KET-QA: A Dataset for Knowledge Enhanced Table Question Answering

Mengkang Hu[1], Haoyu Dong*[2], Ping Luo[1], Shi Han[2], Dongmei Zhang[2]

[1] The University of Hong Kong, [2]Microsoft
mkhu@connect.hku.hk, pluo.lhi@gmail.com,
{hadong, shihan, dongmeiz}@microsoft.com

Abstract

Due to the concise and structured nature of tables, the knowledge contained therein may be incomplete or missing, posing a significant challenge for table question answering (TableQA) systems. However, most existing datasets either overlook the challenge of missing knowledge in TableQA or only utilize unstructured text as supplementary information for tables. In this paper, we propose to use a knowledge base (KB) as the external knowledge source for TableQA and construct a dataset KET-QA with fine-grained gold evidence annotation. Each table in the dataset corresponds to a sub-graph of the entire KB, and every question requires the integration of information from both the table and the sub-graph to be answered. To extract pertinent information from the vast knowledge sub-graph and apply it to TableQA, we design a retriever-reasoner structured pipeline model. Experimental results demonstrate that our model consistently achieves remarkable relative performance improvements ranging from 1.9 to 6.5 times on EM scores across three distinct settings (fine-tuning, zero-shot, and few-shot), in comparison with solely relying on table information. However, even the best model achieves a 60.23% EM score, which still lags behind the human-level performance, highlighting the challenging nature of KET-QA for the question-answering community.

Keywords: Table Question Answering, Knowledge Base

1. Introduction

As a kind of distinct information source, tables are extensively researched for the task of table question answering (TableQA) with numerous practical applications (Pasupat and Liang, 2015; Yu et al., 2018; Chen et al., 2020). Its objective is to answer questions by utilizing specific tables as context. However, owing to the inherent conciseness and organized structure of tables, the information they contain may be incomplete or absent. Consequently, humans, as well as question-answering (QA) systems, may necessitate background knowledge to acquire comprehensive information. These questions that are difficult to answer due to missing information in the table have sparked research interest in addressing the need for external knowledge (Cheng et al., 2023). As is explored in TaCube (Zhou et al., 2022), approximately 10% of the samples in WTQ (Pasupat and Liang, 2015) belong to this category. We define *External Knowledge* as factual information required to answer a given question beyond what is provided in the table. For example, in Figure 1, to answer question 1 *"What was the release date of the studio album from the artist who signed to the record label GOOD Music?"*, a QA system needs to know not only the record label to which each artist belongs in the table but also the release dates of each album, both of which are missing from the table. We consider handling external knowledge required samples a significant challenge for current TableQA systems.

As shown in Table 2, most existing TableQA datasets do not explicitly emphasize the inclusion of external knowledge required questions, although annotators may introduce their prior knowledge during the annotation process. While some efforts have been made to incorporate textual information as external knowledge (Chen et al., 2020; Zhu et al., 2021a; Chen et al., 2021b), there has been a significant oversight in leveraging knowledge graphs, which are widely recognized as an equally prevalent knowledge source. To address this gap, we propose KET-QA, in which each table is associated with a sub-graph from the Wikidata knowledge base (Vrandecic and Krötzsch, 2014), serving as supplementary information for question answering. Each question in KET-QA requires external knowledge to answer, necessitating the integration of the table and knowledge base. To construct KET-QA, we face two main challenges: **(i)** Identifying tables that can be well augmented with an external knowledge base is complex. We find an inherent mapping relationship exists between Wikipedia pages and Wikidata entities (Vrandecic and Krötzsch, 2014), and cells in Wikipedia tables are well linked with Wikipedia pages. These factors naturally connect Wikipedia tables to Wikidata. **(ii)** Proposing natural external knowledge required questions is labour-intensive. Alternatively, we choose to re-annotate natural human-created questions in HybridQA, which leverages unstructured passages in Wikipedia as an external knowledge

*Corresponding author.

LREC-COLING 2024, pages 9705–9719
20-25 May, 2024.

source. We start by substituting the external knowledge source in HybridQA with Wikidata, and cells that initially corresponded to Wikipedia pages are now replaced with entities from Wikidata. Then, we extract a one-hop sub-graph for each table with its corresponding entities. Subsequently, we employ a two-stage annotation approach to re-annotate the question-answer pairs. In the first stage, annotators assess whether a sample is self-contained. That is, to answer the question, it is necessary to incorporate information from the knowledge base, and no additional data is required. In the second stage, annotators annotate the fine-grained gold evidence necessary to answer the given question from the sub-graph. Finally, we collected 9,421 questions and 5,721 tables. Each table corresponds to a sub-graph with 1,696.7 triples on average. Also, we believe that the annotation of fine-grained gold evidence for each question can facilitate a more in-depth exploration of external knowledge required samples within the TableQA research community. Examples from KET-QA are depicted in Figure 1.

Incorporating a knowledge base into the TableQA process poses two challenges: **(i)** The amount of information contained within the grounded sub-graph remains substantial and redundant for a specific question; **(ii)** Integrating three different types of data, namely questions (unstructured text), tables (semi-structured), and a knowledge base (structured), for reasoning purposes. To address the aforementioned challenges, we devise a retriever-reasoner pipeline model, which consists of two steps: first, retrieving relevant triples from the sub-graph based on the given question, and then utilizing a pre-trained language model (PLM) to incorporate the question, table, and retrieved triples, ultimately producing the final answer. This process is illustrated in Figure 3. Benefiting from the fine-grained annotations of gold evidence in KET-QA, we primarily focus on optimizing the retriever in terms of both time efficiency and performance. The resulting **M**ultistage **KB R**etriever (*MKBR*) not only demonstrated state-of-the-art performance with an 83.47 Recall@20 score but also achieved a remarkable speed improvement of 9.3 times. In the final question-answering experiments, incorporating the information from the knowledge graph led to relative performance enhancements ranging from 1.9 to 6.5 times and absolute improvements of 11.66% to 44.64% in terms of EM scores across different models and settings (fine-tuning, few-shot, zero-shot), as compared to solely utilizing table information for answering questions. Moreover, we conducted a comprehensive comparison with two additional external sources of knowledge (LLM-generated knowledge and unstructured text). This extensive comparison further confirmed the benefits and advantages of utilizing a knowledge graph. Despite the substantial performance boost achieved by incorporating the information stored in the knowledge graph into TableQA, its performance still falls short compared to human-level performance, with the highest 60.23% EM score. Consequently, we consider KET-QA a challenging problem for the question-answering community. To analyze further the bottlenecks of the current model, we conducted a manual error analysis, revealing potential areas for further improvement in both the retriever and the reasoner.

2. Task Definition

A table T is a structured arrangement of data that is organized into rows, i.e., $T = \{c_{ij}\} = \{r_i\}$, where i and j represent the coordinates of rows and columns, respectively. A knowledge base G (KB) is a collection of factual information that is formalized as a set of statements that can be categorized into two types, i.e., $G = \{(e_1, p, e_2)\} \cup \{(e, a, v)\}$. The first type represents relational triples, where entities e_1 and e_2 are related by a relation p. The second type represents attribute triples, where an entity e has an attribute a with a value v. The set of all possible entities, relations, and attributes is denoted by E, R, and A, respectively. In KET-QA, a cell in the table may be linked to some entities in the KB. For example, each underlined cell in Figure 1 is linked to an entity in Wikidata (Vrandecic and Krötzsch, 2014). We denote this relation between cells and entities as function f. $E' = \bigcup f(c_{ij})$ represents the union of entities corresponding to all cells in the table. We retrieve a sub-graph by taking E' and their one-hop neighbours. That is, each table corresponds to a sub-graph $\mathcal{G}$ from the entire KB, which serves as supplementary information for TableQA.

The process of KET-QA is as follows: given a table T, the grounded knowledge sub-graph $\mathcal{G}$, and a natural language question q, output a that answers the question according to the context. The source of a could be divided into three categories: **(i)** *In-KB*: an entity or attribute value in the knowledge base; **(ii)** *In-Table*: a cell in the table; **(iii)** *Calculated*: a value that was calculated on the table and knowledge base by using numerical operations like "count", "sum", "difference", etc. Example inputs and outputs are shown in Figure 1.

3. Dataset Construction

3.1. Table Collection

According to the data model of Wikidata (Vrandecic and Krötzsch, 2014) there exists alignments between Wikipedia pages and Wikidata entities. Therefore, in order to incorporate knowledge from Wikidata, our collected table should include hyper-

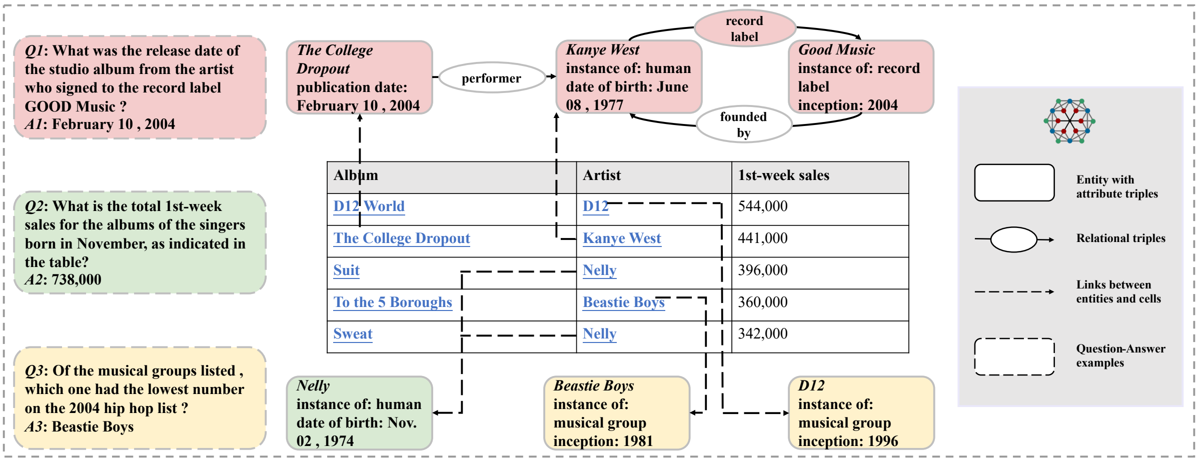

Figure 1: Overview of KET-QA. Only a partial table and knowledge graph are displayed for better visualization. The examples of red, green, and yellow respectively represent three distinct sources of answers: *In-KB*, *Calculated*, and *In-Table*. Corresponding gold evidence for each question is highlighted using the respective colours. Each underlined cell is linked to an entity in Wikidata.

links pointing to Wikipedia pages. With a similar purpose, Chen et al. (2020) has collected 13,000 tables, and 35% of the cells have hyperlinks, thus we took the collection as the table set. Then, we mapped the hyperlinked Wikipedia pages to Wikidata entities by creating an index from a Wikipedia SQL dump with WikiMapper and each table is mapped to 44.3 entities in Wikidata on average.[1]

3.2. Knowledge Base Construction

Following the definition in Section 2, for each table, we took all linked entities for each table and extracted a one-hop Wikidata sub-graph. [2] Then, we performed post-processing on the obtained knowledge sub-graph, discarding attributes such as "globe-coordinate" and "URL" that are not likely to be used.

3.3. Question/Answer Annotation

The collection of question-answer pairs is built on HybridQA, where each question requires the integration of information from Wikipedia passages and tables to be answered. By simply removing the Wikipedia passages, these questions become naturally categorized as external knowledge required samples as defined in Section 1. We hired 43 graduate crowd-sourced workers to manually re-annotate the question-answer pairs. The annotation process consists of two stages. In the first stage, annotators are required to determine whether one given question can be answered by incorporating a knowledge graph and table, which means no additional information sources will be required. In the second stage, annotators are tasked with annotating the fine-grained gold evidence required to answer a specific question from the subgraph corresponding to the current table. One item of gold evidence should include two components: **(i)** the question-relevant triple; **(ii)** the corresponding cell. For convenience, we represent it as $((i, j), t)$, where i and j denote the row and column coordinates of the cell (starting from zero), and t represents a triple. Each question-answer pair would possess multiple pieces of gold evidence. Take the first question in Figure 1 as an example, the gold evidence for this instance would be {((2, 1), (Kanye West, record label, Good Music)), ((2, 0), (The College Dropout, publication date, February 10, 2004))}. Additionally, when there is a mismatch in the answer format or issues arise with the original annotations, the annotators will modify the question-answer pair.

#Words/Ques.	#Words/Answer	#Rows/Table
17.2	3.3	15.8
#Columns/Table	#Entities/Table	#Triples/Table
4.5	41.9	1696.7
Answer in KB	Answer in Table	Calculated Answer
5197	4131	93

Table 1: Core statistics of KET-QA.

3.4. Final Review and Quality Control

Prior to the final review, we assessed the agreement among annotators on a random set of 245 samples. The results indicated "almost perfect agreement" with Fleiss Kappa (Landis and Koch, 1977) scores of 0.89 for the first annotation stage and 0.82 for the second annotation stage. Next, two experienced experts conducted a final review of all annotations, ensuring that any errors or inconsistencies in the annotations were corrected. Then,

[1] https://github.com/jcklie/wikimapper
[2] The data was collected in September 2022 via https://www.wikidata.org/w/api.php

we applied several rules to filter invalid annotations and obtained 9,421 high-quality annotations. Finally, the data is randomly split into train/dev/test sets with 80%/10%/10% respectively.

3.5. Dataset Analysis

Dataset	Size		External Knowledge		
	#Ques.	#Tables	Type	Source	GE
WTQ	22,033	2,108	-	-	-
WikiSQL	80,654	26,521	-	-	-
Spider	10,181	1,020	-	-	-
HiTab	10,672	3,597	-	-	-
FeTaQA	10,330	10,330	-	-	-
HybridQA	69,611	13,000	Text	Wikipedia	No
TAT-QA	16,552	2,757	Text	Financial reports	Yes
FinQA	8,281	2,776	Text	Financial reports	Yes
KET-QA	9,421	5,721	KB	Wikidata	Yes

Table 2: Comparison between datasets. **GE** stands for the annotation of **G**old **E**vidence. Datasets: WTQ (Pasupat and Liang, 2015), WikiSQL (Zhong et al., 2017), Spider (Yu et al., 2018), HiTab (Cheng et al., 2022), FeTaQA (Nan et al., 2022), HybridQA (Chen et al., 2020), TAT-QA (Zhu et al., 2021a), FinQA (Chen et al., 2021b).

Basic Statistics Table 2 shows a comparison of KET-QA with existing table question answering datasets, and Table 1 shows comprehensive statistics. Despite not being one of the largest datasets, KET-QA still has several advantages: **(i)** It is the first TableQA dataset that utilizes a knowledge base as an external knowledge source; **(ii)** It provides alignment between Wikipedia tables and Wikidata entities; **(iii)** It includes fine-grained gold evidence annotations from external knowledge sources, enabling more in-depth analyses.

Question Types As is shown in Figure 2, we analyzed the question types comprehensively and visualized them using the heuristic method proposed by HotpotQA Yang et al. (2018). It is noteworthy that, compared to the original HybridQA (Chen et al., 2020), there is a higher proportion of *Who* questions in KET-QA, accounting for 12.6% as opposed to 9.8% in HybridQA. *Who* questions are typically associated with human entities in Wikidata.

4. Model

4.1. Overview

We propose a retriever-reasoner pipeline model to address the challenges of integrating information from the knowledge graph into TableQA as discussed in Section 1. As shown in Figure 3, our model initially employs a Multistage KB Retriever to retrieve triples from the knowledge sub-graph relevant to the current question. Subsequently, a reasoner (specifically a pre-trained language model) is employed to integrate the table and retrieved triples to answer the current question. The retriever consists of two sub-modules: the Retrieval Bi-Encoder and the Re-Rank Cross-Encoder. These components will be elaborated upon in detail in Section 4.4. The paradigm of incorporating external knowledge into question-answering through a two-stage process has also been studied in the field of Open Domain Question Answering (Zhu et al., 2021b; Chen et al., 2017).

4.2. Probabilistic Formalization

As discussed in Section 2, the task aims to maximize the probability distribution $p(a|T,\mathcal{G},q)$. The whole $\mathcal{G}$ is computationally expensive to process and may contain unrelated related to the specific question. So we retrieve a sub-KB $\mathcal{G}'$ as the evidence for answering the question instead of directly reasoning on $\mathcal{G}$. Considering $\mathcal{G}'$ as latent variables, we rewrite $p(a|T,\mathcal{G},q)$ as follows:

$$p(a|T,\mathcal{G},q) = \sum_{\mathcal{G}'} p_\theta(a|T,\mathcal{G}',q)p_\beta(\mathcal{G}'|T,q)$$

As is indicated in the above equation, the target distribution is jointly modeled by a knowledge base retriever $p_\beta(\mathcal{G}'|T,q)$, and a reasoner conditioned on the table and retrieved triples $p_\theta(a|T,\mathcal{G}',q)$. The goal of training is to find the optimal parameters β and θ which can maximize the log-likelihood.

$$\mathcal{L}(\beta,\theta) = \max_{\beta,\theta} \sum_{\mathcal{D}} \log \sum_{\mathcal{G}'} p_\theta(a|T,\mathcal{G}',q)p_\beta(\mathcal{G}'|T,q)$$

We decouple the two models and train them separately, i.e., first train the retriever p_β and then train the reasoner p_θ on the sub-graph sampled by the retriever (Zhang et al., 2022a; Sachan et al., 2021). The above equation can be approximated as:

$$\mathcal{L}(\beta,\theta) = \max_{\beta,\theta} \sum_{\mathcal{D}} \log p_\theta(a|T,\mathcal{G}',q) + \log p_\beta(\mathcal{G}'|T,q)$$

4.3. Preliminary

Triple Serialization To enable PLMs to handle structured information from a knowledge base, we devised a straightforward approach to transform triples t_1 and t_2 into textual sequence $t_1^* = \texttt{[HEAD]}, \ell(e_1), \texttt{[REL]}, \ell(r), \texttt{[TAIL]}, \ell(e_2)$ and $t_2^* = \texttt{[HEAD]}, \ell(e), \texttt{[REL]}, \ell(a), \texttt{[TAIL]}, v$. Here, [HEAD], [RELATION], [TAIL] are special tokens representing distinct components of triples. The function $\ell(e)$ retrieves the label of e from KB, with the same functionality for r and a.

Table Serialization With a similar purpose, we adopt the same serialization method in Liu et al. (2022c) to flatten a table T into a sequence T^*.

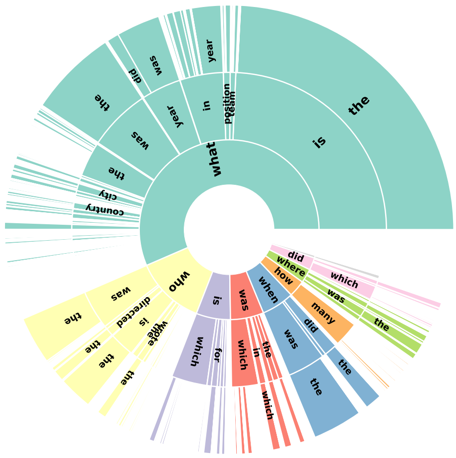

Figure 2: Distribution of question types in KET-QA.

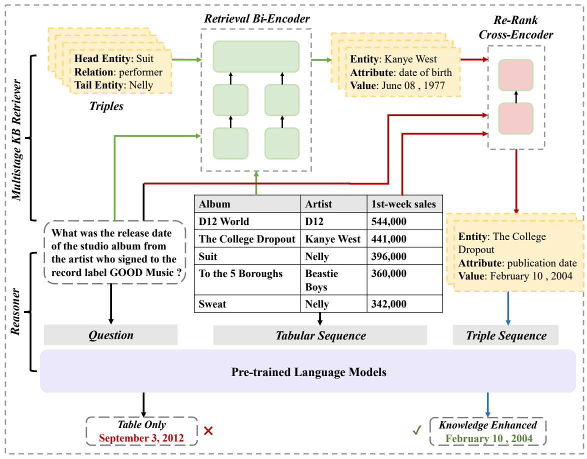

Figure 3: Overview of the retriever-reasoner model.

4.4. Multistage Knowledge Base Retriever

Motivation The retriever we designed draws inspiration from the field of text semantic matching (Giunchiglia and Shvaiko, 2003), among which bi-encoder and cross-encoder are two commonly employed model architectures, as is shown in Figure 4. Cross-encoders can perform better due to their fine-grained cross-attention inside the PLM. However, they tend to have lower efficiency, which can be problematic for real-world TableQA applications. The other model bi-encoders are generally faster and more efficient since they only requires one pass through the input sequence. Therefore, we propose **M**ultistage **KB R**etriever (*MKBR*), which first utilizes a Retrieval Bi-Encoder to retrieve the top N triple candidates and then employ a Re-Ranker Cross-Encoder for more precise scoring.

Retrieval Dataset The i-th instance contains one question q_i , one table T , m relevant (positive) triples $t^+_{i,j}$ and n irrelevant (negative) triples $t^-_{i,j}$. The positive triples are annotated manually. The negative triples are sampled from non-positive triples within the sub-graph $\mathcal{G}$ since considering all negatives would result in an unbearable computational cost. Rather than simply sampling uniformly, we develop a strategy called **k**NN **N**egative **S**ampling (***kNS***). Firstly, the question and all triples are encoded into vectors with a pre-trained sentence embedding model. Then, we take n non-positive triples closest to the question in the vector space as negatives. *kNS* aims to choose informative negative samples, which are also studied in (Robinson et al., 2021; Kumar et al., 2019; Zhang and Stratos, 2021; Xiong et al., 2021).

Retrieval Bi-Encoder consists of a question encoder E_q and a context encoder E_c. The concatenation of the serialized table and triple form the context. E_q and E_c are two independent BERT-style networks (Vaswani et al., 2017), and we take the representation at the `[CLS]` token as the output vector. We define the relevance score using the dot product of the two vectors, i.e., $s(t, q, T) = E_q(q)^\top E_c(T^* \oplus t^*)$. Where $\oplus$ is the concatenation operator. The bi-encoder is optimized with a contrastive loss similar to Karpukhin et al. (2020).

$$\mathcal{L}(q_i, T_i, t^+_{i,1}, \cdots, t^+_{i,p}, t^-_{i,1}, \cdots, t^-_{i,n}) = -\sum_{j=1}^{m} \log \frac{e^{s(t^+_{i,j}, q_i, T_i)}}{e^{s(t^+_{i,j}, q_i, T_i)} + \sum_{k=1}^{n} e^{s(t^-_{i,k}, q_i, T_i)}}$$

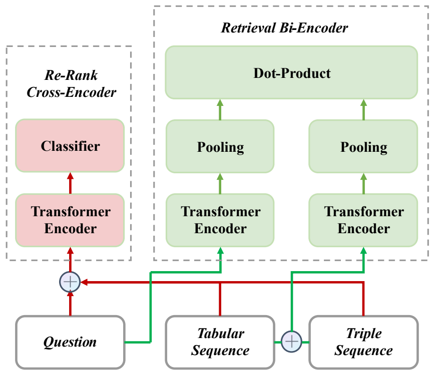

Figure 4: Diagrams of two retrievers. The symbol $\oplus$ denotes the concatenation of text sequences.

Re-Ranker Cross-Encoder directly takes the concatenation of question q, serialized table T^*, and serialized triple t^* as a joint input to the PLM and generate a relevance score ranging from 0 to 1, i.e., $s(t, q, T) = E(q \oplus T^* \oplus t^*)$. Specifically, we take the output logits as the relevance score. The training process of the cross-encoder is modeled as a binary classification problem. Positive triples and negative triples are assigned as 1 and 0, respectively.

Triple-Related Sub-Table The information within the entire table may be redundant for retrieval and could exceed the length limitations of the transformer model. We assume that each row in the table is independent, and the relevance score of a triple t is determined solely by the rows with a mapping relationship with the head entity. Therefore, we propose to improve the performance of the retriever via extracting a triple-related sub-table $\mathcal{T} = \{r_i \in T \mid \exists\, c_{ij} \in r_i, e \in f(c_{ij})\}$ and take $\mathcal{T}$ as the input of retriever. In Section 5.3, we showcase the efficacy of this approach.

4.5. Reasoner

The reasoner aims to answer the question with the information from the table and the retrieved triples. For this phase, we follow the trend of directly generating answers using auto-regressive PLMs in question answering area (Raffel et al., 2020; Lewis et al., 2020). Meanwhile, PLMs are also powerful for fusing and reasoning on heterogeneous data (Zhou et al., 2022; Chen, 2023). Specifically, we concatenate the retrieved information with serialized table and question as the input sequence of PLMs and take the output as the final answer, i.e., $a' = E_r(q \oplus T^* \oplus t_0^* \oplus \cdots \oplus t_k^*)$. Where a' is the predicted answer. E_r stands for the reasoner model. T^* is the serialized table and $\{t_0^*, \cdots t_k^*\}$ is the set of serialized triples obtained from the retriever.

5. Experiments: Evidence Retrieval

5.1. Experimental Setup

Evaluation Metrics: We introduce a modified version of Recall@k (R@k) to evaluate the retrieval performance in the context of KET-QA. The purpose of R@k is to measure the percentage of items of gold evidence that the retriever retrieves:

$$R@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|evidence\ retrieved|_i}{|gold\ evidence|_i}$$

In this equation, the numerator counts the relevant items retrieved up to the k-th position for the i-th instance, while the denominator represents the total number of relevant items for the i-th instance.

Baseline Methods: **(i)** *String Match*: Triples are retrieved based on whether the label of the r or the e_2 for relational triples and a or v for attribute triples matches the words in the question; **(ii)** *Bi-Encoder* and *Cross-Encoder* are used to compare the performance of a single retriever with *MKBR*.

Implementation Details: We use two independent BERT networks (Devlin et al., 2019) (base, uncased) for bi-encoder and a single RoBERTa (Liu et al., 2019) model for cross-encoder. During the training process, we applied *kNS* for Bi-Encoder with $n = 25$, but random sampling for Cross-Encoder with $n = 50$. We chose not to apply *kNS* to the Cross-Encoder because we observed that training it using *kNS* is highly time-consuming, and the model tends to overfit. We also conducted a hyper-parameter search for $n \in \{25, 50, 100\}$ on the dev set to find the optimal n. The optimal *MKBR* model is obtained by selecting the best Retrieval Bi-Encoder and Re-Rank Cross-Encoder separately, based on their performance on the dev set. During the inference process, we set the number of triples retrieved by the Retrieval Cross-Encoder to $N = 200$. We report the R@K of different methods for all experiments on the test set.

5.2. Main Results

Method	Top-1	Top-5	Top-20	Top-100
Random	0.05	0.27	2.94	12.49
String Match	5.87	14.65	28.24	43.66
Cross-Encoder	37.83	63.84	82.14	**94.44**
Bi-Encoder	29.17	51.95	72.12	89.62
MKBR	**38.77**	**66.04**	**83.47**	93.51

Table 3: Comparison between retrieval methods on KET-QA test set using R@k ($k \in \{1, 5, 20, 100\}$).

From Table 3, we can conclude that in scenarios where k is small ($k \leq 20$), *MKBR* consistently outperforms any single retriever model. We attribute this performance improvement primarily to the complementary nature of the bi-encoder and cross-encoder. The Retrieval Bi-Encoder aids the Re-Rank Cross-Encoder in filtering out a subset of triples that are difficult to distinguish, thus enhancing the overall performance.

5.3. Ablation Study

In Section 4.4, we propose to utilize the triple-related sub-table as the final input for the retriever. However, there are two other approaches for table representation: **(i)** Full Table: taking the complete table as input; **(ii)** No Table: not including the table as input. As is shown in Table 4, the table representation method using the triple-related sub-table is superior to the other two approaches. We believe this is because such a representation preserves the relevant information in the table that can aid in retrieval while minimizing redundant information.

5.4. Run-time Efficiency

We conducted run-time tests for *MKBR* on a remote server with four 16G V100 GPUs. When exclusively utilizing a Cross-Encoder, the retrieval

Table Rep.	Top-1	Top-5	Top-20	Top-100
Bi-Encoder				
FT	25.81	49.89	71.75	88.91
NT	24.05	48.67	71.89	89.13
TT	**29.17**	**51.95**	**72.12**	**89.62**
Cross-Encoder				
FT	28.59	58.94	78.32	94.42
NT	12.27	34.41	59.78	85.04
TT	**37.83**	**63.84**	**82.14**	**94.44**

Table 4: Results with different table representations, which can be chosen from {FT(**F**ull **T**able), NT(**N**o **T**able), TT(**T**riple-Related Sub-**T**able)}.

would take 4.92 seconds per question. However, by incorporating *MKBR*, the retrieval process is optimized to 0.53 seconds. On the other hand, the Retrieval Bi-Encoder requires a longer time for the offline generation of knowledge base embeddings. It takes approximately 27.9 seconds per table.

6. Experiments: Question Answering

6.1. Experimental Setup

Evaluation Metrics: We applied two widely-used metrics in the question-answering area: **(i)** Exact Match (EM) is a strict all-or-nothing metric, which represents the percentage of predictions that exactly match the ground truth. **(ii)** F1 is another widely-used metric in QA (Chen et al., 2020; Zhu et al., 2021a), which measures the token overlap between the predicted answer and ground truth.

Baseline Methods: We take table-only models as baselines to explore whether the question can be answered based solely on the table information in the traditional TableQA manner. Specifically, table only models take the concatenation of a question and a table as the input of PLMs and take the output as the predicted answer to the question.

Implementation Details: We selected TaPEx, T5, BART, GPT-3, and ChatGPT as the reasoner models for conducting experiments in fine-tuning, few-shot, and zero-shot settings. The experimental settings for fine-tuning models include using the AdamW optimizer with an initial learning rate of 5e-5, training for 20 epochs, and using a batch size of 24. The few-shot model utilized the KATE (Liu et al., 2022a) method, where five in-context examples were retrieved from the train set for each sample. For zero-shot and few-shot models, we employed a greedy decoding strategy to obtain the final answer. We also performed a hyper-parameter search for the number of retrieved triples k in $\{5, 10, 20, 30\}$ using TaPEX-Large on the dev set. Finally, we chose $k = 20$ based on the F1 score.

6.2. Main Results

As is shown in Table 5, the inclusion of the knowledge base consistently and significantly enhanced the performance across all experimental settings and model types, as reflected by relative improvements of 1.9 to 6.5 times in EM scores and 1.8 to 4.6 times in F1 scores. These findings highlight the effectiveness of leveraging a knowledge base for question-answering tasks and its potential for improving the accuracy and capability of reasoning systems. We also observed that in the table-only scenario, few-shot GPT-3 outperformed the fine-tuned models, indicating that the LLM itself might possess some stored external knowledge. This issue will be further investigated in Section 6.3.

6.3. Comparison of Knowledge Sources

We performed in-depth experiments to compare two distinct external knowledge sources with KB: **(i) LLM-generated Knowledge**: We employed the prompt "*Generate some knowledge about the given question and table*" to instruct an LLM (text-davinci-003) to generate relevant knowledge for answering questions; **(ii) Wikipedia Passages**: We employed the same passage retriever as in Chen et al. (2020) to retrieve relevant knowledge from Wikipedia passages. Note that since KET-QA is built on top of HybridQA. Therefore, each question can be answered by combining information from Wikipedia passages and tables. However, incorporating a KB can still significantly surpass the other two methods regarding EM and F1 scores. We perceive a structured KB to possess several advantages over other forms of knowledge sources: **(i)** It provides structured data and semantic relationships, yielding more precise and consistent knowledge. **(ii)** The semantic relationships between entities can aid the model in comprehending the structure of the table. Furthermore, as KET-QA includes fine-grained gold evidence annotations, we are able to evaluate and optimize the retrieval process.

6.4. Error Analysis

We manually analyzed 100 randomly selected error cases of the few-shot GPT-3 model from the dev set. Errors are categorized into: (1) **Knowledge Uncovered** (*39%*): The provided knowledge does not include the required information. (2) **Erroneous Knowledge** (*1%*): The provided knowledge is detrimental or is factually incorrect. (3) **Reasoning Error** (*42%*): The reasoner failed to provide the correct response even with useful knowledge. (4) **False Negative** (*18%*): Misjudged by the evaluator. The results indicate that the majority of errors are attributed to the inability of the retriever to collect helpful knowledge, as well as the insufficient ability

Model	Table Only				Knowledge Enhanced				Δ			
	Dev		Test		Dev		Test		Dev		Test	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Fine-Tuning												
TAPEX$_{large}$	14.44	18.52	12.83	17.1	**60.62**	**63.22**	56.63	58.75	46.18	44.7	43.8	41.65
BART$_{large}$	9.34	13.41	8.17	12.17	51.7	54.49	52.81	56.16	42.36	41.08	44.64	43.99
BART$_{base}$	7.64	11.57	8.38	11.56	45.65	48.89	46.87	50.28	38.01	37.32	38.49	38.72
T5$_{base}$	9.77	14.05	9.12	12.97	45.54	48.94	46.02	49	35.77	34.89	36.9	36.03
Zero-Shot												
GPT-3	8.07	17.85	10.07	20.11	33.55	45.04	36.69	47.76	25.48	27.19	26.62	27.65
ChatGPT	3.82	7.65	4.03	7.31	17.73	27.53	15.69	26.79	13.91	19.88	11.66	19.48
Few-Shot												
GPT-3	**33.86**	**39.58**	**31.81**	**37.06**	57.86	63.04	**60.23**	**64.89**	24	23.46	28.42	27.83
ChatGPT	20.7	23.95	19.72	23.92	45.01	49.51	43.26	49.53	24.31	25.56	23.54	25.61

Table 5: Performance of different reasoners on KET-QA. Block Δ represents the increase in performance after incorporating knowledge base as an additional source of information. We employ the text-davinci-003 version for GPT-3 and the gpt-3.5-turbo version for ChatGPT.

Knowledge Source	Dev		Test	
	EM	F1	EM	F1
Table Only	14.44	18.52	12.83	17.1
LLM	29.62	34.23	26.51	30.58
Wikipedia Passages	32.27	36.69	28.31	32.24
Knowledge Base	**60.62**	**63.22**	**56.63**	**58.75**

Table 6: Experimental results with various external knowledge sources. We employed TAPEX$_{large}$ as a representative reasoner.

of the reasoner to process knowledge.

7. Related Work

Knowledge Enhanced Models A flurry of QA systems involves using multiple sources of knowledge to answer a wider range of questions (Oguz et al., 2022; He et al., 2023; Zhen et al., 2022; Lan et al., 2021). Available knowledge sources can be divided into: **(i)** unstructured text; **(ii)** structured knowledge bases; **(iii)** semi-structured tables. KG-FiD (Yu et al., 2022) infuses knowledge graph in FiD (Izacard and Grave, 2021) model for Open Domain Question Answering(ODQA) via constructing a graph structure with KB triples and passages. Chen et al. (2020); Zhu et al. (2021a); Chen et al. (2021a) propose to integrate both tabular and textual content to answer questions. UnikQA (Oguz et al., 2022) unifies representations of KB triples and semi-structured tables into unstructured text and performs standard ODQA tasks. Based on previous works, this paper proposes a retriever-reasoner pipeline model, which shares some similarities with the retriever-reader pipeline in ODQA (Chen et al., 2017). However, our work focuses explicitly on the integration of both table and text data into the knowledge base retrieval process, which presents unique challenges. While our model may not introduce significant innovations, the primary contribution of this paper lies in the creation of a valuable language resource KET-QA.

QA over Heterogeneous Information Reasoning over heterogeneous information poses significant challenges. Recent works have demonstrated the potential of a single transformer-based model to fuse heterogeneous information (Xie et al., 2022; Liu et al., 2023, 2022d). These works often unify the representation of different types of information by reducing them to text. Another line of work involves using different models to process data from different structures, e.g., graph neural network for knowledge graph (Yu et al., 2022; Gao et al., 2019). However, the intricate nature of the diverse designs renders it less convenient compared to a transformer-based approach. This paper follows the first line of work, which employs a single transformer model to handle heterogeneous data.

Multistage Retriever Some other works have also utilized a two-stage retriever consisting of retrieval and reranking (Gao et al., 2021; Glass et al., 2022; Zhang et al., 2022b). However, these works typically focus on passage retrieval rather than knowledge base retrieval. While some other works retrieve from a knowledge base with multistage retriever (Baek et al., 2023; Wang et al., 2021), they mainly focus on standard KGQA tasks, without addressing the challenges posed by the tabular data structure. For example, we propose a novel table representation method to address the issue of redundant information in table-based retrievals.

8. Conclusion

We present KET-QA, the first TableQA dataset that incorporates a knowledge base as supplementary information to tables. Significantly, we provide fine-grained gold evidence annotations to facilitate deeper research into the missing knowledge problem of tables resulting from the highly condensed structure. In our experiments, we devise a retriever-reasoner framework to effectively integrate the knowledge base into TableQA. We firmly believe that KET-QA presents an intriguing and demanding challenge for the community to tackle.

9. Limitations

As is shown in Table 2, KET-QA is not large-scale enough compared to other existing datasets due to the complexity of the labeling task. Furthermore, we only considered one-hop connections when extracting the sub-graph from Wikidata. However, suppose we extract sub-graphs with more hops. In that case, it not only provides a broader range of external knowledge for tables but also presents more significant challenges for the reasoner due to the inclusion of more complex structures. We consider the aforementioned limitations to be areas for future improvement and development of the KET-QA. In experiments, we have not provided experiment results of more advanced versions of GPT such as GPT4 due to the access and API frequency limitations.

10. Ethics Statement

This paper introduces KET-QA, an openly accessible English dataset designed for the research community to investigate table question-answering. The annotators we employ possess bachelor's degrees in computer science and are compensated at an hourly wage of $9, which exceeds the local average salary of similar jobs. KET-QA is built on HybridQA (Chen et al., 2020), which is under the MIT license. The knowledge base is constructed on Wikidata (Vrandecic and Krötzsch, 2014), which is under the CC0 public domain license. Both allow us to modify and create new datasets.

11. Bibliographical References

Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. 2023. Direct fact retrieval from knowledge graphs without entity linking. *arXiv preprint arXiv:2305.12416*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. 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, Online. Association for Computational Linguistics.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of bert rerankers in multi-stage retrieval pipeline. In *The 43rd European Conference On Information Retrieval (ECIR)*.

Yan Gao, Jian-Guang Lou, and Dongmei Zhang. 2019. A hybrid semantic parsing approach for tabular data analysis. *ArXiv preprint*, abs/1910.10363.

Fausto Giunchiglia and Pavel Shvaiko. 2003. Semantic matching. *The Knowledge Engineering Review*, 18(3):265–280.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. *arXiv preprint arXiv:2207.06300*.

Xinyi He, Mengyu Zhou, Mingjie Zhou, Jialiang Xu, Xiao Lv, Tianle Li, Yijia Shao, Shi Han, Zejian Yuan, and Dongmei Zhang. 2023. Anameta: A table understanding dataset of field metadata knowledge shared by multi-dimensional data analysis tasks.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Sawan Kumar, Shweta Garg, Kartik Mehta, and Nikhil Rasiwasia. 2019. Improving answer selection and answer triggering using hard negatives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5911–5917, Hong Kong, China. Association for Computational Linguistics.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv preprint arXiv:2105.11644*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

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, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022b. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022c. TAPEX: table pre-training via learning a neural SQL executor. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Qian Liu, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. 2023. From zero to hero: Examining the power of symbolic tasks in instruction tuning. *ArXiv preprint*, abs/2304.07995.

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. 2022d. Uniparser: Unified semantic parser for question answering on knowledge base and database. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8858–8869, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and

Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. 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 6648–6662, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2021. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357, Online. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, Dublin, Ireland. Association for Computational Linguistics.

Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1090–1101, Online. Association for Computational Linguistics.

Yanzhao Zhang, Dingkun Long, Guangwei Xu, and Pengjun Xie. 2022b. Hlatr: enhance multi-stage text retrieval with hybrid list aware transformer reranking. *arXiv preprint arXiv:2205.10569*.

Chaoqi Zhen, Yanlei Shang, Xiangyu Liu, Yifei Li, Yong Chen, and Dell Zhang. 2022. A survey on knowledge-enhanced pre-trained language models. *ArXiv preprint*, abs/2212.13428.

Fan Zhou, Mengkang Hu, Haoyu Dong, Zhoujun Cheng, Fan Cheng, Shi Han, and Dongmei Zhang. 2022. TaCube: Pre-computing data cubes for answering numerical-reasoning questions over tabular data. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2291, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021a. 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, Online. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021b. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

12. Language Resource References

Wenhu Chen and Ming-wei Chang and Eva Schlinger and William Wang and William Cohen. 2021a. *Open Question Answering over Tables and Text*.

Chen, Wenhu and Zha, Hanwen and Chen, Zhiyu and Xiong, Wenhan and Wang, Hong and Wang, William Yang. 2020. *HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data*. Association for Computational Linguistics.

Chen, Zhiyu and Chen, Wenhu and Smiley, Charese and Shah, Sameena and Borova, Iana and Langdon, Dylan and Moussa, Reema and Beane, Matt and Huang, Ting-Hao and Routledge, Bryan and Wang, William Yang. 2021b. *FinQA: A Dataset of Numerical Reasoning over Financial Data*. Association for Computational Linguistics.

Cheng, Zhoujun and Dong, Haoyu and Wang, Zhiruo and Jia, Ran and Guo, Jiaqi and Gao, Yan and Han, Shi and Lou, Jian-Guang and Zhang, Dongmei. 2022. *HiTab: A Hierarchical Table Dataset for Question Answering and Natural Language Generation*. Association for Computational Linguistics.

Nan, Linyong and Hsieh, Chiachun and Mao, Ziming and Lin, Xi Victoria and Verma, Neha and Zhang, Rui and Kryściński, Wojciech and Schoelkopf, Hailey and Kong, Riley and Tang, Xiangru and Mutuma, Mutethia and Rosand, Ben and Trindade, Isabel and Bandaru, Renusree and Cunningham, Jacob and Xiong, Caiming and Radev, Dragomir and Radev, Dragomir. 2022. *FeTaQA: Free-form Table Question Answering*. MIT Press.

Pasupat, Panupong and Liang, Percy. 2015. *Compositional Semantic Parsing on Semi-Structured Tables*. Association for Computational Linguistics.

Denny Vrandecic and Markus Krötzsch. 2014. *Wikidata: a free collaborative knowledgebase*.

Yu, Tao and Zhang, Rui and Yang, Kai and Yasunaga, Michihiro and Wang, Dongxu and Li, Zifan and Ma, James and Li, Irene and Yao, Qingning and Roman, Shanelle and Zhang, Zilin and Radev, Dragomir. 2018. *Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task*. Association for Computational Linguistics.

Victor Zhong and Caiming Xiong and Richard Socher. 2017. *Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning*.

Zhu, Fengbin and Lei, Wenqiang and Huang, Youcheng and Wang, Chao and Zhang, Shuo and Lv, Jiancheng and Feng, Fuli and Chua, Tat-Seng. 2021. *TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance*. Association for Computational Linguistics.

A. More Experimental Results

A.1. Validation Results of Retrievers

The corresponding validation results of comparison between different retrieval methods are shown in Table 7.

	Top-1	Top-5	Top-20	Top-100
Random	0.00	0.61	2.38	13.50
String Match	5.65	15.93	26.49	40.83
Cross-Encoder	36.73	60.89	80.39	**93.10**
Bi-Encoder	25.98	48.74	70.89	88.55
MKBR	**37.58**	**63.10**	**80.99**	91.66

Table 7: Corresponding validation results of Table 3

A.2. Results over Answer Distribution

We divided the dataset based on the answer source and reported the results on the dev set of KET-QA for each subset in Table 8. After incorporating knowledge, the model's performance shows the greatest improvement in the subset where the answer source is In-KB. However, even after incorporating knowledge, the model's performance on calculated answers remains lower compared to the other two categories. This indicates that the model's ability to perform calculations on heterogeneous data still needs improvement.

B. More Details on Dataset

B.1. Dataset Preprocessing

Due to the post-create question-answer annotation approach used in HybridQA, which involves collecting tables and Wikipedia passages and then having annotators label question-answer pairs, there is a significant overlap between some of the questions and the passages from Wikipedia. We consider such examples to be lacking in naturalness. Therefore, we have designed the following rules for filtering: **(i)** We eliminated questions with an LCS Similarity of 0.7 or higher, which was calculated by dividing the length of the Longest Common Sub-sequence (LCS) between the question and its corresponding Wikipedia passage by the length of the question. **(ii)** We employed a fuzzy matching technique to filter and retain question-answer pairs that have corresponding words in the knowledge base but are not found in the table.

B.2. Low-Quality Annotation Filtering Rules

We applied several rules to filter low-quality annotations: **(i)** The source of the answer is invalid. As mentioned in Section 3.3, the answer can only originate from three valid sources. During the annotation process, we instructed the annotators to manually indicate the answer source. In the final review, we employ rule-based methods to trace back the answers and verify their sources as a double-checking measure; **(ii)** Gold evidence was not explicitly marked; **(iii)** The annotated evidence is invalid. For example, the index exceeds the range of the table or there is no corresponding triple in the sub-graph.

C. More Details on Retrievers

C.1. Training of Bi-Encoder

The training process of Bi-Encoder is similar to that in (Karpukhin et al., 2020). We optimize the model using a contrastive learning loss function:

$$\mathcal{L}(q_i, T_i, t^+_{i,1}, \cdots, t^+_{i,p}, t^-_{i,1}, \cdots, t^-_{i,n}) = \\ -\sum_{j=1}^{m} \log \frac{e^{s(t^+_{i,j}, q_i, T_i)}}{e^{s(t^+_{i,j}, q_i, T_i)} + \sum_{k=1}^{n} e^{s(t^-_{i,k}, q_i, T_i)}}$$

C.2. Implementation Details

We use two independent BERT networks (Devlin et al., 2019) (base, uncased) for retrieval bi-encoder and a single RoBERTa (Liu et al., 2019) model for re-ranker cross-encoder. During the training process, both models are trained on the train set of KET-QA with a learning rate of 10-5 using Adam, linear scheduling with warm-up and dropout rate 0.1. Bi-encoder is trained up to 20 epochs with a batch size of 16, while cross-encoder is trained up to 5 epochs with a batch size of 32.

C.3. Negative Sampling

According to Table 9, we conducted experiments with varying numbers of negative examples ($N \in \{25, 50, 100\}$). We observed that increasing the number of negative examples does not necessarily lead to improved model performance. For the cross-encoder, the model achieves its highest performance at $N = 50$, while for the bi-encoder, it is at $N = 25$.

D. More Details on Reasoners

D.1. Introduction

TAPEX (Liu et al., 2022c) guides the language model to mimic a SQL executor on the synthetic corpus, resulting in groundbreaking results on four table-related datasets. We take TAPEX as a representative of tabular language models.

Model	Table Only						Knowledge Enhanced					
	In Table		In-KB		Calculated		In-Table		In-KB		Calculated	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Fine-Tuning												
TAPEX$_{large}$	22.47	27.38	8.04	11.59	**36.36**	**36.36**	55.56	60.47	**59.81**	61.36	**45.45**	**45.45**
BART$_{large}$	15.91	19.72	4.11	8.46	27.27	27.27	49.24	53.46	54.02	56.11	36.36	42.42
BART$_{base}$	11.62	15.77	4.3	8.13	27.27	27.27	40.15	44.68	50.09	52.42	27.27	27.27
T5$_{base}$	15.91	21.9	4.86	7.98	27.27	27.27	45.71	50.83	45.61	47.68	36.36	42.42
Zero-Shot												
GPT-3	13.38	23.93	4.3	13.62	0	4.48	23.48	38.56	44.68	50.53	0	11.89
ChatGPT	7.83	14.14	0.93	2.88	0	5.58	18.69	31.05	17.38	25.42	0	3.21
Few-Shot												
GPT-3	**32.83**	**40.43**	**35.14**	**39.53**	9.09	12.73	**57.32**	**64.38**	58.88	**62.75**	27.27	28.93
ChatGPT	22.22	26.83	20	22.12	0	9.7	46.97	53.89	44.49	47.11	0	8.33

Table 8: Experimental results based on answer distribution on dev set of KET-QA.

#N	Top-1	Top-5	Top-20	Top-100
Bi-Encoder				
25	**29.17**	**51.95**	**72.12**	**89.62**
50	25.68	48.17	68.11	87.94
100	22.42	39.47	62.51	85.03
Cross-Encoder				
25	27.53	54.32	73.76	91.63
50	**37.83**	**63.84**	**82.14**	**94.44**
100	12.49	28.63	44.76	70.22

Table 9: Ablation study on the number of negative numbers (#N). Gray represents the final model selection for *MKBR*

BART (Lewis et al., 2020) and **T5** (Raffel et al., 2020) are representatives of general pre-trained encoder-decoder language models, and have performed exceptionally well on a wide range of NLP tasks.
GPT-3 (Brown et al., 2020) is one of the top-performing models among the large language models with a decoder architecture. It exhibits strong question answering capabilities in both zero-shot and few-shot settings.
ChatGPT is a variant of GPT-3, which is trained using Reinforcement Learning from Human Feedback (RLHF). It excels in natural language conversations and exhibits human-like responses.

Table 10 shows the comparison of parameters of different models.

D.2. Evaluation Metrics

Exact Match The EM score is a strict all-or-nothing metric, which represents the percentage of predictions that exactly match the ground truth.
F1 Score is another widely-used metric in QA (Chen et al., 2020; Zhu et al., 2021a), which measures the token overlap between the predicted answer and ground truth.

D.3. Implementation Details

The experimental settings for fine-tuning models include using the AdamW optimizer with an initial learning rate of 5e-5, training for 20 epochs, and using a batch size of 24. The training process takes about 7.8 hours for BART-Large/TaPEX-Large and 5.2 hours for BART-Base/T5-Base with 4 16G V100 GPUs. The few-shot model utilized the KATE (Liu et al., 2022a) method, where for each sample, five examples were retrieved from the training set. For both the zero-shot model and the few-shot, we employed a greedy decoding strategy ($t = 0$) to obtain the final answer. The inference process typically takes approximately 5.5 seconds per question with OpenAI API.

Model	#Parameter
TAPEX$_{large}$	400 million
BART$_{large}$	400 million
BART$_{base}$	140 million
T5$_{base}$	220 million
GPT-3	175 billion

Table 10: Parameter of reasoners

E. Other External Knowledge Sources

LLM-generated Knowledge Numerous studies have proposed utilizing Large Language Models (LLMs) as databases (Liu et al., 2022b; Yu et al., 2023). We employed the prompt "*Generate some knowledge about the given question and table*" to instruct the LLM (text-davinci-003 in our case) to generate knowledge that is beneficial for answering the current question with a greedy decoding strategy.

Wikipedia Passage As discussed in (Chen et al., 2020), hyperlinked Wikipedia passages can provide additional information that complements the table. Following the methodology outlined in (Chen et al., 2020), we employed the same passage retriever to retrieve relevant knowledge for the current question.

We concatenated the question, the serialized table, and the external knowledge as the input of PLMs and take the output as the final answer, which is the same as in Section 4.5.