

Advanced Semantic Analysis of Research Papers Using a Retrieval-Augmented Architecture

Alexandru MOȚĂȚĂIANU

National University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

*Corresponding author, amotataianu@stud.faima.upb.ro

Ionel-Bujorel PĂVĂLOIU

National University of Science and Technology POLITEHNICA Bucharest, Bucharest, Romania

bujor.pavaloiu@upb.ro

Abstract. *The exponentially increased number of academic papers published lately is raising significant challenges for researchers, requiring new ways for reviewing massive amounts of information, like automated systems capable of analysing the researched content.*

Large Language Models (LLMs) represent an answer to this need, nevertheless, due to the hallucinations, while the research requires rigorous information, a large-scale adoption in this field remains a challenge.

Current scientific literature mentions Retrieval-Augmented Generation (RAG) as a solution (Upadhyay & Viviani, 2025) for LLMs hallucinations, nevertheless (Godinez, 2025) mentions also some difficulties these systems have in providing citations.

This paper addresses these gaps, proposing a scalable RAG architecture built on top of Azure services (Azure Blob Storage, AI Search, OpenAI) that leverages embeddings and multidimensional vector spaces for an advanced semantic analysis, extended with automation tools for accessing curated data (Meacham & Sharafzad, 2025) at the beginning of the flow and for consuming the synthetic answers during the final validations.

Using a progressive methodology, the study confirms the quality of the returned answers gradually, testing different LLM components and scenarios for obtaining the most accurate responses and providing citations for an increased reliability and trustiness of the overall architecture.

Furthermore, the proposed framework prioritizes cost efficiency by using accessible services that are sufficient for supporting the literature review automation while decreasing the articles synthesis process from hours to minutes (Godinez, 2025)

Keywords: Retrieval-Augmented Generation (RAG), Semantic Search, Contextual Embeddings, Vector Spaces.

Introduction

In the past years a consistent increase in the volume of academics' papers disseminated via main publishing journals was remarked, and especially on the emerging technologies fields, creating a need of automated literature analysis (Godinez, 2025).

This is the first hypothesis investigated in the current study, starting with a quantitative analysis of the research articles published on the MDPI portal (<https://www.mdpi.com/>), on the "Remote sensing" journal, during a period of 17 years, between 2009 and 2025 (Figure 1. Number of the research papers published).

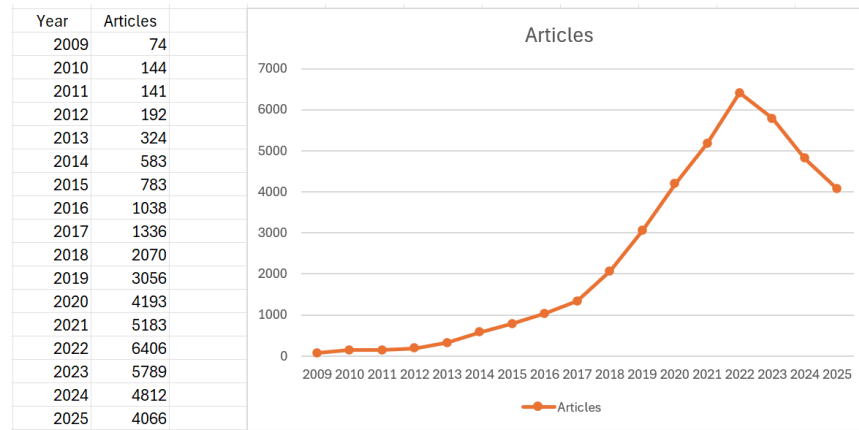


Figure 1. Number of the research papers published on Remote sensing journal

Source: Authors' own research.

If 2009-2013 is characterized mostly by a flat – plateau shape, potentially also linked with the ramp-up activity on the MDPI platform (<https://www.mdpi.com/>) itself, counting only couple of hundreds of published articles, between 2014-2017 a consistent trend upwards can be remarked, with number touching more than 1000 articles. While 2018-2022 represents an abrupt slope that goes beyond 6000 articles/year, the last 3 years depicts a cool-down in this trend, consolidating to a value of over 4000 articles on yearly basis, into the last 5 years frame.

As highlighted visually by the analysis above, the number of published articles has increased dramatically, to an average of 10 to 20 published articles on daily basis in the past 5 years, requiring a similarly increased effort and time for researchers in finding and assessing the relevant content, into a sea of knowledge.

In this context, it is more than obvious that finding more efficient methods of ingesting and analysing these high-volumes of information (Karras, et al., 2025) by the academic community is more than a need, and proposing new research topics, into a higher pace research environment requires new approaches for identifying a fast moving knowledge frontier.

Even if Retrieval-Augmented Generation (RAG) solutions became widely available in different domains, the solution proposed in this paper, relying on the quality of the data (Meacham & Sharafzad, 2025), is following the latest generation concepts (like vector spaces and generative AI), blending with cloud, crawling services and automation tools for supporting an extensive semantic grounding and retrieval (Karras, et al., 2025) of doctoral articles.

There was preferred an architecture based on Azure services (Storage, Cognitive Search, OpenAi, Compute) due to their proved performance in different critical areas, as: reliability of the services, integration between components and with other external tools, but also to high standards related to the data security and privacy.

Literature review

The increased volume of published academic information lately is another dimension added to the already established challenges in integrating AI systems for research, as trust and reliability (Meacham & Sharafzad, 2025).

The solution found during the literature review to this context, is the usage of a RAG solution (Upadhyay & Viviani, 2025), nevertheless despite the advancement of the domain, there

could be found also evidences of RAG systems having difficulties in providing reliable citations for a scientific rigor (Godinez, 2025). It is also revealed the need to ground the responses generated, to curated references, or clean sets of data, to avoid the risk of hallucinations (Meacham & Sharafzad, 2025).

Another direction highlighted in the studied papers is that rigorous testing protocols must be further developed, for confirming the reliability of the RAG systems, for automatically checking the correctness of the responses (Godinez, 2025).

The contribution of this study is represented by the design of an end to end architecture (from data acquisition to aggregated responses), integrating mature software tools as Azure AI Search (for embeddings and vector spaces) and OpenAI for language processing, with complementary custom Python tools developed for automating and scaling up the entire process, plus creation of a progressive methodology for testing the results, achieving a solution that responds to the need of traceability and reducing hallucinations (Meacham & Sharafzad, 2025) in research domain.

There was chosen an Azure architecture, using tools as Azure AI Search and Azure OpenAi, since these tools offers natively features as AI Enrichment, document cracking, data chunking, while the guided Large Language Models (LLM) (Upadhyay & Viviani, 2025) returns next to the responses, the references/citations, minimizing the risk of hallucinations. Adding, the support for Romanian language, it crafts a solution scalable to virtually any volume of raw data.

Methodology

It is well known that keyword searching has represented for decades the main pillar in finding fast and precise information (Meacham & Sharafzad, 2025), but in the current context, it seems to be not sufficient anymore while facing the new volumes of information, in parallel with the evolution of the natural language used by authors in these newly released papers. While keyword searching applies on characters and strings level, the need today is to understand the sense, semantics, or intention. More, when using keywords, synonyms and words with multiple senses were often missed by search, while semantic comes as a solution to these problems.

Semantic search represents a new paradigm, that overpasses the keyword search limitation described above, having the ability to understand the meaning of the words and phrases.

The architectural ground of this technology sits on contextual embeddings, and multi-dimensional vector spaces. Chunks of information are transposed as vectors into a mathematical space, where their coordinates are being characterized by semantic properties, while retrieval mechanism relies on algorithms like cosine similarity for computing and identifying the relevant information.

Generative AI complements it perfectly, with capabilities to create probabilistic synthetic information based on its training data, or, as in this study, curated sources of information.

Vectors embeddings (Figure 2. Embedding process) describe a revolutionary process where each piece of information (chunk) gets multidimensional coordinates, each dimension representing a specific characteristic. Once transformed into a vector format, the information can be easily positioned into virtual space, where the distance between vectors can be easily computed.



Figure 2. Embedding process

Source: Authors' own research.

The distance between vectors is computed with algorithms, as cosine similarity; for example, when the chunks of information describe similar notions, the computed cosine value tends to 1, and in other cases, the value of cosine describes how far the terms are one from the other in terms of meaning (Upadhyay & Viviani, 2025).

A simpler parallel, Figure 3. A parallel between vector space and a 3-dimensional RGB space, can be drawn with a RGB, 3-dimensional space, where the named colors can be grouped together, based on the vectors distances: *Magenta*: RGB(255, 0, 255) and *Fuchsia*: RGB(255, 0, 255), even if they have different naming, they are described by perfectly aligned vectors, or *Cyan*: RGB(0, 255, 255) and *Turquoise*: RGB(64, 224, 208), again, they are described by different words, but placed closely in the RGB vectorial space.

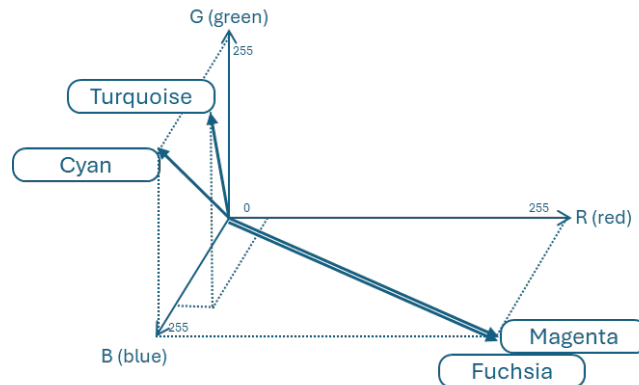


Figure 3. A parallel between vector space and a 3-dimensional RGB space

Source: Authors' own research.

Modern vector embeddings can reach over 1000+ dimensions, while the principle remains the same, turning data various chunks of: text, images, audio, video etc. in strings of values that describes their meaning.

In the proposed solution, the creation of the vector embeddings, and indexation is being orchestrated with Cognitive Search / Azure AI Search, via an *indexer*, that transforms the raw data into searchable, easy to retrieve items into an index.

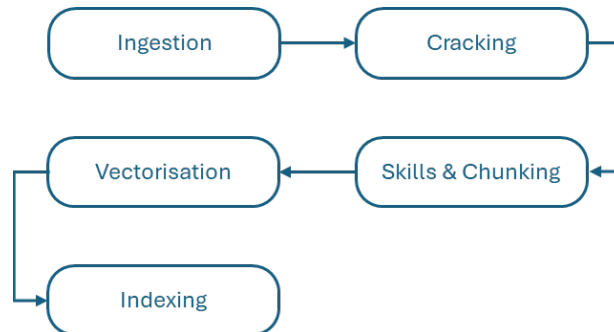


Figure 4. Azure AI/Cognitive search orchestration steps

Source: Authors' own research.

Initially it scans the data source, enumerating the files that are “new”, or “modified” since the last scanning iteration, optimizing this way the resources consumption for this objective. Once a file is identified as being updated, a pull operation starts, gathering updated files from structured

(ex. Azure SQL Database, Cosmos DB, Azure Table Storage) or unstructured (ex. Azure Blob Storage) data sources.

Next, during “documents cracking” it extracts all relevant information/metadata as: filename, path, size last-modified date, essential for later processing, and it opens the file, supporting a consistent list of formats (ex. CSV, PDF, HTML, JSON, XML etc.).

The AI enrichment phase follows, bringing additional information in the final index, using out of the box features provided by Azure as: Entity recognition, Sentiment analysis, Translation etc. next custom skills that can be created and integrated in the flow, using tools like Azure Functions or Machine learning.

The information is further split/chunked into smaller pieces, optimized for AI models, and the embedding tool is being called, in this case text-embedding-3-small, for transforming the chunks into vectors, and place all of them into a multidimensional, vectorial space, creating finally the index that will be further used for queries.

The index remains unchanged, until the indexer runs again, operation that can be either triggered programmatically (ex. update was performed on Azure Blob Storage, so new content needs to be indexed), or scheduled to run on specific time intervals.

The solution architecture used in this study aggregates: crawling, cloud storage, embedding, indexing, LLM, and automation services, as depicted in the Figure 5. RAG architecture. By utilizing a RAG solution, two main LLM related risks are being mitigated, as: *hallucinations*, when the model generates content unlinked to the reality (Upadhyay & Viviani, 2025), or *no access to private/fresh data*, when the LLM model relies when building the answer only on the trained dataset, that has a specific timestamp (in the past), while the content/context has evolved meanwhile (Upadhyay & Viviani, 2025)

Into the proposed RAG architecture, the LLM model is fed with curated source of information, enhancing the reliability and trustworthiness (Meacham & Sharafzad, 2025) of the output provided to end user.

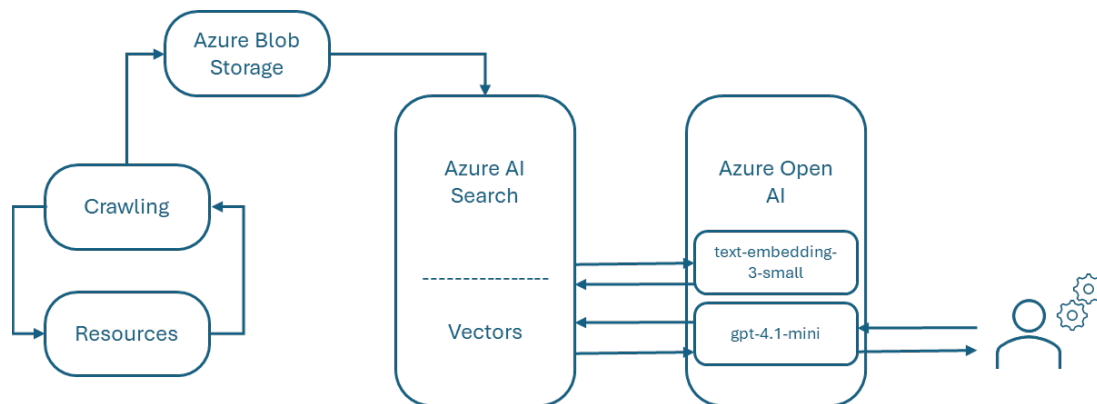


Figure 5. RAG architecture

Source: Authors' own research.

During the data acquisition phase, the crawling service gathers the raw information and stores it, into the Azure Blob Storage. Next the Azure AI Search (former Cognitive Search) ingests the information, enriches it and calling Azure OpenAi embeddings transforms the data in vectors, that are further indexed.

At this stage, when users enter questions into OpenAi chat, these questions are transformed also into vectors for mapping them in the vector space; they are mathematically compared with the content (vectors) already indexed. Through this mechanism the system identifies the most semantically relevant fragments/chunks by calculating the distances between vector with cosine similarity, while the LLM, using its language capabilities build coherent responses based on these fragments, and replies to the users.

For transforming the LLM text generator into a precise analysis tool, a methodology for avoiding hallucinations is mandatory. In this sense, feeding curated data via Azure AI Search to an LLM is the first step. The LLM model will receive a specific context, a specific prompt and it will respond only with information from the private data to queries. It will also be instructed to return citations linked to the response, pointing to the places in the original documents where the info was taken, a decisive factor for raising the trust on the solution (Upadhyay & Viviani, 2025).

Instead of relying only on the original set of data, the model was trained with months or even years ago, the model receives fresh and specific information, like scientific articles in our case, that represents an anchor to the reality, a clean description of the frontier of knowledge the researcher is trying to assess.

Without this fresh information, the model will rely only on its initial ingested and trained information, that might be obsolete or deprecated, a source of potential misinformation the LLM might create.

The knowledge base used into this study is made from articles published on MDPI (<https://www.mdpi.com/>) journal in 2025, on the “Remote sensing” journal. During the data acquisition phase, each published volume was parsed, searching for the “UAV” keyword, and the content was saved as PDF files into an Azure Blob Storage container.

With the Azure AI Search tool, the data was ingested, cracked, enriched and further chunked, meaning it was fragmented in smaller pieces of information preparing it for embedding.

In Azure OpenAI, there were configured two deployments as: *gpt-4.1-nano* model or *gpt-4.1-mini*, for the interface/chat with the end user / automation tool, and *text-embedding-3-small*, an efficient and cost-effective model released by OpenAi, that transforms text into vectors, up to 1536 token dimensions, having support for Romanian language also, used by Azure AI Search for embeddings

The standard Temperature value of 0.7, was decreased to 0.1, since the intention of the solution is to deliver very precise answers, based only on the information indexed into Azure AI. In the same direction, Top P value used was 0.95, so only high probability tokens will be considered when building the response. Here, also the chat Assistant was instructed to provide answers only from the custom added information, via the Azure AI Search functionality.

The implementation strategy has 5 incremental steps, meant to evaluate the solution progressively, from small functional tests to a large-scale volume testing.

In *experiment #1* it was used only one technical article, loaded in the solution, chunked, vectorized and indexed. There was applied a set of 10 questions, 5 questions in English and 5 questions in Romanian, meant to verify a clearly identified information inside the article, with the main purpose of making an initial assessment of the solution and validating it from end-to-end perspective.

Since the solution relies on RAG architecture and the answer will be retrieved from the data source provided and not from the LLM learned knowledge, *gpt-4.1-nano*, an optimized and light model recommended for speed and cost efficiency was preferred first, see Table 1. Experiment #1 setup.

There were evaluated how clean the generated answers against the hallucinations were, the presence of citations and how well were the answers matching the questions intent.

Table 1. Experiment #1 setup

Experiment	Number of papers	Questions number	Questions nature	Questions language	Model used
# 1	1	10	Manually generated	5 Romanian 5 English	gpt-4.1-nano

Source: Authors' own research.

In *experiment #2* due to the observations/results of experiment #1, the gpt-4.1-nano was changed with gpt-4.1-mini, another cost-effective model, with higher reasoning capabilities than the previous one, see Table 2. Experiment #2 setup. The hypothesis was later confirmed, since on the same set of questions, it returned higher figures on the evaluation performed.

Table 2. Experiment #2 setup

Experiment	Number of papers	Questions number	Questions nature	Questions language	Model used
# 2	1	10	Manually generated	5 Romanian 5 English	gpt-4.1-mini

Source: Authors' own research.

In *experiment #3*, the complexity was slightly increased, see Table 3. Experiment #3 setup. Another article was added to the storage, so it was indexed information from 2 articles. Testing was shifted from punctual questions to cross-document synthesis questions, evaluating the capacity of the solution to provide responses, and citations from multiple sources. As model, gpt-4.1-mini was kept, since it has proved previously higher reasoning capabilities. A new set of 10 questions was created, this time, all of them being written in English.

Table 3. Experiment #3 setup

Experiment	Number of papers	Questions number	Questions nature	Questions language	Model used
# 3	2	10	Manually generated	English	gpt-4.1-mini

Source: Authors' own research.

In *experiment #4*, the architecture was enhanced through: procedures for generating lists of questions on one hand, and automated tools built for processing these questions on the other hand, scaling this way the capabilities confirmed in the previous experiments, and transforming the solution from a local prototype to a global research tool able to manage potentially any number of academic articles, so researchers worldwide can interact with large volumes of information in shorter time than ever before, see Table 4. Experiment #4 setup.

This enhancement had two main phases:

A specialized prompt was created and used to ask Azure OpenAI Assistant, running gpt-4.1-mini to generate a list of derived questions, based on the set of articles ingested, plus instructions received via the prompt. The process is conducted by strict prompt constraints, for obtaining a list of relevant, academic, strictly anchored into the article's dataset questions, while the number of questions is a configurable variable proportional with the size of the article's dataset.

A custom software tool, developed in Python, was created for automation the process of asking the Azure OpenAi Assistant multiple questions, handling also the responses collection. This new tool links to Azure OpenAi endpoint, for processing the previously created list of questions, line by line. After awaiting the AI Assistant responses, it aggregates the answers into a structured

output file, for a facile human analysis. For troubleshooting purposes, Python module was designed with an error handler logic, saving the error codes in the output document, if needed, allowing this way the identification of various issues that might appear during the runtime.

The responses were evaluated individually, using the model Gemini 1.5 Pro, a 2-Million tokens LLM, specialized in analysis of massive amounts of information.

Table 4. Experiment #4 setup

Experiment	Number of papers	Questions number	Questions nature	Questions language	Model used
# 4	2	10	AI generated	English	gpt-4.1-mini

Source: Authors' own research.

To further evaluate the solution, the *experiment #5*, was designed as the one to confirm the solution capabilities on higher volume data, a stress test for the entire proposed RAG architecture, see Table 5. Experiment #5 setup.

The knowledge base this time is made of 30 relevant, for the remote sensing and UAV research, recent published articles, having a PDF format, with various sizes from 773KB to 15MB, totalling 230 MB, having more than 25 pages each, being hosted on Azure Blob Storage.

Similar to experiment #4, a specialized prompt is being used to generate a set of 10 relevant questions, focused on cross-document synthesis, in English, starting from a precise set of instructions, based on the 30 articles available. The model used was gpt-4.1-mini.

The previously created Python automation tool is used next to parse the set of questions, calling the Azure OpenAi endpoint, awaiting the responses and references, and merging them into the output file.

The validation, using a specialized prompt, on each question&answer pair is performed using NotebookLM (<https://notebooklm.google.com/>) running Gemini 1.5 Pro.

Table 5. Experiment #5 setup

Experiment	Number of papers	Questions number	Questions nature	Questions language	Model used
# 5	30	10	AI generated	English	gpt-4.1-mini

Source: Authors' own research.

Results and discussions

The methodology results table is depicted below in Table 6. Study results:

Table 6. Study results

Experiment	Questions	Romanian answers	English answers	Validated by
# 1	Manually generated	76%	80%	Researcher
# 2	Manually generated	80%	100%	Researcher
# 3	Manually generated	-	85%	Researcher
# 4	AI generated	-	94%	Gemini 1.5 Pro
# 5	AI generated	-	81%	Gemini 1.5 Pro

Source: Authors' own research.

Based on the observations made on experiment #1 and experiment #2 it was concluded that on both cases, even if the Romanian capabilities of the LLM were present, the Romanian questions

were more troublesome than the English ones, independently of the models used: gpt-4.1-nano or gpt-4.1-mini. For confirming this observation gpt-4.1-mini was asked what its Romanian language is compared with English, and the model confirmed a 90-95% level.

The solution results on experiment #3, English questions/answers, are slightly lower than the ones of previous steps, while using the same gpt-4.1-mini model, mostly because of the increased complexity of the synthesis questions. Nevertheless, the numbers are high, confirming the architecture strength even when the solution needs to synthesize the information from multiple sources for answering a single question.

Experiments #4 and #5 come as natural answer to the need of scaling the solution, developing methods (like prompt engineering) and automation Python tools to achieve this goal and represent the first steps towards a rigorous testing framework. The numbers slightly lower on experiment #5 are due to the increased high information complexity, the high number of tokens (1.05 millions) the solution had to manage and to the limitations of the tools used for testing.

This progressive approach proves that the solution using RAG architecture can deliver very high-quality answers on a scalable data source, while assuring the transparency by the references returned on each result.

All test performed shows clearly the potential of such a solution for synthesizing information and effectively supporting researchers in identifying valuable information to be further assessed in their studies, while the presence of citations represents the mean to increase the level of trustiness in the proposed architecture.

According to (Brysbart, 2019) the average reading speed is 238 words per minute for non-fictional content, value that can be impacted by the complexity of the article, language used, reader experience, etc. On top of it can be added the time needed for taking notes, evaluating and linking ideas exposed, doing data analysis, reviewing citations, fatigue etc. so the estimated duration for a single article synthesis can go up to a couple of hours.

Multiplying this value with 30 articles (our experiment #5 dataset size), the overall estimated effort for manually analysis can reach more than 50 hours, while the time needed to obtain the responses on the experiment #5 was 1 minute and 24 seconds.

Conclusion

The solution proposed in this study embraces the latest technologies like Azure AI Search and OpenAI, transforming them into effective research and productivity tools, that shrink the research work from hours to minutes (Godinez, 2025).

Using curated input data, transforming and indexing it into a vector space, using LLMs to interact with end user, craft answers, and provide citations, represents the mean to validate the solution against hallucinations, and misinformation.

Facing an increased volumes of published research articles today, the utilization of AI tools for extracting the relevant meaning from these articles represents a natural response and alignment to the latest technologies trends also in the researcher's area. For staying relevant on the field, the researchers should be able to process more information and deliver much faster high-quality studies and articles to the community, than ever before. Utilizing AI tools in doctoral research is not a future trend, it is more than actual, since researchers should stay on the knowledge frontier border, that is moving fast today, while constantly tackling the unknow that is beyond it.

Starting from the solution presented in this paper, which validates the RAG architecture using Azure services, several directions can be foreseen, as optimizing of the solution performance across diverse datasets, or automated evaluation frameworks to ensure the quality level on scale.

The proposed architecture in this paper grounds a solid foundation for integrating RAG tools in academic research extensively, and future works should prioritize the design of sophisticated testing ways to reconfirm the reliability of the proposed solution.

An evolution that can be easily anticipated today, and remains open for further analysis, is the usage of Agentic AIs, or independent AI agents (Kostopoulos, Gkamas, Rigou, & Kotsiantis, 2025) that will be able to perform end-to-end crawling and RAG analysis. After receiving goals from the user, they will be independently orchestrating all steps, logic and processing steps described in this paper, taking the necessary means to achieve the goal. Agentic AIs will most probably face several challenges such as navigation drifts or hallucinations, increased costs, or security concerns. This will raise a very delicate legal and ethical question regarding how to legitimate them.

From the challenges perspective, it is well known that all cloud services come with an associated cost, depending on the subscription level, type of service, overall forecasted consume, location, performance, redundancy etc. The solution proposed in this article is built mainly on Azure services, where the cost of Azure Ai Search, Azure Open Ai, Azure Blob Storage (for data storage), Compute (for crawlers and automation), represents the main operational costs to be considered, and correctly assessed before implementation of the solution. Each of them can be fine-tuned to reflect the end user needs, balancing the performance required versus the available budget.

Other important point of attention for adopting a solution like the one proposed in this article, is mainly related to ethics, and the assurance that data remains private (Vishesh & Astha, 2025), being logically isolated on the space dedicated to its specific subscription. More, the data should not be used to train any other models, main or private, by the cloud owner, but used only for subscription owner models, reinforcing the need for privacy of each doctoral research. Modern cloud providers offer strong legal assurances on this matter, nevertheless this problematic needs to be carefully assessed.

References

- Adams, J., & Szomszor, M. (2022). A converging global research system. *Quantitative Science Studies*, 3(3), 3(3), 715-731.
- Brysaert, M. (2019). How many words do we read per minute? A review and meta-analysis of reading rate. *Journal of Memory and Language*.
- Godinez, A. (2025). HYSEM-RAG: A Hybrid Semantic Retrieval-Augmented Generation Framework for Automated Literature Synthesis and Methodological Gap Analysis. arXiv.
- Karras, A., Theodorakopoulos, L., Karras, C., Theodoropoulou, A., Kalliampakou, I., & Kalogeratos, G. (2025). LLMs for Cybersecurity in the Big Data Era: A Comprehensive Review of Applications, Challenges, and Future Directions. *Information*, 16, 957.
- Kostopoulos, G., Gkamas, V., Rigou, M., & Kotsiantis, S. (2025). Agentic AI in Education: State of the Art and Future Directions. *IEEE EDUCATION SOCIETY SECTION*.
- Meacham, S., & Sharafzad, A. (2025). Towards Trustworthy and Effective AI for Academic Policy Navigation: Human Evaluation of a Source-Aware, Domain-Optimized RAG-Based Chatbot. *Research Square*.
- Upadhyay, R., & Viviani, M. (2025). Enhancing Health Information Retrieval with RAG by prioritizing topical relevance and factual accuracy. *Discover Computing*, 28, 27.
- Vishesh, G., & Astha, B. (2025). Redefining Infrastructure: The Strategic ESG Case for Cloud over Traditional Hosting. *The American Journal of Applied Sciences*, 7(8), 133–153.