# Dual Reader-Parser on Hybrid Textual and Tabular Evidence for Open Domain Question Answering

**Alexander Hanbo Li, Patrick Ng, Peng Xu, Henghui Zhu,**
**Zhiguo Wang, Bing Xiang**
AWS AI Labs, Amazon
{hanboli, patricng, pengx, henghui, zhiguow, bxiang}@amazon.com

## Abstract

The current state-of-the-art generative models for open-domain question answering (ODQA) have focused on generating direct answers from unstructured textual information. However, a large amount of world's knowledge is stored in structured databases, and need to be accessed using query languages such as SQL. Furthermore, query languages can answer questions that require complex reasoning, as well as offering full explainability. In this paper, we propose a hybrid framework that takes both textual and tabular evidence as input and generates either direct answers or SQL queries depending on which form could better answer the question. The generated SQL queries can then be executed on the associated databases to obtain the final answers. To the best of our knowledge, this is the first paper that applies Text2SQL to ODQA tasks. Empirically, we demonstrate that on several ODQA datasets, the hybrid methods consistently outperforms the baseline models that only take homogeneous input by a large margin. Specifically we achieve state-of-the-art performance on OpenSQuAD dataset using a T5-*base* model. In a detailed analysis, we demonstrate that the being able to generate structural SQL queries can always bring gains, especially for those questions that requires complex reasoning.

## 1 Introduction

Open-domain question answering (ODQA) is a task to answer factoid questions without a pre-specified domain. Recently, generative models (Roberts et al., 2020; Lewis et al., 2020; Min et al., 2020; Izacard and Grave, 2020) have achieved the state-of-the-art performance on many ODQA tasks. These approaches all share the common pipeline where the first stage is retrieving evidence from the free-form text in Wikipedia. However, a large amount of world's knowledge is not stored as plain text but in structured databases, and need to be accessed using query languages such as SQL. Furthermore, query languages can answer questions that require complex reasoning, as well as offering full explainability. In practice, an ideal ODQA model should be able to retrieve evidence from both unstructured textual and structured tabular information sources, as some questions are better answered by tabular evidence from databases. For example, the current state-of-the-art ODQA models struggle on questions that involve aggregation operations such as counting or averaging.

One line of research on accessing databases, although not open domain, is translating natural language questions into SQL queries (Zhong et al., 2017; Xu et al., 2017; Yu et al., 2018c; Guo et al., 2019; Wang et al., 2018a, 2020; Yu et al., 2018a; Guo and Gao, 2019; Choi et al., 2020). These methods all rely on knowing the associated table for each question in advance, and hence are not trivially applicable to the open-domain setting, where the relevant evidence might come from millions of tables.

In this paper, we provide a solution to the aforementioned problem by empowering the current generative ODQA models with the Text2SQL ability. More specifically, we propose a **dual reader-parser (DUREPA)** framework that can take both textual and tabular data as input, and generate either direct answers or SQL queries based on the context[1]. If the model chooses to generate a SQL query, we can then execute the query on the corresponding database to get the final answer. Overall, our framework consists of three stages: retrieval, joint ranking and dual reading-parsing. First we retrieve supporting candidates of both textual and tabular types, followed by a *joint* reranker that predicts how relevant each supporting candidate is to

[1]Our code is available at https://github.com/AlexanderYogurt/Hybrid-Open-QA

the question, and finally we use a fusion-in-decoder model (Izacard and Grave, 2020) for our reader-parser, which takes all the reranked candidates in addition to the question to generate direct answers or SQL queries.

To evaluate the effectiveness of our DUREPA, we construct a hybrid dataset that combines SQuAD (Rajpurkar et al., 2016) and WikiSQL (Zhong et al., 2017) questions. We also conduct experiments on NaturalQuestions (NQ) (Kwiatkowski et al., 2019) and OTT-QA (Chen et al., 2020a) to evaluate DuRePa performance. As textual and tabular open-domain knowledge, we used textual and tabular data from Wikipedia via Wikidumps (from Dec. 21, 2016) and Wikitables (Bhagavatula et al., 2015). We study the model performance on different kinds of questions, where some of them only need one supporting evidence type while others need both textual and tabular evidence. On all question types, DUREPA performs significantly better than baseline models that were trained on a single evidence type. We also demonstrate that DUREPA can generate human-interpretable SQLs that answer questions requiring complex reasoning, such as calculations and superlatives.

Our highlighted contributions are as follows:

- We propose a multi-modal framework that incorporates hybrid knowledge sources with the Text2SQL ability for ODQA tasks. To the best of our knowledge, this is the first work that investigates Text2SQL in the ODQA setting.
- We propose a simple but effective generative approach that takes both textual and tabular evidence and generates either direct answers or SQL queries, automatically determined by the context. With that, we achieve the state-of-the-art performance on OpenSQuAD using a T5-*base* model.
- We conduct comprehensive experiments to demonstrate the benefits of Text2SQL for ODQA tasks. We show that interpretable SQL generation can effectively answer questions that require complex reasoning in the ODQA setting.

## 2 Related Work

**Open Domain Question Answering** ODQA has been extensively studied recently including extractive models (Chen et al., 2017; Clark and Gardner, 2018; Wang et al., 2019; Min et al., 2019; Yang et al., 2019) that predict spans from evidence passages, and generative models (Raffel et al., 2020; Roberts et al., 2020; Min et al., 2020; Lewis et al., 2020; Izacard and Grave, 2020) that directly generate the answers. Wang et al. (2018b,c); Nogueira and Cho (2019) proposed to rerank the retrieved passages to get higher top-n recall.

**Table Parsing** Text2SQL is a task to translate natural questions to executable SQL queries. Brad et al. (2017) proposed SENLIDB dataset which only contains 29 tables and lacks annotation in their training set. Recently, with datasets like WikiSQL (Zhong et al., 2017), Spider (Yu et al., 2018c) and CoSQL (Yu et al., 2019) being introduced, many works have shown promising progress on these dataset (Yu et al., 2018b; He et al., 2019; Hwang et al., 2019; Min et al., 2019; Wang et al., 2020; Choi et al., 2020; Guo et al., 2019; Lyu et al., 2020; Zhang et al., 2019; Zhong et al., 2020; Shi et al., 2020). Another line of work proposes to reason over tables without generating logical forms (Neelakantan et al., 2015; Lu et al., 2016; Herzig et al., 2020; Yin et al., 2020). However, they are all closed-domain and each question is given the associated table.

**Hybrid QA** Chen et al. (2020a) also proposed an open-domain QA problem with textual and tabular evidence. Unlike our problem, they generate an answer directly from the tabular evidence instead of generating an SQL query. In addition, they assume some contextual information about table is available during retrieval stage (e.g. their fusion-retriever is pretrained using hyperlinks between tables and paragraphs), whereas we don't use any link information between tables and passages. Moreover, Chen et al. (2020b) proposed a closed-domain hybrid QA dataset where each table is linked to on average 44 passages. Different from ours, their purpose is to study multi-hop reasoning over both forms of information, and each question is still given the associated table.

## 3 Method

In this section, we describe our method for hybrid open-domain question answering. It mainly consists of three components: (1) a retrieval system; (2) a joint reranker and (3) a dual Seq2Seq model that uses fusion-in-decoder (Izacard and Grave, 2020) to generate direct answer or SQL query.

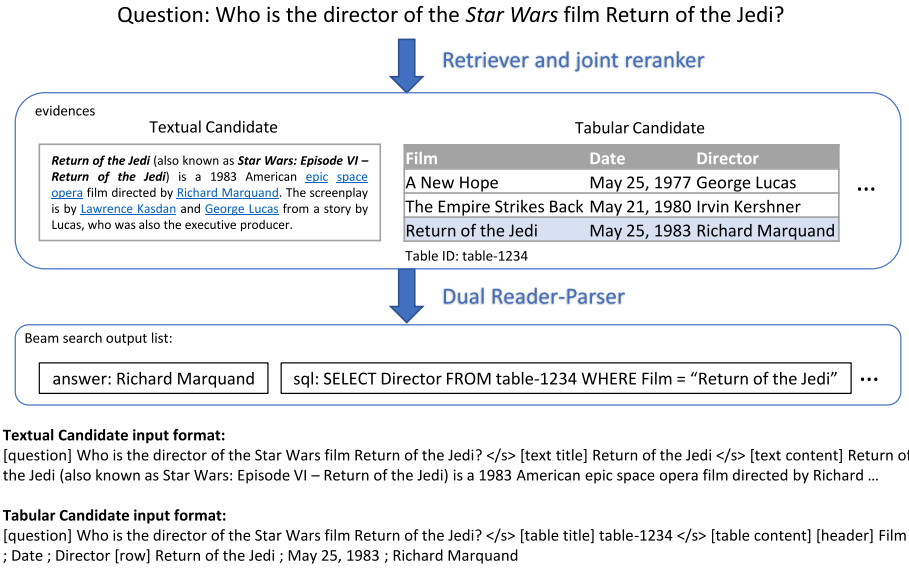


Figure 1: The pipeline of our proposed hybrid model. The candidates are retrieved from knowledge source such as Wikipedia including both paragraphs and tables. Then a generative Seq2Seq model reads the question and all the candidates, and produces $k$ outputs using beam search. Each output can be either a final answer or an intermediate SQL query. The types and order of the outputs are automatically determined by the model itself.

### 3.1 Retrieval

For the hybrid open-domain setting, we build two separate search indices – one for textual input and another for tabular input. For paragraphs, we split them into passages of at most 100 words. For tables, we flattened each table into passages by concatenating cell values along each row. If the flattened table exceeds 100 words, we split it into a separate passage, respecting row boundaries. The column headers are concatenated to each tabular passage. Some examples of flattened tables are given in the Appendix A.1.

Given a natural language question, the retrieval system retrieves 100 textual and 100 tabular passages as the support candidates from the textual and tabular indices, respectively, using BM25 (Robertson et al., 1995) ranking function.

### 3.2 Joint Reranking

The purpose of our reranking model is to produce a score $s_i$ of how relevant a candidate (either an unstructured passage or table) is to a question. Specifically, the reranker input is the concatenation of `question`, a retrieved `candidate-content`, and its corresponding `title` if available[2], separated by special tokens shown in Figure 1. The candidate content can be either the unstructured text or flattened table. We use BERT$_{base}$ model in this paper. Following Nogueira and Cho (2019), we finetune the BERT (Devlin et al., 2019) model using the following loss:

$$L = -\sum_{i \in \mathcal{I}_{pos}} \log(s_i) - \sum_{i \in \mathcal{I}_{neg}} \log(1 - s_i). \quad (1)$$

The $\mathcal{I}_{pos}$ is sampled from all relevant BM25 candidates, and the set $\mathcal{I}_{neg}$ is sampled from all non-relevant BM25 candidates. Different from Nogueira and Cho (2019), during training, for each question, we sample 64 candidates including one positive candidate and 63 negative candidates, that is, $|\mathcal{I}_{pos}| = 1$ and $|\mathcal{I}_{neg}| = 63$. If none of the 200 candidates is relevant, we skip the question. During inference, we use the hybrid reranker to assign a score to each of the 200 candidates, and choose the top 50 candidates as the input to the next module – the reader-parser model. For the top 50 candidates, we choose them from the joint pool of all candidates, according to the scores assigned by the reranker.

### 3.3 Dual Reading-Parsing

Our dual reader-parser model is based on the fusion-in-decoder (FID) proposed in Izacard and Grave (2020), and is initialized using the pretrained T5 (Raffel et al., 2020) model. The overall pipeline of the reader-parser is shown in Figure 1. Each

[2]Wikipedia passages have page titles, and tables have table titles.

retrieved candidate is represented by its title and content, in the following formats:

**Textual Candidate** We represent each textual candidate as the concatenation of the passage title and content, appended by special tokens `[text title]` and `[text content]` respectively.

**Tabular Candidate** In order to represent a structured table as a passage, we first flatten each table into the following format: each flattened table starts with the complete header names and then followed by rows. Figure 1 presents an example for this conversion.

Finally, a tabular candidate is the concatenation of the table title and content flattened as a passage, appended by special tokens `[table title]` and `[table content]` respectively. We use the table ID as the title so that it can be copied to the generated SQL queries by the model.

**Prefix of the Targets** During training, we also add special tokens `answer:` or `sql:` to a targeted sentence depending on whether it is a plain text or a SQL query. For those questions that have both textual answer and SQL query annotations (for example, WikiSQL questions), we create two training examples for each question. During inference, the generated outputs will also contain these two special prefixes, indicating which output type the model has generated.

**Dual Reader-Parser** Our generative Seq2Seq model has reader-parser duality. During inference, the model reads the question and all the candidates, and produces $k$ outputs using beam search. Each output can be either a final answer or an intermediate SQL query. Depending on the context, the types and order of the outputs are automatically determined by the model itself. All the generated SQL queries will then be executed to produce the final answers. In this paper, we fix $k = 3$ and always generate three outputs for each question.

## 4 Experiments

In this section, we report the performance of the proposed method on several hybrid open-domain QA datasets.

### 4.1 Datasets

In this section, we describe all the datasets we use in our experiments. First we summarize the statistics of the open-domain QA datasets we use in Table 1.

| Dataset | #Train&Dev | #Test |
|---|---|---|
| OpenSQuAD | 82,599 | 5,000 |
| OpenNQ | 87,925 | 3,610 |
| OTT-QA | 41,469 | 2,214 |
| OpenWikiSQL | 52,026 | 7,764 |
| Mix-SQuWiki | 134,625 | 12,764 |
| WikiSQL-both | – | 3,029 |

Table 1: Statistics of Datasets

**OpenSQuAD** is an open-domain QA dataset constructed from the original SQuAD-v1.1 (Rajpurkar et al., 2016), which was designed for the reading comprehension task, consisting of 100,000+ questions posed by annotators on a set of Wikipedia articles, where the answer to each question is a span from the corresponding paragraph.

**OpenNQ** is an open-domain QA datasets constructed from the NaturalQuestions (Kwiatkowski et al., 2019), which was desgined for the end-to-end question answering task. The questions were from real google search queries and the answers were from Wikipedia articles annotated by humans.

**OTT-QA** (Chen et al., 2020a) is a large-scale open table-and-text question answering dataset for evaluating open QA over both tabular and textual data. The questions were constructed through "decontextualization" from HybridQA (Chen et al., 2020b) with additional 2200 new questions mainly used in dev/test set. OTT-QA also provides its own corpus which contains over 5 million passages and around 400k tables.

**OpenWikiSQL** is an open-domain Text2SQL QA dataset constructed from the original WikiSQL (Zhong et al., 2017). WikiSQL is a dataset of 80,654 annotated questions and SQL queries distributed across 24,241 tables from Wikipedia.

**Mix-SQuWiki** is the union of OpenSQuAD and OpenWikiSQL datasets.

**WikiSQL-both** is a subset of OpenWikiSQL evaluation data that contains the questions that can be answered by both textual and tabular evidences. The purpose of this dataset is to study when both types of evidence are possible to answer a question, whether the hybrid model can still choose the better one. We select these questions in a weakly-supervised way by only keeping a question if the

| Model | Evidence Corpus Type | OpenSQuAD | OpenNQ | OTT-QA | OpenWikiSQL |
|---|---|---|---|---|---|
| FiD(T5-*base*) | Text-only | 53.4 | 48.2 | - | - |
| FiD(T5-*large*) | Text-only | **56.7** | 51.4 | - | - |
| IR+CR | Text+Table w/o SQL | - | - | 14.4 | - |
| FR+CR | Text+Table w/o SQL | - | - | **28.1**[3] | - |
| Unified Model | Text+NQ Table w/o SQL | - | **54.6**[4] | - | - |
| *Ours* | | | | | |
| FID+ | Text-only | 56.4 | 45.2 | 14.5 | 13.9 |
| FID+ | Table-only w/o SQL | 2.5 | 14.3 | 4.1 | 30.3 |
| DUREPA | Table-only with SQL | 2.7 | 14.8 | 4.7 | 40.2 |
| FID+ | Text+Table w/o SQL | 56.4 | 46.7 | 15.0 | 30.9 |
| DUREPA | Text+Table with SQL | **57.0** | **48.0** | **15.8** | **42.6** |

Table 2: Comparison to the state-of-the-art on open-domain QA datasets. The numbers reported are in EM metric. FiD(T5-*base* & T5-*large*) is reported from (Izacard and Grave, 2020), IR+CR (Iterative Retrieval+Cross-block Reader) and FR+CR (Fusion Retrieval+Cross-block Reader) are from (Chen et al., 2020a), Unified Model is from (Oguz et al., 2020). Comparing DUREPA with FID+ , we observe that having the ability to generate structural queries is always beneficial even for questions with mostly extractive answers like SQuAD and NQ.

groundtruth answer is contained in both textual and tabular BM25 candidates. For example in Figure 1, the answer "Richard Marquand" can be found in both types of passages. We filter out some trivial cases where the answer shows up in more than half of the candidates. [5]

**Wikipedia Passages and Tables** For the textual evidences, we process the Wikipedia 2016 dump and split the articles into overlapping passages of 100 words following (Wang et al., 2019). To create the tabular evidences, we combine 1.6M Wikipedia tables (Bhagavatula et al., 2015) and all the 24,241 WikiSQL tables, and flatten and split each table into passages not exceeding 100 words, in the same format mentioned in the previous section. We use these two collections as the evidence sources for all the QA datasets except for OTT-QA, where we use its own textual and tabular collections.

### 4.2 Implementation Details

**Retriever and Reranker.** We conduct BM25 retrieval using Elasticsearch 7.7 [6] with the default settings. And we use a BERT reranker initialized with pretrained BERT-*base-uncased* model.

**Dual Reader and Parser with fusion-in-decoder.** Similar to (Izacard and Grave, 2020), we initialize the fusion-in-decoders with the pretrained T5 model (Raffel et al., 2020). We only explore T5-*base* model in this paper, which has 220M parameters.

For both reranker and FiD models, we use Adam optimizer (Kingma and Ba, 2014) with a maximum learning rate of $10^{-4}$ and a dropout rate of 10%. The learning rate linearly warms up to $10^{-4}$ and then linearly anneals to zero. We train models for 10k gradient steps with a batch size of 32, and save a checkpoint every 1k steps. For the FiD model, when there are multiple answers for one question, we randomly sample one answer from the list. For the FiD model, during inference, we generate 3 answers for each question using beam search with beam size 3.

### 4.3 Main Results

We present the end-to-end results on the open-domain QA task comparing with the baseline methods as show in Table 2.

We build models with 5 different settings based on the source evidence modality as well as the format of model prediction. Specifically, we consider single modality settings with only textual evidence or tabular evidence and the hybrid setting with both textual and tabular evidence available. For tabular evidence, the models either predict direct answer text or generate structure SQL queries. Note we also consider a baseline model, FID+ , a FiD model that only generates direct answer text, but can make use of both textual and tabular evidence.

[5] For example, some numerical number like "1" is a very common substring and shows up in most of the candidates.

[6] https://www.elastic.co/

[3] Chen et al. (2020a) uses a fusion-retriever to retrieved table-passages blocks as evidences. To construct the fusion blocks, they train a GPT-2 model using extra hyperlink information to link table cell to passages. In contrast, we do not use any hyperlink information.

[4] Oguz et al. (2020) uses tables provided by NQ training data (less than 500k in total), whereas we use all the tables extracted from Wikipedia dumps (around 1.6M in total).

| Index | BM25 **textual** | Reranker **textual** | BM25 **tabular** | Reranker **tabular** | Reranker **hybrid** |
|---|---|---|---|---|---|
| R@1 | 34.40 | 69.76 | 1.60 | 10.16 | **69.92** |
| R@10 | 59.38 | 80.30 | 6.34 | 18.88 | **80.90** |
| R@25 | 65.92 | 81.64 | 8.84 | 21.20 | **82.42** |
| R@50 | 72.16 | 82.50 | 12.36 | 22.62 | **83.26** |
| R@100 | 76.50 | 83.44 | 15.04 | 23.72 | **84.10** |

Table 3: Recalls on top-$k$ textual, tabular or the hybrid candidates for SQuAD questions. The recalls on hybrid inputs are almost the same as or even better than the best recalls on individual textual or tabular inputs, meaning that the reranker is able to jointly rank both types of candidates and provide better evidences to the next component – the reader-parser.

First, in the single modality setting, we observe that for OpenSQuAD, OpenNQ and OTT-QA datasets, textual QA model is performing significantly better than tabular QA models, while for OpenWikiSQL, it is the opposite. This is expected due to the nature of the construction process of those datasets. In the hybrid setting, the hybrid models outperform single modality models consistently across all these datasets. This indicates hybrid models are more robust and flexible when dealing with questions of various types in practice.

Comparing DUREPA with FID+ , we observe that having the ability to generate structural queries is always beneficial even for extractive questions like SQuAD and NQ. And for WikiSQL-type questions, the gain of SQL generation is significant.

On OpenSQuAD dataset, our DUREPA model using hybrid evidences achieves a new state-of-the-art EM score of 57.0. It is worth noting that the previous best score was attained by FiD using T5-*large* model, while our model is using T5-*base*, which has much fewer parameters. On NQ dataset, FID+ with text-only evidences has lower EM score compared with FiD-base, despite having the same underlying model and inputs. We suspect that this is because (1) we truncate all passages into at most 150 word pieces while in FiD paper they keep 250 word pieces, so the actual input (top-100 passages) to our FiD model is much less than that in the FiD paper; and (2) we use BM25 to retrieve the initial pool of candidates instead of trained embedding-based neural retrieval model(Karpukhin et al., 2020; Izacard and Grave, 2020). Nevertheless, the DUREPA model with hybrid evidences still improve the EM by 2.8 points compared to FID+ using only text inputs. On OTT-QA questions, our full model also outperforms the IR+CR baseline by 1.4 points. The FR+CR model is using a different setting where they use hyperlinks between tables and passages to train the fusion-retriever (FR), so the result is not directly comparable to ours. We provide more analysis on OTT-QA in the Appendix. On OpenWikiSQL dataset, enabling SQL generation brings more than 10 points improvement on the EM scores. This is because many questions therein require complex reasoning like COUNT, AVERAGE or SUM on the table evidences. We provide more in-depth analysis in Section 5.2 including some complex reasoning examples in Table 7.

## 5 Analysis

### 5.1 Retrieval and Reranking Performance

In this section, we investigate the performance of the BM25 retriever and the BERT reranker using top-k recalls as our evaluation metric.

During both training and inference, for each question, the textual and tabular passages are reranked jointly using a single reranker. On the Mix-SQuWiki dataset, we report the reranking results on SQuAD questions in Table 3. The result on WikiSQL questions is in Table 9 in Appendix. To provide better insights on the reranker's performance, we show the top-$k$ recalls on textual, tabular and hybrid evidences separately.

From Table 3, on both textual and tabular candidates, recall@25 of the ranker is even higher than recall@100 of the BM25 retriever. This suggest that during inference, instead of providing 100 BM25 candidates to the fusion-in-decoder (FiD), only 25 reranked candidates would suffice.

In Table 9 and 10 in Appendix, we observe similar trend with top-25 recalls comparable to top-100 recalls on both WikiSQL and NQ questions. Finally, across all datasets, the recalls on hybrid inputs are almost the same as or even better than the best recalls on individual textual or tabular inputs, meaning that the reranker is able to *jointly* rank both types of candidates and provide better

evidences to the next component – the dual reader-parser.

### 5.2 Performance of the Reader-Parser

In this section, we discuss the performance of the dual reader-parser on different kinds of questions.

**SQL prediction helps with complex reasoning.** In Table 4, we compare the top-1 EM execution accuracy of DUREPA and FID+ on OpenWikiSQL. If DUREPA generated a SQL, we execute the SQL to obtain its answer prediction. If the ground-truth answer is a list (e.g., What are the names of Simpsons episodes aired in 2008?), we use set-equivalence to evaluate accuracy. DUREPA outperforms FID+ on the test set in most of the settings. We also compare their performance under a breakdown of different categories based on the ground-truth SQL query. DUREPA achieved close to 3x and 5x improvements on WikiSQL questions that have superlative (MAX/MIN) and calculation (SUM/AVG) operations, respectively. For COUNT queries, FID+ often predicted either 0 or 1. Thus, these results support our hypothesis that the SQL generation helps in complex reasoning and explainability for tabular question answering.

| | DUREPA | FID+ | #Test |
|---|---|---|---|
| All | **47.1** | 29.3 | 7764 |
| COUNT $\in \{0,1\}$ | 78.0 | **82.9** | 770 |
| COUNT $\geq 2$ | **44.4** | 0.0 | 9 |
| MIN/MAX | **26.6** | 9.3 | 654 |
| SUM/AVG | **22.6** | 4.7 | 314 |
| Comparison ($<$ or $>$) | **45.8** | 32.0 | 939 |
| AND-condition | **53.0** | 31.8 | 2045 |
| Answer is a list | **34.3** | 0.0 | 160 |
| Direct answers | **78.7** | 75.6 | 933 |

Table 4: Comparison of DUREPA and FID+ on OpenWikiSQL dataset. We compare their accuracy under a breakdown of different categories based on the ground-truth SQL query. "Direct answers" stands for the questions that DUREPA predicts direct answers. DUREPA significantly outperforms on questions that require complex reasoning such as superlatives and calculations.

**Using hybrid evidence types leads to better performance.** Shown in Table 5 is the model performance on the Mix-SQuWiki questions. As the baseline models, if we only use a single evidence type, the best top-1 EM is 34.0, achieved by the model FID+ using only textual candidates. However, if we use both evidence types, the hybrid model DUREPA attains a significantly better top-1 EM of 47.9, which implies that including both textual and tabular evidences leads a better model performance on Mix-SQuWiki. Furthermore, we observe that the model DUREPA has a better top-1 EM compared to FID+, suggesting that the answers for some of these questions need to be obtained by executing SQL queries instead of generated directly. In Table 7, we samples some questions on which the model DUREPA predicts the correct answers but the model FID+ fails.

**What if the questions can be answered by both textual and tabular evidences?** Table 6 shows the model performance on WikiSQL-both dataset. Recall that all these questions in the dataset can be answered by both type of evidence. First of all, the DUREPA model using tabular evidences behaves better than the FID+ model using textual evidences. This implies on WikiSQL questions, using tabular information leads to better answers. Next, when using only one type of evidence, both DUREPA and FID+ models behave significantly worse than their hybrid counterparts. This indicates that the hybrid model can again figure out which evidence type should be used to provide the correct final answer.

## 6 Discussion and Future Work

Our experiments consistently show that the proposed framework DUREPA brings significant improvement on answering questions using hybrid types of evidence. Especially on the questions that can be answered by both supporting evidence types, our multi-modal method still shows clear advantage over models using single-type knowledge, implying that our approach could figure out the most relevant evidence to answer a question. We also demonstrate that the dual reader-parser is essential to the good performance of DUREPA; the ability of generating both direct answers and structural SQL queries help DUREPA perform much better than FID+ and other baselines on questions that require complex reasoning like counting or averaging.

We believe that our methods can be improved in two aspects. First, our general framework Fig. 1 can be improved by a better retrieval system. For example, instead of using BM25, we can use more powerful neural retrieval models (Karpukhin et al., 2020). On the hybrid evidence, one can also use an entity linking module to link the entities between the tables and passages (Chen et al., 2020a) and utilize the structure information for better multi-

| Model | Evidence Corpus Type | % of SQL Answers | Acc of SQL Answers (%) | % of Direct Answers | Acc of Direct Answers (%) | EM (Overall) |
|---|---|---|---|---|---|---|
| FiD+ | Text-only | 0.0 | - | 100.0 | 34.0 | 34.0 |
| FiD+ | Table-only w/o SQL | 0.0 | - | 100.0 | 19.3 | 19.3 |
| DuRePa | Table-only with SQL | 53.9 | 42.5 | 46.1 | 8.4 | 26.8 |
| FiD+ | Text+Table w/o SQL | 0.0 | - | 100.0 | 40.0 | 40.0 |
| DuRePa | Text+Table with SQL | 33.5 | 44.1 | 66.5 | 49.8 | 47.9 |

Table 5: Detailed results on Mix-SQuWiki dataset under various settings.

| Model | Evidence Corpus Type | % of SQL Answers | Acc of SQL Answers (%) | % of Direct Answers | Acc of Direct Answers (%) | EM (Overall) |
|---|---|---|---|---|---|---|
| FiD+ | Text-only | 0.0 | - | 100.0 | 38.7 | 38.7 |
| FiD+ | Table-only w/o SQL | 0.0 | - | 100.0 | 38.4 | 38.4 |
| DuRePa | Table-only with SQL | 38.6 | 30.4 | 61.4 | 57.2 | 46.8 |
| FiD+ | Text+Table w/o SQL | 0.0 | - | 100.0 | 43.2 | 43.2 |
| DuRePa | Text+Table with SQL | 39.8 | 35.5 | 60.2 | 64.0 | 53.6 |

Table 6: Model Performance on WikiSQL-both dataset. The models are trained on Mix-SQuWiki training data.

| | |
|---|---|
| ***Question:*** | Which party won in the election in voting district Kentucky 5? |
| ***Groundtruth:*** | ['democratic'] |
| ***Top-1 generation by* DuRePa:** | `sql: SELECT Party FROM table_1-1342218-17 WHERE District = "Kentucky 5"` |
| ***Execution result:*** | ['democratic'] |
| ***Top-1 generation by* DuRePa–** | answer: republican |
| ***Question:*** | Which Condition has an unaffected Partial thromboplastin time, Platelet count, and a Prothrombin time? |
| ***Groundtruth:*** | ['aspirin', 'uremia', "glanzmann's thrombasthenia"] |
| ***Top-1 generation by* DuRePa:** | `sql: SELECT Condition FROM table_1-14006-1 WHERE Partial thromboplastin time = "Unaffected" AND Platelet count = "Unaffected" AND Prothrombin time = "Unaffected"` |
| ***Execution result:*** | ['uremia', "glanzmann's thrombasthenia", 'aspirin'] |
| ***Top-1 generation by* DuRePa–** | answer: vitamin k deficiency or warfarin |
| ***Analysis:*** | Answer is a list of medical conditions |
| ***Question:*** | How many Wins have Goals against smaller than 30, and Goals for larger than 25, and Draws larger than 5? |
| ***Groundtruth:*** | ['3'] |
| ***Top-1 generation by* DuRePa:** | `sql: SELECT COUNT(Wins) FROM table_2-18017970-2 WHERE Goals against < 30 AND Goals for > 25 AND Draws > 5` |
| ***Execution result:*** | [3] |
| ***Top-1 generation by* DuRePa–** | answer: 0 |
| ***Analysis:*** | COUNT operation |
| ***Question:*** | What is the highest Rd that Tom Sneva had the pole position in? |
| ***Groundtruth:*** | ['7'] |
| ***Top-1 generation by* DuRePa:** | `sql: SELECT MAX(Rd) FROM table_1-10706961-2 WHERE Pole Position = "Tom Sneva"` |
| ***Execution result:*** | [7] |
| ***Top-1 generation by* DuRePa–** | answer: 2.0 |
| ***Analysis:*** | MAX operation |
| ***Question:*** | Name the average ERP W and call sign of w237br |
| ***Groundtruth:*** | [110] |
| ***Top-1 generation by* DuRePa:** | `sql: SELECT AVG(ERP W) FROM table_2-14208614-1 WHERE Call sign = "w237br"` |
| ***Execution result:*** | [110] |
| ***Top-1 generation by* DuRePa–** | answer: 1.0 |
| ***Analysis:*** | AVG calculation |

Table 7: Examples of the SQuWiki and OpenWikiSQL questions that are answered correctly by model DuRePa but incorrectly by model FiD+.

hop reasoning. Second, as we have demonstrated, having the ability of generating structural SQL queries is a very powerful and necessary feature for answering questions that require complex rea-

soning. Given the limited Text2SQL data and the difficulty of obtaining such SQL supervision, two interesting future work include (1) getting SQL annotations more efficiently and (2) adapting weakly-supervised approaches like discrete EM (Min et al., 2019) for model training.

## References


Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. Tabel: entity linking in web tables. In *International Semantic Web Conference*, pages 425–441. Springer.

Florin Brad, Radu Iacob, Ionel Hosu, and Traian Rebedea. 2017. Dataset for a neural natural language interface for databases (nnlidb). *arXiv preprint arXiv:1707.03172*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1870–1879. Association for Computational Linguistics (ACL).

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2020a. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1026–1036.

DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2020. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases. *arXiv preprint arXiv:2004.03125*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855.

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.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-sql in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535.

Tong Guo and Huilin Gao. 2019. Content enhanced bert-based text-to-sql generation. *arXiv preprint arXiv:1910.07179*.

Pengcheng He, Yi Mao, Kaushik Chakrabarti, and Weizhu Chen. 2019. X-sql: reinforce schema representation with context. *arXiv preprint arXiv:1908.08113*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Zhengdong Lu, Hang Li, and Ben Kao. 2016. Neural enquirer: learning to query tables in natural language. *IEEE Data Eng. Bull.*, 39(3):63–73.

Qin Lyu, Kaushik Chakrabarti, Shobhit Hathi, Souvik Kundu, Jianwen Zhang, and Zheng Chen. 2020. Hybrid ranking network for text-to-sql. *arXiv preprint arXiv:2008.04759*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard em approach for weakly supervised question answering. 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 2844–2857.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

Arvind Neelakantan, Quoc V Le, and Ilya Sutskever. 2015. Neural programmer: Inducing latent programs with gradient descent. *arXiv preprint arXiv:1511.04834*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. Unified open-domain question answering with structured and unstructured knowledge. *arXiv preprint arXiv:2012.14610*.

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. *Journal of Machine Learning Research*, 21:1–67.

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

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Peng Shi, Patrick Ng, Zhiguo Wang, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Cicero Nogueira dos Santos, and Bing Xiang. 2020. Learning contextual representations for semantic parsing with generation-augmented pre-training. *arXiv preprint arXiv:2012.10309*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578.

Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Oleksandr Polozov, and Rishabh Singh. 2018a. Robust text-to-sql generation with execution-guided decoding. *arXiv preprint arXiv:1807.03100*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018b. R 3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018c. Evidence aggregation for answer re-ranking in open-domain question answering. In *International Conference on Learning Representations*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. 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 5881–5885.

Xiaojun Xu, Chang Liu, and Dawn Song. 2017. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426.

Tao Yu, Zifan Li, Zilin Zhang, Rui Zhang, and Dragomir Radev. 2018a. Typesql: Knowledge-based type-aware neural text-to-sql generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 588–594.

Tao Yu, Michihiro Yasunaga, Kai Yang, Rui Zhang, Dongxu Wang, Zifan Li, and Dragomir Radev. 2018b. Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1653–1663.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. 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 1962–1979.

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

Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. Editing-based sql query generation for cross-domain context-dependent questions. 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 5341–5352.

Victor Zhong, Mike Lewis, Sida I Wang, and Luke Zettlemoyer. 2020. Grounded adaptation for zero-shot executable semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6869–6882.

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