Representation, Retrieval, and Decision Space: A Case Study on Diagnosing LLM Failures in Column Type Annotation

Jianhao Cao
University of British Columbia
Vancouver, British Columbia, Canada
jhcao@cs.ubc.ca

Rachel Pottinger
University of British Columbia
Vancouver, British Columbia, Canada
rap@cs.ubc.ca

ABSTRACT

Column type annotation (CTA) assigns semantic labels to table columns for table understanding, metadata management, and data integration. Although recent large language model (LLM)-based methods achieve strong performance on CTA, they still produce systematic prediction errors that remain poorly understood. In this paper, we present a case study diagnosing LLM failures in CTA from three perspectives: representation bias, retrieval incompleteness, and decision-space complexity. Our analysis reveals that errors are structured rather than random. Among the universally missed cases, many are already closer to incorrect-label clusters than to their ground-truth clusters in the semantic representation space. Retrieval analysis shows that retrieved examples carry strong label signals, but performance drops sharply when retrieved examples fail to align with ground-truth labels. A controlled binary-choice probing study reveals that some errors are resolved when the candidate label set is narrowed, whereas others persist even under a reduced label space and additional in-table label context. These findings suggest that improving LLM-based CTA requires targeted interventions tailored to error profiles rather than a generic fix.

VLDB Workshop Reference Format:
Jianhao Cao and Rachel Pottinger. Representation, Retrieval, and Decision Space: A Case Study on Diagnosing LLM Failures in Column Type Annotation. VLDB 2026 Workshop: Tabular Data Analysis (TaDA).

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://www.github.com/jhcao1024/cta-llm-probe.

1 INTRODUCTION

The volume of data is rapidly growing, which necessitates high-quality metadata to effectively interpret, manage, and integrate data. As a classical metadata annotation task for tabular data, column type annotation (CTA) refers to the task of assigning a semantic label from a predefined label set to each column in a table [11]. For example, the query table in Figure 1 contains information about an athlete's participation in Olympic Games, and its four columns can be annotated with the semantic labels `age`, `city`, `team`, and `rank`, respectively. CTA is a semantic table interpretation task that supports data discovery, table understanding, data integration, and knowledge-base construction [3, 4]. Large language models (LLMs) have increasingly been applied to CTA [6], and recent work has shown that LLM-based CTA methods, especially when paired with carefully designed task prompts and contextual information, can substantially outperform earlier specialized models [1, 13]. However, despite these improvements, LLM-based methods still exhibit systematic and recurring false predictions. For example, LLMs may incorrectly predict the third column in Figure 1 as `country`. Although this prediction may appear plausible in isolation, it is incorrect when considering the semantic relationship between this column and the `city` column in the same table. In this paper, we focus on predictions that LLMs still struggle with and diagnose these errors, which remain poorly understood in the existing literature, using different LLM-based experimental setups.

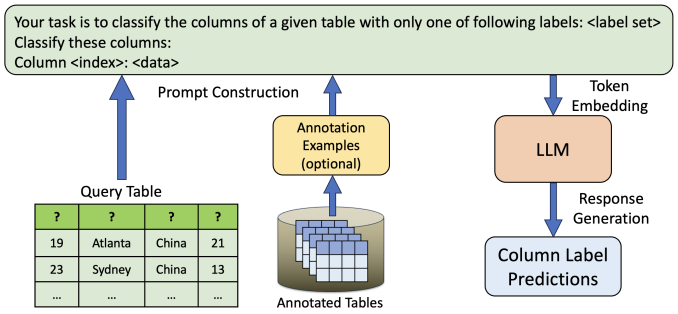


Figure 1: General pipeline of LLM-based CTA approaches. A task prompt is constructed from the query table, optionally with retrieved annotation examples. The prompt is then tokenized, mapped into token embeddings, and used by the LLM to generate responses containing column label predictions.

The standard response to persistent errors is to seek larger models, more data, and task-specific optimization. In this work, we examine these recurring errors through a different lens: rather than proposing another method to boost prediction performance, we take a step back and ask *where* the remaining errors come from. This question matters in practice because different sources of confusion call for different interventions. An error caused by a biased embedding representation cannot be fixed by better context retrieval; similarly, an error caused by incomplete retrieval is unlikely to be resolved by changes applied to the label decision space.

Figure 1 illustrates the general pipeline of LLM-based CTA approaches. (1) The predefined semantic label candidates and the query table are incorporated into a task prompt during **Prompt Construction**. Additionally, previously annotated columns and tables, either randomly selected or retrieved based on their similarity

Proceedings of the VLDB Endowment. ISSN 2150-8097.

to the query table, can be provided as contextual examples within the prompt to assist the LLM during inference. (2) The prompt is tokenized and mapped into token embeddings during **Token Embedding** for LLM inference. (3) The LLM then outputs a prediction for each column in the query table during **Response Generation**.

Once the prompt is transformed into embeddings, response generation proceeds through autoregressive decoding. In this work, we diagnose LLM failures by focusing on the upstream stages prior to response generation, rather than on the internal decoding dynamics of the LLM. Specifically, we examine LLM errors in CTA from three diagnostic perspectives: (1) **representation bias**: the column, when embedded into a high-dimensional semantic space, is already closer to clusters of columns belonging to other labels before retrieval occurs; (2) **retrieval incompleteness**: the correct label is absent from, or not sufficiently represented in, the retrieved context used to support the task; and (3) **decision-space complexity**: the model may fail because the final CTA prediction is made over a large candidate label set, making label selection itself difficult. Our analysis is empirical and diagnostic: rather than proposing a new training algorithm or architecture for LLM-based CTA, we provide structured analysis of these three perspectives across LLM-based CTA methods. Representation bias focuses on semantic representations formed during the token embedding stage, whereas retrieval incompleteness and decision-space complexity concern prompt construction. For retrieval incompleteness, we examine whether the retrieved context provides sufficient label coverage for the query columns. For decision-space complexity, we use probing to study how restricting the candidate label set in the prompt affects prediction behavior.

We view this work as an initial case study for understanding failures that can arise when applying prompting-based LLM methods to CTA. We hope this case study encourages future research on LLM-based CTA to address these previously overlooked perspectives on error attribution. Our contributions are:

- We show that structured and persistent LLM errors exist in CTA, with many concentrated in a small set of semantic confusions that recur across different LLM-based methods.
- We show that many universally missed cases are already closer to incorrect-label clusters than to their ground-truth clusters in the semantic embedding space, indicating that representational ambiguity may contribute to LLM prediction errors.
- We show that the quality of example retrieval influences CTA results: even a simple majority vote over retrieved examples outperforms zero-shot LLM prompting, and downstream performance is substantially lower when the retrieved tables do not cover all ground-truth labels.
- We introduce a binary-choice probing setup for analyzing LLM prediction errors under reduced label decision space and controlled support, showing that some errors are resolved once the candidate label set is narrowed, whereas others remain unresolved even with retrieved examples and additional table context.

The rest of the paper is organized as follows. Section 2 reviews related work on LLM-based CTA. Section 3 describes the experimental setup and diagnostic framework. Section 4 analyzes the structure of LLM errors across methods. Sections 5–7 investigate LLM errors in CTA from the perspectives of representation, retrieval, and decision space, respectively. Section 8 discusses the implications for future research on LLM-based CTA.

2 RELATED WORK

LLMs are a useful tool and a popular paradigm for solving a wide range of tasks. To adapt LLMs for CTA, the task needs to be presented in a form that the LLM can understand and follow. Prompt engineering therefore plays an important role in utilizing LLMs for CTA, where the model is presented with a prompt asking it to assign a label from a predefined label set based on the table content. Korini and Bizer [13] showed that zero-shot prompting, in which the LLM is given only the task prompt without additional examples, can already be effective when the prompt explicitly encodes table structure, column roles, and output constraints. CHORUS [12] further demonstrated that general-purpose LLMs, when paired with output constraints and anchoring, can support CTA within a broader data discovery framework. Subsequent work moved from single-prompt approaches toward more structured prompt construction. ArcheType [5] provides an LLM-based CTA framework with a carefully engineered prompting pipeline. CENTS [22] takes a cost-effective approach for prompt construction, selecting only the most relevant table cells and reducing inference cost.

More recent work has focused on providing additional information to assist LLMs during CTA inference. Korini and Bizer [14] additionally enriched prompts with label definitions and other explanatory cues to disambiguate the semantics of candidate labels. Retrieval-augmented generation (RAG), where external "expert" information is retrieved and provided to the model, is another approach that focuses on improving the evidence supplied to the LLM. RACOON [21] augments prompts with knowledge-graph-derived entity context, and Babamahmoudi et al. [1] showed that RAG-based methods that retrieve semantically similar columns or tables as examples can substantially improve prediction quality in CTA.

A separate line of work explores fine-tuning LLMs for specific downstream tasks. Table-GPT [16] applies instruction tuning to adapt LLMs for various table understanding tasks. PACTA [17] augments the training set with additional prompt variants to reduce LLM sensitivity to prompt phrasing. ZTab [8] generates domain-specific pseudo-tables and fine-tunes the LLM on this synthetic data to improve CTA performance.

Overall, prior work primarily seeks to improve CTA accuracy through better prompts, richer context, post-processing, or model adaptation. Our work differs in that it does not aim to find an optimal strategy to improve LLM-based CTA methods; instead, it focuses on error attribution by diagnosing LLM failures in CTA through the lens of different aspects of the LLM-based CTA pipeline.

3 EXPERIMENTAL SETUP

3.1 Task and Dataset

Definition 3.1 (Column Type Annotation). Given a table T with columns $c_1, \ldots, c_n$ and a predefined semantic label set $\mathcal{L}$, column type annotation (CTA) refers to assigning each column c_i a semantic type $y_i \in \mathcal{L}$ to represent the semantics of values in c_i.

We consider two CTA settings. In the *single-column* (SC) setting, the model observes only the cell values of the target column c_i. In the *multi-column* (MC) setting, the model additionally observes the values of all other columns $c_1, \ldots, c_n$ in the same table. Following prior LLM-based CTA studies [1, 13, 14], we assume that column names are not available.

We evaluate on a curated sample [1] of the VizNet dataset [9], selected to minimize overlap between training and test tables. The test set comprises 300 columns from 90 tables across $|\mathcal{L}| = 32$ semantic types. The label distribution is imbalanced: frequent types such as `status` and `year` appear across many tables, while rare types such as `album` and `symbol` occur only once or twice.

We do not consider fine-tuning any of the LLMs in this work, as doing so would adapt the models to the target dataset distribution through additional training. Better performance on the target domain, however, does not always transfer to cross-domain generalization [15]. In this paper, we focus on treating LLMs as general-purpose prompting models and examining behaviors that are intrinsic to prompting. Therefore, the training set serves exclusively as a retrieval pool for in-context annotation examples. For the SC setting, 2,000 labeled columns are available for retrieval. For the MC setting, retrieval is performed at the table level; a separate subset of 300 tables (1,000 columns total) is used accordingly.

3.2 Diagnostic Framework

We analyze all seven LLM-based CTA methods from Babamahmoudi et al. [1], as these methods represent different design settings within the LLM-based CTA pipeline. We also replace the original LLM with a more recent one. In our experiments, all methods use the same backbone LLM and text embedding model (`gpt-5.4-mini`[1] and `text-embedding-ada-002`[2], respectively), differing only in how much context is provided and how examples are selected:

- **SC-ZS**: zero-shot prompting with only one column at a time.
- **SC-FS**: five-shot single-column prompting with randomly sampled labeled columns as annotation examples.
- **SC-RAG**: retrieval-augmented single-column prompting using the top-five most similar labeled columns from the retrieval corpus, retrieved by embedding cosine similarity.
- **MC-ZS / MC-FS / MC-RAG**: multi-column variants of the above, where the model observes all columns in the same table and annotation examples are provided at the table level.
- **MC-CoT**: **MC-RAG** extended with an explicit chain-of-thought reasoning prompt.

We treat the predictions of these seven methods as fixed artifacts and conduct post-hoc diagnostic analysis on the errors in the predictions from the following three perspectives, each corresponding to a distinct aspect of the LLM-based CTA pipeline:

(1) **Representation bias.** The cell values of a column are embedded into a high-dimensional semantic space. Errors can occur when the embedding model places a column closer to columns of a different semantic type, thereby biasing both retrieval and prediction before later stages take effect.

[1] https://developers.openai.com/api/docs/models/gpt-5.4-mini
[2] https://developers.openai.com/api/docs/models/text-embedding-ada-002

Table 1: Error Counts and Error Rates

Method	# Error	Error rate
SC-ZS	91	30.3%
SC-FS	76	25.3%
SC-RAG	57	19.0%
MC-ZS	78	26.0%
MC-FS	66	22.0%
MC-RAG	32	10.7%
MC-CoT	45	15.0%

(2) **Retrieval incompleteness.** Annotation examples are selected based on embedding similarity. Errors can occur when the correct semantic label is absent from, or underrepresented in, the retrieved context, depriving the LLM of the signal it needs to make a correct prediction.

(3) **Decision-space complexity.** The LLM generates a final prediction conditioned on the provided context. Errors can occur when the final CTA prediction must be made over a large candidate label set, making label selection within that set difficult.

4 LLM PREDICTION ERROR ANALYSIS

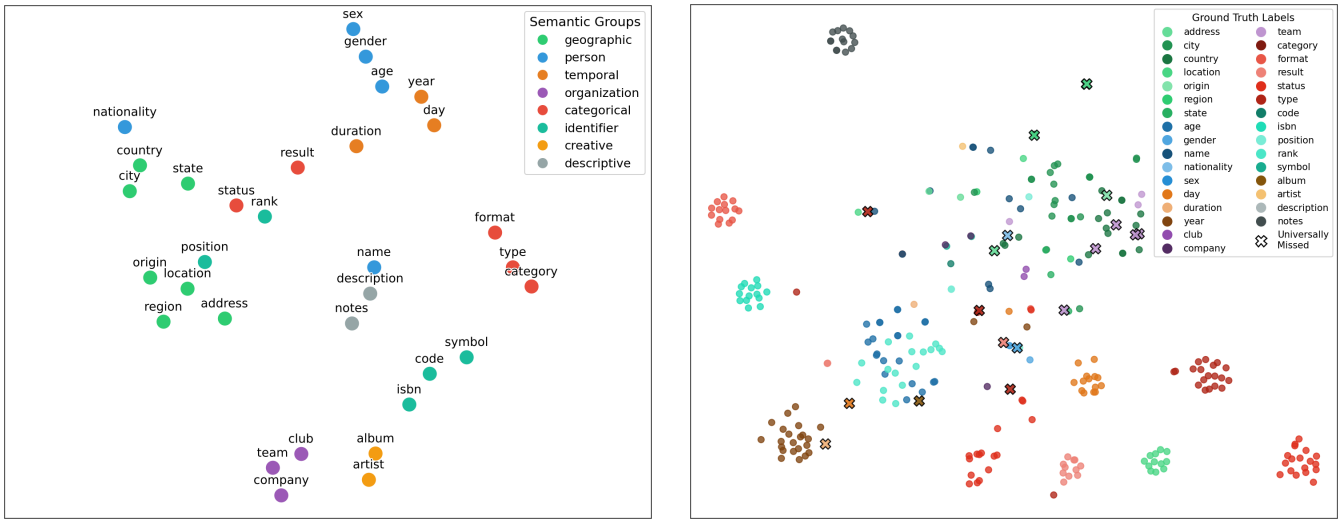
We ran the seven LLM-based CTA methods on the 300 test columns described in Section 3.1 and examined how the observed errors are distributed across methods and labels. Since CTA can be framed as a label-selection task in which the model is presented with a multiple-choice question to assign each column a semantic label from a predefined set, we analyzed the model responses directly to characterize the error distribution.

4.1 Cross-Method Errors

Table 1 shows that the seven methods vary substantially in overall error rate, ranging from 30.3% for **SC-ZS** to 10.7% for **MC-RAG**. Table 2 shows that all seven LLM methods successfully annotated 168 test columns (56.0%), whereas the remaining 132 columns (44.0%) were misclassified by at least one method. Among the error cases, 19 (6.3%) are predicted incorrectly by all seven methods. Within these universally missed cases, 10 exhibit unanimous wrong predictions, meaning every LLM method predicts the same incorrect label. This cross-method consistency suggests that these errors are unlikely to stem from random variability alone, and may instead reflect systematic confusions that persist across different LLM methods.

Table 2: Distribution of test columns by failure frequency.

# Methods Failed	# Test Columns
0	168
1	32
2	30
3	15
4	15
5	11
6	10
7	19



(a) Label-name embeddings **(b) Test-column embeddings**

Figure 2: t-SNE projections of the semantic embedding space.

4.2 Recurring Confusion Patterns

The labeling errors are not uniformly distributed; they are concentrated in a small set of recurring label confusions. In these test cases where the LLM fails, some involve labels that are semantically similar by name, such as `type` → `format` ($n = 17$), whereas others appear to reflect ambiguity in column values, such as `rank` → `age` ($n = 12$) and `team` → `country` ($n = 6$). This pattern suggests that LLM prediction errors are structured around stable semantic confusions rather than random label choices. Section 5 examines these confusions from the perspective of column representation, where semantic similarity is reflected as proximity in the embedding space.

5 SEMANTIC REPRESENTATION ANALYSIS

In our diagnostic framework, we begin with the pipeline stage closest to LLM inference. In this section, we focus on semantic representations and investigate whether label confusion and prediction errors are already reflected in the representation space, where columns associated with different labels are positioned close to one another and are therefore difficult to distinguish prior to LLM inference and response generation. Because `gpt-5.4-mini` is closed-source and its internal token embeddings are inaccessible, we use `text-embedding-ada-002` as an alternative source of semantic representations for analysis.

We embed the labels and column values into a high-dimensional semantic space and examine the representation structure at two levels: the clustering structure of label names, and the distribution of test column embeddings. We visualize the embedding space in Figure 2 using t-SNE [19], where spatial proximity reflects the semantic similarity assigned by the embedding model.

For the 19 universally missed cases, we further compare each column against its ground-truth label cluster and its dominant wrong-label cluster in the embedding space using leave-one-out centroids. This analysis applies to 15 test columns. The remaining 4 are excluded because their ground-truth labels appear only once in the test set, preventing the construction of leave-one-out centroids.

Label clusters. Figure 2a shows a two-dimensional projection of the 32 label-name embeddings, with labels colored by broad manually defined semantic groups. Several frequently confused label pairs occupy nearby regions, notably `type`/`format` and `sex`/`gender`. We interpret this as evidence that semantically similar labels tend to cluster in the embedding space, which may increase label competition when the LLM selects a label for CTA.

Column embedding distribution. Figure 2b shifts the focus from label names to column values. Some labels have column embeddings that cluster within clear boundaries, while others share the same region with columns of different labels. These proximities reflect similarities in column values. For example, `rank` and `age` columns may both contain numeric values, while some `team` columns contain values that resemble those of `country` columns. Notably, most universally missed cases, marked by `x`, do not gather into a single isolated region; instead, many appear in mixed or boundary areas where neighboring semantic types overlap. This is particularly visible in geographic, person-related, and organization-related portions of the space, where several such cases lie closer to adjacent label regions than to the interior of their ground-truth cluster.

To quantify this pattern, we compared each universally missed case with its ground-truth column-cluster centroid and its dominant wrong-label centroid in the embedding space. This centroid analysis characterizes how a query column aligns with label clusters in the embedding space, and it is distinct from the nearest-neighbor retrieval in RAG, which we analyze separately in Section 6. Among the 15 universally missed cases we analyzed, 9 are already closer in the embedding space to the wrong-label centroid than to the ground-truth centroid. This suggests that some prediction errors are associated with column semantic representations that already lean toward the wrong label before any retrieval or final label prediction.

Table 3: Single-Column micro F1 and macro F1 scores.

Method	Micro F1	Macro F1
SC-ZS	69.7%	43.9%
RO-Vote	74.3%	44.8%
RO-Rank	75.7%	40.9%
SC-FS	74.7%	51.1%
SC-RAG	**81.0%**	**58.5%**

The remaining 6 cases are closer to their ground-truth clusters yet are predicted incorrectly by every method. These cases are especially important because their failures are not explained by semantic representation alone and therefore call for other explanations. Overall, the analysis suggests a distinction between two groups of prediction errors: those that already show a representational shift toward the wrong label, and those that remain aligned with their ground-truth labels but still fail. These failures are not explained by cluster centroid representation analysis alone; the local neighbors of these cases play a role even when their embeddings lie closer to their own ground-truth centroid, especially since the RAG-based methods rely on local nearest neighbors rather than cluster centroids. Beyond this representation view, their failures may also reflect retrieval limitations or label decision difficulty.

6 RETRIEVAL AND LABEL COVERAGE

As shown in Table 1, RAG-augmented LLM methods achieve the best performance among the evaluated methods, demonstrating that carefully selected additional annotation examples can benefit LLM inference. In this section, we examine retrieval more closely, focusing on how the quality and label coverage of retrieved examples relate to LLM prediction performance. The RAG-based methods evaluated in Section 4 operate under two different prompting settings: one constructs task prompts using a single target column at a time, whereas the other provides the entire table and annotates multiple columns within a single inference pass. Because these settings also require retrieval at different levels of granularity, we analyze them separately in this section.

6.1 Single-Column Retrieval

Setup. We first analyze the quality and label coverage of retrieved examples in the single-column prompting setting of LLM-based CTA, where each task prompt is constructed using only a single query column. Consequently, retrieval in SC-RAG is also performed at the column level, and annotated columns are selected as contextual examples based on their similarity to the query column. To isolate retrieval quality from downstream LLM inference, we introduce two retrieval-only baselines in addition to SC-RAG. RO-Vote retrieves the top-5 most similar labeled training columns based on cosine similarity of column embeddings and assigns the plurality label by majority vote, without invoking any LLM. RO-Rank applies the same retrieval procedure but weights the retrieved labels by reciprocal rank. We further include the non-RAG LLM methods as additional baselines and compare all methods against SC-RAG.

Results. Table 3 presents the single-column comparison. Because the label distribution is imbalanced, we report micro-F1, which aggregates prediction results across all test columns and therefore emphasizes overall performance, and macro-F1, which computes the F1 score independently for each class and averages them equally to reflect performance across labels regardless of their frequency. Retrieval-only voting (RO-Vote, micro F1 = 74.3%) already outperforms zero-shot prompting (SC-ZS, 69.7%), confirming that retrieved examples already carry strong signal before the final label prediction is made. Reciprocal-rank weighting (RO-Rank, micro F1 = 75.7%) improves the micro F1 score further by assigning greater weight to higher-ranked retrieved examples. SC-FS (micro F1 = 74.7%) uses five randomly selected training samples as examples, whereas SC-RAG (micro F1 = 81.0%) provides the same 5 retrieved training samples to the LLM as the RO methods. The performance gap between SC-RAG and the RO baselines therefore suggests that the additional performance improvement stems from the LLM inference stage rather than retrieval alone.

However, the retrieval-only baselines do not improve macro F1. RO-Vote reaches 44.8%, which is less than one point higher than SC-ZS, while RO-Rank drops to 40.9%, as both baselines are biased toward frequently retrieved labels. By contrast, the LLM-based methods provide an advantage under imbalanced label distributions by recovering less frequent labels during inference, even when those labels are underrepresented in the retrieved examples.

At the individual-case level, RO-Vote and SC-RAG are both correct on 207 of the 300 test columns. SC-RAG correctly predicts an additional 36 cases where RO-Vote fails, while RO-Vote recovers 16 cases missed by SC-RAG. The disagreement cases show that in some instances retrieval evidence alone is stronger than the final LLM prediction. An example is the type label, where RO-Vote reaches 76% accuracy whereas SC-RAG drops to 48%, suggesting that the final LLM prediction can override otherwise useful retrieval evidence.

6.2 Multi-Column Retrieval

Setup. We now investigate the relationship between label coverage in retrieved examples and LLM prediction performance in CTA within MC-RAG. In contrast to the single-column prompting setting, MC-RAG operates at the table level, where multiple columns from a query table are serialized into a single string to construct the task prompt. Retrieval is therefore also performed at the table level: MC-RAG searches over previously annotated tables and retrieves the top-5 most similar tables to the query table based on cosine similarity. The LLM then conditions on these retrieved tables as in-context examples to simultaneously predict semantic labels for all columns in the query table. As shown in Table 1, multi-column methods outperform their single-column counterparts, suggesting that table-level context provides useful signals for label prediction.

Unlike the single-column retrieval setting in SC-RAG, retrieved tables in MC-RAG are not guaranteed to contain columns that directly correspond to those in the query table, meaning that retrieval quality cannot be assessed through per-column alignment. The diagnostic question shifts to whether the retrieved tables provide sufficient label coverage for the query table. To measure this, we aggregate labels from the top-5 retrieved tables using reciprocal-rank weights. For each query table T, label coverage is defined as:

$$\text{Coverage}(T) = \frac{|\mathcal{L}^*(T) \cap \hat{\mathcal{L}}(T)|}{|\mathcal{L}^*(T)|} \tag{1}$$

where $\mathcal{L}^*(T)$ is the ground-truth label set for T, and $\hat{\mathcal{L}}(T)$ denotes the aggregated retrieved label set truncated to size $|\mathcal{L}^*(T)|$.

Table 4: MC-RAG performance by retrieval completeness.

Label Coverage	Table Share	Column-Level Micro F1	Column-Level Macro F1
Complete	65.6%	97.0%	89.6%
Incomplete	34.4%	74.3%	51.7%

Results. Among the 90 test tables, the average label coverage score is 79.2%, and 59 tables achieve exact label-set match, accounting for 65.6% of the test set. We treat these tables as having complete label coverage by the retrieved tables, while the remaining tables have incomplete label coverage. Table 4 summarizes MC-RAG performance stratified by retrieval completeness. The performance gap between the two groups is substantial: when the retrieved tables cover all ground-truth labels, MC-RAG achieves 97.0% micro F1 and 89.6% macro F1; when coverage is incomplete, both metrics drop sharply to 74.3% micro F1 and 51.7% macro F1. The latter is on par with MC-FS, where the example tables are randomly selected rather than retrieved by similarity for each query. These results highlight retrieval quality as a major factor in multi-column RAG: when the retrieved tables do not cover the full ground-truth label set, downstream LLM inference performance degrades substantially. The disproportionately larger drop in macro F1 than in micro F1 further suggests that incomplete label coverage does not affect all labels equally under the imbalanced label distribution.

7 BINARY-CHOICE PROBING

The previous sections showed that LLM prediction errors in CTA are associated with semantic overlap between labels and with incomplete and misaligned label coverage in the retrieved context. In this section, we take a complementary perspective by examining whether these errors persist when the label decision space is reduced to a binary choice.

7.1 Setup

When a CTA prediction is made over a large set of candidate labels, some errors may arise because of the size of the label space. We therefore conducted a binary-choice probing study in which the LLM is prompted to choose between exactly two candidate labels for the target column. This setup allows us to examine which errors are resolved once the candidate label set is narrowed, and which errors remain when the dominant confusion label is the only alternative. The goal is diagnostic rather than comparative: the probing conditions are designed to reveal which errors become recoverable when label competition is reduced and with additional supporting context, rather than to identify an optimal prompting strategy. Consequently, the probing conditions should not be interpreted as an optimization of the LLM methods evaluated in Section 4.

Test cases and distractor labels. We probed all 132 columns that were predicted incorrectly by at least one of the seven LLM methods in Section 4. These 132 cases span 75 distinct tables; 19 are universally missed cases (i.e., wrong across all 7 methods). For each case, we identified the *dominant confusion label*: the wrong prediction appearing most frequently across the seven methods. The dominant confusion label serves as the binary-choice distractor provided to the LLM along with the ground-truth label.

Table 5: Binary-choice prediction behavior on 132 error cases. Error cases are compared against predictions before label space reduction and classified as: C1: unchanged, correct; C2: changed to correct; C3: unchanged, incorrect; C4: correct to distractor; C5: changed between incorrect labels. The last column reports the percentage of positive outcomes (C1+C2)

Condition	Additional Context	C1	C2	C3	C4	C5	P
SC-ZS-bin	None	21	34	44	20	13	41.7%
SC-RAG-bin	5 retrieved columns	68	24	25	7	8	69.7%
MC-ZS-bin	Table + non-target GT labels	45	25	41	9	12	53.0%
MC-RAG-bin	MC-ZS-bin + 5 retrieved tables	94	6	25	6	1	75.8%

Probing conditions. We designed experiments with four probing conditions that vary in the auxiliary information provided to the LLM, as detailed in Table 5. Single-column probing conditions probe one target column at a time. Multi-column probing conditions present the full table in the prompt but probe one target column at a time, with non-target columns and their ground-truth labels provided as auxiliary context. We use this oracle in-table label context to isolate ambiguity at the target column from the uncertainty of the surrounding columns; the multi-column probing conditions should therefore be interpreted as diagnostic probes rather than as an extension of the multi-column CTA methods in Section 4. They are not intended to represent a practical CTA workflow because real systems would not have access to ground-truth oracles at inference time. In the two RAG-based probing conditions, the retrieved examples are provided with their ground-truth labels as reference evidence, while the target-column prediction is restricted to a binary choice between the ground-truth label and the distractor.

7.2 Probing Outcomes

For each probing condition, we compare its binary-choice prediction against the corresponding prediction before the label space was reduced in Section 4. The single-column probing conditions preserve the same context structure as their counterparts in Section 4, while the multi-column probing conditions additionally provide ground-truth labels for the non-target columns in the same table. Each probe case is classified into one of five outcome categories based on whether the binary-choice prediction agrees with the prediction before label space reduction and whether it is correct.

Table 5 summarizes the prediction behavior across all 132 probe cases. The cases in C1 and C2 represent positive outcomes: C1 cases remain correct, whereas C2 cases are recovered by the corresponding probing condition as the prediction changes from a previously incorrect label to the correct label under binary choice. The cases in C3, where the prediction remains the same distractor label, remain unresolved even after the candidate label set is reduced to the ground-truth label and the distractor label. Because the binary-choice probing is restricted to the ground-truth label and the distractor, C4 and C5 distinguish two forms of negative change: C4

represents cases that regress from a previously correct prediction to the distractor label, whereas C5 captures cases where the reduced label space redirects an existing error toward the distractor rather than recovering the correct label.

Because the four original LLM methods in Section 4 make different numbers of errors across the 132 probe cases, which represent the union of all error cases, the outcome counts should be interpreted relative to each corresponding method. Binary-choice probing recovers 34 of the 91 original SC-ZS errors, 24 of the 57 original SC-RAG errors, 25 of the 78 original MC-ZS errors, and 6 of the 32 original MC-RAG errors. In contrast, the same distractor label remains preferred in 44, 25, 41, and 25 cases, respectively. Thus, although MC-RAG produces fewer errors than the other methods in Section 4, most of these errors are not resolved by the reduced label space and oracle in-table label context provided in this probe.

The C4 and C5 cases further show that a reduced label space does not always stabilize predictions. C4 regressions are most common in SC-ZS-bin (20 cases) and lower in SC-RAG-bin, MC-ZS-bin, and MC-RAG-bin (7, 9, and 6 cases, respectively), suggesting that auxiliary support reduces, but does not eliminate, cases where binary probing disrupts an originally correct prediction. C5 appears in 13, 8, 12, and 1 cases for SC-ZS-bin, SC-RAG-bin, MC-ZS-bin, and MC-RAG-bin, respectively, showing that some wrong predictions are redirected toward the distractor instead of being corrected.

Across all binary decisions, the LLM always selects one of the two candidate labels, producing no out-of-given-label predictions in any condition. Since each probing task presents exactly two candidate labels, a uniform random binary choice would achieve 50% accuracy in expectation. Against this random-guessing reference point, SC-ZS-bin reaches 41.7% (C1+C2), suggesting that under this probing setup the model still tends to favor the dominant distractor over the ground-truth label even after the decision space is reduced. By contrast, both retrieval-augmented conditions substantially exceed this random-guessing reference point, suggesting that retrieved examples provide disambiguating evidence that guides the LLM toward the correct label.

7.3 Confusion-Pair Error Profiles

To understand what drives the binary-choice outcomes, we examine how specific confusion pairs, each formed by the ground-truth label and the distractor, respond to the probing conditions. We present binary-choice accuracy for representative pairs in Table 6. These results exemplify three distinct error profiles: label-reduction-responsive patterns, context-dependent patterns, and persistent or unstable patterns.

Label-reduction-responsive patterns. Some original errors become recoverable in low-context probing conditions once the label decision space is reduced to the ground-truth label and the distractor. For `status` → `notes` ($n = 3$), both zero-shot probing conditions recover all three originally incorrect cases (C2=3 in SC-ZS-bin and MC-ZS-bin), while the two RAG conditions are already correct before probing (C1=3). For `rank` → `code` ($n = 8$), SC-ZS-bin and SC-RAG-bin each recover four original errors in C2, while MC-RAG-bin was already correct on all eight cases before binary-choice probing. These patterns suggest that some errors become recoverable when the label space is reduced, particularly in settings with limited context, indicating that label-space reduction itself can contribute to error recovery without requiring richer supporting context.

Table 6: Per-pair binary-choice accuracy for representative confusion pairs.

GT → Distractor	n	SC-ZS -bin	SC-RAG -bin	MC-ZS -bin	MC-RAG -bin
status → notes	3	100.0%	100.0%	100.0%	100.0%
rank → code	8	62.5%	87.5%	75.0%	100.0%
type → format	17	23.5%	100.0%	5.9%	100.0%
location → status	12	0.0%	100.0%	100.0%	100.0%
type → category	7	42.9%	57.1%	42.9%	42.9%
team → country	6	0.0%	0.0%	0.0%	0.0%

Context-dependent patterns. Another profile consists of pairs that become reliably distinguishable only when additional context is provided, whether in the form of retrieved examples or oracle in-table label context for other columns. For `type` → `format` ($n = 17$), SC-ZS-bin reaches only 23.5% and MC-ZS-bin drops further to 5.9%, yet both retrieval-augmented conditions achieve 100%. In this pair, the SC-RAG-bin accuracy comes from both originally correct cases (C1=8) and cases corrected by probing (C2=9), while MC-RAG-bin is already correct for all cases before probing (C1=17). For `location` → `status` ($n = 12$), SC-ZS-bin remains at 0%, whereas SC-RAG-bin, MC-ZS-bin, and MC-RAG-bin all reach 100%; here, retrieved examples or oracle table context are both sufficient to make the pair reliably distinguishable. These cases show that reducing the label decision space is not always sufficient. The useful form of context also differs across confusion pairs: `type` → `format` is resolved by retrieved examples but not by oracle table context alone, whereas `location` → `status` is already resolved when either retrieved examples or oracle in-table label context is provided.

Persistent or unstable patterns. Some confusion pairs remain difficult across all probing conditions. `team` → `country` ($n = 6$) remains at 0% across all four conditions. This pair is dominated by C3 in the first three conditions, while MC-RAG-bin additionally contains two C4 regressions, where cases that were originally correct become incorrect under binary probing. `type` → `category` ($n = 7$) stays between 42.9% and 57.1%, on par with the 50% random-guessing reference point, and shows a mixture of C1–C5 outcomes rather than a stable recovery pattern. These patterns suggest that neither reducing the label decision space nor adding auxiliary context is sufficient to reliably recover the correct label for these pairs.

At the hardest end of the persistent or unstable patterns, the 19 universally missed cases from Section 4 are especially resistant in this probing setup. Because all seven methods predict these cases incorrectly before label space reduction, C1 is zero for all conditions; binary-choice accuracy therefore reflects only C2, genuine recovery against the distractor. This rate reaches only 15.8% under both zero-shot conditions, 10.5% under SC-RAG-bin, and 21.1% under MC-RAG-bin. This suggests that the hardest subset contains errors that remain resistant under all tested reduced-label-space conditions, even when retrieved examples and oracle table context are provided.

8 DISCUSSION

LLM prediction errors in CTA should not be treated as a single failure mode. We conducted a diagnostic case study of error cases through the lens of three different perspectives: error cases that are already shifted toward wrong-label regions in representation space, error cases that are constrained by the quality of retrieved examples provided to the LLM, and error cases that remain difficult even after the label decision space is reduced and auxiliary context is provided. These findings argue against a single generic fix and instead call for case-specific interventions.

The semantic representation analysis highlights one important subset of the problem at the representation level itself. A substantial portion of the hardest cases is already closer to the wrong-label region than to its ground-truth region in embedding space, which suggests that later prompt or retrieval strategies may be insufficient on their own. For these cases, more promising directions include encoders with contrastive learning objectives, or embedding-aware re-ranking designed to improve separation between semantically adjacent labels. This line of work predates LLM-based CTA [20], but it warrants renewed attention in the current LLM era.

The retrieval analysis shows that retrieved examples carry predictive signal beyond merely serving as in-context examples, and performance is often closely tied to the quality of retrieved examples made available to the model. In the single-column setting, retrieval-only voting already outperforms zero-shot prompting, indicating that useful signal is present in the retrieved examples themselves. In the multi-column setting, the sharp drop under incomplete label coverage shows that table-level similarity alone does not guarantee sufficient label coverage. This calls for systems that optimize not only semantic similarity, but also the composition of the retrieved label evidence, for example through hybrid retrieval or retrieval-aware re-ranking strategies.

The binary-choice probing results add another perspective to the preceding analyses. Many prediction errors are corrected once the final decision is reduced to a binary choice between the ground-truth label and a distractor, but the gains remain uneven across cases. A notable pattern emerges for confusion pairs such as `type` → `format`: without retrieval, accuracy remains below chance even in the binary-choice setting (23.5% under `SC-ZS-bin` and 5.9% under `MC-ZS-bin`), while both retrieval-augmented conditions reach 100%. This suggests that for some semantically adjacent label pairs, the model shows a strong preference for the distractor that is not corrected by decision-space reduction alone, but can be reversed when retrieval evidence is provided. By contrast, pairs such as `rank` → `code` already reach above-chance accuracy in the lowest-context condition (62.5% under `SC-ZS-bin`), suggesting that their failures in the full-label setting are more likely attributable to label competition in a large decision space than to persistent ambiguity between the ground-truth label and the distractor. Label-space reduction also introduces regressions in some cases: C4 in Table 5 captures predictions that were originally correct but become incorrect under binary probing, indicating that label-space reduction is more useful as a diagnostic probe than as a prompting strategy.

The universally missed cases in Section 4 remain resistant across all probing conditions, with binary-choice accuracy reaching just 21.1% under the most favorable condition, confirming that neither reducing the decision space nor enriching the context is sufficient to resolve the hardest prediction errors. These patterns suggest that future improvements should target case-specific interventions rather than generic prompting changes. For cases that remain unresolved, the LLM could withhold its prediction when confidence is low and escalate them to human judgment, particularly when the remaining ambiguity stems from label distinctions that cannot be resolved from the observed table content alone and where consistent labeling insights from human annotators could serve as additional context to guide future LLM predictions.

In this case study, our analysis uses a curated sample [1] of VizNet [9]. Future work could apply the same diagnostic framework to additional datasets, such as WikiTables [4] and GitTables [10], as well as to additional model families, particularly open-source LLMs whose internal representations can be inspected directly. Such studies would help assess whether the findings from this case study generalize to other settings and whether new error patterns emerge. Beyond CTA, such diagnostics could also be extended to LLM-based methods for other semantic table interpretation tasks, such as entity disambiguation in tables [2] and table-to-knowledge-graph matching tasks in the SemTab challenge [7, 18].

9 CONCLUSION

We present a diagnostic study of prediction errors in LLM-based CTA from the perspectives of representation, retrieval, and decision space. Our analysis shows that these errors are structured rather than random, and that they display different patterns under these three perspectives. Representation analysis reveals that some cases already lie closer to wrong-label regions in embedding space. Retrieval analysis shows a strong association between performance and the quality of the retrieved label evidence. Reduced-choice probing further shows that some errors are resolved when the label decision space is narrowed, whereas others remain unresolved even with retrieved examples and additional table context.

These findings suggest that improving LLM-based CTA will require targeted interventions tailored to error profiles rather than a generic fix. As an initial case study, our analysis is intended to motivate broader diagnostic evaluations across additional datasets, models, and embedding spaces, as well as future work on solutions that address different patterns of LLM errors in CTA.

ACKNOWLEDGMENTS

This work is supported by an NSERC Discovery Grant.

REFERENCES

[1] Amir Babamahmoudi, Davood Rafiei, and Mario A Nascimento. 2025. Improving Column Type Annotation Using Large Language Models. In *VLDB 2025 Workshop on Tabular Data Analysis*.

[2] Federico Belotti, Marco Cremaschi, Fabio Dadda, Roberto Avogadro, and Matteo Palmonari. 2026. How good are LLMs in disambiguating entities in tabular data? A comprehensive study. *Data & Knowledge Engineering* 164 (2026), 102596.

[3] Marco Cremaschi, Blerina Spahiu, Matteo Palmonari, and Ernesto Jiménez-Ruiz. 2024. Survey on Semantic Interpretation of Tabular Data: Challenges and Directions. arXiv:2411.11891

[4] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proc. VLDB Endow.* 14, 3 (2020), 307–319.

[5] Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. 2024. ArcheType: A Novel Framework for Open-Source Column Type Annotation Using Large Language Models. *Proc. VLDB Endow.* 17, 9 (2024), 2279–2292.

[6] Juliana Freire, Grace Fan, Benjamin Feuer, Christos Koutras, Yurong Liu, Eduardo Pena, Aécio Santos, Cláudio T Silva, and Eden Wu. 2025. Large Language Models for Data Discovery and Integration: Challenges and Opportunities. *IEEE Data Engineering Bulletin* (2025).

[7] Oktie Hassanzadeh, Marco Cremaschi, Fabio D'Adda, Fidèl Jiomekong Azanzi, Jean Petit Bikim, and Ernesto Jiménez-Ruiz. 2025. Results of SemTab 2025. In *20th ISWC Workshop on Ontology Matching*. 216–220.

[8] Ehsan Hoseinzade and Ke Wang. 2026. ZTab: Domain-based Zero-shot Annotation for Table Columns. arXiv:2603.11436

[9] Kevin Hu, Snehalkumar Neil S. Gaikwad, Madelon Hulsebos, Michiel A. Bakker, Emanuel Zgraggen, César Hidalgo, Tim Kraska, Guoliang Li, Arvind Satyanarayan, and Çağatay Demiralp. 2019. VizNet: Towards A Large-Scale Visualization Learning and Benchmarking Repository. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. 1–12.

[10] Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. GitTables: A Large-Scale Corpus of Relational Tables. *Proc. ACM Manag. Data* 1, 1, Article 30 (May 2023), 17 pages.

[11] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *Proceedings of the Conference on Knowledge Discovery & Data Mining (KDD)*. 1500–1508.

[12] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation Models for Unified Data Discovery and Exploration. *Proc. VLDB Endow.* 17, 8 (2024), 2104–2114.

[13] Keti Korini and Christian Bizer. 2023. Column Type Annotation using ChatGPT. In *VLDB 2023 Workshop on Tabular Data Analysis*.

[14] Keti Korini and Christian Bizer. 2025. Evaluating Knowledge Generation and Self-Refinement Strategies for LLM-Based Column Type Annotation. In *Proceedings of the European Conference on Advances in Databases and Information Systems (ADBIS)*. 111–127.

[15] Sven Langenecker, Christoph Sturm, Christian Schalles Schalles, and Carsten Binnig. 2023. Steered Training Data Generation for Learned Semantic Type Detection. *Proc. ACM Manag. Data* 1, 2, Article 201 (June 2023), 25 pages.

[16] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2024. Table-GPT: Table Fine-tuned GPT for Diverse Table Tasks. *Proc. ACM Manag. Data* 2, 3, Article 176 (June 2024), 28 pages.

[17] Hanze Meng, Jianhao Cao, and Rachel Pottinger. 2025. Robust LLM-based Column Type Annotation via Prompt Augmentation with LoRA Tuning. arXiv:2512.22742

[18] SemTab. 2025. *SemTab 2025 - Semantic Web Challenge on Tabular Data to Knowledge Graph Matching*. https://sem-tab-challenge.github.io/2025/.

[19] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.

[20] Runhui Wang, Yuliang Li, and Jin Wang. 2023. Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integration and Preparation . In *Proceedings of International Conference on Data Engineering (ICDE)*. 1502–1515.

[21] Lindsey Linxi Wei, Guorui Xiao, and Magdalena Balazinska. 2024. RACOON: An LLM-based Framework for Retrieval-Augmented Column Type Annotation with a Knowledge Graph. arXiv:2409.14556

[22] Guorui Xiao, Dong He, Jin Wang, and Magdalena Balazinska. 2025. Cents: A Flexible and Cost-Effective Framework for LLM-Based Table Understanding. *Proc. VLDB Endow.* 18, 11 (2025), 4574–4587.