A Retrieval-Augmented Framework for Tabular Interpretation with LLM

Mengyi Yan[1] Weilong Ren[2*] Yaoshu Wang[2] Jianxin Li[1*]

[1]Beihang University [2]Shenzhen Institute of Computing Sciences

北京航空航天大學 BEIHANG UNIVERSITY

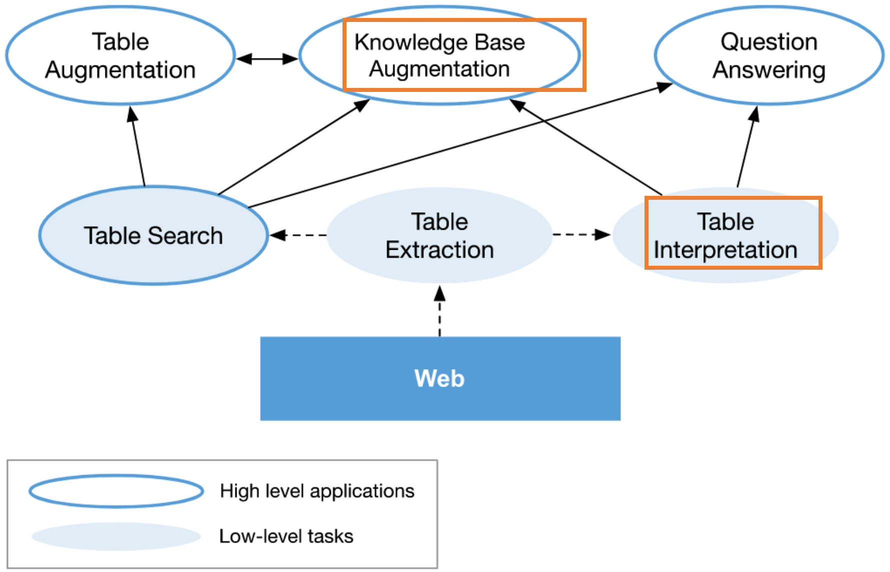
深圳计算科学研究院 Shenzhen Institute of Computing Sciences

Outline

- Background & Motivation
- Problem Definition
- Challege & Solution
- Our framework
  - Retrieval Module $RAFL_{ret}$
  - Re-Ranking System $RAFL_{rank}$
- Experiments
- Conclusion

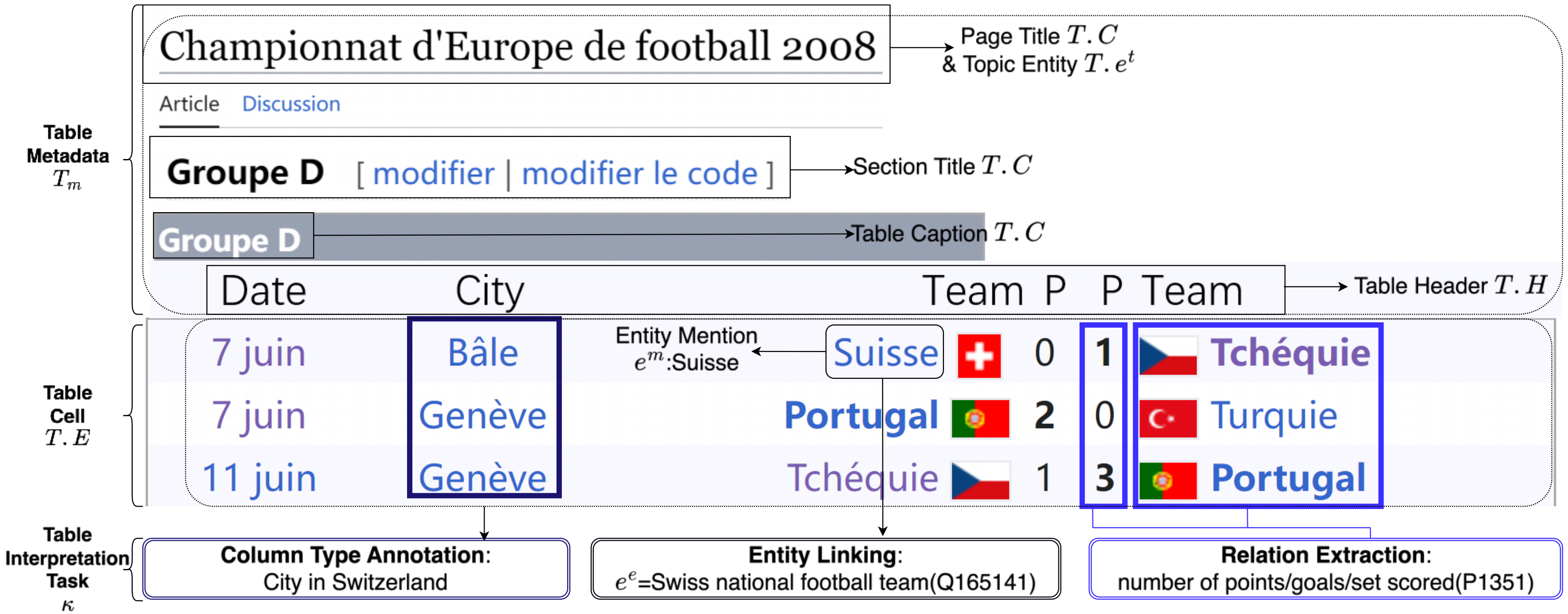
Background

- Table Interpretation: understanding schema-free web tables
  - Goal: Uncover the semantic attributes in relational tables
  - Method: Mapping webtable data(e.g. column/cell) into the node/relation in Knowledge Graph $\mathcal{G}$
  - Task: Column Type Annotation(CTA), Relation Extraction(RE), Entity Linking(EL)



Zhang, Shuo, and Krisztian Balog. "Web table extraction, retrieval, and augmentation: A survey." *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.2 (2020): 1-35.

A Real-World Case of Tabular Interpretation for webtable



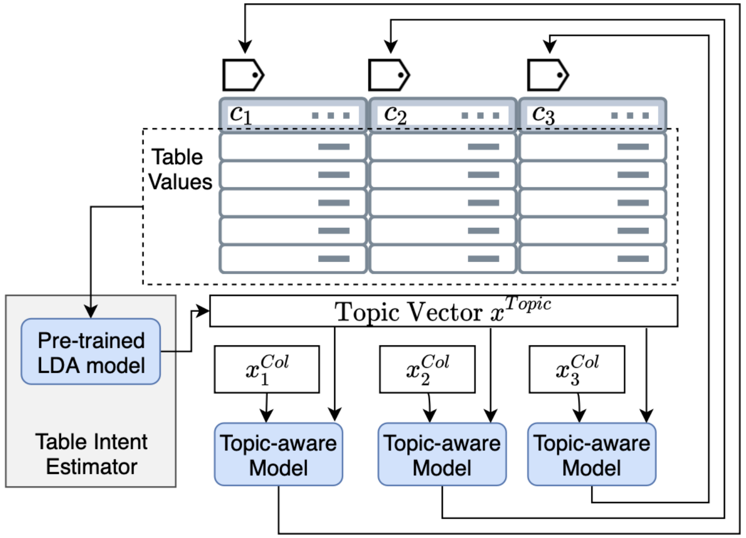
Difference from relational table:

- Various schema-free subtable(e.g. 200k tables for WikiGS Dataset)
  - How to Retrieve similar tables?
- Each table has different metadata
  - How to Cooperate metadata?
- Close relation with Knowledge Graph(i.e. KG entity/relation)
  - How to map and rank KG relations/nodes?

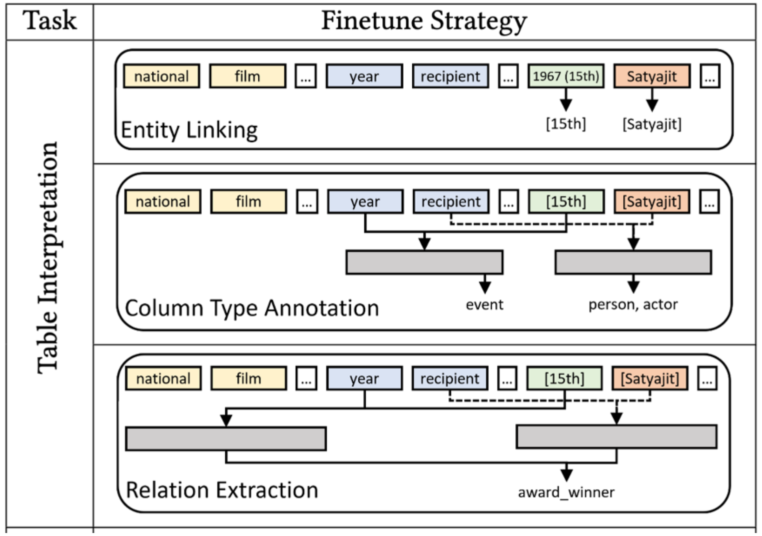
- Column Type Annotation (**CTA**): refers to deciding the column type for column **CITY**;
  - Column type are selected from a pre-defined semantic type set $l \in \mathcal{L}$
- Entity Linking(**EL**): refer to choosing the KG entity linked with cell **Suisse;**
- Relation Extraction(**RE**): refer to decide the KG relation for column pair (**Team-P**).
  - Relation type are selected from a pre-defined relation type set $r \in \mathcal{R}$

https://fr.wikipedia.org/wiki/Championnat_d%27Europe_de_football_2008# 1er_tour_- _phase_de_groupes.

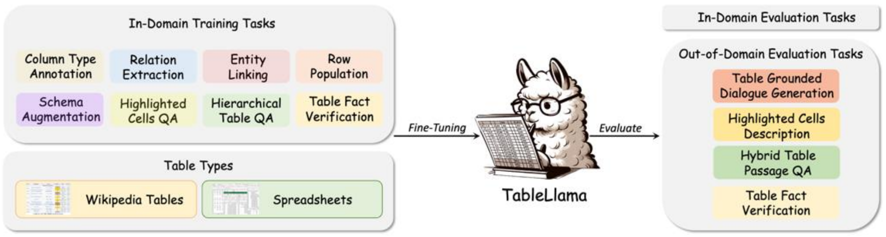
Motivation: Can language model understand webtable well?



2019-SATO[1]: Topic-aware LDA model



2022-TURL[2]: Representation Learning with PLM



2024-TableLLAMA[3]: Unified Generative Method with LLM

- Previous works on language model cannot solve the tabular interpretation task well.
  - ➢ Limited capability of retrieving and incorporating inter-table context
  - ➢ Inadequate ability in handling large tables
  - ➢ PLMs are hard to read tables reliably
- We believe that LLM can be adopted to solve the table interpretation task if we use it properly.
  - ➢ LLM can process a longer query than traditonal PLMs
  - ➢ LLM can read a whole table with additional inter-table contexts
  - ➢ LLM is pretrained on a variety of corpus

1. Zhang, Dan, et al. "Sato: Contextual semantic type detection in tables." *Proceedings of the VLDB Endowment*, *13*(11) 2019.
2. Deng, Xiang, et al. "Turl: Table understanding through representation learning." *ACM SIGMOD Record* 51.1 (2022): 33-40.
3. Zhang, Tianshu, et al. "TableLlama: Towards Open Large Generalist Models for Tables." *NAACL (Volume 1: Long Papers)*. 2024.

Problem definition of table interpretation

- Input:
  - a relational web table T in webtable dataset $\mathcal{T}$
  - a large language model M_G
  - a knowledge graph $\mathcal{G}$
  - a specific task κ
  - the task-related information T^{κ}, instruction Ins^{κ}, and a set D^{κ} of related demonstrations
  - top-k options O^{κ}
- Output
  - an element $o^{\kappa} \in O^{\kappa}$, as the final selection

Challenges

- How to search for related tables from a variety of sub-tables set?
- How to measure structural and semantic similarity among schema-free tables?
- How to alleviate the hallucination problem of LLM?

Our solution

- How to search for related tables from a variety of sub-tables set?
  - ✓We apply a retrieval-augmented module to search related table set from a variety of corpus, in a unified embedding space
- How to measure structural similarity among schema-free tables?
  - ✓We introduce an auxiliary graph structure to measure structural similarity.
- How to alleviate the hallucination problem of LLM?
  - ✓We use pre-ranking model to restrict options and demonstrations, and treat LLM as a re-ranking model.

Our framework: RAFL

1. Pre-Ranking Model($RAFL_{ret}$)

- **Input:** A schema-free web table $T \in \mathcal{T}$, an annotated training set T_{train}, a knowledge graph $\mathcal{G}$
- **Output**: Related table set $T_{related}$ with self-annotation; pre-ranking top-k options O for T

2. Re-Ranking Model($RAFL_{rank}$ with LLM)

- **Input:** Specific task $\kappa \in \{CTA, RE, RL\}$, Instruction Ins^{κ} for task κ, demonstration D^{κ} from $T_{related}$, top-k options O^{κ} for T.
- **Output**: Selection $o^{\kappa} \in O^{\kappa}$ by LLM as re-ranking model.

The light weighed pre-ranking model $RAFL_{ret}$

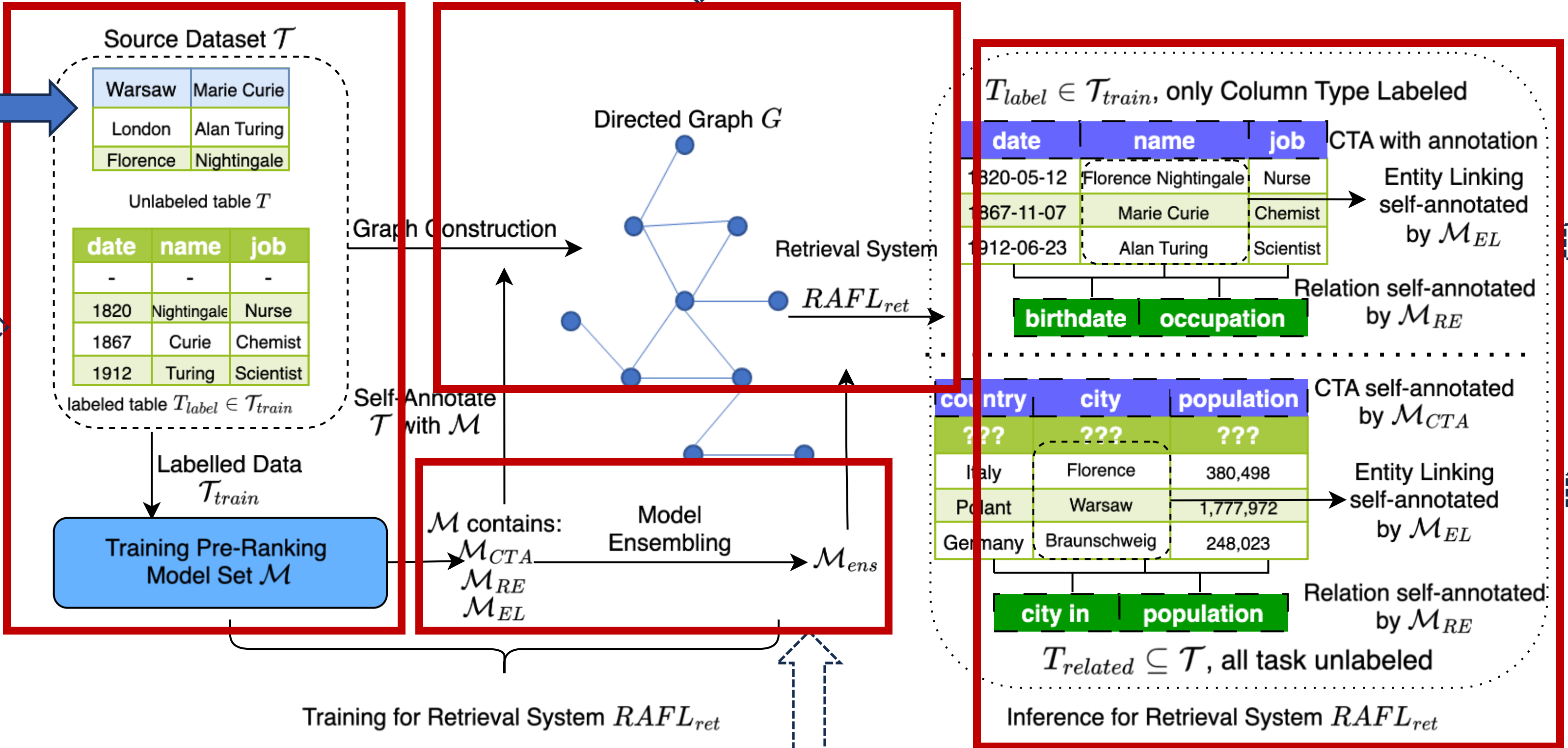
- Restrict different header to pre-defined limited semantic type $\mathcal{L}, \mathcal{R} \in \mathcal{G}$
- Different sub-table T are concatenated to a unified directed graph G
- GSL to learn structural similarity

3. Graph Construction and Graph Structural Learning

Input: Unlabeled sub-table T

1. Bi-level Ranking Model M:

- Cell-Level and Column-Level Serialization
- Trained by Contrastive Learning



Semantic Similar

Related Table Set Retrieval:
Graph Embedding Similarity + Semantic Embedding Similarity

Structural Similar

4. Similarity Calculation

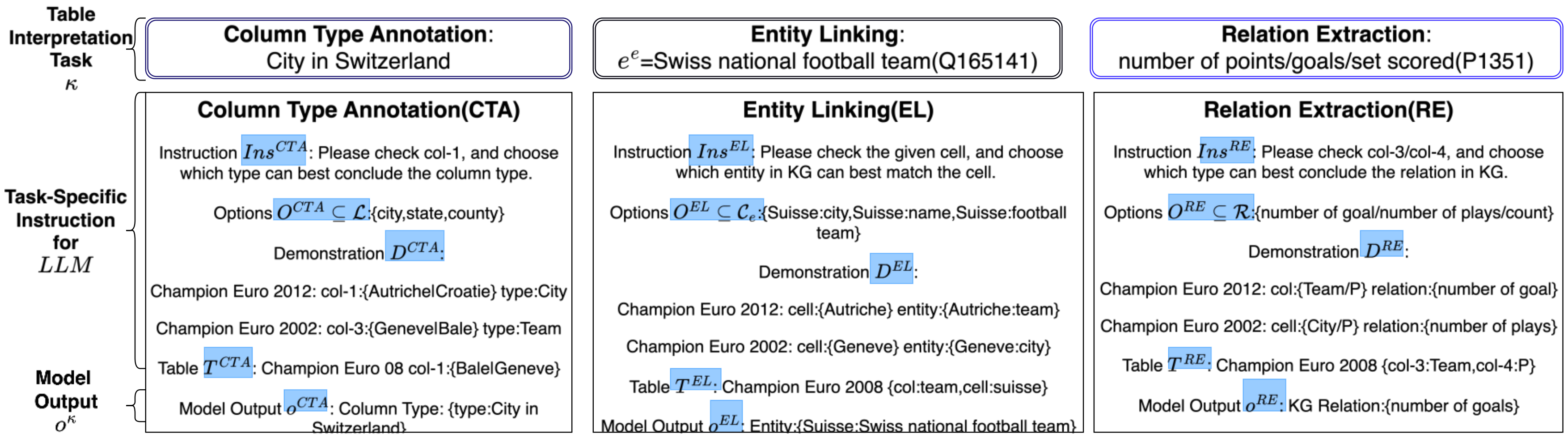
2. Self-Annotation With Ensembled Model

- Model Ensemble
- Calculate Semantic Similarity

Output:
Related Table Set $T_{related}$ for T
Pre-Ranking Options O for T

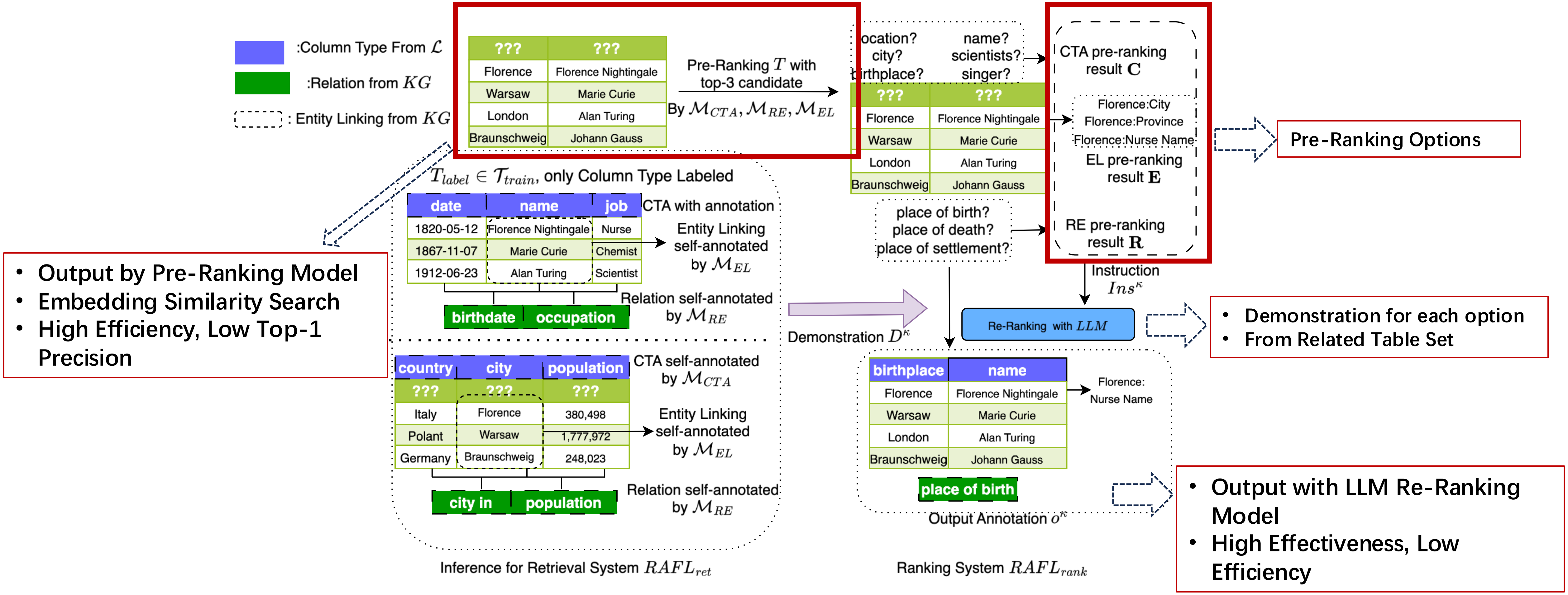
Re-Ranking System $RAFL_{rank}$

- **Avoiding Hallucination of LLM:**
  - ➢ LLM cannot select the correct annotation from hundreds of sematic type set $\mathcal{L} \cup \mathcal{R}$. (Limited Input Token Length)
  - ➢ LLM cannot understand the meaning of each semantic type $l \in \mathcal{L}$ (resp. $r \in \mathcal{R}$) without demonstration.
- **How do RAFL solve such issue?**
  - ➢ **Restrict Selection Domain**: to avoid hallucination, LLM is restricted to select from pre-ranking options O^{κ} from M_{ens}.
  - ➢ **RAG Paradigm**: LLM is also provided with the most related self-annotated table corpus $T_{related}$ as task-specific demonstration, as illustration



Examples of Input and Output for $RAFL_{rank}$

The LLM-based re-ranking model $RAFL_{rank}$



Experiments

- LLM-backboned model: Mistral-7B, Vicuna-13B
  - Fine-Tuned with LLaMA-Factory[1], Inference with vLLM[2]
- RAG Model: bge-large-en
- Non-LLM Baseline:
  - Sherlock/TaBERT/TABBIE/DUDUO/RECA
- LLM Baseline
  - TableLLAMA(applies a 7B LLM model, and pre-trained on millions of tabular data.)
- Metrics:
  - Micro-F1(Overall Evaluation of prediction result)
  - Macro-F1(Prediction Accuracy of Minority Semantic Type Class)
- Hardware:
  - 4 V100 GPU

1. http://github.com/hiyouga/LLaMA-Efficient-Tuning
2. https://github.com/vllm-project/vllm

Experiment: Main Result

Table 2: Results of task CTA on dataset Semtab2019/WebTables

Model	Semtab2019 Micro F1	Semtab2019 Macro F1	WebTables Micro F1	WebTables Macro F1
Sherlock (100%)	0.646	0.440	0.844	0.670
TaBERT (100%)	0.768	0.413	0.896	0.650
TABBIE (100%)	0.799	0.607	0.929	0.734
DODUO (100%)	0.820	0.630	0.928	0.742
RECA(25%)	0.697	0.442	0.909	0.680
RAFL (25%)	0.861	0.743	0.963	0.825
RECA(100%)	0.853	0.674	0.937	0.783
RAFL (100%)	**0.875**	**0.766**	**0.967**	**0.834**

Table 4: Results of task RE and EL on dataset WikiGS

Model	WikiGS-RE Micro F1	WikiGS-RE Macro F1	WikiGS-EL Accuracy
TURL(10%)	0.7350	0.3088	0.6055
RAFL (10%)	0.8930	0.8365	0.8705
TURL(25%)	0.8601	0.6755	0.7394
RAFL (25%)	0.9295	0.8642	0.8861
TURL(100%)	0.9025	0.8016	0.8420
RAFL (100%)	**0.9323**	**0.9153**	**0.9112**
GPT-4	0.5295	0.4326	0.9065

- LLM is inherently suitable with few-shot scenario, without requirement of feature engineering.
- RAG significantly alleviate LLM hallucination, output structural prediction.
- Two-stage ranking strategy compensate the shortage of local LLM ability in understanding long-context multi-table data
- LLM methods have significant higher data efficiency and learning efficiency, it requires fewer label data to achieve higher results.

Experiment: Ablation Study

Table 3: Ablation study of different backbone LLM model for task CTA (resp. RE) on Semtab2019/WebTables (resp. WikiGS-RE) with 25% (resp. 10%) training data.

Model	Semtab2019		WebTables		WikiGS-RE	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
TableLLaMA(7B)	0.822	0.559	0.946	0.805	0.658	0.423
RAFL (Mistral-7B)	**0.862**	0.675	0.961	0.791	0.832	0.621
RAFL (Vicuna-13B)	0.861	**0.743**	**0.963**	**0.825**	**0.893**	**0.836**

Table 5: An Ablation Study on RE task

Model	Micro F1	Macro F1
RAFL w/o ret	0.3272	0.2469
RAFL w/o LLM	0.7427	0.5503
RAFL with LangChain	0.7842	0.5846
RAFL	**0.8930**	**0.8365**

- For Table 3, due to scaling law, a larger model can not only understands the context of downstream tasks but also performs more equitable classifications across minority relations and types.
- For Table 5, we have the following observations:
  - RAFL with LangChain: LangChain can only retrieve related corpus with semantic similarity, neglecting structural similarity
  - RAFL w/o LLM: pre-ranking model may have high top-k precision, but cannot achieve high top-1 precision. A more complex re-ranking model is essential
  - RAFL w/o ret: LLM suffers from hallucination issue

Conclusion

- We aim to solve the tabular interpretation problem with a unified retrieval-augmented framework with LLM. The novelty of our work consist of:
  - Propose a scheme, by unifying GSL, PLM and LLM in the same process
  - Retrieval-Augmented module to search relevant and similar table sets by semantic similarity, leveraging metadata.
  - Graph-Enhanced module to measure structural similarity among schema-free web tables.
  - Learn-to-Rank for LLM: alleviating LLM hallucination in ranking problem
- Our experiment study show LLM-based tabular interpretation is promising in practice, and have high data efficiency and learning efficiency