Knowledge Graph-Enhanced Retrieval-Augmented Generation for Earth Observation Data

Roxanne El Baff, Ben Schluckebier and Tobias Hecking

German Aerospace Center (DLR), Institute of Software Technology, Cologne, Germany

Abstract

Large language models have strong capabilities for different purposes, such as searching and question-answering [1]. However, they hallucinate on domain-specific tasks, leading to potential risks such as misinformation spread or decay of trust between the technology and the user. These risks are more or less prominent depending on the context where an LLM is employed. For example, in a scientific setting, such as the Earth Observation (EO) domain, the LLM must ensure important criteria when answering, such as depth and groundedness [2]. To overcome hallucinations, recent research indicates that compound systems, which employ external tools and knowledge alongside LLMs, outperform standalone LLMs. Therefore, this paper presents a compound system to create a question-answering model for the EO domain. More precisely, our approach employs a Retrieval-Augmented Generation (RAG)-based model, focusing on three sequential components: (1) **Data Curation** to enable the LLM to access a semantically interconnected multi-genre corpora (e.g., scientific and datasets) when answering a question. (2) **RAG-Based Model** to balance between the LLM's existing knowledge and the curated data (from (1)). Lastly, (3) **LLM-Based Evaluation** to compare standalone LLM answers to our RAG-based model. Our evaluation across 70 EO questions shows that our approach achieves the highest score across all criteria (e.g., helpfulness), whereas traditional RAG underperforms zero-shot prompting on larger models. Our code and data are available on GitHub (https://github.com/DLR-SC/RAG-for-Earth-Observation) and Zenodo (https://doi.org/10.5281/zenodo.17106948).

Keywords

Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Question Answering (QA), Earth Observation (EO), Knowledge Graph, Data Curation, Context Engineering, LLM-Based Evaluation, Hallucination Mitigation

1. Introduction

Information search in scientific domains is often of an exploratory nature, with many open-ended tasks executed in parallel. Furthermore, such processes are oftentimes opportunistic, iterative, and multi-faceted [3]. In order to increase efficiency regarding the search for information in an ever-growing landscape of heterogeneous information sources, intelligent tools are needed to find, connect, and extract knowledge beyond classical search engines.

Recently, large language models (LLMs) have become an integral part of scientific information systems. They can engage in human-like interactions and speed up information seeking and processing. However, despite the impression of intelligence, it is crucial to acknowledge two issues that arise when using LLMs for generation: hallucination [2], and limited ability to expand or revise knowledge [4]. These problems prove critical when an LLM is used for question-answering within scientific search, where precision and currency are important.

A common mitigation of these problems is combining LLMs with retrieval systems to provide the LLM with additional context information before generating an answer, which is known as "retrieval-augmented generation" (RAG) [4]. This context can entail domain-specific and recent information, which improves the precision and the currency of the LLM's answers.

This paper tackles the question-answering task (QA) using RAG within the specific scientific domain of Earth Observation (EO) and its application in Earth Sciences. Our approach's novelty stems from

Workshop on AI-driven Data Engineering and Reusability for Earth and Space Sciences (DARES'25), co-located with the 28th European Conference on Artificial Intelligence (ECAI 2025), Bologna, Italy, October 25, 2025

roxanne.elbaff@dlr.de (R. El Baff); ben.schluckebier@dlr.de (B. Schluckebier); tobias.hecking@dlr.de (T. Hecking)
0000-0001-6661-8661 (R. El Baff); 0009-0004-4294-4971 (B. Schluckebier); 0000-0003-0833-7989 (T. Hecking)

incorporating both EO datasets and publications as information sources and interconnecting them on a semantic level. As illustrated in Figure 1, our approach has three sequential steps:

(1) Data curation. Data curation creates a reservoir of recent and interconnected data tailored for EO. More precisely, it creates a multi-genre knowledge graph that interconnects published scientific abstracts (e.g., abstracts from OpenAlex) to scientific artifacts (e.g., datasets from PANGAEA), which are semantically connected via *keywords*.

(2) RAG-based model development Our RAG-based model aims to include the most important information in the LLM context, given the limited length of that context; we refer to this part as *context acquisition*. It initially conducts a *lexical-based* search, fetching all matching nodes (publications and artifacts). Then, it broadens the search by traversing the connecting nodes, resulting in a *subgraph*. However, due to the limited LLM context, the subgraph is further filtered via semantic search.

After the *context acquisition*, our model generates the *answer* by employing a *two-step RAG* that leverages the LLM's knowledge and our curated data by: (step 1.) exploiting the general knowledge of the LLM by using a zero-shot prompt, then (step 2.) refining the zero-shot answer by re-prompting the LLM with the acquired context.

(3) LLM-based evaluation To emphasize the added value of our RAG-based approach compared to simple zero-shot prompting, we use LLM-as-a-judge to score the *answer* based on several criteria such as *groundedness* [4], and *helpfulness* [5].

Our evaluation of 70 EO questions using a small LLM (Mistral 24B) and a large LLM (Llama 70B) reveals an intriguing contrast between our two-step RAG, standard one-step RAG, and zero-shot prompting. Our two-step RAG approach, which first generates an answer and then refines with retrieved context, yields the strongest results on all evaluated criteria (e.g., depth, relevance, helpfulness) with a high effect size. In contrast, the standard one-step RAG underperforms even zero-shot prompting on the larger model, resulting in less helpful and shallower answers. Manual inspection reveals that one-step RAG becomes constrained to the retrieved context and cannot integrate the LLM's existing knowledge.

Our contributions are threefold:

- A reservoir of heterogeneous data sources, and their semantic relationships stored in a graph database with more than two million nodes for Earth Observation.
- An approach to integrate such a knowledge graph within an RAG model for complex question answering over publications and datasets.
- An evaluation study based on the LLM-as-a-judge strategy [6] that shows that our approach leads to more grounded and informative answers to EO-specific questions compared to zero-shot prompting, and standard one-step RAG.

2. Related works

Nedumov and Kuznetsov [3] explores the information needs and specific requirements of scientific search tasks to define essential user interface tools for effective scientific search. They define three sets of tools to improve users' exploratory search tasks, focusing on querying, visualizing, and long search:

Tools for querying. These tools improve a user's queries for their search tasks by refining or extending them. In our work, we refine the *question* into a query (yielding a statement) using LLM with zero-shot prompting, similar to [7]. This step improves retrieval for the lexical and semantic search within our approach.

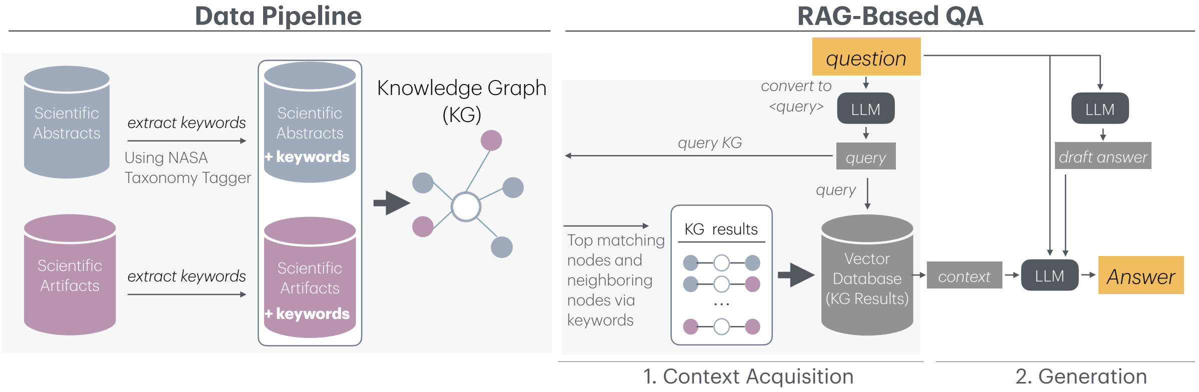

Figure 1: An overview of the two-step RAG approach, including two components: **Data Pipeline** (left), a multi-genre knowledge graph (KG) linking *scientific artifacts* and *scientific abstracts* (publications). Furthermore, the **RAG-Based QA** model incorporates two parts: *1. context acquisition* that, given a question, fetches a subgraph from the KG, embeds the results, and stores them in a vector database (VDB). The top results are then semantically fetched from the VDB to create the *context*. Then, in the *Generation* phase (right), an LLM answers the question by drafting an answer in a Zero-shot manner, then incorporating the *draft* and re-prompting the LLM to generate the final answer.

Tools for analyzing and visualizing search results. These tools help the user make sense of the retrieved documents by visualizing certain aspects of the data and allowing users to filter and restrict the search results (e.g., facets, hierarchical classification, and data visualization). Betz et al. [8] presents a graph-powered visual search engine for Earth Sciences that allows the visual exploration of semantically related datasets. While our work uses similar datasets and a semantically connected knowledge graph, we significantly extend the data included. We then automate exploring and retrieving information from the knowledge graph. More specifically, we broadly acquire the data from a large knowledge graph, explore related work by traversing the graph, and narrow down the acquired data to fit human comprehension. Another difference is that our focus stems from natural language interaction (QA) rather than visualization.

Tools for long search. These tools enable social collaborations for scientists to share their search results among colleagues and save intermediate results to continue search processes over a more extended period [3]. While this work is human-centric, our approach is machine-centric and does not explicitly address *long search*

For science search applications in Earth Science, existing research covers extensive and complex systems, employing traditional information retrieval by hosting scientific data and publications. For instance, Science Discovery Engine [9] provides a comprehensive user interface, allowing complex searches based on topics, tags, and other criteria. Also, PANGAEA[1] provides a data portal for georeferenced data and includes an interactive map for geospatial filtering. These search engines are specialized for EO and can be used as data sources for our *data curation.* Also, in contrast to this work, we focus on question-answering using LLM-based models.

Recent research closely related to our work developed question-answering systems for Earth Observation data archives. EarthQA [10] enables querying satellite images, TerraQ [11] enables spatiotemporal question answering over satellite image archives, and the DA4DTE [12] develops a digital assistant for satellite data archives. Similarly, we focus on natural language queries within a QA system; however, we tackle retrieved, multi-genre, interconnected text rather than images.

Integrating features such as geospatial queries and multimodal support could be significantly beneficial in the future.

[1] https://www.pangaea.de/

3. Approach

This section outlines our approach for a **RAG-based QA** model for the Earth Observation (EO) domain, consisting of two parts: (1) the **knowledge graph creation**, and (2) the two-step answer **generation**, as detailed below.

3.1. Knowledge Graph Creation

A RAG-based model can access external knowledge to improve its answers. The backbone of our system is an integrated knowledge graph including multi-genre Earth Observation data created by an information extraction pipeline similar to Honeder et al. [13].

We create a knowledge graph semantically connecting EO scientific abstracts with scientific artifacts (EO datasets and their metadata) via *keywords*. For that, the data pipeline includes extracting *keywords* from textual contents after fetching and processing them (detailed in section 4).

To this end, we develop TaxoTagger, a tool that matches texts to keywords of a given taxonomy. TaxoTagger is motivated by explicit semantic analysis [14] where the topics of an unknown text are determined by its semantic similarity to documents for which the topic is known.

Similarly, TaxoTagger compares the embedding $e(t)$ of a given text t with the embeddings $e(c)$ of keywords (a.k.a. concept) descriptions c in a given taxonomy, such as NASA's GCMD (Global Change Master Directory), an Earth Science taxonomy[2]. If the cosine similarity between the text and concept description embedding is greater than a given threshold $s_c(e(t), e(c)) > \Theta$, *keyword* c is linked to text t in our knowledge graph.

The semantic similarity score $s_c(e(t), e(c))$ is stored as an *attribute* of the *edge* connecting a document (abstract, or artifact) and a keyword.

In the presented work, we use GCMD[2], a hierarchical set of controlled Earth Science vocabularies as a taxonomy [15, 16] (e.g., earth storable). We detail the implementation and data statistics in Section 4. Figure 2 depicts the ontology of the knowledge graph along with an example.

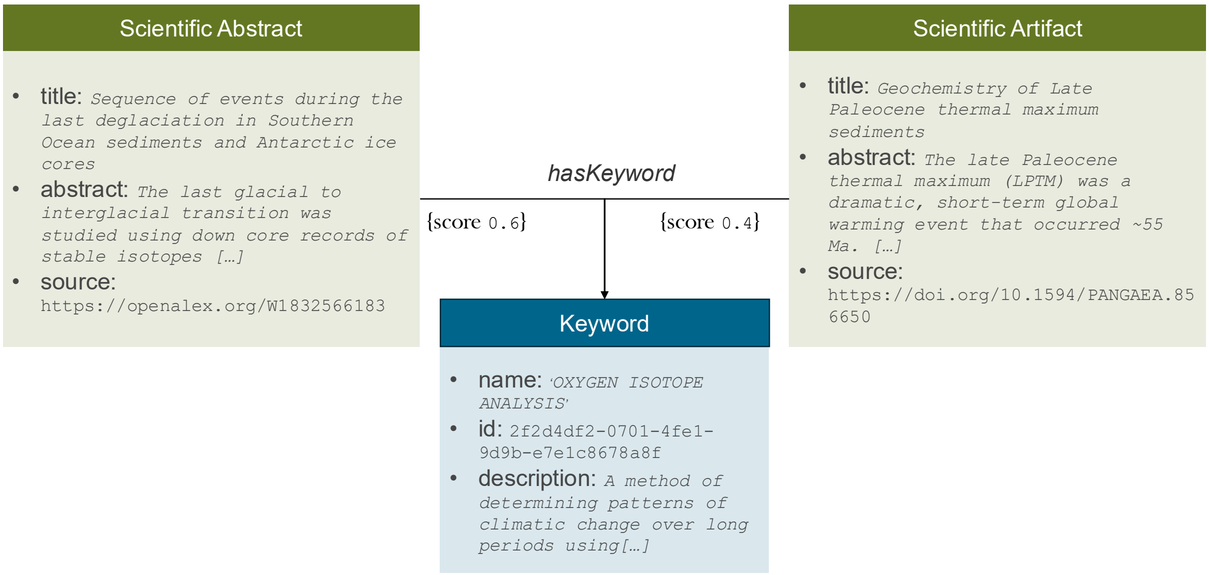
Scientific Abstract
title: Sequence of events during the last deglaciation in Southern Ocean sediments and Antarctic ice cores
abstract: The last glacial to interglacial transition was studied using down core records of stable isotopes […]
source: https://openalex.org/W1832566183
hasKeyword
{score 0.6}
{score 0.4}
Scientific Artifact
title: Geochemistry of Late Paleocene thermal maximum sediments
abstract: The late Paleocene thermal maximum (LPTM) was a dramatic, short-term global warming event that occurred ~55 Ma. […]
source: https://doi.org/10.1594/PANGAEA.856650
Keyword
name: 'OXYGEN ISOTOPE ANALYSIS'
id: 2f2d4df2-0701-4fe1-9d9b-e7e1c8678a8f
description: A method of determining patterns of climatic change over long periods using[…]

Figure 2: The Knowledge Graph Ontology with an example, connecting *Scientific Abstracts* to *Scientific Artifacts* via a *keyword*. The connection strength is stored as a score in the *edge* between the *keyword* and the connecting node.

[2] https://www.earthdata.nasa.gov/data/tools/gcmd-keyword-viewer

3.2. Generation Approach

A RAG-based model requires the user question q and the context c. After fetching c, the answer is generated by prompting the LLM with c and q. This section details the context acquisition and the generation.

Context Acquisition.

This part details the process of creating c for a question q by leveraging the connections between the data points and the exploratory nature of the knowledge graph. We develop a hybrid retrieval process involving three sequential steps, going from a broader exploratory context to a narrower context that fits an LLM prompt: lexical search, graph-based context expansion, and semantic search.

1. **Lexical Search.** When the LLM is prompted with a natural language question (q), this question is first reformulated into a *query* [17, 18] by instructing the LLM to turn q into a statement. The *query* is used to fetch a set of the top $k_{lexical}$ matching nodes, namely scientific abstracts and artifacts (e.g., datasets) using a BM25 search index.
2. **Graph-based Context Expansion.** We expand the context by traversing the graph to include nodes from the neighborhood of the $k_{lexical}$ initially retrieved nodes. This subgraph includes the k_{expand} top-scoring scientific keywords (as described above, the score $s_c(e(t), e(c))$ denotes the semantic relatedness between a document and a keyword), as well as the k_{expand} top-scoring datasets and publications linked to these keywords. This results in a *broad context* enabling information diversification.
3. **Semantic Search.** In the final step, this *broad context* of documents and keywords associated with the retrieved nodes is then chunked, embedded, and saved in a vector database. The n closest documents (chunks) to the query in this semantic vector space form the *narrow context* eventually used to augment the prompt.

c is a formatted *string* representation of the resulting chunks returned from the *Semantic Search* step.

Generation with Context.

This paper compares two strategies to generate an answer based on c. In one-step (or classical) RAG, an augmentation function $f_a(q, c)$ augments q directly with c by creating a prompt that instructs the LLM to consider c explicitly when generating an answer $LLM(f_a(q, c)) = a$.

Since it can be beneficial to condition an LLM to use its own knowledge in addition to the context to avoid context overfitting, we additionally use a two-step RAG as a simple adaptation of multi-step generation [19, 20]. Instead of augmenting q directly, the LLM first generates an initial answer $LLM(q) = \tilde{a}$ in a zero-shot manner, leveraging the LLM's knowledge. Then, the refinement function $f_r(\tilde{a}, c)$ generates a prompt instructing the LLM to refine $\tilde{a}$ taking into account c to generate an answer a

$$a = \mathrm{LLM}\big(f_r(\tilde{a}, c)\big) = \mathrm{LLM}\big(f_r(\mathrm{LLM}(q), c)\big).$$

4. Data for Knowledge Graph Creation

To build the knowledge graph described in Section 3, we acquire scientific abstracts and artifacts mainly from OpenAlex [21]. We additionally extract a relatively small number of datasets from the PANGAEA portal for Earth Sciences [22], and the EOC Geoservice portal of the German Aerospace Center (DLR) [3]. Table 1 shows the count of each node type and its sources. Below, we elaborate on each source:

[3] https://geoservice.dlr.de

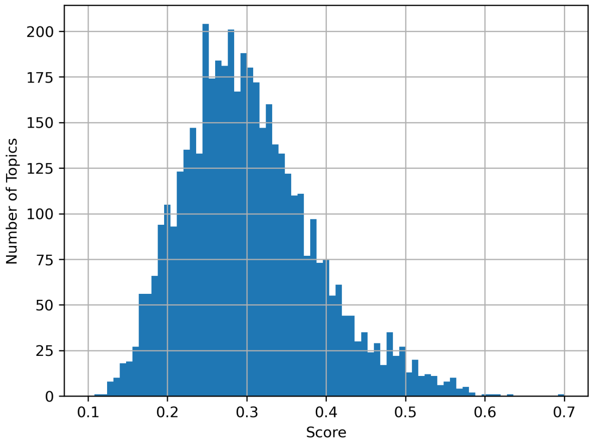

Figure 3: Distribution of the maximum keyword scores, assigned by TaxoTagger for each OpenAlex topic.

OpenAlex for Scientific Abstracts and Artifacts. (2 million entities) OpenAlex (OA) is an open index of scholarly works tagged with topics across all scientific domains. We fetch EO-related datasets $(46,933)$ and publications $(2,021,267)$ metadata like titles and abstracts. To fetch domain-specific data from OA, we do the following steps:

- **OA Topics Selection.** We extract all the OA topics related to the EO domain. More precisely, we extract the highest-scoring keyword for each OA topic using the TaxoTagger. Then, topics (T_{EO}) with a score ≥ 0.4 are selected. This threshold was selected by plotting the histogram of the maximum score and choosing the middle score, as shown in Figure 3.
- **OA Work Selection.** We fetch all the OA works for each topic in (T_{EO}), extract the keywords using TaxoTagger, then filter out works that have a maximum score of < 0.375. This threshold was also selected using the previous approach.

Additionally, we inspected the results manually.

PANGAEA for Scientific Artifacts. (885 datasets) We crawled the datasets' information from the PANGAEA portal for Earth Sciences [22] that contains various georeferenced data, ranging from chemistry to agriculture. The crawler used is part of the OpenSearch project and can be used for multiple data providers[4]. Then, we extracted the content from the resulting WARC[5] files.

EOC Geoservice for Scientific Artifacts. (65 datasets) We acquire Spatio-Temporal Asset Catalog (STAC) [6] collections available via the EOC Geoservice portal of the German Aerospace Center (DLR) [7] using the STAC API[8].

After extracting all relevant metadata and relations, the graph was created with the Corpus Annotation Graph Builder framework (CAG) [23] and stored in an ArangoDB[9] graph database instance. As shown in Table 1, the knowledge graph has 2,021,267 nodes and 41,159,409 edges. Also, Figure 2 shows an example of the knowledge graph content.

5. Experiments and Results

This section reports the evaluation strategy, experiment settings, and results.

[4] https://opencode.it4i.eu/openwebsearcheu-public/opensearch-fill-in-crawls
[5] https://iso.org/standard/68004.html
[6] https://stacspec.org/
[7] https://geoservice.dlr.de
[8] https://geoservice.dlr.de/eoc/ogc/stac/v1/
[9] https://arangodb.com/

Table 1
The count of each node type: Scientific *Abstracts* of publications and *Datasets* (Artifacts), grouped by sources (OpenAlex, PANGAEA, and EOC Geoservice).

Node type	# entries	Source
Scientific Artifact (aka Dataset)	47,883	OpenAlex, PANGAEA, EOC Geoservice
Scientific Abstract (publications)	2,021,267	OpenAlex
Keyword	3,599	NASA's GCMD[2]
Total	**2,072,749**	

We employ an LLM as a judge [6] evaluation approach that relies on generating test data and assessing answers using an LLM, due to the lack of reference data. Then we detail our experiment settings (context acquisition and generation) and report the results.

5.1. LLM-Based Evaluation Approach

Evaluation consists of LLM-based question generation for testing and LLM-based answer assessment.

Question Generation

We generate domain-specific questions by prompting, as shown in Figure 4, an LLM in a zero-shot manner while incorporating two variables: (1) **topic** [24, 25]: we select random topics and their description from the NASA GCMD taxonomy to ensure domain-specific questions, and (2) defining the **intent** behind the question to improve question diversity [26], and are as follows:

- *Exploratory*: asking for discovering new concepts and has no expectations (e.g., *"How do variations in cryospheric indicators, such as glacier retreat and sea ice extent, influence carbon flux dynamics and land surface heat absorption patterns?"*).
- *Comparative*: asking for comparing two scientific concepts, such as methods (e.g., *"How do genetic analysis methods compare to traditional physical characteristic-based approaches in classifying vertebrates, invertebrates, and fungi within the biological classification system?*).
- *Descriptive*: asking for an explanation or description of a concept (e.g., *"What is the role of climate change adaptation in mitigating risks to human settlements?"*.
- *Causal*: asking for cause and effect between phenomena; it usually starts with *why* (e.g., *"How do variations in atmospheric gas concentrations preserved in ice core records influence global temperature changes during glacial and interglacial periods?"*).
- *Relational*: asking for the relationship between two concepts (e.g., *"What is the relationship between cloud cover variability and changes in atmospheric temperature at different altitudes?"*).

We used GPT-4o to generate a set of 70 questions using the prompt in Figure 4. We selected all of the 14 NASA topics (e.g., atmosphere, cryosphere) that exist in our knowledge graph (Section 4) from the NASA GCMD[2] taxonomy.

LLM-as-a-Judge

An LLM is employed to assess several criteria for each answer (a), question (q), and context (c). We adapt the approach of Kim et al. [27], where `gpt-4o-mini-2024-07-18`, as the judging LLM, is prompted to score answers concerning the question and retrieved context along different criteria, with scores ranging from 1 (low) to 5.

We select the 5 rating criteria from the literature:

- *Relevance*: measures the alignment between the question and the answer [5],
- *Groundedness*: measures whether the answer is supported by the given c or a known source [4],

```
### Question Criteria:

INTENT: {intent_category} — {intent_description}

TOPIC:

The topics are extracted from the NASA GCMD Taxonomy. The TOPIC is described and few of its
    SUBJECT AREAS.

<topic>
    {context}
</topic>


### Instructions

The question must strictly reflect the stated INTENT and be centered on the specified TOPIC
    along with its SUBJECT AREAS. Do not include any background or —explanationjust the
    question.


### Question:
```

Figure 4: Question Generation Prompt, to generate *topic* and *intent* specific question.

- *Helpfulness*: measures whether the answer provides useful and contextually appropriate content [5],
- *Depth*: measures whether the answer is detailed and broad [28], and
- *Factuality*: measures whether the answer is scientifically correct and verifiable [29].

5.2. Experimental Settings

This section details our experiment settings by first defining our *contextacquisition* configurations and then describing our experimental configuration for generating answers.

Context Acquisition Configuration. Based on initial experiments, we chose the parameters of the retrieval process (see Section 3.2) as follows: given a query q, we fetch the top $k_{lexical} = 16$ best matching publication and dataset nodes from the knowledge graph (lexical search). Then, for context expansion, for each of the 16 nodes, we fetch the top $k_{expand} = 3$ *keywords* with the highest scores. After that, for each of the *keywords*, the top $k_{secondary} = 2$ most related documents are included. Note that we only consider connections with a score $s_c(e(t), e(c)) > 0.3$ to avoid expansion in unrelated areas. This subgraph is then embedded using the model `mistral-embed` and saved with ChromaDB[10] vector database (VDB) with $chunksize = 256$ and $overlap = 64$. Lastly, the top $k_{semantic} = 10$ documents are fetched to be included in the generation prompt.

Model Configurations. We run our experiments using two open-weight LLMs, varying in their parameter size: Mistral Small 24B, and Llama-3.3 70B. For our implementation, we use the LangGraph Python agent framework[11], along with LangChain. Each of these models answers the 70 questions in a zero-shot (**0shot**) manner, one-step RAG (**rag**), and two-step RAG (**2rag**).

The 420[12] are evaluated using *gpt-4o-mini-2024-07-18*.

Table 2

Mean scores using Mistral Small (left) and Llama 3.3 70B (right), for each experiment for two-step generation RAG (*2rag*), one-step generation RAG (*rag*) and Zero-Shot generation (*0shot*). The *overall* and per criterion mean scores is reported. * denotes approaches that achieve significantly higher scores than the *0shot* baseline, while †indicates scores that are significantly higher than those obtained with *rag*..

	(a) Mistral Small						(b) Llama 3.3 70B					
	Overall	**Factual**	**Relev.**	**Ground.**	**Helpful**	**Depth**	**Overall**	**Factual**	**Relev.**	**Ground.**	**Helpful**	**Depth**
2rag	**4.95*†**	**5.00**	**5.00**	**4.80*†**	**5.00*†**	**4.96*†**	**4.89*†**	**4.97**	4.98	**4.63*†**	**4.95*†**	**4.92*†**
rag	4.80*	4.94	5.00	4.55*	4.80	4.70	4.50	4.83	**5.00**	4.24*	4.33	4.10
0shot	4.71	4.95	**5.00**	3.93	4.88	4.81	4.63†	4.90	**5.00**	3.83	4.77†	4.68†

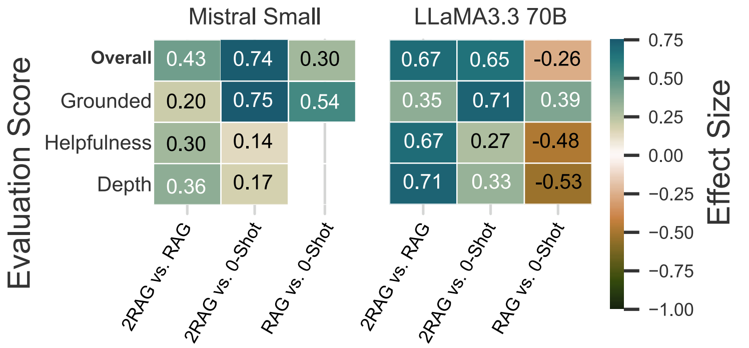

Figure 5: Heatmap for each criterion (Overall, Groundedness, Helpfulness, and Depth) for Mistral Small (left) and Llama 3.3 70B (right). The y-axis represents each assessed criterion, and the x-axis represents each effect-pair (m1 vs m2). Each cube represents the effect size *r*: a green/blue color indicates a positive effect size where m1 has a significantly higher number than m2 (m1 » m2), whereas a brown/orange color indicates the opposite (m1 « m2).

5.3. Results

The results for experiments with Mistral Small 24B, and Llama-3.3 are reported in Table 2 that shows the mean score for all five criteria[13].

We measured significance between scores, per model, using ANOVA in cases of normality (Kruskal-Wallis otherwise). If $p < 0.05$, we conducted post-hoc analysis (independent t-test in case of normality, Mann-Whitney otherwise) with Bonferroni correction. We report below (and in Table 2) the results where $p < 0.05$. Additionally, we calculate the effect size r to quantify the strength of the observed significant differences, as shown in Figure 5.

One-step RAG is rated worse than zero-shot on big LLM. One-step RAG (rag) has a surprisingly bad rating when using Llama 3.3 70B compared to the Zero-shot (0shot) model, which outperforms the RAG-based model *overall* with a medium effect size $r = 0.26$. More specifically, *helpfulness* and *depth* scores decrease significantly with a high effect of $r = 0.48$ and $r = 0.53$. This pattern was not observed when using Mistral; these two criteria had no significant differences.

Answer relevance is constantly rated high for the zero-shot approach, which can be attributed to the fact that, without additional context, the answers are closely aligned with the questions. As a result, they get higher relevance judgments. In contrast, answers produced by RAG sometimes consider the context created from the retrieved information, affecting their relevance. By manual inspection, we found that zero-shot prompting, in most cases, gives longer and more elaborate answers. In contrast,

[10] https://trychroma.com/

[11] https://langchain-ai.github.io/langgraph/

[12] 70 questions, two models, and three approaches yield 420 answers.

[13] The generated questions and answers are available on Zenodo (https://doi.org/10.5281/zenodo.17106948)

one-step RAG answers are very much fitted (or aligned) to the given context. Answers often contain references to publications and datasets but include little additional information.

The Two-Step RAG Generation performs the best. As a hybrid approach that uses RAG to improve a zero-shot generated answer, two-step RAG (2rag) combines the strength of the two other approaches - the grounded answers of RAG and the ability to elaborate on them further by leveraging the LLM's internal information. It outperforms the one-step RAG and zero-shot approaches, using either a small (overall score: 4.95) or a larger LLM (overall score: 4.89). This increase in performance strength is illustrated by the high effect size (Figure 5) ranging between $r = 0.43$ and $r = 0.74$.

6. Conclusion and Future Works

This paper presents a RAG (Retrieval-Augmented Generation) system designed for Earth Science, specifically Earth Observation (EO). It creates a multi-genre, semantically interconnected knowledge graph (KG) compiled from datasets, publications, and an established EO taxonomy. The KG presents a rich, exploratory, and information resource for an LLM to answer EO scientific questions, facilitating the retrieval of accurate, broad and up-to-date information. Our initial LLM-based evaluation shows promising results, demonstrating significant improvement in the quality of the generated answers. These findings establish that our approach is a step forward in increasing trust in generative AI in scientific domains. Consequently, our system can be used as a standalone tool or as a component enriching larger Earth Science information systems.

As future work, we plan to further integrate web search to increase the system's capabilities. To this end, shards of the Open Web Index[14] [30] relevant to Earth Sciences will be used as an additional data source. Furthermore, we will provide more extensive evaluation studies involving human ratings.

Acknowledgments

This work is part of the OpenWebSearch.eu project, which is funded by the EU under GA 101070014. We thank the EU for its support.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4 and Grammarly to conduct a Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] C. Zhai, Large language models and future of information retrieval: opportunities and challenges, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 481–490.

[2] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al., A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, ACM Transactions on Information Systems 43 (2025) 1–55.

[3] Y. Nedumov, S. Kuznetsov, Exploratory search for scientific articles, Programming and Computer Software 45 (2019) 405–416. doi:10.1134/S0361768819070089.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.

[14] https://openwebindex.eu/

[5] F. Santos, X. Wang, K. Roberts, Measuring coverage and completeness in open-domain qa, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2024.

[6] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in neural information processing systems 36 (2023) 46595–46623.

[7] K. D. Dhole, E. Agichtein, Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation, in: European Conference on Information Retrieval, Springer, 2024, pp. 326–335.

[8] P. K. Betz, T. Hecking, A. Schreiber, A. Gerndt, Knowledge graph based visual search application, arXiv preprint arXiv:2410.22846 (2024).

[9] K. Bugbee, A. Acharya, C. Davis, E. Foshee, R. Ramachandran, X. Li, M. Ramasubramanian, NASA's Science Discovery Engine: An Interdisciplinary, Open Science Data and Information Discovery Service, Technical Report, Copernicus Meetings, 2023.

[10] D. Punjani, M. Koubarakis, E. Tsalapati, Earthqa: A question answering engine for earth observation data archives, in: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2023, pp. 1396–1399.

[11] S.-A. Kefalidis, K. Plas, M. Koubarakis, Terraq: Spatiotemporal question-answering on satellite image archives, arXiv preprint arXiv:2502.04415 (2025).

[12] M. Corsi, G. Pasquali, C. Pratola, S. Tilia, S.-A. Kefalidis, K. Plas, M. Pollali, E. Tsalapati, M. Tsokanaridou, M. Koubarakis, et al., Da4dte: Developing a digital assistant for satellite data archives, ????

[13] J. Honeder, R. El Baff, T. Hecking, A. Nussbaumer, C. Guetl, A geo-contextualized multi-genre scientific search engine: A novel conceptual design and prototype evaluation, in: 8th International Conference on Geoinformatics and Data Analysis, Springer, 2025.

[14] E. Gabrilovich, S. Markovitch, et al., Computing semantic relatedness using wikipedia-based explicit semantic analysis., in: IJcAI, volume 7, 2007, pp. 1606–1611.

[15] J. Dutra, J. Busch, Nasa technical white paper-enabling knowledge and discovery: Taxonomy development for nasa, Retrieved January 15 (2003) 2003.

[16] D. Miranda, 2020 NASA technology taxonomy, Technical Report, NASA, 2020.

[17] W. B. Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice, Addison-Wesley, 2010.

[18] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, C. Welty, Building watson: An overview of the deepqa project, AI Magazine 31 (2010) 59–79.

[19] T. Chen, Y. Luo, H. Zhang, et al., Self-refine: Iterative refinement with self-feedback, in: Proceedings of ACL 2023, 2023.

[20] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: Proceedings of ICLR 2023, 2023.

[21] J. Priem, H. Piwowar, R. Orr, Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, arXiv preprint arXiv:2205.01833 (2022).

[22] J. Felden, L. Möller, U. Schindler, R. Huber, S. Schumacher, R. Koppe, M. Diepenbroek, F. O. Glöckner, PANGAEA - Data Publisher for Earth & Environmental Science 10 (2023) 347. URL: https://www.nature.com/articles/s41597-023-02269-x. doi:10.1038/s41597-023-02269-x.

[23] R. El Baff, T. Hecking, A. Hamm, J. W. Korte, S. Bartsch, Corpus annotation graph builder (cag): An architectural framework to build and annotate a multi-source graph, 2023. URL: https://doi.org/10.5281/zenodo.10285155. doi:10.5281/zenodo.10285155.

[24] J. Wang, M. Zhou, X. Liu, Y. Zhang, Towards automatic generation of question-answer pairs for domain-specific qa, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2022.

[25] A. Pipitone, D. Diefenbach, K. Singh, C. Ghidini, Using ontologies for automatic question generation: A case study in domain qa, in: Proceedings of the 21st International Semantic Web Conference (ISWC), 2022.

[26] Y. Pan, C. Zhang, M. Liu, Q. Yang, Intent-guided question generation for improving qa coverage and diversity, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023.

[27] S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, et al., Prometheus: Inducing fine-grained evaluation capability in language models, in: The Twelfth International Conference on Learning Representations, 2023.

[28] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020.

[29] O. Honovich, T. Scialom, L. Choshen, E. Shnarch, O. Abend, et al., True or false? faithful summarization with attribution, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), 2022.

[30] M. Granitzer, S. Voigt, N. A. Fathima, M. Golasowski, C. Guetl, T. Hecking, G. Hendriksen, D. Hiemstra, J. Martinovič, J. Mitrović, et al., Impact and development of an open web index for open web search, Journal of the Association for Information Science and Technology 75 (2024) 512–520.