Mitigating Large Language Model Hallucinations via Autonomous Knowledge Graph-Based Retrofitting

Xinyan Guan[1,2*], Yanjiang Liu[1,2*], Hongyu Lin[1], Yaojie Lu[1†], Ben He[1,2], Xianpei Han[1,3], Le Sun[1,2†]

[1]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]State Key Laboratory of Computer Science Institute of Software, Chinese Academy of Sciences, Beijing, China
{guanxinyan2022,hongyu,luyaojie,xianpei,sunle}@iscas.ac.cn,
liuyanjiang22@mails.ucas.ac.cn,benhe@ucas.edu.cn

Abstract

Incorporating factual knowledge in knowledge graph is regarded as a promising approach for mitigating the hallucination of large language models (LLMs). Existing methods usually only use the user's input to query the knowledge graph, thus failing to address the factual hallucination generated by LLMs during its reasoning process. To address this problem, this paper proposes Knowledge Graph-based Retrofitting (KGR), a new framework that incorporates LLMs with KGs to mitigate factual hallucination during the reasoning process by retrofitting the initial draft responses of LLMs based on the factual knowledge stored in KGs. Specifically, KGR leverages LLMs to extract, select, validate, and retrofit factual statements within the model-generated responses, which enables an autonomous knowledge verifying and refining procedure without any additional manual efforts. Experiments show that KGR can significantly improve the performance of LLMs on factual QA benchmarks especially when involving complex reasoning processes, which demonstrates the necessity and effectiveness of KGR in mitigating hallucination and enhancing the reliability of LLMs.

Introduction

Large Language Models (LLMs) have gained increasing prominence in artificial intelligence. The emergence of potent models such as ChatGPT (OpenAI 2022) and LLaMA (Touvron et al. 2023) has led to substantial influences on many areas like society, commerce, and research. However, LLMs still suffer from severe *factual hallucination* problems, i.e., LLMs can frequently generate unsupported false statements regarding factual information due to their lack of intrinsic knowledge (Ji et al. 2023). For example, in Figure 1, ChatGPT fails to provide an accurate response to the query "*When is Frédéric Chopin's father's birthday?*" due to a wrong belief that Nicolas Chopin's birthday is on June 17, 1771. Factual hallucination poses a severe challenge for LLM applications, particularly in real-world situations where factual accuracy holds significance. Consequently, the endeavor to alleviate factual hallucinations in LLMs has become a research hotspot in NLP field (Liu et al. 2021; Kang and Hashimoto 2020).

On the other hand, Knowledge Graphs (KGs) store a substantial amount of high-quality factual information, which can significantly alleviate factual hallucination if incorporated with LLMs. For example, in Figure 1, we can retrofit the erroneous statement "*Nicolas Chopin was born on June 17, 1771*" by referring to the provided factual knowledge "*(Nicolas Chopin, date of birth, 1771-04-15T00:00:0)*" in Wikidata. Recent work has focused on integrating LLMs with KGs by retrieving the entities in the query within knowledge graphs. Then the obtained factual triples are utilized as an additional context for LLMs to enhance their factual knowledge (Baek, Aji, and Saffari 2023; Chase 2022). Unfortunately, these approaches are limited to retrieving factual knowledge relevant to entities explicitly mentioned within the given query. However, the fundamental capability of large language models involves intricate and multi-step reasoning. Such reasoning processes often necessitate the validation and augmentation of factual knowledge that may be employed during the reasoning process. For example, in the case shown in Figure 1, LLM fails to answer the question because it requires an intermediate knowledge about "*Nicolas Chopin was born on April 15, 1771*". However, such information does not refer to entities appearing in the query. As a result, previous approaches are inadequate in addressing the factual hallucination appearing in the reasoning processes of LLMs.

In this paper, we propose Knowledge Graph-based Retrofitting (KGR), a new framework that incorporates LLMs with KGs to mitigate factual hallucination during the entire reasoning process of LLMs. Instead of retrieving factual information from KGs using original queries, the main idea behind KGR is to autonomously retrofit the initial draft responses of LLMs based on the factual knowledge stored in KGs. However, achieving the above process is challenging because draft responses generated by large language models typically contain a mixture of various information about the reasoning process, making the extraction, verification, and revision of relevant knowledge in it very challenging. Therefore, the key to integrating Knowledge Graphs into the reasoning process of large models to mitigate factual hallucinations lies in efficiently extracting the information requiring

*These authors contributed equally.
† Corresponding authors.

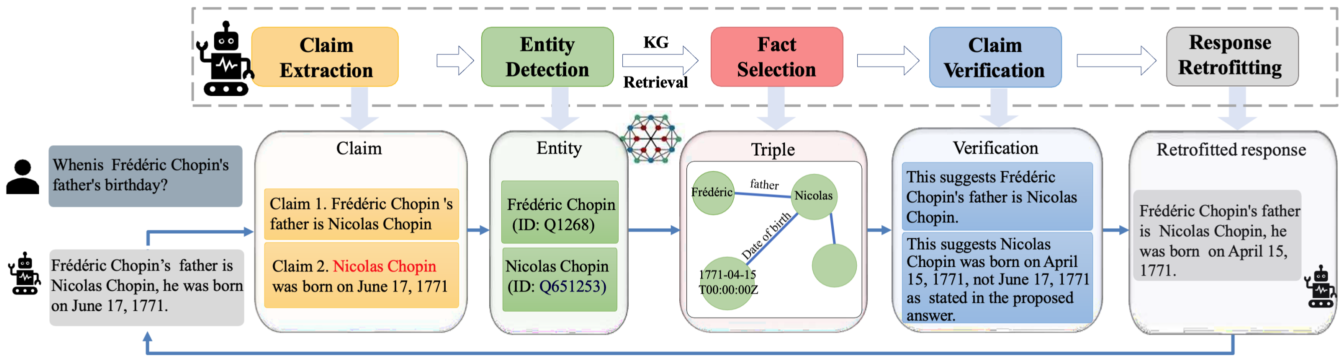

Figure 1: An overview of KGR, our framework consists of five components: (1) claim extraction, (2) entity detection and KG retrieval, (3) fact selection, (4) claim verification, (5) response retrofitting. The core component of these five steps remains the LLM. Given a question and draft response as input, our framework can iteratively mitigate factual errors in LLM's response.

validation from draft responses, querying and selecting relevant knowledge from the knowledge graphs, and using this knowledge to verify and refine draft responses.

To this end, KGR presents a LLMs-based framework to autonomously extract, validate and refine factual statements within the initial draft responses without any manual efforts. Specifically, given an input query and a draft response generated by the LLM that entails the reasoning process of how LLM resolves this problem, KGR will request a LLM to extract factual claims in the reasoning process that require verifying by KGs. As shown in Figure 1, given the draft response "*Frédéric Chopin's father is Nicolas Chopin, he was born on June 17, 1771.*", the claim extraction step will generate factual claims in it like "*Frédéric Chopin's father is Nicolas Chopin*" and "*Nicolas Chopin was born on June 17, 1771*". Then, KGR will identify critical entities in the extracted claims, retrieve relevant factual triples from knowledge graph about the entities, and use a LLM-based fact selector to identify fact triples relevant to the draft response. Subsequently, the retrieved factual knowledge is utilized to compare with the previously extracted factual claims from the draft to verify their correctness. Finally, LLMs are asked to retrofit the draft in accordance with the outcomes of factual verification. This process can be repeated multiple times to ensure that all facts in the generated answers align with the knowledge stored within the knowledge graph. In this way, our method can not only verify the facts in query and response but also the facts used during reasoning. Furthermore, because all phases in the procedure can be automatically executed using a large language model, our method doesn't need any external components and therefore is easy to implement.

We conduct experiments with three representative LLMs on three standard factual QA benchmarks with different levels of reasoning difficulty, including Simple Question (Bordes et al. 2015), Mintaka (Sen, Aji, and Saffari 2022) for complex reasoning, and HotpotQA(Yang et al. 2018) for open domain, multi-hop reasoning. Experiments show that KGR can significantly improve the performance of LLMs on factual QA benchmarks especially when involving complex reasoning processes, which demonstrates the necessity and effectiveness of KGR in mitigating hallucination.

In summary, the contributions are as follows:

- We propose a new framework that incorporates LLMs with KGs to mitigate factual hallucination by effectively extracting, verifying, and refining factual knowledge in the entire reasoning process of LLMs.
- We present an implementation of the above-mentioned procedure by executing all the above-mentioned steps using LLMs without introducing any additional efforts.
- Experiments on 3 datasets and 3 different LLMs confirm that KGR can significantly mitigate the hallucination and enhance the reliability of LLMs.

Related Work

Hallucination Hallucination in Large Language Models has been a prominent research focus within the NLP community (Ji et al. 2023). Automated large-scale data collection processes are prone to collecting erroneous information, which can significantly impact the quality of the generated outputs (Gunasekar et al. 2023). Additionally, excessive repetition of certain data during training can introduce memory biases, further exacerbating the hallucination issue (Lee et al. 2022; Biderman et al. 2023). Imperfections in the encoder backbone and variations in decoding strategies also play a role in determining the extent of hallucination in LLMs outputs (Tian et al. 2019). Recent studies have emphasized the importance of model output confidence as an indicator of potential hallucination occurrences (Manakul, Liusie, and Gales 2023).

Retrieval Augmentation To address hallucination issues in LLMs, two main categories of retrieval augmentation methods have been proposed, which can be concluded as "retrieve before generation" and "retrieve after generation". The retrieve before generation mainly focuses on leveraging information retrieval (IR) to provide additional information to LLMs about the query. Along this line, UniWeb (Li et al. 2023b) introduces an adaptive method for determining the optimal quantity of referenced web text, Chameleon(Lu et al. 2023) leverages an assortment of tools including search engines, to bolster the reasoning capabilities of LLMs, WebGLM (Liu et al. 2023b) augments LLMs with web search and retrieval capabilities. One major limitation of these approaches is the retrieved text is question-related, thus cannot guarantee the correctness of the question-unrelated portions in the generations. The retrieve after generation like RARR (Gao et al. 2023), PURR (Chen et al. 2023), and CRITIC (Gou et al. 2023) automatically edit model generations using evidence from the web. Our method leverages KGs as knowledge base to retrofit the model-generated response while reducing hallucination risk.

KG-Enhanced LLM The Knowledge Graph is regarded as a dependable source of information and is consequently frequently employed to enhance model generations. Traditional approaches involve knowledge representations during the training phase, which often necessitates dedicated model architecture and model-specific training (Zhang et al. 2019, 2022). However, this incurs a substantial cost for contemporary LLMs. In recent years, many researchers propose to inject knowledge during the inference stage. For example, KAPING (Baek, Aji, and Saffari 2023), RHO (Ji et al. 2022), KITLM (Agarwal et al. 2023), and StructGPT (Jiang et al. 2023) try to retrieve knowledge in KG and utilize them as an additional input context for LLMs to enhance their generations. However, these methods only search for question-relevant information, which limits the overall performance. To the best of our knowledge, we're the first to involve knowledge graphs in model response retrofitting.

KGR: Autonomous Knowledge Graph-Based Retrofitting

In this section, we introduce our proposed method KGR, which automatically mitigates factual hallucinations via a chain-of-verification process. As shown in Figure 1, given a query and its draft response, KGR retrofits the response by 1) extracting claims from the draft answer that requires verification; 2) detecting entities in the claims that are critical for retrieving facts from knowledge graph; 3) retrieving relevant fact statements from the knowledge graph; 4) verifying the factual correctness of each extracted claim using the returned factual statements from the knowledge graph; 5) retrofitting the previous draft response based on the verification results. All these steps are autonomously executed using the large language model itself without additional manual efforts. This process can be iterative and repeated multiple times to ensure that all facts in the generated answers align with the factual knowledge stored within the knowledge graph. In the following, we will describe each component in KGR respectively in detail.

Claim Extraction

Given a generated draft response as input, claim extraction will extract all factual claims from previously generated drafts that require validation. The main idea behind claim extraction is that a draft response can frequently contain various factual statements that need to be verified. For the example in Figure 1, the draft response contains at least two factual statements, i.e., "*Frédéric Chopin's father is Nicolas Chopin*" and "*Nicolas Chopin was born on June 17, 1771*". Therefore,

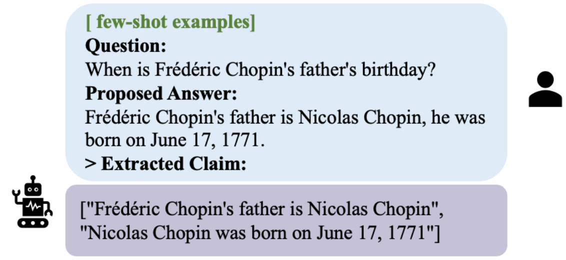

Figure 2: Example for the claim extraction in KGR, which decomposes the proposed answer into two atomic claims.

to make it possible for KG to verify these statements respectively, claim extraction decomposes the draft response to be atomic factual claims.

In this paper, we leverage LLM itself to autonomously extract the claims in the generated draft response. As shown in Figure 2, we prompt LLM with a query and response pair, with the anticipation of receiving a list of decomposed factual claims.

After extracting claims, entity detection identifies the mentioned critical entities for knowledge graph retrieval.

Entity Detection and Knowledge Graph Retrieval

Given a list of claims extracted from the draft response, entity detection will detect the critical entities mentioned in the claims. Then, we retrieve the detected entities' local subgraph from the KG and expressed it in the form of triples. The main idea behind entity detection and knowledge graph retrieval is that we need to identify entities in claims so as to retrieve the relevant knowledge in the KG. Meanwhile, we can ensure recalling relevant triples as much as possible by retrieving the local subgraph in the knowledge graph. For the example in Figure 1, we identify the entity *Frédéric Chopin* and its entity id *Q1268*, so we can search the identified entity to acquire knowledge relevant to *Claim1* in the KG.

In this paper, we prompt LLMs to detect entities. As illustrated in Figure 3, our approach shows powerful generalization ability by capitalizing on the information extraction capabilities of LLMs(Li et al. 2023a) through the utilization of few-shot prompt. Based on the few-shot prompts, we can make LLMs understand which entities merit fact selection. After detecting the entities, we retrieve the knowledge graph for the local subgraph and send it to fact selection in the form of triples. Concretely, we retrieve the node identifiers for these entities in Wikidata using heuristic rules and then query the knowledge graph for the neighboring nodes of the identified entity and obtain its local subgraph with SPARQL.

Fact Selection

Given the retrieved triples based on the detected entities, fact selection will select relevant fact statements among them. The main idea behind fact selection is the limited ability of LLMs in long-context modeling (Liu et al. 2023a) and the constraint on the context window size of LLMs, which make it impractical to select critical triples at one time.

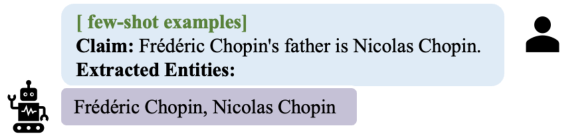

Figure 3: Example for the entity detection in KGR, which only extracts the essential entities from the claim.

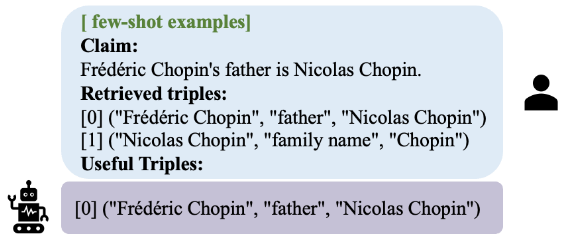

Figure 4: Example for the fact selection in KGR, in which LLMs are prompted to select critical items among all retrieved triples.

In this paper, we partition retrieved triples into several chunks and leverage LLM itself to extract the critical triples in the retrieved triples respectively, illustrated in Figure 4. In this way, we can avoid introducing excessive irrelevant knowledge into claim verification. Once we have selected the critical triples, the claim verification will verify the factual correctness of claims and subsequently offer suggestions.

Claim Verification

Given the critical triples selected by the fact selection, we utilize LLM to compare the model-generated claims with the factual information present in the KGs. The main idea behind claim verification is to propose a detailed revision suggestion for each claim, as retrofitting solely based on the selected knowledge may not convince LLMs. As illustrated in Figure 5, we employ LLMs to verify each claim and propose revision suggestions respectively based on the retrieved fact knowledge, so as to boost the execution of the following retrofitting step. Then, we send the claim verification result to LLM to ask it to retrofit the draft response accordingly.

Response Retrofitting

Given the verification of all claims, the response retrofitting step retrofits the generated draft response in accordance with the verification suggestions.

In this paper, we capitalize on the capabilities of LLMs for the purpose of retrofitting. This approach involves employing LLMs with a few-shot prompt, a strategy that has exhibited efficacy in prior researches (Gou et al. 2023; Zheng et al. 2023). As illustrated in Figure 5, we merge the entire KGR process into a singular prompt. This allows LLMs to leverage their in-context learning ability, comprehending the

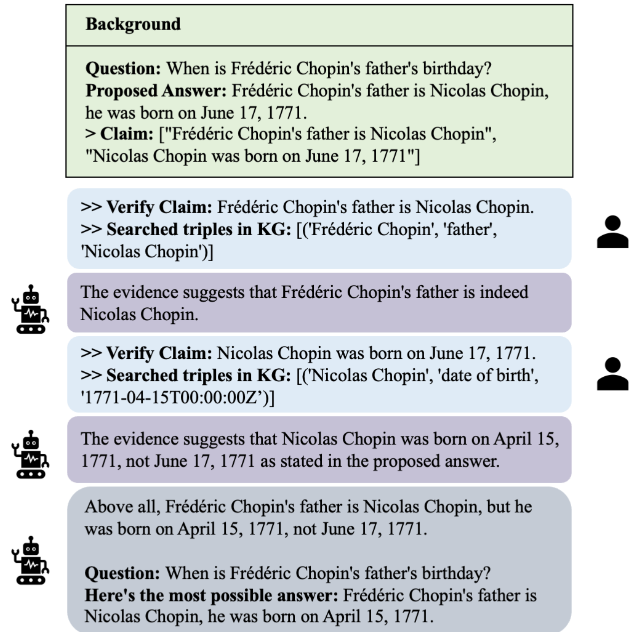

Figure 5: Example for the claim verification and response retrofitting in KGR. The claim verification judges whether the claim aligns with searched triples and gives revision suggestions respectively. The response retrofitting incorporates the revision suggestions from all claims and gives a refined response.

KGR process and enhancing their comprehension of factual retrofitting based on verification suggestions

By following the cycle of "Extraction - Detection - Selection - Verification - Retrofitting ", our KGR framework can be iterated multiple times to ensure all facts in the generated answers align with the factual knowledge stored within the knowledge graph.

Experiments

We evaluate our KGR framework on three datasets with different levels of reasoning difficulty, including Simple Question (Bordes et al. 2015), Mintaka (Sen, Aji, and Saffari 2022), and HotpotQA (Yang et al. 2018). We also compare KGR with information retrieval-based approaches and previous question-relevant knowledge graph retrieval approaches.

Experiment Settings

Dataset and Evaluation We conduct experiments on three representative factual QA benchmarks, including:

- **Simple Question (Bordes et al. 2015)** is a simple QA dataset that contains 100k questions constructed from Freebase knowledge graph, requiring no deep reasoning procedure. Therefore, we can evaluate the ability to retrieve relevant evidence in KG based on Simple Question.
- **Mintaka (Sen, Aji, and Saffari 2022)** is a complex, natural and multilingual dataset, composing 20k questions collected in 8 different languages. We only use English

	Simple Question		Mintaka		HotpotQA	
	ChatGPT	*text-davinci-003*	*ChatGPT*	*text-davinci-003*	*ChatGPT*	*text-davinci-003*
Vanilla	22.0/28.9	34.7/45.1	42.9/56.1	36.7/44.8	18.4/31.6	22.4/34.6
CoT	10.0/11.8	38.0/46.7	53.1/59.3	46.3/57.9	24.5/34.3	29.2/40.5
CRITIC	12.0/14.3	38.0/46.7	51.0/58.6	44.4/54.0	30.6/**41.7**	27.1/38.9
QKR	54.0/60.2	**56.0/62.0**	48.0/54.6	44.0/51.7	28.0/38.1	22.0/31.9
KGR (ours)	**58.0/60.7**	54.0/57.6	**53.1/60.8**	**52.0/60.2**	**32.7**/39.2	**34.0/47.2**

Table 1: Results on three datasets using ChatGPT and text-davinci-003. We implement CoT using the prompt provided by CRITIC. QKR uses the same entity detection and fact selection method as KGR. We report both EM and F1 scores in the table.

	Simple Question	Mintaka	HotpotQA
CoT	14.0/21.9	26.0/28.3	12.0/19.6
QKR	40.0/44.0	**26.5**/32.4	**12.2**/17.6
KGR(ours)	**46.0/46.9**	**26.5/34.0**	10.2/**20.6**

Table 2: Results on three datasets using Vicuna 13B. We report both EM and F1 scores in the table.

test sets. In this setting, we focus on the ability to logically refine and revise answers based on the evidence gathered.

- **HotpotQA (Yang et al. 2018)** is a Wikipedia-based [1] dataset with 113k questions that requires finding and reasoning over multiple supporting documents to answer, which are diverse and not constrained to any pre-existing knowledge bases or knowledge schemas. Therefore, we evaluate the KGR framework's robustness in handling generalized scenarios, requiring LLMs to answer involving the incorporation of both parametric knowledge and Knowledge Graph-based information.

We reported the results in terms of EM and F1 scores respectively on 50 samples from the validation set of each dataset. By comparing performance across these three datasets, we can evaluate how well different methods mitigate factual hallucinations and handle complex tasks.

LLMs and KG Implementation We evaluate the effectiveness of KGR on both close-source and open-source large language models. For close-source models, we evaluate on text-davinci-003 and ChatGPT (gpt-3.5-turbo-0301) to see whether alignment will have an impact on KGR. For the open-source model, we evaluate KGR on Vicuna 13B, a representative aligned open-source model, to see whether KGR can work well on compact size LMs. We choose Wikidata[2](Vrandečić and Krötzsch 2014) as our knowledge base, which encompasses structured data from various sources such as Wikipedia, Wikimedia Commons(Commons 2023), and other wikis associated with the Wikimedia movement(Meta 2023).

[1]https://wikipedia.org

[2]https://www.wikidata.org/

Baseline

We compared our KGR framework with the following methods, including:

- Vanilla, which adopts a straightforward approach to prompt the model to generate answers for the given question.
- Chain of Thought (CoT) (Wei et al. 2022), which aims to generate more reliable answers by prompting LLMs to generate more comprehensive and detailed explanations for the generated answers.
- Self-Correcting with Tool-Interactive Critiquing (CRITIC) (Gou et al. 2023), which revises the answer based on text from the web. Since CRITIC did not release their web crawling method, in this experiment we adopt the crawling pipeline provided in RARR (Gao et al. 2023) via Bing Search[3]
- Question-relevant Knowledge Retrieval method (QKR), which prompts LLMs with the question-relevant retrieved facts in the knowledge graph to generate answers. We aim to demonstrate the superior effectiveness of our response-relevant retrofitting method over the question-relevant knowledge graph augmentation approach. Inspired by KAPING (Baek, Aji, and Saffari 2023), QKR leverages extracted facts from KGs as prompts to enhance response correctness. In our implementation, we replace the fact extracts process with our entity detection and fact selection to strictly compare the difference between the response-relevant and query-relevant methods.

Overall Results

As shown in Table 1, our method demonstrates significant superiority over other methods across various conditions.

1) **Our framework can mitigate large language model hallucination via Knowledge Graph-based Retrofitting and achieve significant improvements on 3 datasets.** Compared with the CoT and CRITIC, our KGR framework gains improvements on all three datasets. This indicates that our KG-based approach is more effective due to its reliance on a reliable knowledge base, whereas IR-based methods like CRITIC might introduce noise from external. Additionally, we observed that the CoT

[3]https://www.bing.com/.

method performed worse than the vanilla approach in ChatGPT. This could be attributed to the CoT method's tendency to ask for more information, which is amplified in ChatGPT due to Reinforcement Learning from Human Feedback (Ouyang et al. 2022).

2) **By verifying the facts used during reasoning via chain-of-verification, our method can achieve significant performance improvement in complex reasoning tasks in Mintaka and HotpotQA datasets.** As shown in Table 1, compared to the QKR method, our KGR framework achieves F1 improvement for at least 6.2 and 1.1 on Mintaka and HotpotQA. Both of them pose complex reasoning question-answering challenges, and the success of our method with chain-of-verification on these datasets demonstrates its capability to handle complex questions effectively. It is worth noting that the text-davinci-003 outperformed QKR in Simple Question. We attribute this to the fact that Simple Question consists of straightforward, one-hop questions, which makes the question-relevant method more effective.

3) **By automatically generating and executing chain-of-verification via LLMs, our KGR approach exhibits remarkable generalization capabilities in different datasets and is robust on open-domain settings.** In HotpotQA, KGR performs well compared to the CoT and CRITIC methods. The HotpotQA presents an open-domain QA scenario where finding related triples in the KG can be challenging. Despite this difficulty, our method displayed the ability to effectively utilize the searched triples and effectively leverage parametric knowledge even when no evidence was returned.

4) **Our framework can work well on compact size LMs, aligned LLMs, and misaligned LLMs, showing the generalizability of KGR.** We compare KGR with the strong baselines CoT and QKR on Simple Question, Mintaka, and HotpotQA using Vicuna 13B. The result is shown in Table 2. We can find that the KGR framework outperforms both CoT and QKR, demonstrating the generalizability of our framework even leveraging a compact size LM. Moreover, the significant improvement with ChatGPT and text-davinci-003 shows the generalizability of both aligned LLMs and misaligned LLMs.

In summary, our method consistently outperforms other methods across various conditions and exhibits strong generalization ability. The results suggest that our KGR framework is more reliable and effective, especially in handling complex factual reasoning tasks. Furthermore, it showcases the robustness of our method in open-domain QA settings, where knowledge retrieval may be more challenging.

Case Study

We present a multi-round retrofitting process of a multi-hop case which needs to be retrofitted iteratively in Figure 3. In this case, the model-generated response shows a factual error in the initial reasoning step. It erroneously states that *Alex Shevelev died in Moscow, Russia*, whereas he actually passed away in Rome, Italy. After retrofitting this mistake, we encounter another factual error, which asserts that *Rome is the capital of the Central Federal District*. So, we need to retrofit it again based on the retrofitted response in the first iteration.

From this case, we show KGR's intermediate results, including atomic claim, critical triples, detailed verification, and iterative retrofitting. All these show the effectiveness of KGR, especially on reasoning with multi-hop complex tasks, verifying the feasibility of multi-turn retrofit to ensure that all facts in the generated answers align with the factual knowledge stored in the knowledge graph.

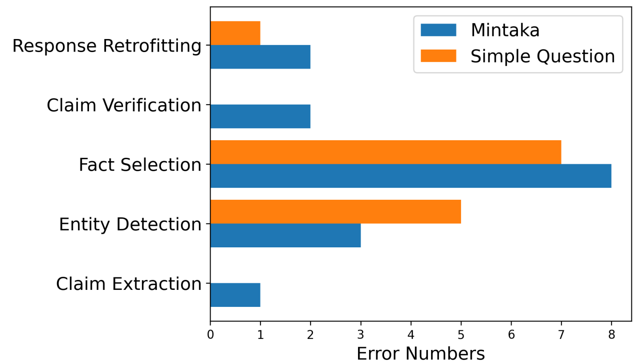

Figure 6: Distribution of error case numbers across KGR stages: Analysis conducted on a sample of 50 instances from the Mintaka dataset and 50 instances from the Simple Question dataset reveals the occurrence of error numbers across various stages of the KGR process.

Error Analysis

In order to gain a comprehensive understanding of the KGR approach, we conducted an exhaustive analysis of incorrect cases based on the Mintaka and Simple Question datasets. After carefully examining the errors, we identified which component causes revision failures. The outcomes of this analysis are visualized in Figure 6. On closer inspection, the main issues are inaccuracies in entity detection and fact selection, while claim extraction, verification, and response retrofitting are more reliable, underscoring the need to improve entity detection and fact selection.

On the other hand, our analysis explored the error reason in different stages. The claim extraction often fails to express the central claim adequately, sometimes due to excessive use of pronouns that confuse the model's comprehension. Entity detection has issues with entity extraction granularity. It captures too many common entities like "films" or "apple", leading to excessive and useless triples for the claim verification. The fact selection has difficulty extracting the critical triples between multiple triples that contain noise. For the claim verification and response retrofitting, the focus shifts to the model's ability to adhere to the cues provided by the few-shot prompts. Effectively discerning and subsequently rectifying answers within this framework presents a central challenge. The process of fact selection encounters challenges in extracting essential triples from a collection of triples that include irrelevant information or noise.

Question: The city where Alex Shevelev died is the capital of what region?
Answer: the Lazio region
Initial prediction: Let's think step by step. *Alex Shevelev*† died in *Moscow*‡, Russia. And it is the capital of the Central Federal District. So the answer is: Central Federal District.

	Claims	Fact Knowledge	Verification	Retrofitted Response
KGR round1	*Alex Shevelev*† died in Moscow, Russia.	(*Alex Shevelev*†, place of death, Rome)	The evidence suggests Alex Shevelev died in *Rome**, not Moscow, Russia.	Let's think step by step. Alex Shevelev died in *Rome, Italy**. And it is the capital of the Central Federal District. So the answer is: Central Federal District.
	Moscow‡ is the capital of the Central Federal District.	(Central Federal District, capital, *Moscow*‡)	The evidence suggests Moscow is the capital of the Central Federal District.	
KGR round2	*Alex Shevelev*† died in *Rome*$^\diamond$, Italy.	(*Alex Shevelev*†, place of death, *Rome*$^\diamond$)	The evidence suggests Alex Shevelev died in Rome.	Let's think step by step. Alex Shevelev died in Rome, Italy. And it is the capital of *Lazio**. So the answer is: Lazio
	Rome$^\diamond$ is the capital of the Central Federal District.	(*Rome*$^\diamond$, capital of, Lazio)	The evidence suggests Rome is the capital of Lazio* not the Central Federal District.	

Table 3: Gnerated examples in HotpotQA by ChatGPT. We show the multi-turn retrofitting process of KGR. The *entity* with superscript* refers to retrofitted factual statements. The other green colors refers to critical entities in claims.

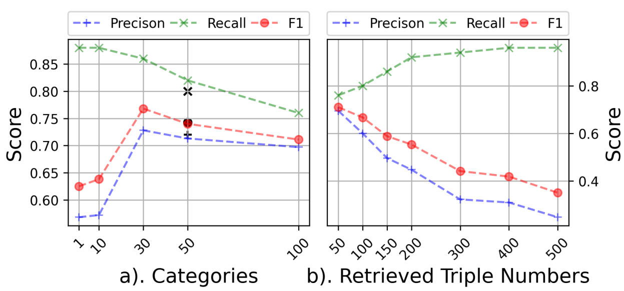

Figure 7: Impact of chunk size and numbers of retrieved triples on the efficacy of the fact selection. a). Results for fact selection were conducted on a sample of 50 instances from Simple Question with different chunk sizes. b). Results for fact selection were conducted on a sample of 50 instances from Simple Question with different retrieved triple numbers.

Impact of Chunk Size&Numbers of Retrieved Triples

As discussed above, considering the limitation of maximum input length for LLMs, we partition the retrieved triples into chunks for fact selection. We evaluate the effectiveness of fact selection when retrieved triples are in random order, referring black point in Figure 7 a) on the Simple Question using ChatGPT. These experiments help us understand fact selection behavior under various hyperparameters, optimize chunk size, and refine triple retrieval strategies for improved efficiency.

As shown in Figure 7 a), the chunk size has minimal impact on triple selection capability, except for a chunk size of 100, which may cause worse long-distance dependency modeling. However, reducing the chunk size leads to lower precision and higher recall scores. This indicates that a smaller chunk size increases the chance of selecting both critical and irrelevant triples. Additionally, we observe that prompting LLMs with triples in random orders doesn't significantly affect triple selection.

As shown in Figure 7 b), increasing the number of retrieved triples has a gradual positive impact on recall but significantly reduces precision. More retrieved triples may boost recall for critical knowledge and introduce numerous irrelevant triples, potentially compromising the effectiveness of the claim verification and negating the benefits of fact selection.

All in all, experiments show that the core difficulty of retrieving fact knowledge based on LLMs is the tradeoff between precision and recall. This observation points to future research on fact selection based on LLMs.

Conclusion

In this paper, we propose a knowledge graph-based retrofitting framework that effectively mitigates factual hallucination during the reasoning process of LLMs based on the factual knowledge stored in KGs.

Experiment results show that KGR can significantly improve the performance of LLMs on factual QA benchmarks especially when involving complex reasoning, which demonstrates the necessity and effectiveness of KGR in mitigating hallucination and enhancing the reliability of LLMs. As for future work, we plan to improve the effectiveness in each step of our KGR framework.

Acknowledgements

We thank all reviewers for their valuable suggestions. This research work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27020200, the National Natural Science Foundation of China under Grants no. 62122077, 62106251, 62306303 and 62076233.

References

Agarwal, A.; Gawade, S.; Azad, A. P.; and Bhattacharyya, P. 2023. KITLM: Domain-Specific Knowledge InTegration into Language Models for Question Answering. *arXiv preprint arXiv:2308.03638*.

Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. *arXiv preprint arXiv:2306.04136*.

Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2397–2430. PMLR.

Bordes, A.; Usunier, N.; Chopra, S.; and Weston, J. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Chase, H. 2022. LangChain.

Chen, A.; Pasupat, P.; Singh, S.; Lee, H.; and Guu, K. 2023. PURR: Efficiently Editing Language Model Hallucinations by Denoising Language Model Corruptions. *arXiv preprint arXiv:2305.14908*.

Commons, W. 2023. Main Page — Wikimedia Commons, the free media repository.

Gao, L.; Dai, Z.; Pasupat, P.; Chen, A.; Chaganty, A. T.; Fan, Y.; Zhao, V.; Lao, N.; Lee, H.; Juan, D.-C.; et al. 2023. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 16477–16508.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C. C. T.; Del Giorno, A.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; et al. 2023. Textbooks Are All You Need. *arXiv preprint arXiv:2306.11644*.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Ji, Z.; Liu, Z.; Lee, N.; Yu, T.; Wilie, B.; Zeng, M.; and Fung, P. 2022. RHO: Reducing Hallucination in Open-domain Dialogues with Knowledge Grounding. *arXiv preprint arXiv:2212.01588*.

Jiang, J.; Zhou, K.; Dong, Z.; Ye, K.; Zhao, W. X.; and Wen, J.-R. 2023. StructGPT: A general framework for Large Language Model to Reason on Structured Data.

Kang, D.; and Hashimoto, T. 2020. Improved Natural Language Generation via Loss Truncation. arXiv:2004.14589.

Lee, K.; Ippolito, D.; Nystrom, A.; Zhang, C.; Eck, D.; Callison-Burch, C.; and Carlini, N. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8424–8445. Dublin, Ireland: Association for Computational Linguistics.

Li, B.; Fang, G.; Yang, Y.; Wang, Q.; Ye, W.; Zhao, W.; and Zhang, S. 2023a. Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness. *arXiv preprint arXiv:2304.11633*.

Li, J.; Tang, T.; Zhao, W. X.; Wang, J.; Nie, J.-Y.; and Wen, J.-R. 2023b. The Web Can Be Your Oyster for Improving Large Language Models. *arXiv preprint arXiv:2305.10998*.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2023a. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172.

Liu, T.; Zheng, X.; Chang, B.; and Sui, Z. 2021. Towards Faithfulness in Open Domain Table-to-text Generation from an Entity-centric View. arXiv:2102.08585.

Liu, X.; Lai, H.; Yu, H.; Xu, Y.; Zeng, A.; Du, Z.; Zhang, P.; Dong, Y.; and Tang, J. 2023b. WebGLM: Towards An Efficient Web-Enhanced Question Answering System with Human Preferences. arXiv:2306.07906.

Lu, P.; Peng, B.; Cheng, H.; Galley, M.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; and Gao, J. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.

Manakul, P.; Liusie, A.; and Gales, M. J. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Meta. 2023. Wikimedia movement — Meta, discussion about Wikimedia projects.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2024-02-20.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Sen, P.; Aji, A. F.; and Saffari, A. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. *arXiv preprint arXiv:2210.01613*.

Tian, R.; Narayan, S.; Sellam, T.; and Parikh, A. P. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

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

Zhang, T.; Wang, C.; Hu, N.; Qiu, M.; Tang, C.; He, X.; and Huang, J. 2022. DKPLM: Decomposable Knowledge-enhanced Pre-trained Language Model for Natural Language Understanding. arXiv:2112.01047.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1441–1451. Florence, Italy: Association for Computational Linguistics.

Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; and Chang, B. 2023. Can We Edit Factual Knowledge by In-Context Learning? *arXiv preprint arXiv:2305.12740*.