RoCEL: Advancing Table Entity Linking through Distinctive Row and Column Contexts

Yuanzheng Wang, Yixing Fan*, **Jiafeng Guo, Ruqing Zhang, Xueqi Cheng**
CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences;
School of Computer Science and Technology, University of Chinese Academy of Sciences;
Beijing, China
{wangyuanzheng19z, fanyixing, guojiafeng, zhangruqing, cxq}@ict.ac.cn

Abstract

Table entity linking (TEL) aims to map entity mentions in the table to their corresponding entities in a knowledge base (KB). The core of this task is to leverage structured contexts, specifically row and column contexts, to enhance the semantics of mentions in entity disambiguation. Most entity linking (EL) methods primarily focus on understanding sequential text contexts, making it difficult to adapt to the row and column structure of tables. Additionally, existing methods for TEL indiscriminately mix row and column contexts together, overlooking their semantic differences. In this paper, we explicitly distinguish the modeling of row and column contexts, and propose a method called *RoCEL* to capture their distinct semantics. Specifically, for row contexts in tables, we take the attention mechanism to learn the implicit relational dependencies between each cell and the mention. For column contexts in tables, we employ a set-wise encoder to learn the categorical information about the group of mentions. At last, we merge both contexts to obtain the final mention embedding for link prediction. Experiments on four benchmarks show that our approach outperforms the state-of-the-art (SOTA) baseline by about 1.5% on the in-domain dataset, and by 3.7% on average across three out-of-domain datasets.

1 Introduction

TEL is an important task in natural language processing (NLP), which serves as a building block of table understanding. It benefits many downstream table-oriented tasks. For instance, in table question answering (Chemmengath et al., 2021), it helps locate the answer for entity-centric questions. In knowledge base population (Zhang et al., 2020a), it aids in mining relationships between entities from the tables. The task of TEL is to map the mentions in a table to their referent entities in a given KB.

*Corresponding author

Best-selling albums of the 21st century, 13–19 million copies:

Release Year	Album	Artist/s	Nationality	Worldwide sales (in millions)
2008	The Fame	Lady Gaga	United States	18.0
2003	Fallen	Evanescence	United States	17.0
2002	Let Go	Avril Lavigne	Canada	16.0

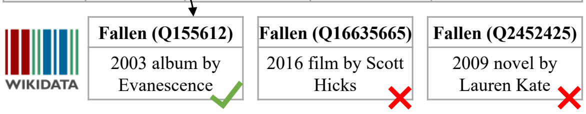

Figure 1: An example of table entity linking. The blue texts represent entity mentions.

Figure 1 shows an example, where the mention "Fallen" is linked to the entity "Fallen (Q155612)" in WikiData.

The major challenge of TEL is how to make use of the structured contexts in tables to help link mentions to their corresponding entities. Most of the existing works have focused on text EL, which aims to model the sequential context of the mention (Ledell Wu, 2020; De Cao et al., 2021; Tom Ayoola, 2022). However, tables often exhibit different structures and relationships between mentions and contexts compared to text. Firstly, the context in tables is presented in a two-dimensional structure, i.e., row and column, whereas in text, it is displayed in a one-dimensional sequential form. Secondly, the row context typically includes descriptional information about the mention, and the column context contains similar entities to the mention. However, the sequential context in texts is often natural language sentences with syntactic and grammatical constraints. In this way, the TEL entails a more complex structured context, rendering it more challenging than the text.

Recently, some works have tried to model the structured context in tables. For example, Herzig et al. (2020) proposes a pretrained language model TAPAS for table question answering, and Sui et al. (2024) adopts markup languages to leverage large language model (LLM) for table-to-text generation.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15284–15298
November 12-16, 2024

The most relevant works to ours are TURL (Deng et al., 2020) and NPEL (Wu et al., 2024), where TURL employs a Transformer with a masked self-attention mechanism, and NPEL uses GCN to encode row and column contexts. However, neither method can well distinguish the semantic differences between row and column contexts. This results in a coarse representation of table information, and limits the accuracy of TEL.

In fact, the row and column in the table usually provide contexts of different aspects about the mention (Liu et al., 2023), which calls for distinct modeling strategies. In the row context, each cell is either the properties or related objects to the mention. For example in Figure 1, the row context of the mention "Fallen" contains the release year (i.e., "2003"), the performer (i.e., "Evanescence"), the nationality (i.e., "United States") and the sales (i.e., "17.0"). In the column context, each cell typically represents an entity similar to the mention, all belonging to the same category specified by the column header. As is shown in Figure 1, the column context of the mention "Fallen" includes other albums such as "The Fame" and "Let Go".

In this study, we propose **Ro**w-**C**olumn differentiated table **E**ntity **L**inking (RoCEL) to learn the distinct context in the row and column of the mention. Specifically, (1) to learn the descriptional information of mentions, we serialize row contexts with implicit dependencies, and encode them with the self-attention mechanism. (2) To capture the categorical information of mentions, we treat a column of independent mentions as an unordered set, and learn the category through a set-wise encoder. (3) Finally, we fuse the representation of both contexts to obtain the mention embedding, and link it to the target entity in the KB.

To evaluate the effectiveness of our method, we conduct experiments on four benchmark datasets. The results demonstrate that our method outperforms all SOTA baselines, including cell-based, text-based, and table-based methods. Further experiments show that our method can effectively utilize row and column contexts on both in-domain and out-of-domain data. Additionally, we evaluate the open-sourced LLM Llama-3 (AI@Meta, 2024b) on this task, and find it challenging for the model to leverage the structured contexts for TEL.

The contribution of this work is as follows:

• We propose a novel table entity linking method RoCEL, which highlights the semantic difference between row and column contexts.

• We conduct systematic experiments on four benchmarks, showing that RoCEL outperforms all SOTA baselines on both in-domain and out-of-domain datasets.

2 Preliminary

Table Contexts. In this paper, the term "table" refers to "relational table" (Liu et al., 2023), which is similar to a table in a database. It contains multiple rows and columns, where each row represents an individual record and each column represents a field of the records. Given a table T with I rows (excluding headers) and J columns, as illustrated in Figure 2, we denote: (1) T_Θ as the metadata of T, such as its title, caption, or surrounding text; (2) T_{ij} as the cell at the i-th row and j-th column, $1 \leq i \leq I$, $1 \leq j \leq J$; (3) T_{i*} as the i-th row (excluding headers); (4) T_{*j} as the j-th column; (5) T_{Hj} as the header of the j-th column; (6) T_{H*} as the set of all headers.

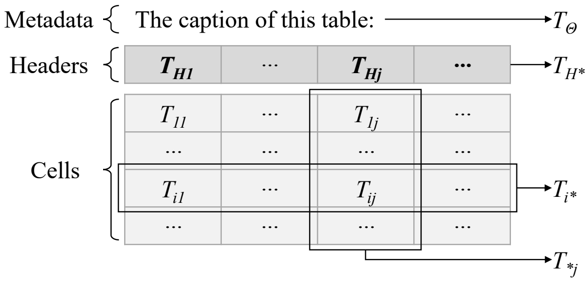

Figure 2: An illustration of table contexts.

Table Entity Linking. In this task, we only link cells that are entity mentions. For example, in Figure 1, the cell "Fallen" is a mention, while the cell "17.0" is not. Given a table T, a mention cell T_{ij}, and an entity base $\mathcal{E}$, the task is to find the corresponding entity $e_{ij} \in \mathcal{E}$ for T_{ij}.

3 Method

In this section, we will give a detailed introduction to our model, namely RoCEL. Figure 3 shows an overview of the RoCEL, which comprises of three major components: (1) *Row Context Encoding*, aiming to represent the descriptional information from row contexts; (2) *Column Context Encoding*, aiming to represent the categorical information from column contexts; and (3) *Semantics Fusion & Entity Linking*, which fuses the embeddings of row and column contexts into a final mention embedding, and links it to the target entity.

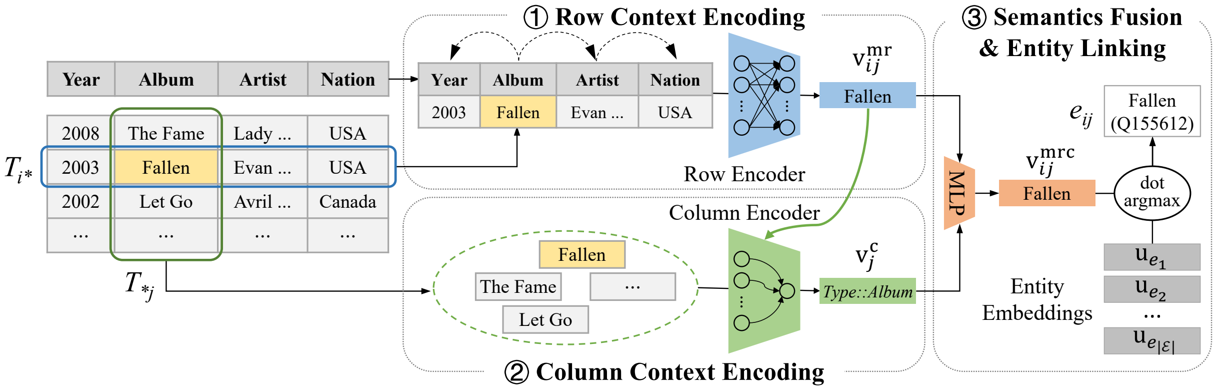

Figure 3: An overall architecture of the proposed RoCEL. "Fallen" is the mention cell T_{ij} to be linked.

3.1 Row Context Encoding

We first encode the row context of a mention to represent its descriptional information, including its properties and related objects. This information is provided by the implicit and relational dependencies between the mention and other cells. Figure 4 shows an example of the dependencies, where "2008" is the release year of "The Fame", not of "Lady Gaga". To capture the implicit dependencies, we serialize the row context into a sequential text, and encode it using BERT (Devlin et al., 2019). This approach aims to (1) capture dependencies between tokens through the self-attention mechanism, thereby approximating the dependencies between cells, and (2) understand the texts of headers and cells with the linguistic knowledge of BERT.

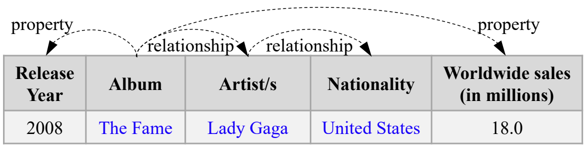

Figure 4: An example of the implicit dependencies within the row context.

Following Deng et al. (2020), the metadata is also provided for better performance. Formally, given a mention cell T_{ij}, a row of cells $T_{i1}, \dots, T_{iJ} \in T_{i*}$, headers $T_{H1}, \dots, T_{HJ} \in T_{H*}$, and metadata T_Θ, we first convert each pair of $\langle T_{Hk}, T_{ik} \rangle$ into a text piece p_{ik}, then concatenate them with T_Θ to obtain the sequential text txt_{ij}:

$$\mathrm{p}_{ik} = \begin{cases} T_{Hk} : [\mathrm{START}]\ T_{ik}\ [\mathrm{END}], & k = j, \\ T_{Hk} : T_{ik}, & k \neq j, \end{cases}$$
$$\mathrm{txt}_{ij} = [\mathrm{CLS}]\ T_\Theta \mid \mathrm{p}_{i1}\ ; \cdots ; \mathrm{p}_{iJ}\ [\mathrm{END}], \tag{1}$$

where $1 \leq k \leq J$, characters "|", ":", and ";" are delimiters. [START] and [END] mark the mention T_{ij}. Finally, we encode txt_{ij} using BERT, and take [CLS] vector in the last layer as the row-contextualized mention embedding $\mathbf{v}_{ij}^{\mathrm{mr}} \in \mathbb{R}^n$:

$$\mathbf{v}_{ij}^{\mathrm{mr}} = \mathrm{BERT}_{[\mathrm{CLS}]}\left(\mathrm{txt}_{ij}\right),$$

where n is the hidden size of the embedding. $\mathbf{v}_{ij}^{\mathrm{mr}}$ integrates the descriptional information of row T_{i*} into the embedding of mention T_{ij} .

3.2 Column Context Encoding

After encoding the row context, we encode the column context for categorical information. Column contexts are looser than row contexts. Specifically, column cells are dependence-free and equal, with the only commonality sharing the same type (which is also called the column type[1]). As a result, we treat a column of cells as an unordered set, and represent it through a set-wise encoder. We choose FSPool (Zhang et al., 2020b) as the encoder, and the reason will be discussed in Section 3.4.

Specifically, given a column of mentions $T_{1j}, \dots, T_{Ij} \in T_{*j}$ and the set of their row-contextualized embeddings[2] $\left\{\mathbf{v}_{1j}^{\mathrm{mr}}, \dots, \mathbf{v}_{Ij}^{\mathrm{mr}}\right\}$, we use FSPool to aggregate them into a column embedding $\mathbf{v}_j^{\mathrm{c}} \in \mathbb{R}^n$:

$$\mathbf{v}_j^{\mathrm{c}} = \mathrm{FSPool}\left(\left\{\mathbf{v}_{1j}^{\mathrm{mr}}, \dots, \mathbf{v}_{Ij}^{\mathrm{mr}}\right\}\right).$$

$\mathbf{v}_j^{\mathrm{c}}$ is expected to capture the type of column T_{*j}, However, in the early stages of training, the randomly initialized encoder struggles to extract meaningful information and may introduce noise. To address this issue, we introduce auxiliary tasks to warm up the column encoder at the beginning, which will be introduced in Section 3.4.

[1]The name of the column type often differs from the column header. For further discussion, see Appendix B.

[2]We use row-contextualized mention embeddings rather than the embeddings of each mention cell, because the latter contains too little information.

3.3 Semantics Fusion & Entity Linking

After obtaining the embeddings of row and column contexts, we fuse them together. Specifically, for a mention T_{ij}, we concatenate the row-contextualized embedding $\mathbf{v}_{ij}^{\mathrm{mr}}$ with the column embedding $\mathbf{v}_j^{\mathrm{c}}$ and mix them through a multi-layer perceptron (MLP):

$$\mathbf{v}_{ij}^{\mathrm{mrc}} = \mathrm{MLP}\left(\mathbf{v}_{ij}^{\mathrm{mr}}, \mathbf{v}_j^{\mathrm{c}}\right),$$

where $\mathbf{v}_{ij}^{\mathrm{mrc}} \in \mathbb{R}^n$ is the final mention embedding of T_{ij}, and it incorporates the descriptional information from the row context and categorical information from the column context.

Finally, we link T_{ij} to the most similar entity through dense retrieval. Specifically, given a knowledge base $\mathcal{E}$, the number of entities $|\mathcal{E}|$, we also obtain an embedding[3] $\mathbf{u}_e$ for each entity $e \in \mathcal{E}$. Then we compute the similarity score $s_{ij}(e) \in \mathbb{R}$ using dot product:

$$s_{ij}(e) = \mathbf{v}_{ij}^{\mathrm{mrc}} \cdot \mathbf{u}_e,$$
$$e_{ij} = \operatorname*{argmax}_{e \in \mathcal{E}} s_{ij}(e),$$

where e_{ij} is the predicted entity for T_{ij}. The loss $\mathcal{L}^{\mathrm{EL}}$ for entity linking is ranking cross-entropy:

$$\mathbf{s}_{ij}^{\mathrm{EL}} = \mathrm{Softmax}\left(s_{ij}(e_1), \ldots, s_{ij}(e_{|\mathcal{E}|})\right),$$
$$\mathcal{L}^{\mathrm{EL}} = \mathrm{CrossEntropy}\left(\mathbf{s}_{ij}^{\mathrm{EL}}, \mathbf{y}_{ij}^{\mathrm{EL}}\right),$$

where $\mathbf{s}_{ij}^{\mathrm{EL}} \in \mathbb{R}^{|\mathcal{E}|}$ is the vector of softmax-normalized scores, and $\mathbf{y}_{ij}^{\mathrm{EL}} \in \{0,1\}^{|\mathcal{E}|}$ is the vector of one-hot label.

3.4 Warm-up of the Column Encoder

To better represent categorical information, we take two auxiliary tasks to warm up the column encoder. The first and most straightforward task is to train column typing in a supervised manner, where column types can be distant-labeled using the shared types of target entities (Deng et al., 2020). Specifically, given the set of all types Ω, the number of types $|\Omega|$, the column T_{*j}, the column embedding $\mathbf{v}_j^{\mathrm{c}}$, and the vector of multi-hot label $\mathbf{y}_j^{\mathrm{CT}} \in \mathbb{R}^{|\Omega|}$ for column typing, we use Linear and Sigmoid to obtain the scores $\mathbf{s}_j^{\mathrm{CT}} \in \mathbb{R}^{|\Omega|}$ for each type, and the loss $\mathcal{L}^{\mathrm{CT}}$ for column typing is binary cross-entropy:

$$\mathbf{s}_j^{\mathrm{CT}} = \mathrm{Sigmoid}\left(\mathrm{Linear}\left(\mathbf{v}_j^{\mathrm{c}}\right)\right),$$
$$\mathcal{L}^{\mathrm{CT}} = \mathrm{BinaryCrossEntropy}\left(\mathbf{s}_j^{\mathrm{CT}}, \mathbf{y}_j^{\mathrm{CT}}\right).$$

[3]We utilize the entity embeddings generated by Ledell Wu (2020), which are obtained by encoding the Wikipedia descriptions of entities using BERT. These embeddings are kept frozen during training.

However, sometimes even distant supervision is still challenging, for example, if types of target entities are not provided. In such cases, we can employ an unsupervised warm-up task instead. We choose a set reconstruction task based on autoencoder, as extracting shared types from mentions is a form of information compression. The autoencoder should satisfy: (1) Compress-restore: The encoder aggregates I elements into an intermediate vector, and the decoder reconstructs it back into I elements; (2) Order-invariance: The intermediate vector does not change with the order of input elements; (3) Order-preservation: The input elements and the reconstructed elements maintain the same order. To meet the above conditions, we choose FSPool and its inversion FSUPool as the set-wise encoder and decoder, respectively (Zhang et al., 2020b). Formally, given a column embedding $\mathbf{v}_j^{\mathrm{c}}$ encoded from FSPool, we decode it to a set of I vectors $\left\{\hat{\mathbf{v}}_{1j}^{\mathrm{mr}}, \ldots, \hat{\mathbf{v}}_{Ij}^{\mathrm{mr}}\right\}$ through FSUPool , where $\hat{\mathbf{v}}_{ij}^{\mathrm{mr}} \in \mathbb{R}^n$ is the reconstructed embedding of $\mathbf{v}_{ij}^{\mathrm{mr}}$, then we use mean-square loss as the loss $\mathcal{L}^{\mathrm{SR}}$ for set reconstruction:

$$\left\{\hat{\mathbf{v}}_{1j}^{\mathrm{mr}}, \ldots, \hat{\mathbf{v}}_{Ij}^{\mathrm{mr}}\right\} = \mathrm{FSUPool}\left(\mathbf{v}_j^{\mathrm{c}}\right),$$
$$\mathcal{L}^{\mathrm{SR}} = \sum_{i=1}^{I}\left(\hat{\mathbf{v}}_{ij}^{\mathrm{mr}} - \mathbf{v}_{ij}^{\mathrm{mr}}\right)^2.$$

The complete training process is as follows: First, we warm up the column encoder using either the column typing or set reconstruction task. During this stage, BERT is frozen, and only the column encoder (FSPool) and auxiliary modules (i.e., Linear for column typing, and FSUPool for set reconstruction) are trained until convergence. The loss function is either $\mathcal{L}^{\mathrm{CT}}$ or $\mathcal{L}^{\mathrm{SR}}$. Next is the training of entity linking, where auxiliary modules are removed, and all remaining parameters are finetuned. The loss at this stage is only $\mathcal{L}^{\mathrm{EL}}$.

4 Experimental Settings

In this section, we will introduce our experimental datasets, baselines, and implementation details.

4.1 Datasets

We conduct experiments on four benchmarks of table entity linking: TURL-Data (Deng et al., 2020), T2D (Lehmberg et al., 2016), Wikilinks-LARGE (**Wikilinks-L**) (Wu et al., 2024), and Wikilinks-RANDOM-2020 (**Wikilinks-R**) (Wu et al., 2024). Note that tables in T2D come from various websites, while tables in other datasets are sourced from Wikipedia. We use the largest dataset TURL-Data for training, validation, and testing. The other three datasets are small and only for out-of-domain testing. Unless otherwise specified, ablation experiments are conducted on TURL-Data by default. Moreover, some columns in TURL-data are annotated with types, which are used in the column typing warm-up. More statistics and other details of the datasets are listed in Appendix A.

4.2 Baselines

We consider three types of SOTA baselines, based on how they utilize contexts:

Cell-based: These baselines rely solely on the mention cell, without using any other contexts: (1) Wikidata API (**WK**): the Wikidata Lookup service, taking mentions as input; (2) OSS (**OS**) (Zhang et al., 2020a), a random forest with manual features between mentions and entities.

Text-based: There are four SOTA baselines designed for text EL: (1) BLINK (**BL**) (Ledell Wu, 2020): a two-stage method, with a bi-encoder for generating candidates, and a cross-encoder for reranking; (2) GENRE (**GE**) (De Cao et al., 2021): an auto-regressive method; (3) ReFinED (**RE**) (Tom Ayoola, 2022): a bi-encoder method; (4) ExtEnD (**EX**) (Barba et al., 2022): an extractive method. To adapt the input of these models, we serialize the table contexts into sequential texts, which is shown in Section 5.2.

Table-based: There are two baselines which directly process the structured context of tables: (1) TURL (**TU**) (Deng et al., 2020) encodes row and column contexts indiscriminately with a variant Transformer; (2) NPEL (**NP**) (Wu et al., 2024) encodes row and column contexts separately with GCN, and calibrates the predictions differently through a heuristic post-processing.

We use the official implementations of the baselines for our experiments, except for OSS and NPEL whose codes are unavailable. We implement OSS ourselves, and report the performance of NPEL from their paper directly.

Additionally, the performance of LLMs on this task is shown in Section 5.5. Due to the budget constraints, we only test on sampled data rather than the full.

4.3 Implementation Details

We initialize the BERT of RoCEL using BERT-large-uncased with 340M parameters. We employ the Adam (Kingma and Ba, 2015) optimizer with batch_size=16. We train the warm-up tasks with a learning rate of 2×10^{-3} for 4 epochs, and the entity linking task with a learning rate of 2×10^{-5} for 4 epochs. For model input, we use a maximum context length of 32 tokens. For Llama-3, we conduct experiments on a single 80GB A800 GPU, while for other models, we use four 32GB V100 GPUs. Hyper-parameters are determined through grid search. Experimental results are averaged over 3 different random seeds.

5 Experimental Results

In this section, we will conduct experiments to answer the following research questions:

- **RQ1:** How does our method perform compared to the SOTA baselines on both in-domain and out-domain datasets?
- **RQ2:** How does each kind of context affect the linking performance?
- **RQ3:** How should row and column contexts be best encoded?
- **RQ4:** How does the warm-up task for the column encoder contribute to our model?
- **RQ5:** How do LLMs perform on table entity linking?

5.1 Main Results

To answer **RQ1**, we compare the linking performance of our method with all the baselines. Following Wu et al. (2024), we report the accuracy (Acc). The result is shown in Table 1, where RoCEL is warmed up with set reconstruction (**R-S**), column typing (**R-C**), and both tasks together (**R-SC**), respectively.

Firstly, the cell-based methods perform relatively poorly, indicating that utilizing context is essential for this task. For the text-based methods, they all outperform the cell-based methods, but none of them could consistently outperform the others across all datasets.

Secondly, among the four table-based methods, although TURL surpasses the cell-based methods, it inferiors to the text-based methods. This stems from its coarse representation of table contexts, where row and column contexts are blended together. On the other hand, although the uncalibrated NPEL achieves only 69.4% and 74.0% accuracy on Wikilinks-R and Wikilinks-L respectively, the predictions with distinct calibrations for rows and columns surpass all text-based methods. This demonstrates the importance of context modeling

Dataset		Cell-based		Text-based				Table-based				
		WK	OS	BL	GE	RE	EX	TU	NP	R-S	R-C	R-SC
In-domain	TURL-Data	59.7	65.3	85.0	85.4	84.3	81.2	65.9	-	86.3	**86.9**	86.6
Out-of-domain	T2D	74.8	78.2	91.3	87.5	91.5	89.1	88.1	-	92.2	**92.6**	**92.6**
	Wikilinks-R	62.6	64.6	76.6	79.2	80.0	75.4	75.2	81.1	**84.4**	84.0	84.3
	Wikilinks-L	69.1	69.9	81.4	80.6	80.5	76.6	77.8	84.8	**86.5**	84.8	85.7
	Average	68.8	70.9	83.1	82.4	84.0	80.4	80.4	-	**87.7**	87.1	87.5

Table 1: Accuracy on four table entity linking benchmarks. **Bold** and underline indicate the best and second-best performance respectively.

by distinguishing the row context and the column context for TEL.

Finally, our method outperforms almost all SOTA baselines. Specifically, it outperforms the best baseline by 1.5% (e.g., 86.9% versus 85.4%) on in-domain dataset TURL-Data, and by 3.7% (i.e., 87.7% versus 84.0%) on the average of three out-of-domain datasets. The better performance compared to NPEL shows the effectiveness of our proposed RoCEL in learning table contexts, without a heuristic calibration. Furthermore, for RoCEL warmed up with different tasks, the one with column typing achieves better in-domain performance, while the other with set reconstruction performs better on average on out-of-domain datasets, indicating better generalization. The performance of using both warm-up tasks together (R-SC) lies between the performance of using each task individually. The reason might be that the optimization objectives of the two warm-up tasks are generally consistent, but have slight conflicts.

5.2 Study of different contexts

To answer **RQ2**, we conduct ablation studies with different combinations of contexts, and investigate the impact on linking performance.

Ablation Study on Our Method. To investigate the contribution of each kind of context (rows, columns, metadata, and headers) to our model, we remove one context at a time while keeping the others. When the row/column context is removed, the table remains only one column/row that contains the mention.

We report the accuracy on TURL-Data and T2D, as shown in Table 2. It can be observed that removing each kind of context leads to a performance drop. The removal of rows causes a greater performance decrease than columns, indicating that descriptional information in row contexts is more important for TEL than categorical information in column contexts. Additionally, metadata also contributes to the performance, since the non-tabular information can complement the table itself.

Context	TURL-Data	T2D
all contexts	**86.9**	**92.6**
- metadata	81.4	92.4
- headers	80.8	89.8
- rows	83.2	88.7
- columns	84.0	91.8

Table 2: Accuracy of RoCEL (warmed up with column typing), with the removal of different contexts.

Ablation study on Text-based Methods. To investigate the adaptability of text-based methods to tables, we provide surrounding cells of a mention in four layouts: (1) one-row: the row containing the mention; (2) one-column: the mention itself, along with k cells above and below; (3) cross: the combination of one-row and one-column; (4) multi-row: the row containing the mention, along with k rows above and below them. Constrained by input length, we set $k = 2$. Figure 5 visualizes these layouts. Metadata and headers are also provided. The structured contexts are serialized into texts by Equation 1, where each row is separated by "|".

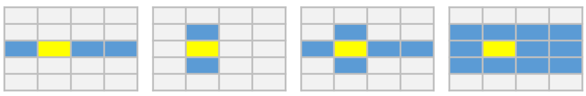

Figure 5: The layouts of the table contexts for text-based methods. Yellow cells are mentions, and blue cells are other provided cells.

The experimental results are shown in Table 3. It is observed that (1) The performance of one-row is better than one-column, further confirming that descriptional information is more important than categorical information for this task; (2) The performance of cross and multi-row is worse than one-row, indicating that these models struggle to utilize additional row and column contexts. We attribute this to the misalignment of cells when flattening a two-dimensional table into one-dimensional text, making it difficult to restore the table semantics.

Layout	**Method**			
	BL	GE	RE	EX
One-row	**85.0**	**85.4**	**84.3**	**81.2**
One-column	81.4	81.8	83.1	79.7
Cross	84.2	85.1	84.1	81.0
Multi-row	83.4	84.7	84.0	80.5

Table 3: Accuracy of text-based methods, with contexts of different layouts.

5.3 Study of Row and Column Encoders

To answer **RQ3**, we apply set-wise and attention-based encoders in different combinations to row and column contexts. Specifically,

• When applying an attention-based encoder on the column context, we convert it into a sequential text like the row context:

$$\mathrm{p}'_{kj} = \begin{cases} T_{Hj} : \text{[START]}\ T_{kj}\ \text{[END]}\,, & k = i, \\ T_{Hj} : T_{kj}, & k \neq i, \end{cases}$$
$$\mathrm{txt}'_{ij} = \text{[CLS]}\ \ T_{\Theta} \mid \mathrm{p}_{1j}\ ; \cdots ; \mathrm{p}_{Ij}\ \text{[END]}\,,$$

, where $1 \leq k \leq I$.

• When applying set-wise encoder on the row context, the metadata T_{Θ}, the mention "[CLS] T_{Hj} : [START] T_{ij} [SEP]", and the other pairs "[CLS]T_{Hk} : T_{ik} [SEP]" $(k \neq j)$ in the row context, are firstly encoded into vectors. Then we treat the vectors as an unordered set, and aggregate them into a single vector using FSPool.

Table 4 shows the performance of different row and column encoders, including using different encoders, using the same encoder, and using each encoder separately. It can be observed that: (1) Applying an attention-based encoder to column contexts yields slightly better results than without a column encoder, but it is not as effective as a set-wise encoder. This indicates that column contexts are better modeled as unordered sets rather than sequential texts. (2) Applying a set-wise encoder to row context performs worse than without a row encoder. The reason is that the set-wise encoder equally treats each cell in the row context, making it difficult to distinguish necessary information from noise. The column encoder then takes these noisy embeddings as input, further increasing the encoding difficulty. Therefore, it is better to encode row contexts with an attention-based encoder.

Row-encoder	**Column-encoder**	TURL-Data	T2D
attention	set-wise	**86.9**	**92.6**
attention	×	84.0	91.8
attention	attention	85.2	92.1
×	set-wise	83.2	88.7
set-wise	set-wise	79.5	81.1

Table 4: Accuracy of RoCEL (warmed up with column typing), with different row and column encoders.

5.4 Impact of Column Encoder Warm-up

We answer **RQ4** from two aspects: (1) whether the warm-up tasks enhance linking performance; and (2) whether they help to capture categorical information in column contexts. To this end, we use 1%, 10%, and 100% of the data to warm up the column encoder and auxiliary modules.

Impact on Entity Linking. Figure 6 shows the linking accuracy of using two warm-up tasks and without warm-up. We find that: (1) Compared to no warm-up, both tasks can enhance linking performance, and the improvement increases with the amount of warm-up data. (2) Column typing is more data-efficient than set reconstruction. The former achieves higher accuracy with only 1% distant-labeled data compared to the latter with 100% unlabeled data. However, set reconstruction remains valuable when the types are unknown.

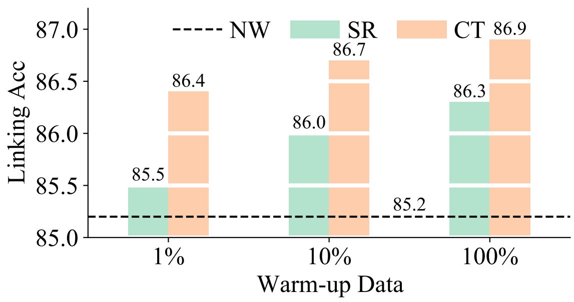

Figure 6: Linking accuracy of RoCEL. NW, SR, and CT stand for no warm-up, set reconstruction, and column typing, respectively.

Impact on Column Representation. We use the column typing task to evaluate the representation of categorical information in columns. For column typing warm-up, we directly report the classification F_1. For set reconstruction warm-up, we freeze BERT and the column encoder after the warm-up, replace the set decoder FSUPool with a Linear layer, and train it until convergence with the loss $\mathcal{L}^{\mathrm{CT}}$ and 1% of column typing data. The experimental results are shown in Figure 7. The F_1 of both models reach around 80% or higher, showing they can somewhat capture column types.

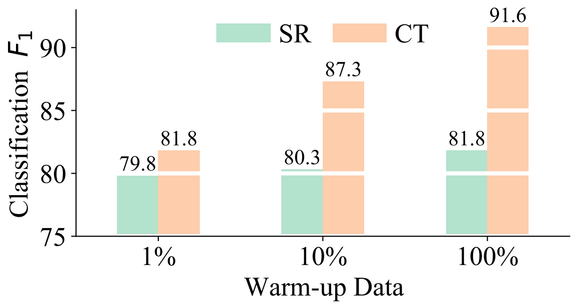

Figure 7: Column typing F_1 of RoCEL. SR and CT stand for set reconstruction and column typing, respectively.

5.5 LLM Performance

To answer **RQ5**, we evaluate the capability of a representative LLM Llama-3-8B-Instruct (AI@Meta, 2024b) on this task, specifically in the following three aspects:

• **Context Utilization.** To investigate whether Llama-3 can effectively utilize table contexts, we provide contexts with one-row and multi-row layouts as described in Section 5.2. Metadata and headers are also provided.

• **Table Format.** Since Llama-3 is a text-based method, we follow Sui et al. (2024) to explore which format of the sequential text is more suitable for serializing tables. For the multi-row layout, we include four formats: plain-text (same as the input of other text-based methods in Section 5.2), JSON-lines, Markdown, and HTML. For the one-row layout, we only consider plain-text.

• **Task Modeling.** Following Anonymous (2024), we model the task in two ways: (1) generating the Wikipedia URL directly, which contains the entity name; (2) reranking top-10 candidates from the retriever of BLINK. The prompts are shown in tables 10 and 11 in Appendix C.

We sample one mention from each table, resulting in a total of 4k mentions, each has a ground truth entity within the candidates. We do inference only, and report the accuracy in Table 5. Note that in over 97.5% of cases, Llama-3 successfully follows instructions and provides meaningful predictions. We can observe that: (1) For context utilization, the performance under multi-row shows a significant decline compared to one-row. It indicates that Llama-3 also struggles to utilize table contexts like other text-based methods. (2) For table formats under multi-row layout, plain-text and JSON-lines perform similarly, and are significantly better than HTML and Markdown. This indicates that in this task, it is easier for Llama-3 to understand simple and natural-language-like formats, rather than complex and markup-language-like formats. (3) For task modeling, the accuracy of generating URLs is higher than reranking candidates, which is consistent with Anonymous (2024). One possible reason is that the reranking prompts are longer and more complex, making them harder to understand. Moreover, Llama-3 performs significantly worse than task-specific methods.

Layout	**Format**	**Modeling**	
		Generate	Rerank
One-row	Plain-text	**58.7**	52.5
Multi-row	Plain-text	**57.3**	41.1
	JSON-Lines	**58.5**	40.7
	Markdown	**54.1**	28.6
	HTML	**52.1**	33.7

Table 5: Accuracy of Llama3 across different context layouts, table formats, and task modelings.

Model	**Accuracy**	**Finetuned**
RoCEL	86.9	✓
Llama3	52.5	×
Llama3 (Generate)	58.7	×
Llama2*	31.8	×
TableLlama*	**93.7**	✓
GPT-3.5*	72.2	×
GPT-4*	90.8	×

Table 6: The comparison with various LLMs. Results marked with * are cited from Zhang et al. (2024a).

We also compare RoCEL with other LLMs. Most of the results are cited from a recent work, TableLlama (Zhang et al., 2024a). It is a finetuned Llama2-7B (AI@Meta, 2024a) using various table tasks (including TEL). The performance on TURL-data is shown in Table 6. All models rerank candidate entities except for Llama3 (Generate), which directly generates Wikipedia URLs. It can be seen that although RoCEL is not as good as GPT-4 and the finetuned TableLLaMA, it outperforms

GPT-3.5, and non-finetuned Llama2 and Llama3. Additionally, RoCEL has fewer parameters than the LLMs (0.34B vs ≥7B) and faster inference speed, making it competitive in low-resource scenarios.

6 Related Works

Early work on TEL typically employs heuristic methods or traditional machine learning models. For example, Ritze et al. (2015) proposes a rule-based approach that iteratively maps entity mentions or column types to a knowledge base. Bhagavatula et al. (2015) introduces a graph-based method. Efthymiou et al. (2017) presents a hybrid linking method that combines rule-based exact matching and embedding-based semantic matching. Zhang et al. (2020a) trains a random forest with lexical and semantic features. However, these methods rely heavily on human expertise to utilize table contexts, thus lacking flexibility.

Recently, several neural networks have been proposed for automatically understanding tables (Dong et al., 2022). These methods can be divided into two categories, namely text-based and table-based methods. Text-based methods serialize tables into sequential texts to leverage models for textual data. For example, TableGPT (Gong et al., 2020) employs GPT-2 (Radford et al., 2019) and convert tables into natural language sentences through templates; GPT4Table (Sui et al., 2024) adopts GPT-4 (OpenAI, 2023) and serializes tables into markup languages, such as HTML and JSON. In contrast, table-based methods are specialized model architectures for structured tables. For example, TAPAS (Herzig et al., 2020) is a variant of Transformer with row and column position embeddings, which can mark the coordinates of each cell; TUTA (Wang et al., 2021) is another variant with a bi-dimensional tree. These methods target different tasks, such as table classification (Wang et al., 2021), table question answering (Herzig et al., 2020; Sui et al., 2024), and table-to-text generation (Gong et al., 2020). However, none of them can handle all these tasks.

The most relevant works to our research are TURL (Deng et al., 2020) and NPEL (Wu et al., 2024), both are table-based methods for TEL. TURL is a variant of Transformer with a masked self-attention mechanism. In this model, each cell interacts only with the cells in its row and column, thus implicitly encoding the structure of a table. However, it does not distinguish whether the cells are from the same row or column, which ignores their semantic differences. NPEL, a contemporaneous work with ours, recognizes this issue but addresses it through a non-learnable post-processing. Specifically, it represents row and column contexts separately, but employs GCN (Kipf and Welling, 2017) uniformly as the encoder. Then in the post-processing stage, it heuristically calculates entity coherence within rows and columns in two different ways, thereby globally calibrating the predictions. In contrast, we deal with semantic differences by encoding rows and columns in different manners from the beginning, without post-processing.

Some works try to employ LLMs in table-related tasks, such as TableQA (Zhang et al., 2024b), table entity matching (Li et al., 2024), and table-to-text generation (Sundararajan et al., 2024). The survey (Lu et al., 2024) summarizes the application of LLMs on tabular data. However, there are very few works targeting on TEL. The most relevant LLM-based work to ours is TableLlama (Zhang et al., 2024a). It constructs a dataset with various table-related tasks including TEL, and finetunes LLama2 (AI@Meta, 2024a) to obtain the TableLlama.

7 Conclusion

In this work, we propose a table entity linking method RoCEL, in which the row and column contexts are encoded in two manners due to their semantic differences. Our approach outperforms all SOTA baselines on four benchmark datasets, demonstrating its better utilization for table contexts. In contrast, further evaluations of text-based methods including Llama-3 indicate their difficulty in leveraging table contexts.

For future work, we plan to achieve a global entity linking method by imposing different constraints on the predictions within rows and columns. Additionally, the warming up of the row encoder remains to be explored, since we only warm up the column encoder.

8 Limitations

There are some limitations of this work in methodology and experiments:

Methodology: (1) We focus only on relational tables with simple structures, and do not extend to more complex tables in the real world, such as nested tables. (2) The cells within a row are order-irrelevant. However, an order is introduced to the cells when the row context is serialized into text.

This partially contradicts the nature of tables.

Experiments: (1) Due to computational and accessibility limitations, we only evaluate Llama-3-8B-instruct, and do not include other LLMs like GPT-4 (OpenAI, 2023) and Mistral (Jiang et al., 2023). (2) As the emphasis of this work is not on LLMs, we only conduct basic evaluations without further prompt engineering or tool augmentation.

9 Ethical Considerations

In this section, we briefly discuss the ethical considerations of our work.

Privacy. Entity linking involves real-world named entities, including people, places, and organizations, which cannot be anonymized in this study. The datasets are created by Deng et al. (2020); Lehmberg et al. (2016); Wu et al. (2024), which are widely adopted in previous research. These datasets are collected from publicly available Internet, including Wikipedia and various other websites. However, it is not guaranteed that these public data are free from privacy breaches.

Languages. The data in this study is in English.

Intended use. Our method is intended for table entity linking and other related tasks, such as table question answering, table retrieval, and knowledge base population. The datasets we used are all created for table entity linking. All baselines are compatible with this study, although some are designed for table EL and others for text EL.

Environmental Impact. The training and inference of models result in energy consumption and carbon emissions. However, compared to LLMs, our method has fewer parameters and requires less computation, thus consuming less energy.

Licenses and terms. The licenses of the knowledge base (KB), codes, and datasets discussed in this paper are listed in Table 7.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62372431 and 62472408, the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, XDB0680301, the National Key Research and Development Program of China under Grants No. 2023YFA1011602 and 2021QY1701, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039.

	Resource	License
KB	Wikipedia	CC BY-SA 4.0
Dataset	TURL-Data	Apache 2.0
	T2D	Apache 2.0
	Wikilinks-R	not found
	Wikilinks-L	not found
Code	BLINK	MIT
	GENRE	CC BY-NC 4.0
	ReFinED	CC BY-NC 4.0
	ExtEnD	CC BY-NC-SA 4.0
	TURL	Apache 2.0

Table 7: Licenses of the knowledge base, datasets, and codes in this work.

References

AI@Meta. 2024a. Llama 2 model card.

AI@Meta. 2024b. Llama 3 model card.

Anonymous. 2024. Llmael: Large language models are good context augmenters for entity linking.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2022. ExtEnD: Extractive entity disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Online and Dublin, Ireland. Association for Computational Linguistics.

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: Entity linking in web tables. In *The Semantic Web - ISWC 2015*, pages 425–441, Cham. Springer International Publishing.

Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Jaydeep Sen, Mustafa Canim, Soumen Chakrabarti, Alfio Gliozzo, and Karthik Sankaranarayanan. 2021. Topic transferable table question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4159–4172, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: table understanding through representation learning. *Proceedings of the VLDB Endowment*, 14(3):307–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. 2022. Table pre-training: A survey on model architectures, pre-training objectives, and downstream tasks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5426–5435. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. 2017. Matching web tables with knowledge base entities: From entity lookups to entity embeddings. In *The Semantic Web – ISWC 2017*, pages 260–277, Cham. Springer International Publishing.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. TableGPT: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

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 Companion on World Wide Web*, WWW '16 Companion, page 75–76, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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).

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. *Journal of Web Semantics*, 76:100761.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2024. Large language model for table processing: A survey. *Preprint*, arXiv:2402.05121.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Dominique Ritze, Oliver Lehmberg, and Christian Bizer. 2015. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS '15, New York, NY, USA. Association for Computing Machinery.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Barkavi Sundararajan, Yaji Sripada, and Ehud Reiter. 2024. Improving factual accuracy of neural table-to-text output by addressing input problems in ToTTo. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7350–7376, Mexico City, Mexico. Association for Computational Linguistics.

Joseph Fisher Christos Christodoulopoulos Andrea Pierleoni Tom Ayoola, Shubhi Tyagi. 2022. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *NAACL*.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

Tianxing Wu, Lin Li, Huan Gao, Guilin Qi, Yuxiang Wang, and Yuehua Li. 2024. Npel: Neural paired entity linking in web tables. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* Just Accepted.

Shuo Zhang, Edgar Meij, Krisztian Balog, and Ridho Reinanda. 2020a. Novel entity discovery from web tables. In *Proceedings of The Web Conference 2020*, WWW '20, page 1298–1308, New York, NY, USA. Association for Computing Machinery.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024a. TableLlama: Towards open large generalist models for tables. In *NAACL*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2020b. Fspool: Learning set representations with featurewise sort pooling. In *International Conference on Learning Representations*.

Zhehao Zhang, Yan Gao, and Jian-Guang Lou. 2024b. e^5: Zero-shot hierarchical table analysis using augmented LLMs via explain, extract, execute, exhibit and extrapolate. In *NAACL*, pages 1244–1258, Mexico City, Mexico. Association for Computational Linguistics.

A Dataset Details

Statistics of the datasets are shown in Table 8, including the table resources, usage, and table number of each dataset.

Deng et al. (2020) distantly annotate the types of certain columns in the TURL-data dataset. Specifically, they only consider columns with at least three linkable mentions. For such a column, they retrieve the FreeBase types of each target entity. The shared types of these entities are then labeled as the types of the column. These columns involve a total of 255 FreeBase types. Since an entity can have multiple types, a column can also have multiple types. This makes column typing a multi-class, multi-label classification task.

Dataset	Table Resource	Usage	Table Number
TURL-data	Wikipedia	train	570k
		valid	5k
		test	5k
T2D	Web pages	test	233
Wikilinks-R	Wikipedia	test	168
Wikilinks-L	Wikipedia	test	2.3k

Table 8: Dataset statistics.

B Headers and Column Types

The relationships between headers and column types are diverse and complex. Table 9 are some examples of real-world tables, each annotated with the column types. In these examples, we can discover several patterns in the relationship between headers and types (not limited to these patterns):

- The header is the same as the type name, such as the header and the type "Album" in Table 9c.
- The header is semantically similar to the type name, but in different words. For instance, the header "Nationality" in Table 9c, where the standard type name is "Country".
- The granularity of the header is coarser than the type. For example, the header "Artist/s" in Table 9c, where the column type is finer-grained "Musician/s".
- The header itself is completely unrelated to the type. For instance, the header "Name" with the type "Museum" in Table 9a, and the header "From" with the type "Football Team" in Table 9b.

C LLM Prompts

Our prompts are adapted from Anonymous (2024), which are originally proposed for text EL rather than table EL. Table 10 and 11 show examples of our prompts and responses of Llama-3. The prompt in Table 10 directly generates the Wikipedia URL, while the prompt in Table 11 reranks the top-10 retrieved candidates. Each prompt consists of a description of the task, three in-context examples (only one is shown), and a query. The in-context example and query share similar formats, both including the serialized table contexts and table caption. The in-context example additionally contains an answer (and an explanation for the reranking prompt), while the query does not include an answer or explanation.

Museum	*Town/City*	
Name	**Town/City**	**Type**
Alnwick Castle	Alnwick	Multiple
Aydon Castle	Aydon	Historic house
Bailiffgate Museum	Alnwick	Local

(a) List of museums in Northumberland.

Football Player	*Football Team*	*Sports League*	
Player	**From**	**League from**	**Date**
Adrian Caceres	Perth Glory	A-League	13 February 2006
Claudinho	Atlético Paranaense	Brazil Serie A	24 May 2006
Fred	Guarani	Brazil Serie B	24 May 2006

(b) 2006–07 Melbourne Victory season. Transfers.

	Album	*Musician/s*	*Country*	
Release Year	**Album**	**Artist/s**	**Nationality**	**Worldwide sales (in millions)**
2008	The Fame	Lady Gaga	United States	18.0
2003	Fallen	Evanescence	United States	17.0
2002	Let Go	Avril Lavigne	Canada	16.0

(c) List of best-selling albums of the 21st century, 13–19 million copies:

Table 9: Some examples of real-world tables. *Italics* and **bold** indicate the types and headers respectively. Non-entity columns are not labeled with types.

Prompt:
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Gives a JSONLINES-like format table, the caption of the table, and a mention within the table highlighted by <MENTION> and </MENTION>.Please give which page in Wikipedia this mention is most likely to be? Please answer me directly in this form: ”mention”:”Wikipedia page url”.

Table:
{’home team’: ’Collingwood’, ’venue’: ’Victoria Park’, ’crowd’: ’22,519’, ’date’: $\cdots$ }
{’home team’: ’Richmond’, ’venue’: **’<MENTION> MCG </MENTION>’**, ’date’: $\cdots$ }
{’home team’: ’Melbourne’, ’venue’: ’VFL Park’, ’crowd’: ’31,535’, ’date’: $\cdots$ }
{’home team’: ’Hawthorn’, ’venue’: ’Princes Park’, ’crowd’: ’15,081’, ’date’: $\cdots$ }
{’home team’: ’Geelong’, ’venue’: ’Kardinia Park’, ’crowd’: ’26,574’, ’date’: $\cdots$ }
Caption: 1984 VFL season. Round 4.
Answer: ”MCG”: ”https://en.wikipedia.org/wiki/Melbourne_Cricket_Ground”

<|eot_id|><|start_header_id|>user<|end_header_id|>

Table:
{’year’: ’1956’, ’coach’: ’Forest Evashevski’, ’overall record’: $\cdots$ }
{’year’: ’1958’, ’coach’: ’Forest Evashevski’, ’overall record’: $\cdots$ }
{’year’: ’1960’, ’coach’: **’<MENTION> Forest Evashevski </MENTION>’**, ’overall record’: $\cdots$ }
{’year’: ’1981’, ’coach’: ’Hayden Fry’, ’conference record’: ’6–2–0’, ’overall record’: $\cdots$ }
{’year’: ’1985’, ’coach’: ’Hayden Fry’, ’conference record’: ’7–1–0’, ’overall record’: $\cdots$ }
Caption: Iowa Hawkeyes football. Conference championships.
Answer:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Response:
Forest Evashevski”: ”https://en.wikipedia.org/wiki/Forest_Evashevski

Table 10: An example of the prompt and model response that directly generates the Wikipedia URL. **’<MENTION> ... </MENTION>’** marks the mentions.

Prompt:
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Gives a JSONLINES-like format table, the caption of the table, and a mention within the table highlighted by <MENTION> and </MENTION>.Please select from the options below which Wikipedia page this mention is most likely to be from? Please answer me directly in this form: "(letter): Wikipedia entity name and url. And I also want you to give an explanation in the next line.

Options:
(a): ['Docklands Stadium', 'https://en.wikipedia.org/wiki?curid=541428', 'stadium in Melbourne, Victoria, Australia']
...
(j): ['Melbourne Cricket Ground', 'https://en.wikipedia.org/wiki?curid=19765', 'Stadium in Melbourne, Victoria, Australia']
Table:
{'home team': 'Collingwood', 'venue': 'Victoria Park', 'crowd': '22,519', 'date': ··· }
{'home team': 'Richmond', 'venue': **'<MENTION> MCG </MENTION>'**, 'date': ··· }
{'home team': 'Melbourne', 'venue': 'VFL Park', 'crowd': '31,535', 'date': ··· }
{'home team': 'Hawthorn', 'venue': 'Princes Park', 'crowd': '15,081', 'date': ··· }
{'home team': 'Geelong', 'venue': 'Kardinia Park', 'crowd': '26,574', 'date': ··· }
Caption: 1984 VFL season. Round 4.
Answer: (j): ['Melbourne Cricket Ground', 'https://en.wikipedia.org/wiki?curid=19765']
Explanation: For mention of <MENTION> MCG </MENTION>, the most similar option is (j) Melbourne Cricket Ground. MCG is the abbreviation for (j) Melbourne Cricket Ground. Moreover, this table is about sport events, with MCG as a venue, and (j) Melbourne Cricket Ground is a sports stadium, matching the type with MCG, making it the most likely source.

<|eot_id|><|start_header_id|>user<|end_header_id|>

Options:
(a): ['Forest Evashevski', 'https://en.wikipedia.org/wiki?curid=5779566', 'American football player and coach']
...
(j): ['Ed Stefanski', 'https://en.wikipedia.org/wiki?curid=14556015', 'American basketball player']
Table:
{'year': '1956', 'coach': 'Forest Evashevski', 'overall record': ··· }
{'year': '1958', 'coach': 'Forest Evashevski', 'overall record': ··· }
{'year': '1960', 'coach': **'<MENTION> Forest Evashevski </MENTION>'**, 'overall record': ··· }
{'year': '1981', 'coach': 'Hayden Fry', 'conference record': '6–2–0', 'overall record': ··· }
{'year': '1985', 'coach': 'Hayden Fry', 'conference record': '7–1–0', 'overall record': ··· }
Caption: Iowa Hawkeyes football. Conference championships.
Answer:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Response:
(i): ['Forest Evashevski', 'https://en.wikipedia.org/wiki?curid=5779566']
Explanation: For mention of <MENTION> Forest Evashevski </MENTION>, the most suitable option is (i) Forest Evashevski. The mention Forest Evashevski can exactly match option (i) Forest Evashevski. Moreover, this table is about the Iowa Hawkeyes football team, and Forest Evashevski was the coach of the team, making it the most likely source.

Table 11: An example of the prompt and model response that reranks the top-10 retrieved candidates. **'<MENTION> ... </MENTION>'** marks the mentions.