GDAŃSK UNIVERSITY OF TECHNOLOGY

TASK Quarterly 29 (3) 2025

An Analysis of Retrieval-Augmented Generation: A Systematic Review Addressing Architectures, Components, and Evaluation

Adam Ślusarek*, Oskar Wilda, Jakub Wojtalewicz, Jakub Stachowicz, Piotr Wesołowski, Błażej Szutenberg

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Narutowicza 11/12, Gdańsk, 80-233, Poland
*slusarekadam1@gmail.com

https://doi.org/10.34808/tq2025/29.3/a

Abstract

Retrieval-Augmented Generation (RAG) enhances Large Language Models (LLMs) by integrating external retrieval mechanisms to improve factuality and currency. This systematic literature review characterizes current RAG architectures, components, and evaluation practices in peer-reviewed studies published between 2021 and 2025 across IEEE Xplore, Scopus, and Web of Science. Conducted in accordance with the PRISMA guidelines, this review analyzes 41 studies that met the predefined inclusion criteria. Most research addresses Question Answering (QA) and dialogue systems, employing diverse encoders and retrieval optimization methods. Key findings reveal a strong trend toward integrating OpenAI's GPT models, alongside growing adoption of open-source alternatives. Persistent challenges include hallucination control, computational efficiency, and inconsistent evaluation metrics. Despite the potential of RAG, the evidence base is limited by a focus on English-language, high-resource domains. Furthermore, reproducibility is constrained by heterogeneous evaluation standards and a lack of open-access code or datasets. This review maps the RAG research landscape and identifies gaps in standardization, scalability, and application to low-resource languages. The protocol was not prospectively registered, and no funding was received for this work.

Keywords:

Large language models, Retrieval-augmented generation, Systematic literature review

1. Introduction

In recent years, large language models (LLMs) have gained significant popularity. Their applications span across various domains due to their high effectiveness in text generation and question answering [1, 2]. However, even the most advanced models are not without flaws – responses generated by LLMs can be imprecise, outdated, or even factually incorrect [3]. This phenomenon is commonly referred to as hallucination [1, 4].

To address these issues, the retrieval-augmented generation (RAG) approach was developed [5]. RAG combines the advantages of generative language modeling with mechanisms for retrieving information from external knowledge sources [6]. Its architecture assumes that, prior to generating a response, the model employs a retriever to search for the most relevant documents, which are then used as context for the generator [7, 8]. As a result, responses can be more fact-based and grounded in up-to-date knowledge [9, 10].

With the growing adoption of RAG, numerous research studies have emerged that explore its applications, limitations, and technical characteristics [11–13]. This has created a need for a comprehensive review of the existing literature to organize and synthesize current knowledge about RAG.

This paper presents the findings of a systematic literature review (SLR) on RAG. The analysis covers existing research in the context of nine specific research questions (RQ1-RQ9). The aim of this work is to provide a structured and comprehensive overview of the current state of knowledge on RAG architectures and to identify areas for future investigation.

2. Research method

To conduct a reliable and well-structured literature review, the SLR approach was applied. This method is based on established methodological frameworks that ensure the repeatability and transparency of the entire review process. By using this methodology, it is possible to systematically collect, evaluate, classify, and synthesize existing scientific publications related to RAG.

2.1. Research questions

For the purpose of this review, nine specific research questions (RQ1–RQ9) have been defined, covering key aspects related to the RAG architecture. These questions focus on applications, limitations, evaluation methods, utilized components (such as encoders, retrievers, and generative models), as well as the types of knowledge bases and optimization techniques employed. The exact research questions are presented below.

RQ1. What applications of RAG are studied?

RQ2. What are the challenges and limitations in implementing RAG models?

RQ3. How is RAG evaluated in terms of accuracy, efficiency, and scalability?

RQ4. What datasets and benchmarks are commonly used to evaluate RAG models?

RQ5. What query encoders are used in RAG architectures?

RQ6. What retriever search and fusion techniques are used in RAG architectures?

RQ7. What techniques are employed to optimize retrieval in RAG models?

RQ8. Which generative models are integrated into RAG architectures?

RQ9. What types of knowledge bases are used in RAG architectures?

2.2. Search strategy and study selection

This review followed a structured search and selection process to ensure comprehensiveness and reproducibility.

2.2.1 Information sources and search strategy

To identify relevant scientific publications, three well-established bibliographic databases were used: IEEE Xplore, Scopus, and Web of Science. These databases were selected due to their high quality, broad thematic coverage, and the availability of peer-reviewed publications in computer science and engineering. The search was performed uniformly across all three databases, and the final search in each source was completed on March 20, 2025.

In addition to the database searches, backward citation searching was conducted using the reference lists of the studies that met the inclusion criteria. Each reference list was manually reviewed to identify additional potentially relevant publications that were not captured in the initial database search. This manual snowballing process ensured comprehensive coverage of the literature and helped identify supplementary studies aligned with the predefined research questions.

All databases were accessed via their official web interfaces.

2.2.2 Eligibility criteria

Studies were selected based on pre-defined eligibility criteria, structured using the PICO framework [14]:

- **Population / Phenomenon of Interest:** The review considered studies focusing on LLMs and their applications where the core subject was the RAG architecture.
- **Intervention / Core Subject:** The primary subject of the study was required to be the implementation, analysis, evaluation, or theoretical discussion of a RAG system. A RAG system was defined as one comprising at least a *retriever* component (for sourcing external information) and a *generator* component (an LLM for generating responses based on the retrieved context).
- **Context:** Any application domain (e.g., healthcare, law, general QA) was included.
- **Study Design:** Original research articles, empirical evaluations, and systematic literature reviews were included. Conceptual papers and theoretical frameworks were included only if they provided a novel architectural proposition or a synthesis of RAG challenges.

2.2.3 Report characteristics and exclusion criteria

- **Publication Period:** Studies published between January 2021 and March 2025 were included. The year 2021 was chosen as a cutoff because it marks the period of accelerated research and adoption of RAG following the widespread proliferation of powerful transformer-based LLMs, ensuring the review captures the most current and relevant state of the art.
- **Language:** Only articles written in English were considered.
- **Publication Status:** Only peer-reviewed conference papers and journal articles were included. Preprints (e.g., from arXiv) and grey literature were excluded to maintain a baseline of quality and verifiability through the peer-review process.
- **Exclusion Criteria:** Studies were excluded if they: (1) only mentioned "information retrieval" in a general context without a specific focus on the RAG architecture; (2) were tutorials, position papers, or editorials without original research; (3) did not report on any outcomes relevant to the research questions defined in this review.

2.2.4 Grouping studies for synthesis

For the purpose of qualitative and quantitative synthesis, the included studies were grouped according to the themes of the research questions (RQ1–RQ9). This allowed for a structured analysis across the following dimensions:

- **Application Groups:** Based on RQ1, studies were categorized by their primary use case (e.g., Question Answering, Summarization, Domain-Specific applications).
- **Architectural Component Groups:** Based on RQ5–RQ9, studies were grouped by the technical components they employed or analyzed (e.g., types of Encoders, Retrievers, Generative Models, and Knowledge Bases).
- **Evaluation Groups:** Based on RQ3 and RQ4, studies were sorted by the metrics and benchmarks they used for assessment (e.g., studies reporting on Accuracy, Efficiency, or Scalability metrics).

2.3. Keywords and search string

For the purpose of the search and selection process, a set of keywords was identified to reflect the core aspects of the RAG architecture. The selected keywords included: Retrieval-Augmented Generation, RAG, retrieval optimization, challenges, applications, language models, and information retrieval. Based on these keywords, the following query string was constructed:

```
("Retrieval-Augmented Generation")
AND (("applications" OR "use cases" OR
"implementations") AND
("challenges" OR "limitations" OR "barriers" OR "bottlenecks") OR
("retrieval optimization" OR
"retrieval techniques" OR "search efficiency"))
```

2.4. Papers extraction and selection

The publication selection process was carried out in several stages. The first stage involved analyzing the titles and abstracts to preliminarily assess the relevance of the publications to the topic. The second stage included a detailed review of the full texts to evaluate their substantive content and verify compliance with the predefined inclusion and exclusion criteria. The third stage focused on extracting data from the qualified articles into analytical tables, covering aspects such as applications, methods, metrics, generative models, optimization techniques, and limitations. At this stage, the *snowballing* technique was also applied – references cited in the already accepted articles were reviewed to identify additional potentially relevant publications. This process identified 2 additional reports. These were retrieved and assessed for eligibility, but both were ultimately excluded as they did not fully meet the inclusion criteria upon detailed review.

The list of articles was split evenly across three groups consisting of two people each. Each group was responsible for analyzing the articles, reviewing them and

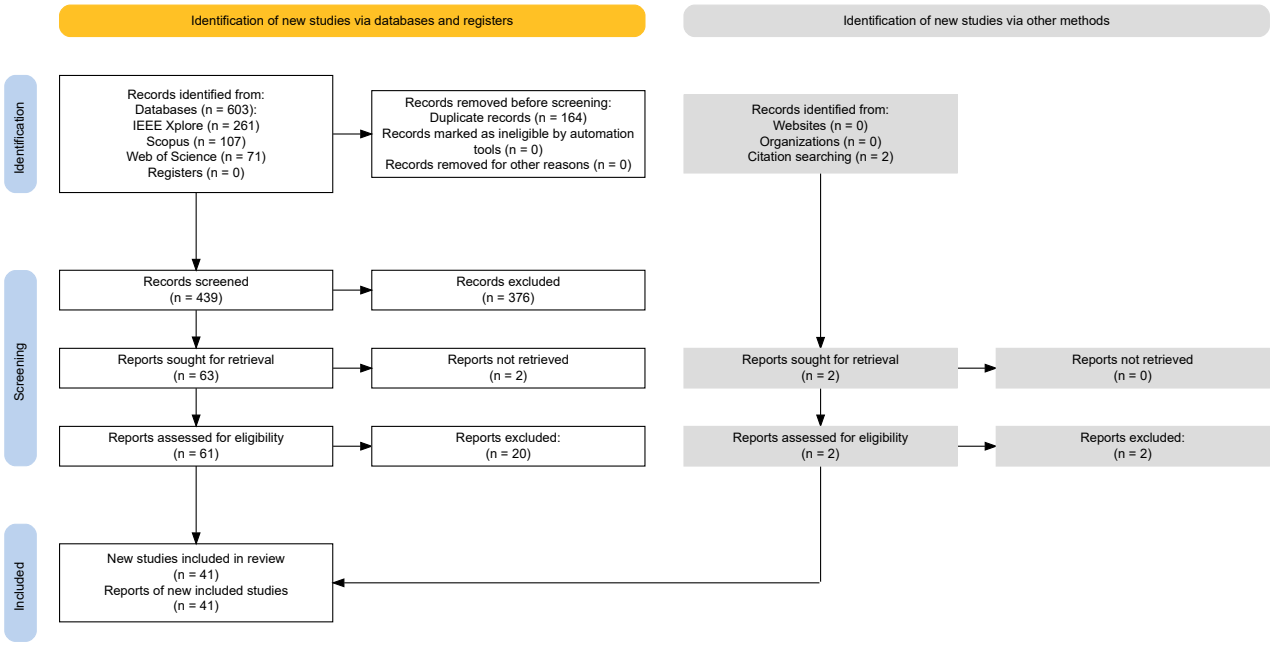

Figure 1: PRISMA flow diagram for the systematic literature review, detailing the number of records identified, screened, and included.

extracting data into analytical tables from their respective article subset. After each step, the subsets were swapped between groups in order to perform a cross-verification of the work done.

If there were any disagreements between groups, a discussion and a vote was held on the specific article by all groups. No external software was used to process the articles and no translation was required at any point.

The analytical tables consisted of rows representing articles, and columns representing specific aspects, such as limitations, methods, metrics etc.

Articles were assessed regarding the risk of being biased. Again, the same method of performing a split across three groups consisting of 2 people was used. Every group objectively reviewed the content of the assigned articles, and then a swap was performed, so that in case of any disagreements, the discussion could be held. No external tools were used to check for bias. The checklist for verifying bias for articles was as follows:

- Was the RAG architecture clearly and thoroughly described?
- Was the evaluation methodology reliable (e.g., use of standard benchmarks, comparison with a baseline model)?
- Are the results reproducible (e.g., availability of code or data)?
- Did the authors clearly discuss the study's limitations?

The selection process, aligned with the PRISMA guidelines [15], is fully transparent and documented in the PRISMA flow diagram (Figure 1).

Operational coding of research questions and inclusion threshold

For each of the nine research questions (RQ1–RQ9), we developed a category codebook and coded every full-text paper at the level of binary indicators (present/absent) for each category within a given RQ. This produced, for each paper p, a per-question count

$$s_{p,r} = \sum_{c \in \mathscr{C}_r} \mathbf{1}\{\text{paper } p \text{ is coded in category } c \text{ for RQ } r\} \tag{1}$$

which we then aggregated across all RQs:

$$S_p = \sum_{r=1}^{9} s_{p,r} \tag{2}$$

To focus the synthesis on studies providing substantive extractable evidence, we applied a frequency-based threshold to the total number of category occurrences. Papers were retained only if

$$S_p \geq 14 \tag{3}$$

The threshold $S_p \geq 14$ was selected based on the distribution of coding scores (mean $S_p \approx 16.6$) to balance coverage across all research questions while excluding studies with marginal relevance; exploratory analysis showed that lower thresholds substantially increased noise without improving thematic diversity.

Borderline cases (near the threshold) were adjudi-

cated by joint discussion and majority vote. This step resulted in the final set of 41 studies while preserving balanced coverage across all RQs.

Studies that appeared to meet the inclusion criteria but were excluded

Two records identified via Web of Science appeared eligible at screening but were excluded at full-text assessment because their full texts could not be retrieved through institutional access. As the information required for RQ-level extraction could not be verified, these papers were not included in the quantitative synthesis. The records are listed in Table 1.

2.5. Review protocol and registration

This systematic literature review was not formally registered in a public registry (e.g., PROSPERO) prior to its commencement. The research method detailed in this paper, including the research questions (Section 2.1), search strategy (Section 2.2), eligibility criteria (Section 2.2.2), and data extraction process (Section 2.4), serves as the protocol for this study. No amendments were made to this protocol during the review process.

3. Data items and selection of results

3.1. Scope of data extraction

The data extraction process focused exclusively on obtaining all available information directly relevant to Research Questions 1 through 9 (RQ1–RQ9). This process was performed manually; no external or automated data scraping tools were utilized. Data were extracted verbatim or as conceptual phrases corresponding to the defined outcomes. Crucially, no assumptions were made regarding missing or ambiguous data items, ensuring the integrity and traceability of the extracted corpus.

3.2. Outcome domains

The outcome domains were rigorously pre-specified based on the research questions. For every included study, we extracted all reported results matching the definitions of these domains. To maintain comprehensive data fidelity, if a single study reported multiple measures, time points, or analyses within one domain, all such instances were extracted separately. We applied no pre-selection or statistical imputation to the study outcomes.

3.3. RQ1 – Applications

- **Outcome:** The specific application domain(s) in which RAG was implemented.
- **Definition:** Domain labels (e.g., Question Answering, Summarization). The complete set of extracted labels and their definitions are provided in Table 2.

3.4. RQ2 – Challenges and Limitations

- **Outcome:** Challenges and limitations explicitly reported for RAG architectures.
- **Definition:** Categorical labels describing reported problems (e.g., Hallucinations, Computational Complexity, Scalability). Detailed definitions of each category are provided in Table 3.

3.5. RQ3 – Evaluation (Accuracy, Efficiency, Scalability)

- **Outcome:** Evaluation methods and performance indicators used to assess RAG systems.
- **Definition:** For each included study, all reported evaluation metrics were extracted and grouped into three overarching domains:
  1. **Accuracy:** Measures assessing the quality and correctness of generated outputs, such as automated text similarity scores and human judgments.
  2. **Efficiency:** Indicators describing computational or operational performance, including latency, processing time, and resource utilization.
  3. **Scalability:** Metrics capturing the ability of RAG systems to maintain performance across larger datasets, benchmarks, or retrieval workloads.

  The complete distribution of evaluation measures and their occurrence across studies is presented in Tables 4, 5, and 6.

3.6. RQ4 – Datasets and Benchmarks

- **Outcome:** Dataset and benchmark identifiers and attributes (name, domain, size, configuration).
- **Definition:** Extracted datasets are categorized by domain and type, as detailed in Tables 7 and 8.

3.7. RQ5 – Query Encoders

- **Outcome:** Encoder models utilized for queries or embeddings.
- **Definition:** Extracted data include the exact model

Table 1: Records excluded at full-text screening.

Study (source / DOI)	Year	Reason
Benchmarking Retrieval Augmented Generation in Quantitative Finance (Web of Science) 10.1007/978-3-031-67195-1_9	2024	Full text unavailable (publisher paywall).
Development of a Liver Disease–Specific Large Language Model Chat Interface Using Retrieval-Augmented Generation (Web of Science) 10.1097/HEP.0000000000000834	2024	Full text unavailable (publisher paywall).

name and version, classified by nature (domain-specific, multilingual, general-purpose, commercial, sparse/classic) and fine-tuning details. Model types are defined in Tables 9 through 12.

3.8. RQ6 – Retriever Search and Fusion Techniques

- **Outcome:** Retrieval, similarity, and fusion algorithms and implementation details.
- **Definition:** Extracted methods include core retrieval (sparse, dense, hybrid), similarity metrics (cosine, Euclidean), rank fusion, and reranking strategies. These are defined and categorized in Tables 13 through 17.

3.9. RQ7 – Retrieval Optimization Techniques

- **Outcome:** Applied optimizations to enhance retrieval performance.
- **Definition:** Optimization categories include chunking methods, vector database usage, knowledge-graph augmentation, query expansion, and reranking. The specific techniques and parameters are detailed in Table 18.

3.10. RQ8 – Generative Models Integrated

- **Outcome:** Generative model(s) used (commercial or open-source).
- **Definition:** Model name, version, usage context, and size where reported. Models are categorized by family (e.g., OpenAI, MetaAI, Chinese-specific) in Table 19.

3.11. RQ9 – Knowledge Bases

- **Outcome:** Types of knowledge stores used in RAG systems.
- **Definition:** Extracted knowledge base types, including vector, domain-specific, graph, and SQL databases, are defined in Table 20.

4. Quantitative results

In this chapter, the quantitative results of the conducted systematic literature review are presented. These results correspond to the nine previously defined research questions and indicate the number of publications that addressed each question to some extent. For each research question, relevant summaries are provided, including the number of studies and the most frequently recurring answers. The results are presented in the form of tables and brief summaries related to each research question.

4.1. RQ1. What applications of RAG are studied?

As summarized in Table 2, the literature analysis shows that RAG is applied across various domains, with Question Answering and Dialogue Systems being the most frequently studied area. Out of the 41 analyzed papers, 35 addressed this application (e.g., [7, 10, 16]). The second most common category was other domain-specific text generation (25 papers), followed by summarization tasks (14 papers) found in works such as [17] and [3]. Other application areas include enterprise knowledge management [18, 19], recommendation systems [20, 21], process automation [11, 22], compliance [8, 23], as well as medical [12, 24] and legal domains [13].

Additionally, several studies investigated low-resource conversational systems. In this context, "low-resource" refers to languages or domains with limited available data. These works examined how RAG can enhance dialogue systems for under-represented or low-data languages such as Swahili [16], as well as for highly specialized fields with scarce textual resources.

4.2. RQ2. What are the challenges and limitations in implementing RAG models?

Table 3 presents a detailed overview of the challenges and limitations associated with the implementation of RAG architectures. The two most frequently mentioned issues were the accuracy of generated responses (appearing in 20 studies, e.g., [17, 24]) and hallucinations — the generation of incorrect or fabricated information (reported in 20 publications such as [6, 7, 25]). Dynamic

Table 2: RQ1. What applications of RAG are studied?

Application Domain	Count	Paper(s)
Question Answering and Dialogue Systems	35	[16], [17], [25], [10], [3], [1], [6], [26], [11], [7], [27], [28], [12], [20], [13], [18], [29], [4], [9], [30], [31], [32], [33], [34], [2], [8], [35], [36], [37], [38], [39], [19], [40], [41], [42]
Other domain-specific text generation	25	[17], [1], [26], [11], [27], [12], [20], [13], [18], [29], [9], [30], [31], [32], [34], [21], [2], [23], [35], [36], [37], [38], [39], [41], [42]
Summarization	14	[17], [3], [6], [26], [11], [20], [29], [4], [30], [34], [21], [2], [36], [42]
Other enterprise search and knowledge management	10	[10], [6], [18], [32], [33], [21], [2], [8], [23], [19]
Conversational Systems (low-resource)	6	[16], [33], [34], [21], [40], [42]
Process automation	6	[22], [43], [11], [27], [21], [37]
Medical	5	[26], [24], [28], [12], [38]
Compliance	4	[21], [8], [23], [36]
Recommendation Systems	4	[20], [21], [2], [8]
Appraisal/Scoring	2	[25], [26]
Legal conflict resolution	2	[13], [36]

knowledge integration was the next major issue, cited in 19 studies [28, 44]. Other significant challenges included semantic confusion (11) and greater computational complexity (11) highlighted by [16] and [17]. Additional limitations involved lack of information in external databases [10, 29], lack of transparency and explainability, trustworthiness concerns [1, 11], scalability issues [2, 7], high computational costs, data privacy [12, 36], and longer response times. Additionally, data-centric challenges were observed, specifically *data parsing/format issues* [3, 37] and *retrieval granularity mismatch* [29], where retrieved chunks were either too broad or too narrow for the query.

4.3. RQ3. How is RAG evaluated in terms of accuracy, efficiency, and scalability?

Methodological note on efficiency metrics. In this review we distinguish *retrieval time*, *inference time*, and end-to-end *response time*. Retrieval time is the latency to fetch and (if applicable) re-rank candidate chunks prior to generation; inference time is the latency of the generator given its input context; response time is the end-to-end user-perceived latency, which includes retrieval time, inference time, and any orchestration overhead (e.g., network/API calls, serialization). Several included studies report retrieval time explicitly, while others report response time for the whole pipeline or provide both. We therefore record metrics at the granularity reported by each paper and compare like with like across studies.

In essence, inference time is a subset of response time:

$$\text{Response Time} = \text{Retrieval} + \text{Inference} + \text{Other Latencies}$$

The performance of RAG models is typically assessed across three key dimensions: accuracy, efficiency, and scalability. Each of these aspects is discussed in the literature using specific metrics and examples drawn from the analyzed publications.

As summarized in Tables 4, 5, and 6, across the 41 retained studies, accuracy is assessed using a mixture of automated and human-judgment metrics. The most frequent automated metrics are the RAGAS correctness family (e.g., used in [7, 10]) and ROUGE-L (e.g., [6, 16]). BLEU and BERTScore appear in several works, typically for summarization or low-resource QA, such as [16] and [44]. Human evaluation is used where domain correctness or faithfulness is hard to capture automatically (e.g., legal/medical [26] or Arabic QA [9]). These choices align with task design across the corpus: long-form QA and summarization favor ROUGE/BERTScore, while retrieval-grounded QA increasingly adopts RAG-specific scores such as RAGAS. Tables 4 and 6 summarize usage and frequency.

Efficiency is typically reported as (i) *inference time* for the generator (e.g., [16, 18, 27, 43]), (ii) *retrieval time* for candidate search and re-ranking, or (iii) end-to-end *response time* (e.g., [9, 28]). In our corpus, inference time and response time are the most common aggregate measures; a subset reports retrieval time separately, especially when approximate nearest-neighbor search (e.g., FAISS or ScaNN) and re-ranking are varied. Scalability is addressed via approximate vector search, top-k management, and

Table 3: RQ2. What are the challenges and limitations in implementing RAG models?

Challenge / Limitation	Count	Paper(s)
Accuracy	20	[17], [24], [28], [12], [20], [13], [29], [4], [9], [30], [31], [32], [33], [34], [21], [2], [23], [35], [36], [37], [40]
Hallucinations	20	[25], [6], [7], [13], [29], [4], [9], [30], [31], [33], [34], [21], [8], [23], [35], [36], [38], [19], [41], [42]
Dynamic knowledge integration	19	[44], [28], [20], [13], [18], [29], [30], [31], [33], [34], [21], [2], [8], [23], [35], [36], [38], [39], [42]
Trustworthiness	13	[1], [11], [24], [44], [28], [29], [9], [32], [33], [34], [21], [2], [8]
Explainability/Transparency	13	[17], [1], [7], [44], [29], [4], [9], [32], [21], [2], [8], [39], [42]
Lack of information in external database	12	[10], [3], [22], [6], [11], [29], [9], [34], [21], [2], [35], [38]
Semantic confusion	11	[25], [27], [30], [31], [32], [33], [21], [35], [19], [40], [42]
Greater computational complexity	11	[16], [17], [11], [7], [27], [28], [32], [34], [21], [2], [37]
Scalability	11	[7], [28], [20], [18], [34], [21], [2], [8], [36], [39], [42]
Data Privacy	8	[1], [11], [12], [31], [34], [21], [36], [41]
Cost-effectiveness	7	[17], [28], [18], [29], [30], [31], [32]
Longer response time	5	[16], [17], [32], [34], [38]
Data Parsing/Format Issues	5	[3], [29], [4], [9], [37]
Retrieval Granularity Mismatch	4	[3], [29], [4], [9]

modular pipeline design, with several studies noting practical trade-offs between retrieval speed, re-ranking cost, and overall response time.

The scalability dimension was addressed less frequently and often indirectly in the reviewed literature. Scalability in RAG systems can be understood as the ability to maintain performance as the system's demands increase, such as the size of the knowledge base, the volume of concurrent queries, or the complexity of the retrieval task. Only a few studies employed dedicated benchmarks designed to stress-test these aspects. Notably, one publication utilized comprehensive evaluation frameworks, using the CRAG (Comprehensive RAG Benchmark) and RAGFoundry [8] to assess performance across diverse datasets and retrieval workloads. Other studies implied scalability through discussions of their system's architecture (e.g., use of vector databases like FAISS for efficient search in large corpora) rather than through explicit, targeted metrics.

4.4. RQ4. What datasets and benchmarks are commonly used to evaluate RAG models?

Tables 7 and 8 detail the wide range of datasets and benchmarks identified in the reviewed literature to evaluate RAG models. These can be categorized into two main groups: domain-specific datasets and general-purpose benchmarks.

Among the domain-specific datasets, Biomedical datasets (e.g., BioASQ, GenMedGPT-5K [3, 38]) and low-resource language corpora (Arabic History-QA [9, 35]) appeared most frequently (3 publications each). These were followed by legal texts such as CAIL2018 and cross-jurisdictional legal corpora [13, 25] (2 publications), and summarization datasets (CNN/Daily Mail) [4, 43] (2 publications). A few studies employed specialized datasets from the healthcare (HealthcareMagic-101 [44]) and privacy (OPP-115, GDPR) domains.

Regarding general-purpose benchmarks, the most frequently used were large retrieval corpora such as Wikipedia and dbpedia-openai (5 papers, e.g., [27, 29]). These were followed by question answering and retrieval benchmarks like Natural Questions (NQ), SQuAD, CoQA, and WebQSP [7, 38]. Less frequently used were conversational QA datasets (e.g., CMCQA, LaMP), semantic similarity benchmarks (e.g., SICK), and specialized evaluation tools created for RAG, such as the Retrieval-Augmented Generation Benchmark (RGB) [6].

Examples verified in this corpus. To ground the above distributions, we highlight three representative, fully documented cases from our included papers: (i) a large Arabic hadith corpus (34,542 texts) with a 123-question eval-

Table 4: RQ3. How is RAG evaluated in terms of accuracy?

Metric	Count	Paper(s)
Correctness score (RAGAS)	9	[10], [7], [13], [18], [23], [38], [39], [40], [41]
ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation – Longest Common Subsequence)	9	[16], [6], [26], [44], [12], [9], [30], [19], [42]
Human evaluation	6	[26], [11], [7], [9], [35], [36]
Contextual Relevance	6	[26], [11], [7], [28], [9], [36]
BLEU (Bilingual Evaluation Understudy)	4	[16], [6], [44], [19]
Precision, Recall, F1	4	[27], [29], [4], [38]
Mean reciprocal rank (MRR)	3	[43], [13], [35]
Hit rate	2	[13], [38]
BERTScore	2	[6], [12]

Table 5: RQ3. How is RAG evaluated in terms of efficiency?

Metric	Count	Paper(s)
Inference Time	7	[16], [43], [7], [27], [18], [9], [34]
Response Time	4	[28], [9], [31], [32]
Cost-efficiency	3	[28], [31], [32]

Table 6: RQ3. How is RAG evaluated in terms of scalability?

Metric / Framework	Count	Papers
CRAG (Comprehensive RAG Benchmark), RAGFoundry	1	[8]

Table 7: RQ4. Domain-Specific datasets that are commonly used to evaluate RAG models.

Dataset	Count	Paper(s)
Biomedical (BioASQ, GenMedGPT-5K)	3	[3], [38], [41]
Low-Resource Languages (Arabic History-QA)	3	[29], [9], [35]
Legal (Domain-Specific Corpora, e.g., CAIL2018)	2	[25], [13]
Journalism/Text summarization (CNN/Daily Mail dataset)	2	[43], [4]
Healthcare (HealthcareMagic-101)	1	[44]
Privacy/GDPR (OPP-115)	1	[23]

Table 8: RQ4. General-Purpose Benchmarks that are commonly used to evaluate RAG models.

Task/Benchmark	Count	Paper(s)
Retrieval Corpora (Wikipedia, dbpedia-openai-1M-1536-angular)	5	[27], [29], [9], [2], [35]
QA & Retrieval (Natural Questions (NQ), SQuAD, CoQA, WebQSP)	3	[6], [7], [38]
Conversational QA (CMCQA, LaMP)	1	[38]
Specialized RAG Benchmarks (Retrieval-Augmented Generation Benchmark (RGB))	1	[6]

uation set for low-resource QA [9]; (ii) multi-document technical and scientific PDFs for image-aware RAG retrieval and MMR-style re-ranking [32]; and (iii) a domain dataset comprising flood-education and policy materials used to evaluate an operational RAG platform with reported end-to-end response time [34].

4.5. RQ5. What query encoders are used in RAG architectures?

As categorized in Tables 9 through 12, the reviewed literature employs a wide variety of query encoders used in RAG architectures, which can be grouped into four main categories: language-specific and multilingual encoders, domain-specific encoders, general-purpose encoders, and classic sparse models.

Among language-specific and multilingual encoders, the most frequently used was mBERT (4 publications, e.g., [9, 16]). Other models such as BGE-Large-zh-v1.5 (1 publication), primarily applied in Chinese-language contexts, were also noted. AraBERTv2 and BERT-large-Arabic were also used in Arabic NLP tasks [35].

Domain-specific encoders included specialized models such as BioBERT (biomedical [41]), PrivBERT (privacy-related), and a Chinese medical sentence embedding encoder. These were each used in only one publication.

The majority of the encoders came from the general-purpose group. The most commonly used were models based on the BERT architecture (11 publications), followed by Sentence-BERT (SBERT) (5) [8, 10], RoBERTa (2) [25, 37], and specific high-performing encoders like all-mpnet-base-v2 (3) [7, 21, 43]. Several commercial embedding models were also applied, including text-embedding-ada-002 (5 publications, e.g., [10]), gemini-embedding-001, and Snowflake-arctic-embed. Lightweight variants such as distilbert and text-embedding-3-small were also used, along with other dense encoders like Dense Passage Retrieval (DPR) [1], bge-base, stella-base, and Contriever.

In the classic / baseline encoder category, the most frequently used techniques were TF-IDF (5 publications) [11, 27] and Word2Vec (4 publications) [4, 29], highlighting their continued relevance as baseline retrieval approaches.

4.6. RQ6. What retriever search and fusion techniques are used in RAG architectures?

Table 13 through Table 17 summarize the retriever search and result fusion techniques, which play a crucial role in RAG architectures. The reviewed literature presents a variety of approaches, ranging from traditional keyword-based retrieval to advanced vector-based semantic matching and reranking strategies.

Among search techniques, the most widely used was semantic search (22 publications, e.g., [17, 29]), where queries and documents are embedded into a shared vector space. This was followed by vector search (15 papers) [26, 43], and classic sparse retrieval methods such as BM25 (9 papers, e.g., [1, 13, 16]) and TF-IDF (8) [11, 23]. Within dense retrieval, techniques such as Dense Passage Retrieval (DPR), deep learning models, and Contriever were employed. Hybrid retrieval, especially combinations like DPR + BM25 (3 papers), as well as graph-based retrieval (6 papers, e.g., [8, 25]), were also present.

For vector similarity, cosine similarity was by far the dominant metric (17 publications) [3, 7], while Euclidean distance was used in only one.

In terms of rank fusion, works applied Reciprocal Rank Fusion (RRF) (1 paper) and probabilistic fusion (1 paper).

Reranking strategies included diversity-enhancing methods such as Maximal Marginal Relevance (MMR) (1 paper) [32] and neural reranking, e.g., using the bge-reranker-large model (1 paper).

To improve efficiency and scalability, authors employed approximate search methods—most notably with FAISS (6 papers, including [16, 24]) and ScaNN (1 paper). Additional techniques included TOP-N retrieval (12 papers) and consolidation strategies to manage retrieved candidates.

4.7. RQ7. What techniques are employed to optimize retrieval in RAG models?

As detailed in Table 18, a variety of retrieval optimization strategies were identified across the analyzed literature to enhance the relevance and performance of RAG systems. The most commonly used technique was chunking methods and vector database–based retrieval, both appearing in 20 publications.

Chunking involves breaking source documents into smaller units (such as paragraphs or sentences) to improve the alignment between queries and documents and enable more accurate retrieval, as employed in [25] and [10]. Meanwhile, vector databases (e.g., FAISS, Pinecone) facilitate efficient similarity search over embedded representations [33].

The third most frequent method was the use of knowledge graphs (6 papers), which provide contextual relationships between entities, allowing for more semantically informed retrieval [11, 17].

Other techniques included query expansion [18, 29], query rewriting [7, 33], result reranking [13, 30], and adding metadata to improve matching accuracy and personalization [3, 11].

Table 9: RQ5. Language-Specific and Multilingual Query Encoders used in RAG architectures.

Encoder Model	Count	Paper(s)
mBERT (Multilingual BERT)	4	[16], [9], [35], [42]
AraBERTv2	2	[9], [35]
BGE-Large-zh-v1.5	1	[31]
BERT-large-Arabic	1	[35]
m3e-base	1	[6]

Table 10: RQ5. Domain-Specific Query Encoders used in RAG architectures.

Encoder Model	Count	Paper(s)
BioBERT	1	[41]
PrivBERT	1	[23]
Chinese Medical Sentence Encoder	1	[24]

Table 11: RQ5. General-purpose Query Encoders used in RAG architectures.

Model	Count	Paper(s)
BERT Family		
BERT	11	[22], [1], [11], [44], [13], [29], [4], [9], [30], [8], [40]
SBERT (Sentence-BERT)	5	[16], [10], [8], [35], [40]
MPNet-v2 (all-mpnet-base-v2)	3	[43], [7], [21]
RoBERTa	2	[25], [37]
Commercial (Proprietary APIs)		
OpenAI Ada-002	5	[10], [29], [33], [39], [19]
Gemini Embedding Model (gemini-embedding-001)	1	[29]
Snowflake Arctic Embed	1	[36]
Lightweight Variants		
DistilBERT	2	[29], [4]
OpenAI Text Embedding 3-Small	1	[33]
Other Dense		
DPR (Dense Passage Retrieval)	3	[1], [28], [30]
BGE-Base	2	[6], [44]
BGE-Large	1	[44]
GTE-Base	1	[6]
Stella-Base	1	[6]

Table 12: RQ5. Classic / Baseline Encoders used in RAG architectures.

Model	Count	Paper(s)
TF-IDF – Term Frequency-Inverse Document Frequency	5	[11], [27], [9], [30], [23]
Word2Vec – Word to Vector model	4	[29], [4], [30], [23]

4.8. RQ8. Which generative models are integrated into RAG architectures?

Table 19 provides a comprehensive overview of the generative models integrated into RAG architectures. The analysis reveals a wide variety of both commercial and open-source generative models used in RAG systems.

The most commonly integrated model was GPT-4 (14 papers), reflecting its dominance in precision-driven generation (e.g., [3, 25]). Other popular models included GPT-3.5 Turbo (10) [10, 43] and GPT-3.5 (9) [12, 25]. Overall, OpenAI's GPT series was by far the most frequently used among commercial solutions.

Among open-source models, Meta's LLaMA-2 (9 papers) stood out [22, 33], followed by newer versions like LLaMA-3-8B, LLaMA-3-70B-Instruct, and LLaMA-3.1.

Table 13: RQ6. Core retriever search techniques used in RAG architectures.

Method	Count	Paper(s)
Sparse Retrieval		
BM25	9	[16], [22], [1], [43], [6], [13], [18], [9], [30]
TF-IDF	8	[16], [1], [11], [27], [29], [9], [30], [23]
Keywords	4	[17], [43], [6], [26]
Dense / Neural Retrieval		
Semantic Search	22	[17], [11], [7], [12], [20], [18], [29], [4], [9], [30], [31], [32], [33], [34], [21], [2], [8], [35], [38], [40], [41], [42]
Vector Search	15	[43], [26], [11], [7], [27], [12], [20], [29], [9], [30], [33], [34], [2], [35], [36]
Dense retrieval algorithms	2	[6], [28]
Deep learning model retrieval	2	[25], [2]
Hybrid Retrieval		
DPR + BM25	3	[1], [13], [41]
Hybrid Search	2	[17], [18]
Graph-Based Retrieval		
Graph-based	6	[25], [26], [11], [8], [38], [42]
Other / Cross-cutting		
Similarity Search	4	[10], [9], [34], [21]

Table 14: RQ6. Vector similarity metrics used in RAG architectures.

Method	Count	Paper(s)
Cosine similarity	17	[3], [7], [29], [4], [9], [30], [31], [32], [33], [34], [21], [23], [35], [36], [39], [19], [40]
Euclidean Distance	1	[27]

Table 15: RQ6. Rank fusion techniques used in RAG architectures.

Method	Count	Paper(s)
Reciprocal Rank Fusion (RRF)	1	[43]
Probabilistic fusion	1	[24]

Table 16: RQ6. Reranking strategies used in RAG architectures.

Reranking Strategy	Count	Paper(s)
Diversity Reranking – MMR (Maximal Marginal Relevance)	1	[32]
Neural Reranking – Bge-reranker-large	1	[24]

Table 17: RQ6. Efficiency and scalability techniques for retrieval and fusion used in RAG architectures.

Approximate Search		
FAISS (Facebook AI Similarity Search)	6	[16], [24], [13], [31], [8], [35]
ScaNN	1	[27]
Candidate Management		
TOP-*N* retrieval	12	[25], [3], [43], [6], [29], [31], [33], [34], [21], [2], [39], [19]
Consolidation strategy	2	[3], [36]

Legacy models like mBART and BART were also used, mostly in earlier research [16].

There was notable representation of Chinese-specific models, including Qwen-7B, Baichuan, ChatGLM2, and GLM4, illustrating a strong interest in multilingual and localized RAG applications [6, 25].

Table 18: RQ7. Retrieval optimization techniques in RAG models.

Method	Count	Paper(s)
Chunking methods	20	[25], [10], [3], [6], [26], [11], [20], [13], [29], [9], [30], [31], [32], [33], [34], [21], [2], [36], [39], [19]
Vector Database based RAG	20	[10], [26], [11], [27], [24], [12], [20], [29], [9], [30], [31], [32], [33], [34], [21], [2], [23], [35], [36], [39]
Knowledge Graphs-based RAG	6	[17], [11], [8], [23], [38], [42]
Query expansion	6	[22], [27], [18], [29], [4], [36]
Reranking	6	[43], [6], [13], [18], [9], [30]
Adding meta-data	4	[3], [11], [9], [30]
Query rewriting	3	[22], [7], [33]

Google's models (e.g., FlanT5-XXL, Gemini, Code Bison) were present in a few works, but with less frequency [27, 29].

Other interesting mentions include Mistral-7B, Vicuna, Claude 3 Sonnet, and Hermes 2 Pro [37]. Though less frequent, these highlight growing diversity and experimentation beyond OpenAI and Meta ecosystems.

4.9. RQ9. What types of knowledge bases are used in RAG architectures?

As shown in Table 20, vector databases are by far the most commonly used knowledge source in RAG architectures, appearing in 22 publications (e.g., [10,11,43]). These databases support efficient and scalable search over embedded representations, making them well-suited for semantic document retrieval.

The second most frequent type is domain-specific databases (6 papers), such as those related to medical, legal, or financial domains [1, 7, 13, 26]. These sources are particularly valuable in tasks that require domain precision and contextual understanding.

Graph databases were used in 7 studies [25, 42], allowing for the modeling and retrieval of relationships between concepts or entities. SQL databases were mentioned in 3 papers, mainly in the context of integrating with structured enterprise systems [10, 12].

5. Discussion

This section provides an integrative discussion of the results of this Systematic Literature Review on RAG. The discussion interprets the findings in the context of previous research, identifies limitations of the evidence base and of the review process, and outlines implications for research and practice.

5.1. Principal findings and interpretation

The qualitative synthesis revealed consistent insights into how RAG architectures are applied and developed across domains. The majority of the 41 included studies focused on applications such as question answering, dialogue systems, and domain-specific text generation. These findings reflect the natural alignment of RAG with information-seeking and reasoning tasks, where grounding model outputs in retrieved context enhances factual reliability.

A clear trend toward the use of OpenAI's GPT series—particularly GPT-4 and GPT-3.5—was observed. This dominance can be attributed to their high accessibility via APIs, superior performance in multilingual and factual reasoning tasks, and extensive community adoption. Open-source models such as LLaMA-2, LLaMA-3, and Mistral were also represented, indicating growing diversification and the maturing open ecosystem for RAG research.

Across the analyzed literature, authors frequently emphasized the benefits of RAG in improving response faithfulness, contextuality, and currency. In several domains (e.g., medicine, education, law), RAG-based systems were reported to outperform traditional retrieval-only or generation-only baselines by combining structured search with generative flexibility. However, challenges remained regarding consistency, hallucination control, and integration of dynamic external knowledge.

5.2. Limitations of the evidence base

The strength of evidence across the included studies was limited by several recurring issues. Many works lacked standardized evaluation procedures or failed to report reproducible experimental setups. Only a minority provided open access to datasets or implementation code, which constrains transparency and verification.

Furthermore, evaluation metrics were highly heterogeneous. While correctness-oriented metrics such

Table 19: RQ8. Generative models integrated into RAG architectures.

Model	Count	Paper(s)
OpenAI GPT models		
GPT-2	1	[16]
GPT-3	3	[22], [29], [30]
GPT-3.5	9	[25], [12], [29], [4], [32], [37], [38], [40], [42]
GPT-3.5-Turbo	10	[10], [43], [6], [26], [44], [32], [33], [34], [21], [19]
GPT-4	14	[25], [3], [22], [6], [29], [4], [9], [31], [32], [36], [38], [39], [41], [42]
GPT-4 Turbo	2	[8], [19]
GPT-4o	4	[6], [7], [44], [19]
GPT-4o-mini	1	[9]
MetaAI models		
LLaMA-2	9	[22], [4], [33], [34], [21], [35], [36], [38], [41]
LLaMA-2-7B	3	[33], [23], [38]
LLaMA-3-70B-Instruct	2	[8], [35]
LLaMA-3	2	[7], [8]
LLaMA-3-8B	4	[10], [44], [12], [35]
LLaMA-3.1	3	[20], [13], [42]
mBART	1	[16]
BART	2	[22], [1]
Chinese specific models		
Qwen-7b	4	[25], [6], [24], [40]
Qwen-14b	2	[25], [6]
Qwen-72b	2	[25], [24]
Qwen2-7B	2	[10], [6]
Baichuan2-7b	1	[25]
Baichuan2-13b	2	[25], [6]
SUS-Chat-34B	1	[25]
ChatGLM2-6B	1	[6]
GLM4	2	[9], [31]
Google models		
mT5	1	[16]
T5	2	[22], [1]
Gemini-1.0-Pro	1	[29]
Gemma-2-9b	2	[10], [27]
Other		
GPT-Neo	1	[16]
Mistral-7B	3	[10], [29], [4]
Falcon-RW-1B-Cha	1	[43]
BERT-large-Arabic	1	[35]
Hermes 2 Pro	1	[37]

Table 20: RQ9. Knowledge bases used in RAG architectures.

Knowledge Base type	Count	Paper(s)
Vector database	22	[10], [43], [11], [7], [27], [12], [20], [13], [18], [9], [30], [31], [32], [33], [34], [21], [2], [8], [35], [37], [19], [41]
Domain-specific databases	6	[1], [26], [11], [7], [13], [35]
Graph database	7	[25], [22], [26], [11], [8], [38], [42]
SQL database	3	[10], [22], [12]

as RAGAS and ROUGE-L were common, their usage was inconsistent across tasks, making cross-comparison difficult. Several studies relied solely on human evaluation without quantitative baselines, further reducing comparability. Additionally, the reviewed research tended to focus heavily on English-language datasets and high-resource domains, limiting the generalizability of the conclusions.

5.3. Limitations of the review process

This review was conducted using three major bibliographic databases (IEEE Xplore, Scopus, and Web of Science) and was limited to peer-reviewed English-language publications from 2021–2025. While this ensured a high-quality corpus, it may have excluded relevant non-English or non-indexed studies, particularly in regions where RAG is actively researched (e.g., China or Korea).

The review was not preregistered in a public registry (e.g., PROSPERO), which limits procedural transparency. Furthermore, although backward snowballing was performed, no automated text-mining or grey literature search was included, potentially omitting recent preprints or industrial implementations.

5.4. Implications for research and practice

From a practical standpoint, the findings suggest that RAG systems can meaningfully improve factual consistency and adaptability in domain-specific applications. Practitioners should, however, prioritize the integration of mechanisms for hallucination detection, metadata-aware retrieval, and ongoing evaluation of knowledge base freshness. Vector databases and hybrid retrievers emerged as key enablers of scalability and efficiency, making them crucial components in applied RAG pipelines.

For researchers, the synthesis highlights several promising directions: (1) establishing unified evaluation benchmarks for accuracy, efficiency, and scalability; (2) extending empirical studies to underrepresented languages and domains; and (3) exploring interpretability and bias mitigation techniques within retrieval-augmented pipelines. Further research is also needed to systematically compare open-source and proprietary models under identical retrieval conditions.

Finally, bridging the gap between research metrics and practical reliability requires a unified approach to evaluation. Specifically, to address identified inconsistencies in evaluation standards, we propose a Decoupled Evaluation Strategy as a fundamental requirement for future RAG research and deployment. Our analysis indicates that traditional end-to-end metrics, such as ROUGE or BLEU, often conflate errors, making it impossible to discern whether a failure originated in the retrieval stage (e.g., irrelevant context) or the generation stage (e.g., hallucination despite correct context). Therefore, we provide the following guidance: (1) Retrieval Performance must be isolated using rank-aware metrics such as Mean Reciprocal Rank (MRR) or Hit Rate to verify context quality; and (2) Generation Faithfulness should be assessed using specialized frameworks like RAGAS to evaluate groundedness. Adopting this granular approach is critical for identifying system bottlenecks and ensuring the reliability of RAG pipelines in production environments.

6. Conclusion

This systematic literature review has analyzed 41 peer-reviewed studies published between 2021 and 2025, mapping the architectural landscape, applications, and evaluation methodologies of RAG. The synthesis of evidence confirms that RAG has become a pivotal architecture for mitigating the limitations of standalone LLMs, particularly regarding factual grounding and the integration of up-to-date information.

The structural analysis reveals that RAG architectures have coalesced around specific dominant components. The majority of implementations rely on vector databases for knowledge storage and semantic search for retrieval, often augmented by hybrid retrieval techniques (combining dense and sparse methods) to enhance precision. While OpenAI's GPT series remains the standard for generative components due to its reasoning capabilities, there is a significant and growing trend toward the adoption of open-source models such as LLaMA and specialized Chinese-language models, reflecting a shift toward more accessible and customizable research ecosystems.

Despite the demonstrated efficacy of RAG in Question Answering and Dialogue Systems, this review identifies critical challenges that persist in the field. The primary bottleneck is no longer the retrieval mechanism itself, but rather the standardization of evaluation. The heterogeneity of metrics—ranging from traditional ROUGE scores to automated frameworks like RAGAS and ad-hoc human evaluation—hinders the ability to objectively compare performance across different studies. Furthermore, while hallucination reduction is a primary goal of RAG, it remains a persistent challenge, necessitating further research into dynamic knowledge integration and robust verification layers.

Finally, the review highlights significant gaps in reproducibility and inclusivity. The current evidence

base is heavily skewed toward English-language and high-resource domains, with limited attention paid to low-resource languages where RAG could offer substantial benefits. Additionally, the scarcity of open-access code and datasets in published research restricts community-driven validation. Future research efforts must therefore prioritize the development of standardized benchmarking frameworks, the expansion of RAG to multilingual contexts, and a commitment to open science practices to ensure the scalability and reliability of these systems in real-world applications.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability

The analysis in this systematic literature review is based on 41 primary studies, which are fully cited in the bibliography. These articles are publicly available or were accessed by the authors through the scientific databases and resources provided by the Gdańsk University of Technology. The detailed data extraction tables (analytical tables) generated for the internal analysis during this review are available upon request.

References

[1] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, p. 6491–6501, ACM, August 2024.

[2] A. F. Chao, "Blending visitor familiarity and site significance: Rag enhanced heritage interpretation," in *2024 IEEE 13th Global Conference on Consumer Electronics (GCCE)*, p. 286–287, IEEE, October 2024.

[3] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, and M. Abdelrazek, "Seven failure points when engineering a retrieval augmented generation system," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN 2024, p. 194–199, ACM, April 2024.

[4] M. V. Nezafat and S. Samet, "Fake news detection with retrieval augmented generative artificial intelligence," in *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, p. 160–167, IEEE, November 2024.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 9459–9474, Curran Associates, Inc., 2020.

[6] Y. Lyu, Z. Li, S. Niu, F. Xiong, B. Tang, W. Wang, H. Wu, H. Liu, T. Xu, and E. Chen, "Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models," *ACM Transactions on Information Systems*, vol. 43, p. 1–32, January 2025.

[7] M. H. Heydari, A. Hemmat, E. Naman, and A. Fatemi, "Context awareness gate for retrieval augmented generation," in *2024 15th International Conference on Information and Knowledge Technology (IKT)*, p. 260–264, IEEE, December 2024.

[8] V. Kamra, L. Gupta, D. Arora, and A. K. Yadav, "Enhancing document retrieval using ai and graph-based rag techniques," in *2024 5th International Conference on Communication, Computing amp;amp; Industry 6.0 (C2I6)*, p. 1–7, IEEE, December 2024.

[9] M. Alshammary, M. N. Uddin, and L. Khan, "Rfpg: Question-answering from low-resource language (arabic) texts using factually aware rag," in *2024 IEEE 10th International Conference on Collaboration and Internet Computing (CIC)*, p. 107–116, IEEE, October 2024.

[10] J. Byun, B. Kim, K.-A. Cha, and E. Lee, "Design and implementation of an interactive question-answering system with retrieval-augmented generation for personalized databases," *Applied Sciences*, vol. 14, p. 7995, September 2024.

[11] L. Zhou, S. Yan, Z. Li, and J. Ma, "Exploring the application of retrieval-augmented generation technology in defense technology intelligence," in *2024 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC)*, p. 664–669, IEEE, September 2024.

[12] H. Y. Leong, Y. Gao, and S. Ji, "A gen ai framework for medical note generation," in *2024 6th International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, p. 423–429, IEEE, November 2024.

[13] R. S. M. Wahidur, S. Kim, H. Choi, D. S. Bhatti, and H.-N. Lee, "Legal query rag," *IEEE Access*, vol. 13, p. 36978–36994, 2025.

[14] B. A. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report, 07 2007.

[15] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, R. Chou, J. Glanville, J. M. Grimshaw, A. Hróbjartsson, M. M. Lalu, T. Li, E. W. Loder, E. Mayo-Wilson, S. McDonald, L. A. McGuinness, L. A. Stewart, J. Thomas, A. C. Tricco, V. A. Welch, P. Whiting, and D. Moher, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, vol. 372, 2021.

[16] E. V. Ndimbo, Q. Luo, G. C. Fernando, X. Yang, and B. Wang, "Leveraging retrieval-augmented generation for swahili language conversation systems," *Applied Sciences*, vol. 15, p. 524, January 2025.

[17] J. C. dos Santos Junior, R. Hu, R. Song, and Y. Bai, "Domain-driven llm development: Insights into rag and fine-tuning practices," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, p. 6416–6417, ACM, August 2024.

[18] P. Sharma and A. K. Mohammad, "Dynamic retriever selection in rag systems: An rl approach to user-centric nlp," in *2024 International Conference on Electrical and Computer Engineering Researches (ICECER)*, p. 1–6, IEEE, December 2024.

[19] G. Şahin, K. Varol, and B. K. Pak, "Llm and rag-based question answering assistant for enterprise knowledge management," in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, p. 1–6, IEEE, October 2024.

[20] M. Arslan, S. Munawar, and Z. Riaz, "Sustainable urban water decisions using generative artificial intelligence," in *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, p. 1–5, IEEE, December 2024.

[21] A. R. Wibowo, S. Fauziati, and R. Hartanto, "Government product recommendation systems in e-katalog: leveraging large language models with retrieval-augmented generation," *IET Conference Proceedings*, vol. 2024, p. 603–610, March 2025.

[22] B. Han, T. Susnjak, and A. Mathrani, "Automating systematic literature reviews with retrieval-augmented generation: A comprehensive overview," *Applied Sciences*, vol. 14, p. 9103, October 2024.

[23] L. Garza, L. Elluri, A. Piplai, A. Kotal, D. Gupta, and A. Joshi, "Privcomp-kg: Leveraging kg and llm for compliance verification," in *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, p. 97–106, IEEE, October 2024.

[24] C. Li, J. Luo, F. Jing, Y. Han, H. Ren, H. Zhang, and W. Cheng, "Development of a meta-question enhanced retrieval-augmented generation model and its application in dermatology," in *2024 17th International Conference on Advanced Computer Theory and Engineering (ICACTE)*, p. 281–285, IEEE, September 2024.

[25] F. Zhang, Y. Luo, Z. Gao, and A. Han, "Injury degree appraisal of large language model based on retrieval-augmented generation and deep learning," *International Journal of Law and Psychiatry*, vol. 100, p. 102070, May 2025.

[26] M. Rani, B. K. Mishra, D. Thakker, and M. N. Khan, "To enhance graph-based retrieval-augmented generation (rag) with robust retrieval techniques," in *2024 18th International Conference on Open Source Systems and Technologies (ICOSST)*, p. 1–6, IEEE, December 2024.

[27] M. Gao, P. Lu, Z. Zhao, X. Bi, and F. Wang, "Leveraging large language models: Enhancing retrieval-augmented generation with scann and gemma for superior ai response," in *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)*, p. 619–622, IEEE, October 2024.

[28] V. S, Seshasai, and L. Joseph, "A hybrid deep learning algorithm for improved chatbot accuracy and relevance through advanced retrieval-augmented generation," in *2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, p. 1–7, IEEE, December 2024.

[29] M. Barochiya, P. Makhijani, H. N. Patel, P. Goel, and B. Patel, "Evaluating rag pipeline in multimodal llm-based question answering systems," in *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, p. 69–75, IEEE, December 2024.

[30] S. Roy, M. Goswami, N. Nargund, S. Mohanty, and P. K. Pattnaik, "Conversational text extraction with large language models using retrieval-augmented systems," in *2024 6th International Conference on Computational Intelligence and Networks (CINE)*, p. 1–6, IEEE, December 2024.

[31] L. Xu and J. Liu, "A chat bot for enrollment of xi 'an jiaotong-liverpool university based on rag*," in *2024 8th International Workshop on Control Engineering and Advanced Algorithms (IWCEAA)*, p. 125–129, IEEE, November 2024.

[32] S. Bag, A. Gupta, R. Kaushik, and C. Jain, "Rag beyond text: Enhancing image retrieval in rag systems," in *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET*, p. 1–6, IEEE, July 2024.

[33] B. B. İrican, M. Sivri, V. Kokach, B. Kocaçınar, and F. P. Akbulutl, "Qbot: Domain-specific chatbots with retrieval-augmented generation and vector embedding for complex documentation queries," in *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, p. 1–6, IEEE, October 2024.

[34] L. Addison, A. Hosang, T.-A. Tuitt, K. Manohar, and P. Hosein, "A llm-based platform for flood risk education and weather alerts in sids," in *2024 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*, p. 1–6, IEEE, November 2024.

[35] S. Maged, A. ElMaghraby, A. Marzban, M. Essawey, A. Ahmed, E. Negm, and W. H. Gomaa, "Historyquest: Arabic question answering in egyptian history with llm fine-tuning and transformer models," in *2024 Intelligent Methods, Systems, and Applications (IMSA)*, p. 135–140, IEEE, July 2024.

[36] K. K, P. T, O. V. G, D. J, and S. S. S, "Interactive legal assistance system using large language models," in *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, p. 931–937, IEEE, October 2024.

[37] F. L. Cesista, R. Aguiar, J. Kim, and P. Acilo, "Retrieval augmented structured generation: Business document information extraction as tool use," in *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, p. 227–230, IEEE, August 2024.

[38] B. Zhan, A. Li, X. Yang, D. He, Y. Duan, and S. Yan, "Rarok:retrieval-augmented reasoning on knowledge for medical question answering," in *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 2837–2843, IEEE, December 2024.

[39] M. Guettala, S. Bourekkache, O. Kazar, and S. Harous, "Building advanced rag qamp;a with multiple data sources using langchain: A multi-search agent rag application in ubiquitous learning," in *2024 2nd International Conference on Computing and Data Analytics (ICCDA)*, p. 1–7, IEEE, November 2024.

[40] L. Huang, H. Lan, Z. Sun, C. Shi, and T. Bai, "Emotional rag: Enhancing role-playing agents through emotional retrieval," in *2024 IEEE International Conference on Knowledge Graph (ICKG)*, p. 120–127, IEEE, December 2024.

[41] S. Ranasinghe, D. De Silva, N. Mills, D. Alahakoon, M. Manic, Y. Lim, and W. Ranasinghe, "Addressing the productivity paradox in healthcare with retrieval augmented generative ai chatbots," in *2024 IEEE International Conference on Industrial Technology (ICIT)*, p. 1–6, IEEE, March 2024.

[42] S. L. Phyu, S. Jaman, M. Uchkempirov, and P. Kulkarni, "Myanmar law cases and proceedings retrieval with graphrag," in *2024 IEEE International Conference on Big Data (BigData)*, p. 2506–2513, IEEE, December 2024.

[43] A. Chen and S. Tran, "Supercharging document composition with generative ai: A secure, custom retrieval-augmented generation approach," in *2024 11th IEEE Swiss Conference on Data Science (SDS)*, p. 123–130, IEEE, May 2024.

[44] J. He, C. Liu, G. Hou, W. Jiang, and J. Li, "Press: Defending privacy in retrieval-augmented generation via embedding space shifting," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5, IEEE, April 2025.