HeLM: Highlighted Evidence augmented Language Model for Enhanced Table-to-Text Generation

Anonymous ACL submission

Abstract

To harness the powerful text generation capabilities of recent large language models (LLMs) in the Table-to-Text task, employing parameter-efficient fine-tuning on open-source LLMs is a viable approach. However, how to enhance the model's table reasoning ability during LLM fine-tuning presents a challenge. In this study, we propose a two-step solution called HeLM. Different from previous fine-tuning-based methods that directly expand tables as inputs, our approach injects reasoning information into the input table by emphasizing table-specific row data. Our model consists of two modules: a table reasoner that identifies relevant row evidence, and a table summarizer that generates sentences based on the highlighted table. To facilitate this, we propose a method to train and construct reasoning labels for obtaining the table reasoner. On both the FetaQA and QTSumm datasets, our approach achieved state-of-the-art results in ROUGE and BLEU scores. Additionally, it is observed that highlighting input tables significantly enhances the model's performance and provides valuable interpretability.

1 Introduction

Tabular data is important and ubiquitous, serving as the fundamental format for data storage in databases. Analyzing and processing tabular data is important in the field of Natural Language Processing (NLP), such as table-based fact verification (Chen et al., 2019; Aly et al., 2021) and table-based question answer (Pasupat and Liang, 2015; Nan et al., 2022). The recent emergence of Large Language Models (LLMs) (Brown et al., 2020; OpenAI, 2023; Wei et al., 2022; Touvron et al., 2023a; Chung et al., 2022; Workshop et al., 2022) showcase impressive capabilities, unveiling vast potential in handling tabular data. Therefore, this paper delves specifically into the application of LLMs in table-to-text generation.

Current approaches (Ye et al., 2023; Chen, 2023) in utilizing LLMs for table-based tasks usually rely on invoking APIs for few-shot learning or integrating methods like chain-of-thought (Wei et al., 2022) or in-context learning. Although LLMs can achieve comparable performance even without fine-tuning, frequent API calls can be costly and pose information security risks. Therefore, a lightweight LLM system specialized in handling tabular data independently stands as an effective solution. With the availability of open-sourced LLMs like LLaMA (Touvron et al., 2023a,b), ChatGLM (Du et al., 2022), and the introduction of parameter-efficient training methods such as LoRA (Hu et al., 2021), fine-tuning a large model with limited computational resources is now available. In this study, we employ QLoRA (Dettmers et al., 2023) to fine-tune the LLaMA2 (Touvron et al., 2023b) base model specifically for table-to-text generation.

To enable the model to adeptly handle tabular data, it requires the capability to reason intricately across textual, numerical, and logical domains. Some methods (Abdelaziz et al., 2017; Hui et al., 2022) achieve this by synthesizing executable languages, such as SQL. Others (Herzig et al., 2020; Liu et al., 2021; Jiang et al., 2022; Gu et al., 2022) pretrain on additional table data to acquire table reasoning capabilities. However, some lightweight LLMs often lack table reasoning capabilities due to their pretraining text containing minimal tabular data content. In this paper, we conceptualize table reasoning as the capacity to identify crucial evidence within a table according to the output requirements. In this context, we define evidence as the specific row-level data crucial for answering the final output. Considering that input tables are often extensive, essential information usually resides within a small portion. Identifying and conveying this row data effectively to the model can significantly enhance the model's output quality.

In real-world scenarios, inputs often consist solely of tables and queries, necessitating an automated process to gather evidence data. This paper introduces a two-step methodology designed to tackle these challenges. The first is a LLM-based table reasoner, aimed at identifying and **H**ightlighting **e**vidence given input table. Then another Large **L**anguage **M**odel based table summarizer model generates the final output. This methodology is termed as **HeLM**.

The pivotal component of HeLM lies in the table reasoner, which outputs row indexes based on the given table and query. HeLM utilizes fine-tuned LLMs for this task. However, most datasets lack evidence labels for fine-tuning the reasoner. To address this issue, one direct approach is to distill evidence labels from more powerful LLMs. Additionally, we designed an algorithm that, without relying on other models or data, automatically constructs evidence labels using only the original input-output data from the table2text dataset. After that, combining evidence labels obtained from different methods can further enhance the quality of evidence. The table reasoner trained in this manner not only improves the overall performance of HeLM but also provides valuable interpretability.

The contributions of this paper are as follows: (1) We propose a two-step table-to-text approach named HeLM, which utilizes a table reasoner to highlight input tables, aiding downstream table summarizers in producing better text outputs. (2) We introduce a search-based evidence label construction method and a workflow for training HeLM's reasoner and summarizer. (3) HeLM attains state-of-the-art results in terms of BLEU and ROUGE scores on both the FetaQA and QTSumm datasets.

2 Related work

2.1 Table to Text Generation

Some table-to-text tasks (Chen et al., 2020; Parikh et al., 2020; Cheng et al., 2022a) focus on generating descriptions that correspond to the content within a selected range of tables. While tasks that limit the table scope for text output are relatively simple, they are not consistent with real-world applications. In contrast, other tasks (Suadaa et al., 2021; Moosavi et al., 2021) emphasize the analysis of tables within specific domains. Furthermore, tasks such as QTSumm (Zhao et al., 2023) and FeTaQA (Nan et al., 2022) involve text generation for tables based on provided queries.

Generally, these tasks are accomplished using neural encoder-decoder models that directly generate sentences through the fine-tuning of language models such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

2.2 Reasoning Over Tables

Enhancing a model's table reasoning capabilities is pivotal for table-related tasks. One prevalent strategy is pre-training models with reasoning data that includes both tables and text (Yin et al., 2020; Chen et al., 2020; Liu et al., 2021; Deng et al., 2022; Xie et al., 2022). However, these models often generate texts in an end-to-end manner, sacrificing explainability. An alternative approach, named by REFACTOR (Zhao et al., 2023), suggests generating query-relevant facts from tables as intermediate results for LLM's input. Another noteworthy method, as proposed in (Cheng et al., 2022b), employs Codex (Chen et al., 2021) to synthesize SQL for executing logical forms against tables in question-answering tasks. Dater (Ye et al., 2023) takes an approach by reducing the original table into relevant sub-tables. Unlike Dater, we choose to preserve the entire table, as reducing it to sub-tables may lead to the loss of critical global information. ToTTo (Parikh et al., 2020) also highlights the human observed evidence with the entire table retained, but its performance was found to be extremely poor.

3 Methodology

3.1 Table-to-Text Formulation

Table-to-text is a generative task $\mathcal{X} \rightarrow \mathcal{Y}$, where the input $\mathcal{X}$ comprises the table T and its metadata, typically involving a query Q to direct the output content. The output $\mathcal{Y}$ can be either a sentence or a paragraph.

3.2 HeLM Framework

Our framework consists of two components: a table reasoner $\mathcal{M}_R$ and a table summarizer $\mathcal{M}_S$. The table reasoner identifies the indexes of row data relevant to the query within the table. This component can be implemented using various model types. In this study, we employ a generative form of LLM. To achieve this, we design a prompt (see appendix) that concatenates rows of the table into a string along with the query and task description, forming the input for the table reasoner:

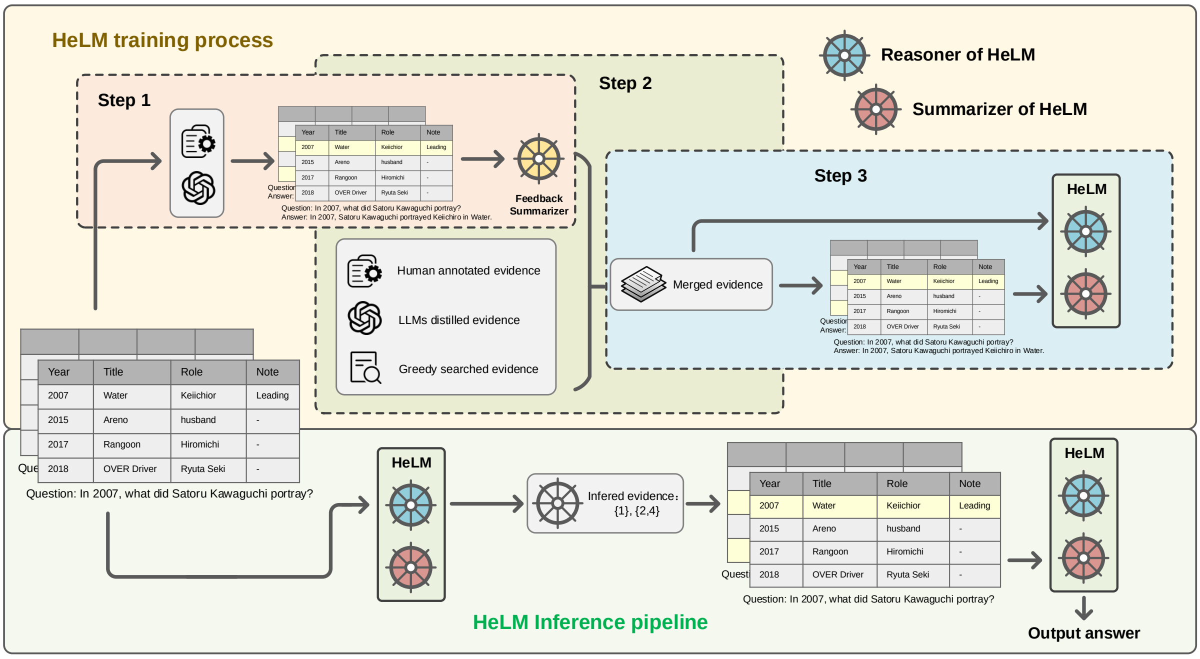

Figure 1: The overall framework of HeLM. The upper part demonstrates the training process, while the lower part illustrates the inference process.

$$E = \mathcal{M}_R\left(\text{Prompt}(T, Q)\right) \tag{1}$$

The output of $\mathcal{M}_R$ is a list of indices $E = \{e_i, \ldots\}$, where e_i corresponds to the row number in the table.

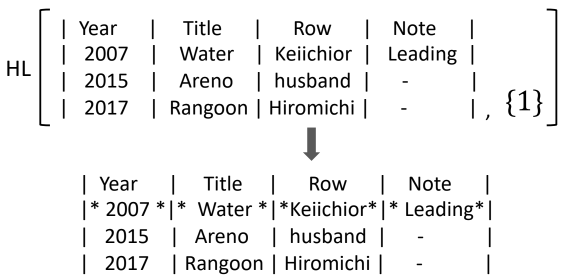

Figure 2: Case of table highlighting, where $\{1\}$ corresponds to E in equation 2.

Utilizing the information of key evidence (E) in the original table (T), we highlight this information to obtain the modified table (T^*). The highlighting operation HL($\cdot$) is to decorate each data cell of a key row with a special character '*' symbolizing its significance, as depicted in Figure 2.

$$T^* = \text{HL}(T, E) \tag{2}$$

The table summarizer will subsequently produce the final result based on the prompt generated by highlighted table T^* associated with query Q and task description:

$$\mathcal{Y} = \mathcal{M}_S\left(\text{Prompt}(T^*, Q)\right) \tag{3}$$

3.3 Table Reasoner Labels

Training evidence is necessary for fine-tuning a table reasoning module, and we summarize three sources for obtaining these evidence labels.

- **Human annotated evidence** E_{man}: Some datasets, such as QTSumm (Zhao et al., 2023), inherently include labels for relevant evidence, and they are obtained through manual annotation.

- **LLMs distilled evidence** E_{gpt}: Labels can also be distilled from other LLMs such as ChatGPT (OpenAI, 2023). To better capture evidence, we designed an in-context learning prompt (see appendix A), incorporating golden labels $\mathcal{Y}$ to better capture evidence.

$$E_{gpt} = \text{LLMs}(\text{Prompt}(T, Q, \mathcal{Y})) \tag{4}$$

- **Searched evidence** E_{se}: Evidence labels can also be obtained through search algorithms, which require feedback for different E. This feedback system has two requirements: one is the golden output $\mathcal{Y}$ corresponding to the input table and query, and the other is a feedback table summarizer $\mathcal{M}_F$. For more details of this algorithm, please refer to section 3.4.

$$E_{se} = \text{Search}(T, Q, \mathcal{Y}, \mathcal{M}_F) \tag{5}$$

The table evidence labels obtained through various methods showcases significant disparities. By integrating these evidence labels, higher-quality evidence can be attained. This process entails using highlighted tables associated with different evidence and getting sentences via the feedback summarizer $\mathcal{M}_F$. The evidence label for the current sample is chosen based on the sentence that receives the highest evaluated score. The formula for generating the merged label E_{merge} is outlined as follows:

$$E_{merge} = \text{Merge}(\boldsymbol{E}, T, Q, \mathcal{Y}, \mathcal{M}_F) \tag{6}$$

Here, $\mathbf{E}$ represents the available evidence label set. For datasets lacking human annotated evidence, $\boldsymbol{E} = \{E_{se}, E_{gpt}\}$.

3.4 Reasoning labels by Searching

As mentioned earlier, the search algorithm requires a feedback system. The feedback system includes the golden output $\mathcal{Y}$ corresponding to the table query and a feedback summarizer $\mathcal{M}_F$. $\mathcal{M}_F$'s output is evaluated by computing the BLEU score against $\mathcal{Y}$ to derive numerical feedback.

In label searching, the input for $\mathcal{M}_F$ is the sub-table corresponding to the evidence. Parikh et al. (2020); Ye et al. (2023) have proved that using only the sub-table corresponding to evidence as input yields satisfactory results when using LLMs. Therefore, we can use the sub-table corresponding to E as input, allowing the summarizer to obtain results for comparison with $\mathcal{Y}$ for feedback.

Another reason for using the sub-table as input to search for evidence is that $\mathcal{M}_F$ is more sensitive to sub-table evidence compared to the input of the complete table. Because even when relevant row data is not highlighted as evidence in the complete table input, $\mathcal{M}_F$ might still capture it.

Assuming the table has n rows of data, and each row can be either selected or not, the search space for this algorithm is 2^n. This implies that for each training example, one would need to invoke LLMs (summarizer) 2^n times to construct the optimal evidence, which is impractical. Therefore, we propose a greedy search method to construct labels, reducing the searching complexity from 2^n to n.

Algorithm 1: Reasoning evidence labels by greedy search

Input: Table T(n rows), Query Q, Answer $\mathcal{Y}$,
  Feedback summarizer M_F
Output: Searched evidence Label E_{se}
Generate n evidence labels $\boldsymbol{E} = \{E_1, E_2, ..., E_n\}$,
  where $E_i = \{i\}$
for $i \leftarrow 1$ *to* n **do**
  $\mathcal{Y}_i = M_s(\text{Prompt}(\text{SubTab}(T, E_i), Q))$
  $R_i = \text{eval}(\mathcal{Y}_i, \mathcal{Y})$
end
Reorder the E according to reward R.
Evidence label E_s is initialized with empty set.
Evidence label reward: $R_s = 0$
for $i \leftarrow 1$ *to* n **do**
  $\mathcal{Y}_i = M_F(\text{Prompt}(\text{SubTab}(T, E_i + E_s), Q))$
  $R_i = \text{eval}(\mathcal{Y}_i, \mathcal{Y})$
  if $R_i > R_s$ **then**
    $R_s = R_i$
    $E_s = E_i + E_s$
  end
end

The core idea of this algorithm is that query-relevant evidence can enhance the summarizer feedback score, while irrelevant evidence cannot enhance the feedback score. During evidence construction, the initial evidence is an empty set. Based on feedback results, we expand this evidence by adding row index one by one, and we repeat this process until the score no longer increases or reach a certain step. We have also designed heuristic steps to efficiently select evidence rows. A more detailed procedure is shown in Algorithm 1.

3.5 HeLM Training

HeLM comprises two modules: Reasoner and Summarizer. Training a complete HeLM modules involves the following steps:

Step 1 Obtain feedback summarizer: Distilling E_{gpt} through LLMs, and training a rough table summarizer $\mathcal{M}_S^0$ using either E_{gpt} or E_{man}.

$$\big(\text{HL}(E_{gpt/man}, T), Q, \mathcal{Y}\big) \rightarrow \mathcal{M}_S^0 \tag{7}$$

Step 2 Obtain merged evidence: Treating the table summarizer $\mathcal{M}_S^0$ as $\mathcal{M}_F$ to obtain E_{se} using Algorithm 1, then combining the existing evidence through Equation 6 to obtain E_{merge}.

Step 3 Train reasoner and summarizer: Train reasoner $\mathcal{M}_S$ using E_{merge}, and train summarizer $\mathcal{M}_R$ using $\mathcal{Y}$ and T^* corresponding to E_{merge}.

$$(\text{HL}(E_{merge}, T), Q, \mathcal{Y}) \rightarrow \mathcal{M}_S \tag{8}$$

$$(T, Q, E_{merge}) \rightarrow \mathcal{M}_R \tag{9}$$

Figure 1 displays the comprehensive training and inference process of HeLM.

Facing the immense size of recent language models, conducting full-parameters fine-tuning is prohibitively expensive. As a practical alternative, we adopt the parameter-efficient finetuning strategy, QLoRA (Dettmers et al., 2023; Hu et al., 2021), to train our reasoner and summarizer. This approach significantly reduces trainable parameters to 0.6% of the original, enabling fine-tuning of LLMs on consumer devices.

4 Experiments

4.1 Dataset and Evaluation

FeTaQA: FeTaQA is a dataset designed for free-form table question-answering, constructed using information from Wikipedia. It introduces a table question answering scenario, where questions are answered in natural language. The FeTaQA dataset comprises 7,326 question-answer pairs in the training set, 1,000 in the validation set, and 2,006 in the test set. For the evaluation of results on the FeTaQA dataset, we employ commonly adopted metrics, including ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004), and the BLEU (Papineni et al., 2002; Post, 2018) score.

QTSumm: QTSumm is a query-focused table summarization dataset, requiring text generation models to engage in human-like reasoning and analysis over the provided table to generate a tailored summary. The training and validation sets consist of 4,981 and 1,052 examples respectively, and the test set comprises 1,078 examples. Notably, in comparison to the FeTaQA dataset, QTSumm exhibits longer output lengths. For the evaluation of results on QTSumm, we employ not only ROUGE-L and BLEU scores but also the METEOR (Banerjee and Lavie, 2005) as the evaluation metric.

4.2 Implementation Details

All models are executed on a single NVIDIA-A100 GPU with 80G of memory. We optimized our baseline LLMs through 4-bit QLoRA finetuning, utilizing an effective batch size of 8 for 2 epochs. The optimization process employed the AdamW (Loshchilov and Hutter, 2018) optimizer with default momentum parameters and a constant learning rate schedule set at 2e-4. For QLoRA, NormalFloat4 with double quantization was applied to the base models, and LoRA adapters were added to all linear layers with parameters $r = 16$ and $\alpha = 32$. The maximum input length was constrained to 2048. For generating outputs from the LLMs, we employed nucleus sampling (Holtzman et al., 2019) with parameters $p = 0.9$ and a temperature of 0.1.

Our model, $\text{HeLM}_{\text{LLaMA2-13B}}$, denotes that both the summarizer and reasoner utilize LLaMA2-13B as the backbone model for parameter-efficient fine-tuning. The prompt used for fine-tuning is detailed in Appendix A.

Models	R-1	R-2	R-L	BLEU
Fine-tuning based methods				
T5-small	55	33	47	21.60
T5-base	61	39	51	28.14
T5-large	63	41	53	30.54
UnifiedSKG	64	42	54	31.5
TAPEX	62	40	51	30.2
OmniTab	63	41	52	30.7
PLOG	64	43	55	31.8
$\text{HeLM}^{\dagger}_{\text{LLaMA2-7B}}$	65.4	43.5	55.4	<u>32.95</u>
$\text{HeLM}^{\dagger}_{\text{LLaMA2-13B}}$	**67.8**	**46.4**	**57.9**	**35.10**
Few-shot LLMs methods				
TabCot(GPT-3)	61	38	49	27.02
Dater(Codex)	<u>66</u>	<u>45</u>	<u>56</u>	30.92

Table 1: Results on FeTaQA dataset. The † marked models are trained using QLoRA.

Models	BLEU	R-L	METEOR
Fine-tuning based methods			
T5-Large	20.3	38.7	40.2
BART-large	21.2	40.6	43.0
OmniTab	22.4	42.4	44.7
TAPEX	23.1	42.1	45.6
LLaMA2-13B†	23.3	42.8	46.7
$\text{HeLM}^{\dagger}_{\text{LLaMA2-13B}}$	<u>23.9</u>	<u>43.7</u>	48.1
- E_{man}	**25.0**	**45.3**	<u>50.0</u>
Few-shot LLMs methods			
LLaMA2-7B	14.0	31.2	37.3
LLaMA2-13B	17.5	33.2	42.3
LLaMA2-70B	19.0	38.0	46.4
GPT-3.5	20.0	39.9	<u>50.0</u>
GPT-4	19.5	40.5	**51.1**

Table 2: Results on QTSumm dataset. The † marked models are trained using QLoRA.

4.3 Baselines

There are primarily two types of baselines for Table-to-Text task, that is, fine-tuning methods, and few-shot methods using LLMs. In the FeTaQA dataset, fine-tuning baselines contain the T5-based (Raffel et al., 2020) models (T5-Small, T5-Base, and T5-Large), as well as TAPEX (Liu et al., 2021), OmniTab (Jiang et al., 2022), and PLOG (Liu et al., 2022). TAPEX and OmniTab are both BART-based models, with additional pre-training on custom training data.

In FeTaQA dataset, methods using LLMs for few-shot learning include Dater (Codex) (Ye et al., 2023) and TabCOT (Chen, 2023). The few-shot LLMs baselines for the QTSumm dataset are directly adapted from (Zhao et al., 2023), including methods such as LLaMA2 and GPT-4.

4.4 Main Results

HeLM$_{\text{LLaMA2-13B}}$ demonstrate superior performance on both the QTSumm and FeTaQA datasets. Specifically, on the FeTaQA dataset, HeLM$_{\text{LLaMA2-13B}}$ outperforms the previous leading method, Dater, with a 1.8 and 1.9 improvement in Rouge-1 and Rouge-L respectively. More notably, there is a substantial improvement in the BLEU score, with an increase of 3.26. Additionally, BLEU scores of fine-tuning methods are consistently higher than few-shot-based LLMs.

Models	Fluency	Correct	Adequate
TabCot(GPT3)	2.05	1.98	2.02
LLaMA2-QLoRA	2.00	2.11	2.06
HeLM	**1.96**	**1.92**	**1.91**

Table 3: Human evaluation on FeTaQA. The numbers in the table indicate the average ranking.

On the QTSumm dataset, the method fine-tuned based on LLaMA2-13B demonstrates significant improvement compared to other fine-tuning methods. Due to QTSumm providing manual evidence E_{man}, we can use this evidence to directly highlight the table as input for the HeLM's summarizer in evaluation. The corresponding result is denoted as HeLM-E_{man}. In comparison with other LLMs, HeLM-E_{man} and HeLM achieve the highest and second-highest scores in ROUGE-L and BLEU. However, GPT-4 achieves the highest score in the METEOR.

Relying solely on the ROUGE and BLEU scores cannot comprehensively assess the model's performance. Therefore, human evaluation is necessary. We conducted a human evaluation in three aspects: (1) fluency (whether the output sentences are fluent and without grammar errors), (2) correctness (the accuracy of numerical values and logical correctness of sentences), and (3) adequacy (whether the output results cover all aspects of the questions). Models we compared included LLaMA2-13B LoRA, which is also based on efficient fine-tuning, as well as Tabcot (GPT3), an LLMs-based few-shot method. We randomly sampled 100 examples from the test set and recruited three annotators to rank the three models.

Among these three metrics, correctness is the most indicative table reasoning ability. TabCot performs lower on fluency compared to LLaMA2 LoRA, but its correctness is significantly better. This suggests that fine-tuning on a specific dataset is more focused on learning surface-level features. Regarding the table reasoning ability as indicated by correctness, LLMs like GPT-3 showcases superior capabilities. HeLM performs best in correctness, indicating the positive impact of HeLM's reasoner on the overall accuracy of the results.

Models	R-1	R-2	R-L	BLEU
Different highlight evidence				
LLaMA2-13B†	66.5	44.7	56.2	33.24
HeLM$^\dagger_{\text{LLaMA2-13B}}$	67.8	46.4	57.9	35.10
- w/o HL	66.6	44.7	56.6	33.13
- subTab	65.0	43.3	55.5	32.28
- $E_{gpt\ w/o\ \mathcal{Y}}$	68.1	46.6	58.1	35.34
- E_{gpt}	69.4	47.8	59.2	36.33
- E_{se}	68.1	46.6	58.0	34.96
- E_{merge}	69.6	48.2	59.5	36.74
Different model size				
LLaMA2-7B†	65.0	43.0	54.8	32.68
HeLM$^\dagger_{\text{LLaMA2-7B}}$	65.4	43.5	55.4	32.95
- w/o HL	64.7	43.0	54.8	32.28

Table 4: Ablation study on FeTaQA dataset. The † marked models are trained using QLoRA.

4.5 Ablation Study

4.5.1 Impact of model size

As shown in Table 4, when using LLaMA2-7B as the base model for fine-tuning, HeLM$_{\text{LLaMA2-7B}}$ showed a 2.15 decrease in BLEU score and a 2.5 decrease in ROUGE-L score compared to

(1) Table caption: 2014 Newark by-election

party	Candidate	Votes	%	±
Conservative	Robert Jenrick	17,431	45.0	8.9
UKIP	Roger Helmer	10,028	25.9	22.1
Labour	Michael Payne	6,842	17.7	4.7
Independent	Paul Baggaley	1,891	4.9	N/A
Green	David Kirwan	1,057	2.7	N/A
...	...	...	...	...
Majority	Majority	7,403	19.1	12.4

Query: Who won the 2014 Newark by-election and by what margin?

Golden: Conservative Robert Jenrick won with a majority of 7,403.

HeLM: The 2014 Newark by-election was won by Robert Jenrick of the Conservative Party with a majority of 7,403 votes.

LLaMA2-QLoRA: The Conservative Party's Robert Jenrick won the 2014 Newark by-election with 45.0% of the vote, a majority of 19.1%.

TabCot (GPT3): The Conservative party held the seat in the 2014 Newark by-election. The margin of victory was 19.1%.

(2) Table caption: Paul Williams (footballer, born 1971)

Club	Season	League		Total	Total
Club	Season	Division	...	Apps	Goals
Dearby County	1989-90	First Division	...	10	1
Dearby County	1990-91	First Division	...	19	4
...	...	...	...	...	...
Coventry City	1995-96	Premier League	...	37	2
Coventry City	1995-96	Premier League	...	38	2
Coventry City	1995-96	Premier League	...	25	0
...	...	...	...		
Coventry City	2001-02	Premier League	...	5	0
Coventry City	Total	Total	...	199	5
Southampton	2001-02	Premier League	...	30	0
...	...	...	...		

Query: How many appearances did Williams make for Coventry City, and for how many seasons did he play for the club?

Golden: Williams made 199 appearances for Coventry City in 7 consecutive seasons.

HeLM: Williams made 199 appearances for Coventry City in seven seasons.

LLaMA2-QLoRA: Williams made 58 appearances for Coventry City in two seasons.

TabCot (GPT3): Williams made 169 appearances for Coventry City over six seasons.

(3) Table caption: Cliff Gorman

Year	Title	Role	Note
1969	Justine	Toto	-
1970	The Boys in the band	Emory	
...	...	...	...
1980	Night of the Juggler	Gus Soltic	-
1981	The Bunker	Joseph Gobbels	TV movie
1984	Angel	Lt. Andrews	-
1992	Night and the City	Phil Nasseros	-
1992	Hoffa	Solly Stein	-
...	...	...	...

Query: What roles did Cliff Gorman play between 1981 to 1984?

Golden: Cliff Gorman played as Joseph Goebbels in the 1981 TV movie The Bunker and as Lt. Andrews in the film Angel (1984).

HeLM: Cliff Gorman played the role of Joseph Goebbels in the TV movie The Bunker (1981), Lieutenant Andrews in Angel (1984), and Solly Stein in Hoffa (1992).

LLaMA2-QLoRA: Cliff Gorman played the role of Joseph Goebbels in the TV movie The Bunker (1981) and the role of Lt. Andrews in Angel (1984).

TabCot (GPT3): Cliff Gorman played the roles of Lt. Andrews in Angel (1981), Joseph Goebbels in The Bunker (1981), and Solly Stein in Hoffa (1992).

(4) Table caption: Swimming at the 2012 Summer Olympics – Women's 200 metre backstroke

Rank	Lane	Name	Nationality	Time	Notes
-	5	Missy Franklin	United States	2:04.06	WR
-	6	Anastasia Zuyeva	Russia	2:05.92	-
-	4	Elizabeth Beisel	United States	2:06.55	-
4	1	Elizabeth Simmonds	Great Britain	2:07.26	-
5	3	Meagen Nay	Australia	2:07.43	-
6	7	Kirsty Coventry	Zimbabwe	2:08.18	-
7	2	Alexianne Castel	France	2:08.43	-
8	8	Sinead Russell	Canada	2:09.86	-

Query: Who were the last two finishers of the at the 2012 Summer Olympics for the Women's 200 metre backstroke?

Golden: France's Alexianne Castel (2:08.43) and Canada's Sinead Russell (2:09.86) closed out the championship field.

HeLM: Canada's Sinead Russell finished eighth in 2:09.86 and Australia's Belinda Hutchison rounded out the finale in ninth with a time of 2:10.00.

LLaMA2-QLoRA: The last two finishers of the Women's 200 metre backstroke were Kirsty Coventry with a time of 2:08.18 and Alexianne Castel with a time of 2:08.43.

TabCot (GPT3): The last two finishers were Sinead Russell in 8th place with a time of 2:09.86 and Kirsty Coventry in 6th place with a time of 2:08.18.

Figure 3: Cases from the FeTaQA Dataset. The reasoner of HeLM has highlighted specific parts of the table using red boxes. The rows in the table with a green background represent manually observed evidence related to the query.

$\text{HeLM}_{\text{LLaMA2-13B}}$. Directly fine-tuning LLaMA2-7B using LoRA also exhibited a 1.4 decrease in ROUGE-L compared to LLaMA2-13B. This indicates that LLM's size significantly affects the results for the Table-to-Text task.

4.5.2 Impact of table highlighting

HeLM's summarizer takes tables highlighted with evidence as input, and different evidence will have different effects on the output results of the summarizer. When the summarizer of $\text{HeLM}_{\text{LLaMA2-13B}}$ receives unmodified tables as input, specifically, the result of **-w/o HL** showed a decrease of both BLEU and ROUGE-L. This signifies the effectiveness of highlighting crucial information in LLM's input tables. Additionally, when using the same evidence for test data, and constructing a sub-table with only key row information as input instead of retaining all table data, the approach **-subTab** has a 2.82 decrease in BLEU score. This suggests the benefit of retaining sufficient table information. Another observation is that when no highlighting is applied to the input table, LLaMA2 outperformed HeLM-w/o HL. This happens because HeLM's summarizer generated dependency on highlighted evidence during training. However, during testing, when the highlighting is absent, it results in poorer performance compared to LLaMA2.

4.5.3 Impact of evidence labels

Keeping the summarizer of $\text{HeLM}_{\text{LLaMA2-13B}}$ fixed, we examine the output derived from employing various evidence labels for table highlighting, aiming to illustrate the impact of evidence label quality.

During testing, the evidence used for highlighting the input table in our base model $\text{HeLM}_{\text{LLaMA2}}$ is generated by the $\text{HeLM}_{\text{LLaMA2}}$'s reasoner. E_{gpt} and E_{se} are reasoning evidence mentioned in section 3.3, while E_{merge} is a combination of the two evidence labels. It's important to note that all three labels were obtained with knowledge of the golden summary $\mathcal{Y}$. $E_{gpt\ w/o\ \mathcal{Y}}$, based on the method used for E_{gpt}, eliminates $\mathcal{Y}$ from the prompt. Thus, the BLEU score also decreased by 1.01. According to the Table 4, the evaluation score corresponding to E_{merge} is the highest, indicating that the evidence quality of E_{merge} is the best. This also indicates that although the overall quality of E_{se} obtained through greedy search is lower than E_{gpt}, some samples perform better than E_{gpt}.

4.6 Cases Analysis

We showcase some instances of accurate and inaccurate predictions generated by HeLM's table reasoner, alongside outputs from TabCot(GPT3) and LLaMA2-QLoRA respectively, as shown in Figure 3. For instance, case (2) shows the results given two questions about numerical calculation. HeLM's reasoner accurately finds the player's records during their tenure at Coventry, aiding the table summarizer in precisely calculating the player's tenure and total appearances. In contrast, both LLaMA2-QLoRA and TabCot(GPT3) give wrong answers for the two questions.

Cases (3) and (4) represent instances where the reasoner made inaccurate judgments. In case (3), the reasoner highlighted two irrelevant rows, one of which appeared in the summarizer's output. In case (4), the reasoner missed highlighting one row, leading the summarizer to fabricate a ninth-ranking entry, but the table only contained data for the top eight ranks. Therefore, it's evident that the summarizer places significant emphasis on the highlighted segments of the table as identified by the reasoner.

5 Conclusion

In this paper, leveraging existing open-source LLM, we devised a lightweight two-step table-to-text solution named HeLM. HeLM comprises two modules: table reasoner and table summarizer. Both modules adopt LLaMA2 as the backbone model and conduct efficient fine-tuning using designed prompts. Additionally, we explored diverse methods for constructing reasoning evidence, encompassing distillation from ChatGPT and construction by a searching algorithm. Our experimental findings showcase that leveraging the reasoner to highlight important row data of the input table significantly elevates the quality of the output and provides valuable interpretability.

Limitations and Future work

Despite HeLM achieving good results on two table-to-text datasets, there are still some limitations and space for further improvement: (1) We haven't extensively investigated table highlighting formats, and there might be more effective ways. (2) Currently, HeLM is trained for specific datasets, lacking generalization; training HeLM on a mixture of table-to-text datasets could be a better solution. (3) The evidence labels generated by greedy search in table reasoner could be further improved. For instance, we can employ reinforcement learning to search for more optimal evidence labels. (4) Despite our model achieving high scores in BLEU and ROUGE metrics, its advantages in numerical and textual accuracy aren't notably pronounced compared to some powerful LLMs.

Ethics Statement

We acknowledge the importance of the ACL Ethics Policy and agree with it. The objective of HeLM systems in this paper is to enhance data processing efficiency. The datasets, QTSumm and FeTaQA, utilized in this paper are both public datasets under the MIT license.

In human evaluation, we recruit 3 graduate students in computer science and statistic majors (2 male and 1 female) each student is paid \$11.2 (above average local payment of similar jobs) per hour.

References

Ibrahim Abdelaziz, Razen Harbi, Zuhair Khayyat, and Panos Kalnis. 2017. A survey and experimental comparison of distributed sparql engines for very large rdf data. *Proceedings of the VLDB Endowment*, 10(13):2049–2060.

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of* *the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wenhu Chen. 2023. Large language models are few (1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1090–1100.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022a. Hitab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2022b. Binding language models in symbolic languages. *arXiv preprint arXiv:2210.02875*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Zihui Gu, Ju Fan, Nan Tang, Preslav Nakov, Xiaoman Zhao, and Xiaoyong Du. 2022. Pasta: Table-operations aware fact verification via sentence-table cloze pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4971–4983.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, 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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Binyuan Hui, Ruiying Geng, Lihan Wang, Bowen Qin, Yanyang Li, Bowen Li, Jian Sun, and Yongbin Li. 2022. S2sql: Injecting syntax to question-schema interaction graph encoder for text-to-sql parsers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1254–1262.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022. Plog: Table-to-logic pretraining for logical table-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5531–5546.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Matt Post. 2018. A call for clarity in reporting bleu scores. *WMT 2018*, page 186.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1451–1465.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning. *arXiv preprint arXiv:2301.13808*.

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.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Xiangru Tang, Yumo Xu, Arman Cohan, et al. 2023. Qtsumm: A new benchmark for query-focused table summarization. *arXiv preprint arXiv:2305.14303*.

Prompt to get E_{gpt}

System:
You are a linguistic expert, and you need to determine which parts of the row data in table are relevant to the question and answer. I will first give you two examples of input and Sub-table Row output. Note that your output format can only be 'row(*)' !!!
User:
[example 1]
[example 2]
###Input:[table] \n ##Question:[query] \n ##Answer:[golden output] \n ###Sub-table Row:

Figure 4: The prompt used to obtain E_{gpt} using GPT-3.5-trubo. The content within the green brackets represents two examples used for few-shot learning. The inputs such as tables, queries, etc., within the red brackets, can be replaced according to specific input requirements.

A Prompt Design

Prompt to get E_{gpt}: To achieve better results, we incorporate two examples of input-output pairs within the prompt to help LLMs understand the output format. Additionally, we include the golden summary $\mathcal{Y}$ to guide a better identification of evidence. The details of the prompt template for obtaining E_{gpt} can be found in Figure 4. Additionally, the two samples used in the prompt can be found in Figure 6.

Prompt of Reasoner

System:
You are an expert table reasoner, your task is to output the relative row indexes which might be helpful for answering the query.
User:
###Table:[table] \n ###Query:[query] \n ###Output:[golden output]

Prompt of Summarizer

System:
You are an expert table reasoner, your task is to output the answer given Table and Query. Relative table units to query are surrounded by "*".
User:
###Table:[table] \n ###Query:[query] \n ###Output:[golden output]

Figure 5: The top section shows the prompt corresponding to the reasoner, and the bottom section shows the prompt corresponding to the summarizer. The elements within the red brackets can be replaced based on different examples.

Prompt of reasoner and summarizer In HeLM, both the table reasoner and summarizer utilize LLMs, necessitating the construction of input prompt text. Figure 5 displays the prompt templates used for the two components in HeLM. During inference, leave the area after "###Output" in the prompt blank.

B Code libraries

HeLM utilizes the *PyTorch* deep learning framework, loading models provided by *Huggingface*, and utilizes the *PEFT* package for parameter-efficient fine-tuning. The training framework of the model is modified based on the *LLM-finetuning HUB*.

Example 1

###Input: Table caption: University of Oregon Admissions
col: - | 2014 | 2013 | 2012 | 2011 | 2010
row 1: Applicants | 21,359 | 21,938 | 21,263 | 23,012 | 18,515
row 2: Admits | 15,997 | 16,206 | 15,770 | 16,790 | 14,588
row 3: % Admitted | 74.9 | 73.9 | 74.2 | 73.0 | 78.8
row 4: Avg GPA | 3.58 | 3.60 | 3.57 | 3.59 | 3.52
row 5: Enrolled | 3,961 | 3,966 | 4,031 | 4,167 | 3,978
row 6: SAT range* | 990–1230 | 990–1240 | 991–1224 | 993–1223 | 991–1218

##Question: How many students were accepted from the 21,359 people who applied for the University of Oregon in 2014 and how many students enrolled?
##Answer: For students entering University of Oregon 2014, 15,997 freshmen were accepted out of 21,359 applicants, a 74.9% acceptance rate, and 3,961 enrolled.
###Sub-table Row: row(1, 2, 3, 5)

Example 2

###Input:
Table caption: Annette Taddeo Early elections, 2008–2016
col: Party | Party | Candidate | Votes | %
row 1: - | Republican | Rick Scott/Carlos López-Cantera | 2,865,343 | 48.1%
row 2: - | Democratic | Charlie Crist/Annette Taddeo | 2,801,198 | 47.1%
row 3: - | Libertarian | Adrian Wyllie/Greg Roe | 223,356 | 3.8%
row 4: - | No Party Affiliation | Glenn Burkett/Jose Augusto Matos | 41,341 | 0.7%
row 5: - | No Party Affiliation | Farid Khavari/Lateresa A. Jones | 20,186 | 0.3%
row 6: Total votes | Total votes | Total votes | 5,951,561 | -

##Question: What duo finished second in the election, what duo won the election, what percentage of vote did each duo receive, and what party was victorious?
##Answer: The Crist-Taddeo lost the election to Republican Rick Scott and Carlos López-Cantera, 48.1 to 47.1%.
###Sub-table Row: row(1, 2)

Figure 6: Two examples used in figure 4