Leveraging GPT Models For Semantic Table Annotation

Jean Petit Bikim[1,2], Carick Atezong[2], Azanzi Jiomekong[1,2], Allard Oelen[1], Gollam Rabby[3], Jennifer D'Souza[1] and Sören Auer[1,3]

[1] *TIB – Leibniz Information Centre for Science and Technology, Hannover, Germany*

[2] *Department of Computer Science, University of Yaounde I, Yaounde, Cameroon*

[3] *L3S Research Center, Leibniz University Hannover, Hanover, Germany*

Abstract

This paper outlines our contribution to the Accuracy Track and the Semantic Table Interpretation (STI) & Large Language Models (LLMs) track of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab). Our approach involves using LLMs to address the various tasks presented in the challenge. Specifically, we employed zero-shot and few-shot prompting techniques for most of the tasks, which facilitated the LLMs ability to interpret and annotate tabular data with minimal prior training. For the Column Property Annotation (CPA) task, we took a different approach by applying a set of predefined rules, tailored to the structure of each dataset. Our method achieved notable results, with an $f1-score$ exceeding 0.92, demonstrating the effectiveness of LLMs in tackling the SemTab challenge. These results suggest that LLMs hold significant capabilities as a robust solution for semantic table annotation and knowledge graph matching, highlighting their potential to advance the field of semantic web technologies.

Keywords

Tabular Data, Semantic Table Annotation, Semantic Table Interpretation, Knowledge Graph, Large Language Models, SemTab, Prompt Engineering

1. Introduction

Tabular data are currently the most used data for structuring and organizing data on the Web, in companies, etc. The form these data are presented makes it difficult to access [1], and analyses[1]. To solve this problem, SemTab (Semantic Web Challenge on Tabular Data to Knowledge Graph Matching) proposes a challenge consisting of annotating tabular datasets using a knowledge graph (KG). The addition of semantic to tabular data may enhance a large range of applications such as Web Search, Question Answering [2], Knowledge Graph construction and refinement [3, 4], etc. To solve the SemTab challenge, we are proposing an LLM-based approach. To this end, the GPT-3 model was fine-tuned using a prompt engineering technique. Due to the limited time to submit the test data, we were able to participate in the STI & LLMs, Accuracy tracks only. The zero-shot prompting [5] was used for the CTA (Column Type Annotation), RA (Row Annotation) and TD (Table Topic Detection) tasks. The few-shot prompting [5] was used to solve the CEA (Cell Entity Annotation) task in the accuracy track and the only task of the LLM track. To solve the CPA (Column Property Annotation) task, we used a set of rules to identify relevant properties that link two columns of a table. In this paper, we also present how the GPT-3 was fine-tuned (see Section 2.2) and the results with the test data (see Section 3).

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.

jean.bikim@facsciences-uy1.cm (J. P. Bikim); carick.atezong@facsciences-uy1.cm (C. Atezong); fidel.jiomekong@facsciences-uy1.cm (A. Jiomekong); Allard.Oelen@tib.eu (A. Oelen); gollam.rabby@l3s.de (G. Rabby); jennifer.dsouza@tib.eu (J. D'Souza); Auer@tib.eu (S. Auer)

0009-0001-6535-7516 (C. Atezong); 0000-0002-8005-2067 (A. Jiomekong)

[1] https://sem-tab-challenge.github.io/2024/

2. Applying GPT-3 for Semantic Table Annotation

This section details the methodology we employed during the SemTab'24 challenge to address the various tasks set by the organizers. The challenge involved multiple stages, each with distinct objectives requiring customized strategies. In Section 2.1, we present a comprehensive overview of the SemTab'24 challenge, outlining its goals, structure, and key requirements. Following that, Section 2 delves into the specific approach we implemented to tackle the challenge's diverse tasks, including data processing, LLM selection, and performance optimization. Each component of our approach was carefully designed to align with the challenge's demands while maximizing accuracy and efficiency. Overall, our strategy reflects a combination of innovative techniques and established methods, ensuring robust results across all tasks.

2.1. Overview of the Challenge

The SemTab challenge [6], as described by the organizers, focuses on bench-marking datasets and systems for semantic table annotation. The primary goal of this challenge is to assess and improve the capabilities of automated systems in interpreting and annotating structured data, such as tables, by linking them to relevant KGs. The SemTab challenge serves as an important platform for evaluating advancements in semantic technologies and encouraging the development of novel approaches to table annotation. Participants are required to apply their techniques across diverse tasks and datasets, reflecting real-world scenarios. By setting standardized evaluation metrics and promoting reproducible results, the SemTab challenge plays a crucial role in advancing the field of semantic data annotation.

2.1.1. SemTab Challenge Tracks

This year, the SemTab challenge introduced five distinct tracks, each designed to focus on specific aspects of table annotation: the STI & LLMs track, the accuracy track, the dataset track, the metadata-to-KG track, and the IsGold? track. The STI & LLMs track, alongside the accuracy track, includes a series of critical tasks that highlight key table annotation processes, as illustrated in Fig. 1. The main tasks within these tracks are as follows:

- **Column Entity Annotation (CEA):** This task involves linking the elements in a table's cells to their corresponding entities in a KG. For example, in Fig. 1, the entity "Kelso Township" in Table (a) match to the QID "Q6386554" in Wikidata.
- **Column Type Annotation (CTA):** This task requires identifying the most specific semantic type to be assigned to a column in the table. For instance, in Table (a) of Fig. 1, The Wikidata entity type for "Kelso Township" and "Ohio Township" has the QID "Q17201685" (Township of Indiana).
- **Column Property Annotation (CPA):** The objective here is to determine the property within the KG that links two columns in a table. For example, in Table (a) of Fig. 1, the Wikidata property that connects columns col0 and col1 is P2044 (elevation above sea level).
- **Table Topic Detection (TD):** This task focuses on assigning an overarching semantic type to an entire table by identifying its primary subject within the KG. For instance, The Wikidata entity that describes the topic of Table (b) in Fig. 1 has the QID Q16823610 (Blue Christmas).
- **Row Annotation (RA):** In this task, participants must link entire rows in the table to the corresponding entities in the KG. For example, the first row of Table (c) in Fig. 1 has the Wikidata QID "Q26689963".

These tasks, while diverse, collectively assess the robustness and flexibility of participating systems in accurately interpreting and annotating tabular data. Each track is designed to target different challenges faced in real-world applications, ensuring that systems are tested comprehensively across a wide range of scenarios.

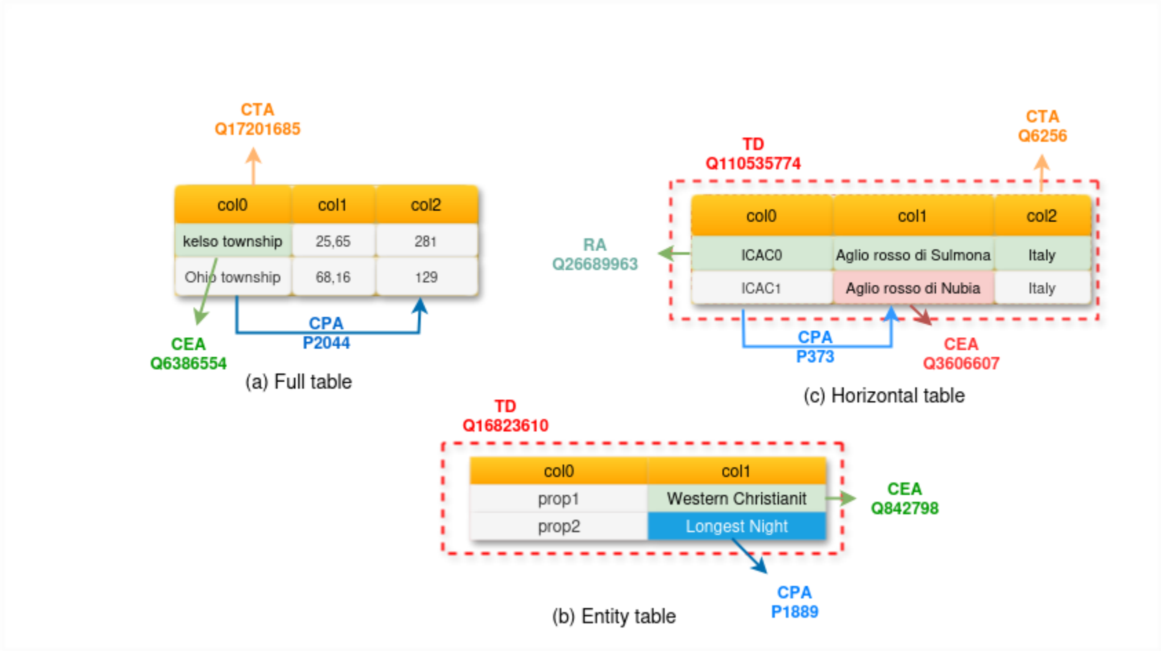


Figure 1: Illustration of the different tasks of the LLMs and accuracy tracks. CEA: Cell Entity Annotation, CTA: Column Type Annotation, CPA: Column Property annotation, TD: table topic detection, RA: Row Annotation.

2.1.2. SemTab Datasets

Our focus on semantic table annotation led us to benchmark various datasets from the SemTab challenges published since 2019[2], allowing us to establish a system that will adapt to different datasets.

Table 1 provides a detailed overview of the datasets we employed for the CEA task. The datasets vary in size, complexity, and domain coverage, offering a comprehensive range of challenges for CEA systems. The datasets tfood [7] (entity, horizontal) and WikidataTableR1 from the 2023 edition, along with Semantic_annotation[3](a dataset automatically constructed from 15,000 entities on Wikidata retrieved through API queries and their descriptions as context), were primarily used before the challenge. They served as the foundation for our various experiments and also enriched our training data during the actual challenge phase. Additionally, the training data contained in tbiomed [8], tbiodiv [9] and SuperSemtab24 [10] were used to further enhance our models.

For the CTA, CPA, RA, and TD tasks, we used the datasets proposed by the challenge organizers for the 2024 edition. These datasets cover a diverse range of domains and tasks, which allows for a more comprehensive evaluation of different semantic table annotation techniques. Table 2 summarizes the statistics of these datasets, indicating the number of valid and test data for each task. Each dataset provides both validation and test sets to ensure rigorous evaluation and to facilitate fine-tuning during the development process.

2.2. Fine-tuning GPT-3 for LLM and Accuracy Track

In this experiment, we focused on leveraging the capabilities of the GPT-3 model, which contains 175 billion parameters, for addressing various semantic table annotation tasks. Fine-tuning LLMs like GPT-3 can be approached in two main ways: probing and prompt engineering. Probing involves deeper adjustments of the LLMs weights for task-specific learning, while prompt engineering optimizes the input format to guide the model's responses. For our experiments, we primarily relied on prompt engineering techniques.

[2] https://orkg.org/comparison/R642266
[3] https://huggingface.co/datasets/yvelos/semantic_annotation

Table 1
Overview of the SemTab Datasets Used for the CEA Task

Datasets	Set	Year	Tables	Targets
WikidataTableR1	train	2023	500	4,247
tfood Entity	train	2023	849	2,265
tfood Horizontal	train	2023	438	24,951
Semantic_annotation	train	2024	-	15,000
WikidataTableR1	train	2024	500	4,247
WikidataTableR1	test	2024	30,000	2,276,095
tbiodiv entity	train	2024	1,539	11
tbiodiv entity	test	2024	13,852	136
tbiodiv horizontal	train	2024	4,203	568
tbiodiv horizontal	test	2024	37,832	4,544
tbiomed entity	train	2024	1,056	3,576
tbiomed entity	test	2024	9,511	31,550
tbiomed horizontal	train	2024	1,621	58,492
tbiomed horizontal	test	2024	14,590	497,905
SuperSemtab24	train	2024	16,180	305,453
SuperSemtab24	test	2024	4,044	74,837

Table 2
An Overview of the Datasets Used for CTA, CPA, RA, and TD Tasks

Datasets	Task	# Valid Targets	# Test Targets
Accuracy Track			
WikidataTableR1	CTA	623	36,626
	CPA	710	44,952
tbiodiv Entity	TD	1,539	13,852
	RA	11,109	97,113
tbiodiv Horizontal	CTA	18,100	169,908
	CPA	45,573	422,406
	TD	4,203	37,832
	RA	79,457	737,773
tbiomed Entity	TD	1,056	9,511
	RA	6,240	55,108
tbiomed Horizontal	CTA	5,227	46,002
	CPA	11,065	96,799
	TD	1,621	14,590
	RA	27,142	233,848

Specifically, few-shot prompting was employed to address the CEA task within the accuracy track, as well as the task in the LLM track. Few-shot prompting allows the model to learn patterns from a small set of examples provided during inference. On the other hand, we adopted zero-shot prompting for the CTA, RA, and TD tasks. Zero-shot prompting does not require any training examples; instead, it relies solely on the LLMs pre-trained knowledge to interpret the prompts. To facilitate these approaches, the datasets were structured such that the different SemTab tasks could be effectively interpreted and solved by GPT-3.

For the CPA task, instead of using GPT-3, we used a symbolic rule-based method. The CPA task often requires precise identification of relationships between table columns, which can be more effectively handled by deterministic rules. This hybrid strategy allowed us to exploit the strengths of both LLMs and symbolic methods.

The architecture used in this experiment is illustrated in Fig. 2. It involves several key modules, each serving a specific function in the overall system:

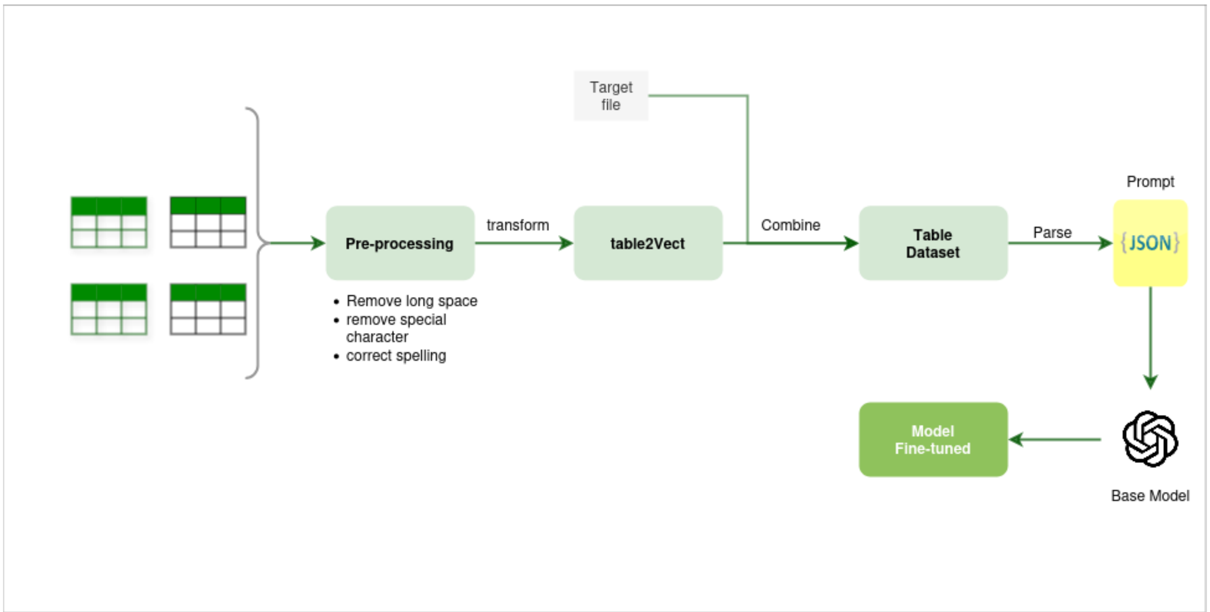
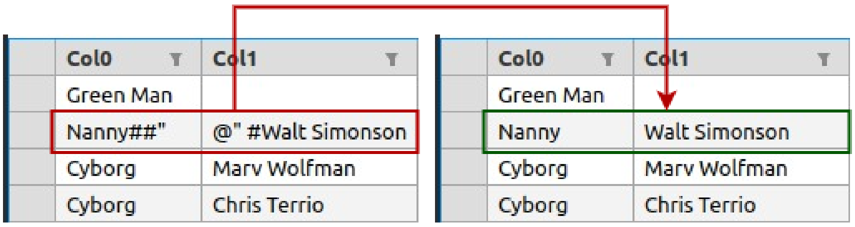


Figure 2: The Architecture of the Solution Proposed in this experiment for LLM and Accuracy Track

- **Pre-processing Module:** This module takes as input a set of tables and applies various cleaning operations such as removing blank spaces, stripping HTML tags, and eliminating special characters. An example of how a cell is processed through this module is shown in Fig 3.



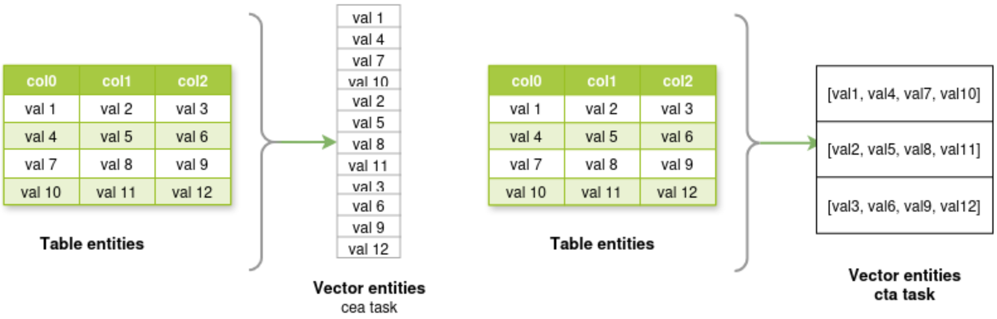
Col0	Col1
Green Man	
Nanny##"	@" #Walt Simonson
Cyborg	Marv Wolfman
Cyborg	Chris Terrio

Col0	Col1
Green Man	
Nanny	Walt Simonson
Cyborg	Marv Wolfman
Cyborg	Chris Terrio

Figure 3: Example of cell pre-processing

- ***table2vect* Module:** The *table2vect* module, as described by Algorithm 1, processes the cleaned dataset and generates task-specific vectors for CEA, CTA, CPA, RA, and TD tasks. These vectors are structured based on the requirements of each annotation task. The Fig. 4 show an example of *table2vect* process.

- **Table dataset Modules**: This module accepts a vector as input, along with a target file if provided, and then maps the vector elements to their corresponding targets. The output is a new table that represents our dataset.

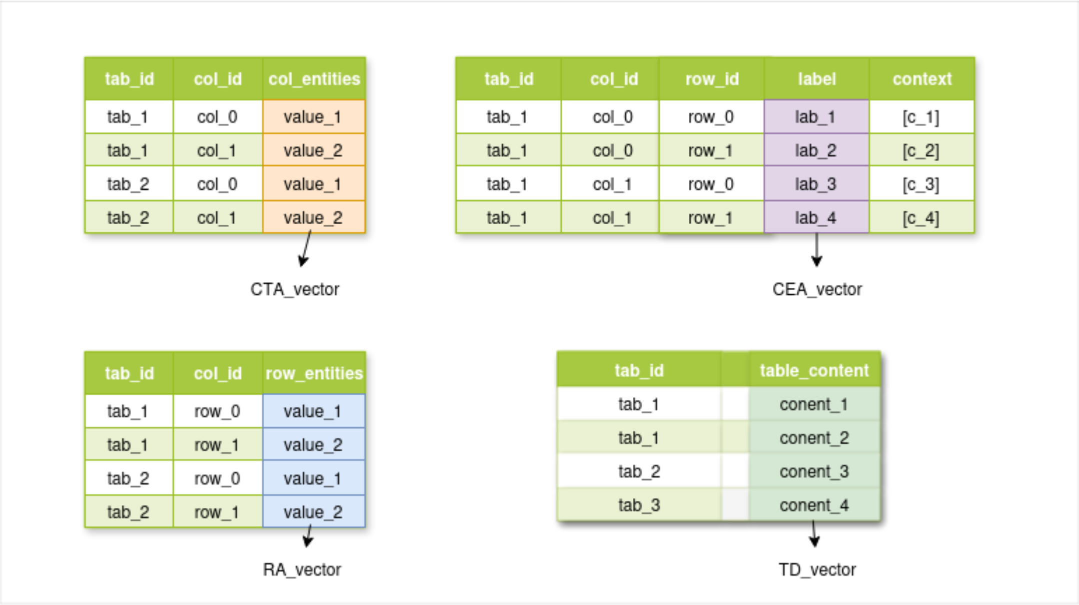


col0	col1	col2
val 1	val 2	val 3
val 4	val 5	val 6
val 7	val 8	val 9
val 10	val 11	val 12

col0	col1	col2
val 1	val 2	val 3
val 4	val 5	val 6
val 7	val 8	val 9
val 10	val 11	val 12

Table entities
[val1, val4, val7, val10]
[val2, val5, val8, val11]
[val3, val6, val9, val12]
Vector entities
cta task

Figure 4: Example of *table2ect* process for the CEA and CTA task



tab_id	col_id	col_entities
tab_1	col_0	value_1
tab_1	col_1	value_2
tab_2	col_0	value_1
tab_2	col_1	value_2

tab_id	col_id	row_id	label	context
tab_1	col_0	row_0	lab_1	[c_1]
tab_1	col_0	row_1	lab_2	[c_2]
tab_1	col_1	row_0	lab_3	[c_3]
tab_1	col_1	row_1	lab_4	[c_4]

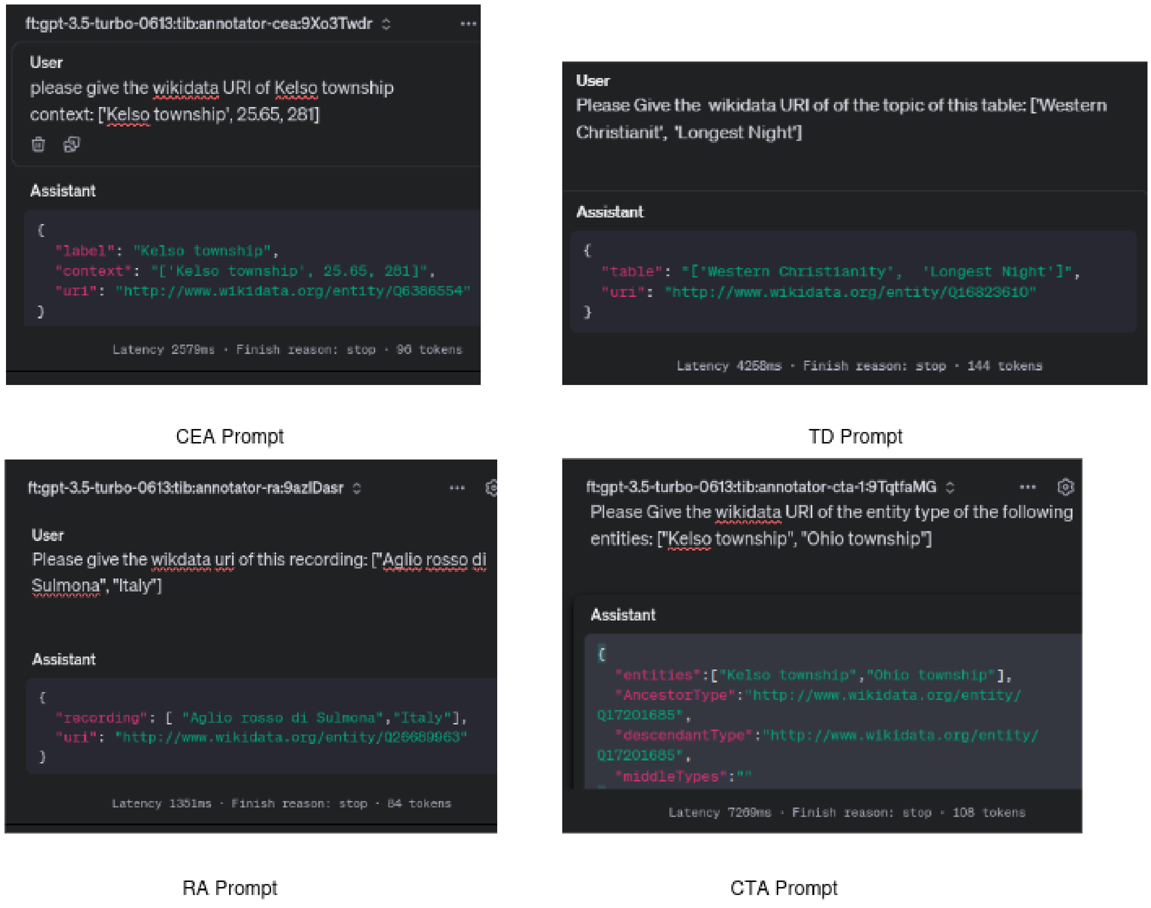
tab_id	col_id	row_entities
tab_1	row_0	value_1
tab_1	row_1	value_2
tab_2	row_0	value_1
tab_2	row_1	value_2

tab_id	table_content
tab_1	conent_1
tab_1	conent_2
tab_2	conent_3
tab_3	conent_4

Figure 5: This figure illustrates the structure of the datasets following the combination of vectors with the target files.

- **Prompt Generation Modules (ceaPrompt, ctaPrompt, raPrompt, tdPrompt):** These modules transform the rows of Table dataset into a set of questions and answers tailored for each task. For example, in the CEA task, a table cell and its context are framed as a question, while the corresponding entity serves as the answer. Examples of these question-answer pairs are embedded in Fig. 6.
- **Fine-Tuning Base GPT model:** The generated questions and answers are used to fine-tune GPT-3 or GPT-4, ensuring that the model can accurately perform the semantic annotation tasks across different datasets.

This modular architecture allows for a flexible and scalable approach to semantic table annotation, enabling the system to adapt to different tasks by simply modifying the input prompts and vectors. While GPT-3 handles most of the annotation tasks, the use of a rule-based approach for CPA underscores the importance of integrating symbolic reasoning in cases where relationship extraction is critical.



ft:gpt-3.5-turbo-0613:tib:annotator-cea:9Xo3Twdr

User
please give the wikidata URI of Kelso township
context: ['Kelso township', 25.65, 281]

Assistant

```
{
  "label": "Kelso township",
  "context": "['Kelso township', 25.65, 281]",
  "uri": "http://www.wikidata.org/entity/Q6386554"
}
```

Latency 2579ms · Finish reason: stop · 96 tokens

CEA Prompt

User
Please Give the wikidata URI of of the topic of this table: ['Western Christianit', 'Longest Night']

Assistant

```
{
  "table": "['Western Christianity',  'Longest Night']",
  "uri": "http://www.wikidata.org/entity/Q16823610"
}
```

Latency 4258ms · Finish reason: stop · 144 tokens

TD Prompt

ft:gpt-3.5-turbo-0613:tib:annotator-ra:9azlDasr

User
Please give the wikdata uri of this recording: ["Aglio rosso di Sulmona", "Italy"]

Assistant

```
{
  "recording": [ "Aglio rosso di Sulmona","Italy"],
  "uri": "http://www.wikidata.org/entity/Q26689963"
}
```

Latency 1351ms · Finish reason: stop · 84 tokens

RA Prompt

ft:gpt-3.5-turbo-0613:tib:annotator-cta-1:9TqtfaMG

Please Give the wikidata URI of the entity type of the following entities: ["Kelso township", "Ohio township"]

Assistant

```
{
  "entities":["Kelso township","Ohio township"],
  "AncestorType":"http://www.wikidata.org/entity/
Q17201685",
  "descendantType":"http://www.wikidata.org/entity/
Q17201685",
  "middleTypes":""
```

Latency 7269ms · Finish reason: stop · 108 tokens

CTA Prompt

Figure 6: Expample of prompt for the CEA, CTA, RA and TD task based on Fig. 1

2.3. Annotating the test data using the model fine-tuned

After fine-tuning the GPT-3 model for semantic table annotation tasks, the resulting model was employed to annotate the test data. The annotation workflow closely follows the first three steps of the fine-tuning process, as outlined in Fig. 1. This process is structured to handle the different annotation tasks efficiently by leveraging the pre-processing pipeline and vector generation approach discussed earlier.

The annotation process begins with inputting the set of tables to be annotated. These tables go through a pre-processing phase, which involves removing irrelevant characters, normalizing formats, and cleaning the data to ensure consistency. Following this, the *table2vect* algorithm is applied to convert the tables into a set of task-specific vectors. These vectors capture the essential elements needed for annotation, such as table cells and their context. However, for all tasks, the vectors includes a URI cell that is initially left blank. This placeholder will be populated with the correct URI during the inference stage, using the fine-tuned GPT-3.

The fine-tuned LLMs, when performing inference, processes the task-specific prompts generated from these vectors and fills in the blank spaces with the corresponding URIs or semantic labels. For example, in the CEA task, the model identifies the most relevant entity from a knowledge graph, while in the CTA task, it assigns the appropriate semantic type. The transformation from vectors to answers is handled seamlessly by GPT-3, which was trained on similar tasks during fine-tuning.

It is important to note that while GPT-3 was primarily used for tasks such as CEA, CTA, RA, and TD,

Algorithm 1 Table to Vector Conversion Algorithm (Dataset D)

```
Require: Dataset $D$
Ensure: A vector representation of the dataset
Initialize:
    vector ← list() {List to store the final vector representation}
    c ← list() {List to store non-NaN elements of the current row}
for each table $S$ in $D$ do
    for each column $col$ in $S$ do
        for each row $i$ in $S$ do
            if the value of $row[col]$ is NaN then
                vector.append([“NIL”, “NIL”])
            else
                c.clear() {Clear the list $c$ for new row elements}
                for each element in $row$ do
                    if element is not NaN then
                        c.append(element)
                    end if
                end for
                if length of $c$ > 10 then
                    Select 10 random elements from c
                else
                    Use all elements in c
                end if
                vector.append([$row[col]$, c])
            end if
        end for
    end for
end for
return vector
```

the CPA task required a different approach. The CPA task involves determining the property that links two columns in a table, a challenge that often benefits from deterministic logic rather than generative language models. Therefore, a rule-based method was applied to solve this task, as illustrated in Fig. 7. This rule-based approach relies on predefined relationships and patterns in the data, making it highly effective for capturing the structured nature of properties in knowledge graphs.

By integrating both the generative power of GPT-3 for complex annotation tasks and symbolic methods for rule-based tasks, this hybrid architecture ensures a robust and adaptable annotation pipeline. The resulting annotated datasets maintain high accuracy across all tracks, leveraging the strengths of both AI-driven models and traditional symbolic techniques.

3. Results

This section presents the evaluation results for the SemTab’24 challenge, focusing on both the STI & LLMs track (see Section 3.1) and the accuracy track (see Section 3.2). The outcomes are discussed in detail, highlighting the strengths and limitations observed during the testing phase.

3.1. LLMs Track

In the LLMs track, we fine-tuned the GPT-3 as outlined in Section 2. GPT-3 was also evaluated on the test data by the challenge organizers. Table 3 presents the results, focusing on the CEA task.

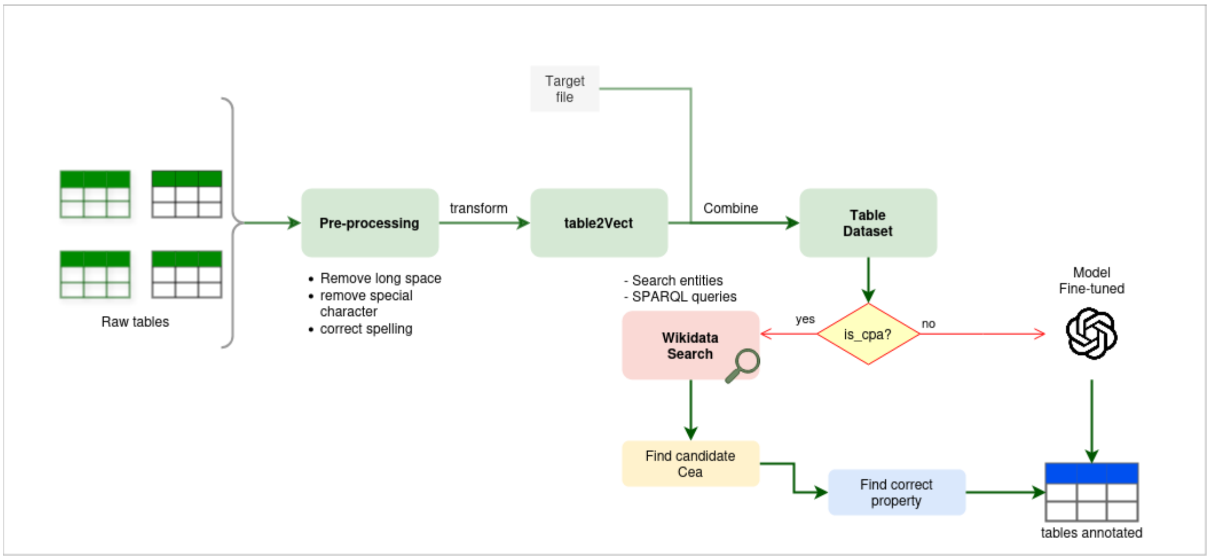


Figure 7: Annotation process using the GPT-3 fine-tuned

Table 3
Results for the Semantic Table Interpretation (STI) and Large Language Model (LLM) Track (Round 1)

Datasets	Task	F1-Score	Precision
SuperSemTab24	CEA	0.899	0.899

The results in Table 3 demonstrate the LLMs ability to perform entity annotation tasks with high accuracy. The fine-tuned LLM achieved an $f1 - score$ of 0.899 for the CEA task, which aligns closely with its precision score, indicating a balanced performance. The success in this track can be attributed to effective few-shot prompting and careful data pre-processing, which allowed the LLM to grasp the complex semantic relationships present in the tables.

3.2. Accuracy Track

For the accuracy track, the results cover a broader range of tasks, including CEA, CTA, CPA, RA, and TD, across multiple datasets. The results are summarized in Table 4.

Table 4
Results for Different Tasks in the Accuracy Track (Round 1)

Tasks	WikidataTableR1		WikidataTableR2		tbiodiv Entity		tbiodiv Horizontal		tbiomed Entity		tbiomed Horizontal	
	F1	P	F1	P	F1	P	F1	P	F1	P	F1	P
CEA	0.069	0.24	-	-	0.926	0.926	0.740	0.740	0.938	0.938	0.575	0.806
CTA	0.717	0.717	0.194	0.279	-	-	0.648	0.648	-	-	0.749	0.749
CPA	0.770	0.734	-	-	-	-	0.016	0.016	-	-	0.069	0.060
RA	-	-	-	-	0.020	0.020	0.719	0.719	0.008	0.008	0.411	0.411
TD	-	-	-	-	0.055	0.055	0.780	0.780	0.029	0.029	0.621	0.621

During the challenge, the fine-tuned model was primarily evaluated on the following datasets and tasks:

- **CEA**: WikidatableR1, tbiodiv Entity, tbiodiv Horizontal, tbiomed Entity, tbiomed Horizontal
- **CTA**: WikidataTableR1, tbiodiv Horizontal, tbiomed Horizontal

- **TD**: tbiodiv Entity, tbiodiv Horizontal, tbiomed Entity, tbiomed Horizontal
- **RA**: tbiodiv Horizontal, tbiomed Horizontal

The results indicate that the model performed well on the CEA task, particularly for the tbiodiv Entity and tbiomed Entity datasets, achieving an $f1 - score$ above 0.92. The tbiodiv Horizontal dataset, with its unique table structure, saw a slightly lower performance, with an $f1 - score$ of 0.74. This decline is likely due to the complexity introduced by the horizontal orientation of the data, which poses challenges in capturing relationships between entities.

For the CTA task, the model delivered strong results with an $f1 - score$ greater than 0.7 for the WikidataTableR1 and tbiomed Horizontal datasets, while scoring 0.648 for tbiodiv Horizontal. The TD task showed a range of $f1 - score$, from 0.78 for tbiodiv Horizontal to 0.621 for tbiomed Horizontal, reflecting the varying difficulty levels of semantic topic detection across datasets.

The RA task produced a high $f1 - score$ of 0.719 for tbiodiv Horizontal but a lower $f1 - score$ of 0.411 for tbiomed Horizontal. The disparity in performance for these tasks can be attributed to the limited availability of high-quality training data, which likely hindered the model's ability to generalize effectively.

Lastly, the CPA task suffered from incomplete test data runs, particularly for the WikidataTableR1 dataset, where only 80% of the test data was covered. The incomplete data coverage explains the lower $f1 - score$, as the model had less data to work with, leading to reduced precision and recall.

Overall, while the results show promising performance in several areas, they also highlight the challenges posed by diverse table structures, limited training data, and incomplete test coverage.

4. Conclusion

This paper presented an exploration of utilizing GPT-3 for addressing the SemTab challenge, which involves a series of complex tasks related to entity annotation and classification. To approach this, we employed the base GPT-3 model and refined its capabilities through both few-shot and zero-shot prompting techniques. The model demonstrated promising performance when applied to the complete dataset, achieving commendable results across various tasks. Specifically, for the CEA task, we observed an impressive $f1 - score$ exceeding 0.92 when the model was tested on the tbiodiv Entity and Tbiomed Entity datasets. This indicates a high level of accuracy and reliability in the model's ability to correctly annotate entities within these datasets. However, for other tasks such as CTA and TD , the $f1 - score$ ranged between 0.6 and 0.8. This variability in performance can be attributed to the limited size of the training data, which constrained the model's ability to fully generalize and optimize its predictions across these tasks. Moving forward, future work will focus on completing the remaining annotations that were not finalized before the deadline of this study. Once these annotations are completed, the results will be submitted to the SemTab challenge organizers for formal evaluation. This subsequent evaluation will provide further insights into the model's performance and its applicability to similar challenges in the field.

References

[1] G. C. Azanzi Jiomekong, Hippolyte Tapamo, An ontology for tuberculosis surveillance system, SpringerLink (2023).

[2] Y. L. Ruizhe Ma, f-kgqa: A fuzzy question answering system for knowledge graphs, ScienceDirect (2024). URL: https://www.sciencedirect.com/science/article/abs/pii/S016501142400263X.

[3] A. Jiomekong, Towards an approach based on knowledge graph refinement for tabular data to knowledge graph matching, CEUR-WS (2022). URL: https://ceur-ws.org/Vol-3320/paper12.pdf.

[4] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, Semant. Web 8 (2017) 489–508 (2016). URL: http://dx.doi.org/10.3233/SW-160218.

[5] M. I. Sander Schulhoff, The prompt report: A systematic survey of prompting techniques, arxiv (2024). URL: https://arxiv.org/abs/2406.06608.

[6] O. Hassanzadeh, Semantic tabular data annotation to knowledge graph matching, in: Semtab challenge, 2024. URL: https://sem-tab-challenge.github.io/2024/.

[7] N. Abdelmageed, tfood: Semantic table annotations benchmark for food domain, Zenodo (2023). URL: https://zenodo.org/records/10048187.

[8] N. Abdelmageed, tbiomed: Semantic table annotations benchmark for biomedical domain, Zenodo (2024). URL: https://zenodo.org/records/10996334.

[9] N. Abdelmageed, tbiodiv: Semantic table annotations benchmark for biodiversity domain, Zenodo (2024). URL: https://zenodo.org/records/10996688.

[10] Cremaschi, Semtab 24: Semantic table annotations benchmark for llm-based approaches, Zenodo (2024). URL: https://zenodo.org/records/11031987.