Building a System for Text Information Extraction from Georock Research Papers

Mohd Uwaish

University of Göttingen

Abstract—Efficient retrieval of geochemical data remains challenging due to its heterogeneous nature, encompassing elemental compositions, isotopic analyses, mineralogical datasets, and computational models. These datasets require specialized contextual understanding, such as normalization in rare earth element (REE) analysis or mass spectrometry protocols for stable isotope interpretation. Traditional methods struggle with semantic variability, unstructured formats, and context-aware extraction, limiting efficient knowledge retrieval.

This study introduces Hybrid Retrieval-Augmented Generation (HybridRAG) to enhance knowledge extraction from Georock research papers. HybridRAG integrates VectorRAG, leveraging dense embeddings for semantic retrieval, and GraphRAG, utilizing entity-relationship modeling for structured knowledge extraction. A query stratification mechanism classifies queries into explicit fact retrieval, implicit reasoning, and rationale-driven explanations, dynamically applying retrieval strategies to improve contextual relevance.

HybridRAG is evaluated against a baseline RAG system that employs a vector similarity search retrieval mechanism. Performance assessment is conducted using Response Quality Metrics, including Factual Correctness, Semantic Similarity, BLEU, and ROUGE, which help determine the accuracy, coherence, and relevance of generated responses. These metrics identify how well HybridRAG aligns with reference answers, preserves semantic meaning, and maintains textual similarity. Retrieval-Based Metrics, such as Context Precision, Context Recall, Entity Recall, Noise Sensitivity, and Faithfulness, measure the system's ability to retrieve complete and relevant knowledge while minimizing retrieval errors and ensuring consistency between retrieved contexts and generated outputs. These evaluations provide insights into HybridRAG's effectiveness in retrieving high-quality information and reducing misinformation.

Results demonstrate that HybridRAG outperforms baseline RAG in complex queries by leveraging structured knowledge retrieval. It retrieves more precise and relevant contexts while reducing retrieval noise, particularly excelling in implicit and rationale-driven queries where structured knowledge enhances response quality. These findings highlight the advantages of hybrid retrieval strategies in geochemical data extraction. By combining vector- and graph-based retrieval, HybridRAG offers a more adaptive and accurate approach to information retrieval in scientific literature. Future work may refine retrieval selection, integrate geoscientific ontologies, and optimize retrieval fusion techniques for improved performance.

Keywords— *Retrieval-Augmented Generation (RAG), HybridRAG, VectorRAG, GraphRAG, Knowledge Extraction, Information Retrieval, Large Language Models (LLMs), Georock Research Papers, Scientific Text Processing, Semantic Retrieval, Knowledge Graphs, Query Stratification, Context-Aware Retrieval*

1. Introduction

Geochemistry research involves analyzing heterogeneous datasets, including elemental compositions, isotopic ratios, mineralogical properties, and experimental measurements. Extracting relevant insights from these diverse and unstructured data sources remains challenging due to semantic variability, domain-specific terminology, and implicit relationships between geochemical parameters. Traditional keyword-based retrieval methods fail to effectively capture these complexities, often leading to inaccurate or incomplete knowledge extraction (Sarmah et al., 2024).

Recent advances in Retrieval-Augmented Generation (RAG) have introduced hybrid approaches that combine vector-based retrieval (VectorRAG) and graph-based retrieval (GraphRAG) for more accurate and context-aware information retrieval (Zhao et al., 2024). However, existing solutions often lack dynamic query stratification, which is crucial for adapting retrieval strategies based on query type. To address this, we propose a HybridRAG system that integrates VectorRAG for capturing semantic similarities and GraphRAG for structured knowledge representation. A query classification mechanism ensures that retrieval methods are dynamically selected based on query intent, enhancing efficiency and contextual accuracy (Li et al., 2025).

Our HybridRAG framework consists of five key components: (1) Query Classification – categorizes queries into fact-based retrieval, reasoning-based retrieval, or explanation-driven retrieval, optimizing retrieval strategy (Zhao et al., 2024); (2) Vector-Based Retrieval (VectorRAG) – uses ChromaDB to retrieve semantically similar documents (Sarmah et al., 2024); (3) Graph-Based Retrieval (GraphRAG) – extracts structured knowledge using Neo4j knowledge graphs, improving interpretability (Khemakhem et al., 2024); (4) HybridRAG Pipeline – merges retrieved contexts and feeds them into GPT-4 LLM for response generation; and (5) Context-Augmented Response Generation – ensures that final responses are faithful to retrieved knowledge while maintaining geochemical interpretability (Ganesh et al., 2024).

This study aims to optimize retrieval efficiency by dynamically selecting VectorRAG or GraphRAG based on query classification; improve contextual accuracy in knowledge extraction by leveraging semantic and structured retrieval methods; ensure interpretability by structuring retrieved responses with geochemistry-specific relationships; and develop a scalable retrieval framework applicable to other scientific disciplines requiring knowledge extraction.

2. Related Work

The field of Retrieval-Augmented Generation (RAG) has evolved to address limitations of traditional LLMs that rely exclusively on parametric knowledge. RAG systems enhance factual accuracy and domain relevance by retrieving external information during inference (Zhao et al., 2024). Initial RAG architectures predominantly employed vector-based retrieval utilizing dense passage embeddings, which improved contextual alignment but exhibited limitations in structured knowledge representation and explicit reasoning capabilities (Khemakhem et al., 2024).

Vector-based retrieval (VectorRAG) systems leverage embedding models such as BERT and Sentence Transformers to encode semantically meaningful representations in high-dimensional vector spaces. These representations enable efficient similarity search through approximate nearest neighbor algorithms implemented in platforms like ChromaDB and FAISS (Sarmah et al., 2024). Despite advantages in scalability and semantic relevance, VectorRAG demonstrates inherent constraints in handling structured reasoning tasks that require explicit knowledge representation.

Graph-based retrieval (GraphRAG) architectures utilize Neo4j and RDF-based knowledge graphs to model entity-relationship structures and enable multi-hop reasoning pathways. These systems implement entity-linking and relation extraction models to improve knowledge representation, while ontology-based reasoning enhances retrieval precision through domain-specific relation structuring (Ganesh et al., 2024). GraphRAG demonstrates particular efficacy in geochemistry research, where chemical properties, isotopic compositions, and mineralogical relationships adhere to well-defined structural patterns.

Hybrid retrieval approaches integrate vector-based semantic similarity with graph-based structural reasoning, creating flexible knowledge extraction pipelines. The HybridRAG methodology enables dynamic switching between retrieval mechanisms based on query classification, facilitating semantic context alignment for unstructured queries and structured knowledge retrieval for entity-centric inquiries (Li et al., 2025). Contemporary evaluation frameworks have expanded beyond traditional precision metrics to incorporate Contrastive In-Context Learning (CICL), Context-Aware Retrieval Evaluation (CARE), and Factuality Scoring (FactScore) methodologies.

In domain-specific applications, particularly geochemistry, RAG systems must process complex interdependencies between elemental compositions, isotopic fractionation patterns, and mineralogical transformations. Context-Augmented Retrieval (CAR) techniques optimize retrieval workflows by implementing dynamic partitioning of the information space based on real-time query classification, thereby reducing retrieval noise and enhancing computational efficiency for large scientific datasets (Ganesh et al., 2024).

Despite significant advancements, critical research gaps persist: predominant implementations employ static retrieval strategies rather than dynamic selection methods; current architectures lack efficient hybrid pipelines optimized for specialized scientific domains; and domain-specific reasoning validation metrics remain insufficiently explored (Li et al., 2025). Our HybridRAG system addresses these limitations through a comprehensive query classification-based pipeline that effectively bridges vector-based retrieval efficiency with knowledge graph interpretability for geochemistry research and domain-specific knowledge extraction.

3. Methodology

The proposed HybridRAG system is designed to enhance geochemical knowledge extraction by dynamically selecting VectorRAG or GraphRAG based on query classification. The methodology follows a structured pipeline comprising data preprocessing, knowledge graph extraction, model training, vector-based retrieval, graph-based retrieval, and hybrid response generation.

3.1. Data Preprocessing Pipeline

To facilitate efficient retrieval and response generation, a structured and cleaned dataset is essential. The data preprocessing pipeline consists of the following stages:

3.1.1 Text Extraction

Scientific papers from the Georock Database are processed using Grobid, which converts PDFs into a structured XML format. The extracted XML content is then transformed into JSON format, including metadata, sections, and references for organized storage.

3.1.2 Text Cleaning and Normalization

To enhance data quality, unwanted characters, special symbols, and redundant metadata are removed. Additionally, stopwords are filtered, and stemming and lemmatization techniques are applied to standardize the text.

3.1.3 Chunking and Metadata Assignment

A fixed-window chunking strategy (512 tokens per chunk with a 100-token overlap) is implemented to divide the text into retrievable units. Each chunk is assigned relevant metadata and stored in MongoDB for structured indexing.

3.1.3 Embedding Generation

To enable efficient vector-based retrieval, text chunks are converted into high-dimensional embeddings using OpenAI's text-embedding-ada-002 model. These embeddings are stored in ChromaDB, ensuring fast and accurate retrieval.

3.2. Knowledge Graph Extraction

Since VectorRAG lacks explicit structured relationships, we extract knowledge graphs from processed text to capture geochemical dependencies.

3.2.1 Chunk Refinement

Extracted chunks are further refined to identify key scientific relationships. The system processes relationships between chemical elements, isotopic compositions, mineralogical data, and experimental measurements.

3.2.2 Triplet Extraction

Using Named Entity Recognition (NER) and relation extraction models, structured triplets are formed in the format:

```
(Head Entity, Relationship, Tail Entity)
(Element A, Has Isotope, Isotopic Ratio)
(Mineral X, Contains, Element Y)
```

Code 1. Triplets Format.

The triplets are stored with metadata to retain context. Knowledge Graph Construction: Extracted triplets are appended to a Neo4j-based knowledge graph. Metadata, such as source documents, references, and experiment details, is linked to nodes in the graph. This step ensures that GraphRAG retrieval can retrieve not just semantic text passages but also structured relationships for reasoning-based queries.

3.3. Query Classification Model Training

To optimize retrieval strategies, user queries are categorized into four distinct types: Explicit Fact Retrieval, Implicit Reasoning, Hidden Rationale, and Interpretable Rationale. The system classifies queries dynamically to determine the most suitable retrieval method.

3.3.1 Dataset Preparation

A synthetic dataset was generated using Claude AI, ensuring a diverse and well-balanced set of queries across the four categories. Example classifications include:

- Explicit Fact Retrieval → "What is the isotopic fractionation of Uranium?"
- Implicit Reasoning → "How does mineral composition affect element diffusion?"
- Hidden Rationale → "Why do certain isotopes exhibit fractionation under pressure?"
- Interpretable Rationale → "What factors influence isotopic fractionation trends?"

The dataset was stored in CSV format, with fields for query text and category labels.

3.3.2 Text Vectorization

Queries were vectorized using TF-IDF with the following parameters:

- max-features = 7000 (selecting the most informative terms)
- ngram-range = (1,3) (capturing single words, bigrams, and trigrams)

This transformation ensures that the classifier captures meaningful patterns in the query text.

3.3.3 Model Training and Evaluation

Multiple classification models were trained, and Logistic Regression with the following hyperparameters outperformed other models:

- C = 50 (regularization strength)
- max-iter = 3000 (ensuring convergence)
- class-weight = balanced (handling class imbalance)

The trained model was saved as a pickle file for real-time query classification.

3.3.4 Classification at Query Time

When a user submits a query, it is classified into one of the four predefined categories, and the corresponding retrieval strategy is applied:

- **Explicit Fact Retrieval** – *Simple Similarity Search*
  - Utilizes **vector-based retrieval** to fetch direct fact-based information.
  - Implemented via **ChromaDB** similarity search.
- **Implicit Reasoning** – *Multi-Hop Retrieval*
  - Uses **MultiQueryRetriever** to retrieve multiple relevant passages for inferential queries.
  - Employs **OpenAI's LLM** to generate multi-hop reasoning responses.
- **Interpretable Rationale** – *Keyword Expansion Retrieval*

  - Expands the query with **LLM-based keyword expansion** to improve search accuracy.
  - Uses **ContextualCompressionRetriever** to refine retrieved documents.
- **Hidden Rationale** – *Hybrid Retrieval*
  - Combines **sparse (keyword-based) and dense (vector-based) retrieval** for complex reasoning.
  - Uses an **LLM-powered agent** to interact with **ChromaDB** and select optimal information sources.

This classification-driven approach ensures that retrieval is **adaptive** and **context-aware**, optimizing information access for geochemical queries.

3.4. Vector-Based Retrieval (VectorRAG)

The **VectorRAG** approach is designed for **fact-based and semantic retrieval** using **vector embeddings** and **FAISS similarity search**.

3.4.1 Query Embedding

The user query is first **preprocessed** to remove unnecessary characters and stopwords. It is then converted into a **high-dimensional vector representation** using **OpenAI's text-embedding-ada-002 model**.

3.4.2 Similarity Search in ChromaDB

The vectorized query is used to perform a **FAISS similarity search** within **ChromaDB**. The system retrieves the **top-k most relevant document chunks** based on vector similarity.

3.4.3 Context Retrieval and Aggregation

Retrieved chunks are **ranked** based on similarity scores. **Duplicate information is filtered**, and the most relevant passages are **merged** into a final retrieved context.

This method ensures **fast, efficient retrieval** of semantically relevant information. However, it **does not capture structured relationships** for reasoning-based queries.

3.5. Graph-Based Retrieval (GraphRAG)

The **GraphRAG** approach enables **structured retrieval** for **reasoning-based queries** by leveraging a **Neo4j knowledge graph** for multi-hop reasoning.

3.5.1 Entity Extraction from Query

The system extracts key **geochemical entities** from the query, such as **isotopes, elements, and minerals**, using **GPT-4 for Named Entity Recognition (NER)**. These extracted entities serve as the starting points for querying the knowledge graph.

3.5.2 Knowledge Graph Query Execution

A **Cypher query** is executed in **Neo4j** to retrieve relevant **subgraphs** based on the extracted entities. The system utilizes **APOC procedures** to expand the subgraph and include relevant relationships and connections within the graph. The query depth is dynamically adjusted depending on the complexity of the reasoning task.

```
MATCH (start {name: $entity})
CALL apoc.path.subgraphAll(start, {
    maxLevel: $depth
})
YIELD nodes, relationships
UNWIND nodes AS node
UNWIND relationships AS rel
RETURN
    node.name AS entity_name,
    labels(node) AS entity_types,
    type(rel) AS relationship_type,
    startNode(rel).name AS start_node,
    endNode(rel).name AS end_node
```

Code 2. CYPHER query to extract the subgraph

3.5.3 Context Retrieval and Graph Expansion

The retrieved **subgraph** provides structured facts that form the basis of the response. Extracted relationships and entities are formatted into structured contextual information. **Redundant information is filtered**, and if necessary, the query scope is **expanded dynamically** to incorporate additional related knowledge from the graph.

Triplets extracted using **GraphRAG** are combined with the retrieved textual information from **VectorRAG** and passed to the **LLM**, enabling the model to generate responses that integrate both structured knowledge and semantic text retrieval. This ensures **factual consistency, multi-hop reasoning, and enriched context generation** while leveraging the strengths of both retrieval methods.

3.6. HybridRAG: Integrating Vector and Graph Retrieval

The **HybridRAG** pipeline integrates **VectorRAG** and **GraphRAG** to enable both **semantic text retrieval** and **structured knowledge retrieval**, ensuring a comprehensive response generation process. By combining these retrieval strategies, the system effectively retrieves factual knowledge while incorporating structured relationships essential for reasoning-based queries.

3.6.1 Merging Retrieved Contexts

For every query, the system retrieves information from both **VectorRAG** and **GraphRAG**. The extracted **text chunks** from VectorRAG provide direct factual insights, while the **knowledge graph responses** from GraphRAG ensure structured reasoning. These two contexts are merged into a unified representation, ensuring that structured facts complement semantic text retrieval.

3.6.2 Final Context Formatting

The combined response is formatted into a structured **context dictionary**, preserving both textual and graph-based knowledge. The response follows the format:

```
{
  "VectorRAG_Context": {
    "0": "Uranium isotopic fractionation in sedimentary
     ↪ environments is primarily influenced by redox
     ↪ conditions.
          Under reducing conditions, uranium
     ↪ precipitates as U(IV) and accumulates in organic-
     ↪ rich sediments.",
    "1": "Rare earth elements (REEs) in hydrothermal
     ↪ fluids show systematic variations that provide
     ↪ insights
          into fluid source, temperature, and
     ↪ mineralization processes.",
    "2": "Strontium isotope ratios in carbonate deposits
     ↪  are widely used as tracers for reconstructing
     ↪ paleoceanographic
          conditions and fluid-rock interactions.",
    "3": "Magmatic differentiation processes are often
     ↪ inferred from the geochemical trends of major and
     ↪  trace elements
          in igneous rock suites.",
    "4": "The Makran subduction zone shows significant
     ↪ geochemical heterogeneity, with volcanic arc
     ↪ magmatism influenced
          by sediment subduction and slab-derived fluid
     ↪ metasomatism."
  },
  "GraphRAG_Context": {
    "0": "Uranium  Under Reducing Conditions
     ↪ Precipitates as U(IV)",
    "1": "REEs  Show Systematic Variations  Hydrothermal
     ↪  Fluids",
    "2": "Strontium Isotopes  Used for
     ↪ Paleoceanographic Reconstruction",
    "3": "Igneous Rock Suite  Displays  Magmatic
     ↪ Differentiation Trends",
    "4": "Makran Subduction Zone  Influenced by  Slab-
     ↪ Derived Fluid Metasomatism"
  }
}
```

Code 3. Context Passed to LLM

3.6.3 Response Generation using GPT-4

The structured context from both **VectorRAG** and **GraphRAG** is

passed to **GPT-4**, where it generates a **context-grounded response**. The system ensures that both factual and structured knowledge are incorporated into the final response, improving interpretability and accuracy. By consistently including both retrieval approaches, HybridRAG guarantees a well-rounded and knowledge-rich response generation process.

This integration of **vector-based semantic retrieval** and **graph-based structured reasoning** ensures that HybridRAG provides **factually accurate, contextually rich, and reasoning-aware** responses. The system always utilizes both retrieval mechanisms in tandem, implementing a comprehensive knowledge retrieval framework that enhances response quality and reasoning depth.

4. Evaluation Metrics

To assess the performance of the proposed HybridRAG system, we evaluate it based on a combination of retrieval-based and response quality metrics.

4.1. Context Precision

Context Precision measures the proportion of relevant chunks retrieved within the retrieved contexts. It is computed as the mean of Precision@K for each chunk in the retrieved context list.

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K}(\text{Precision@k} \times v_k)}{|R_K|} \tag{1}$$

$$\text{Precision@k} = \frac{\text{TP@}k}{\text{TP@}k + \text{FP@}k} \tag{2}$$

where K represents the total number of chunks in the retrieved contexts, $|R_K|$ is the number of relevant items in top K results, and $v_k \in \{0, 1\}$ is a binary indicator of relevance at rank k. TP and FP represent true positives and false positives.

LLM-Based Context Precision: Two variations of this metric are used:

- **Without Reference:** Compares retrieved context with the generated response.
- **With Reference:** Uses both retrieved context and reference answer.

A **higher Precision** indicates highly relevant retrieved documents. A **lower Precision** suggests inclusion of irrelevant contexts.

4.2. Context Recall

Context Recall measures the proportion of reference claims supported by the retrieved context.

$$\text{Context Recall} = \frac{|C_{sup}|}{|C_{ref}|} \tag{3}$$

where $|C_{sup}|$ is the number of supported claims and $|C_{ref}|$ is the total claims in reference.

LLM-Based Context Recall: Evaluates reference information captured in the retrieved context by checking claim support.

Higher Recall values indicate comprehensive retrieval. **Lower recall** suggests missing important claims.

4.3. Context Entities Recall

Context Entities Recall evaluates preservation of named entities from the reference answer.

$$\text{CER} = \frac{|RCE \cap RE|}{|RE|} \tag{4}$$

where RE represents entities in reference answer, and RCE is entities in retrieved context.

Interpretation:

- **Higher CER:** Better retention of key entities.
- **Lower CER:** Missing critical named entities.

4.4. Noise Sensitivity

Noise Sensitivity quantifies incorrect claims due to retrieval noise.

$$\text{NS} = \frac{|C_{incorrect}|}{|C_{total}|} \tag{5}$$

Lower NS indicates more accurate claims. **Higher NS** suggests the model is prone to using misleading contexts.

4.5. Faithfulness

Faithfulness measures factual consistency between generated response and retrieved context.

$$\text{FS} = \frac{|C_{supported}|}{|C_{response}|} \tag{6}$$

Interpretation:

- **Higher FS:** Claims supported by retrieved context.
- **Lower FS:** Contains unsupported/hallucinated claims.

Faithfulness is crucial for ensuring the reliability of the RAG system, especially in scientific applications where factual accuracy is paramount.

4.6. Factual Correctness

Factual Correctness measures how accurately the generated response aligns with the reference answer. It ensures that the system does not introduce misinformation or hallucinated claims. This metric is computed using Precision, Recall, and F1 Score, which assess the factual overlap between the response and reference.

4.6.1. Computing True Positives, False Positives, and False Negatives

$$TP = \text{Number of claims in response that are present in reference} \tag{7}$$

$$FP = \text{Number of claims in response that are not present in reference} \tag{8}$$

$$FN = \text{Number of claims in reference that are not present in response} \tag{9}$$

4.6.2. Precision, Recall, and F1 Score

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

Interpretation:

- A **higher F1 Score** (closer to 1) means the response accurately reflects the reference.
- A **lower F1 Score** (closer to 0) indicates factual inconsistency or hallucinations in the response.

4.7. Semantic Similarity

Semantic Similarity evaluates the meaning-based closeness between the generated response and the reference. It is computed using a bi-encoder model that converts both texts into vector embeddings and calculates their cosine similarity.

$$\text{Semantic Similarity} = \cos(\theta) = \frac{A \cdot G}{\|A\|\|G\|} \tag{13}$$

where:

- A is the embedding of the generated response.
- G is the embedding of the reference answer.
- $A \cdot G$ is the dot product of the two vectors.

Interpretation:

- Higher Semantic Similarity (closer to 1) → The response conveys the same meaning as the reference.
- Lower Semantic Similarity (closer to 0) → The response deviates significantly from the reference in meaning.

4.8. Non-LLM String Similarity

Non-LLM String Similarity assesses the textual resemblance between the generated response and the reference using traditional string distance measures, without relying on language models. It is computed using techniques like Levenshtein Distance, Hamming Distance, and Jaro Similarity.

4.8.1. Levenshtein Similarity

$$1 - \frac{\text{Levenshtein Distance(Response, Reference)}}{\max(\text{len(Response)}, \text{len(Reference)})} \tag{14}$$

$$\frac{\hbar^2}{2m}\nabla^2\Psi + V(\mathbf{r})\Psi = -i\hbar\frac{\partial\Psi}{\partial t} \tag{15}$$

4.8.2. Hamming Similarity

$$\text{Hamming Similarity} = 1 - \frac{\text{Hamming Distance(Response, Reference)}}{\text{len(Reference)}} \tag{16}$$

4.8.3. Jaro Similarity

$$\text{Jaro Similarity} = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \tag{17}$$

where:

- m is the number of matching characters.
- t is the number of transpositions.
- s_1, s_2 are the two compared strings.

4.9. BLEU Score

The BLEU (Bilingual Evaluation Understudy) score measures the similarity between the response and the reference based on n-gram precision and a brevity penalty to prevent overly short responses.

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{18}$$

where:

- p_n is the n-gram precision.
- w_n is the weight assigned to each n-gram.
- BP (brevity penalty) is calculated as:

$$BP = \begin{cases} 1, & \text{if } c > r \\ \exp(1 - r/c), & \text{if } c \leq r \end{cases} \tag{19}$$

where c is the response length and r is the reference length.

4.10. ROUGE Score

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score is used to evaluate the similarity between the generated response and the reference based on n-gram overlap. The default ROUGE score provided by RAGAS was used in this study, without modifications to the ROUGE type.

The ROUGE score is calculated as follows:

$$\frac{\text{Number of overlapping words between response and reference}}{\text{Total words in reference}} \tag{20}$$

Interpretation:

- Higher ROUGE Score (closer to 1) → Indicates that the response contains a high proportion of words from the reference, suggesting better alignment.
- Lower ROUGE Score (closer to 0) → Suggests that the response differs significantly from the reference in terms of word overlap.

5. Comparison of HybridRAG vs. BaselineRAG

The performance of our **HybridRAG** system was compared with the traditional **RAG system**, which follows a **Vector Similarity-Based Retrieval** approach. This traditional RAG system acts as a **baseline** for our system.

5.1. BaselineRAG: Traditional Vector Similarity-Based Retrieval System

The **BaselineRAG** system follows a traditional **retrieval-augmented generation (RAG)** approach, where document retrieval is performed using **vector similarity search**. Given a user query, it retrieves relevant text chunks from a vector database using **cosine similarity** or other distance metrics. The retrieved contexts are then passed to a language model to generate the response.

5.2. Evaluation Data Preparation

To ensure a systematic comparison between HybridRAG and BaselineRAG, evaluation data was carefully prepared before computing the metrics. Two queries were selected from each category, covering Explicit Facts, Implicit Facts, Hidden Rationale, and Interpretable Rationale queries. For each query, a ground truth reference answer was generated using ChatGPT, which served as the baseline for factual correctness and similarity-based metrics. Along with the reference answers, the retrieved contexts and system-generated responses from both HybridRAG and BaselineRAG were collected. These elements, including retrieved contexts, generated responses, and reference answers, were used to compute various evaluation metrics.

5.3. Evaluation Plots

All the metrics were plotted to evaluate the performance of hybridRAG compared to baselineRAG, using both Retrieval Metrics and Response Metrics as our evaluation criteria.

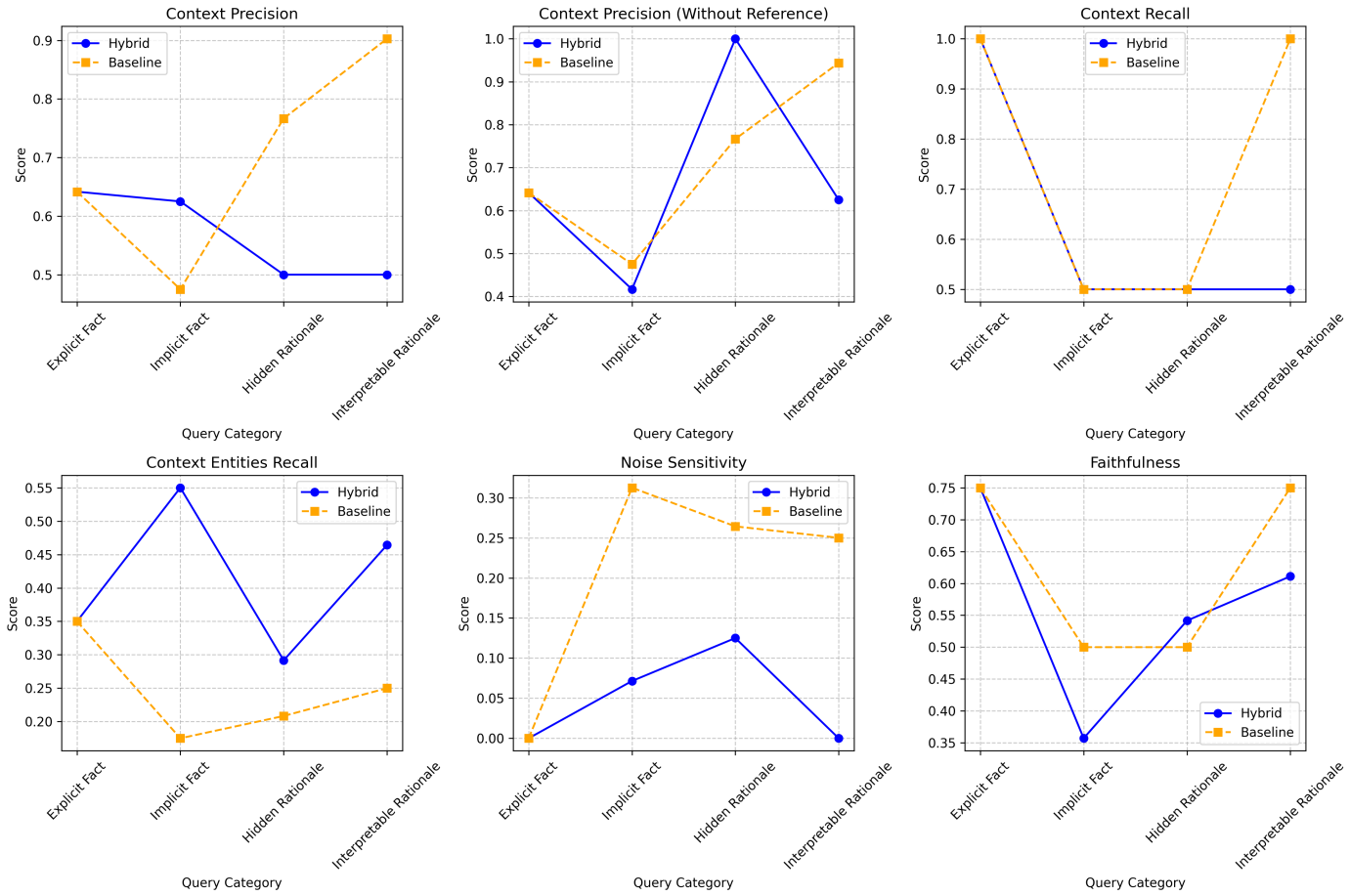
5.3.1. Response Quality Metrics

These metrics evaluate the correctness and quality of the generated responses compared to reference answers. We plotted the following metrics:

- **Factual Correctness**
- **Semantic Similarity**
- **Non-LLM String Similarity**
- **BLEU Score**
- **ROUGE Score**

5.3.2. Retrieval-Based Metrics

Retrieval metrics analyze the effectiveness of retrieving relevant contexts, ensuring that the system provides useful information for response generation. We plotted the following retrieval-based metrics:

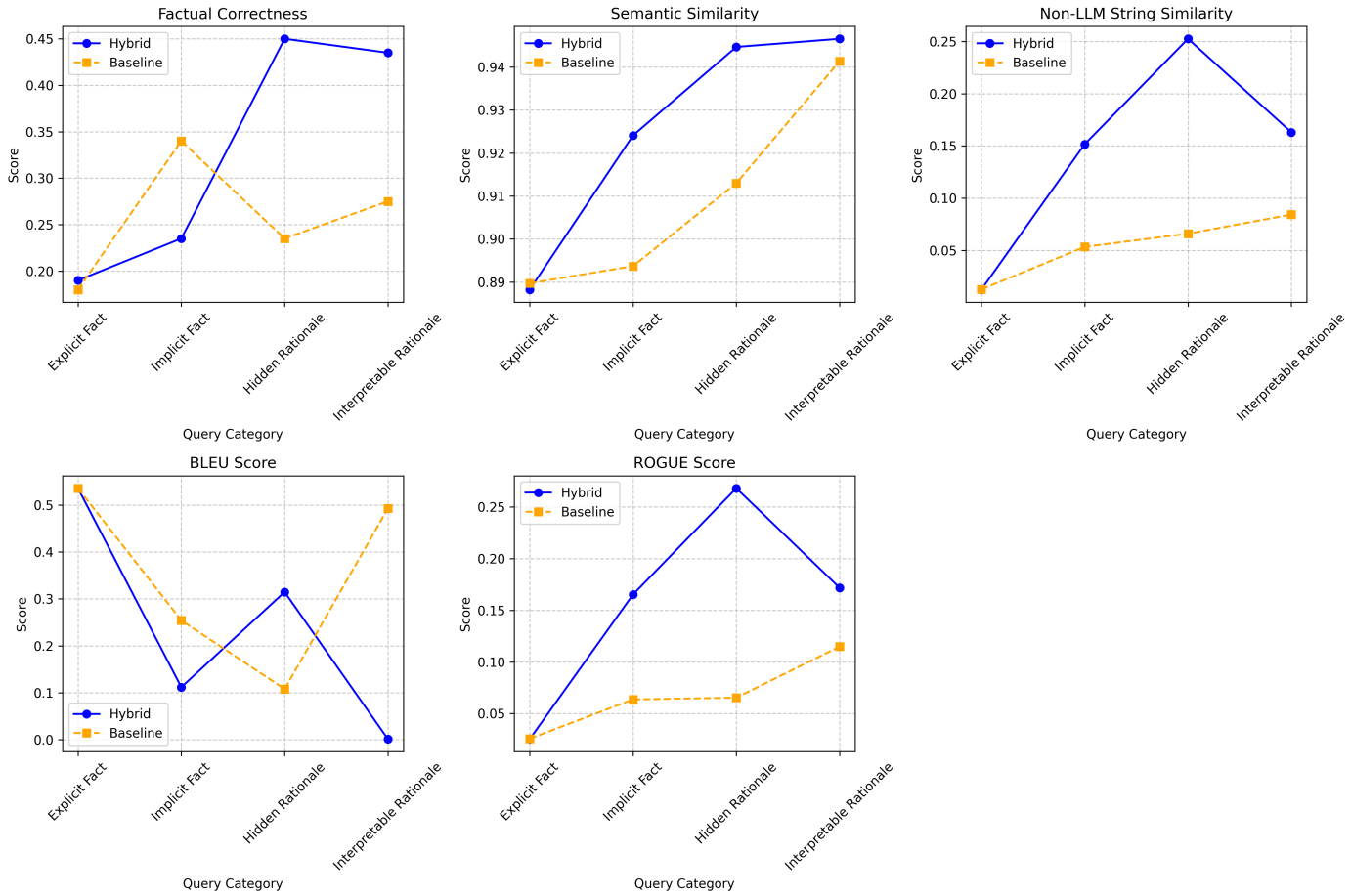


(a) fig: Retrieval-based Metrics

Figure 1. Comparison of HybridRAG and BaselineRAG in Retrieval-Based Metrics: Context Precision, Context Precision (Without Reference), Context Recall, Context Entities Recall, Noise Sensitivity, and Faithfulness.

- **Context Precision**
- **Context Precision (Without Reference)**
- **Context Recall**
- **Context Entities Recall**
- **Noise Sensitivity**
- **Faithfulness**

The following figures illustrate the comparative performance of **HybridRAG** and **BaselineRAG** across different query categories.



(a) fig: Response quality Metrics

Figure 2. Comparison of HybridRAG and BaselineRAG in Response Quality Metrics: Factual Correctness, Semantic Similarity, Non-LLM String Similarity, BLEU Score, and ROUGE Score. **PFGPlots**.

6. Analysis of Results

The comparative evaluation between **HybridRAG** and **BaselineRAG** provides valuable insights into the effectiveness of both retrieval and response generation mechanisms. The analysis of both **retrieval-based metrics** and **response-based metrics** helps determine the strengths and limitations of each approach.

6.1. Response Quality Analysis

The response quality metrics assess the accuracy and fluency of generated responses in comparison to reference answers. The observations from the evaluation plots are as follows:

- **Factual Correctness:** HybridRAG achieves higher factual correctness scores Hidden Rationale and Interpretable query categories, indicating that it produces more factually accurate responses than BaselineRAG.
- **Semantic Similarity:** HybridRAG outperforms BaselineRAG in maintaining semantic consistency with reference answers, especially for more complex queries.
- **Non-LLM String Similarity:** The HybridRAG system exhibits better text-level similarity with reference answers, demonstrating its ability to generate responses that closely match expected outputs.
- **BLEU Score:** Performance fluctuates between both systems, but HybridRAG shows a noticeable improvement in some query categories, indicating better structured responses.
- **ROUGE Score:** HybridRAG achieves higher ROUGE scores, reflecting stronger word overlap between generated responses and reference texts.

Overall, **HybridRAG outperforms BaselineRAG** in response quality by producing more factually correct and semantically aligned answers.

6.2. Retrieval Effectiveness Analysis

Retrieval-based metrics evaluate how well the system retrieves relevant contexts to aid response generation. The key findings are:

- **Context Precision:** BaselineRAG performs better in some cases, especially for explicit facts, but HybridRAG maintains stable precision across query categories.
- **Context Precision (Without Reference):** HybridRAG achieves significantly higher scores, highlighting its ability to retrieve more relevant contexts.
- **Context Recall:** Both systems perform similarly in retrieving relevant contexts, though HybridRAG maintains a slight advantage in some query types.
- **Context Entities Recall:** HybridRAG consistently recalls more entities from the reference, making it more reliable for entity-focused queries.
- **Noise Sensitivity:** HybridRAG has lower noise sensitivity, meaning it introduces fewer incorrect claims in responses.
- **Faithfulness:** HybridRAG achieves better faithfulness scores, ensuring that generated responses align with retrieved contexts more effectively.

These results indicate that **HybridRAG excels in retrieving precise and relevant contexts** while minimizing noise, making it more reliable than BaselineRAG.

6.3. Overall Performance Comparison

- **HybridRAG outperforms BaselineRAG** in factual correctness, semantic similarity, and faithfulness, proving its superiority in response quality.
- **HybridRAG demonstrates stronger retrieval effectiveness,** particularly in precision, recall, and entity recall.
- **BaselineRAG performs competitively in certain retrieval scenarios**, particularly in cases involving explicit fact queries.
- **HybridRAG minimizes noise sensitivity,** ensuring fewer incorrect claims, making it a more reliable choice for knowledge-intensive applications.

Based on these findings, **HybridRAG outperforms the baselineRAG system**, providing more accurate, relevant, and contextually faithful responses while maintaining strong retrieval effectiveness.

7. Conclusion

This study focused on developing a Retrieval-Augmented Generation (RAG) system tailored for the Georock database to enhance knowledge extraction and response generation for geochemistry-related queries. The HybridRAG system integrates a query classification mechanism that dynamically selects between vector-based and graph-based retrieval, ensuring more precise and contextually relevant retrieval, particularly for complex queries requiring structured knowledge representation.

To assess its effectiveness, HybridRAG was compared against BaselineRAG, a traditional RAG system using vector similarity-based retrieval. The results show that HybridRAG achieves higher factual correctness, semantic similarity, and faithfulness, demonstrating its ability to generate more accurate and contextually grounded responses. In retrieval-based metrics, HybridRAG consistently retrieves more relevant contexts with lower noise sensitivity, as reflected in improved Context Precision and Context Entities Recall.

For explicit fact-based queries, both HybridRAG and BaselineRAG exhibit similar performance since HybridRAG also employs vector similarity retrieval for such queries. However, HybridRAG outperforms BaselineRAG in implicit and rationale-driven queries, where its hybrid retrieval approach incorporating graph-based knowledge enhances response quality.

The evaluation methodology, involving a standardized dataset with ChatGPT-generated reference answers, ensured reliable metric computations. The findings highlight that integrating structured knowledge graphs with vector retrieval significantly improves retrieval and response accuracy.

In conclusion, HybridRAG offers significant improvements for complex queries, making it a valuable tool for scientific knowledge extraction. Future work may explore expanding the system with domain-specific ontologies, refining query classification models, and optimizing retrieval strategies to further enhance performance.

References

[1] B. Sarmah, B. Hall, R. Rao, S. Patel, S. Pasquali, and D. Mehta, "HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction," *arXiv*, vol. 2408.04948, 2024.

[2] S. Zhao, Y. Yang, Z. Wang, Z. He, L. Qiu, and L. Qiu, "Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make Your LLMs Use External Data More Wisely," *arXiv*, vol. 2409.14924, 2024.

[3] S. Li, L. Stenzel, C. Eickhoff, and S. A. Bahrainian, "Enhancing Retrieval-Augmented Generation: A Study of Best Practices," *arXiv*, vol. 2501.07391, 2025.

[4] M. Khemakhem, H. E. Rekik, and O. Bouaziz, "Enhancing Technical Knowledge Acquisition with RAG Systems: The TEI Use Case," *HAL Science*, 2024.

[5] S. Ganesh, G. Balakrishnan, and A. Purwar, "Context-augmented Retrieval: A Novel Framework for Fast Information Retrieval Based Response Generation Using Large Language Models," *ResearchGate*, 2024.

[6] RAGAS Team, "RAGAS: A framework for evaluating retrieval-augmented generation," 2023. [Online]. Available: `https://github.com/explodinggradients/ragas`. [Accessed: 10-Mar-2025].