AI Open 4 (2023) 1–12

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

A survey on complex factual question answering

Lingxi Zhang [a], Jing Zhang [a,*], Xirui Ke [a], Haoyang Li [a], Xinmei Huang [a], Zhonghui Shao [a], Shulin Cao [b], Xin Lv [b]

[a] Information School, Renmin University of China, Beijing, China
[b] Tsinghua University, Beijing, China

ARTICLE INFO

Keywords:
Question answering
Complex question
Factual question
Knowledge base question answering
Text2SQL
Document-based question answering
Table question answering
Multi-source question answering

ABSTRACT

Answering complex factual questions has drawn a lot of attention. Researchers leverage various data sources to support complex QA, such as unstructured texts, structured knowledge graphs and relational databases, semi-structured web tables, or even hybrid data sources. However, although the ideas behind these approaches show similarity to some extent, there is not yet a consistent strategy to deal with various data sources. In this survey, we carefully examine how complex factual question answering has evolved across various data sources. We list the similarities among these approaches and group them into the analysis–extend–reason framework, despite the various question types and data sources that they focus on. We also address future directions for difficult factual question answering as well as the relevant benchmarks.

1. Introduction

Question answering (QA) tasks attempt to seek the answers for a given natural language question from accessible data sources. Such data source often contains large amount of information and can be webpages, wiki corpus, knowledge bases (KBs), or even relational databases. Compared with information retrieval (IR), instead of retrieving the relevant context according to the given question, QA directly locates or reasons the answers, which can help users access their demanding information more efficiently.

With the advances of deep learning, existing researches have fully explored the simple QA problem which only requires one-hop reasoning on a single piece of evidence. For instance, "Who created Batman?" is a simple question, since it can be easily answered by a sentence "BATMAN is created by Bob Kane" or a triplet (Bob Kane, Create, BATMAN) from KBs. Machines can already beat human beings (Lan et al., 2019; Zhang et al., 2020, 2021c) or obtain approximate 100% accuracy (He et al., 2021a; Huang et al., 2021b; Shi et al., 2021a) on such QA benchmarks, such as MetaQA (Zhang et al., 2018) for knowledge base question answering and SQuAD (Rajpurkar et al., 2018) for closed-domain document question answering. Beside simple questions, complex questions are also frequently asked by users and play a key role in knowledge acquisition and analysis. Answering complex questions requires model to handle functional operators between evidences, such as join, numerical computing, intersection, and union, which results in multi-hop, constrained, numerical, and set logical questions that remain to be challenges. Therefore, much attention has been drawn and researchers design various QA models for complex QA on different data sources.

For unstructured data sources such as text, a mainstream approach (Zhang et al., 2021b,d; Tu et al., 2020) is a retriever–reader framework that retrieves relevant documents as candidate evidence and then reasons over the document candidates to extract or generate the text spans as answers. For structured data sources such as KBs, a similar retriever–reasoner framework (He et al., 2021b; Zhao et al., 2022; Jin et al., 2021) is also proposed, which retrieves the relevant subgraphs as candidate evidence and then reasons over the subgraphs to extract the entities as answers. In addition, a more promising idea for structured data sources is the semantic parsing (SP)-based framework (Kapanipathi et al., 2021; Purkayastha et al., 2022; Ye et al., 2022; Cao et al., 2022b) that attempts to translate the given question into an executable logical expression. Such expressions could be executed against the structured data sources to obtain entities or the aggregation result as answers. Moreover, structured relational databases (Xie et al., 2022; Scholak et al., 2021; Shaw et al., 2021) and semi-structured web tables (Herzig et al., 2021b, 2020; Eisenschlos et al., 2021) are also commonly investigated to answer complex questions. Although these methods are devised for different data sources, they show similar patterns in the architecture and could fall into a unified framework,

* Corresponding author.
E-mail addresses: zhanglingxi@ruc.edu.cn (L. Zhang), zhang-jing@ruc.edu.cn (J. Zhang), kexirui@ruc.edu.cn (X. Ke), lihaoyang.cs@ruc.edu.cn (H. Li), huangxinmei.spring@gmail.com (X. Huang), shaozhonghui@ruc.edu.cn (Z. Shao), caosl19@mails.tsinghua.edu.cn (S. Cao), lv-x18@mails.tsinghua.edu.cn (X. Lv).

https://doi.org/10.1016/j.aiopen.2022.12.003
Received 25 November 2022; Accepted 5 December 2022
Available online 23 December 2022

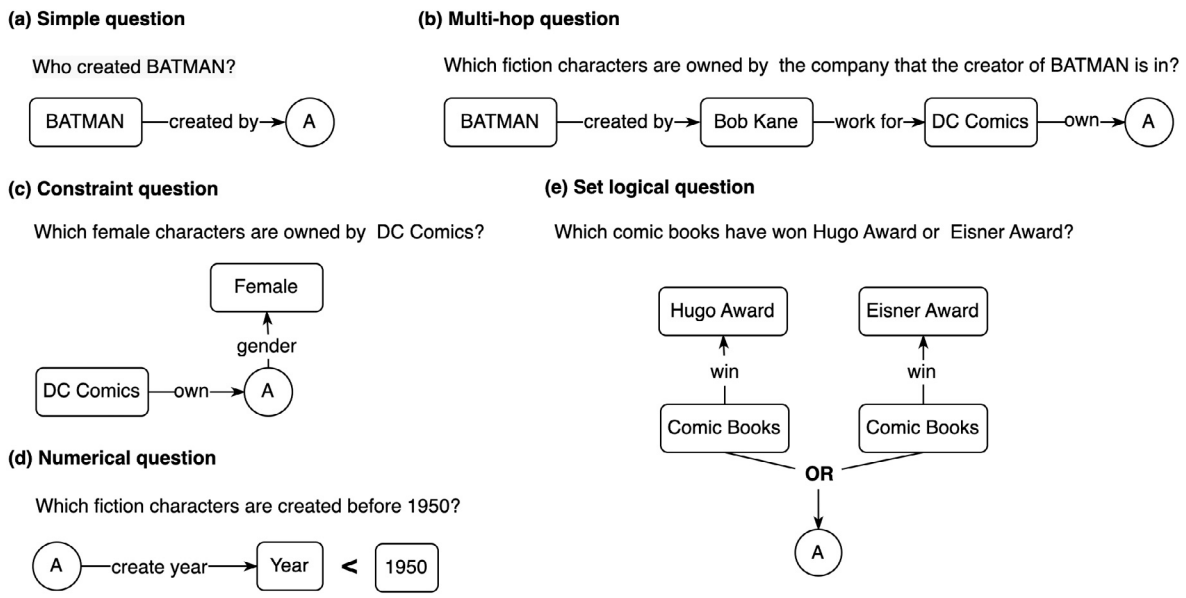

Fig. 1. Illustration of question types. For each question type, we provide an example question and its corresponding query graph, where the rectangles represent evidence and the circles represent answers.

analysis–extend–reason (Cf. Fig. 2). More specifically, the analysis module first understands the question and outputs an implicitly or explicitly reformulated question (*i.e.*, question representation or question skeleton). Then, according to the given question information, extend module retrieves or encodes relevant evidence from the data sources. Finally, the reason module performs reasoning over above relevant evidence to obtain the final answers.

Recently, there are also related surveys to summarize existing QA approaches. For example, Abbasiantaeb and Momtazi (2021), Jin et al. (2022), and Wu et al. (2019) review QA models in different data sources: documents, tables, and KBs respectively. However, they do not pay much attention to how these existing models handle complex questions. Lan et al. (2021), on the other hand, focus on reviewing complex QA task and category methods by the challenges existing methods manage to solve, but are limited to QA on KBs. Pandya and Bhatt (2021) discuss both complex questions and various data sources, but there is no abstract and overall summary of the substantial methods on various data source types. Both Zhang et al. (2021a) and Mavi et al. (2022) attempt to sum up different QA methods into a unified framework, but they only target a single data source KB or a single complex question type multi-hop. In fact, as we observed, all QA models share significant similarities, despite the differences in question types and data sources. Thus, our survey compiles QA methods for solving complex questions from various data sources into a unified framework.

The remaining sections of this survey are organized as follows. Section 2 formally introduces three mainstream natural language data sources for QA tasks, the categorization of question types and the definition of complex factual questions. Section 3 provides a unified framework for all recent approaches in complex QA, followed by reviews of structured QA, including KBQA and Text2SQl, unstructured QA, semi-structured QA, and hybrid QA. Section 4 summarizes the common datasets and evaluation metrics used in the above QA tasks listed above. Finally, Section 5 explores possible future directions in complex factual QA.

2. Background knowledge and problem definition

Question answering (QA) seeks to obtain an answer to a natural language question by retrieving and reasoning over provided data sources . Given a question q, a QA model is required to find the set of answers A_q based on the data source D. This survey develops the QA categorization by the structure of data sources and the type of questions. As stated in the introduction, the focus of our survey is on complex factual question answering using natural language sources. In this section, We will provide a detailed definition of these concepts.

2.1. Data source

We investigate the task of question answering on various natural language data sources such as text and database. To be clear, image and video are not covered because their patterns significantly differ from natural language. We classify natural language sources into three types based on how data is organized in the source: structured, unstructured, and semi-structured.

Structured Data Source. Data in a structured data source is organized in a unified and well-defined format, such as a tuple or table, and is easily accessible. Knowledge bases (KBs), a type of classical structural data source, are formally denoted as $S_{KB} = \{E \times R \times E\}$, where E is the entity set and R is the relation set. The knowledge in KB is organized in triplets (e_h, r, e_t), where $e_h/e_t \in E$ is the head/tail entity, and $r \in R$ is the relation that connects them. To be noted, numerical numbers or text strings are stored as literals in KBs, which is a special type of entity. Relational databases (DBs) are another common type of structural data sources. DB is made up of relational tables with multiple columns and rows and is formally denoted as $S_{DB} = \{T, C\}$, where T denotes tables, and C denotes columns. Textual names are assigned to both tables and columns and each cell in tables is in text format. Some columns serve as primary keys for unique indexing, and some serve as foreign keys, referring to columns in other tables. We define schema as all atomic knowledge involved in a structured data source, such as entities and relations in KBs or columns in DBs.

Unstructured Data Source. Unstructured data, along with structured data, makes up the majority of web data information. The set of documents S_D is referred to as the unstructured data source. Each $d \in S_D$ is a text-heavy document in pure text format that contains knowledge facts or numerical data.

Semi-structured Data Source. Aside from relatively structured tables in relational databases, a large corpus of tables is semi-structured, with an emphasis on web tables and spreadsheet tables, are semi-structured. The schema of such web tables is structured but is not formulated

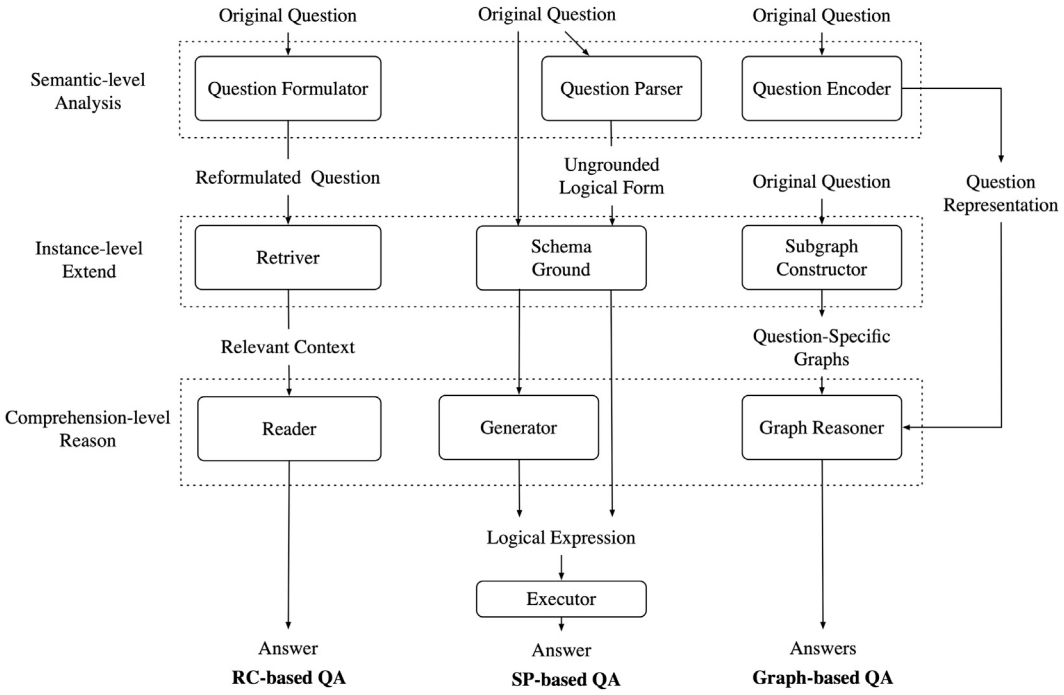

Fig. 2. Illustration of the Analysis–Retrieve–Reason framework for Question Answering. The analysis module first understands the question and then creates an implicitly or explicitly reformulated question (emphasis on question representation or question skeleton). Then, based the information provided in the question, the extend module retrieves or encodes relevant evidence from the data sources. Finally, the reason module utilizes reasoning over the prior relevant evidence to achieve the final answers.

or classified in a consistent manner. They are frequently massive, dispersed, and surrounded by rich text. We refer to $S_T = \{T_{web}\}$ as a semi-structured data source. S_T, like S_{DB}, contains multiple tables, but unlike S_{DB}, there are no foreign keys connecting the tables.

2.2. Complex factual question answering

We can divide questions into close-ended and open-ended. The former are often factual questions and frequently begin with "What", "Where" and "Which", leading to objective answers that can be evaluated accurately. In contrast, the latter frequently begins with "How" and is answered in a descriptive text with no specific facts as answers. Our survey is primarily concerned with answering factual questions.

A factual question can be further categorized into simple and complex questions based on the difficulty of obtaining the answers from the data source, given the question and the corresponding data source. The former only requires one-hop reasoning with a join-based operation on a single piece of evidence, such as a triple from KBs or a sentence from the documents. The latter, on the other hand, requires multi-hop reasoning or other complex operations in addition to the join-based operations such as numerical computing, intersection and union. Fig. 1 shows the corresponding examples. If a question requires multiple support evidence but can only be answered using join-based operations, it is referred to as a multi-hop question if the evidence is joined one by one and can be formulated as a direct straight chain (Fig. 1(b)), otherwise, it is referred to as a constrained question (Fig. 1(c)). In addition, we refer to a question that requires numerical computing (*e.g.* comparison and superlative) as a numerical question (Fig. 1(d)). Furthermore, we call a question involving intersection or union a set logical question (Fig. 1(e)). To be noted, a complex questions can belong to more than one category, for example, a complex question can contain both numerical computing and constraint property.

3. Overview of methods

We divide QA methods into three categories: reading comprehension (RC)-based method, semantic parsing (SP)-based method, and graph-based method. As the answer, RC-based methods extract a text span from the question-relevant text. To obtain the answer, SP-based methods map the given question to a logical expression and then execute it against a structured data source. Based on its encoded embedding, graph-based methods construct a question-relevant subgraph from the entire data source and predict whether one node is the answer. Regardless of the implemented details, we cast them in a unified analysis–extend–reason framework as illustrated in Fig. 2:

- **The semantic-level analysis** module attempts to convert the original input question into an intermediate form in order to facilitate the subsequent evidence gathering and answer reasoning. In particular, in RC-based QA, it could be a question rewriter that generates more appropriate questions for subsequent document rereiever. It could be a question parser in SP-based QA to extract the ungrounded logical form from the question, such as the AMR tree or the SPARQL skeleton. It could be a question encoder in Graph-based QA to represent the semantics of the question.
- **The instance-level extend** module aims to interact with the source data through the question. Given the question information, the module explicitly or implicitly collects the question-relevant evidence from the source data. In particular, in RC-based QA, it explicitly retrieves question-relevant documents, in SP-based QA, it links question-relevant schema, and in Graph-based QA, it constructs the question-relevant subgraph.
- **The comprehension-level reason** module performs reasoning over the input relevant evidence to obtain the final answers. In particular, in RC-based QA, it could be a text reader that extracts text span as answers. It contains an executor in SP-based QA that executes the logical expressions to obtain answers. The executor may be equipped with a previous generator that decodes the final executable logical expressions. A schema-grounded question or a question with auxiliary schema could be fed into the generator. The reason module in graph-based QA could be a GNN-based reasoner that predicts the answer nodes based on the learned representation of the question and the nodes.

3.1. Structured question answering

3.1.1. KBQA

Large-scale knowledge bases, such as Wikidata (Vrandečić and Krötzsch, 2014) and Freebase (Bollacker et al., 2008), are emerging and have supported a lot of natural language processing tasks. Among them, knowledge base question answering (KBQA) has been a recent surge of research interest, providing users with an easy and friendly way to seek factual knowledge. Existing KBQA systems can be divided into two categories: semantic parsing (SP)-based methods and graph-based methods. They both conform the proposed analysis–extend–reason framework. The former is adaptable to dealing with any types of complex questions, whereas the latter has some ability to deal with the multi-hop, constrained, and set logical questions, but struggles with numerical questions. The two KBQA methods in this unified framework are explained below.

SP-based methods aim to convert the question into an executable logical expression that can be directly executed against the KB to obtain the answers, which fall under the category of symbolic reasoning. As a result, they are adaptable in dealing with a wide range of complex questions. With the analysis–extend–reason framework, SP-based methods typically first parse the question into some intermediate ungrounded logic forms in the semantic-level analysis stage (Nie et al., 2022). Then in the next instance-level extending stage, they need to ground the logical form to the underlying KB explicitly or implicitly. Finally, in the comprehension-level reasoning stage, they derive the final logical expressions and apply them to get the answers. The SPARQL (Pérez et al., 2009), S-expression (Gu et al., 2021), and KoPL (Cao et al., 2022a) are all examples of logical expressions. To parse out such logical expressions, some researchers elaborately design the query parser and directly obtain the final logical expression after the schema ground, while others leverage neural-based generation models to generate the final logical expression.

Query Parser and Schema Ground. To bridge the gap between the natural language question and the logical expression, many works elaborately design intermediate logical forms rather than directly parsing to the final logical expressions. Some methods select the syntactic skeleton of the question as intermediate logical form. For example, NSQA (Kapanipathi et al., 2021) leverages abstract meaning representation(AMR) as the intermediate logical form to represent the question. NSQA begins by parsing the question into an AMR graph that is rooted, directed, and acyclic, and then performs entity linking in the schema ground stage to align the entity nodes in the AMR graph to the KBs. Starting from these linked entities, NSQA explore a path-expanding approach to transform the AMR graph into the pre-defined query graph, which can be directly translated into executable logical expressions. Similarly, Li and Ji (2022) define six semantic structures and compose these structures to form a query graph as the intermediate logical form.

Also, some approaches utilize the abstract sketch of the target logical expression as the intermediate logical form. Purkayastha et al. (2022) generate the target SPARQL sketch using a Seq2Seq model, and then use entity and relation linker to complete the SPARQL sketch. Cao et al. (2022b) first parse the original question into the skeleton of KoPL program, a sequence of symbolic functions, and then train an argument parser to retrieve corresponding arguments of these functions.

While above methods generate the intermediate logical forms with one step directly, other methods alternately update the logical forms through schema grounding. Mo et al. (2022) transform the parsing process into a multi-turn dialogue, allowing for corrections via human feedback. To address the challenge of large search space, ArcaneQA (Gu and Su, 2022) leverages a dynamic program induction process in which the final logical form is gradually expanded from subprograms while adhering to KB constraints and pre-defined admissible actions.

Generator. With the advance of pre-trained language models (PLMs), many works have started to cast the semantic parsing task to a sequence-to-sequence (Seq2Seq) logical expression generation task. The most intuitive approach is to fine-tune a pre-trained encoder–decoder model to directly generate the logical expression from the natural language question. KoPL (Cao et al., 2022a), for example, investigate RNN (Dong and Lapata, 2016) and BART (Lewis et al., 2020) as the generators. Because it is composed of pre-defined atomic operations on KBs, such as Find, Relate, QueryName, SelectBetween, and so on, the KoPL program can model many complex reasoning processes. GraphQ IR (Nie et al., 2022) defines a natural-language-like logical expression to unify multiple graph query languages. Generating such logical expression can bridges the semantic gap and formally defined syntax that preserves the graph structure.

However, such a generator can only produce complex logical expressions from the training set. Many works add a schema ground stage to solve the problem, which augments the original question with auxiliary information such as linked entities, relations, candidate logical forms, and so on. As auxiliary information, Hu et al. (2022) propose two additional modules for retrieving candidate entities and relations. Then they employ a multi-task learning paradigm to simultaneously learn relation classification, entity disambiguation, and logical form generation, all of which use the same encoder model of the generation model. Similarly, ReTrack (Chen et al., 2021b) retrieves schema relevant to the question as additional input by a dense retriever based on BERT (Kenton and Toutanova, 2019). They do, however, use a checker during the generation process to avoid syntax errors and ensure executability. Differently, RNG-KBQA (Ye et al., 2022) uses the top-ranked candidate logical expressions as additional input for the generator. It first enumerates all of the candidate logical forms that are within two hops of the detected entities in the question and then use a rank model to determine the top-ranked candidates.

Graph-based methods aim to embed the original question as well as the entities and relations in KBs. And based on its learning representation, each entity in KB is predicted to be the answer or not. Because graph-based methods are a type of neural reasoning, they have limited expression and reasoning ability for complex questions, particularly numerical questions. With the same analysis–extend–reason framework, in the semantic-level analysis stage, graph-based methods typically encode the original question into an embedding by neural network methods such as Glove (He et al., 2021b) or RoBERTa (Zhang et al., 2022). Then in the next instance-level extending stage, they frequently restrict the reasoning space within a question-relevant subgraph expanding from the detected topic entities in the question. Finally, they obtain the answers in the comprehension-level reasoning stage by reasoning over the question-relevant subgraph. Graph-based methods focus primarily on subgraph construction and graph reasoning.

Subgraph Constructor. Some works construct appropriately sized question-specific subgraphs to both include the target answers and reduce noise, reducing the difficulty of subsequent graph reasoning. Zhang et al. (2022), for example, propose a subgraph retriever to obtain a semantically relevant subgraph to the question. It expands paths from detected entities in the question, and at each extension, it chooses top-K paths relevant to the question. The semantic relevance between the question and the path is calculated using the concatenation as input of RoBERTa (Liu et al., 2019). It expands the scope of the subgraph as much as possible while keeping the scale of the subgraph as small as possible using top-K semantic-relevant retrieval.

Graph Reasoner. To answer multi-hop complex questions, graph-based methods attempt to explicitly and implicitly model the multi-hop path connecting the topic entities and answers in the reasoning module. Because of the powerful reasoning ability of graph neural networks, these reasoners can address constrained and the set logical questions to some extent. NSM (He et al., 2021b) proposes a teacher–student framework in which the student network learns to infer the final answer while the teacher network learns to find intermediate signals. The

teacher network considers both forward reasoning starting from topic entities and backward reasoning starting from answers. The intermediate entity distributions from the teacher network will be offered to the student network as intermediate supervision signals. Instead of using the entity distributions to model multi-hop reasoning, Some approaches use relation chains(paths) in the KB. ImRL (Zhao et al., 2022), for example, first identifies the entities and relation phrases in the question, then finds candidate relation paths in the KB, and finally ranks these candidates using BERT (Kenton and Toutanova, 2019) and RotatE (Sun et al., 2019b) to embed relation phrases in the question as well as candidate relation paths in the KB. The answers are inferred following the top-1 ranked path. Rce-KGQA (Jin et al., 2021) use knowledge graph embedding techniques to capture the implicit relation chains in the KB to overcome the missing implied relations between the topic entities and answers. Feng et al. (2021) pretrain a transformer model before the graph reasoner to enable numerical computing.

3.1.2. Text2SQL

Relational databases, which store a large amount of data, are another important source of structured data. The Text2SQL task, given a relational database, aims to automatically translate natural language questions into SQL queries that can be executed on the database to obtain answers. Text2SQL approaches, as a pure SP-based QA method, can answer a wide range of complex questions. We integrate Text2SQL approaches into an encoder–decoder paradigm, in which the encoder and decoder play the roles of "Schema Ground" and "Generator", respectively, in Fig. 2.

Encoder (Schema Ground). The encoder intends to encode both the question and database schema, which includes tables, columns,primary keys, and foreign keys. Schema linking, which aligns schema items (including tables and columns) to the phrases in the question, is a critical function of the encoder. When generating the target SQL query, the schema linking will assist the model in recognizing and selecting required schema items. The encoder can be divided into sequence-based and graph-based models on the form of the input data.

Sequence-based encoders (Xie et al., 2022; Scholak et al., 2021; Shaw et al., 2021; Lin et al., 2020; Qi et al., 2022) take the concatenation of the question and the serialized database schema as input. Then, a pre-trained language model (PLM), such as BERT (Kenton and Toutanova, 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), and the encoder part of T5 (Raffel et al., 2020) is leveraged to encode the tokens in the sequence into context-aware embeddings, which are then fed into the decoder to generate SQL queries. In this way, the schema grounding is accomplished implicitly via the self-attention mechanism in PLMs. The original PLMs, on the other hand, are pre-trained on large-scale corpora using some specific pre-training objectives, such as the mask language model (MLM). Due to the large difference in datasets and training objectives between pre-training and fine-tuning, directly fine-tuning PLMs on the downstream Text2SQL task may be ineffective. To address the issue, Yu et al. (2021) and Shi et al. (2021b) pre-train the PLMs on existing or synthetic database-related data. Through pre-training, Text2SQL-related knowledge and ability are injected into these PLMs improving performance.

Graph-based encoders (Wang et al., 2020b; Cai et al., 2021; Cao et al., 2021) take one or more heterogeneous graphs as input. The graph' nodes typically represent question tokens, tables, or columns. The graph's edges represent the relationships between two nodes, such as "ForeignKey" relationship between two table nodes and "PrimaryKey" relationship between a table node and a column node. In the graph-based encoder, the schema linking is already modeled in the input graph, which assigns relations between question tokens and schema items based on string match degree (*i.e.*, exact match, partial match, and no match). Then, for each input graph, a relational graph neural network like RGAT (Wang et al., 2020a) and RGCN (Schlichtkrull et al., 2018) or a relation-aware self-attention transformer (Shaw et al., 2018) is used to perform message passing on the graph. If the input is made up of multiple graphs, some additional interaction operations are usually introduced for information exchange across different graphs (Cao et al., 2021; Cai et al., 2021). To obtain better initial representations of the nodes on the graph, a sequence-based encoder plus a pooling module is sometimes inserted before the graph-based encoder (Cao et al., 2021).

Decoder (Generator). The decoder takes the outputs of the encoder to generate desired SQL queries. The generated SQL query, on the other hand, may contain errors due to type mismatch or incorrect grammar, which have a significant impact on the Text2SQL model's performance. Grammar-based and execution-guided decoders are proposed to control the auto-regressive decoder to generate sequences that satisfy specific constraints (*e.g.*, the context-free grammar of SQL).

Grammar-based methods train the decoder on SQL grammar. The AST-based decoder (Yin and Neubig, 2017) generates an action sequence that depicts a SQL query's abstract syntax tree (AST) in depth-first traversal order. PICARD (Scholak et al., 2021) employs incremental parsing to constrain the auto-regressive decoder, which is a semi-auto-regressive bottom-up decoding method (Rubin and Berant, 2021). In particular, PICARD rejects invalid tokens at each decoding step to assist the decoder in generating valid SQL queries.

To avoid introducing complex grammar knowledge into the decoder, some works, such as (Suhr et al., 2020; Wang et al., 2018; Xuan et al., 2021), use an off-the-shelf SQL executor (such as SQLite) to validate the validity of generated SQL queries (or partial SQL queries). Wang et al. (2018) proposes a decoding strategy with an executor-in-the-loop. Specifically, the execution engine is used at appropriate decoding steps to determine whether a candidate (*i.e.*, generated partial SQL) contains semantic errors. In the beam, the candidate with errors will be discarded. On top of that, Sead (Xuan et al., 2021) extends the decoding process by proposing a clause-sensitive execution-guided decoder. Suhr et al. (2020) propose a generate-and-check decoding procedure. Concretely, it performs beam search on the decoder to generate several candidate SQL queries before selecting the highest-probability candidate SQL query without execution errors.

The mismatch between natural language questions and the corresponding SQL queries causes neural network learning difficulties. To bridge the gap above, some works, such as Gan et al. (2021), Guo et al. (2019), Yu et al. (2018a), Wolfson et al. (2020), propose a SQL intermediate representation (IR) that preserves core SQL functionalities while simplifying some operators (such as JOIN ON). The decoder is encouraged to generate IRs instead of the original SQL queries during training and inference. Finally, a non-trainable translator can convert these IRs to SQL queries.

3.1.3. Correlation of SP-based KBQA and Text2SQL

Both SP-based KBQA and Text2SQL approaches use the schema information in the data source to translate a natural language question into a logical expression that adheres to the grammar of structured query language. Intuitively, methods in these two tasks can easily adapt to each other because the only difference is the distinct ways to access data. However, despite the similar macro pattern (both involve schema-ground module and generator module), approaches in such two tasks differ significantly in terms of model design and focusing challenges.

First, while filtering and pruning of schema candidates is critical in KBQA approaches, it is often overlooked in Text2SQL approaches. The reason is that the KBs have far more schema items than relational databases. For example, the most famous KB, FreeBase (Bollacker et al., 2008), has over 6,000 relations, whereas the relational DBs in SPIDER (Yu et al., 2018b), the most popular Text2SQL benchmark, have only 27.6 columns on average. Because of the large scale, KBQA approaches must carefully filter and retrieve the appropriate number of schema candidates for the following reason module, while attempting to strike a balance between efficiency and precision. Text2SQL methods, on the other hand, can take all schema in the data source and focus more on encoding questions and such schema.

Second, KBQA models require instance-level entity connections in addition to the schema-level connections to help generate accurate logical expressions, whereas Text2SQL models rarely consider such instance-level connections between specific database rows. In relational databases, such instance-level reasoning is unnecessary because once a foreign key determines a schema-level connection, the corresponding rows are also connected. That, however, is not held in KBs. Even if a relation and a entity satisfy a schema-level connection where the domain class is matched, they still can be disconnected from each other.

Third, most Text2SQL approaches focus on translating the correct operation between schema items with appropriate grammar, whereas obtaining the target logical form skeleton is not a challenge in most KBQA approaches. The reason is that the usage of the relational DB and the KB are different. Because the relational DB serves operational and analytical purposes, the questions in Text2SQL prefer statistic analysis, which requires a large number of operation. On the contrary, the KB is typically used for reasoning and uncovering hidden connections between relations or entities at the semantic level. Therefore, questions in KBQA are designed to elicit factual knowledge, and the skeleton of the corresponding logical expression is often simple and basic.

To summarize, reasoning in KB is much more difficult than reasoning in relational DB. In contrast to Text2SQL methods, recent KBQA methods continues to focus on improving the performance on relatively basic complex questions. In particular, the most recent Text2SQL approaches can achieve promising results on difficult questions requiring more than four operations, whereas answering questions requiring more than three hops remains a challenge in KBQA.

3.2. Unstructured question answering

The unstructured resource is also a substantial corpus for quality assurance. The use of a large collection of unstructured texts to answer open-domain questions has been extensively researched (Chen et al., 2020a). Most of the QA models use an RC-based QA approach, as shown in Fig. 2, with an optional query reformulator, a text retriever, and a text reader. We will go over these three modules in depth, especially for complex questions.

Question Reformulator. The question is usually reformulated so that more relevant documents can be recalled in the next retrieval step (Xiong et al., 2020; Yadav et al., 2020, 2021; Zhang et al., 2021b). Most studies, such as MDR (Xiong et al., 2020), simply concatenate the retrieved passages with the question to reformulate the question. While Zhang et al. (2021b) trains a model to extract clues and concatenate clues with the question, Yadav et al. (2021) leverage an align-based model (Yadav et al., 2019) to extract the most similar tokens in passages to do query reformulation.

Document Retriever. It is hard for the retriever to directly reason answers for multi-hop complex questions within in a single step, since intermediate information is required. So most approaches utilize question reformulation or hyperlinking to enable multi-hop reasoning. The former updates the question based on the retrieved documents in the previous step, and then continues to retrieve document candidates based on the new question (Xiong et al., 2020; Zhang et al., 2021b). The latter leverages the provided hyperlinks in the retrieved documents to obtain documents for next step continuously (Zhang et al., 2021d). Compared to question reformulation, hyperlinking could find new documents faster but may introduce noise. To address such issue, some methods (Seonwoo et al., 2021; Xiong et al., 2020) utilize a ranking model is exploited to further rank the candidate documents at each stage, which is initialized by PLM and take the question and the document candidates as input.

Document Reader. Document reader aims to reason on retrieved documents to obtain answers. The existing methods leverage the reason module to address multi-hop, constrained, and numerical questions.

Multi-hop questions relies on the iterative retrieved documents and their relationships to reason the answer. Many works construct a graph with the retrieved documents as nodes and the citation relationship as edges, and then use a graph neural network for multi-hop reasoning. Many of them (Fang et al., 2019; Huang and Yang, 2021) use graph attention network (GAT) (Veličković et al., 2017) to propagate messages in the above graph and then predict the likelihood of a node in the graph as the answer. Some studies, such as (Fang et al., 2019; Huang and Yang, 2021; Luo et al., 2021), have shown that training answer prediction with other tasks, such as support sentence prediction and answer type prediction, can benefit answer prediction. Fang et al. (2019) revise the graph by including paragraphs, sentences, entities, and questions as nodes in their graph, which can significantly improve the subsequent subtasks. To avoid irrelevant propagation, Huang and Yang (2021) propose a question-relevant message-passing algorithm on the graph. Tu et al. (2020) use a multi-relational graph neural network to distinguish the different edge types in addition to homogeneous graph neural networks. Luo et al. (2021) simply combine the retrieval and reading stages with a unified PLM model in addition to graph neural networks. By pre-training the reasoning model on the simple QA dataset, Li et al. (2022) improve the multi-hop reasoning performance.

Constrained QA outputs each answer along with a constraint or condition instead of a single answer. For example, the answer to the question "Whether one can be honored as Officer of the Order of the British Empire?" is dependent on whether he has made public achievements, according to UK government documents. As a result, the condition "he has made achievements in public life" must constrain the answer "yes". The document retriever is not needed since a closed-domain context is usually provided for constrained QA. Although proposed for simple QA, the Fusion-in-Decode (FID) method (Izacard and Grave, 2020), which fuses multiple retrieved documents for unified decoding, is still adequate for solving constrained QA. A typical work (Sun et al., 2022b) parses the HTML structure of the given context to obtain the conditions and then uses the FID model to encode multiple conditions with the given question to predict the answer under various conditions.

Numerical reasoning is concerned with numerical operations between numbers, such as addition, subtraction, sorting, and counting. For numerical QA, a closed-domain context is also provided. Graph-based reasoning methods are commonly used in numerical reasoning over texts (Chen et al., 2020b; Huang et al., 2021a). For example, Chen et al. (2020b) create a typed homogeneous graph of entities and numbers. Huang et al. (2021a) divide the context into discourse units and treat them as nodes to construct the graph. The entity nodes associated with the question and their neighbors are iteratively updated, allowing message propagation across the graph guided by the question. Another common method is to equip the PLM with the numerical reasoning skills (Geva et al., 2020; Kim et al., 2022). Geva et al. (2020) generate synthetic datasets for the pre-training task to improve PLM's numerical reasoning ability. To adapt to the numerical pre-training task, they propose an extended BERT structure. Because Geva et al. (2020) may fail to obtain the contextual knowledge, Kim et al. (2022) propose a numerical-contextual BERT with an extra mask layer on top of BERT encoder to emphasize the number-related contextual information.

3.3. Semi-structured question answering

As discussed in Section 3.1.2, table question answering in relational databases is regarded as a purely semantic parsing (SP) task. However, besides relational databases, many tables are obtained form web and are less standard and structured. We refer the reasoning over such semi-structured data sources as web-table QA. Since the data in web tables can support various operation, existing web-table datasets typically contain a wide range of complex questions.

Due to the non-strict data structures, the SP-based method cannot easily applied on semi-structured QA as it is hard to obtain standard

logical expressions annotation. Therefore, the majority of existing research (Herzig et al., 2021a, 2020; Eisenschlos et al., 2021; Yang et al., 2022) do not produce logical expressions but employs RC-based QA methods which apply a retriever–reader framework and treat tables as text. Differ from RC-based methods in unstructured QA, above methods inject tables' structure information in both retriever and reader. They first apply a table reader which could encode structure information in web tables, and then pre-train the reader with some table-relevant tasks to improve their ability in encoding the table structures. Since questions are usually not formulated in web-table QA, we only detailed introduce the table retriever and the table reader in the following.

Table Retriever. The retriever measures the similarity of the question to each web table. DTR (Herzig et al., 2021b) is a typical model that applies a bi-encoder to represent the question and each web table separately. The model is trained in a weak supervised manner by predicting the table that appears closely with the text span. The retriever can assist in locating the most relevant tables to the question. In addition, Ma et al. (2022) develop a viable unified interface for a semi-structured knowledge source that contains both data and text. The key idea is to augment the retriever with a data-to-text verbalizing step for accessing heterogeneous knowledge sources, such as more WikiData data and more text from Wikipedia.

Table Reader. To improve the table reasoning ability, existing research typically pre-trains a table-sensitive PLM and then fine-tunes it to reason the cells from the given tables as the answers. Various strategies is applied to encode the table structures. Herzig et al. (2020), for example, pre-train a revised BERT on millions of text segments and tables crawled from Wikipedia. It adds additional row and column embeddings, as well as positional embeddings, to BERT's input to encode tabular structures. During pre-training, it predicts the masked tabular context in addition to the traditional masked tokens. Then, during fine-tuning, it predicts a set of selected cells and the corresponding aggregation operators, on which the final answers could be based.

Because many web tables have more than 20 rows, the transformer architecture have to deal with a long input which may cause time consuming. To improve the efficiency of the implementation, Eisenschlos et al. (2021) observe that a cell's context is only restricted to its neighbor, for example, the cells that share same column and row, and propose MATE with a sparse attention mechanism to enable the inner column attention and inner row attention mechanism. Specifically, MATE divides the heads in the multi-head attention into row and column heads to realize the above context locality.

To further alleviate the row and column order biases, TableFormer (Yang et al., 2022) injects 13 types of attention bias, such as "same row", "same column", and 'cell to column header' into the self-attention layer, to predict correct answers regardless of the row or column perturbation.

3.4. Multi-source question answering

Unstructured sources, such as text, have a broad coverage of knowledge due to their extremely large scale. However, they can hardly support complex question answering because the relationship or reason link between passages is frequently unknown. On the other hand, semi-structured and structured sources can easily answer complex questions, which benefits from the corresponding schema logical expressions, but suffer from incompleteness due to the restricted structured schema. Therefore, in order to effectively combine the benefits of both sources, multi-source question answering, also known as hybrid QA, is proposed to generate answers with heterogeneous knowledge. We divide hybrid QA into table-text QA and KB-text QA based on the supporting data sources, and will introduce each in turn in the following.

3.4.1. Table-Text QA

Most Table-Text QA approaches are RC-based methods and follow a retriever-and-reader framework. Given the question, they retrieve table sources and text sources respectively, and then link table cells to the retrieved passages or questions based on the hyperlink or content matching. Finally, the subsequent model exploits a retriever–reader pipeline to extract text spans or table cells as answers from the passage-enhanced tables. Because table cells and passages are connected as a graph through hyperlinks, such framework can handle multi-hop questions as well as numerical questions, since the hyperlink type between question and table cells can include numerical operation such as ">", which means that the value of a cell is greater than the value mentioned in the question.

Retriever and Reader. Following the hyperlink, MITQA (Kumar et al., 2021) splits each table into rows of records, with each record containing the table header, a row, and its linked passages. Following that, a PLM-based retriever is used to select the top k records that is relevant to the question, and then a reader is used to reason over such record candidates to obtain answers. Instead of reasoning over top-k records implicitly, CARP (Zhong et al., 2022) retrieves the hybrid chain directly and explicitly models the intermediate reasoning process. CAPR first lists all candidate hybrid chains based on the hyperlink between table cells and passages. Following that, a similar PLM-based retriever and reader framework is proposed, in which the retriever finds the most relevant hybrid chains and the reader reasons over the retrieved hybrid chains to extract the final answers. Some methods, in addition to utilizing a PLM-based reader, elaborately design the reasoning module. The reason process is viewed as a multi-hop inference path by HYBRIDER (Chen et al., 2020c). To begin, HYBRIDER encodes each cell along with its linked passages and uses a rank model to determine the most relevant cell. Starting from this cell, HYBRIDER then uses a hop model to jump to the next connected cell based on the question. Finally, HYBRIDER extracts a span as answer if the last hop is a text; otherwise, HYBRIDER directly takes the cell as the answer. DEHG (Feng et al., 2022) leverages a heterogeneous graph network to reason instead of modeling the paths. According to the hyperlink, the model first creates three sub-graphs: cell-passage sub-graphs, cell-question sub-graphs, and passage-question sub-graphs. DEHG learns embeddings for each node in the sub-graphs using a BERT-based context encoder. Finally, DEHG uses subgraph message passing and information propagation on the sub-graph to update node representation and exploits an LSTM to decode a text span as answers.

3.4.2. KB-Text QA

Combining KB and text is more difficult than combining web-tables and text, because unlike semi-structured web-tables, KBs are fully structured and have a larger gap with text format due to the reformulated schema items. A simple way to combine KB knowledge and text is directly converting KB triplets into text. For example, Unik-qa (Oguz et al., 2020) reformulates the triplet into a sentence and then employs a retriever-and-reader RC-based framework, similar to unstructured QA. However, Unik-qa disregards KB structure information. Recent KB-Text methods, on the other hand, use both KB knowledge and KB structure information to improve the subsequent retriever-and-reader module. Confined by the expression of neural reasoning methods, most KB-Text have limited reasoning ability for hard complex questions.

Retriever and Reader. One common approach is to build relationships between passages based on KB structure. For example, GRAFT-Net (Sun et al., 2018) retrieves relevant entities and documents from the KB and the text corpus in parallel, and then combines them to construct a question-specific subgraph based on the entity linking results. Given the retrieved subgraph, GRAFT-Net uses CNN variant to reason over the graph and select nodes as answers, which is equipped with heterogeneous update rules. However, the such heuristic building of subgraph suffers from coverage issues. To address this issue, PullNet (Sun et al.,

Table 1
Datasets overview.

Dataset	Size	Source	Multi-hop	Constrained	Numerical	Set logical
WebQSP (Yih et al., 2016)	4,737	FreeBase [KB]	✓	✓	✗	✗
CWQ (Talmor and Berant, 2018)	34,689	FreeBase [KB]	✓	✓	✓	✓
LC-QuAD (Trivedi et al., 2017)	5,000	DBPedia [KB]	✓	✓	✓	✗
LC-QuAD (Dubey et al., 2019) 2.0	30,000	DBPedia [KB]	✓	✓	✓	✓
GRAPHQ (Su et al., 2016)	5,166	FreeBase [KB]	✓	✓	✓	✗
GrailQA (Gu et al., 2021)	64,331	FreeBase [KB]	✓	✗	✓	✓
KQA Pro (Cao et al., 2022a)	117,970	Wikidata [KB]	✓	✓	✓	✓
Restaurants (Giordani and Moschitti, 2012)	378	Specific domain database [DB]	✓	✓	✓	✓
ATIS (Dahl et al., 1994)	5,280	Specific domain database [DB]	✓	✓	✓	✓
GeoQuery (Zelle and Mooney, 1996)	877	Specific domain database [DB]	✓	✓	✓	✓
Scholar (Iyer et al., 2017)	817	Specific domain database [DB]	✓	✓	✓	✓
Academic (Li and Jagadish, 2014)	196	Specific domain database [DB]	✓	✓	✓	✓
Yelp (Yaghmazadeh et al., 2017)	128	Website [DB]	✓	✓	✓	✓
IMDB (Yaghmazadeh et al., 2017)	131	Specific domain database [DB]	✓	✓	✓	✓
Advising (Finegan-Dollak et al., 2018)	3,898	Specific domain database [DB]	✓	✓	✓	✓
Spider (Yu et al., 2018b)	10,181	Wikipedia&Others [DB]	✓	✓	✓	✓
TriviaQA (Joshi et al., 2017)	95,956	Wikipedia&Web [Text]	✓	✗	✓	✗
NatualQuestions (Kwiatkowski et al., 2019)	307,373	Wikipedia [Text]	✓	✓	✓	✗
SearchQA (Dunn et al., 2017)	14,0461	Webpage [Text]	✓	✓	✓	✓
HoppotQA (Yang et al., 2018)	112779	Wikipedia [Text]	✓	✗	✓	✗
ConditionalQA (Sun et al., 2021)	3287	Website [Text]	✓	✓	✗	✗
DROP (Dua et al., 2019a)	96567	Wikipedia [Text]	✓	✗	✓	✓
WTQ (Pasupat and Liang, 2015)	2,108	Wikipedia[Table]	✓	✓	✓	✓
WikiSQL (Zhong et al., 2017)	80,654	Wikipedia [DB&Table]	✗	✓	✓	✓
AIT-QA (Katsis et al., 2021)	517	Wikipedia[Table]	✓	✓	✓	✓
HiTab (Cheng et al., 2022)	10,672	Wealth reports&Wikipedia[Table]	✓	✓	✓	✓
HybridQA (Chen et al., 2020c)	69,611	Wikipedia [Text&Table]	✓	✓	✓	✓
OTT-QA (Chen et al., 2020a)	45,000	Wikipedia [Text&Table]	✓	✓	✓	✓
NQ-Table (Herzig et al., 2021a)	12,000	Wiki [Text&Table]	✓	✓	✓	✗
TAT-QA (Zhu et al., 2021)	16,552	Financial Reports [Text&Table]	✗	✓	✓	✓
FinQA (Chen et al., 2021a)	8,281	Financial Reports [Text&Table]	✗	✓	✓	✓

2019a) leverages retrieval operations. PullNet begins with small entity sets detected from questions and then expands the graph through the "pull" operations which retrieve new information from the KB or the corpus. GRAPHREADER (Min et al., 2019), on the other hand, introduces a BERT-based reader model to extend and propagate information from related passages and obtain knowledge-rich representations for each node in the subgraph in order to take advantage of the strong reasoning ability.

Instead of using the KBs to build the heterogeneous subgraph, another approach uses KB to help with document retrieval scoring. KAQA (Zhou et al., 2020) uses the KB connection to select neighbor documents for each retrieved document and modifies the retrieval score by the scores of the corresponding neighbors. The answers are then extracted from the retrieved documents by a BERT-based reader, which is followed by an answer re-ranker. Similarly, KG-FiD (Yu et al., 2022) uses KB-based re-scoring on both retriever and reader. Specifically, KG-FiD uses a graph attention network to learn better representations for passages based on the passage graph as constructed by KB, and then uses the representations to re-score the retriever and reader.

4. Datasets and evaluation metrics

4.1. QA dataset

Many datasets have been proposed to facilitate the development of complex question answering. We summarize all natural language based QA datasets and list them in Table 1 to provide a clear overview of the question types involved in each dataset.

KBQA. WebQSP (Yih et al., 2016) is the first large-scale KBQA dataset based on Freebase with semantic-parsing annotation. Most questions in WebQSP are derived from google search logs and are simple multi-hop questions with no more than three hops. On top of that, CWQ (Talmor and Berant, 2018) is built by automatically extending and aggregating questions in WebQSP, which includes more complicated multi-hop questions, constrained questions, set logical questions, and numerical questions. LC-QuAD (Trivedi et al., 2017) and LC-QuAD 2.0 (Dubey et al., 2019) are complex KBQA datasets based on Wikidata and DBpedia rather than Freebase. The latter is an advanced version of the former, with more complicated logical questions. In addition, some datasets are proposed to test generalization in complex KBQA. GRAPHQ (Su et al., 2016) leverage paraphrased complex questions to test natural language robustness, whereas GrailQA (Gu et al., 2021) supports all levels of generalization problems on complex questions. KQA Pro (Cao et al., 2022a) provides large-scale natural language questions with corresponding SPARQL and KoPL annotations, and covers all reasoning types.

Text2SQL. Most early Text2SQL datasets, such as Scholar (Iyer et al., 2017), Academic (Li and Jagadish, 2014), GeoQuery (Iyer et al., 2017), Restaurants (Giordani and Moschitti, 2012), ATIS (Dahl et al., 1994), Yelp (Yaghmazadeh et al., 2017), and IMDB (Yaghmazadeh et al., 2017), mainly focus on the complex but single-domain questions. Advising (Finegan-Dollak et al., 2018) introduces a single-domain dataset as well as improved versions of existing datasets. Based on Wikipedia, WIKISQL (Zhong et al., 2017) is designed to provide a large body of question-SQL pairs focusing on cross-domain Text2SQL. However, SQL queries in WIKISQL are not complex enough because they only use a few simple SQL operators. Furthermore, each WIKISQL question involves only one table. Spider (Yu et al., 2018b) is a large-scale human-annotated cross-domain dataset containing complex question-SQL pairs involving multi-table reasoning and requiring many complex SQL operators such as "INTERSECT", "GROUP BY", and "JOIN ON".

Document QA. Many open domain text-based QA datasets, such as TriviaQA (Joshi et al., 2017), NatualQuestions (Kwiatkowski et al., 2019) and SearchQA (Dunn et al., 2017), have recently been proposed, which contain complex question–answer pairs accompanied by multiple documents as the context. However, because the documents are chosen separately through simple information retrieval, the reasoning path among them is not guaranteed to be interesting. Furthermore, HotpotQA (Yang et al., 2018) is proposed as the first explainable multi-hop QA dataset on text corpus, with two golden paragraphs and eight distracting paragraphs in each sample. Besides multi-hop

complex questions, ConditionalQA (Sun et al., 2022a) contains complex questions with conditional answers that are only applicable in a specific condition, and DROP (Dua et al., 2019b) contains all the other types of complex questions, including constrained questions, numerical questions, and set logical questions.

Web-Table QA. Most Web-table QA datasets are based on Wikipedia tables, which have over 14 billion HTML tables available for table retrieval. More specifically, Pasupat et al. create WTQ (Pasupat and Liang, 2015) which contains 22,033 question–answer pairs on 2,108 tables, and Zhong et al. construct WikiSQL (Zhong et al., 2017) which includes SQL annotation on the Wiki tables. The datasets presented above only consider flat layout tables. On the contrary, AIT-QA (Katsis et al., 2021) and HiTab (Cheng et al., 2022) are proposed to address complex questions on hierarchical tables with hierarchical row and column headers.

Table-Text Hybrid QA. HybridQA (Chen et al., 2020c) is the first proposed table-text hybrid dataset, with each question aligned with a Wikipedia table and multiple paragraphs linked with table cells. Instead of providing table candidates for each question, OTT-QA (Chen et al., 2020a) and NQ-table (Herzig et al., 2021a) are a open-domain datasets that require the model to retrieve both tables and text to obtain answers. Unlike the above datasets that focus on multi-hop reasoning, TAT-QA (Zhu et al., 2021) and FinQA (Chen et al., 2021a) constructs a large-scale dataset that samples from the real financial reports and contains a large number of numerical questions.

KB-Text Hybrid QA. To the best of our knowledge, KB-Text does not have any constructed datasets. They primarily evaluate on the open domain text QA datasets, such as TriviaQA (Joshi et al., 2017) and WebQuestions (Berant et al., 2013), or KBQA datasets, such as CWQ (Talmor and Berant, 2018) and WebQAP (Yih et al., 2016), which are introduced in the former. Most methods choose WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), and ConceptNet (Speer et al., 2017) as KB sources, and Wikipedia (Vrandečić and Krötzsch, 2014) as text corpus for the external sources.

4.2. Evaluation metrics

Exact match and F1 scores are two standard evaluation metrics in QA tasks. We will introduce them respectively in the following sections.

Exact Match. Exact match (EM) measures the perfect match between the predicted answers and ground truth. For semi-structured or unstructured sources, EM checks the string matching between the predict text span and the target text span. For structured sources, EM metrics are commonly used in SP-based methods, which can be classed into string-based, set-based, and graph-based. String-based EM checks the string match between the predict and the target logical expressions. Set-based EM first executes the logical expression to obtain the answer set, and then compares the answer set for exact matching. Graph-based EM converts the logical expression in the structured source to graph queries before determining the graph isomorphism.

F1 Score. F1 score represents the overlap ratio between the predict and the target answer sets, taking into account both precision and recall:

$$F1score = \frac{2 \times recall \times precision}{recall + precision}$$

where precision represents the ratio of the correctly predicted answers over the predicted answer set and recall represents the ratio of the correctly predicted answers over the target answer set. For unstructured and semi-structured source, F1 score treats the answer as a token set and evaluates the token overlap between the predict and the target. For structured data source like KB, the answer is a entity set, thus F1 score directly evaluates the overlap between the predict entity set and the target entity set.

5. Conclusion and future directions

This paper examines complex factual question answering on various data sources, including structured knowledge bases and tables, unstructured web-tables, semi-structured documents, and hybrid data sources. We attempt to unify existing methods from various data sources into a unified analysis–extend–reason framework and list datasets from various data sources with different question types. We also discuss potential future challenges, which are listed below.

Improving Reasoning Capability of Unstructured QA. For unstructured QA, existing models primarily use the retriever–reader framework, and focus on multi-hop reasoning. Recently, datasets involving numerical reasoning such as Dua et al. (2019a) and Amini et al. (2019) have been created. We believe that improving reasoning capability of unstructured QA is a promising research direction, with a focus on numerical and constrained reasoning. A possible solution is to investigate question decomposition (Wolfson et al., 2020) technique and numerical representation learning to make numerical reasoning differentiable.

Improving Generalization Ability of QA Models. The robustness and generalizability of QA systems are critical for realistic application. Existing research focuses primarily on the I.I.D. setting of models, without taking into account O.O.D. settings, where different dimensions of generalization, such as domain and compositional generalization, can be considered (Gu et al., 2021). Recently, several benchmarks have been proposed such as GrailQA (Gu et al., 2021) and KQA Pro (Cao et al., 2022a), and we believe that developing QA models with compositional reasoning ability and few/zero-shot schema grounding ability is a promising direction.

Developing Hybrid QA Models. Another important direction is the combination of multiple data sources. On one hand, existing KB-Text Hybrid QA models typically use different datasets for evaluation, such as dropping a portion of the existing KB to obtain an incomplete one and replying on external text information for Hybrid QA. We believe that a well-designed Hybrid benchmark with a consistent standard is beneficial for KB-involved Hybrid QA. Improving model explainability, on the other hand, requires more attention for model design because existing models typically use a retriever–reader framework.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abbasiantaeb, Z., Momtazi, S., 2021. Text-based question answering from information retrieval and deep neural network perspectives: A survey. Wiley Interdisc. Rev. Data Min. Knowl. Discov. 11 (6), e1412.

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., Hajishirzi, H., 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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). Association for Computational Linguistics, Minneapolis, Minnesota.

Berant, J., Chou, A., Frostig, R., Liang, P., 2013. Semantic parsing on freebase from question-answer pairs. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1533–1544.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J., 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. pp. 1247–1250.

Cai, R., Yuan, J., Xu, B., Hao, Z., 2021. SADGA: structure-aware dual graph aggregation network for text-to-SQL. In: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021. NeurIPS 2021, December 6-14, 2021, Virtual, pp. 7664–7676.

Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S., Yu, K., 2021. LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Association for Computational Linguistics, pp. 2541–2555.

Cao, S., Shi, J., Pan, L., Nie, L., Xiang, Y., Hou, L., Li, J., He, B., Zhang, H., 2022a. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 6101–6119.

Cao, S., Shi, J., Yao, Z., Lv, X., Yu, J., Hou, L., Li, J., Liu, Z., Xiao, J., 2022b. Program transfer for answering complex questions over knowledge bases. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 8128–8140.

Chen, W., Chang, M.-W., Schlinger, E., Wang, W., Cohen, W.W., 2020a. Open question answering over tables and text. arXiv preprint arXiv:2010.10439.

Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.-H., Routledge, B.R., et al., 2021a. FinQA: A dataset of numerical reasoning over financial data. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3697–3711.

Chen, S., Liu, Q., Yu, Z., Lin, C.-Y., Lou, J.-G., Jiang, F., 2021b. Retrack: a flexible and efficient framework for knowledge base question answering. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. pp. 325–336.

Chen, K., Xu, W., Cheng, X., Xiaochuan, Z., Zhang, Y., Song, L., Wang, T., Qi, Y., Chu, W., 2020b. Question directed graph attention network for numerical reasoning over text. arXiv preprint arXiv:2009.07448.

Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y., 2020c. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1026–1036.

Cheng, Z., Dong, H., Wang, Z., Jia, R., Guo, J., Gao, Y., Han, S., Lou, J.-G., Zhang, D., 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1094–1110.

Clark, K., Luong, M., Le, Q.V., Manning, C.D., 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In: 8th International Conference on Learning Representations. ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net.

Dahl, D.A., Bates, M., Brown, M.K., Fisher, W.M., Hunicke-Smith, K., Pallett, D.S., Pao, C., Rudnicky, A., Shriberg, E., 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In: Human Language Technology: Proceedings of a Workshop Held At Plainsboro. New Jersey, March 8-11, 1994.

Dong, L., Lapata, M., 2016. Language to logical form with neural attention. In: 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), pp. 33–43.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M., 2019a. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. arXiv preprint arXiv:1903.00161.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M., 2019b. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 2368–2378. http://dx.doi.org/10.18653/v1/N19-1246.

Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J., 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In: International Semantic Web Conference. Springer, pp. 69–78.

Dunn, M., Sagun, L., Higgins, M., Guney, V.U., Cirik, V., Cho, K., 2017. Searchqa: A new q&a dataset augmented with context from a search engine. arXiv preprint arXiv:1704.05179.

Eisenschlos, J.M., Gor, M., Müller, T., Cohen, W.W., 2021. Mate: Multi-view attention for table transformer efficiency. In: EMNLP.

Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., Liu, J., 2019. Hierarchical graph network for multi-hop question answering. arXiv preprint arXiv:1911.03631.

Feng, Y., Han, Z., Sun, M., Li, P., 2022. Multi-hop open-domain question answering over structured and unstructured knowledge. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 151–156.

Feng, Y., Zhang, J., He, G., Zhao, W.X., Liu, L., Liu, Q., Li, C., Chen, H., 2021. A pretraining numerical reasoning model for ordinal constrained question answering on knowledge base. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 1852–1861.

Finegan-Dollak, C., Kummerfeld, J.K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., Radev, D., 2018. Improving text-to-SQL evaluation methodology. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 351–360.

Gan, Y., Chen, X., Xie, J., Purver, M., Woodward, J.R., Drake, J.H., Zhang, Q., 2021. Natural SQL: making SQL easier to infer from natural language specifications. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021, Association for Computational Linguistics, pp. 2030–2042.

Geva, M., Gupta, A., Berant, J., 2020. Injecting numerical reasoning skills into language models. arXiv preprint arXiv:2004.04487.

Giordani, A., Moschitti, A., 2012. Automatic generation and reranking of SQL-derived answers to NL questions. In: Proceedings of the Second International Conference on Trustworthy Eternal Systems Via Evolving Software, Data and Knowledge. pp. 59–76.

Gu, Y., Kase, S., Vanni, M., Sadler, B., Liang, P., Yan, X., Su, Y., 2021. Beyond IID: Three levels of generalization for question answering on knowledge bases. In: Proceedings of the Web Conference 2021. pp. 3477–3488.

Gu, Y., Su, Y., 2022. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. arXiv preprint arXiv:2204.08109.

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J., Liu, T., Zhang, D., 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, pp. 4524–4535.

He, G., Lan, Y., Jiang, J., Zhao, W.X., Wen, J., 2021a. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In: WSDM.

He, G., Lan, Y., Jiang, J., Zhao, W.X., Wen, J.-R., 2021b. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 553–561.

Herzig, J., Mueller, T., Krichene, S., Eisenschlos, J., 2021a. Open domain question answering over tables via dense retrieval. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 512–519.

Herzig, J., Müller, T., Krichene, S., Eisenschlos, J., 2021b. Open domain question answering over tables via dense retrieval. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Online, pp. 512–519. http://dx.doi.org/10.18653/v1/2021.naacl-main.43.

Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., Eisenschlos, J.M., 2020. Tapas: Weakly supervised table parsing via pre-training. arXiv:Abs/2004.02349.

Hu, X., Wu, X., Shu, Y., Qu, Y., 2022. Logical form generation via multi-task learning for complex question answering over knowledge bases. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 1687–1696.

Huang, Y., Fang, M., Cao, Y., Wang, L., Liang, X., 2021a. Dagn: Discourse-aware graph network for logical reasoning. arXiv preprint arXiv:2103.14349.

Huang, X., Kim, J.-J., Zou, B., 2021b. Unseen entity handling in complex question answering over knowledge base via language generation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, Punta Cana, Dominican Republic, pp. 547–557. http://dx.doi.org/10.18653/v1/2021.findings-emnlp.50.

Huang, Y., Yang, M., 2021. Breadth first reasoning graph for multi-hop question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 5810–5821.

Iyer, S., Konstas, I., Cheung, A., Krishnamurthy, J., Zettlemoyer, L., 2017. Learning a neural semantic parser from user feedback. In: 55th Annual Meeting of the Association for Computational Linguistics 2017.

Izacard, G., Grave, E., 2020. Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282.

Jin, N., Siebert, J., Li, D., Chen, Q., 2022. A survey on table question answering: Recent advances. arXiv preprint arXiv:2207.05270.

Jin, W., Yu, H., Tao, X., Yin, R., 2021. Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning. arXiv preprint arXiv:2110.12679.

Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L., 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1601–1611.

Kapanipathi, P., Abdelaziz, I., Ravishankar, S., Roukos, S., Gray, A., Astudillo, R.F., Chang, M., Cornelio, C., Dana, S., Fokoue-Nkoutche, A., et al., 2021. Leveraging abstract meaning representation for knowledge base question answering. In: Findings of the Association for Computational Linguistics. ACL-IJCNLP 2021, pp. 3884–3894.

Katsis, Y., Chemmengath, S., Kumar, V., Bharadwaj, S., Canim, M., Glass, M., Gliozzo, A., Pan, F., Sen, J., Sankaranarayanan, K., et al., 2021. AIT-QA: Question answering dataset over complex tables in the airline industry. arXiv preprint arXiv:2106.12944.

Kenton, J.D.M.-W.C., Toutanova, L.K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186.

Kim, J., Kang, J., Kim, K.-m., Hong, G., Myaeng, S.-H., 2022. Exploiting numerical-contextual knowledge to improve numerical reasoning in question answering. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 1811–1821.

Kumar, V., Chemmengath, S., Gupta, Y., Sen, J., Bharadwaj, S., Chakrabarti, S., 2021. Multi-instance training for question answering across table and linked text. arXiv preprint arXiv:2112.07337.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al., 2019. Natural questions: A benchmark for question answering research. Trans. Assoc. Comput. Linguist. 7, 453–466.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2019. ALBERT: A lite BERT for self-supervised learning of language representations. In: International Conference on Learning Representations.

Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W.X., Wen, J.-R., 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. arXiv preprint arXiv:2105.11644.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880.

Li, F., Jagadish, H.V., 2014. Constructing an interactive natural language interface for relational databases, Proc. VLDB Endowment 8 (1), 73–84,.

Li, M., Ji, S., 2022. Semantic structure based query graph prediction for question answering over knowledge graph. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 1569–1579.

Li, X.-Y., Lei, W.-J., Yang, Y.-B., 2022. From easy to hard: Two-stage selector and reader for multi-hop question answering. arXiv preprint arXiv:2205.11729.

Lin, X.V., Socher, R., Xiong, C., 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In: Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020. In: Findings of ACL, vol.EMNLP 2020, Association for Computational Linguistics, pp. 4870–4888.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Luo, M., Chen, S., Baral, C., 2021. A simple approach to jointly rank passages and select relevant sentences in the OBQA context. arXiv preprint arXiv:2109.10497.

Ma, K., Cheng, H., Liu, X., Nyberg, E., Gao, J., 2022. Open domain question answering with a unified knowledge interface. arXiv:2110.08417v2.

Mavi, V., Jangra, A., Jatowt, A., 2022. A survey on multi-hop question answering and generation. arXiv preprint arXiv:2204.09140.

Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM 38 (11), 39–41.

Min, S., Chen, D., Zettlemoyer, L., Hajishirzi, H., 2019. Knowledge guided text retrieval and reading for open domain question answering. arXiv preprint arXiv:1911.03868.

Mo, L., Lewis, A., Sun, H., White, M., 2022. Towards transparent interactive semantic parsing via step-by-step correction. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 322–342.

Nie, L., Cao, S., Shi, J., Sun, J., Tian, Q., Hou, L., Li, J., Zhai, J., 2022. GraphQ IR: Unifying the semantic parsing of graph query languages with one intermediate representation. ArXiv, arXiv:2205.12078.

Oguz, B., Chen, X., Karpukhin, V., Peshterliev, S., Okhonko, D., Schlichtkrull, M., Gupta, S., Mehdad, Y., Yih, S., 2020. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. arXiv preprint arXiv:2012.14610.

Pandya, H.A., Bhatt, B.S., 2021. Question answering survey: Directions, challenges, datasets, evaluation matrices. arXiv preprint arXiv:2112.03572.

Pasupat, P., Liang, P., 2015. Compositional semantic parsing on semi-structured tables. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, pp. 1470–1480. http://dx.doi.org/10.3115/v1/P15-1142.

Pérez, J., Arenas, M., Gutierrez, C., 2009. Semantics and complexity of SPARQL. ACM Trans. Database Syst. 34 (3), 1–45.

Purkayastha, S., Dana, S., Garg, D., Khandelwal, D., Bhargav, G.S., 2022. A deep neural approach to KGQA via SPARQL Silhouette generation. In: 2022 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.

Qi, J., Tang, J., He, Z., Wan, X., Cheng, Y., Zhou, C., Wang, X., Zhang, Q., Lin, Z., 2022. RASAT: Integrating relational structures into pretrained Seq2Seq model for text-to-SQL. ArXiv, arXiv:2205.06983.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21, 140:1–140:67.

Rajpurkar, P., Jia, R., Liang, P., 2018. Know what you don't know: Unanswerable questions for squad. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 784–789.

Rubin, O., Berant, J., 2021. Smbop: Semi-autoregressive bottom-up semantic parsing. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics, pp. 311–324.

Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M., 2018. Modeling relational data with graph convolutional networks. In: The Semantic Web - 15th International Conference. ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, In: Lecture Notes in Computer Science, vol.10843, Springer, pp. 593–607.

Scholak, T., Schucher, N., Bahdanau, D., 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Association for Computational Linguistics, pp. 9895–9901.

Seonwoo, Y., Lee, S.-W., Kim, J.-H., Ha, J.-W., Oh, A., 2021. Weakly supervised pre-training for multi-hop retriever. arXiv preprint arXiv:2106.09983.

Shaw, P., Chang, M., Pasupat, P., Toutanova, K., 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, Association for Computational Linguistics, pp. 922–938.

Shaw, P., Uszkoreit, J., Vaswani, A., 2018. Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), Association for Computational Linguistics, pp. 464–468.

Shi, J., Cao, S., Hou, L., Li, J., Zhang, H., 2021a. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 4149–4158.

Shi, P., Ng, P., Wang, Z., Zhu, H., Li, A.H., Wang, J., dos Santos, C.N., Xiang, B., 2021b. Learning contextual representations for semantic parsing with generation-augmented pre-training. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence. EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, pp. 13806–13814.

Speer, R., Chin, J., Havasi, C., 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In: Thirty-First AAAI Conference on Artificial Intelligence.

Su, Y., Sun, H., Sadler, B., Srivatsa, M., Gür, I., Yan, Z., Yan, X., 2016. On generating characteristic-rich question sets for qa evaluation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 562–572.

Suhr, A., Chang, M., Shaw, P., Lee, K., 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, pp. 8372–8388.

Sun, H., Bedrax-Weiss, T., Cohen, W., 2019a. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. 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, pp. 2380–2390.

Sun, H., Cohen, W.W., Salakhutdinov, R., 2021. Conditionalqa: A complex reading comprehension dataset with conditional answers. arXiv preprint arXiv:2110.06884.

Sun, H., Cohen, W., Salakhutdinov, R., 2022a. ConditionalQA: A complex reading comprehension dataset with conditional answers. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Dublin, Ireland, pp. 3627–3637. http://dx.doi.org/10.18653/v1/2022.acl-long.253.

Sun, H., Cohen, W.W., Salakhutdinov, R., 2022b. Reasoning over logically interacted conditions for question answering. arXiv preprint arXiv:2205.12898.

Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J., 2019b. Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197.

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W., 2018. Open domain question answering using early fusion of knowledge bases and text. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4231–4242.

Talmor, A., Berant, J., 2018. The web as a knowledge-base for answering complex questions. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 641–651.

Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J., 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In: International Semantic Web Conference. Springer, pp. 210–218.

Tu, M., Huang, K., Wang, G., Huang, J., He, X., Zhou, B., 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. pp. 9073–9080.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

Vrandečić, D., Krötzsch, M., 2014. Wikidata: A free collaborative knowledgebase. Commun. ACM 57 (10), 78–85.

Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R., 2020a. Relational graph attention network for aspect-based sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, pp. 3229–3238.

Wang, B., Shin, R., Liu, X., Polozov, O., Richardson, M., 2020b. 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. ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, pp. 7567–7578.

Wang, C., Tatwawadi, K., Brockschmidt, M., Huang, P.-S., Mao, Y., Polozov, O., Singh, R., 2018. Robust text-to-SQL generation with execution-guided decoding. ArXiv: Computation and Language.

Wolfson, T., Geva, M., Gupta, A., Goldberg, Y., Gardner, M., Deutch, D., Berant, J., 2020. Break it down: A question understanding benchmark. Trans. Assoc. Comput. Linguist. 8, 183–198.

Wu, P., Zhang, X., Feng, Z., 2019. A survey of question answering over knowledge base. In: China Conference on Knowledge Graph and Semantic Computing. Springer, pp. 86–97.

Xie, T., Wu, C.H., Shi, P., Zhong, R., Scholak, T., Yasunaga, M., Wu, C.-S., Zhong, M., Yin, P., Wang, S.I., Zhong, V., Wang, B., Li, C., Boyle, C., Ni, A., Yao, Z., Radev, D., Xiong, C., Kong, L., Zhang, R., Smith, N.A., Zettlemoyer, L., Yu, T., 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. EMNLP.

Xiong, W., Li, X.L., Iyer, S., Du, J., Lewis, P., Wang, W.Y., Mehdad, Y., Yih, W.-t., Riedel, S., Kiela, D., et al., 2020. Answering complex open-domain questions with multi-hop dense retrieval. arXiv preprint arXiv:2009.12756.

Xuan, K., Wang, Y., Wang, Y., Wen, Z., Dong, Y., 2021. SeaD: End-to-end text-to-SQL generation with schema-aware denoising. CoRR abs/2105.07911.

Yadav, V., Bethard, S., Surdeanu, M., 2019. Alignment over heterogeneous embeddings for question answering. 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). pp. 2681–2691.

Yadav, V., Bethard, S., Surdeanu, M., 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. arXiv preprint arXiv:2005.01218.

Yadav, V., Bethard, S., Surdeanu, M., 2021. If you want to go far go together: Unsupervised joint candidate evidence retrieval for multi-hop question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Yaghmazadeh, N., Wang, Y., Dillig, I., Dillig, T., 2017. SQLizer: query synthesis from natural language. Proc. ACM Programm. Lang. 1 (OOPSLA), 1–26.

Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., Paul, S., 2022. Tableformer: Robust transformer modeling for table-text encoding. arXiv:Abs/2203.00274.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D., 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600.

Ye, X., Yavuz, S., Hashimoto, K., Zhou, Y., Xiong, C., 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 6032–6043.

Yih, W.-t., Richardson, M., Meek, C., Chang, M.-W., Suh, J., 2016. The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 201–206.

Yin, P., Neubig, G., 2017. A syntactic neural model for general-purpose code generation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, Association for Computational Linguistics, pp. 440–450.

Yu, T., Wu, C., Lin, X.V., Wang, B., Tan, Y.C., Yang, X., Radev, D.R., Socher, R., Xiong, C., 2021. GraPPa: Grammar-augmented pre-training for table semantic parsing. In: 9th International Conference on Learning Representations. ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net.

Yu, T., Yasunaga, M., Yang, K., Zhang, R., Wang, D., Li, Z., Radev, D.R., 2018a. SyntaxSQLNet: Syntax tree networks for complex and cross-DomainText-to-SQL task. CoRR abs/1810.05237.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., Radev, D.R., 2018b. 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. Brussels, Belgium, October 31 - November 4, 2018, Association for Computational Linguistics, pp. 3911–3921.

Yu, D., Zhu, C., Fang, Y., Yu, W., Wang, S., Xu, Y., Ren, X., Yang, Y., Zeng, M., 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4961–4974.

Zelle, J.M., Mooney, R.J., 1996. Learning to parse database queries using inductive logic programming. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2. pp. 1050–1055.

Zhang, J., Chen, B., Zhang, L., Ke, X., Ding, H., 2021a. Neural, symbolic and neural-symbolic reasoning on knowledge graphs. AI Open 2, 14–35.

Zhang, Y., Dai, H., Kozareva, Z., Smola, A.J., Song, L., 2018. Variational reasoning for question answering with knowledge graph. In: AAAI.

Zhang, Y., Nie, P., Ramamurthy, A., Song, L., 2021b. Answering any-hop open-domain questions with iterative document reranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 481–490.

Zhang, Z., Wu, Y., Zhou, J., Duan, S., Zhao, H., Wang, R., 2020. SG-net: Syntax-guided machine reading comprehension. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. pp. 9636–9643.

Zhang, Z., Yang, J., Zhao, H., 2021c. Retrospective reader for machine reading comprehension. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. pp. 14506–14514.

Zhang, X., Zhan, K., Hu, E., Fu, C., Luo, L., Jiang, H., Jia, Y., Yu, F., Dou, Z., Cao, Z., et al., 2021d. Answer complex questions: Path ranker is all you need. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 449–458.

Zhang, J., Zhang, X., Yu, J., Tang, J., Tang, J., Li, C., Chen, H., 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 5773–5784.

Zhao, Y., Huang, J., Hu, W., Chen, Q., Qiu, X., Huo, C., Ren, W., 2022. Implicit relation linking for question answering over knowledge graph. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 3956–3968.

Zhong, W., Huang, J., Liu, Q., Zhou, M., Wang, J., Yin, J., Duan, N., 2022. Reasoning over hybrid chain for table-and-text open domain question answering. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. IJCAI-22, pp. 4531–4537.

Zhong, V., Xiong, C., Socher, R., 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103.

Zhou, M., Shi, Z., Huang, M., Zhu, X., 2020. Knowledge-aided open-domain question answering. arXiv preprint arXiv:2006.05244.

Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., Chua, T.-S., 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 3277–3287.