Beyond Boundaries: A Human-like Approach for Question Answering over Structured and Unstructured Information Sources

Jens Lehmann◇† and **Dhananjay Bhandiwad**† and **Preetam Gattogi**† and **Sahar Vahdati**†
◇Amazon, Germany
jlehmnn@amazon.com
†ScaDS.AI / TU Dresden, Germany
{dhananjay.bhandiwad|preetam.gattogi|sahar.vahdati}@tu-dresden.de

Abstract

Answering factual questions from heterogenous sources, such as graphs and text, is a key capacity of intelligent systems. Current approaches either (i) perform question answering over text and structured sources as separate pipelines followed by a merge step or (ii) provide an early integration, giving up the strengths of particular information sources. To solve this problem, we present ''HumanIQ'', a method that teaches language models to dynamically combine retrieved information by imitating how humans use retrieval tools. Our approach couples a generic method for gathering human demonstrations of tool use with adaptive few-shot learning for tool augmented models. We show that HumanIQ confers significant benefits, including i) reducing the error rate of our strongest baseline (GPT-4) by over 50% across 3 benchmarks, (ii) improving human preference over responses from vanilla GPT-4 (45.3% wins, 46.7% ties, 8.0% loss), and (iii) outperforming numerous task-specific baselines.

1 Introduction

The ability to answer factual questions reliably is one of the hardest problems in AI. Large language models (LLMs) have shown promise on factual question answering (QA) tasks. However, LLMs that rely only on internal parameters can become stale and do not perform well on tail knowledge (Sun et al., 2023). Moreover, it is difficult to curate LLM's parametric knowledge or attribute responses to information sources. This has led to research in which LLMs are equipped with external *tools* to retrieve information from external sources (Thoppilan et al., 2022; Nakano et al., 2021) to reduce hallucinations and establish a connection between sources of evidence and the generated response to a question. Such external information sources can be in structured form, e.g., in the shape of a knowledge graph, or in unstructured form, e.g., text corpora. Textual sources usually provide higher recall whereas structured sources offer potentially higher precision and support executing complex queries. The field of *hybrid question answering*, which pre-dates the rise of LLMs, has investigated approaches to answer questions by retrieving information from structured and unstructured information sources.

A main approach to hybrid QA (a field sometimes also referred to as heterogeneous QA) is to use *late fusion*: In this line of research, the input question is processed by separate systems optimised for specific types of information sources and the responses are merged or selected afterwards. While this has the advantage of being easy to parallelize, a drawback is that the different information sources do not interact with each other, i.e., evidence found in one source does not influence pipelines for other sources. To remedy this, *early fusion* QA approaches integrate information sources into a unified format before performing inference steps. This allows to run a single pipeline over all sources. However, the unified format represents a least common denominator of the input sources and does not allow to use the strength of different input sources: (i) the flexibility and generality of unstructured representations and (ii) the ability to perform compositional queries to aggregate, combine, sort, and order structured data.

To overcome these restrictions, we propose to follow a new direction in this field: Rather than performing early or late fusion, we integrate inference steps over heterogeneous information sources by mimicking how humans find responses to question. We find that human reasoning tends to transcend the boundaries of different information sources and can benefit from mutually complementary information. Specifically, we propose

Transactions of the Association for Computational Linguistics, vol. 12, pp. 786–802, 2024. https://doi.org/10.1162/tacl_a_00671
Action Editor: Jimmy Lin. Submission batch: 10/2023; Revision batch: 12/2023; Published 6/2024.

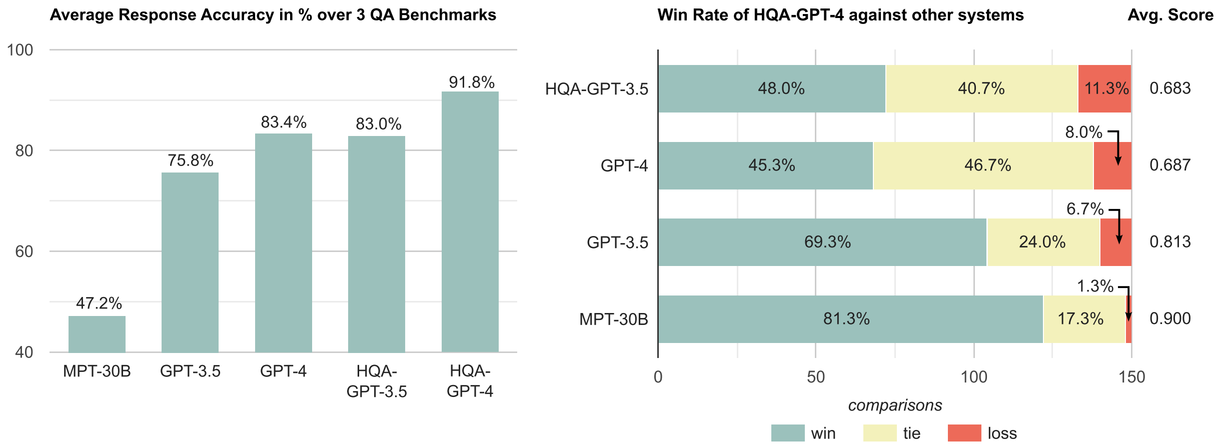


Figure 1: Overview of aggregated evaluation results for HumanIQ applied to hybrid question answering (HQA): On the left, the percentage of correctly answered questions over 3 QA benchmarks is plotted. On the right, the results of a human evaluation using questions from the same benchmarks is shown. For each question, annotators could indicate preference for the output of a system. Detailed results can be found in Section 5.

the HumanIQ (''Human-like Intelligent Queries'') methodology for collecting human solution processes and using those to prompt LLMs. The methodology can be applied to tasks that use tools and benefit from human-like thought processes.[1] The methodology is divided into a preparation stage, performed by humans, and an inference stage. During the preparation stage, the method requires to identify suitable tools and solution processes using those tools. For hybrid QA, this includes search and retrieval over unstructured information, entity linking, and querying over structured information. Solution processes describe how those tools can be combined to find an answer for a particular question, in particular, they also contain *thoughts* in natural language humans have when using those tools. At inference stage, we aim to replicate those solution processes. We use an LLM-based approach in which we provide tools to an LLM and the reasoning is done by generating thoughts about the usage of those tools and the interpretation of their results (also called *observations*). We achieve human-like reasoning by injecting complete solution processes as few-shot examples in the LLM prompt. We adapt the prompt to the input by selecting few-shot examples based on their similarity to the input as well as maximizing the diversity of their tool usage. The method does not require fine-tuning the LLM.

We tested our approach on three different QA benchmarks and observed performance improvements over the used GPT-3.5 and GPT-4 base models as illustrated in Figure 1 (left). An additional benefit over vanilla LLMs is that we also obtain a reasoning trace including used sources and thoughts about observations. We observed large improvements over task-specific baselines (see Section 5) and conducted a human evaluation study showing that users prefer our system over GPT-4, GPT-3.5, and MPT (Figure 1 right).

2 HumanIQ Methodology

The HumanIQ methodology is illustrated in Figure 2. We first cover its task preparation stage. The input to this stage is a set of instances of a task sampled from the training set. Humans proficient at the task then need to decide which tools (also called services or APIs) are most relevant for solving the task. Those tools are then registered at one of the available LLM-tool-augmentation frameworks, e.g., Langchain[2] or OpenAI Function Calling,[3] and connected to the HumanIQ user interface (see Figure 3). The user interface follows the ReAct (Yao et al., 2023) approach dividing steps into actions (= commands for invoking tools), observations (= tool output), and thoughts (= natural language reasoning) until the LLM outputs a final response. We call the combination of all of those steps a *solution process*. The user

[1] Code available at https://github.com/NIMI-research/Human-IQ.

[2] https://www.langchain.com.

[3] https://openai.com/blog/function-calling-and-other-api-updates.

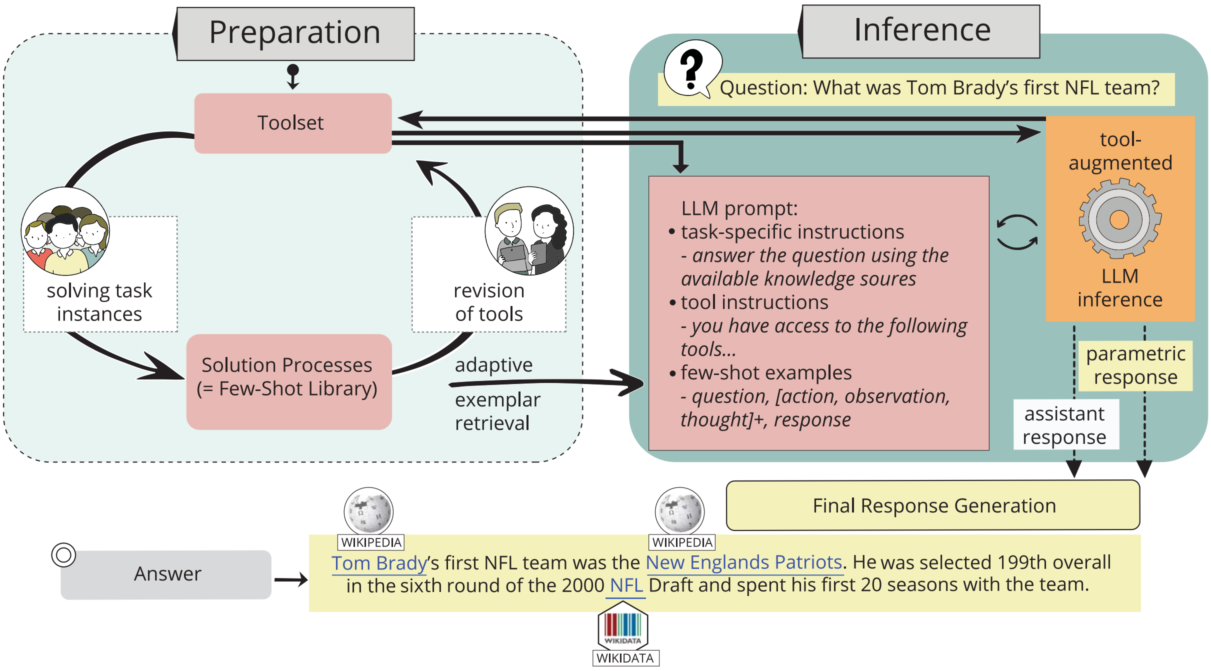
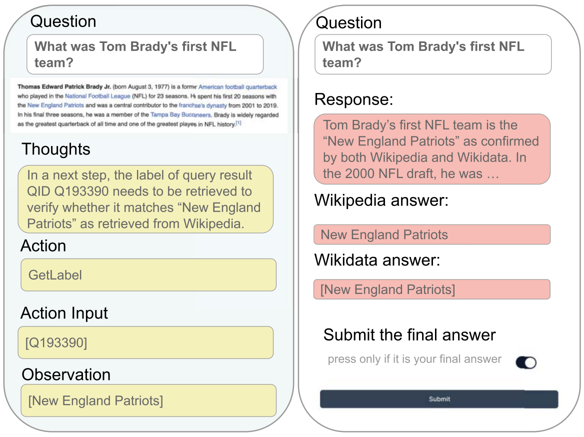

Preparation
Toolset
solving task instances
revision of tools
Solution Processes (= Few-Shot Library)
adaptive exemplar retrieval
Inference
Question: What was Tom Brady's first NFL team?
tool-augmented
LLM inference
LLM prompt:
• task-specific instructions
- answer the question using the available knowledge soures
• tool instructions
- you have access to the following tools...
• few-shot examples
- question, [action, observation, thought]+, response
parametric response
assistant response
Final Response Generation
WIKIPEDIA
WIKIPEDIA
Answer
Tom Brady's first NFL team was the New Englands Patriots. He was selected 199th overall in the sixth round of the 2000 NFL Draft and spent his first 20 seasons with the team.
WIKIDATA

Figure 2: HumanIQ workflow – the parts marked yellow are specific to hybrid question answering (Section 3) and all other elements are part of the generic HumanIQ methodology (Section 2).



Question

What was Tom Brady's first NFL team?

Thomas Edward Patrick Brady Jr. (born August 3, 1977) is a former American football quarterback who played in the National Football League (NFL) for 23 seasons. He spent his first 20 seasons with the New England Patriots and was a central contributor to the franchse's dynasty from 2001 to 2019. In his final three seasons, he was a member of the Tampa Bay Buccaneers. Brady is widely regarded as the greatest quarterback of all time and one of the greatest playes in NFL history.[1]

Thoughts

In a next step, the label of query result QID Q193390 needs to be retrieved to verify whether it matches "New England Patriots" as retrieved from Wikipedia.

Action

GetLabel

Action Input

[Q193390]

Observation

[New England Patriots]

Question

What was Tom Brady's first NFL team?

Response:

Tom Brady's first NFL team is the "New England Patriots" as confirmed by both Wikipedia and Wikidata. In the 2000 NFL draft, he was …

Wikipedia answer:

New England Patriots

Wikidata answer:

[New England Patriots]

Submit the final answer

press only if it is your final answer

Figure 3: HumanIQ user interface for collecting for collecting thoughts and actions from human users.

interface can be applied on concrete instances of a task to construct a human-like solution process. Action sequences can differ substantially for different task instances. For example, some factual questions require more work around entity disambiguation, e.g., distinguishing between the ''The Hunger Games'' movie, movie series, or book. In some cases multiple tries are needed to construct a query returning the desired information or retrieving documents useful for solving a question. While applying the tool-augmented LLM on particular task instances, humans can identify whether particular tools are missing or should be implemented differently to follow a human-like solution process. For example, in this initial phase we added a linker to map from individual documents to entity IDs enabling a bridge between unstructured and structured information sources. This improvement cycle is depicted in the left in Figure 2. Once the tool set is established, a set of solution processes demonstrating desired behavior is constructed resulting in *few-shot library*. To be sample efficient, this can be done in the spirit of active learning using a verify & fix approach: Tool-augmented LLMs can be automatically applied on task instances in the training set and, when those fail, humans can manually build successful solution processes for those instances. During this process, the tool outputs (observations) can be rated (e.g., on a 1–5 Likert scale). This can be used as supervised feedback signal for individual tools.

The second part of the HumanIQ methodology is the inference stage aiming to replicate human solution processes: We use a base LLM that we instruct to follow the general pattern for solution processes as well as announcing the tools that are available along with instructions for solving the task (e.g., ''Answer the question below by accessing the information sources with the given tools!'') and few-shot examples. Each few-shot example contains the entire solution process and can typically span several hundred tokens, i.e., we primarily instruct the model to follow the process, which is similar in spirit to (Lightman et al., 2023), rather than focusing only on output. Given LLM context length limitations, we can usually not include all elements on the few-shot library when applying HumanIQ in an in-context learning setting. To select a set of few-shot examples, we therefore use a Determinantal Point Process (DPP) (Kulesza et al., 2012) to obtain solution processes that are *relevant* and *diverse*. Relevance in our case is measured by the similarity to the input question using SentenceBERT. Intuitively, similar questions should be more relevant demonstrations of the desired behavior. However, using this as sole criterion can lead to selecting examples which all exhibit very similar solution processes. Therefore, we include example diversity as objective. The diversity of two given solution processes is measured via the Levenshtein distance of their actions with each action encoded by a different single symbol. The DPP based on using the above similarity and diversity metrics is then conditioned to a fixed number k of desired few-shot examples. We search the configuration of examples with the highest probability density using a greedy optimiser (Kaushal et al., 2022) and include those in the LLM prompt. Executing the inference stage successfully is a demanding task for LLMs: The model needs to learn human thought patterns by example, understand the intricate relationship between different tools, observations, and needs to support long context as we inject full solution processes with multiple thoughts and tool invocations.

3 HumanIQ Applied to Hybrid QA

We applied the HumanIQ methodology to three hybrid QA[4] benchmarks—Mintaka (Sen et al., 2022), QALD9-Plus (Perevalov et al., 2022) and CompMix (Christmann et al., 2023a)—and call the resulting approaches *HQA*. We use Wikipedia as unstructured and Wikidata (Vrandečić and Krötzsch, 2014) as structured knowledge sources.

Tool specification: We organized two workshops, each with 10 participants with gender and ethnic diversity. All the participants were computer scientists. There was no overlap with the authors of this submission, nor between the two workshop participants. Half of the participants in each workshop were recruited in-house, and half of them were recruited from external institutions. The first workshop yielded tools that helped humans succeed in tasks:

1. wikisearch(): identifying suitable documents given a search query using the Wikipedia API
2. read(): extracting an answer given a document and an input question using GPT-4 with the question and document as input
3. getWikiDataID(): returning the Wikidata ID for a given Wikipedia document
4. generateQuery(): taking a set of relevant entity IDs as input and returning a SPARQL query based on (Lehmann et al., 2023)
5. runQuery(): executing a SPARQL query
6. getLabel(): returning natural language labels for entity IDs

The tools allow for different action sequences to obtain a solution: for example, a query against the knowledge graph can be generated directly, but it is also possible to first identify relevant Wikipedia pages and map those to Wikidata entities, which are then used as the base for query construction.

Solution Process Collection: In the second workshop, we collected 20 solution processes for each benchmark which then constitute the respective few-shot libraries. The libraries are separate for each benchmark to avoid training leakage.

Final Response Generation: In addition to the default HumanIQ workflow, we implemented a final response generation step to combine the evidence obtained from heterogeneous data sources into a final response. This is done using a separate LLM request, which takes the question, the parametric response and the HumanIQ workflow response (summarizing obtained evidence from different sources, thoughts and relevant surrounding factual information) as input. This allows us to instruct the LLM to deliver the desired user experience, e.g., preferring short vs. long answers or including or omitting source attribution for the different parts of the response.

4 Experimental Setup

4.1 Datasets and Base LLMs

In our work, we investigate closed book QA. Therefore, we selected the Mintaka, QALD 9

[4] Code available at `https://github.com/NIMI-research/Hybrid-QA`.

Plus, and CompMix benchmarks, since all of them contain manually written and curated questions—i.e., are not templated or machine generated, and reflect questions answered in a production QA system well. CompMix was specifically developed for hybrid QA across Wikipedia texts, tables, and Infoboxes as well as the Wikidata KG. QALD targets Wikidata specifically. For constructing Mintaka, crowdworkers could freely phrase questions and answers which were afterwards linked to Wikidata. For QALD, we evaluate on the full test set of 150 questions whereas for CompMix and Mintaka we limit our evaluation to a random sample of 200 questions of the test due to the associated effort for manual evaluation. From the GPT model series, we use GPT-3.5-Turbo-16k and GPT-4-0314 via API as base LLMs. To include open models, we also attempted to use the MPT-30B Instruct model,[5] since it is instruction-tuned and allows a large context (8k tokens) but it was not able to follow the provided instructions. We select 3 few-shot examples as described in Section 2 from the given 20 examples per benchmark.

4.2 Baselines

Knowledge Graph QA Models: KVMemNet (Miller et al., 2016) proposes key-value networks for storing knowledge graph triples with corresponding retrieval operations. EmbedKGQA (Saxena et al., 2020) uses an answer scoring module that scores and selects answer entities using a KG embedding and an answer embedding module. Rigel (Saffari et al., 2021) is an end-to-end question answering system using differentiable knowledge graphs. DiFAR (Baek et al., 2023) is an approach for QA via direct fact retrieval from a knowledge graph. Both input question as well as all facts in the knowledge graph are embedded into a dense embedding space. A similarity metric in combination with a reranking approach is then used to retrieve relevant facts.

Text-focused QA Models: DPR (Karpukhin et al., 2020) is a retriever-reader approach using a dense retriever for identifying (Wikipedia) passages for the input question and a reader model to score retrieved passage spans. We use the base reader trained on Natural Questions (Kwiatkowski et al., 2019). For the reader, we report scores from (Sen et al., 2022) for the zero-shot setting and for a model trained on Mintaka.

Hybrid QA Models: CONVINSE (Christmann et al., 2022) is a conversational hybrid question answering system using a frame-like representation to capture evidence from text, knowledge graph and tables with a fusion-in-decoder model for answer generation. UniK-QA (Oguz et al., 2022) is an open-domain hybrid QA system homogenising all sources by converting them into text and then applying a retriever-reader model. EXPLAIGNN (Christmann et al., 2023b) is a conversational hybrid QA system which constructs a heterogeneous graph from entities and evidence snippets retrieved from knowledge graphs, text corpora, tables and infoboxes.

LLMs: T5 (Kwiatkowski et al., 2019), short for Text-To-Text Transfer Transformer, frames NLP problems as text-to-text problems in a consistent and adaptable manner. T5 for closed book QA (Roberts et al., 2020) is an extension fine-tuned as a QA model that can implicitly store and retrieve knowledge. MPT-30B-Instruct (Liu et al., 2024) is an open instruction-tuned language model. GPT3 (Brown et al., 2020), GPT-3.5 and GPT-4 are autoregressive Transformer-based language models by OpenAI.

4.3 Metrics

We compute two metrics per benchmark: The first metric is implemented as described in the benchmark papers (hits@1 exact match for CompMix and Mintaka, Macro F1 for QALD) and allows comparing against other approaches. For generative models, using those measures only provides a rough (under)estimate given that they can return correct responses which do not exactly match the gold standard. For this, we use manual evaluation (marked ''M'') as second metric. We verify all responses manually which are not exact matches. In this process, we also fix incorrect gold standard responses and skip nonsensical questions. We will release the specific instructions for manual evaluation as well all score sheets.

5 Evaluation Results

RQ1: Is the approach competitive against task-specific baselines?

We analyze the performance of HumanIQ against QA baselines in Table 1, Table 2, and Table 3.

[5] `https://huggingface.co/mosaicml/mpt-30b-instruct`.

Model	Hits@1 in %	
	EM	M
KVMemNet	12.0	–
EmbedKGQA	18.0	–
Rigel	20.0	–
DiFAR	34.0	–
DPR (zero-shot)	15.0	–
DPR (trained)	31.0	–
T5	28.0	–
T5 for CBQA (zero-shot)	20.0	–
T5 for CBQA (fine-tuned)	38.0	–
MPT-30B-Instruct*	36.5	47.4
GPT-3.5*	55.0	80.0
GPT-4*	60.5	91.9
HQA-GPT-3.5* (ours)	55.5	85.9
HQA-GPT-4* (ours)	62.0	95.9

Table 1: Response accuracy on the Mintaka data set. The first 9 rows are from Sen et al. (2022) and Baek et al. (2023) and the rest from us. Entries marked with * used 200 random samples from the Mintaka test set.

On all three benchmarks, our approach outperforms task-specific baselines by a wide margin even though some of those were optimized on the training set whereas we only use three few-shot examples (selected from 20 in total). A caveat is that our evaluation was only performed on a random sample of 200 questions for CompMix and Mintaka. However, applying a Z-test comparing the scores with different sample sizes shows that the HQA results are better than all task-specific baselines with high confidence ($p < 0.01$). To a large degree, the performance improvement is due to the capabilities of the base language models in analyzing and reasoning over the different information sources.

RQ2: Is HQA more accurate than base LLMs?

Given the wide performance gap to task-specific approaches, we focused most of our evaluation efforts on comparisons against LLM baselines. Tables 1 to 3 indicate that using the HumanIQ method for retrieval from heterogeneous information sources leads to performance gains over the LLM baselines including the most powerful model GPT-4. Aggregated results are shown in Figure 1 (left) and show that HQA-GPT-4 has a remaining error rate of 8.2%, which is less than half of the GPT-4 error rate. In addition to obtaining higher accuracy than vanilla LLMs, our approach is also explainable in the sense that it (i) pinpoints information sources of evidence and (ii) the thoughts contain explanations of how the final response was derived. However, there is a latency cost for generating additional tokens (575 additional tokens generated on average) and executing tools (5.0 tool calls per question on average).

Model	QALD9-Plus En in %	
	Macro F1	M
DeepPavlov	12.4	–
Platypus	15.0	–
QAnswer	30.4	–
MPT-30B-Instruct	32.0	51.0
GPT 3.5	39.0	75.2
GPT 4.0	43.0	77.5
HQA-GPT-3.5 (ours)	43.0	83.5
HQA-GPT-4 (ours)	50.0	88.4

Table 2: Performance of different approaches on QALD9-Plus EN with the first three rows taken from Perevalov et al. (2022) and the rest by us.

Model	hits@1 in %	
	EM	M
CONVINSE	40.7	–
UniK-QA	44.0	–
EXPLAIGNN	44.2	–
GPT-3.0 (text-davinci-003)	50.2	–
MPT-30B-Instruct	31.5	43.2
GPT 3.5*	53.0	72.3
GPT 4.0*	58.5	80.7
HQA-GPT-3.5* (ours)	61.0	79.6
HQA-GPT-4* (ours)	65.5	91.1

Table 3: Performance on the CompMix data set. The first four rows are taken from Christmann et al. (2023a) and other rows done by us. Entries marked with * used 200 random samples from the CompMix test set.

In addition to measuring performance on benchmarks, we also performed a human evaluation. In this evaluation, we used a different final response generation prompt, which lets the LLM output normal length responses rather than very brief responses potentially matching the gold standard.

For vanilla LLMs, we use exactly the question as input with no further instruction. We employed 10 annotators who are disjoint from the authors and the HumanIQ workshop participants in Section 3 as well as diverse in gender and location. The annotators performed a win rate comparison, i.e., they decided which system generated a better response for a given question. We sampled 50 questions from the test set of each of the 3 benchmarks and compared our strongest system (HQA-GPT-4) against GPT-4, GPT-3.5, MPT and HQA-GPT-3.5. In addition to scoring which response is better, annotators provide the following metrics for each response:

(1) Truthfulness: Rated ‘‘false’’ if any element of the response is factually incorrect and ‘‘true’’ otherwise. The outcome is ‘‘cannot judge’’ if the verification could not be performed within 5 minutes using web search.

(2) Informativeness: Rated as ‘‘not informative’’ if the response provides insufficient information to answer the question, ‘‘somewhat informative’’ if the response provides sufficient information but not further interesting information beyond this and ‘‘informative’’ otherwise.

(3) Quality: This is a compound assessment of several factors with definitions and examples provided to the annotators. The ratings are ‘‘terrible’’, ‘‘poor’’, ‘‘acceptable’’, or ‘‘high’’.

After judging the individual response quality, annotators are asked to decide whether one system response is better or the result is a tie. We duplicated 20% of the comparisons to measure annotator reliability via Cohen’s Kappa. The obtained value of 0.328 indicates ‘‘fair’’ agreement between annotators. Overall, 720 comparisons were made requiring approximately 40 annotator hours of work, most of which was spent on checking factual accuracy. The overall results for response rating are shown in Table 4.

HQA responses were generally judged to be of higher quality and more informative, e.g., 62.8% of HQA-GPT-4 answers are informative compared to 36.7% for GPT-4. This is usually achieved by including surrounding information from the documents it retrieves while keeping a lower hallucination rate than the vanilla models. GPT-4 and GPT-3.5 generated not truthful responses in 11-12% of all cases (approximately 10% when ignoring questions requiring information after the pre-training cutoff). Many hallucinations are not in the fact(s) answering the question, but in the provided additional information. HQA-GPT-4 generated incorrect responses in 5.4% of the cases that were not rated as ‘‘cannot judge’’, which is lower than GPT-4 by a factor of 2.3.

	HQA-4	HQA-3.5	GPT-4	GPT-3.5	MPT-30B
			Truthfulness		
True	91.2	86.7	84.0	80.7	48.0
False	5.2	7.3	12.0	11.3	46.0
Can’t j.	3.7	6.0	4.0	8.0	6.0
			Informativeness		
Inf.	62.8	55.3	36.7	29.5	16.0
Som.inf.	32.3	40.0	57.3	62.4	46.0
Not inf.	4.8	4.7	6.0	8.1	38.0
			Answer quality		
High	55.4	50.7	29.7	25.5	13.3
OK	39.3	40.7	58.8	59.7	34.0
Poor	3.8	8.0	8.1	12.1	28.7
Terrible	1.5	0.7	3.4	2.7	24.0

Table 4: Human evaluation scores in % from overall 720 comparisons of system outputs rated in terms of truthfulness, informativeness and answer quality.

Figure 1 (right) contains the win rates of HQA-GPT-4 against other systems. The average score on the right is obtained by rating a win as 1 point, a tie as 0.5 points and a loss with 0 points and aggregating this on all questions. HQA-GPT-4 is rated significantly above 0.5 for all systems indicating that users prefer its responses on the tested QA benchmarks.

RQ3: How do structured, unstructured and parametric knowledge affect performance?

We analyzed the importance of structured, unstructured and parametric knowledge in obtaining the final solution. To do this, we observed the responses from those sources before the final response generation and report the results in Table 5. All three forms of knowledge are relevant, but play different roles: We almost always obtain a parametric response (recall close to 100) since our benchmarks do not contain input, such as sensitive content, that the LLM is trained not to reply to. The precision of parametric knowledge is on average lower than the information sources. Unstructured information has a higher recall in our experiments than structured information. This is primarily the case because the system is not

Model	Source	Mintaka			QALD			CompMix			Aggregated		
		re	prec	prev	re	prec	prev	re	prec	prev	re	prec	prev
GPT-3.5	stru.	33.3	87.8	20.2	32.4	87.9	27.0	40.9	87.3	26.9	35.5	87.6	24.7
	unstru.	67.6	87.3	47.5	58.5	88.0	48.0	57.5	87.3	43.5	61.2	87.5	46.3
	param.	100.0	79.8	72.7	97.7	70.0	67.2	100.0	74.6	65.2	99.3	74.8	68.3
GPT-4	stru.	52.5	82.6	36.3	62.0	95.6	48.9	65.2	83.3	44.5	59.9	87.1	43.2
	unstru.	81.8	93.8	71.2	69.0	85.7	54.9	71.5	92.0	60.1	74.1	90.5	62.0
	param.	100.0	90.4	84.3	97.8	79.0	64.8	100.0	80.8	65.8	99.2	83.4	71.6

Table 5: Analysis of information sources: For each benchmark, we list how often a particular source of information could provide a response to a question (re = recall), how often it was correct when provided (prec = precision) and how often it was selected in the final response generation step (prev = prevalence). Prevalence can add up to more than 100% since responses from several sources can be equal.

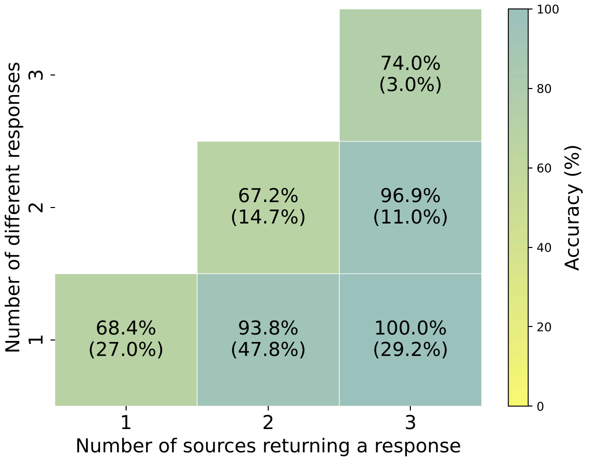


Figure 4: Comparison of HQA-GPT-4 accuracy relative to confirming/disagreeing information sources (frequency in parentheses).

able to construct an executable KG query for each question—either because there is no query that would answer the question or because it cannot find it. The precision of structured and unstructured information is similar in our experiments. For structured information, the main source of imprecision are incorrectly identified entities and incorrectly constructed queries. For unstructured information, the main source of imprecision are incorrectly identified entities. We report more error types in RQ6.

We also investigated the effect of multiple source types (structured/unstructured/parametric) either confirming or disagreeing on a particular response and visualized this in Figure 4. For HQA-GPT-4, in 29.2% of the cases all source types report the same answer. Questions in this category have 100% precision across all datasets compared to an HQA baseline of 91.8% (see Figure 1). Generally, having either two non-conflicting sources or 3 sources with at at most 2 different responses results in high accuracy above the baselines. Other responses can be hedged, e.g., HQA sometimes uses phrases like ‘‘While I could not verify this information, I believe . . . ’’). This makes the risk explicit for the user.

RQ4: Is there a performance benefit of in-prompt integration compared to late fusion?

HumanIQ keeps the solution process for structured and unstructured knowledge retrieval in the same prompt. We want to assess whether this is beneficial by comparing this to separate pipelines. We create a structured pipeline by only allowing it to use the generateQuery(), runQuery(), and getLabel() tools. The unstructured pipeline is allowed to use the search() and read() tools. After the pipelines are executed, we use two types of response generation steps: 1.) If only one of the systems responds, we return this response. If both respond, we randomly select an answer since retrieval from both sources has similar precision. 2.) We use an LLM-based response generation as in our approach, i.e., we give the LLM the result of the structured and unstructured pipelines as well as the parametric generation and ask it to provide a final response. The results are shown in Figure 5. There is a substantial performance delta between using late fusion on separate pipelines versus the HQA approach of applying both in the same prompt. This suggests that evidence obtained from one type of source benefits retrieval for the other type of source.

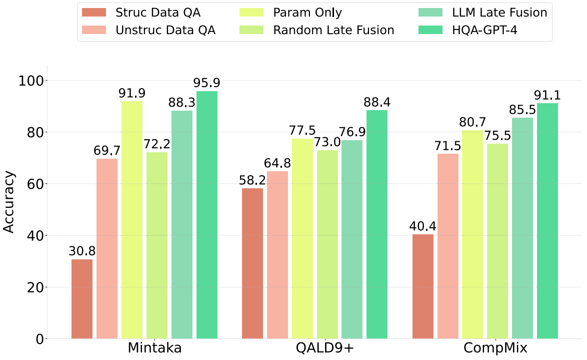


Figure 5: Comparison against baselines which only use either unstructured or structured data and two late fusion approaches. HQA uses in-prompt integration.

RQ5: How do the solution processes as few-shot examples affect performance?

We perform three ablation studies here. 1. Using random few-shot examples instead of using adaptive selection. 2. Using zero-shot prompting instead of few-shot prompting. 3. Using only relevance and only diversity in construction of few-shot examples. Results averaged over all benchmarks are in Figure 6. Generally, we observe that including few-shot examples, which were designed to follow a human-like process, improves performance by 8.4% compared to only giving the LLM general tool usage instructions. We observed that the adaptive few-shot selection plays a significant role increasing performance by 2.8% compared to a random selection, i.e., example similarity and diversity has a relevant impact.

For the third ablation study to analyze the effect of relevance and diversity in few-shot example construction. Keeping the same procedure as before, we observed that examples selected solely based on relevance gave an average hits@1(M%) of 85.03% across all three datasets. Conversely, when diversity was the sole criterion to select the examples, the average hits@1(M%) is 86.4%.

To conclude, our findings indicate that employing an adaptive few-shot selection approach enhances performance significantly, demonstrating a 6.77% improvement over relevance-based selection and a 5.4% increase compared to diversity-based selection.

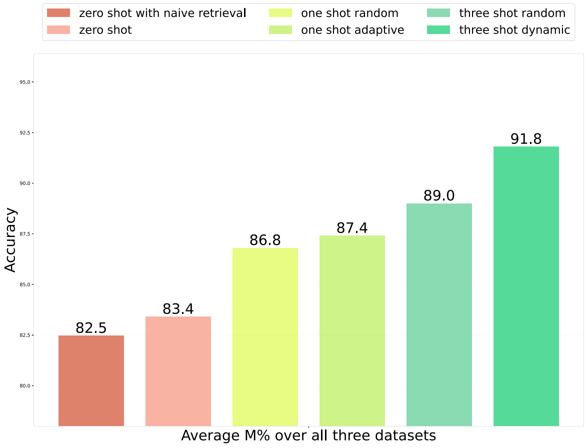


Figure 6: HQA-GPT-4 performance with different few-shot settings (aggregated over all benchmarks).

	Mintaka	QALD9	CompMix
Static	88.8	84.6	87.0
Dynamic	95.9	88.4	91.1
	(+7.1%)	(+3.8%)	(+4.1%)

Table 6: Response accuracy ablation study of HQA-GPT-4 against a static baseline.

RQ6: Is it beneficial to employ an interactive workflow compared to a static workflow?

We aim to quantify whether there is a benefit from following a flexible interactive tool-augmented LLM approach compared to following a static pipeline. The interactive approach has the advantage that it can react based on the observations, i.e., when a call does not deliver the expected information it can retry with a different input or change to a different strategy. However, this type of pipeline does not lend itself easily to parallelization techniques like REWOO (Xu et al., 2023) that allow substantial efficiency improvements and reduce API token consumption by decoupling parametric modules from non-parametric tool calls. We compare our dynamic pipeline with a static pipeline that corresponds to the most frequently used execution pipeline in our setting, which is the sequence search(), getWikidataID(), generateQuery(), runQuery(), getLabel(). The results in Table 6 shows a performance drop of 3.8% to 7.1% response accuracy and affects questions, which require multi-hop inference across several documents or more complex KG queries.

RQ7: How much are results affected by the LLM pre-training cutoff?

Vanilla LLMs can usually not answer questions requiring information beyond their pre-training cutoff. To analyze the effect of this, we manually

	Mintaka P@1 M		QALD9+ P@1 M		CompMix P@1 M	
	HQA-GPT-4	GPT4	HQA-GPT-4	GPT4	HQA-GPT-4	GPT4
Non-transient / before cutoff (539)	96.2	93.0	87.9	79.5	90.9	81.8
Post pre-training cutoff (36)	91.6	75.0	94.4	61.1	100.0	50.0

Table 7: Analyzing the effect of the GPT-4 pre-training cutoff on response accuracy.

marked all questions in our test sets requiring information beyond the September 2021 pre-training cutoff of GPT-3.5 (and GPT-4). We then analyzed the performance of HQA-GPT-4 relative to GPT-4 on this subset as well as all other questions and present the results in Table 7. We observe that there is still a performance improvement of HQA-GPT-4 compared to the vanilla model baseline. For questions requiring recent information, the delta is naturally bigger. GPT-4 could still answer some of those questions correctly, which we attribute to its supervised fine-tuning and learning from human feedback cycles after pre-training which can lead to an integration of newer information in the models.

RQ8: Are the solution processes human-like?

The question can be broken down into several sub-questions: 1. Are the generated action sequences human-like? 2. Are the thoughts generated in the solution process human-like?

In our earlier analysis in Figure 6, we compared the performance of the approach with zero, one and three examples using both random and adaptive settings. Beyond this, we also measured the number of tool calls and generated tokens in solution processes. We observed that the number of tool calls is reduced when we add adaptive few-shot examples: 7.2, 5.9, and 5.0 calls for zero, one, and three examples, respectively. Similarly, the number of generated tokens also decreases (769, 652, 576). For human-created solution processes in the workshops, we measured an average of 6.75 tool calls and 642 used tokens. This indicates the following: 1. Using human-like solution processes leads to a more efficient use of tools (while being more effective as the comparison to the standard ReAct retrieval in Figure 6 shows). 2. This efficiency improvement is not bound by human efficiency since we see no convergence but rather a reduction below the number of tool calls and tokens used in human-created solution processes.

We conducted three additional experiments on solution processes drawing inspiration from Zaitsu and Jin (2023). For comparing human-likeness, we look at three different types of solution processes: human-written (i.e., manually done during the workshops), GPT-4 (standard ReAct approach), and HQA-GPT-4 (including 3 adaptive few-shot examples). For each type, we randomly selected 15 steps (consisting of action, observation and thought) of solution processes per dataset, i.e., overall 45 steps across datasets for each setting. The ‘‘thought’’ part of the solution processes were used for Experiment 1 and 2 while 126 steps with entire solution processes was used for Experiment 3.

Experiment 1. One part of our evaluation involves stylometric analysis. We consider stylometric features of thoughts, in particular (1) bigrams of tokens, (2) frequency of stop words in thoughts (‘‘a’’, ‘‘the’’, ‘‘is’’, etc.), and (3) the use nouns/adjectives (‘‘worse’’, ‘‘better’’, etc.), (4) use of words that describes the narrative flow (‘‘first’’, ‘‘next’’, ‘‘finally’’, etc.). Figure 7 shows that human-written thoughts tend to align more closely with HQA-GPT-4 thoughts in terms of stylometric features. Thoughts extracted from both of the HQA-GPT-4 as well as the human-written thoughts are distinct from GPT-4 thoughts.

Experiment 2. In this experiment, we compared the thought distribution for different pairs of sources, specifically GPT4-H (GPT-4 vs human-written) and HQA-H (HQA-GPT-4 vs human-written). For this, we used Jensen-Shannon divergence distance (JSD) for comparison, with lower values indicating higher similarity in the distribution. JSD is defined as follows:

$$\mathrm{JSD}(P \parallel Q) = \sqrt{\frac{1}{2}\mathrm{D_{KL}}(P \parallel M) + \frac{1}{2}\mathrm{D_{KL}}(Q \parallel M)} \quad (1)$$

where P and Q are the two probability distributions obtained by the softmax of the BERT embeddings of the compared texts. For example, for the GPT4-H comparison, P represents the embedding distribution of GPT-4 generated

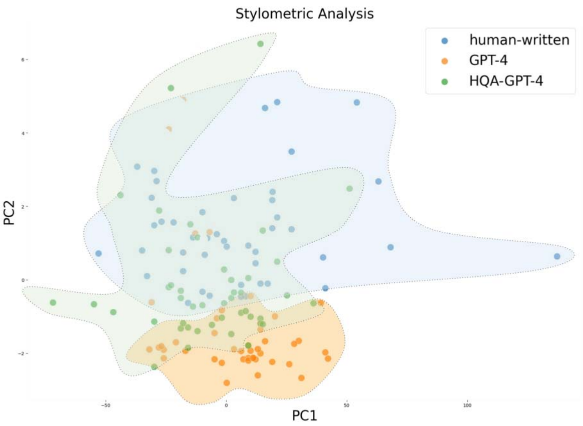


Figure 7: Plot of a stylometric analysis with each point representing a sample from three types of thoughts: human-written, GPT-4 generated, and HQA-GPT-4 generated. The stylometric input features were reduced to two dimensions (PC1, PC2) via principal component analysis. HQA-GPT-4 generated thoughts show a closer resemblance to human thoughts compared to GPT-4 generated thoughts.

n	GPT4-H	HQA-H	Δ(%)
1	0.0889	0.0897	−0.89
≤ 2	0.0700	0.0674	3.71
≤3	0.0630	0.0618	1.90
≤4	0.0600	0.0576	4.00
≤5	0.0579	0.0549	5.18
≤6	0.0570	0.0540	5.26
≤7	0.0569	0.0537	5.62
≤8	0.0568	0.0535	5.80

Table 8: Jensen-Shannon Divergence Scores for GPT4-H and HQA-H outputs where n is the n-th thought in the sequence.

text and Q represents the embedding distribution from human-written text. $M = \frac{1}{2}(P + Q)$ is the midpoint distribution defined and D_{KL} denotes the Kullback-Leibler divergence between two distributions.

For each pair of distributions the JSD scores are calculated. The average JSD scores across all pairs of embeddings in the solution process are shown in Table 8 where n is the n-th thought in the sequence. The table shows a trend of decreasing JSD scores, i.e. scores later in the solution process are closer to human-written ones.

Additionally, JSD scores for HQA-H are smaller than those of GPT4-H except for the first thought in the sequence ($n = 1$). This could imply that human-like few-shot examples have an influence on making the AI-generated thoughts more closely resemble human thought patterns, as represented by the embeddings.

Error Category	Count
(1) Lack of support for tables	6
(2) SPARQL query generation	4
(3) Entity selection	2
(4) Iteration limit error	2
(5) Final response generation step	9
(6) Search: incorrect document picked	20
(7) Answer not found in document	5

Table 9: Error categories observed across all test sets.

Experiment 3. Seven independent annotators, distinct from the authors, were engaged for this experiment. The analysis involves 126 samples across three datasets, each containing complete solution processes generated by both HQA-GPT-4 and GPT-4. Their task was to assess all three types of solution processes, i.e., human written, GPT-4 generated, HQA-GPT-4 generated, side-by-side in terms of stylistic features, coherence and plausibility of thoughts. Specifically, the annotators could see the solution processes of two systems A_1 and A_2 which were taken from HQA-GPT-4 and GPT-4 in random order and then had to judge which of those has a closer proximity to a system B which were human-written solution processes. They were instructed to pay attention to whether the thoughts reflect and analyse the observations (i.e. tool outputs) and provide a coherent chain of reasoning. The analysis of 126 samples demonstrated a preference for HQA-GPT-4 over GPT-4 in 67.4% of instances, indicating that HQA-GPT-4 aligned with the human-written processes. This preference is statistically significant (p-value of < 0.0001 in a sign test).

To calculate the Cohen's kappa score for inter-rater agreement, 27% of the samples were double annotated which resulted in a score of 0.71 indicating ‘‘substantial’’ agreement between annotators.

RQ9: What types of errors do we observe?

Here, we qualitatively analyze the errors made by HQA-GPT-4 and group them into categories as shown in Table 9. The main source of error is the search tool, i.e., a document to answer the question exists but was not found.

The second largest error bucket is not selecting the correct response in the final response generation step followed by the lack of support for tables. We aim to address this via more powerful tools in the future.

6 Related Work

The main areas related to our work are early/late fusion approaches for hybrid QA, and information integration and human-like thoughts in LLMs.

Late Fusion: In late fusion (Sawant et al., 2019; Savenkov and Agichtein, 2016; Xu et al., 2016b) over a set of heterogeneous information sources, a pool of candidate answers is collected using QA pipelines for each type of source. The answers of those pipelines are then combined in a final merge step, e.g., employing a waterfall model based on individual pipeline result precision. Several late fusion models focus on one type as the primary source and use the other source to gather supporting evidence. The final merge step usually only considers the final responses of the individual pipelines and not the evidence collected within their processing steps. Generally, a drawback of those methods is that evidence in one type of knowledge source does not influence the inference steps for other types of information sources (Sun et al., 2018). Some approaches perform an integration via a query language that supports text retrieval (Bahmid and Zouaq, 2018; Xu et al., 2016a; Usbeck et al., 2015). While this allows for a more direct integration of sources, the responses are restricted by the expressiveness of the query language and the text retrieval, which is usually limited to simple facts.

Early Fusion / Unified Representations: In early fusion approaches (Sun et al., 2019; Wang et al., 2022) different types of sources are combined at an early pipeline stage to find an answer to an input question. The most common way to achieve this is to build unified representations, i.e., represent all information sources in a uniform way; most commonly (i) converting facts in a knowledge graph to text or (ii) converting or integrating text in a KG. GRAFT-Net (Sun et al., 2018) allows to query information from a KG and textual sources linked from its entities. They construct a query subgraph for an input question and then apply a graph convolution based neural network. Pramanik et al. (2023) build context graphs from KG and text inputs on the fly. An advantage of their method is that it allows to run complex queries over the context graphs. However, the graph construction from text is in itself a challenging task with high error rates. Oguz et al. (2022) flatten a KG to text and leverage dense passage retrieval (DPR) in combination with fusion-in-decoder (FiD) to pass facts into an LLM. While this approach is promising, it does not allow to use queries over the input KG. Generally, early fusion approaches have the drawback that a) building a unified representation can lead to errors that propagate into the QA approach and b) the selected representation is often the least common denominator of the input sources and does not exploit source-specific advantages such as compositional queries for structured data.

Information Integration and Human-like Reasoning for LLMs: Retrieval-augmented language models (Guu et al., 2020; Borgeaud et al., 2022) enable the integration of external information into language models but do not allow querying structured information. API-augmented language models allow to connect to different services, including text retrieval and graph querying, by letting the LLM generate actions and feeding the output of those actions back into the LLM context. Specifically, we use ReAct (Yao et al., 2023) as the API augmentation framework in our approach. Several related frameworks have been developed over the past years. For example, the recently introduced DSPy approach (Khattab et al., 2022, 2023) provides an abstraction layer over LLM calls and retrieval modules. DSPy was developed in parallel to HumanIQ and focuses mostly on automatic few-shot generation and optimization whereas HumanIQ focuses on improving inference by appropriately selecting human few-shot examples. Combining both may be an interesting avenue for future work. Chameleon (Lu et al., 2024) is another LLM-based planner, which supports a wide range of tools and tasks. A difference to HumanIQ is that our few-shot prompts include complete solution processes and the reasoning about which tool to invoke with what argument is made during inference whereas Chameleon generates a program consisting of the sequence of tools to invoke in advance. Several LLM reasoning frameworks have been applied to question answering. In Trivedi et al. (2023), similar to our approach, retrieval and action generation are interleaved for QA tasks based on the observation that the next action can depend on the actual outcome of the previous retrieval step. A difference to our method is that we investigate the use of full solution processes in the LLM prompt and support heterogenous retrieval including structured knowledge. Dua et al. (2022) and Khot Khot et al. (2022) use LLMs for decomposing complex questions. In our case decomposition is done implicitly by the provided LLM instructions and few-shot examples.

We are also inspired by using human-like reasoning techniques, since those were shown to be very sample-efficient (Hu and Clune, 2024). The integration of contextual knowledge via in-context learning has been shown to work particularly well for very large language models (Wei et al., 2023), i.e., it appears to be an emergent property of LLMs. The implication of those insight for our approach, which was also confirmed by our experiments, is that this type of information integration and reasoning via in-context learning is primarily applicable in today's most powerful LLMs.

7 Conclusions and Future Work

We introduced the HumanIQ methodology and showed that applying it to hybrid question-answering allows reasoning over heterogeneous information sources in the LLM prompt leading to state-of-the-art results on three benchmarks. Users preferred the system over vanilla LLMs in a human evaluation study. A natural next step is to apply HumanIQ in other settings than factual QA. Furthermore, we aim to investigate prompt compression and memory techniques. Another interesting direction are multilingual NLP tasks, in particular retrieving from sources in different languages and using this to build evidence in the required target language.

Acknowledgments

We thank the reviewers and editors for their feedback on the submission. We also acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by Sächsisches Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research ‘‘Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig’’ (ScaDS.AI). The work is also supported by the SECAI project (grant 57616814) funded by DAAD (German Academic Exchange Service). We also thank the Center for Information Services and High Performance Computing [Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH)] at TU Dresden for providing its facilities for high throughput calculations.

References

Jinheon Baek, Alham Fikri Aji, Jens Lehmann, and Sung Ju Hwang. 2023. Direct fact retrieval from knowledge graphs without entity linking. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10038–10055. https://doi.org/10.18653/v1/2023.acl-long.558

Rawan Bahmid and Amal Zouaq. 2018. Hybrid question answering using heuristic methods and linked data schema. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 446–451. IEEE. https://doi.org/10.1109/WI.2018.00-54

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances* *in Neural Information Processing Systems*, 33:1877–1901.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational question answering on heterogeneous sources. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 144–154. https://doi.org/10.1145/3477495.3531815

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2023a. Compmix: A benchmark for heterogeneous question answering. *arXiv preprint arXiv:2306.12235*.

Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2023b. Explainable conversational question answering over heterogeneous sources via iterative graph neural networks. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 643–653. https://doi.org/10.1145/3539618.3591682

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. In *Successive prompting for decomposing complex questions*. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.81

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Shengran Hu and Jeff Clune. 2024. Thought cloning: Learning to think while acting by imitating human thinking. *Advances in Neural Information Processing Systems*, 36.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Vishal Kaushal, Ganesh Ramakrishnan, and Rishabh Iyer. 2022. Submodlib: A submodular optimization library. *arXiv preprint arXiv:2202.10680*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286. https://doi.org/10.1561/9781601986290

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466. https://doi.org/10.1162/tacl_a_00276

Jens Lehmann, Preetam Gattogi, Dhananjay Bhandiwad, Sébastien Ferré, and Sahar Vahdati. 2023. Language models as controlled natural language semantic parsers for knowledge graph question answering. In *European Conference on Artificial Intelligence (ECAI)*. IOS Press. https://doi.org/10.3233/FAIA230411

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan

Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12. https://doi.org/10.1162/tacl_a_00638

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1147

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546. https://doi.org/10.18653/v1/2022.findings-naacl.115

Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pages 229–234. IEEE. https://doi.org/10.1109/ICSC52841.2022.00045

Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, and Gerhard Weikum. 2023. Uniqorn: Unified question answering over rdf knowledge graphs and natural language text. *arXiv preprint arXiv:2108.08614*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426. https://doi.org/10.18653/v1/2020.emnlp-main.437

Amir Saffari, Armin Oliya, Priyanka Sen, and Tom Ayoola. 2021. End-to-end entity resolution and question answering using differentiable knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4193–4200. https://doi.org/10.18653/v1/2021.emnlp-main.345

Denis Savenkov and Eugene Agichtein. 2016. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 235–244. https://doi.org/10.1145/2911451.2911536

Uma Sawant, Saurabh Garg, Soumen Chakrabarti, and Ganesh Ramakrishnan. 2019. Neural architecture for question answering using a knowledge graph and web corpus. *Information Retrieval Journal*, 22:324–349. https://doi.org/10.1007/s10791-018-9348-8

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507. https://doi.org/10.18653/v1/2020.acl-main.412

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. 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)*, pages 2380–2390. https://doi.org/10.18653/v1/D19-1242

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 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*, pages 4231–4242. https://doi.org/10.18653/v1/D18-1455

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: How knowledgeable are large language models (llm)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.557

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, and Christina Unger. 2015. Hawk–hybrid question answering using linked data. In *The Semantic Web. Latest Advances and New Domains: 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31–June 4, 2015. Proceedings 12*, pages 353–368. Springer. https://doi.org/10.1007/978-3-319-18818-8_22

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85. https://doi.org/10.1145/2629489

Yingyao Wang, Junwei Bao, Chaoqun Duan, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022. Muger2: Multi-granularity evidence retrieval and reasoning for hybrid question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6687–6697. https://doi.org/10.18653/v1/2022.findings-emnlp.498

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016a. Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2397–2407.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016b. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336. https://doi.org/10.18653/v1/P16-1220

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Wataru Zaitsu and Mingzhe Jin. 2023. Distinguishing chatgpt(-3.5, -4)-generated and human-written papers through Japanese stylometric analysis. *PLOS ONE*. https://doi.org/10.1371/journal.pone.0288453, PubMed: 37556434