applied sciences

MDPI

Article

Evaluating Retrieval-Augmented Generation Models for Financial Report Question and Answering

Ivan Iaroshev [1], Ramalingam Pillai [1], Leandro Vaglietti [1] and Thomas Hanne [2,*]

[1] School of Business, University of Applied Sciences and Arts Northwestern Switzerland, 4600 Olten, Switzerland
[2] Institute for Information Systems, University of Applied Sciences and Arts Northwestern Switzerland, 4600 Olten, Switzerland
* Correspondence: thomas.hanne@fhnw.ch; Tel.: +41-62-957-22-92

Abstract: This study explores the application of retrieval-augmented generation (RAG) to improve the accuracy and reliability of large language models (LLMs) in the context of financial report analysis. The focus is on enabling private investors to make informed decisions by enhancing the question-and-answering capabilities regarding the half-yearly or quarterly financial reports of banks. The study adopts a Design Science Research (DSR) methodology to develop and evaluate an RAG system tailored for this use case. The study conducts a series of experiments to explore models in which different RAG components are used. The aim is to enhance context relevance, answer faithfulness, and answer relevance. The results indicate that model one (OpenAI ADA and OpenAI GPT-4) achieved the highest performance, showing robust accuracy and relevance in response. Model three (MiniLM Embedder and OpenAI GPT-4) scored significantly lower, indicating the importance of high-quality components. The evaluation also revealed that well-structured reports result in better RAG performance than less coherent reports. Qualitative questions received higher scores than the quantitative ones, demonstrating the RAG's proficiency in handling descriptive data. In conclusion, a tailored RAG can aid investors in providing accurate and contextually relevant information from financial reports, thereby enhancing decision making.

Keywords: retrieval-augmented generation; large language models (LLMs); financial reports; question and answering with LLM

Citation: Iaroshev, I.; Pillai, R.; Vaglietti, L.; Hanne, T. Evaluating Retrieval-Augmented Generation Models for Financial Report Question and Answering. *Appl. Sci.* **2024**, *14*, 9318. https://doi.org/10.3390/app14209318

Academic Editor: Pedro Couto

Received: 16 September 2024
Revised: 7 October 2024
Accepted: 9 October 2024
Published: 12 October 2024

1. Introduction

Since the inception of Artificial Intelligence (AI), researchers have pursued the ambitious goal of developing machines capable of reading, writing, and conversing like humans. This occurred in the domain of natural language processing (NLP). According to Zhao et al. [1], NLP has a rich history: from the development of statistical language models (SLMs) to the rise of neural language models (NLMs), linguistic capabilities have significantly increased, enabling AI systems to understand and generate complex language patterns [1].

In recent years, NLP has witnessed remarkable progress with the introduction of large language models (LLMs). These models, which were extensively trained on vast amounts of textual data, demonstrated unprecedented proficiency in generating human-like text and performing language-based tasks accurately [2]. However, despite their advancements, LLMs still face several limitations. Burtsev et al. [3] outline three main limitations of LLMs. First, they struggle with complex reasoning, hindering their ability to draw accurate conclusions. Second, their knowledge or expertise is limited to training data, which may lead to failure in providing relevant information. Third, they may produce inaccurate outputs due to a lack of understanding of prompts, which makes them marginally helpful.

The rise of LLMs has also sparked interest in their application to specific tasks, prompting the emergence of an approach called retrieval-augmented generation (RAG). This

approach was first introduced by Guu et al. [4] and Lewis et al. [5]. RAG was devised to extend the capabilities of LLMs beyond conventional training data. By integrating a specialized body of knowledge, RAG enables LLMs to provide more accurate responses to user queries. In essence, RAG comprises two distinct phases: retrieval and generation. During the retrieval phase, defined sources (e.g., indexed documents) are analyzed to extract relevant information that is aligned with the user's prompt or question. The retrieved information is then seamlessly integrated with the user's prompt and forwarded to the language model. In the generative phase, the LLM leverages the augmented prompt and internal understanding to craft a tailored response that addresses the user's query effectively [6,7].

RAG is gaining momentum for its potential to improve LLM-generated responses by grounding models on the said external sources, thereby reducing issues such as inconsistency and hallucination [8,9].

1.1. Problem Statement

The integration of RAG into LLMs has rapidly gained traction, and it has emerged as a key strategy to increase the practical applicability of LLMs [8]. However, despite its potential to reduce factual inaccuracies or "hallucinations", there remains a notable gap in the literature regarding RAG's application in question and answering for a specific use case: private investors seeking to analyze the financial reports of banks effectively (see Section 2.4). The manual analysis of such reports is both time-consuming and labor-intensive. However, by leveraging RAG, investors can access domain-specific information from half-yearly or quarterly reports previously published by several banks. Through RAG-facilitated question-and-answer sessions, investors can make informed decisions based on accurate insights.

This study focuses on half-yearly and quarterly reports due to their regular publication frequency, ensuring an ample number of samples for analysis. In addition, these reports contain a better-manageable number of pages than annual reports, which may overwhelm an RAG system with their extensive content. Further details about the data collection process are provided in Section 3.2.

Currently, there is a lack of a straightforward RAG approach tailored to this use case, which hinders the empirical demonstrations of how RAG can be effectively adapted (Section 2.3). These experiments with different model configurations should focus on both the retrieval and generation components (Section 2.2) to achieve desirable outcomes across evaluation aspects (context relevance, answer faithfulness, and answer relevance, Section 2.3).

Therefore, the motivation to research RAG encompasses several crucial aspects. First, it aims to enhance the quality of responses across evaluation dimensions such as context relevance, answer faithfulness, and answer relevance by exploring different model configurations (e.g., different LLMs enriched with the same data sources). Second, it attempts to address the persistent challenges encountered by state-of-the-art LLMs, such as factual inaccuracies and hallucinations. By examining the practical implementation and effectiveness of RAG, particularly in specific domains like financial analysis for private investors, this research aims to provide valuable insights that bridge the gap between theoretical advancements and real-world applicability.

1.2. Thesis Statement and Research Questions

Our study aims to explore the potential of RAG for improving the accuracy and reliability of LLMs for analyzing half-yearly or quarterly reports. Through a range of experiments featuring various RAG model configurations, such as different LLMs and retrieval methods, we aim to examine and gauge differences and improvements in context relevance, answer faithfulness, and answer relevance in financial report analysis. Ultimately, the goal is to empower users to make informed decisions by providing accurate answers. To

achieve this, a thesis statement and main research question were formulated, supplemented by additional sub-questions.

Thesis Statement: The effectiveness of RAG in enhancing context relevance, answer faithfulness, and answer relevance for analyzing half-yearly or quarterly reports can be evaluated through empirical experiments with different model configurations.

Main Research Question: How can RAG be adapted to increase the context relevance, answer faithfulness, and answer relevance of LLM conclusions about several types of financial reports?

1. How can the retrieval component of RAG be set up to extract relevant information from half-yearly or quarterly reports?
2. What strategies can be employed to ensure that the generation component of RAG produces answers to the questions posed about half-yearly or quarterly reports?
3. How can the effectiveness of the RAG system for analyzing half-yearly or quarterly reports in the banking sector be reliably evaluated and validated?
4. How accurately do RAG-generated responses represent the information extracted from the half-yearly or quarterly reports of banking institutions?

2. Literature Review

In this review, RAG systems, including evaluation metrics, methods, and frameworks, are briefly discussed. In addition, a research gap is identified that requires further investigation.

Methodologically, the literature review was initiated with extensive searches on Google Scholar using broad keywords such as "Retrieval Augmented Generation" and "RAG" to capture general papers on the topic. As the investigation progressed, the search terms were refined to focus on specific aspects of RAG, including "Evaluation methods of RAG". This approach includes both forward and backward searches to ensure the comprehensive coverage of the relevant literature. While priority was given to peer-reviewed academic papers, a selection of non-peer-reviewed papers with substantial citations and reputable authors was also integrated, acknowledging the rapid advancements in LLM and RAG research.

2.1. History and Current State of RAG

The field of NLP has seen significant advancements with the emergence of LLMs like the GPT and LLama series, as well as models like Gemini [2,10,11]. Despite their successes, these models still need to cope with challenges such as outdated knowledge and a lack of domain-specific expertise [12,13]. One major issue is the generation of incorrect information, known as "hallucinations", especially when faced with queries beyond their training data [3,14]. These limitations highlight the need for additional safeguards before deploying LLMs in real-world settings [6].

A solution to address these challenges is RAG, which integrates external data retrieval into the generative process. RAG, pioneered by Lewis et al. [5] and Guu et al. [4], revolutionizes generative tasks within the LLM domain. In RAG, LLMs initially query external data sources to gather relevant information before generating text or answering questions. This retrieval step ensures that the subsequent generation phase is grounded in the retrieved evidence, which significantly improves the accuracy and relevance of the output [8,15].

By dynamically retrieving information from knowledge bases during inference, RAG effectively tackles issues such as factual inaccuracies or "hallucinations" [9]. The adoption of RAG in LLMs has rapidly gained momentum, and it has become a central approach to enhance the capabilities of chatbots and make LLMs more suitable for practical applications [8].

2.2. Evaluating Targets of RAG

Despite the proven effectiveness of RAG as outlined in Section 2.1, their successful implementation still requires careful tuning. Various factors, including the retrieval model, construction of the external knowledge base, and language model complexities, collectively influence the overall performance of the RAG system. Therefore, the evaluation of RAG

systems is critical [16]. However, before proceeding with evaluation, it is important to define evaluation targets.

The workflow of a traditional RAG typically comprises three key steps according to Gao et al. [6]: indexing, retrieval, and generation. Indexing involves extracting raw data from various formats, converting the data to plain text, segmenting the data into smaller chunks, and encoding these chunks into vector representations. Vector representations are the numerical representations of data, such as words or documents, that are used to capture semantic meaning and enable computational operations. During retrieval, user queries are transformed similarly into vector representations using the same encoding model applied during indexing. The query vectors are then compared with the vector representations of the indexed chunks to compute similarity scores. The chunks with the highest similarity scores are retrieved and serve as the expanded context for generating responses. In the generation phase, the query and selected chunks are synthesized as a prompt for an LLM to formulate a response. This process allows the model to leverage the encoded representations of the input data to generate contextually relevant and coherent responses [6].

Let us note that there are slightly different approaches for RAG conceptualizations such as in IBM Research [7] where only the two phases retrieval and content generation are distinguished. However, both approaches agree that evaluation should ensure a comprehensive assessment of both the quality of the retrieved context and the coherence of the generated content. The retrieval quality assesses the effectiveness of the sourced context, while the generation quality evaluates the relevance and coherence of the responses [7].

2.3. Evaluation Aspects of RAG

In the evaluation of RAG models, there are three primary quality metrics to assess the two main targets retrieval and generation. These quality metrics, including context relevance (retrieval metric), answer faithfulness (generation metric), and answer relevance (generation metric), serve to gauge the effectiveness of the RAG system [17,18].

Context relevance scrutinizes the precision and specificity of the retrieved context, ensuring relevance while minimizing the processing costs associated with irrelevant content. Answer faithfulness ensures the fidelity of the generated responses to the retrieved context, thereby maintaining consistency and avoiding contradictions and "hallucinations". Answer relevance requires that the generated responses directly address the posed questions, effectively tackling the main inquiry [17,18].

RAG evaluation methods can be categorized into two types: reference-required and reference-free evaluation. In the reference-free approach, the quality of outputs is determined based on intrinsic text characteristics, bypassing the need for human-annotated ground truth labels in the evaluation dataset. Conversely, the reference-required approach compares outputs to a ground truth annotated by humans [16].

Reference-free evaluation frameworks leverage LLMs to assess quality scores, such as context relevance, answer faithfulness, and answer relevance, by analyzing the alignment of the generated text with the retrieved context. However, this method's reliability can be compromised if the retrieved external information is of low quality [16]. Frameworks such as RAGAS (Retrieval-Augmented Generation Assessment) by Es et al. [17,19], ARES (Automated RAG Evaluation System) by Saad-Falcon et al. [18], or RAG Triad by TruLens [20] are classified as reference-free frameworks according to [16]. Nonetheless, both RAGAS and ARES require a form of "ground truth" for evaluation besides using synthetic training data for assessing the quality of individual RAG components. Let us note that various other approaches related to reference-free evaluation methods are available [21–25], which, for instance, may require the manual evaluation of RAG's output.

In contrast to reference-free evaluation methods, reference-required benchmarks involve comparing outputs to a reference or ground truth annotated by humans. These benchmarks provide quantitative metrics to measure the performance of the RAG model and deepen the understanding of its capabilities across various evaluation aspects. For in-

stance, the Retrieval-Augmented Generation Benchmark (RGB) [8] evaluates four essential abilities for RAG: noise robustness, negative rejection, information integration, and counterfactual robustness. However, this approach focuses on question-and-answering tasks to measure RAG performance. Other approaches, such as those discussed by Lyu et al. [16], support a wider range of RAG applications but are more challenging to implement.

2.4. RAG as an Approach to Financial Report Question and Answering

RAG finds application in scenarios centered around evaluating the question and answering of financial reports, offering access to a wealth of freely available domain-specific information, such as reports of banks. However, there is limited literature on this topic.

One approach targets financial analysts: Sarmah et al. [26] demonstrated the use of LLMs to efficiently extract information from earnings report transcripts while ensuring high accuracy. Combining the retrieval-augmented generation technique with metadata, they streamline the extraction process and mitigate hallucination. Their evaluation, based on various objective metrics for Q&A systems, demonstrated the superiority of their method. However, they focus on earnings call transcripts. Additionally, their approach incorporates manual labeling to evaluate the answer quality. Therefore, their evaluation methodology focuses on reference-required evaluation (as discussed in Section 2.3).

Two other papers explore very specific and narrow aspects of RAG in financial contexts. Yepes et al. [27] focused on effective chunking in RAG and proposed an enhanced approach to chunk financial reports by identifying structural elements in such documents. The proposed method optimizes the chunk size without tuning and improves context and accuracy.

Another important angle is financial sentiment analysis, which is crucial for valuation and investment decision making. Zhang et al. [28] introduced a retrieval-augmented LLM framework for this purpose, which features an instruction-tuned LLM module. This module uses pretrained LLMs with human-like text to guide the model's execution based on task descriptions and desired outputs, which are often labeled by humans. In addition, a retrieval-augmentation module gathers context from reliable external sources. However, this approach focuses on sentiment analysis rather than Q&A from financial reports.

2.5. Research Gap

In Sections 2.1–2.5, RAG applications were discussed that can effectively tackle the challenges of LLMs, such as inaccuracies and hallucinations. However, there is a significant lack of empirical knowledge and literature on optimizing RAG systems and their retrieval and generation components for financial report question and answering. Specifically, there is insufficient empirical evidence on the quality of RAG systems developed to handle complex financial data, such as quarterly and half-yearly bank reports.

Although automatic evaluation methods like RAGAS, as well as manual evaluation methods, exist to assess RAG systems without heavily relying on ground truths, there is limited literature on evaluating RAG systems specifically for bank reports. These reports pose unique challenges, such as accurately interpreting financial data, understanding regulatory compliance, and assessing business performance metrics. Therefore, evaluating RAG systems and their components in such scenarios to ensure minimal deviations in interpretation requires a thorough examination of the retrieval and generation components.

3. Research Design

This section presents the research methodology used to address the research questions outlined in Section 1.2 and delineates the data sources and collection methods employed.

The research tackles a real-world challenge by developing a novel and innovative artifact. Hence, Design Science Research (DSR), as proposed by Hevner and Chatterjee [29], is recommended. DSR involves the creation and evaluation of an artifact. In this context, it entails the establishment of an RAG system, along with the adaptation of its components,

such as the embedding model or LLM. These adaptations will be tailored to varying model configurations, which will be evaluated. This application aims to tackle an important problem by addressing the lack of a straightforward RAG approach tailored for the use case of private investors analyzing half-yearly or quarterly bank reports.

3.1. Process Steps of Design Science Research

This research adheres to the 5-step approach of DSR as proposed by Hevner and Chatterjee [29]. The first step, Awareness, involves identifying the research problem and requirements. Subsequently, the second and third steps, Suggestion and Development, involve designing an artifact that addresses the identified problems. Following the design phase, the fourth step, Evaluation, assesses the effectiveness and functionality of the developed artifact. Finally, the fifth step, Conclusion, summarizes by synthesizing the findings and drawing conclusions.

Most steps of the DSR approach focus on a specific research question and its corresponding objective. This will be outlined in the following subsections.

3.1.1. Awareness of Problem and Identification of Requirements

In the initial step of the 5-step DSR approach proposed by Hevner and Chatterjee [29], the focus is on recognizing a problem that can be addressed through new artifacts. This step corresponds with the problem statement outlined in Section 1.1, the literature review in Section 2, and the research gap identified in Section 2.5. In short, the problem statement and research gap illustrate the limited exploration of RAG for assisting private investors in analyzing the financial reports of banks. Despite these potential benefits, tailored RAG approaches for this specific case are lacking. This gap hinders the empirical demonstration of RAG's effectiveness in providing accurate insights for investors. The motivation behind this research is to develop and test RAG implementations tailored to the analysis of the half-yearly and quarterly reports of banks with the aim of enhancing response quality, addressing challenges in dialog models, and contributing to advancing RAG applications in this domain.

3.1.2. Suggestion

The second step of DSR involves determining the type of artifact that could solve the current problem [29]. In this case, the proposed artifact is an RAG system and its corresponding implementation. This RAG system should be designed to facilitate the analysis of the half-yearly and quarterly reports of banks. It is essential to note that the RAG system, respectively, the retrieval and generation components, should be able to encompass various scenarios, thereby allowing for comprehensive evaluation and adaptation as needed.

3.1.3. Development

After defining the problem and identifying the artifact type, as per Hevner and Chatterjee [29], the Development phase should concentrate on designing and creating an artifact that offers a solution to the defined problem. The sub-questions one and two (see Table 1), outlined in Section 1.2., are integral to this phase.

Table 1. Sub-questions for the Development phase.

No.	Sub-Questions	Activities
1.	How can the retrieval component of RAG be set up to extract relevant information from half-yearly or quarterly reports?	Data collection through literature review Artifact development
2.	What strategies can be employed to ensure that the generation component of RAG produces answers to questions posed about half-yearly or quarterly reports?	Data collection through literature review Artifact development

In this phase, the focus is on developing and implementing an RAG system. Establishing an RAG system requires the definition and selection of certain components. Figure 1 shows the simplified architecture.

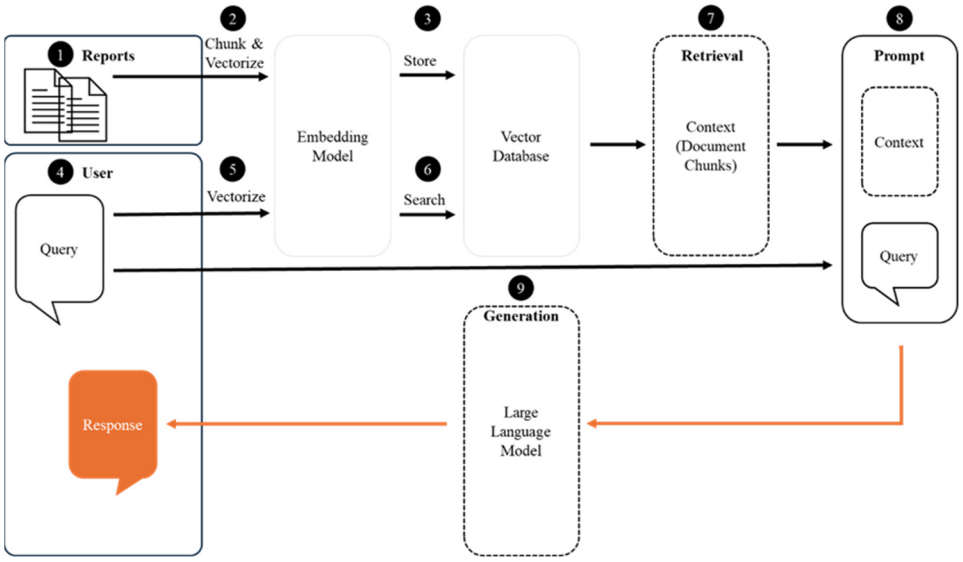

Figure 1. RAG system according to [6].

The flow is outlined based on the following 9 steps (identified by the circled number in Figure 1) based on [6]:

1. Begin by gathering essential reports for the RAG system's operation, specifically targeting half-yearly and quarterly reports from banks. Further insights into the data collection procedures are provided in Section 3.2.
2. Break down the collected data into manageable chunks to streamline information retrieval and improve efficiency by avoiding the processing of entire documents. This segmentation ensures that each data chunk remains focused on a specific topic, which increases the likelihood of retrieving relevant information for user queries. Subsequently, these segmented data are transformed into vector representations known as embeddings, which capture the semantic essence of the text.
3. The resulting embeddings are stored in a dedicated vector database to facilitate the efficient retrieval of pertinent information, transcending traditional word-to-word comparison methods.
4. The user query formulation is initiated.
5. Upon entry into the system, the user query is converted to an embedding or vector representation. To ensure consistency, the same model is used for both document and query embeddings.
6. Use the transformed query to search the vector database and perform comparisons with document embeddings.
7. The most relevant text chunks are retrieved, and a contextual framework is established to address the user query.
8. The retrieved text chunks are integrated with the original user query to provide a unified prompt for LLM.
9. The unified prompt, which comprises the retrieved text chunks and the original user query, is sent to the LLM. The LLM utilizes its advanced natural language processing capabilities and extensive knowledge to generate coherent responses tailored to address the user's queries effectively, leveraging the additional context provided by the text chunks to enhance the accuracy and depth of the responses.

Therefore, for the development of a newly built RAG system, the essential components include an embedding model, a vector database, an LLM, and a comprehensive orchestra-

tion tool to manage the entire system seamlessly. The following tools were identified as potential setups:

- Embedding Model: MiniLMEmbedder [30] and BedRockEmbeddings [31];
- Vector Database: FAISS [32] and Chroma [33];
- LLM: GPT-4o [34], Llama 3 [35], and Gemini 1.5 Pro [36];
- Orchestrator: LangChain [37] and LlamaIndex [38].

Furthermore, there exists an alternative approach involving the use of a pre-built RAG system that can be readily deployed. One such system is the Verba RAG by Weaviate. In this case, it is also possible to create different model configurations by using different embedding models (e.g., OpenAI ADA, MiniLMEmbedder) and LLMs (e.g., GPT-4o, Gemini 1.5 Pro) [39,40].

In any case, it is the goal to create three distinct technical model configurations, each using different components. For each model configuration, a selection of quarterly and half-yearly reports is gathered from various banks, as detailed in Section 3.2. Using these documents, querying and answering via the RAG system will be conducted. Ten questions per bank are formulated for this purpose and tested with each model configuration.

3.1.4. Feedback and Evaluation

In the fourth step of the DSR approach, the focus shifts to evaluating whether the artifact effectively addresses the identified problem and assessing its strengths and weaknesses [29]. In this context, the sub-questions 3 and 4 from Section 1.2. are relevant (see Table 2).

Table 2. Sub-questions for the Feedback and Evaluation phase.

No.	Sub-Questions	Activities
3.	How can the effectiveness of the RAG system for analyzing half-yearly or quarterly reports in the banking sector be reliably evaluated and validated?	The designing, conduction, and evaluation of tests
4.	How accurately do RAG-generated responses represent the information extracted from half-yearly or quarterly reports of banking institutions?	The designing, conduction, and evaluation of tests

The evaluation (sub-question 3 and 4) is crucial for determining the viability of an artifact in practical application scenarios, as it provides insights into the quality of the RAG system's performance. Three model configurations are generated for the subsequent evaluation. The evaluation of the answers provided by the RAG system encompasses context relevance, answer faithfulness, and answer relevance, as discussed in Section 2.3.

Given the considered use case, practical and accessible evaluation methods are needed that do not require extensive technical expertise. Due to the project's timeline and scope, we will set up an evaluation manually.

3.1.5. Conclusions

In the final step of the DSR process, the focus is on discussing the findings and evaluating their generalizability, as well as considering future aspects, such as open questions and plans for further development [29].

Thus, the aim is to illuminate the potential of the specific RAG system for broader implementation, identify avenues for continued development and enhancement, and address any open questions regarding its functionality, model configuration, output, or evaluation.

3.2. Data Collection and Analysis

This section discusses the necessary activities and the chosen data collection approach. As mentioned in Section 1.1, the half-yearly and quarterly reports of banks were used to evaluate the RAG system in combination with qualitative methods like document reviews by directly checking the considered quarterly and half-yearly reports. The reporting periods include the first quarters of 2022 and 2023 and the interim reports of the first half of 2022 and 2023. These periods were chosen because the data in these reports are sufficiently up-to-date. Another reason for the report's types of choice is that we would like to conduct and assess RAG performance under different conditions. Thus, the financial situation of each bank is represented by two half-yearly reports from 2022 and 2023 and two quarterly reports from the aforementioned years.

The chosen sample banks include Barclays, HSBC, Credit Suisse, Credit Agricole, and Banco Santander. The reasons for choosing these banks are diverse. First, we need to obtain direct access to their reports of the considered periods, which are published officially and are written in English. Second, the sample banks are from European countries with local economic features: the UK, EU (France, Austria, Spain, and Czech Republic), and Switzerland. It is well known that the British economic system is quite different from the EU's because of the domination of English law, which establishes other rules on property, trade, etc. At the same time, these three economic systems are not too far from each other unlike the US and China, which have varying standards of financial reporting. However, three additional banks (Bawag Bank, Erste Bank & Sparkasse, and KB Bank) were chosen. These extra documents are not relevant to the experiments; however, they test the RAG system's ability to filter out irrelevant data and focus on important information.

4. Solution Development

The following sections provide the research findings according to the main Design Science Research phases of Hevner and Chatterjee [29].

4.1. Suggestion Phase: Verba RAG

For the development of an RAG system and to address the identified problems (as stated in Section 4.1), Verba was used. Verba is an open-source RAG application that leverages Weaviate, a vector database, for information retrieval. Using the Weaviate Cloud Service (WCS), one can easily implement Verba RAG without extensive technical knowledge [39].

4.1.1. Verba RAG Architecture

Verba's modular architecture enables the customization of various components (see Table 3), such as readers, chunkers, embedders, retrievers, and generators [39], as illustrated in Figure 2.

Table 3. Verba RAG architecture.

Name	Task
Read-Chunk Manager	The component (https://weaviate.io/blog/verba-open-source-rag-app#chunker-manager-breaking-data-up, accessed on 10 June 2024) receives a list of strings representing uploaded documents. In our case, it handles the reports collected from various banks. The Read-Chunk Manager takes a list of documents and breaks each document's text into smaller segments. For this use case, the Read-Chunk Manager divides each document into chunks of 100-word tokens with a 50-token overlap. This method will remain consistent across all the tests. This approach is based on preliminary testing, which clearly indicated that larger chunk sizes (e.g., 250- and 400-word tokens) negatively affected the quality of the final output of the proposed RAG system when applied to the financial reports.

Table 3. *Cont.*

Name	Task
Embedding Manager	The Embedding Manager (https://weaviate.io/blog/verba-open-source-rag-app#embedding-manager-vectorizing-data, accessed on 10 June 2024) receives a list of documents and embeds them as vectors into Weaviate as the relevant database. It is also used to retrieve chunks and documents from Weaviate. The specific embedding model used will vary according to the model configuration described in Section 4.2.1.
Retrieve Manager	The Retrieve Manager (https://weaviate.io/blog/verba-open-source-rag-app#retrieve-manager-finding-the-context, accessed on 10 June 2024) communicates with the Embedding Manager to retrieve chunks and apply custom logic. They return a list of chunks. For this use case, the "WindowRetriever" is employed for all the tests, which retrieves relevant chunks and their surrounding context using a combination of semantic and keyword search (hybrid approach).
Generation Manager	The Generation Manager (https://weaviate.io/blog/verba-open-source-rag-app#generation-manager-writing-the-answer, accessed on 10 June 2024) uses a list of chunks and a query to generate an answer. Then, it returns a string as the answer. Like the Embedding Manager, the specific generating model used will vary according to the model configuration described in Section 4.2.1.

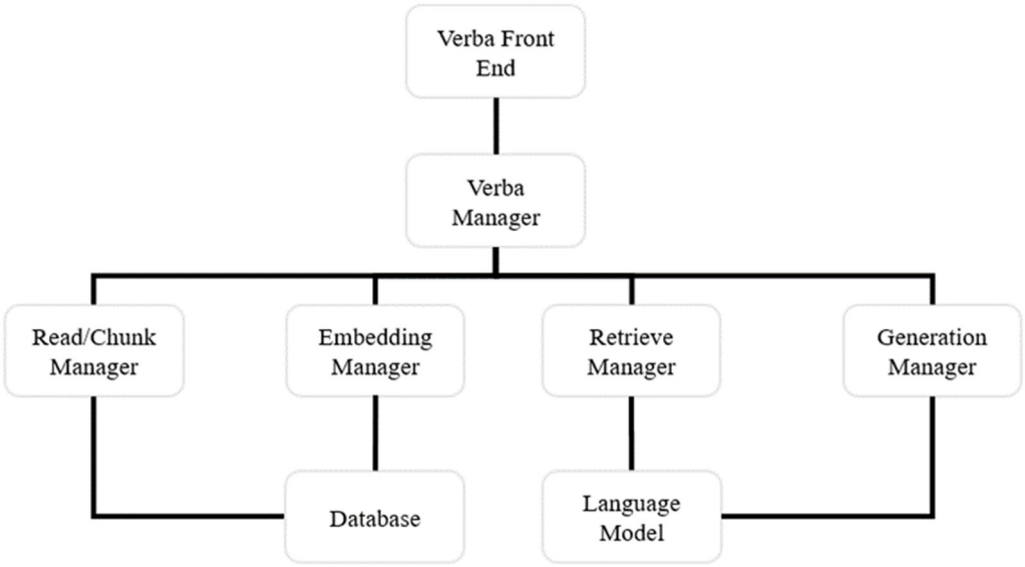

Figure 2. Illustration of the Verba RAG architecture based on [39].

4.1.2. Verba Flow

Verba by Weaviate [39] is designed to be customizable and scalable. The flow of this RAG system is generally the same as described in Section 3.1.3: Users can upload or import data files via the Verba web interface. Once uploaded, Verba automatically chunks the documents and stores the embeddings in the WCS Weaviate instance. When a query is entered into the Verba web interface, relevant document chunks are retrieved from the WCS Weaviate instance based on the selected query embeddings. These retrieved chunks are then passed to a language model (e.g., GPT-4o) to generate a contextual answer. Verba displays the generated answer along with the relevant document chunks used for context.

4.2. Development Phase: Experiments with Verba RAG

As detailed in Section 3.1.2, the artifact for this project involves the construction of an RAG pipeline with Verba by Weaviate and various other components. To address research

questions one and two of Section 1.2, the RAG system will be developed, and three model configurations will be created, each using different components. The setup and design of the experiments are outlined in the following sections.

4.2.1. Model Configurations

Three model configurations with different components were developed for the evaluation of the RAG system, as shown in Table 4.

Table 4. Model configurations.

Model	Embedding Model (Part of "Retrieval")	LLM (Part of "Generation")	Purpose
Baseline Model	OpenAI ADA	OpenAI GPT-4o	Establish a benchmark for performance based on widely used, reliable components.
Second Model	OpenAI ADA (consistent with baseline)	Google Gemini 1.5 Pro	Evaluate the impact of substituting the LLM (generator component) with a newly released LLM [1] while keeping the embedding model constant to isolate the effect.
Third Model	MiniLM Embedder	OpenAI GPT-4o (consistent with baseline)	Assess the effect on the overall performance of changing the embedding model (part of the retrieval component) to an open-source component while using the same LLM.

[1] according to the official website of OpenAI [41], GPT-4o was launched on 13 May 2024, whereas Google [42,43] released Gemini 1.5 Pro on 9 April 2024 to the public and upgraded it on May 14, 2024. The latest models are used.

4.2.2. Data

For each model configuration, two quarterly and two half-yearly reports from the five selected banks are used, as described in Section 3.2. To introduce complexity and simulate real-world challenges, reports from three additional banks are included.

4.2.3. Questions

Each bank will be subjected to ten specific questions and three model configurations. Seven of these questions are general, applicable across all the banks, and designed to evaluate the system's general capabilities. These general questions are a mix of quantitative (e.g., financial metrics like revenue or profit) and qualitative types (e.g., risks, challenges, and trends), but all are questions that a private investor could ask. They were developed iteratively, starting with simple queries and refining them through experiments to determine the optimal length and style. We ended up with relatively long questions that accommodate different wordings, such as revenue, total income, and gross earnings, as banks often use these terms synonymously. These questions can be found in Appendix A.

The remaining three questions are tailored to each individual bank to assess the system's ability to handle specific and detailed inquiries.

4.2.4. Experiments

The experiments were conducted by providing the RAG system with ten questions for each bank and model configuration. To ensure precise analysis and response, these questions were submitted individually in a single session rather than in batches. Each question, answer, and the retrieved context were compiled into an Excel. The complete

file includes data for five banks for three model configurations with ten questions per model configuration.

The RAG system was set up according to the identified requirements, developing three distinct model configurations with different components. This setup addresses research questions 1 and 2 from Section 1.2, particularly defining the retrieval and generation components. On the other hand, the defined and developed setup of the RAG system and the experiments form the basis for answering research questions three and four.

5. Evaluation and Discussion of Results

This section outlines how the evaluation of the experiments was conducted. The results of the evaluation are then discussed. This section addresses research questions 3 and 4 of Section 1.2. There is also a discussion of the academic relevance of this study at the end of this section.

5.1. Realization of Evaluation

To evaluate the three model configurations and assess the RAG systems, three quality metrics were used: context relevance (retrieval metric), answer faithfulness (generation metric), and answer relevance (generation metric) (described in Section 2.3).

Each quality metric is evaluated manually for each response, corresponding to a specific question for a bank and model configuration (as outlined in Section 3.1.3). Each metric is rated on a scale from 1 to 5 specifically developed for this purpose (see Table 5).

Table 5. Scale for evaluation.

Scale	Definition
1	The retrieved context is irrelevant to the question, the generated response is inaccurate and inconsistent, and it fails to address the question.
2	The retrieved context has limited relevance and contains several inaccuracies, the generated response exhibits notable inconsistencies, and it only partially addresses the given question.
3	The retrieved context is somewhat relevant but has some inaccuracies, the generated response is generally consistent with minor errors, and it adequately addresses the question.
4	The retrieved context is relevant and mostly accurate, the generated response is consistent with minor or no errors, and it effectively addresses the question.
5	The retrieved context is highly relevant and accurate, the generated response is entirely faithful to the context with no errors, and it thoroughly addresses the question.

A total of 750 points are possible for each model configuration, as there are five banks with 10 questions for each model, with five points possible per question in three different categories. The maximum number of points per model configuration is calculated as follows: $5 \times 10 \times 5 \times 3 = 750$ points.

5.2. Main Results and Discussion

The primary outcome is the evaluation of the three model configurations (see Table 6). Our initial goal was to determine which model configuration performed best and to address research questions 3 and 4 from Section 1.2.

Table 6. Evaluation results: three model configurations (best results are indicated in bold).

No.	Embedding Model	LLM	Points	Metric 1: Context Relevance	Metric 2: Answer Faithfulness	Metric 3: Answer Relevance	Total
1	OpenAI ADA	OpenAI GPT-4o	Total Points	175	**189**	**188**	**552**
			Average Points	**3.5**	**3.78**	**3.76**	**3.68**
2	OpenAI ADA (consistent with baseline)	Google Gemini 1.5 Pro	Total Points	**176**	185	182	543
			Average Points	**3.52**	3.7	3.64	3.62
3	MiniLM Embedder	OpenAI GPT-4o (consistent with baseline)	Total Points	137	155	146	438
			Average Points	2.74	3.1	2.92	2.92

Model one (OpenAI ADA and OpenAI GPT-4o) achieved the highest overall score (552 of 750 points). With an average score of 3.7 out of 5, this model configuration falls between the 3- and 4-point definitions on the scale presented in Section 5.1. Thus, the retrieved context was generally relevant and mostly accurate, and the generated responses were consistent with minor errors to address the posed questions. The lowest scores were assigned to "Context Relevance", often due to retrieving over 10–15 chunks, many of which were not pertinent. However, even with irrelevant chunks, the LLM still provided mostly accurate answers. This highlights GPT-4o's robustness in handling excess information.

Model two (OpenAI ADA and Gemini 1.5 Pro) received a slightly lower overall score. The "Context Relevance" score remained nearly stable due to using the same embedding model. However, the scores for "Answer Faithfulness" and "Answer Relevance" were slightly lower, reflecting the impact of using a different LLM. This observation aligns with benchmarks from the LLM Leaderboard of the Large Model Systems Organization [44], where GPT-4o was rated with 1287 points compared to Gemini 1.5 Pro with 1266 points. In addition, the Massive Multitask Language Understanding (MMLU) benchmark developed by Hendrycks et al. [45] obtained a score of 88.7% for GPT-4o and 81.9% for Gemini 1.5 Pro [46]. These benchmarks indicate a slight performance edge for GPT-4o, which was reflected in the evaluation results.

Model three (MiniLM Embedder and OpenAI GPT-4) demonstrated a significantly lower overall score of 438 points, which was more than 20% lower than model one and lower than model two. Changing the embedding model from OpenAI's ADA to the MiniLM Embedder resulted in lower scores for all the metrics, which is not unexpected as the new model is much smaller than ADA. The "Context Relevance" score was considerably lower because much of the retrieved context was not relevant to the posed questions. We observed that this model configuration often retrieved fewer chunks than the previous models, which also affected "Answer Faithfulness" and "Answer Relevance". Without sufficient (quantity) and relevant (quality) context, the LLM struggled to generate accurate answers. This underscores the importance of using a high-quality embedding model to retrieve relevant information.

In summary, to answer research questions three and four from Section 1.2, models one and two demonstrated a more accurate representation of the necessary information, while model three lagged behind. The evaluation approach and setup were effective for reliably evaluating and validating these experiments.

5.3. Additional Results and Discussion

Table 7 presents the evaluation results for each bank, independent of the model configuration. The main difference was observed between Barclays (290 of 450 points) and HSBC (338 of 450 points). Barclays, with an average score of 3.2, had a notably low score for "Context Relevance". A comparison of the half-yearly and quarterly reports of Barclays and HSBC reveals that they use completely different layouts. Barclays reports are in landscape mode and contain a large amount of data displayed in tables and various types of charts

(e.g., line charts and bar charts) with minimal written text. On the contrary, HSBC's reports are text-heavy with well-explained tables and minimal charts. This layout makes it easier for retrievers to gather the relevant context. Consequently, the HSBC reports are more suitable for the RAG system.

Table 7. Results of evaluation—total points per bank (best results are indicated in bold).

Bank	Metric 1: Context Relevance	Metric 2: Answer Faithfulness	Metric 3: Answer Relevance	Total
Banco Santander	96.0	97.0	99.0	292.0
CS	104.0	97.0	97.0	298.0
Barclays	81.0	107.0	102.0	290.0
Credit Agricole	**106.0**	107.0	102.0	315.0
HSBC	101.0	**121.0**	**116.0**	**338.0**

Table 8 presents the results for each of the seven standards and three individual questions. The highest scores were obtained on individual questions. This indicates that when specific questions and wording align with the report, the RAG system performs well across all three metrics, thereby providing accurate answers.

Table 8. Results of evaluation—total points per question (best results are indicated in bold).

Questions	Metric 1: Context Relevance	Metric 2: Answer Faithfulness	Metric 3: Answer Relevance	Total
Indiv. Question 1	57	58	59	174
Indiv. Question 2	54	59	58	171
Indiv. Question 3	**63**	**63**	**62**	**188**
Standard Question 1 (quantitative)	42	56	51	149
Standard Question 2 (quantitative)	45	49	45	139
Standard Question 3 (quantitative)	45	46	45	136
Standard Question 4 (quantitative)	42	46	44	132
Standard Question 5 (qualitative)	43	46	46	135
Standard Question 6 (qualitative)	47	54	53	154
Standard Question 7 (qualitative)	50	52	53	155

The qualitative questions also received higher average scores than the quantitative questions. It was found that when multiple similar numbers appeared in a report, the RAG sometimes had difficulties handling them accurately. For example, question 4 asked for the total assets held by the bank at the end of H1 2023. Banks typically report various types of assets, such as risk-weighted assets, customer assets, or high-quality liquid assets. If the RAG system selected the wrong number, it received a low score. However, minimal hallucination was observed; instead, the primary issue was selecting an incorrect but actually appearing number. On the other hand, for qualitative questions such as market trends, the RAG system often produced at least a partially correct answer, which resulted in higher scores.

5.4. Academic Relevance of Artifact Evaluation

The evaluation highlights the academic significance of the RAG system in assisting private investors with bank financial report analysis (see Section 2.4). By leveraging RAG, investors can access domain-specific information from half-yearly or quarterly reports. Among the evaluated model configurations, model one was the top performer, delivering accurate answers with minor errors. However, due to the variability in performance, the importance of selecting reliable system components was underscored.

This study shows the critical role of well-structured financial reports, as poorly organized documents, such as those from Barclays, received lower scores. This highlights the necessity for clear and coherent financial documentation. In addition, this study identifies the types of questions that the RAG system handles most proficiently, providing valuable academic insights.

Importantly, the evaluation has shown that the RAG concept addresses challenges in LLMs, particularly issues related to factual accuracy and the avoidance of hallucinations. These findings underscore the crucial importance of this research area for both academic inquiry and practical applications in financial report analysis.

6. Conclusions

This section consolidates our research findings, assesses the success of addressing the main research question, and presents the key insights gained from the study. Practical implications, limitations, and the potential future developments of the suggested artifact are discussed.

The study focused on the main research question: "How can the RAG be adapted to increase the context relevance, answer faithfulness, and answer relevance of LLM's conclusions about several types of financial reports?". By developing an RAG system and exploring three technical RAG model configurations employing different components (embedding models and LLMs), variations in these metrics were identified. The development and evaluation process led us to conclude that the first model configuration was particularly effective. Ultimately, the findings suggest that we can empower private investors to make informed decisions when provided with accurate answers derived by such an RAG system from relevant bank reports.

Although the main research question was addressed successfully, the study revealed several areas for improvement. A significant limitation was the RAG system's difficulty in processing complex PDF layouts. Future research should aim to optimize the system by integrating more robust components, such as computer vision models, to handle visual information more effectively. Moreover, adding a domain-specific repository for financial terminology would enhance context understanding and mitigate issues arising from different terminology used by different banks for similar concepts (e.g., assets).

Future research holds various promising avenues to provide investors with even more reliable insights. Expanding the system to include a broader spectrum of banks is a critical aspect. In addition, exploring more advanced evaluation methods could enhance the robustness and accuracy of the assessment process, thereby improving RAG's overall efficacy.

By addressing these limitations and exploring the outlined research directions, the RAG systems' capability to analyze financial reports and deliver valuable insights to private investors could be enhanced.

Author Contributions: Methodology, software, investigation, and writing—original draft preparation, I.I., R.P., and L.V.; writing—review and editing and supervision, T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author due to legal reasons.

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

Appendix A

Table A1. Questions for Experiments.

No.	Question/Prompt	Period	Type
1.	What was the total revenue for [Bank Name] in quarter Q1 2022? Retrieve the total revenue for [Bank Name] from the report of the quarter Q1 2022. Also, consider alternative terms such as 'total income' or 'gross earnings' if they are used in the document.	Quarter	Quantitative
2.	What is the percentage change in total revenue for [Bank Name] in Q1 2023 compared to the same quarter of the previous year? Find or calculate the percentage change in total revenue for [Bank Name] of Q1 2023 compared to the to the same quarter of the previous year. Include terms like 'percentage growth' or 'percentage decline' if mentioned.	Quarter	Quantitative
3.	What is the net profit reported by [Bank Name] for the half-year period H1 2022? Identify the net profit for [Bank Name] for the half-year period H1 2022, including any references to 'net income' or 'net earnings'. Ensure the figure reflects income after all expenses.	Half-yearly	Quantitative
4.	What are the total assets held by [Bank Name] at the end of the half-yearly period H1 2023? Extract the total assets held by [Bank Name] at the end of the half-yearly period H1 2023. Include all categories such as cash, investments, loans, and other financial instruments.	Half-yearly	Quantitative
5.	What strategic initiatives or projects did [Bank Name] undertake during the reporting period Q1 2022? Summarize the strategic initiatives or projects undertaken by [Bank Name] during the reporting period, including new product launches, market expansions, and digital transformation efforts. There might be also similar strategic initiatives or projects.	Quarter	Qualitative
6.	What trends or market conditions affecting the banking sector were highlighted by [Bank Name] in their report of the half-yearly period H1 2022? Highlight the trends or market conditions affecting the banking sector as discussed by [Bank Name], such as interest rate changes, regulatory impacts, and technological advancements. There might be also similar trends and market conditions.	Half-yearly	Qualitative
7.	What key risks and challenges for the upcoming period did [Bank Name] identify in their report of the half-yearly period H1 2023?? Detail the key risks and challenges for the upcoming period identified by [Bank Name] include, including any mention of credit risk, market volatility, and operational challenges. There might be also similar risks and key challenges.	Half-yearly	Qualitative

References

1. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
3. Burtsev, M.; Reeves, M.; Job, A. The Working Limitations of Large Language Models. *MIT Sloan Manag. Rev.* **2024**, *65*, 8–10.
4. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M. REALM: Retrieval-Augmented Language Model Pre-Training. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 13–18 July 2020; pp. 3929–3938.
5. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
6. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Guo, Q.; Wang, M.; et al. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2024**, arXiv:2312.10997.
7. IBM Research. What Is Retrieval-Augmented Generation? Available online: https://research.ibm.com/blog/retrieval-augmented-generation-RAG (accessed on 9 February 2021).
8. 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.
9. Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; Weston, J. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv* **2021**, arXiv:2104.07567.
10. Anil, R.; Borgeaud, S.; Wu, Y.; 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.
11. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288.
12. He, H.; Zhang, H.; Roth, D. Rethinking with Retrieval: Faithful Large Language Model Inference. *arXiv* **2022**, arXiv:2301.00303.
13. Shen, X.; Chen, Z.; Backes, M.; Zhang, Y. In ChatGPT We Trust? Measuring and Characterizing the Reliability of ChatGPT. *arXiv* **2023**, arXiv:2304.08979.
14. Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv* **2023**, arXiv:2309.01219.
15. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G.B.; Lespiau, J.-B.; Damoc, B.; Clark, A. Improving language models by retrieving from trillions of tokens. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 2206–2240.
16. Lyu, Y.; Li, Z.; Niu, S.; Xiong, F.; Tang, B.; Wang, W.; Wu, H.; Liu, H.; Xu, T.; Chen, E.; et al. CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models. *arXiv* **2024**, arXiv:2401.17043.
17. Es, S.; James, J.; Espinosa-Anke, L.; Schockaert, S. RAGAS: Automated Evaluation of Retrieval Augmented Generation. *arXiv* **2023**, arXiv:2309.15217.
18. Saad-Falcon, J.; Khattab, O.; Potts, C.; Zaharia, M. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. *arXiv* **2023**, arXiv:2311.09476.
19. Ragas. Core Concepts. 2023. Available online: https://docs.ragas.io/en/latest/concepts/index.html (accessed on 10 July 2024).
20. TruLens. RAG Triad. 2024. Available online: https://truera.com/ai-quality-education/generative-ai-rags/what-is-the-rag-triad/ (accessed on 10 July 2024).
21. Besbes, A. A 3-Step Approach to Evaluate a Retrieval Augmented Generation (RAG). Towards Data Science. Available online: https://towardsdatascience.com/a-3-step-approach-to-evaluate-a-retrieval-augmented-generation-rag-5acf2aba86de (accessed on 23 November 2023).
22. Besbes, A. Quickly Evaluate Your RAG without Manually Labeling Test Data. Towards Data Science. Available online: https://towardsdatascience.com/quickly-evaluate-your-rag-without-manually-labeling-test-data-43ade0ae187a (accessed on 21 December 2023).
23. Frenchi, C. Evaluating RAG: Using LLMs to Automate Benchmarking of Retrieval Augmented Generation Systems. Willow Tree Apps. Available online: https://www.willowtreeapps.com/craft/evaluating-rag-using-llms-to-automate-benchmarking-of-retrieval-augmented-generation-systems (accessed on 1 December 2023).
24. Leal, M.; Frenchi, C. Evaluating Truthfulness: Benchmarking LLM Accuracy. Willow Tree Apps. Available online: https://www.willowtreeapps.com/craft/evaluating-truthfulness-a-deeper-dive-into-benchmarking-llm-accuracy (accessed on 21 September 2023).
25. Nguyen, R. LlamaIndex: How to Evaluate Your RAG (Retrieval Augmented Generation) Applications. Better Programming. Available online: https://betterprogramming.pub/llamaindex-how-to-evaluate-your-rag-retrieval-augmented-generation-applications-2c83490f489 (accessed on 1 October 2023).
26. Sarmah, B.; Zhu, T.; Mehta, D.; Pasquali, S. Towards reducing hallucination in extracting information from financial reports using Large Language Models. *arXiv* **2023**, arXiv:2310.10760.
27. Yepes, A.J.; You, Y.; Milczek, J.; Laverde, S.; Li, R. Financial Report Chunking for Effective Retrieval Augmented Generation. *arXiv* **2024**, arXiv:2402.05131.

28. Zhang, B.; Yang, H.; Zhou, T.; Ali Babar, M.; Liu, X.-Y. Enhancing Financial Sentiment Analysis via Retrieval Augmented Large Language Models. In Proceedings of the Fourth ACM International Conference on AI in Finance, Brooklyn, NY, USA, 27–29 November 2023; pp. 349–356.
29. Hevner, A.; Chatterjee, S. (Eds.) Design Science Research in Information Systems. In *Design Research in Information Systems: Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 9–22. [CrossRef]
30. HuggingFace. MiniLMEmbedder. Available online: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed on 18 January 2024).
31. LangChain. Bedrock Embeddings. 2023. Available online: https://python.langchain.com/v0.1/docs/integrations/text_embedding/bedrock/ (accessed on 10 July 2024).
32. Meta. Faiss: A Library for Efficient Similarity Search. Available online: https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/ (accessed on 29 March 2017).
33. Chroma. Chroma Docs. 2024. Available online: https://docs.trychroma.com/getting-started (accessed on 10 July 2024).
34. OpenAI. OpenAI Platform. 2024. Available online: https://platform.openai.com/docs/overview (accessed on 10 July 2024).
35. Meta. Meta Llama 3. 2024. Available online: https://llama.meta.com/llama3/ (accessed on 10 July 2024).
36. Google DeepMind. Gemini Pro 1.5. Available online: https://deepmind.google/technologies/gemini/pro/ (accessed on 20 May 2024).
37. LangChain. Introduction to LangChain. 2023. Available online: https://python.langchain.com/v0.1/docs/get_started/introduction/ (accessed on 10 July 2024).
38. LlamaIndex. LlamaIndex Docs. 2024. Available online: https://docs.llamaindex.ai/en/stable/ (accessed on 10 July 2024).
39. Weaviate. Verba Docs. GitHub. 2024. Available online: https://github.com/weaviate/Verba (accessed on 10 July 2024).
40. Weaviate. Verba—Demo Tool. 2024. Available online: https://verba.weaviate.io/ (accessed on 10 July 2024).
41. OpenAI. Hello GPT-4o. Available online: https://openai.com/index/hello-gpt-4o/ (accessed on 13 May 2024).
42. Google. Gemini 1.5 Pro Now Available in 180+ Countries. Available online: https://developers.googleblog.com/en/gemini-15-pro-now-available-in-180-countries-with-native-audio-understanding-system-instructions-json-mode-and-more/ (accessed on 9 April 2024).
43. Google. Gemini 1.5 Pro Updates. Available online: https://blog.google/technology/developers/gemini-gemma-developer-updates-may-2024/ (accessed on 14 May 2024).
44. Large Model Systems Organization. LLM Leaderboard. Available online: https://chat.lmsys.org/?leaderboard (accessed on 6 June 2024).
45. Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; Steinhardt, J. Measuring Massive Multitask Language Understanding. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021; pp. 11260–11285.
46. Papers with Code. Multi-task Language Understanding on MMLU. 2024. Available online: https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu (accessed on 10 July 2024).