# RAG-based Question Answering over Heterogeneous Data and Text

Philipp Christmann, Gerhard Weikum

Max Planck Institute for Informatics
Saarland Informatics Campus, Germany
{pchristm, weikum}@mpi-inf.mpg.de

## Abstract

This article presents the Quasar system for question answering over unstructured text, structured tables, and knowledge graphs, with unified treatment of all sources. The system adopts a RAG-based architecture, with a pipeline of evidence retrieval followed by answer generation, with the latter powered by a moderate-sized language model. Additionally and uniquely, Quasar has components for question understanding, to derive crisper input for evidence retrieval, and for re-ranking and filtering the retrieved evidence before feeding the most informative pieces into the answer generation. Experiments with three different benchmarks demonstrate the high answering quality of our approach, being on par with or better than large GPT models, while keeping the computational cost and energy consumption orders of magnitude lower.

## 1 Introduction

**Motivation and Problem.** The task of question answering, QA for short, arises in many flavors: factual vs. opinions, simple lookups vs. multi-hop inference, single answer vs. list of entities, direct answers vs. long-form, one-shot questions vs. conversations, and other varieties (see, e.g., surveys [28, 29]). The state-of-the-art for this entire spectrum has been greatly advanced in the past decade. Most notably, incorporating deep learning into retriever-reader architectures (e.g., [2, 13, 18]) has boosted answering quality, and most recently, large language models (LLM) [25, 40] have pushed the envelope even further (e.g., [16]).

Today's LLMs alone are capable of accurately answering many *factoid* questions, simply from their pre-trained parametric memory which latently encodes huge text corpora and other online contents. However, this critically depends on the frequency of evidence in the underlying contents and the complexity of the information need. For example, asking for the MVP of the 2024 NBA season would easily return the correct answer Nikola Jokic, but asking for the highest-scoring German NBA player or the MVP of the 2024 German basketball league pose a big challenge. The reason is that LLMs alone do not easily recall information about not so popular or even long-tail entities [17, 32], and that they are mainly geared for direct look-ups as opposed to connecting multiple pieces of evidence [24, 39].

[10, 12, 21, 41] known as RAG, address these bottlenecks. In addition to cleverly crafted prompts and few-shot examples, the LLM is provided with the top-ranked results of an explicit retrieval step, like web search or knowledge graph (KG) lookups. The former is often necessary for freshness of answers, and the latter may help with long-tail entities and also mitigate the notorious risk of hallucinations. Still, this generation's RAG architectures are limited in how broad and how deep they tap into external sources. Popular AI assistants like Gemini or ChatGPT seem to primarily retrieve from the text of web pages (incl. Wikipedia articles), and academic research has additionally pursued knowledge augmentation by enhancing prompts with facts from large KGs (e.g., Wikidata).

An additional content modality that is still underexplored are online tables: a wide range of tabular data including HTML tables in web pages, spreadsheets and statistics, all the way to CSV and JSON datasets that are abundant on the Internet. There is prior work on joint support for text and KGs and for text and tables, but very little on all of these together – some notable exceptions being [3, 5, 27, 38].

**Examples.** All three heterogeneous types of sources are crucial not only for answering different questions from different kinds of evidence, but also for combining multiple pieces of evidence of different modalities to infer correct and complete answers. To illustrate the need for tapping all sources, consider the following questions:

*Q*1: *Which Chinese basketballers have played in the NBA?*
*Q*2: *Who was the first Chinese NBA player?*
*Q*3: *Which Chinese NBA player has the most matches?*

Q1 can be cast into querying a KG, but the list there is not necessarily complete and up-to-date, so additional evidence from text or tables would be desired. Q2 needs information about who played in which seasons, found only in web pages or sports-statistics tables. Finally, Q3 may be lucky in finding prominent textual evidence (e.g., in biographies, Wikipedia etc.), but this often faces divergent statements, and resolving contradictions needs to dive into more evidence. Besides, when textual evidence is rare and hard to find or not trustworthy enough, then information from multiple tables and text snippets may have to be aggregated (e.g., totals of per-season counts). Some of this may perhaps become feasible for an industrial LLM's RAG capabilities in the near future, but there are always harder scenarios by moving from Chinese NBA players deeper into the long tail, such as asking for Lithuanian players in the German handball league.

**Approach and Contribution.** This paper presents a simple but powerful and versatile RAG system with unified access to text, KG and tables. We call our method QUASAR (for Question Answering over Heterogeneous Sources with Augmented Retrieval). Its architecture is relatively straightforward: all heterogeneous content is verbalized and indexed for retrieval; a retriever finds top-ranked results for the given question (from different source types), and these are fed into the LLM for answer generation. This is the unsurprising bird-eye's view. Specific details that are key factors for the strong performance of QUASAR are:

i) automatically casting user questions into a structured representation of the information need, which is then used to guide
ii) judicious ranking of search results, with multiple rounds of re-ranking and pruning, followed by
iii) extracting faithful answers from an LLM in RAG mode, with answers grounded in tangible evidence.

The paper presents experiments with three different benchmarks, covering various flavors of questions. We focus on one-shot questions; conversational QA is out of scope here, but QUASAR itself is well applicable to this case, too. Our experiments demonstrate that our methods are competitive, on par with big GPT models and often better, while being several orders of magnitude lower in computational and energy cost. The experimental findings also highlight that question understanding, with structured representation of user intents, and iterative re-ranking of evidence are crucial for good performance.

Overall, our contribution lies in proposing a unified system architecture for RAG-based question answering over a suite of different data sources, with strong points regarding both effectiveness (i.e., answer quality) and efficiency (i.e., computational cost).

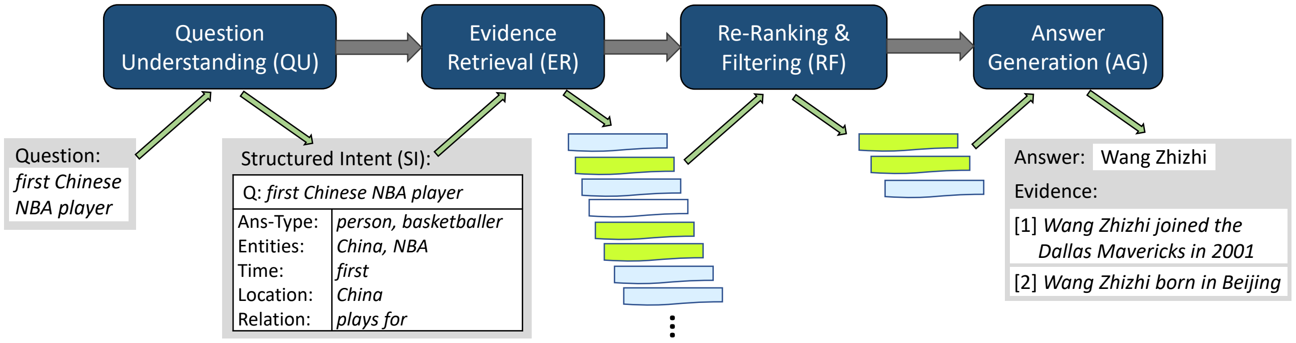

Figure 21: Overview of the Quasar system.

## 2 Related Work

The RAG paradigm came up as a principled way of enhancing LLM factuality incl. provenance and mitigating the risk of hallucination [12, 21]. It is highly related to the earlier retriever-reader architectures for QA [2, 18], especially when the reader uses the fusion-in-decoder method [13, 27]. Since its invention, RAG methodology has been greatly advanced, introducing a wide suite of extensions, such as batched inputs, interleaving retrieval and generation steps, and more (see the recent surveys [10, 41]).

On question answering (QA), there is a vast amount of literature including a wealth of differently flavored benchmarks (see, e.g., [28]). The case of interest here is QA over heterogeneous sources, tapping into both unstructured content and structured data. A variety of works has pursued this theme by combining knowledge graphs with text sources, using graph-based methods, neural learning and language models (e.g., [30, 31, 37]).

Most relevant for this article is the research on jointly leveraging all different sources: text, KGs, and tables (incl. CSV and JSON files). This includes the UniK-Qa system [27], the Spaghetti/SUQL project [23, 38], the Matter method [19], the STaRK benchmarking [34], and our own prior work [3, 5] (without claiming exhaustiveness). Out of these, we include UniK-Qa, Spaghetti and our own systems Convinse and Explaignn as baselines in the experimental evaluation. Their architectures are similar to ours, but UniK-Qa and Spaghetti do not have our distinctive elements of question understanding and iterative re-ranking (originally introduced in Explaignn [5]).

## 3 Methodology

The Quasar system is a pipeline of four major stages, as illustrated in Figure 21. First, the input question is analyzed and decomposed, in order to compute a structured intent (SI) representation that will pass on to the subsequent steps, along with the original question. Second, the SI is utilized to retrieve pieces of evidence from different sources: text, KG and tables. Third, this pool of potentially useful evidence is filtered down, with iterative re-ranking, to arrive at a tractably small set of most promising evidence. The final stage generates the answer from this evidence, passing back the answer as well as evidence snippets for user-comprehensible explanation.

The second and fourth stage, Evidence Retrieval (ER) and Answer Generation (AG), are fairly standard. Such a two-phase architecture was called a retriever-reader architecture [42]. With a modern LLM replacing the earlier kinds of neural readers, this is the core of every RAG system [10].

Stages 1 and 3 are unique elements of our architecture, judiciously introduced to improve both effectiveness (i.e., answer quality) and efficiency (i.e., computational cost). Question Understanding (QU) provides the ER component with crisper and semantically refined input, and the Re-Ranking &

Filtering (RF) stage is beneficial for distilling the best evidence from the large pool of retrieved pieces. The following subsections elaborate on the four stages of the pipeline, emphasizing the Quasar-specific steps QU and RF.

### 3.1 Question Understanding (QU)

To prepare the retrieval from different kinds of sources, including a KG, ad-hoc tables and text documents, it is useful to analyze and decompose the user question. In this work, we aim to cast a question into a structured intent (SI) representation: essentially a frame with faceted cues as slots, or equivalently, a concise set of key-value pairs. Figure 21 gives an idealized example for the question about the first Chinese NBA player. The facets or keys of potential interest here are:

- Ans-Type: the expected answer type (or types when considering different levels of semantic refinement),
- Entities: the salient entities in the question, and
- Relation: phrases that indicate which relation (between Q and A entities) the user is interested in.

In addition, as questions can have temporal or spatial aspects, the SI also foresees slots for:

- Time: cues about answer-relevant time points or spans, including relative cues (e.g., "before Covid") and ordinal cues (e.g., "first"), and
- Location: cues about answer-relevant geo-locations.

The ideal SI for example question Q2 would look like:

> Ans-Type: person, basketballer; Entities: China, NBA; Time: first; Location: China; Relation: plays for.

Note that the values for these slots can be crisp like entity names or dates, but they can also take the form of surface phrases. The SI purpose and value lie in the decomposition. In practice, many questions would only lead to a subset of faceted cues, leaving some slots empty. For the example in Figure 21, an alternative SI could simply consist of

> Ans-Type: person; Entities: China, NBA; Time: first.

Even this simplified SI can be highly beneficial in guiding the subsequent evidence retrieval.

To generate the SI from a user question, we employ a (small-scale) LM, specifically BART [20], a Transformer-based auto-encoder with 140M parameters.[1] BART is pre-trained for language representation; its power for our purpose comes from fine-tuning. To this end, we generate (question, SI) pairs by using an instruction-trained LLM like GPT-4, with few-shot in-context learning (following our earlier work [15]). Note that this is a one-time action; at inference-time we only use much smaller LMs. The generated silver-standard pairs are then used to fine-tune BART. In the experiments in this article, we leverage pre-existing collections of silver pairs, based on the training data of the CompMix benchmark [6], comprising 3,400 such pairs.

Although this paper focuses on single-shot questions, the Quasar architecture is also geared for conversational QA. In that setting, the SI can play an even bigger role, as (follow-up) questions are often formulated in a rather sloppy manner – all but self-contained. For example, a conversation could start with a clear question When did Wang Zhizhi join the NBA?, followed a few dialog steps later, by a user utterance like Which teams did he play for? or simply Which teams?. In such an informal conversation,

[1] `https://huggingface.co/facebook/bart-base`

the system needs to contextualize each user utterance based on the preceding turns in the dialog (e.g., inferring the relevant entities Wang Zhizhi and NBA from the conversational history). For details on conversational QA, based on our architecture, see our earlier works [3, 5].

### 3.2 Evidence Retrieval (ER)

The ER stage taps into a knowledge graph, a corpus of text documents, and a collection of web tables. Specifically, for the experiments, we use the Wikidata KG, all English Wikipedia articles, and all tables that are embedded in Wikipedia pages (incl. infoboxes, which can be seen as a special case of tables).

**Retrieval from KG:** To retrieve evidence from the KG, we utilize our earlier work Clocq [4], which provides entity disambiguations and a relevant KG-subgraph for a given query. Unlike most other works on QA-over-KG, Clocq fetches all KG-facts that are relevant for a given entity in a single step. For example, when querying for NBA players, it can traverse the KG neighborhood and pick up top teams, also considering so-called qualifier nodes in Wikidata which are often used for temporal scopes. As the disambiguation of entity names onto the KG can be tricky and noisy (e.g., China could be mapped to Chinese sports teams in all kinds of sports), Clocq considers several possible disambiguations [4] (typically in the order of 10 result entities). The queries for Clocq are constructed by concatenating all slots of the question's SI. For the example query about the first Chinese NBA player, good result entities would be Dallas Mavericks, lists about NBA seasons, MVP awards etc., and their associated facts. These provide cues, but are likely insufficient to answer the question.

**Retrieval from Text and Tables:** The disambiguated entities returned by Clocq form anchors for tapping into text and tables. Quasar first identifies relevant text documents and tables that refer to the anchor entities. With focus on Wikipedia, these are simply the articles for the respective entities. Quasar then constructs a keyword query that concatenates all available fields of the SI. The query is evaluated against a linearized and verbalized representation (see below) of all sentences and all table rows in the selected documents. This returns a set of sentences and and individual table rows, ranked by BM25 scores.

**Evidence Verbalization:** All results from the different data sources are uniformly treated by linearizing and verbalizing them into token sequences. For KG results, the entity-centric triple sets are linearized via breadth-first traversal of the mini-graph starting from the entity node. For tables, results are individual rows, which are contextualized by including labels from column headers and from the DOM-tree path of the article where the table comes from. For example, a table row about Wang Zhizhi playing for Dallas (Mavericks) in the 2000-2001 season, would be expressed as:

Wang Zhizhi / NBA Career / Season: 2000-2001, Team: Dallas, Games Played: 5 . . .

Finally, results from the text corpus are already in the form of token sequences, but we can additionally prefix these with the DOM-tree labels. We can think of this entire pool of evidence as an on-the-fly corpus of potentially relevant pseudo-sentences, forming the input of the subsequent RF stage.

**Result Ranking:** Overall, the ER stage compiles a substantial set of evidence, possibly many thousands of entities, text snippets and table rows. Therefore, we practically restrict the pool to a subset of high-scoring pieces, like the top-1000. For scoring, a simple BM25 model (a classical IR method) is applied. By default, we treat all evidence pieces uniformly with global scoring, no matter whether they come from KG, text or tables.

### 3.3 Re-Ranking and Filtering (RF)

With a pool of top-1000 evidence pieces, we could invoke an LLM for answer generation. However, that would face a large fraction of noise (i.e., misleading evidence) and incur high costs of computation and energy consumption.

For both of these reasons, we have devised light-weight techniques for iteratively reducing the top-1000 pieces to a small subset, say top-30 or top-10, that can be fed into an LLM at much lower cost (as LLM computations and pricing are at least linear in the number of input tokens). The difficulty is, of course, to do this without losing good evidence and reducing answer presence. Our techniques for this task are based on graph neural networks (GNNs) [35] or cross-encoders (CEs) [7, 22].

**GNN-based RF**. Given a large pool of evidence pieces from all sources, a bipartite graph is constructed:

- nodes being evidence pieces or entities that occur in these pieces, and
- edges connecting an evidence piece and an entity if the entity occurs in the evidence.

The task for the GNN is to jointly score the evidence and the entity nodes in a multi-task learning setup. The latter are the answer candidates, and the evidence should give faithful explanation for an answer. We build on our earlier work on explainable QA [5].

The node encodings are initialized with cross-encoder embeddings (see below) for node contents and the SI of the question. The inference iteratively adjusts the encodings based on message passing from neighboring nodes. The GNN is trained via weak supervision from question-answer pairs: evidence nodes are labeled as relevant if they are connected to a gold answer. More technical details are given in [5].

QUASAR invokes the GNN in multiple rounds, iteratively reducing top-$k$ to top-$k^*$ nodes with $k^* \ll k$. In practice, we would typically consider two rounds: re-ranking top-1000 and pruning to top-100, and then reducing to top-30 or top-10, which are passed to the answer generation stage. Note that this keeps the GNN at a tightly controlled size, so that its computational costs at inference-time are much smaller than those of an LLM.

**CE-based RF**. An alternative to the GNN inference is to employ a cross-encoder for scoring and re-ranking the evidence pieces. These are transformers (typically with a small LM like BERT) that are fine-tuned for scoring the relatedness between a query and a document [26]. In our case, the comparison is between the question SI and the evidence piece. In our experiments, we make use of two different cross-encoders, both trained on the MS-MARCO benchmark for passage retrieval [1], and fine-tuned on the respective benchmark (leveraging the same weak supervision data as for the GNNs), the difference being in model size.[2] We use the smaller model to reduce top-1000 to top-100, and the larger model to go further down from top-100 to top-30.

### 3.4 Answer Generation (AG)

The last stage follows mainstream practice to invoke an LLM in a retrieval-augmented manner. We call a 'small-scale' LLM, specifically a fine-tuned LlaMA-3.1 model (8B-Instruct)[3], with a prompt [4] consisting of:

- the concatenated SI of the original question, and
- the top-30 (or other top-$k^*$ with small $k^*$) evidence pieces.

[2] `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-4-v2` and `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2`

[3] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

[4] The specific prompt is *"SI: <`concatenated SI`> Evidence: <`evidence pieces`>"*.

By the previous down-filtering of the original pool of evidence pieces, this last step has affordable cost in terms of computation time and energy consumption.

**Fine-Tuning the LLM:** We considered adding an instruction to the prompting, such as "answer this question solely based on the provided evidence snippets". However, this turned out to be ineffective. The reason why the model works well without such instructions is our task-specific fine-tuning. We perform this by running the training data of benchmarks through the QUASAR pipeline, and training the AG stage with the top-30 evidence pieces as input. Thus, the fine-tuning makes the model learn the role of evidence for RAG-based QA.

**Explanations:** The top-30 evidence pieces can be used to provide users with explanation of answers. Optionally, these could be reduced further for comprehensibility. Alternatively, we can fine-tune the LLM to provide both answers and concise explanations. Since we can infer which evidences in the input mention the annotated ground-truth answers, our method could be fine-tuned to provide such *answering evidences* as well (cf. [11]).

# 4 Experiments

## 4.1 Experimental setup

**Benchmarks**. We run experiments on three benchmarks with different characteristics of questions.

- **COMPMIX**. COMPMIX [6] is a benchmark which was specifically designed for evaluating QA systems operating over heterogeneous sources. The dataset has 9,410 questions, out of which 2,764 are used for testing. Answers are crisp entity names, dates, or other literals.
- **CRAG**. We further evaluate on a subset of the CRAG [36] dataset, which was recently released as a testbed for RAG-based QA systems. We utilize the same pipeline and sources as outlined in Section 2, without using the web snippets or APIs provided with CRAG. This way we focus on entity-centric questions that do not require access to live web data (e.g., news feeds), and disregard cases where the results would be up-to-date quantities. This restricts the test data to 436 entity-centric questions, still enough for a proof of concept.
- **TIMEQUESTIONS**. To showcase the generalizability of our pipeline, we conduct experiments on TIMEQUESTIONS [14], a benchmark for temporal QA. The dataset requires temporal understanding and reasoning, which are well-known limitations of LLMs [8]. TIMEQUESTIONS has 16,181 questions (3,237 for testing).

  Typical examples for the questions in these three benchmarks are:

  > COMPMIX: *Which player won the most number of Man-of-the-Match titles in the FIFA world cup of 2006?*
  > CRAG: *What was the worldwide box office sales for little hercules?*
  > TIMEQUESTIONS: *Which club did Cristiano Ronaldo play for before joining Real Madrid?*

**Baselines**. As competitors or reference points to QUASAR, we study the performance of the following methods:

- **Generative LLMs**. We compare QUASAR against out-of-the-box LLMs: **GPT-3** (`text-davinci-003`), **GPT-4** (`gpt-4`) and **LLAMA3** (`meta-llama/Llama-3.1-8B-Instruct`). The same prompt is used for all LLMs, consistent with previous work [6, 38]: *"Please answer the following question by providing the crisp answer entity, date, year, or numeric number. Q: <*`question`*>"*.

- **Heterogeneous QA methods**. Convinse [3], UniK-Qa [27], Explaignn [5] are QA methods designed to integrate heterogeneous sources: text, tables and KG. All of these integrate the exact same sources as Quasar.
- **State-of-the-art**. For CompMix and TimeQuestions, we also compare against state-of-the-art methods from the literature: Spaghetti [38] and Un-Faith [15], which are among the best performing systems.

  Results are taken from the literature whenever applicable. On Crag, we use the models trained on CompMix for Quasar and heterogeneous QA baselines.

**Metrics**. We measure *precision at 1* (**P@1**) as our main metric [29] on all benchmarks. On Crag, we manually annotate answer correctness, as the ground-truth answer formats vary (e.g., entity name variants, lists, sentences).

We also compute the number of neural parameters aggregated over all sub-modules (#**Parameters**). Parameter counts for GPT-models are taken from [25] (Gpt-4 might have less active parameters during inference).

For further analysis we measure *answer presence* (**AP@k**), i.e. whether the answer is present in the top-$k$ ranked evidence pieces, and *mean reciprocal rank* within the top-$k$ evidences (**MRR@k**).

**Configuration**. Our implementation uses the `Llama3.1-8B-Instruct` model for the AG stage. For the QU, ER and RF stages we adopt code from the Explaignn project.[5] For the ER stage, we use Clocq, setting its specific parameters to $k = 10$ and $p = 1{,}000$.

As default, we use the GNN technique for the RF stage. For efficiency, we use light-weight models for initializing the GNN encoders – the same models used for the CE-based RF.[6] The GNNs are trained for 5 epochs with an epoch-wise evaluation strategy, i.e. we choose the model with the best performance on the respective dev set. We train the GNNs on graphs with a maximum of 100 evidence and 400 entity nodes (as scored by BM25). During inference, the first GNN is applied on graphs with 1,000 evidence and 4,000 entity nodes, shrinking the pool of evidence pieces to the top-100. The second GNN then runs on graphs with 100 evidence and 400 entity nodes. The factor of 4 entities per evidence (on average) holds sufficient for the observed data, and enables batched inference. Other parameters are kept as is.

The AG model, based on `Llama3.1-8B-Instruct`, is fine-tuned for 2 epochs with a warm-up ratio of 0.01 and a batch size of 8, again with an epoch-wise evaluation strategy. Other parameters are set to the default Hugging Face training parameters.[7]

### 4.2 Main results

**Quasar is competitive on all benchmarks**. Main results of our experiments are shown in Table 5. First of all, we note that Quasar achieves competitive performance across all three benchmarks.

On CompMix, baselines for heterogeneous QA and Llama3 perform similarly, whereas GPT-based LLMs can answer more than 50% of the questions correctly. Quasar exhibits substantially higher performance, on par with the state-of-the-art method Spaghetti [38] (which is based on Gpt-4).

On the Crag dataset, P@1 drops for all methods except for Gpt-4. The benchmark includes realistic questions, which can be ambiguous/confusing (*"who was the director for the report?"*), on "exotic" entities with answers in social media (*"how many members does the teknoist have?"*), or require up-to-date

[5] `https://explaignn.mpi-inf.mpg.de`

[6] `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-4-v2` and `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2`

[7] `https://huggingface.co/docs/transformers/v4.46.2/en/main_classes/trainer#transformers.TrainingArguments`

information (*"when did chris brown release a song or album the last time?"*), and other cases that are challenging for all methods.

Finally, QUASAR establishes new state-of-the-art performance on the TIMEQUESTIONS benchmark. Interestingly, all of the tested LLMs show greatly reduced performance on this benchmark, which inherently requires temporal understanding and reasoning – a known weakness of stand-alone LLMs.

| **Method ↓ / Benchmark →** | **COMPMIX** | **CRAG** | **TIMEQUESTIONS** | **#Parameters** |
|---|---|---|---|---|
| **GPT-3** | 0.502 | – | 0.224 | 175,000 M |
| **GPT-4** | 0.528 | **0.633** | 0.306 | 1,760,000 M |
| **LLAMA3 [33]** (8B-Instruct) | 0.431 | 0.385 | 0.178 | 8,030 M |
| **CONVINSE [3]** | 0.407 | 0.298 | 0.423 | 362 M |
| **UNIK-QA [27]** | 0.440 | 0.280 | 0.424 | 223 M |
| **EXPLAIGNN [5]** | 0.442 | 0.303 | 0.525 | 328 M |
| **STATE-OF-THE-ART** | **0.565**<br>(SPAGHETTI [38]) | – | 0.571<br>(UN-FAITH [15]) | – |
| **QUASAR (ours)** | 0.564 | 0.362 | **0.754** | 8,218 M |

Table 5: End-to-end P@1 of QUASAR and baselines on three benchmarks. Results for GPT-3 and GPT-4 are taken from the literature [6, 15]. GPT-3 is not accessible anymore, hence no results on CRAG.

**Integration of heterogeneous sources is vital**. QUASAR integrates evidence from text, KG and tables into a unified framework. We aim to better understand how this affects the answering performance of the method. Table 6 shows end-to-end answering performance of QUASAR with different combinations of the input sources. The results clearly indicate that all types of sources contribute, with option Text+KG+Tables performing best, with a large margin over tapping only single source types.

### 4.3 Analysis

**Unified retrieval enhances performance**. In the RF stage, we re-rank and filter evidence from different source types, and feed the unified top-$k^*$ into the AG stage. We conduct a comparison in which we consider the top-10 evidence pieces from each source type individually. This gives equal influence to KG, text and tables, whereas our default is based on global ranking. Table 7 shows the results for this analysis, showing our default choice performs better. The reason is that different questions require different amounts of evidence from each of the source types.

**QUASAR works well with small amounts of evidence**. We investigate the influence of the number of evidence pieces fed into the AG stage, varying it from 5 to 100. Results are shown in Figure 22. As the curve shows, there is a sharp increase in precision as we add evidence up to 30 or 40 pieces, which is around our default of top-30. This indicates that a certain amount of evidence is needed, to overcome the inherent noise and arrive at sufficient answer presence. As we increase the amount of evidence further, we observe a saturation effect, and eventually a degration of performance. Too much evidence not only has diminishing returns, but can actually be confusing for the AG stage. This reconfirms our heuristic choice of top-30: enough for good answering while keeping computational costs reasonably low.

**Ablation study on re-ranking**. For more insight on the possible configurations of the RF stage, we conducted an ablation study with different options, including solely relying on the initial BM25 scoring without explicit re-ranking. The results are shown in Table 8. We observe that the iterative reduction in

| Benchmark → | CompMix | | | TimeQuestions | | |
|---|---|---|---|---|---|---|
| **Input sources ↓ / Metric →** | **P@1** | **AP@100** | **AP@30** | **P@1** | **AP@100** | **AP@30** |
| **Text** | 0.455 | 0.563 | 0.531 | 0.539 | 0.515 | 0.487 |
| **KG** | 0.481 | 0.677 | 0.637 | 0.724 | 0.701 | 0.674 |
| **Tables** | 0.432 | 0.501 | 0.482 | 0.536 | 0.347 | 0.328 |
| **Text+KG** | 0.537 | 0.749 | 0.706 | 0.745 | **0.776** | 0.748 |
| **Text+Tables** | 0.503 | 0.632 | 0.594 | 0.567 | 0.578 | 0.549 |
| **KG+Tables** | 0.524 | 0.728 | 0.692 | 0.743 | 0.731 | 0.703 |
| **Text+KG+Tables** | **0.564** | **0.759** | **0.724** | **0.754** | **0.776** | **0.749** |

Table 6: Answer presence and answering precision of Quasar with different combinations of input sources (on the respective test sets).

| Input evidences ↓ / Metric → | P@1 | AP@30 |
|---|---|---|
| **Top-30 Text+KG+Tables (ours)** | **0.574** | **0.710** |
| **Top-10 Text + Top-10 KG + Top-10 Tables** | 0.560 | 0.709 |

Table 7: Answer presence and precision of Quasar for different choices of top-30 (on CompMix dev set).

two steps is slightly better than the single-step variants (going down from top-1000 to top-30 in one RF step). Between the two options of using a GNN or a CE, the differences are negligible. A notable effect is that our RF techniques retain the answer presence at a very high level, only a bit lower than for the initial top-1000. The last two rows of Table 8 demonstrate that RF is crucial: without explicit re-ranking, the technique of just picking smaller top-$k$ from the original BM25 model leads to substantial degradation in both answer presence and precision.

**Quality of SI**. To assess the quality and robustness of the Structured Intents, we inspected a sample of questions and their SIs. Table 9 gives three anecdotic examples. We show SIs generated by Quasar, which makes use of the pre-existing collection from the CompMix benchmark for training. This training data was obtained via different heuristics, which can be a limiting factor when user intents become more complex.

Therefore, we also looked at SIs derived via in-context learning (ICL) using Gpt-4 with 5 handcrafted examples. As shown in our earlier work on temporal QA [15], such data can be used for training smaller models (e.g., BART), which can greatly boost the completeness and overall quality of the generated SIs.

From the sampled set, we observed that the ICL-based SIs are more complete with all slots filled, whereas the BART-based SIs focused more on the main slots Answer-type, Entities and Relation. However, both approaches achieve very high quality in filling the slots, capturing the user's information need very well.

Interestingly, when questions get complicated, with nested phrases, the ICL-based variant succeeds in decomposing the questions, based on only 5 ICL examples. For example, for the question "which German state had the most Corona-related death cases in the first year after the outbreak?" the Time slot becomes "first year after Corona outbreak", which can be resolved to identify the temporal scope. In general, we believe that such question decomposition, beyond simple temporal constraints, would be an interesting theme for future work.

**Refraining from answering**. We can train our model to refrain from answering in scenarios where the

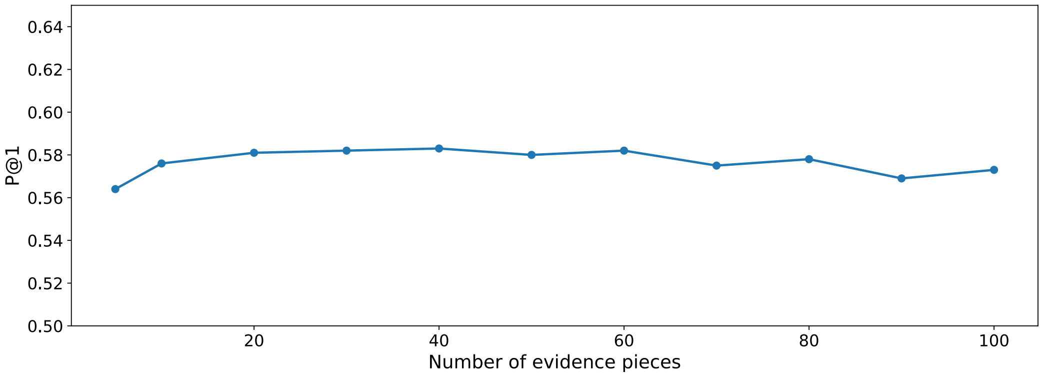

Figure 22: Performance of Quasar on the CompMix dev set with different numbers of evidence.

| | **CompMix** (dev set) | | | |
|---|---|---|---|---|
| **RF Method ↓ / Metric →** | **P@1** | **AP@100** | **AP@30** | **MRR@100** |
| **GNN: 1000 → 100 → 30** | **0.574** | 0.738 | 0.710 | **0.572** |
| **CE: 1000 → 100 → 30** | 0.573 | **0.740** | **0.721** | 0.553 |
| **GNN: 1000 → 30** | 0.567 | *n/a* | 0.710 | 0.567 |
| **CE: 1000 → 30** | 0.570 | *n/a* | 0.715 | 0.558 |
| **BM25: 100 (w/o GNN or CE)** | 0.490 | 0.652 | *n/a* | 0.259 |
| **BM25: 30 (w/o GNN or CE)** | 0.468 | *n/a* | 0.534 | 0.259 |

Table 8: Ablation study for different RF strategies of Quasar on the CompMix dev set. The answer presence in the RF input with top-1000 evidence pieces is 0.760.

provided evidence does not contain an answer to the question. Specifically, during training, when the answer is not present in the evidence, we change the target answer to unknown. This variant is referred to as Quasar(faithful).

We measure the ratio of questions for which unknown is provided as answer, and the P@1 restricted to questions that are answered. The accuracy of refraining from answering is measured as well, based on whether the answer is present in the evidence or not. We conduct this experiment on CompMix and TimeQuestions, for which we can compute answer presence exactly. We also compute results for Llama3, which is already instructed with the option to answer “don’t know”. Table 10 shows the results. For CompMix, we observe that Quasar has high accuracy on refraining when appropriate, whereas Llama3 tends to be overconfident with a very small rate of unknowns, leading to incorrect answers.

## 5 Insights, Limitations, and Challenges

**Benchmark Performance.** Our method, RAG-based Quasar with an 8B LLaMA model, outperforms much larger LLMs like Gpt-4 on two of the three benchmarks, with a very large margin for temporal questions. Obviously, pre-trained LLMs have only limited sense of properly positioning “remembered” facts on the timeline even with training data that exceeds ours by several orders of magnitude. This confirms our intuition that LLMs alone are not good at “recalling” higher-arity relations that require combining distant pieces of evidence. This is a sweet spot for RAG. Only for the Crag benchmark, Quasar is substantially inferior to a full-blown LLM. This is likely due to the nature of the questions: not necessarily the complexity of the information needs, but the need for more web sources (beyond what our experiments tap into).

| **Question** | **Current SI by Quasar** | **SI via ICL** |
|---|---|---|
| *what was disneys first color movie?* | Ans-Type: *animated feature film*<br>Entities: *disneys*<br>Relation: *was first color movie* | Ans-Type: *film, animated film*<br>Entities: *Disney*<br>Relation: *first color movie*<br>Time: *first* |
| *at the oscars, who won best actor in 2018?* | Ans-Type: *human*<br>Entities: *at the oscars*<br>Relation: *who won best actor in 2018* | Ans-Type: *person, actor*<br>Entities: *Oscars, 2018*<br>Relation: *won best actor*<br>Time: *2018* |
| *which German state had the most Corona-related death cases in the first year after the outbreak?* | Ans-Type: *state*<br>Entities: *Germany, Corona*<br>Relation: *which state had the most related death cases in the first year after the outbreak* | Ans-Type: *location, state*<br>Entities: *Germany, Corona-related deaths*<br>Relation: *highest count of death cases*<br>Location: *Germany*<br>Time: *first year after Corona outbreak* |

Table 9: Examples for pairs of question and generated SI.

| | **CompMix** | | | | **TimeQuestions** | | | |
|---|---|---|---|---|---|---|---|---|
| **Metric →**<br>**Method ↓** | **P@1** | **P@1 (answered)** | **Refrain rate** | **Refrain accuracy** | **P@1** | **P@1 (answered)** | **Refrain rate** | **Refrain accuracy** |
| **Llama3** | 0.431 | 0.471 | 0.089 | *n/a* | 0.177 | 0.276 | 0.392 | *n/a* |
| **Quasar (faithful)** | 0.497 | 0.713 | 0.303 | 0.838 | 0.597 | 0.804 | 0.257 | 0.864 |

Table 10: Performance of Quasar with option to refrain from answering (“don’t know”).

**Cost/Performance Ratio.** The most important take-away from our experiments is that Quasar achieves its competitive performance at a much lower cost than the full LLMs. Assuming that the consumed GFlops are proportional to the number of model parameters, Quasar achieves a cost reduction by a factor of 200x for Gpt-3 and 2000x for Gpt-4. This does not only mean less computation, but also a massively lower electricity bill and climate impact.

**Role of Question Understanding.** We did not systematically investigate the influence of the Structured Intent in the Quasar pipeline. However, the comparison to the big GPT models reflects the role of the SI, as we prompt the GPT models in their natural mode with the original questions. The linearized sequence of available SI slots does not always have major advantages, but there are enough cases where specific facets provide crucial cues. This holds especially for the Entities slot, as this drives the gathering of evidence in the ER stage (cf. [3], and for the Time slot, as these cues are often decisive for temporal questions (cf. [15]).

**Role of Re-Ranking.** As our ablation studies show, merely using top-$k$ evidence from an initial BM25-style ranking does not provide good performance. Also, there seems to be sweet spot in the choice of $k$: we need enough evidence for connecting the dots if the question requires multiple pieces of information, or for corroborating candidates if the question finds many useful but noisy pieces. In the experiments, $k = 30$ turns out to be good choice; much lower $k$ results in insufficient evidence, and much larger $k$ leads to saturation and ultimately degrading performance. Our argument for iteratively shrinking the candidate set in multiple rounds of re-ranking is substantiated in our experiments, but the gain of doing this, compared to GNN- or CE-based re-ranking from 1000 to 30, is not big. More research is called to better understand the role of ranking in RAG.

**Limitations of Evidence Retrieval.** For ER, we adopted more or less standard techniques. The results showed very good answer presence, in the order of 75% in the top-100 or even top-30. An important case where this is insufficient are questions that require aggregating information over a large number of evidence pieces. An example is asking for the life-time total of 3-point scores of the basketball player Dirk Nowitzki. This requires collecting a set of per-season tables with NBA player statistics, but also other web sources with numbers for his career before he joined the NBA (including his youth teams). Of course, there are sometimes shortcuts like a Wikipedia article or biography mentioning the total number, but this cannot be universally assumed. The bottom line is that ER should be reconsidered as well, striving to improve the recall dimension.

**Limitations of Answer Generation.** For AG, we simply rely on a LLM, using it as an extractor ("reader") from the given evidence. Despite the wide belief that LLMs can perform deep reasoning over many pieces of evidence, our experience is that the extraction works only well – robustly and faithfully – for relatively simple questions with a few multi-hop joins or simple aggregation over a few pieces. However, complicated questions such as asking for the top-100 NBA players with the largest number of life-time 3-point scores (again including their pre-NBA careers) are currently out of scope and will likely remain so for quite some time. This offers many opportunities for pushing the envelope further.

**Trust in Data Sources.** In our experiments, we considered all heterogeneous sources as trustworthy and unbiased. With focus on Wikidata and Wikipedia, this assumption has been well justified. In the wild, however, input data for RAG-based systems likely exhibit a wide spectrum of quality issues, in terms of stale information, biased positions, or simply false statements. Identifying trustworthy and up-to-date evidence and dealing with conflicting data, has been explored in other contexts (e.g., for KG curation [9]), but remains a major challenge for RAG-based QA.

**Open Challenges and Future Work.** The best-performing methods in our experiment, mostly QUASAR, reach P@1 values of 56% for COMPMIX and 75% for TIMEQUESTIONS. For the latter, the answer presence in the top-100 is only slightly higher; so the AG stage hardly misses anything. However, for COMPMIX, the answer presence is 75% – much higher than what our system can actually answer. Obviously, closing this gap is a major direction to pursue, with focus on the RF and AG stages. However, missing one fourth of the answers completely in the top-100 pool, is a big problem as well. This requires improving recall at the ER stage, possibly with better guidance by the QU, which in turn needs more sources beyond the scope of our experiments (currently limited to Wikidata and Wikipedia).

In general, we need to think beyond this kind of "benchmark mindset". Even if we reached 80% or 90% precision and recall, we would still have a substantial fraction of questions that are answered incorrectly or not at all. The remaining errors may not be a problem for chatbots, but they would be a showstopper for the deployment of mission-critical applications in business or science. We believe that this big gap is a shortcoming of all methods, not an issue that comes from the data alone. For trivia-style QA, as looked at in this paper, a smart human in "open book" mode and no time limitation should be able to properly answer practically all questions, just by reading pieces of web contents and putting things together. Neither LLMs nor state-of-the-art RAG are the final solution; substantial research and creative ideas are needed to further advance QA.

# References

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In arXiv 2018.

[2] Danqi Chen, Adam Fisch, Jason Weston and Antoine Bordes. Reading Wikipedia to Answer Open-Domain Questions. In ACL 2017.

[3] Philipp Christmann, Rishiraj Saha Roy, Gerhard Weikum. Conversational Question Answering on Heterogeneous Sources. In SIGIR 2022.

[4] Philipp Christmann, Rishiraj Saha Roy, Gerhard Weikum. Beyond NED: Fast and Effective Search Space Reduction for Complex Question Answering over Knowledge Bases. In WSDM 2022.

[5] Philipp Christmann, Rishiraj Saha Roy, Gerhard Weikum. Explainable Conversational Question Answering over Heterogeneous Sources via Iterative Graph Neural Networks. In SIGIR 2023.

[6] Philipp Christmann, Rishiraj Saha Roy, Gerhard Weikum. CompMix: A Benchmark for Heterogeneous Question Answering. In WWW 2024.

[7] Herve Dejean, Stephane Clinchant, Thibault Formal. A Thorough Comparison of Cross-Encoders and LLMs for Reranking SPLADE. In arXiv 2024.

[8] Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. Time-Aware Language Models as Temporal Knowledge Bases. In TACL 2022.

[9] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, Wei Zhang. Knowledge-Based Trust: Estimating the Trustworthiness of Web Sources. In PVLDB 2015.

[10] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. In arXiv 2023.

[11] Tianyu Gao, Howard Yen, Jiatong Yu, Danqi Chen. Enabling Large Language Models to Generate Text with Citations. In EMNLP 2023.

[12] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang. Retrieval Augmented Language Model Pre-Training. In ICML 2020.

[13] Gautier Izacard, Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In EACL 2021.

[14] Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex Temporal Question Answering on Knowledge Graphs. In CIKM 2021.

[15] Zhen Jia, Philipp Christmann, Gerhard Weikum. Faithful Temporal Question Answering over Heterogeneous Sources. In WWW 2024.

[16] Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, Davood Rafiei. Evaluating Open-Domain Question Answering in the Era of Large Language Models. In arXiv 2023.

[17] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, Colin Raffel. Large Language Models Struggle to Learn Long-Tail Knowledge. In ICML 2023.

[18] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In EMNLP 2020.

[19] Dongkyu Lee, Chandana Satya Prakash, Jack FitzGerald, Jens Lehmann. MATTER: Memory-Augmented Transformer Using Heterogeneous Knowledge Sources. In ACL 2024.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In ACL 2020.

[21] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In NeurIPS 2020.

[22] Jimmy Lin, Rodrigo Frassetto Nogueira, Andrew Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. In Morgan & Claypool Publishers 2021.

[23] Shicheng Liu, Jialiang Xu, Wesley Tjangnaka, Sina J. Semnani, Chen Jie Yu, Monica Lam. SUQL: Conversational Search over Structured and Unstructured Data with Large Language Models. In NAACL-HLT 2024.

[24] Vaibhav Mavi, Anubhav Jangra, Adam Jatowt. Multi-hop Question Answering. In Foundations and Trends in Information Retrieval 2024.

[25] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large Language Models: A Survey. In arXiv 2024.

[26] Rodrigo Frassetto Nogueira, Kyunghyun Cho. Passage Re-ranking with BERT. In arXiv 2019.

[27] Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Sejr Schlichtkrull, Sonal Gupta, Yashar Mehdad, Scott Yih. UniK-QA: Unified Representations of Structured and Unstructured Knowledge for Open-Domain Question Answering. In NAACL-HLT 2022.

[28] Anna Rogers, Matt Gardner, Isabelle Augenstein. QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. In ACM Computing Surveys 2023.

[29] Rishiraj Saha Roy, Avishek Anand. Question Answering for the Curated Web: Tasks and Methods in QA over Knowledge Bases and Text Collections. In Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers 2021.

[30] Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, Gerhard Weikum. UNIQORN: Unified Question Answering over RDF Knowledge Graphs and Natural Language Text. In Journal of Web Semantics 2024.

[31] Haitian Sun, Tania Bedrax-Weiss, William W. Cohen. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. In EMNLP/IJCNLP 2019.

[32] Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, Xin Luna Dong. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs? In NAACL-HLT 2024.

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample. Llama: Open and efficient foundation language models. In arXiv 2023.

[34] Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Subbian, James Zou, Jure Leskovec. STaRK: Benchmarking LLM Retrieval on Textual and Relational Knowledge Bases. In arXiv 2024.

[35] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. In IEEE Transactions on Neural Networks and Learning Systems 2021.

[36] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze D. Gui, Ziran W. Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, Xin Luna Dong. CRAG – Comprehensive RAG Benchmark. In arXiv 2024.

[37] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, Jure Leskovec. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In NAACL-HLT 2021.

[38] Heidi C. Zhang, Sina J. Semnani, Farhad Ghassemi, Jialiang Xu, Shicheng Liu, Monica S. Lam. SPAGHETTI: Open-Domain Question Answering from Heterogeneous Data Sources with Retrieval and Semantic Parsing. In ACL 2024.

[39] Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, Shen Huang. End-to-End Beam Retrieval for Multi-Hop Question Answering. In NAACL-HLT 2024.

[40] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, Ji-Rong Wen. A Survey of Large Language Models. In arXiv 2023.

[41] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Bin Cui. Retrieval-Augmented Generation for AI-Generated Content: A Survey. In arXiv 2024.

[42] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, Tat-Seng Chua. Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. In arXiv 2021.