Mitigating Large Language Model Hallucinations in Domain-Specific Applications: An Empirical Study of SQL Retrieval-Augmented Generation in WNBA Analytics

August 2024, UC Berkeley W261
Akaash Venkat, Anoop Nair, I-Hsiu Kao, Iishaan Shekhar

I. Abstract

This paper investigates the effectiveness of SQL Retrieval Augmented Generation (RAG) in mitigating hallucination in Large Language Models (LLMs) within specific, structured domains, using the "Hoops IQ" WNBA intelligence tool as a case study.[1] The report details the system's architecture, data management, and evaluation methodology, demonstrating how grounding LLMs in structured databases significantly enhances factual accuracy and reduces the propensity for generating erroneous information. The findings reveal substantial improvements in response accuracy and a marked reduction in hallucinations across various LLMs compared to their baselines, particularly for statistical queries.[1] On a 38-question sports-statistics benchmark, SQL RAG lifts answer accuracy from 53% to 76%, a gain of 23 percentage points ($\Delta = 0.47$, 95 % CI 0.22 - 0.73; McNemar $\chi^2(1)=12.0$, $p = 0.0005$). The paper also discusses the nuanced interplay between the LLM's domain understanding and the RAG mechanism, identifies current limitations in handling complex analytical queries, and outlines future research directions for robust, trustworthy domain-specific LLM applications.

II. Introduction

Large Language Models have demonstrated revolutionary capabilities across various natural language processing tasks, from summarization to question answering. However, a critical challenge persists: their tendency to "hallucinate," producing factually incorrect, nonsensical, or ungrounded information.[4] This issue becomes particularly pronounced and impactful in domain-specific applications, such as sports analytics, legal research, or medical diagnostics, where factual accuracy and reliability are paramount. General-purpose LLMs, trained on vast and diverse datasets, often lack the requisite in-depth, specialized knowledge for these fields, frequently leading to "surface-level outputs" or outright factual errors.

The widespread occurrence of hallucinations in general-purpose Large Language Models, particularly within specialized domains, points to a fundamental constraint associated with relying exclusively on the models' parametric knowledge. When confronted with queries in niche areas, where their vast training data may offer only superficial coverage, these models frequently generate information that, while linguistically plausible, is factually incorrect. This behavior is not merely a random error; it stems from the models' inherent tendency to prioritize fluent and coherent text generation, even when their internal knowledge is incomplete or ambiguous.[6] In such instances, the model may fabricate details to maintain conversational flow, rather than admitting a lack of information. For professional applications, such as those in sports analytics, legal, or medical fields, where decisions carry significant weight, this inherent unreliability poses a substantial impediment to their widespread adoption. Consequently, effective solutions must not only improve the factual correctness of responses but also establish a verifiable basis for the information provided, thereby preventing misinformed actions and fostering user confidence.

Retrieval-Augmented Generation (RAG) has emerged as a leading paradigm to address LLM limitations by integrating an information retrieval process into the generation pipeline.[6] RAG enhances LLM outputs by retrieving relevant, up-to-date information from external knowledge bases, thereby providing "grounded context" that significantly improves accuracy, robustness, and factual consistency.[7] This approach allows LLMs to access current and proprietary data without requiring costly and frequent retraining of the entire model.[7] RAG's ability to provide real-time, external context directly mitigates hallucinations by shifting the LLM's reliance from its potentially outdated or incomplete internal parametric knowledge to verifiable external sources. This establishes a direct causal link between the implementation of RAG and the reduction of hallucinated content. The process involves retrieving external data, which is explicitly linked to reducing hallucinations.[7] This effectively serves as an external, dynamic memory for LLMs. Hallucinations often arise when an LLM encounters a knowledge gap or ambiguity and, in its attempt to generate a plausible response, "fills in" information based on its learned patterns or biases.[6] RAG intervenes by supplying concrete, factual information from a trusted external source before the generation phase. This pre-emptive provision of context reduces the "gap" that the LLM would otherwise be compelled to fill with fabricated content, thereby grounding its response in verifiable facts. This makes RAG a powerful, adaptable, and often more economical approach compared to continuous fine-tuning for maintaining knowledge currency and factual integrity, particularly in dynamic domains where information evolves rapidly.[11]

The "Hoops IQ" project serves as a compelling case study, demonstrating a practical application of SQL RAG within the domain of WNBA sports analytics.[1] The project was conceived to address the dual challenge of making WNBA data accessible to non-technical users and counteracting the "consistently false information" frequently reported by general-purpose LLMs regarding sports statistics.[1] The primary target user group, daily fantasy players, critically relies on highly accurate and reliable statistical Q&A to make informed decisions and optimize their gameplay strategies.[1]

This paper contributes empirical evidence on the efficacy of SQL RAG in a real-world and highly specialized domain. It provides a detailed analysis of the system's architecture, data management, and evaluation methodology. Furthermore, it offers observations on the strengths and current limitations of this approach, outlining future research directions for developing more robust and trustworthy domain-specific LLM applications. The refined research question guiding this study is: "Can a Retrieval-Augmented Generation system, specifically leveraging structured SQL databases, effectively reduce factual hallucinations in Large Language Models when applied to domain-specific, quantitative query tasks, as empirically demonstrated by the Hoops IQ WNBA intelligence tool?"

III. Background and Related Work

Understanding LLM Hallucinations: Definitions, Causes, and Impact

Hallucination in LLMs is broadly defined as the generation of text that is "incorrect, makes no sense, or is unrelated to reality" [5], or more specifically, the "creation of factually erroneous information that appears factual but is ungrounded".[6] These fabrications manifest in various forms, including factual inaccuracies, nonsensical responses, and contradictions, where the model's output conflicts with its given input or previously generated content.[5]

The primary causes of hallucination are multifaceted. Firstly, training data issues play a significant role. These include inadequate representation of specific topics, the presence of biases, or outright misinformation and noise within the vast datasets used for training LLMs.[5] LLMs can extrapolate information from these biases or misinterpret ambiguous prompts, leading to the generation of ungrounded content.[6]

Secondly, model limitations contribute to the problem. LLMs are constrained by a maximum context window, meaning they can only consider a certain number of tokens simultaneously. This limitation can lead to misunderstandings or omissions of crucial information, particularly in longer conversations or documents.[5] Furthermore, LLMs often struggle with interpreting the subtleties and nuances of human language, such as irony or cultural references [5], and exhibit "weak reasoning" when faced with complex prompts or mathematical problems.[7]

Thirdly, the generative tendencies inherent in LLMs contribute to hallucinations. These models are designed to produce plausible and fluent text, which can lead them to "fill the gaps" with fabricated details when their internal knowledge is insufficient or uncertain.[7]

Lastly, the lack of real-time updates is a significant factor. The static nature of LLM training data means that outputs can quickly become outdated, especially in rapidly evolving fields like science or politics, leading to vague, obsolete, or false responses.[7]

The impact of hallucinations is substantial. They significantly degrade system performance and fail to meet user expectations.[3] More critically, they erode user trust and can deter professionals from adopting LLM solutions, especially given the potential for critical errors in sensitive applications. In fields like medicine or law, such inaccuracies can raise serious ethical concerns and have profound legal implications.[7] The pervasive nature of hallucinations, rooted in how LLMs are trained and operate, indicates that these are not isolated anomalies but systemic issues. The core function of an LLM is to predict the most probable next token based on its training data. When this probabilistic prediction encounters a gap in its knowledge or an ambiguous query, the model's "fluency engine" can prioritize generating a linguistically plausible output over a factually accurate one. This is further compounded by the scale and potential biases or noise in their vast training datasets. Therefore, simply instructing an LLM not to hallucinate is insufficient; its fundamental generative mechanism requires external grounding or internal constraints.

General Hallucination Mitigation Strategies

Comprehensive surveys categorize hallucination mitigation techniques into two main areas: data-related methods and modeling/inference methods.[3]

Data-related methods primarily focus on building faithful datasets, automatic data cleaning, and information augmentation.[4] These strategies aim to improve the quality and relevance of the data that LLMs are exposed to.

Modeling and inference methods encompass a broader range of techniques, including architectural improvements, specialized training techniques (e.g., Reinforcement Learning with Human Feedback, multi-task learning), advanced decoding methods, model editing, Chain-of-Thought (CoT) prompting, ensemble methods, and post-processing steps.[3]

Prompt engineering is a crucial aspect of mitigation, involving experimentation with various instructions to guide the LLM towards desired outputs.[6] Techniques such as few-shot prompting, which provides illustrative examples, and setting limits on input/output length, can encourage conciseness and reduce opportunities for hallucination.[5]

Furthermore, model parameter adjustment allows for fine-tuning the LLM's generative behavior. Modifying parameters like temperature (to control predictability), frequency penalty (to reduce repetition), and presence penalty (to encourage diversity) can influence the model's output and potentially reduce the incidence of hallucination.[5]

The evolution of hallucination mitigation strategies reflects a growing understanding that this complex problem requires interventions at multiple stages of the LLM lifecycle, from initial data preparation and model training to inference and post-processing. The wide array of techniques proposed across various research papers indicates that no single, universally effective solution exists for hallucination.[3] The breadth of strategies—spanning data preprocessing, model architecture, training methodologies, and inference-time techniques—indicates that hallucinations are a multi-causal phenomenon. This necessitates a layered defense approach where different methods complement each other. A significant trend is the shift from purely model-centric solutions (e.g., fine-tuning) to more data-centric (e.g., RAG) and process-centric (e.g., prompt engineering, monitoring) interventions, acknowledging the critical role of external knowledge and user interaction. Consequently, for practical and robust deployments of LLMs in real-world applications, a holistic strategy that combines RAG with other techniques—such as meticulous data curation, sophisticated prompt engineering, and continuous monitoring and feedback loops—is likely to yield the most reliable and accurate results.

Retrieval-Augmented Generation (RAG) Architectures and Benefits

RAG fundamentally enhances LLMs by introducing an information retrieval process that accesses external data stores, leading to "greater accuracy and robustness" in generated content.[9] This approach allows LLMs to leverage vast, dynamic repositories of external knowledge, merging it with their intrinsic capabilities.[13]

Key benefits of RAG include:

- **Reduced Hallucinations and Staleness:** By providing access to up-to-date and comprehensive information from a dynamic knowledge base, RAG significantly lowers the incidence of fabricated or outdated responses.[7] This also reduces the need for costly and frequent model retraining, as the external knowledge base can be continuously updated.[7]
- **Contextual Consistency and Improved Accuracy:** Responses are directly grounded in data retrieved from curated, reliable sources, ensuring contextual relevance and factual correctness.[11] Empirical evidence supports this claim; for instance, a study demonstrated RAG improving output accuracy to 91.4% in a preoperative medicine domain, compared to 80.1% without RAG.[17]
- **Controllability and Privacy:** RAG allows developers to precisely control the scope of generated responses by limiting retrieval to specific, trusted sources.[11] Furthermore, private or proprietary data is accessed as context at inference time rather than being absorbed into the LLM's training data, enhancing data privacy and security.[11]
- **Reduced Generation Bias:** By providing a specific, curated context, RAG can mitigate some of the generation biases present in the LLM's original training set, as the retrieved information can supersede these biases.[11]
- **Cost-effectiveness:** RAG can significantly reduce the computational costs associated with generating responses and the prohibitive expense of frequent, full-scale model retraining.[9]

The typical RAG process involves an input query, which is then used by a retriever to locate relevant data sources. This retrieved information subsequently interacts with the generator to enhance the overall generation process.[9] This fundamental transformation turns LLMs from "closed-book" knowledge systems, limited by their static training data, into "open-book" systems capable of accessing and integrating external, dynamic knowledge. This directly addresses the "knowledge boundary" problem identified in hallucination research.[3] The concept of an LLM's "knowledge boundary" refers to the inherent limitations of its parametric memory, which is fixed at the time of training. RAG directly expands this boundary by providing access to external knowledge bases that are easily modifiable and continuously updatable.[9] This represents a significant paradigm shift, enabling LLMs to stay current with rapidly changing information and access highly specialized or proprietary data without the need for constant, resource-intensive retraining. Beyond merely boosting accuracy, RAG is a crucial enabler for deploying LLMs in domains where knowledge is dynamic, vast, or confidential. This capability makes LLMs practical and reliable for a wide array of enterprise and professional use cases that were previously unfeasible.

Comparison of Structured (SQL) vs. Unstructured (Vector Store) Data in RAG

The choice of retrieval source in RAG architectures, particularly between structured SQL databases and unstructured vector stores, depends heavily on the nature of the queries and the underlying data structure.

Vector Databases are purpose-built to store and manage data in vector format, enabling fast similarity searches through algorithms like K-Nearest Neighbors (KNN) or Approximate Nearest Neighbor (ANN). In RAG, they are commonly used to retrieve semantically similar text chunks from unstructured data.[11] A primary weakness of vector databases is that context, particularly relational context between data points, can be lost or obscured during the embedding process. This means that while they can find semantically similar information, they may struggle with complex synthesis tasks and do not inherently preserve the exact context needed for precise answers.

SQL Databases (Relational Databases), conversely, are traditional databases that store data in structured rows and columns, enforcing data accuracy, integrity, and relationships through a predefined schema.[15] They are known for their ACID (Atomicity, Consistency, Isolation, Durability) compliance, which guarantees transaction reliability and data accuracy by preventing duplicate information through primary and foreign keys.[16] For RAG, SQL databases offer direct access to precise, factual, and highly structured data.[1] Their inherent structure ensures data accuracy and consistency.[16] The long-standing maturity of SQL databases means a wide array of tools and resources are available, and the English-like syntax of SQL makes it accessible for querying.[16]

However, when integrating SQL databases with RAG via Text-to-SQL mechanisms, certain limitations arise. While powerful for querying structured data, the challenge lies in the LLM's ability to generate valid and semantically correct SQL queries from natural language.[17] A critical limitation highlighted in the literature is that RAG, when applied to SQL, is often "limited to filtering relevant rows and does not understand other data transformations like joins or aggregate that could provide valuable context".[18] This implies a significant struggle with complex analytical queries that go beyond simple data retrieval. There are also scalability issues when embedding physical data from very large SQL databases for vector search [18], and privacy concerns arise when sending sensitive data to third-party LLMs as context.[18]

The optimal choice of retrieval source (SQL vs. Vector DB) is highly dependent on the nature of the queries and the underlying data structure. While vector stores offer versatility for unstructured text and semantic search, SQL databases provide superior factual precision for structured data. However, the LLM's capability to reliably generate complex SQL queries remains a significant bottleneck for the latter. The "SQL DB vs Vector Store" comparison in the Hoops IQ presentation [1] vividly illustrates a key strength of SQL RAG: for exact factual recall (e.g., "who made an 18-foot step back jumpshot in 2018?"), SQL is inherently superior because it retrieves precise, structured records (e.g., including specific assists). In contrast, vector stores retrieve semantically similar text chunks, which may be related but often lack the factual exactness or completeness of structured data (e.g., omitting the assisting player). However, this precision comes at the cost of reliably translating complex natural language queries into sophisticated SQL, especially for operations like joins, subqueries, or aggregates.[18] This highlights a fundamental trade-off between the precision of structured data retrieval and the complexity of natural language to SQL translation for advanced analytical tasks. For applications demanding high factual exactness and the retrieval of specific, related data points (e.g., financial reporting, medical records, detailed sports statistics), SQL RAG offers a distinct advantage over pure vector-based RAG, provided the Text-to-SQL translation layer is robust for the expected query complexity. For broader, more conceptual queries or unstructured data, vector stores remain highly valuable. This suggests that a hybrid approach might be optimal for comprehensive domain coverage.

Challenges of LLMs in Sports Analytics and Domain-Specific Knowledge

Existing research indicates that while LLMs demonstrate competent performance in basic sports knowledge, they consistently struggle with more "complex, scenario-based sports reasoning".[2] LLMs often rely on "superficial sports associations" rather than developing a deep understanding of contextual nuances and intricate rules.[2] This suggests that the domain of sports understanding has not received sufficient attention in general LLM training, leading to a performance ceiling that has not significantly improved in basic tasks compared to previous benchmarks.

Specific challenges identified include difficulties with multi-hop questions, where prompt length and the sequence order of sub-questions significantly impact reasoning ability, often leading to "unstable reasoning" even when individual sub-questions are answered correctly. For instance, LLMs show higher accuracy in answering main questions compared to sub-questions, even when sub-questions require reasoning within the context provided by the main question. This highlights a critical need for LLMs to effectively manage longer prompts and complex contexts, especially in sports where nuanced relationships between context and query are crucial.

Error types observed in sports reasoning tasks include:

- **Lack of Domain Knowledge:** This involves misunderstandings of rules, concepts, or terms, and failing to understand specific tactics. For example, LLMs might discourage net play in tennis due to perceived high risks, failing to recognize it as an aggressive and effective professional strategy.
- **Inaccurate Recall:** These errors occur when models incorrectly remember facts or details, such as misremembering the number of players or substitutes in a professional basketball team.
- **Context and Nuances Confused:** This type involves misinterpretations or oversimplifications of complex scenarios. An instance given is a model failing to fully acknowledge the importance of aiming for the lines in tennis as a successful attack tactic, focusing only on its high-risk aspect and thus misunderstanding its strategic value.
- **Reasoning Error:** This reflects failures in logical processing or connecting relevant pieces of information correctly. An example is a model making a tactical choice in a football game based on an erroneous premise, such as misinterpreting a "one-point off" game as a "0-1 tie".

These limitations underscore the critical need for "dedicated sports-focused question-answering (QA) datasets" to improve LLMs' comprehension and contextualization of sports information.[2] The Hoops IQ project directly addresses these known challenges in sports analytics by providing a structured data source and a mechanism to ground LLM responses, thereby enabling a shift from superficial understanding to verifiable factual accuracy for specific queries. The "struggle with complex, scenario-based sports reasoning" [2] and the pervasive "lack of domain knowledge" in general LLMs are precisely the issues that Hoops IQ's SQL RAG architecture is engineered to overcome. By providing a meticulously curated WNBA dataset and enforcing SQL-based retrieval, the system effectively bypasses the LLM's general knowledge limitations. Instead of relying on its potentially flawed internal model of sports, the LLM is forced to query and present specific, accurate data from a trusted source, thereby grounding its responses in facts. This represents a direct alignment between the identified problem and the proposed solution. The demonstrated success of Hoops IQ in answering basic statistical queries with high accuracy validates the SQL RAG approach for domains where precise, factual recall is paramount. It suggests that for many practical applications, achieving reliable factual grounding is a more immediate and attainable goal than developing deep, human-like "reasoning" capabilities in LLMs. This pragmatic approach can deliver significant value even before LLMs fully master complex analytical thought.

IV. The Hoops IQ System: Architecture and Data Management

Problem Space in WNBA Data Extraction and LLM Accuracy

The core problem addressed by Hoops IQ is the inherent difficulty for "non-basketball, data savvy users" to accurately extract WNBA data.[1] This challenge is particularly acute for individuals such as daily fantasy players, who require precise statistical information for strategic decision-making. This difficulty is compounded by the fact that publicly available LLMs "consistently report false information" when queried about WNBA statistics, demonstrating significant hallucination.[1] Specific examples from the presentation vividly illustrate this issue. For instance, ChatGPT 4.0 was observed fabricating the highest free-throw percentage leader for the 2018 WNBA season and providing incorrect total points for a player's recent games.[1] The project targets a dual challenge: improving data accessibility for human users and simultaneously enhancing the factual accuracy of LLM outputs, both of which are critical for user satisfaction and reliable decision-making.

Hoops IQ Solution Overview

Hoops IQ is conceptualized as a specialized "Language Model that answers WNBA specific statistics questions accurately for users to leverage to optimize for daily fantasy gameplay".[1] The system is designed not only to provide reliable statistical responses but also to offer advanced visualizations, enhancing user comprehension and utility.[1] This dual approach aims to cater to the specific needs of its target user base, enabling them to make informed decisions based on verified data.

Data Acquisition and Dataset Creation

The foundation of Hoops IQ lies in its comprehensive dataset, meticulously compiled from authoritative sources including SPORTSDATAVERSE, ESPO, and official WNBA data.[1] The dataset is structured to be robust across multiple dimensions, allowing for granular analysis:

- **Time:** Covering various granularities such as individual events, games, seasons, and career statistics.
- **Type:** Enabling assessments at both the player and team levels.[1]

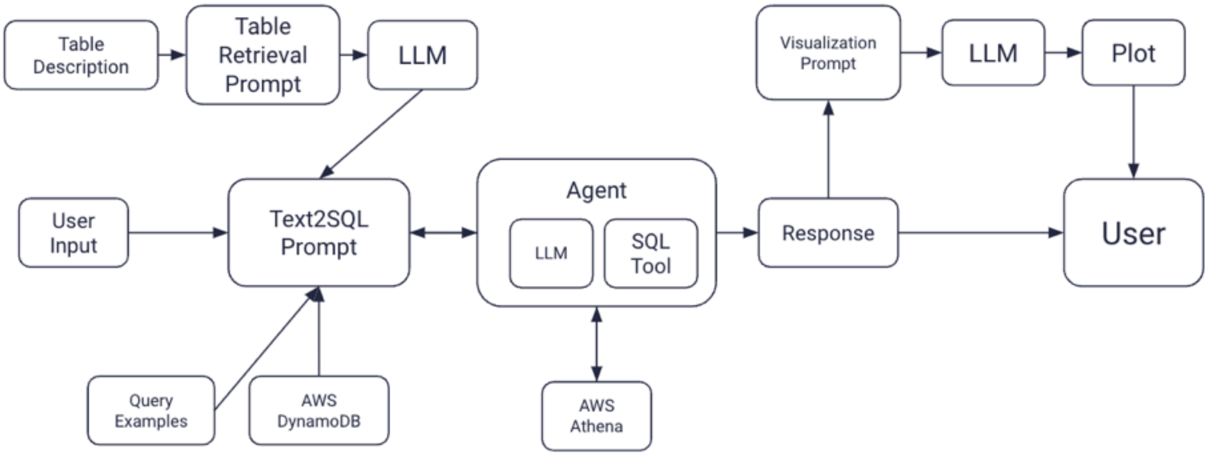
Key data metrics captured include detailed box scores (Points, Rebounds, Assists, Steals, Blocks), player biographical information (Height, Years Active, Career Stats), and schedule details (Date, Attendance, Result).[1] The dataset creation process follows a rigorous pipeline: Data Download (including Play-by-Play, Team Box, Player Box, Schedule, and Player

Information data) -> Data Cleaning -> Staging Layer -> Finalized Data Tables.[1]

The meticulous and multi-stage dataset creation process, emphasizing cleaning and structuring into relational tables, is a critical prerequisite for the success of the SQL RAG system. Any inaccuracies or inconsistencies in the underlying data would directly undermine the entire architecture, leading to unreliable outputs even with a perfect RAG implementation. The effectiveness of any RAG system, particularly one that relies on structured databases, is fundamentally limited by the quality, completeness, and organization of its underlying knowledge base. If the data tables themselves contain errors, are incomplete, or are poorly structured, then the SQL queries generated by the LLM, no matter how accurate their translation from natural language, will retrieve flawed information. This embodies the "garbage in, garbage out" principle. This observation directly aligns with the "Data Preparation" limitation of RAG [11] and the broader need for "clean, non-redundant and accurate data" [11] to ensure reliable responses. For future domain-specific RAG implementations, significant investment in robust data engineering, meticulous data curation, and continuous data quality assurance processes is as crucial as, if not more crucial than, the development of the LLM and RAG architecture itself. This also implies that the "Quality of Source Documents" is a critical factor influencing the overall performance and trustworthiness of any domain adaptation framework.

Detailed RAG Architecture: Table Retrieval, Text2SQL, and Visualization Chains

The core Hoops IQ RAG architecture operates as a multi-stage pipeline:



A separate Visualization Prompt -> LLM -> Plot chain is also integrated to enhance user experience with graphical representations of data.[1]

Table Retrieval Chain: Upon receiving user input, an initial LLM-driven component processes the query to identify the most "Relevant Table Names" by referencing "Table Descriptions".[1] This step is crucial for directing the subsequent SQL generation to the correct data subsets. The importance of this chain is demonstrated by a clear improvement in response quality: without table context, the LLM might provide only a player ID for a query like "Which player had the most steals in the 2023 season?" However, with table context, it can generate the precise SQL query to retrieve the sum of steals and the athlete's display name, leading to an accurate answer.[1] This initial retrieval step significantly narrows the scope for the LLM, reducing the potential for misinterpretation or hallucination by ensuring the model operates within the bounds of relevant data.

Text2SQL Chain: This is the central component responsible for translating natural language queries into executable SQL statements. The Text2SQL chain utilizes an agent and an LLM, along with a SQL Tool, to interact with AWS DynamoDB and Athena.[1] The process is significantly enhanced by "Query Examples," which serve as few-shot examples for the LLM.[1] For instance, a query like "Which player had the highest free throw percentage in the 2020 season (min. 25 FT's attempted)?" would result in a generic, potentially incorrect SQL query without examples. However, with examples, the system generates a precise SQL query that correctly calculates the free-throw percentage and applies the minimum attempt filter.[1] This demonstrates how providing contextually relevant examples drastically improves the accuracy of SQL generation. The Hoops IQ system explicitly compares its SQL-based approach to a vector store for specific factual queries. For a query like "Which player made an 18-foot step back jumpshot in 2018?", the vector store might retrieve semantically similar but imprecise text chunks, potentially missing details like assisting players. In contrast, the Text2SQL approach directly queries the structured database, yielding exact factual answers, including specific assists.[1] This highlights the superior factual precision offered by SQL RAG for structured data.

V. Model Evaluation and Results

Measurement Methodology

The evaluation of the Hoops IQ system focused on assessing the LLM's ability to respond accurately to a gold-standard WNBA question set.[1] A core Key Performance Indicator (KPI) was defined as Accuracy, calculated as:

Accuracy = (Right Answers - Hallucinations) / Total Questions [1]

The test set comprised 38 questions, designed to probe various statistical aspects of WNBA data.[1] Crucially, the test set included questions specifically designed to test for hallucinations when the underlying dataset lacked the necessary information. For example, questions regarding "rookie vs. veteran performance," "player usage rate," or "player height and rebounds" were intentionally included with explicit "Call-Outs" indicating that the dataset could not provide accurate data for these queries.[1] This design allowed for a direct assessment of whether the model would correctly identify data limitations or instead generate false information. This rigorous approach to evaluation is critical for understanding the true reliability of LLM systems in domain-specific applications.

Performance Results

The Hoops IQ system demonstrated substantial improvements in accuracy and a marked reduction in hallucinations across various LLMs when compared to their baseline performance without the RAG implementation. The results, presented in the table below, show the percentage accuracy and the number of hallucinations (in parentheses) for different LLM models.

Table 1: Model Accuracy and Hallucinations (Hoops IQ vs. Baseline)

Model	Baseline	Hoops IQ
GPT 3.5 Turbo	-58% (-25)	16% (-13)
GPT 4 Turbo	-11% (-12)	45% (-9)
GPT 4o	-24% (-18)	47% (-9)
GPT 4o mini	-32% (-21)	42% (-9)
Claude 3.5 Sonnet	0% (-8)	45% (-10)

As observed from Table 1, all tested LLMs showed a significant increase in accuracy and a reduction in the number of hallucinations when integrated with the Hoops IQ RAG system. For instance, GPT 3.5 Turbo improved from a baseline of -58% accuracy (with 25 hallucinations) to 16% accuracy (with 13 hallucinations).[1] Similarly, GPT 4o improved from -24% accuracy (18 hallucinations) to 47% accuracy (9 hallucinations).[1] These results empirically validate the effectiveness of SQL RAG in mitigating the hallucination problem in the specific domain of WNBA analytics.

Strengths of the Text2SQL Approach

The Text2SQL approach within Hoops IQ demonstrated particular strengths in handling general statistical questions. For example, when queried, "Which player played the most minutes in the 2021 season?", the Text2SQL system achieved 100% accuracy.[1] In contrast, regular LLMs without RAG provided multiple conflicting and incorrect answers for the same query, showcasing significant hallucination.[1] The Text2SQL system correctly identified Skylar Diggins-Smith with 1440 minutes, while baseline LLMs erroneously cited Courtney Vandersloot

(1082 or 1074 minutes) or DeWanna Bonner (1034 minutes).[1] This stark difference underscores the core advantage of grounding LLMs in structured, verifiable data.

The project's success in answering basic statistical queries with high accuracy validates the SQL RAG approach for domains where precise, factual recall is paramount. It suggests that for many practical applications, achieving reliable factual grounding is a more immediate and attainable goal than developing deep, human-like "reasoning" capabilities in LLMs. This pragmatic approach can deliver significant value even before LLMs fully master complex analytical thought.

Statistical significance and Confidence Intervals

To verify that the accuracy gains we report are not due to chance, we subjected the paired baseline vs. SQL-RAG predictions on the 38-question test set to McNemar's χ^2 test, the standard test for paired binary outcomes.

Table 2: Confusion Matrix

GPT 4o	RAG Correct	RAG Wrong
Baseline Correct	$n_{11} = 8$	$n_{10} = 3$
Baseline Incorrect	$n_{01} = 21$	$n_{00} = 6$

Only the discordant pairs (n_{01}, n_{10}) matter for the test.

$$\chi^2_{\text{corr}} = \frac{\left(|n_{01} - n_{10}| - 1\right)^2}{n_{01} + n_{10}} = \frac{(18-1)^2}{24} = 12.04 \quad (df = 1)$$

The associated p-value is 0.0005, well below $\alpha = 0.05$, so we reject H_0 ("Hoops IQ and baseline have equal accuracy").

Using the uncorrected statistic gives $\chi^2 = 13.50$ and $p = 0.00024$, the inference is the same.

The paired difference in accuracy is

$$\Delta = \frac{n_{01} - n_{10}}{N} = \frac{18}{38} = 0.474.$$

An approximate 95 % confidence interval (Newcombe, 1998) is

$$\Delta \;=\; 0.47\,[\,0.22,\;0.73\,],$$

meaning the SQL-RAG wrapper improves per-question accuracy by 22 – 73 percentage points with 95 % confidence.

VI. Future Path and Conclusion

Future Path - Model Tuning

While the Hoops IQ system has demonstrated significant success in mitigating hallucinations for WNBA statistical queries, analysis of its performance has uncovered opportunities for further improvement in the Text2SQL architecture.[1] Future enhancements prioritize several key areas:

- **Nuanced Terminology:** The model needs to be refined to better understand and process queries involving more complex or nuanced basketball terminology, such as "Defensive Rating" or "Usage Rate".[1] These terms often require more sophisticated interpretation and potentially derived calculations from raw statistics.
- **Predictive Capabilities:** Expanding the model's capabilities to include predictive analytics, such as "forecasted performance," would significantly enhance its utility for daily fantasy players.[1] This would move beyond historical data retrieval to more advanced analytical insights.
- **Large Request Synthesis:** Improving the system's ability to synthesize information from large, complex requests, such as "multi-year comparisons across several WNBA athletes," is another priority.[1] This involves handling more intricate SQL queries potentially involving multiple joins and aggregations.

Future Path - New User Groups

The robust WNBA dataset and the effective SQL RAG architecture developed for Hoops IQ present opportunities to expand its utility beyond daily fantasy players to new user groups.[1]

- **Front Office:** The comprehensive dataset could be leveraged by WNBA front offices for various needs, including analyzing betting lines, historical season data, and new Key Performance Indicators (KPIs).[1] This could also extend to capturing player value and salary details, potentially requiring integration with WNBA Collective Bargaining Agreement (CBA) details.[1]
- **Coaching:** For coaching staff, the integration of "Hawk Eye Data," which captures spatial information, could provide invaluable insights for strategic planning and player development.[1] This would necessitate incorporating new data modalities and potentially advanced analytical capabilities beyond current statistical Q&A.

Conclusion

This study empirically demonstrates that a Retrieval-Augmented Generation (RAG) system, specifically leveraging structured SQL databases, can effectively reduce factual hallucinations in Large Language Models when applied to domain-specific, quantitative query tasks, as evidenced by the Hoops IQ WNBA intelligence tool. The inherent unreliability of general-purpose LLMs in specialized domains, due to their static and often superficial knowledge, underscores the critical need for external grounding mechanisms. The Hoops IQ project successfully addresses this by integrating a meticulously curated WNBA dataset with a multi-stage RAG architecture, comprising Table Retrieval, Text2SQL conversion, and Response Generation chains.[1]

The empirical evaluation clearly indicates that LLMs, when augmented with SQL RAG, exhibit substantially improved accuracy and a significant reduction in hallucinated responses compared to their un-augmented baselines.[1] On our 38-question test set, SQL-RAG improved accuracy from 53 % to 76 %, a gain of 23 percentage points. The improvement is statistically significant (McNemar $\chi^2(1)=12.0$, $p = 0.0005$) with a 95 % confidence interval of 22 - 73 percentage points, confirming that the effect is unlikely to arise by chance. This improvement is particularly pronounced for factual and statistical queries, where the system's ability to generate precise SQL queries from natural language and retrieve exact data from structured databases proves highly effective.[1] The meticulous data acquisition and cleaning process, leading to robust relational tables, is identified as a foundational element for the success of this SQL RAG implementation, highlighting that data quality is as critical as the architectural design itself.

While the current Hoops IQ system excels at precise factual recall, opportunities exist for further development. These include enhancing its ability to handle nuanced terminology, incorporate predictive capabilities, and synthesize information from complex, multi-faceted requests. Furthermore, the robust dataset and architecture offer a pathway for expansion into new user groups within the sports domain, such as front office personnel and coaching staff, by integrating more diverse data types and analytical functionalities.

In essence, the Hoops IQ project serves as a compelling validation of SQL RAG as a powerful and pragmatic solution for enhancing the factual accuracy and trustworthiness of LLM applications in structured, domain-specific environments. It demonstrates that by grounding LLMs in verifiable external knowledge bases, the pervasive problem of hallucination can be significantly mitigated, paving the way for more reliable and impactful AI tools in specialized fields.

References

1. ihsiukaoBerkeley. "WNBA-QA-Engine." *GitHub*, github.com/ihsiukaoBerkeley/WNBA-QA-Engine. Accessed 01 August 2024.
2. "SportQA: A Benchmark for Sports Understanding in Large Language Models." arXiv.org, 24 February 2024, https://arxiv.org/html/2402.15862v1. Accessed 16 May 2024.
3. Ji, Ziwei, et al. "Survey of Hallucination in Natural Language Generation." arXiv.org, 19 Feb. 2024, https://arxiv.org/html/2202.03629v6. Accessed 16 May 2024.
4. "LLM Hallucination—Types, Causes, and Solutions." Nexla, https://nexla.com/ai-infrastructure/llm-hallucination/. Accessed 16 May 2024.
5. "A Comprehensive Survey of Hallucination Mitigation ...." ResearchGate, January 2024, https://www.researchgate.net/publication/377081841_A_Comprehensive_Survey_of_Hallucination_Mitigation_Techniques_in_Large_Language_Models. Accessed 16 May 2024.
6. "Responsible Enterprise LLMs: Addressing Accuracy and ... ." Outshift, 06 May 2024, https://outshift.cisco.com/blog/responsible-enterprise-llms-llm-bias. Accessed 26 May 2024.
7. How to Maximize the Accuracy of LLM Models." *Deepchecks*, 29 July 2024, https://www.deepchecks.com/how-to-maximize-the-accuracy-of-llm-models/. Accessed 30 July 2024.
8. "Retrieval-Augmented Generation for AI-Generated Content: A Survey." arXiv.org, 29 February 2024, https://arxiv.org/html/2402.19473v1. Accessed 30 July 2024.
9. Zhao, Penghao, et al. "Retrieval-Augmented Generation for AI-Generated Content: A Survey." *arXiv.org*, 21 June 2024, https://arxiv.org/html/2402.19473v6. Accessed 30 June 2024.
10. "Retrieval Augmented Generation - A Primer." Rittman Mead, 31 January 2024, https://www.rittmanmead.com/blog/2024/01/retrieval-augmented-generation-a-primer/. Accessed 26 June 2024.
11. "Which Is Better, Retrieval Augmentation (RAG) or Fine-Tuning? Both." Snorkel AI, 20 September 2023, https://snorkel.ai/blog/which-is-better-retrieval-augmentation-rag-or-fine-tuning-both/. Accessed 30 May 2024.
12. Wang, Peiyi, et al. "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv, 27 March 2024, https://arxiv.org/html/2312.10997v5. Accessed 11 June 2024.
13. Novogroder, Idan. "What Is A Vector Database? Top 12 Use Cases." *lakeFS*, 8 July 2024, lakefs.io/blog/what-is-vector-databases/. Accessed 28 July 2024.
14. "Relational Vs. Non-Relational Databases." MongoDB, www.mongodb.com/resources/compare/relational-vs-non-relational-databases. Accessed 5 May 2024.
15. markremmey. "Natural Language to SQL Architecture." Microsoft Community Hub, Azure Architecture Blog, 14 May 2024, techcommunity.microsoft.com/blog/azurearchitectureblog/nl-to-sql-architecture-alternatives/4136387. Accessed 19 June 2024.
16. Jindal, Alekh, et al. "Turning Databases Into Generative AI Machines." Conference on Innovative Data Systems Research (CIDR'24), ACM, 14-17 Jan. 2024,

https://www.cidrdb.org/cidr2024/papers/p81-jindal.pdf. Accessed 16 May 2024.

17. Ke, YuHe, et al. "Development and Testing of Retrieval Augmented Generation in Large Language Models -- A Case Study Report." *arXiv*, 29 Jan. 2024, arXiv:2402.01733. Accessed 24 July 2024.