1024040810 LaTeX

Hu Yuzhe
Nanjing University of Posts and Telecommunications

May 19, 2025

Abstract

Graph-based Retrieval-Augmented Generation (RAG) methods have significantly enhanced the performance of large language models (LLMs) in domain-specific tasks. However, existing RAG methods do not adequately utilize the naturally inherent hierarchical knowledge in human cognition, which limits the capabilities of RAG systems. In this paper, we introduce a new RAG approach, called HiRAG, which utilizes hierarchical knowledge to enhance the semantic understanding and structure capturing capabilities of RAG systems in the indexing and retrieval processes. Our extensive experiments demonstrate that HiRAG achieves significant performance improvements over the state-of-the-art baseline methods.

1 Introduction

Retrieval Augmented Generation (RAG) (Gao et al., 2023[2]) (Lewis et al., 2020[8]) (Fan et al., 2024[10]) has been introduced to enhance the capabilities of LLMs in domain-specific or knowledge-intensive tasks. Naive RAG methods retrieve text chunks that are relevant to a query, which serve as references for LLMs to generate responses, thus helping address the problem of ”Hallucination” (Zhang et al.,2023[1]) (Tang and Yang, 2024[3]). However, naive RAG methods usually overlook the relationships among entities in the retrieved text chunks. To address this issue, RAG systems with graph structures were proposed (Edge et al., 2024[11]) (Liang et al., 2024[6]) (Zhang et al., 2025[7]) (Peng et al., 2024a[8]), which construct knowledge graphs (KGs) to model relationships between entities in the input documents. Although existing RAG systems integrating graph structures have demonstrated outstanding performance on various tasks, they still have some serious limitations. GraphRAG (Edge et al., 2024[9]) introduces communities in indexing using the Leiden algorithm (Traag et al., 2019[8]), but the communities only capture the structural proximity of the entities in the KG. KAG (Liang et al., 2024[4]) indexes with a hierarchical representation of information and knowledge, but their hierarchical structure relies too much on manual annotation and requires a lot of human domain knowledge, which renders their method not scalable to general tasks. LightRAG (Guo et al., 2024[5]) utilizes a dual-level retrieval approach to obtain local and global knowledge as the contexts for a query, but it ignores the knowledge gap between local and global knowledge, that is, local knowledge represented by the retrieved individual entities (i.e., entity-specific details) may not be semantically related to the global knowledge represented in the retrieved community summaries (i.e., broader, aggregated summaries), as these individual entities may not be a part of the retrieved communities for a query.

We highlight two critical challenges in existing RAG systems that integrate graph structures: (1) distant structural relationship between semantically similar entities and (2) knowledge gap between local and global knowledge. We illustrate them using a real example from a public dataset, as shown in Figure 1.

Challenge (1) occurs because existing methods overrely on source documents, often resulting in constructing a knowledge graph (KG) with many entities that are not structurally proximate in the KG even though they share semantically similar attributes. For example, in Figure 1, although the entities ”BIG DATA” and ”RECOMMENDATION SYSTEM” share semantic relevance under the concept of ”DATA MINING”, their distant structural relationship in the KG reflects a corpus-driven disconnect. These inconsistencies between semantic relevance and structural proximity are systemic in KGs, undermining their utility in RAG systems where contextual coherence is critical.

Challenge (2) occurs as existing methods (Guo et al., 2024) (Edge et al., 2024) typically retrieve context either from global or local perspectives but fail to address the inherent disparity between these knowledge layers. Consider the query ”Please introduce Amazon” in Figure 1, where global context emphasizes Amazon’s involvement in technological domains like big data and cloud computing, but local context retrieves entities directly linked to Amazon (e.g., subsidiaries, leadership). When these two knowledge layers are fed into LLMs as the contexts of a query without contextual alignment, LLMs may struggle to reconcile their distinct scopes, leading to disjointed reasoning, incomplete answers, or even contradictory outputs. For instance, an LLM might conflate Amazon’s role as a cloud provider (global) with its e-commerce operations (local), resulting in incoherent or factually inconsistent responses as the red words shown in the case. This underscores the need for new methods that bridge hierarchical knowledge layers to ensure cohesive reasoning in RAG systems.

To address these challenges, we propose Retrieval-Augmented Generation with Hierarchical Knowledge (HiRAG), which integrates hierarchical knowledge into the indexing and retrieval processes. Hierarchical knowledge (Sarrafzadeh and Lank, 2017) is a natural concept in both graph structure and human cognition, yet it has been overlooked in existing approaches. Specifically, to address Challenge (1), we introduce Indexing with Hierarchical Knowledge (HiIndex). Rather than simply constructing a flat KG, we index a KG hierarchically layer by layer. Each entity (or node) in a higher layer summarizes a cluster of entities in the lower layer, which can enhance the connectivity between semantically similar entities. For example, in Figure 1, the inclusion of the summary entity ”DATA MINING” strengthens the connection between ”BIG DATA” and ”RECOMMENDATION SYSTEM”. To address Challenge (2), we propose Retrieval with Hierarchical Knowledge (HiRetrieval). HiRetrieval effectively bridges local knowledge of entity descriptions to global knowledge of communities, thus resolving knowledge layer disparities. It provides a three- level context comprising the global level, the bridge level, and the local level knowledge to an LLM, enabling the LLM to generate more comprehensive and precise responses.

In summary, we make the following main contributions:

- We identify and address two critical challenges in graph-based RAG systems: distant structural relationships between semantically similar entities and the knowledge gap between local and global information.

- We propose HiRAG, which introduces unsupervised hierarchical indexing and a novel bridging mechanism for effective knowledge integration, significantly advancing the state-of-the-art in RAG systems.

- Extensive experiments demonstrate both the effectiveness and efficiency of our approach, with comprehensive ablation studies validating the contribution of each component.

2 Related Work

In this section, we discuss recent research concerning graph-augmented LLMs, specifically RAG methods with graph structures. GNN-RAG (Mavromatis and Karypis, 2024) employs GNN-based reasoning to retrieve query-related entities. Then they find the shortest path between the retrieved entities and candidate answer entities to construct reasoning paths. LightRAG (Guo et al., 2024) integrates a dual-level retrieval method with graph-enhanced text indexing. They also decrease the computational costs and speed up the adjustment process. GRAG (Hu et al., 2024) leverages a soft pruning approach to minimize the influence of irrelevant entities in retrieved subgraphs. It also implements prompt tuning to help LLMs comprehend textual and topological information in subgraphs by incorporating graph soft prompts. StructRAG (Li et al., 2024) identifies the most suitable structure for each task, transforms the initial documents into this organized structure, and subsequently generates responses according to the established structure. Microsoft GraphRAG (Edge et al., 2024) first retrieves related communities and then let the LLM generate the response with the retrieved communities. They also answer a query with global search and local search. KAG (Liang et al., 2024) introduces a professional domain knowledge service framework and employs knowledge alignment using conceptual semantic reasoning to mitigate the noise issue in OpenIE. KAG also constructs domain expert knowledge using human-annotated schemas.

3 Preliminary and Definitions

In this section, we give a general formulation of an RAG system with graph structure referring to the definitions in (Guo et al., 2024) and (Peng et al., 2024b).

In an RAG framework $\mathcal{M}$ as shown in Equation 1, LLM is the generation module, $\mathcal{R}$ represents the retrieval module, φ denotes the graph indexer, and ψ refers to the graph retriever:

$$\mathcal{M} = (LLM, \mathcal{R}(\varphi, \psi)). \quad (1)$$

When we answer a query, the answer we get from an RAG system is represented by a^*, which can be formulated as

$$a^* = arg \max_{a \in A} \mathcal{M}(a|q, \mathcal{G}), \quad (2)$$

$$\mathcal{G} = \varphi(\mathcal{D}) = \{(h, r, t)|h, t \in \mathcal{V}, r \in \mathcal{E}\}, \quad (3)$$

where $\mathcal{M}(a|q, \mathcal{G})$is the target distribution with a graph retriever $\psi(G|q, \mathcal{G})$ and a generator $LLM(a|q, G)$, and A is the set of possible responses. The graph database G is constructed from the original external database $\mathcal{D}$. We utilize the total probability formula to decompose M(a|q, G), which can be expressed as

$$\mathcal{M}(a|q, \mathcal{G}) = \sum_{G \in \mathcal{G}} LLM(a|q, G) \cdot \psi(G|q, G) \quad (4)$$

4 The Balala Framework

HiRAG consists of the two modules, HiIndex and HiRetrieval, as shown in Figure 2. In the HiIndex module, we construct a hierarchical KG with different knowledge granularity in different layers. The summary entities in a higher layer represent more coarse-grained, high-level knowledge but they can enhance the connectivity between semantically similar entities in a lower layer. In the HiRetrieval module, we select the most relevant entities from each retrieved community and find the shortest path to connect them, which serve as the bridge-level knowledge to connect the knowledge at both local and global levels. Then an LLM will generate responses with these three-level knowledge as the context.

4.1 Indexing with Hierarchical Knowledge

In the HiIndex module, we index the input documents as a hierarchical KG. First, we employ the entity-centric triple extraction to construct a basic KG G0 following (Carta et al., 2023). Specifically, we split the input documents into text chunks with some overlaps. These chunks will be fed into the LLM with well-designed prompts to extract entities V0 first. Then the LLM will generate relations (or edges) E0 between pairs of the extracted entities based on the information of the corresponding text chunks. The basic KG can be represented as

$$\mathcal{G}_0 = \{(h, r, t)|h, t \in \mathcal{V}_0, r \in \mathcal{E}_0\} \quad (5)$$

The basic KG is also the 0-th layer of our hierarchical KG. We denote the set of entities (nodes) in the i-th layer as Li where L0 = V0. To construct the i-th layer of the hierarchical KG, for $i \geq 1$, we first fetch the embeddings of the entities in the $(i-1)$-th layer of the hierarchical KG, which is denoted as

$$\mathcal{Z}_{i-1} = \{Embedding(v)|v \in \mathcal{L}_{i-1}\} \quad (6)$$

where $Embedding(v)$ is the embedding of an entity v. Then we employ Gaussian Mixture Models (GMMs) to conduct semantical clustering on Li−1 based on Zi−1, following the method described in RAPTOR (Sarthi et al., 2024). We obtain a set of clusters as

$$\mathcal{C}_{i-1} = GMM(\mathcal{L}_{i-1}, \mathcal{Z}_{i-1}) = \{\mathcal{S}_1, ..., \mathcal{S}_c\} \quad (7)$$

4.2 Retrieval with Hierarchical Knowledge

We now discuss how we retrieve hierarchical knowledge to address the knowledge gap issue. Based on the hierarchical KG Gk constructed in Section 4.1, we retrieve three-level knowledge at both local and global levels, as well as the bridging knowledge that connects them.

To retrieve local-level knowledge, we extract the top-n most relevant entities $\hat{\mathcal{V}}$ as shown in Equation 8, where Sim(q, v) is a function that measures the semantic similarity between a user query q and an entity v in the hierarchical KG Gk. We set n to 20 as default.

$$\hat{\mathcal{V}} = TopN(\{v \in \mathcal{V}_k | Sim(q, v)\}, n) \tag{8}$$

To access global-level knowledge related to a query, we find the communities $\hat{\mathcal{P}} \subset \mathcal{P}$ that are con- nected to the retrieved entities as described in Equa- tion 13, where $\mathcal{P}$ is computed during indexing in Section 4.1. Then the community reports of these communities are retrieved, which represent coarse- grained knowledge relevant to the user's query.

$$\hat{\mathcal{P}} = \bigcup_{p \in \mathcal{P}} \left\{ p | p \cap \hat{\mathcal{V}} \neq \varnothing \right\} \tag{9}$$

4.3 Why is HiRAG effective?

HiRAG's efficacy stems from its hierarchical architecture, HiIndex (i.e., hierarchical KG) and HiRetrieval (i.e., three-level knowledge retrieval), which directly mitigates the limitations outlined in Challenges (1) and (2) as described in Section 1.

Addressing Challenge (1): The hierarchical knowledge graph Gk introduces summary entities in its higher layers, creating shortcuts between entities that are distantly located in lower layers. This design bridges semantically related concepts efficiently, bypassing the need for exhaustive traversal of fine-grained relationships in the KG.

Resolving Challenge (2): HiRetrieval constructs reasoning paths by linking the top-n entities most semantically relevant to a query with their associated communities. These paths represent the shortest connections between localized entity descriptions and global community-level insights, ensuring that both granular details and broader contextual knowledge inform the reasoning process.

5 Environmental Evaluation

We report the performance evaluation results of HiRAG in this section.

Baseline Methods. We compared HiRAG with state-of-the-art and popular baseline RAG methods. **NaiveRAG** (Gao et al., 2022) (Gao et al., 2023) splits original documents into chunks and retrieves relevant text chunks through vector search. **GraphRAG** (Edge et al., 2024) utilizes communities and we use the local search mode in our experiments as it retrieves community reports as global knowledge, while their global search mode is known to be too costly and does not use local entity descriptions. **LightRAG** (Guo et al., 2024) uses both global and local knowledge to answer a query. **FastGraphRAG** (Circlemind, 2024) integrates KG and personalized PageRank as proposed in HippoRAG (Gutiérrez et al., 2025). **KAG** (Liang et al., 2024) integrates structured reasoning of KG with LLMs and employs mutual indexing and logical-form-guided reasoning to enhance professional domain knowledge services.

Datasets and Queries. We used four datasets from the UltraDomain benchmark (Qian et al., 2024), which is designed to evaluate RAG systems across diverse applications, focusing on long- context tasks and high-level queries in specialized domains. We used Mix, CS, Legal, and Agriculture datasets like in LightRAG (Guo et al., 2024). We also used the benchmark queries provided in Ultra- Domain for each of the four datasets. The statistics of these datasets are given in Appendix A.

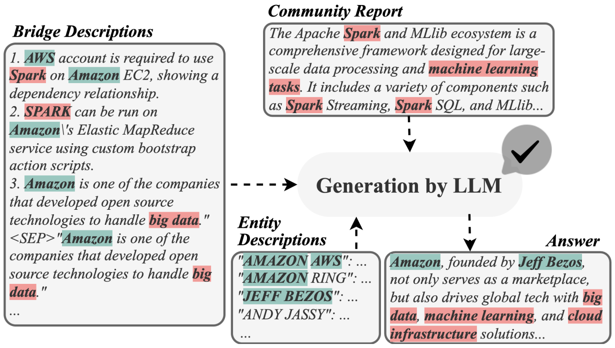
LLM. We employed DeepSeek-V3 (DeepSeek- AI et al., 2024) as the LLM for information extraction, entity summarization, and answer generation in HiRAG and other baseline methods. We utilized GLM-4-Plus (GLM et al., 2024) as the embedding model for vector search and semantic clustering because DeepSeek-V3 does not provide an accessible embedding model.

5.1 Overall Performance Comparison

Evaluation Details. Our experiments followed the evaluation methods of recent work (Edge et al., 2024)(Guo et al., 2024) by employing a powerful LLM to conduct multi-dimensional comparison. We used the win rate to compare different methods, which indicates the percentage of instances that a method generates higher-quality

Table 1: Win rates (%) of HiRAG, its two variants (for ablation study), and baseline methods.

	Mix		CS		Legal		Agriculture	
	NaiveRAG	**HiRAG**	NaiveRAG	**HiRAG**	NaiveRAG	**HiRAG**	NaiveRAG	**HiRAG**
Comprehensiveness	16.6%	<u>83.4%</u>	30.0%	70.0%	67.5%	34.0%	34.0%	66.0%
Empowerment	42.1%	<u>57.9%</u>	40.5%	59.5%	51.5%	49.0%	49.0%	51.0%
Diversity	12.7%	<u>87.3%</u>	14.5%	85.5%	22.0%	21.0%	21.0%	79.0%
Overall	12.4%	<u>87.6%</u>	26.5%	73.5%	24.5%	28.5%	28.5%	71.5%



1. **AWS** account is required to use **Spark** on **Amazon** EC2, showing a dependency relationship.
2. **SPARK** can be run on **Amazon**\'s Elastic MapReduce service using custom bootstrap action scripts.
3. **Amazon** is one of the companies that developed open source technologies to handle **big data**." <SEP>"**Amazon** is one of the companies that developed open source technologies to handle **big data**."
...

*The Apache **Spark** and MLlib ecosystem is a comprehensive framework designed for large-scale data processing and **machine learning tasks**. It includes a variety of components such as **Spark** Streaming, **Spark** SQL, and MLlib...*

"**AMAZON AWS**": ...
"**AMAZON** RING": ...
"**JEFF BEZOS**": ...
"ANDY JASSY": ...
...

***Amazon**, founded by **Jeff Bezos**, not only serves as a marketplace, but also drives global tech with **big data**, **machine learning**, and **cloud infrastructure** solutions...*

Figure 1: Answer to the query in Figure 1 with additional bridge-level knowledge.

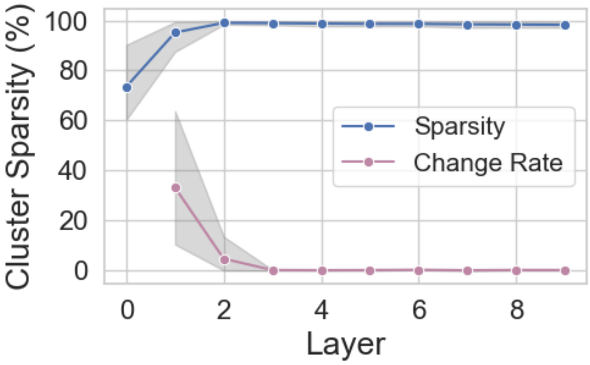


Figure 2: Cluster sparsity CS_i and change rate from CS_i to CS_{i+1}, where the shadow areas represent the value ranges of the four datasets and the blue/pink lines are the respective average values.

answers com- pared to another method as judged by the LLM. We utilized GPT-4o (Achiam et al., 2023) as the evaluation model to judge which method generates a superior answer for each query for the following four dimensions: (1) **Comprehensiveness**: how thoroughly does the answer address the question, covering all relevant aspects and details? (2) **Empowerment**: how effectively does the answer pro- vide actionable insights or solutions that empower the user to take meaningful steps? (3) **Diversity**: how well does the answer incorporate a variety of perspectives, approaches, or solutions to the prob- lem? (4) Overall: how does the answer perform overall, considering comprehensiveness, empower- ment, diversity, and any other relevant factors? For a fair comparison, we also alternated the order of the answers generated by each pair of methods in the prompts and calculated the overall win rates of each method.

5.2 Hierarchical KG vs. Flat KG

To evaluate the significance of the hierarchical KG, we replace the hierarchical KG with a flat KG (or a basic KG), denoted by w/o HiIndex as reported in Table 1. Compared with HiRAG, the win rates of w/o HiIndex drop in almost all cases and quite significantly in at least half of the cases. This ablation study thus shows that the hierarchical indexing plays an important role in the quality of answer generation, since the connectivity among semantically similar entities is enhanced with the hierarchical KG, with which related entities can be grouped together both from structural and semantical perspectives.

5.3 HiRetrieval vs. Gapped Knowledge

To show the effectiveness of HiRetrieval, we also created another variant ofHiRAG without using the bridge-level knowledge, denoted by w/o Bridge in Table 1. The result shows that without the bridge- layer knowledge, the win rates drop significantly across all datasets and evaluation dimensions, because there is knowledge gap between the local- level and global-level knowledge as discussed in Section 1.

Case Study. Figure 1 shows the three-level knowledge used as the context to an LLM to answer the query in Figure 1. The bridge-level knowledge contains entity descriptions from different commu- nities, as shown by the different colors in Figure 1, which helps the LLM correctly answer the question about Amazon's role as an e-commerce and cloud provider.

5.3.1 Determining the Number of Layers

One important thing in HiIndex is to determine the number of layers, k, for the hierarchical KG, which should be determined dynamically according to the quality of clusters in each layer. We stop build- ing another layer when the majority of the clusters consist of only a small number of entities, mean- ing that the entities can no longer be effectively grouped together. To measure that, we introduce the notion of cluster sparsity CSi, as inspired by graph sparsity, to measure the quality of clusters in the i-th layer as described in Equation 17.

$$CS_i = 1 - \frac{\sum_{\mathcal{S} \in \mathcal{C}_i} |\mathcal{S}|(|\mathcal{S}| - 1)}{|\mathcal{L}_i|(|\mathcal{L}_i| - 1)} \qquad (10)$$

6 Conclusion

We presented a new approach to enhance RAG systems by effectively utilizing graph structures with hierarchical knowledge. By developing (1) HiIndex which enhances structural and semantic connectivity across hierarchical layers, and (2) HiRetrieval which effectively bridges global conceptual abstractions with localized entity descriptions, Hi-RAG achieves superior performance than existing methods.

References

[1] Salvatore Carta, Alessandro Giuliani, Leonardo Piano, Alessandro Sebastian Podda, Livio Pompianu, and Sandro Gabriele Tiddia. Iterative Zero-Shot LLM Prompting for Knowledge Graph Construction. *arXiv e-prints*, page arXiv:2307.01128, July 2023.

[2] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise Zero-Shot Dense Retrieval without Relevance Labels. *arXiv e-prints*, page arXiv:2212.10496, December 2022.

[3] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv e-prints*, page arXiv:2410.05779, October 2024.

[4] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. GRAG: Graph Retrieval-Augmented Generation. *arXiv e-prints*, page arXiv:2405.16506, May 2024.

[5] Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting Knowledge Intensive Reasoning of LLMs via Inference-time Hybrid Information Structurization. *arXiv e-prints*, page arXiv:2410.08815, October 2024.

[6] Lei Liang, Mengshu Sun, Zhengke Gui, Zhongshu Zhu, Zhouyu Jiang, Ling Zhong, Yuan Qu, Peilong Zhao, Zhongpu Bo, Jin Yang, Huaidong Xiong, Lin Yuan, Jun Xu, Zaoyang Wang, Zhiqiang Zhang, Wen Zhang, Huajun Chen, Wenguang Chen, and Jun Zhou. KAG: Boosting LLMs in Professional Domains via Knowledge Augmented Generation. *arXiv e-prints*, page arXiv:2409.13731, September 2024.

[7] Costas Mavromatis and George Karypis. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. *arXiv e-prints*, page arXiv:2405.20139, May 2024.

[8] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia

Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, and Alec Radford. GPT-4 Technical Report. *arXiv e-prints*, page arXiv:2303.08774, March 2023.

[9] Yixuan Tang and Yi Yang. MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries. *arXiv e-prints*, page arXiv:2401.15391, January 2024.

[10] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *arXiv e-prints*, page arXiv:1809.09600, September 2018.

[11] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv e-prints*, page arXiv:2309.01219, September 2023.