TACR: A Table-alignment-based Cell-selection and Reasoning Model for Hybrid Question-Answering

Jian Wu[1*], **Yicheng Xu**[2*], **Yan Gao**[3], **Jian-Guang Lou**[3]
Börje F. Karlsson[4], **Manabu Okumura**[1]
[1]Tokyo Institute of Technology [2]Nanyang Technological University
[3]Microsoft Research Asia [4]Beijing Academy of Artificial Intelligence
wu.j.as@m.titech.ac.jp, yxu040@e.ntu.edu.sg,
{yan.gao, jlou}@microsoft.com, borje@baai.ac.cn, oku@pi.titech.ac.jp

Abstract

Hybrid Question-Answering (HQA), which targets reasoning over tables and passages linked from table cells, has witnessed significant research in recent years. A common challenge in HQA and other passage-table QA datasets is that it is generally unrealistic to iterate over all table rows, columns, and linked passages to retrieve evidence. Such a challenge made it difficult for previous studies to show their reasoning ability in retrieving answers. To bridge this gap, we propose a novel **T**able-**a**lignment-based **C**ell-selection and **R**easoning model (TACR) for hybrid text and table QA, evaluated on the HybridQA and WikiTableQuestions datasets. In evidence retrieval, we design a table-question-alignment enhanced cell-selection method to retrieve fine-grained evidence. In answer reasoning, we incorporate a QA module that treats the row containing selected cells as context. Experimental results over the HybridQA and WikiTableQuestions (WTQ) datasets show that TACR achieves state-of-the-art results on cell selection and outperforms fine-grained evidence retrieval baselines on HybridQA, while achieving competitive performance on WTQ. We also conducted a detailed analysis to demonstrate that being able to align questions to tables in the cell-selection stage can result in important gains from experiments of over 90% table row and column selection accuracy, meanwhile also improving output explainability.

1 Introduction

Text-based question-answering datasets derive answers based on reasoning over given passages (Rajpurkar et al., 2016; Chen et al., 2017; Joshi et al., 2017; Yang et al., 2018), while table-based QA datasets collect tables from sources such as WikiTables (Pasupat and Liang, 2015a; Zhong et al., 2017; Chen et al., 2019). However, datasets combining textual passages and tables, like HybridQA (Chen et al., 2020b), OTT-QA (Chen et al., 2020a), and TAT-QA (Zhu et al., 2021) are more realistic benchmarks. As the answer to a given question may come from either table cells or linked passages, current hybrid QA models usually consist of two components, a retriever to learn evidence and a reasoner to leverage the evidence to derive answers. Such models retrieve evidence from different granularities, coarse-grained (e.g., row or column) or fine-grained (e.g., cell), and directly use a span-based reading comprehension model to reason the answer.

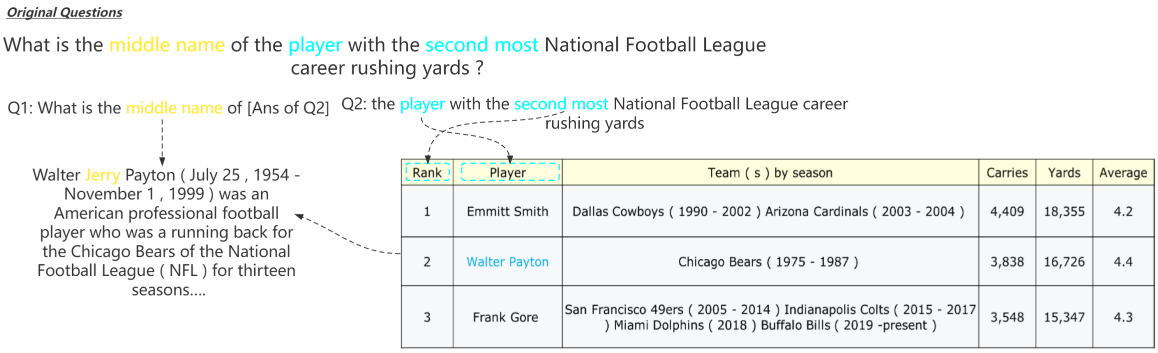
Kumar et al. (2021), for example, chooses coarse-grained regions as evidence, e.g., a table row. Chen et al. (2020b) and Eisenschlos et al. (2021), however, focus on fine-grained units, table cells and linked passages. To preserve the advantages and eliminate the disadvantages of different-granularity evidence, Sun et al. (2021a) propose MuGER,[2] which performs multi-granularity evidence retrieval and answer reasoning.

Wang et al. (2022) conducts extensive experiments to prove that a coarse-grained retriever contributes less than a fine-grained retriever. Moreover, fine-grained methods, although giving an exact position of candidate cells, fail to illustrate why the selected cells are chosen, while our method is based on row and column selection probabilities. We thus further extend the fine-grained method by aligning questions with tables, letting our approach know which parts of questions are accounted for by which modalities. Intuitively, multi-hop questions in the text-table QA task usually contain two pieces of information from different modalities, tables and passages. Moreover, tables and passages are connected with evidence contained in tabular data. Our method implicitly decomposes the questions for different modalities to locate evidence and improve cell-selection accuracy.

As illustrated in Figure 1, an example from the HybridQA dataset shows how humans work on

[*] indicates equal contribution.

Findings of the Association for Computational Linguistics: ACL 2023, pages 6535–6549
July 9-14, 2023 ©2023 Association for Computational Linguistics


Original Questions
What is the middle name of the player with the second most National Football League career rushing yards ?
Q1: What is the middle name of [Ans of Q2]
Q2: the player with the second most National Football League career rushing yards
Walter Jerry Payton (July 25 , 1954 - November 1 , 1999) was an American professional football player who was a running back for the Chicago Bears of the National Football League (NFL) for thirteen seasons....

Rank	Player	Team (s) by season	Carries	Yards	Average
1	Emmitt Smith	Dallas Cowboys (1990 - 2002) Arizona Cardinals (2003 - 2004)	4,409	18,355	4.2
2	Walter Payton	Chicago Bears (1975 - 1987)	3,838	16,726	4.4
3	Frank Gore	San Francisco 49ers (2005 - 2014) Indianapolis Colts (2015 - 2017) Miami Dolphins (2018) Buffalo Bills (2019 -present)	3,548	15,347	4.3

Figure 1: Example from the HybridQA dataset. The top sentence is the original question, and words in different colors show different parts of questions required for reasoning in different modalities. the two headers in blue-dashed boxes are column names aligned with the given question. TACR first uses a method based on table-question-alignment to align the original question with table columns to help obtain golden table cells and then retrieve the final answer based on linked passages.

multi-hop and multi-modal QA tasks. The original question *"What is the middle name of the player with the second most National Football League career rushing yards ?"* can be divided into two parts, *"What is the middle name of"* and *"the player with the second most National Football League career rushing yards?"* for passages and tables, respectively. Such sub-questions are connected with the evidence of a cell (*"Walter Payton"*). For humans, we first locate who was the player in the second rank, which requires information from two columns: *"Rank"* and *"Player"*. After locating the cell, we can finally determine Walter Payton's middle name from the passage. Such reasoning process inspired us to develop TACR, a **T**able-**a**lignment-based **C**ell-selection and **R**easoning model, which incorporates a fine-grained evidence-retrieval module that utilizes table-question-alignment to learn which parts of the question are used for retrieving evidence from different modalities and reasoning towards answers.

To explicitly and correctly show the reasoning process in the text-table QA task, in the evidence retrieval stage, TACR first selects the golden cells and avoids redundant information in multi-granularity evidence that would lower the performance of the answer-reasoning module. The table-cell-selection module of TACR is designed to navigate the fine-grained evidence for the reader by fusing well-learned information from the table-question-alignment module. Compared with current fine-grained retrievers, the table-question-alignment module of TACR can help our model learn which parts of questions are used for reasoning in which modality, and which parts of tables contain candidate cells. Together with the alignment module, TACR preserves both high golden cell-selection accuracy and shows competitive performance on the HybridQA and WikiTableQuestions (WTQ) datasets, while providing improved explainability.

Our contributions are as follows: (1) TACR is the first model able to explicitly show its reasoning process in the passage-table QA task; (2) We jointly train the cell-selection and table-question alignment modules to improve golden cell selection performance and preserve the QA reader's performance; and (3) We conduct extensive experiments on the HybridQA and WTQ datasets to demonstrate the effectiveness of TACR.

2 Related Work

2.1 Table Question Answering

Table QA has gained much attention, as shown by benchmark datasets such as WikiTableQuestions (Pasupat and Liang, 2015b), WikiSQL (Zhong et al., 2018), SPIDER (Yu et al., 2018), and TABFACT (Chen et al., 2019). However, these datasets mainly focus on reasoning on tables and ignore important knowledge stored in the textual corpus. Consequently, QA covering both tabular and textual knowledge has gained increasing interest. Chen et al. (2020b) pioneered a passage-table QA benchmark, HybridQA, with Wikipedia tables linked to relevant free-form text passages (e.g., Wikipedia entity-definition pages). The OTT-QA (Chen et al., 2020a) benchmark extended HybridQA to the open domain setting, where a system needs to retrieve a relevant set of tables and passages first before trying to answer questions. Moreover, the links from the table and passage are not provided explicitly.

2.2 Table-Question Alignment

There are several table-question-alignment methods. Schema-linking-based methods, such as RAT-SQL (Wang et al., 2019), introduce a relation-aware transformer encoder to improve the joint encoding of a question and schema. Liu et al. (2022) propose a similarity learning-based question-schema-alignment method to obtain a semantic schema-linking graph and observed how the pre-trained language model (PLM) embeddings for the schema items are affected. Zhao and Yang (2022) use the same words that appear in both the natural language statement and the table as weak supervised key points and design an interaction network to explore the correlation between the representations of the natural language statements and tables.

2.3 Hybrid QA

Studies on hybrid QA usually retrieve different granularities of evidence from heterogeneous data to retrieve the final answer. Hybrider, proposed by Chen et al. (2020b), is a two-phase pipeline framework to retrieve gold table cells as evidence and input their values and linked passages into a QA model to extract the final answer. Sun et al. (2021b) proposes Dochopper, an end-to-end multi-hop retrieval model that directly concatenates rows with related textual evidence as its inputs. Pan et al. (2020) explores an unsupervised multi-hop QA model, called MQA-QG, which can generate human-like multi-hop questions by building a reasoning graph from heterogeneous data resources. Kumar et al. (2021) propose MITQA, which applies multiple-instance training objectives to retrieve coarse-grained evidence. On the contrary, Eisenschlos et al. (2021) introduce a transformer-based model with row- and column-wise attentions for fine-grained evidence retrieval, e.g., table cells. Wang et al. (2022) propose a unified retriever that tries to preserve the advantages and eliminates the disadvantages of different-granularity evidence retrieval methods.

TACR differs from the above models mainly in two aspects: (1) TACR focuses on providing an explicit reasoning process by aligning multi-hop questions to tables, so it learns which parts of multi-hop questions are accounted for by retrieving evidence from which modality; and (2) The table-question alignment can enhance the reasoning ability of the table cell selection module with the help of our generated hybrid alignment dataset. TACR shows competitive performance to that of other table QA models on the HybridQA and WTQ datasets on the basis of high row, column, and cell selection accuracy. To the best of our knowledge, no text-table QA system handles the challenge of explicitly showing its reasoning process and multi-hop question table alignment.

2.4 Table Cell Retrieval

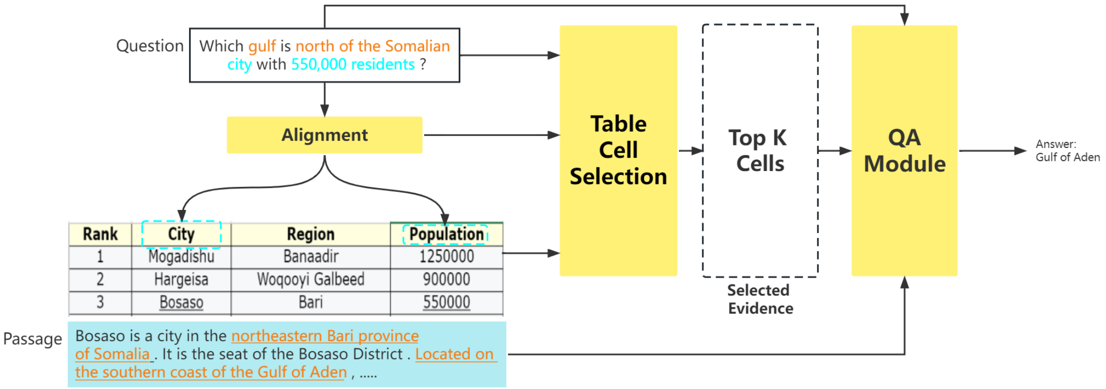
Jauhar et al. (2016) construct a multiple-choice table QA benchmark that includes over 9000 question-table pairs via crowd-sourcing and proposed a table-cell search model based on calculating all relevance scores between each cell and question. Such a model is reasonable and intuitive but time-consuming. TACR selects gold cells based on row and column selection. Suppose that a table contains n rows and m columns; the table cell search method must calculate $n * m$ scores for each cell, while TACR needs to calculates only $n + m$ scores for each row and column, and selects the gold cell in the row and column with the highest score. Sun et al. (2016) focus on extracting entities from questions and building a row graph and then mapping the question to the pair of cells in the same row of a table. However, some entities may not appear in both questions and table cells, e.g., an entity of the question in Figure 1 that should be extracted is *National Football League*, but it cannot be mapped into any cells.

3 Framework

As described in the previous section, both coarse- and fine-grained approaches fail to provide a reasoning process showing which parts of multi-hop questions map to which modality and evidence. Here we describe the details of TACR and its three main components: (1) data augmentation for training the table-question alignment module; (2) a multi-task learning module for table-question alignment and table-cell-selection; and (3) a text-based multi-hop QA module for retrieving answers. Figure 2 shows the overall architecture of TACR.

3.1 Task Definition

Given a question Q (a sequence of tokens) and N rows of table T together with linked passages P, where each table column has a header $h_{i=1}^{i=M}$ (M is



Which gulf is north of the Somalian city with 550,000 residents ?

Rank	City	Region	Population
1	Mogadishu	Banaadir	1250000
2	Hargeisa	Woqooyi Galbeed	900000
3	Bosaso	Bari	550000

Bosaso is a city in the northeastern Bari province of Somalia . It is the seat of the Bosaso District . Located on the southern coast of the Gulf of Aden , ...

Figure 2: TACR model architecture. From left to right, we first construct a hybrid alignment dataset to jointly train the table-question-alignment and table-cell-selection modules. We then concatenate filtered linked passages with selected top-k candidate cells as paragraphs and feed them into a text-based multi-hop QA module to retrieve answers.

the number of table headers), the task is to find a candidate cell $c_{i,j}$ that contains the answer α.

3.2 Data Construction

We generate multi-hop questions from tables and linked passages, as well as table-question alignment labels from questions and table columns for training the table-question-alignment module. However, such supervision information is not offered in the HybridQA dataset and other text-table QA datasets, which makes the alignment task difficult. We use an unsupervised text-table QA-generation method to generate questions as well as alignment labels.

Alignment Generation. We follow the settings of the MQA-QG method (Pan et al., 2020), using a pre-trained Google T5 (Raffel et al., 2019), fine-tuned on the SQuAD dataset (Rajpurkar et al., 2018), to generate multi-hop questions from tables and passages based on a bridge entity, a table cell that contains the bridge entity, and a linked passage that describes the bridge entity. The bridge entity is critical in reasoning because it connects the tables and passages, which are difficult to locate in the original HybridQA dataset.

Such bridge entity provides us with additional information to align table headers with generated questions based on the column containing golden cells and the column containing the bridge entity. We align the columns which contain bridge entities and answers to questions following two schema-linking alignment rules:

Name-based Linking. This rule refers to exact or partial occurrences of the column/table names in the question, such as the occurrences of "player" in the question in Figure 1. Textual matches are the most explicit evidence of table-question alignment and, as such, one might expect them to be directly beneficial to the table-question alignment module.

Value-based Linking. Table-question alignment also occurs when the question mentions any values that occur in the table and consequently participate in the table-cell selection, such as "the second most" in Figure 1. While it is common for examples to make the alignment explicit by mentioning the column name (e.g., "Rank"), many real-world questions do not (like in the example). Consequently, linking a value mentioned in the question to the corresponding column also requires background knowledge.

3.3 Passage Filtering

In this stage, we aim to filter out linked passages unrelated to a question, namely keeping almost noise-free passages for the following modules. Moreover, the total number of tokens in passages linked to table cells can be large, exceeding the maximum input sequence length of current LMs. Thus, we utilize Sentence-BERT (Reimers and Gurevych, 2019) to obtain question and passage embeddings and rank the top-k sentences based on their text similarities. We expand the cells with the filtered top k-related sentences to both fit in the max input length of language models and to preserve the useful information from passages. More details on this stage are provided in Appendix A.

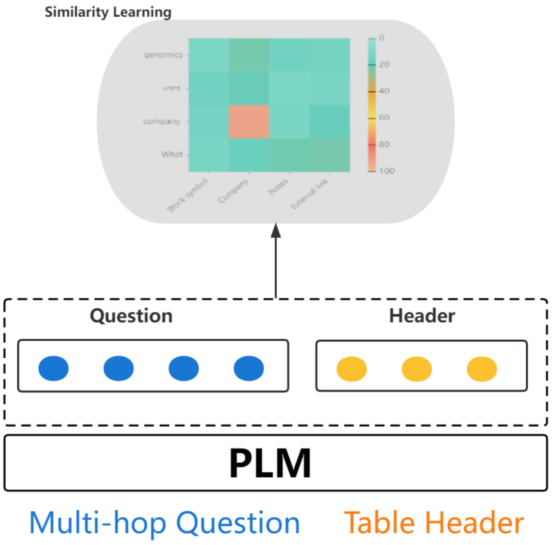


Figure 3: The table-question-alignment module of TACR. We treat the alignment objective as a similarity learning task.

3.4 Table Alignment & Cell Selection

In this stage, we jointly train a multi-task model with the objectives of selecting the expanded cell that contains the answer and table-question alignment to different modalities to enhance the previous objective. TACR accepts the full table as inputs and outputs the probabilities of selected cells based on the probabilities of row and column selection.

3.4.1 Table-Question Alignment

Given a natural language question $Q = \{q_1, ...q_{|Q|}\}$, a table consisting of several column headers $C = \{c_1....c_{|C|}\}$, and the corresponding table-question alignment labels $L = \{l_1, ...l_{|C|}\}$ where $l_i \in [0, 1]$ (0 meaning the column header is unrelated to the question Q and 1 meaning the column header is related to Q). The goal of our table-question alignment module is to learn the relevance between table-column headers and questions. Table-question relations aid TACR by aligning column references in the question to the corresponding table columns.

We first feed the questions and table columns into the pre-trained model and map them into hidden representations. The question and table-column headers can be denoted as $\{q_1, ...q_{|Q|}\}$ and $\{c_1....c_{|C|}\}$, respectively. Our goal is to induce a function $f(q_i, c_j)$ to capture the relevance of a question word q_i has on the representation of column header c_j. Figure 3 shows the structure of the alignment module.

Specifically, we use ALBERT (Lan et al., 2019) as the encoder to learn the representations of tables and column headers. Here we concatenate column headers as a pseudo sentence. The representations of the question (h_q) and the column headers sequence (h_c) are first computed independently. The relevance where each column header c_i is the target of the question is then given by using softmax. The respective equations are as follows:

$$h_q = \texttt{BERT(Q)}, \tag{1}$$

$$h_c = \texttt{BERT(C)}, \tag{2}$$

$$p(C_i \in C) = \texttt{softmax}(W(h_q * h_c) + b). \tag{3}$$

3.4.2 Table-Cell Selection

Inspired by the previous idea of modeling the attention on rows and columns (Eisenschlos et al., 2021), we design a cell-selection module based on row and column selection. The probabilities of each row and column are given and the cells with the top-k highest scores are returned as the candidate answers, or to aid in locating the relevant passage. However, unlike in MATE (Eisenschlos et al., 2021), we can derive probabilities of candidate cells from the probabilities of row and column.

We utilize the Row-Column-Intersection (RCI) model, designed for the single-hop table-QA task (Glass et al., 2021) (based on ALBERT (Lan et al., 2019)), as our backbone and decompose the table QA task into two subtasks: projection - corresponding to identifying columns; and selection - identifying rows. Every row and column identification is a binary sequence pair classification. We concatenate the question as the first sequence and the row or column as the second sequence. We feed concatenated two sequences, with standard separator tokens $[CLS]$ and $[SEP]$, as the input to the model. The representation of the final hidden state is sent to the linear layer, followed by a softmax to classify whether the column or row contains the answer or not. Each row and column is assigned a probability of containing the answer. This module finally outputs the top-k cells with the sum of row and column probabilities. Therefore, given a table T with N rows and M columns, we can obtain two sets of scores produced from the RCI model: $P_r = p_1, ...p_N$ for rows and $P_c = p_1, ...p_M$ for columns. We then calculate the overall probability score for each cell.

The final training loss is the summation of table-question-alignment loss, table-row-selection loss, and table-column-selection loss:

$$L = \texttt{L\_row} + \texttt{L\_column} \\ + \sigma \times \texttt{BCE}(pred\_headers, target\_headers), \quad (4)$$

where σ is a hyper-parameter to balance cell-selection training and table-question-alignment training. The details of choosing the best σ are provided in Appendix C.

3.5 Passage Question-Answering

Previous research often simply treat the answer-reasoning task as a span-extraction task, considered the first span matching the answer text as the gold span, and use that for training. Such consideration is incorrect because the answer text may appear in multiple passages, but only one of them is right. Therefore, using all text matches for training span extraction may introduce a large amount of training noise. As not all instances are the gold answer text that has relations with questions, after obtaining the top-k cells from the cell-selection module, we train the text-based QA module to predict the final answer that also takes into account the cell-selection scores.

Specifically, we select clean training instances where the gold answer text appears only once and train an initial QA model. In this stage, we use RoBERTa (Liu et al., 2019) as our backbone model. Other BERT variants, e.g., either SpanBERT (Joshi et al., 2019) or DeBERTa (He et al., 2020), could be also used in this module. Our goal is to obtain a span s in a given expanded table cell c with its filtered passage p and the input question q. We compute a span representation as follows:

$$h_{start} = \texttt{RoBERTa}_r(q, c)[\texttt{START}(s)], \quad (5)$$

$$h_{end} = \texttt{RoBERTa}_r(q, c)[\texttt{END}(s)], \quad (6)$$

$$S_{span}(q, p) = \texttt{MLP}([h\_start, h\_end]). \quad (7)$$

We also consider other cells in the same row as the retrieved candidate gold cells as the necessary context. We linearize and concatenate the row into a passage with the designed template: "The <column header> is <cell content>". We retrieve the top-k cells and thus have k samples. Since not all selected cells contain the gold answer text, we treat one sample as positive and the others as negative samples. For each data point, we generate k samples and match these with the answer text. Let $K = \{q_i, A_i, P_i^+, P_{i,1}^-, , P_{i,k-1}^-\}_{i=1}^k$ be the training data that consist of k instances, where k is the number of selected candidate cells. Each instance contains one question q_i, the gold answer text A_i, and one correct (positive) passage text P_i^+, along with $k-1$ wrong passages $P_{i,j}^-$. For positive samples, the answer is the text span of the passage, while for negative samples, the answers are -1.

Split	Train	Dev.	Test	Total
In-Passage	35215	2025	2045	39285
In-Table	26803	1349	1346	29498
Compute	664	92	72	864
Total	62682	3466	3463	69611

Table 1: Statistics of HybridQA dataset

4 Experiments

4.1 Datasets

HybridQA (Chen et al., 2020b) is the first large-scale multi-hop QA dataset that requires reasoning over hybrid knowledge, including tables and linked Wikipedia passages. The dataset contains 62,682 instances in the training set, 3,466 instances in the development set, and 3,463 instances in the test set.

WikiTableQuestions (Pasupat and Liang, 2015a), WTQ for short, consists of 22033 complex questions and 2108 semi-structured Wikipedia tables. The questions are designed by crowd-sourcing to contain a wide range of domains. The answers are derived from several operations such as table lookup, aggregation, superlatives, arithmetic operations, joins, and unions.

To verify the performance of TACR, we first conduct experiments on HybridQA (Chen et al., 2020b), a dataset of multi-hop question-answering over tabular and textual data. The basic statistics of HybridQA are listed in Table 1. The dataset contains three partitions: 'In-Table', where the answer derives from table cell values; 'In-Passage', where the answer exists in a linked passage; and 'Compute', where the answer should be computed by executing numerical operations. We mainly focus on the first two types. We also provide results over WTQ to illustrate TACR's capabilities in table-focused QA.

4.2 Baselines

MQA-QG, proposed by (Pan et al., 2020), is an unsupervised question-generation framework that generates multi-hop questions from tables and linked passages, and uses the generated questions to train an HQA model.

Table-Only (Chen et al., 2020b) only retrieves the tabular information to find an answer by parsing the question into a symbolic form and executing it.

Passage-Only (Chen et al., 2020b) only retrieves answers from the table-linked passages.

Hybrider (Chen et al., 2020b) addresses HQA using a two-stage pipeline framework to retrieve the gold table cell and extract an answer in its value or linked passages.

Dochopper (Sun et al., 2021b) first converts a table with its hyperlinked passages into a long document then concatenates column headers, cell text, and linked passages in each row of tables as a paragraph.

MATE (Eisenschlos et al., 2021) applies sparse attention to rows and columns in a table. To apply it to the HybridQA dataset, the authors propose a PointR module, which expands a cell using the description of its entities, selects the golden cells, then retrieves answers from them.

MITQA (Kumar et al., 2021) designs a multi-instance training method based on distant supervision to filter the noisy information from multiple answer spans.

4.3 Quantitative Analysis

We use exact match (EM) and F1 scores as evaluation metrics on the HybridQA dataset to compare the performance of TACR with that of previous baselines. As shown in Table 2, TACR outperforms most baselines and achieved competitive performance to state-of-the-art (SOTA) models (e.g., MITQA) in both EM and F1 scores over the HybridQA dataset. Table 3 reports the accuracy performance on WTQ. Though TACR is trained on a base model, it presents comparable accuracy to the large SOTA models and outperforms other base models. It is important to note that, besides both using much larger LMs than TACR (GPT-3 and BART-large respectively, versus RoBERTa-base), neither Binder nor Omnitab-large provide explainability. With the help of the table-question-alignment module, TACR boosts relative accuracy by $+18.5\%$ on the test set compared with RCI (Glass et al., 2021), which is also based on cell selection. This competitive performance is mainly based on the high cell selection along with table-question alignment. We further verified the effectiveness of the table-question-alignment module in an ablation study discussed in Section 4.5.

4.4 Qualitative Analysis

We compare the cell-selection accuracy of TACR and baseline models, as shown in Table 4. The high cell selection accuracy is based on the high row- and column-selection accuracies shown in Table 6. On the HybirdQA dataset, TACR shows SOTA performance and 0.4% higher than that of MATE (Eisenschlos et al., 2021) in the top 3 cell-selection accuracies due to its 89.3% row-selection accuracy and 98.3% column-selection accuracy, as shown in Table 6. Moreover, by achieving soft question decomposition (i.e., showing which parts of questions are connected to reasoning in the different modalities), TACR both improves the explainability of its results and provides valuable signals for future improvements.

4.5 Ablation Study

To evaluate the impact of the table-question-alignment module, we conduct an ablation study, shown in Table 5. We test DeBERTa-base, ALBERT-base, and RoBERTa-base models as TACR backbones for generality. Different top-k results show that the alignment module consistently significantly improves results; with the best model based on ALBERT improving cell-selection accuracy by 2.5, 3.9, and 4.3% in top 1, 3, and 5 cell selection respectively; and mean reciprocal rank (MRR) improving by 3.7%. The results indicate that the table-question-alignment module has an important role in the table-question-reasoning stage to select the most related cells that support the answer to the question.

4.6 Case Study

To illustrate TACR can successfully learn which parts of tables contain golden cells and which parts of questions are required for reasoning in the different modalities, we choose two examples from the HybridQA development set. Appendix B includes Figures 4 and 5 showing their word relevances heatmap and analysis.

The question in Case 1 is *"Who is the athlete in a city located on the Mississippi River ?"*. The concatenated table headers string for the corresponding table is *"Year Score Athlete Place"*. The table-question-alignment module helps TACR learn that header terms *"Athlete"* and *"Place"* have higher relevance to the question than the headers of other columns, thus guiding cell-selection. Figure 4 shows its relevance heatmap. TACR again learns

Model	Dev.						Test					
	In-Table		In-Passage		Total		In-Table		In-Passage		Total	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Table-Only	14.7	19.1	2.4	4.5	8.4	12.1	14.2	18.8	2.6	4.7	8.3	11.7
Passage-Only	9.2	13.5	26.1	32.4	19.5	25.1	8.9	13.8	25.5	32.0	19.1	25.0
Hybrider (τ=0.8)	54.3	61.4	39.1	45.7	44.0	50.7	56.2	63.3	37.5	44.4	43.8	50.6
PointR + SAT	66.5	71.8	60.3	69.2	61.2	68.7	64.6	70.1	59.6	68.5	60.1	67.4
PointR + TAPAS	68.1	73.9	62.9	72.0	63.3	70.8	67.8	73.2	62.0	70.9	62.7	70.0
PointR + TABLEETC	36.0	42.4	37.8	45.3	36.1	42.9	35.8	40.7	38.8	45.7	36.6	42.6
PointR + LINFORMER	65.5	71.1	59.4	69.0	60.8	68.4	66.1	71.7	58.9	67.8	60.2	67.6
PointR + MATE	68.6	74.2	62.8	71.9	63.4	71.0	66.9	72.3	62.8	71.9	62.8	70.2
MQA-QG (unsupervised)	–	–	–	–	–	–	36.2	40.6	19.8	25.0	25.7	30.5
Dochopper	–	–	–	–	47.7	55.0	–	–	–	–	46.3	53.3
MITQA	68.1	73.3	66.7	75.6	65.5	72.7	68.5	74.4	64.3	73.3	64.3	71.9
MuGER^2	58.2	66.1	52.9	64.6	53.7	63.6	56.7	64.0	52.3	63.9	52.8	62.5
TACR (ours)	66.7	70.3	63.4	72.5	64.5	71.6	64.1	69.6	65.4	70.7	66.2	70.2
Human											88.2	93.5

Table 2: EM and F1 results of models on the HybridQA dataset. In-Table and In-Passage subsets refer to the location of answers.

Model	Dev Acc	Test Acc
TAPEX-Large (Liu et al., 2021)	57.0	57.5
Binder (Cheng et al., 2022)	65.0	64.6
OmniTab-Large (Jiang et al., 2022)	62.5	63.3
TAPAS_base (pre-trained on SQA) (Herzig et al., 2020)	-	48.8
UnifiedSKG (Xie et al., 2022)	50.7	49.3
TaBERT_base (Yin et al., 2020)	51.6	51.4
RCI (Glass et al., 2021)	45.3	41.7
TACR_RoBERTa-base (ours)	58.9	60.2

Table 3: Execution-accuracy results of models on WTQ

Model	Hits@1	Hits@3	Hits@5
TABLEETC (Ainslie et al., 2020)	51.1	72.0	78.9
LINFORMER (Wang et al., 2020)	77.1	86.5	90.0
MATE (Eisenschlos et al., 2021)	80.1	86.2	90.5
TACR (ours)	**83.3**	**87.8**	**91.2**

Table 4: Comparison of cell-retrieval results on HybridQA dataset (dev set)

which parts of the question account for retrieving evidence in tables.

The question in Case 2 is *"What is the middle name of the player with the second most National Football League career rushing yards ?"*. The concatenated table headers string for it is *"Rank Player Team(s) by season Carries Yards Average"*. The table-question-alignment module helps TACR learn that the sub-question *"the player with the second most National Football League career rushing yards"* has a higher relevance to the table headers than that of other parts of the original question, thus guiding modality relevance. Figure 5 shows its relevance heatmap.

4.7 Error Analysis

To further analyze TACR, we also calculate statistics for error cases in the model predictions. The error statistics are based on the development set of HybridQA. Through the cell-selection accuracy statistics in Table 4, we find there are 347 tables whose cells are incorrectly selected.

To better understand the advantages and disadvantages of table-question alignment-based cell selection, we manually sample and examined 20 such error cases (i.e., where TACR does not provide the correct answer in the correct row, column, and cell position). Out of the 20 samples, we find that five error cases (25%) are due to requiring numerical reasoning operations that cross several cells (which is out of scope for TACR). The majority of errors, 13 of the remaining incorrect cases, are in the same column with a correct answer while in the wrong row. Only one case is from a different row but the same column with the correct answer and only one incorrect case is in a completely different row and column to the correct answer.

5 Conclusion

This paper presents TACR, a **T**able question **A**lignment-based **c**ell selection and **R**easoning model for hybrid text and table QA, evaluated on the HybridQA and WikiTableQuestions datasets. When answering questions given retrieved table

Model	MRR	Hits@1	Hits@3	Hits@5
TACR-DeBERT_base w/o alignment	78.9	74.9	79.4	83.7
TACR-Roberta_base w/o alignment	80.7	74.3	82.6	84.4
TACR-ALBERT_base w/o alignment	80.1	77.1	82.8	85.4
TACR-DeBERTa_base w/ alignment	82.4	78.3	83.4	86.2
TACR-RoBERTa_base w/ alignment	82.5	76.5	85.5	88.9
TACR-ALBERT_base w/ alignment	**83.8**	**79.6**	**86.7**	**89.7**

Table 5: Ablation study of table-question-alignment module impact. Experiment results of cell-retrieval on HybridDQA (dev set) show the effectiveness of this module in the table-cell-selection stage.

Model	HybridQA		WTQ	
	Row	Col	Row	Col
top 1				
TACR_DeBERTa_base	85.1	95.3	53.2	93.9
TACR_ALBERT_base	86.7	96.1	56.8	94.4
TACR_RoBERTa_base	86.0	96.2	52.3	94.7
top 3				
TACR_DeBERTa_base	86.2	96.2	57.6	94.2
TACR_ALBERT_base	88.3	97.1	62.4	95.1
TACR_RoBERTa_base	87.9	97.3	59.3	94.9
top 5				
TACR_DeBERTa_base	87.5	97.8	59.1	94.8
TACR_ALBERT_base	89.9	98.3	68.1	95.4
TACR_RoBERTa_base	89.3	98.4	64.5	95.2

Table 6: Performance of TACR with different backbone models. Top-k rows and columns selection accuracies on HybridQA and WTQ datasets, where k=1, 3, 5. Results demonstrate the effectiveness of TACR.

cells and passages, TACR attempts to align multi-hop questions to different modalities for correct evidence retrieval. To enhance the QA module with better table cell-selection and table-question-alignment ability, we construct a hybrid alignment dataset generated from the HybridQA dataset. TACR shows state-of-the-art performance in retrieving intermediate gold table cells and competitive performance on the HybridQA and WikiTableQuestions datasets, while improving output explainability.

6 Limitations

In this paper, we focus on the hybrid QA task, where the answers to most questions can be extracted from cell values in tables and linked passages using a reading comprehension model. Although TACR performs well in cell selection, one of its limitations is that it lacks numerical reasoning ability across different cells, such as counting and comparing. To enable TACR to answer numerical questions, we will further develop its numerical reasoning capabilities in future work. Another limitation of TACR is that it shows a strong ability in column selection while performing relatively worse in row selection. For future work, we plan to try to improve its row-selection accuracy.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

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, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2020a. Open question answering over tables and text. *ArXiv*, abs/2010.10439.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020b. HybridQA: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, R.K. Nadkarni, Yushi Hu, Caiming Xiong, Dragomir R. Radev, Marilyn Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Binding language models in symbolic languages. *ArXiv*, abs/2210.02875.

Julian Martin Eisenschlos, Maharshi Gor, Thomas Müller, and William W. Cohen. 2021. MATE: Multi-view attention for table transformer efficiency. In *Conference on Empirical Methods in Natural Language Processing*.

Michael R. Glass, Mustafa Canim, A. Gliozzo, Saneem A. Chemmengath, Rishav Chakravarti, Avirup Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *North American Chapter of the Association for Computational Linguistics*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with disentangled attention. *ArXiv*, abs/2006.03654.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. *ArXiv*, abs/2004.02349.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tables as semi-structured knowledge for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 474–483, Berlin, Germany. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *NAACL*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-Guang Lou. 2021. TAPEX: Table pre-training via learning a neural SQL executor. *ArXiv*, abs/2107.07653.

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. *ArXiv*, abs/1907.11692.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2020. Unsupervised multi-hop question answering by question generation. In *North American Chapter of the Association for Computational Linguistics*.

Panupong Pasupat and Percy Liang. 2015a. 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, Beijing, China. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015b. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Annual Meeting of the Association for Computational Linguistics*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2021a. End-to-end multihop retrieval for compositional question answering over long documents. *ArXiv*, abs/2106.00200.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2021b. Iterative hierarchical attention for answering complex questions over long documents.

Huan Sun, Hao Ma, Xiaodong He, Wen tau Yih, Yu Su, and Xifeng Yan. 2016. Table cell search for question answering. *Proceedings of the 25th International Conference on World Wide Web*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2019. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In *Annual Meeting of the Association for Computational Linguistics*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768.

Yingyao Wang, Junwei Bao, Chaoqun Duan, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022. MuGER^2: Multi-granularity evidence retrieval and reasoning for hybrid question answering. *ArXiv*, abs/2210.10350.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 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 R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unified-SKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *ArXiv*, abs/2201.05966.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *ArXiv*, abs/2005.08314.

Tao Yu, Rui Zhang, Kai-Chou Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Z Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Conference on Empirical Methods in Natural Language Processing*.

Guangzhen Zhao and Peng Yang. 2022. Table-based fact verification with self-labeled keypoint alignment. In *International Conference on Computational Linguistics*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Seq2sql: Generating structured queries from natural language using reinforcement learning. *ArXiv*, abs/1709.00103.

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. *arXiv preprint arXiv:2105.07624*.

A Passage Filtering

Passage filtering plays an important role in cell selection as well as answer extraction. Pre-trained language models such as BERT, RoBERTa, and LLMs have the limitation of max input sequence length. Passage filtering ensures that it is unlikely to lose information relevant to the questions, while fitting model input limits. We used the well-trained DistilBert-based model to obtain question and passage embeddings to rank and filter relevant passages.[1]

B Alignment Analysis

Here we provide example heatmaps showing the relevance of questions and table headers. The relevance is in the [0,1] range, where the higher relevance between words from questions and column headers is shown in the warmer colors and vice versa. Figure 4 shows that the column headers "athlete" and "place" have more relevance to the question, which helps TACR identify which columns contain potential gold cells. In Figure 5, the words "player with second most national football league" from the question have more relevance to columns, which help TACR learn which parts of the question better use to retrieve gold cells.

C Implementation Details of Cell Selection and Alignment

TACR is implemented using Pytorch version 1.13 and the Huggingface transformers (Wolf et al.,

[1] https://huggingface.co/sebastian-hofstaetter/distilbert-dot-tas_b-b256-msmarco

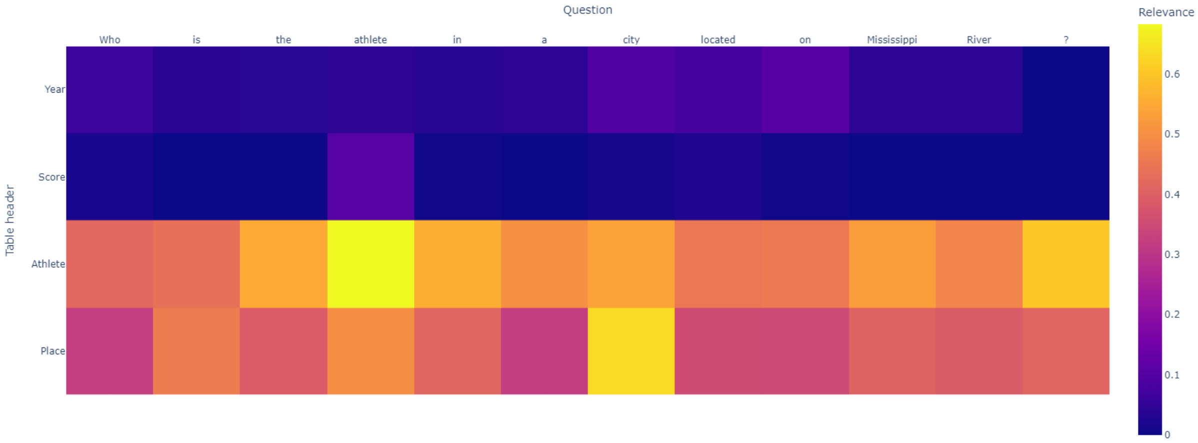


Figure 4: Heatmap of question and table-header relevance - Case 1

2020) library. We trained TACR using two NVIDIA A6000 GPUs. The cell selection and table–question-alignment modules are trained for four epochs and we selected the best model based on the dev fold performance. AdamW is used as optimizer algorithm with a learning rate of 5×10-5 and a batch size of 32. We set the per-GPU train batch size to 16 while training the span-based QA model. Final answers are evaluated using EM and F1 scores. We also automatically iterated through increments of 0.1 in the range [0, 1] to select the best σ to balance the multi-task training.

Hyper-parameter Details: We tune hyper-parameters based on the loss on the development set and use the following range of values for selecting the best hyper-parameters:

- Batch size: [8, 16, 32, 64]
- Learning rate: [1e-3, 1e-4, 1e-5, 1e-6, 3e-3, 3e-4, 3e-5, 3e-6, 5e-3, 5e-4, 5e-5, 5e-6]
- σ : [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]

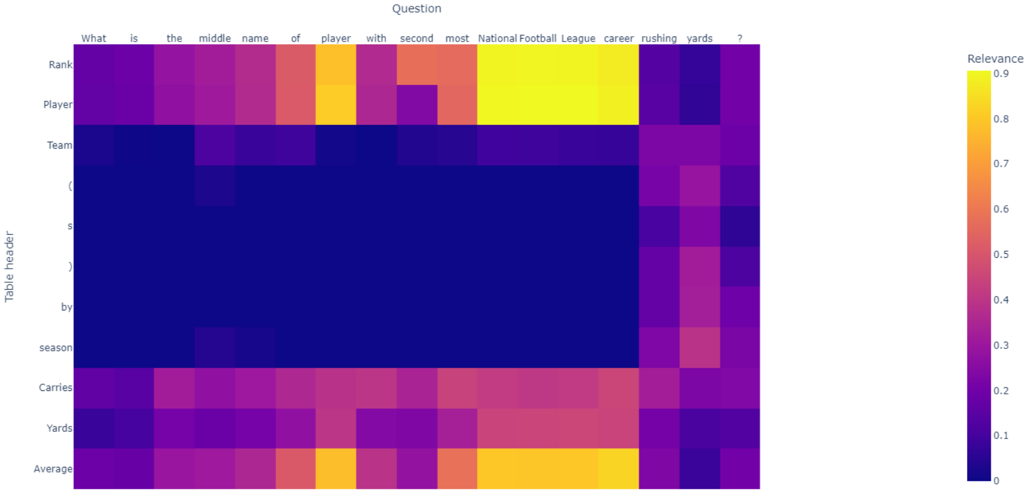


Figure 5: Heatmap of question and table header relevance - Case 2

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 6

☑ A2. Did you discuss any potential risks of your work?
Left blank.

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

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

B ☑ Did you use or create scientific artifacts?

section 3

☑ B1. Did you cite the creators of artifacts you used?
references

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
appendix

☑ 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)?
section 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?
Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
section 2

☑ 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.
section 4

C ☑ Did you run computational experiments?

section 4

☑ 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?
No response.

□ 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?
No response.

□ 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.)?
No response.

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.