Knowledge Graph Integration and Self-Verification for Comprehensive Retrieval-Augmented Generation

Chenyuan Wu*
Tingjia Shen*
Ruiran Yan*
wuchenyuan@mail.ustc.edu.cn
jts_stj@mail.ustc.edu.cn
yanruiran@mail.ustc.edu.cn
University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China

Hao Wang
University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
wanghao3@ustc.edu.cn

Zheng Liu
Beijing Academy of Artificial Intelligence
Beijing, China
zhengliu1026@gmail.com

Zhen Wang
Sun Yat-Sen University
Guangzhou, Guangdong, China
wangzh665@mail.sysu.edu.cn

Defu Lian
University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
liandefu@ustc.edu.cn

Enhong Chen
University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence
Hefei, Anhui, China
cheneh@ustc.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) has gained significant attention from both academic researchers and the industry as a promising solution to address the knowledge limitations of large language models (LLMs). However, LLMs often exhibit hallucination phenomena when employing RAG. To effectively address hallucination phenomena in a wide range of question types, we employ various choices and strategies. Specifically, we utilize LLaMA3's emergent self-verification capability to determine whether the given reference can adequately answer a particular question, thereby avoiding hallucination phenomena. Subsequently, by utilizing knowledge graphs to augment our knowledge base, we enhance contextual understanding and reduce hallucinations on RAG. LLM's advanced capabilities further enable us to effectively integrate and interpret the contents of knowledge graphs, ensuring more coherent and accurate responses. Finally, the effective handling of these diverse question types allows us to provide precise and informative answers, tailored to the specific requirements of each query. In general, our work comprehensively utilizes the advanced capabilities of LLM to enhance the robustness and credibility of our information retrieval system. This multi-faceted approach, coupled with a meticulous evaluation of references, ensures the delivery of high-quality responses, irrespective of the complexity of the questions.

*All authors contributed equally to this research.

KDD '24, August 25–29, 2024, Barcelona, Spain

CCS Concepts

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Natural language generation**.

Keywords

Retrieval Augmented Generation, Large Language Model

1 Introduction

In recent years, there has been a surge of interest in leveraging knowledge graphs in the development of Open-Domain Retrieval-Augmented Generation (RAG) models. These models aim to improve the generation of human-like responses by retrieving relevant information from extensive databases and incorporating it into the generated content. Knowledge graphs, which represent information in a structured and relational manner, have proven to be a valuable asset in enhancing the quality, relevance, and factual accuracy of generated outputs. Conventional methods usually entail a retriever-reader framework [5]. The evolution of retriever models has progressed from BM25 [17] and TF-IDF [1] to dense-vector-based approaches, such as DPR [12], SEAL [2] and BGE [26]. Concurrently, reader models have diversified, spanning from extractive readers like DPR-reader [4] to generative readers, such as FiD [10] and RAG [13], targeting span and free answering tasks respectively. Recently, the rapid development of Large Language Models (LLMs) has motivated researchers to incorporate them into OpenQA, driven by the models' advanced abilities in natural language reasoning.

However, despite these advancements, several critical challenges remain to be addressed. One prominent issue is the format complexity and diversity of the data utilized in these systems. This wide range of formats can lead to inconsistencies and difficulties in retrieval, representation, and integration of the information within

the RAG models. Additionally, these systems are prone to generating 'hallucinations' – fabricated information that appears plausible but is not grounded in the provided data or factual knowledge. Such misinformation can undermine the reliability and trustworthiness of the outputs produced by these AI systems.

In this report, we aim to explore these challenges in-depth and present potential strategies and methodologies for mitigating these issues, thereby enhancing the performance and reliability of Open-Domain RAG models integrated with knowledge graphs.

Our solution to the competition leverages a comprehensive approach that integrates knowledge graph integration and a self-check strategy. By combining these mechanisms, our model is able to generate high-quality answers. Specifically, in Task 3 of the competition, within the *simple_w_condition* category, our solution achieved an impressive score of 42.2%, securing the top position among all participants.

2 Related Work

2.1 LLMs-based Open-domain Question Answering

Recent years have witnessed a surge in interest in LLMs for their remarkable capabilities across various NLP tasks [9, 18, 22, 24]. Amidst this trend, handling OpenQA based on LLMs is emerging as a popular research direction. Research in this domain can be mainly divided into two categories: discriminative language models-based approaches and generative language models-based approaches. In the first type of approaches, researchers typically fine-tune BERT [6] or RoBERTa [16] to build a reader aligned to answering tasks [11, 25, 28]. Following the occurrence of generative language models, such as GPT [3], GLM [7], LLaMA [21], the researchers began adopting these models for OpenQA. This shift was driven by their proven capability to handle OpenQA without relying on retrievers [23]. Nevertheless, several studies persist in exploring the role of retrievers in enhancing the performance of generative large models on these tasks, such as REPLUG [19] or dataset regeneration [29], which employs retrieval systems to fetch relevant knowledge as prompts. There is also some research trying to enhance LLM itself by in-context learning [14], scaling [30] or knowledge editing [15], using certain prompts to make a better integration of external knowledge on LLMs or editing located parameters to address compositional failure.

3 Preliminary

3.1 CRAG Problem Definition

Given a domain knowledge base (e.g., Wikipedia) comprising a set of sentences $P = \{p_1, p_2, ..., p_{n_1}\}$, a specified answer format request F, and knowledge graph $G = \{V, E\}$, and each query question q_i within $Q = \{q_1, q_2, ..., q_{n_2}\}$, CRAG aims to train a model M that can extract relevant knowledge and generate the ideal answer $a_i = M(q_i, F, P, G)$ in response to each query, guided by the information in the knowledge base.

In authentic knowledge bases like Wikipedia, information is typically structured within web pages or documents. By segmenting it into sentence form, we can derive the set of sentences denoted as P. Take, for instance, the question q_i: "*Where was the initial awareness of the Chernobyl incident triggered?*". Given that the desired format F is Entity Style, the model is expected to leverage insights extracted from these sentences to generate an answer such that $a_i = M(q_i, F, P, G)$ ="*Sweden*".

3.2 Instruct Tuning on LLMs

In the OpenQA pipeline with LLMs, we employ a fine-tuned LLM denoted as the model M for answer formulation. The fine-tuning process of the LLM facilitates its adaptation to the distribution and domain knowledge relevant to downstream tasks. This process essentially encompasses an equivalent final objective loss, resembling that of autoregressive training, outlined as follows:

$$max_{\Phi} \sum_{x_i, a_i \in T} \sum_{t=1}^{|a_i|} log(P_{\Phi}(a_{i,t}|x, a_{i,<t})), \tag{1}$$

where Φ represents the parameters of LLM to be optimized, T denotes the training set, x refers to the input context encompassing both an instruction and a query question, and $a_{i,t}$ is the t-th token of the generated answer word. For all tunings of our experiment, we adopted the LoRA [8] method of lightweight fine-tuning by reducing the required GPU memory consumption.

4 Methodology

To enhance the RAG performance of our model, we focused on two primary objectives: improving the quality of external knowledge and reducing hallucination in generated responses. To address these objectives, we have designed two modules as depicted in Figure 1:

- **External Knowledge Module**: We combines retrieved web information with high quality structured knowledge form knowledge graph. As the knowledge graph knowledge is more accurate, we prioritized its use in our question answering system.
- **Self-Verification Module**: It enables the model to self-assess whether it can answer a query based on known external information.

In the following sections, we will delve into the specific mechanism of each module. The detailed prompts for each module are provided in A.1. Entity extraction prompts for other domains can be constructed analogously.

4.1 External Knowledge Module

Based on our analysis of public datasets, we observed that a significant portion of queries for which web pages do not provide direct answers can be answered using knowledge graphs. For instance, the query "what is the latest stock price of gdtc that's available today?" is a typical example where knowledge graphs can provide accurate and specific information with condition.

However, there are three main challenges when using knowledge graphs to enhance generation. (1) The knowledge graph offers APIs across multiple domains. To accurately determine which API to invoke for a given query, a highly precise domain classifier is essential. (2) Extracting entities accurately from queries and converting them into formats compatible with knowledge graph API calls presents a significant challenge. The inherent ambiguity in

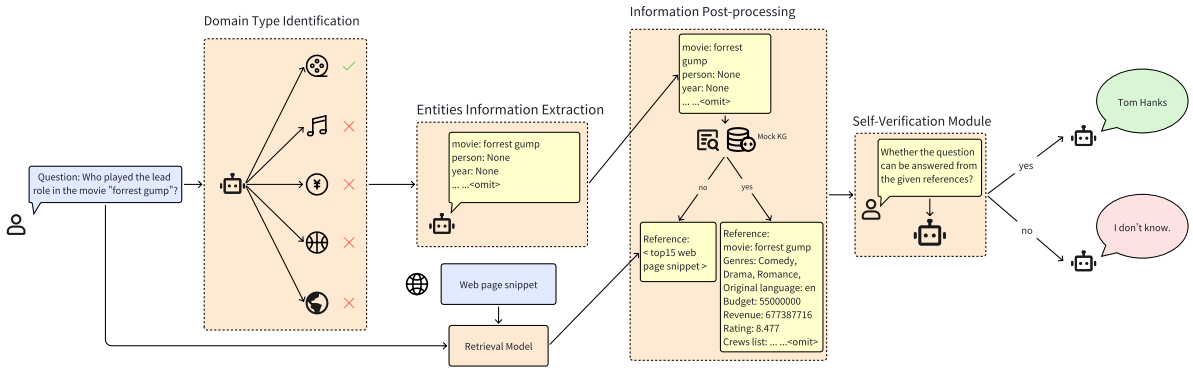

Figure 1: An illustration of our proposed pipeline for Comprehensive Retrieval-Augmented Generation.

entity expressions, such as "Last Tues" or "the year Kobe Bryant won MVP," necessitates complex reasoning processes to infer specific dates based on the current context. Furthermore, temporal references can be expressed as ranges, like "past five years". Additionally, the same entity can be represented in multiple ways, while knowledge graph APIs often require precise input. For instance, "Lakers", "LALakers", and "Los Angeles Lakers" refer to the same entity, but only the latter is accepted as a valid input for a specific knowledge graph API. (3) Knowledge graphs contain a vast amount of structured data, but only a small portion is relevant to answering a specific query. Extraneous data, such as stock prices at each time point on the given date, can potentially mislead the model, leading to hallucinations. Refining this structured data into concise and informative natural language representations that can be easily understood by large language models is a significant challenge. To address these challenges, we propose the following solution:

4.1.1 Domain Type Identification. We adopted a two-stage approach to identify the question domain, laying the groundwork for subsequent API calls. In the first stage, we employed a rule-based character matching approach to categorize questions into domains. This can be represented by the following equation:

$$D = \varphi(Q) \tag{2}$$

where D represents the specific domain and φ denotes the rule-based classifier applied to the query Q. For instance, we search for the phrase "stock price" to identify questions about stock prices. If no terms match the predefined rules, the query proceeds to the second stage. While this method offers high precision, it suffers from low recall. This limitation arises because entities can be expressed in multiple ways, and character-based matching may overlook queries containing alternative phrasings.

Therefore, we add a second stage identification based on LLM. Given a query Q, we prepend an instruction about domain classification to it. The domain is then determined by the LLM, as represented by the following equation:

$$D = LLM([I_{cls}; Q]) \tag{3}$$

where I_{cls} denotes the classification instruction. When a query is classified into a particular domain, knowledge graph augmented generation is activated. If not, retrieval-based generation is applied.

4.1.2 Entities Information Extraction. Once the specific domain and corresponding function are determined, our next critical step involves extracting entity information from the query to serve as parameters for API calls. To accomplish this, we leverage the efficient information extraction capabilities of LLMs. This process can be formally represented as:

$$E = LLM([I_{ext}; Q]) \tag{4}$$

where E represents the extracted entities, I_{ext} denotes the instruction for entity extraction, and Q is the input query.

However, LLMs may extract entities incorrectly, leading to downstream hallucination. We observed that LLMs have limited ability in performing date calculations. For instance, when tasked with calculating the date of "Last Tues", LLMs often mistakenly identify it as the Tuesday of the current week rather than the previous week. To mitigate this issue and enhance extraction accuracy, we augmented the instructions with common sense knowledge, such as the current day of the week and the explicit meaning of "Last Tues" as the Tuesday of the previous week.

Moreover, a given entity can be expressed in multiple ways, while knowledge graphs often require a standardized representation. To address this, we propose two strategies for aligning extracted entities with the knowledge graph's schema: (1) Restricting the entity extraction scope: When the target entity set is relatively small (e.g., NBA team names), we can constrain the extraction process within the instruction. (2) Post-processing for entity alignment: For larger entity sets, we can perform post-processing to align entity names with a standardized representation. For instance, stock names can be converted to ticker symbols for subsequent price queries.

4.1.3 Information Post-processing. We have designed a manual method to perform post-processing on the knowledge extracted from the knowledge graph, with the aim of enhancing the final performance. Specifically, we take the structured information extracted from the Mocked API, perform information filtering, and then convert it into natural language descriptive text as external knowledge for downstream LLM generation.

4.2 Self-Verification Module

For knowledge generated from the knowledge graph, given its high quality, we directly input it as context. For queries that yield no results from the knowledge graph or do not belong to specific domains, we employ web page retrieval augmentation. Specifically, for tasks 1 and 2, we use snippets from all five retrieved web pages as external knowledge. For task 3, we utilize the all-MiniLM-L6-v2 as the retrieval model and conduct a grid search on the number of pages, discovering that the top 15 web page snippets yield the best performance.

We observed that many web pages either lacked relevant information or contained misleading content, leading to elevated hallucination scores. To mitigate this issue, we introduced a self-verification module that enables the model to autonomously assess the utility of a page and reduce hallucinations.

We simply prompted the LLM to determine whether the given passages could answer the question. If not, the response will be "I don't know." By decomposing the generation process into a two-step procedure—first checking feasibility and then generating—we observed a significant reduction in hallucinations compared to a one-step generation approach.

5 Experimental Evaluation

We present the model setup, datasets, and evaluation metrics evaluating the effectiveness of our model below.

5.1 Model setup and Datasets

Due to its performance and efficiency, we selected the Llama-3-8B-Instruct model as our generator. For the retrieval model, we employ the all-MiniLM-L6-v2 model as the embedding model. The Mock API was utilized as the knowledge graph module. In addition, we extract more real-time data from Wikipedia to enhance retrieved data quality and the overall RAG performance.

We evaluated our model using the Comprehensive RAG (CRAG) dataset [27]. The dataset includes web search results and mock KGs to mimic real-world RAG retrieval sources. Web search contents were created by storing up to 50 pages from search queries related to each question. Mock KGs were created using the data behind the questions, supplemented with "hard negative" data to simulate a more challenging retrieval environment. Mock APIs facilitate structured searches within these KGs, and we provide the same API for all five domains to simulate Knowledge Graph access. This dataset is composed of three tasks:

- WEB-BASED RETRIEVAL SUMMARIZATION Participants receive 5 web pages per question, potentially containing relevant information. The objective is to measure the systems' capability to identify and condense this information into accurate answers.
- KNOWLEDGE GRAPH AND WEB AUGMENTATION This task introduces mock APIs to access information from underlying mock Knowledge Graphs (KGs), with structured data possibly related to the questions.
- END-TO-END RAG The third task increases complexity by providing 50 web pages and mock API access for each question, encountering both relevant information and noises.

5.2 Evaluation Metrics

The evaluation of the model's output consists of three possible states: correct, hallucination, and missing, which are scored as 1, -1, and 0 points, respectively. When the model outputs 'I don't know,' the output state is recorded as missing. Otherwise, the output is evaluated by GPT-4 with the input format: 'Question: {query} Ground truth: {ground truth} Prediction: {prediction}.' If the output contains the word 'accurate,' it is scored as correct; otherwise, it is scored as a hallucination. The final score is the average score across the entire dataset.

5.3 Overall Performance

5.3.1 Offline Performance. Table 1 presents the results of different designs of our methods on public test set. KG refers to Knowledge Graph Integration while Self-V. refers to Self-Verification. The results reveal that the integration of a knowledge graph significantly enhances the RAG performance. Specifically, this incorporation resulted in a 17% increase in accuracy and a 22% improvement in overall score. This increase can be attributed to the superior quality of knowledge provided by the knowledge graph, particularly in relation to long-tail entities [20] and time-sensitive information. Moreover, the Self-Verification module effectively mitigates hallucinations, leading to a notable improvement in the overall score.

Table 1: Ablation results on Task 2's public test set.

KG	Self-V.	Accuracy	Hallucination	Missing	Overall Score
✗	✗	0.260	0.290	0.449	-0.030
✗	✓	0.222	0.190	0.588	0.032
✓	✓	**0.395**	**0.150**	0.455	**0.245**

Table 2: Grid search results on web page snippets number.

Snippets Number	Top 5	Top 10	Top 15	Top 20
Overall Score	0.170	0.196	**0.214**	0.184

In task 3, a grid search was conducted to determine the optimal number of web page snippets. As shown in Table 2, the RAG model achieves its highest performance when using the top 15 snippets as the reference from web pages.

5.3.2 Online Performance. We present the performance of our model in phase 2 of the competition in Table 3.

Table 3: Performance of our model in phase2 Leaderboard.

Task	Task 1	Task 2	Task 3
Rank	6	4	3
Overall Score	0.174	0.228	0.210
Accuracy Score	0.363	0.405	0.365
Hallucination Score	0.189	0.177	0.155
Missing Score	0.447	0.417	0.480

We achieved commendable results across all tasks, which can be attributed to the unique design advantages of our model. Specifically, our outstanding performance in the END-TO-END RAG tasks highlights the model's exceptional capability to handle and process complex information. End-to-end training ensures that the model can learn to perform tasks in a holistic manner, integrating various sub-tasks seamlessly. This means that the model not only excels in understanding and generating responses for individual queries but also in maintaining coherence and context across a series of interactions. Furthermore, the model's ability to interpret and execute conditional logic enhances its performance, particularly in tasks that require high-level contextual reasoning and decision-making processes. The use of large-scale datasets during training also plays a significant role, as it enables the model to generalize well across different types of queries and scenarios.

In summary, the combination of these design advantages—advanced neural architectures, attention mechanisms, extensive training, and the capability to handle complex information—enables our model to achieve superior results across various tasks, especially in END-TO-END RAG, where processing complex information is crucial.

5.3.3 Final Performance. In evaluating the online performance of our model on the Comprehensive RAG (CRAG) dataset, one significant highlight is our remarkable achievement in Task 3, specifically within the *simple_w_condition* category. Our model secured an impressive score of 42.2%, earning the top spot among all participants. This accomplishment underscores several key advantages of our model, aligning with the objectives and benchmarks set in natural language processing and machine learning communities.

The high performance in the *simple_w_condition* category demonstrates the robustness and adaptability of our model in handling conditional statements and queries efficiently. Conditional queries often require models to interpret context, understand logical dependencies, and generate precise responses based on given conditions. The exceptional ability of our model to excel in this aspect indicates a strong and profound understanding of linguistic structures and semantics, which is crucial for tasks involving if-then scenarios, contextual reasoning, and complex decision-making processes.

6 Conclusion

In this paper, we propose an effective solution for comprehensive retrieval-augmented generation (RAG). By integrating a knowledge graph to enhance information and employing a self-verification strategy to mitigate hallucinations, our approach demonstrates superior performance in generating high-quality responses. Notably, our model achieves the top results in Task 3 within the simple-with-conditions category.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U23A20319, 62202443) and the Anhui Provincial Science and Technology Major Project (No. 2023z020006).

References

[1] Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management* 39, 1 (2003), 45–65.

[2] Michele Bevilacqua, Giuseppe Ottaviano, Patrick S. H. Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/cd88d62a2063fdaf7ce6f9068fb15dcd-Abstract-Conference.html

[3] 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 (2020), 1877–1901.

[4] Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. 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)*. Association for Computational Linguistics, Hong Kong, China, 5886–5891. https://doi.org/10.18653/v1/D19-1600

[5] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=HkfPSh05K7

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360* (2021).

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[9] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716* (2024).

[10] Gautier Izacard and Edouard Grave. 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *CoRR* abs/2007.01282 (2020). arXiv:2007.01282 https://arxiv.org/abs/2007.01282

[11] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics* 8 (2020), 64–77.

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[13] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[14] Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2023. Self-Prompting Large Language Models for Zero-Shot Open-Domain QA. arXiv:2212.08635 [cs.CL]

[15] Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and Patching Compositional Reasoning in LLMs. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 9668–9688. https://aclanthology.org/2024.findings-acl.576

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[17] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (apr 2009), 333–389. https://doi.org/10.1561/1500000019

[18] Tingjia Shen, Hao Wang, Jiaqing Zhang, Sirui Zhao, Liangyue Li, Zulong Chen, Defu Lian, and Enhong Chen. 2024. Exploring User Retrieval Integration towards Large Language Models for Cross-Domain Sequential Recommendation. arXiv:2406.03085 [cs.LG] https://arxiv.org/abs/2406.03085

[19] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652* (2023).

[20] Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and

Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 311–325. https://doi.org/10.18653/v1/2024.naacl-long.18

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[22] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1064–1072.

[23] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[24] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.

[25] Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-Gram: Pre-Training with Explicitly N-Gram Masked Language Modeling for Natural Language Understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 1702–1715. https://doi.org/10.18653/v1/2021.naacl-main.136

[26] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 641–649. https://doi.org/10.1145/3626772.3657878

[27] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024. CRAG – Comprehensive RAG Benchmark. arXiv:2406.04744 [cs.CL] https://arxiv.org/abs/2406.04744

[28] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining Language Models with Document Links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 8003–8016. https://doi.org/10.18653/v1/2022.acl-long.551

[29] Mingjia Yin, Hao Wang, Wei Guo, Yong Liu, Suojuan Zhang, Sirui Zhao, Defu Lian, and Enhong Chen. 2024. Dataset Regeneration for Sequential Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3954–3965.

[30] Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2024. Entropy Law: The Story Behind Data Compression and LLM Performance. *arXiv preprint arXiv:2407.06645* (2024).

A Appendix / supplemental material

A.1 List of Prompts

Domain Classification

You are provided with a question. Determine what domain the question is about. The domain should be one of the following: 'finance', 'sports', 'music', 'movie', 'encyclopedia'. If none of the domains apply, use 'other'. Don't answer other words except the domain. Answer without quotes.

Examples:
Question: how many family movies were there that came out in 1994?
Domain: movie

Question: what car did roman pearce drive in the second fast and furious movie?
Domain: movie

Question: can you provide me with the dates when indb's stock price closed at a lower value last week?
Domain: finance

Question: what's the eps of dks?
Domain: finance

Question: who has won more world series championships as a player and manager, babe ruth or joe torre?
Domain: sports

Question: what are the major sanctioning bodies in professional boxing?
Domain: sports

Question: what's the percentage change in spotify premium subscribers from the start of the 2015 fiscal year and the end of the 2020 fiscal year?
Domain: music

Question: how many different record labels has eminem been signed for?
Domain: music

Question: what's the cooling source of the koeberg nuclear power station?
Domain: encyclopedia

Question: which university has a higher student-to-faculty ratio, harvard or princeton?
Domain: encyclopedia

Question: {query}
Domain:

Prompt 1: Domain Classification

Movie Domain Entities Extraction

You are given a Text, if the Text is in movie domain please response 'Yes' in Answer1, otherwise answer 'No'. If 'Yes' in Answer1 and one or more film name appears in the Text, please provide them all in Answer2 and split with ';' , otherwise answer with 'None'. If 'Yes' in Answer1 and one or more people names appear in the Text, please provide them all in Answer3 and split with ';' , otherwise answer with 'None'.
Your answer should be simple and concise.
Here are some examples:

Question: who has been in more tv shows, emma stone or jennifer lawrence?
Answer1: No
Answer2: None
Answer3: None

Question: what's the language that eagle eye was released publicly in?
Answer1: Yes
Answer2: eagle eye
Answer3: None

Question: which film received better critic ratings from rotten tomatoes, shutter island or the wolf of wall street?
Answer1: Yes
Answer2: shutter island;the wolf of wall street
Answer3: None

Question: which director has the most movies under their belt, harrison smith or susan muska?
Answer1: Yes
Answer2: None
Answer3: harrison smith;susan muska

Question: what was mike epps's age at the time of next friday's release?
Answer1: Yes
Answer2: next friday
Answer3: mike epps

Question: {query}

Prompt 2: Movie Domain Entities Extraction

Stock Price Entities Extraction

You are given a query about stock price or volume and the query time. Your answer needs to follow the following rules.

1. What is the company name or ticker symbol entity in the query? Make sure your response appears within the query. If there are multiple company names, output them all separated by commas.
2. Then you should answer what date is the query ask for. If the queried date is in a range, provide one possible date values within the range.
3. Choose a time range for answer3 from options of "day, week, month, year, all time".
4. Choose a price option for answer4 from options of "Open, Close, High, Low, Volume".
5. Your answer should following the format of "Answer1: XXX, Answer2: YYYY-MM-DD, Answer3: XXX, Answer4: XXX". Your answer should be simple and concise.
6. It is known that 2024-02-28 is a Wednesday.
7. "tues" means Tuesday, "thurs" means Thursday, "fri" means Friday, "sat" means Saturday, "sun" means Sunday, "mon" means Monday, "wed" means Wednesday.
8. "last tues" or "past tues" means last Tuesday, not the Tuesday of this week.
9. Consider using yesterday's date if the query asked about the last trading day.
10. If the query is about "the past week", make sure answer2 is the last trading day of the past week.

Answer following the rules.
Query Time: {query_time}
Question: {query}

Prompt 3: Stock Price Entities Extraction

Self-Verification

You are provided with a question and various references. You need to judge whether the question can be answered from the given references.

Reference 1: page name: {page_name}, url: {url}, last modified: {last_modified}, snippet: {page_snippet}
Reference 2: page name: {page_name}, url: {url}, last modified: {last_modified}, snippet: {page_snippet}
Reference 3: page name: {page_name}, url: {url}, last modified: {last_modified}, snippet: {page_snippet}
.....<omit>

Using only the references listed above, can you answer the question? Answer with 'Yes' or 'No'.
Current Time: {query_time}
Question: {query}

Prompt 4: Self-Verification

Answer Generation
You are given a question and references which may or may not help answer the question. Your answer needs to follow the following rules.

1. Your goal is to answer the question in as few words as possible.
2.If the References do not contain the necessary information to answer the question, please respond 'i don't know' with no other words.
3. All False Premise questions should be answered with a standard response 'invalid question'.

Reference 1: {External knowledge from knowledge graph or web pages}.
Reference 2: {External knowledge from knowledge graph or web pages}.
Reference 3: {External knowledge from knowledge graph or web pages}.
.....<omit>

Using only the references listed above, answer the following question as few words as possible:
Current Time: {query_time}
Question: {query}

Prompt 5: Answer Generation