Column Vocabulary Association (CVA): Semantic Interpretation of Dataless Tables

Margherita Martorana*, Xueli Pan, Benno Kruit, Tobias Kuhn and Jacco van Ossenbruggen

Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1105, Amsterdam, The Netherlands

Abstract

Traditional Semantic Table Interpretation (STI) methods rely primarily on the underlying table data to create semantic annotations. This year's SemTab challenge introduced the "Metadata to KG" track, which focuses on performing STI by using only metadata information, without access to the underlying data. In response, we introduce a new term: Column Vocabulary Association (CVA). This term refers to the task of semantic annotation of column headers solely based on metadata information. This study evaluates several methods for the CVA task, including a Large Language Models (LLMs) approach combined with Retrieval Augmented Generation (RAG), using three commercial GPT models and four open-source models, along with temperature setting variations. We also evaluate a traditional similarity approach using SentenceBERT. Our experiments operate in a zero-shot setting, without fine-tuning or examples for the LLMs, to maintain a generalized and domain-agnostic application.

Initial findings indicate that LLMs generally perform well at temperatures below 1.0, achieving an accuracy of 100% on the challenge test set. Traditional methods outperforms several LLMs, instead, when metadata and glossary are closely related. However, interim results on the full data set show that our approaches reach an accuracy of 70%, suggesting possible discrepancies in test representativeness, though further investigation is needed.

Keywords

Large Language Models, Metadata Enrichment, Retrieval Augmented Generation, Semantic Table Interpretation, Semantic Web

1. Introduction

Tabular data is the most common format used for data storage and sharing [1]. However, tabular data often lacks semantic annotations and can contain inaccurate or missing information. Semantic Table Interpretation (STI) aims to find semantic annotations for table cells and columns, as well as column relationships, using existing Knowledge Graphs (KGs). Semantic annotations are particularly important when used to enrich and augment metadata. In fact, several studies [2, 3, 4] have shown that high-quality metadata supports data Findability, Accessibility, Interoperability and Reusability (FAIR Guiding Principles) [5]. Rich metadata plays a critical role

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.

m.martorana@vu.nl (M. Martorana); x.pan2@vu.nl (X. Pan); b.b.kruit@vu.nl (B. Kruit); t.kuhn@vu.nl (T. Kuhn); j.r.van.ossenbruggen@vu.nl (J. v. Ossenbruggen)

0000-0001-8004-0464 (M. Martorana); 0000-0002-3736-7047 (X. Pan); 0000-0002-0228-4823 (B. Kruit); 0000-0002-1267-0234 (T. Kuhn); 0000-0002-7748-4715 (J. v. Ossenbruggen)

when dealing with confidential data, as the underlying data is generally not open and freely accessible. Enhancing the FAIRness for this type of data has gained more attention in recent years, and in previous work we have shown that high-quality and rich metadata improves the discovery and reuse of these resources [6].

Nevertheless, the automatic enrichment of metadata - when only the metadata is available - presents a significant challenge. In such cases, much of the contextual information is lacking, and the underlying data cannot be used to identify the most suitable annotations. Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) [7] can offer promising approaches to address these challenges. LLMs' training data can be leveraged as background knowledge, while RAG systems can further integrate external resources - such as knowledge graphs, controlled vocabularies, and glossaries - to extend the LLMs' adaptability across various domains. Also, tuning the LLM's temperature parameter allows for adjusting the model's creativity or determinism, which could be used to balance between precision and flexibility when identifying relevant annotations.

In this year's SemTab challenge, participants in the "Metadata to KG" track aim to annotate tables using only table metadata (e.g. column and table names) without accessing the underlying data. This approach tests the ability to enrich metadata effectively under similar conditions imposed by restricted access data. To guide our investigation we have formulated the following research questions:

- How do traditional semantic similarity methods compare to newer methods using Large Language Models (LLMs) in the semantic annotation of table metadata when the underlying data is not available?
- How does the temperature setting of LLMs impact their performance in this task?
- How do different combinations of metadata information in traditional methods affect their performance?
- How does the nature of the input data and glossary influence the results?

In the pages that follow, we further describe the importance of metadata, especially in settings where the underlying data is not available. We also introduce the term "Column Vocabulary Association", before discussing our main methodology and results.

2. Background

In the next section, we present the key background concepts relevant to this research. First, we introduce the concept of "Dataless Tables", referring to tables where the data is confidential and cannot be accessed. Next, we provide an overview of LLMs and RAG. Finally, we introduce the term "Column-Vocabulary Association", which relates directly to the specific task set by this year's SemTab Challenge.

2.1. Dataless tables

There has been a recent rise in solutions for sharing confidential or restricted-access data. For example, multiple online Open Government Data (OGD) portals, such as the Central Bureau

for Statistics Netherlands (CBS)[1], the U.S. Government's Open Data[2], and Canada's Open Government Portal[3], have been developed to enhance innovation and research by allowing users to explore population data. These portals typically provide aggregated statistics, giving the general public and researchers access to data for use in fields such as journalism, software development, and research [8]. However, much of the population data remains inaccessible due to confidentiality concerns, including sensitive data like patient records, individual-level statistics, and other data containing Personally Identifiable Information (PII).

Various solutions have been proposed to facilitate the reuse of restricted-access data. For instance, the Personal Health Train [9] enables users to send their algorithms to where the data is stored, allowing analysis without needing direct access to the data. However, users still need to know that the data exists and its structural details. Detailed metadata descriptions play a crucial role in addressing this challenge. In previous work, we introduced the DataSet Variable Ontology (DSV) [10], a metadata schema designed to capture information at both the dataset and variable levels. This demonstrated that high-quality metadata can enable the discovery of restricted access data by annotating non-confidential information, such as column descriptions, dataset structure, and summary statistics.

In this context, we introduce the concept of "Dataless Tables". These tables allow for the description of structural elements, summary statistics, and metadata, like column descriptions, while keeping the actual data confidential and inaccessible. Although the raw data is unavailable, such tables retain important features of the dataset, which can be annotated using frameworks like DSV, making them valuable for data discovery and analysis under restricted conditions. We refer to them as "dataless" because, while the data itself is not directly available, the tables still carry structural and contextual information useful for various applications.

2.2. LLMs and RAG

Large Language Models (LLMs) have brought significant advances in the field of Natural Language Processing (NLP). LLMs are trained on a vast amount of data, which allows them to handle a wide range of tasks, including those they weren't explicitly trained for [11, 12]. Studies have shown that ChatGPT has outperformed crowd-workers in tweet classification [13], and the open-source model SOTAB in Column Property Annotation tasks [14]. Additionally, it has been shown that incorporating semantic technologies and Knowledge Graphs (KGs) can further enhance the accuracy of text classification [15]. However, the LLM's training data often lacks real-time updates, resulting in outdated or incomplete information [16], which can lead to factual inaccuracies or irrelevant content, a phenomenon commonly referred to as "hallucinations" [17, 18, 19, 20]. Furthermore, LLMs have limited domain-specific expertise, which can impact their reliability in specialized tasks [21, 22].

To address these challenges, Retrieval-Augmented Generation (RAG) [7] systems have emerged as a promising solution to these challenges [23, 24, 25, 26]. By combining the generative capabilities of LLMs with external, high-quality information sources, RAG systems improve the accuracy and reliability of information retrieval. These systems have been shown to enhance

[1] https://www.cbs.nl
[2] https://data.gov
[3] https://search.open.canada.ca/data/

performance in various applications, including code generation [27] and both domain-agnostic and domain-specific question answering [28, 29, 30].

2.3. Column Vocabulary Association (CVA)

In the domain of Semantic Table Interpretation (STI), there are some well known challenges, including the Column Type Annotation (CTA), the Column Entity Annotation (CEA), and the Column Property Annotation (CPA) tasks. The **CTA** task involves identifying the semantic type (e.g. dates or geographical locations) of each column in the table. The **CEA** task, instead, involves linking each cell to an entity in a knowledge graph: for example, the cell containing the string "New York" to be linked to the WikiData entity for New York City (Q60). **CPA** requires the identification of relationships between columns of a table: for example, recognising that the columns with headers "Mayor's Name" and "City" are related to each other by the property eg:isMayorOf.

In this work we introduce a new term: the "Column Vocabulary Association" (CVA) task. This task differs significantly from the previous ones because it does not rely on any information from the underlying data within the table. Instead, it aims to associate column headers with entries in controlled vocabularies purely based on semantic similarities. The distinction between the word *association* and *annotation* is also important in this context. Annotation typically refers to the labeling of data with tags or categories. In contrast, with the term association we refer more on the conceptual linkage between the textual information in a column header, and an external knowledge repository. This approach emphasizes understanding and leveraging the semantic meaning of the column headers themselves, without using any underlying data. By focusing on semantic similarities, we aim to create a method for interpreting and integrating restricted access datasets, to facilitate metadata enrichment and data discovery.

3. SemTab Challenge

Since its start in 2019, the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab) has focused on benchmarking systems and approaches that support and enhance Semantic Table Interpretation (STI). The SemTab challenge typically consists of two main tracks: the *"Accuracy Track"*, where participants annotate tables with tasks like CTA and CEA, and the *"Dataset Track"*, focused on submitting new datasets and benchmarks across different domains. This year, SemTab introduced the new "Metadata to KG" track, where participants are asked to map table metadata to KGs without having access to the underlying data. This presents a unique challenge due to the limited available context, making traditional STI methods less applicable, as they typically rely on actual data for annotation. To better define this metadata-only task, we introduced the term **Column Vocabulary Association (CVA)**. As previously described in section 2.3, CVA involves annotating columns using only KGs and table metadata, without accessing the underlying data. This approach is especially relevant in scenarios where the data is confidential and inaccessible.

In a prior research [31], we investigated a similar concept, using a Large-Context Window approach to enhance the model's performance, instead of a RAG system. We primarily focused on whether we could use, in principle, LLMs to annotate column headers with an external

vocabulary without any use of the underlying data, and whether contextual and hierarchical information had any impact in performance. In that work we proposed three metrics for evaluation. The first - "LLM Internal Consistency" - assesses how consistently LLMs perform the annotation. The second metric - "Inter-LLMs Alignment" - evaluates how multiple LLMs annotate the same column headers. The third metric - "Human-Computer Agreement" - involves comparing LLM-generated annotations against human annotations. Importantly, we did not treat human annotations as the groundtruth or gold standard. Instead, we focus on comparing the variance among human annotations with those produced my the LLMs. In this approach, we highlight that humans can also disagree on finding the most suitable annotations, in the same way LLMs can. Also, all our metrics involved performing the same annotations task multiple times, allowing us to measure the consistency of the LLM's outputs over several iterations rather than just once. The goal of this year SemTab challenge and the "Metadata to KG" track align with our previous research, as creating annotations is often challenging, requiring domain expertise and difficult to automate.

3.1. Metadata to KG - Round 1

Round 1 of the "Metadata to KG" track required participants to map a set of table metadata to DBpedia properties. Participants were provided with tables metadata and DBpedia properties files in both JSONL and OWL formats, all of which were accessible in the following GitHub repository[4]. The tables metadata file included information about 141 columns derived from different tables. For each column, the provided information included the column ID, column label, table ID, table name, and a list of the other column labels within the same table. The DBpedia properties file contained 2,881 properties. For each DBpedia property, the information included the property ID (the actual URI of the property in DBpedia), the property label, and the description. Below, we report examples of a table metadata entry (Listing 1) and of DBpedia property (Listing 2).

```
{
    "id": "58891288_0_1117541047012405958_Director(s)",
    "label": "Director(s)",
    "table_id": "58891288_0_1117541047012405958",
    "table_name": "Film",
    "table_columns": ["Rank", "Title", "Year", "Director(s)", "Overall Rank"]
}
```

Listing 1: Example of table metadata

```
{
    "id": "http://dbpedia.org/ontology/director",
    "label": "film director",
    "desc": "A film director is a person who directs the making of a film."
}
```

Listing 2: Example of DBpedia property

[4]https://github.com/sem-tab-challenge/2024/blob/main/data/metadata2kg/round1/README.md

Additionally, the SemTab organizers supplied a sample table metadata file and a sample ground truth (which contains only 9 metadata entries and their corresponding annotations), along with a Python evaluation script. The objective was to develop approaches for mapping each table metadata with up to 5 DBpedia properties for each column, based on semantic similarities and relevance, and then rank the mappings from the most to least accurate. The evaluation script assessed the mappings by calculating two metrics: hit@1, which checks if the first mapping is correct, and hit@5, which checks if the correct mapping is within the top five. Participants tested their systems on sample metadata (containing only 9 columns) and submitted their complete results to the track organisers for evaluation against the overall ground truth.

3.2. Metadata to KG - Round 2

Round 2 of the "Metadata to KG" track introduced a level of complexity by using a collection of custom vocabularies for the mapping task. Participants were again provided with tables metadata and custom vocabularies files in JSONL and OWL formats, accessible in the following GitHub repository[5]. In this round, the tables metadata file contained 1181 entries (one entry corresponds to one column) from various datasets, with each column having the same information as in round 1 included: column ID, column label, table ID, table name, and the other columns labels. The custom vocabularies file consisted of 1192 entries, where each entry had, again, the same information as in round 1: ID (in this case not a URI, but a minted ID), label and description. The tables metadata included a very diverse set of topics: including but not limited to: COVID-19 clinical trials, Indian movies ratings and Saudia Arabia stock exchange data. As in Round 1, the SemTab organizers supplied a sample table metadata file and a sample ground truth for validation and testing purposes (containing 11 metadata entries and their corresponding annotations), with the Python evaluation script. As before, the objective was to map each table column metadata to up to 5 relevant custom vocabulary terms based on semantic similarities and relevance, and rank these mappings by accuracy. The evaluation script again used hit@1 and hit@5 metrics to assess the quality of the mappings.

4. Methods

Here we outline our methodology, which considers the CVA task as a textual information retrieval challenge. Given that most of the table metadata and glossary information are described in text, the goal is to retrieve the most similar glossary entries from the glossary files based on the table metadata only.

4.1. Implementation details

Our approach combines two main methods: one leveraging LLMs (both open-source and commercial) and the other utilizing a traditional semantic similarity technique. We tested several LLMs, including three GPT models (`gpt-3.5-turbo-0125`, `gpt-4o` and `gpt-4-turbo`), two LLama models (`llama3-70b` and `llama3-8b`), a Gemma model (`gemma-7b`) and a Mixtral

[5] https://github.com/sem-tab-challenge/2024/blob/main/data/metadata2kg/round2/README.md

model (`mixtral-8x7b`). To explore how LLMs' creativity impacts performance on our task, we experimented with varying temperature settings (0.5, 0.75, 1.0, 1.25, and 1.5) for each model. In general, temperature is a key hyperparameter in LLMs that controls the randomness of generated outputs, with higher temperature typically producing more and diverse responses, and lower temperatures more deterministic ones. In this study, we only changed the temperature setting to focus on one parameter at the time and explore wheter the creativity or determinism of the LLMs had an impact on task performance. Other parameters, such as top-k and top-p sampling, which control token selection probability and distribution, were kept at default values. While these additional parameters may also affect performance, investigating them was out of scope of this paper and could be explored in future research. Further, we utilized RAG systems to enhance the LLMs' performance. Specifically, we employed two RAG systems. The first is OpenAI's RAG, which uses the OpenAI Assistant API with a built-in file search tool to process and retrieve relevant data. The second is an open-source setup with LlamaIndex, ChromaDB, and Groq, where LlamaIndex integrates the data, ChromaDB serves as a vector database, and Groq accelerates inference with custom hardware.

For our semantic similarity method, we implemented SentenceBERT [32] using the model `all-MiniLM-L6-v2` to calculate cosine similarity scores between sentence embeddings. Here, we embedded both the metadata and glossary information, computing the cosine similarity for all possible pairs and selecting the glossary term with the highest similarity score for each metadata entry. We explored whether the choice of information embedded (e.g. table name, column headers, glossary term description) impacted similarity results, experimenting with various configurations to determine the most effective approach.

All experiments with LLMs were conducted in a zero-shot setting, meaning the models received no fine-tuning and were provided no examples through prompts or assistant instructions. This approach is a key feature of our methodology. We chose this strategy because we aim to develop a method that is domain-agnostic. Fine-tuning models with specific examples from particular mappings and vocabularies could bias the reported accuracy towards that specific domain. This is particularly relevant in scenarios where a large amount of datasets requires annotation, or where we are dealing with a new vocabulary without ground truth or suitable examples to provide. While a few-shot approach could also be a reasonable solution in cases where domain experts provide examples or ground truth, our focus in this work is on the more naive zero-shot setting. This allows us to propose a method that can be applied more in general settings, regardless of the data domain or vocabulary.

4.2. CVA with LLMs

Prompt Engineering

Through trial and error, we developed effective prompts, both user queries and assistant instructions. We found that repeating some information from the assistant instructions within the prompt resulted in more precise results by ensuring the models only used the data we provided, thus minimizing hallucinations. Both the prompt and instructions specified to return the 5 most similar glossary entries for each metadata. Below, we show the instructions given to the assistants and the query template for the user prompt used in Round 1 of the SemTab Challenge. The

instructions and prompts for Round 2, which are quite similar, can be found on the GitHub page [6].

```
ASSISTANT INSTRUCTIONS
Your task is to match column metadata to DBpedia properties.
The full set of DBpedia properties will be provided in the vector.
Columns metadata, instead, will be provided by the user and it will contain the following
information: column ID, column label, table ID, table name and the labels of the other
columns within that table. The matching between the column and the DBpedia properties is
to be made based on the semantic similarities between the metadata (i.e. what the column
express), and DBpedia properties.
You can add multiple properties, but no more 5.
Return the results in the following format:
'colID': '00000_0_0000_XXX', 'propID': ['http://dbpedia.org/ontology/PROPERTY_ID', ...,
'http://dbpedia.org/ontology/PROPERTY_ID'].
Sort the matched DBpedia in descending order of relevance, starting with the most relevant.
Choose ONLY from the DBpedia properties.
Return ONLY the results, no other text.
Return results for each and every single column metadata.


QUERY TEMPLATE
Based on the instruction given to you, find the most relevant DBpedia property, for each of
the following metadata in json format:
{input_metadata}
Each json element is an independent column metadata. The metadata do not have any relationship,
so the matching with the DBpedia properties should only be based on the information provided
within its own metadata.
You can add multiple properties, but no more 5.
Return the results in the following format:
'colID': '00000_0_0000_XXX', 'propID': ['http://dbpedia.org/ontology/PROPERTY_ID', ...,
'http://dbpedia.org/ontology/PROPERTY_ID'].
Sort the matched DBpedia in descending order of relevance, starting with the most relevant.
Choose ONLY from the DBpedia properties provided in the vector.
Return ONLY the results, no other text.
Return results for each and every single column metadata.
```

Model-temperature selection

We ran the queries three times for each LLM and temperature combination, then evaluated the preliminary performance using an evaluation script and groundtruth provided by the organisers. Based on these results, we selected the best-performing LLM-temperature combination to compute results on the full dataset.

CVA on full metadata set

In the first round, we built a vector representation of the complete glossary JSON file containing the DBpedia properties through the RAG system. We then processed the metadata entries in batches of 25 when using the OpenAI API. For the open-source LLMs, instead, each metadata entry was added individually. While processing each metadata individually may lead to better performance, we implemented this batching strategy primarily to reduce costs associated with running some of the more expensive OpenAI models (i.e. GPT-4o and GPT-4-turbo). In the second round, given the larger size of the glossary, we split it into smaller, topic-based

[6] https://github.com/sem-tab-challenge/2024/blob/main/data/metadata2kg/round1/README.md

glossaries. We created 75 smaller glossary files and divided the full metadata set into 75 corresponding files. Each metadata file was then processed one at a time against the vector containing the 75 glossary files.

4.3. Semantic similarity using SentenceBERT

Our second method involved computing the semantic similarity between table metadata and glossary entries using SentenceBERT [32]. First, we generated a vector representation for each metadata and glossary entry. Next, we calculated the cosine similarity between the embedding of each table metadata and the glossary entries to identify the top 5 glossary entries with the highest cosine similarity scores.

The initial steps of this method posed two challenges: first, selecting the appropriate table metadata and glossary information for vector generation, and second, determining the optimal approach for vectorizing the textual content. Specifically, we needed to decide whether to concatenate all textual elements before vectorization or to vectorize each component separately and then combine the vectors to create a final embedding. To address these questions, we experimented with various combinations of textual information, namely metadata column headers and table names, and glossary term labels and descriptions. We computed the cosine similarities for each combination and evaluated the results against the ground truth using the evaluation script provided by the organizers. Based on these analyses, we selected the most effective combinations for application across the complete metadata set.

4.4. Evaluation

The evaluation was conducted using a script provided by the track organizers. This script computed the accuracy of the generated mappings for a sample metadata file and a sample ground truth. It calculated two metrics: hit@1 and hit@5. To reiterate, users are supposed to generate the 5 most relevant mapping between the table metadata and the glossary, sorted from the most relevant. Hit@1 checks if the first mapping (thus the one considered to be the most relevant) is correct, while hit@5 checks if the correct mapping is among the top five results. Participants were then asked to generate the mappings for the entire table metadata file and submit them to the organizers. The organizers can then run the evaluation script again using the complete ground truth, which has not yet been shared with the participants.

5. Results

In the following sections, we present the preliminary challenge's results. These results show the accuracy scores obtained from the evaluation script on the sample metadata and the sample groundtruth provided. At this point, we are not aware on how our methods performed for the full set of metadata file, as the complete groundtruth has not yet been provided to participants.

5.1. CVA with LLMs

Here we present the results from our initial analysis using different LLMs and temperature settings. We employed three models from OpenAI and four open-source models, testing them at five different temperatures, as detailed in table 1. Table 1 shows the average accuracy results for each model-temperature combination, evaluated using the evaluation script with the sample metadata and sample ground truth. Each query was run three times per model-temperature combination, and accuracy results were then averages. The numbers in bold correspond to the best-performing model-temperature combinations. `gpt-4o` outperformed other models in both Rounds 1 and 2, specifically at temperatures 0.5, 0.75, and 1.0. We observed that the LLMs did not perform very well in Round 2. In the discussion section6 we explore possible reasons for this outcome. Based on these preliminary results, for Round 1, we used `gpt-4o` at temperatures 0.5, 0.75, and 1.0 on the full metadata file for final analysis. For Round 2, we used `gpt-4o` only at temperatures 0.5 and 0.75.

Table 1
Results of different models for sample data in Round 1 and 2. The cells with the "X" refers to tries where the LLM could not compute the task, and either the API was not returning any results over a long period of time or, in the case of `gemma-7b` the model was returning "failure" message. In bold, instead, we show the best performing results. In the table h1 and h5 refers to Hit@1 and Hit@5 metrics from the evaluation script.

LLM Models	Round 1									
	0.5		0.75		1.0		1.25		1.5	
	h1	h5	h1	h5	h1	h5	h1	h5	h1	h5
gpt-3.5-turbo-0125	0.42	0.42	0.33	0.42	0.47	0.53	0.41	0.41	0.36	0.36
gpt-4o	**0.59**	**0.89**	**0.62**	**0.81**	**0.64**	**0.75**	0.47	0.67	0.11	0.22
gpt-4-turbo	0.58	0.58	0.56	0.56	0.61	0.61	0.59	0.59	0.36	0.36
llama3-8b	0.22	0.44	0.22	0.44	0.22	0.44	0.33	0.44	0.33	0.33
llama3-70b	0.11	0.11	0.11	0.22	0.11	0.22	0.11	0.22	0.11	0.11
gemma-7b	0.37	0.4	0.44	0.52	0.33	0.47	0.48	0.61	0.37	0.53
mixtral-8x7b	0.33	0.33	0.44	0.56	0.33	0.44	0.44	0.44	0.44	0.44
	Round 2									
	0.5		0.75		1.0		1.25		1.5	
	h1	h5	h1	h5	h1	h5	h1	h5	h1	h5
gpt-3.5-turbo-0125	0.67	0.67	0.3	0.33	0.63	0.67	0.67	0.67	X	X
gpt-4o	**0.73**	**1**	**0.45**	**1**	0.45	0.58	0.54	0.64	X	X
gpt-4-turbo	0	0	X	X	X	X	X	X	X	X
llama3-8b	0	0	0	0	0	0	0	0	0	0
llama3-70b	0	0	0	0	0	0	0	0	0	0
gemma-7b	X	X	X	X	X	X	X	X	X	X
mixtral-8x7b	0	0	0	0	0	0	0	0	0	0

5.2. CVA with SentenceBERT

Below we show the results from our initial analysis with SentenceBERT. Table 2 includes the possible combinations of information from the table metadata and the glossary, and the accuracy results for both Round 1 and 2, which we obtained by running the evaluation script against the ground truth for the sample metadata file. We used these results to find the best performing combinations, which were then applied to the full metadata file.

In Round 1, we did not have a single combination that performed best for both hit@1 and hit@5. The best hit@1 (0.56) is obtained when we use the column label and/or table name to represent the table metadata embedding, and use the property label for the DBpedia property embedding. The best hit@5 (0.67) is obtained when we use the sum of the vectors of the column label and the table name as the table metadata embedding, and use the vector of the property label as the DBpedia property embedding.

For Round 2, the best hit@1 and hit@5 are both obtained when we use the sum of the vectors of the column label and the table name as the table metadata embedding, and encode the vocabulary description as the vocabulary embedding. Based on this results, we did perform SentenceBERT on the full data for Round 1. For Round 2, instead, we sent the results from SentenceBERT for final analysis using the setting with the sum of the vectors of the column label and the table name as the table metadata embeddings.

Table 2
Results of various embedding combinations for sample data in Round 1 and 2 are presented, with best performing results in bold. The table includes Hit@1 (h1) and Hit@5 (h5) metrics from evaluation script.

Metadata Embeddings	Glossary Embeddings	Round 1		Round 2	
		h1	h5	h1	h5
encode(label)	encode(label)	**0.56**	0.56	0.36	0.55
encode(label)	encode(lable + desc)	0.22	0.56	0.45	0.82
encode(label + table_name)	encode(label)	**0.56**	0.56	0.09	0.27
encode(label + table_name)	encode(lable + desc)	0.33	0.44	0.64	0.73
encode(label)	encode(desc)	0.11	0.33	0.45	0.82
encode(label + table_name)	encode(desc)	0.22	0.44	0.64	0.73
encode(label) + encode(table_name)	encode(desc)	0	0.33	**0.64**	**0.91**
encode(label) + encode(table_name)	encode(desc) + encode(label)	0.22	0.44	0.55	0.91
encode(label) + encode(table_name)	encode(label)	0.44	**0.67**	0.27	0.45

5.3. Interim results on full metadata set

The results presented above are based on the test metadata provided by the track organizers. These results reflect the performance of our methods solely on the test metadata, which we used for selecting the most effective approaches for application to the complete metadata set. After, the full set of results was submitted and evaluate by the organizers against the complete groundtruth set (which is not accessible to participants). The organizers subsequently provided us with interim results, shown in Table 3.

In Round 1, the model that performed best on the test data was GPT-4o, with temperature settings of 0.5, 0.75, and 1.0. This model achieved a top accuracy of 0.89 on the test data, as illustrated in Table 1. However, when applied to the full dataset, our performance dropped to 70%. In Round 2, the best approaches based on the test data were the traditional semantic similarity method (SentenceBERT) and GPT-4o with temperatures of 0.5 and 0.75. The GPT models reached an accuracy of 100% on the test data, while SentenceBERT achieved 91%. Nevertheless, the performance on the full dataset was lower, with SentenceBERT reaching only 68% accuracy and GPT models achieving up to 52%.

While we have received these interim results, the complete results and ground truth data have not yet been provided to participants. One potential reason for the lower performance on the full dataset could be the quality of the test data, which may include mapping that are "impossible" or particularly challenging even for humans, especially since there is no human performance baseline to compare against. Additionally, the discrepancy between the interim results and test performance may suggest that the test data is not fully representative of the entire dataset, which could further explain the variations in performance.

Table 3
Interim results on full data set in Round 1 and 2 are presented, with the best performing results in bold. ∗ The embedding settings for this approach included the following: metadata embeddings (computed as encode(column label) + encode(table name)) and glossary term embeddings (computed as encode(term description)).

Approach	Specifics	Round 1	
		h1	h5
gpt-4o	0.50 (temp)	0.50	0.50
gpt-4o	0.50 (temp)	0.49	0.49
gpt-4o	0.75 (temp)	**0.55**	**0.70**
gpt-4o	1.00 (temp)	0.53	0.56
		Round 2	
		h1	h5
gpt-4o	0.50 (temp)	**0.49**	0.52
gpt-4o	0.75 (temp)	0.45	0.49
SentenceBert	∗	0.37	**0.68**

6. Conclusion

Our analysis provides several key observations. Firstly, we found that repeating phrases in both the assistant instructions and user prompts improved the LLM's adherence and reduced hallucinations by preventing irrelevant entries. Also, the effectiveness of traditional semantic similarity methods can outperform that of more advanced techniques like LLMs and RAG, depending on the nature of the glossary and metadata. In fact, in Round 1, there was no

clear link between the table metadata and the glossary, which consisted of DBpedia properties. However, we saw a strong connection in Round 2, likely because both the metadata and glossary were developed by the same institution. We think that these variations impacted the performance of the methods proposed in this study. LLMs, particularly `gpt-4o`, performed much better in Round 1, where leveraging the LLM's background knowledge was crucial for identifying the most relevant mappings. In round 2, instead, the more traditional semantic similarity method using SentenceBERT was sufficient and sometimes even outperformed LLMs. This improvement is attributed to the high degree of semantic similarity between the metadata and the glossary, as both likely originated from the same institution and were intentionally designed to be compatible and consistent. Regarding temperature settings for LLMs, we observed that lower temperatures generally led to better performance. Higher temperatures, particularly above 1.25, often resulted in no outputs or errors, especially with models like `gpt-4-turbo` and `gemma-7b`. This suggests that lower temperatures may be more effective for tasks requiring precise mapping.

It is also essential to note that the interim results from the full dataset show lower performance compared to results derived from the test data alone. Currently, we don't have yet the complete ground truth and results for further examination, as these remain with the task organizers. Our observations suggest that the test ground truth and data provided may not be suitable to assess our approaches for generalization to the full dataset. Only a small number of test results were provided in both rounds (9 ground truth metadata entries in Round 1 and 11 in Round 2), and we are uncertain about how the ground truth was generated (whether automatically, through extraction from resources, or manually). The evaluation's aim might need to focus more on comparing performance against human annotations, as parts of metadata annotation using external vocabularies can be challenging even for humans. This difficulty can happen if the vocabulary lacks suitable terms to describe the metadata or if a term is too complicated to explain clearly. In this context, trying to limit hallucinations might not be the best approach. In previous research [31], where we labeled column headers with terms from a specific vocabulary (CESSDA)[7] using also a zero-shot LLMs approach, we showed that hallucinations can indicate problems with the vocabulary used: columns about mental health were labeled "Mental Health", even though this topic wasn't included in CESSDA. These findings could help in creating better and more comprehensive vocabularies. Another consideration is that we utilized the same prompt for all LLMs. While we found that commercial models performed generally better, open-source LLMs might require different prompts than commercial ones, and further investigation is needed to address this.

In conclusion, our study examined methods for mapping column headers to glossaries in a zero-shot setting. This approach allows for broader evaluation across domains. We introduced the concept of "Column Vocabulary Association" (CVA), distinguishing it from other Semantic Table Interpretation (STI) tasks. Additionally, we defined the term of "Dataless Tables", referring to tables where the structure and metadata are available but the actual data is confidential and inaccessible. Our findings suggest that LLMs have good performance when there is no clear connection between metadata and glossaries (as seen in Round 1), while traditional methods perform better in cases where there is a strong relationship between them (Round 2).

[7] https://vocabularies.cessda.eu/vocabulary/TopicClassification

Acknowledgments

We acknowledge that ChatGPT was utilized to generate and debug part of the python and latex code utilised in this work. This work is funded by the Netherlands Organisation of Scientific Research (NWO), ODISSEI Roadmap project: 184.035.014.

Author M.M. led the research by developing the main ideas, conceptualising the research design, participating in the programming tasks, reviewing the literature and drafting most of the manuscript. Author X.P. provided most of the technical expertise, particularly in programming and APIs integration to perform our analysis, and has helped in reviewing the manuscript. Authors B.K., T.K and J.v.O. provided guidance through their supervisory roles and gave feedback to improve the overall study.

References

[1] R. Shwartz-Ziv, A. Armon, Tabular data: Deep learning is not all you need, Information Fusion 81 (2022) 84–90.

[2] M. Boeckhout, G. A. Zielhuis, A. L. Bredenoord, The fair guiding principles for data stewardship: fair enough?, European journal of human genetics 26 (2018) 931–936.

[3] B. Mons, Data stewardship for open science: Implementing FAIR principles, Chapman and Hall/CRC, 2018.

[4] A.-L. Lamprecht, L. Garcia, M. Kuzak, C. Martinez, R. Arcila, E. Martin Del Pico, V. Dominguez Del Angel, S. Van De Sandt, J. Ison, P. A. Martinez, et al., Towards fair principles for research software, Data Science 3 (2020) 37–59.

[5] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016) 1–9.

[6] M. Martorana, T. Kuhn, R. Siebes, J. van Ossenbruggen, Aligning restricted access data with fair: a systematic review, PeerJ Computer Science 8 (2022) e1038.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[8] G. M. Begany, E. G. Martin, X. J. Yuan, Open government data portals: Predictors of site engagement among early users of health data ny, Government Information Quarterly 38 (2021) 101614.

[9] T. M. Deist, F. J. Dankers, P. Ojha, M. S. Marshall, T. Janssen, C. Faivre-Finn, C. Masciocchi, V. Valentini, J. Wang, J. Chen, et al., Distributed learning on 20 000+ lung cancer patients–the personal health train, Radiotherapy and Oncology 144 (2020) 189–200.

[10] M. Martorana, T. Kuhn, R. Siebes, J. Van Ossenbruggen, Advancing data sharing and reusability for restricted access data on the web: introducing the dataset-variable ontology, in: Proceedings of the 12th Knowledge Capture Conference 2023, 2023, pp. 83–91.

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
[12] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, Advances in neural information processing systems 35 (2022) 22199–22213.
[13] F. Gilardi, M. Alizadeh, M. Kubli, Chatgpt outperforms crowd workers for text-annotation tasks, Proceedings of the National Academy of Sciences 120 (2023) e2305016120.
[14] K. Korini, C. Bizer, Column property annotation using large language models, in: Proceedings of the ESWC Conference, 2024.
[15] Y. Shi, H. Ma, W. Zhong, Q. Tan, G. Mai, X. Li, T. Liu, J. Huang, Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs, in: 2023 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2023, pp. 515–520.
[16] H. He, H. Zhang, D. Roth, Rethinking with retrieval: Faithful large language model inference, arXiv preprint arXiv:2301.00303 (2022).
[17] G. Marcus, The next decade in ai: four steps towards robust artificial intelligence, arXiv preprint arXiv:2002.06177 (2020).
[18] M. Cao, Y. Dong, J. Wu, J. C. K. Cheung, Factual error correction for abstractive summarization models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6251–6258.
[19] V. Raunak, A. Menezes, M. Junczys-Dowmunt, The curious case of hallucinations in neural machine translation, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 1172–1183. URL: https://aclanthology.org/2021.naacl-main.92. doi:10.18653/v1/2021.naacl-main.92.
[20] 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) 1–38.
[21] X. Li, S. Chan, X. Zhu, Y. Pei, Z. Ma, X. Liu, S. Shah, Are ChatGPT and GPT-4 general-purpose solvers for financial text analytics? a study on several typical tasks, in: M. Wang, I. Zitouni (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, Association for Computational Linguistics, Singapore, 2023, pp. 408–422. URL: https://aclanthology.org/2023.emnlp-industry.39. doi:10.18653/v1/2023.emnlp-industry.39.
[22] X. Shen, Z. Chen, M. Backes, Y. Zhang, In chatgpt we trust? measuring and characterizing the reliability of chatgpt, arXiv preprint arXiv:2304.08979 (2023).
[23] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 (2023).
[24] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, G. Neubig, Active retrieval augmented generation, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 7969–7992.
[25] K. Guu, K. Lee, Z. Tung, P. Pasupat, M. Chang, Retrieval augmented language model pre-training, in: International conference on machine learning, PMLR, 2020, pp. 3929–3938.
[26] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, et al., Improving language models

by retrieving from trillions of tokens, in: International conference on machine learning, PMLR, 2022, pp. 2206–2240.
[27] S. Zhou, U. Alon, F. F. Xu, Z. Jiang, G. Neubig, Docprompting: Generating code by retrieving the docs, in: The Eleventh International Conference on Learning Representations, ????
[28] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, et al., Check your facts and try again: Improving large language models with external knowledge and automated feedback, arXiv preprint arXiv:2302.12813 (2023).
[29] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 874–880. URL: https://aclanthology.org/2021.eacl-main.74. doi:10.18653/v1/2021.eacl-main.74.
[30] D. Li, A. S. Rawat, M. Zaheer, X. Wang, M. Lukasik, A. Veit, F. Yu, S. Kumar, Large language models with controllable working memory, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 1774–1793.
[31] M. Martorana, T. Kuhn, L. Stork, J. van Ossenbruggen, Zero-shot topic classification of column headers: Leveraging llms for metadata enrichment, in: Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI, IOS Press, 2024, pp. 52–66.
[32] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).