

TRACER: An Information Entropy-Grounded Framework for Efficient Large-table Semantic Annotation

Miaoqin Zheng¹, Shuangyin Li^{1*}

¹School of Computer Science, South China Normal University, Guangzhou, China
miaoqinzheng@m.scnu.edu.cn, shuangyinli@scnu.edu.cn

Abstract

Large-table semantic annotation aims to assign ontology-grounded semantic labels to columns and column pairs in large relational tables, while considering the efficiency bottleneck caused by the increasing scale of table contexts. However, as relational tables continue to grow in scale, existing methods face a critical effectiveness–efficiency trade-off: directly modeling the full table preserves rich context but introduces substantial computational overhead, while aggressive filtering may discard useful semantic cues. To address this issue, we propose **TRACER**, an information entropy-grounded framework for efficient large-table semantic annotation. TRACER ranks table rows according to ontology-related semantic uncertainty and constructs compact evidence-oriented contexts through hybrid compression, enabling a lightweight semantic reader to perform column type and column property annotation efficiently. From an information-theoretic perspective, TRACER formulates large-table compression as a conditional entropy minimization problem, providing a principled basis for preserving key semantic evidence while filtering redundant information. Experiments on several benchmarks show that TRACER achieves a strong balance between annotation effectiveness and inference efficiency compared with task-specific and large language model baselines. The source code and data are available at <https://github.com/miaoqin0402/TRACER>.

1 Introduction

Relational tables are a fundamental format for organizing structured data and are widely used in open data platforms and real-world data management systems (Cafarella et al., 2008; Lehmberg et al., 2016; Hulsebos et al., 2023a). To make such tables machine-interpretable, semantic table annotation

assigns ontology-grounded semantic labels to table elements, including semantic types for columns and semantic properties for column pairs. These annotations provide essential metadata for schema matching, dataset discovery, data integration, and knowledge graph construction (Liu et al., 2023; Zhang et al., 2020; Suhara et al., 2022).

However, as relational tables continue to grow in scale, *Large-Table* semantic annotation requires not only accurate semantic prediction but also efficient context modeling. Directly encoding the full table can preserve rich semantic evidence, but it introduces substantial computational overhead and may expose the model to redundant or noisy rows. Conversely, aggressively reducing the table context improves efficiency, but may discard discriminative cues required for ontology-grounded prediction. This makes large-table semantic annotation inherently constrained by an effectiveness–efficiency trade-off problem.

Although recent studies have recognized the importance of selecting useful table context, existing solutions remain insufficient for efficient large-table semantic annotation. Some methods serialize the whole table or a large portion of table context into pretrained language models, relying on self-attention to implicitly identify useful evidence (Deng et al., 2022; Suhara et al., 2022). This design is costly for large table and may introduce irrelevant or misleading content. Other methods attempt to reduce context by selecting relevant columns or incorporating inter-table information (Ding et al., 2025; Sun et al., 2023), which is effective for mitigating noisy column contexts or handling wide table, but they mainly focus on column-level context construction rather than row-level evidence compression. As a result, they do not directly address the substantial row redundancy commonly observed in large relational table. Simple alternatives such as truncating table contexts or selecting rows based on shallow lexical signals are

*Corresponding author.

also inadequate (Yin et al., 2020; Iida et al., 2021), because they are either insensitive to ontology labels or unable to guarantee that the retained rows contain discriminative semantic evidence.

Therefore, the task of efficient large-table semantic annotation still faces two major challenges. The first is *how to identify ontology-discriminative semantic evidence to improve annotation effectiveness*. Among the massive table contents, it should identify and retain the semantic evidence that is truly informative for ontology-grounded label annotation. The second is *how to remove repetitive, homogeneous, or noisy table content to improve inference efficiency without losing useful context*. It means that we should remove the redundant contents to reduce inference cost without discarding the useful semantic cues. These two challenges are tightly coupled and naturally form an effectiveness-efficiency trade-off. Over-compression may remove critical evidence and hurt annotation accuracy, while insufficient compression preserves excessive redundant content, leading to unnecessary computation and potential performance degradation.

To address these challenges, we propose **TRACER**, an information entropy-grounded framework for efficient large-table semantic annotation, which preserves ontology-discriminative evidence while filtering redundant table content from an information-theoretic perspective. Specifically, TRACER consists of an Entropy-Guided Evidence Ranker (EGER) that estimates ontology-related semantic uncertainty, and a Hybrid Evidence Compression and Annotation (HECA) module that constructs compact table contexts by integrating semantic confidence, structural preservation, and contextual diversity. The compressed context is then encoded by a lightweight semantic reader for column type and column property annotation. In this way, TRACER explicitly addresses the effectiveness-efficiency trade-off by retaining label-discriminative semantic evidence for accurate annotation while reducing unnecessary computation caused by redundant table content.

The main contributions of this work are summarized as follows:

- We propose TRACER, an efficient framework for large-table ontology metadata parsing, which reduces unnecessary context encoding by constructing compact evidence-oriented contexts.

- We introduce an information-theoretic based table compression strategy that ranks table evidence according to ontology-grounded semantic uncertainty.
- We provide a conditional entropy minimization objective for large-table semantic annotation, which guides compact context construction.

2 Related Work

Feature-engineering-based Semantic Table Annotation. Feature-engineering-based methods annotate tables by designing lexical, statistical, and knowledge-graph-derived signals for table elements. Early STI systems combine string matching, entity lookup, type voting, and probabilistic inference to support cell entity annotation, column type annotation, and column property annotation (Limaye et al., 2010; Mulwad et al., 2013). Subsequent studies introduce more task-specific features, such as TF-IDF similarity, value distributions, header compatibility, entity occurrence, and row- or column-wise semantic consistency (Venetis et al., 2011; Pham et al., 2016). For numerical or mixed-type columns, prior work further models distributional properties and KG-based context to improve semantic labeling (Neumaier et al., 2016; Kacprzak et al., 2018). Recent systems such as MTab4D and AdaTyper continue this line with richer matching signals, adaptive predictors, or weak supervision to improve robustness under schema heterogeneity and distribution shifts (Nguyen et al., 2024; Hulsebos et al., 2023b). Although these methods are often efficient and interpretable, their dependence on handcrafted features and KG matching limits their ability to model complex contextual semantics in noisy or large-scale relational tables.

Deep-learning-based Semantic Table Annotation. Deep-learning-based methods learn contextualized table representations with neural encoders, reducing the reliance on manual feature design. Early neural approaches learn column-level semantic representations from cell values, table context, or candidate semantic annotations to support semantic type prediction and broader table annotation tasks (Hulsebos et al., 2019; Chen et al., 2019a,b). Table pretraining models such as TaBERT, TURL, TABBIE, TAPAS, and TAPEX encode textual and tabular inputs through structure-aware or task-oriented pretraining objectives to sup-

port downstream table understanding tasks (Yin et al., 2020; Deng et al., 2022; Iida et al., 2021; Herzig et al., 2020; Liu et al., 2022). For column annotation, SATO combines column representations, table-level topic signals, and structured dependencies, while DODUO serializes tables column-wise and jointly learns CTA and CPA with pretrained language models (Zhang et al., 2020; Suhara et al., 2022). Recent studies further enhance table representations through related-table retrieval, contrastive learning, or LLM-based annotation, including RECA, Watchog, and ArcheType (Sun et al., 2023; Miao and Wang, 2023; Feuer et al., 2024). Closely related to our work, REVEAL and REVEAL+ explicitly select target-specific column contexts to reduce noisy evidence in wide tables (Ding et al., 2025). However, existing methods mainly focus on full-table serialization, column-level context selection, or inter-table context construction, whereas TRACER targets row-level redundancy by constructing compact, label-discriminative table contexts under a fixed compression budget.

3 The TRACER Framework

3.1 Overview

Task Formulation. Given a relational table T with N rows and M columns, we denote its column set as $\mathcal{C}_T = \{c_1, c_2, \dots, c_M\}$. The goal of this work is to assign ontology-grounded semantic labels to target columns or column pairs in a large table. We consider two fundamental subtasks: Column Type Annotation (CTA) and Column Property Annotation (CPA).

For CTA, let $\mathcal{O}$ denote the predefined ontology type space. Given a target column $c_i \in \mathcal{C}_T$, the model predicts its semantic type as

$$\hat{y}_i = \arg \max_{y \in \mathcal{O}} p_\theta(y | T, c_i), \quad (1)$$

where y denotes a candidate type label in $\mathcal{O}$, and $\hat{y}_i$ denotes the predicted semantic type of column c_i .

For CPA, let $\mathcal{R}$ denote the predefined property space. Given a target column pair (c_i, c_j) with $i \neq j$, the model predicts their semantic relation as

$$\hat{r}_{ij} = \arg \max_{r \in \mathcal{R}} p_\theta(r | T, c_i, c_j), \quad (2)$$

where r denotes a candidate property label in $\mathcal{R}$, and $\hat{r}_{ij}$ denotes the predicted semantic relation between columns c_i and c_j .

Framework Overview. To mitigate the efficiency and accuracy degradation caused by redundant and noisy rows in massive tables, we propose compressing the table into a compact row-level context. As illustrated in Figure 1, TRACER consists of two main components: an **Entropy-Guided Evidence Ranker (EGER)** and a **Hybrid Evidence Compression and Annotation (HECA)** module.

From an information-theoretic perspective, the goal of **EGER** is to select rows that are maximally informative for the semantic label variables. Let $Y_g \in \mathcal{L}$ denote the semantic label variable for each target $g \in \mathcal{G}$, where $\mathcal{G}$ is the set of target columns or column pairs.

Proposition 1 (Information-theoretic compression objective). *Given a fixed row budget K , maximizing the conditional mutual information between the compressed table context and the semantic label variables,*

$$T' = \arg \max_{\substack{T^* \subseteq_{\text{row}} T \\ |T^*|=K}} \sum_{g \in \mathcal{G}} I(Y_g; T^* | g), \quad (3)$$

is equivalent to minimizing the conditional label uncertainty induced by the selected context.

Proof sketch. By the definition of conditional mutual information, we have

$$I(Y_g; T^* | g) = H(Y_g | g) - H(Y_g | T^*, g). \quad (4)$$

Since $H(Y_g | g)$ is independent of the selected context T^* , maximizing $I(Y_g; T^* | g)$ with respect to T^* is equivalent to minimizing $H(Y_g | T^*, g)$.

Guided by Proposition 1, the **HECA** module constructs the compressed table T' via a hybrid selection strategy. Subsequently, the semantic reader performs column-wise contextual encoding over T' and applies task-specific prediction heads to yield the final annotations.

3.2 Entropy-Guided Evidence Ranker

To enhance the efficiency of tabular semantic annotation, we employ a lightweight pre-trained language model (PLM) as the semantic reader. However, lightweight models typically impose strict constraints on the maximum input sequence length. Actually, an extensive relational table intrinsically contains substantial redundant information. For tabular semantic annotation, identifying the corresponding semantic types typically necessitates only a few pivotal rows, demonstrating an inherent **intra-column homogeneity** within tables.

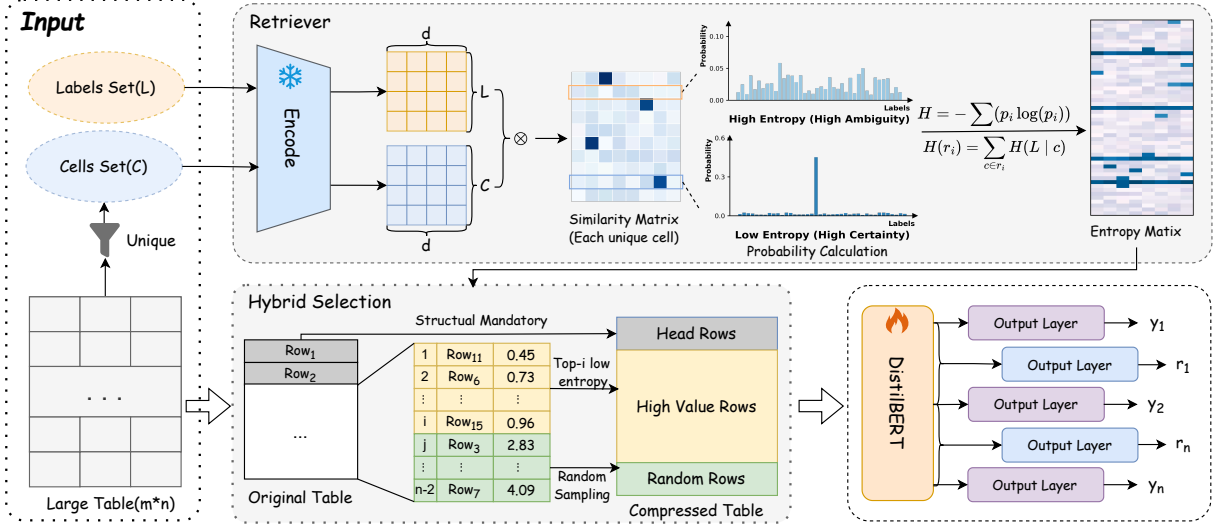


Figure 1: Overview of the TRACER framework. TRACER estimates ontology-grounded cell uncertainty, aggregates it into row-level entropy scores, and constructs a compact table context through hybrid evidence compression. The compressed table is then serialized and encoded by a lightweight semantic reader for semantic annotation.

To achieve effective compression of the large input table, we develop a training-free row ranker **EGER**. From an information-theoretic perspective, let $\mathcal{L}$ denote the ontology label space, and let $Y \in \mathcal{L}$ be the corresponding semantic label variable. For a candidate row r_i , we measure its row-level semantic informativeness by the mutual information between r_i and Y :

$$I(Y; r_i) = D_{\text{KL}}(p(Y | r_i) \| p(Y)) = \sum_{\ell \in \mathcal{L}} p(\ell | r_i) \log \frac{p(\ell | r_i)}{p(\ell)}, \quad (5)$$

where $p(Y)$ denotes the prior distribution over ontology labels, $p(Y | r_i)$ denotes the label distribution, and $I(Y; r_i)$ denotes the row-wise mutual information contribution. The KL divergence term measures the information gain provided by a row r_i relative to the prior label distribution. According to Eq. (4), the relationship between the conditional entropy $H(Y | r_i)$ and the prior entropy $H(Y)$ is rigorously formulated as:

$$H(Y | r_i) = H(Y) - \sum_{\ell \in \mathcal{L}} p(\ell | r_i) \log \frac{p(\ell | r_i)}{p(\ell)}. \quad (6)$$

Since $H(Y)$ is independent of the candidate row r_i , maximizing the mutual information is equivalent to minimizing the conditional entropy $H(Y | r_i)$. A lower conditional entropy means reduced uncertainty regarding the labels given the row, implying a more deterministic semantic indication.

However, directly computing the row-level conditional entropy is impractical, as the cells in a complete row $r_i = \{c_{i1}, c_{i2}, \dots, c_{iM}\}$ are not strictly independent. Crucially, according to the principle that conditioning reduces entropy, we have $H(Y | r_i) \leq \sum_{c \in r_i} H(Y | c)$. This motivates the aggregated cell-level conditional entropy as a tractable upper-bound surrogate for row-level uncertainty. By minimizing this surrogate, EGER favors rows whose cells provide confident and discriminative semantic evidence, without explicitly estimating the joint posterior over the entire row. The detailed implementation is described as follows.

First, we employ a lightweight encoder to project ontology labels and deduplicated table cells into a shared semantic space. Let $\mathcal{U}_T$ denote the set of unique cells in table T . For each ontology label $\ell \in \mathcal{L}$ and each cell $c \in \mathcal{U}_T$, the encoder produces embeddings $\mathbf{e}_\ell$ and $\mathbf{v}_c$, yielding the label embedding matrix $\mathbf{E}_{\text{label}} \in \mathbb{R}^{|\mathcal{L}| \times d}$ and the cell embedding matrix $\mathbf{V}_{\text{cell}} \in \mathbb{R}^{|\mathcal{U}_T| \times d}$. To compute cosine similarity through inner products, both embeddings are L_2 -normalized as $\hat{\mathbf{v}}_c = \mathbf{v}_c / \|\mathbf{v}_c\|_2$ and $\hat{\mathbf{e}}_\ell = \mathbf{e}_\ell / \|\mathbf{e}_\ell\|_2$.

Subsequently, EGER computes the similarity between each unique cell and all candidate labels in the normalized embedding space. The initial label distribution induced by cell c is obtained by applying a softmax over the cosine similarities:

$$\tilde{p}_0(\ell | c) = \frac{\exp(\hat{\mathbf{v}}_c^\top \hat{\mathbf{e}}_\ell)}{\sum_{\ell' \in \mathcal{L}} \exp(\hat{\mathbf{v}}_c^\top \hat{\mathbf{e}}_{\ell'})}. \quad (7)$$

To control the sharpness of the induced distribution, we introduce a temperature coefficient τ . The temperature-scaled label distribution is then defined as:

$$\tilde{p}_\tau(\ell | c) = \frac{\exp(\hat{\mathbf{v}}_c^\top \hat{\mathbf{e}}_\ell / \tau)}{\sum_{\ell' \in \mathcal{L}} \exp(\hat{\mathbf{v}}_c^\top \hat{\mathbf{e}}_{\ell'} / \tau)}, \quad (8)$$

where τ controls how concentrated the distribution is over ontology labels. A smaller τ makes the distribution sharper and emphasizes the most similar labels, while a larger τ produces a smoother distribution.

Based on the temperature-scaled distribution, we estimate the cell-level label uncertainty using Shannon entropy:

$$\hat{H}(Y | c) = - \sum_{\ell \in \mathcal{L}} \tilde{p}_\tau(\ell | c) \log \tilde{p}_\tau(\ell | c). \quad (9)$$

This entropy serves as a proxy for the conditional uncertainty of the semantic label given the cell, rather than an unbiased estimate of the true posterior entropy. A lower value indicates that the cell induces a more confident and ontology-discriminative label distribution.

Finally, EGER aggregates cell-level uncertainties into a row-level uncertainty score, defined as $U(r_i) = \sum_{c \in r_i} \hat{H}(Y | c)$, where $U(r_i)$ measures the estimated semantic uncertainty of row r_i . Rows are ranked in ascending order of $U(r_i)$, so lower-scoring rows are selected as high-value evidence rows for constructing the compressed table.

3.3 Hybrid Evidence Compression and Annotation

Hybrid Evidence Compression. Given the row-level uncertainty score $U(r_i)$ produced by EGER, HECA constructs the compressed table by combining structural preservation, entropy-guided evidence selection, and diversity-aware sampling. Let $\mathcal{I} = \{1, \dots, N\}$ denote the row index set of the original table, and let $K = K_{\text{head}} + K_{\text{ent}} + K_{\text{rand}}$ be the compression budget.

Structural Preservation. We first retain the first K_{head} rows as structural anchors $I_{\text{head}} = \{i \in \mathcal{I} | i \leq K_{\text{head}}\}$. Although these rows are not necessarily the lowest-entropy rows, they often provide stable contextual patterns and preserve the original input structure expected by the downstream reader.

Entropy-guided Evidence Selection. After removing the structural anchors, HECA selects rows with the most confident ontology-related evidence.

Specifically, the entropy-guided subset is obtained by minimizing the aggregated row uncertainty:

$$I_{\text{ent}} = \arg \min_{\substack{A \subseteq \mathcal{I} \setminus I_{\text{head}} \\ |A| = K_{\text{ent}}}} \sum_{i \in A} U(r_i), \quad (10)$$

which is equivalent to selecting the K_{ent} rows with the lowest uncertainty scores. These rows are expected to contain more confident and ontology-discriminative semantic evidence.

Diversity-aware Sampling. Relying only on low-entropy rows may bias the compressed context toward highly deterministic patterns. To retain broader distributional coverage, HECA further samples K_{rand} rows uniformly without replacement from the residual pool $\mathcal{I} \setminus (I_{\text{head}} \cup I_{\text{ent}})$. We denote the sampled index set as I_{rand} . This random component complements the deterministic entropy-guided selection and helps preserve non-dominant but potentially useful table evidence. The final row index set is then defined as $I_{\text{final}} = I_{\text{head}} \cup I_{\text{ent}} \cup I_{\text{rand}}$. The compressed table T' is obtained by retaining the rows indexed by I_{final} .

Subsequently, T' is fed into a lightweight semantic reader designed to capture both intra-column and inter-column semantic information. Unlike conventional unstructured text in natural language processing, tables inherently possess a two-dimensional structure. To comply with the input format of the reader model while preserving the semantic coherence of individual columns, we adopt a column-wise serialization strategy. Specifically, for each column c_j in the compressed table T' , we concatenate the textual content of all constituent cells, prepending and appending special tokens [CLS] and [SEP] to each sequence, respectively. The resulting input sequence S is formulated as follows:

$$S = \bigoplus_{j=1}^{|\mathcal{G}|} ([\text{CLS}] \oplus c_j \oplus [\text{SEP}]), \quad (11)$$

where the symbol $\oplus$ denotes the sequence concatenation operator. The resulting sequence S is processed by a DistilBERT encoder, which leverages multi-layer self-attention mechanisms to facilitate global interactions and capture the contextualized representations for each column. To address the Column Type Annotation (CTA) and Column Property Annotation (CPA) tasks, we append task-specific classification heads to the derived column embeddings to yield the final semantic predictions.

3.4 Training Strategy

Given the semantic interdependence among columns within the same table, we adopt an end-to-end joint training strategy that performs multi-column context modeling and semantic classification in a unified stage. Specifically, the serialized packed sequence S in Eq. (11) is fed into a DistilBERT encoder, whose self-attention layers capture global tabular context and inter-column dependencies across column boundaries.

The model is optimized by minimizing a joint cross-entropy loss over all $|\mathcal{G}|$ targets within a single forward pass:

$$Loss = -\frac{1}{|\mathcal{G}|} \sum_{i=1}^{|\mathcal{G}|} \log P(y_i | \mathbf{x}_i), \quad (12)$$

where $|\mathcal{G}|$ denotes the number of target columns, y_i is the ground-truth semantic label for column c_i , and $\mathbf{x}_i \in \mathbb{R}^d$ represents its corresponding contextualized hidden representation derived from the encoder.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate TRACER on two representative benchmarks covering the two core tasks of semantic table annotation. GitTable is used for column type annotation (CTA), where the goal is to assign ontology-grounded semantic types to target columns in large relational tables. SemTab-CPA is used for column property annotation (CPA), where the goal is to predict semantic relations between subject-object column pairs. These two benchmarks differ in table scale, label space, and annotation granularity, enabling us to evaluate both the effectiveness and efficiency of TRACER under diverse semantic annotation settings.

Metrics. Following prior studies, we report Accuracy (Acc), Macro-F1 (MaF1), Precision (Pre), and Recall (Rec) to evaluate annotation effectiveness. For efficiency, we report online end-to-end inference time and throughput, which measure test-time latency and large-scale annotation capacity, respectively.

Baselines. We compare TRACER with representative task-specific semantic annotation methods, including SATO, DODUO, RECA, and REVEAL, as well as two instruction-tuned LLM baselines, Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct. The task-specific baselines cover feature-

enhanced, PLM-based, retrieval-augmented, and context-selection-based annotation paradigms. For LLM baselines, we adopt a closed-set prompt-based setting, where each model is provided with raw table evidence and a prototype-retrieved candidate label set, and is required to output a candidate label ID without task-specific fine-tuning.

4.2 Main Results

Table 1 reports the main results on GitTable and SemTab. Existing task-specific baselines achieve competitive accuracy, but their class-balanced metrics remain limited, especially on GitTable, indicating that they are less effective in handling long-tail ontology labels and fine-grained semantic distinctions. On SemTab, REVEAL benefits from column-level context filtering and obtains strong results, but existing methods still mainly rely on full-table serialization, column-level selection, or external related-table evidence, leaving row-level redundancy insufficiently addressed. In contrast, the prompt-only LLM baselines perform poorly on both datasets, suggesting that general instruction-following ability alone is insufficient for closed-set ontology-grounded annotation over large and fine-grained label spaces.

TRACER achieves the strongest overall performance, particularly on class-balanced metrics, demonstrating the effectiveness of entropy-guided evidence compression. Across both CTA and CPA, it improves all metrics while maintaining competitive accuracy, indicating that compact evidence-oriented contexts help the reader focus on ontology-discriminative rows and reduce interference from redundant content. Results with different encoder backbones further show that TRACER is not tied to a specific PLM architecture; its main advantage comes from informative context construction before reader-based prediction.

4.3 Efficiency Analysis

To further examine the practical efficiency of TRACER, we conduct an efficiency comparison on GitTable using two complementary metrics: online end-to-end inference time and throughput. The former measures the total wall-clock time of the complete test-time annotation pipeline, including context construction and model inference, while the latter reflects the number of target columns or column pairs processed per second.

As shown in Figure 2, TRACER achieves the best efficiency among all task-specific baselines. It

Methods	GitTable				SemTab			
	Acc	MaF1	Pre	Rec	Acc	MaF1	Pre	Rec
<i>Existing Baselines</i>								
SATO	95.24	42.49	52.73	39.28	81.49	43.17	46.72	43.50
DODUO	94.65	45.22	52.81	42.97	91.63	58.55	59.13	60.87
RECA	93.90	39.38	45.11	38.69	89.72	48.12	50.43	49.51
REVEAL	91.36	44.61	54.46	41.99	<u>93.03</u>	65.17	65.90	66.51
<i>Large Language Models</i>								
Qwen2.5-7B	24.02	7.87	9.17	11.23	26.70	15.73	14.70	24.96
Llama-3.1-8B	11.56	2.63	3.99	3.42	23.03	11.13	12.10	16.04
<i>Our Method with Different Backbones</i>								
Ours (DistilBERT)	96.54	71.12	74.32	70.73	92.55	73.28	74.58	74.05
Ours (DeBERTa)	<u>96.81</u>	<u>61.84</u>	<u>69.77</u>	<u>59.10</u>	92.50	69.31	70.38	70.69
Ours (RoBERTa)	97.02	60.69	67.83	58.39	93.15	<u>72.47</u>	<u>73.67</u>	<u>73.46</u>

Table 1: We compare existing baselines, large language model baselines, and our method with different encoder backbones. Bold and underlined values denote the best and second-best results for each metric, respectively.

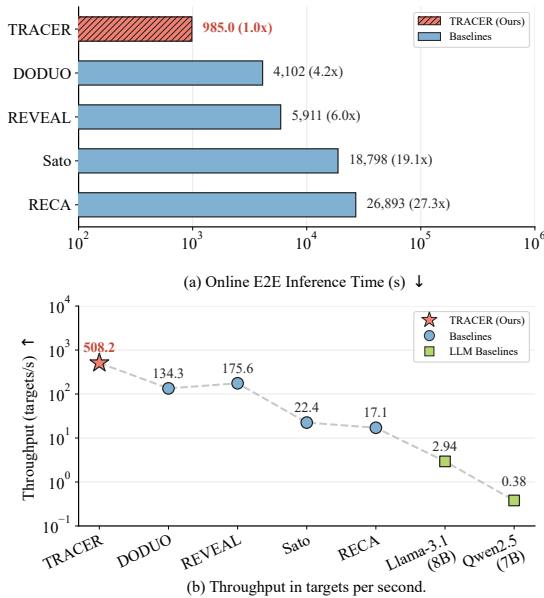


Figure 2: Efficiency comparison on GitTable.

substantially reduces online inference time compared with DODUO, REVEAL, Sato, and RECA, while also achieving the highest throughput. This advantage mainly comes from entropy-guided row compression, which restricts expensive reader encoding to compact evidence-oriented contexts instead of processing redundant full-table content or relying on additional retrieval and feature construction. The gap is even more pronounced when compared with prompt-based LLM baselines, whose autoregressive generation and large ontology search space make them inefficient for large-scale table annotation. Overall, these results show that TRACER improves deployment efficiency without sacrificing

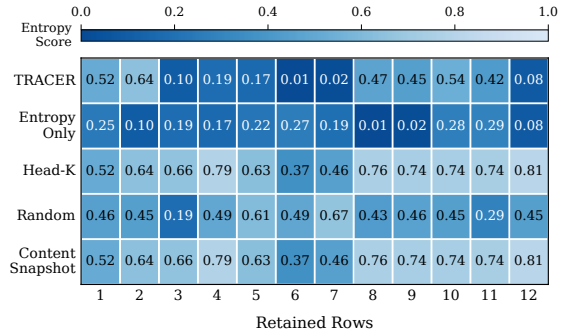


Figure 3: Comparison of row-level entropy scores under different compression strategies. Darker cells indicate lower entropy and stronger semantic evidence.

annotation effectiveness, providing a more favorable effectiveness–efficiency trade-off for scalable ontology-grounded metadata parsing.

4.4 Ablation Studies

To evaluate the contribution of different components in TRACER, we conduct ablation studies on GitTable by removing the entropy-guided retriever and the hybrid compression strategy.

w/o EGER. This variant removes the entropy-guided evidence selection in HECA. It preserves the head rows as structural anchors, while the remaining rows are randomly sampled from the original table.

w/o Hybrid Selection Strategy (HSS). This variant removes the hybrid compression strategy. It directly selects rows according to the entropy ranking produced by EGER.

w/o Both. This variant removes both the entropy-guided retriever and the compression strategy. It

Method	Macro-F1	Pre.	Rec.	Acc.	Time
TRACER	71.12	74.32	70.73	96.54	985.0
w/o EGER	30.92 (↓56.53%)	35.05 (↓52.84%)	29.66 (↓58.07%)	65.08 (↓32.59%)	211.29 (↓78.55%)
w/o HSS	64.48 (↓9.34%)	68.24 (↓8.18%)	64.26 (↓9.15%)	95.09 (↓1.50%)	933.65 (↓5.21%)
w/o BOTH	41.38 (↓41.82%)	46.07 (↓38.01%)	43.49 (↓38.51%)	75.40 (↓21.90%)	1458.46 (↑48.07%)

Table 2: Ablation results on the GitTable dataset. Values in parentheses denote the changes compared with TRACER.

feeds the original uncompressed table into the semantic reader.

As shown in Table 2, removing EGER causes the largest degradation, indicating that random compression cannot reliably retain ontology-discriminative evidence. Although w/o HSS remains competitive, its gap to TRACER shows the benefit of combining entropy-guided ranking with structural preservation and diversity-aware sampling. The weaker and slower w/o BOTH variant further confirms that full-table encoding may introduce redundant or noisy content.

Figure 3 shows that TRACER retains both low-entropy evidence and structural context, unlike heuristic strategies with less stable evidence. This confirms that EGER and HSS are complementary for compact and informative context construction.

We further analyze the effect of the compression budget K on GitTable. As shown in Figure 4, increasing K initially improves performance, indicating that an overly small budget cannot retain sufficient ontology-discriminative evidence. The best overall results are achieved with a moderate budget, while further increasing K leads to slight degradation, suggesting that additional rows may introduce redundant or noisy context rather than useful semantic signals. These results show that TRACER benefits from balancing evidence preservation and redundancy filtering, with $K = 15$ used as the default setting in our experiments.

4.5 Hyperparameter Analysis

τ	Accuracy (%)	Δ Acc. (pp)
0.03	96.52	-0.02
0.05	96.54	0.00
0.10	96.49	-0.05

Table 3: Sensitivity analysis of the temperature parameter τ on GitTable.

We further analyze the sensitivity of the temperature parameter τ used in the entropy estimation. As shown in Table 3, TRACER achieves stable per-

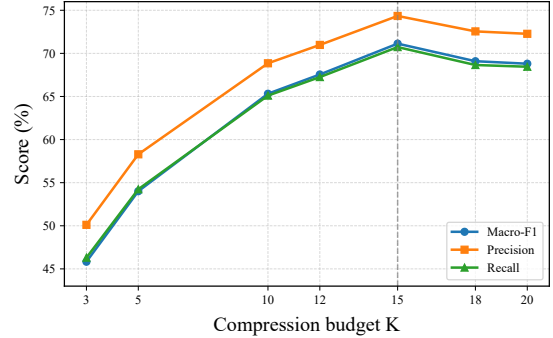


Figure 4: Sensitivity analysis of the compression budget K on GitTable.

formance across different values of τ , with only minor variations in accuracy. The maximum difference is only 0.05 percentage points, suggesting that EGER does not require careful tuning of τ and remains stable under moderate changes in the temperature setting. Accordingly, We use $\tau = 0.05$ as the default value in our experiments.

5 Conclusion

In this paper, we proposed TRACER, an information entropy-grounded framework for efficient large-table semantic annotation. TRACER constructs compact contexts through entropy-guided row ranking and hybrid compression, enabling the semantic reader to preserve ontology-discriminative evidence while reducing redundant table content. Experiments on CTA and CPA benchmarks demonstrate that TRACER achieves a strong effectiveness–efficiency trade-off, highlighting the value of principled row-level evidence compression for scalable ontology-grounded table annotation. These findings suggest that large table annotation should prioritize evidence quality over context size, with entropy-guided compression balancing semantic reliability and computational efficiency.

Limitations

TRACER still has several limitations. First, it is evaluated under a closed-set ontology setting, and

extending it to open-world or dynamically evolving label spaces remains an important direction. Second, the current framework uses a fixed compression budget for different tables, which may not fully adapt to variations in table size and evidence density. Future work can explore adaptive budget allocation and more robust uncertainty estimation to further improve large-table semantic annotation.

Ethics Statement

TRACER aims to improve the efficiency and effectiveness of large-table semantic annotation under a closed-set ontology setting. In this work, all datasets and baseline models are publicly available, and we do not collect new user data or introduce additional personally identifiable information. The dataset construction and label filtering procedures are used only for building reliable evaluation settings and are applied uniformly to all compared methods. Since TRACER predicts ontology-grounded labels rather than generating open-ended text, the risk of producing harmful free-form content is limited. Nevertheless, incorrect semantic annotations may affect downstream applications such as schema matching, data integration, or knowledge graph construction, so the predictions should be used with appropriate validation in sensitive scenarios. There are no conflicts of interest among the authors.

References

- Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. [Webtables: exploring the power of tables on the web](#). *Proc. VLDB Endow.*, 1(1):538–549.
- Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. 2019a. [Colnet: Embedding the semantics of web tables for column type prediction](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 29–36. AAAI Press.
- Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. 2019b. [Learning semantic annotations for tabular data](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2088–2094. ijcai.org.
- Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. [TURL: table understanding through representation learning](#). *SIGMOD Rec.*, 51(1):33–40.
- Zhihao Ding, Yongkang Sun, and Jieming Shi. 2025. [Retrieve-and-verify: A table context selection framework for accurate column annotations](#). *Proc. ACM Manag. Data*, 3(6).
- 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):2279–2292.
- Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisen-schlos. 2020. [Tapas: Weakly supervised table parsing via pre-training](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4320–4333. Association for Computational Linguistics.
- Madelon Hulsebos, Çağatay Demiralp, and Paul Groth. 2023a. [Gittables: A large-scale corpus of relational tables](#). *Proc. ACM Manag. Data*, 1(1):30:1–30:17.
- Madelon Hulsebos, Paul Groth, and Çağatay Demiralp. 2023b. [Adatyper: Adaptive semantic column type detection](#). *CoRR*, abs/2311.13806.
- Madelon Hulsebos, Kevin Zeng Hu, Michiel A. Bakker, Emanuel Zraggen, Arvind Satyanarayan, Tim Kraska, Çağatay Demiralp, and César A. Hidalgo. 2019. [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 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1500–1508. ACM.
- Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. [TABBIE: Pretrained representations of tabular data](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.
- Emilia Kacprzak, José M. Giménez-García, Alessandro Piscopo, Laura Koesten, Luis-Daniel Ibáñez, Jeni Tennison, and Elena Simperl. 2018. [Making sense of numerical data - semantic labelling of web tables](#). In *Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings, Lecture Notes in Computer Science*, pages 163–178. Springer.
- Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. [A large public corpus of web tables containing time and context metadata](#). In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 75–76. ACM.

- Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. [Annotating and searching web tables using entities, types and relationships](#). *Proc. VLDB Endow.*, 3(1):1338–1347.
- Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. [From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods](#). *J. Web Semant.*, 76:100761.
- Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. [TAPEX: table pre-training via learning a neural SQL executor](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Zhengjie Miao and Jin Wang. 2023. [Watchog: A light-weight contrastive learning based framework for column annotation](#). *Proc. ACM Manag. Data*, 1(4):272:1–272:24.
- Varish Mulwad, Tim Finin, and Anupam Joshi. 2013. [Semantic message passing for generating linked data from tables](#). In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, Lecture Notes in Computer Science, pages 363–378. Springer.
- Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. 2016. [Multi-level semantic labelling of numerical values](#). In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, Lecture Notes in Computer Science, pages 428–445.
- Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. 2024. [Mtab4d: Semantic annotation of tabular data with dbpedia](#). *Semantic Web*, 15(6):2613–2637.
- Minh Pham, Suresh Alse, Craig A. Knoblock, and Pedro A. Szekely. 2016. [Semantic labeling: A domain-independent approach](#). In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, Lecture Notes in Computer Science, pages 446–462.
- Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çagatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. [Annotating columns with pre-trained language models](#). In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 1493–1503. ACM.
- Yushi Sun, Hao Xin, and Lei Chen. 2023. [RECA: related tables enhanced column semantic type annotation framework](#). *Proc. VLDB Endow.*, 16(6):1319–1331.
- Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. 2011. [Recovering semantics of tables on the web](#). *Proc. VLDB Endow.*, 4(9):528–538.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. [TaBERT: Pretraining for joint understanding of textual and tabular data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.
- Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çagatay Demiralp, and Wang-Chiew Tan. 2020. [Sato: Contextual semantic type detection in tables](#). *Proc. VLDB Endow.*, 13(11):1835–1848.