Research and Analysis on the Mechanism of Suppressing Large Model Hallucination Based on Modular RAG Architecture

Zihan Lin [1, *], Xinle Yang [2]

[1] College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China

[2] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

* Corresponding Author Email: 20231650@mail.suep.edu.cn

Abstract. Large Language Models (LLMs) perform remarkably well in knowledge-intensive tasks, yet they are difficult to deploy in high-stakes scenarios due to 'hallucinations'—outputs that contradict factual information. Although Retrieval-Augmented Generation (RAG) is widely regarded as a mainstream approach to mitigate hallucinations, most existing studies treat it as a black box and rarely analyze the heterogeneous functions of its internal modules. To address this, we propose a 'functionally decomposed' modular RAG taxonomy, dividing the entire process into three stages: retrieval, refinement, and generation, from which three technical pathways are derived: Direct Injection RAG (DI-RAG), Relevance-Focused RAG (RF-RAG), and Fact-Checking RAG (FC-RAG). Utilizing the large-scale real-world Q&A dataset MS MARCO, we constructed four benchmarks simulating high-risk scenarios such as information noise, knowledge conflicts, and outdated knowledge. Using Qwen1.5-7B-Chat as the generative backbone, we systematically evaluate the marginal benefits of the three architectures in suppressing hallucinations and quantify the contributions of components like re-rankers and multi-query verifiers in specific hallucination scenarios, providing actionable empirical guidance and optimization pathways for building high-fidelity RAG systems.

Keywords: Large Language Models, RAG Architecture, retrieval, refinement, generation.

1. Introduction

In just the past few years, generative AI—powered by ever-larger language models—has vaulted from academic curiosity to industrial backbone. Systems such as Qwen and GPT-4 now compose prose, answer questions, and write code at or above human parity, reshaping everything from search to journalism [1]. Yet this prodigious fluency masks a fundamental fragility: hallucination. A hallucination is a sentence that sounds authoritative but is unmoored from reality—factually wrong, contextually incompatible, or wholly fabricated [2]. In high-stakes domains like medicine or law, a single such lapse can propagate dangerous misinformation, turning a helpful assistant into a liability.

To tether generation to ground truth, the field has coalesced around Retrieval-Augmented Generation (RAG). Instead of relying solely on static parameters encoded during training, RAG dynamically retrieves relevant, up-to-date evidence from external corpora—web pages, knowledge graphs, proprietary document stores—and feeds it to the model as a verifiable "fact anchor." By conditioning generation on this real-time context, RAG constrains the creative process, replacing opaque memorization with inspectable citation and sharply reducing the rate of hallucination [3].

Although the effectiveness of RAG has been widely recognized, and it can significantly improve factual consistency while keeping the generation cost within a reasonable range [4], existing studies often evaluate it as a single, holistic framework. A RAG system is essentially a complex pipeline composed of multiple functional components, mainly including three core stages: retrieval, refinement, and generation. The design choices of each stage - for example, whether to use sparse retrieval or dense retrieval, whether to re-rank the retrieval results to filter out noise, and how the generator handles conflicting retrieval information - may have a decisive impact on the factuality of its final output. If the effectiveness of RAG is discussed indiscriminately, it will not be possible to

reveal the specific mechanism of its internal components in combating hallucinations. Current classification research on RAG is mostly divided according to technical implementation paths (such as query-based or latent representation), which is helpful for understanding the system architecture, but fails to clearly answer a more practical question: how to minimize hallucinations by optimizing the configuration of the various components of RAG.

To fill this gap, this study proposes a modular RAG taxonomy based on component functionality. Instead of viewing RAG as an indivisible whole, this paper deconstructs it into a series of pluggable functional modules. Based on the complexity of the intermediate "knowledge processing" stage, these modules are divided into three progressive categories: direct-injection RAG (DI-RAG), relevance-focused RAG (RF-RAG), and fact-checking RAG (FC-RAG).

The core goal of this research is to systematically evaluate the performance of these three RAG architectures through controlled comparative experiments under specific hallucination-inducing scenarios, including information noise, factual conflicts, and outdated knowledge. This allows us to isolate and quantify the specific contributions of different processing modules to hallucination suppression. The research results will provide developers with practical guidance on how to optimize RAG components to build more hallucination-resistant systems.

2. Related Work

2.1. Tracing and Mitigating Hallucinations in Large Language Models

The hallucination problem of LLM has been widely studied in academia. The root causes of hallucinations are multifaceted, including factual errors in the training data, incomplete knowledge encoding in the model during parameterized memory, and random deviations caused by probabilistic sampling during the decoding phase [5]. Early mitigation methods mainly focused on the internal aspects of the model, such as by improving the training objectives or introducing factual reinforcement learning [6]. However, these methods have difficulty in solving the problem of the static nature of model knowledge, that is, they cannot adapt to the dynamic changes in the real world, which is precisely the main reason for the "outdated knowledge" hallucination.

2.2. RAG as a key technology to combat hallucinations

RAG fundamentally solves the problem of knowledge staticity by decoupling the vast world knowledge from model parameters and storing it in an external knowledge base that can be updated at any time. Its pioneering work was proposed by Lewis et al. (2020), who demonstrated that combining pre-trained retrievers and generators can achieve significant results on knowledge-intensive tasks [7]. Since then, RAG has quickly become a research hotspot.

2.3. Classification Perspective of RAG Component Technology

To clarify the exact contribution of different improvement strategies to hallucination suppression, this article maps existing work into the three major components of "retrieval-refinement-verification" based on functional roles, thereby illustrating the necessity of modular comparison.

Retrieval Component: Researchers have found that a single retrieval strategy has its limitations. Therefore, hybrid retrieval techniques have been proposed to combine the keyword matching capabilities of sparse retrieval (such as BM25) with the semantic understanding capabilities of dense retrieval [8], aiming to improve the comprehensiveness of retrieval recall.

Refinement Component: To address the problem of a large amount of noise in the preliminary retrieval results, re-ranking technology is widely used. Nogueira & Lin showed that using a strong cross-encoder like BERT to re-rank the preliminary retrieved paragraphs [9] can significantly improve the performance of the final task [10]. This constitutes the core idea of RF-RAG classification in this paper.

Verification Component: More cutting-edge research is beginning to give models a more proactive role. The Self-RAG framework proposed by Asai et al. allows the model to learn self-reflection and

judge the quality of the retrieved content [11], which coincides with the concept of “factual verification” in FC-RAG in this paper.

Although these component techniques exist independently, few studies have placed them within a unified framework, systematically comparing their performance in isolation when faced with specific hallucination triggers. This study aims to accomplish this.

3. Research Methodology

The core methodology of this study is controlled comparative experiments. This paper constructs three RAG systems with progressive complexity, which share the same underlying LLM and basic retrieval. The only variable is their "knowledge processing module" between retrieval and generation.3.1 Design Scheme: Three Modular RAG Architectures.

3.1. Design Scheme: Three Modular RAG Architectures

3.1.1. Basic Model

Throughout the experiment, Qwen1.5-7B-Chat was used as the generator. This model offers a good balance between parameter count and performance, and exhibits excellent command-following capabilities, making it the core generation component of this research. To leverage the complementary strengths of lexical and semantic matching, the base retriever employs a hybrid strategy, fusing the sparse signal of BM25 with the dense vector signal of BAAI/bge-large-en-v1.5. By performing parallel recall and merging the two results, the base retriever achieves the highest possible recall rate in the initial phase, providing a unified set of candidate documents for subsequent architectures.

3.1.2. Architecture 1: Direct Injection RAG

Direct injection RAG maintains a minimal pipeline. The original text blocks returned by the basic retriever are not processed in any additional way. Instead, they are simply spliced into a continuous context in order of score and directly fed into the Qwen1.5-7B-Chat generator. The resulting answers rely entirely on the knowledge learned by the model during the pre-training phase and the explicit information of the retrieval fragment. As a baseline, this architecture can clearly demonstrate the "unmodified" retrieval enhancement effect.

3.1.3. Architecture 2: Relevance Focus RAG

Relevance Focused RAG inserts a reranking step between retrieval and generation. After the base retriever delivers initial candidates, the BAAI/bge-reranker-large model fine-grainedly scores and reranks these fragments for relevance, retaining only the highest-scoring texts as the final context. This filtering step significantly reduces the noise encountered by the generator, making it easier to focus on information highly relevant to the question, thereby improving the accuracy and conciseness of the answer.

3.1.4. Architecture 3: Fact Checking RAG

The fact-checking RAG introduces a multi-query cross-validation module. It first uses the same LLM to generate multi-angle sub-queries around the original question, then independently calls the hybrid retriever for each sub-query to obtain corresponding evidence, and then aggregates the retrieval results of all sub-queries into the LLM, which comprehensively compares, identifies potential contradictions, and performs consistency checks. Finally, it outputs a verification text that has been conflict-resolved and factually self-consistent. This refined context is sent to the Qwen1.5-7B-Chat generator, thereby significantly reducing the risk of hallucinations caused by one-sided or conflicting information.

3.2. Data Collection: Building a Targeted Test Benchmark

To accurately evaluate hallucinations in different scenarios, this paper constructed four test sets based on the large-scale real question answering dataset MS MARCO and manually created another two.

FactualQA (ground truth question and answer set): Question and answer pairs with clear answers are selected from MS MARCO for evaluating basic performance.

DistractorQA (noise injection test set): uses the "negative samples" naturally existing in MS MARCO that are relevant to the question but have wrong answers to construct a test environment full of high-quality distractors.

3.3. Data Analysis Method: Three-Tier Evaluation System

In order to comprehensively measure the "accuracy" and degree of hallucination, this article adopts a multi-dimensional evaluation system:

Traditional automated metrics: Calculate the Exact Match (EM) and F1 Score to measure literal similarity.

Semantic similarity index: BERT Score is used to calculate the semantic similarity between the generated answer and the standard answer, making up for the shortcomings of traditional indicators.

LLM-as-a-Judge: Use the more powerful GPT-4o model as a judge, scoring each answer based on a "factual correctness" scale (1-5). This is the core metric used in this paper to evaluate hallucinations.

Both BERT Score and F1 Score are values between 0 and 1, with higher values representing better performance.

4. Project Implementation

4.1. Experimental Environment

This project was completed on the Alibaba Cloud DSW (Data Science Workshop) platform, using the GPU instance ecs.gn6i-c8g1.2xlarge. This instance is equipped with a single V100 accelerator card with 16GB of video memory, an 8-core vCPU, and 32 GiB of memory, which is sufficient to support parallel computing for large model inference and dense vector retrieval. At the software level, an isolated environment was built based on Conda, with Python version locked to 3.9. It also integrates mainstream toolkits such as PyTorch, Transformers, FAISS, and Sentence-Transformers to ensure that experiments are reproducible and dependencies are clear.

4.2. Experimental Procedure

The entire experimental process is shown in Figure 1. It begins with data preparation and index construction. By downloading the MS MARCO dataset and generating embeddings using the bge-large-en model, a dense (FAISS) and sparse (BM25) dual index is constructed. Next, during the test set generation phase, FactualQA and DistractorQA are automatically created from MS MARCO, and ConflictQA and FreshQA are manually augmented to comprehensively evaluate the RAG system. The core RAG pipeline implementation encodes four different strategies: Baseline, DI-RAG, RF-RAG, and FC-RAG. Subsequently, the batch experiment execution script automatically loads all test problems, runs each pipeline in sequence, and saves the results. Finally, the result evaluation script reads the experiment results, automatically calculates various metrics (including calling the GPT-4o API for high-level judgment), and generates a final scoring report, completing the entire experimental evaluation loop.

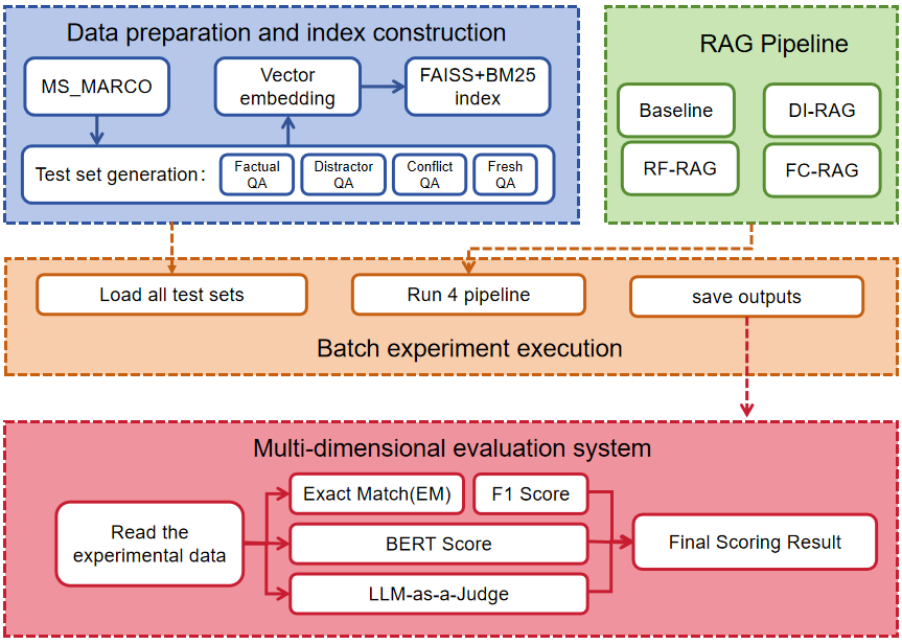

Figure 1. Experimental flow chart

4.3. Problems and Solutions

When scaling the RAG pipeline to the industrial scale of MS MARCO, which handles millions of queries and tens of millions of paragraphs, we assumed that the data could be loaded into memory all at once and that the field structure would be consistent with the mainstream training set. However, at the very beginning of the project, we encountered two major bottlenecks simultaneously: the BM25 index burst the memory immediately upon startup, and the training script crashed outright due to missing fields. Although these issues appeared different on the surface, they both stemmed from the rapid increase in data scale and complexity. To address this, we adopted a unified strategy of 'streaming reads, on-the-fly computation, dynamic parsing, and semantic alignment.' We split 8.8 million paragraphs into micro-batches of 10k shards according to shard order, feeding them into the BM25 builder one segment at a time. After calculating the term frequencies for a batch, we immediately released the original text, retaining only the incremental inverted index and global statistics, reducing peak memory usage from tens of GB to under 4 GB, allowing a single 32 GB instance to run the full-scale index stably. Meanwhile, by carefully comparing the schema of the MS MARCO dataset on Hugging Face, we found that it only uses `passage_text` and `is_selected` to implicitly label positive and negative samples. Therefore, we rewrote the parsing logic by labeling texts with `is_selected == 1` as positives and randomly sampling the rest as negatives, then grouping them by `query_id` into standard (query, positive, negative) triplets. This allowed the subsequent training process to proceed smoothly without any modifications, effectively isolating external uncertainty outside the training loop.

By combining streaming batching with field adaptation, we not only solved the bottlenecks of index building and training startup at one go, but also lowered the hardware threshold of the entire process to a common GPU cloud instance. Any subsequent new datasets or larger corpora can be quickly implemented using the same framework.

5. Results and Analysis

This section presents the core results of this experiment. We first tabulate the quantitative performance of four models (Baseline, DI-RAG, RF-RAG, and FC-RAG) on the FactualQA and DistractorQA test sets, using BERT Score and F1 Score metrics. We then conduct an in-depth analysis and discussion of this data to assess the specific role of different RAG components in suppressing model hallucinations.

5.1. Experimental Results

Through batch automated evaluation of 400 test samples (200 per dataset), we obtained performance data for each model in different scenarios. Detailed average scores are summarized in Table 1.

Table 1. Performance of each model on different test sets

Dataset	Metric	Baseline (No RAG)	DI-RAG	RF-RAG	FC-RAG
ConflictQA	LLM-Judge Score	3.8	3.933	3.667	4.2
	BERTScore	0.872	0.88	0.877	0.89
	F1 Score	0.221	0.254	0.255	0.306
DistractorQA	LLM-Judge Score	3.46	3.8	3.915	3.83
	BERTScore	0.82	0.84	0.842	0.841
	F1 Score	0.076	0.144	0.15	0.148
FactualQA	LLM-Judge Score	3.475	3.86	3.87	3.835
	BERTScore	0.821	0.841	0.842	0.842
	F1 Score	0.077	0.147	0.151	0.153
FreshQA	LLM-Judge Score	3.6	2.933	3.133	3
	BERTScore	0.886	0.893	0.891	0.897
	F1 Score	0.288	0.388	0.361	0.397

5.2. Data Analysis

As shown in the experimental results of Figure 2, we can clearly observe that the RAG paradigm has a significant effect in suppressing the illusion of large models. All RAG architectures significantly outperform the baseline model without RAG in terms of F1 Score and BERT Score metrics, especially in the FactualQA and DistractorQA datasets, where the F1 Score has increased nearly twice, indicating that the introduction of external knowledge can effectively enhance the model's ability to align with facts. Under various hallucination scenarios, the performance differences among different architectures are significant: in the ConflictQA dataset, which targets knowledge conflicts, FC-RAG, with its multi-query cross-verification mechanism, leads in all three metrics—large language model evaluation, BERT Score, and F1 score—demonstrating its advantage in handling contradictory information; meanwhile, in the DistractorQA dataset, which introduces a large amount of distracting information, RF-RAG, through its rearranger, effectively filters out noise and performs best across all evaluations, proving its enhanced resilience to interference. However, in the FreshQA dataset, which tests knowledge timeliness, all RAG architectures perform below the baseline, with a marked decline in large language model evaluation scores, indicating the RAG system's shortcomings in identifying outdated knowledge and its potential to introduce misinformation when faced with obsolete knowledge, exposing the limitation of relying on static knowledge bases. Moreover, although the differences among RAG architectures in BERT Score and F1 score are minor, the large language model evaluation scores reveal more pronounced performance gaps, suggesting the inadequacy of traditional automated metrics in measuring factual hallucinations and emphasizing the necessity of introducing more fact-oriented evaluation methods. Overall, the RAG system demonstrates positive effects in specific illusion scenarios through the collaborative action of different components, but its ability to handle knowledge timeliness still needs to be further enhanced. In the future, it should be combined with dynamic knowledge updates and a stronger fact-oriented evaluation mechanism to build a more reliable and trustworthy generation system.

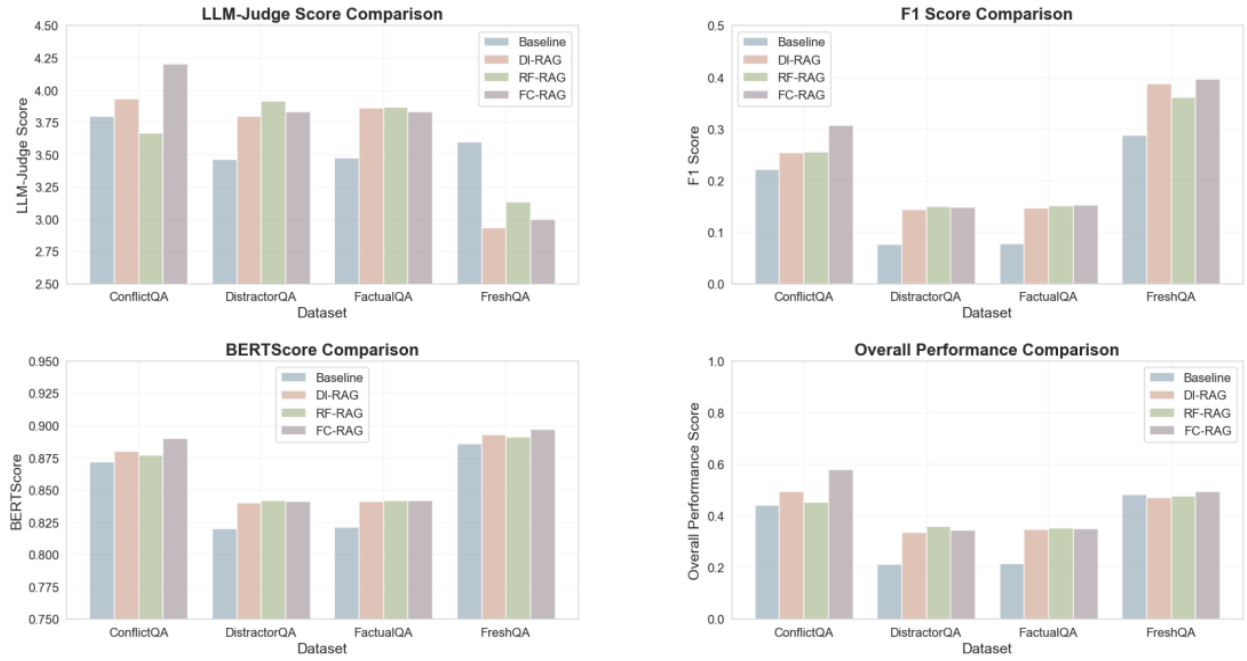

Figure 2. RAG Component Performance Analysis

5.3. Results and Discussion

Although the results of this experiment preliminarily verified the value of RAG components, they also triggered deeper thinking.

First, the fundamental capabilities of RAG have proven to be robust. With just the simplest direct injection (DI-RAG), the model's performance doubled. This demonstrates that providing LLM with external context is one of the most effective ways to suppress hallucinations and enhance factuality.

Second, the value of advanced components is not fully measured. The advantages of RF-RAG and FC-RAG over DI-RAG are not as significant as expected. This paper infers that this is mainly due to the limitations of the evaluation metrics. Metrics such as BERT Score and F1 Score, which are based on word overlap and semantic similarity, have limited ability to detect fact-based hallucinations. For example, for the question "What is the capital of France?", the answer "Berlin is the capital of France" has a certain degree of semantic and lexical similarity with the standard answer "Paris is the capital of France" and may obtain a high BERT Score, but it is actually completely wrong. The DistractorQA dataset was originally designed to induce such errors, and the existing metrics in this paper may not accurately capture the true advantage of RF-RAG in avoiding such errors.

Therefore, these preliminary results strongly suggest that to truly measure a RAG system's ability to combat hallucinations, and especially to distinguish the subtle but critical role of higher-level components like rearrangement and verification, an evaluation method more focused on factual correctness is necessary. The LLM-as-a-Judge evaluation method originally proposed in this paper will be key to addressing this issue. It can go beyond literal and semantic constraints and directly score the "factualness" of the answer.

6. Conclusion

This study focuses on the core issue of "how to decompose the RAG system into modules to suppress the hallucinations of large language models", and proposes a RAG classification method based on component functions. It systematically evaluated the performance of three architectures - Direct Injection (DI-RAG), Relevance Focusing (RF-RAG), and Fact Verification (FC-RAG) - in various hallucination-inducing scenarios. Experimental results show that the RAG paradigm significantly improves the factual consistency and generation accuracy of models, with its advantages being more pronounced in complex scenarios such as information noise and knowledge conflicts: RF-

RAG demonstrates stronger resistance to interference, while FC-RAG leads in resolving contradictions. However, in knowledge timeliness tests, all RAG architectures performed poorly, revealing the difficulty of static knowledge bases in eliminating outdated information. At the same time, traditional metrics have limited capability in measuring 'hallucinations,' further underscoring the necessity of introducing fact-oriented evaluation methods such as LLM-as-a-Judge. In summary, this study not only clarifies the differentiated roles of RAG internal modules in suppressing hallucinations but also provides empirical evidence and optimization pathways for building more reliable and scalable retrieval-augmented generation systems. In the future, it is urgent to tackle dynamic knowledge update mechanisms and establish more sensitive fact-oriented evaluation systems to enable large models to truly be applied in critical domains.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Z. Ji, et al. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys. (2023).

[2] G. P. Reddy, Y. V. Pavan Kumar and K. P. Prakash, Hallucinations in Large Language Models (LLMs), 2024 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, pp. 1 - 6, (2024). doi: 10.1109/eStream61684.2024.10542617.

[3] F. Cuconasu, G. Trappolini, F. Siciliano, et al. The Power of Noise: Redefining Retrieval for RAG Systems. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24). Association for Computing Machinery, New York, NY, USA, 719 – 729. (2024). https://doi.org/10.1145/3626772.3657834.

[4] Y. Mao, X. Dong, W. Xu, Y. Gao, B. Wei, & Y. Zhang. FIT-RAG: Black-box RAG with factual information and token reduction. ACM Transactions on Information Systems, 43 (2), 1 - 27. (2025).

[5] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, ... & T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems, 43 (2), 1 - 55. (2025).

[6] W. Su, C. Wang, Q. Ai, Y. Hu, Z. Wu, Y. Zhou, & Y. Liu. Unsupervised real-time hallucination detection based on the internal states of large language models. arXiv preprint arXiv: 2403.06448. (2024).

[7] P. Lewis, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems (NeurIPS). (2020).

[8] M. G. Arivazhagan, L. Liu, P. Qi, X. Chen, W. Y. Wang, & Z. Huang. Hybrid hierarchical retrieval for open-domain question answering. In Findings of the Association for Computational Linguistics: ACL (pp. 10680 - 10689). (2023).

[9] J. Devlin, M. W. Chang, K. Lee, & K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171 - 4186). (2019).

[10] R. Nogueira, & J. Lin. Passage Re-ranking with BERT. arXiv preprint arXiv: 1901.04085. (2019).

[11] A. Asai, Z. Wu, Y. Wang, A. Sil, & H. Hajishirzi. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. ArXiv, abs/2310.11511. (2023).