Assessing RAG System Capabilities on Financial Documents

Oscar Lithgow-Serrano[1], David Kletz[1], Vani Kanjirangat[1], David Adametz[2], Marzio Lunghi[2], Claudio Bonesana[1], Matilde Tristany Farinha[2], Yuntao Li[2], Detlef Repplinger [2], Marco Pierbattista[2], Stefania Stan[2], Oleg Szehr[1]
[1]SUPSI, IDSIA, Switzerland
[2]UBS Switzerland AG and its affiliates
Correspondence: {david.kletz, oleg.szehr}@supsi.ch, {david.adametz, stefania.stan}@ubs.com

Abstract

Financial institutions are increasingly using Retrieval-Augmented Generation (RAG) systems for document processing. However, there is still limited systematic evaluation focused on industry-specific content. In this study, we evaluated four state-of-the-art RAG architectures for processing of financial documents using FinDoc-RAG, a benchmark we developed for this purpose. This benchmark consists of over 600 question-answer pairs derived from 46 documents from a banking institution. Source materials include product descriptions, investment guides, legal policies, and marketing brochures, all of which contain dense numerical content and complex layouts. Our evaluation shows significant performance gaps: while leading systems achieve an accuracy of 0.91 on factual extraction, performance drops to 0.44 on cross-document synthesis tasks. Our experiments demonstrate varying strengths of the explored RAG approaches across different question complexities in the financial services sector and position *FinDoc-RAG* as a benchmark for measuring progress in this area.

1 Introduction

Financial institutions process thousands of documents that require human interpretation for client advisory tasks, regulatory compliance, and product inquiries. Large Language Models (LLMs) offer automation potential but face deployment challenges such as regulatory constraints that prevent external data transfer and operational complexity challenges, such as documents that combine textual content with numerical data, complex layouts, and requirements. Current LLMs show limitations in quantitative reasoning and cross-document synthesis essential for financial applications. Retrieval-Augmented Generation (RAG) addresses privacy constraints while leveraging LLM capabilities, but systematic evaluation on financial documents remains limited.

Financial documents present unique challenges: they require factual extraction, quantitative reasoning with numerical data, and information synthesis across multiple documents for comprehensive responses. Evaluating RAG performance on these distinct task types requires specialized benchmarks that reflect real-world financial complexity.

Existing Question-Answer (QA) datasets focus on Wikipedia articles (Yang et al., 2018, Kwiatkowski et al., 2019), academic papers (Pramanick et al., 2024), or single-domain sources (Pipitone and Alami, 2024, Ngo et al., 2024), and therefore fail to capture the heterogeneous nature of financial document collections. While financial QA benchmarks often focus on narrow regulatory domains, they neglect the broader spectrum of client-facing content. Moreover, no benchmark systematically evaluates the intersection of financial materials and the diverse types of tasks critical for the deployment of RAG.

We introduce *FinDoc-RAG*, a QA benchmark comprising 600+ QA pairs from 46 documents in English from UBS AG and other UBS entities. Documents include product descriptions, investment guides, legal policies, and marketing materials with dense numerical content and regulatory references. The questions span nine complexity levels (L0-L8) that target factual extraction, quantitative reasoning, and multi-document synthesis. Evaluation of five RAG architectures –vector-based indexing, graph-enhanced RAG, hierarchical summary-style retrieval (e.g., Raptor 4.2), and Knowledge Graph (KG)– reveals systematic performance gaps: leading systems achieve 0.91 accuracy in factual extraction but only 0.44 on multi-document synthesis tasks.

Contributions:

1. We present FinDoc-RAG, a RAG-focused question-answer benchmark over heterogeneous financial documents. It comprises

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

nine task levels, each associated with a predefined difficulty ranging from single-document extraction to multi-document synthesis. The data are published at https://gitlab-core.supsi.ch/dti-idsia/ai-finance-papers/findoc-rag.

2. We evaluate four representative RAG architectures, demonstrating their individual strengths and weaknesses. The dataset is released to foster research and compare RAG systems in the financial domain.

Our analysis identifies specific failure modes in current RAG approaches, with quantitative reasoning showing high performance variability and multi-document synthesis proving most challenging across all systems. The benchmark enables systematic evaluation of financial RAG systems and provides deployment readiness assessment for different task types.

2 Related Work

By retrieving relevant passages from external document collections prior to generation, RAG systems improve factual grounding, enhance domain-specific accuracy, and support local deployment with preserved data privacy. This architecture is especially promising in specialized domains like finance, where even state-of-the-art LLMs struggle, when used in isolation, with quantitative reasoning, factual consistency, and multi-document synthesis (Rasool et al., 2024).

The creation of information-seeking QA datasets has been pivotal in driving progress in RAG-based approaches.

2.1 Domain-Specific and Heterogeneous QA Benchmarks

General-purpose benchmarks such as Natural Questions (Kwiatkowski et al., 2019) evaluate QA over real-world queries and Wikipedia passages. Domain-specific datasets target deeper comprehension in specialized settings. For instance, Qasper (Dasigi et al., 2021) covers academic articles in NLP, SPIQA (Pramanick et al., 2024) addresses reasoning over complex figures and tables, and datasets such as MedRGB (Ngo et al., 2024) and LegalBench-RAG (Pipitone and Alami, 2024) focus on medical and legal domains, respectively.

Recent efforts have extended QA evaluation to longer and more complex contexts. HOTPOTQA (Yang et al., 2018) and MultiHop-RAG (Tang and Yang, 2024) test multihop reasoning, while QuALITY (Pang et al., 2022) and MMLongBench-Doc (Ma et al., 2024) challenge models with long documents and structured layouts. Multimodal benchmarks such as VisDoMBench (Suri et al., 2025), MRAG-Bench (Hu et al., 2025), and MuRAG (Chen et al., 2022) further evaluate the integration of textual and visual information.

However, most existing datasets assume homogeneous, well-structured sources and do not reflect the heterogeneity of real-world document collections. In industry settings, especially in finance, documents range from reports and contracts to internal memos, with various formats, styles, and terminology. Financial QA datasets such as FinTextQA (Chen et al., 2024a), FinDER (Choi et al., 2025), and GBS-QA (Sohn et al., 2021) typically focus on narrow domains or single-source documents, limiting their generalizability.

To bridge this gap, we introduce *FinDoc-RAG*, a benchmark designed for QA over heterogeneous financial documents. It captures cross-document reasoning, contextual variability, and structural diversity characteristic of real-world financial information ecosystems.

2.2 Evaluation Strategies for QA Benchmarks

Evaluating QA benchmarks—particularly those involving long, heterogeneous, or domain-specific documents—remains a major challenge. Broadly, evaluation strategies fall into two categories: model-centric, which assess the performance of different LLMs, and method-centric, which compare paradigms such as extractive, abstractive, or RAG.

While early QA benchmarks focused primarily on comparing model performance, recent efforts have shifted toward approach-specific evaluations, particularly in the context of RAG. Despite the strong general QA capabilities of state-of-the-art LLMs such as GPT-4, studies show persistent limitations in multistep reasoning and numerical understanding (Rasool et al., 2024). In contrast, RAG-based methods demonstrate improved factual grounding and reduced hallucination in domain-specific tasks (Chen et al., 2024b).

However, recent findings indicate that no single approach consistently outperforms others across all task types. The LaRA benchmark (Li et al., 2025), for example, demonstrates that both RAG and long-context methods succeed in different sce-

narios, highlighting the need for nuanced, task-aware evaluation frameworks that account for document complexity, question type, and reasoning depth.

These insights emphasize the importance of benchmarks that capture real-world document heterogeneity while enabling multifaceted evaluations aligned with the strengths and trade-offs of both models and methodologies.

3 *FinDoc-RAG* Benchmark

FinDoc-RAG comprises 600+ QA pairs extracted from 46 documents in English from UBS AG and other UBS entities.

Documents span four categories: product descriptions, investment guides, legal policies, and marketing materials. The collection includes two distinct subsets: V1 contains concise factsheets with dense numerical content averaging 2,400 words, while V2 features comprehensive reports with complex layouts averaging 12,000 words and rich structural elements including tables, footnotes, and cross-references.

3.1 Question Generation Methodology

Questions are structured across nine complexity levels (L0-L8) targeting three task types: factual extraction, information integration, and multi-document synthesis. Each level introduces specific constraints on document scope, quote requirements, and reasoning complexity based on our initial design expectations (see Table 1). However, empirical results reveal that expected difficulty progression does not always align with actual model performance.

Question generation is carried out using two methodological approaches: raw document content and clustered document summaries. Raw document approaches (L0-L4) generate questions directly from the original text, enabling extraction and role-based query formulations. For the other levels, the cluster-based approach first creates document summaries, embed them semantically, and clusters related content using Gaussian Mixture Models. The optimal number of clusters is selected using the Bayesian Information Criterion. The resulting clusters serve as the basis for generating questions that integrate information across related contexts, yielding multi-aspect queries that test narrative understanding rather than isolated fact retrieval.

3.1.1 Factual Extraction Tasks

Levels L0, L1, L4, and L5 involve single-document retrieval tasks that require direct text extraction, without the need for computational reasoning.

L0 Generic prompts applied uniformly to V1 documents (concise factsheets) generate self-contained QA pairs. Questions avoid generic formulations and reference relevant topics when needed. Each generated pair undergoes manual review to ensure groundedness and eliminate hallucinations.

L1 Targets V2 documents (comprehensive reports) using stratified processes that identify specific textual quotes including numerical values, dates, and key definitions. The questions remain strictly answerable from isolated textual details.

L4 Role-based prompting simulates heterogeneous user perspectives through three financial personas: young student exploring digital financial tools, elderly widow prioritizing stability with limited resources, and high-earning digital nomad navigating minimal traditional banking reliance. Each persona generates questions reflecting typical concerns and levels of knowledge.

L5 This method relies on thematic document selection, using the cluster-based approach to identify coherent topic groups. By drawing questions from individual summaries within these clusters, the LLM is guided to generate queries tied to the cluster's topic. This approach provides controlled complexity by focusing on specific topic areas while maintaining single-document question scope.

3.1.2 Information Integration Tasks

Levels L2 and L6 require combining information from multiple document sections or sentence fragments, which may require mathematical operations, numerical comparisons, or logical synthesis of related concepts.

L2 Aggregates information from different sections within a single document, connecting thematically related passages to generate abstract queries requiring numerical synthesis across document parts.

L6 Synthesizes 2–3 direct quotes from distinct sentence fragments, sometimes involving basic mathematical operations, percentage calculations, or numerical comparisons that are strictly entailed by the quoted content.

3.1.3 Cross-Document Synthesis Tasks

The integration of cross-document information that requires reasoning across multiple sources is the core of levels L3, L7, and L8.

L3 Extracts key concepts and topics from multiple documents, combining information into QA pairs that require understanding relationships between different source documents.

L7 Complex multi-document synthesis requiring >=3 quotes from >=3 different documents, testing the ability to integrate information across diverse source materials.

L8 Cluster-based maximum complexity synthesis where N quotes from N documents equal the cluster size. Uses the same clustering methodology as L5 but operates across document boundaries, requiring synthesis within clusters containing multiple document summaries. Similar to L5, the aspects from different summaries enable the generation of more complex and diverse QA pairs, while in L8, the complexity further increases, attributed to its multi-document summarization strategy.

Detailed questions generation settings and parameters can be found in the Appendix B.

3.2 Benchmark Statistics

Document Distribution The dataset consists of 46 documents spanning four radically different templates: product factsheets, legal documents, investment guides, and marketing materials. This heterogeneity ensures a diverse and representative dataset, yielding over 600 high-quality QA pairs. *Legal* documents have the highest average word count (4,000 words), with significant outliers exceeding 10,000 words, while *Research and Reports* are the shortest, averaging ∼500 words. Further details are provided in Appendix C.2, Figure 5.

Lexical Analysis Using Type-Token Ratio (TTR) as a measure of lexical diversity we found that *Research and Reports* documents have more diverse vocabulary (~ 0.09 TTR) compared to other types of documents (Figure 7). The density of financial terminology peaks in *Product Information* documents (6-7% of tokens) and *Forms and Guides* (4-6%), remaining lowest in *Marketing Materials* (1-2%). The complete analysis is available in Appendix C.2, Figure 11.

Layout Complexity *Legal* documents show the highest structural complexity (scores 10-25+, including outliers at 120), featuring tables, sections, and complex nesting levels. *Product Information* exhibit minimal complexity (scores ~ 5). Details can be found in Appendix C.2, Figure 9.

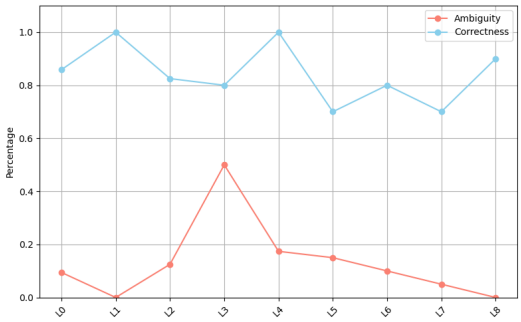

Figure 1: Ambiguity and correctness across levels with human validation.

3.3 Quality Validation

Human validation at all levels reveals correctness rates above 80% for most levels, with L3 and L7 achieving 95% correctness. Ambiguity ranges from 0% (L1, L8) to 50% (L3), with most levels maintaining low-to-moderate ambiguity (see Figure 1 and Appendix C.2).

The validation process assessed maximum 50 randomly sampled questions per level using two annotators with financial domain expertise [1]. Additionally, QA pairs were deduplicated and reviewed by a domain expert to remove those that appeared incorrect based on domain expertise, without formal fact-checking. For all levels, the financial experts performed internal validation of the QA pairs to filter out the suprious pairs.

4 Evaluating Financial RAG Systems

Our objective is to identify which RAG approaches achieve sufficient accuracy across various financial document processing tasks and to uncover fundamental limitations that point to necessary architectural improvements. We evaluated four representative RAG systems, each corresponding to a distinct architectural approach.

4.1 Task Setup

We adopt a standard RAG setup, where each system receives a query and retrieves the text spans from a fixed corpus that the system considers relevant to generate an answer to the query. In our

[1]The publicly released dataset may contain fewer QA pairs per level than manually validated. Additional pairs were validated but cannot be shared due to confidentiality constraints.

Level	# Quotes	# Docs	Expected Difficulty	Special Features
L0	1	1	Easy	V1 document set
L1	1	1	Easy	V2 document set
L2	Multiple	1	Medium	V2 docs, multiple supporting quotes
L3	Multiple	Multiple	Hard	Cross-document information synthesis
L4	1	1	Medium	Non-expert phrasing style
L5	≥ 1	1	Easy	Based on cluster summaries
L6	2-3	≥ 1	Medium	Quotes from different sentence fragments
L7	≥ 3	≥ 3	Hard	Quotes from different summaries
L8	$\geq N$	N	Very Hard	N equals cluster size

Table 1: Question Complexity Levels Description.

case, the corpus consists of the 46 financial documents included in the *FinDoc-RAG* benchmark. The set of questions comprises questions spanning the nine complexity levels (L0–L8) defined in the benchmark.

Each architecture under evaluation (detailed in 4.2) processes the full set of questions in a zero-shot setting. The retriever has access to all 46 documents and selects the subset of documents that it considers relevant to answer the input query. The retrieved documents, along with the query, are then passed to the generator component, which produces the answer. All systems utilize default configurations without fine-tuning, hyperparameter optimization, or preprocessing customization (e.g., chunking, enrichment, propositionalization) to provide baseline performance assessment representative of out-of-the-box deployment scenarios.

The generated answers are evaluated against the expected answers provided in *FinDoc-RAG*. Multiple evaluation metrics are used to assess different aspects of system performance (see Section 4.3). Our assessment evaluates end-to-end RAG system performance, rather than isolated retrieval or generator components. The systems handle document selection from the collection and passage identification within selected documents as integrated processes, with the final answer generation completing the pipeline. This holistic evaluation reflects real-world deployment scenarios in which RAG systems must complete a document-to-answer workflow without human intervention in retrieval decisions.

4.2 Selected RAG Architectures

Vector-based (Vector-RAG) This baseline method is based on an index of dense vector representations. It starts by encoding documents into high-dimensional embeddings using a neural encoder model. These document vectors are stored in a vector index that supports efficient similarity search. When a query is submitted, it is encoded using the same model, and the system retrieves documents with embeddings most similar to the query vector, in this case using cosine as the similarity metric. During retrieval, the system ranks documents based on their vector similarity scores to determine relevance. By leveraging the semantic representativeness of dynamic embeddings produced by neural encoders, this approach can identify topically relevant information even when exact keyword matches are absent. This method is computationally efficient for large-scale retrieval.

Recursive Abstractive Processing for Tree-Organized Retrieval (RAPTOR) Sarthi et al. (2024) is a semi-structured method based on hierarchical summarization organized on a tree structure. It starts by breaking down large texts into smaller chunks, which are then embedded using a BERT-based encoder. These chunks are grouped into clusters using a Gaussian Mixture Model, and a language model summarizes each cluster. This process is repeated to build a tree with multiple levels of summaries. During retrieval, RAPTOR can either traverse the tree layer-by-layer or evaluate nodes across all layers to find the most relevant information. By capturing high- and low-level details about a text, this approach helps with handling a wide range of questions and improves the integration and relevance of the retrieved information.

Graph RAG (GraphRAG) Edge et al. (2025) is a multistep method to answer questions from large text collections. First, it creates a graph-based index by building an entity knowledge graph and generating summaries for groups of related entities.

When a question is asked, these summaries help create partial answers, which are then combined into a final response. Specifically, it uses algorithms such as the Leiden algorithm (Traag et al., 2019) to detect communities within the graph by identifying groups of closely related elements, including nodes, edges, and covariates. By partitioning the graph into these communities, the method can perform parallel summarization and employs a hierarchical structure to provide different levels of detail. It also uses a map-reduce technique to combine the partial answers from the parallel summaries. This approach is designed to handle broad questions and large amounts of text.

Graph Foundation Model for Retrieval Augmented Generation (GFM-RAG) Luo et al. (2025) is a query-aware Graph Neural Network (GNN) pretrained on more than 60 knowledge graphs with over 14M triples and 700k documents. This foundation model is intended to generalize to similar Knowledge Graphs (KGs) independently of the domain. Following Luo et al. (2025), we created our KG by prompting an LLM over the source documents to generate the KG triplets. The pretrained query-aware GFM retriever model is then used to extract relevant entities from the KG with respect to the given query. Based on the relevance scores of the entities, the top entities are selected and then used by a document ranker that retrieves the ranked set of relevant documents. The final top K documents are given as the context for the LLM along with the query to generate the respective answer.

Detailed experimental settings and parameters can be found in Appendix B.

4.3 Evaluation Metrics and Scoring

To robustly assess the quality of the response provided by an architecture, we try to capture distinct dimensions of correctness beyond the exact text overlap by using different metrics.

The most direct evaluation of QA performance is the degree of overlap between the generated answer and the expected reference answer. This surface-level correspondence is captured by traditional text-matching metrics. Following the SQuAD evaluation protocol (Rajpurkar et al., 2016), we report two standard metrics: Exact Match (EM), which assigns a binary score based on exact string equivalence after normalization, and F1 Score, which computes word-level overlap to capture the trade-off between precision and recall.

Alternatively, for a fairer and more robust evaluation, it is key to recognize answers that are semantically equivalent to the reference, even when they differ in phrasing (Bulian et al.; Li et al.; Thakur et al.; Reiter). To capture this aspect, we introduce a second evaluation based on semantic similarity. We use BERTScore (Zhang* et al., 2020), which measures the alignment between predicted and reference answers by computing token-level similarities using contextualized BERT embeddings. The metric performs greedy matching between the tokens in both texts, aligning each token with its most similar counterpart based on cosine similarity. From these alignments, it computes precision, recall, and F1 scores that reflect the degree of semantic correspondence between the two answers, even when their surface forms differ.

We also include an LLM-based metric, which is more sensitive to semantic meaning that depends on subtle contextual cues. This approach, inspired by Zheng et al. (2023); Friel et al. (2024), uses an LLM to evaluate the semantic equivalence between predicted and reference answers within the context of the question. The LLM is prompted to determine whether the candidate's answer accurately preserves the meaning of the ground truth. To improve reliability and better reflect the model's confidence, we frame the evaluation as a factual correctness task with a binary classification—labeling answers as either *CORRECT* or *INCORRECT*. The LLM provides a brief explanation for its judgment while applying semantic flexibility for minor phrasing differences that preserve core meaning, tolerating reasonable omissions that do not introduce ambiguity, and ignoring stylistic differences unless they impact clarity. The answers are marked *INCORRECT* if they contain factual errors, false claims, significant omissions, or distortions of the core meaning. Each question was evaluated by the LLM judge across 3 independent runs. We computed accuracy as the mean proportion of *CORRECT* responses per question (ranging from 0 to 1), then averaged across all questions per system.

To capture a more fine-grained measure of answer quality, we finally adopt the *LLMLogScore* (L3Score) metric introduced by Pramanick et al. (2024). This approach leverages the log-likelihood probabilities generated by an LLM when prompted to evaluate semantic similarity between a candidate answer and the ground truth. By comparing the model's predicted probabilities of "yes" and "no"

responses, L3Score computes a continuous similarity score normalized between 0 and 1, enabling a more sensitive and graded evaluation without relying on arbitrary predefined scales.

Detailed evaluation settings and parameters can be found in Appendix B.

5 Results & Analysis

The observed performance patterns should be interpreted as indicators of the benchmark's inherent challenges across question complexity levels rather than definitive assessments of the approaches' capabilities.

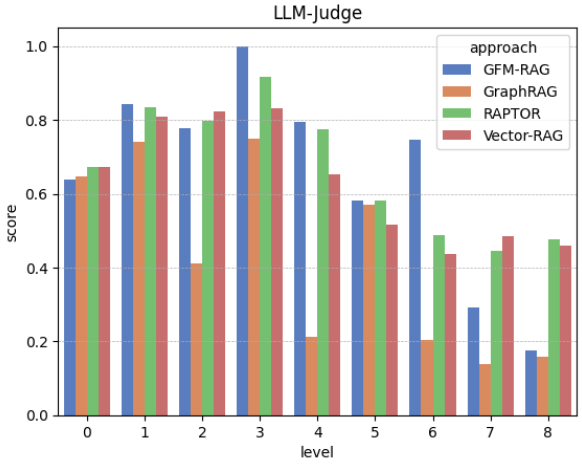

Figure 2: RAG approaches across question levels measured by LLM-Judge.

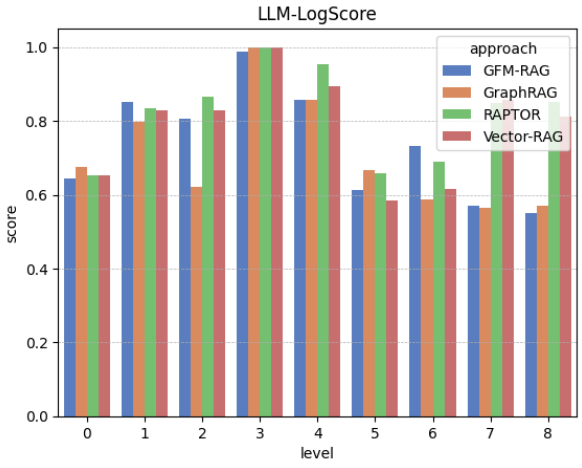

Figure 3: RAG approaches across question levels measured by LLMLogScore.

5.1 Benchmark complexity analysis

Task type analysis reveals distinct performance patterns across RAG architectures. Factual extraction tasks (L0, L1, L4, L5) have relatively stable performance with accuracy ranging from 0.51 to 0.84 0.51-0.84 across systems –except for a significant drop by GraphRAG on L4 (see details in subsection 5.2)– demonstrating suitability for production deployment in basic extraction tasks. Information integration tasks (L2, L6) show substantial performance variability (0.20-0.82 range) with GFM-RAG excelling at L6 (0.75) and GraphRAG struggling in both levels, suggesting system-dependent capabilities for intra-document synthesis. Cross-Document Synthesis tasks (L3, L7, L8) demonstrate extreme performance ranges (0.14-1.00), highlighting fundamental architectural differences in multi-document reasoning capabilities. In the individual-level difficulty analysis, L1 is the easiest level with an average score (across systems) of 0.81 and, counterintuitively, L3 achieved the highest individual system performance (GFM-RAG: 1.00). Levels L6-L8 show substantial performance degradation across most systems, L8 is the one with the lowest average score (0.32) but, L4 presents the highest variance across systems.

5.2 System performance analysis

Vector-RAG exhibits competitive baseline performance, particularly excelling at L2 and L3 (0.82-0.83), but its performance drops on synthesis tasks. RAPTOR, the other top performer, shows consistent mid-to-high performance across all complexity levels, achieving the second-highest scores on L3 (0.92) and maintaining stability across diverse question types. With the highest average score across levels (0.66) with low standard deviation ($\sigma = 0.17$), RAPTOR is a robust general purpose approach. GraphRAG demonstrates mixed performance patterns: strong capability on basic tasks (L1: 0.742) but severe degradation on complex synthesis (L7-L8: 0.14-0.16), which suggests architectural limitations in multi-document reasoning. GFM-RAG is one of the strongest performers, achieving perfect accuracy on L3 (1.00) and leading performance on L6 (0.74), demonstrating good capabilities for multi-document reasoning tasks.

System Selection Guidance For basic document extraction, all systems except GraphRAG achieve adequate performance ($\geq$ 0.65), indicating reliable potential for factual retrieval tasks. Complex synthesis scenarios require targeted system selection: GFM-RAG for multi-document reasoning (L3: 1.00, L6: 0.74), RAPTOR for consistent cross-level performance, and Vector-RAG for mid-complexity applications (L2-L3: 0.82-0.83).

Approach	L0	L1	L2	L3	L4	L5	L6	L7	L8	All
Vector-RAG	**0.673**	0.808	**0.824**	0.833	0.653	0.516	0.449	**0.486**	0.460	0.691
RAPTOR	**0.673**	0.835	0.797	0.917	0.776	**0.581**	0.500	0.444	**0.476**	**0.711**
GraphRAG	0.647	0.742	0.410	0.750	0.211	0.570	0.203	0.139	0.159	0.542
GFM-RAG	0.638	**0.844**	0.779	**1.000**	**0.796**	**0.581**	**0.746**	0.292	0.175	0.705

Table 2: LLM-as-Judge score for each approach across question levels.

GraphRAG demonstrates limited utility beyond basic extraction tasks.

5.3 Metrics comparative analysis

Multi-metric evaluation (details are available in Table 6 of Appendix E) reveals significant measurement discrepancies across assessment approaches. BERTScore maintains consistently high scores (0.86-0.91) across all systems and levels, suggesting preservation of semantic similarity even when factual accuracy suffers. SQUAD Exact Match demonstrates extremely low performance (0.0-0.07) across all systems, indicating minimal exact string matching between generated and reference answers. SQUAD F1 shows moderate performance (0.3-0.5) with significant fluctuations, suggesting partial word overlap between predictions and references.

LLM-based metrics (Figures 13, 14) provide more nuanced assessment. LLMLogScore shows convergence at L3-L4 (0.95-1.00 across systems), then diverges substantially, with some systems recovering at L7-L8 while others decline. LLM-Judge shows varying patterns by system, with performance peaking at level 3 then generally declining, though with significant differences between systems at higher levels.

GraphRAG shows a significant discrepancy between LLMLogScore and LLM-Judge, particularly at higher difficulty levels. Both metrics peak at Level 3, suggesting this is GraphRAG's optimal complexity zone. However, at levels 6-8, LLMLogScore remains moderate (~ 0.6) while LLM-Judge drops severely (~ 0.2). This suggests GraphRAG retrieves semantically relevant information but fails to synthesize factually correct answers at high complexity. The system appears to hit a complexity ceiling beyond Level 3-4, where it likely produces semantically similar but structurally different answers that fall into a "gray area" - good enough for high semantic similarity scores but not meeting the binary judge's correctness threshold.

The benchmark appears to pose increasingly challenging questions at higher levels, as evidenced by the decline in performance in LLM-based metrics. The contrast between BERTScore (consistently high) and SQUAD metrics (consistently low) suggests that responses maintain word-level semantic similarity to references, without exact matching. In combination, this indicates the challenging nature of the benchmark, where similar but incorrectly retrieved context can lead to responses with good token-level semantic similarity, but where nuances in compositionality significantly impact the conveyed meaning.

Our multi-metric evaluation approach uses BERTScore and LLM-Judge to assess semantic correctness while maintaining awareness of formatting precision. The universally low EM scores across all systems and complexity levels suggest that reference answer formatting, rather than content accuracy, drives these results. Moreover, FinDocRAG evaluates general financial document understanding—spanning numerical data, legal terms, and marketing content—rather than specialized numerical reasoning tasks. In deployment scenarios, post-processing can standardize formatting, making semantic accuracy the primary criterion for RAG system selection. Developing unified metrics encompassing both semantic understanding and numerical precision represents important future work beyond this paper's scope.

6 Conclusion

Financial institutions increasingly rely on RAG systems for document processing, yet systematic evaluation on industry-specific content has remained limited. We assessed four state-of-the-art RAG architectures using *FinDoc-RAG*, a benchmark comprising 600+ question-answer pairs from real financial documents across nine complexity levels targeting factual extraction, information integration, and cross-document synthesis.

Our evaluation reveals that architectural choice

impacts performance on different question types: while leading systems achieve 0.84 accuracy on basic extraction, performance drops substantially for complex synthesis tasks (0.31 average), with architectural differences amplifying at higher complexity levels. Semi-structured approaches (RAPTOR) provide the most consistent performance across complexity levels, while knowledge graph augmentation (GFM-RAG) excels at complex reasoning but shows variable baseline performance. Our analysis reveals that no single architecture dominates across all task types.

These findings highlight the need of benchmarks like *FinDoc-RAG* for measuring progress toward reliable financial document understanding systems.

Limitations

We acknowledge several limitations of *FinDoc-RAG*. First, while our approach to generating QA pairs using LLMs across different complexity levels provides a comprehensive evaluation framework, automatically generated questions may occasionally lack the depth that human-crafted questions might offer. Despite our manual validation showing high correctness rates, there remains inherent variability in the LLM output that could affect the quality of the question. Second, while our collection of documents spans multiple types of financial documents, it still represents a subset of the vast landscape of financial documentation. Finally, our metrics comparison reveals challenges in accurately measuring RAG system performance, suggesting that even our multifaceted evaluation approach may not capture all dimensions of answer quality relevant to financial domain experts.

It is important to note that our evaluation focuses on RAG architectures, which currently represent the most prevalent and practical approach for incorporating external knowledge into LLM-based question answering. This is due to their ability to maintain data locality and provide retrieval transparency - critical requirements in regulated financial environments. In contrast, long-context LLMs face significant deployment challenges in financial institutions, including prohibitive high computational costs for inference over a large document collections and challenges in explaining retrieval decisions.

Acknowledgements

This work has been supported by UBS Switzerland AG and its affiliates.

References

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305. Association for Computational Linguistics.

Jian Chen, Peilin Zhou, Yining Hua, Loh Xin, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024a. FinTextQA: A dataset for long-form financial question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6047, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. 2022. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. *Preprint*, arXiv:2504.15800.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. 2025. MRAG-bench: Vision-centric evaluation for retrieval-augmented multimodal models. In *The Thirteenth International Conference on Learning Representations*.

Shantanu Jain. 2025. [link].

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. 2025. Lara: Benchmarking retrieval-augmented generation and long-context llms – no silver bullet for lc or rag routing. *Preprint*, arXiv:2502.09977.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. Leveraging Large Language Models for NLG Evaluation: A Survey. *Preprint*, arXiv:2401.07103.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. Gfm-rag: Graph foundation model for retrieval augmented generation. *Preprint*, arXiv:2502.01113.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems*, volume 37, pages 95963–96010. Curran Associates, Inc.

Nghia Trung Ngo, Chien Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2024. Comprehensive and practical evaluation of retrieval-augmented generation systems for medical question answering. *Preprint*, arXiv:2411.09213.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *Preprint*, arXiv:2408.10343.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. SPIQA: A dataset for multimodal question answering on scientific papers. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Zafaryab Rasool, Stefanus Kurniawan, Sherwin Balugo, Scott Barnett, Rajesh Vasa, Courtney Chesser, Benjamin M. Hampstead, Sylvie Belleville, Kon Mouzakis, and Alex Bahar-Fuchs. 2024. Evaluating llms on document-based qa: Exact answer selection and numerical extraction using cogtale dataset. *Natural Language Processing Journal*, 8:100083.

Ehud Reiter. *Natural Language Generation*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Kyunghwan Sohn, Sunjae Kwon, and Jaesik Choi. 2021. The global banking standards QA dataset (GBS-QA). In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 19–25, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. 2025. Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation. *Preprint*, arXiv:2412.10704.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges. *Preprint*, arXiv:2406.12624.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Dataset generation details

A.1 Converting documents to Markdown text

The original documents are provided in PDF[2] and are professionally typeset for human consumption. As a result, they feature multi-column layouts, text flowing around tables and figures, and a variety of typographical elements, including bold headers, bullet points, footnotes, superscripts, and subscripts. Since PDF documents are designed for visual presentation, text elements are absolutely positioned. As such, they may not be stored in the same order as they appear on the page. Further, headings and body text are kept in separate text boxes that differ only in font size. Without any additional distinguishing features, this poses a challenge for machines. Consequently, many PDF software packages rely on a complex set of heuristics that are prone to error and may introduce artifacts during content extraction. To generate a faithful Markdown representation from PDF documents, we developed a three-step conversion pipeline:

1. **Image Conversion:** Each PDF page is converted into a flat image that captures its complete visual layout, including multi-column arrangements, figures, and other graphical elements, which helps preserve the original presentation.

2. **Text Extraction:** In parallel, we programmatically extract the text content from each page. Although this extraction is a best-effort process that likely omits layout details or includes minor artifacts, it produces a reliable reference of the page's content.

3. **Markdown Generation:** Both the page image and the extracted text are supplied to a vision-enabled Large Language Model (LLM). The text acts as guidance to reduce hallucinations while the image provides visual context. The LLM generates a Markdown output that preserves key stylistic and structural elements such as headers, bold and italic text, bullet points, tables, and hyperlinks. Further, explicit rules are applied to handle layout features that do not have direct Markdown counterparts (e.g., footnotes, super, and subscripts).

A key challenge in our pipeline is maintaining continuity across pages, especially given variations in header levels and layout elements that span multiple pages, such as tables without repeated headers. To address these issues, we implemented a rolling *'continuity bridge'* text (see algorithm 1 in Appendix A). The process is repeated for each subsequent page until the entire document is processed. No additional cleaning was applied beyond the structured conversion process described above. All original features were preserved exactly as they appear in the source documents, including spellings, product names or hyphenated words at page breaks. This ensures that the Markdown representation most accurately mirrors the original PDF documents.

A.2 Document Processing

The pseudo-code 1 describes the algorithm used to process a document.

[2]Portable Document Format

Cluster	Field	Content
0	Filenames	95be, 91f6, f059, 095f
	Main product or service	UBS duo Saving; UBS Investment Fund Account; UBS Fixed Term Deposit; Foreign Exchange (FX) & Precious Metal (PM) Spot, Forward & Swaps
	Coherence	95be, 91f6 and f059 cover UBS retail savings/investment products with similar terms, while 095f lists FX & PM mark-ups for the same clientele, making it a mild outlier.
1	Filenames	3996, b7e1, 3000, e3da, 6592, 6253, 2f9d, 8b09
	Main product or service	UBS Visa Corporate Card; UBS Commercial Credit Cards; Power of Attorney for UBS Commercial Cards; UBS Platinum Credit Card; UBS Travel Insurance Plus; UBS Gold Credit Card
	Coherence	All files concern UBS credit cards: offering, product sheets, insurance add-ons, and legal/administrative details.
2	Filenames	eec1, 9082, a02d, 7f44
	Main product or service	UBS Vitainvest Funds Sustainable; UBS Vitainvest World 25 Sustainable U; UBS Vitainvest Swiss 75 Sustainable U; BVG 21 Reform
	Coherence	eec1, 9082 and a02d are Vitainvest Sustainable fund sheets sharing Swiss-pension and ESG themes; 7f44 adds broader BVG reform context. All fit a "Swiss sustainable retirement investing" topic.
3	Filenames	29e9, 9542, fde3, 873f, 6c47
	Main product or service	UBS Investment Fund Account; UBS key4 smart investing; UBS Personal Account; UBS Manage [CH]; UBS key4 Banking; UBS me Banking Package; UBS Fisca Account; UBS Vested Benefits Account; UBS Investment Funds
	Coherence	29e9, 9542 and 873f focus on investment or discretionary-management offers; fde3 and 6c47 outline the core account and fee framework. Common threads are low entry thresholds, digital access and sustainability, with the payments documents forming the loosest link.

Table 3: Overview of document clusters generated during creating of L5-L8 QAs.

Algorithm 1 Document Processing Across Pages

For Page 1:
Input: Image and extracted text of page 1
Output:
    Markdown text for page 1
    Continuity bridge text describing the structural context on page 1

For Page k+1:
Input:
    Image and extracted text of page k+1
    Continuity bridge text from page k
Output:
    Markdown text for page k+1
    Updated continuity bridge text for page k+1

B Settings for Question Generation, Experiments and Evaluation

B.1 Question generation Settings

QA pairs of levels L0-L3 were generated with the OpenAI model `gpt-4o`, whereas levels L4-L8 were generated using `gpt-4o-mini`. For generating embeddings of Levels L5-L8, the OpenAI `text-embedding-ada-002` model was used.

B.2 RAG-QA Settings

General setup: All algorithm use OpenAI's `gpt-4o-mini` as completion model LLM and `text-embedding-ada-002` as embedding type.

Vector-based: This is an ad hoc implementation of the standard RAG pipeline (Lewis et al., 2021). The preprocessing step includes a chunking of each document using Tiktoken (Jain, 2025)'s tokenizer with the `text-embedding-ada-002` encoding. Each chunk has a size of 100 tokens. The selection of the best chunks is made by minimizing the cosine distance between the input query and all the chunks available in the embedding space.

RAPTOR: (Sarthi et al., 2024) Using the RAPTOR's building algorithm, we created a tree for each document, then all the trees have been merged

together (this is a custom change since the available implementation does not support multiple documents). RAPTOR retrieval process has been used with the collapsed tree parameter set to `TRUE`. All other parameters have been kept with default values.

Graph-RAG: (Edge et al., 2025) Despide the changes to use the standard models, this experiment has been run with default parameters using local search mode.

KG-RAG: For KG-RAG, we followed the baseline settings from (Luo et al., 2025) with the 8M-pretrained model [3]. Using the default LLM prompting set-ups, we create the KG from the document sets. For the entity-linking module, the ColBERTv2 model (Santhanam et al., 2022) was employed with a baseline cosine similarity of 0.8 and a maximum default of 100 similar neighbours. This controls the number of synonymous edges to be added between similar entities during the entity-linking phase.

B.3 Evaluation settings

BERTScore evaluations were run with an off-the-shelf `roberta-large` model trained on English texts. LLMJudge and LLMLOgScore were both run using the OpenAI model gpt-4o-mini.

C Dataset Analysis

C.1 Dataset composition

Dataset composition analysis is presented in Figures 4, 5 and 6.

C.2 Complexity and Diversity

Figures 7, 8, 9, and 10 depict the statistical analysis of the dataset in terms of complexity and diversity.

C.3 Information analysis

The information analysis is depicted in Figures 11 and 12.

[3] https://huggingface.co/rmanluo/GFM-RAG-8M

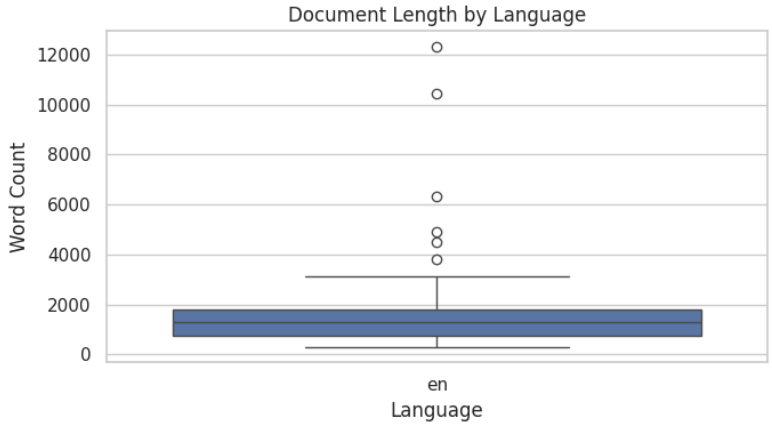

Figure 4: Document length in words.

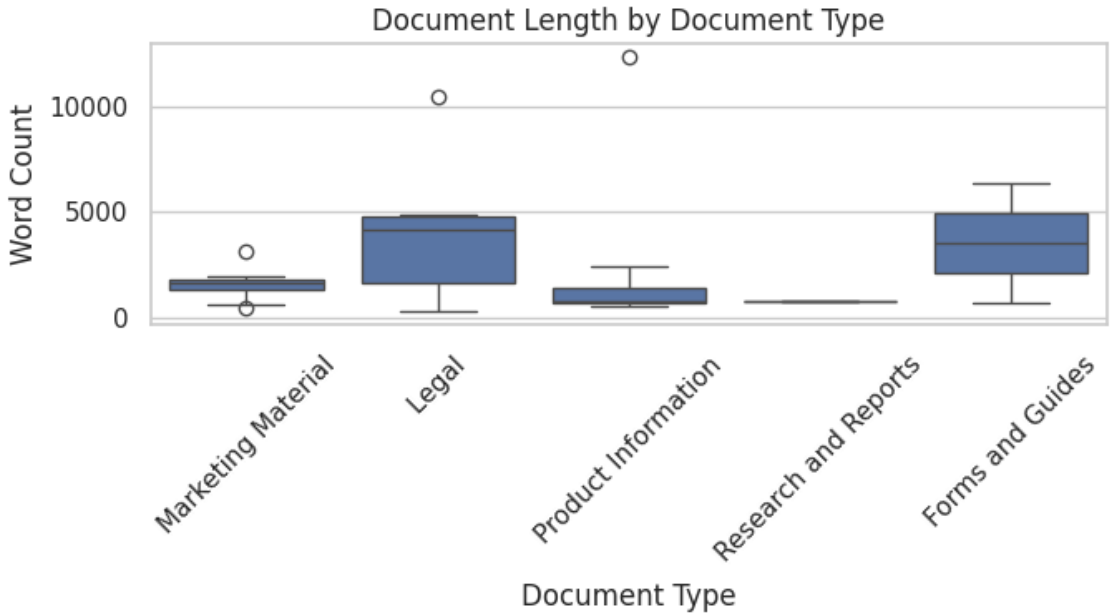

Figure 5: Document length in words by document type.

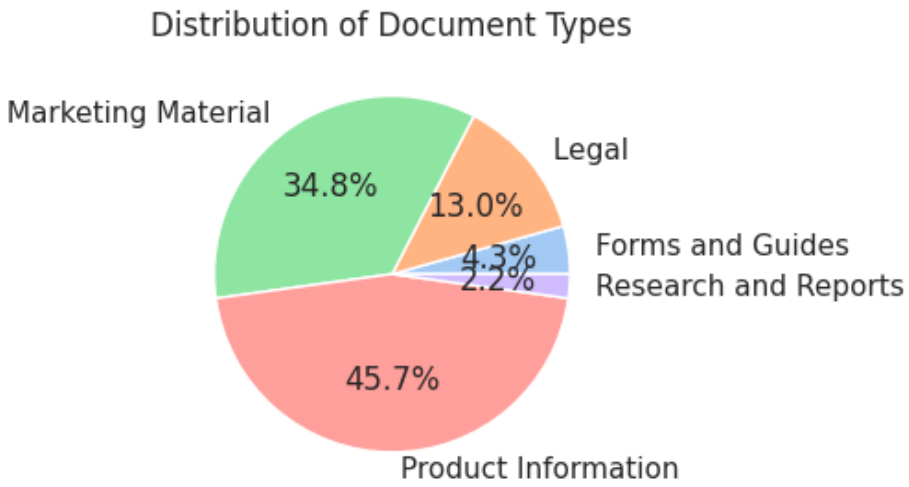

Figure 6: Distribution by document types.

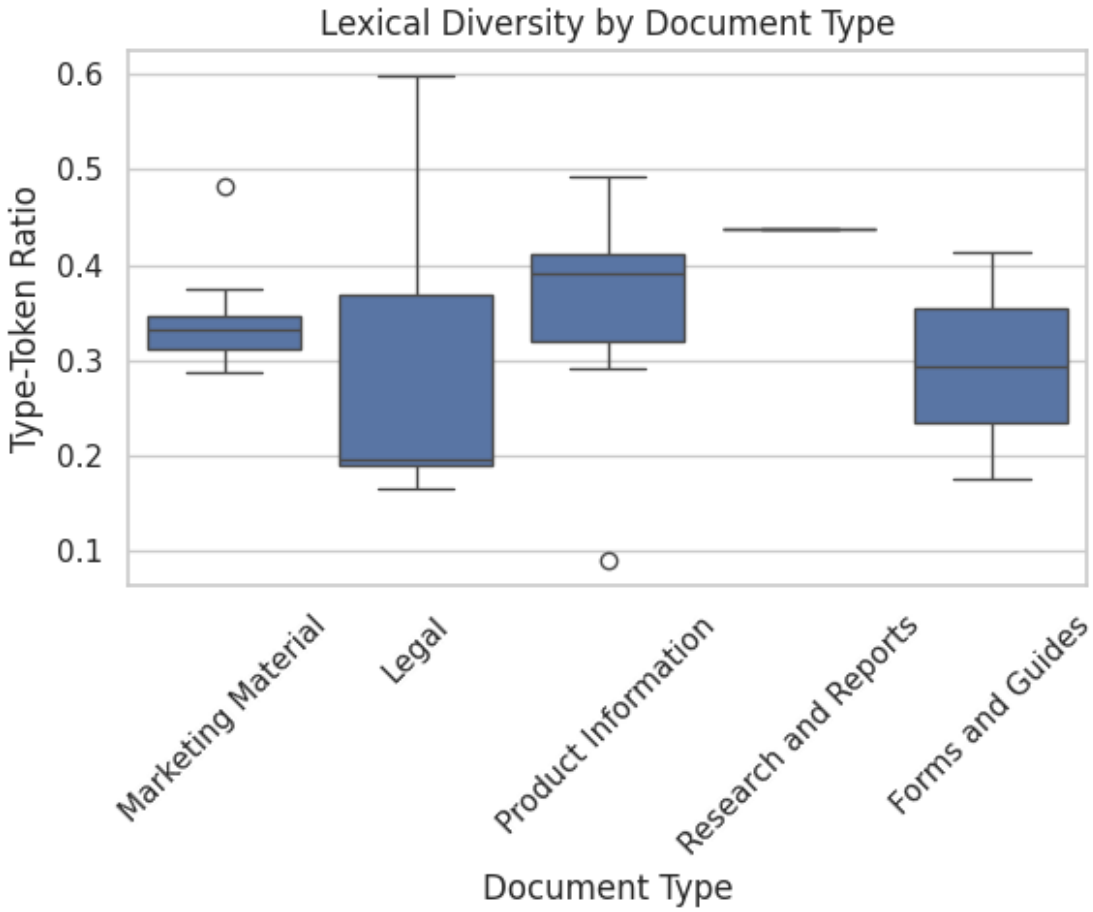

Figure 7: Lexical diversity (i.e., type-token ratio - TTR) by type.

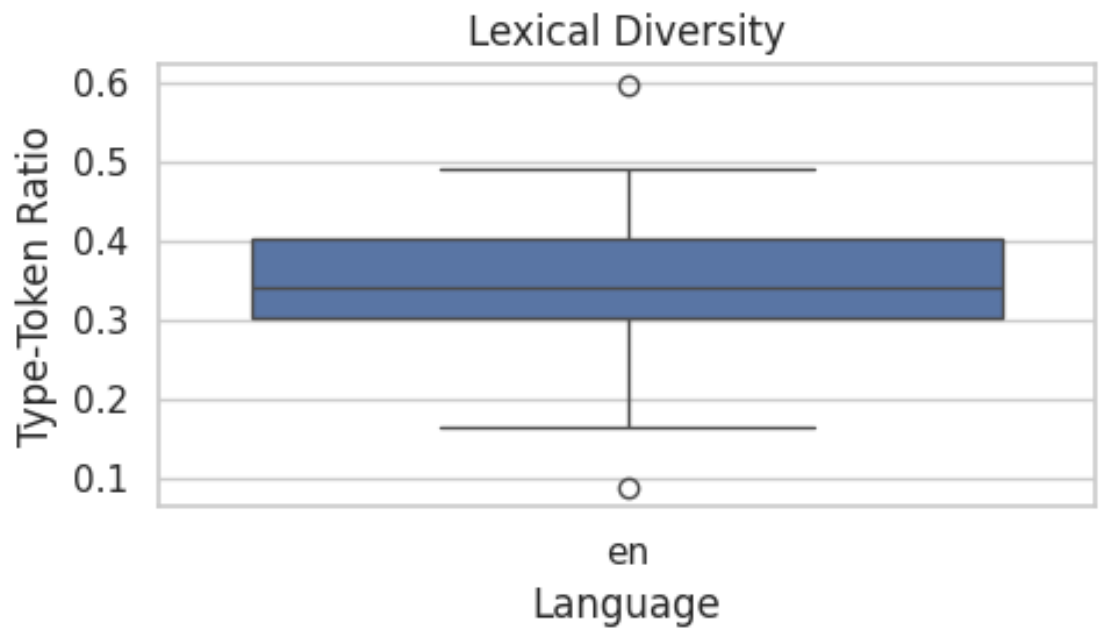

Figure 8: Lexical diversity (i.e., type-token ratio - TTR) by language.

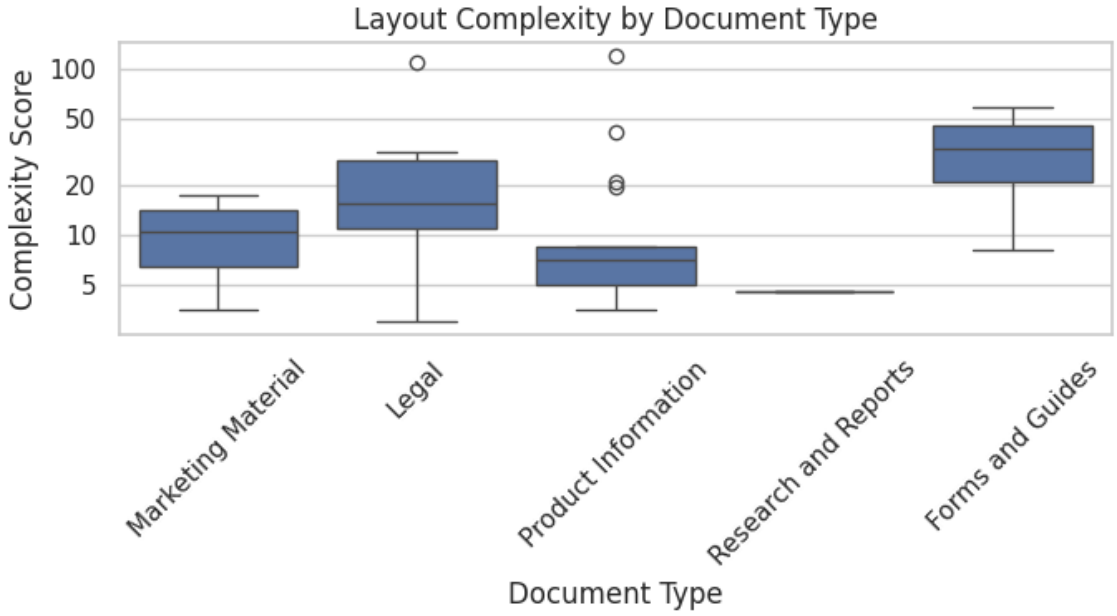

Figure 9: Layout complexity consists in an heuristic based on number of tables, section and images and, the maximum nesting level of sections.

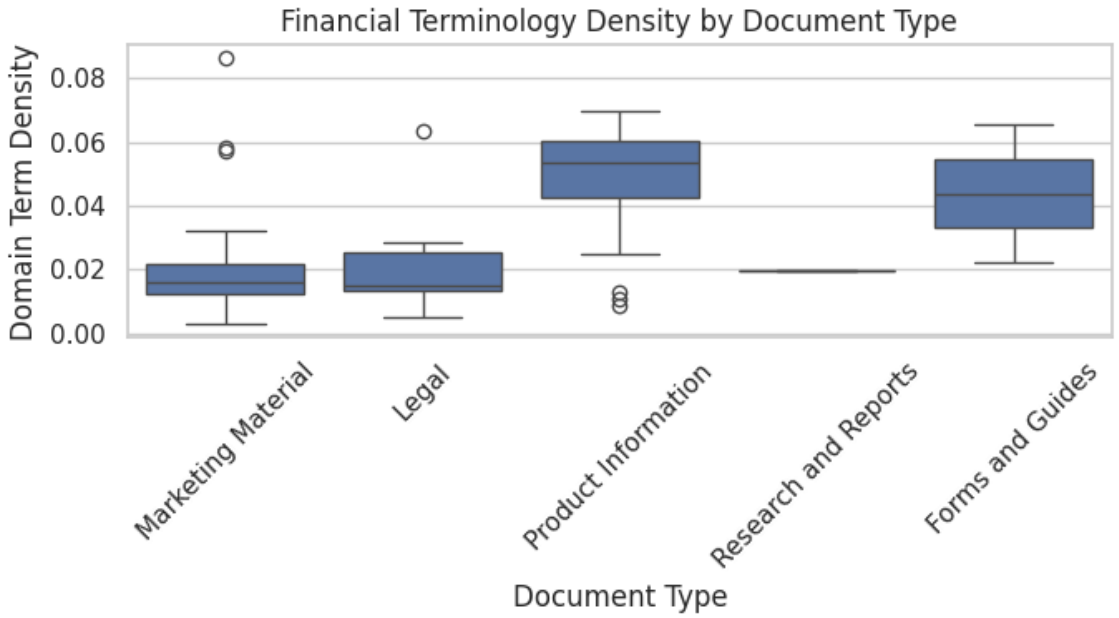

Figure 10: Density of financial terms (account, asset, balance, bond, capital, credit, debt, dividend, equity, fund, interest, investment, liability, mortgage, portfolio, risk, share, stock, tax, yield) by document type.

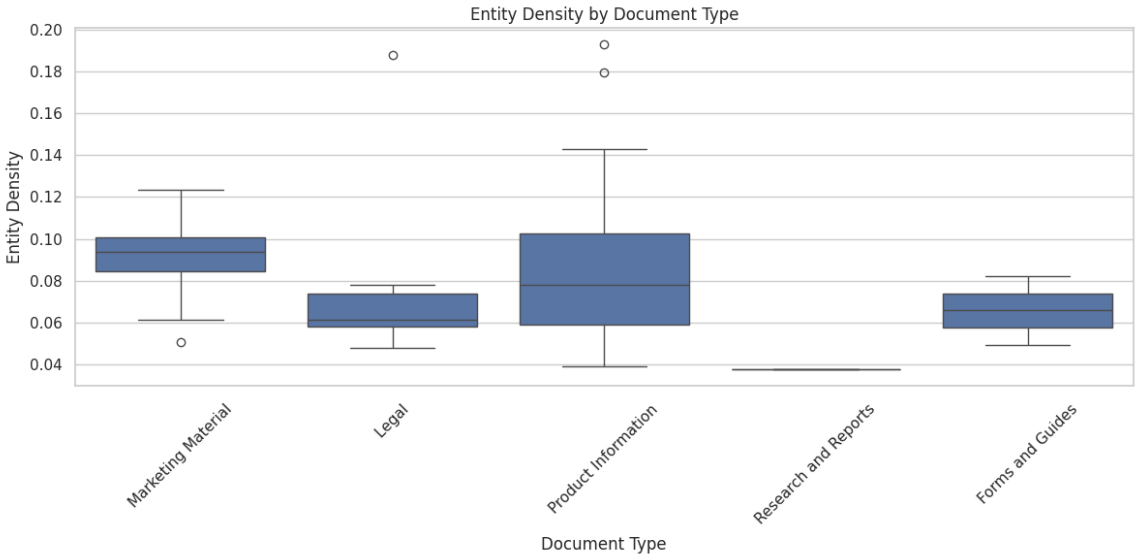

Figure 11: Measure information density using entity density as a proxy (i.e., ratio of entities by tokens).

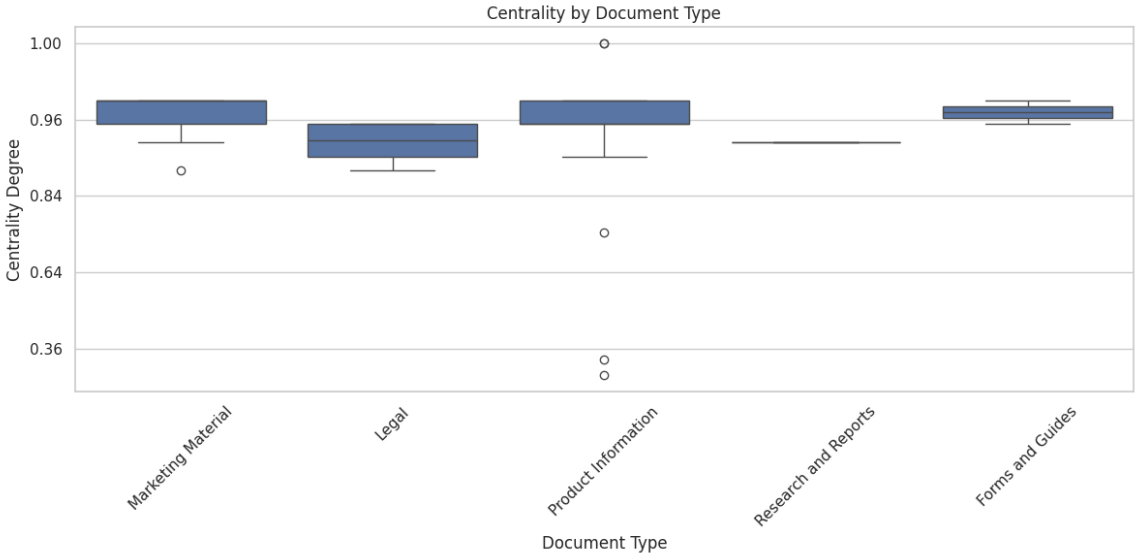

Figure 12: Document's centrality degree by document type.

D Human Validation Details

To evaluate the quality of the Q&A dataset we constructed, we conducted a human evaluation focusing on two key aspects: **Ambiguity** and **Correctness**.

For this evaluation, a randomly sampled representative subset of Q&A pairs was manually reviewed for each level. Correct and non-ambiguous examples are shown in table 5, meanwhile incorrect and/or ambiguous examples are shown in table 4. The evaluation criteria were defined as follows:

- **Ambiguity (Question-level)**: A question was marked as ambiguous if it met **any** of the following conditions:
    1. The question was unclear or poorly formulated.
    2. The question was too general, allowing multiple distinct answers to be considered valid.
    3. The question included vague or forbidden referents (e.g., "in this story"), which are not self-contained or interpretable without external context.
- **Correctness (Answer-level)**: An answer was marked as correct if it satisfied **all** of the following conditions:
    1. It directly addressed the question being asked.
    2. It was factually accurate and faithful to the content of the source document(s).
    3. It was complete, providing all relevant and necessary details to fully answer the question.

For levels **4 through 8**, the Q&A generation process follows a slightly different paradigm: questions at these levels are intended to reflect the perspective of users with limited prior knowledge of the subject matter. As a result, the evaluation criteria were applied with a more relaxed interpretation. Specifically:

- Questions were allowed to be more general, provided they remained clear and self-contained.
- Answers were not required to include every possible detail, as long as they remained accurate and sufficiently informative given the context and intent.

E Benchmark evaluation

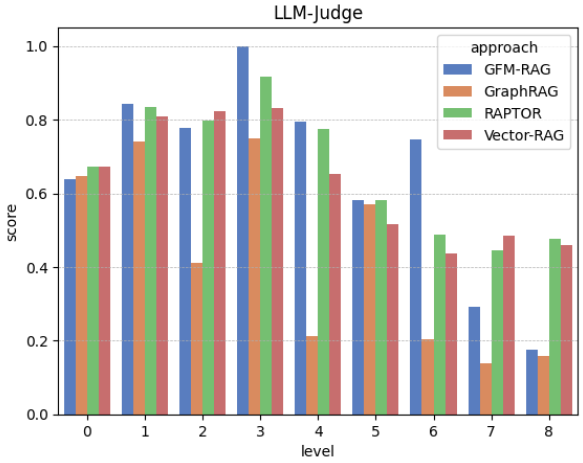

Figure 13: LLM-as-Judge comparison of RAG-QA approaches across question levels.

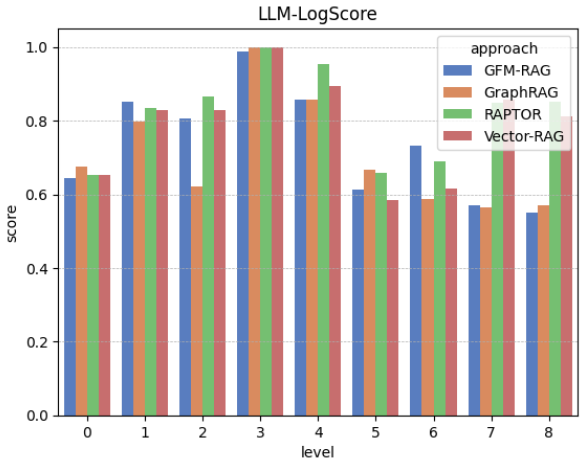

Figure 14: LLMLogScore comparison of RAG-QA approaches across question levels.

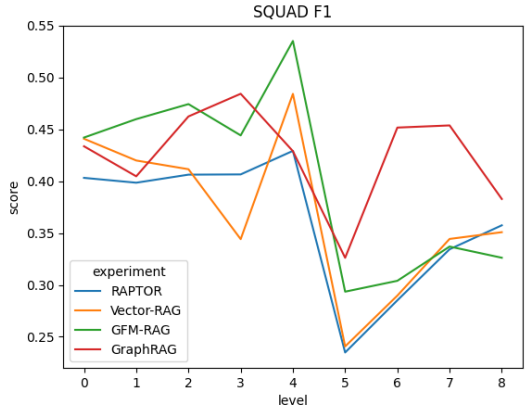

Figure 15: SQUAD-F1 comparison of RAG-QA approaches across question levels.

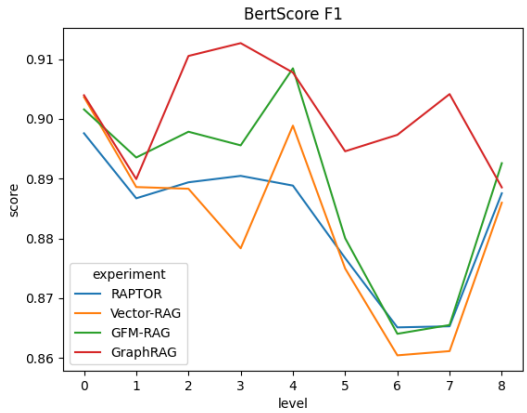

Figure 16: BERTScore-F1 comparison of RAG-QA approaches across question levels.

Group: Incorrect and/or non-ambiguous examples

Example 1
Q: What does the UBS Gold Credit Card offer for airport lounge access?
A: The card provides membership to Priority Pass for access to over 1300 lounges worldwide.
LEVEL: 5
ID: 99288f2fd571eb545bb18ba4b33caa31b6adb8eb7f6c92953b7665fae34fdc48
Ambiguity: No
Correctness: No - Answer is missing key detials, and therefore is uncomplete.
Source: 8b09

Example 2
Q: What is this "infrastructure investment" that you are talking about?
A: Infrastructure investment refers to investments in large-scale assets that are critical to economic productivity, such as regulated utilities, transportation systems, power stations, and telecommunications capabilities. These investments are expected to provide a combination of steady income and potential for capital appreciation. They are seen as resilient to macroeconomic and interest rate pressures, offering diversification benefits and stable long-term cash flows. The document highlights that infrastructure investments could be particularly timely due to fiscal initiatives like the Inflation Reduction Act in the U.S. and the Green Deal in Europe, which may funnel investments into infrastructure assets for years to come.
LEVEL: 4
ID: 70b7b8445c454663b7a9abb5420495910155266f6ed035657e9e823d93108c13
Ambiguity: Yes - the term "you are talking about" is ambiguous. The question includes a vague or forbidden referent.
Correctness: Yes
Source: 2642

Example 3
Q: For UBS Commercial Credit Cards, what happens if the card is used for illegal purposes?
A: The use of the card for illegal purposes is prohibited.
LEVEL: 0
ID: 42142c0776f5e06328d75874d498712e1740d9f8f35fbcfd1c81cf03eeaaf70e
Ambiguity: Yes - The question is poorly formulated. It is unclear whether it refers to (1) the actions a cardholder must take in the event of illegal card use, (2) UBS's or holder response or liability in such a case, or (3) the legal consequences for the cardholder. As written, it leaves room for multiple interpretations.
Correctness: No - The answer do not address the question and it is not faithful to the content of the source document, nor presents all the necessary detials.
Source: 2f9d

Table 4: Incorrect and/or non-ambiguous examples in our human evaluation.

Group: Correct and non-ambiguous Examples

Example 1
Q: What is included in the UBS key4 Banking package?
A: The UBS key4 Banking package includes personal and savings accounts, debit and credit cards, and digital banking services.
LEVEL: 5
ID: 0d43c3a2df5f78dc8704238b5e43c093e502be215e8704dd88decde92ed62ab1
Ambiguity: No
Correctness: Yes
Source: 6c47

Example 2
Q: How do the investment strategies of the UBS Duo Saving and the UBS Fixed Term Deposit cater to different investor needs?
A: The UBS Duo Saving caters to investors looking for both stability and growth by splitting investments between a fixed-term deposit and an investment component, while the UBS Fixed Term Deposit is tailored for risk-averse investors seeking security with a fixed interest rate. The Duo Saving states, the product aims to balance stability and growth, while the Fixed Term Deposit is described as designed for risk-averse investors.
LEVEL: 6
ID: d975a031c131f7a4b050e84f42acf45f6fd27cfc6f3ae027a263f768073060de
Ambiguity: No
Correctness: Yes
Source: f059, 95be

Example 2
Q: What is the management fee per annum for the UBS Vitainvest Swiss 75 Sustainable U fund?
A: 1.20%
LEVEL: 0
ID: e88bf375f47e9bb251dcdeefc5eba2a0682b7f0cb4e1e477bd5426991d7c69bf
Ambiguity: No
Correctness: Yes
Source: eec1

Table 5: Correct and non-ambiguous examples in our human evaluation.

Approach	level	Q_count	llmjudge (std)	llmlogscore	bertscore_F1	squad_EM	squad_F1
GFM-RAG	0	201	0.638 (0.010)	0.644	0.902	0.070	0.442
GFM-RAG	1	182	0.844 (0.005)	0.852	0.894	0.033	0.460
GFM-RAG	2	74	0.779 (0.023)	0.806	0.898	0.000	0.474
GFM-RAG	3	4	1.000 (0.000)	0.988	0.896	0.000	0.444
GFM-RAG	4	49	0.796 (0.033)	0.858	0.908	0.020	0.535
GFM-RAG	5	31	0.581 (0.000)	0.613	0.880	0.000	0.294
GFM-RAG	6	46	0.746 (0.037)	0.732	0.864	0.000	0.304
GFM-RAG	7	24	0.292 (0.000)	0.569	0.866	0.000	0.337
GFM-RAG	8	21	0.175 (0.059)	0.551	0.893	0.000	0.326
GraphRAG	0	201	0.647 (0.015)	0.676	0.904	0.035	0.434
GraphRAG	1	182	0.742 (0.027)	0.797	0.890	0.000	0.405
GraphRAG	2	74	0.410 (0.006)	0.622	0.911	0.000	0.463
GraphRAG	3	4	0.750 (0.204)	1.000	0.913	0.000	0.484
GraphRAG	4	49	0.211 (0.019)	0.856	0.908	0.000	0.429
GraphRAG	5	31	0.570 (0.040)	0.667	0.895	0.000	0.326
GraphRAG	6	46	0.203 (0.027)	0.588	0.897	0.000	0.452
GraphRAG	7	24	0.139 (0.052)	0.565	0.904	0.000	0.454
GraphRAG	8	21	0.159 (0.022)	0.570	0.889	0.000	0.383
RAPTOR	0	201	0.673 (0.009)	0.654	0.898	0.025	0.403
RAPTOR	1	182	0.835 (0.008)	0.835	0.887	0.027	0.399
RAPTOR	2	74	0.797 (0.029)	0.866	0.889	0.014	0.406
RAPTOR	3	4	0.917 (0.118)	1.000	0.890	0.000	0.407
RAPTOR	4	49	0.776 (0.044)	0.954	0.889	0.000	0.429
RAPTOR	5	31	0.581 (0.026)	0.659	0.877	0.000	0.235
RAPTOR	6	46	0.500 (0.031)	0.690	0.865	0.000	0.285
RAPTOR	7	24	0.444 (0.104)	0.848	0.865	0.000	0.334
RAPTOR	8	21	0.476 (0.000)	0.851	0.888	0.000	0.358
Vector-RAG	0	201	0.673 (0.006)	0.652	0.904	0.035	0.441
Vector-RAG	1	182	0.808 (0.009)	0.830	0.889	0.027	0.420
Vector-RAG	2	74	0.824 (0.022)	0.828	0.888	0.000	0.412
Vector-RAG	3	4	0.833 (0.118)	0.998	0.878	0.000	0.344
Vector-RAG	4	49	0.653 (0.050)	0.893	0.899	0.000	0.484
Vector-RAG	5	31	0.516 (0.000)	0.584	0.875	0.000	0.241
Vector-RAG	6	46	0.449 (0.041)	0.617	0.860	0.000	0.290
Vector-RAG	7	24	0.486 (0.071)	0.858	0.861	0.000	0.344
Vector-RAG	8	21	0.460 (0.045)	0.811	0.886	0.000	0.351

Table 6: System performance across question difficulty levels. Q_count indicates number of questions per level. Evaluation metrics: llmjudge (LLM-as-judge accuracy), llmlogscore (log probability scores), bertscore_F1 (semantic similarity), squad_EM (exact match), squad_F1 (token-level F1).

F Prompts for generating level 4 questions

Prompt 1 : question generation prompt

system : You are client_profile_name: Here is the description of your profile: client_profile Ensure that you always write in the style associated with your assigned profile.
user: Read the following markdown document describing a banking product and generate a **simple, naïve question** about it. The question should be something a person with **no prior knowledge of banking** might ask when encountering this product for the first time. Assume the person has **little to no financial expertise** and is genuinely curious about basic concepts. **Guidelines for the question:** - It should be **basic and straightforward**, avoiding complex financial terminology. - It should reflect **genuine curiosity**, as if someone is trying to understand the very basics. - The question **must explicitly reference the banking product** (e.g., *"a savings account"*, *"this type of loan"*, *"this investment plan"*) instead of using vague words like "this" or "it." - The answer **must be found within the document**—do not ask questions unrelated to the content. - Do **not** add explanations or extra context—**just generate the question**. Wrap the question with the «Q» and «/Q» tags. ### Banking Product Description (Markdown Format): "'markdown banking_markdown "'

Prompt 2 : answer generation prompt

system: Your task is to: 1. Read the provided markdown document describing a banking product, and the provided question. 2. First, answer the question using the information from the markdown document. Wrap the answer with the «A» and «/A» tags. 3. If it's not possible to answer given the document, answer with «A» No answer «/A».
user: ### Banking Product Description (Markdown Format): "'markdown banking_markdown "' ### Question: question

Prompt 3 : quotation generation prompt

system: Your task is to: 1. Read the provided markdown document describing a banking product, and the provided question. 2. Provide a quotation from the document that answers the provided question. When quoting, wrap the quotation with «Quot» and «/Quot» tags. 3. If no quotation answers the question, answer with «Quot» No quotation «/Quot».
user:### Banking Product Description (Markdown Format): "'markdown banking_markdown "' ### Question: question