Evaluating Multi-Document Inference in RAG Systems

Samuel Ostrove
Advisor: Robert Frank

Submitted to the faculty of the Department of Linguistics
in partial fulfillment of the requirements
for the degree of Bachelor of Arts
in Computing and Linguistics

May 2, 2025

Contents

Introduction

My first encounter with hallucination in a large language model (LLM) came early last Fall, when I asked ChatGPT about my great-grandfather Sam Ostrove, a one-time president of the New York City coffee shop chain Chock Full o'Nuts. The answer was rambling, generic, and inaccurate, focusing on Sam Ostrove's supposed legacy of "financial stewardship" enabling the brand to "sustain its operation and growth over the years." In fact, my great-grandfather's legacy was not one of sustainable growth: he was fired after three years for a failed product launch. ChatGPT's answer was mostly baseless, and certainly a gross mis-representation: such an answer is often called "hallucinated". Hallucination, or the generation of such "content that is nonsensical or unfaithful to the provided source content" (Huang et al., 2023, p. 1) significantly hinders the trustworthiness of LLMs.

Hallucination has three broad causes: data, training, and inference. Training data might be incomplete, biased, or poorly utilized. The model's architecture could be flawed, pre-training might be lacking, or alignment may induce errors and overconfidence. Finally, the model output methods (both the use of softmax and of long-tailed distributions) and context attention might lead to errors (Huang et al., 2023). Methodically exploring the impact of training on LLM hallucinations is impractical because of the large amount of computing power it requires, and because training hyperparameters for state-of-the-art models are closely guarded secrets even when the architecture is known. While it is always possible to change the methods by which language models sample from output distributions and to improve training data quality, obtaining broader coverage within the training data is not always possible: the knowledge boundary is a persistent problem.

Thus, this senior thesis seeks to contribute to our understanding of LLM-based systems in limited-information conditions. As explained above, training relevant models

from scratch on campus is not feasible. Fine-tuning, a type of secondary training procedure which aims to give models desirable conversational characteristics, could be interesting but would remain expensive—one might imagine fine-tuning a model several times with varying datasets, and examining the impact of datasets on model behavior. On top of the cost issue, interpreting the results objectively would be difficult as fine-tuning only affects model parameters. The clearest path forward is to study Retrieval-Augmented Generation (RAG), an architecture developed to deal with knowledge boundary issues which relies on augmenting a query to an LLM with contextual information from an external knowledge source (Lewis et al., 2021).

This project examines how the performance of RAG systems is affected by the structure of the external knowledge source. The goal is to understand when different types of RAG models fail to accurately answer a question which requires them to reason over multiple documents from the knowledge source. We hypothesize that the RAG system will do poorly when documents that must be retrieved do not share referents with the question, particularly due to limitations with the retrieval algorithm.

We begin this report with background on the RAG method: why it was proposed, where it is used, what variants are available and how they work, and what evaluation frameworks are currently available. Then, we describe the experiment in more detail: the hypothesis, the experimental protocol, and the current status of the project.

Hallucination as the Motivation for RAG

Hallucination

The basic impulse for this project was an observation about hallucination. This is a recent topic of inquiry in computational linguistics: the term first appeared in its current meaning in machine translation papers in 2017. The transformer architecture introduced that year (Vaswani et al., 2017) enabled such a large jump in performance for language models that hallucination went from an afterthought in neural machine translation to a major consideration across NLP subfields. Since 2017, over 2500 papers published to arXiv mention hallucination in their abstracts.[1]

No exact definition of the term "hallucination" has been widely accepted. The term can refer to false or confabulated answers, unreasoned imitation of training data, or over-generalizations (Maleki et al., 2024). In the case of this thesis, we will consider any response that does not answer the query truthfully (with respect to the external knowledge source) to be a hallucination.

Huang et al. (2023) usefully divide hallucinations into "factuality hallucination" and "faithfulness hallucination". The first of these concerns the relation between model output and the real world, whereas the second one describes cases when the model's output is not consistent with the input provided. My concern here lies within the latter, especially "factual fabrication" when models produce "facts that are unverifiable against established real-world knowledge" (Huang et al., 2023, p. 5).

[1] See arXiv.org, retrieved 2024/12/18.

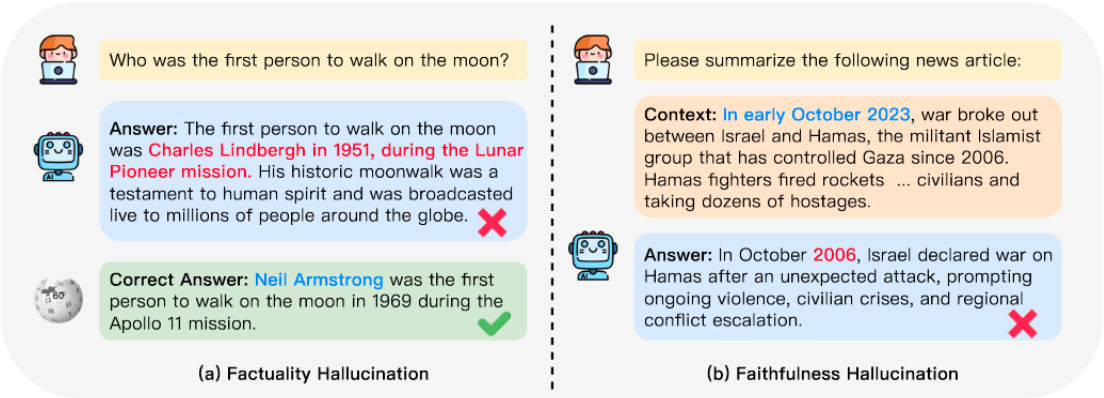

Figure 1 Examples of factuality and faithfulness hallucination, from Huang et al. (2023)

We should first ask how bad the problem of factuality hallucination is. Because language models generate open-ended responses, fact-checking models' outputs is difficult to reliably automate (Wang et al., 2024, p. 19521). One elegant approach to this problem is that taken by TruthfulQA (Lin et al., 2022), which was designed to examine whether language models "mimic popular misconseptions". The benchmark included questions which are commonly falsely answered by humans in texts that could make it into LLM training data. For example, questions tested adherence to well-known conspiracy theories ("Who really caused 9/11?") and common misconceptions ("Can coughing effectively stop a heart attack?"). The model under examination was evaluated twice on each question, once in an open-ended manner and once with a multiple-choice prompt. The answer to the open-ended question was evaluated by humans on the basis of truthfulness and informativeness, and the multiple-choice answer was evaluated automatically.

On the generation task, a human participant was used as a performance reference and produced 94% true answers. GPT-3-175-B scored highest of the models evaluated in the study at 58% true and informative answers. On the multiple-choice task, GPT-3-175B scored around 35%; GPT-4 with RLHF scored 59% (OpenAI et al., 2024). Interestingly, the authors noted a definite tendency of larger models to perform less well, both on the

generation and on the multiple-choice task. This would indicate that larger models imitate human falsehoods more.

Another frequently used benchmark is Massive Multi-Task Language Understanding (MMLU), which "tests both world knowledge and problem-solving ability" using a multiple-choice framework (Hendrycks et al., 2021). It is designed to evaluate use of knowledge acquired by models in pre-training, across a very wide variety of domains. Here comparison with a single human is not relevant, though human specialists do excel at this benchmark in their own field. GPT-3-175B performed at 42% average accuracy, though nearing 70% on some topics. More recently, GPT-4o achieved 88% and GPT-o1 got to 92% (OpenAI, 2024).

These high scores should nonetheless be set within the context of data availability. Zhao et al., (2024) use the existence or absence of a Wikipedia page for an entity as a proxy for the scale of its prevalence online. They show that factuality is significantly lower when answers concern entities without Wikipedia pages (less online information) for open-ended answers by a number of mainstream models including GPT-4o, Gemini-1.5-Pro, Claude 3 Opus, and Mixtral-8x22B.

Mitigating factuality hallucination

Multiple approaches have been proposed to deal with factuality hallucination: improving the quality of the pre-training data, inference (by changing decoding strategies or adding self-reasoning steps), or even implementing an additional fact-checking layer on top of an existing LLM (Wang et al., 2024, pp. 19522–19526). LLM providers also suggest "prompt engineering", a widespread practice which involves modifying the prompt to a language model in ways that encourage factuality.

Yet these techniques do not address a significant issue: they do not increase the number of true and informative answers on topics where information is not readily available on the internet or in the training corpora. One example of this was ChatGPT's

hallucinated biography of my great-grandfather, about whom online information is scarce.

Let us consider cases where a LLM's performance in a specific domain must improve. The basic requirement is information about this domain beyond what was used to train the model. This could be because such information is proprietary or because it was created after the model's training corpus was finalized. Two main approaches are possible. One is to fine-tune an existing model on domain-specific information which is either not; the other is to retrieve additional knowledge from a reliable source at the time of response generation. While fine-tuning might be more streamlined than retrieval, it requires significant computing resources (or budget) to be expended each time new information must be added to the model. In contrast, retrieval methods use an external source of knowledge which can be updated whenever necessary: they are the focus of this project. Moreover, (Ovadia et al., 2024) show that retrieval methods offer consistent improvement over unsupervised fine-tuning methods on knowledge-intensive tasks.

Multiple retrieval-based methods are available. We will concentrate on retrieval-augmented generation (Lewis et al., 2021), a method which has brought widespread attention (though actual adoption is near-impossible to measure.) We introduce the RAG architecture, then move on to existing methods for evaluating RAG systems.

The RAG Architecture

As initially proposed (Lewis et al., 2021), the RAG architecture is designed for sequence-to-sequence processing by LLMs. First, a set of documents is translated into embeddings to serve as an external knowledge source. Then, a prompt is submitted and embedded. A pre-trained retrieval mechanism selects the k most relevant documents from the knowledge source: these are the documents whose embedding produces the highest dot-product with the prompt's embedding. Those k documents are finally submitted to the LLM as supporting materials along with the initial prompt. The k most relevant documents

could either be selected once based on the prompt or selected anew for each token based on the entire sequence (prompt and previously-generated tokens.) In essence, then, RAG is an adaptation of semantic search to the LLM era.

In the initial proposal, the document embedding and retrieval component is borrowed from the question-answering system DPR (Karpukhin et al., 2020) whose goal is to index a very large number of documents "in a low-dimensional and continuous space" to then retrieve a small number of highly relevant documents. The documents and queries are embedded using two separate BERT networks, and the dot-product of the document embedding and the query embedding are used as a similarity score. The system then provides the k documents with highest similarity scores to the generator LLM, which is BART 400M in the original paper. The authors train the query-encoding BERT and output-generating BART together via stochastic gradient descent, holding the document-embedding BERT fixed to avoid re-embedding and re-indexing all the documents. Though there are variants on this system depending on the type of external documents in use, and how exactly the retrieval and augmentation work.

(Gao et al., 2024) give a comprehensive review of four years' worth of RAG research, from 2020 to 2023; they divide the different types of systems into three categories:

- Naïve RAG, which functions as described above. It is limited by issues of precision (how relevant the retrieved documents are) and recall (what proportion of the relative documents are retrieved), as well as by the integration of retrieved information with the initial prompt.
- Advanced RAG, which improves on the original embedding algorithm—changing how documents are split into chunks or incorporating ancillary information—and on how the query is processed—perhaps by splitting it into a chain of reasoning—and improves on how the retrieved data and the original query are written into a prompt.

- Modular RAG, which adds expanded functionality to RAG systems (like different types of external knowledge, or on-the-fly decision between different data sources.)

Only the first type of systems will be under evaluation here, as the heterogeneity of methods in the latter two categories would enlarge analysis beyond the scope of this project.

Existing Multi-Document RAG Benchmarks

Such a significant number of systems has led to a significant variety of evaluation frameworks, each serving slightly varying purposes (Yu et al., 2024). Only two, MultiHop-RAG (Tang & Yang, 2024) and BeerQA (Qi et al., 2021) focus on questions which require retrieving more than one document from the knowledge base. However, both of these methods share problematic characteristics:

- Dataset issues: BeerQA's knowledge source uses Wikipedia articles, which are highly likely to have been included in LLM training data. MultiHop-RAG's knowledge source and questions are LLM-generated based on news articles, a method which lacks the fine control of template-based synthetic data creation.
- MultiHop-RAG's questions always share semantic content (at least an entity name) with each of the documents to be retrieved, giving the RAG retriever more of a hint than we would like. It is not entirely clear whether this is the case with BeerQA.
- Finally, neither system controls for the number of documents in the external knowledge source, a factor which we consider to be crucial in illustrating the limits of RAG systems.

Because of the shortage in RAG evaluation frameworks which evaluate answers based on more than one document, and because of the methodological issues outlined

above, this project proposes a new, compact RAG evaluation framework for questions requiring more than one document to answer. The goal is to achieve repeatable and generalizable experiments with easily-interpretable results to guide further development of RAG and associated systems.

The experiment

The goal of this experiment is to understand how the structure of the external knowledge source affects the performance of a RAG system. By "structure" we mean the directness or indirectness of the information needed to answer a question, and how it is dispersed throughout an external knowledge source. By "performance" we mean how accurately and informatively a system can answer a question. We will evaluate different types of RAG systems, from the three categories proposed by (Gao et al., 2024). We will then attempt to draw detailed conclusions about the relative benefits of different architectures.

We approach this problem through the lens of the information in the external knowledge source, which we attempt to control quite thoroughly. Control over the external knowledge source allows the experimenter to clearly decide on the correct and incorrect answers to questions without much computational overhead. While a more complex version of this evaluation framework will be based on a relational graph as the basic structure for world knowledge, this version relies on simpler data structures which are described in the methods section below. This fully-synthetic ground truth allows us to explicitly generate both documents for the RAG system, and question-answer pairs for the evaluation process.

Knowledge Source and Query Generation

Family relations provide a simple, unified framework of relations between entities. Moreover, kinship relations relate to each other through a defined set of implications. For instance, if a is the child of b, and b is the child of c, then a is the grandchild of c (by the definition of the "grandchild(x,y)" relation). This framework evaluates the ability of RAG systems to answer questions using these implications.

We first provide two equally-long lists of names (one male and one female) to serve as entities in our synthetic family trees. We use gendered names and because of the ease they provide in specifying questions, though extending the framework to non-

gendered names should prove fairly simple. **The length of the name lists is the first variable parameter in this framework, determining the overall size of the knowledge source.** The smallest possible length is four male names and four female names, as will become clear below.

Then, we simulate generations by splitting the lists of male and female names. In each generation, we pair male and female names into "couples". Then we shuffle the lists and pair names again into sets of two "siblings", re-shuffling if needed to ensure that there are no "couples" of "siblings" within a generation. **The number of generations is the second variable parameter in this framework, determining the maximum number of documents required to answer a given question.** The smallest possible number of generations is two. A further version of this framework should allow for same-sex couples, sibling-less individuals, and sets of more than two siblings.

We establish parent (mother-father) and child (son-daughter) relationships by linking spouses from generation n with siblings from generation n+1. These entail grandparent (grandfather-grandmother) relations from each set of spouses in generation n-1 to two sets of siblings in generation n+1, and grandchild (grandson-granddaughter) relations. If there are more than three generations, great-grandparent relations are of course possible to implement. They also entail uncle and aunt, niece and nephew relations. Each relation is written into a csv file as "[name$_1$] is the [relation] of [name$_2$]." Each of these becomes a single "document" which is provided to the RAG system.

Simultaneously, for each relation we create a query and list the answers. For the relation "[name$_1$] is the [relation] of [name$_2$]", we create the query "Who is the [relation] of [name$_2$]?" and the answer "[name$_1$]". In our very simple model, only queries about the grandmother-grandfather and grandson-granddaughter relations admit more than one correct answer. These queries and answers are written to a separate csv file.

This results in two csv files, with a one-to-one correspondence between documents and queries. **We use this complete set of single documents answering single queries as**

a control for baseline performance of the retrieval portion of the system. Then, we create two csv files for each relation: a file containing all of the queries and answers about that relation, and a file containing all of the documents *except those with the given relationship.* **These sets of queries and documents serve as the experimental condition, where the model must use more than one document to answer the query.**

Evaluation mechanism

One goal of this framework is to have a simple evaluation mechanism which allows for open-ended answers (as opposed to multiple-choice ones) but does not involve using an LLM as a judge. An explicit evaluation mechanism has the advantage of eliminating that potential source of bias or error. It is also more portable, as using an LLM as a judge would require running two language models at once. Using an explicit evaluation mechanism is made possible by the simplicity of the answers required: just a name suffices.

Therefore, an answer is deemed correct if it includes one of the possible answers: a simple string comparison suffices. If the question is "Who is the brother of Amelia" and one of the possible answers is "Oliver", then any model output which contains "Oliver" counts as correct. Observation of model outputs and results shows no problematic cases (for example, "Oliver is not the brother of Amelia" never occurs).

Testing Procedure

A RAG system is evaluated first on the control questions, and then on questions from each of the experimental conditions. For each condition, the system's score is the proportion of queries that it answers correctly. We should note that each evaluation is carried out on a query-by-query basis, so that the prior queries and documents retrieved do not end up in the context window.

Many RAG systems also offer some degree of internal adjustability. The most common modifyiable parameter is the number of documents selected by the retrieval

system. **The number of retrieved documents forms the third parameter of this testing framework.** For this preliminary study, we tested the retrieval of 5, 15, 30, 45, and 80 documents.

Results

For this thesis, we evaluate two naïve RAG systems: the first RAG system (Lewis et al., 2021) and the implementation suggested on the LangChain blog using OpenAI embedding and generation models[2]. We evaluate the original system on a set of eighty relations, and the LangChain system on the same set of eighty relations as well as a separate set of 160 relations. In both cases, we limit ourselves to three generations.

The original RAG system of Lewis and colleagues does very poorly on our benchmark. With its default settings, on our smallest dataset size (12 entities, 80 relations) it answers 68% of the control questions accurately, and 0% of the experimental questions accurately. Increasing the purported number of retrieved documents seems to have no effect on the performance of the model. This might be an implementation issue, though. The LangChain implementation shows more promising results, which are displayed in Table 1 and discussed below.

Small Knowledge Source: 12 Names, 80 Relations

# Retrieved	Control	Grandfather	Grandmother	Sibling	Uncle	Aunt
5	0.95	0	0	0	0	0
15	0.99	0.5	0	0.08	0	0
30	1	0.75	0.75	0.33	0.5	0
45	1	1	0.75	0.33	0.25	0
80	0.986	1	0.5	0.25	0.25	0.25
120						

[2] https://python.langchain.com/docs/tutorials/rag/

Medium Knowledge Source: 24 Names, 160 Relations

# Retrieved	Control	Grandfather	Grandmother	Sibling	Uncle	Aunt
5	0.99	0	0	0	0	0
15	0.99	0.12	0	0.04	0	0
30	1	0.375	0.125	0.08	0	0
45	1	0.5	0.37	0.04	0	0
80	1	0.75	0.5	0.17	0	0
120	1	0.75	0.375	0.2	0	0

Table 1: Response accuracy for the LangChain RAG system. Divided by knowledge source size, and organized by number of documents retrieved for each query and type of relation interrogated by the query.

Discussion

The results of the original RAG system are not entirely surprising. The retrieval mechanism is only searching for the 5 best documents, using older BERT embeddings. Even if it retrieves the right documents, we observe in exploratory analysis that the BART-based language model has severe trouble with inference across multiple documents.

The LangChain system does much better overall. This is unsurprising, as it relies on much more modern embedding and generation models (OpenAI's "text-embedding-3-large" and GPT 4o-mini respectively). Its retrieval mechanism is very robust to large numbers of very similar documents: in the control condition, it scores at or above 95% correct answers on both smaller and larger knowledge sources with only 5 documents retrieved to answer each query.[3]

However, the number of retrieved documents strongly modulates the score on experimental conditions. This would tend to indicate that the retriever is not preferentially retrieving the necessary documents in cases when they do not directly answer the query. If the retriever were always retrieving the documents necessary to

[3] Strangely enough, the rare errors here were due to the generation model seemingly ignoring certain retrieved documents.

answering the query, the number of retrieved documents would have no effect on the accuracy score (as is the case for the control condition).

The LangChain system also exhibits unexpected patterns. Across experiments, the easiest multi-document inference to make is to infer the grandfather relation. There is a noticeable gendering effect: the grandfather relation is easier to infer than the grandmother relation, and the uncle relation is easier to infer than the aunt relation. Yet these relations only require wo documents to be retrieved: in theory, they should all be equally challenging. These discrepancies are puzzling, and their sources are unclear—though they could be related to frequency effects.

Explaining these disparities could be the object of future research using this framework. An expansion of this testing framework should consider relations outside of kinship to determine how widespread relation-based effects are. The effects are nonetheless interesting to note.

Another interesting testing parameter that could be included in a further version of this framework could be the proportion of documents in the knowledge source which are semantically unrelated to the question. One might hypothesize that it would be easier to retrieve documents relative to a query about a kinship relation if only a small percentage of the documents in the knowledge source relate to kinship relations.

Conclusion

This senior project proposes a new framework for the evaluation of retrieval-augmented generation systems. The framework tests a system's ability to reason over multiple documents which do not share referents with the query. Both the original RAG system and a more recent implementation (LangChain) perform much worse on the experimental task than on the control task, thereby quantifying a concerning issue. With the LangChain model, increasing the number of retrieved documents positively affected the accuracy of answers in the experimental condition but not in the control condition. This contrast tends to confirm our hypothesis about the source of this issue lying within the retrieval system.

Further work could extend this evaluation framework to different types of relations and use it to evaluate more complex types of retrieval-augmented systems. We expect that certain modular RAG systems which employ chain-of-reasoning query reformulation strategies and multiple retrieval steps will perform better than the naïve systems explored here.

My sincerest thanks to Professor Frank for his knowledge, time, and support over the past three years, and especially on this project. Great appreciation as well to Hannah Szabo and Will Min for their comments on preliminary stages of this work. Finally, my deepest gratitude goes to my family (Ariela Katz, Michael Ostrove, Shulamit Ostrove, Miriam Huerta, and others) for all their encouragement and perseverance.

References

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). *RAGAS: Automated Evaluation of Retrieval Augmented Generation* (arXiv:2309.15217). arXiv. https://doi.org/10.48550/arXiv.2309.15217

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey* (arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). *Measuring Massive Multitask Language Understanding* (arXiv:2009.03300). arXiv. https://doi.org/10.48550/arXiv.2009.03300

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2023). *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions* (arXiv:2311.05232). arXiv. https://doi.org/10.48550/arXiv.2311.05232

Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). *Dense Passage Retrieval for Open-Domain Question Answering* (arXiv:2004.04906). arXiv. https://doi.org/10.48550/arXiv.2004.04906

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

Lin, S., Hilton, J., & Evans, O. (2022). *TruthfulQA: Measuring How Models Mimic Human Falsehoods* (arXiv:2109.07958). arXiv. https://doi.org/10.48550/arXiv.2109.07958

Maleki, N., Padmanabhan, B., & Dutta, K. (2024). *AI Hallucinations: A Misnomer Worth Clarifying* (arXiv:2401.06796). arXiv. https://doi.org/10.48550/arXiv.2401.06796

OpenAI. (2024). *OpenAI o1 System Card*.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (arXiv:2303.08774; Version 5). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Ovadia, O., Brief, M., Mishaeli, M., & Elisha, O. (2024). *Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs* (arXiv:2312.05934). arXiv. https://doi.org/10.48550/arXiv.2312.05934

Qi, P., Lee, H., Sido, T., & Manning, C. (2021). Answering Open-Domain Questions of Varying Reasoning Steps from Text. In M.-F. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 3599–3614). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.292

Tang, Y., & Yang, Y. (2024). *MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries* (arXiv:2401.15391). arXiv. https://doi.org/10.48550/arXiv.2401.15391

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762; Version 5). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wang, Y., Wang, M., Manzoor, M. A., Liu, F., Georgiev, G. N., Das, R. J., & Nakov, P. (2024). Factuality of Large Language Models: A Survey. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 19519–19529). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.1088

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., & Liu, Z. (2024). *Evaluation of Retrieval-Augmented Generation: A Survey* (arXiv:2405.07437). arXiv. https://doi.org/10.48550/arXiv.2405.07437

Zhao, W., Goyal, T., Chiu, Y. Y., Jiang, L., Newman, B., Ravichander, A., Chandu, K., Bras, R. L., Cardie, C., Deng, Y., & Choi, Y. (2024). *WildHallucinations: Evaluating Long-form Factuality in LLMs with Real-World Entity Queries* (arXiv:2407.17468). arXiv. https://doi.org/10.48550/arXiv.2407.17468