EWEK-QA: Enhanced Web and Efficient Knowledge Graph Retrieval for Citation-based Question Answering Systems

Mohammad Dehghan[1], Mohammad Ali Alomrani[1], Sunyam Bagga[1],
David Alfonso-Hermelo[1], Khalil Bibi[1], Abbas Ghaddar[1], Yingxue Zhang[1],
Xiaoguang Li[1], Jianye Hao[1], Qun Liu[1], Jimmy Lin[2], Boxing Chen[1],
Prasanna Parthasarathi[1], Mahdi Biparva[1], Mehdi Rezagholizadeh[1]
[1] Huawei Noah's Ark Lab, [2] University of Waterloo
{mohammad.dehghan@uwaterloo.ca, mehdi.rezagholizadeh@huawei.com}

Abstract

The emerging *citation-based QA* systems are gaining more attention especially in generative AI search applications. The importance of extracted knowledge provided to these systems is vital from both accuracy (completeness of information) and efficiency (extracting the information in a timely manner). In this regard, citation-based QA systems are suffering from two shortcomings. First, they usually rely only on web as a source of extracted knowledge and adding other external knowledge sources can hamper the efficiency of the system. Second, web-retrieved contents are usually obtained by some simple heuristics such as fixed length or breakpoints which might lead to splitting information into pieces. To mitigate these issues, we propose our enhanced web and efficient knowledge graph (KG) retrieval solution (EWEK-QA) to enrich the content of the extracted knowledge fed to the system. This has been done through designing an adaptive web retriever and incorporating KGs triples in an efficient manner. We demonstrate the effectiveness of EWEK-QA over the open-source state-of-the-art (SoTA) web-based and KG baseline models using a comprehensive set of quantitative and human evaluation experiments. Our model is able to: first, improve the web-retriever baseline in terms of extracting more relevant passages (>20%), the coverage of answer span (>25%) and self containment (>35%); second, obtain and integrate KG triples into its pipeline very efficiently (by avoiding any LLM calls) to outperform the web-only and KG-only SoTA baselines significantly in 7 quantitative QA tasks and our human evaluation. [1].

1 Introduction

Large language models (LLMs) have shown great potentials to be used for question answering (QA) (Tan et al., 2023) tasks. However, relying only on the internal knowledge (gained from pre-training or fine-tuning) of LLMs for answering questions may lead to issues such as hallucination, lack of knowledge, or outdated knowledge (Gao et al., 2023b). To address these problems, retrieval augmented generation (RAG) (Gao et al., 2024) can be leveraged to assist with grounding the answer of LLMs to some external knowledge bases such as web or knowledge graphs (KGs). While this approach can be very useful in practice to reduce the hallucination of LLMs (Huang and Yu, 2023), it still remains challenging to identify which parts of the answer come from the external knowledge or internal knowledge of LLMs (i.e. knowledge grounding).

Citation-based QA systems, such as generative

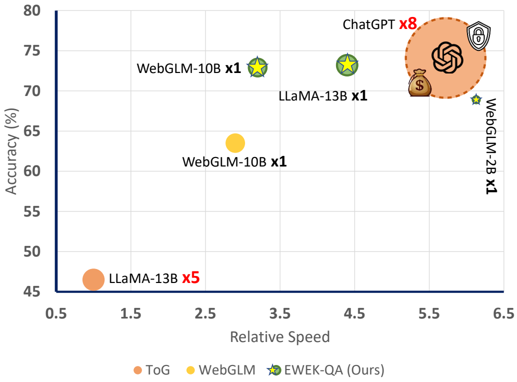


Figure 1: Overview of performance vs. efficiency of EWEK-QA (KG+Web), WebGLM (Web-only), and ToG (KG-only) on the WebQSP dataset (See Table 4 for details). Each circle represents one solution with its LLM's name and the number of calls to the LLM (indicated as $\times n$). The size of each circle indicates the relative size of its corresponding backbone LLM. The relative speed represents the speed with respect to ToG with LLaMA-2-13B. Bear in mind that ToG with ChatGPT needs to call the closed-source ChatGPT system 8 times on average, which can increase the expenses and also raise privacy concerns for sensitive applications.

[1] The codes of this work will be available at https://github.com/huawei-noah/Efficient-NLP/tree/main/EWEK-QA.

AI search applications (e.g. Microsoft's new Bing [2] or YOU.com [3]), aim at addressing the knowledge grounding issue by adding proper citations from relevant retrieved passages (so-called *quotes* in this paper hereafter) to their answer.

In this regard, instruction-tuned LLMs learn to give citations by supervised fine-tuning or in-context learning (Liu et al., 2023).

Moreover, considering the abundant number of users and queries to these citation-based QA systems, the whole pipeline should be designed to run very efficiently while providing accurate answers.

A case in point is WebGLM (Liu et al., 2023)

which is an efficient web-enhanced question answering system based on the 10B GLM model (Du et al., 2022). To the best of our knowledge, WebGLM is the first of its kind to efficiently use open-source models for QA systems with citation capability. In this paper, we aim to improve the accuracy of WebGLM while keeping its efficiency. While WebGLM (Liu et al., 2023) outperforms a similar size (13B) WebGPT model (Nakano et al., 2022) significantly and works on-par with the large WebGPT model (175B), there remains some major challenges to deal with. First, WebGLM only relies on the web as a source of external knowledge (Liu et al., 2023), which might not be sufficient on its own for answering a diverse set of questions (e.g. multi-hop reasoning questions (Yang et al., 2018b) or knowledge graph question answering (KGQA) (Perevalov et al., 2022) tasks). Second, its web-retrieval module, usually breaks the pages based on some simple heuristics such as length or breakpoints (to form the quotes), which can give rise to splitting complete information into independent pieces.

To address these problems, we propose our *E*nhanced *W*eb and *E*fficient *K*nowledge graph retrieval for citation-based *QA* systems (EWEK-QA) which tries to enrich the content of the extracted quotes fed to the LLM in WebGLM through incorporating KGs and extracting adaptive quotes rather than fixed-length quotes from the web. It is worth mentioning that state-of-the-art (SoTA) KGQA techniques (Sun et al., 2024; Luo et al., 2024), which extract informative triples from KGs, require several calls to LLMs. In this regard, using open-source LLMs can significantly increase the end-to-end latency of the models and using closed-source LLMs (such as ChatGPT) can bring-up extensive costs and privacy concerns (see Fig. 1). Additionally, the sheer size of modern KGs (e.g. Freebase) makes it challenging to efficiently extract the most relevant sub-graphs. Hence, our solution focuses on retrieving the most informative triples from KGs with *minimal* calls to the LLMs (to maintain the efficiency of the entire pipeline).

We evaluate our EWEK-QA using several qualitative (i.e. human evaluations) and quantitative experiments on different types of QA tasks such as open-domain QA (ODQA), multi-hop reasoning, and KGQA. The results show that our adaptive web-retriever

can significantly improve the quality of extracted quotes in terms of relevance to the queries, coverage of the answer span, and self-containment (i.e. containing complete information to answer questions). EWEK-QA with its efficient graph retriever and adaptive web-retriever is able to outperform both WebGLM and SoTA Think-on-Graph (ToG) (Sun et al., 2024) significantly on KGQA and ODQA datasets by >10% and >3% on average respectively. Moreover, EWEK-QA achieves between $\sim 3\times$ to $\sim 6\times$ speedup compared to ToG when using open-source LLaMA-2-13B model (Touvron et al., 2023) (see Fig. 1). Finally, our human evaluation shows that EWEK-QA answers questions >20% more accurately than SoTA baselines. The results indicate the importance of combining web-extracted knowledge with KG-extracted triples in designing citation-based QA systems. Our contributions are summarized as follows:

- We propose an efficient citation-based QA system that utilizes two external knowledge modalities: web text, and KGs simultaneously without hampering the efficiency. Our efficient KG extraction module does not use any open-source or close-source LLM calls but still the triplets provide valuable information to the system.
- EWEK-QA introduces an adaptive web-retrieval module which is able to extract more informative and more relevant quotes.
- We demonstrate that our solution is able to outperform KG-only and Web-only QA baselines in wider range of QA tasks such as KGQA, ODQA, and multi-hop reasoning datasets

[2] https://www.microsoft.com/en-us/edge/features/bing-chat?form=MA13FJ

[3] https://you.com/?chatMode=default

based on comprehensive quantitative and human evaluations.

2 Related Work

Citation-based Question Answering Systems Citation-based QA systems can be viewed as an enhanced version of regular retrieval augmented generation (RAG) solutions (such as REALM (Guu et al., 2020), RAG (Lewis et al., 2020), and Atlas (Izacard et al., 2023)) which are able to add citation from relevant retrieved quotes during the answer generation. RAG are able to integrate external knowledge into the generation process; however, they cannot add citation to the answers. WebGPT (Nakano et al., 2022) is one of the pioneering works on citation-based QA which fine-tuned GPT-3 (175B) to answer open-domain questions using web by browsing through most relevant pages and adding references to the answers from the relevant pages. GopherCite (Menick et al., 2022) is another case in point where a 280B model LLM was fine-tuned using reinforcement learning based on human preference to generate answers with proper citations. Although the models such as WebGPT and GopherCite rely on the power of large scale LLMs, WebGLM (Liu et al., 2023) introduced an efficient citation-based QA approach based on fine-tuning much smaller LLMs. WebGLM showed better performance compared to a similar size (13B) WebGPT model (Nakano et al., 2022) significantly and works on-par with the large WebGPT model (175B). To the best of our knowledge, WebGLM is the first of its kind to efficiently use open-source models for QA systems with citation capability. We found WebGLM as the most relevant work to ours and we keep that as one of our main SoTA baselines. We aim at improving the accuracy of WebGLM while keeping its efficiency.

Knowledge Graph-Augmentation for Reasoning in LLMs Many recent works leverage the rich and structured knowledge of KGs to mitigate the hallucination and reliability issues of LLMs. Early studies (Yasunaga et al., 2021; Zhang et al., 2021; Yasunaga et al., 2022) integrate KG embedding methods using GNNs with LLM models during the finetuning or pre-training stage. While such approaches have shown promising results in explainability and reasoning, they tend to require extensive task-specific finetuning.

Another line of works combines external knowledge from KGs into LLMs during the prompting stage. KB-Binder (Li et al., 2023) leverages the Codex LLM (Chen et al., 2021) to generate a SPARQL query that extracts the answer from the KG. KAPING (Baek et al., 2023) retrieves the most relevant one-hop KG triples via dense retrieval models and provides them to the LLM as context. KD-CoT (Wang et al., 2023) proposes to verify and modify CoT reasoning traces with KG knowledge in order to alleviate hallucination and error propagation. MindMap (Wen et al., 2023) builds a prompting pipeline where the LLM reasons over the extracted KG subgraphs and generates answers grounded by the "reasoning pathways" within the KG. RoG (Luo et al., 2024) uses an LLM to retrieve reasoning paths from the KGs based on relation "plans" grounded by KGs. ToG (Sun et al., 2024) performs beam search on KGs using LLMs to dynamically extract the most relevant reasoning paths. While it displays impressive performance on KGQA datasets, it requires many LLM calls per questions, and it degrades on open-domain datasets where the answer may not exist in the KG. Additionally, it heavily relies on closed-source LLMs (e.g. ChatGPT) for SoTA performance.

3 Methodology

Our approach consists of two main components: Knowledge Extraction (§3.1) and Answer Composition (§3.2), which operate sequentially. When presented with a question, we initiate the knowledge extraction phase to gather the pertinent information required for answering. Subsequently, we employ an open-source LLM to generate a cohesive final answer *solely* based on the collected information. See Fig. 2 for the full pipeline.

3.1 Knowledge Extraction: Web and KG

We concurrently extract information from two distinct knowledge sources – Web (§3.1.1) and Knowledge Graph (§3.1.2) – to gather external data for addressing a given query.

3.1.1 Adaptive Web Retrieval

We introduce an adaptive module designed to extract pertinent passages (referred to as quotes) from unstructured web text (refer to Figure 5 in the Appendix). Initially, relevant webpages are retrieved using the Bing search engine, followed by a multi-step process to extract quotes, as detailed below. Our approach is termed "adaptive" because it integrates a heuristic-based parser, the Paragraph Split-

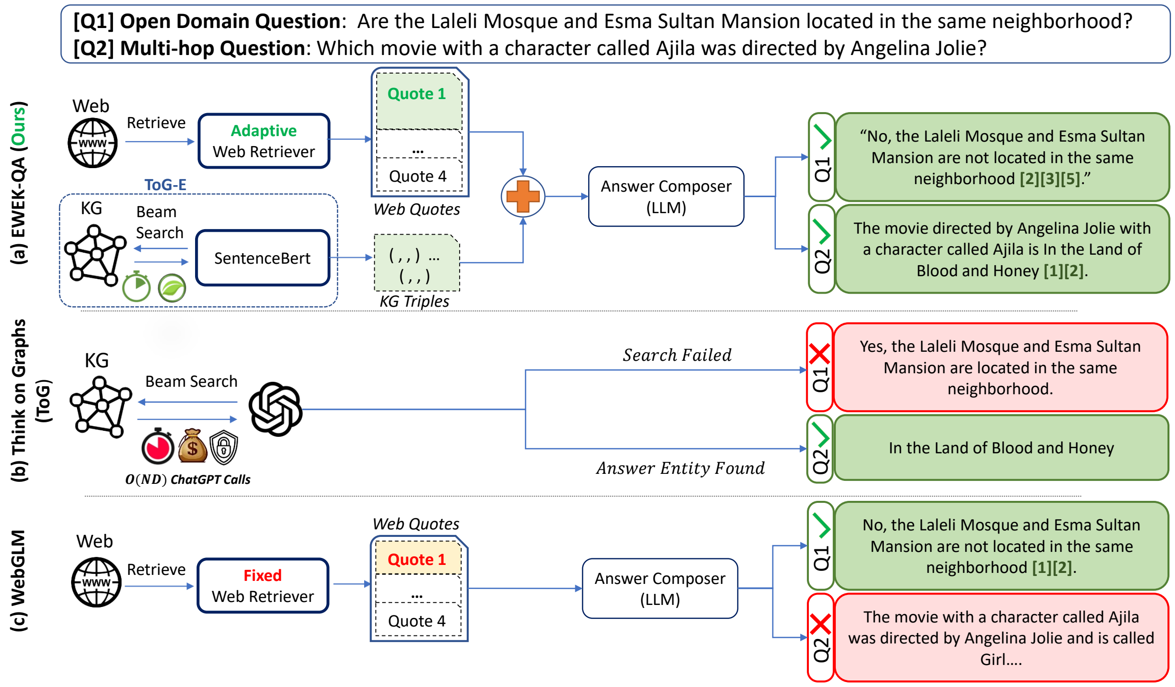

[Q1] Open Domain Question: Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?
[Q2] Multi-hop Question: Which movie with a character called Ajila was directed by Angelina Jolie?
(a) EWEK-QA (Ours)
Web
Retrieve
Adaptive Web Retriever
Quote 1
...
Quote 4
Web Quotes
ToG-E
KG
Beam Search
SentenceBert
(, ,) ...
(, ,)
KG Triples
Answer Composer (LLM)
Q1
"No, the Laleli Mosque and Esma Sultan Mansion are not located in the same neighborhood [2][3][5]."
Q2
The movie directed by Angelina Jolie with a character called Ajila is In the Land of Blood and Honey [1][2].
(b) Think on Graphs (ToG)
KG
Beam Search
O(ND) ChatGPT Calls
Search Failed
Answer Entity Found
Q1
Yes, the Laleli Mosque and Esma Sultan Mansion are located in the same neighborhood.
Q2
In the Land of Blood and Honey
(c) WebGLM
Web
Retrieve
Fixed Web Retriever
Web Quotes
Quote 1
...
Quote 4
Answer Composer (LLM)
Q1
No, the Laleli Mosque and Esma Sultan Mansion are not located in the same neighborhood [1][2].
Q2
The movie with a character called Ajila was directed by Angelina Jolie and is called Girl....

Figure 2: Comparison of EWEK-QA , ToG (Sun et al., 2024), and WebGLM (Liu et al., 2023) pipelines. EWEK-QA utilizes both knowledge modalities which enables to correctly answer both question types using a single LLM call. ToG requires $O(ND)$ costly calls where N and D represent the beam search width and depth respectively. WebGLM relies solely on the web which makes it unfit for multi-hop reasoning questions.

ter (PS), with a small language model, the Evidence Extractor (EE). This dynamic extraction process adjusts according to both the specific query and the format of the webpage, managed by PS and EE respectively. Once webpages are transformed into candidate quotes, our Retrieve and Rerank module selects the most relevant quotes, followed by removal of redundant quotes by a deduplicator module. We provide a detailed explanation of each module below.

Paragraph Splitter (PS). This module is similar to the WebGLM retriever. As in Liu et al. (2023), we divide the webpage contents into a list of candidate passages using line breaks. We apply additional constraints to further improve the quality of the quotes: we utilise <p> tags in the webpage's HTML to produce candidate passages; any passage with less than 10 tokens is discarded; any passage with more than 80 tokens is broken down into shorter passages while respecting sentence boundaries (see §A.7 for more details).

Evidence Extractor (EE). The task is similar to the machine reading comprehension (MRC) (Devlin et al., 2019) task. Instead of pursuing an answer span (like in MRC), the target here is to extract *evidence spans* from the webpage contents that can provide support to answer the question. We fine-tune a pre-trained MRC model – DeBERTa (He et al., 2021) – on the MS Marco dataset (Bajaj et al., 2016) to identify text spans from webpages that are relevant to the user query (see §A.7.2 for more details).

Retrieve and Rerank. Given all the candidate quotes produced by PS and EE, we retrieve and re-rank them based on semantic relevance using two cross-encoder models of different sizes: (1) a small filtration model to remove irrelevant quotes and (2) a larger cross-encoder model to re-rank the filtered passages. Specifically, we use a six-layer MiniLM (Wang et al., 2020) with $22M$ parameters as the filtration model and a large DeBERTa (He et al., 2021) with $900M$ parameters as the re-ranker model. Both models are trained on the MS Marco dataset for the passage ranking task.

Deduplicator. Since both PS and EE extract quotes from the same webpages, there is a need to eliminate duplicate quotes. We use a small bi-encoder six-layer MiniLM to produce sentence embeddings for each passage and compute the cosine similarity between each pair of embeddings. Any passage with cosine similarity > 0.9 is removed.

3.1.2 ToG-E: Sub-graph Retrieval

Knowledge Graphs (KGs) serve as structured and dynamic repositories of information. Integrating LLMs with KGs presents an adjunctive strategy to mitigate LLM's limitations in answering multi-hop reasoning questions (Sun et al., 2024). Prior KG sub-graph extraction modules demonstrated enhanced effectiveness across diverse question types (Wang et al., 2023; Luo et al., 2024).

Adhering to the Think-on-Graph (ToG) methodology (Sun et al., 2024), we employ beam search (Jurafsky and Martin, 2000) on the KG to extract a relevant sub-graph given a question. In a nutshell, ToG iteratively invokes an LLM to explore possible reasoning paths on the KG until the LLM determines that the question can be answered based on the current reasoning paths. At each iteration, ToG constantly updates the top-N reasoning paths until a max depth D is reached. To enhance this execution, we introduce an efficient variant of the ToG method, denoted as ToG-E. In contrast to the original ToG methodology, the ToG-E returns a sub-graph in the form of "entity, relation, entity" triples without invoking any LLM.

Three primary distinctions characterize ToG-E in comparison to ToG: 1) During the pruning step, ToG relies on an LLM to acquire scores for candidate relations and entities in the beam search, whereas ToG-E utilizes SentenceBert (Reimers and Gurevych, 2019) embeddings of the question, relations, and entities to compute cosine similarity scores for each relation and entity. 2) ToG-E omits a reasoning step employed by ToG to halt before reaching the maximum depth in the beam search. 3) The ToG-E method produces a sub-graph as its output, regardless of whether the extracted sub-graph is informative or not. In contrast, the ToG methodology validates the extracted sub-graph (triples) by engaging an LLM and may further prompt the LLM with a CoT (Wei et al., 2022) instruction. This additional step aims to elicit an answer solely based on the parametric knowledge of the LLM in case the sub-graph is deemed to lack sufficient informativeness.

In contrast to ToG, which relies heavily on closed-source LLMs such as ChatGPT to achieve effectiveness, our approach sidesteps the use of any LLM during the triple extraction process from the KG (see Figure 2). Furthermore, our system does not require any KGQA supervised dataset for training or fine-tuning. As a result, we compare our method with prompt-based approaches.

3.2 Answer Composition

After extracting KG triples and web quotes, we utilize an open-source pre-trained LLM to process this data and construct a coherent response, supplemented with citations to relevant knowledge sources. The KG triples constitute the initial passage, while the subsequent passages consist of web quotes, all serving as inputs to the answer composer model (see §A.4 for details).

In our experiments, we employed the WebGLM-10B model (Liu et al., 2023) for answer composition by default. This model has been fine-tuned specifically for the task of composing answers: given a question and a set of text passages (5 to 10) containing relevant information, the LLM is trained to produce an accurate answer grounded in these passages.

4 Experiments

4.1 Experimental Setup

Datasets. We use 4 KGQA datasets to evaluate the multi-hop reasoning abilities of our approach: WebQSP (Yih et al., 2016), CWQ (Talmor and Berant, 2018), GrailQA (Gu et al., 2021), and SimpleQA (Bordes et al., 2015). Additionally, we evaluate on 3 ODQA datasets: HotpotQA (Yang et al., 2018a), WebQuestions (Berant et al., 2013), and Natural Questions (NQ) (Kwiatkowski et al., 2019a). We evaluate our models on a random sample of 1000 instances from each dataset to manage computational costs. However, for WebQSP and CWQ, we use the full test set, and for Natural Questions, we assess a random subset of 400 samples. Freebase KG (Bollacker et al., 2008) is utilized for all datasets. See §A.1 for details.

Evaluation Metrics. We compute Hits@1 to evaluate the models' answers following prior works (Baek et al., 2023; Jiang et al., 2023; Li et al., 2023). That is, each question receives a score of 1 if the target answer is present within the predicted LLM answer, and 0 otherwise. The metrics used in our human evaluation studies are discussed in §4.2.2 and §4.3.

Baseline Methods. We use standard prompting (IO Prompt) (Brown et al., 2020) and Chain of Thought (CoT) prompting (Wei et al., 2022) with 6 in-context examples as prompting baselines *with no external knowledge*. Additionally, we compare

Method	WebQSP	CWQ	WebQuestions	HotpotQA	GrailQA	SimpleQA	Natural Questions	Avg.
				w/o External Knowledge				
IO Prompt w/ChatGPT	59.8	39.4	53.7	31.2	27.4	18.5	51.1	40.1
CoT w/ChatGPT	61.0	37.8	54.1	33.1	29.6	18.8	52.8	41.0
				fine-tuned w/External Knowledge				
DeCAF (Yu et al., 2023)	76.6	56.6	-	-	-	-	-	-
KD-CoT (Wang et al., 2023)	73.7	50.5	-	-	-	-	-	-
				Prompting w/External Knowledge				
ToG (ChatGPT) (Sun et al., 2024)	74.1	46.7	59.3	28.3	70.8	56.7	44.8	54.3
WebGLM (Liu et al., 2023)	63.5	42.3	54.3	38.7	34.3	29.7	57.6	45.8
EWEK-QA (Ours) w/ KG	59.9	40.1	50.7	20.6	70.0	57.9	17.8	45.2
EWEK-QA (Ours) w/ Web	68.0	48.1	58.1	42.9	36.7	33.0	64.7	50.2
EWEK-QA (Ours) w/ KG + Web	71.3 (+**7.8**)	52.5 (+**10.2**)	61.2 (+**6.9**)	43.6 (+**4.9**)	60.4 (+**26.1**)	50.9 (+**21.2**)	62.5 (+**4.9**)	57.4

Table 1: Hits@1 accuracy ↑ for different datasets. **KGQA Datasets**: WebQSP, CWQ, GrailQA, and SimpleQA. **Open-domain Datasets**: WebQuestions, Hotpot, and Natural Questions. For EWEK-QA , we compare having access to only KG triples, only web quotes, or both. The parentheses represent improvement over WebGLM.

our method to 4 SoTA baselines that can access external knowledge: ToG (Sun et al., 2024), KG-CoT (Wang et al., 2023), DeCaF (Yu et al., 2023), and WebGLM (Liu et al., 2023).

ToG performs beam search on KGs using ChatGPT to keep track of the most relevant reasoning paths. KD-CoT generates faithful reasoning traces based on retrieved external knowledge to produce precise answers. DeCaF is a finetuning-based method that jointly generates search queries over KGs and predicts the final answer. Finally, WebGLM is a web-based QA system that composes an answer based on external knowledge retrieved through a web search.

We experiment with two backbone language models for ToG: ChatGPT (Table 1) and Llama-2-13B (Table 4) (Touvron et al., 2023). For EWEK-QA and WebGLM, we use the WebGLM-10B answer composer in all experiments unless specified otherwise. Refer to §A.1 for more details.

Human Evaluation Setup. We perform two human annotation experiments to evaluate the quality of: (i) final answer generated by the model (§4.2.2) and (ii) web quotes extracted by our adaptive retriever (§4.3). Four professional annotators are carefully selected from a pool of 20 candidates based on their demonstrable skills and expertise. They are trained and continuously monitored by our domain expert. We present them with detailed task-specific instructions and illustrative examples covering potential scenarios that they might encounter during the annotation process. They contribute 600 hours of total annotation time on both tasks and receive a compensation of $19 USD per hour.

4.2 Main Results

In this section, we present the end-to-end evaluation of our proposed system. We evaluate the quality of the final generated answer using Hits@1 accuracy (§4.2.1) and human annotators (§4.2.2).

4.2.1 Automatic Evaluation

The performance comparison of EWEK-QA with different prompting and finetuning baselines is presented in Table 1. We observe that standard IO prompting results in particularly low performance for multi-hop open-domain and KGQA datasets (e.g. HotpotQA and GrailQA). Interestingly, CoT delivers only a minor improvement over standard prompting which shows that such multi-hop domains require more than relying on the internal parametric knowledge of ChatGPT. Although DeCAF outperforms KG-CoT and exhibits top performance for WebQSP and CWQ, it requires extensive finetuning for each dataset making it an impractical choice for a generic QA system. ToG shows competitive performance on KGQA datasets but falls short on open-domain datasets such as HotpotQA. Note that, unlike the open-domain setting, KGQA methods assume that the answer is an entity that exists in the KG; making them unsuitable for generic questions (e.g. Yes/No questions).

We observe that EWEK-QA with only web quotes surprisingly performs well on some KGQA datasets (e.g. CWQ) and significantly outperforms all baselines on HotpotQA. This demonstrates the power of web-based QA systems for multi-hop open-domain questions which require reasoning over more than one supporting passage to answer. Having access to both external knowledge modalities inherits both the benefits of WebGLM and ToG, achieving competitive performance on 5/7 datasets while using a fraction of computational cost (efficiency details in Table 4 and §4.4.1).

4.2.2 Human Evaluation

Metrics and Data. We adopt a 3-level *Correctness* score inspired by Gao et al. (2023a). Each model output receives one of three labels: `IDK` (Does not know or is unable to answer), `Incorrect` (Is fully or partially incorrect), or `Correct` (Is fully correct with, possibly, extra information). We create a special test set of 92 challenging queries: 20 factual queries adapted from SimpleQA and CWQ, 17 verbose factual queries developed for this research, 15 recent factual queries answerable with $\sim$ one-year-old data, 20 'yes/no' reasoning queries, and 20 factual reasoning queries from CWQ.

The annotation results for all methods are presented in Table 2. EWEK-QA w/KG + Web reports a significant improvement of 21% over the baseline WebGLM. ToG has competitive performance as WebGLM but has a higher proportion of "IDK" answers. We believe this is due to the reliance on ChatGPT's parametric knowledge when beam search fails.

Model	IDK	Incorrect	Correct
`IO Prompt w/ChatGPT`	0.38	0.21	0.41
`CoT w/ChatGPT`	0.41	0.18	0.40
`ToG`	0.25	0.25	0.50
`WebGLM`	0.00	0.47	0.53
`EWEK-QA w/KG`	0.00	0.48	0.52
`EWEK-QA w/Web`	0.01	0.41	0.58
`EWEK-QA w/KG + Web`	0.00	0.26	**0.74**

Table 2: Human evaluation performance measured in Correctness (*is it able to correctly answer the question?*) on a small 92 hand-picked challenging queries dataset using 3 annotation labels: `IDK` (unable to answer), `Incorrect`, `Correct`.

4.3 Web Retriever Evaluation

To measure the quality of Adaptive retriever quotes, we obtain human annotations for the top quotes retrieved by Adaptive retriever and the existing WebGLM retriever on 100 random queries from two datasets: ELI5 (Fan et al., 2019) and Natural Questions (Kwiatkowski et al., 2019b). The results are presented in Table 3. The quotes are annotated along three dimensions: *Answer Span* (AS), *Self-Containment* (SC), and *Pertinence* (Per). AS measures what proportion of the quote is used verbatim by the LLM Answer Composer, SC measures whether the answer in the quote is complete or truncated, and Pertinence measures how relevant the quote is to the query and whether it is able to answer the query (annotated on a scale of $0-3$). Refer to §A.5 for more details.

As can be seen in Table 3, the quality of our quotes is significantly better than WebGLM quotes on both datasets across all three dimensions. This holds true whether we consider only the topmost quote or all top five quotes extracted by the retriever. This can be attributed to our improved heuristic parser and the additional evidence extractor component that is able to extract more relevant and complete quotes from webpages.

4.4 Ablation and Analysis

4.4.1 Efficiency Analysis.

In Table 4, we study the efficiency of EWEK-QA over ToG. In order to ensure a fair comparison, we stick to open-source LLMs for both methods. We observe that ToG's performance significantly degrades when using small open-source LLMs (i.e. LLaMA-2-13B). On the other hand, EWEK-QA uses even smaller open sourced LLMs and achieves SoTA performance. Moreover, EWEK-QA requires only one LLM call to compose an answer as opposed to ToG which can require up to $2ND + D + 1$, where N and D are the width and depth of the beam search respectively. Our method relies on fast retrievers and small embedding models (e.g. SentenceBert) allowing for $5\times$ decrease in LLM calls per question.

In Table 4, we also present an analysis of how variations in the answer composer model impact the performance of our system. This examination leverages two distinct datasets: WebQSP, a KGQA dataset, and WebQuestions, an ODQA dataset. Specifically, we conducted experiments employing three different models: WebGLM-10B, WebGLM-2B, and LLaMA-2-13B. The LLaMA-2-13B (chat version) model is fine-tuned on the WebGLM-QA dataset (Liu et al., 2023) [4]. This dataset, previously utilized by Liu et al. (2023) for fine-tuning WebGLM-2B/10B answer composer models, comprises of questions, reference passages, and corresponding answers grounded within these passages. Fine-tuning of the LLaMA-2-13B model was conducted across 8 V100 GPUs. More details can be found in §A.4.

Our findings indicate that the performance of our system remains robust across various off-the-shelf open-source LLMs serving as answer composers.

[4] https://huggingface.co/datasets/THUDM/webglm-qa

		ELI5			NQ		
Quotes	**Retriever**	**Per**	**AS**	**SC**	**Per**	**AS**	**SC**
Top-1	WebGLM	1.76	0.37	0.35	2.07	0.28	0.49
	EWEK-QA (EE)	2.07	0.48	0.49	1.76	0.37	0.35
	EWEK-QA (PS)	2.2	0.57	0.54	2.51	0.47	0.73
	EWEK-QA (EE + PS)	2.23	0.5	0.58	2.66	0.53	0.8
Top-5	WebGLM	1.71	0.35	0.34	1.86	0.3	0.38
	EWEK-QA (EE)	1.9	0.45	0.37	2.2	0.45	0.57
	EWEK-QA (PS)	1.99	0.49	0.41	2.17	0.39	0.55
	EWEK-QA (EE + PS)	2.02	0.5	0.43	2.36	0.47	0.64

Table 3: Human evaluation of the web quotes retrieved by our system and WebGLM measured using Pertinence (Per), Answer Span (AS), and Self-containment (SC). Evaluated by professional human annotators on 200 random queries ELI5 and Natural Questions datasets.

Dataset	**Method**	**LLM**	**Avg. Runtime (s)**	**Avg. # LLM calls**	**Hits@1**
WebQSP	ToG	LLaMA-2-13B	128.7	5.6	45.6
	WebGLM	WebGLM-10B	44	1	65
	EWEK-QA (Ours)	LLaMA-2-13B	29	1	73.2
		WebGLM-10B	40	1	72.9
		WebGLM-2B	21	1	68.9
WebQuestions	ToG	LLaMA-2-13B	124.4	5.7	37.8
	WebGLM	WebGLM-10B	45	1	54.3
	EWEK-QA (Ours)	LLaMA-2-13B	26	1	60.8
		WebGLM-10B	35	1	61.2
		WebGLM-2B	20	1	58.4

Table 4: Efficiency vs Performance Analysis: Comparing the baseline ToG with our method locally using the LLaMA-2 and WebGLM models respectively. *"Avg. # LLM calls"* represents the average number of LLM calls performed by ToG during search and reasoning stages per question. *"LLM"* represents the LLM used by the method to predict the answer. Here, EWEK-QA uses both KG and web external knowledge. *"Hits@1"* reports the performance on 1000 samples.

Notably, despite similarities in LLM utilization between our approach and the baselines, our method enhances the quality of the extracted knowledge and outperforms both in terms of effectiveness and runtime.

Table 5 provides a comprehensive runtime (sec/query) comparison between the EWEK-QA web retriever and WebGLM retriever, conducted on a subset of 230 questions from the Natural Question dataset. We have run WebGLM with the default hyperparameters provided by the authors. We ran both EWEK-QA and WebGLM retriever models on V100 Nvidia GPUs. The efficiency of our web retriever stems from our deliberate design choices. Our design facilitates parallelism during web page scraping and segmentation through the Evidence Extractor and Paragraph Splitter modules, as these modules operate independently. Also, each web page is parsed and segmented independently of others, optimizing efficiency. Unlike employing a single retriever or re-ranker, we adopt a double-module format for faster inference. In this format, a smaller, faster language model (LM) with 22 million parameters filters out noisy quotes, retaining only the most promising 70 quotes, which are then reranked by a larger LM with 900 million parameters to enhance final ranking. Notably, this reranking process with the larger model is swift due to the small number of quotes involved in a single forward pass.

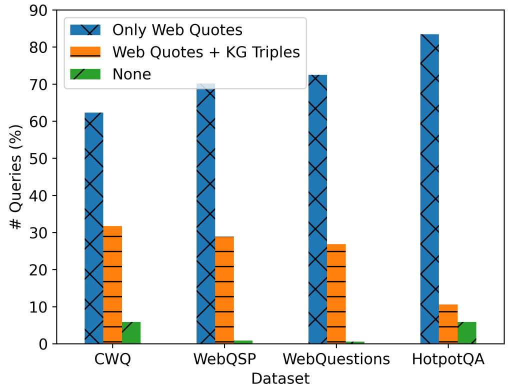


Figure 3: Sources of the quotes cited by the Answer Composer across queries from two KGQA and two ODQA datasets. "None" denotes that the answer contains no citations. "Web Quotes + KG Triples" indicate that both KG Triples and Web Quotes are cited.

	Total Time	Extract (Split)	Fetch (Crawling)	Filter (rank)	Search Engine
WebGLM Retriever	12.2093	0.785376	8.44552	1.8486	1.12285

	Total Time	Deduplicator	EE + PS	Retrieve & Re-rank	Search Engine
EWEK-QA Web Retriever	11.4965	0.12887	9.47622	1.0043	1.0028

Table 5: Average run time (in seconds) of each module within EWEK-QA web retriever and WebGLM retriever for a single query. EE and PS refer to Evidence Extractor and Paragraph Splitter, respectively. The experiment was performed on 230 questions randomly selected from the Natural Questions dataset.

4.4.2 Citation Analysis

In order to study the efficacy and usefulness of our knowledge extraction approach, we conduct an analysis on the quotes cited by the LLM Answer Composer for queries from two KGQA and two ODQA datasets. Figure 3 presents the distribution of citation sources in the EWEK-QA answers across the four datasets. Interestingly, the model relies only on the Web Quotes for most of the queries across all datasets (70% of WebQSP questions). We attribute this to the richness and diversity of information available on the Web as compared to knowledge graphs. Nevertheless, KG Triples are utilized by the model for a large number of queries (29% of WebQSP questions).

Citation Accuracy. As hallucination is a common problem with LLMs, we verify the accuracy of these citations. We extract cited sentences from the answer and use GPT-3.5 to assess whether it is supported by the corresponding citation. We achieve an average citation accuracy of 89.6% across the four datasets, thereby strengthening the claim that hallucination is not biasing the citation analysis. Refer to §A.3 for per-dataset accuracy, prompt details and an analysis on the number of cited quotes.

5 Conclusion

We introduce EWEK-QA, an efficient and generic QA system that is capable of answering both open domain and multi-hop reasoning questions. Contrary to prior works, our system relies on two external knowledge modalities: KGs and the web. We develop an adaptive web retriever to extract coherent and complete quotes from webpages. Furthermore, our ToG-E method eschews reliance on LLMs to extract the most relevant KG triples. Extensive experimental results on a variety of dataset types show significant efficiency gains over baselines. For future work, we aim to further improve the KG subgraph extraction module through more powerful embedding methods and experiment with using bigger backbone LLM models.

Limitations

The datasets used in this work successfully benchmark the multi-hop reasoning abilities of all methods. However, we have found the exact match (i.e. Hits@1) evaluation to be constraining for this setting. For example, there exists many cases where the answer composer will correctly answer a question but in a different wording than expected. Therefore, it is worthwhile to scale up the human evaluation to gain more reliable results. Moreover, we do not utilize the most up-to-date KGs such as WikiData[5] which can limit our performance on temporal questions.

References

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *ACL 2023 Workshop on Matching Entities*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

[5] https://www.wikidata.org/wiki/Wikidata:Introduction

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks.

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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, page 3477–3488, New York, NY, USA. Association for Computing Machinery.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Lei Huang and Weijiang Yu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason on structured data.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

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. 2019a. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980, Toronto, Canada. Association for Computational Linguistics.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4549–4560, New York, NY, USA. Association for Computing Machinery.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.

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. 2022. Webgpt: Browser-assisted question-answering with human feedback.

Aleksandr Perevalov, Xi Yan, Liubov Kovriguina, Longquan Jiang, Andreas Both, and Ricardo Usbeck. 2022. Knowledge graph question answering leaderboard: A community resource to prevent a replication crisis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2998–3007, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family. In *The Semantic Web – ISWC 2023*, pages 348–367, Cham. Springer Nature Switzerland.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018b. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems (NeurIPS)*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2023. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations*.

A Appendix

A.1 Implementation Details

Since initial topic entities for HotPotQA and NQ are not provided by default, we use the ReFinED model (Ayoola et al., 2022) to identify the WikiData (Vrandečić and Krötzsch, 2014) entities in each question and map them to the corresponding FreeBase entities via the “FreeBase ID" relation. We discard the questions where no Freebase entities are found. Moreover, we do not use the provided context for HotpotQA questions to test the retrieval performance of the baselines.

All method outputs were reproduced, except for DeCAF and KD-CoT. DeCAF outputs were obtained from their GitHub repository[6], evaluated using our script, while KD-CoT numbers were extracted from their paper. All results are on 1000 random samples except for WebQSP and CWQ which are on the full test set and for Natural Questions which is on random 400 samples.

We use LLaMA-2 with ToG. LLaMA-2 was run on 8 V100-32G GPUs without quantization, with temperature parameter 0.4 for pruning and 0 for the reasoning process. The maximum token length for the generation is set to 256. We use 5 shots in the ToG prompts for all the datasets. The maximum depth and width of the beam search is fixed to 3. ToG-E uses the same hyper-parameters except that there is no reasoning/stopping stage and the pruning is performed via SentenceBert (Reimers and Gurevych, 2019). The final answer is generated by the answer composer given the extracted KG triples and web quotes. For EWEK-QA and WebGLM, we retrieve the top 5 relevant quotes of max length 128 tokens. The web-pages are retrieved via the Bing search API.

A.2 Computing Infrastructure

All experiments were done on a Ubuntu 20.4 server with 72 Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz cores and a RAM of size 755 Gb. We use a NVIDIA Tesla V100-PCIE-32GB GPU.

A.3 Citation Accuracy

The citation accuracy on the individual datasets is presented in Table 6. We used the following prompt with GPT-3.5 to assess if the citation was accurate or not:

```
You are given an Answer and a Context. Your task is
to identify whether the information in the Answer
```

[6] https://github.com/awslabs/decode-answer-logical-form

Dataset	Accuracy
CWQ	86.6
WebQSP	92.7
WebQuestions	93.2
HotpotQA	86.1

Table 6: Citation Accuracy for the four datasets. GPT-3.5 is used for judgements.

```
is present in (or supported by) the information
in the Context. Output "Yes" if the Answer is
supported by the Context.
    Answer: {sub-answer}
    Context: {quote}
```

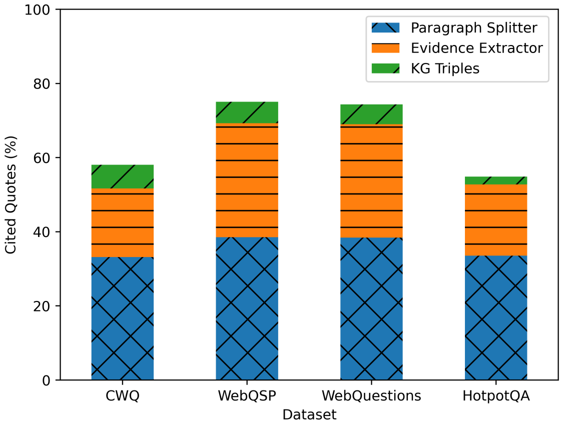


Figure 4: This figure depicts what percentage of the quotes provided by our knowledge extraction approach are cited in the final answer. Note that Paragraph Splitter and Evidence Extractor correspond to Web Quotes, and KG Triples come from the knowledge graph.

We also analyse *how many* of the quotes provided by our knowledge extraction approach are deemed useful by the answer composer. Figure 4 shows that, on average, 65% of the quotes provided are cited by the answer composer across the four datasets. More importantly, quotes originating from both streams of our adaptive web retrieval – Paragraph Splitter and Evidence Extractor – are used in the final answer.

A.4 Answer Composer

Given the quotes, we prompt the WebGLM 10B answer composer model with the following:

```
[CLS] Reference [1]: {Quote1} \Reference [2]:
{Quote2} \Reference [3]: {Quote3} \Reference
[4]: {Quote4} \Reference [5]: {Quote5} \Question:
{Question} \Answer: [gMASK] <|endoftext|>
<|startofpiece|>
```

When using the LLaMA-2-13B answer composer, we use the following prompt:

```
<s> [INST] <<SYS>> Given the following quotes
answer the question. You are given five quotes with
their numbers. Each quote used in the answer should
be cited with [ and ] symbols and the number of the
quote in between.  <</SYS>><s> [INST] <|QUESTION|>
{Question} <s> [INST] <|QUESTION|> <|QUOTES|>
1: {Quote}
2: {Quote2}
3: {Quote3}
4: {Quote4}
5: {Quote5}
<s> <|ANSWER|>
```

When KG triples are included, they are passed in as the first quote.

A.5 Web Retriever Quotes Annotation

For this task, the annotators were asked to evaluate the quality of the extracted quotes. They were given files containing queries and quotes. To evaluate how good is the Quote at answering the Query or at contributing in answer the query, the annotators were asked to work on three different metric for each quote:

- **Pertinence:** a score that measures how relevant the quote is to the query and whether the answer can be found in the quote. The given score must be $[0, 3] \uparrow$:
  - 0 means that the quote does not answer the query AND the query-quote pair is irrelevant (different subjects).
  - 1 means that the quote does not answer the query BUT the query-quote pair is relevant (have same subject).
  - 2 means that the quote partially answers the query AND the query-quote pair is relevant (have same subject).
  - 3 means the quote completely answers the query AND the query-quote pair is relevant (have same subject).
- **Answer-span:** this metric is used to know where the answer is. The annotators were asked to highlight the part of the quote which answer (even partly) the query. The highlighted text must be a single continuous string of text; if the answer to the query appears in multiple sections, separated by non-related data, all the sections must be highlighted. The score for this metric is computed by dividing the length (number of characters) of the highlighted text by the length (number of characters of the quote), so the score would be $[0, 1] \uparrow$.
- **Self-containment:** this metric is used to measure if the answer was cut-off or absent. A

binary score was assigned to each quote; 0 means that the quote does not contain the answer or only partially, 1 means that the quote correctly contains and mentions the answer.

See Tables 10 and 9 for examples of human quote evaluation on queries from the Eli5 and NQ datasets respectively.

A.6 Generated answer Annotation

For this task, the annotators were asked to evaluate the evaluate if and how correctly does the generated answer respond to the query. They were given files containing queries and generated answers. To do so, the annotators were asked to use a single metric:

- **Correctness:** this metric indicates how well and how completely did the generated answer correctly respond the query? The given score must be $[0, 2] \uparrow$:
  - 0 means the generated answer indicates it does not know or is unable to answer the query.
  - 1 means the generated answer responds fully or partially incorrectly. This includes the answers that are only partially correct.
  - 2 means the generated answer responds correctly and might even include additional information.

A.7 Adaptive Web Retrieval

In this section, we layout additional details about the two components that produce candidate quotes, namely Paragraph Splitter and Evidence Extractor.

A.7.1 Paragraph Splitter

For every webpage returned by the search engine, we scrape the contents of each page using BeautifulSoup[7]. Like in Liu et al. (2023), we divide the webpage contents into a list of candidate passages using line breaks. Since web-scraping is a time consuming task, we use multi-threading to scrape and parse each webpage in parallel. Additionally, for efficiency, we cache the search engine results and the scraped URL contents in a database. We apply additional constraints to improve the quality of the candidate passages produced by the heuristic parser. In addition to using the newline character, we also utilise the <p> tags in the webpage's HTML to produce candidate passages; any passage that has less than 10 tokens is discarded. If a passage contains more than 80 tokens, we further break it down into shorter passages no longer than 80 tokens such that the sentence boundaries are respected. This is slightly different from WebGLM's parser since they just relied on line breaks and used `html2text` to extract the text from HTML pages. Furthermore, in WebGLM's parser, lines shorter than 50 characters were dropped and longer lines are truncated with first 1200 characters followed by "...".

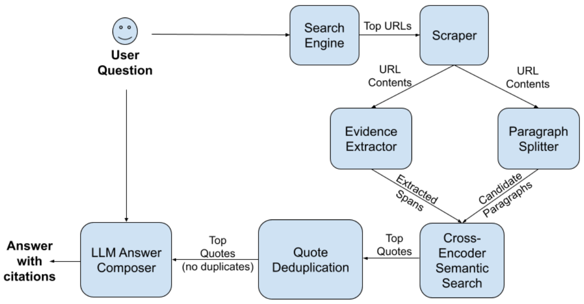
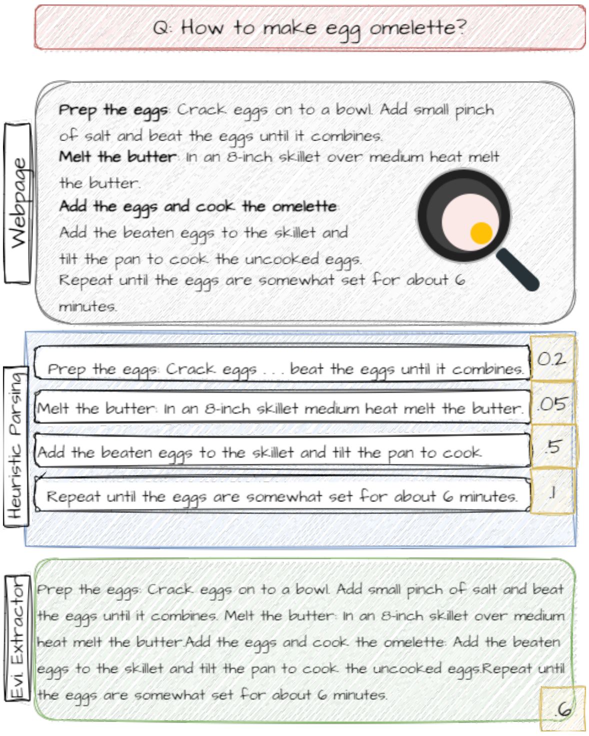


Figure 5: The pipeline for our adaptive web retrieval module.



Prep the eggs: Crack eggs on to a bowl. Add small pinch of salt and beat the eggs until it combines.
Melt the butter: In an 8-inch skillet over medium heat melt the butter.
Add the eggs and cook the omelette
Add the beaten eggs to the skillet and tilt the pan to cook the uncooked eggs.
Repeat until the eggs are somewhat set for about 6 minutes.

Prep the eggs: Crack eggs . . . beat the eggs until it combines.	0.2
Melt the butter: In an 8-inch skillet medium heat melt the butter.	.05
Add the beaten eggs to the skillet and tilt the pan to cook	.5
Repeat until the eggs are somewhat set for about 6 minutes.	.1

Prep the eggs: Crack eggs on to a bowl. Add small pinch of salt and beat the eggs until it combines. Melt the butter: In an 8-inch skillet over medium heat melt the butter.Add the eggs and cook the omelette: Add the beaten eggs to the skillet and tilt the pan to cook the uncooked eggs.Repeat until the eggs are somewhat set for about 6 minutes. .6

Figure 6: The candidate quotes produced by the Paragraph Splitter (Heuristic Parser) can be incomplete if they break on newlines or <p> tags. In comparison, a trained Evidence Extractor model can extract self-contained quotes.

[7] https://www.crummy.com/software/BeautifulSoup/

A.7.2 Evidence Extractor

By exploiting the similarity between evidence extraction and machine reading comprehension (MRC), we fine-tune the pre-trained MRC model—DeBERTa (He et al., 2021)— to extract a span of text from the documents. Instead of pursuing an answer span (like in MRC), the target here is to extract *evidence spans* that can provide support to answer the question. In comparison to the Paragraph Splitter, the candidate passages produced by Evidence Extractor are likely to be more complete and *self-contained* since it is not restricted to any predefined heuristic for chunking which can lead to incomplete chunks. See Figure 6 for an example.

Training Data. For this task, we use the MS Marco dataset which is a collection of about 1M queries sampled from Bing's search logs. The human editors are shown web passages relevant to the query and asked to compose a well-formed answer. Each query is accompanied with a set of 10 passages which may contain the answer to the question. The editors annotate the passage as *is_selected* if they use it to compose their final answer. Specifically, we create a training set of $110K$ instances from the *Train* split of MS Marco dataset using only the passages that the annotators tagged as useful while composing their answer (marked in the metadata with *is_selected* = 1). Similar to span-prediction based MRC, each train instance is a three-tuple of (q_i, s_i, c_i) where q_i refers to the query, s_i refers to the relevant passage span that the annotators used when composing their answer, and c_i is the full text of the webpage (u_i) from where s_i was derived. For every useful passage (*is_selected* = 1) per query, we use it and the corresponding webpage with the question to create a data-point in our dataset. The model trained is expected to predict the span, s_i, when jointly provided with query and context webpage text (q_i,c_i).

Training Details. We experimented with DeBERTa and Longformer models for this task. We train the model for one epoch on four V100 GPUs using a learning rate of 3×10^{-5} and training batch size of 32. The best model is selected using the $f1$ measure on a set-aside validation set of $10k$ instances.

A.8 Example Outputs

A sample answer for EWEK-QA and several baselines can be found in Table 7. Table 8 includes the KG triples and quotes extracted by EWEK-QA for a sample question in. Tables 10 and 9 include examples of human quote evaluation on queries from Eli5 and NQ datasets respectively.

Model	**Answer**
Which movie with a character called Ajila was directed by Angelina Jolie?	
WebGLM	The movie with a character called Ajila was directed by Angelina Jolie and is called Girl, Interrupted. Angelina Jolie stars as Lisa Rowe, a sociopath in the 1960s psychiatric hospital who becomes Susanna Kaysen's unlikely friend on her journey to self-discovery[4].
ToG	Based on the given knowledge triples, the movie directed by Angelina Jolie that features a character called Ajila is "In the Land of Blood and Honey". Therefore, the answer to the question is {In the Land of Blood and Honey}.
EWEK-QA	The movie directed by Angelina Jolie with a character called Ajila is In the Land of Blood and Honey[1][4]

Table 7: Output example for the 3 models. The question is picked from the HotpotQA dataset.

Quote	Content
Are the Laleli Mosque and Esma Sultan Mansion located in the same neighborhood?	
KG Triples	(’Esma Sultan Mansion’, ’architecture.architect.structures_designed’, ’Balyan family’), (’Esma Sultan Mansion’, ’architecture.structure.architect’, ’Balyan family’), (’Laleli Mosque’, ’religion.place_of_worship.religion’, ’Islam’),(’Balyan family’, ’architecture.structure.architect’, ’Esma Sultan Mansion’), (’Balyan family’,’architecture.structure.architect’, ’Beylerbeyi Palace’), (’Balyan family’, ’architecture.structure.architect’, ’Dolmabahçe Mosque’),(’Beylerbeyi Palace’, ’architecture.architectural_style.examples’, ’Ottoman architecture’), (’Dolmabahçe Clock Tower’, ’architecture.architectural_style.examples’, ’Ottoman architecture’), (’Dolmabahçe Clock Tower’,’architecture.structure.architectural_style’, ’Ottoman architecture’)",’The Esma Sultan Mansion (Turkish: Esma Sultan Yals)
Quote 1	The Esma Sultan Mansion (Turkish: Esma Sultan Yals), a historical yal located on the Bosphorus in the Ortakoy neighborhood of Istanbul, Turkey and named after its original owner Princess Esma Sultan, is used today as a cultural center after being redeveloped.
Quote 2	The Laleli Mosque (Laleli Camii) is the centerpiece of the Laleli neighborhood in Istanbul , Turkey . It sits along Ordu Street (Ordu Caddesi), which is part of the historic Divan Yolu
Quote 3	The Laleli Mosque (Turkish: Laleli Camii, lit. ’Tulip Mosque’) is an 18th-century Ottoman imperial mosque located in Laleli, Fatih, Istanbul, Turkey.
Quote 4	Laleli is a neighborhood in Istanbul, Turkey, with a few points of interest. It’s located in the Fatih district between Beyazt and Aksaray. Laleli Laleli runs along Ordu Street (Ordu Caddesi), which is part of the historic Divan Yolu.
Quote 5	Location At the intersection of Ordu and Fethi Bey Streets in Laleli Neighborhood, Eminonu District., Istanbul, Turkey Directions Associated Names Mustafa III, Ottoman Sultan Turkey patron Mehmet Tahir Aga Turkey architect/planner Istanbul Turkey place Events / AH damaged in earthquake in 1766/1179 AH Show all 3 Style Periods Ottoman Variant Names Laleli Kulliyesi Alternate transliteration Laleli Kulliye Variant. About Home Sites Authorities Collections Search Laleli Kulliyesi Istanbul, Turkey The Laleli Complex is a mosque complex commissioned and partly designed by Ottoman sultan Mustafa III. The word "Lale", in Turkish, is a tulip.

Table 8: KG triples and quotes extracted by EWEK-QA .

Retriever	Quote	Per	AS	SC
	who was the king of england in 1756			
WebGLM	King of Great Britain and Ireland from 1727 to 1760	1	0.0	0
Ours	George II (George Augustus; German: Georg August; 30 October / 9 November 1683[a] - 25 October 1760) was King of Great Britain and Ireland, Duke of Brunswick-Luneburg (Hanover) and a prince-elector of the Holy Roman Empire from 11 June 1727 (O.S.) until his death in 1760.	3	1.0	1
	what is the name of the hyena in lion king			
WebGLM	In the first book of the novelization series, The Lion King: Six New Adventures, A Tale of Two Brothers, the young hyenas appear at the beginning where they surround a helpless Rafiki who had just arrived in the Pride Lands. King Ahadi comes to his rescue just in time, with his sons Mufasa and Taka (Scar) in tow, and demands that the hyenas leave. Later, Scar meets with the hyenas, the story revealing that he asked for their help. Shenzi advises Scar to make himself look good by making Mufasa look bad. Taka is intrigued by the suggestion and plans to incite a cape buffalo called Boma to fight Mufasa. His plan goes badly and his eye is wounded by the buffalo herd in the process, hence the nickname "Scar".	1	0.0	0
Ours	According to Timon in The Lion King 1, her full name is Shenzi Marie Predatora Veldetta Jacquelina Hyena. Her first name means s̈avage,̇ p̈agan,̇ üncouth,̇ or b̈arbarousïn Swahili. Her distinguishing features are five prominent bangs hanging over her face and a mane that reaches all the way to her bangs.	3	0.34	1
	who is the biggest selling female group of all time			
WebGLM	Opening up the top three best-selling girl groups of all time is the American R&B trio TLC. The girl group was formed in 1991 and it consisted of T-Bozm Chilli, Left Eye, and Crystal Jones, however, Crystal left the group early on.	1	0.0	0
Ours	The Spice Girls (here in 2008) are the best-selling girl group in history.	3	1.0	1

Table 9: Examples of human quote evaluation on queries from NQ.

Retriever	Quote	Per	AS	SC
	Why do we prefer cold pillows to warm pillows?			
WebGLM	- Firmness: Preferred firmness will differ from person to person. Firm pillows can provide greater support for the head and neck, while softer pillows can reduce pressure. A person can consider choosing pillows from brands that offer sleep trials or allow returns so they can replace or exchange their pillow if it is not comfortable.	1	0.0	0
Ours	Is it better to sleep on a cold pillow? Research shows that a warm environment may keep people awake and disrupt the body's regular sleep-wake cycle, resulting in poor quality sleep. A cold pillow may help a person feel cool enough to fall asleep.	3	0.84	1
	PayPal, how it works, if it's trustworthy/worthwhile and should I get it?			
WebGLM	PayPal has various measures in place to make sure PayPal is safe for both buyers and sellers, whenever they make or receive an online payment. It's one of the reasons PayPal grew to be such a popular payment platform. For buyers, PayPal offers protection on eligible PayPal purchases, as well as 24/7 monitoring of accounts to watch out for suspicious transactions, dispute resolution services and more.	2	1.0	0
Ours	PayPal is a service that allows you to use your credit card to pay for things on the internet. It's not trustworthy or worthwhile, but if you're willing to put in the time and effort to learn how to use it, it's worth it. If you don't trust PayPal, don't use it.	3	1.0	1

Table 10: Examples of human quote evaluation on queries from Eli5.