

Reasoning Knowledge Filter for Logical Table-to-Text Generation

Yu Bai^{1,4}, Baoqiang Liu¹, Shuang Xue², Fang Cai³, Na Ye^{1,4}, Guiping Zhang^{1,4}

¹School of Computer Science, Shenyang Aerospace University

²School of Foreign Language, Shenyang Aerospace University

³Department of Statistics, Stanford University

⁴National and Local Joint Engineering Laboratory for Multilingual Collaborative Translation Technology

Correspondence: {baiyu, liubaoqiang}@sau.edu.cn

Abstract

Logical table-to-text generation (LT2T) seeks to produce logically faithful textual descriptions base on tables. Current end-to-end LT2T models, which use descriptions directly as learning objectives, frequently face challenges in maintaining logical faithfulness due to the lack of a reasoning knowledge. Recent research have introduced reasoning knowledge generated by models for LT2T task, but the noise along with it limited its performance. We therefore propose a framework *reasoning knowledge filter* that leverages the collaboration between large language models and smaller models to filter data points with high-quality reasoning knowledge. This framework aims to provide highly matched table, description and reasoning knowledge triplets for LT2T. The results obtained on LogicNLG database demonstrate that the efficiencies of the method in this paper has achieved optimal performance with a reduced amount of data. Specifically, it enhances SP-Acc by 1.4 points and NLI-Acc by 0.7 points compared to the current state-of-the-art model.

1 Introduction

The objective of the table-to-text generation task is to generate a sentence or a paragraph based on a table. In recent years, it has been an important research direction in natural language processing (Lebret et al., 2016; Wiseman et al., 2017; Parikh et al., 2020; Cheng et al., 2021).

However, previous work is still limited to surface-level descriptions that simply use language to describe the basic facts about the tables (Parikh et al., 2020; Wiseman et al., 2017; Novikova et al., 2017; Lebret et al., 2016). As shown in Figure 1, a surface-level description of the table might be “The attendance for the game held at the Wachovia Center on December 2nd was 19,227”. Chen et al. (2020a) argue that new research should go beyond surface-level descriptions. To this end, they proposed a new task called logical table-to-text gen-

2008 - 09 tampa bay lightning season

date	opponent	location	...	attendance
december 2	...	wachovia center	...	19227
december 4	...	st pete times forum	...	15598
december 6	...	st pete times forum	...	17154
...
december 18	...	st pete times forum	...	16333
...
december 27	...	st pete times forum	...	18226

description : 4 game were played at the St Pete Time Forum.

Figure 1: Example for logicNLG dataset.

eration (LT2T) with a dataset named LogicNLG. In LT2T, the descriptions generated from tables are no longer surface-level, instead, they are required to be logically faithful to the tables. The facts included in such descriptions require multi-step reasoning based on the table. For example, the description in Figure 1 requires the following steps of reasoning: (1) Select all rows with the “location” column value as “st pete times forum”. (2) Count the resulting rows to get “4”.

Since LogicNLG only provides tables and descriptions, most research focuses on modifying model architectures or training objectives to ultimately achieve a model that can outperform end-to-end training (Chen et al., 2021; Nan et al., 2022; Zhao et al., 2023b). These models need to implicitly learn how to generate logically faithful descriptions by reasoning on tables.

A recent study RKT (Liu et al., 2024) suggests that incorporating explicit reasoning knowledge into LogicNLG can enhance the performance of existing generation models. Based on the idea of transfer learning, RKT generates the corresponding reasoning knowledge for each LogicNLG data by learning from out-of-domain datasets. For example, RKT generate a reasoning knowledge for the description in Figure 1 : “select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4”.

We have observed that this reasoning knowledge

vehicles & animals			
country	date	label	format
united kingdom	...	parlophone	lp
united kingdom	...	parlophone	cd
united kingdom	...	parlophone	cd digipak
united states	...	astralwerks	cd
australia	...	capitol records	cd

description : Australia and United State made their Label using the same Format.

reasoning knowledge : select the rows whose country record fuzzily matches to Australia. for the format records of these rows, all of them fuzzily match to Cd. there is only one such row in the table. the label record of this unique row is Capitol Records.

Figure 2: The incorrect reasoning knowledge generated by RKT.

is not manually annotated but generated by models, which introduces a significant amount of noisy data. As shown in Figure 2, the reasoning knowledge generated by RKT for description is incorrect, it may have a negative impact on model training process.

We hypothesize that if we can filter out some of the noisy data and train the model to use only the remaining clean data, we can further improve the performance of existing generation models. However, relying on human to label nearly 30,000 such pieces of data could take more time and effort. With powerful reasoning capabilities, Large Language Models (LLMs) has the potential to replace humans in the annotation process(Zhao et al., 2023a). But with large token size of serialized datas, LLMs may have higher economic costs.

To filter reasoning knowledge and minimize costs, we propose a framework *reasoning knowledge filter* (RKF) based on knowledge distillation. RKF first splits the LogicNLG dataset, which contains reasoning knowledge introduced by RKT(Liu et al., 2024), into two parts of different sizes. Then, GPT-4o is tasked with adding correctness labels to the reasoning knowledge in the smaller part. Subsequently, the data with correctness labels is used to train a reasoning knowledge correctness classification model. Finally, this classification model is employed to determine the correctness of the reasoning knowledge in the remaining larger part of the data. The results obtained on LogicNLG dataset demonstrate that our method further improves the performance of existing models, with an increase of 1.4 points in SP-Acc and 0.7 points in NLI-Acc over the current state-of-the-art model.

Overall, our contributions are as follows:

- We proposed a framework *reasoning knowledge filter* to filter out noisy data from the

existing reasoning knowledge in LogicNLG, successfully reducing the training bias in existing generation models.

- We provided a updated version of the LogicNLG dataset, which offers a higher degree of alignment between reasoning knowledge, tables, and descriptions compared to provided in RKT.
- We achieved optimal performance using less data than the original LogicNLG training set.

2 Related Work

With the recognition of pre-trained language models (PLMs) by researchers, most recent works have adopted the approach of end-to-end fine-tuning of PLMs to accomplish table-to-text generation. Kale and Rastogi (2020) obtained notable results across various datasets by solely fine-tuning T5; Wang et al. (2022a) changed the attention and position encoding base on T5; and An et al. (2022) proposed optimizing the loss function during the fine-tuning of PLMs using contrastive learning. For LT2T, R2D2 modified the input and loss function used during the fine-tuning of T5(Nan et al., 2022).

Notably, in the data-to-text generation field, some researchers have been dedicated to integrating PLMs with traditional natural language generation generation method content planning and surface realisation(Holmes-Higgin, 1994). This type of model, which uses PLMs for content planning and surface realization, is referred to as a pipeline model. The generation of descriptions using a pipeline model typically involves two steps: (1) Content Planning: selecting and ordering key information from the input data, and (2) Surface Realization: generating descriptions based on the key information. PlanGen(Su et al., 2021) is a representative of such pipeline models.

The reasoning capabilities of LLMs have garnered significant attention in recent years. Chain-of-Thought (CoT) is a crucial technique for further unlocking the reasoning capabilities of LLMs. Widely recognized research on CoT includes few-shot CoT(Wei et al., 2022), zero-shot CoT(Kojima et al., 2022), auto-CoT(Zhang et al., 2022) and Self-consistency(Wang et al., 2022b). Our work partially draws on few-shot CoT and self-consistency research.

Knowledge Distillation (KD)(Xu et al., 2024) in LLMs is a technique used to transfer knowledge

from a large, complex model to a smaller, simpler model. This process aims to retain the performance of the larger model while reducing computational requirements. KD methods are mainly used for white-box classification models or for training smaller models to replicate the behavior of black-box model APIs, such as ChatGPT. Many KD efforts achieve text classification tasks by imitating the output distribution of the teacher model(Song et al., 2020; Liang et al.; Zhang et al., 2023).

3 Method

This section begins with some preliminary information for our work (3.1), followed by *reasoning knowledge filter* framework (3.2).

3.1 Preliminaries

3.1.1 LogicNLG Datasets

The LogicNLG dataset is divided into three parts: train, dev, and test. We define these parts as follows:

$$DT = \{(T, S)_i\}_{i=1}^{|DT|} \quad (1)$$

$$DV = \{(T, S)_i\}_{i=1}^{|DV|} \quad (2)$$

$$DS = \{(T, S)_i\}_{i=1}^{|DS|} \quad (3)$$

in which T is a structured table, S is a description, and $|DT||DV||DS|$ is the size of dataset.

3.1.2 LogicNLG Datasets With Reasoning Knowledge

RKT introduces explicit reasoning knowledge to the LogicNLG train set, expanding the original (T, S) pairs into triplets. We can redefine DT as

$$DTRK = \{(T, S, RK)_i\}_{i=1}^{|DTRK|} \quad (4)$$

where RK is reasoning knowledge.

Reasoning knowledge is a formal process of reasoning out descriptions from tables(Liu et al., 2024). It based on seven categories of logical operation functions: count, superlative, comparative, aggregation, majority, unique, and ordinal (details of the logical operation functions are provided in the AppendixA). High-quality reasoning knowledge can be seen as a graph structure. As shown in Figure 3, the bottom-up execution order in graph structures provides a more intuitive explanation of the formal process of inferring descriptions from tables.

reasoning knowledge : select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4.

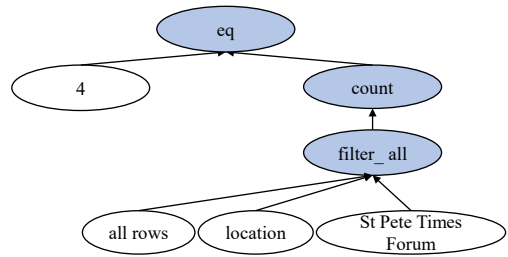


Figure 3: The correspondence between reasoning knowledge and graph structures for example in Figure 1.

3.1.3 Table Serialization

In table-to-text generation, tables are typically transformed into a sequence format before being fed into the model. Current research suggests various methods for table serialization. Early research tended to serialize tables using XML formats(An et al., 2022). With the advent of pre-trained language models, it has become common to use natural language templates for table serialization(Chen et al., 2020a). For LLMs, directly adding special tokens between cells and rows is also an effective serialization choice. Moreover, Chen et al. (2020a) highlighted that serializing the entire table can negatively impact model performance. Consequently, we adhere to Chen et al. (2020a) approach and choose specific sub-columns of the table for serialization.

3.2 Reasoning Knowledge Filter

To illustrate the dataset’s transformation, the process of *reasoning knowledge filter* is depicted in Figure 4. The numbers enclosed in brackets within the figure correspond to the equations provided in this paper. Now, we will separately introduce the implementation details.

3.2.1 Dataset Partition

As mentioned in Section 1, we had planned to use LLMs to replace human annotation because manually annotating the correctness of reasoning knowledge incurs significant time and financial costs. Therefore, we initially planned to use LLMs to completely replace human annotation. However, while LLMs can significantly reduce time costs, they do not lower economic costs. According to our estimates, with even the most token-efficient method to serialize the tables, the entire training set would still need over 30,000,000 prompt tokens and 45,000,000 completion tokens (for each

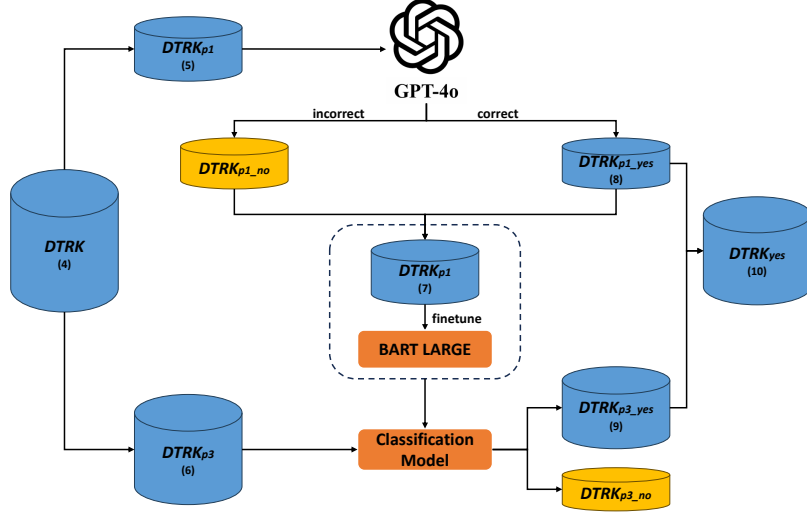


Figure 4: Reasoning knowledge filter process.

data point, we assume that the prompt requires a minimum of 1,024 tokens, and three completions require 1,536 tokens). To further reduce economic costs, we decided to partition $DTRK$ in a 1:3 ratio. The partitioned data is defined as follows:

$$DTRK_{p1} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p1}|} \quad (5)$$

$$DTRK_{p3} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p3}|} \quad (6)$$

3.2.2 Annotation of Reasoning Knowledge Correctness Based on LLMs

It is widely acknowledged that LLMs possess powerful multi-step reasoning capabilities. In scenarios where data is scarce, numerous studies have attempted to use LLMs to generate training data (Long et al., 2024). We believe that employing LLMs to assess the correctness of reasoning knowledge is feasible.

The recently released GPT-4o has demonstrated outstanding performance in multi-step reasoning capabilities, particularly excelling in handling complex logical reasoning tasks. It has shown significant advantages in accurately generating and verifying intricate reasoning chains. Therefore, we have chosen GPT-4o to assess the correctness of reasoning knowledge in $DTRK_{p1}$.

Figure 5 illustrates the entire process of using GPT-4o to assess the correctness of reasoning knowledge. Each prompt can be divided into two parts: the instruction description and the serialized data. The instruction description remains constant in each prompt. We prompt GPT-4o to perform step-by-step reasoning based on the reasoning knowledge to determine whether the facts

included in the description can be derived. To quickly pinpoint the judgment results of GPT-4o, we also require GPT-4o to return a definitive response, specifically either “*The result is yes*” or “*The result is no*”. In the completions generated by GPT-4o, there remains a small portion of data that does not include the two specified markers. We consider this subset of data as containing incorrect reasoning knowledge. In serialized data, we choose to add special tokens between cells and rows when table serialization. We distinguish cells and rows with “#” and newlines, respectively. Based on our tests, LLMs perform well with each serialization scheme mentioned in Section 3.1.3. However, the scheme involving the addition of special characters is the most token-efficient. Due to the possibility of errors in single-instance inferences generated by LLMs, we require GPT-4o to randomly generate three completions for each data point to maximize the collection of accurate reasoning knowledge. The conclusions derived from these three completions may differ. Inspired by research on the self-consistency of CoT, we select the conclusion that appears most frequently as the final result.

After GPT-4o completes the correctness assessment of reasoning knowledge for all $DTRK_{p1}$ data, the definition of $DTRK_{p1}$ will be updated as follows:

$$DTRK_{p1} = \{(T, S, RK, FLAG)_i\}_{i=1}^{|DTRK_{p1}|} \quad (7)$$

where the $FLAG$ is used to indicate the correctness of the reasoning knowledge. The candidate values for $FLAG$ are “*YES*” or “*NO*”. Further-

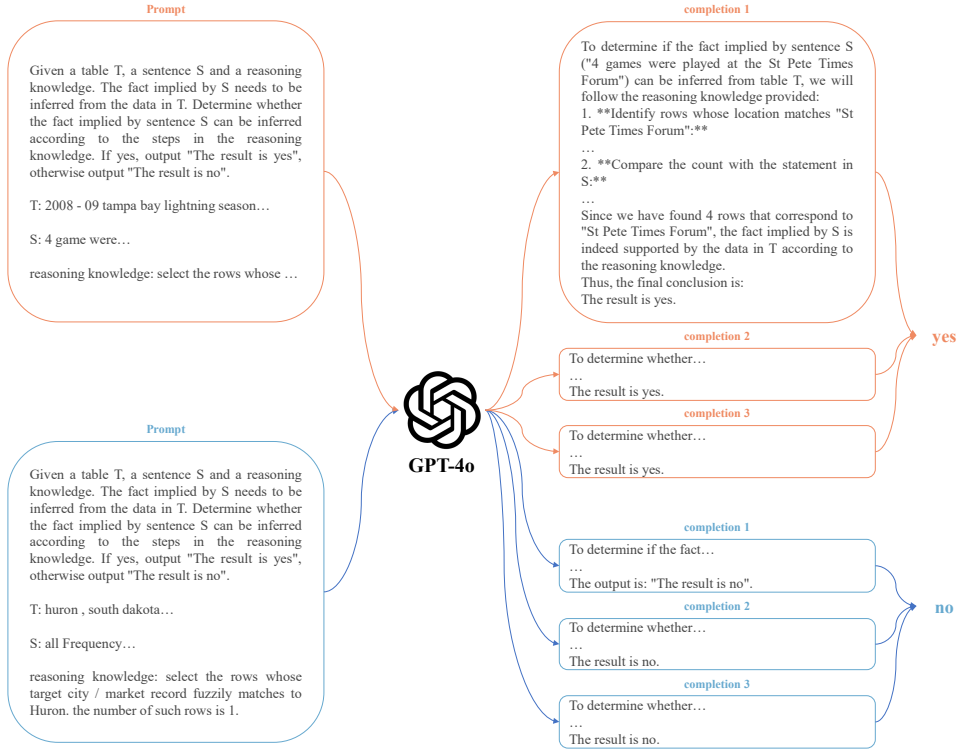


Figure 5: Annotation of reasoning knowledge correctness based on LLMs. Detailed examples will be presented in the Appendix A.

more, we define the subset of $DTRK_{p1}$ data where the $FLAG$ is set to “YES” as follows:

$$DTRK_{p1_yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p1_yes}|} \quad (8)$$

3.2.3 Reasoning Knowledge Correctness Classification Model

In previous section, we annotated $DTRK_{p1}$ using GPT-4o. Now, we need to complete the annotation for $DTRK_{p3}$. Knowledge distillation (Xu et al., 2024), as a technique where smaller models learn from the capabilities of LLMs, is widely utilized by researchers. Inspired by knowledge distillation, we decided to use a smaller text classification model to learn the prediction distribution of GPT-4o on $DTRK_{p1}$. Subsequently, we will employ this classification model to annotate the reasoning knowledge correctness for dataset $DTRK_{p3}$.

We selected the BART-large (Lewis, 2019) model as the foundation for our classification model. By jointly training the pre-trained weights of BART-large and the randomly initialized weights of the linear classification layer, we can assess the correctness of reasoning knowledge. We use $DTRK_{p1}$ to train the classification model, requiring the classification model to determine the correctness of

reasoning knowledge based on the input table, description, and reasoning knowledge. Figure 8 in appendix illustrates the inputs and outputs during the training of the classification model. Note that, following previous table-to-text research (Chen et al., 2020a, 2021; Nan et al., 2022; Liu et al., 2024), we use natural language templates to achieve table serialization during training.

After completing the training, we use the classification model to assess the correctness of all reasoning knowledge in $DTRK_{p3}$. Using the same input format as during training, we provide the classification model with a table, a description, and a reasoning knowledge, and require it to determine the correctness of the reasoning knowledge. We merge the data points that the classification model deems as correct reasoning knowledge into a set, which can be defined as:

$$DTRK_{p3_yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{p3_yes}|} \quad (9)$$

3.2.4 Merge Data

Through Sections 3.2.2 and 3.2.3, we successfully filtered all data split according to a 1:3 ratio in Section 3.2.1. The data marked as “YES” for the correctness of reasoning knowledge are the ones

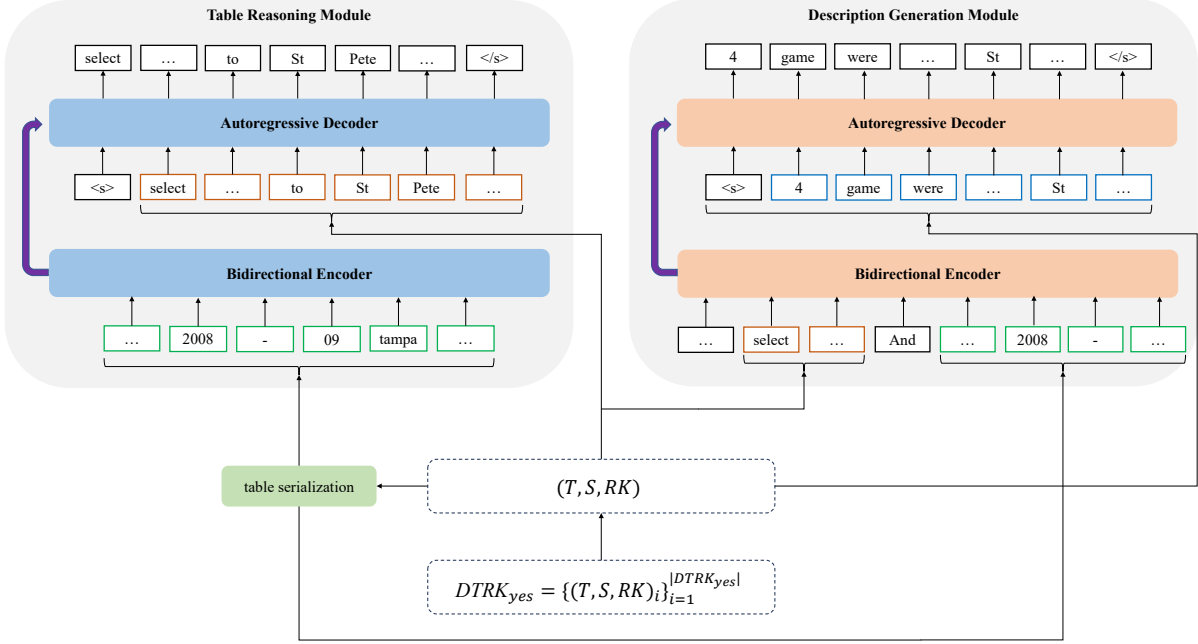


Figure 6: Training details of the generation model.

we need. Now, we need to merge the two parts of the data.

We merge $DTRK_{p1_yes}$ and $DTRK_{p3_yes}$ together and define them as follows:

$$DTRK_{yes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{yes}|} \quad (10)$$

where $|DTRK_{yes}|$ equals $|DTRK_{p1_yes}|$ plus $|DTRK_{p3_yes}|$.

3.2.5 Generation Model

$DTRK_{yes}$ and DT differ in two aspects: first, the former has a smaller size; second, the former provides an additional piece of reasoning knowledge for each data point compared to the latter. This reasoning knowledge should be an intermediate process that the model can generate on its own, with the aim of better generating descriptions.

Following the work of Liu et al. (2024), we decompose the LT2T task into two subtasks: table reasoning and description generation. We implemented a pipeline-based generation model using the BART-large model as the foundation to accomplish these two subtasks. Figure 6 illustrates the training details of the generation model using the data from Figure 1 as an example. The generation model is divided into two independent modules: table reasoning module and description generation module.

The training objective of the table reasoning module is to generate a reasoning knowledge based

on table. We expect the table reasoning module to learn the experience knowledge embedded in reasoning knowledge, thereby enabling it to perform logical reasoning based on tabular data.

The training objective of the description generation module is to generate a description based on table and reasoning knowledge. Since the table reasoning module has already completed the reasoning based on the table, the description generation module's task now shifts to generate a summarizing description based on the provided reasoning knowledge and table.

This pipeline generation model decomposes the originally complex task of table-to-text generation into two relatively simpler subtasks: reasoning and generation. Each module focuses solely on its respective subtask, which facilitates the model's ability to uncover logical relationships within the table and produce accurate and logically faithful descriptions.

The prediction process of this generation model is as follows: (1) Input a serialized table into the table reasoning module, and the table reasoning module generates a reasoning knowledge based on the table information. (2) The reasoning knowledge generated by the table reasoning module is appended to the serialized table. (3) The concatenated sequence is then input into the description generation module, which combines the table and the reasoning knowledge to generate a description

that is logically faithful to the facts presented in the table.

4 Experiments

In this section, we describe our evaluation methodology, including the evaluation metrics and comparison baselines, evaluation and ablation study to thoroughly assess our model.

4.1 Evaluation Metrics

The evaluation metrics for LT2T are categorized into two types: surface-level matching and logical fidelity.

Similar to LT2T, we employ BLEU-1/2/3 (Papineni et al., 2002) as our evaluation metric, which assesses the consistency between the model’s output and the reference by using n-gram matching.

Following previous work (Chen et al., 2020a; Nan et al., 2022; Zhao et al., 2023b), we assessed our method using two evaluation metrics: SP-Acc, based on semantic parsing, and NLI-Acc, based on entailment scores. During evaluation, SP-Acc converts the predicted description into a logical form (Chen et al., 2020a) and executes it, while NLI-Acc computes the entailment score between the predicted description and the table. Both metrics determine the proportion of descriptions that meet specific criteria, thereby measuring whether a description is logically faithful to the table.

4.2 Baselines

Our method will be compared with the following studies:

GPT-Coarse-to-Fine, a method proposed by Chen et al. (2020a) when introducing LT2T. This method involves copying the reference description and masking key entity information, then appending it to the front of the reference description as training data for the model. This approach partially alleviates the issue of the model generating descriptions that are not logically faithful.

DCVED, an encoder-decoder for the LT2T, which employ the causal intervention method to mitigate spurious correlations (Chen et al., 2021).

R2D2, which addresses the issue of unfaithful data-to-text generation by replacing key information in the description with elements from the same column or by sampling from the model’s prediction distribution. It also introduces a new unlikelihood loss function, training the system to act as both a generator and a faithfulness discriminator (Nan et al., 2022).

LoFT, which employs a model to transform descriptions into logic forms (Chen et al., 2020b) and subsequently trains the model to generate descriptions based on both the tables and the logic forms. By incorporating candidate logic form synthesizers, LoFT improves the fidelity and diversity of the model’s predictions (Zhao et al., 2023b).

HISTALIGN, which enhances the context dependence of language models by introducing a novel Cache-LM training objective to ensure proper cache alignment. This allows the model to effectively utilize historical signals, thereby improving the coherence and faithfulness of the generated descriptions (Wan et al., 2023).

RKT, a framework that introduces explicit reasoning knowledge into LogicNLG, mitigating the issue of erroneous information often generated by end-to-end models that directly infer descriptions from tables (Liu et al., 2024).

4.3 Automatic Evaluation

Table 1 presents the results of our experiment. It can be observed that our method surpasses previous approaches in terms of logical fidelity evaluation metrics. Our method improves SP-Acc by up to 1.4 points and NLI-Acc by 0.7 points. The enhancement in these two metrics demonstrates that our table reasoning module effectively generates reliable reasoning knowledge from tables, and the description generation module uses this reasoning knowledge to guide the generation of more logically faithful descriptions.

Regarding BLEU scores, our method is competitive with previous approaches. However, it appears that further research is necessary. BLEU evaluates the n-gram token matching between reference descriptions and predicted descriptions, without assessing the correctness of the descriptions. However, our method shows significant improvement in logical fidelity metrics, indicating that the descriptions we generated are indeed more accurate. By observing the prediction results, we believe that the low BLEU score was attributed to our generated descriptions being logically faithful but not matching the reference descriptions. This situation may have occurred because the table reasoning module generated reasoning knowledge that did not match the reference descriptions. We provide an example of this situation in Figure 7. The reference description indicates that Pierre Lamine has more points than Shinji Someya, whereas the predicted description states that Mark Cockerell had the highest number

model	bleu1	bleu2	bleu3	SP-Acc	NLI-Acc
GPT-Coarse-to-Fine (sm)(Chen et al., 2020a)	46.6	26.8	13.3	42.7	72.2
GPT-Coarse-to-Fine (med)(Chen et al., 2020a)	49.0	28.3	14.6	45.3	76.4
DCVED(Chen et al., 2021)	49.5	28.6	15.3	43.9	76.9
R2D2(Nan et al., 2022)	51.8	32.4	18.6	50.8	85.6
LOFT(Zhao et al., 2023b)	48.1	27.7	14.9	57.7	86.9
HISTALIGN(Wan et al., 2023)	56.7	37.6	26.3	53.1	85.7
RKT(Liu et al., 2024)	54.8	34.1	19.7	59.6	88.1
Ours	55.2	34.3	20.0	61.0	88.8

Table 1: Performance results on the LogicNLG test set.

1976 world junior figure skating championships

...	name	...	point	...
...	mark cockerel	...	172.42	...
...	takashi mura	...	165.70	...
...	brian pockar	...	166.62	...
...	pierre lamine	...	150.50	...
...	shinji someya	...	150.34	...

Reference description : Pierre Lamine has a mere 0.16 more Point than Shinji Someya.

Prediction reasoning knowledge : select the row whose points record of all rows is maximum. the name record of this row is Mark Cockerell.

Prediction description : Mark Cockerell had the highest number of Point in the 1976 World Junior Figure Skating Championship.

Figure 7: Prediction example in LogicNLG test set.

model	SP-Acc	NLI-Acc
Previous best	59.6	88.1
ours (L+C)	61.0	88.8
ours (C)	60.4	88.5

Table 2: Ablation study result. (L+C) indicates that the filtering process utilized both the GPT-4o model and the classification model. (C) indicates that the filtering process used only the classification model.

of points.

5 Ablation Study

In Section 3.2.2 and 3.2.3, we utilized the GPT-4o model and a reasoning knowledge correctness classification model to filter out data from *DTRK* that contained correct reasoning knowledge. To further validate the effectiveness of the classification model, we applied it alone to filter dataset *DTRK*, resulting in a new dataset defined as follows:

$$DTRK_{cyes} = \{(T, S, RK)_i\}_{i=1}^{|DTRK_{cyes}|} \quad (11)$$

Subsequently, we retrained the generation model using *DTRK*_{cyes} according to the methodology

outlined in Section 3.2.5.

As shown in Table 2, significant performance improvements were observed even when the classification model was used to filter *DTRK*. This demonstrates the utility of our classification model base on knowledge distillation.

6 Conclusion

In this paper, we propose a framework *reasoning knowledge filter* based on large language models and knowledge distillation. This framework successfully filters out a dataset that is smaller in size compared to the original LogicNLG dataset but has a higher match quality among table, description, and reasoning knowledge triplets. Through this method, we are able to significantly enhance the performance and logical faithfulness of the generation model.

7 Limitations

Although our method improved logical faithfulness, it still falls short of human performance, which indirectly highlights the difficulty of the task. Meanwhile, three-quarters of *DTRK* was filtered by a smaller classification model, which learned the output distribution of GPT-4o through knowledge distillation. However, it is foreseeable that its filtering performance is not as effective as that of GPT-4o.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1908216).

References

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems*, 35:2197–2210.

- Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942.
- Wenqing Chen, Jidong Tian, Yitian Li, Hao He, and Yao-hui Jin. 2021. De-confounded variational encoder-decoder for logical table-to-text generation. 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 5532–5542.
- Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyu Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020b. Logic2text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111.
- Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2021. Hitab: A hierarchical table dataset for question answering and natural language generation. *arXiv preprint arXiv:2108.06712*.
- Paul Holmes-Higgin. 1994. Text generation—using discourse strategies and focus constraints to generate natural language text by kathleen r. mckeown, cambridge university press, 1992, pp 246, £ 13.95, isbn 0-521-43802-0. *The Knowledge Engineering Review*, 9(4):421–422.
- Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. Mixkd: Towards efficient distillation of large-scale language models. In *International Conference on Learning Representations*.
- Baoqiang Liu, Yu Bai, Fang Cai, Shuang Xue, Na Ye, and XinYuan Ye. 2024. Reasoning knowledge transfer for logical table-to-text generation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*.
- Linyong Nan, Lorenzo Jaime Flores, Yilun Zhao, Yixin Liu, Luke Benson, Weijin Zou, and Dragomir Radev. 2022. R2d2: Robust data-to-text with replacement detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6903–6917.
- Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2017. The e2e dataset: New challenges for end-to-end generation. In *18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 201–206. Association for Computational Linguistics.
- 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.
- Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. Lightpaff: A two-stage distillation framework for pre-training and fine-tuning. *arXiv preprint arXiv:2004.12817*.
- Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909.
- David Wan, Shiyue Zhang, and Mohit Bansal. 2023. Hystalign: Improving context dependency in language generation by aligning with history. *arXiv preprint arXiv:2305.04782*.
- Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022a. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

- 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.
- Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.
- Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. 2023. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. *arXiv e-prints*, pages arXiv–2305.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Yilun Zhao, Zhenting Qi, Linyong Nan, Lorenzo Jaime Flores, and Dragomir Radev. 2023b. Loft: Enhancing faithfulness and diversity for table-to-text generation via logic form control. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 554–561.

A Appendix

Table 3 presents all logical operation functions related to reasoning knowledge. Figure 9 and 10 illustrate how data is annotated using GPT-4o.

Name	Arguments	Output	Description
count	view	number	returns the number of rows in the view
only	view	bool	returns whether there is exactly one row in the view
hop	row, header string	object	returns the value under the header column of the row
and	bool, bool	bool	returns the boolean operation result of two arguments
max/min/avg/sum	view, header string	number	returns the max/min/average/sum of the values under the header column
nth_max/nth_min	view, header string	number	returns the n-th max/n-th min of the values under the header column
argmax/argmin	view, header string	row	returns the row with the max/min value in header column
nth_argmax/nth_argmin	view, header string	row	returns the row with the n-th max/min value in header column
eq/not_eq	object, object	bool	returns if the two arguments are equal
round_eq	object, object	bool	returns if the two arguments are roughly equal under certain tolerance
greater/less	object, object	bool	returns if argument 1 is greater/less than argument 2
diff	object, object	object	returns the difference between two arguments
filter_eq/not_eq	view, header string, object	view	returns the subview whose values under the header column is equal/not equal to argument 3
filter_greater/less	view, header string, object	view	returns the subview whose values under the header column is greater/less than argument 3
filter_greater_eq/less_eq	view, header string, object	view	returns the subview whose values under the header column is greater/less or equal than argument 3
filter_all	view, header string	view	returns the view itself for the case of describing the whole table
all_eq/not_eq	view, header string, object	bool	returns whether all the values under the header column are equal/not equal to argument 3
all_greater/less	view, header string, object	bool	returns whether all the values under the header column are greater/less than argument 3
all_greater_eq/less_eq	view, header string, object	bool	returns whether all the values under the header column are greater/less or equal to argument 3
most_eq/not_eq	view, header string, object	bool	returns whether most of the values under the header column are equal/not equal to argument 3
most_greater/less	view, header string, object	bool	returns whether most of the values under the header column are greater/less than argument 3
most_greater_eq/less_eq	view, header string, object	bool	returns whether most of the values under the header column are greater/less or equal to argument 3

Table 3: Logical Operation Functions, same as in logic2text(Chen et al., 2020b).

input	<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge.</p> <p>S:only Gary Player represented South Africa at the 2001 Open Championship.</p> <p>reasoning knowledge: select the rows whose country record fuzzily matches to South Africa. there is only one such row in the table. the player record of this unique row is Gary Player.</p> <p>T: The caption is "2001 open championship". In row 1, the player is Justin Leonard, the country is United States. In row 2, the player is Nick Faldo, the country is England. In row 3, the player is Tom Lehman, the country is United States. In row 4, the player is John Daly, the country is United States. In row 5, the player is Seve Ballesteros, the country is Spain. In row 6, the player is Bob Charles, the country is New Zealand. In row 7, the player is Tony Jacklin, the country is England. In row 8, the player is Tom Watson, the country is United States. In row 9, the player is Gary Player, the country is South Africa.</p>
output	1(YES)
input	<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge.</p> <p>S:only 1 Mass Suicide occured in the united state, in California in 1997.</p> <p>reasoning knowledge:select the rows whose type record fuzzily matches to Mass Suicide. among these rows, select the row whose date record of all rows is 1st minimum. the location record of this row is California.</p> <p>T: The caption is "list of battles and other violent events by death toll". In row 1, the type is Mass Suicide, Murder, the location is Japan, the date is 1945. In row 2, the type is Mass Suicide, the location is Mariana Islands, the date is 1944. In row 3, the type is Mass Human Sacrifice, the location is Templo Mayor, Tenochtitlan, the date is 1487. In row 4, the type is Mass Suicide, the location is Judea, the date is 73. In row 5, the type is Mass Suicide, Murder, the location is Guyana, the date is 1978. In row 6, the type is Mass Suicide, the location is Uganda, the date is 2000, March. In row 7, the type is Mass Suicide, the location is Switzerland, Canada, the date is 1994. In row 8, the type is Mass Suicide, the location is California, the date is 1997. In row 9, the type is Mass Suicide, the location is France, the date is 1995, December 23.</p>
output	0(NO)

Figure 8: Train data for reasoning knowledge correctness classification model.

Prompt
<p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge. If yes, output "The result is yes", otherwise output "The result is no".</p> <p>T: 2008 - 09 tampa bay lightning season ...</p> <p>row number#location 1#wachovia center 2#st pete times forum 3#st pete times forum 4#td banknorth garden 5#hsbc arena 6#bell centre 7#scotiabank place 8#st pete times forum 9#philips arena 10#mellon arena 11#bankatlantic center 12#st pete times forum ...</p> <p>S: 4 game were played at the St Pete Time Forum. reasoning knowledge: select the rows whose location record fuzzily matches to St Pete Times Forum. the number of such rows is 4.</p>
Completion 1
<p>To determine if the fact implied by sentence S ("4 games were played at the St Pete Times Forum") can be inferred from table T, we will follow the reasoning knowledge provided:</p> <ol style="list-style-type: none"> Identify rows whose location matches "St Pete Times Forum": <ul style="list-style-type: none"> Looking at the table T, we can see the following rows with the location "st pete times forum": <ul style="list-style-type: none"> Row 2 Row 3 Row 8 Row 12 We count the matches: <ul style="list-style-type: none"> There are 4 rows that match the location "st pete times forum". Compare the count with the statement in S: <ul style="list-style-type: none"> Sentence S states that "4 games were played at the St Pete Times Forum." The reasoning knowledge states that we should find 4 rows that match. <p>Since we have found 4 rows that correspond to "St Pete Times Forum", the fact implied by S is indeed supported by the data in T according to the reasoning knowledge. Thus, the final conclusion is: The result is yes.</p>
Completion 2
<p>To determine whether the fact implied by sentence S can be inferred from the data in table T, we will follow the reasoning knowledge provided.</p> <ol style="list-style-type: none"> The reasoning knowledge states to "select the rows whose location record fuzzily matches to St Pete Times Forum." We need to count the number of rows in table T that match "St Pete Times Forum." <p>Looking at the data in T:</p> <ul style="list-style-type: none"> Row 2: st pete times forum Row 3: st pete times forum Row 8: st pete times forum Row 12: st pete times forum <p>From the analysis, we see that there are 4 rows that match "St Pete Times Forum".</p> <p>Now, let's evaluate the fact implied by S: "4 games were played at the St Pete Time Forum." Since we have identified 4 rows that match the location "St Pete Times Forum," this supports the implication stated in S.</p> <p>According to the reasoning knowledge, since the number of rows that match is indeed 4, we can conclude that the implication is valid.</p> <p>Thus, the output is: The result is yes.</p>
Completion 3
<p>To determine whether the fact implied by sentence S can be inferred from the data in table T using the reasoning knowledge provided, we can follow these steps:</p> <ol style="list-style-type: none"> Identify the target location from S: The sentence states that 4 games were played at the St Pete Times Forum. Apply the reasoning knowledge: According to the reasoning knowledge, we need to select the rows from T whose location records fuzzily match "St Pete Times Forum". Examine the data in table T: Looking at the table, we can see the locations listed: <ul style="list-style-type: none"> Row 2: st pete times forum Row 3: st pete times forum Row 8: st pete times forum Row 12: st pete times forum Count the matches: There are 4 rows where the location matches "St Pete Times Forum" (rows 2, 3, 8, and 12). <p>Since the reasoning knowledge states that we need to find 4 rows that fuzzily match "St Pete Times Forum", and we have found exactly 4 such rows, we can conclude that the fact implied by sentence S can indeed be inferred from the data in T.</p> <p>Therefore, the final output is: The result is yes.</p>

Figure 9: GPT-4o Annotation Detail.

<p>Prompt</p> <p>Given a table T, a sentence S and a reasoning knowledge. The fact implied by S needs to be inferred from the data in T. Determine whether the fact implied by sentence S can be inferred according to the steps in the reasoning knowledge. If yes, output "The result is yes", otherwise output "The result is no".</p> <p>T: huron , south dakota ...</p> <p>row number#frequency#target city / market#city of license 1#88.7 fm#huron#huron 2#90.5 fm#huron#huron 3#93.3 fm#huron / mitchell#wessington springs 4#105.1 fm#huron#huron 5#89.7 fm#huron#rocklin , california ...</p> <p>S: all Frequency Target the City Of Huron. reasoning knowledge: select the rows whose target city / market record fuzzily matches to Huron. the number of such rows is 1.</p>
<p>Completion 1</p> <p>To determine if the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the table T according to the provided reasoning knowledge, let's break down what needs to happen.</p> <ol style="list-style-type: none"> reasoning knowledge Analysis: The reasoning knowledge states to select rows whose "target city / market" record fuzzily matches "Huron" and the number of such rows must be 1. Table Analysis: Let's examine the rows in table T: <ul style="list-style-type: none"> - Row 1: target city / market is "huron" (matches) - Row 2: target city / market is "huron" (matches) - Row 3: target city / market is "huron / mitchell" (fuzzily could match "huron") - Row 4: target city / market is "huron" (matches) - Row 5: target city / market is "huron" (matches) <p>Thus, we have the following evaluation:</p> <ul style="list-style-type: none"> - Row 1: matches - Row 2: matches - Row 3: fuzzily matches (due to the presence of "huron") - Row 4: matches - Row 5: matches Counting Matches: The total number of rows that match or fuzzily match "Huron" is 5 (Rows 1, 2, 3, 4, and 5). Conclusion Based on reasoning knowledge: The reasoning knowledge requires that the number of such rows must be exactly 1. Since we found 5 rows that match or fuzzily match "Huron", we do not satisfy the condition of having exactly 1 row. <p>Therefore, following the steps outlined in the reasoning knowledge, we find that the fact implied by sentence S cannot be inferred from the data in table T. The output is: "The result is no".</p>
<p>Completion 2</p> <p>To determine whether the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the data in table T according to the specified reasoning knowledge, we can follow the steps outlined in the rule.</p> <ol style="list-style-type: none"> The rule states to select the rows whose "target city / market" record fuzzily matches to "Huron". In table T, we will look for rows where the "target city / market" includes "Huron": <ul style="list-style-type: none"> - Row 1: target city / market = "huron" (matches) - Row 2: target city / market = "huron" (matches) - Row 3: target city / market = "huron / mitchell" (matches fuzzily, as it includes "Huron") - Row 4: target city / market = "huron" (matches) - Row 5: target city / market = "huron" (matches fuzzily, as it is part of the broader match) Counting the matched rows, we have: <ul style="list-style-type: none"> - Row 1: matches - Row 2: matches - Row 3: matches (fuzzy) - Row 4: matches - Row 5: matches (fuzzy) - Total matched rows = 5 The reasoning knowledge also states that the expected number of such rows is 1. Since we found 5 rows that match, this contradicts the rule. <p>Based on this reasoning, the fact implied by sentence S cannot be inferred because the number of matching rows does not meet the condition specified in the reasoning knowledge. Thus, the output is: The result is no.</p>
<p>Completion 3</p> <p>To determine whether the fact implied by sentence S ("all Frequency Target the City Of Huron") can be inferred from the data in table T according to the reasoning knowledge provided, we follow these steps:</p> <ol style="list-style-type: none"> Examine the reasoning knowledge: The reasoning knowledge states that we need to select the rows whose target city / market record fuzzily matches "Huron" and that the number of such rows should be 1. Check the Table: Let's review the rows in table T: <ul style="list-style-type: none"> - Row 1: Target city / market is "huron" (matches) - Row 2: Target city / market is "huron" (matches) - Row 3: Target city / market is "huron / mitchell" (fuzzily matches) - Row 4: Target city / market is "huron" (matches) - Row 5: Target city / market is "huron" (fuzzily matches, but does not exactly match Huron) Count Matching Rows: <ul style="list-style-type: none"> - Rows that match "Huron" (fuzzily or exactly): Row 1, Row 2, Row 3, Row 4, Row 5. - Total matches: 5 rows. Compare with the reasoning knowledge: The reasoning knowledge indicates that there should be exactly 1 row that matches. Since we found 5 rows that match "Huron", this does not meet the criteria of the reasoning knowledge. <p>Given that the condition of having exactly 1 matching row is not fulfilled, we conclude the following: The result is no.</p>

Figure 10: GPT-4o Annotation Detail.