S^{3}HQA: A Three-Stage Approach for Multi-hop Text-Table Hybrid Question Answering

Fangyu Lei[1,2], **Xiang Li**[1,2], **Yifan Wei**[1,2],
Shizhu He[1,2], **Yiming Huang**[1,2], **Jun Zhao**[1,2], **Kang Liu**[1,2]
[1]The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
{leifangyu2022, lixiang2022, weiyifan2021}@ia.ac.cn
{shizhu.he, jzhao, kliu}@nlpr.ia.ac.cn

Abstract

Answering multi-hop questions over hybrid factual knowledge from the given text and table (TextTableQA) is a challenging task. Existing models mainly adopt a retriever-reader framework, which have several deficiencies, such as noisy labeling in training retriever, insufficient utilization of heterogeneous information over text and table, and deficient ability for different reasoning operations. In this paper, we propose a three-stage TextTableQA framework **S^{3}HQA**, which comprises of *retriever*, *selector*, and *reasoner*. We use a *retriever with refinement training* to solve the noisy labeling problem. Then, a *hybrid selector* considers the linked relationships between heterogeneous data to select the most relevant factual knowledge. For the final stage, instead of adapting a reading comprehension module like in previous methods, we employ a *generation-based reasoner* to obtain answers. This includes two approaches: a row-wise generator and an LLM prompting generator (first time used in this task). The experimental results demonstrate that our method achieves competitive results in the few-shot setting. When trained on the full dataset, our approach outperforms all baseline methods, ranking first on the HybridQA leaderboard.[1]

1 Introduction

Question answering systems devote to answering various questions with the evidence located in the structured knowledge base (e.g., table) (Pasupat and Liang, 2015; Yu et al., 2018) or unstructured texts (Rajpurkar et al., 2016). Considering that many questions need to utilize multiple sources of knowledge jointly in real-world applications, the hybrid form of question answering over texts and tables (TextTableQA) has been proposed and attracted more and more attention (Chen et al., 2020b,a; Zhu et al., 2021; Chen et al., 2021; Zhao et al., 2022; Wang et al., 2022a). Fact reasoning (Chen et al., 2020a,b) is a critical question type of TextTableQA. It requires jointly using multiple evidence from tables and texts to reasoning the answers with different operations, such as correlation (e.g., multi-hop) and aggregation (e.g., comparison). Hyperlinks among some table cells and linked passages are essential resources to establish their relationship and support the retrieval and reasoning for multi-hop questions. As shown in Figure 1, answering a complex question Q1 requires jointly reasoning from textual evidence (P1) to table evidence ([R2, Place]) and then to other table evidence ([R2, Athlete]).

H	Year	Score	Athlete	Place
R$_1$	1960	8,683	Rafer Johnson	Eugene
R$_2$	1960	8,709	Philip Mulkey	Memphis
R$_3$	1963	8,089	Chuan-Kwang Yang	Walnut

P$_1$ …Memphis is a city located along the Mississippi River in southwestern Shelby County, Tennessee, United States …

P$_2$ …Chuan-Kwang Yang competed in the decathlon at the 1960 Olympic Games in Rome…

Q1: Who is the athlete in a city located on the Mississippi River?
A1: Philip Mulkey
Q2: In which year did Walnut-born athletes participate in the Rome Olympics?
A2: 1960
Q3: Who is the higher scoring athlete from the cities of Eugene and Walnut? (Comparison)
A3: Rafer Johnson

Figure 1: The examples of HybridQA.

Existing methods consist of two main stages: *retriever* and *reader* (Chen et al., 2020b; Feng et al., 2022). The *retriever* filters out the cells and passages with high relevance to the question, and then the *reader* extracts a span from the retrieval results as the final answer. However, current methods with two stages still have three limitations as follows.

1) **Noisy labeling for training retriever.** Existing retrieval methods usually ignore the weakly supervised answer annotation (Chen et al., 2020b; Wang et al., 2022b; Feng et al., 2022). For the Q2 of Figure 1, we cannot know the specific location

[1]https://codalab.lisn.upsaclay.fr/competitions/7979.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1731–1740

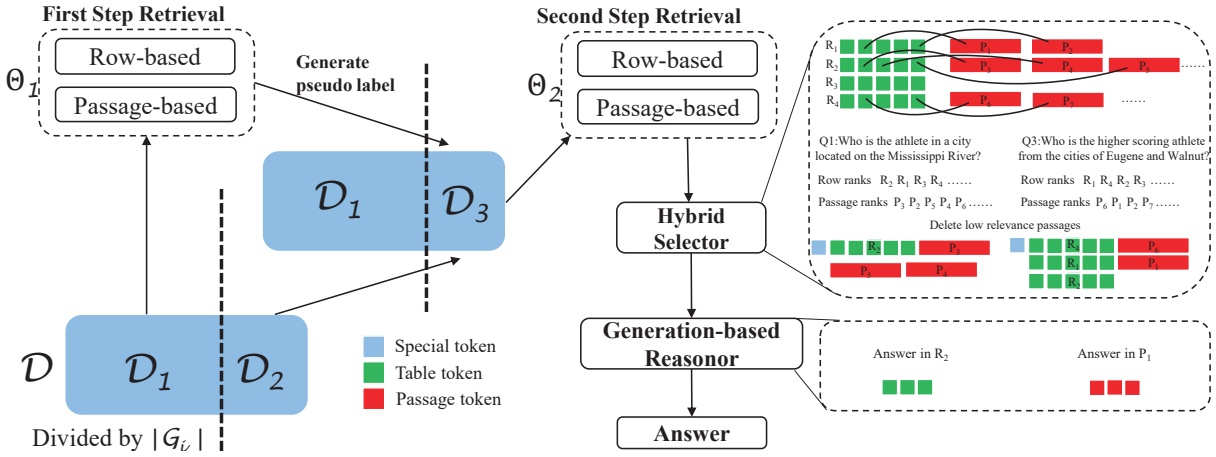

Figure 2: An overview of S^{3}HQA framework. The retrieval stage is divided into two steps. The hybrid selector considers the linked relationships between heterogeneous data to select the most relevant factual knowledge.

of the hybrid evidence, only given the final answer "1960". Therefore, there is a lot of pseudo-true evidence labeled (Marked in green) automatically by string matching, which introduces a lot of evidence noise.

2) **Insufficient utilization of heterogeneous information.** After retrieval, existing methods selected a particular cell or passage for reading to extract the final answer (Chen et al., 2020b; Wang et al., 2022b). As for Q1 in Figure 1, previous models were more likely to choose P1 or the coordinates [R2,Place] to extract the answer. However, these methods seldomly used the hybrid information of table schema and cell-passage hyperlinks, which is the key factor in answering multi-hop questions.

3) **Deficient ability for different reasoning operations.** Previous methods (Eisenschlos et al., 2021; Kumar et al., 2021; Wang et al., 2022b) mainly used an extraction module to obtain answers, which cannot support knowledge reasoning that requires comparison, calculation, and other operations.

In this paper, we propose a three-stage approach S^{3}HQA to solve the above problems. (1) **Retriever with Refinement Training**, we propose a two-step training method, splitting the training data into two parts, so that the noise in the retrieval phase can be alleviated. (2) **Hybrid Selector** has been proposed and selects supporting facts with different granularity and resources depending on the question type. By considering the hybrid data of tables and text, this paper proposes a hybrid selection algorithm that can effectively utilize the heterogeneous information of tables and passages. (3) **Generation-based reasoner** utilizes a generation-based model for addressing different question types. The model allows better aggregation of information on the input side, which not only have better multi-hop reasoning capabilities but also be able to handle comparison and counting questions. Furthermore, we are the first to use the LLM in-context learning approach for table-text hybrid question-answering tasks.

We evaluate our proposed model on the challenging TextTableQA benchmark HybridQA. The empirical results show that our approach outperforms all the existing models[2].

2 Our Approach

2.1 Problem Definition

Given a natural language question $\mathcal{Q} = \{q_i\}_{i=1}^{|\mathcal{Q}|}$ and a table $\mathcal{T}$ with $\langle \mathcal{H}, \mathcal{R} \rangle$, $\mathcal{H}$ indicates the table headers, and $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{R}|}$ indicates the rows with number $|\mathcal{R}|$. Each row r_i is consists of N cells $r_i = \{c_{ij}\}_{j=1}^{N}$. The header's number is also N. Some cells have a linked passage $\mathcal{P}_{ij}$. Our goal aims to generate the answer $\mathcal{A}$ with model Θ, which is a span from table cells or linked passage or a derivation result of counting questions.

2.2 Retriever with Refinement Training

The retriever aims to perform initial filtering of heterogeneous resources. However, accurately labeling the location of answers consumes high labeling costs. For TextTableQA data, the answer $\mathcal{A}$ usually appears in multiple locations, which makes it difficult for us to generate precise retrieval la-

[2]We released the source code at https://github.com/lfy79001/S3HQA

bels. We use a two-step training method, with a row-based retriever and a passage-based retriever for each step.

Inspired by (Kumar et al., 2021), the retrieval has two steps. First, we divide the data $\mathcal{D}$ into two folds according to the string matching labels G_i. Specifically, for a question-answer instance, the answer $\mathcal{A}$ appears one time as $\mathcal{D}_1$, and the instance whose answer $\mathcal{A}$ appears multiple times as $\mathcal{D}_2$. Take the example in Figure 1, Q1, Q3 belongs to $\mathcal{D}_1$ while Q2 belongs to $\mathcal{D}_2$. The data is organized in the form of $[\mathrm{CLS}]q_1q_2...q_{|Q|}[\mathrm{SEP}]\mathrm{c_{i1}c_{i2}...c_{iN}}[\mathrm{SEP}]$ or $[\mathrm{CLS}]q_1q_2...q_{|Q|}[\mathrm{SEP}]\mathrm{p_{ij}}[\mathrm{SEP}]$.

In the first step, we only use $\mathcal{D}_1$ to train a model Θ_1, which data are noiseless. Then in the second step, we use the trained weight Θ_1 to train the model Θ_2. For the input x, the loss function is:

$$L(\Theta_2, x, \mathcal{R}) = \sum_{z \in \mathcal{R}} -q(z) \log p_{\Theta_1}(z|x)$$

where $q(z) = p_{\Theta_1}(z|x, z \in \mathcal{R})$ is the probability distribution given by the model restricted to candidate rows $\mathcal{R}$ containing the answer span, taken here as a constant with zero gradients (Eisenschlos et al., 2021).

Meanwhile, we use a passage-based retriever to enhance the performance of a row-based retriever (PassageFilter). Specifically, we use the passage-based retriever to obtain a prediction score of passage relevance. Based on this score, we reorder the input of the row-based retriever. It avoids the limitation on input sequence length imposed by the pre-trained model.

2.3 Hybrid Selector

This module needs to combine the results of the two granularity retrievers. As for this task, we consider the question type and the relationships between the table and linked passages essential. As shown in Figure 2, the hybrid selector chooses the appropriate data source from the two retrieval results depending on question types.

Specifically, for general *bridge* multi-hop questions, we use a single row and its linked passage. While for *comparison/count* questions, we consider multiple rows and further filter the related sentences, delete the linked paragraphs with the low scores. This not only enables the generation module to obtain accurate information, but also prevents the introduction of a large amount of unrelated information. The selector algorithm outputs a mixed sequence with high relevance based on the relationship between the question, the table, and the passages. The algorithm is shown in Algorithm 1.

Algorithm 1 Hybrid Selector Algorithm.

Input: question $\mathcal{Q}$, table rows $\mathcal{R}$, linked passages $\mathcal{P}$, row-based retriever Θ_R, passage-based retriever Θ_P, selector target row count N_S
Output: generator input $\mathcal{S}$
Get the row/passage ordered list by relevant scores
$\mathcal{O}_\mathcal{R} \leftarrow sort(\Theta_R(\mathcal{Q}, \mathcal{R}))$
$\mathcal{O}_\mathcal{P} \leftarrow sort(\Theta_P(\mathcal{Q}, \mathcal{P}))$
$\mathrm{p^{type}} \leftarrow Classification(Q)$
if $\mathrm{p^{type}} = bridge$ **then**
  if $\mathcal{O}_\mathcal{P}[0]$ in $\mathcal{O}_\mathcal{R}[0]$ **then**
    $\mathcal{S} \leftarrow \mathcal{Q} + \mathcal{O}_\mathcal{R}[0]$
  else
    $\mathcal{S} \leftarrow \mathcal{Q} + \mathcal{O}_\mathcal{R}[0] + \mathcal{O}_\mathcal{P}[0]$
  end if
else
  $\mathcal{O}_{\mathcal{PC}} \leftarrow \mathcal{P}[len(\mathcal{O}_\mathcal{P})//2 :]$
  $\mathcal{S} \leftarrow \mathcal{Q} + \mathcal{O}_\mathcal{R}[0 : N_S] - \mathcal{O}_{\mathcal{PC}}$
end if
return $\mathcal{S}$

2.4 Generation-based Reasoner

The results of the selector take into account both two granularity. Unlike the previous approaches, which were based on a span extraction module, we use a generation-based model for answer prediction.

2.4.1 Row-wise generator

To generate an accurate answer string $\mathcal{A} = (a_1, a_2, ..., a_n)$ given the question $\mathcal{Q}$ and selection evidence $\mathcal{S}$, we perform lexical analysis to identify the question type, such as counting or comparison, by looking for certain keywords or comparative adjectives. We utilize two special tags $\langle\mathrm{Count}\rangle$ and $\langle\mathrm{Compare}\rangle$, which indicates the question types.

We then use the results of the passage retriever to rank the passages in order of their relevance, eliminating the impact of model input length limitations. Finally, we train a Seq2Seq language model with parameters Θ, using the input sequence $\mathcal{Q}, \mathcal{S}$ and the previous outputs $a_{<i}$ to optimize the product of the probabilities of the output sequence $a_1, a_2, ..., a_n$:

$$\mathcal{A} = argmax \prod_{i=1}^{n} P(a_i|a_{<i}, \mathcal{Q}, \mathcal{S}; \Theta)$$

2.4.2 LLM prompting generator

With the emergence of large language models, In-Context Learning (Dong et al., 2022) and Chain-of-Thought prompting (Wei et al., 2022) have become

	Table				Passage				Total			
	Dev		Test		Dev		Test		Dev		Test	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Unsupervised-QG (Pan et al., 2021)	-	-	-	-	-	-	-	-	25.7	30.5	-	-
HYBRIDER (Chen et al., 2020b)	54.3	61.4	56.2	63.3	39.1	45.7	37.5	44.4	44.0	50.7	43.8	50.6
DocHopper (Sun et al., 2021)	-	-	-	-	-	-	-	-	47.7	55.0	46.3	53.3
MuGER[2] (Wang et al., 2022b)	60.9	69.2	58.7	66.6	56.9	68.9	57.1	68.6	57.1	67.3	56.3	66.2
POINTR (Eisenschlos et al., 2021)	68.6	74.2	66.9	72.3	62.8	71.9	62.8	71.9	63.4	71.0	62.8	70.2
DEHG (Feng et al., 2022)	-	-	-	-	-	-	-	-	65.2	**76.3**	63.9	75.5
MITQA (Kumar et al., 2021)	68.1	73.3	68.5	74.4	66.7	75.6	64.3	73.3	65.5	72.7	64.3	71.9
MAFiD (Lee et al., 2023)	69.4	75.2	68.5	74.9	66.5	75.5	65.7	75.3	66.2	74.1	65.4	73.6
S[3]HQA	**70.3**	**75.3**	**70.6**	**76.3**	**69.9**	**78.2**	**68.7**	**77.8**	**68.4**	75.3	**67.9**	**75.5**
Human	-	-	-	-	-	-	-	-	-	-	88.2	93.5

Table 1: Performance of our model and related work on the HybridQA dataset.

two particularly popular research topics in this field. In this paper, we introduce a prompting strategy for multi-hop TextTableQA.

We utilize selection evidence $\mathcal{S}$ and apply LLM-based prompting. We conducted experiments on both vanilla prompting and chain-of-thought prompting in zero-shot and few-shot scenarios.

3 Experiment

3.1 Experiment Setup

Datasets We conduct experiments on HybridQA (Chen et al., 2020b). The detailed statistics are shown in Appendix A. For evaluation, we followed the official evaluation to report exact match accuracy and F1 score.

Implementation details The implementation details are shown in Appendix B. The experimental results are the average of five times results.

3.2 Fully-supervised Results

Table 1 shows the comparison results between our models with previous typical approaches on both development and test sets. It shows that our proposed S[3]HQA works significantly better than the baselines in terms of EM and F1 on HybridQA. The results indicate that S[3]HQA is an effective model for multi-hop question answering over tabular and textual data. Specifically, it can effectively handle multi-hop reasoning and make full use of heterogeneous information.

However, we found that our approach was outperformed by the DEHG model (Feng et al., 2022) in terms of F1 score on the Dev set. We speculate that this might be because the DEHG approach uses their own Open Information Extraction (OIE) tool.

Model	Dev	
	EM	F1
Zero-shot prompt		
GPT3.5 direct	33.1	50.5
GPT3.5 CoT	52.9	66.6
Few-shot prompt (2-shot)		
GPT3.5 direct	57.1	68.8
GPT3.5 CoT	**60.3**	**72.1**

Table 2: Performance Comparison of LLM-Prompting Method on Zero-Shot and Few-Shot Scenarios for HybridQA Dataset.

3.3 LLM-prompting Results

We present our zero-shot and few-shot results in Table 2. "**Direct**" refers to a simple prompting method where only the question, context, and answer are provided to the model without any additional reasoning process. In contrast, "**CoT**" involves a human-authored Chain-of-Thought reasoning process that provides a more structured and logical way of prompting the model. The experiments demonstrate that in-context learning used to prompt large language models can achieve promising results. Specifically, utilizing the Chain-of-Thought prompt method can significantly enhance the model's performance.

However, it's worth noting that there is still a performance gap compared to fine-tuning the model on the full dataset (Table 1). Fine-tuning allows the model to learn more specific information about the TextTableQA task, resulting in better performance. Nevertheless, our results show that the LLM-prompting method can be a useful alternative to fine-tuning, especially when there is a limited amount of labeled data available.

3.4 Ablation Studies

We conduct ablation studies on the test set. We validate the effects of three modules: *retriever with refinement training*, *hybrid selector*, and *generation-based reasoner*. The retriever performs initial filtering of heterogeneous resources; Selectors combined with hyperlinks further identify the exact evidence needed to answer multi-hop questions; and the reasoner uses the selection evidence to obtain the final answer.

Model	Top1
S^3HQA-Retriever$_{DB}$	**88.0**
S^3HQA-Retriever$_{BE}$	87.3
w/o Refinement training	84.1
w/o PassageFilter	85.3
Vanilla-Retriever$_{BE}$	82.0

Table 3: Ablation study of retrieval results. DB and BE denote models based on Deberta-base (He et al., 2020) and BERT-base-uncased (Devlin et al., 2018), respectively

Model	EM	F1
S^3HQA	**67.9**	**76.5**
w/o hybrid selector	65.0	74.9
w/o special tags	67.2	76.0
BERT-large reader	66.8	75.8

Table 4: Ablation study of S^3HQA.

Effect of proposed retriever. As shown in the Table 3, under the setting of using the BERT-base-uncased model, sing the BERT-base-uncased model setting, the retriever with *refinement training* achieved 87.2. When we use Deberta-base, the top1 retrieval performance improved by 0.8%. For *w/o refinement training*, we use the entire data directly for training, the top1 recall drops about 3.2%. For *w/o PassageFilter*, we remove the mechanism, the top1 recall drops about 3.2%. For *Vanilla-Retriever*, we use the row-based retriever (Kumar et al., 2021) and remove all our mechanisms, the top1 score drops about 5.3%. This shows that our model can solve the weakly supervised data noise problem well.

Effect of hybrid selector. As shown in the Table 4, we removed the selector of S^3HQA and replaced it with the previous cell-based selector (Wang et al., 2022b). This method directly uses the top1 result of the row retriever as input to the generator. *w/o hybrid selector* shows that the EM drops 2.9% and F1 drops 1.6%, which proves the effectiveness of our selector approach.

Effect of reasoner. As shown in the Table 4, we design two baselines. *BERT-large reader* (Chen et al., 2020b; Wang et al., 2022b) uses BERT (Devlin et al., 2018) as encoder and solves this task by predicting the start/end tokens. *w/o special tags* deletes the special tags. Both the two experiments demonstrate our S^3HQA reasoner performs the best for HybridQA task.

4 Related Work

The TextTableQA task (Wang et al., 2022a) has attracted more and more attention. As for multi-hop type dataset, previous work used pipeline approach (Chen et al., 2020b), unsupervised approach (Pan et al., 2021), multi-granularity (Wang et al., 2022b), table pre-trained language model (Eisenschlos et al., 2021), multi-instance learning (Kumar et al., 2021) and graph neural network (Feng et al., 2022) to solve this task. As for numerical reasoning task, which is quite different from multi-hop type dataset, there is also a lot of work (Zhu et al., 2021; Zhao et al., 2022; Zhou et al., 2022; Lei et al., 2022; Li et al., 2022; Wei et al., 2023) to look at these types of questions. Unlike these methods, our proposed three-stage model S^3HQA can alleviate noises from weakly supervised and solve different types of multi-hop TextTableQA questions by handling the relationship between tables and text.

5 Conclusion

This paper proposes a three-stage model consisting of retriever, selector, and reasoner, which can effectively address multi-hop TextTableQA. The proposed method solves three drawbacks of the previous methods: noisy labeling for training retriever, insufficient utilization of heterogeneous information, and deficient ability for reasoning. It achieves new state-of-the-art performance on the widely used benchmark HybridQA. In future work, we will design more interpretable TextTableQA models to predict the explicit reasoning path.

Limitations

Since the multi-hop TextTableQA problem has only one dataset HybridQA, our model has experimented on only one dataset. This may lead to a lack

of generalizability of our model. Transparency and interpretability are important in multi-hop question answering. While our model achieves the best results, the model does not fully predict the reasoning path explicitly and can only predict the row-level path and passage-level path. In future work, we will design more interpretable TextTableQA models.

Acknowledgements

This work was supported by the National Key R&D Program of China (2022ZD0160503) and the National Natural Science Foundation of China (No.U1936207, No.61976211). This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDA27020100), the Youth Innovation Promotion Association CAS, Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004) and CCF-DiDi GAIA Collaborative Research Funds for Young Scholars.

References

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W Cohen. 2020a. Open question answering over tables and text. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Julian Eisenschlos, Maharshi Gor, Thomas Mueller, and William Cohen. 2021. Mate: Multi-view attention for table transformer efficiency. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619.

Yue Feng, Zhen Han, Mingming Sun, and Ping Li. 2022. Multi-hop open-domain question answering over structured and unstructured knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 151–156.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Vishwajeet Kumar, Saneem Chemmengath, Yash Gupta, Jaydeep Sen, Samarth Bharadwaj, and Soumen Chakrabarti. 2021. Multi-instance training for question answering across table and linked text. *arXiv preprint arXiv:2112.07337*.

Sung-Min Lee, Eunhwan Park, Daeryong Seo, Donghyeon Jeon, Inho Kang, and Seung-Hoon Na. 2023. Mafid: Moving average equipped fusion-in-decoder for question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2292–2299.

Fangyu Lei, Shizhu He, Xiang Li, Jun Zhao, and Kang Liu. 2022. Answering numerical reasoning questions in table-text hybrid contents with graph-based encoder and tree-based decoder. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1379–1390.

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.

Xiao Li, Yin Zhu, Sichen Liu, Jiangzhou Ju, Yuzhong Qu, and Gong Cheng. 2022. Dyrren: A dynamic retriever-reranker-generator model for numerical reasoning over tabular and textual data. *arXiv preprint arXiv:2211.12668*.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5866–5880.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Haitian Sun, William W Cohen, and Ruslan Salakhutdinov. 2021. End-to-end multihop retrieval for compositional question answering over long documents.

Dingzirui Wang, Longxu Dou, and Wanxiang Che. 2022a. A survey on table-and-text hybridqa: Concepts, methods, challenges and future directions. *arXiv preprint arXiv:2212.13465*.

Yingyao Wang, Junwei Bao, Chaoqun Duan, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022b. MuGER2: Multi-granularity evidence retrieval and reasoning for hybrid question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6687–6697, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Yifan Wei, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. Multi-view graph representation learning for answering hybrid numerical reasoning question. *arXiv preprint arXiv:2305.03458*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600.

Yongwei Zhou, Junwei Bao, Chaoqun Duan, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022. Unirpg: Unified discrete reasoning over table and text as program generation. *arXiv preprint arXiv:2210.08249*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. 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 3277–3287.

A HybridQA Dataset

HybridQA is a large-scale, complex, and multi-hop TextTableQA benchmark. Tables and texts are crawled from Wikipedia. Each row in the table describes several attributes of an instance. Each table has its hyperlinked Wikipedia passages that describe the detail of attributes. It contains 62,682 instances in the train set, 3466 instances in the dev set and 3463 instances in the test set.

Split	Train	Dev	Test	Total
In-Passage	35,215	2,025	20,45	39,285 (56.4%)
In-Table	26,803	1,349	1,346	29,498 (42.3%)
Computed	664	92	72	828 (1.1%)
Total	62,682	3,466	3,463	69,611

Table 5: Data Split: In-Table means the answer comes from plain text in the table, and In-Passage means the answer comes from certain passage.

B Implementation Details

B.1 Fully-supervised Setting

We utilize PyTorch (Paszke et al., 2019) to implement our proposed model. During pre-processing, the input of questions, tables and passages are tokenized and lemmatized with the NLTK (Bird, 2006) toolkit. We conducted the experiments on a single NVIDIA GeForce RTX 3090.

In the retriever stage, we use BERT-base-uncased (Devlin et al., 2018) and Deberta-base (He et al., 2020) to obtain the initial representations. For the first step, batch size is 1, epoch number is 5, learning rate is 7e-6 (selected from 1e-5, 7e-6, 5e-6). The training process may take around 10 hours. For the second step, we use a smaller learning rate 2e-6 (selected from 5e-6, 3e-6, 2e-6), epoch number is 5. The training process may take around 8 hours. In the selector stage, target row count N_S is 3. In the generator stage, we use BART-large language model (Lewis et al., 2020), the learning rate is 1e-5 (selected from 5e-5, 1e-5, 5e-6), batch size is 8, epoch number is 10, beam size is 3 and max generate length is 20.

B.2 LLM-prompting Setting

We use the OpenAI GPT-3.5 (text-davinci-003) API model with the setting $temperature = 0$ in our experiments. For the few-shot setting, we use 2 shots. To elicit the LLM's capability to perform multi-hop reasoning, we use the text "Read the following table and text information, answer a question. Let's think step by step." as our prompt.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitation

☑ A2. Did you discuss any potential risks of your work?
Section Limitation

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section2, Section3

☑ B1. Did you cite the creators of artifacts you used?
Section1, 2, 4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section3.2, Section3.3

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section3

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

Section3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section3.1 and Appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.