Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Bachelor's Thesis

Study program B.Sc. Computer Science

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

by
Schluckebier, Ben

First examiner	Prof. Dr. Andreas Hackelöer
Second examiner	Dr. Tobias Hecking (DLR)
Supervisor	Roxanne El Baff (DLR)
submitted on	September 29, 2025

Declaration

I hereby declare that I have written the work submitted by me independently. All passages that are taken verbatim or paraphrased from published or unpublished works by others have been identified as such. All sources and aids used for this work have been cited. This work has not been submitted to any other examination authority with the same content or in substantial parts.

Rheinbach, September 29, 2025

Eidesstattliche Erklärung

Ich versichere an Eides statt, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Rheinbach, 29. September 2025

Abstract

The search for scientific information is frequently exploratory, involving several open-ended tasks conducted simultaneously. These searches tend to be iterative, opportunistic, and utilize various tactics, resulting in a multi-faceted research process [1]. The sheer mass of available data has a big impact on efficiency, but this can be improved upon by the usage of intelligent tools, discovering, linking and extracting knowledge beyond the capabilities of conventional search engines.

Large language models (LLMs) are increasingly being woven into scientific information retrieval infrastructures [2], [3]. Their ability to engage in human-like conversations allows for acceleration in both the search itself and the processing of the retrieved knowledge. Nevertheless, despite their seeming intelligence, key limitations have to be recognized: (i) hallucinations [4] and (ii) a limited ability to revise or expand their internal knowledge [5]. These shortcomings become especially problematic when employing an LLM in scientific question-answer scenarios, where up-to-date information and accuracy are of utmost importance.

Retrieval-augmented generation (RAG) addresses the mentioned limitations, being a recent technique for providing additional context information to the LLM. By injecting domain-specific and recent information, RAG enhances the answer's accuracy and bypasses knowledge limitations [6]. Further enhancements can be made using knowledge graphs incorporating the semantic relationships of the data [7].

This thesis creates such a RAG system for the scientific domain of Earth Observation (EO), a collective term for various earth sciences like oceanography or atmospheric chemistry. The presented approach contributes (i) the creation of a heterogeneous knowledge graph, compromised of EO datasets and publications, (ii) a RAG integration utilizing the aforementioned graph for the execution of question-answer tasks and (iii) an automatic evaluation using LLM-as-judge, based on the criteria context-relevance, answer-relevance and groundedness [8], showing the RAG system surpassing the zero-shot approach in context-relevance and groundedness.

Contents

List of Figures

List of Tables

Nomenclature

AI	Artificial Intelligence
API	Application Programming Interface
AQL	ArangoDB Query Language
CAG	Corpus Annotation Graph Builder
DLR	German Aerospace Center
EO	Earth Observation
EOC	Earth Observation Center
LLM	Large Language Model
NASA	National Aeronautics and Space Administration
RAG	Retrieval-Augmented Generation
RDF	Resource Description Framework
STAC	SpatioTemporal Asset Catalog
UUID	Universally Unique IDentifier
XML	Extensible Markup Language

1 Introduction

Artificial intelligence (AI) has conquered the world. What once was only known to experts in the field of computer science has spread to a term known by billions of people. Probably by far the best known AI product is ChatGPT[1] by OpenAI[2], a chatbot giving the impression of something / someone intelligent on the other side of the screen. This type of AI is called *large language model* (LLM). These LLMs can engage in text-based human-like interactions and have a (not necessarily correct) answer to almost everything you can think of.

Despite the impression of being intelligent, their knowledge and conceptual understanding are limited [9]. They are constrained by the data they were trained on [5] and can „hallucinate“. *Hallucination* is the term that describes when an LLM makes up information. For the user, there is no difference indicating whether the provided output is based on facts or not [4].

A more recent mitigation of these problems is so called *retrieval-augmented generation* (RAG) [6]. This approach provides the LLM during or right before the conversation with data relevant to the question. For example, an LLM tasked to provide information about the current weather, could retrieve this information from the German weather service[3].

RAG systems are especially useful for tasks where factual correctness is imperative, one of them being scientific search. Searches in its various domains explore many different open-ended questions [1], opening up a well suited use case for LLMs. One of these domains is Earth Observation (EO), incorporating various fields such as oceanography and environmental science.

Based on data from multiple providers, an interconnected, multi-genre knowledge graph can be created, serving as basis for a tailored RAG system accelerating scientific research and providing consistently accurate information.

1.1 Research Question / Aim

This thesis examines the central question of „How can a knowledge graph-based RAG model improve the accuracy and relevance of responses compared to a zero-shot language model, in the context of the domain of Earth Observation?“. To investigate this question, the use of a domain specific knowledge graph is proposed, incorporating data from various providers, which feeds into a RAG system to combat the challenges of hallucinations and restricted knowledge. The resulting system is evaluated and compared against the naive zero-shot approach, based on the metrics of context relevance, answer relevance and groundedness. The hypothesis is that integrating a knowledge graph with a RAG model significantly outperforms the baseline zero-shot model.

[1] `https://chatgpt.com`
[2] `https://openai.com/`
[3] `https://dwd.de/`

1.2 Contributions

The contributions of this work are 3-fold:

- Creation of a heterogeneous scientific knowledge graph, including over 2 million semantically linked data nodes from multiple domain specific providers.
- A RAG integration enabling the execution of question-answer tasks based on data from the knowledge graph.
- An automated evaluation using LLM-as-a-judge, showing an improvement in EO related question-answer tasks of the RAG system compared to zero-shot prompting.

1.3 Structure

This thesis consists of six chapters and starts with chapter 2, presenting related work and elaborates on theoretical foundations essential to this work. Following is chapter 3, which discusses the design considerations regarding obtaining and processing domain specific data, the ontological model of the knowledge graph and the RAG system, including retrieval and processing of the context. The actual implementation in chapter 4 outlines data acquisition and processing, the creation of the knowledge graph and the RAG system including its multiple stages. Evaluation of the system is discussed in chapter 5, followed by the conclusion and future works in chapter 6.

2 Background and Related Work

This chapter provides an overview about the core concepts of this work. Section 2.1 discusses relevant definitions, including Large Language Models, Knowledge Graphs, Retrieval-Augmented Generation and Earth Observation. An overview of related work is given in section 2.2.

2.1 Background

In this section the theoretical foundations and core concepts essential to this thesis are presented.

2.1.1 Large Language Models

Large language models (LLMs) are AI models specifically trained to parse and generate human-like text. Based on the transformer architecture introduced in „Attention Is All You Need“ [10] they use millions, billions or even over a trillion parameters [11] to estimate patterns and predict the subsequent word or sequence of words. LLMs are capable of a variety of tasks, including summarization, translation, coding, providing world knowledge and much more. Although their knowledge is constrained by the data they were trained on, newer models incorporate tool calling, which allows for the use of external services and knowledge [12].

Inputs to an LLM are called *prompts* and can be any text, ranging from simple questions to multi-step instructions with extensive examples. Giving only a question or instruction to the model without any examples is called *zero-shot* prompting. With this approach the LLM relies exclusively on its knowledge gained during training. *Few-shot* prompting on the other hand includes some examples or task demonstrations, allowing the model to infer any patterns from the provided context. *Reflection-based* approaches built upon these strategies by tasking the LLM with evaluating and revising it own output. This allows for the detection of flaws and improves reasoning as well as general output quality [13], [14].

Across this thesis multiple different LLMs will be used for query formulation, user interaction and finally automatic evaluation.

2.1.2 Knowledge Graphs

The core concept of knowledge graphs is to represent data using graphs. Using vertices and edges enables abstraction in various domains, allowing for relations capturing biological interactions, transport networks and much more. Opposing to classical relational databases, a knowledge graph allows postponing the schema definition, enabling flexible data evolution.

An additional benefit of having a graph lies in advanced querying, enabling navigational operations leveraging recursive searches using the entities relations, whilst relational models only allow for generic operations like joins and unions [7].

Such a knowledge graph will be used to systemically store all data used in this thesis, providing both information about the data itself as well as its semantic relationships.

2.1.3 Retrieval-Augmented Generation

Approaching or leaving the limits of an LLMs' „knowledge", one encounters the phenomenon of hallucination. An LLM hallucinates when the output no longer corresponds to the facts or the training data, even if this is not apparent to the user apart from the content itself. The above-mentioned limits are quickly reached, particularly with specialized questions. Retrieval-augmented generation (RAG) is a possible solution here.

RAG enables the LLM to use external knowledge by the means of a retriever and thus to answer questions whose answers were not previously available to the LLM. This process can be carried out once at the start or repeatedly during the interaction [6]. The external knowlede in this thesis will be the earth observation data stored in a knowledge graph.

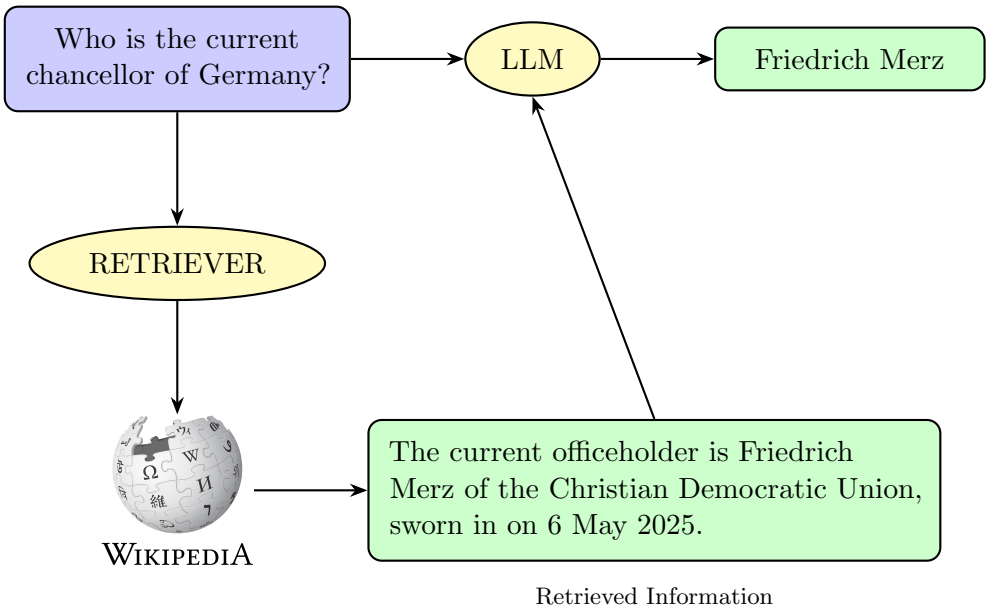

Figure 2.1: Example for a simple RAG system implementing Wikipedia as data provider. (a) Logo courtesy of Wikimedia Foundation [15]; (b) Text excerpt from [16].

2.1.4 Earth Observation

Earth Observation (EO) is a term used to describe various methods to observe the earth's surface and atmosphere using remote sensing instruments. Such observations can be made from planes, drones or, as most commonly known, by satellites.

Imagery is either passive or active. Passive imagery senses e.g. light intensity in the visible and not-visible spectrum, whereas active imagery needs a transmitter sending out signals that reflect from their target and can be sensed by the observation unit. Although images or videos like time-lapses of clouds are commonly associated with the term *imagery*, a variety of instruments can monitor many more different things, such as atmospheric chemistry or even gravitational fields [17].

However, EO data is not only collected for scientific reasons. Image data in particular is extremely relevant for the military, allowing for unparalleled capabilities in surveillance and reconnaissance [18]. Private operators also collect EO data, using it for scientific research or map services like e.g. Google Maps [1] [19].

[1] `https://google.com/maps/`

2.2 Related Work

Previous work explored the usage of knowledge graphs and RAG algorithms in different domains. Bensch et al. presented the usage of RAG with an underlying knowledge graph for various spaceflight procedure AI assistants [20]. Work from VanGundy et al. showed akin methods on the example of NASA's Air Traffic Management-Exploration (ATM-X) project [21].

In closer context of EO, Kao et al. evaluated the usage of RAG for automating EO tasks, without using an underlying knowledge graph. Their evaluations highlighted the effectiveness of LLMs in zero-shot, few-shot, and reflection-based approaches. Although the domain overlaps with this work, the approach differs in that the actual EO data (satellite imagery) was used, whilst this thesis focuses on associated metadata like titles, abstracts and authorships [22].

Previous work by Honeder focused on a small EO dataset and traditional search engines, incorporating a knowledge graph for data storage and retrieval [23].

This thesis builds on these foundations by applying RAG models to a larger dataset and integrating a novel simplistic knowledge graph, leveraging the power of semantic relations. Data is provided by the data publishers OpenAlex [24], PANGAEA [25] and the EOC Geoservice[2]. The knowledge graph itself will be constructed using the „Corpus Annotation Graph Builder“ (CAG) [26] tool.

[2] `https://geoservice.dlr.de/`

3 Approach

This chapter presents the 3-fold approach for building a RAG based model for Earth Observation (EO): Data selection and curation, contents and ontology of the knowledge graph, and design of the retrieval process and LLM implementation.

3.1 Data

In this section, the various types and origins of the data, as well as the processing steps applied to it, are presented.

3.1.1 Science Keywords

Underlying all other data is the science keyword taxonomy, a subset of NASA's Global Change Master Directory (GCMD)[1]. This database contains more than 30 000 descriptions of earth science related data. To establish semantic connections between these science keywords to the data described in the rest of this chapter **TaxoTagger**[2] is employed.

This tool, developed by the German Aerospace Center (DLR), allows assigning taxonomy concepts (e.g. science keywords) to any text. It uses a combined strategy of sentence level embedding with aggregation and saturation, as well as graph traversal over various taxonomy concepts. All assigned keywords provide a score σ between 0 and 1, showing how well the keyword fits the text.

3.1.2 EOC Geoservice

The EOC Geoservice is a data provider for selected geospatial data, run by the Earth Observation Center (EOC)[3] of the DLR. Data can be accessed in a variety of ways, one of them being a SpatioTemporal Asset Catalog (STAC)[4] API. STAC provides a common interface to handle geospatial data like satellite imagery.

3.1.3 OpenAlex

OpenAlex was launched as a drop-in replacement for the Microsoft Academic Graph[5], which was discontinued on December 31, 2021. It is a fully-open dataset of scholarly metadata and is growing ever since its release on January 1st 2022. Data is divided into a total of five different scholarly entity types, of which only the *work* type will be used in this thesis.

The work type entails a variety of scholarly works, including books, datasets and theses. For the purposes of this thesis the only relevant work types are *dataset* and *article*. Datasets are of particular interest since EO research relies on the collection of different kinds of data. Articles include all sorts of publications like papers, journal articles and theses. Analyzing datasets usually leads to the publication of some kind of article and is therefore also relevant for this work [24].

[1] `https://data.nasa.gov/dataset/global-change-master-directory-gcmd`
[2] The version used is still in beta and not publicly available.
[3] `https://dlr.de/de/eoc/`
[4] `https://stacspec.org/`
[5] `https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/`

The data provided by OpenAlex is not domain specific, resulting in the need of filtering out anything unrelated to EO. This can be achieved using two distinct filtering stages and a set of well-defined criteria, to which every work must adhere to be considered. The common criteria are defined as follows:

- The work must either be of the type **dataset** or **article**.
- If the work is of the type **article** it must have been **peer-reviewed**.
- The work must **provide a title**.
- The work must **provide an abstract**[6].
- The language of the work must be **english**.
- The date of the work's **publication** must be **after 1984-04-07**[7].

Topic Filtering The first of the two filtering steps is based on the *primary topic* of a work. Topics are part of the taxonomy used by OpenAlex connecting all works and can be used as a reference to decide whether a work belongs to a certain domain. Akin to the TaxoTagger, topics also are assigned with a score, the primary topic being the topic with the highest score. In this first step science keywords are assigned to all topics, allowing the removal of all topics with a highest keyword score below a common threshold θ_T.

Work Filtering The second step builds upon the first one and assigns science keywords to all remaining works of the chosen topics. A second threshold θ_W will be set, removing even more works with a highest keyword score below. The first step is needed, as tagging all works of OpenAlex would not be possible in a sensible amount of time. After these two steps all remaining works can be considered as relevant to the domain of EO.

3.1.4 PANGAEA

PANGAEA is a data publisher for earth and environmental science, providing access to a plethora of georeferenced data. It is hosted by the Alfred Wegener Institute, Helmholtz Centre for Polar- and Marine Research (AWI)[8] and the Center for Marine Environmental Sciences (MARUM)[9] and has a 30-year history in publishing data [25]. Opposing to OpenAlex and in accordance with the Geoservice, datasets are not filtered. This step can be omitted as all datasets on PANGAEA are related to EO anyway. Due to the absence of a STAC or similar easy to use API, a crawler will be used to acquire data.

3.2 Knowledge Graph

All data will be stored as a knowledge graph, using nodes to represent datasets and publications, referred to as *documents* from this point onward, authors and science keywords that are connected by edges representing their relations. *HasAuthor* denotes the relations of documents and their authors and *HasKeyword* connects documents to science keywords with a score σ. The relations *IsReferencedBy* and *IsRelatedTo* are unique to documents originating from OpenAlex. Documents citing each other are connected via *IsReferencedBy*, whilst the *IsRelatedTo* relation is algorithmically computed by OpenAlex to link documents sharing concepts from the OpenAlex taxonomy.

[6] This is important because the *tagging* uses the abstract as reference to assign the science keywords.
[7] This date was chosen as it marks the first publication of a STAC item know to the author.
[8] `https://awi.de`
[9] `https://marum.de/`

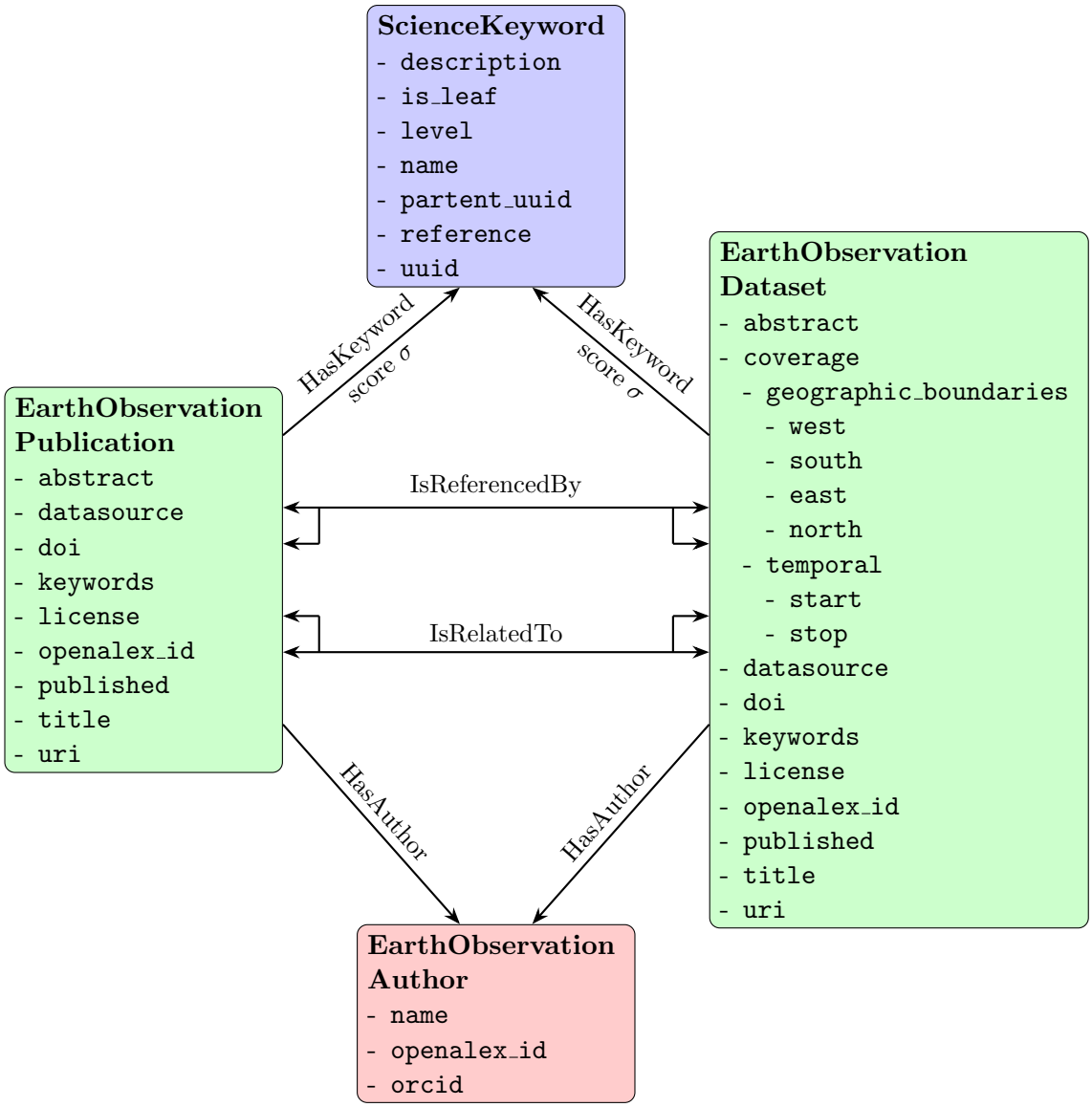

Figure 3.1: Ontology of the knowledge graph.

3.3 RAG System

The RAG system facilitates the connection between the knowledge graph and the user. A two-step process will be implemented, aiming to provide highly relevant and accurate knowledge.

Broad Context In a first step a lexical search on the knowledge graph will be run, returning a set K of best matching documents. Utilizing the graph structure, traversal to the connecting science keywords will be run. For each document $\kappa \in K$ a set Φ_κ of the highest scoring science keywords is retrieved. Additionally, for each science keyword $\varphi \in \Phi_\kappa$ the set Ψ_φ, containing the highest scoring documents, is fetched. The score of a document $\sigma_{\max}$ is the maximum score of all associated keywords. Therefore, the *highest scoring* science keywords or documents are those with the highest $\sigma_{\max}$, with $\sigma_{\max} \geq \theta_{expand}$ to rule out less relevant relations. The now established tree structure (see fig. 3.2) is called the *broad context*.

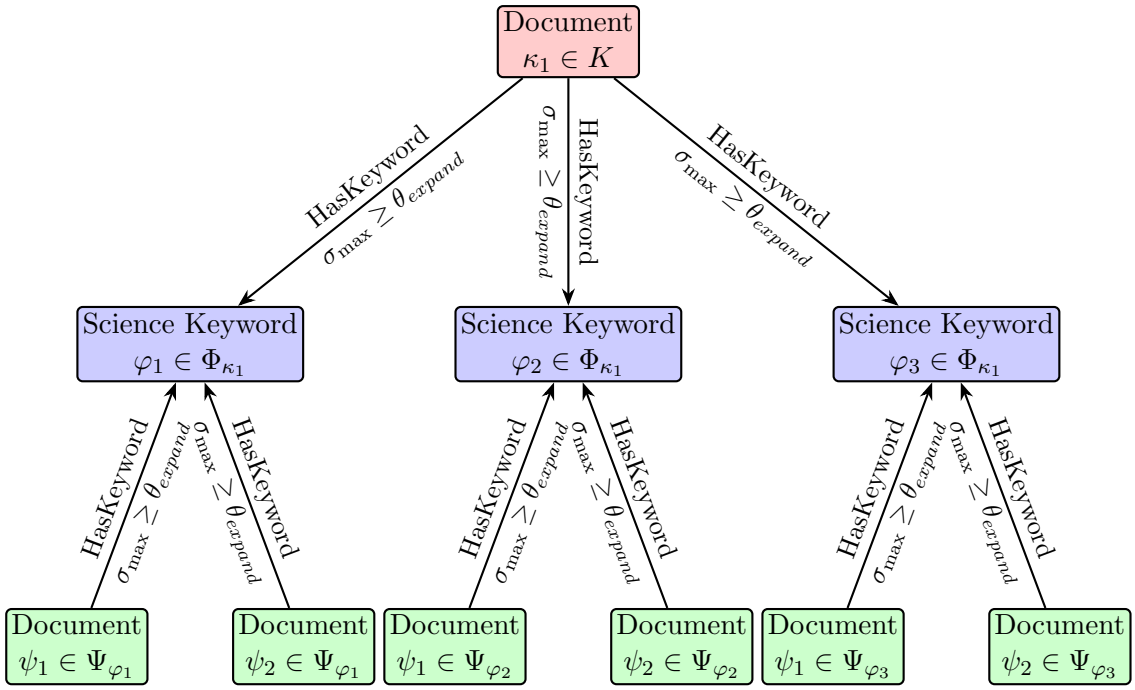

Figure 3.2: Exemplary subgraph with one lexical document $\kappa \in K$, three science keywords $\varphi \in \Phi_\kappa$ and two expand documents $\psi_k \in \Psi_\varphi$, all with a common threshold of θ_{expand} i.e. $\sigma_{\max} \geq \theta_{expand}$.

Narrow Context The now established *broad context* contains highly relevant but also less relevant data to the user's query. To only inject the most relevant context an intermediate step is taken. A vector database will be created from the subgraph and only the most relevant parts of the broad context are passed down to the LLM. This ensures the model is only provided with the most relevant data.

The created *narrow context* will then be injected into the LLMs prompt so it can answer the user's question based on the factual content extracted and processed from the knowledge graph.

4 Implementation

This chapter presents the implementation of the approach outlined in chapter 3. The acquisition of data from multiple providers and the subsequent processing will be discussed in section 4.1. Additionally, the creation of the knowledge graph will be shown in section 4.2. The chapter concludes with the implementation of the RAG system in section 4.3.

4.1 Data Acquisition and Processing

This section outlines the different types of data, how it is obtained, and which processing steps are executed.

4.1.1 Science Keywords

NASA provides an API endpoint[1] to access its GCMD taxonomies in various file formats. The science keywords taxonomy is fetched as RDF[2], an XML[3] syntax general-purpose language. This file format allows for hierarchical / tree-like data structures, such as the science keyword taxonomy. The resulting data is then combined into a graph and subsequently further processed by removing irrelevant metadata, including versioning information and changelogs. The following attributes are extracted:

Attribute	Description
`description`	A short description what the keyword is about
`is_leaf`	If the keyword is a leaf (has no children)
`level`	The hierarchical level of the keyword
`name`	The name of the keyword
`parent_uuid`	The UUID[4] of the parent keyword
`reference`	A reference to where the keyword originates from
`uuid`	Keyword UUID

Table 4.1: Relevant metadata of a science keyword.

The level of a keyword hints at its role in the taxonomy. Categories are fields of science, which are refined by a topic. Topics have associated terms, followed by variables level 1 to 3, going more and more specific with each layer. An example for the keyword *Subtropical Depression Track* can be found in table 4.2.

Assigning these keywords is done by TaxoTagger as detailed in section 3.1.1. The text used for assignment is the abstract of a document. Each edge connecting a science keyword with a document includes a score σ between 0 and 1, whereas $\sigma = 1$ would represent a perfect match.

[1] `https://gcmd.earthdata.nasa.gov/static/kms/`
[2] Resource Description Framework, `https://w3.org/TR/REC-rdf-syntax/`
[3] Extensible Markup Language, `https://w3.org/TR/xml/`
[4] Universally Unique IDentifier, `https://rfc-editor.org/rfc/rfc9562.html`

Keyword Level	Example
Category	Earth Science
Topic	Atmosphere
Term	Weather Events
Variable Level 1	Subtropical Cyclones
Variable Level 2	Subtropical Depression
Variable Level 3	Subtropical Depression Track

Table 4.2: Example GCMD hierarchy of the science keyword **Subtropical Depression Track** with all parents and levels [27].

4.1.2 EOC Geoservice

All STAC Collections offered by the EOC Geoservice can be directly accessed via the provided STAC API[5]. Author information cannot be extracted via this interface. The resulting 65 datasets are then tagged and stored as single `.json`.

4.1.3 OpenAlex

OpenAlex provides a snapshot of its database[6] as a public S3[7] bucket on Amazon Web Services servers, which allows for a straightforward data acquisition. The whole snapshot contains over 200 million works and is more than 400 GB compressed and over 2 TB uncompressed in size. The relevant data is only a very small subset, so filtering is crucial.

Topic Filtering As a first step to rule out the vast majority of items the OpenAlex topics[8] are used. Topics are part of a taxonomy used to categorize works. There are a total of 4561 topics (at time of writing), each with a short description. To determine whether a topic matches the domain of EO or not, TaxoTagger assigns science keywords to each topic. Now that every topic got a list of keywords associated, a threshold $\theta_T = 0.4$ can be set. Every topic with $\sigma_{\max} \geq \theta_T$ is deemed as relevant. The decision to opt for 0.4 was made after plotting distributions (see fig. 4.1) and manually inspecting the topics for certain threshold ranges. After careful considerations the mentioned threshold of 0.4 was chosen, leaving a total of 536 topics with 4 261 224 works. Additional to the primary topics matching the now created list, each work also has to meet the criteria defined in 3.1.3.

Threshold θ_T	# Topics	Topics Loss (%)	# Works	Works Loss (%)
0.100	4516	0.00 %	28 573 425	0.00 %
0.200	4100	9.21 %	25 698 915	10.06 %
0.300	2125	52.95 %	13 439 863	52.96 %
0.400	563	87.53 %	4 261 224	85.09 %

Table 4.3: Selection of topic-keyword thresholds with number of topics and works where $\sigma_{\max} \geq \theta_T$.

[5] `https://geoservice.dlr.de/eoc/ogc/stac/v1/`
[6] `https://docs.openalex.org/download-all-data/openalex-snapshot`
[7] Amazon object storage, `https://aws.amazon.com/s3/`
[8] `https://docs.openalex.org/api-entities/topics`

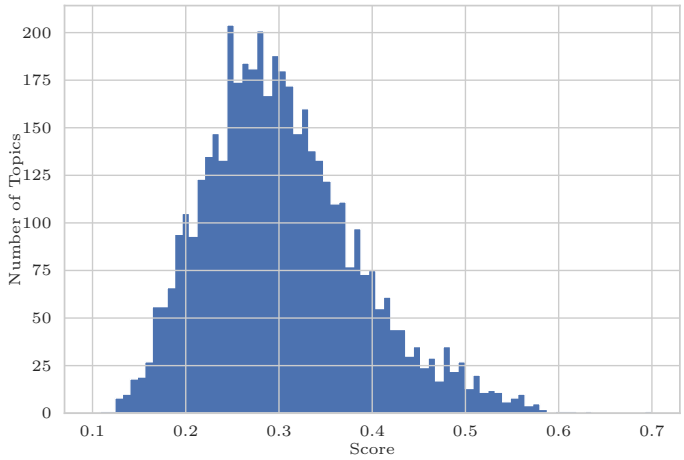

Figure 4.1: Distribution of highest keyword scores of OpenAlex topics.

Work Filtering The topic filtering already reduced the number of works by a lot, but there is no guarantee that every work belonging to an EO related topic is actually relevant to EO. To address this issue further filtering needs to be done. With the amount of works left it is feasible to tag every remaining work and decide in the same manner as with the topics for a threshold. After plotting and manual probing the threshold θ_W is set to 0.375, leaving 2 105 943 works.

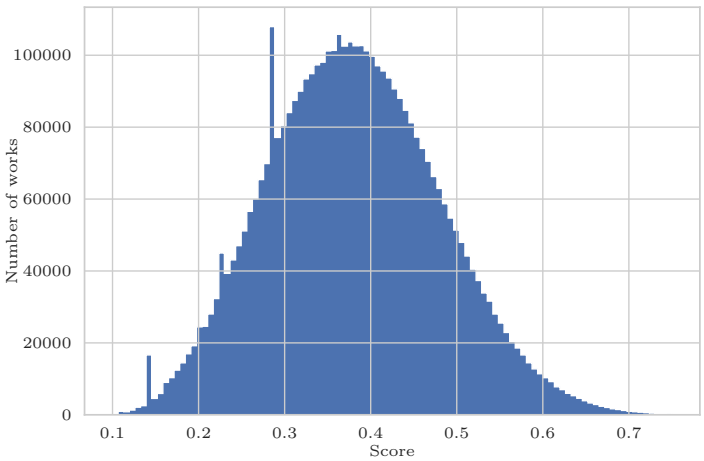

Figure 4.2: Distribution of highest keyword scores of filtered OpenAlex works.

Due to faulty processing not all works marked as English are actually English. Additional language detection using `lingua-py`[9] was done and removed 696 works with differing abstract language and an additional 26 783 works with differing title language. In the same iteration 5195 more works were discarded where no title existed.

[9] `https://github.com/pemistahl/lingua-py`

Threshold θ_W	# Works	Works Loss (%)
0.100	4 164 369	0.00 %
0.200	4 037 355	3.05 %
0.300	3 212 787	22.85 %
0.350	2 502 090	39.92 %
0.375	2 105 943	49.43 %
0.400	1 711 957	58.89 %

Table 4.4: Selection of work-keyword thresholds with number of works where $\sigma_{\max} \geq \theta_W$.

4.1.4 PANGAEA

Accessing PANGAEA datasets is a bit more challenging. There are APIs, but they are either unreliable or very limited. Data was subsequently crawled using a modified version of the *Opensearch Fill In Crawls* crawler[10], resulting in `.warc.gz`[11] files containing the most recent 10 000 datasets. Due to mentioned API limitations it was not possible to acquire more data, although in the future this limitation could be tackled with direct support from the PANGAEA team.

Parsing all files included extracting all relevant metadata (as specified in fig. 3.1) and assigning science keywords to all abstracts. As all data published by PANGAEA is domain specific there is no need to filter based on keyword score, although many datasets (∼90 %) had to be discarded due to the lack of an abstract. Without an abstract there can be no keywords assigned, which prevents the context expansion as specified in section 3.3.

4.2 Knowledge Graph

The knowledge graph is created using the Corpus Annotation Graph Builder (CAG) framework [26]. Built on top of ArangoDB[12], CAG provides methods to create a graph with multiple sources of origin (OpenAlex, ...) and update an existing graph with additional or updated nodes and annotations, albeit annotations will not be used in this work.

In accordance with the ontology defined in fig. 3.1 node and edge wrapper classes are defined. With the extracted data instances, classes are created and then inserted in the graph via CAG. Science keyword nodes are only inserted if connected to a dataset or publication, preventing the insertion of isolated nodes.

Disregarding nodes and edges related to the structure of graphs created by CAG, the graph consists of 2 075 599 nodes and 41 159 409 edges.

Node Type	# Nodes	Source
`EarthObservationDataset`	47 883	OpenAlex, PANGAEA, EOC Geoservice
`EarthObservationPublication`	2 021 267	OpenAlex
`Author`	2850	OpenAlex, PANGAEA
`ScienceKeyword`	3599	NASA
Total	2 075 599	

Table 4.5: Knowledge Graph node composition.

[10] `https://opencode.it4i.eu/openwebsearcheu-public/opensearch-fill-in-crawls`
[11] `https://iso.org/standard/68004.html`
[12] `https://arangodb.com`

Edge Type	# Edges	Source
`HasAuthor`	11 871 165	OpenAlex, PANGAEA
`HasKeyword`	13 238 018	OpenAlex, PANGAEA, EOC Geoservice, NASA
`IsReferencedBy`	13 766 993	OpenAlex
`IsRelatedTo`	2 283 233	OpenAlex
Total	41 159 409	

Table 4.6: Knowledge Graph edge composition.

The edges `IsReferencedBy` and `IsRelatedTo` are exclusive to OpenAlex and not further used for the retrieval process outlined in section 4.3. Future work might use these relations to further enhance the RAG system.

4.3 RAG System

The RAG system implements multiple steps as it is responsible for everything that happens between the graph and the user. It (1) retrieves a subgraph from the database, (2) narrows down the documents to a few important chunks, (3) generates the actual answer and interacts with the user.

(1) Subgraph Retrieval To generate an answer it is imperative to have a factual basis. This basis is scattered across the knowledge graph and needs to be fetched. All titles and abstracts of all datasets and publications are full-text indexed, which allows retrieval using `arangosearch`[13]. Since the user's question can be formatted in any way, the LLM is firstly tasked to create a query to be run against the database.

```
Convert the following question into an optimized search query. Do not
    include any SQL phrases, just plain text. Regardless of the language
    of the question, ALWAYS create an English query.

<question>
    {question}
</question>

Only print the query itself. Do not elaborate on why it is the way it is.
```

Figure 4.3: Query generation prompt.

With this generated query, an AQL (ArangoDB Query Language) is run against the database, returning the best matching documents K, ranked using BM25[14]. These documents are then extended to their associated top science keywords Φ_κ. Furthermore, each science keyword $\varphi \in \Phi_\kappa$ gets extended to its top documents Ψ_φ. This established subgraph constitutes the *broad context*.

[13] ArangoDB implementation of a full-text search, `https://docs.arangodb.com/stable/index-and-search/arangosearch/`

[14] Okapi BM25 ranking function (BM = best match)

(2) Establishing the narrow context Providing the whole subgraph to the LLM has proven to be a sub-optimal solution. The first challenge are the context-window sizes of the LLMs, which limit the amount of tokens[15] an LLM can take as input. Although new LLMs get bigger and bigger context-windows, their attention is not equal across the whole context, tending to ignore the middle part [28]. A big improvement can be achieved using a vector database as intermediate step, drastically cutting down the context passed to the LLM.

In a first step the retrieved documents are split, embedded and then added to the vector database. Sentence embeddings are vectors that represent the semantic context of a text. These vectors are then used to create an ephemeral vector storage.

The same query used in the AQL is now used again for a `max_marginal_relevance_search`[16] on the vector storage to retrieve the most relevant chunks. These document chunks make up the *narrow context*, which is then fed into the LLM.

(3) Answer Generation In the final step the generated narrow context is added into the prompt template (see fig. 4.4). With the gathered information the model can finally answer the user's question.

The implementation does not require any specific way to interact with and offers a public interface method `ask(self, question: str) -> str`, which takes any question and returns the generated answer. Additionally, an interactive option is available (see fig. 4.5).

[15] Tokens are a unit of size measurement and can be approximated by using the length of a given text.

[16] `https://python.langchain.com/api_reference/chroma/vectorstores/langchain_chroma.vectorstores.Chroma.html#langchain_chroma.vectorstores.Chroma.max_marginal_relevance_search`

```
### Context:

The following context was extracted from a scientific Graph Database and
    contains Earth Observation (EO)-related information:

- A list of publications (titles, abstracts and sources) that match the
    user query.
- A list of scientific keywords ('name'), along with their 'descriptions'
    and 'references' (where the keyword originate from), based on the
    NASA GCMD taxonomy.
- Links between publications and scientific keywords based on their
    content relevance ('secondary_nodes').

Use this context to answer the question, if relevant.

The documents in the provided context have metadata, including title or
    name and source (uri) or reference. Only use this metadata, ignore
    references / citations in the 'page_content' field. If you use a
    document cite it in IEEE style. Mentioning the metadata (title / name
    / source / reference) is ONLY allowed in the 'Reference' section
    (bibliography). If you use a document multiple times include it only
    once in 'References'.

<context>
    {context}
</context>

### Query:

{question}

### Answer

[<answer>]

References:
[<number>] [<title / name>], [<uri / reference>]
```

Figure 4.4: System prompt used to generate RAG answers.

```
User: How does ocean circulation influence the distribution of chemical
    constituents in seawater?

RAG: Ocean circulation significantly influences the distribution of
    chemical constituents in seawater through various processes. Ocean
    currents transport water masses over long distances, distributing
    dissolved chemicals and nutrients across different regions. This
    movement can lead to the mixing of water with different chemical
    compositions, altering the concentrations of trace metals and other
    elements.

The redistribution of these chemical components is crucial for
    understanding oceanic biogeochemical cycles and the overall health of
    marine ecosystems. For instance, the movement of water masses can
    transport nutrients from nutrient-rich areas to nutrient-poor
    regions, supporting phytoplankton growth and influencing carbon
    cycling [1].

References:
[1] A fresh look at element distribution in the North Pacific Ocean,
    https://openalex.org/W3046450536
```

Figure 4.5: Example of an interaction using the LLM `mistral-small:24b-instruct-2501-q8_0` to generate a RAG answer.

5 Evaluation

This chapter elaborates on the evaluation approach, settings the samples were generated with and results using RAG triad evaluation. Evaluation was done automatically employing LLM-as-a-judge [29], rating multiple question-answer quality metrics including the context as reference.

5.1 Approach

Question Generation 70 Questions[1] were generated using zero-shot prompting with the LLM `gpt-4o`[2] by selecting random topics[3] and varying the intent of the question (see fig. 5.1).[4] This approach ensures questions to be domain specific and allows for improvements in question diversity. The intent categories and descriptions with examples are as follows:

- **Exploratory**: Asking for discovery of new concepts without any expectations (e.g., *„How do variations in solar activity influence the dynamics of the Earth's ionosphere and magnetosphere over different phases of the solar cycle?“*)
- **Comparative**: Asking for comparing two scientific concepts, such as methods (e.g., *„How do variations in solar activity during solar maximum and solar minimum differentially impact ionosphere dynamics and radiowave propagation?“*)
- **Descriptive**: Asking for an explanation or description of a concept (e.g., *„What is solar activity and how does it influence the Earth's ionosphere?“*)
- **Causal**: Asking for an explanation or description of a concept (e.g., *„How does solar activity influence the dynamics of the Earth's ionosphere and magnetosphere?“*)
- **Relational**: Asking for relationship between two concepts (e.g., *„What is the relationship between solar activity levels and variations in ionosphere dynamics, particularly during solar maximum and minimum phases?“*)

LLM-as-a-Judge Evaluation was done in a fully automatic approach using `gpt-4o-mini-2024-07-18`[5] as judge. Answers were rated on a scale of 1 to 5 based on the three criteria established by Saad-Falcon et al. [8]:

- **Context Relevance**: How relevant is the retrieved context to the query?
- **Answer Relevance**: How relevant is the generated answer to the query?
- **Groundedness**: How faithful is the generated answer to the retrieved context?

The LLM was provided a prompt[6] (see fig. 5.2) including the user's question (`query`), the context retrieved from the vector storage (`context`) and the answer generated (`answer`).

[1] Available at `https://zenodo.org/records/16421449`
[2] `https://platform.openai.com/docs/models/gpt-4o`
[3] Science keywords of the level *topic*, e.g. atmosphere.
[4] Implementation of this question generator was executed by El Baff.
[5] `https://platform.openai.com/docs/models/gpt-4o-mini`
[6] The prompt used for RAG triad was created with the help of `gpt-4` on 2025-07-30 09:15 CET

```
### 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 explanation-just the question.
This question will be used to evaluate the performance of a scientific QA
    system under Retrieval-Augmented Generation (RAG) and zero-shot
    prompting conditions.

### Question:
```

Figure 5.1: Prompt used to generate domain specific test questions.

```
Act as an impartial RAG evaluator. Your task is to assess the quality of
    a Retrieval-Augmented Generation (RAG) system across three
    dimensions:

1. Context Relevance: Does the retrieved context help answer the query?
2. Groundedness: Is the answer strictly based on the retrieved context?
3. Answer Relevance: Does the answer clearly and completely address the
    query?

Scoring scale (string between "1" and "5"):
"1": None - irrelevant or unsupported
"2": Low - weakly related or grounded
"3": Medium - somewhat relevant/grounded
"4": High - mostly correct and grounded
"5": Perfect - fully relevant, accurate, and supported

Think step by step for each score before deciding the final number. Only
base your judgments on the provided CONTEXT and ANSWER.

# QUERY
{query}

# CONTEXT
{context}

# ANSWER
{answer}

## OUTPUT FORMAT:
Context relevance: [<REASONING>] - Score: [<SCORE>]
Answer relevance: [<REASONING>] - Score: [<SCORE>]
Groundedness: [<REASONING>] - Score: [<SCORE>]
```

Figure 5.2: RAG triad evaluation prompt.

5.2 Settings

Retrieval Configuration The retrieval parameters deemed to yield the best results are as follows: For a query q get the 16 best matching documents $\kappa \in K$ from the knowledge graph. Get for each document κ the top scoring 3 science keywords $\varphi \in \Phi_\kappa$ with a score $\sigma_{\max} \geq 0.3$, and for each keyword $\varphi \in \Phi_\kappa$, get the 2 highest scoring documents $\psi \in \Psi_\varphi$, also with $\sigma_{\max} \geq 0.3$.

After the retrieval of the broad context, the documents get recursively split and chunked. Chunking breaks a piece of text into smaller segments, allowing easier processing for LLMs. The two main parameters are *chunk size*, specifying the maximum number of tokens per segment, and *chunk overlap*, describing the number of tokens shared between consecutive chunks. The parameters chosen are a chunk size of 256 and a chunk overlap of 64. These chunks are then embedded using the model `mistral-embed`[7] and saved in a Chroma[8] vector database. Given the query q, the top 4 document chunks are retrieved and passed to the LLM.

Model Configurations All 70 questions were answered by the LLMs `mistral-small:24b-instruct-2501-q8_0`[9] and `llama3.3:latest`[10]. Both models are open-weight[11] but vary in parameter size, whereas mistral small has24 billion and llama3.3 70 billion parameters. Each model answered all questions with the RAG approach and in a zero-shot manner.

5.3 Results

The results show the RAG triad evaluation results with RAG and zero-shot results for mistral small (fig. 5.3) and llama3.3 (fig. 5.4). Although the zero-shot answers are technically unrelated to the retrieved context, due to a lack of a ground truth, they were also evaluated with the narrow context in mind.

A first observation is, that the zero-shot answers always score high in answer-relevance. This can be explained with the LLMs response closely aligning with the question. Manual inspection reveals long elaborated answers that are not always factually grounded. RAG answers in contrast stick to the context they are provided with and thus do not elaborate further than the scope of the information enables them to do.

Low scores of context-relevance are a direct result of the limited data stored in the knowledge graph. Although more than 2 million documents are included, they are not sufficient to answer every possible question, a limitation not present with zero-shot. High scores of context-relevance on the other hand (mostly) lead to high scores in both answer-relevance and groundedness, surpassing the zero-shot results.

Another important observation is low performance of llama3.3 compare to mistral small, both in RAG and zero-shot, but especially in the RAG evaluation. There is no definite explanation for these results, although architectural differences might provide some hints.

[7] `https://docs.mistral.ai/capabilities/embeddings/text_embeddings/`
[8] `https://trychroma.com/`
[9] `https://ollama.com/library/mistral-small:24b-instruct-2501-q8_0`
[10] `https://ollama.com/library/llama3.3`
[11] Open-weight models can be downloaded freely and deployed locally.

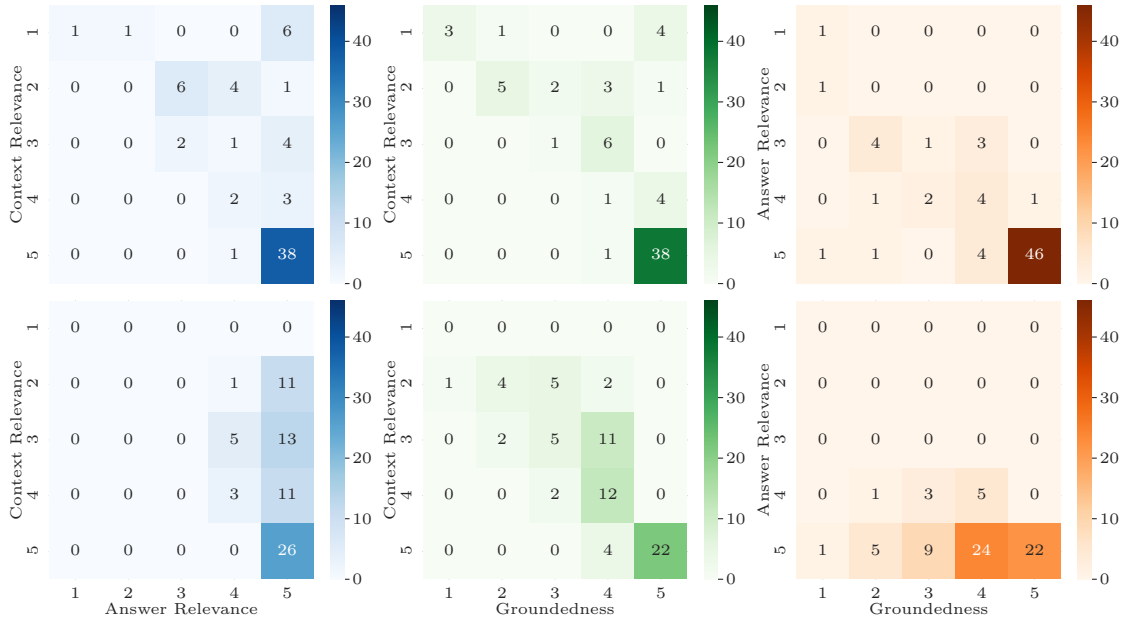

Figure 5.3: RAG triad evaluation for RAG (top) and zero-shot (bottom) answers generated by `mistral-small:24b-instruct-2501-q8_0`.

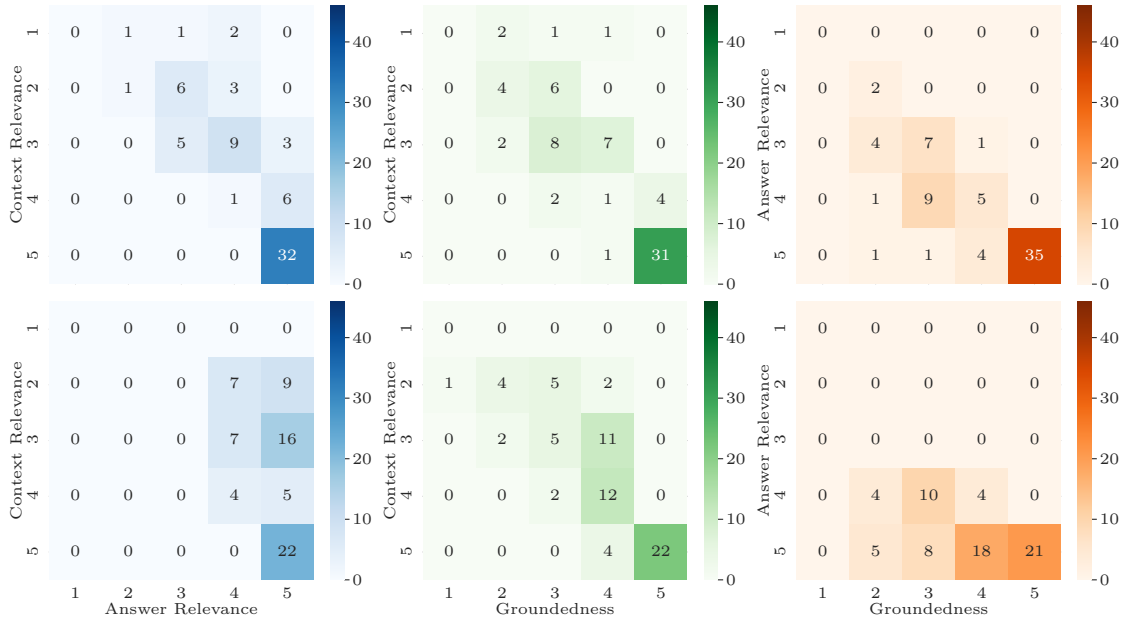

Figure 5.4: RAG triad evaluation for RAG (top) and zero-shot (bottom) answers generated by `llama3.3:latest`.

6 Conclusion

This thesis investigated the question of „How can a knowledge graph-based RAG model improve the accuracy and relevance of responses compared to a zero-shot language model, in the context of the domain of Earth Observation“. A design including multiple domain specific data sources represented in a semantically connected knowledge graph and a RAG system including a two-step retrieval process was presented.

The Implementation detailed data acquisition and processing from the data providers OpenAlex, PANGAEA and the EOC Geoservice. Processing included filtering and establishing semantic connections using the TaxoTagger. This tagger and the science keywords taxonomy were presented, allowing assignment of taxonomy concepts to abstracts of the datasets and publications included in the knowledge graph. A RAG system retrieving a broad context and creating a narrow context with the help of vector databases was also presented.

Evaluation used LLM-as-a-judge to rate the generated answers based on a RAG triad consisting of context relevance, answer relevance and groundedness. Results showed that the RAG system surpasses the zero-shot approach in context relevance and groundedness, whilst the zero-shot comparison revealed a close alignment towards the questions resulting in high answer relevance. Manual inspection though revealed a lack of factual correctness and unnecessary elaborations of the zero-shot answers.

6.1 Subsequent Work

Further work by El Baff et al. [30] was able to improve results by implementing a two-step generation RAG approach. Instead of generating an answer directly from the retrieved context the LLM first generates a zero-shot answer and then refines it with the retrieved context. This method results in the new *2rag* approach outperforming both zero-shot and single generation RAG.

6.2 Future Work

Future work can increase the system's capabilities by extending the knowledge graph and include web search data in the form of Open Web Index shards [31]. Usage of *IsRelatedTo* and *IsReferencedBy* relations to establish the broad context can also be explored. Additionally, human based evaluation is recommended.

An error that needs to be addressed is related to the subgraph retrieval outlined in section 4.3. The current implementation lexically fetches unique documents and creates a tree structure with science keywords and more documents. There is no check for duplicates, so if multiple documents got the same science keyword, the duplicates get inserted multiple times. Although the vector storage prohibits duplicate chunks by checking hash collisions, duplicates reduce the number of unique nodes that are available to the LLM, potentially worsening performance.

Acknowledgments

This work is part of the OpenWebSearch.eu project. The OpenWebSearch.eu project is funded by the EU under the GA 101 070 014 and I thank the EU for their support.

I want to especially thank my supervisor Roxanne El Baff for her guidance, support and expertise throughout the course of this thesis. Her insights and encouragement have been invaluable in shaping this work.

Bibliography

[1] Y. R. Nedumov and S. D. Kuznetsov, „Exploratory search for scientific articles“, *Programming and Computer Software*, vol. 45, no. 7, pp. 405–416, Dec. 1, 2019. DOI: 10.1134/S0361768819070089. [Online]. Available: https://doi.org/10.1134/S0361768819070089 (visited on 08/15/2025).

[2] Y. Zheng, S. Sun, L. Qiu, *et al.*, „OpenResearcher: Unleashing AI for accelerated scientific research“, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, D. I. Hernandez Farias, T. Hope, and M. Li, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 209–218. DOI: 10.18653/v1/2024.emnlp-demo.22. [Online]. Available: https://aclanthology.org/2024.emnlp-demo.22/ (visited on 08/15/2025).

[3] Y. Ma, Z. Gou, J. Hao, *et al.*, „SciAgent: Tool-augmented language models for scientific reasoning“, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 15 701–15 736. DOI: 10.18653/v1/2024.emnlp-main.880. [Online]. Available: https://aclanthology.org/2024.emnlp-main.880/ (visited on 08/15/2025).

[4] Z. Xu, S. Jain, and M. Kankanhalli, *Hallucination is inevitable: An innate limitation of large language models*, 2025. arXiv: 2401.11817 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2401.11817 (visited on 08/13/2025).

[5] L. Thede, K. Roth, M. Bethge, Z. Akata, and T. Hartvigsen, *Understanding the limits of lifelong knowledge editing in llms*, 2025. arXiv: 2503.05683 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2503.05683 (visited on 08/13/2025).

[6] P. Lewis, E. Perez, A. Piktus, *et al.* „Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks“. arXiv: 2005.11401 [cs]. (Apr. 12, 2021), [Online]. Available: https://arxiv.org/abs/2005.11401 (visited on 07/07/2025), pre-published.

[7] A. Hogan, E. Blomqvist, M. Cochez, *et al.*, „Knowledge Graphs“, *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37, May 31, 2022, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3447772. arXiv: 2003.02320 [cs]. [Online]. Available: https://arxiv.org/abs/2003.02320 (visited on 07/07/2025).

[8] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, *Ares: An automated evaluation framework for retrieval-augmented generation systems*, 2024. arXiv: 2311.09476 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2311.09476 (visited on 08/13/2025).

[9] P. Shojaee, I. Mirzadeh, K. Alizadeh, M. Horton, S. Bengio, and M. Farajtabar, *The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity*, 2025. [Online]. Available: https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf (visited on 07/22/2025).

[10] A. Vaswani, N. Shazeer, N. Parmar, *et al.* „Attention Is All You Need“. arXiv: 1706.03762 [cs]. (Aug. 2, 2023), [Online]. Available: https://arxiv.org/abs/1706.03762 (visited on 07/10/2025), pre-published.

[11] X. Ren, P. Zhou, X. Meng, *et al.* „PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing“. arXiv: 2303.10845 [cs]. (Mar. 20, 2023), [Online]. Available: https://arxiv.org/abs/2303.10845 (visited on 07/10/2025), pre-published.

[12] S. Minaee, T. Mikolov, N. Nikzad, *et al.* „Large Language Models: A Survey“. arXiv: 2402.06196 [cs]. (Mar. 23, 2025), [Online]. Available: https://arxiv.org/abs/2402.06196 (visited on 07/10/2025), pre-published.

[13] T. B. Brown, B. Mann, N. Ryder, *et al.* „Language Models are Few-Shot Learners“. arXiv: 2005.14165 [cs]. (Jul. 22, 2020), [Online]. Available: https://arxiv.org/abs/2005.14165 (visited on 07/07/2025), pre-published.

[14] F. Liu, N. AlDahoul, G. Eady, Y. Zaki, and T. Rahwan, *Self-reflection makes large language models safer, less biased, and ideologically neutral*, 2025. arXiv: 2406.10400 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.10400 (visited on 08/04/2025).

[15] Wikimedia Foundation, *Wikipedia logo 2.0, only logo and wordmark 'Wikipedia'*, Jan. 29, 2018. [Online]. Available: https://commons.wikimedia.org/wiki/File:Wikipedia-logo-v2-wordmark.svg (visited on 07/11/2025).

[16] Wikipedia contributors. „Chancellor of germany“. (2025), [Online]. Available: https://en.wikipedia.org/wiki/Chancellor_of_Germany (visited on 08/16/2025).

[17] European Space Agency. „Newcomers Earth Observation Guide“. (Aug. 11, 2020), [Online]. Available: https://business.esa.int/newcomers-earth-observation-guide (visited on 07/07/2025).

[18] N. Raghuvanshi, A. Johri, M. Saxena, and P. Rout, *Space based surveillance and reconnaissance technologies*, Mar. 2025. [Online]. Available: https://www.researchgate.net/publication/391978297_Space_Based_Surveillance_and_Reconnaissance_Technologies (visited on 08/16/2025).

[19] M. F. McCabe, M. Rodell, D. E. Alsdorf, *et al.*, „The future of earth observation in hydrology“, *Hydrology and Earth System Sciences*, vol. 21, no. 7, pp. 3879–3914, Jul. 2017, ISSN: 1607-7938. DOI: 10.5194/hess-21-3879-2017. [Online]. Available: http://dx.doi.org/10.5194/hess-21-3879-2017 (visited on 08/18/2025).

[20] O. Bensch, L. Bensch, T. Nilsson, *et al.*, „AI Assistants for Spaceflight Procedures: Combining Generative Pre-Trained Transformer and Retrieval-Augmented Generation on Knowledge Graphs With Augmented Reality Cues“, in *SPAICE Conference 2024*, Harwell, UK, Sep. 2024. (visited on 07/04/2025).

[21] B. VanGundy, N. Phojanamongkolkij, B. Brown, R. Polavarapu, and J. Bonner, „Requirement Discovery Using Embedded Knowledge Graph With ChatGPT - Poster“. [Online]. Available: https://ntrs.nasa.gov/citations/20240004949 (visited on 07/04/2025).

[22] C. H. Kao, W. Zhao, S. Revankar, *et al.* „Towards LLM Agents for Earth Observation“. arXiv: 2504.12110 [cs]. (Apr. 16, 2025), [Online]. Available: https://arxiv.org/abs/2504.12110 (visited on 07/04/2025), pre-published.

[23] J. Honeder, „Bridging science and web: An open search application for earth observation and environmental research“, M.S. thesis, Graz University of Technology, Graz, May 2024, 171 pp. [Online]. Available: https://doi.org/10.3217/1p3yn-gm455 (visited on 05/23/2025).

[24] J. Priem, H. Piwowar, and R. Orr. „OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts“. arXiv: 2205.01833 [cs]. (Jun. 17, 2022), [Online]. Available: https://arxiv.org/abs/2205.01833 (visited on 05/27/2025), pre-published.

[25] J. Felden, L. Möller, U. Schindler, *et al.*, „PANGAEA - Data Publisher for Earth & Environmental Science“, *Scientific Data*, vol. 10, no. 1, p. 347, Jun. 2, 2023, ISSN: 2052-4463. DOI: 10.1038/s41597-023-02269-x. [Online]. Available: https://www.nature.com/articles/s41597-023-02269-x (visited on 05/27/2025).

[26] R. El Baff, T. Hecking, A. Hamm, J. W. Korte, and S. Bartsch, *Corpus Annotation Graph Builder (CAG): An Architectural Framework to Build and Annotate a Multi-source Graph*, version v1.6.0, Zenodo, Dec. 7, 2023. DOI: 10.5281/zenodo.10285155. [Online]. Available: https://zenodo.org/records/10285155 (visited on 07/07/2025).

[27] Global Change Master Directory (GCMD). „Gcmd keywords, version 21.6“. GCMD Keyword Forum Page. (2025), [Online]. Available: https://forum.earthdata.nasa.gov/app.php/tag/GCMD+Keywords (visited on 07/28/2025).

[28] N. F. Liu, K. Lin, J. Hewitt, *et al.*, *Lost in the middle: How language models use long contexts*, 2023. arXiv: 2307.03172 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.03172 (visited on 08/12/2025).

[29] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.* „Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena“. arXiv: 2306.05685 [cs]. (Dec. 24, 2023), [Online]. Available: https://arxiv.org/abs/2306.05685 (visited on 07/15/2025), pre-published.

[30] R. El Baff, B. Schluckebier, and T. Hecking, „Knowledge graph-enhanced retrieval-augmented generation for earth observation data“, Submitted to DARES'25 workshop on July 29, 2025.

[31] M. Granitzer, S. Voigt, N. A. Fathima, *et al.*, „Impact and development of an open web index for open web search“, *Journal of the Association for Information Science and Technology*, vol. 75, no. 5, pp. 512–520, 2024. DOI: https://doi.org/10.1002/asi.24818. eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24818. [Online]. Available: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24818 (visited on 08/07/2025).