

Article

# Retrieval-Augmented Generation to Generate Knowledge Assets and Creation of Action Drivers

Antony James <sup>1,\*</sup>, Marcello Trovati <sup>1,\*</sup> and Simon Bolton <sup>2</sup><sup>1</sup> Department of Computer Science, Edge Hill University, St Helens Road, Ormskirk L39 4QP, Lancashire, UK<sup>2</sup> Business School, Edge Hill University, St Helens Road, Ormskirk L39 4QP, Lancashire, UK; boltons@edgehill.ac.uk

\* Correspondence: antony.james@edgehill.ac.uk (A.J.); marcello.trovati@edgehill.ac.uk (M.T.)

**Abstract:** This article explores the application of Retrieval-Augmented Generation (RAG) to enhance the creation of knowledge assets and develop actionable insights from complex datasets. It begins by contextualising the limitations of large language models (LLMs), notably their knowledge cut-offs and hallucination tendencies, and it will present RAG as a promising solution that integrates external knowledge retrieval to improve factual accuracy and relevance. This study reviews current RAG architectures, including naïve and advanced models, emphasising techniques such as optimised indexing, query refinement, metadata utilisation, and the incorporation of autonomous AI agents in agentic RAG systems. Methodologies for effective data preprocessing, semantic-aware chunking, and retrieval strategies—such as multihop retrieval and reranking—are also discussed to address challenges such as irrelevant retrieval and semantic fragmentation. This work further examines embedding models, notably the use of state-of-the-art vector representations, to facilitate precise similarity searches within knowledge bases. A case study demonstrates the deployment of an RAG pipeline for analysing multisheet datasets, highlighting challenges in data structuring, prompt engineering, and ensuring output consistency.

**Keywords:** RAG; Retrieval-Augmented Generation; LLM; large language models; AI; artificial intelligence



Academic Editors: Marija Brkić Bakarić and Maja Matetic

Received: 28 April 2025

Revised: 27 May 2025

Accepted: 30 May 2025

Published: 1 June 2025

**Citation:** James, A.; Trovati, M.; Bolton, S. Retrieval-Augmented Generation to Generate Knowledge Assets and Creation of Action Drivers. *Appl. Sci.* **2025**, *15*, 6247. <https://doi.org/10.3390/app15116247>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the contemporary digital age, characterised by an unprecedented volume and velocity of information, effective management and utilisation of knowledge are paramount for individuals and organisations alike. The rise of large language models (LLMs) has marked a significant leap in natural language processing capabilities, enabling machines to understand, generate, and interact with human language at a sophisticated level. However, standalone LLMs often face limitations such as the propensity for hallucinations, i.e., generating factually incorrect or non-sensical information—and reliance on static training data that can quickly become outdated [1].

The use of LLMs for the generation of insight and analysis has recently emerged as a crucial application, demonstrating the potential to automate processes that traditionally require extensive human input [1]. Although extracting relevant information from the data remains a fundamental challenge, LLMs have demonstrated particular value in generating qualitative insights and explaining complex concepts in accessible terms in various organisations and business contexts [2].

However, LLMs face two significant limitations: their knowledge cut-off dates [3] and their tendency to generate hallucinations (producing incorrect or unverified information) [1].

These challenges have made it difficult to consistently obtain factually accurate information. Retrieval-Augmented Generation (RAG) [4,5] has emerged as a promising solution to address these limitations by augmenting LLM outputs with retrieved factual information. The effectiveness of RAG implementations depends heavily on both the nature of the input data and the methodologies used to organise and feed this information into the system [6]. Recent studies have shown that RAG effectiveness is significantly dependent on both the quality of knowledge retrieval and the complexity of the target task [7].

This article explores the current state of the art in RAG technology and examines various approaches to its implementation, with a particular focus on optimising the creation of knowledge assets and developing actionable insights [8]. The main contribution of this research focusses on the implementation of an RAG-based system to address a real-world scenario, based on the Baseline Review 2024 for the Liverpool City Region [9], as discussed in Section 6, which provides an economic assessment of the Professional Business Services (PBS) sector. This review is based on large and diverse datasets, suggesting several challenges associated with the use of AI-based analysis. In this work, novel approaches are implemented to address data structuring, prompt development, and testing/retesting for consistency.

Evaluation metrics, including context relevance and answer faithfulness, are employed to assess system performance. The findings demonstrate that RAG systems, particularly when enhanced with AI agent capabilities for autonomous query optimisation and adaptive response generation, possess substantial potential to generate reliable, comparable, and insightful knowledge assets, thereby significantly advancing data-driven decision-making processes in complex organisational contexts and establishing practical frameworks for implementing intelligent information systems in real-world applications.

## 2. Related Works

Retrieval-Augmented Generation (RAG) has emerged as a key technique in the application of LLMs to business contexts. This approach complements the generative capabilities of LLMs with the factual grounding of external knowledge bases, addressing critical limitations of standalone LLMs such as knowledge cut-offs, hallucinations, and lack of domain specificity. The core principle of RAG involves an LLM retrieving relevant information from an external data source before generating a response to a query [4]. This mechanism allows businesses to leverage their proprietary data or up-to-date information sources, ensuring that the LLM's output is not only coherent and contextually appropriate, but also accurate and reflective of current knowledge. Current research indicates that RAG improves LLM performance by establishing responses in dynamically accessible content, thus improving reliability, a crucial factor in enterprise applications [10,11].

The integration of RAG offers several distinct advantages for businesses, such as

**Enhanced Accuracy and Reduced Hallucination.** By grounding responses in verifiable external data, RAG significantly mitigates the tendency of LLMs to generate factually incorrect information. This is particularly vital in business domains where accuracy is paramount, such as finance, legal, and healthcare [12].

**Domain-Specific and Up-to-Date Knowledge.** RAG enables LLMs to access and utilise current specialised information from enterprise-specific databases or real-time data feeds. This overcomes the static knowledge limitation of pretrained LLMs, allowing businesses to deploy AI solutions that are conversant with their unique operational context and the latest developments [11].

**Improved Traceability and Trust.** Responses generated by RAG systems can often be linked back to the source documents used for retrieval. This traceability improves user

trust and allows verification, which is essential for compliance and critical decision-making processes [11].

**Cost-Effective Customisation.** Compared to the extensive resources required to fine-tune or retrain an entire LLM with new or proprietary data, RAG offers a more cost-effective and agile way to customise LLM outputs for specific business needs by updating the external knowledge base [12].

In [13], the authors proposed an automated customer service system for mobile operators utilising LLMs combined with a Retrieval-Augmented Generation (RAG) system. The research aims to improve the quality of automated customer service for customers of mobile operators. The authors conducted a comprehensive analysis of open-source LLMs and methods for model adaptation and pretraining. They designed the architecture and deployment diagram for the RAG system. The paper also describes the main stages of system training and provides performance evaluations for different LLMs.

In [14], TrumorGPT, a generative artificial intelligence solution, is introduced to verify facts in the health domain. Its primary goal is to distinguish trumors, which are health-related rumours that turn out to be true, providing a crucial tool to differentiate between speculation and verified facts. TrumorGPT leverages a large language model (LLM) with few-shot learning for semantic health knowledge graph construction and semantic reasoning. To address common issues in LLMs, such as hallucinations and limitations of static training data, TrumorGPT incorporates graph-based Retrieval-Augmented Generation (GraphRAG). GraphRAG accesses and uses information from regularly updated semantic health knowledge graphs that contain the latest medical news and health information, ensuring that fact checking is based on the most recent data.

However, the effectiveness of an RAG system is highly dependent on the quality of the retrieval mechanism. Poor retrieval can lead to suboptimal or incorrect responses generated [4]. Issues such as “lost in the middle”, where models struggle to utilise information from long contexts, also affect retrieval. Furthermore, implementing and scaling RAG systems can be complex, involving the integration of multiple components and potentially incurring computational overhead [5]. Finally, the assessment of the performance of RAG systems applied to real-world scenarios can be challenging, requiring evaluation of both the retrieval and generation components, as well as the multidisciplinary aspects related to the specific areas of investigation [15].

### 3. Materials and Methods

#### 3.1. Retrieval Augmented Generation

Retrieval-Augmented Generation [4] (RAG), a complex hybrid model architecture, enhances natural language generation (NLG) by incorporating external retrieval techniques to expand the model’s knowledge base. Traditional large language models (LLMs), such as GPT-3 and BERT, which have been pretrained on massive corpora, are vulnerable to issues such as hallucinations, where the models generate believable but incorrect information due to the employment on only their internal representations of knowledge. Due to their inability to efficiently update their knowledge bases without retraining, these models are less successful for dynamic, knowledge-intensive tasks such as fact verification and addressing open-domain questions [16]. The authors in [4] introduced the RAG architecture, which grounds the generated text on factual knowledge by retrieving relevant current external materials.

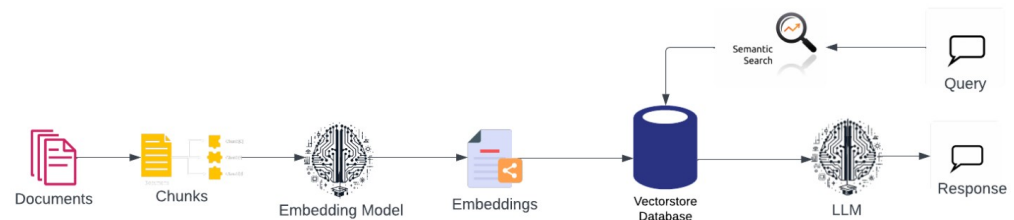
Retrieval-Augmented Generation (RAG) has been shown to be an effective and successful way to leverage external knowledge in large language models (LLMs). RAG retrieves relevant data based on the query and then instructs an LLM to provide a response within

the parameters of the retrieved data. This approach significantly expands LLM's low-cost access to a vast amount of data.

RAG has been used to improve the output of LLMs and to reduce hallucinations caused by them. LLMs can generate more consistent and factual results by providing the appropriate and relevant context.

### 3.2. Naïve RAGs

Naïve RAGs consist of the following components, as shown in Figure 1.



**Figure 1.** Naïve RAG architecture: a simple Retrieval-Augmented Generation architecture.

#### 3.2.1. Indexing

Raw data (PDF, HTML, Word, Markdown) is cleaned, extracted, and converted to plain text. The text is chunked, encoded into vectors using an embedding model, and stored in a vector database for an efficient similarity search.

#### 3.2.2. Retrieval

A user query is encoded into a vector and similarity scores are computed with indexed chunks. The top-K relevant chunks are retrieved for context expansion.

#### 3.2.3. Generation

The query and retrieved chunks form a prompt for the language model, which generates a response using either its internal knowledge or the provided context. Multiturn conversations incorporate dialogue history for coherent interactions.

#### 3.2.4. Challenges

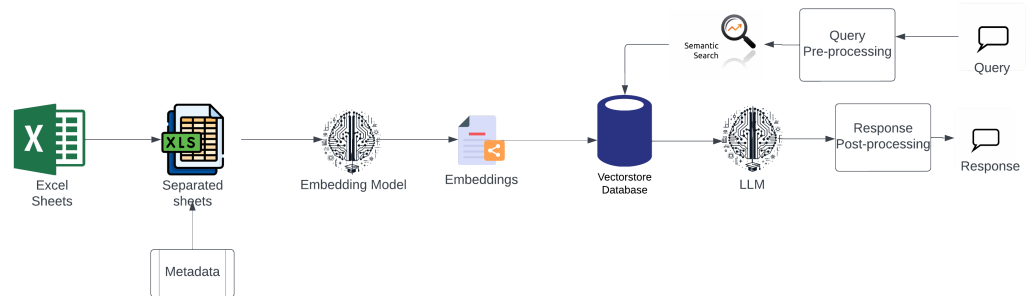
Naïve RAG faces significant challenges, including retrieval issues where irrelevant or crucial information is missed due to poor precision and recall. During generation, models can hallucinate, producing unsupported, biased, or toxic content. Augmentation adds complexity with disjointed or redundant outputs, difficulties in maintaining coherence, and assessing relevance. Over-reliance on retrieved content can result in repetitive responses lacking synthesis and insight, especially when a single retrieval fails to provide sufficient context.

More specifically, the main challenges with the Naïve RAG include:

1. Splitting into different chunks of data removes the relevant context and breaks the semantic structure of the data. When we query the data, we retrieve the closest-k (top  $k$  documents) similarity document, which causes many documents to be omitted, even if they were relevant to the query. Hence, it does not provide a concrete answer.
2. The chunking parameters (chunk size and chunk overlap) are highly data-dependent, and hence, we may not be able to generalise these variables for the entire data. Different RAG models need to be defined for different datasets.
3. Naïve RAG (and RAG in general) works best for continuous textual data that can be broken into chunks to query about a particular question. For textual summarisation or for querying information regarding the complete data, it fails considerably.

### 3.3. Advanced RAG

Advanced RAG, as shown in Figure 2, improves on naïve RAG by enhancing retrieval through pre- and post-retrieval strategies, refining indexing with techniques such as sliding windows, fine-grained segmentation, and metadata integration, and optimising the overall retrieval process for better efficiency and accuracy.



**Figure 2.** Advanced RAG architecture. Adding metadata and a series of preprocessing and postprocessing steps improves the quality of extraction and generation significantly. The processing methods used is strictly data-dependent.

The preretrieval process in Advanced RAG focuses on optimising indexing and refining user queries. Indexing improvements involve improving data granularity, structuring indexes, adding metadata, and employing mixed retrieval. Query optimisation ensures clarity and relevance through techniques such as rewriting, transformation, and expansion (Please see Table 1). Post-retrieval efforts prioritise reranking chunks and compressing context to highlight essential information, reducing information overload for LLMs by emphasising critical sections and streamlining the input.

**Table 1.** Complete RAG architecture for extracting and querying information from an Excel spreadsheet.

Component	Techniques/Approaches
Data Extraction & Preprocessing	Data cleaning, normalisation, handling missing values, chunking rows/columns, metadata annotation
Indexing	Dense vectorisation (e.g., using language model embeddings), column-specific indexing, hybrid indexing (combining dense and sparse retrieval)
Query Optimisation	Query parsing, query expansion (e.g., synonyms), column filtering, semantic matching
Metadata Utilisation	Adding row-level and sheet-level metadata, hierarchical metadata organisation for filtering and context
Retrieval	Contextual retrieval (extracting related rows/columns), multihop retrieval across sheets, reranking using graph or similarity models
Post-Retrieval Processing	Answer extraction (cell or range selection), data aggregation (sums, averages, counts), formatting before passing to the generator
Generator Optimisation	Prompt engineering, using templates for predictable responses, building in error handling for missing or ambiguous data
Automation & Integration	Automated data syncing when Excel updates occur, API integration (e.g., using Python libraries such as pandas and openpyxl), version control, and external tool integration

### 3.4. Indexing

During the *Indexing phase*, documents are processed, segmented, and converted to embeddings stored in a vector database. Effective index construction ensures accurate context retrieval. Key approaches include:

**Chunk Strategy:** Documents are divided into fixed-size chunks (for example, 100–512 tokens). Larger chunks provide more context but increase noise and cost, while smaller ones reduce noise but risk losing meaning. Advanced methods like recursive splits, sliding windows, and *Small2Big* improve retrieval by balancing semantic completeness and context, with sentences as retrieval units and surrounding sentences as additional context.

**Metadata Attachments:** Chunks are enriched with metadata (e.g., page number, author, timestamp) for filtered and time-aware retrieval, ensuring updated knowledge. Artificial metadata, such as paragraph summaries or hypothetical questions (Reverse HyDE), reduces semantic gaps by aligning document content with user queries.

**Structural Index:** Hierarchical indexing organises documents in parent–child structures, linking chunks with summaries for efficient traversal. *Knowledge Graphs (KG)* [17] further enhance retrieval by mapping relationships between concepts and structures, improving reasoning, and reducing inconsistencies in multidocument environments.

### 3.5. Query Optimisation

Query Optimisation in RAG systems addresses problems that arise from imprecise, ambiguous, or complex user queries that hinder effective retrieval. To improve outcomes, query expansion techniques like multiquery generation and subquery planning enrich the context by breaking down complex queries or generating parallel, well-designed queries through prompt engineering. Validation methods, such as Chain-of-Verification [18] (CoVe), ensure reliability by reducing hallucinations in expanded queries. Query transformation further enhances retrieval by refining the original query using techniques such as query rewriting with LLMs or smaller models (e.g., RRR [19,20]), generating hypothetical documents with HyDE [21], or employing step-back prompting [22] to create high-level conceptual queries that complement the original. Additionally, query routing directs queries to specific RAG pipelines according to context. Metadata routers filter results using keywords and metadata, while semantic routers leverage the query’s meaning, with hybrid approaches combining both methods for robust routing. Together, these techniques improve query clarity, retrieval precision, and adaptability across diverse scenarios.

For tabular data like excel sheets, we can use query optimisation techniques like (a) Query Parsing, which break down the user query into structured components; (b) Column filtering, i.e., query keywords, which can be used to filter relevant columns; and (c) semantic matching by using embeddings to match queries with semantically similar rows and columns.

### 3.6. Metadata Utilisation

Different kinds of metadata may be utilised to improve the precision of retrieval.

1. Row-level metadata: Add tags or labels for each row according to its content.
2. Sheet-level metadata: Annotate each sheet with the descriptive metadata.
3. Hierarchical metadata: Organise the metadata hierarchically for better filtering.

### 3.7. Retrieval Techniques

Optimising the retrieval step is critical in an RAG pipeline (please see Table 2), and some approaches to address this aspect are as follows.

1. Contextual retrieval: Retrieval of not just the top row but also related rows and columns that provide context.
2. Multihop Retrieval: For complex queries, retrieve data from multiple sheets and combine results.
3. Reranking: Using a reranking model to prioritise retrieved rows based on relevance.

Furthermore, optimisation techniques include:

- Prompt Engineering: Designing prompts that clearly instruct the language model on how to use retrieved tabular data.
- Template-Based Responses: Using templates for predictable queries and to get specific responses.
- Error handling: Including instructions in the prompts to handle exceptional cases where no relevant data are found so that the generator does not hallucinate.

**Table 2.** Summary of techniques for improving RAG models.

Step	Techniques
Preprocessing	Data cleaning, chunking, metadata annotation
Indexing	Vectorisation, column-specific indexing, hybrid indexing
Query Optimisation	Query parsing, column filtering, semantic matching
Metadata Utilisation	Row-level tags, sheet-level metadata, hierarchical organisation
Retrieval	Contextual retrieval, multihop retrieval, reranking
Post-Retrieval	Answer extraction, aggregation, formatting
Generator Optimisation	Prompt engineering, template-based responses, error handling
Automation	Automated updates, API integration, version control

### 3.8. Tools and Frameworks Used

For the purpose of our case, we are implementing RAG using different industry standard tools, such as Langchain and LlamaIndex to make the pipeline and LlamaParse and Pandas to read and parse the files. OpenAI LLM models are used for word embedding and as generators.

### 3.9. Preprocessing

The datasets employed in this study consisted of Excel-based spreadsheets characterised by non-standardised formatting and color-coding schemes optimised for human readability rather than computational processing. Prior to analysis, extensive data preprocessing was performed to transform these files into a machine-readable format. This preprocessing pipeline encompassed column standardisation, header normalisation, and the strategic insertion of delimiters between council-specific data segments to facilitate automated extraction workflows.

### 3.10. Agentic RAG

Agentic RAG (see Figure 3) transcends the limitations of traditional RAG systems by incorporating AI agents that can perceive their environment, make decisions, and take actions autonomously [23]. While conventional RAG follows static workflows with limited adaptability, agentic RAG enables dynamic decision making and workflow optimisation through agent-based autonomy. This paradigm shift addresses challenges in multistep reasoning and complex task management that traditional systems struggle to handle.

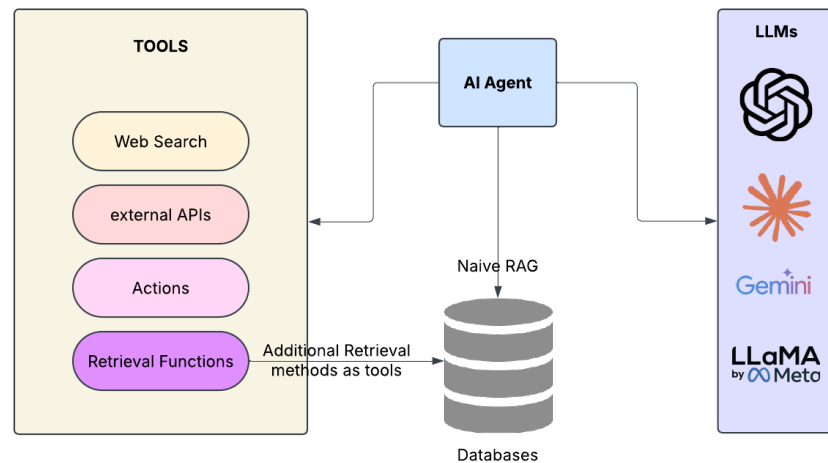
The fundamental distinction lies in how agentic RAG transforms the retrieval process from passive information retrieval to proactive problem solving. Traditional RAG systems are reactive data retrieval tools with no inherent ability to adapt to changing contexts, whereas agentic RAG transitions to adaptive, intelligent problem solving with self-assessment capabilities.

The architecture of agentic RAG systems (Figure 3) typically involves autonomous agents that leverage key agentic design patterns, such as

1. Reflection: Agents evaluate retrieved information and self-critique their outputs;

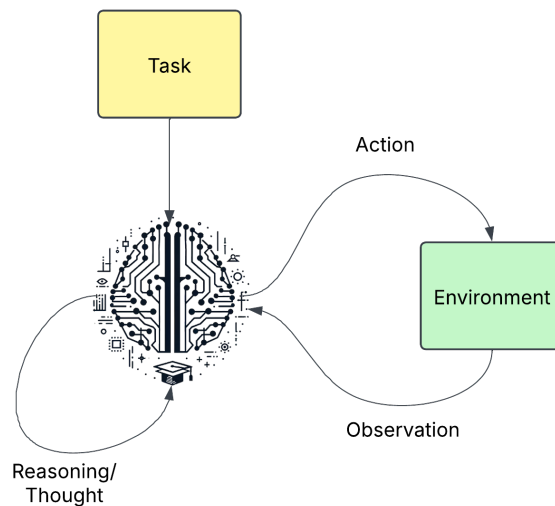
2. Planning: Systems break down complex queries into manageable subtasks;
3. Tool use: Agents can access external resources beyond the initial knowledge base;
4. Multiagent collaboration: Networks of specialised agents work together, checking each other’s work.

These systems operate on iterative refinement cycles that progressively improve both the relevance and diversity of retrieved documents. The autonomous agents dynamically manage retrieval strategies and adapt workflows to meet complex requirements, enabling unparalleled flexibility and context awareness.



**Figure 3.** Retrieval agent: personalised tools are provided to the retrieval AI agent to automate the process to retrieve and generate the best response.

In general, AI agents follow the cycle of (see Figure 4) Thought, Action, and Observation [24].



**Figure 4.** ReAct: reasoning action cycle for AI Agents.

1. Autonomous Agent Integration: Agentic Retrieval-Augmented Generation (RAG) [23] integrates autonomous AI agents capable of reflection, planning, tool use, and multi-agent collaboration. These agents operate beyond static retrieval pipelines by dynamically adapting retrieval strategies based on intermediate outputs and evolving task demands. This agentic control enables complex reasoning, improved interpretability,

and task-specific optimisation, making the system more responsive and intelligent in handling diverse and evolving information needs.

2. **Iterative Refinement:** A key differentiator of Agentic RAG is its iterative approach to information retrieval. Rather than relying on a single retrieval step, the system performs multiple refinement cycles, progressively improving both the relevance and diversity of retrieved documents. For example, Vendi-RAG [25] introduced a new diversity metric—the Vendi Score—alongside large language model (LLM) judges to evaluate and enhance the quality of responses across iterations. This iterative refinement mechanism leads to more robust and contextually nuanced outputs.
3. **Context-Aware Retrieval:** Agentic RAG systems place significant emphasis on semantic and structural context. Frameworks such as Citation Graph RAG [26] (CG-RAG) leverage citation graphs to model interdocument relationships, thereby improving contextual grounding and retrieval precision. Similarly, systems like Golden-Retriever [27] improve retrieval quality by preprocessing queries to resolve domain-specific terminology and jargon, ensuring better alignment between user intent and retrieved content. These strategies collectively improve the depth and relevance of the knowledge retrieved, particularly in technical or specialised domains.

#### 4. RAG for Knowledge Asset Generation

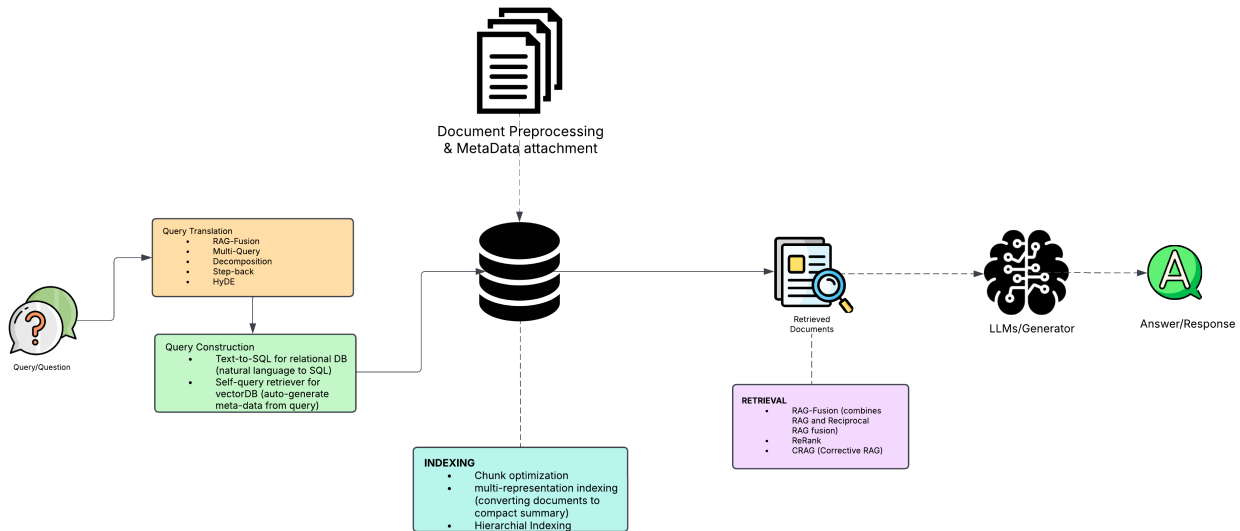
Embedding models in RAG (Retrieval-Augmented Generation) systems are neural network models that convert text into dense vector representations (embeddings) that capture semantic meaning. These vectors enable similarity-based retrieval of relevant information from knowledge bases to augment language model outputs.

Embedding models serve as the foundation of the retrieval component in RAG systems by transforming document content into vector embeddings that capture semantic relationships between texts [28], allowing similarity-based searches that go beyond simple keyword matching [29], creating searchable vector representations of information stored in knowledge bases [30] and supporting efficient document retrieval from vector databases for context augmentation [31]. These models essentially bridge the gap between raw text and mathematical representations that algorithms can process, allowing RAG systems to identify and retrieve the most relevant information based on meaning rather than just lexical overlap.

The Massive Text Embedding Benchmark (MTEB) [32] is a comprehensive evaluation framework designed to assess the performance of text embedding models across diverse tasks and languages. Introduced to address the limitations of evaluating embeddings on only a small set of datasets from a single task, MTEB spans 8 embedding tasks covering 58 datasets and 112 languages.

The MTEB leaderboard [33] of the company Hugging Face contains the state-of-the-art embedding models benchmarked and evaluated on different metrics. Generally, in all RAG models, we see that the openAI 'text-embedding-ada-002' model has been used. Using the latest models from the benchmark is bound to get better performance. At the time of this writing, 'gemini-embedding-exp-03-07' seems to be the state-of-the-art embedding model available.

Figure 5 shows the Advanced RAG that specifically addresses knowledge generation. This will be the focus of the approach introduced in this work.



**Figure 5.** Advanced RAG architecture. This figure depicts different techniques that can be used to improve the RAG architecture.

## 5. Knowledge Asset Generation

After developing the RAG pipeline for our use, knowledge assets will be generated using this for multiple sheets of data. This will provide qualitative and analytical information and insights derived from the data. Using a particular RAG pipeline ensures that the generated results are consistent and comparable. A consistent prompt template shall be used to initialise the RAG pipeline, and further prompts from the user shall only be used to fine-tune the answers. This shall ensure that the initially generated answers are all in a similar format and comparable to each other. It would make it easier for the reader or the analyst to compare between different sectors and save a significant amount of time.

## 6. Case Study: Challenges of Using LLMs to Analyse Complex Datasets

In this section, a case study based on the Baseline Review 2024 for the Liverpool City Region [9] is introduced and evaluated to illustrate and discuss the particular problems faced when dealing with large datasets using large language models.

The Baseline Review 2024 for the Liverpool City Region (LCR) provides an in-depth economic assessment of the Professional Business Services (PBS) sector, using large, complex, and diverse datasets. The study highlighted several challenges associated with the use of AI to analyse such data effectively. These challenges are categorised into three main areas: data structuring issues, prompt development issues, and testing/retesting for consistency.

### 6.1. Data Structuring Issues

AI systems face significant hurdles when processing diverse datasets due to inconsistencies in format, quality, and structure. Key challenges include:

1. **Inconsistent Formats:** Data from sources such as CSVs, databases, and APIs often contain mismatched fields, naming conventions, or missing values. For example, the report had to standardise SIC codes and reconcile data across LCR boroughs, tasks that require robust preprocessing pipelines.
2. **Unstructured Data Complexity:** Although the review primarily used quantitative data, qualitative insights (e.g., growth needs and strategic support plans) required transformation techniques such as tokenisation or embedding before AI models could process them meaningfully.

3. Noise and redundancy: Duplicated or outdated company records can mislead AI models during analysis. The report noted potential oversimplification risks when using SIC codes.
4. Scalability: Processing datasets for over 12,000 firms across various lifecycle stages demands scalable AI systems in terms of both computational power and data governance.

### 6.2. Prompt Development Issues

Effective AI analysis is highly dependent on well-designed prompts tailored to specific contexts. However, several challenges were identified:

1. Ambiguity in prompts: Vague prompts such as “what do scale-ups need in LCR?” yield generic results. Tailored prompts are necessary to address specific business lifecycle stages (e.g., scale-up vs. repeat scale-up).
2. Overfitting to Context: Customised prompts for the LCR context may not generalise well to other regions, limiting their utility for national benchmarking.
3. Context Limitations: Token limits in AI models like GPT can truncate or lose critical information when attempting to include sectoral comparisons, borough-level insights, and lifecycle requirements in a single prompt.
4. Skill Gap in Prompt Engineering: Creating effective prompts requires expertise in both AI behaviour and domain knowledge. Without domain-aware developers, replicating expert analysis becomes risky.

### 6.3. Testing and Retesting for Consistency

Ensuring consistent outputs from AI models is vital for longitudinal studies and policy recommendations. However, several obstacles arise:

1. Non-Determinism of AI Models: Generative AI can produce different results for identical inputs, complicating the task of generating repeatable recommendations.
2. Version Drift: Updates to AI models (for example, new GPT versions) may alter the output over time. For studies spanning multiple years, version control is essential.
3. Validation Difficulty: Defining “correct” AI analysis is challenging without ground-truth benchmarks, especially when forecasting economic outcomes.
4. Feedback loops: Refining of prompts based on AI-generated insights risks confirming user bias rather than uncovering objective findings.

The Baseline Review 2024 underscores the importance of structured processes when applying AI to analyse layered, regional, and longitudinal datasets. Key recommendations include:

- Implement robust data structuring practices to standardise input and maintain accuracy.
- Design clear and context-sensitive prompts that reflect the complexity of business lifecycles, regional variations, and sector interdependencies.
- Establish thorough testing and validation protocols to ensure repeatability and reliability of AI outputs.

By systematically addressing these challenges, AI can enhance regional economic analysis while maintaining precision and consistency.

## 7. RAG vs. Large Context Window LLMs

The evolution of large language models (LLMs) has sparked a critical debate between the extension of context windows and the use of Retrieval-Augmented Generation (RAG) for the handling of long-context tasks. Since LLMs have been becoming stronger with larger context windows (Gemini [34] 2.0 Flash, Claude 3.7 Sonnet, etc.), and even beyond 2 million tokens [35], it has become easier to process entire codebases and lengthy scientific papers.

Longer contexts risk incorporating irrelevant details, reducing accuracy [36]. Also, the computational cost is high for longer context models, as quadratic scaling of attention mechanisms strains resources. Another problem is that LLMs are more apt to pick up on important information appearing at the start or end of a long prompt rather than buried in the middle [37], so a lot of context information is lost when using these large context models.

RAG shines in these aspects, as the selective retrieval of information avoids processing entire corpora, making it cost-efficient. The dynamic knowledge integration makes it critical for real-time data.

### 8. Experiments, Results, and Evaluation

The RAG TRIAD framework, consisting of Context Relevance, Groundedness or Faithfulness, and Answer Relevance, can be used to evaluate the RAG system to assess the core components as per Table 3.

**Table 3.** TRIAD framework: component metrics and data mapping.

Component	Metrics
Context Relevance	Precision, Recall, MRR (Mean Reciprocal Rank), MAP (Mean Average)
Faithfulness	Factual consistency
Answer Relevance	BLEU, ROUGE, embedding similarity

There are multiple metrics and tools to measure the RAG system; here, we use the library/system RAGAS [5] (Retrieval-Augmented Generation assessment) to evaluate our RAG model and compare the different architectures. The most important stage of building the RAG system was preprocessing (see Table 4) the data since context and semantics are very important for any tabular data to be used in the RAG system. RAG systems have usually been used in text document fetching, and most methods rely on chunking the texts into different documents. In doing so, context and semantics are lost in the RAG system. Therefore, to compare between various sections in tabular data, a lot of preprocessing has to be performed so the semantic and structural information is not lost in the indexing/chunking process.

**Table 4.** Table comparing queries and responses to different models. This shows how preprocessing improves the retrieval and accuracy of the retrieval model.

No.	Question/Query	Naive RAG Answer/Response	RAG (With Preprocessing Using LlamaParse) Retrieval k = 7
1	Tell me all the locations mentioned in this data?	The location mentioned in the data are Liverpool	The locations mentioned in the data are LCR, Wirral, Liverpool, Knowsley, St Helens, Halton, and Sefton.
2	What are the total companies and scale-ups in Sefton?	In Sefton, there are a total of 793 companies and 32 visible scale-ups.	In Sefton, there are a total of 2095 companies. Among these, there are 81 visible scale-ups.
3	What are the total companies and scale-ups in Liverpool?	Liverpool has a total of 1833 companies and 72 visible scale-ups.	In Liverpool, there are a total of 5142 companies. Among these, there are 185 visible scale-ups.

In this particular case, the sheet data with different councils have to be preprocessed thoroughly to make a more holistic and simpler tabular sheet for our use in RAG. With the advent of large context windows LLMs, it is easier to feed the whole data to the LLMs, but RAG systems are still used for relevancy and accuracy and to make sure that the LLMs do not hallucinate.

The following metrics measure the different components of the architecture.

1. Retriever Component: Context Precision, Context Recall;

## 2. Generator Component: Faithfulness, Answer Relevancy.

$$\text{Context Precision} = \frac{\text{Number of relevant contexts retrieved}}{\text{Total number of contexts retrieved}}$$

$$\text{Context Recall} = \frac{\text{Number of relevant contexts retrieved}}{\text{Total number of relevant contexts available}}$$

$$\text{Faithfulness} = \frac{\text{Number of claims supported by context}}{\text{Total number of claims in the answer}}$$

$$\text{Answer Relevancy} = \frac{\text{Relevance score of answer to query}}{\text{Maximum possible relevance score}}$$

Context Precision is a metric for evaluating Retrieval-Augmented Generation (RAG) systems that measures the proportion of relevant chunks in retrieved contexts. It calculates the mean precision@k across all retrieved chunks, where precision@k represents the ratio of relevant chunks at rank k to the total number of chunks at that rank. Context Recall measures the proportion of relevant information from the ground truth that is successfully retrieved by the system. It evaluates the completeness of retrieval by determining what fraction of all relevant information available was actually captured in the retrieved contexts. Faithfulness measures how accurately the generated response reflects the information present in the retrieved contexts. It evaluates whether the model's output is grounded in and consistent with the provided source material, ensuring the response does not introduce hallucinations or contradictions to the retrieved evidence.

In Table 5, (k = 3) 3 documents are retrieved, each for every query. Since each of these documents refers to a specific English council, the retrieval metrics are likely to be suboptimal.

**Table 5.** Evaluation metrics: Context Precision measures the proportion of relevant chunks in retrieved contexts by calculating the mean precision@k across all retrieved chunks. Context Recall measures the proportion of relevant information from the ground truth that is successfully retrieved by the system. Faithfulness measures how accurately the generated response reflects the information present in the retrieved contexts.

S No.	Questions	Context Precision	Context Recall	Faithfulness	Answer Relevancy
1	Tell me the number of the Scale-up Potential SMEs in Sefton	0	0	0	0.98
2	Tell me the number of the Visible Scale-ups businesses in Wirral	0.5	0	0	0.97
3	Tell me the number of the Scale-up Potential SMEs in Wirral	0.5	0	0	0.98
4	Which council has the highest number of Scale-up Potential SMEs?	0.33	0	0	0.94
5	Tell me all the councils mentioned	1	1	1	0.94

**Ground Truth and Reference Data:** For our evaluation, we used a curated set of representative queries about council data (as shown in Table 5). The ground truth for Context Recall was established based on the complete dataset knowledge, allowing RAGAS to determine what relevant information should ideally be retrieved for each query.

**Limitations and Human Evaluation:** While our current evaluation relies on automated metrics through RAGAS, we acknowledge that human evaluation would provide additional validation of our results. The relatively low Context Precision and Context Recall scores (particularly for queries 1–4) reflect the challenge of retrieving relevant information from council-specific documents when k = 3 documents are retrieved per query. Future work

will incorporate human evaluation to validate these automated assessments and provide qualitative analysis of answer quality.

The RAGAS framework has been validated against human judgments in prior work, providing confidence in the reliability of these automated metrics for comparative analysis across different RAG architectures.

Agentic RAG (as shown in Figure 6) has been used to improve the retrieval and generation as the agent goes through a *thought, action, and observation* cycle before coming up with the actual result.

```

New run
What are the locations/councils mentioned in this dataset located at
/Users/james/Developer/TPG_Research/ehu/sheet1.xlsx?
OpenAIServerModel - gpt-4o-mini

Initial plan
Here are the facts I know and the plan of action that I will follow to solve the task:

## Facts survey
### 1.1. Facts given in the task
- There is a dataset located at the path '/Users/james/Developer/TPG_Research/ehu/sheet1.xlsx'.
- The task is to identify the locations or councils mentioned in this dataset.

### 1.2. Facts to look up
- The actual content of the Excel file 'sheet1.xlsx' to see what locations or councils are mentioned within it.
- This can be found by using the 'extract_excel_data' tool to access and extract the data from the provided Excel file.

### 1.3. Facts to derive
- The names of locations or councils identified within the dataset from the extracted data.
- Any additional context about the locations or councils (e.g., their geographical or administrative significance) may also be derived through further web searches if necessary.

## Plan
1. Use the 'extract_excel_data' tool to extract and parse the data from the Excel file located at '/Users/james/Developer/TPG_Research/ehu/sheet1.xlsx'.
2. Analyze the extracted data to identify the specific mentions of locations or councils.
3. If further information on these locations or councils is needed, perform a 'web_search' for each location or council identified.
4. Compile the results from the dataset and any additional information gathered from web searches.
5. Use the 'final_answer' tool to present the completed answer regarding the locations or councils mentioned in the dataset.

Step 1
- Executing parsed code:
data = extract_excel_data(file_path="/Users/james/Developer/TPG_Research/ehu/sheet1.xlsx")
print(data)

Started parsing the file under job_id 2fc314ac-fcb4-4e3b-9a80-732506351f81

Step 5
- Executing parsed code:
# Extracting the actual text content from the document to analyze its structure.
dataset_text = dataset_text
lines = dataset_text.split('\n')

# Initialize an empty set to hold unique locations and councils.
locations_councils = set()

# Iterate through the lines to find potential locations.
for line in lines:
    columns = line.split(',')
    # Assuming the first column indicates location/council names
    if len(columns) > 1:
        location = columns[1].strip() # Get the name and strip whitespace
        if location:
            locations_councils.add(location)

# Convert the set back to a list for easier readability.
locations_councils_list = list(locations_councils)
print("Locations/Councils Found:", locations_councils_list)

Step 7
- Executing parsed code:
# Given that we have already identified a preliminary list of locations/councils,
# I will filter them to ensure we only keep recognized council/entity names.

# Initial list of results to filter
raw_results = [
    'Knowsley', 'Phases of Growth', 'Wirral', 'Sefton',
    'St Helens', 'Phases of Growth (SMEs)', 'LCR', 'Liverpool',
    'Scaleup Potential Businesses (10 to 20 years)', 'Start-up Businesses (0 to 5 years)',
    'Scaleup Businesses (20 to 50 years)', 'Repeat Scaleup Businesses (50 years plus)',
    'Large', 'SME', 'Halton', 'High Growth Potential Entrepreneurial Enterprises (5 to 10 years)'
]

# Validating and cleaning results to exclude non-location entries
valid_locations_councils = [
    location for location in raw_results
    if location in ['Knowsley', 'Wirral', 'Sefton', 'St Helens', 'LCR', 'Liverpool', 'Halton']
]

print("Validated Locations/Councils Found:", valid_locations_councils)

Step 8
- Executing parsed code:
final_answer(valid_locations_councils)

Out - Final answer: ['Knowsley', 'Wirral', 'Sefton', 'St Helens', 'LCR', 'Liverpool', 'Halton']

[Step 8: Duration 4.35 seconds] Input tokens: 114,499 | Output tokens: 1,364

['Knowsley', 'Wirral', 'Sefton', 'St Helens', 'LCR', 'Liverpool', 'Halton']

```

**Figure 6.** Agentic RAG workflow: this figure shows some steps the AI agent takes to solve the task using the thought, action, and observation approach. At each step, the agent makes a plan, implements an action, observes the result, and repeats the process until it finds the optimal answer to the task.

## 9. Conclusions and Discussion

The results of this study highlight both the promise and the complexity of using Retrieval-Augmented Generation (RAG) systems for knowledge asset generation and action driver creation. RAG architectures, especially when advanced preprocessing and indexing techniques are applied, can significantly improve the factual consistency and relevancy of outputs compared to standard LLM approaches. This is particularly evident in scenarios involving large heterogeneous datasets, where naïve RAG methods often struggle due to context fragmentation and data-dependent chunking parameters. Incorporating metadata, semantic-sensitive segmentation, and query optimisation into advanced RAG pipelines helps to preserve context and improve retrieval precision, ultimately leading to more accurate and actionable insights.

However, several challenges remain. The effectiveness of RAG systems is highly dependent on robust data structuring and preprocessing, as inconsistencies or redundancies in the source data can propagate errors through the pipeline. Additionally, the development of effective prompts remains a non-trivial task, requiring both domain expertise and a nuanced understanding of AI behaviour. The evaluation results also reveal limitations in recall and faithfulness metrics, particularly when dealing with complex or tabular data, underscoring the need for further refinement in retrieval and generation strategies. In the future, agentic RAG systems—which employ autonomous agents for iterative reasoning and context-aware retrieval—show promise in addressing some of these challenges by enabling more adaptive and robust information synthesis. However, systematic validation and careful pipeline design remain essential to fully realise the benefits of RAG in practical, knowledge-intensive applications. Future research will focus on complementing the potential of domain-specific knowledge LLMs with erroneous and outdated information mitigation. This will lead to the design and creation of intelligent and tailored applications that can drive innovation, improve decision making, and improve customer and employee experiences. As businesses increasingly navigate complex information environments, the ability of RAG to provide accurate, contextually relevant, and timely information positions it as a key enabler technology to achieve competitive advantage and foster data-driven operational excellence.

**Author Contributions:** Conceptualisation, A.J. and M.T.; methodology, A.J. and M.T.; software, A.J.; validation, A.J., M.T., and S.B.; formal analysis, A.J. and M.T.; investigation, A.J., M.T., and S.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available in a publicly accessible repository, which can be retrieved here: <https://www.liverpoolchamber.org.uk/wp-content/uploads/2024/07/baseline-review-2024-executive-summary.pdf> (accessed on 20 March 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

RAG	Retrieval-Augmented Generation
LLM	Large language models
MTEB	Text Embedding Benchmark

## References

1. Ferreira Barros, C.; Borges Azevedo, B.; Graciano Neto, V.V.; Kassab, M.; Kalinowski, M.; do Nascimento, H.A.D.; Bandeira, M.C. Large Language Model for Qualitative Research—A Systematic Mapping Study. *arXiv* **2024**, arXiv:2411.14473.
2. Fischer, T.; Biemann, C. Exploring Large Language Models for Qualitative Data Analysis. In Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities, Miami, FL, USA, 15–16 November 2024; pp. 423–437.
3. Xu, Z.; Jain, S.; Kankanhalli, M. Hallucination is inevitable: An innate limitation of large language models. *arXiv* **2024**, arXiv:2401.11817.
4. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
5. Es, S.; James, J.; Espinosa-Anke, L.; Schockaert, S. Ragas: Automated evaluation of retrieval augmented generation. *arXiv* **2023**, arXiv:2309.15217.
6. Chen, J.; Lin, H.; Han, X.; Sun, L. Benchmarking large language models in retrieval-augmented generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 17754–17762.
7. Li, X.; Ouyang, J. A Systematic Investigation of Knowledge Retrieval and Selection for Retrieval Augmented Generation. *arXiv* **2024**, arXiv:2410.13258.
8. Hoshi, Y.; Miyashita, D.; Ng, Y.; Tatsuno, K.; Morioka, Y.; Torii, O.; Deguchi, J. Ralle: A framework for developing and evaluating retrieval-augmented large language models. *arXiv* **2023**, arXiv:2308.10633.
9. Liverpool Chamber of Commerce. *Baseline Review 2024: Professional Business Services—Identifying Future Growth Potential Liverpool City Region—Executive Summary*; Technical Report; Liverpool Chamber of Commerce: Liverpool, NY, USA, 2024.
10. Arslan, M.; Munawar, S.; Cruz, C. Business insights using RAG–LLMs: A review and case study. *J. Decis. Syst.* **2024**. [[CrossRef](#)]
11. Arslan, M.; Cruz, C. Business-RAG: Information Extraction for Business Insights. In *21st International Conference on Smart Business Technologies*; SCITEPRESS-Science and Technology Publications: Setúbal, Portugal, 2024; pp. 88–94.
12. Ramalingam, S. *RAG in Action: Building the Future of AI-Driven Applications*; Libertatem Media Private Limited: Ahmedabad, India, 2023.
13. Lovtsov, V.A.; Skvortsova, M.A. Automated Mobile Operator Customer Service Using Large Language Models Combined with RAG System. In Proceedings of the 2025 7th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Cairo, Egypt, 8–10 April 2025; pp. 1–6.
14. Hang, C.N.; Yu, P.D.; Tan, C.W. TrumorGPT: Graph-Based Retrieval-Augmented Large Language Model for Fact-Checking. *IEEE Trans. Artif. Intell.* **2025**. [[CrossRef](#)]
15. Bruckhaus, T. Rag does not work for enterprises. *arXiv* **2024**, arXiv:2406.04369.
16. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
17. Wang, Y.; Lipka, N.; Rossi, R.A.; Siu, A.; Zhang, R.; Derr, T. Knowledge graph prompting for multi-document question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–28 February 2024; Volume 38, pp. 19206–19214.
18. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-Verification Reduces Hallucination in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 3563–3578.
19. Ma, X.; Gong, Y.; He, P.; Zhao, H.; Duan, N. Query Rewriting in Retrieval-Augmented Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023.
20. Peng, W.; Li, G.; Jiang, Y.; Wang, Z.; Ou, D.; Zeng, X.; Xu, D.; Xu, T.; Chen, E. Large Language Model based Long-tail Query Rewriting in Taobao Search. In Proceedings of the ACM Web Conference, Singapore, 13–17 May 2024; pp. 20–28.
21. Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise Zero-Shot Dense Retrieval without Relevance Labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; Association for Computational Linguistics: Toronto, ON, Canada, pp. 1762–1777.
22. Zheng, H.S.; Mishra, S.; Chen, X.; Cheng, H.T.; Chi, E.H.; Le, Q.V.; Zhou, D. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
23. Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, T.T. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. *arXiv* **2025**, arXiv:2501.09136.
24. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
25. Rezaei, M.R.; Dieng, A.B. Vendi-rag: Adaptively trading-off diversity and quality significantly improves retrieval augmented generation with llms. *arXiv* **2025**, arXiv:2502.11228.

26. Hu, Y.; Lei, Z.; Dai, Z.; Zhang, A.; Angirekula, A.; Zhang, Z.; Zhao, L. CG-RAG: Research Question Answering by Citation Graph Retrieval-Augmented LLMs. *arXiv* **2025**, arXiv:2501.15067.
27. An, Z.; Ding, X.; Fu, Y.C.; Chu, C.C.; Li, Y.; Du, W. Golden-Retriever: High-Fidelity Agentic Retrieval Augmented Generation for Industrial Knowledge Base. *arXiv* **2024**, arXiv:2408.00798.
28. Geetha, M.; Thirukumar, G.; Pradakshana, C.; Sudharsana, B.; Ashwin, T. Conversational AI Meets Documents Revolutionizing PDF Interaction with GenAI. In Proceedings of the 2024 International Conference on Emerging Research in Computational Science (ICERCS), Coimbatore, India, 12–14 December 2024; pp. 1–6.
29. Kang, B.; Kim, J.; Yun, T.R.; Kim, C.E. Prompt-rag: Pioneering vector embedding-free retrieval-augmented generation in niche domains, exemplified by korean medicine. *arXiv* **2024**, arXiv:2401.11246.
30. An, H.; Narechania, A.; Wall, E.; Xu, K. VITALITY 2: Reviewing Academic Literature Using Large Language Models. *arXiv* **2024**, arXiv:2408.13450.
31. Gopi, S.; Sreekanth, D.; Dehbozorgi, N. Enhancing Engineering Education Through LLM-Driven Adaptive Quiz Generation: A RAG-Based Approach. In Proceedings of the 2024 IEEE Frontiers in Education Conference (FIE), Washington, DC, USA, 13–16 October 2024; pp. 1–8.
32. Muennighoff, N.; Tazi, N.; Magne, L.; Reimers, N. MTEB: Massive Text Embedding Benchmark. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–6 May 2023; pp. 2014–2037.
33. Muennighoff, N.; Tazi, N.; Magne, L.; Reimers, N. MTEB Leaderboard. Hugging Face Spaces. 2023. Available online: <https://huggingface.co/spaces/mteb/leaderboard> (accessed on 20 March 2025).
34. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
35. Ding, Y.; Zhang, L.L.; Zhang, C.; Xu, Y.; Shang, N.; Xu, J.; Yang, F.; Yang, M. Longrope: Extending llm context window beyond 2 million tokens. *arXiv* **2024**, arXiv:2402.13753.
36. Li, X.; Cao, Y.; Ma, Y.; Sun, A. Long Context vs. RAG for LLMs: An Evaluation and Revisits. *arXiv* **2024**, arXiv:2501.01880.
37. Wan, D.; Vig, J.; Bansal, M.; Joty, S. On Positional Bias of Faithfulness for Long-form Summarization. *arXiv* **2024**, arXiv:2410.23609.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.