Large Language Model Result Post-Processing and Verification - RAG system evaluation

1st Sicheng Dong
Chair of Robotics, Artificial Intelligence, and Real-Time Systems
Technical University of Munich
Munich, Germany
go98xah@tum.de

***Abstract*—Large language modeling (LLM) has become a significant research focus and is utilized in various fields, such as text generation and dialog systems. One of the most essential applications of LLM is Retrieval Augmented Generation (RAG), which greatly enhances the reliability and relevance of generated content. However, evaluating RAG systems remains a challenging task. Traditional evaluation metrics struggle to effectively capture the key features of modern LLM-generated content, which often exhibits high fluency and naturalness. In this study, we propose an LLM-driven evaluation method for RAG systems. Our approach leverages LLMs to assess and analyze RAG performance across multiple dimensions, including Response Relevancy in the retrieval component , Factual Correctness and Faithfulness in the generation component. By incorporating these comprehensive evaluation criteria, we hope to gain a deeper understanding of the RAG system. In order to validate our proposed approach, we compare our results with a semantic similarity-based evaluation, which shows a strong correlation between the two. Furthermore, we compare our approach with RAGAS, a well-known RAG evaluation framework. We find similar patterns in specific evaluation metrics. Finally, we discuss potential future research directions for the evaluation of RAG systems Challenges and opportunities**

***Index Terms*—Retrieval Augmented Generation, Large Language Model, Assessment.**

I. Introduction

Large Language Models (LLMs) are one of the hottest research topics in artificial intelligence today, and they have proven to be extremely powerful in a variety of fields, including healthcare and education. [1] [2] Despite their strong performance, LLMs nevertheless have a number of serious drawbacks. For instance, they frequently lack the knowledge required to respond to domain-specific queries. [3] Furthermore, LLM databases eventually get out of date and can no longer address today's issues. [4]

Researchers have taken two primary approaches to solving these issues: **fine-tuning the model using domain-specific data** and **connecting the model to additional external information sources**. [5] Although fine-tuning is a straightforward and effective approach, it has some obvious drawbacks, such as the scarcity of high-quality domain data and the high computational cost of the training process. [6] As a result, the second strategy, known as the **Retrieval Augmented Generation (RAG) system**, is increasingly being used in research. By accessing external data sources, this approach can search for domain-specific data in real-time without the need for extensive training. [7] In addition, RAG is also regarded as an effective structure to solve the problem that the system to generate inaccurate or misleading information (*LLM hallucination*). [8]

A RAG system consists of two key components: a *retriever* and a *generator*. The retriever will fetch relevant information based on the given input, and the generator then utilizes the information from the retriever to produce the final output. [9]

When applying them to real-world projects, we must understand the reliability and effectiveness of RAG systems. The whole evaluation process not only includes assessing the quality of the final generated output but also analyzing the retriever's ability to fetch relevant information and examine the interaction between the retriever and generator components. Traditional evaluation methods, such as word-overlap-based metrics (e.g., BLEU [10], ROUGE [11]) or pre-trained model-based methods (e.g., BERTScore [12]), struggle to effectively capture the semantic richness of modern LLM-generated text and give a perfect evaluation.

Therefore, researchers have begun to focus on LLMs as evaluators for assessing RAG systems. Several well-known evaluation frameworks, such as RAGAS [13], have already achieved significant progress in this field. Compared to traditional evaluation methods, our LLM-driven approach demonstrates great advantages in both efficiency and accuracy since most of the work can be done by LLMs themselves, reducing manual intervention and enhancing sensitivity to linguistic nuances. [14]

In this study, inspired by evaluation frameworks like RAGAS, we propose a novel evaluation method designed to provide a more comprehensive assessment of RAG systems. Furthermore, we validate the effectiveness of the proposed approach to ensure its reliability and applicability in real-world scenarios.

A. Research Questions

The Research Questions (RQs) in this work are:

- **RQ1:** How can a novel evaluation method be designed to assess RAG system performance?
- **RQ2:** To what extent is the proposed evaluation method reliable?

- **RQ3:** How does the performance of our evaluation method compare to the performance of RAGAS framework?

The contributions of our work are as follows:

- For **RQ1**, we propose a novel evaluation framework for quantitatively assessing the generation and context retrieval capabilities of RAG systems. This framework leverages LLMs as evaluators and incorporates multiple dimensions with distinct metrics to evaluate comprehensively the generated outputs.
- For **RQ2**, we compute the semantic similarity between the final results generated by our method and the given reference. The reliability of our approach is supported by analyzing the relationship between different LLM-driven evaluation scores and corresponding semantic similarity scores.
- For **RQ3**, we utilize the RAGAS framework and conduct experiments under controlled conditions using the same evaluation LLM, embedding model, and dataset. We assess the correlation between our method and the RAGAS framework by comparing their scores on the same evaluation metrics.

All related results can be seen here.

II. Related Work

A. Atomic facts

The definition of atomic facts states that they are the smallest units of information that can stand alone and be evaluated independently. [15] By segmenting a passage into distinct atomic facts, we can better understand its central meaning. Particularly in question answering and RAG evaluation, methodologies based on atomic facts have achieved significant success. [13] [15] [16] A simple example is illustrated in Table I for decomposition.

Original Sentence:
Theron Shan is a man who has given over his life in service to the Republic, using work to try and cope with abandonment issues gained from being hurt too many times by those who were supposed to love him.
Atomic Facts 1:
Theron Shan is a man.
Atomic Facts 2:
Theron Shan has given over his life in service to the Republic.
Atomic Facts 3:
Theron Shan uses work to try and cope with abandonment issues.
Atomic Facts 4:
Abandonment issues are gained from being hurt too many times by those who were supposed to love him.

TABLE I: Decomposition Of The Original Sentence Into Atomic Facts

[17]

B. RAGAS

RAGAS is a comprehensive testing framework for RAG systems, which implements diverse evaluation metrics. [13] Besides traditional metrics, it also leverages LLMs as evaluators to systematically assess RAG systems. The framework defines a wide range of metrics, some of which are outlined below:

FactualCorrectness is a metric that compares and evaluates how factually accurate the generated response is compared with the reference. This metric is used to determine the extent to which the response aligns with the the reference. [13]

Faithfulness measures the factual consistency between a response and the retrieved context. Higher scores mean better consistency between the generated response and the source material. [13]

Response Relevancy evaluates how relevant the generated response is to the user input. Higher scores are assigned to responses that align more closely with the user input, while lower scores are given to responses that are incomplete or include redundant information. [13]

Among the methods used in RAGAS are techniques such as splitting sentences into atomic statements and employing embedding models to compute similarity values. [13]

III. Environment Setup

In this section, we will discuss in detail the specific implementation steps of our experiment environment.

A. Datasets

All experiments in this work were conducted using the test set of the **Retrieval-Augmented Generation (RAG) Dataset 12000**, an English-language dataset specifically designed for RAG evaluation. [17] The dataset is structured around three components: **context**, **question**, and **reference answer**.

B. RAG system implementation

We constructed a basic Retrieval-Augmented Generation (RAG) system. [18] The detailed pipeline is illustrated in Figure 1.

1) Pre-processing: First, we extract all **context** entries from the database, storing each entry as an individual document. We then segment them into smaller chunks and utilize the `thenlper/gte-small` [19] model to encode them into vector representations. The resulting embeddings are then stored in a vector database.

2) Retriever: For each user query, we first embed it and then compute the cosine similarity to retrieve the Top-K most relevant documents from the vector database.

3) Generator: We use OpenAI's `GPT-4o-mini` [20] as our generator (LLM). The retrieved relevant documents are combined with the user query into a structured prompt (LLM Prompt), fed into the generator to produce the final output.

C. Evaluation set

We integrate the previously mentioned datasets with the RAG system to generate the final LLM-produced answers. These answers, along with the corresponding questions, references, and contexts from the dataset, form a new evaluation dataset. The detailed structure of the evaluation dataset is illustrated in Table II. This structure is also specified by the RAGAS evaluation framework. [13]

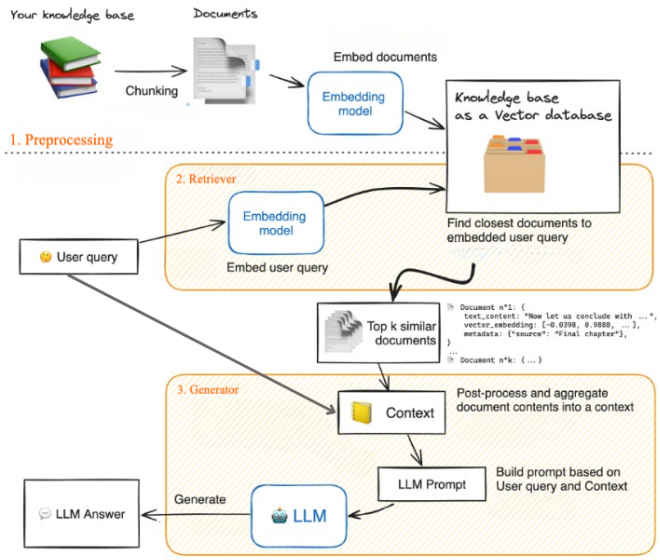


Fig. 1: RAG System [18]

Question	Context	Answer	Reference
Who is Peter?	Peter....	Peter works as a musician.	Peter is a musician.

TABLE II: Example Of An Evaluation Set Entry

IV. Methodology

The overall evaluation metrics can be categorized into two aspects: tests conducted on the retriever and tests performed on the generator. The evaluator LLM is `GPT-4o-mini` [20] and the embedding model utilized for semantic similarity is `text-embedding-3-large` [21].

A. Retriever

1) Response Relevancy: This metric measures how relevant the retrieved context is with the input (question), allowing us to assess the effectiveness of the retriever in selecting relevant contexts.

a) Experimental idea: The metric is based on the hypothesis that if a context is highly relevant to a question, we should be able to reconstruct a similar question from the given context. The higher the similarity, the more relevant the retrieved context is. [13]

As the Pipeline shown in Figure 2, the overall evaluation method consists of three steps:

1) We provide the **context** as input to the LLM, prompting it to generate three related questions. The generated questions should accurately reflect the content of the context.
2) We compute the cosine similarity between each generated question and the original question to quantify their similarity.
3) We take the **maximum cosine similarity value** among the three generated questions as our final evaluation score.

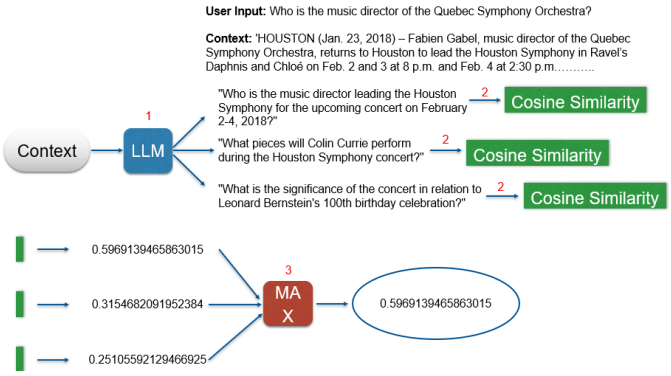
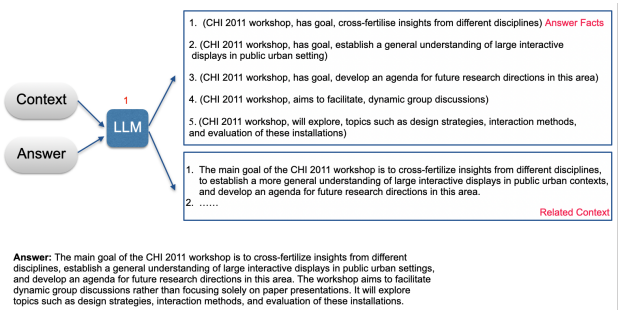
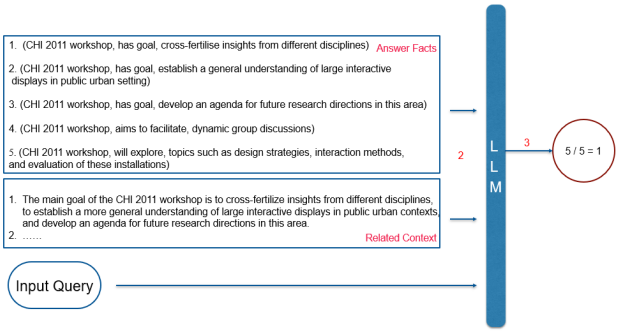


Fig. 2: Response Relevancy Pipeline

B. Generator



(a) First Step Of Faithfulness Evaluation



(b) Second And Third Step Of Faithfulness Evaluation

Fig. 3: Pipeline For Faithfulness

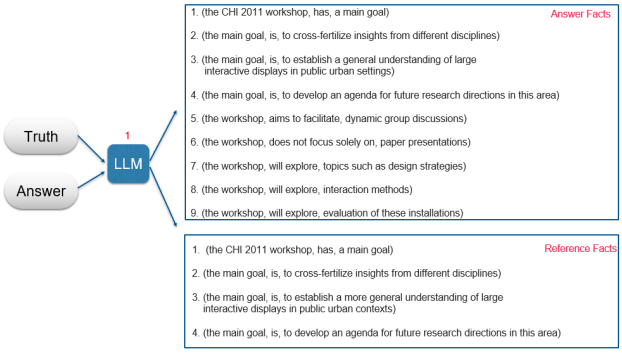
1) Faithfulness: This metric measures how relevant the answer is with the context retrieved by the retriever, thereby evaluating how well the generated response aligns with the context.

a) Experimental idea: The metric is based on the hypothesis that if the context and the generated answer are relevant, the context should support each atomic fact in the answer. The more atomic facts from the answer verified by the context, the higher the relevance. [13] [16]

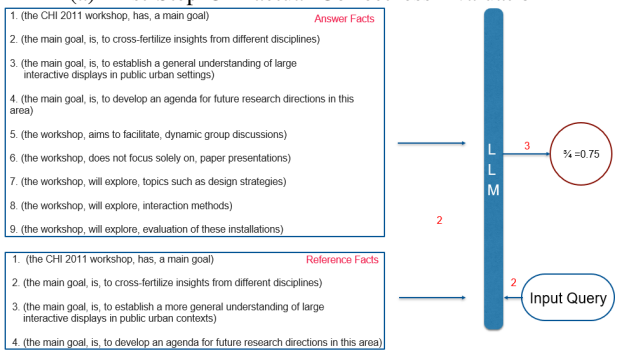
As the Pipeline shown in the Figure 3, the overall evaluation method consists of three parts:

1) We provide the **answer** and the retrieved related **context** to the LLM, prompting it to extract atomic facts from the answer and identify the necessary supporting context passages. We define the desired atomic fact structure using a triplet format: `(subject, predicate, object)`. By enforcing this structured representation, the LLM can better extract atomic facts in a standardized manner.
2) We then embedded the extracted atomic facts and context into a prompt and provided it to the LLM again, instructing it to determine whether the retrieved context supports each atomic fact.
3) Finally, we compute the proportion of supported atomic facts out of the total extracted facts as the final evaluation score. The specific calculation formula is shown in the figure below:

$$\text{Faithfulness} = \frac{\text{Number of supported atomic facts}}{\text{Number of total extracted facts}} \quad (1)$$



(a) First Step Of Factual Correctness Evaluation



(b) Second and Third Step Of Factual Correctness Evaluation

Fig. 4: Pipeline For Factual Correctness

2) Factual Correctness: This metric measures the accuracy of the generator's answer compared to the reference answer, thus evaluating the correctness of the generated answer comparing with the reference.

a) Experimental idea: The metric is based on the idea that if the answer and reference are similar, they should share a significant number of atomic facts. [13] [16]

As shown in Figure 4, the overall process consists of three steps:

1) We provide both the **generated answer** and the **reference** to the LLM, prompting it to extract atomic facts for both the answer and the reference. As before, we use the triplet structure for atomic facts: `(subject, predicate, object)`.
2) We then embed the extracted atomic facts from both the answer and the reference into a prompt and provide it to the LLM, asking it to compute the following two values:
   - True Positive (TP): Atomic facts that appear in both the answer and the reference. [13]
   - False Negative (FN): Atomic facts present in the reference but missing in the answer. [13]
3) Finally, we calculate the recall value as the final evaluation score by applying the standard formula [22]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

V. Results Analysis

In this section, I will provide a detailed explanation of the validation of our method's result and a performance comparison with the RAGAS framework. We utilize **cosine similarity** as a metric for semantic similarity validation for both validations.

A. Validation Result compared with Semantic Similarity

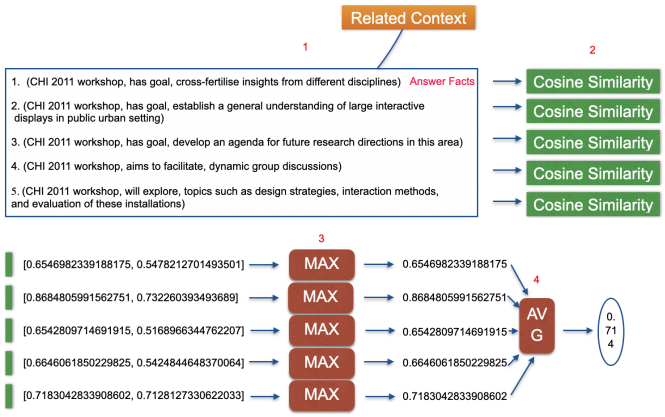


Fig. 5: Validation Of Faithfulness Pipeline

1) Result of Faithfulness:

*a) **Hypothesis**:* If an LLM assigns a high score to a particular example, the atomic facts in the answer and the related context should exhibit a significant similarity value.

As shown in Figure 5, we continue using the examples from the previous Faithfulness pipeline.

The steps can be divided into four:

1) **Validation Step 1**: We extract the atomic facts and related context from the Faithfulness score generated in the first step of the Pipeline.
2) **Validation Step 2**: We calculate the cosine similarity between each Faithfulness and each related context.

3) **Validation Step 3**: For each set of cosine similarity values associated with an atomic fact, we take the **maximum value** as the degree of Faithfulness in the validation.
4) **Validation Step 4**: We calculate the **average** of all the maximum values as the final validation score.

We apply the aforementioned validation steps to the entire dataset and divide the dataset into two groups. The corresponding table is shown in Figure 6. When LLM Faithfulness > 0.6, the average cosine similarity of the corresponding atomic facts is 0.75, whereas when LLM Faithfulness < 0.4, the average cosine similarity is 0.43. We observe a gap between these values, and a correlation exists between LLM Faithfulness and cosine similarity.

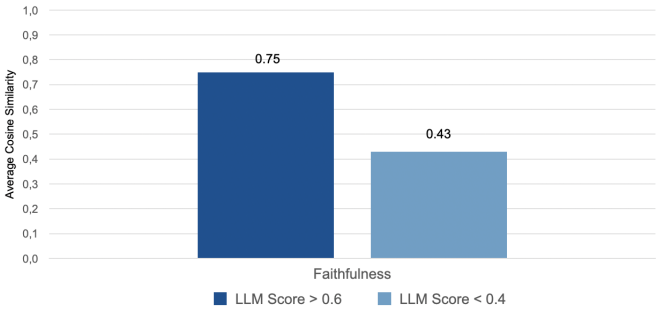


Fig. 6: Validation Result of Faithfulness

To further quantify this relationship, we computed the **Spearman correlation coefficient** [23] between the cosine similarity scores and the LLM-driven scores, obtaining $\rho = 0.3447$ with $p < 10^{-10}$, as shown in Figure 7. This result indicates a **statistically significant but weak monotonic positive correlation** between the two, which to some extent supports the hypothesis.

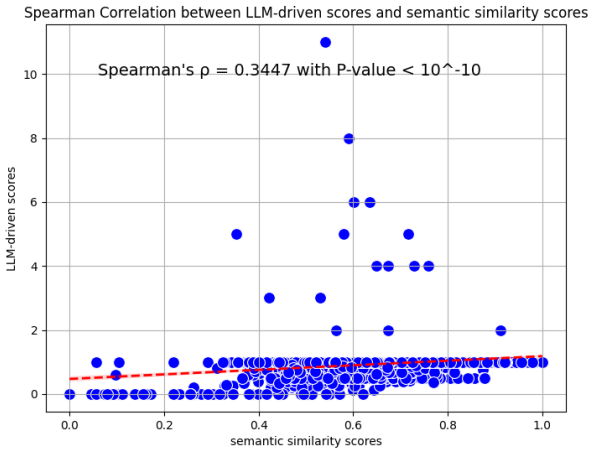


Fig. 7: Spearman Correlation For Faithfulness

2) Validation of Factual Correctness: In our validation of *Factual Correctness*, we employ a methodology similar to that used for *Faithfulness* validation.

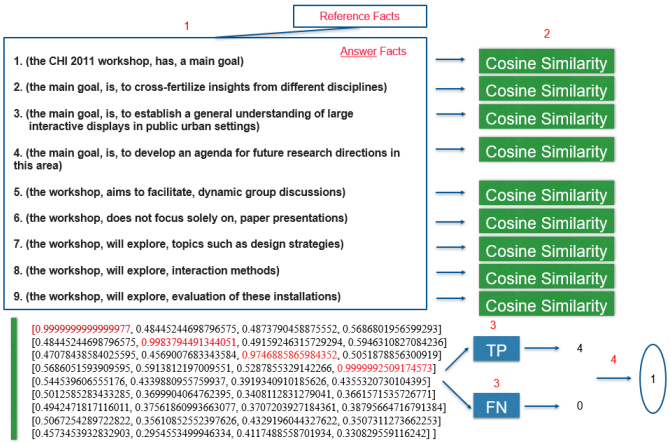


Fig. 8: Validation Of Factual Correctness Pipeline

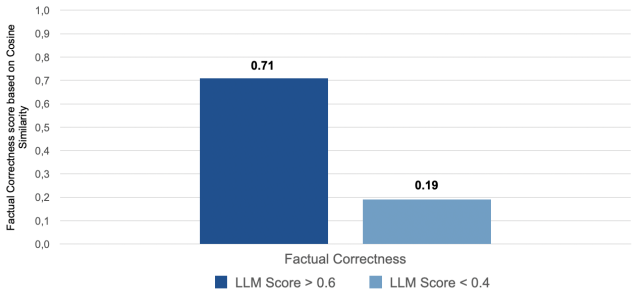


Fig. 9: Validation Result of Factual Correctness

*a) **Hypothesis**:* if the *LLM's Factual Correctness score is high*, then the *similarity score between atomic facts in the answer and the reference should also be high*, indicating a correlation between the two.

As illustrated in Figure 8, our validation process consists of four steps:

1) **Validation Step 1**: We extract atomic facts from both the *LLM-generated answer* and the *reference*.
2) **Validation Step 2**: We compute the *cosine similarity* between each atomic fact in the answer and the reference, obtaining a *cosine similarity matrix*.
3) **Validation Step 3**: We set **0.9** as a threshold and determine TP and FN values based on the following criteria:
   - A **true positive (TP)** occurs if, for a given row in the similarity matrix, there exists a value bigger than the threshold.
   - A **false negative (FN)** occurs if, for a given column in the similarity matrix, **no cosine similarity** value is bigger than the threshold.
4) **Validation Step 4**: We compute the final *score* using formular 2.

Similarly, we apply the above method to the entire dataset. As in Faithfulness validation, we define two groups based on the LLM-driven scores. As shown in Figure 9, when the LLM score exceeds 0.6, the cosine similarity-based method yields

an average score of 0.71. In contrast, when the LLM score is less than 0.19, the cosine similarity-based method produces an average score of only 0.19. This reveals an even more significant gap compared to Faithfulness validation.

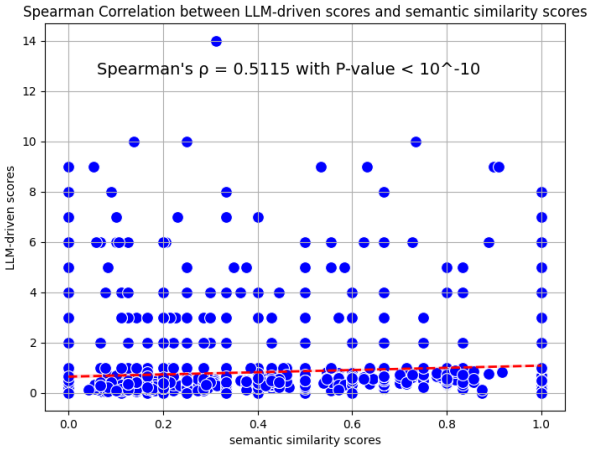


Fig. 10: Spearman Correlation For Factual Correctness

We also computed the **Spearman correlation coefficient** [23], getting $\rho = 0.5115$ with $p < 10^{-10}$, as illustrated in Figure 10. This result indicates a **statistically significant moderate monotonic positive correlation** between the two, strongly supporting our hypothesis.

B. Comparison with Ragas

We use the same embedding model and LLM evaluator to run our dataset within the RAGAS framework and compare their performance in various metrics. The detailed results are presented in Figure 11. We compare the two methods based on two key metrics: **standard deviation** and **average score**.

We observe that both methods exhibit similar values in Response Relevancy (0.26 to 0.21) and Factual Correctness regarding standard deviation (0.31 to 0.36). However, in Faithfulness, a significant gap exists between the two approaches (0.2 to 0.4).

In contrast, when considering the average score, the two methods produce identical values in Faithfulness (0.93 to 0.93), whereas notable differences appear in Factual Correctness (0.63 to 0.85) and Response Relevancy (0.83 to 0.65).

Based on the observed result, we can draw a preliminary conclusion. Compared to RAGAS, our method demonstrates similar distribution patterns in **Factual Correctness** and **Response Relevancy**, despite differences in the absolute score ranges. This suggests that both approaches capture similar trends in these two metrics. However, in **Faithfulness**, the two methods exhibit distinct characteristics.

The potential cause of the deviation in Faithfulness could be due to the differences in the specific implementation of the two methods. Unlike RAGAS, we also emphasize the extraction of context segments related to each atomic fact. The purpose of this is to better assess the relevancy of atomic facts within a smaller context window, thereby improving both efficiency and accuracy while reducing unnecessary redundant information. Due to this, we may obtain a higher overall score (i.e., the mean value). However, LLM may still have inaccuracies when finding most relevant part of context, this could lead to variations in the overall score distribution pattern (i.e., the standard deviation). Further research is needed to evaluate the effectiveness of this metric.

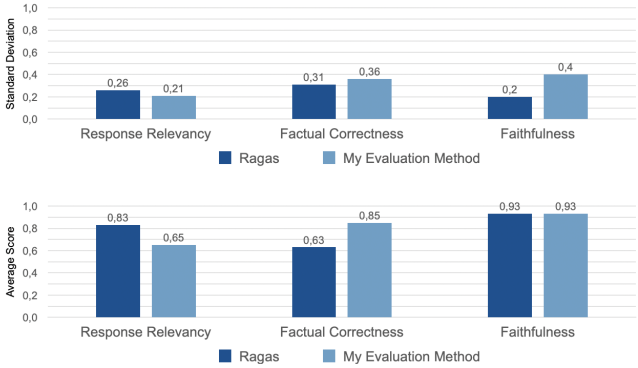


Fig. 11: Comparison With RAGAS In Average Score And Standard Deviation

VI. Conclusion and Future Scope

This paper proposes an LLM-driven approach for evaluating RAG systems. By leveraging an LLM as an evaluator and defining multi-dimensional metrics, we conduct an efficient and accurate assessment of RAG systems. To verify the reliability of the evaluation results, we employ a semantic validation method based on cosine similarity, which confirms the robustness of the proposed evaluation approach. Furthermore, to assess the performance of our evaluation framework, we compare it against the baseline framework, RAGAS. Our findings reveal similar patterns in Response relevancy and Factual Correctness between the two methods, while Faithfulness exhibits a significant discrepancy.

Future research directions include investigating the underlying reasons for the substantial differences in Faithfulness scores between our approach and RAGAS, incorporating additional evaluation metrics such as context recall, and further testing the capabilities of RAG systems in diverse scenarios, such as negative rejection and long-context accuracy. [24] [25]

References

[1] K. He et al., "A Survey of Large Language Models for Healthcare: from Data, Technology, and Applications to Accountability and Ethics," arXiv [cs.CL], 2025.

[2] Z. Zhang et al., 'Simulating Classroom Education with LLM-Empowered Agents', arXiv [cs.CL]. 2024.

[3] A. Szymanski, N. Ziems, H. A. Eicher-Miller, T. J.-J. Li, M. Jiang, and R. A. Metoyer, "Limitations of the LLM-as-a-Judge Approach for Evaluating LLM Outputs in Expert Knowledge Tasks," arXiv preprint arXiv:2410.20266, 2024.

[4] R. Dolphin, J. Dursun, J. Chow, J. Blankenship, K. Adams, and Q. Pike, "Extracting structured insights from financial news: An augmented LLM driven approach," arXiv preprint arXiv:2407.15788, 2024.

[5] J. C. dos Santos Junior, R. Hu, R. Song, and Y. Bai, "Domain-Driven LLM Development: Insights into RAG and Fine-Tuning Practices," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6416–6417, 2024.

[6] C. Jeong, "Fine-tuning and utilization methods of domain-specific LLMs," arXiv preprint arXiv:2401.02981, 2024.

[7] K. K. Y. Ng, I. Matsuba, and P. C. Zhang, "RAG in Health Care: A Novel Framework for Improving Communication and Decision-Making by Addressing LLM Limitations," *NEJM AI*, vol. 2, no. 1, p. AIra2400380, 2025.

[8] G. Perković, A. Drobnjak, and I. Botički, "Hallucinations in LLMs: Understanding and addressing challenges," in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pp. 2084–2088, 2024.

[9] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, "A Survey on RAG with LLMs," *Procedia Computer Science*, vol. 246, pp. 3781–3790, 2024.

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[11] N. Schluter, "The limits of automatic summarisation according to ROUGE," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 41–45, 2017.

[12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, 'Bertscore: Evaluating text generation with bert,' arXiv preprint arXiv:1904. 09675, 2019.

[13] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated Evaluation of Retrieval Augmented Generation," arXiv [cs.CL], 2023.

[14] L. Zhu, X. Wang, and X. Wang, "JudgeLM: Fine-tuned Large Language Models are Scalable Judges," arXiv [cs.CL], 2023.

[15] N. E. Kriman, 'Measuring text summarization factuality using atomic facts entailment metrics in the context of retrieval augmented generation', arXiv [cs.CL]. 2024.

[16] S. Min et al., 'FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation', arXiv [cs.CL]. 2023.

[17] Neural Bridge AI, 'rag-dataset-12000'. [Online]. Available: Website. [Accessed: Feb. 15, 2025].

[18] A. Roucher, 'Advanced RAG on Hugging Face documentation using LangChain'. [Online]. Available: Website. [Accessed: Feb. 15, 2025].

[19] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, 'Towards general text embeddings with multi-stage contrastive learning', arXiv preprint arXiv:2308. 03281, 2023.

[20] OpenAI, "GPT-4o-mini," May 2024. [Online]. Available: Website. [Accessed: Feb. 15, 2025].

[21] OpenAI, "OpenAI Embeddings," May 2024. [Online]. Available: Website. [Accessed: Feb. 15, 2025].

[22] D. M. W. Powers, 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,' CoRR, vol. abs/2010.16061, 2020.

[23] Clark Wissler ,The Spearman Correlation Formula.Science22,309-311(1905).DOI:10.1126/science.22.558.309

[24] S. Simon, A. Mailach, J. Dorn, and N. Siegmund, "A Methodology for Evaluating RAG Systems: A Case Study On Configuration Dependency Validation," arXiv preprint arXiv:2410.08801, 2024.

[25] Q. Leng, J. Portes, S. Havens, M. Zaharia, and M. Carbin, "Long Context RAG Performance of Large Language Models," arXiv preprint arXiv:2411.03538, 2024.