Reducing Hallucinations in Multi-Hop Question Answering Using Wikidata-Based Retrieval-Augmented Generation

Razi Alsayyed
Master's Program in AI
Department Of Natural Sciences
Faculty Of Graduate Studies
An-Najah National University
Nablus, P.O.Box:7, Palestine
razi@najah.edu

Shukri Khelfa
Master's Program in AI
Department Of Natural Sciences
Faculty Of Graduate Studies
An-Najah National University
Nablus, P.O.Box:7, Palestine
s12557805@stu.najah.edu

Amjad Hawash*
Department of ICS
Faculty of IT & AI
An-Najah National University
Nablus, P.O.Box:7, Palestine
amjad@najah.edu

***Abstract*—Hallucinations continue to significantly restrict large language models (LLMs), particularly in fact-finding and knowledge-intensive tasks. Retrieval-Augmented Generation (RAG), which bases model outputs on external knowledge sources, addresses this problem. This study examines the impact of integrating structured knowledge from Wikidata into an LLM-based question-answering pipeline to reduce hallucinated material. The proposed Wikidata-based RAG technique injects this structured context into the LLM prompt before response synthesis by first extracting entities from the input question and then gathering candidate entities and verified attributes from Wikidata. Both the prompt-only baseline models and the Wikidata-grounded RAG system were used to evaluate 80 multi-hop compositional problems. The proposed RAG model is evaluated using four hallucination evaluation methods. The model is evaluated against three prompt-only baselines: the AimonLabs hallucination-detection model, the Vectara HHEM, and an LLM-as-a-Judge rubric. Retrieved-Context Faithfulness (RAGTruth-style evaluation) is assessed using the last evaluation method. (Vectara HHEM: 77.5% vs 38.75% factual replies; AIMON: 60% vs 32.5% factual responses; LLM-as-a-Judge: 75% vs 38.75% factual responses) RAG consistently surpasses the baseline across all four evaluators. 54 of 79 assessed examples had a retrieved-context fidelity of 68.35%. This suggests that the observed frequency of hallucinations is influenced by the evaluation approach used. These results show how fact accuracy may be greatly increased by basing LLMs on structured knowledge from Wikidata. They also show how important it is to have several assessors when studying hallucinations.**

***Index Terms*—LLMs, retrieval-augmented generation, Wikidata, hallucination detection.**

I. Introduction

Large Language Models (LLMs) have revolutionised the field of natural language processing and enabled breakthroughs in multiple areas such as education, healthcare, research, and industry [1]–[3]. Nevertheless, despite their fluency and breadth of knowledge, LLMs frequently make hallucinations, which are responses that appear accurate but are either incorrect or lack supporting data [1]. Because hallucinations provide information that sounds plausible but is inaccurate or unsubstantiated, they jeopardize the safety, dependability, and trustworthiness of AI systems [3]. Users may lose faith in the system and begin to doubt its legitimacy if an AI produces inaccurate information, made-up references, or deceptive justifications. Such mistakes might result in bad choices and possible harm in crucial fields like healthcare, law, or education. For AI systems to be reliable, accurate, and safe to use, hallucinations must be reduced. Due to hallucination and static knowledge, LLMs must be used with caution as standalone systems in fact-seeking tasks. Apart from not having access to up-to-date or verifiable information beyond their training data, LLMs can generate confident but false claims [4].

Instead of accessing a factual database or verifying real-world reality, large language models (LLMs) create text by predicting the next word based on statistical patterns learned during training. This implies that there is no guarantee that the model is grounded in verifiable information, and errors may still occur [5]. This "probabilistic" generation may provide information that is fluent but inaccurate or unsupported. This shortcoming is directly related to how internal parametric knowledge—information contained in model weights—is acquired and used. Such parametric knowledge is opaque and difficult to verify since the vast network of parameters lacks visible logic, and it is static and subject to becoming obsolete after the model's training cutoff, meaning it won't reflect new or altered facts without outside intervention [6].

An important technique for addressing this problem is Retrieval-Augmented Generation (RAG) [3], [7], [8]. RAG systems often use document databases and knowledge graphs as external knowledge sources that support the LLM [2], [7]. Adding fresh information to the model enables it to provide responses that are more accurate [2], [8]. Reducing the model's reliance on its own knowledge -which may be inaccurate or outdated- usually helps decrease hallucinations and improve response accuracy [3], [8]. By using externally acquired data at inference time, Retrieval-Augmented Generation (RAG) has become a popular method for reducing hal-

*Corresponding Author: amjad@najah.edu

lucinations in Large Language Models (LLMs). The majority of the unstructured textual sources used by current RAG frameworks—such as Wikipedia articles, PDFs, and online documents—are retrieved via dense vector similarity [9]. Although these techniques enhance response grounding, they are nevertheless vulnerable to issues such as retrieval noise, semantic ambiguity, and redundant material. Furthermore, trustworthy entity disambiguation and relational reasoning are limited in unstructured text due to the lack of defined structural semantics. Despite these difficulties, despite structured knowledge graphs' benefits in accuracy, consistency, and interpretability, their use for hallucination reduction has gotten relatively little attention [10].

This study focuses on using structured knowledge from Wikidata through its SPARQL service [8]. It also uses the abilities of LLMs to generate SPARQL queries. The goal is to study the effect of using knowledge graphs RAG on LLM hallucinations. However, our contribution can be summarised in:

1) A Wikidata-grounded, agentic retrieval-augmented generation pipeline is introduced that performs entity discovery, candidate disambiguation, and verified property retrieval through SPARQL before answer synthesis.
2) A benchmark of 80 multi-hop compositional question-answering tasks is constructed with manually curated ground-truth references primarily derived from Wikipedia, providing a controlled setting for factuality comparison between prompt-only and Wikidata-grounded generation.
3) Hallucination reduction is evaluated using three complementary automated factuality evaluators—Vectara HHEM, AIMon HDM-2, and an LLM-as-a-Judge rubric—demonstrating consistent factuality improvements of Wikidata-grounded RAG over prompt-only prompting across evaluation frameworks.

The rest of this paper is organised as follows: Previous works are presented in Section II. Section III discusses the proposed methodology of the work. Section IV discusses the tests conducted to measure the reliability of the proposed methodology. Section V discusses the results of the tests. Section VI illustrates the limitations of the work. Finally, Section VII concludes this paper.

II. Previous Works

In LLMs, hallucinations occur when the model generates text that appears accurate but is not supported by solid evidence. This problem appears because the model mostly relies on statistical patterns that were discovered during training. Because of this, it might overgeneralise, misinterpret ambiguous queries, or omit information that calls for current or domain-specific expertise [1], [7], [11]. In fields that primarily rely on accurate knowledge and open-ended activities, where inaccurate information might have detrimental effects, hallucination becomes more severe [1], [3], [11]. This is particularly evident in recent deep learning applications for complex medical diagnoses, such as thalassemia detection, where the cost of factual error is high [12].

Modern LLMs that rely on transformer-based architectures, such as the Generative Pre-trained Transformer (GPT) family, are trained on large text datasets [13]. Models such as GPT-3 and GPT-4 have strong natural language understanding and generation abilities, including question answering, summarisation, and cross-domain reasoning. These models can adapt to a range of tasks without explicit fine-tuning thanks to self-supervised pretraining and in-context learning [14]. Despite their fluency and generalisation ability, GPT-like models have well-known limitations, including hallucinations, a lack of explicit grounding, and dependency on static training material. These models do not automatically verify facts from external references when generating responses, so they may provide information that appears reliable but is incorrect or outdated. This has led to incorporating external retrieval techniques and organised knowledge sources into LLMs [15].

By linking LLMs with retrieval components that deliver pertinent information or documents from outside sources, Retrieval-Augmented Generation (RAG) can lessen hallucinations. In order to help the model base its output on actual and verifiable facts, this collected information is added to the model's prompt or internal processing [1], [3], [8]. In various tasks, including question answering, summarising, and creating structured outputs, RAG frequently reduces hallucinations and improves factual accuracy, according to prior studies [2], [11], [16], [17].

RAG is helpful, but it is not a comprehensive answer. Hallucinations may still occur when the model ignores or misinterprets the evidence, when the retrieval system returns missing or irrelevant information, or when the prompt fails to help the model make effective use of the retrieved content [3], [8]. RAG increases factual correctness and strengthens models against noise, according to recent standards, but there are still a number of issues. These include dealing with contradictory or incomplete knowledge, correctly merging retrieved facts, and rejecting incorrect information [2], [3]. Any RAG system's performance is also heavily influenced by how well the retrieval step works and how well the content that is retrieved fits the user's query [3], [18], [19]. Consequently, adaptive query approaches have been proposed to optimize the extraction of specialized information, particularly in medical image applications where precise retrieval is essential for downstream detection tasks [20].

To further reduce hallucinations in RAG systems, recent research has investigated post-generation correction procedures, enhanced retrieval designs, and new fine-tuning techniques [3], [11]. Researchers can assess hallucination more precisely at the word and passage levels using benchmark data sets like RAGTruth [21], which helps develop better detection and mitigation strategies [2]. Furthermore, domain adaptation and retrieval techniques informed by domain knowledge have demonstrated positive outcomes in fields where high accuracy is crucial, such as customer service and medical applications [7], [16].

Because RAG grounds outputs in external knowledge, it offers a substantial advancement in reducing hallucinations

in LLMs [3], [8]. However, retrieval quality, integration processes, and persistent evaluation and domain adaptation issues limit its efficacy. To create more dependable, contextually aware, and explicable RAG systems for practical implementation, additional research is required [7], [11].

Although previous works demonstrate that retrieval-augmented generation enhances factual grounding and reduces hallucinations, the majority of current methodologies mainly depend on unstructured textual retrieval and dense similarity search.

Although previous works demonstrate that retrieval-augmented generation enhances factual grounding and reduces hallucinations, the majority of current methods mainly depend on unstructured text retrieval and dense similarity search. While these methods are showing good results in answering fact retrieval questions, it is commonly shown that unstructured text retrieval is not good enough in answering multi-hop questions [22], where clear relational reasoning is required to retrieve the verified information to answer the question. For example, asking a question like *what is the capital of the country where Albert Einstein was born?* requires the LLM to follow the path: Albert Einstein → Place of Birth → Country → Capital. This kind of information retrieval is hard to be implemented using a simple vector similarity search. The purpose of this study is to examine the effect of using a structured data source like Wikidata on multi-hop question-answering accuracy.

III. Methodology

This study evaluates the effect of Wikidata-based RAG on reducing hallucinations in LLMs. To achieve this, an agentic pipeline has been developed that grounds model responses in Wikidata and Wikipedia before answer synthesis. $qwen2.5:32b-instruct-q4\_K\_M$ was used in the study. We selected this model because it shows strong performance in following detailed instructions [23], which is essential for RAG systems that depend on precise tool calls.

The model uses ReAct prompting to guide its behaviour. With ReAct prompting, the model alternates between reasoning steps and actions [24]. By combining structured reasoning, careful tool selection, and grounded evidence, the system aims to limit unsupported inferences and reduce hallucinations [3], [17].

A. Agent Tools

The Wikidata-RAG model uses different tools to fetch structured facts about entities found in the user question. All tools presented in this section are custom-built for this experiment. The code can be found at https://github.com/razisayyed/wikidata-rag.

1) $search\_entity\_candidates$: This is the first tool that the model uses when finding any entity in the question. It is a mandatory tool that must be called for each entity. This tool uses **SPARQL** to return basic information from **Wikidata**. It takes a $entity\_name$ and an optional $entity\_type$ as parameters. The $entity\_type$ is used to distinguish between similar entities. A basic **SPARQL** query is executed that returns `QID, Label, and Description` for a list of candidates. The LLM decides if any candidate is relevant to the context of the question. The SPARQL query used in this tool is shown in Listing 1.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX mwapi: <https://www.mediawiki.org/ontology#API/>

SELECT ?item ?itemLabel ?itemDescription ?
    instanceOfLabel WHERE {
  SERVICE wikibase:mwapi {
    bd:serviceParam wikibase:api "EntitySearch" ;
                    wikibase:endpoint "www.wikidata.org
    " ;
                    mwapi:search "..." ;
                    mwapi:language "en" .
    ?item wikibase:apiOutputItem mwapi:item .
  }
  OPTIONAL { ?item wdt:P31 ?instanceOf . }
  SERVICE wikibase:label { bd:serviceParam wikibase:
    language "en". }
}
LIMIT ...
```

Listing 1. SPARQL template used by $search\_entity\_candidates$.

2) $fetch\_entity\_properties$: this tool is mandatory; it must be executed by the LLM after picking a relevant candidate. It takes the QID of the candidate and a list of **Wikidata** properties as parameters. The tool fetches the properties and returns them to the LLM. The SPARQL query used in this tool is shown in Listing 2.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX schema: <http://schema.org/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>

SELECT ?itemLabel ?itemDescription ?wikipediaUrl ?value
     ?valueLabel WHERE {
  BIND(wd:Q... AS ?item)

  OPTIONAL {
    ?item p:P... ?statement .
    ?statement ps:P... ?value .
  }

  OPTIONAL {
    ?wikipediaUrl schema:about ?item ;
                  schema:isPartOf <https://en.wikipedia
    .org/> .
  }

  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en".
  }
}
LIMIT 500
```

Listing 2. SPARQL template used by $fetch\_entity\_properties$ (no qualifiers).

3) $wikidata\_sparql$: This tool is also optionally executed in cases where limited information is returned from the $fetch\_entity\_properties$ tool. It takes a **SPARQL** query written by the LLM, executes it, and returns the results to the model. This tool is considered a last-resort way to let the LLM decide how the data will be retrieved from **Wikidata**.
4) $fetch\_wikipedia\_article$ This tool is optionally exe-

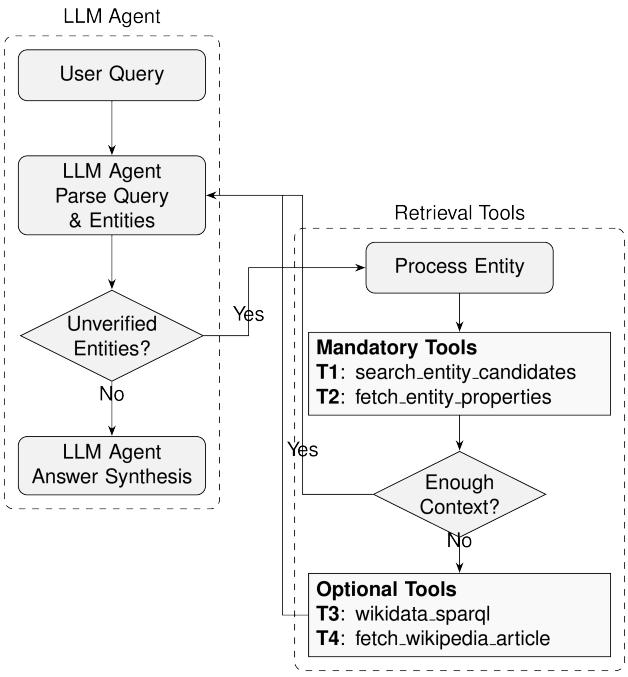

Fig. 1. Workflow of the Wikidata-based RAG system showing LLM-controlled entity verification and tool-based context retrieval with numbered tools.

cuted when there is insufficient information from all other tools. It returns the first 8000 words found on the Wikipedia page of the entity, if it exists. Wikipedia page URL is returned using a **SPARQL** query.

B. Detailed Retrieval Procedure

Figure 1 illustrates the overall retrieval and grounding workflow. The process begins with a user query and proceeds through entity identification, retrieval of Wikidata properties, extraction of a relevant Wikipedia article, executing custom `SPARQL` queries, and final answer synthesis.

1) Step 1: Query Parsing and Entity Identification: Upon receiving a user question, the main LLM identifies all named entities and determines the types of information required. For example, in the query *"What are the major scientific contributions of Niels Bohr and when was he born?"*, the model recognises the entity *Niels Bohr* and infers that the required properties include notable works (P800), fields of work (P101), and date of birth (P569).

The agent invokes the lookup tool with an entity name and an optional semantic type. The tool performs a **SPARQL Search** – Query a limited amount of **Wikidata** items for exact or close label matches.

The query returns **QID**, **Label**, and **Description** for each result found. The LLM then tries to pick a relevant candidate.

If a relevant candidate is found, then the LLM will execute $fetch\_entity\_properties$ with the relevant QID and the already inferred properties. $fetch\_entity\_properties$ will return all the properties found on **Wikidata** with their corresponding values.

In case of the absence of enough context from the gathered data, the LLM can optionally use $wididata\_sparql$, a tool to execute a query that is entirely constructed by the LLM itself. This way, we give the LLM more freedom in extracting facts that help answer the question.

After gathering structured data, the LLM decides if the gathered data is enough to answer the question. In case of the need for more context, then the LLM will execute $fetch\_wikipedia\_article$ tool. This tool will first use **SPARQL** to fetch the URL of the entity's Wikipedia page. If the SPARQL found a result, this tool will fetch the HTML content of the found page, clean it, and return the final result to the LLM.

Wikipedia retrieval is used only as supplementary descriptive context after structured facts have been retrieved, and never as a replacement for Wikidata properties.

This retrieval procedure will be repeated for each entity found in the query. Also, tools can be executed multiple times for each entity if needed. Once the LLM has gathered all materials. It will synthesise these materials into a final response. Crucially, no part of the answer is produced without explicit grounding in retrieved content.

C. Traceability and Reproducibility

The system keeps a record of every tool call, including the list of entity candidates, the chosen QIDs, the SPARQL outputs, and the generated summaries. This level of tracking makes it possible to link each final answer to the exact structured facts and retrieved text that were used to build it.

In addition to that, *langsmith* (https://smith.langchain.com/) service has been used to manually investigate tools' invocation and the returned context by each tool. Langsmith was a crucial part of the experiments conducted.

D. Baseline Model

The experiment was conducted by comparing the Wikidata-RAG model with a baseline Prompt-Only model. The same LLM and settings are used to evaluate both models to make sure the comparison was fair and controlled. The main difference between the two models was whether the retrieved structured data from Wikidata was injected into the prompt or not. No additional prompt tuning or adjustments have been done to either model.

E. Evaluation

This study employs four automated hallucination evaluation methods: Vectara's HHEM score-based hallucination evaluation model, the AimonLabs hallucination-detection model, LLM-as-a-Judge [25] rubric, and RAGTruth style retrieved-context faithfulness evaluator. Among these, the Vectara evaluator is designated as the primary evaluation metric.

Vectara's model is selected as the primary evaluator because it provides a continuous factuality score based on direct comparison between the generated response and a

reference context, rather than relying on binary or subjective judgments.

AimonLabs and LLM-as-a-Judge evaluators are included to capture alternative perspectives on hallucination detection. In other words, they are used to check whether the conclusions remain consistent when different evaluation methods are applied.

$vectara/hallucination\_evaluation\_model$ compares generated answers against a ground-truth reference, returning a factuality score in the range [0, 1], where lower values suggest a greater risk of hallucination [26]. However, the model has notable shortcomings: it prioritises lexical and semantic alignment over deeper reasoning, and rather than providing interpretable feedback, it only gives a scalar score—offering little insight into the nature of any detected hallucinations.

$AimonLabs/hallucination{-}detection{-}model$ is used as a secondary semantic validator [27]. This model performs a deeper comparison between the generated answer and the ground-truth context and outputs a binary hallucination judgement. It is particularly useful for identifying cases where correct answers are phrased differently from the reference text, which may lead to false negatives under lexical comparison.

The third method of evaluating is the LLM-as-a-Judge approach. A big language model is asked to compare the answer provided with the ground-truth reference and see if the answer gives facts that are not supported or are wrong. This method seems close to human judgment and gives a reasoning-based view that is easy to follow and understand, especially in unclear cases.

RAGTruth-style evaluator is used to measure source-grounding directly, not just answer correctness. In other words, an answer can be factually correct relative to ground truth while still being weakly grounded in the retrieved evidence. This provides a complementary signal by checking whether each response is explicitly supported by the retrieved context/tool output.

1) Ground Truth Construction: All the evaluation methods use the same ground truth references, which were constructed manually due to the absence of well-established and widely accepted datasets specifically designed for factual hallucination assessment in open-domain question answering [1]. For each test question, a ground truth reference was compiled primarily from Wikipedia and supplemented, when necessary, by a limited number of additional high-reputation web sources.

The ground truth was designed to include only factual and verifiable statements that directly address the scope of each question. Semantically equivalent formulations of the same fact were considered valid; however, exhaustive enumeration of all possible correct facts related to an entity was not feasible. As a result, the ground truth represents a conservative factual reference rather than a complete knowledge representation.

IV. Experimental Tests

This section presents the experimental results comparing the Prompt-Only baseline with the proposed Wikidata-based RAG pipeline.

A. Overall Performance

Across the evaluated test cases, the Wikidata-based RAG system consistently produces more factually grounded responses than the prompt-only baseline. Using the Vectara evaluator, the RAG system achieves higher factuality scores in most cases. The LLM-as-a-Judge evaluator further confirms this trend by explicitly identifying hallucinated spans in prompt-only outputs, while RAG responses are generally labelled as factual.

Table I summarises the winner distribution across evaluators. The Wikidata-based RAG system is preferred more frequently overall, especially by Vectara and the LLM-as-a-Judge, while AimonLabs assigns an equal number of wins to both systems, reflecting evaluator-dependent variation.

TABLE I
Overall Winner Distribution Across Different Evaluators

Evaluator	RAG	Prompt	Tie
Vectara	53	24	3
AimonLabs	46	10	24
LLM-as-a-Judge	37	4	39

The main reason why Vectara results vary from AimonLabs and LLM-as-a-Judge is the fact that Vectara uses continuous score comparison, while other evaluators use label-based comparison (factual vs hallucination). If we compare Vectara results the same way, its results will become **34 RAG**, **3 Prompt**, and **43 Ties**, which complies with other evaluators' results.

Table II shows the number of factual and hallucinated responses assigned by each evaluator. The results show that the Wikidata-based RAG system outperforms the prompt-only baseline model according to all the comparison evaluators. It also shows a relative agreement across the different evaluators, with the exception of AimonLabs. This is due to the fact that AimonLabs produces stricter factuality classifications because it penalizes locally unsupported phrasing even when the overall answer is largely correct.

TABLE II
Factual vs. Hallucinated Responses by Evaluator

Evaluator	RAG		Prompt-Only	
	Factual	Halluc.	Factual	Halluc.
Vectara	62	18	31	49
AimonLabs	48	32	26	54
LLM-as-a-Judge	60	20	31	49

Overall, the results suggest that integrating Wikidata improves factual reliability relative to Prompt-Only, but the measured hallucination rate depends on the evaluation method.

B. Retrieved-Context Faithfulness

A retrieved-context faithfulness metric was used to check whether the RAG answer is supported by the retrieved tool outputs (not just correct against ground truth). This metric is RAG-only since the baseline has no retrieved context. On the 80-question benchmark, the evaluator completed 79 cases (1 skipped due to the fact that RAG decided to ignore the tools and reject answering the question) and marked 54/79 responses as faithful (68.35%) and 25/79 as unfaithful (31.65%).

Among the 25 unfaithful RAGTruth-style cases, the other evaluators also marked RAG as hallucinated, as summarized in Table III.

TABLE III
HALLUCINATION DETECTION AGREEMENT ACROSS EVALUATORS (25 UNFAITHFUL RAGTRUTH-STYLE CASES)

Evaluator / Condition	Marked as Hallucinated
Vectara HHEM	14/25 (56.0%)
AimonLab	20/25 (80.0%)
LLM-as-a-Judge	17/25 (68.0%)
At least one of the three evaluators	20/25 (80.0%)
All three evaluators	14/25 (56.0%)
None of the three evaluators	5/25 (20.0%)

These results show that the RAG system often gives correct answers, but some answers are still not fully supported by the retrieved evidence trace.

V. RESULTS DISCUSSION

This section interprets the experimental results, focusing on evaluator behavior, grounding effectiveness, and observed failure modes.

A. Evaluator Hierarchy and Metric Sensitivity

Evaluator disagreement must be interpreted in light of the predefined evaluation hierarchy. While Vectara serves as the primary evaluator, AimonLabs and LLM-as-a-Judge apply different operational definitions of hallucination, producing partially divergent judgments. These variations illustrate how hallucination measurement is sensitive to the choice of evaluator, but do not invalidate the primary findings.

B. Effectiveness of Wikidata Grounding

Grounding with Wikidata consistently reduces hallucinations across evaluators. However, the results do not support the claim that score-based evaluators always reward grounded answers; the main observable difference is that the prompt-only baseline more frequently produces hallucinated responses.

C. Qualitative Error Analysis

Inspection of individual test cases reveals three recurring failure patterns:

Long multi-hop chain errors: Questions requiring several connected reasoning steps are the most common source of hallucination. The LLM may correctly follow part of a reasoning chain but miss one critical link, producing an answer that appears plausible but is unsupported by the full evidence chain.

Role ambiguity with multiple valid entities: When a role can refer to more than one valid person (e.g., multiple founders of an organisation), the model may return a correct but unexpected answer. Such responses may be marked as hallucinated because they do not match the ground truth exactly, even when the reasoning path is valid.

Temporal mismatch in ground-truth normalization: Questions involving historical figures may yield historically correct answers that conflict with ground truth normalized to modern country names or current capitals, making reasonable historical answers appear incorrect.

D. Implications for Hallucination Evaluation

These results confirm that Wikidata-grounded RAG substantially improves factual accuracy, but does not fully eliminate hallucinations, and the retrieval step can introduce new error types—particularly when entity disambiguation relies on incomplete temporal or relational constraints. Using multiple evaluators, clear ground truth, and detailed error analysis remains essential when studying hallucinations in RAG systems.

VI. LIMITATIONS

The main limitation of this study is the fact that the evaluation dataset is relatively small. The experimental results are obtained from 80 AI-generated multihop questions that are manually reviewed and corrected, which is inadequate for robust statistical generalisation. Thus, the results should be considered exploratory rather than definitive. To reach statistically valid conclusions, we need a bigger and more varied benchmark.

A second limitation relates to the dependence on model-based hallucination evaluators. The three comparison evaluators in this study use different definitions of hallucination and don't always agree on judgment. Therefore, hallucination rates are highly dependent upon the evaluator, and no particular metric can be considered an absolute source of truth.

The difficulty of constructing ground-truth references adds extra limitations. Some facts can be missed during the construction of ground truth. As a result, evaluators may incorrectly mark correct answers as hallucinations based on the absence of valid facts from the ground-truth context.

In addition, the proposed approach is constrained by the coverage and expressiveness of Wikidata. Some facts are missing, difficult to model, or require complex SPARQL queries that are not always generated or executed successfully.

Finally, the reported results are sensitive to implementation details such as prompt design, tool invocation order, and decoding parameters. While these factors were fixed to ensure internal consistency, the conclusions may not directly generalize to other RAG configurations or LLM backends.

VII. CONCLUSION

This paper studied how Wikidata-based Retrieval-Augmented Generation (RAG) affects hallucination in LLM

outputs by injecting verified Wikidata facts into the answer-generation process. Under the Vectara hallucination evaluator, the proposed Wikidata-RAG system achieved a 77.5% factuality rate (62/80) versus 38.75% (31/80) for the prompt-only baseline model. The LLM-as-a-Judge evaluator scores comparable scores: 75% (60/80) versus 38.75% (31/80) factuality rate. The AimonLab model flagged more hallucinations for both models: 60% (48/80) versus 32.5% factuality rate. Overall, the results support Wikidata grounding as a practical step toward improved factual reliability, but any hallucination-rate claim must be reported together with the evaluator used.

Although limitations remain—especially in entity ambiguity, partial coverage in Wikidata, and evaluator-dependent definitions of hallucination—these findings suggest that combining structured retrieval with controlled prompting is a promising direction for building more trustworthy AI systems.

Future work should investigate scalable benchmarks, improved temporal entity disambiguation, and hybrid grounding strategies that combine structured knowledge graphs with curated document retrieval.

REFERENCES

[1] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3703155

[2] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented generation," pp. 17 754–17 762, 2023.

[3] W. Zhang and J. Zhang, "Hallucination mitigation for retrieval-augmented large language models: A review," *Mathematics*, 2025.

[4] M. Mahaut, L. Aina, P. Czarnowska, M. Hardalov, T. Müller, and L. Màrquez, "Factual confidence of llms: on reliability and robustness of current estimators," *arXiv preprint arXiv:2406.13415*, 2024.

[5] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, 2023.

[6] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," *International Conference on Machine Learning (ICML)*, 2023.

[7] S. Gupta, "Retrieval-augmented generation and hallucination in large language models: A scholarly overview," *Scholars Journal of Engineering and Technology*, 2025.

[8] P. B'echard and O. M. Ayala, "Reducing hallucination in structured outputs via retrieval-augmented generation," pp. 228–238, 2024.

[9] X. Zhu, Y. Xie, Y. Liu, Y. Li, and W. Hu, "Knowledge graph-guided retrieval augmented generation," in *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computational Linguistics, 2025, pp. 8912–8924.

[10] K. Xu, K. Zhang, J. Li, W. Huang, and Y. Wang, "Crp-rag: A retrieval-augmented generation framework for supporting complex logical reasoning and knowledge planning," *Electronics*, vol. 14, no. 1, p. 47, 2025.

[11] S. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, "A comprehensive survey of hallucination mitigation techniques in large language models," *ArXiv*, vol. abs/2401.01313, 2024.

[12] M. U. Nasir, M. T. Naseem, T. M. Ghazal, M. Zubair, O. Ali, S. Abbas, M. Ahmad, and K. M. Adnan, "A comprehensive case study of deep learning on the detection of alpha thalassemia and beta thalassemia using public and private datasets," *Scientific Reports*, vol. 15, no. 1, Apr. 2025. [Online]. Available: http://dx.doi.org/10.1038/s41598-025-97353-0

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Technical Report*, 2019.

[15] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[16] J. Miao, C. Thongprayoon, S. Suppadungsuk, O. A. G. Valencia, and W. Cheungpasitporn, "Integrating retrieval-augmented generation with large language models in nephrology: Advancing practical applications," *Medicina*, vol. 60, 2024.

[17] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *CoRR*, vol. abs/2005.11401, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[18] X. Qu, Q. Chen, W. Wei, J. Sun, and J. Dong, "Alleviating hallucination in large vision-language models with active retrieval augmentation," *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.

[19] V. Karpukhin, B. Oguz, S. Min, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense passage retrieval for open-domain question answering," *CoRR*, vol. abs/2004.04906, 2020. [Online]. Available: https://arxiv.org/abs/2004.04906

[20] A. Migdady, Y. Khamayseh, O. AlZoubi, and M. Bani Yassein, "An adaptive query approach for extracting medical images for disease detection applications," *Arabian Journal for Science and Engineering*, vol. 50, 05 2024.

[21] Y. Wu, J. Zhu, S. Xu, K. Shum, C. Niu, R. Zhong, J. Song, and T. Zhang, "Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models," 2023.

[22] Y. Tang and Y. Yang, "Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries," *ArXiv*, vol. abs/2401.15391, 2024.

[23] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

[24] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," 2023. [Online]. Available: https://arxiv.org/abs/2210.03629

[25] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023. [Online]. Available: https://arxiv.org/abs/2306.05685

[26] M. Li, R. Luo, and O. Mendelevitch, "HHEM-2.1-Open," 2024. [Online]. Available: https://huggingface.co/vectara/hallucination_evaluation_model

[27] B. Paudel, A. Lyzhov, P. Joshi, and P. Anand, "Hallucinot: Hallucination detection through context and common knowledge verification," 2025. [Online]. Available: https://arxiv.org/abs/2504.07069