CogMG: Collaborative Augmentation Between Large Language Model and Knowledge Graph

Tong Zhou[1] and **Yubo Chen**[1,2*] and **Kang Liu**[1,2,3] and **Jun Zhao**[1,2*]
[1]The Laboratory of Cognition and Decision Intelligence for Complex Systems
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Shanghai Artificial Intelligence Laboratory
tong.zhou@ia.ac.cn, {yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Large language models have become integral to question-answering applications despite their propensity for generating hallucinations and factually inaccurate content. Querying knowledge graphs to reduce hallucinations in LLM meets the challenge of incomplete knowledge coverage in knowledge graphs. On the other hand, updating knowledge graphs by information extraction and knowledge graph completion faces the knowledge update misalignment issue. In this work, we introduce a collaborative augmentation framework, CogMG, leveraging knowledge graphs to address the limitations of LLMs in QA scenarios, explicitly targeting the problems of incomplete knowledge coverage and knowledge update misalignment. The LLMs identify and decompose required knowledge triples that are not present in the KG, enriching them and aligning updates with real-world demands. We demonstrate the efficacy of this approach through a supervised fine-tuned LLM within an agent framework, showing significant improvements in reducing hallucinations and enhancing factual accuracy in QA responses. Our code[1] and video[2] are publicly available.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Achiam et al., 2023) have witnessed a surge in adoption for question-answering (QA) applications (Stelmakh et al., 2022). Despite their ability to produce engaging and coherent responses, these models are susceptible to generating hallucinated content and frequently encompass factually inaccurate information (Rawte et al., 2023). Xu et al., 2024 indicated that this inevitable symptom imputes their data (Kandpal et al., 2023), training (Liu et al., 2024a), and inference stages (Dziri et al., 2021). Fortunately, LLMs can leverage their comprehension and reasoning ability by referring to external knowledge sources to relieve hallucinations, such as documents (Lewis et al., 2020) and knowledge graphs (Sun et al., 2023). We concentrate on utilizing knowledge graphs (KGs), which provide a complementary strength to Large Language Models (LLMs) through their structured format and precise encapsulation of factual information. However, the utility of KGs in QA scenarios is hindered by the challenges of *incomplete knowledge coverage* and *knowledge update misalignment*.

Incomplete Knowledge Coverage: In principle, knowledge graphs possess the capability to encompass a vast array of information; however, they are also confronted with the challenge of achieving comprehensive coverage in their storage of knowledge. The explicitly encoded triples within the KG prove inadequate to exhaustively cover the knowledge required for practical QA scenarios. Existing approaches to augmenting QA systems with KG have primarily focused on improving parsing formal language (Xiong et al., 2024) or semantic relevance in retrieval knowledge triples (Wu et al., 2023), pursuing the corresponding knowledge prestorage in the KG for pre-defined questions. There is relatively limited attention given to the subsequent handling of queries that do not hit the knowledge graph.

Knowledge Update Misalignment: Current approaches to updating knowledge graphs primarily depend on two strategies: extracting knowledge triples from unstructured text (Wang et al., 2023; Xiao et al., 2023) (Information Extraction) and inferring unseen linkages through the analysis of existing connections between nodes (Yang et al., 2023) (Knowledge Graph Completion). These paradigms employed for updating KGs are characterized by their aimless and seemingly infinite nature and, therefore, do not fully address the mis-

[*] Corresponding author
[1] Project: https://github.com/tongzhou21/CogMG
[2] Video: https://youtu.be/WnkS0Qk_0OM

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 365–373
August 11-16, 2024 ©2024 Association for Computational Linguistics

alignment between the newly acquired knowledge and real-world user needs. This highlights a lack of proactive consideration in updating the knowledge graph to align better with user demands.

To address the above two challenges, this paper proposes a framework called CogMG for collaborative augmentation between LLM and KG. When a query exceeds the knowledge scope of the current KG, the LLM is encouraged to explicitly decompose the required knowledge triples. Subsequently, completion is done based on the extensive knowledge encoded in the LLM's parameters, serving as the reference for the final answer. The explicit identification of necessary knowledge triples serves as a means for model introspection to mitigate hallucination and proactively highlights deficiencies in the KG in meeting real-world demands. Moreover, identifying these triples allows for their automatic verification through retrieval augmented generation (RAG) with external documents. The retrieved relevant documents can also be a reference for manual review before incorporating triples into the knowledge graph. This continual and proactive process of knowledge updating enables the knowledge graph to meet actual knowledge demands gradually. Consequently, the LLM can leverage the augmented KG to improve its factualness in answering questions, forming a collaborative augmentation between LLM and KG. The main contributions of this paper are shown below:

- We propose the collaborative augmentation framework between LLM and KG, which is called CogMG. Address knowledge deficiency in LLMs and advocate actively updating the knowledge within the KG according to user demand.

- We fine-tune an open-source LLM to adapt the collaborative augmentation paradigm CogMG in an agent framework and demonstrate it by implementing a website system. The agent framework is modular and pluggable, and the system is interactive and user-friendly.

- According to a use-case presentation and the experimental results in various situations, we demonstrate the effectiveness of CogMG in updating knowledge proactively and enhancing response quality in various real-world QA scenarios.

2 Framework Design

The single iteration of CogMG framework comprises three steps: (1) Querying the Knowledge Graph: Large models utilize reasoning and planning capabilities to decompose queries and generate formalized query statements for querying the knowledge graph. (2) Processing Results: If results are returned successfully, detailed answers preferred by humans are integrated. If unsuccessful, the required triples are explicitly identified and broken down before being integrated into the answer. (3) Graph Evolution: Utilizing external knowledge verification and modification to incorporate triples that were not hit into the knowledge graph.

2.1 Querying Knowledge Graph

Given a knowledge-intensive question, we initiate our approach by deconstructing the corresponding formal query into sub-steps in natural language. This decomposition aids in elucidating the necessary and universal logic for querying knowledge graphs, ensuring our method's generalizability across various KG schemas. The LLM then calls a formal language parsing tool to execute the query. This tool receives the logically decomposed steps in natural language as input, translates them into a formal query language tailored to the target knowledge graph, and returns the query results.

2.2 Processing Result

Upon receiving the query results from KG, the LLM leverages its comprehension and reasoning capabilities to organize the final answer. If the query execution encounters errors, the LLM delineates the essential knowledge triples with unknown components based on decomposed steps. Suppose the complement of these triples could provide the necessary knowledge to answer the question. Subsequently, knowledge encoded within the model's parameters is utilized to complete these triples. And then, the model generates the final answer according to these facts. Note that the completion step is applicable to LLMs with capabilities of any level. Explicit the necessary knowledge not only mitigates the hallucination effect due to snowballing in the current output but also identifies knowledge gaps within the graph, thereby facilitating the enhancement of the graph's knowledge coverage. The incomplete knowledge triples, and their completions are logged for potential incorporation into the graph or further verification.

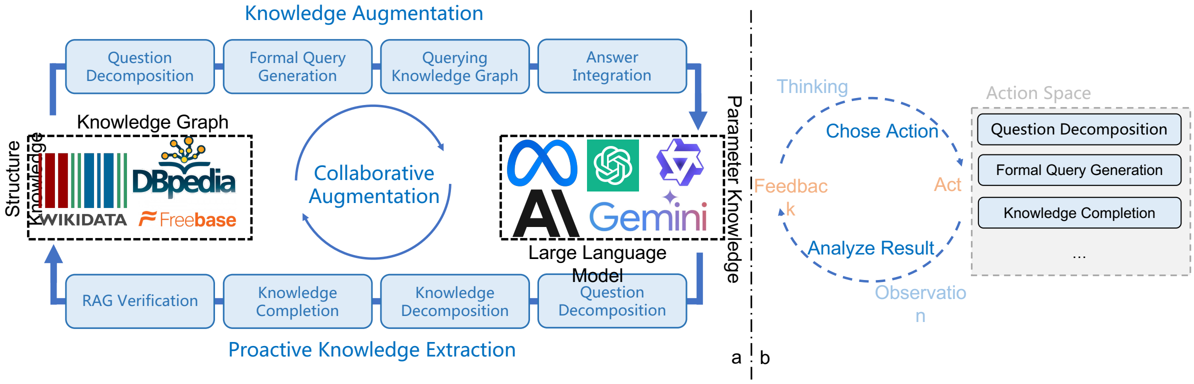


Figure 1: Left part: A schematic diagram illustrating the overall design of the collaborative augmentation framework CogMG, involving LLM and KG. Right part: We implement CogMG using an agent-based framework, with each module designed to be plug-and-play to ensure generalizability.

2.3 Knowledge Graph Evolution

The high generality and broad coverage of parameter knowledge encoded within LLM can supplement the more specialized knowledge in KG. These triples completed by LLM can be added to KG directly. However, the LLM struggles with rare, long-tail, and domain-specific knowledge and lacks robustness in its knowledge statement. We offer an option for manual intervention, where administrators can choose to (1) directly incorporate the completed triples into the knowledge graph, (2) manually adjust them before addition, or (3) verify them automatically according to external knowledge sources.

To automatically validate and correct these triples, CogMG searches related documents within unstructured corpora and makes comparisons in facts between documents and triples. These documents, which could drawn from domain-specific texts, general encyclopedias, or rapidly updated search engines, not only enhance the factual accuracy of the knowledge but also provide interpretable references for manual review. Based on the insights from these external sources, the model adjusts the proposed knowledge triples, making them suitable for manual inclusion into the knowledge graph.

3 Implementation and Usecase

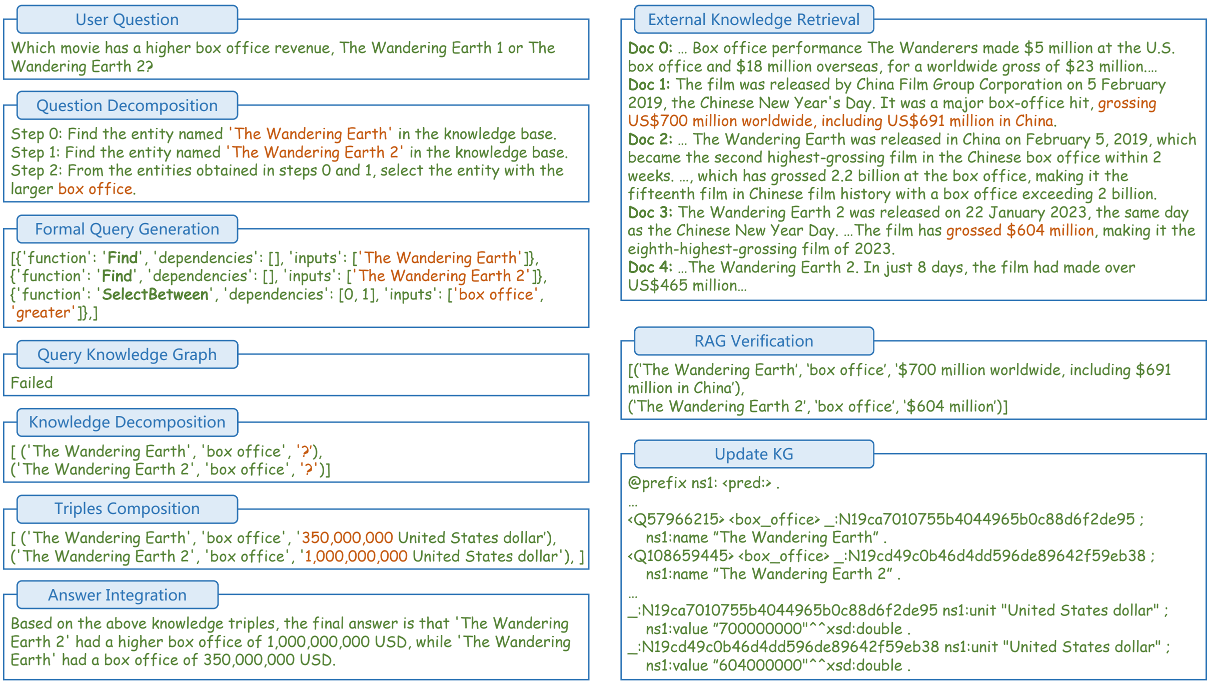
We fine-tune an open-source LLM for implementing the CogMG and develop an online system to demonstrate and evaluate our proposed collaborative augmentation framework.

3.1 Model and Components

LLM is capable of serving as an agent to plan and call tools independently (Qin et al., 2023). We adopted ReAct's agent framework (Yao et al., 2022) to adapt LLM to our proposition of modularization and generalization in CogMG's philosophy. We utilize a subset of Wikidata as the knowledge graph, KoPL (Cao et al., 2022) as the query engine, and the KQA-Pro dataset as the backbone of the fine-tuning dataset. KQA-Pro contains natural language questions with corresponding KoPL queries, SPARQL queries, and the gold answer. To ensure that the agent exhibits the expected behavior across various scenarios, we construct customized SFT datasets to fine-tune or utilize in-context learning to prompt the model. Qwen-14B-Chat (Bai et al., 2023) is responsible for all the SFT data generation and the agent backbone. Notably, our framework is applicable across various knowledge graphs and LLM. We will introduce our solution scenario by scenario.

Question Decomposition: Utilizing decomposition steps as intermediaries between questions and formal languages clarifies problem-solving logic and enhances robustness against different expressions of questions. We manually write several natural language explanations of query steps to obtain parallel data cases between questions and natural language explanations according to corresponding KoPL function calls. With these parallel examples, we prompt an LLM to generate sub-steps across the entire dataset and get 50k pairs of data. These data are reserved to construct agent behavioral SFT dataset.

Formal Query Generation: A model that parses the natural language explanations of query



User Question

Which movie has a higher box office revenue, The Wandering Earth 1 or The Wandering Earth 2?

Question Decomposition

Step 0: Find the entity named 'The Wandering Earth' in the knowledge base.
Step 1: Find the entity named 'The Wandering Earth 2' in the knowledge base.
Step 2: From the entities obtained in steps 0 and 1, select the entity with the larger box office.

Formal Query Generation

```
[{'function': 'Find', 'dependencies': [], 'inputs': ['The Wandering Earth']},
{'function': 'Find', 'dependencies': [], 'inputs': ['The Wandering Earth 2']},
{'function': 'SelectBetween', 'dependencies': [0, 1], 'inputs': ['box office',
'greater']},]
```

Query Knowledge Graph

Failed

Knowledge Decomposition

```
[ ('The Wandering Earth', 'box office', '?'),
('The Wandering Earth 2', 'box office', '?')]
```

Triples Composition

```
[ ('The Wandering Earth', 'box office', '350,000,000 United States dollar'),
('The Wandering Earth 2', 'box office', '1,000,000,000 United States dollar'), ]
```

Answer Integration

Based on the above knowledge triples, the final answer is that 'The Wandering Earth 2' had a higher box office of 1,000,000,000 USD, while 'The Wandering Earth' had a box office of 350,000,000 USD.

External Knowledge Retrieval

Doc 0: ... Box office performance The Wanderers made $5 million at the U.S. box office and $18 million overseas, for a worldwide gross of $23 million....
Doc 1: The film was released by China Film Group Corporation on 5 February 2019, the Chinese New Year's Day. It was a major box-office hit, grossing US$700 million worldwide, including US$691 million in China.
Doc 2: ... The Wandering Earth was released in China on February 5, 2019, which became the second highest-grossing film in the Chinese box office within 2 weeks. ..., which has grossed 2.2 billion at the box office, making it the fifteenth film in Chinese film history with a box office exceeding 2 billion.
Doc 3: The Wandering Earth 2 was released on 22 January 2023, the same day as the Chinese New Year Day. ...The film has grossed $604 million, making it the eighth-highest-grossing film of 2023.
Doc 4: ...The Wandering Earth 2. In just 8 days, the film had made over US$465 million...

RAG Verification

```
[('The Wandering Earth', 'box office', '$700 million worldwide, including $691
million in China'),
('The Wandering Earth 2', 'box office', '$604 million')]
```

Update KG

```
@prefix ns1: <pred:> .
...
<Q57966215> <box_office> _:N19ca7010755b4044965b0c88d6f2de95 ;
   ns1:name "The Wandering Earth" .
<Q108659445> <box_office> _:N19cd49c0b46d4dd596de89642f59eb38 ;
   ns1:name "The Wandering Earth 2" .
...
_:N19ca7010755b4044965b0c88d6f2de95 ns1:unit "United States dollar" ;
   ns1:value "700000000"^^xsd:double .
_:N19cd49c0b46d4dd596de89642f59eb38 ns1:unit "United States dollar" ;
   ns1:value "604000000"^^xsd:double .
```

Figure 2: A complete example of how the system handles queries not found in the knowledge graph and the processing of related knowledge.

steps to the KoPL formal program could be rapidly trained using the parallel data. Since the parsing process is relatively undemanding on the model's capabilities, we fine-tuned a 7B model to create a dedicated model in the tool of querying knowledge graph.

Querying Knowledge Graph: We wrapped the execution of the KoPL engine to uniformly return "Failed" upon errors, facilitating the model's decision-making and recognition. The query tool processes decomposed step inputs through a parsing model predicts the KoPL query program and returns the results of the knowledge graph query.

Answer Integration: The gold answers provided by KQA Pro are brief and precise at the word level and have gaps with the more detailed explanations preferred by humans. Hence, we supply the inference model with questions and gold answers from KG execution, instructing it to generate more exhaustive, explanatory responses to each question in the dataset. The answer integration scenario is a part of the agent behavior.

Knowledge Decomposition: We explicitly decompose the formal query's target triples to clarify the facts necessary for answering questions. This step is essential for manually annotating some query statements to incomplete triple, with unknown parts of facts expressed as question marks, and then using these samples as examples for the model to infer the triple decomposition for all data. Given the precise label names in the KoPL program as entity linking, we added label name constraints during triple inference, regenerating triples if non-standard label names were produced. All the knowledge decomposition data are utilized to simulate handling questions that the knowledge graph uncovered.

Knowledge Completion: We directly instruct the model to undertake knowledge completion tasks, referring to manually written examples. To fit the entire ReAct agent framework and ensure modularity, we encapsulate the knowledge completion part as a tool, inputting questions and corresponding incomplete knowledge triples to output the mappings of the parameter's knowledge with these triples.

Retrieval Augmented Generation Verification: Since LLM with general instruction tuning and preference alignment are familiar with RAG, we utilize prompt engineering to request the model to generate the correction of knowledge triples based on retrieved relevant documents, incomplete triples with question marks, and corresponding triples with parameter knowledge completion. We adopt Wikipedia as a retrieval corpus and segment every 256 tokens into a chunk. We build a document index by BM25, searching via concatenated knowledge triples and the origin question and selecting the top ten chunks as external knowledge references.

For the entire ReAct agent framework, we constructed two routes for the agent's planning and calling tools, differentiating whether the necessary knowledge is contained in the knowledge graph. Utilizing the built parallel training data, we construct two Thought-Action-Observation execution routes of SFT data, considering every scenario elaborated above. The agent is tuned using a total of 100k behavior SFT data.

3.2 System and Use Case

Knowledge Augmented Generation: Users can input and submit knowledge-intensive questions into the dialogue box at the bottom. The agent LLM is responsible for dealing with these questions and processes with pre-defined routes. The Thought-Action-Observation paradigm will be displayed in real time at the corresponding dropdown tab. When the knowledge graph cannot support the question-answering process, the model decomposes knowledge and invokes itself for knowledge completion before providing a final answer, as shown in the left of Figure 2. Meanwhile, these knowledge triples are recorded in the database.

Knowledge Management: In the Knowledge Management section of our system, we design an interactive interface to display all pending instances of knowledge that are not yet covered by the knowledge graph. The interface presents the origin of the query that highlighted the knowledge gap, the specific knowledge that is missing, and the results of the model's attempt to complete this knowledge based on its internal parameters. Administrators can (1) directly integrate this newly completed knowledge into the knowledge graph or opt for (2) further verification through RAG. A dropdown tab within the interface provides access to related documents and the outcomes of modifications, facilitating a rigorous validation process. Once the verification is complete and any necessary adjustments are made, administrators can seamlessly add the refined knowledge to the graph. This process not only ensures the continuous expansion and refinement of the knowledge graph but also leverages the administrators' expertise to validate the model-generated knowledge. By integrating these human-in-the-loop verification steps, our system enhances the reliability and accuracy of the knowledge graph, making it a more robust resource for answering real-world questions.

Method	Accuracy
Direct Answer	40%
CogMG w/o Knowledge	44%
CogMG Update	86%

Table 1: Comparison results of the accuracy of question answering in three different scenarios.

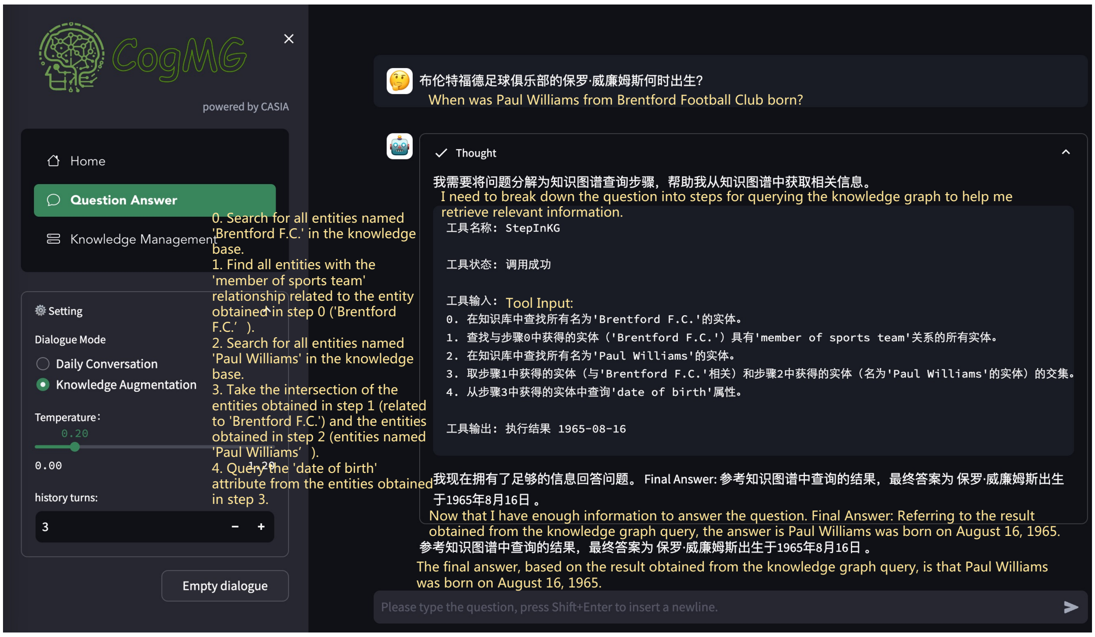
3.3 Experiments

We further designed and conducted experiments to demonstrate the effectiveness of the CogMG framework. Sampling questions from the KQA Pro dataset, we tested the following scenarios: (1) **Direct Answer**: answering using only the backbone LLM without utilizing the knowledge graph; (2) **CogMG w/o Knowledge**: deleting relevant knowledge from the graph and answering using parameter completion of knowledge; (3) **CogMG Update**: updating all relevant knowledge, utilizing the graph query results for answering. Due to the difficulty of exact matching in reflecting the correctness of real answers, we manually evaluated the correctness of 50 questions. Table 1 illustrates the accuracy under these three scenarios. Experimental results indicate that directly answering questions using LLM results in lower accuracy due to the lack of precise factual knowledge. Besides, leveraging the model's knowledge clarification and completion can alleviate some hallucinations and improve accuracy. Finally, the accuracy of subsequent inquiries is improved after utilizing the collaborative augmentation framework to update the knowledge graph.

4 Related Work

4.1 Knowledge Base Question Answering

Knowledge Base Question Answering (KBQA) aims to provide answers to natural language questions using Knowledge Bases (KBs) as their primary source of information (Bordes et al., 2015; Lan et al., 2019). Semantic parsing plays a crucial role by mapping questions to a formal language (Yih et al., 2016; Cai and Yates, 2013), enabling precise queries on knowledge graphs (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014). This task format can be regarded as a Seq2Seq paradigm, where formal language sequences are generated based on input question sequences. From RNN (Dong and Lapata, 2016) to BART (Cao et al., 2022) and GPT (Luo et al., 2023), the accuracy of formal language increases gradually with the ca-



Dialogue Mode
○ Daily Conversation
● Knowledge Augmentation

Temperature: 0.20
0.00 1.20

history turns: 3

布伦特福德足球俱乐部的保罗·威廉姆斯何时出生?
When was Paul Williams from Brentford Football Club born?

我需要将问题分解为知识图谱查询步骤，帮助我从知识图谱中获取相关信息。
I need to break down the question into steps for querying the knowledge graph to help me retrieve relevant information.

```
工具名称: StepInKG

工具状态: 调用成功

工具输入:
0. 在知识库中查找所有名为'Brentford F.C.'的实体。
1. 查找与步骤0中获得的实体（'Brentford F.C.'）具有'member of sports team'关系的所有实体。
2. 在知识库中查找所有名为'Paul Williams'的实体。
3. 取步骤1中获得的实体（与'Brentford F.C.'相关）和步骤2中获得的实体（名为'Paul Williams'的实体）的交集。
4. 从步骤3中获得的实体中查询'date of birth'属性。

工具输出: 执行结果 1965-08-16
```

Tool Input:
0. Search for all entities named 'Brentford F.C.' in the knowledge base.
1. Find all entities with the 'member of sports team' relationship related to the entity obtained in step 0 ('Brentford F.C.').
2. Search for all entities named 'Paul Williams' in the knowledge base.
3. Take the intersection of the entities obtained in step 1 (related to 'Brentford F.C.') and the entities obtained in step 2 (entities named 'Paul Williams').
4. Query the 'date of birth' attribute from the entities obtained in step 3.

我现在拥有了足够的信息回答问题。 Final Answer: 参考知识图谱中查询的结果，最终答案为 保罗·威廉姆斯出生于1965年8月16日 。
Now that I have enough information to answer the question. Final Answer: Referring to the result obtained from the knowledge graph query, the answer is Paul Williams was born on August 16, 1965.

参考知识图谱中查询的结果，最终答案为 保罗·威廉姆斯出生于1965年8月16日 。
The final answer, based on the result obtained from the knowledge graph query, is that Paul Williams was born on August 16, 1965.

Figure 3: System screenshot.

pability of generative models. Besides end-to-end generation, Chen et al., 2021 suggested first identifying the entities and schema involved in the problem separately and then utilizing the transducer to generate logical expressions, ensuring the accuracy of logical syntax. Finally, it employs a checker to enhance the semantic consistency of the logical form. With the assistance of LLM, KB-BINDER (Li et al., 2023) generates a draft of logical expressions using codex and then matches executable programs based on BM25 scores. Thanks to in-context learning, the process can be accomplished with just a few annotated examples. Moreover, LLMs' reasoning and planning capabilities can also serve as better assistants for utilizing knowledge graphs without additional training (Jiang et al., 2023a; Sun et al., 2023; Jiang et al., 2024; Liu et al., 2024b). However, these works mainly focus on answering questions within the confines of a given dataset without addressing scenarios where the knowledge graph lacks the necessary information for the question. The agent framework (Yao et al., 2022; Qin et al., 2023; Liu et al., 2023) allows for the autonomous selection of alternative tools when faced with knowledge graph misses by design. However, it does not utilize these gaps as opportunities to enhance the knowledge graph.

In summary, existing research either overlooks the issue of insufficient knowledge graph coverage or fails to use these deficiencies to improve knowledge graphs actively.

4.2 Updating Knowledge Graph

Information extraction concentrates on extracting triples from a wide range of unstructured texts to augment knowledge graphs with new knowledge. Subject to the model capabilities, the target needs to be split into serval sub-tasks. Named entities need to be identified in the text (Lample et al., 2016; Yu et al., 2020; Qu et al., 2023), followed by the classification of relationships among these entities (Miwa and Bansal, 2016; Peng et al., 2020; Cheng et al., 2021). OpenIE (Etzioni et al., 2008; Stanovsky et al., 2018; Kolluru et al., 2020), on the other hand, identifies subject-predicate-object triples in one go without being limited by predefined schemas in the knowledge graph. LLMs have unified various information extraction tasks, allowing a single model to generalize across all sub-tasks with supervised fine-tuning or a few examples as a demonstration (Lu et al., 2022; Lou et al., 2023; Wang et al., 2023; Zhu et al., 2023). On the other hand, knowledge graph completion (Zhang et al., 2023; Jiang et al., 2023b) through reasoning over existing knowledge can augment the graph by establishing connections between existing nodes.

Our advocated approach of active knowledge updating is more targeted and complements large-scale knowledge updates without contradiction, providing a supplementary mechanism.

5 Conclustion

We address two relatively overlooked issues in integrating Large Language Models (LLMs) and Knowledge Graphs (KGs): Incomplete Knowledge Coverage and Knowledge Update Misalignment. In response to these challenges, we introduce CogMG, a framework for the collaborative enhancement of LLMs and KGs. CogMG tackles the problem of answering questions with knowledge not covered in the graph by explicitly defining and completing relevant knowledge. Additionally, it actively collects and verifies knowledge requirements to update the graph. Furthermore, we fine-tune an LLM based on an agent framework to implement CogMG and develop a user-friendly interactive system to visualize its capabilities. Use cases and experimental results demonstrate the effectiveness of CogMG.

6 Limitations

In enhancing large language models with knowledge graphs, we do not introduce more complex and advanced methods such as planning, reasoning, and interaction. We believe that the application of these methods can further improve the effectiveness of the CogMG framework.

On the other hand, actively acquiring updated triples in the real world and automatically incorporating this knowledge into the knowledge graph without human intervention remains challenging. The operation and management of knowledge graphs by large language models are directions for our future work.

7 Acknowledgments

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDA27020203), the National Natural Science Foundation of China (No. 62176257).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qingqing Cai and Alexander Yates. 2013. Semantic parsing freebase: Towards open-domain semantic parsing. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 328–338.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. Kqa pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6101–6119.

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021. Retrack: A flexible and efficient framework for knowledge base question answering. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: system demonstrations*, pages 325–336.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43.

Nouha Dziri, Andrea Madotto, Osmar R Zaiane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2024. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. *arXiv preprint arXiv:2402.11163*.

Pengcheng Jiang, Shivam Agarwal, Bowen Jin, Xuan Wang, Jimeng Sun, and Jiawei Han. 2023b. Text augmented open knowledge graph completion via pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11161–11180.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. Knowledge base question answering with a matching-aggregation model and question-specific contextual relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(10):1629–1638.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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.

Bingbin Liu, Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2024a. Exposing attention glitches with flip-flop language modeling. *Advances in Neural Information Processing Systems*, 36.

Jiaxiang Liu, Tong Zhou, Yubo Chen, Kang Liu, and Jun Zhao. 2024b. Enhancing large language models with pseudo-and multisource-knowledge graphs for open-ended question answering. *arXiv preprint arXiv:2402.09911*.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, et al. 2023. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *arXiv preprint arXiv:2308.05960*.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13318–13326.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772.

Haoran Luo, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, et al. 2023. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *arXiv preprint arXiv:2310.08975*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Xiaoye Qu, Jun Zeng, Daizong Liu, Zhefeng Wang, Baoxing Huai, and Pan Zhou. 2023. Distantly-supervised named entity recognition with adaptive teacher learning and fine-grained student ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13501–13509.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. 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 885–895.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*.

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

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023. Instructuie: multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Yike Wu, Nan Hu, Guilin Qi, Sheng Bi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2023. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction. *arXiv preprint arXiv:2312.15548*.

Guanming Xiong, Junwei Bao, and Wen Zhao. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. *arXiv preprint arXiv:2402.15131*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Rui Yang, Li Fang, and Yi Zhou. 2023. Cp-kgc: Constrained-prompt knowledge graph completion with large language models. *arXiv preprint arXiv:2310.08279*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Wen-tau Yih, Matthew Richardson, Christopher 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.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.

Yichi Zhang, Zhuo Chen, Wen Zhang, and Huajun Chen. 2023. Making large language models perform better in knowledge graph completion. *arXiv preprint arXiv:2310.06671*.

Tong Zhu, Junfei Ren, Zijian Yu, Mengsong Wu, Guoliang Zhang, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, and Min Zhang. 2023. Mirror: A universal framework for various information extraction tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8861–8876.