Cell Entity Annotation for SemTab 2025 MammoTab via Iterative Refinement with Transposed Contexts and Unsupervised Scoring*

Yuuki Tachioka[1], Yasunori Terao[1]

[1]Denso IT Laboratory, 13F Shintora Yasuda Bldg., 4-3-1 Shimbashi, Minato-ku, Tokyo, Japan

Abstract

This paper presents a robust approach to Cell Entity Annotation (CEA) in the ISWC 2025 SemTab Challenge MammoTab task, where tables must be linked to Wikidata entities without gold labels. We propose a multi-stage Wikidata entity identifier candidate generation pipeline combined with an iterative process alternating between Column Type Annotation (CTA) and CEA. Candidate sets are refined through both the original and transposed table orientations, with final candidates taken as the union of both to leverage complementary contextual cues. We also introduce unsupervised evaluation metrics, consistency and entropy, which enable performance estimation and iteration control without labeled data. Experiments on 84,907 entities and 3,576 columns show that our method improves label coverage and semantic coherence, and the best-selection strategy achieved the highest scores. The results demonstrate that combining multi-orientation candidate generation with iterative refinement and unsupervised evaluation provides a practical and effective solution for large-scale, label-free entity linking in tabular data.

Keywords

Cell Entity Annotation, Entity Linking, Wikidata, Large Language Models, Transpose Strategy, Unsupervised Evaluation

1. Introduction

In recent years, the rapid development of Large Language Models (LLMs) has enabled the understanding of not only text but also multiple modalities such as images and tables. Tabular data, in particular, exhibit a high degree of structural clarity while often lacking explicit column names or contextual information, making it challenging to accurately capture their intended meaning. Understanding the semantics of table data plays an important role in data integration, knowledge discovery, information retrieval, and downstream decision-making tasks [1, 2, 3].

Among research initiatives that address these challenges, SemTab provides an internationally recognized benchmark specifically designed for the semantic interpretation of cell contents and table structures by linking them to knowledge bases [4]. In the ISWC 2025 SemTab Challenge, the MammoTab task [5] focuses on Cell Entity Annotation (CEA) as a primary objective. CEA refers to the entity linking task of associating the content of each cell with a Wikidata entity identifier (QID) [6], which requires deep semantic understanding beyond simple string matching.

The challenge evaluates systems not only by F1 score, but also across multiple complex and realistic dimensions, including:

- Robustness to noise and ambiguity (Noise Robustness, Disambiguation, Alias Resolution, etc.)
- NIL Detection (identifying cases where no corresponding entity exists)
- Collective Inference (leveraging correlations with other cells and columns for reasoning)

These require strategies that consider the semantics of the entire table, rather than improving accuracy at the single-cell level alone.

OM 2025: The 20th International Workshop on Ontology Matching collocated with the 24th International Semantic Web Conference (ISWC 2025), November 2nd, 2025, Nara, Japan

*Corresponding author.

tachioka.yuki@core.d-itlab.co.jp (Y. Tachioka); terao.yasunori@core.d-itlab.co.jp (Y. Terao)
0009-0002-0587-2943 (Y. Tachioka)

In this work, we propose a multi-stage QID candidate generation strategy combined with an iterative process alternating between Column Type Annotation (CTA) [9] and CEA, with the aim of achieving high accuracy and reduced noise[7]. In particular, NIL detection requires minimizing noise during the candidate generation phase [8]. Our approach applies CTA and CEA in a loop to mutually reinforce the CTA and the selection of QID candidates. In addition, recent studies have explored LLM-based table generation and interpretation methods [1, 2, 3], demonstrating their strong potential to capture relational semantics in complex tabular structures [10]; therefore, we also explore the automatic estimation of column names using ChatGPT. Furthermore, to maximize the benefits of collective inference, we utilize both the original table and its transposed form, thereby capturing semantic relationships along both rows and columns.

Moreover, since no development set is provided for this task, unsupervised methods for performance estimation are necessary. We propose calculating both consistency scores and entropy scores for predicted labels and selecting outputs based on the principle of minimization of entropy [11].

2. ISWC 2025 SemTab Challenge: MammoTab Task

Tabular data, particularly in the CSV format, is widely used in data analysis pipelines. However, the lack of an explicit semantic structure often hinders effective analysis. Tables available on the Web also serve as valuable information sources and, by enriching them with semantic annotations, they can be leveraged for applications such as search, question answering, and knowledge base construction.

The SemTab Challenge provides a benchmark for fair evaluation and comparison of systems that match knowledge graphs (KGs) and tables. In the MammoTab task, the focus is on CEA based on Wikidata (version 20240720), with participating systems required to address the following challenges:

- Disambiguation
- Homonymy resolution
- Alias resolution
- NIL detection
- Noise robustness
- Collective inference

Moreover, the task restricts approaches to those based on LLMs, either through fine-tuning or Retrieval-Augmented Generation.

3. Proposed Method

3.1. Overview

Figure 1 illustrates the overall architecture of the proposed method. The process begins with a six-stage candidate generation pipeline (3.2) that extracts potential QIDs from table cells. In the CTA stage, the column-level semantics are estimated. The estimated column types serve as contextual cues for the CEA stage, which links individual cells to their most appropriate QIDs. The CTA and CEA modules operate in an iterative loop (3.4), allowing each to reinforce the other. CTA refines the context of columns, while CEA updates the predictions of QIDs of cell entities based on improved column semantics. This mutual refinement progressively enhances the overall semantic coherence of the annotations.

Furthermore, the proposed framework applies not only to the original table, but also to its transposed version (3.5). This dual-orientation processing allows the model to capture complementary contextual cues along both rows and columns. To input tables into LLM, the tables are vectorized. For example, in the case of TableLlama [12], which is a baseline model in this paper, the input of tables is in row-major order and the entities that belong to the same column are remote for each row in the input prompt, thus, TableLlama may struggle with long vertical columns; processing the transposed table mitigates this issue by aligning semantically related cells closer in textual space.

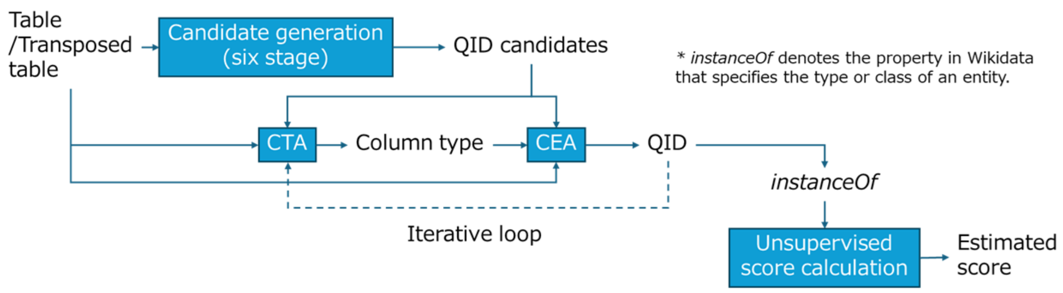
Table /Transposed table
Candidate generation (six stage)
QID candidates
* instanceOf denotes the property in Wikidata that specifies the type or class of an entity.
CTA
Column type
CEA
QID
instanceOf
Iterative loop
Unsupervised score calculation
Estimated score

Figure 1: Overview of the proposed method. The process applies the same six-stage candidate generation, iterative CTA–CEA refinement, and unsupervised score calculation to both the original and transposed tables. By leveraging complementary contexts from both orientations, the method enhances robustness and achieves more consistent entity annotations.

After performing the same candidate generation, iterative refinement of CTA-CEA and unsupervised scoring (3.6) in both orientations, the results are compared and the best performing annotations are selected. This orientation-based ensemble enhances robustness and consistency across different table structures, effectively improving performance in various data layouts.

3.2. Multi-stage Entity Candidate Generation

For each cell, QID candidates are generated from the latest Wikidata dump using a six-stage process designed to minimize the inclusion of unrelated entities into the candidate set:

1. Year and Digit Detection:
   - If the values in the column are numeric and within the range 1770–2030, the column is classified as a *year* column.
   - If the values form a sequential series, the column is classified as an *ID* column.
   - Appropriate QIDs are assigned to each type accordingly.
2. Exact Match: If the cell value exactly matches a Wikidata label string, the corresponding QID is added to the candidate set.
3. Set Match: The cell value is tokenized in words, and if the resulting set exactly matches a Wikidata label word set, the corresponding QID is added.
4. Set Match (Description Removal): Parenthetical text, comma-separated suffixes, and other supplementary descriptions are removed from the cell value; if the resulting word set matches a Wikidata label set, the QID is added.
5. Partial Set Match (label in cell value): Word sets partially matching a label's word set are considered, but restricted to sets of size at most $\min(2 \times |S|, |S| + 5)$ where S is the word set of the original label. For example:
   - If $|S| = 2$, allow up to 4 words.
   - If $|S| = 6$, allow up to 11 words.
6. Partial Set Match (cell value in label): Similar to step 5, but checking whether the cell value's word set is contained in the label's word set, with the same size restrictions.

This staged approach ensures high coverage while reducing the likelihood of introducing noisy QID candidates.

3.3. Initial Column Type Annotation (CTA)

3.3.1. Majority Voting

Column labels are estimated from the generated candidate sets as follows:

- For each column, retrieve the `instanceOf` labels of its multi-stage QID candidates from Wikidata.
- Flatten the label set, and if the most frequent label exceeds a majority threshold, assign it as the column's representative label.
- Aggregate the labels of all columns to form a *pseudo-title* for the table.

3.3.2. ChatGPT-based Column Name Generation

In addition, we prompt ChatGPT with the contents of each column to predict plausible column names, which are then combined to form another pseudo-title for the table.

3.4. Iterative CEA and CTA

Using the initial CTA results, we perform CEA with the state-of-the-art model, TableLlama [12]. Based on the obtained QIDs, the CTA is updated to re-estimate column semantics. CEA is then performed again using the updated CTA. By alternating CTA and CEA in this way, we aim for mutual performance improvements.

3.5. Transpose Table Strategy

Some tables contain long vertical columns, which models based on row-major input such as TableLlama may find challenging to interpret. By transposing the table before input, cell values that are semantically related can be positioned closer together in text space, reinforcing contextual signals:

- This often improves the consistency of the type within columns.
- For cases where row-wise relationships are important, the original orientation is preferable.

We perform a CTA on both the original and transposed versions of the table, improving QID accuracy through complementary perspectives.

3.6. Unsupervised Score Calculation

Since no development set with labels is provided, the performance of the model must be estimated without supervision, following the principle of minimization of entropy [11].
Consistency Score: for each column, it computes the agreement ratio of the `instanceOf` label mode:

$$\text{mode ratio} = \frac{\text{count}(\text{mode}(\mathscr{L}))}{|\mathscr{L}|}$$

where $\mathscr{L}$ is the set of predicted labels.
Entropy Score: for each column, it computes the entropy of the `instanceOf` label distribution, where a lower entropy indicates more consistent predictions:

$$\text{entropy} = -\sum_{l\in\mathscr{L}} \frac{\text{count}(l)}{N} \log_2 \frac{\text{count}(l)}{N}$$

where N is the number of cells in the column.

4. Experiment

4.1. Setup

Experiments were conducted on the MammoTab task of the ISWC 2025 SemTab Challenge. The primary objectives were to improve the accuracy of the CEA for tables and to evaluate the proposed unsupervised scoring methods on unlabeled data. The experimental dataset contained a total of 84,907 unique entities and 3,576 target columns to be annotated.

For comparison, we prepared the following experimental variations:

- Initial CTA based on candidate sets
- Initial CTA based on ChatGPT-generated column names
- Comparison and integration of results from the original and transposed table representations
- Iterative CEA and CTA with 1 to 4 alternating steps

4.2. LLM setup

For LLM-based column name estimation, we used GPT-4.1 accessed via the OpenAI API. The model was prompted with the entire table (converted to Markdown format) and instructed to infer meaningful names for all columns labeled “Unknown.X”. In a single inference, the CTA for all columns was executed simultaneously. The complete prompt used for the generation of ChatGPT-based column names is shown in Listing 1. This prompt guides the model to infer meaningful names for columns labeled as “Unknown.X” and return the results in a structured JSON format.

```
You are given a table with column names, some of which are labeled as "Unknown.X"
(e.g., Unknown.1, Unknown.2, etc.). Your task is to infer the most likely meaning
of each "Unknown.X" column based on the data it contains. For each "Unknown.X"
column, provide:

1. A list of multiple possible column name candidates.
2. The single best column name you believe fits the data.

For columns that are not labeled "Unknown.X", you may suggest a better name if it
is clearly more descriptive or accurate, but this is optional. If you do suggest
a new name, also provide candidates and the single best choice, similar to the
Unknown columns.

The output must be in valid JSON format, using the following structure:

{
  "columns": {
    "<original_column_name>": {
      "candidates": ["candidate_1", "candidate_2", "..."],
      "best_guess": "best_candidate"
    },
    ...
  }
}

Do not include any explanations outside of the JSON.

Here is the table to analyze:
{{ table }}
```

Listing 1: Prompt used for ChatGPT-based CTA generation.

A total of 756 API calls were made, corresponding to the number of Markdown-converted tables. We used the Batch API, with a pricing rate of $1.00 per 1 M input tokens and $4.00 per 1 M output tokens. Given approximately 0.5M input tokens (2.1 M words) and 0.4M output tokens (1.4 M words), the total cost was approximately $2.1. The relatively low computational cost demonstrates the practicality of using LLM-based column name estimation at scale.

4.3. Multi-stage Entity Candidate Generation

The results in Table 1 show how the proposed multi-stage QID candidate generation pipeline progressively improves coverage while controlling noise. Stage 1, which performs exact matching and

Table 1
Multi-stage Entity Candidate Generation results and cumulative coverage rates.

Stage	Found	Unmatched	Cumulative Found	Coverage (%)
1	24596	10482	24596	70.1
2	674	9808	25270	72.0
3	934	8800	26204	74.7
4	3110	5690	29314	83.6
5	1035	4655	30349	86.5
6	2040	2615	32389	92.3

Table 2
Example of correctly labeled columns using initial CTA (multi-stage candidate generation + `instanceOf`-based majority voting).

Year	film	human.film director
1976	Eat My Dust!	Charles Byron Griffith
1976	Hollywood Boulevard	Joe Dante

high-confidence heuristics, immediately recovers 70.1% of the entities (24,596 matches), leaving 10,482 unmatched. Stages 2 and 3 provide modest gains (0.6% and 2.7% additional coverage, respectively), indicating that set-based matching and description-removed matching capture only a small fraction of the remaining entities. A substantial improvement occurs at Stage 4, where controlled partial set matching (label in cell value) produces a jump of 8.9 percentage points, reducing the unmatched set to 5,690 entities. This suggests that carefully expanding match criteria is highly effective in bridging coverage gaps. Stages 5 and 6 add further recall through reversed partial matching strategies, reaching a final coverage of 92.3%.

Overall, the stage-wise progression reflects a deliberate trade-off: early stages prioritize precision with strict matching, while later stages boost recall by relaxing constraints in a controlled manner. The large gain at Stage 4 underscores its central role in balancing coverage expansion against noise suppression.

4.4. Initial CTA Performance

Table 2 shows an example in which the initial CTA, estimated from the candidate set using majority voting, correctly assigned column labels. Given that the evaluation target consisted of a total of 3,576 columns, the Step 1 results in Table 3 show that the columns $328 + 23 + 303 + 942 = 1,596$ remained unlabeled. This corresponds to approximately 55.4% of the columns being successfully labeled in the initial CTA stage. These results indicate that the combination of multi-stage candidate generation and `instanceOf`-based CTA enables a reasonable degree of column semantic estimation even without explicit column names.

4.5. Iterative CTA and CEA Performance

4.5.1. Labeling Coverage (Table 3, Table 4)

Tables 3 and 4 show the number of columns for which labels could not be assigned at each step for different initialization strategies. Significant improvements are observed in the early iterations (Step 1 and Step 2), after which the performance gains diminish, indicating that the iterative process quickly approaches a stable state. The transpose-based approach (Step 2($\top$) in Table 3) is particularly effective in reducing the number of unlabeled cells, especially in challenging cases where the most frequent label is a *Wikimedia disambiguation page*. When starting with ChatGPT-based column name estimation (Table 4), the initial steps also produce improvements; however, later iterations show limited additional

Table 3
Number of columns without labels for candidate set–based initialization and corresponding labeling rates.

Step	Empty type list	Max instance < 50%	Disambiguation page majority	Low element variation	Labeling rate (%)
1	328	23	303	942	55.4
2	329	463	122	942	48.3
2(⊤)	328	158	88	942	59.1
3(⊤)	328	142	92	942	59.1
4(⊤)	328	135	85	1002	58.0

Table 4
Number of columns without labels for ChatGPT-based initialization and corresponding labeling rates.

Step	Empty type list	Max instance < 50%	Disambiguation page majority	Low element variation	Labeling rate (%)
1	311	126	79	1034	58.9
2	310	126	77	1015	59.4
2(⊤)	330	134	90	1002	58.1

gains, and candidate set-based initialization ultimately achieves more stable and consistent performance across iterations.

Given that the evaluation target comprised a total of 3,576 columns, we also calculated the labeling rate for each step, defined as the percentage of columns with assigned labels out of the total. As shown in Table 3, the candidate set–based initialization labeled approximately 55.4% of columns at Step 1, while the ChatGPT-based initialization (Table 4) achieved a slightly higher rate of 58.9% at the same step. Interestingly, the labeling rate temporarily decreases at Step 2 in both settings. This drop can be explained by the fact that the first CEA pass fixes entity linking decisions, which in turn reduces the diversity of available candidates for subsequent CTA, leaving some columns without any dominant type label. In practice, this means that low-confidence or noisy candidates are eliminated, which may reduce coverage, but can also improve overall precision in later iterations. Thus, the Step 2 decrease should not necessarily be interpreted as a performance degradation, but rather as a selective filtering effect that prioritizes high-confidence assignments. In particular, the transpose-based approach in Step 2(⊤) for candidate set–based initialization reached the highest coverage of 59.1%, indicating that transposition can recover labels for some columns that remain unlabeled in the original orientation. In our implementation, the set of candidates for each column is constructed as the union of candidates obtained from the original and transposed table representations, thereby leveraging complementary contextual cues from both orientations.

4.5.2. Unsupervised Score (Tables 5–10)

We evaluated the proposed *unsupervised scores* to quantitatively assess prediction stability and uncertainty without explicit gold labels. For the original table orientation (Table 5), there was a substantial improvement from Step 1 to Step 2 for both metrics, after which the scores plateaued, indicating that the iterative process quickly reaches a stable state. When using the transposed table (Table 6), both the consistency and entropy scores were generally higher than those of the original table, suggesting that the transposed format facilitates more coherent type predictions across columns.

In the case of ChatGPT-based initialization (Table 7), Step 1 achieved better scores than the initialization based on the candidate set, showing the benefit of leveraging the LLM-generated column names for the initial step. However, this advantage diminished in Step 2, where the scores decreased noticeably, implying that the initial labels derived from LLM may introduce inconsistencies during subsequent iterations. For the transposed setting (Table 8), Step 1 achieved slightly higher consistency and lower

Table 5
Iterative CEA unsupervised scores (original table).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	0.538	(0.292, 0.440, 0.862)
	Entropy	1.689	(0.701, 1.588, 2.442)
2	Consistency	**0.549**	(0.308, 0.455, 0.902)
	Entropy	1.638	(0.592, 1.585, 2.350)
3	Consistency	**0.549**	(0.304, 0.444, 0.909)
	Entropy	1.641	(0.589, 1.585, 2.370)
4	Consistency	**0.549**	(0.308, 0.450, 0.898)
	Entropy	**1.636**	(0.592, 1.585, 2.359)

Table 6
Iterative CEA unsupervised scores (transposed table).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	0.543	(0.286, 0.449, 0.875)
	Entropy	1.676	(0.650, 1.585, 2.420)
2	Consistency	**0.552**	(0.307, 0.467, 0.900)
	Entropy	**1.629**	(0.597, 1.561, 2.360)
3	Consistency	0.549	(0.300, 0.455, 0.889)
	Entropy	1.641	(0.624, 1.577, 2.371)
4	Consistency	0.550	(0.304, 0.462, 0.900)
	Entropy	1.636	(0.599, 1.585, 2.371)

Table 7
Iterative CEA unsupervised scores (ChatGPT-based initialization).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	0.547	(0.302, 0.444, 0.886)
	Entropy	**1.648**	(0.643, 1.585, 2.371)
2	Consistency	**0.549**	(0.297, 0.462, 0.895)
	Entropy	1.657	(0.628, 1.585, 2.406)

Table 8
Iterative CEA unsupervised scores (transposed table, ChatGPT-based initialization).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	**0.548**	(0.300, 0.447, 0.889)
	Entropy	**1.650**	(0.627, 1.585, 2.396)
2	Consistency	**0.548**	(0.300, 0.455, 0.892)
	Entropy	1.662	(0.625, 1.585, 2.413)

entropy compared to the original ChatGPT start (Table 7), indicating modest gains from improved contextual proximity. However, Step 2 did not yield further improvements, suggesting a limited benefit from iterative refinement in this configuration.

Finally, selecting the best score from the original or transposed results at each step (Table 9), we were able to combine the strengths of both orientations, achieving the highest performance in all steps (mean consistency = 0.563, mean entropy = 1.585). This indicates that the two orientations provide

Table 9
Iterative CEA unsupervised scores (best of original/transpose).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	0.557	(0.308, 0.482, 0.903)
	Entropy	1.621	(0.581, 1.573, 2.326)
2	Consistency	**0.563**	(0.325, 0.495, 0.923)
	Entropy	**1.585**	(0.463, 1.530, 2.281)
3	Consistency	0.562	(0.318, 0.483, 0.924)
	Entropy	1.586	(0.465, 1.549, 2.285)
4	Consistency	0.561	(0.324, 0.490, 0.923)
	Entropy	1.587	(0.481, 1.554, 2.292)

Table 10
Iterative CEA unsupervised scores (best of original/transpose, ChatGPT-based initialization).

Step	Score	Mean	Quantile (0.25, 0.50, 0.75)
1	Consistency	0.560	(0.316, 0.474, 0.912)
	Entropy	**1.596**	(0.503, 1.549, 2.307)
2	Consistency	**0.562**	(0.308, 0.500, 0.918)
	Entropy	1.604	(0.507, 1.544, 2.322)

complementary information that can be exploited to improve unsupervised performance estimation. When applying best selection (Table 10), both consistency and entropy scores improved over single-orientation ChatGPT start results. This confirms that, as with candidate set–based initialization, combining complementary orientations improves robustness. However, even in the best selection setting, the initialization of ChatGPT did not exceed the highest scores obtained from the best selection based on the candidate set (Table 9), indicating that the LLM-derived column names, while helpful in the early stages, require additional filtering to match the stability of candidate-based methods.

4.5.3. Visualization of Unsupervised Score Distributions

Visualizing the distributions of unsupervised scores allows for an intuitive comparison between steps and methods. For Step 1 with the original table orientation (Figure 2), the scores are widely distributed without clear mode, indicating a high variability and low consistency in the predictions. This reflects the insufficient contextual information available when only the initial CTA and CEA are combined.

In Step 2 (Figure 3), the score distribution becomes more concentrated and the mode agreement rate improves. This suggests that the iterative reasoning process effectively enhances semantic coherence between columns, enabling more stable label estimation for many columns. When using the transposed table in Step 2 (Figure 4), the score distribution becomes even sharper, with an increased number of high-scoring columns. This trend implies that transposition emphasizes textual proximity between semantically related cells, improving the contextual understanding of the model, especially for columns with strong context dependency.

Finally, when selecting the better prediction between the original and transposed results for each column in Step 2 (Figure 5), the score distribution reaches its highest concentration, with almost no columns of low scores remaining. This shows that choosing the better orientation for each column serves as an effective and practical ensemble strategy, maximizing overall consistency and prediction reliability.

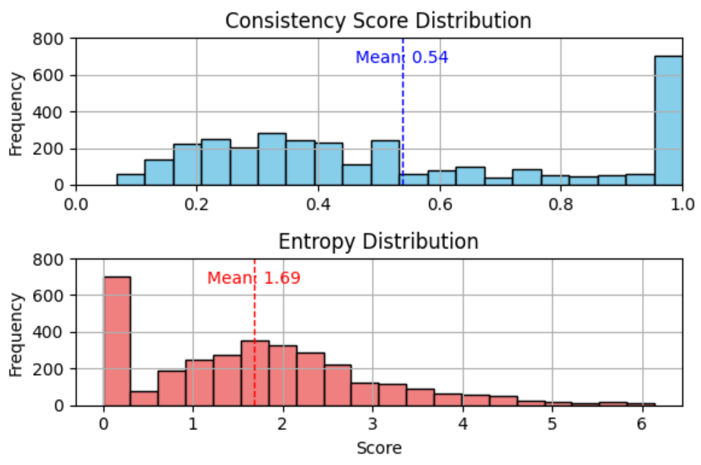

Figure 2: Step 1 score distribution for the original table.

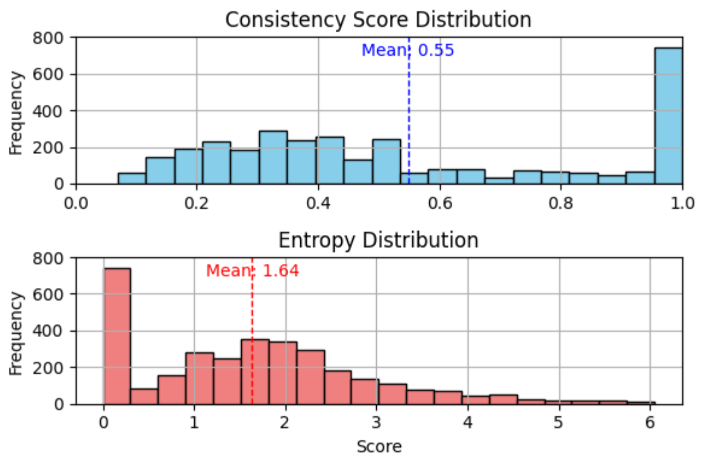

Figure 3: Step 2 score distribution for the original table.

4.6. Discussion

The experimental results clearly demonstrate that collective inference enhances column-wise consistency and refines cell-level entity predictions. Through the iterative interaction between CTA and CEA, even columns with insufficient initial information benefited from relationships with other columns, leading to improved accuracy. This effect is particularly evident in the notable increase in the consistency score from Step 1 to Step 2 (Figures 2 and 3), indicating that semantic coherence across the table was strengthened in the early stages of iteration.

The utility of the transpose strategy was also confirmed. By increasing textual proximity between semantically related cells, the model's contextual understanding was significantly enhanced, especially when the model input is in row-major order. In Step 2 with the transposed table (Figure 4), the score distribution reached its peak, with higher agreement rates and lower entropy compared to the original orientation. This suggests that transposition acts as an effective form of context reinforcement, particularly for columns requiring strong contextual cues.

Moreover, selecting the better result between the original and transposed orientations for each column proved to be a powerful ensemble-like strategy. This "best selection" approach (Figure 5) produced the most concentrated score distribution, with a substantial reduction in low-scoring columns. The outcome highlights that structural uncertainty in the interpretation of the table can be mitigated by integrating complementary perspectives from different orientations, thus improving the robustness of the model predictions.

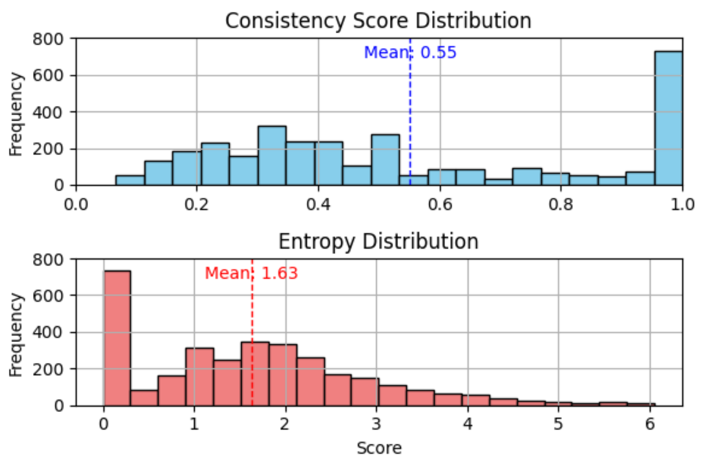

Figure 4: Step 2 score distribution for the transposed table.

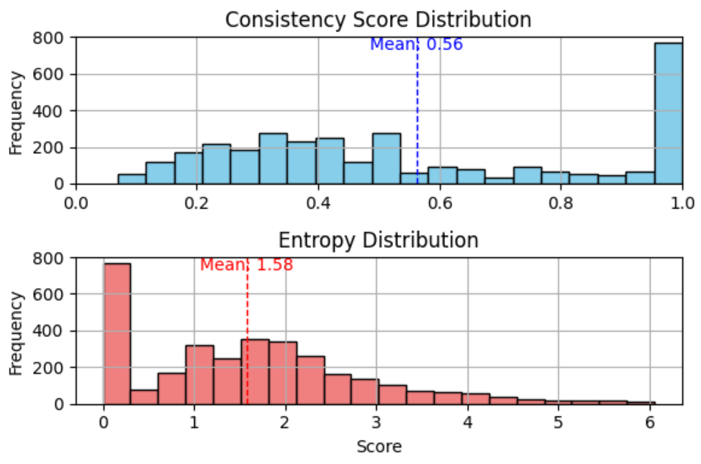

Figure 5: Step 2 score distribution for the best of original and transposed tables.

On the other hand, the use of ChatGPT-generated column names for initialization exhibited clear limitations. Although this approach provided a temporary advantage in Step 1 over candidate set–based initialization, it failed to adapt effectively in later iterations. The column names generated by ChatGPT were sometimes ambiguous or overly specific, introducing noise during the iterative process. As shown in Table 7, the initial benefit was not sustained and subsequent scores lagged behind those from initialization according to the set of candidates.

Finally, the proposed unsupervised evaluation metrics, consistency and entropy, proved to be reliable indicators of model performance trends, even in the absence of gold labels. The marked improvement from Step 1 to Step 2, followed by gradual stabilization, quantitatively supports the qualitative enhancement of the label predictions. These metrics could be further applied for early stopping in iterative processes or for detecting overfitting, offering practical benefits for unsupervised table annotation pipelines.

5. Conclusion

This study presented a method for CEA in the ISWC 2025 SemTab Challenge MammoTab task, combining multi-stage QID candidate generation with an iterative process of CTA and CEA. The approach was designed to operate without gold labels, using unsupervised evaluation metrics based on consistency and entropy to monitor and guide performance.

Experimental results demonstrated that the iterative CTA–CEA framework substantially improved column-wise semantic coherence, particularly between Step 1 and Step 2, and that the transpose strategy further enhanced contextual understanding for table-data LLMs thanks to better table data vectorization. Selecting the better result between the original and transposed tables yielded the highest overall performance, confirming the value of integrating complementary structural perspectives.

Although ChatGPT-based initialization provided a temporary advantage in early iterations, it proved less effective in later stages due to the introduction of noise. The unsupervised metrics reliably reflected performance trends, suggesting their applicability for early stopping and overfitting detection in unsupervised annotation settings.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 and Writefull for grammar and spelling checks, as well as for paraphrasing and rewording. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Y. Sui, M. Zhou, M. Zhou, S. Han, D. Zhang, Table meets llm: Can large language models understand structured table data? a benchmark and empirical study, in: The 17th ACM International Conference on Web Search and Data Mining (WSDM '24), 2024.

[2] X. Fang, W. Xu, F. A. Tan, Z. Hu, J. Zhang, Y. Qi, S. H. Sengamedu, C. Faloutsos, Large language models (LLMs) on tabular data: Prediction, generation, and understanding - a survey, Transactions on Machine Learning Research (2024). URL: https://openreview.net/forum?id=IZnrCGF9WI.

[3] X. Wu, A. Ritter, W. Xu, Tabular data understanding with llms: A survey of recent advances and challenges, 2025. URL: https://arxiv.org/abs/2508.00217. arXiv:2508.00217.

[4] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems, in: A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, M. Cochez (Eds.), The Semantic Web, Springer International Publishing, Cham, 2020, pp. 514–530.

[5] M. Marzocchi, M. Cremaschi, R. Pozzi, R. Avogadro, M. Palmonari, Mammotab: A giant and comprehensive dataset for semantic table interpretation, in: SemTab@ISWC, 2022, pp. 28–33. URL: https://ceur-ws.org/Vol-3320/paper3.pdf.

[6] C. S. Bhagavatula, T. Noraset, D. Downey, TabEL: Entity linking in web tables, in: M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, K. Thirunarayan, S. Staab (Eds.), The Semantic Web - ISWC 2015, Springer International Publishing, Cham, 2015, pp. 425–441.

[7] P. Nguyen, N. Kertkeidkachorn, R. Ichise, H. Takeda, MTab4D: Semantic annotation of tabular data with dbpedia, Semantic Web 15 (2024) 2613–2637. URL: https://journals.sagepub.com/doi/abs/10.3233/SW-223098. doi:10.3233/SW-223098. arXiv:https://journals.sagepub.com/doi/pdf/10.3233/SW-223098.

[8] P. Ruas, F. M. Couto, NILINKER: Attention-based approach to nil entity linking, Journal of Biomedical Informatics 132 (2022) 104137. URL: https://www.sciencedirect.com/science/article/pii/S1532046422001526. doi:https://doi.org/10.1016/j.jbi.2022.104137.

[9] M. Hulsebos, K. Hu, M. Bakker, E. Zgraggen, A. Satyanarayan, T. Kraska, c. Demiralp, C. Hidalgo, Sherlock: A deep learning approach to semantic data type detection, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1500–1508. URL: https://doi.org/10.1145/3292500.3330993. doi:10.1145/3292500.3330993.

[10] J. P. Bikim, C. A. A. Ymele, A. Jiomekong, A. Oelen, G. Rabby, J. D'Souza, S. Auer, Leveraging

GPT models for semantic table annotation, in: SemTab@ISWC, 2024, pp. 43–53. URL: https://ceur-ws.org/Vol-3889/paper3.pdf.

[11] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, in: Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS'04, MIT Press, Cambridge, MA, USA, 2004, p. 529–536.

[12] T. Zhang, X. Yue, Y. Li, H. Sun, TableLlama: Towards open large generalist models for tables, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 6024–6044. URL: https://aclanthology.org/2024.naacl-long.335/. doi:10.18653/v1/2024.naacl-long.335.