

Building an AI Teaching Assistant With RAG: Implementation and Performance Scoring Framework

Alexandre Moudjeb, Karlsruhe University of Applied Sciences, Germany
Fahmi Bellalouna, Karlsruhe University of Applied Sciences, Germany

The European Conference on Education 2026
Official Conference Proceedings

Abstract

AI-based assistants are increasingly used in educational contexts, yet large language models (LLMs) often produce inaccurate or hallucinated information, limiting their reliability for learning. This study investigates how Retrieval-Augmented Generation (RAG) can enhance the factuality and coherence of AI responses without additional model fine-tuning. We implement a multimodal educational assistant using RAG, capable of integrating textual documents, code, and images, and evaluating it on a dataset of 2,000 queries sourced from real student interactions and standardized educational exercises. Performance is measured using two complementary metrics: scientific factuality, assessing the accuracy of generated content against verified sources, and multimodal coherence, evaluating consistency across text, code, and images. A novel scoring method based on Jensen-Shannon divergence and temperature softmax is introduced, enabling automatic and dynamic evaluation of the system's responses. Validation against a set of human-annotated queries confirms the robustness of the scoring framework. Experimental results show that integrating RAG improves response accuracy by 9% and increases overall performance, demonstrating that RAG is a decisive factor in delivering reliable educational content. The multimodal approach further enhances the assistant's ability to address diverse query types, from textual explanations to code examples and visual illustrations. These findings suggest that RAG-based AI assistants can provide more trustworthy and pedagogically effective support, potentially improving student engagement and learning outcomes.

Keywords: large language models, RAG, education, scoring, performance

iafor

The International Academic Forum
www.iafor.org

Introduction

In Germany and everywhere in the world, AI-based learning assistants are increasingly part of learning contexts, from helping learners ask questions to clarify misunderstandings or solve problems. Large Language Models (LLMs) have significantly enhanced the quality of natural language-based interactions; they still suffer from important limitations. They can produce plausible but wrong or unsupported answers like a hallucination phenomenon (Ji et al., 2023; Maynez et al., 2020). Such limitations indicate that generative models essentially rely on static knowledge acquired during training, without intrinsic mechanisms by which the model can check its responses against external sources.

These limitations have been addressed through a variety of approaches, including domain-specific adaptation or fine-tuning. However, these methods usually require large amounts of annotated data and are computationally expensive, besides being difficult to keep up to date in evolving domains. In such a context, Retrieval-Augmented Generation has emerged as a paradigm that appears very effective (Lewis et al., 2020). It combines text generation by an LLM with an information retrieval phase that resorts to relevant external sources at inference time. By combining retrieval and generation techniques, RAG thus allows answers grounded in explicit documents, reducing the model's reliance on internal memory, and improving factual accuracy without expensive fine-tuning.

In the context of education, RAG is particularly interesting because it allows for integration with various sources such as textbooks, course materials, code snippets, images and can answer complex questions by explicitly leveraging these resources (Holmes et al., 2019; Radford et al., 2021). Despite this interest, few studies offer a systematic analysis of multimodal RAG architectures for educational assistants, or an evaluation framework adapted to these systems (Lewis et al., 2020; Radford et al., 2021).

Evaluating such multimodal assistants remains a major challenge. Standard automatic metrics, designed for unimodal tasks, often fail to capture essential criteria such as scientific correctness, conceptual coherence, or alignment across modalities such as text, image, or code (Maynez et al., 2020). Human evaluations, though instructive, are expensive, subjective, and hard to generalize on a large scale (Holmes et al., 2019). There is, therefore, an increasing need for automated yet rigorous evaluation frameworks that not only measure response accuracy but also multimodal coherence (Radford et al., 2021). That is the challenge that this paper aims to address by proposing a structured framework to design, implement, and evaluate a multimodal educational assistant based on RAG. This paper presents an implementation developed in FastAPI, in a modular software architecture that conforms to the principles of object-oriented programming; it embeds a multimodal pipeline that can process textual documents, code snippets, and images. Besides implementation, this paper proposes a light, but rigorous scoring framework based on two complementary dimensions: scientific factuality and multimodal coherence. This approach allows the analysis of the respective and combined impact of dynamically retrieved documents and those indexed in the RAG knowledge base. Experimental results, with state-of-the-art multimodal models, show that the integration of RAG significantly improves response accuracy and overall performance, underlining the interest of such an approach for deploying reliable and effective educational assistants.

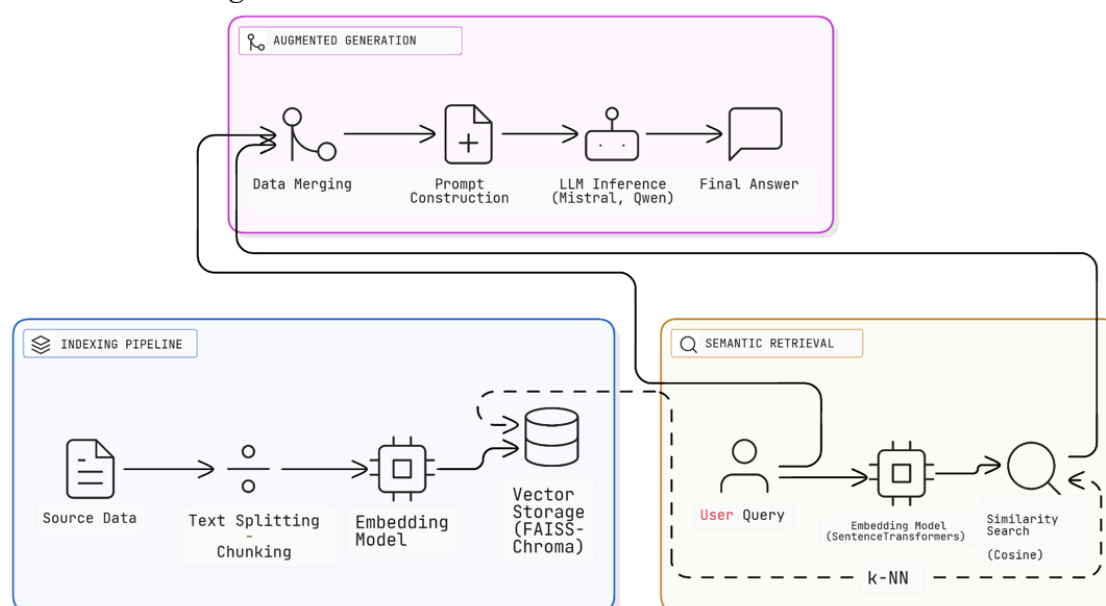
Background and Related Works

Retrieval Augmented Generation for Educational Assistants

In the context of AI-automated pedagogy, RAG is a powerful tool for increasing factuality. Indeed, finely trained models can provide correct answers across a wide range of domains to the extent that they have been trained, although certain niches or precise details may remain problematic. Consequently, for more specific knowledge or fine details of reasoning in problem solving, the model alone may not have been previously trained, and it may prove difficult to obtain a reliably generated answer, which can increase the risk of hallucination (Lewis et al., 2020).

Nowadays, there are several ways to implement an RAG system. The most common system is the use of a vector database (FAISS, Chroma) containing documents vectorized beforehand by an embedding model (e.g., a model from Sentence-Transformers). This database becomes a limited source, whose reliability depends on the quality of the indexed information, and through various processes (development of utility functions), by defining the Euclidean dot product between the components of the database vectors and those of the query, it is possible to select the vectors closest to the query and, ultimately, retrieve the important passages that the large language model can reinterpret to generate its answer (Abo El Enen et al., 2025). It would therefore be consistent to use this tool for pedagogical purposes.

Figure 1
Architectural Diagram



Note. Indexing of documents via FAISS, the vectorization of documents and the query, as well as the post-query processing leading to the generation of a response.

The use of vector database technology in the RAG process is called Dense Retrieval (Karpukhin et al., 2020; Lewis et al., 2020). Indeed, vectorization followed by comparison using similarity computation via L2 distance and cosine similarity allows for a precise quantification of the proximity between queries and passages, which makes it possible to characterize meaning and perform semantic search (Devlin et al., 2019; Karpukhin et al., 2020). What may be interesting is to couple this system with an algorithm such as BM25, which allows counting the frequency of exact words and can give weight to passages

containing a particular word. This technology is called Sparse Retrieval (Robertson & Zaragoza, 2009).

Informations Regarding FastAPI Application Development

FastAPI is designed as an efficient framework for backend development with applications in data-intensive applications as well as artificial intelligence projects (Ji et al., 2023; Maynez et al., 2020). Founded on Starlette for handling requests asynchronously and Pydantic for structuring data, it is built using Python's type annotations system that fosters reliability, automatic input validation, and the generation of OpenAPI documents dynamically. With capabilities tailored for managing multiple requests asynchronously, it can successfully process multiple document uploads, vector search, or large-scale language model evaluations at an efficient rate (Bednarz & Milosz, 2025; Chen, 2023).

In retrieval augmented generation (RAG) models, FastAPI manages these tasks: preprocessing of documents, embedding computation, vector indexing, and response generation. The modularity of FastAPI along with dependency injection enables experiments to be performed as well as replacing tasks such as embedding models without affecting other tasks (Brown et al., 2025; Chen, 2023; Gao et al., 2023).

It also allows file upload, response streaming, and multimodal data, which is necessary when working with educational assistants that process texts, code, and scientific documents. Response streaming improves interaction in conversational systems. With the help of ASGI-based solutions, FastAPI provides an efficient and reproducible method used in the development of superior artificial intelligence applications (Bednarz & Milosz, 2025; Chen, 2023; Haque, 2025).

Current Scoring Methods and Performance Evaluation Metrics

It is pertinent to mention that multimodal assistants are assessed using a set of evaluation measures to judge alignment, coherence, as well as reasoning across various modalities like text, images, or more. Among various evaluation measures used to judge alignment between multimodal pairs like an image and text, CLIPScore facilitates evaluating semantic coherence between an image and text based on the cosine similarity between them in an embedding space learned using contrastive learning (Hessel et al., 2021). Though this is helpful in judging alignment across various multimodal pairs, this measure is restricted to representational similarity only between the multimodal pairs in question (Radford et al., 2021).

On the same line, the image and text-based retrieval scores, like Recall@K, Mean Reciprocal Rank (MRR), and scores like EM1 and EM2, are used largely for the assessment of the quality of the ranks and the accuracy level of the respective responses in the context of searching and/or questioning and answering processes (Antol et al., 2015; Manning et al., 2008). EM1 calculates the exact level of the produced answer retrieved against the reference answer, whereas EM2 accounts for the tolerant form variations in the respective correct responses to be retrieved. Additionally, semantic similarity scores like SS1 and SS2 can be used for measuring the semantic similarity levels between the answer and the reference, where SS2 considers the overall weights of the segments or the modalities (Antol et al., 2015; Yu et al., 2023).

For those tasks involving complex reasoning with multimodal information, such as Visual Question Answering, ScienceQA, or MMMU challenges, the tasks are assessed using accuracy

measures to calculate the ratio of correct answers on labeled examples (Clark et al., 2022). Furthermore, recent work suggests the utilization of language models acting as independent assessors to judge OPEN ENDED responses to multimodal tasks with respect to notions of coherence, relevance, and multimodal response faithfulness.

While the strategies are highly promising, some level of subjectivity and continued reliance upon the evaluator model are still incorporated. Another method that has been attempted in the calculations of dependence among modalities on a theoretical level is the Mutual Information (MI) metric, which estimates the difference between the joint distribution of the modalities and the product of their marginal distributions (Cover & Thomas, 2006). Nevertheless, despite the robust theoretical background of the measure, there exist several limitations in applying the MI metric in a multimodal framework. The measure has no upper boundary and faces the problem of inaccurate estimates in high-dimensional representation spaces, besides being statistically noisy (Belghazi et al., 2018). Besides, the extension of this measure to large modalities remains a difficult task, making it less preferred for the task of multimodal evaluation in large scenarios.

Global System Design and Architecture

Functional Design of the Assistant and Pedagogical Objectives

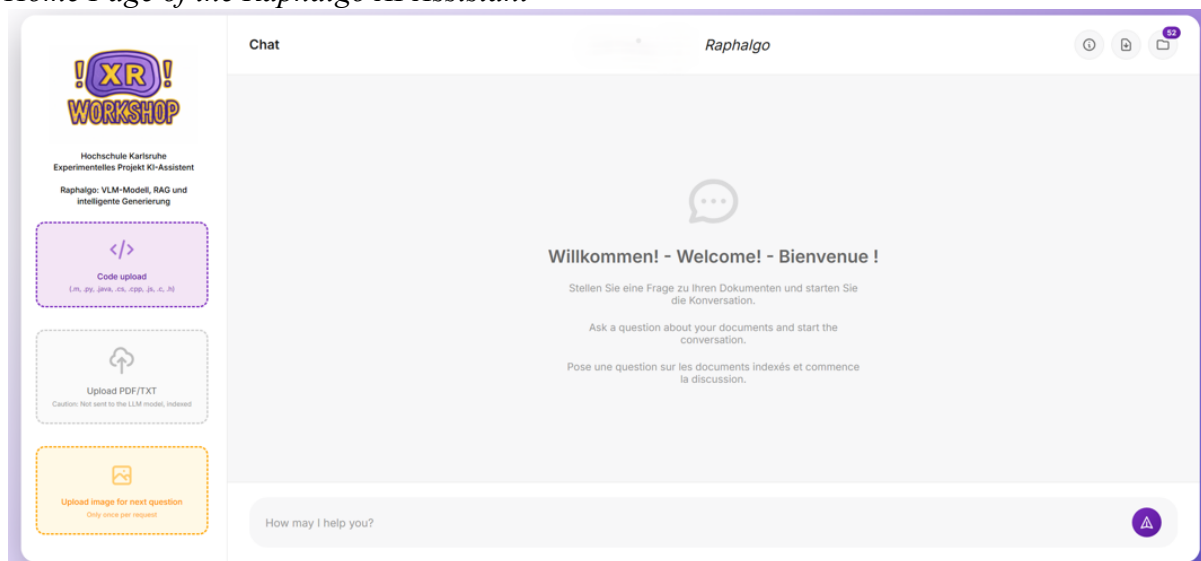
The proposed pedagogical assistant is designed as a multimodal educational system capable of answering questions in natural language, explaining technical concepts, analyzing source code, and interpreting visual statements. The assistant is restricted to an academic context in which factual accuracy, source traceability, and pedagogical coherence constitute, in our view, requirements to limit the hallucination phenomena observed in language generation models (Ji et al., 2023; Maynez et al., 2020).

The user can upload PDF or TXT documents, which constitute the assistant's knowledge base. These documents are never transmitted directly to the language model: they are indexed locally to serve as support for the information retrieval mechanism. This architectural choice is consistent with the principles of retrieval-augmented generation architecture, which aim to anchor generation in explicit and verifiable sources (Brown et al., 2025; Gao et al., 2023; Lewis et al., 2020). It also guarantees better data confidentiality and enhanced control over the content used during the generation of responses (Haque, 2025).

The assistant also supports the upload of code files (MATLAB, Python, C/C++, Java, etc.), which are stored at the user session level. This design enables iterative code analysis and pedagogical continuity across multiple interactions, in line with the educational uses of AI assistants specialized in engineering and computer science (Holmes et al., 2019). Finally, images can be associated with a single query to process exercises or problems containing visual elements, aligning with multimodal approaches studied in the fields of Visual Question Answering and AI-assisted scientific learning (Antol et al., 2015; Clark et al., 2022).

A bounded conversation history is maintained to ensure contextual coherence while limiting the size of the requests sent to the model. The assistant systematically adapts the language of its responses to that of the user, promoting multilingual use in an international educational context.

Figure 2
Home Page of the Raphalgo AI Assistant



Note. Based on Mistral (to the left, the drag-and-drop areas; at the bottom, the search bar; and at the top right, the HTML export, the document registry, and an information tab).

RAG-Based Architecture and Inference Pipeline

The architecture of the assistant is based on a Retrieval-Augmented Generation (RAG) approach combining a dense vector search engine and a large language model. This architecture, initially introduced for knowledge-intensive tasks, aims to improve the factuality and robustness of the responses generated (Gao et al., 2023; Lewis et al., 2020).

The uploaded documents are first preprocessed through text extraction (PDF parsing or direct reading), then segmented into overlapping textual fragments to preserve semantic continuity. Each fragment is then transformed into a dense vector using a sentence-transformer-type embedding model based on MiniLM (all-MiniLM-L6-v2) (Reimers & Gurevych, 2019; Wang et al., 2020). These approaches rely on pre-trained bidirectional transformer architectures, widely validated for the semantic representation of sentences and passages (Devlin et al., 2019; Goodfellow et al., 2016).

The generated vectors are stored in a FAISS (Facebook AI Similarity Search) index, designed for large-scale vector similarity search, and widely used in dense retrieval systems (Johnson et al., 2019). The associated metadata, including the document name and the textual content of the fragments, are preserved to ensure source traceability, a key point in educational and scientific systems.

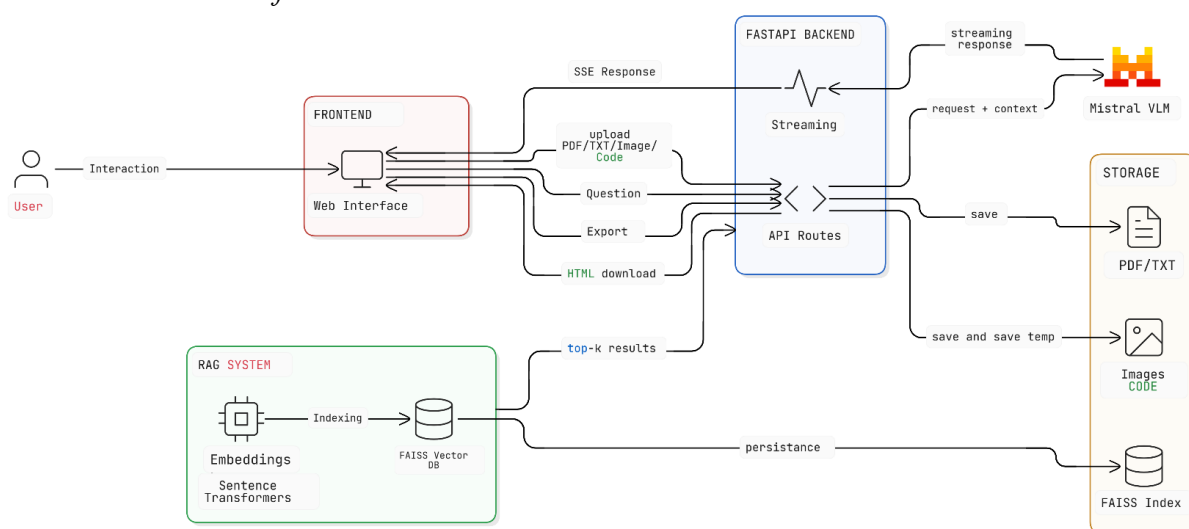
When a user query is received, it is vectorized and compared against the FAISS index to extract the k most relevant fragments, according to a dense semantic proximity measure (Karpukhin et al., 2020). These fragments are concatenated to form a documentary context injected into the prompt sent to the language model. In the absence of relevant results in the index, a fallback mechanism extracts a limited number of fragments directly from the available documents.

The FAISS index is persistent and automatically synchronized with the document directory via a process executed in the background. This approach enables the dynamic addition or removal of content without service interruption. It also makes it possible to overcome the limitations of

traditional lexical search methods, such as BM25, which rely primarily on exact term matching (Manning et al., 2008; Robertson & Zaragoza, 2009). In scientific fields such as physics or control theory, this ability to capture conceptual proximity is essential for linking physical principles to their mathematical formulations, even when terminology varies across pedagogical materials. For indexing, the text is split into fragments of 600 tokens with an overlap of 150 tokens, ensuring semantic continuity across the 32,678 generated vectors and limiting context loss in complex scientific demonstrations.

The entire system is orchestrated by a FastAPI API, which manages document ingestion, user queries, and the streaming delivery of responses. The choice of FastAPI is based on its performance and low latency in intensive application contexts (Bednarz & Milosz, 2025; Chen, 2023). The language model is executed locally and queried via a generation interface allowing the progressive return of tokens.

Figure 3
Global Architecture of the RAG AI Assistant



Prompt Orchestration, Multimodality, and Output Control

The behavior of the assistant is driven by a prompt orchestration strategy integrating several components: system instructions, documentary context derived from the RAG, conversation history, and optional multimodal inputs. This structuring aligns with recent work showing that explicit prompt design improves reasoning quality and reduces unfounded generation errors (Wei et al., 2022).

A fixed system instruction explicitly defines the pedagogical role of the assistant, enforces linguistic consistency, and specifies strict formatting rules, particularly for mathematical formulas and code blocks. When a code file is present, it is included in the prompt and treated as a central reference for the analysis and explanations provided.

In the case of multimodal queries, the uploaded images are encoded and attached to the request to enable joint text–image reasoning, in accordance with work on multimodal evaluation and reasoning in large-scale models (Antol et al., 2015; Hessel et al., 2021; Yu et al., 2023). The generation parameters are deliberately conservative, notably using a low temperature, to limit degeneration of the generated text and to improve factual stability and reproducibility of the responses (Holtzman et al., 2020).

The responses are produced in enriched Markdown, compatible with LaTeX rendering of mathematical expressions and subsequent export of conversations. Finally, information related to the retrieved context, the modalities used, and the dialogue history is logged, providing the elements necessary for the quantitative performance evaluation presented in Section 5.

Figure 4

HTML Preview of the Report Generated by the User, Using Markdown and LaTeX Formatting



Scoring Method

Scientific Factuality Score and RAG Precision

Let us consider a panel of n numbered queries. For a query i , we denote by T_i the expected rigor corresponding to the degree of expectation assigned to the assistant, and by $\{T_i\}_{i \in [1, n]}$ the associated family. We proceed similarly for the estimated values of the assistant's responses, which we denote by M_i and $\{M_i\}_{i \in [1, n]}$ the associated family. Each of these T_i , M_i given as a percentage, is defined on a scale from 0 to 100%, with 1 if the assistant answers the question rigorously and completely.

Thus, one can deduce an average divergence Δ of scientific factuality given by the formula:

$$\Delta = \frac{1}{n} \sum_{i \in [1, n]} |T_i - M_i| \quad (1)$$

and an associated score S given by $S = 1 - \Delta$.

For a RAG precision score, a simple formula is used by assigning each T_i to the number of sources expected/that should be retrieved for a query i , and M_i to the number of sources retrieved among the expected sources for this same query i . Note that in the case of omission, if a source is retrieved that was not anticipated in T_i , if it is coherent, it will trigger a +1 for both T_i and M_i , namely:

$$\Delta = \frac{1}{n} \sum_{i \in [1, n]} \frac{M_i}{T_i} \quad (2)$$

Note that this divergence formula only makes sense if there exists a document dealing with the question. It should be noted that this score does not consider the ranking of the retrieved information, which is a minor issue if the top-k retrieved items are sufficient on their own to produce a complete answer.

Multimodal Consistency Score

The objective of this scoring is to evaluate the internal consistency of the reasoning of a scientific assistant when it relies on heterogeneous sources (text, code, documents). By comparing the quality profiles obtained from responses conditioned on a single source, the score measures whether these sources support a compatible scientific interpretation or produce divergent reasoning.

It does not aim to judge the absolute correctness of a response, but to quantify the structural alignment of the contributions of each source to the overall reasoning.

We therefore explain the method for obtaining this score. In a first step, a query q is submitted to the assistant with a particular prompting strategy; that is, the same question is asked, but the assistant is instructed to respond using only one type of source at a time (either PDF documents, text, or code).

Table 1

Adapted Prompting

Prompt 1 – Text only
Answer the following question using only text excerpts from the provided TXT documents. You may use an unlimited number of words, as long as the reasoning is, in your judgment, sufficiently thorough and complete. If you cannot find information that you consider coherent and relevant to the query, return “not retrieved or insufficient context”. Context: “TEXT documents”.
Prompt 2 – PDF only
Answer the following question using only text excerpts from the provided PDF documents. You may use an unlimited number of words, as long as the reasoning is, in your judgment, sufficiently thorough and complete. If you cannot find information that you consider coherent and relevant to the query, return “not retrieved or insufficient context”. Context: “PDF documents”.

Prompt 3 – Code only
Answer the following question using only text excerpts from the provided Code documents. You may use an unlimited number of words, as long as the reasoning is, in your judgment, sufficiently thorough and complete. If you cannot find information that you consider coherent and relevant to the query, return “not retrieved or insufficient context”. Context: “.py, .m, .cs, .java, .c, .h documents”.

From these generated responses, one can assess (either manually or via an LLM-judge) a certain score according to a few common criteria B (e.g., scientific clarity, logical coherence, absence of hallucinations, structuring, possible numerical applications, etc.), such that for a query q , we obtain the score of modality m , denoted $s_{q,m}$ composed of:

$$s_{q,m} = (s_{q,m}(1), \dots, s_{q,m}(B)) \text{ with } s_{q,m} \in [0, 1]^B \quad (3)$$

To this per-query score, one can associate a statistical distribution $L_{q,m}$, where the score vector is interpreted as an empirical distribution over the criteria:

$$L_{q,m} = (L_{q,m}(1), \dots, L_{q,m}(B)) \text{ with } L_{q,m} \in [0, 1]^B \text{ such as } L_{q,m}(b) = \frac{s_{q,m}(b) + \varepsilon}{\sum_{b \in [0, B]} s_{q,m}(b) + \varepsilon}$$

$$\varepsilon = 10^{-12} \text{ (Holtzman et al., 2020), } b \in [1, B] \quad (4)$$

Through the normalization imposed by the distribution, information related to exactness is lost; nevertheless, this is not problematic since the goal is to obtain a score related to the coherence between several statistical distributions. There currently exist several functions for measuring coherence between multiple distributions, including the Jensen–Shannon divergence, which in our case allows comparison of the relative structure of reasoning qualities induced by each source involved.

The objective is to compare statistical distributions associated with different modalities, each distribution representing the way a given source supports reasoning according to several criteria. Since there is no natural reference distribution nor dominant modality, the comparison must be symmetrical and independent of the order of the sources. As divergence between distributions is defined for two distributions at a time, multimodal comparison naturally reduces to the study of all binary comparisons between modalities. The triangular sum then aggregates these pairwise divergences exhaustively and non-redundantly, providing a global measure of the intermodal coherence of the reasoning.

Ultimately, one first computes the mean of the pairwise Jensen–Shannon divergences between modalities for a given query and then averages these divergences over the number of queries performed. Considering Q as the number of queries asked and M as the number of modalities, the expression is:

$$\Delta = \frac{2}{M(M-1)Q} \sum_{q \in [1, Q]} \sum_{1 \leq i < j \leq M} D_{JS}^*(L_{q,i} \parallel L_{q,j}) \quad (5)$$

Where D_{JS}^* denotes the normalized Jensen–Shannon divergence function.

In terms of algorithmic complexity, we have $O(M^2Q)$. For a limited number of modalities (e.g., three modalities: code, text, and PDF documents), the computation of the sum can be performed quickly.

An associated score is defined using the previous formula $S = 1 - \Delta$.

Multimodal Performance Indicator

From these scores, one can define a global performance indicator measuring how coherent and reliable a scientifically oriented AI assistant could be. We perform an aggregation of our scores. This represents a structural measure of trust.

Let C denote the set of score components (RAG, Multimodal Consistency Score, scientific scores, latency, etc.).

We formalize this indicator by computing a weighted sum of the scores with carefully chosen positive weights w_c , whose sum equals 1. Thus, denoting the indicator by IP, we have:

$$IP = \sum_{c \in C} w_c S_c \quad (6)$$

The experimenter has full freedom in choosing the weights w_c . We propose, for each weight, the use of the softmax formula applied to the divergences associated with the score, so that for each weight we have:

$$w_c = \frac{\exp(-\lambda \Delta_c)}{\sum_{j \in C} \exp(-\lambda \Delta_j)} \quad (7)$$

with λ the sensitivity parameter, which we define as:

$$\lambda = \frac{1}{\text{med}(\{\Delta_j\}_{j \in C}) + \varepsilon} \quad (8)$$

The sensitivity parameter allows us to adjust the weights according to our tolerance for errors. This expression for λ is self-adaptive, and the inverse of the median is a robust estimator of the scale of the divergences, allowing the parameter to be normalized by the natural scale of the observed phenomenon (Huber, 1981).

Nevertheless, if the experimenter wishes to discriminate errors by penalizing the weights of scores whose divergence is maximal, they can choose this λ .

Thus, assuming that each score can be written in the form $S_c = 1 - \Delta_c$, the general form of the performance indicator is:

$$IP = \sum_{c \in C} (1 - \Delta_c) \left(\frac{\exp(-\lambda \Delta_c)}{\sum_{j \in C} \exp(-\lambda \Delta_j)} \right) \quad (9)$$

Results

Evaluation Method

In the first step, we are going to calculate standard metrics of precision, recall, and ranking scores to globally assess the performance of the assistant with and without RAG, then compare its added value in the modeling by using the proposed scoring method.

RAG Impact

To evaluate the performance of our AI assistant, we will first carry out an initial measurement of the effectiveness of RAG by computing the standard metrics Mean Reciprocal Rank (MRR), Global Precision (P), Global Recall (R), and top-6 Precision/Recall (P@6/R@6) on a panel of 2,000 queries covering the following scientific topics (thermodynamics, mechanism dynamics, electronics, mathematics, computer programming, optics) (Manning et al., 2008).

Table 2

Evaluation Metric Values

Metrics	MRR	P	R	P@6	R@6
RAG	0,756	0,772	0,827	0,723	0,781
NO RAG	/	0,689	0,694	/	/

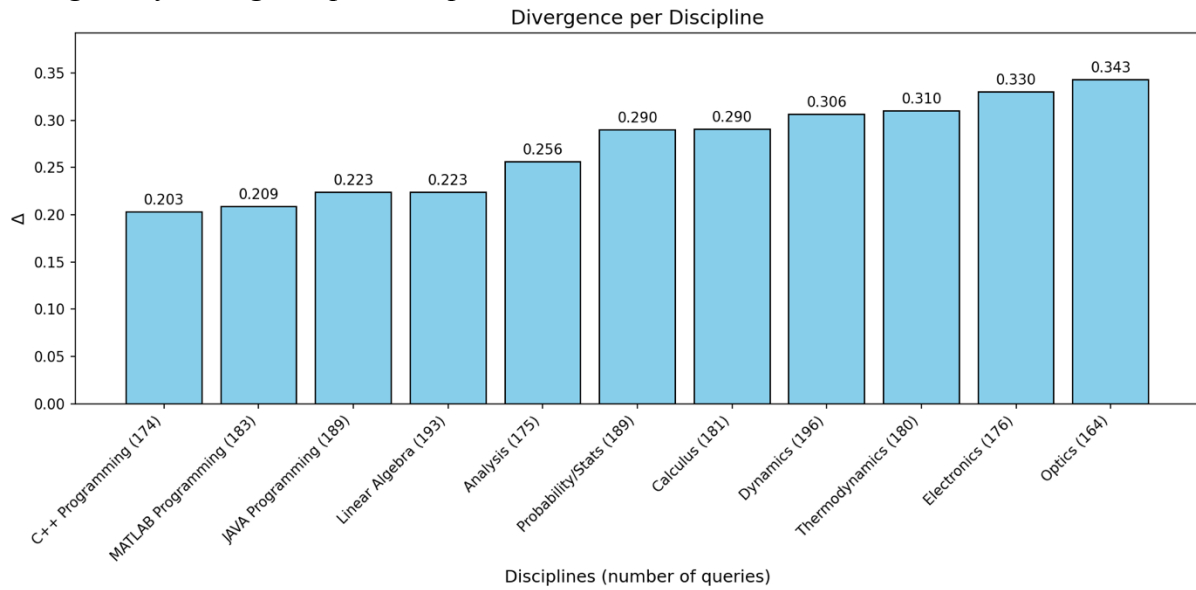
The model achieves an R@6 of 0.781 and a P@6 of 0.723, indicating that most relevant items are retrieved among the top six results while maintaining a high level of precision. Combined with an MRR of 0.756, these performances show that relevant items are generally ranked in the very first position or very close to the top of the list, reflecting the model's effective ability to correctly order results according to their relevance. It is also observed that the RAG system increases precision by 9% and global recall by 12%, which implies a performance improvement of 10.5% according to the F1-score. The RAG system is effective. MRR and Recall are not defined in the NO-RAG setting, as no retrieval mechanism is involved.

Scoring Method

For our scoring method, we will establish a scientific factuality score for each of the considered disciplines. We will also consider our overall RAG score. For the modality score, we will consider the provided text, code, images, and PDF documents. Next, we will compute the associated parameters and the corresponding performance indicator (IP).

We plot the histogram of the divergences Δ for the tested academic subjects, using the number of queries derived from university-level exercises.

Figure 5
Histogram of Divergence per Discipline



We observe that the most satisfactory responses correspond to programming, as they minimize the divergence. This may be due to several reasons, including the choice of the embedding model, which ignores image files that are more prevalent in physics courses. The assistant struggles more with the physical sciences, which is consistent with the experimenter's observations. These values are averaged to facilitate the analysis.

On the same set of 2,000 queries, the RAG system made it possible to retrieve, on average, 74.7% of the expected sources. For the multimodality score, we considered four criteria: scientific clarity, logical coherence, absence of hallucinations, and correctness of numerical applications. For this purpose, we applied the prompting shown in Fig. 5 to obtain five responses per query (code, text, PDF, image, and intermodal response) for the joint evaluation of Δ_S and Δ_{RAG}).

Table 3
Study Values for the Performance Indicator

	M	Q	Δ_S	Δ_{RAG}	Δ_M
Values	4	2000	0,271	0,253	0,324

We obtain a modality divergence of 0.324. This means that the AI assistant can synchronize the different sources to promote a structured and coherent response across these sources with an alignment of 67.6%. The reasoning is generally well aligned.

Table 4
Performance Indicator Measurements

	Δ_S	Δ_{RAG}	Δ_M	λ	w_S	w_{RAG}	w_M	IP
Values	0,271	0,253	0,324	3,690	0,346	0,370	0,284	0,720

Our scoring method evaluates our RAG-based AI assistant with a score of 7.2/10, which at first glance is relatively satisfactory. The model used may (or may not) discriminate strongly between high values, which can influence the final score by diverging from the original intent

of the scoring scheme. We observe that the model assigns a relatively balanced importance to all scores.

Conclusion

This work shows that integrating RAG clearly improves the reliability of an educational assistant by reducing hallucinations and increasing the accuracy of responses. The proposed architecture functions effectively and remains adaptable to different types of data.

The scoring method provides a more relevant evaluation, particularly regarding multimodal coherence, although it remains imperfect and strongly dependent on the experimenter. The results confirm a real performance gain, but also highlight limitations on certain types of content, especially in physics or when dealing with visual data.

RAG therefore appears to be an effective solution, but one that strongly depends on the quality of the sources and the technical choices made.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

The authors declare that Grammarly, an AI-assisted writing software, was used in proofreading and refining the language used in the manuscript. The usage was limited to correcting grammatical and spelling errors and rephrasing statements for accuracy and clarity. The authors further declare that, apart from Grammarly, no other AI or AI-assisted technologies have been used to generate content in writing the manuscript. The ideas, design, procedures, findings, analyses, and discussion are originally written and derived from careful and systematic conduct of the research.

References

- Abo El Enen, M., Saad, S., & Nazmy, T. (2025). A survey on retrieval-augmented generation (RAG) models for healthcare applications. *Neural Computing and Applications*, 37(33), 28191–28267. <https://doi.org/10.1007/s00521-025-11666-9>
- Antol, A., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2425–2433). <https://doi.org/10.1109/ICCV.2015.279>
- Bednarz, B., & Milosz, M. (2025). Benchmarking the performance of Python web frameworks. *Journal of Computer Sciences Institute*, 36, 336–341. <https://doi.org/10.35784/jcsi.7738>
- Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., & Hjelm, R. D. (2018). Mutual information neural estimation. In *Proceedings of the International Conference on Machine Learning*, (Vol. 80, pp. 531–540). PMLR.
- Brown, A., Roman, M., & Devereux, B. (2025). *A systematic literature review of retrieval-augmented generation: Techniques, metrics, and challenges*. arXiv. <https://arxiv.org/abs/2508.06401>
- Chen, J. (2023). Model algorithm research based on Python FastAPI. *Frontiers in Science and Engineering*, 3(9), 7–10. <https://doi.org/10.54691/fse.v3i9.5591>
- Clark, P., Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., & Kalyan, A. (2022). Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, 35. <https://arxiv.org/abs/2209.09513>
- Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). *Retrieval-augmented generation for large language models: A survey*. arXiv. <https://arxiv.org/abs/2312.10997>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Haque, I. C. R. (2025). Implementation of RAG and LLM for a document-based chatbot system. *Journal of Electronics Technology Exploration*, 3(1), 19–23. <https://doi.org/10.52465/joetex.v3i1.588>

- Hessel, J., Holtzman, A., Forbes, Y., Bras, R., & Choi, Y. (2021). CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 7514–7528). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.emnlp-main.595>
- Holmes, W., Bialik, M., & Fadel, C. (2019). *Artificial intelligence in education: Promises and implications for teaching and learning*. OECD Publishing.
- Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations*. <https://arxiv.org/abs/1904.09751>
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), 1–38. <https://doi.org/10.1145/3571730>
- Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547. <https://doi.org/10.1109/TBDDATA.2019.2921572>
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6769–6781). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.550>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 33 (pp. 9459–9474).
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1906–1919). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.173>
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 8748–8763). PMLR.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 3982–3992). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D19-1410>

- Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
<https://doi.org/10.1561/15000000019>
- Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep self-attention distillation for labeled and unlabeled pre-trained language models. *Advances in Neural Information Processing Systems*, 33, 12941–12952.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35.
- Yu, F., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., & Wang, L. (2023). *MM-Vet: Evaluating large multimodal models for integrated capabilities*. arXiv.
<https://arxiv.org/abs/2308.02490>