OmniEval: An Omnidirectional and Automatic RAG Evaluation Benchmark in Financial Domain

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) has emerged as a key application of large language models (LLMs), especially in vertical domains where LLMs lack domain-specific knowledge. Nevertheless, current RAG benchmarks often suffer from narrow scenarios and limited evaluation dimensions, hindering an all-sides understanding of RAG models in real-world vertical applications. This paper introduces OmniEval, an omnidirectional and automatic RAG benchmark for the financial domain, featured by its omnidirectional evaluation framework: First, we categorize RAG scenarios by five task classes and 16 financial topics, leading to a *matrix-based structured assessment*. Next, we leverage a *multi-dimensional and auto-chained data generation pipeline* that integrates LLM-based automatic generation and human annotation approaches, creating high-quality evaluation instances. Further, we adopt a *multi-stage evaluation* to assess both retrieval and generation performance, resulting in a holistic RAG evaluation. Finally, rule-based and LLM-based metrics are combined to build a *multi-level evaluation system*. Our experiments indicate the performance of RAG systems varies across topics and tasks, highlighting the importance of multi-aspectives and structured assessments to better locate the advantages and disadvantages of RAG systems. We anonymize our code in https://anonymous.4open.science/r/OmniEval-anonymous-8E48.

1 Introduction

RAG techniques have gained prominence as one of the most widespread and practical applications of LLMs. Particularly in specialized domains where LLMs often lack in-domain expertise, RAG models effectively incorporate external domain corpora and the internal knowledge of LLMs to enhance the overall quality of generative AI systems. Despite advances, the challenge of automatically building high-quality omnidirectional benchmarks to yield all-sided evaluation profiles for RAG models remains unresolved. In this study, we introduce an omnidirectional and automatic benchmark, OmniEval, designed to assess RAG systems in a widely adopted vertical domain, finance. Its versatility and automaticity are indicated by the following angles:

Matrix-based RAG scenario evaluation. Versatile response capabilities are essential for RAG systems to handle user queries spanning various scenarios. For example, some queries seek factual information that can be extracted from web pages, while others may require complex financial computations. To assess such versatility, we classified RAG scenarios into five common tasks, *i.e.*, extractive question-answering (QA), multi-hop reasoning, contrast QA, long-form QA, and conversational QA. Besides, in specialized domains like finance, user queries often fall into distinct domain topics. Consequently, we also distinguish RAG scenarios based on topical categories of queries, recognizing 16 common subcategories in the finance domain. These two orthogonal taxonomies lead to matrix-based RAG evaluation scenarios and support all-sided profiles for RAG systems (an example is visualized in Figure 1).

Multi-dimensional and auto-chained data generation. To create extensible and high-quality evaluation datasets, we integrate the GPT-4-based automated generation and human annotation approaches. The former provides flexibility, allowing the data generation pipeline to adapt to various domains, and the latter guarantees the quality of the datasets. Automatic topic recognition and quality inspection are further introduced to ensure the reliability of generated instances. *Multi-stage evaluation.* The quality of the retrieval and generation processes are both important when evaluating the RAG pipeline, especially for vertical domains, since gen-

*Corresponding author.

Benchmark	Evaluation Scenarios		Data Generation		Evaluation Metrics			Evaluation Models	
	Task-Spe.	Topic-Spe.	Manual	Auto.	Rule	Model	Human	Retriever	Generator
PIXIU (Xie et al., 2023)	✓	✗	✗	✗	✓	✗	✓	✗	✓
DISC-FinLLM (Chen et al., 2023)	✓	✗	✗	✓	✓	✓	✗	✗	✓
FinanceBench (Islam et al., 2023)	✓	✓	✓	✗	✗	✗	✓	✗	✓
AlphaFin (Li et al., 2024)	✓	✗	✗	✗	✓	✓	✓	✗	✓
FinBen (Xie et al., 2024)	✓	✗	✗	✗	✓	✓	✗	✗	✓
FinTextQA (Chen et al., 2024a)	✓	✗	✗	✗	✓	✓	✗	✓	✓
OmniEval	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: The comparison between our proposed benchmark and existing financial benchmarks. "Auto." is short for "Auto-generated", "Spe." is short for "Specific".

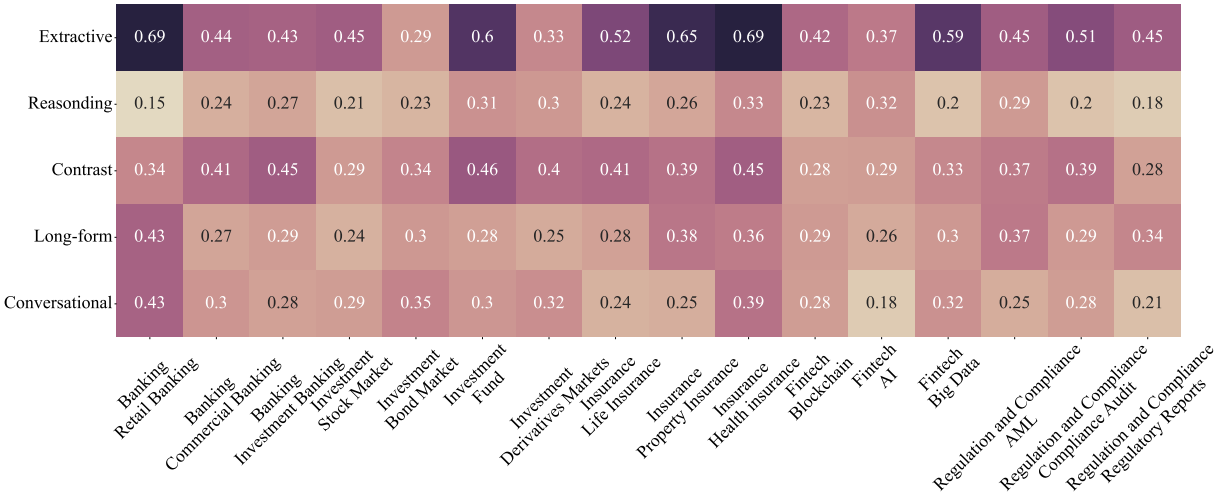

Figure 1: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Llama3.1-70B-Instruct on human-annotated sets.

eral retrievers may lack expert knowledge and potentially compromise the response quality. Therefore, OmniEval evaluates both retriever and generator performance to provide a comprehensive assessment for RAG systems. *Multi-level evaluation metrics.* For the evaluation systems, we build our evaluation metrics by combining rule-based and LLM-based metrics together. The former embodies widely used evaluation metrics, such as MAP and Rouge, offering solid evaluation results. The latter is produced from fine-tuned LLMs to achieve high-level evaluation beyond term-level matching, such as hallucination detection and numerical accuracy. To ensure the reliability of our LLM-based evaluation, we further manually annotate some evaluation samples and fine-tune Qwen2.5-7B-Instruct (Team, 2024) to build LLM evaluators.

As a result, OmniEval contains 11.4k automatically generated test examples and 1.7k human-annotated test examples. We further split out 3k automatically generated examples as a training set for future investigations.[*] The preliminary assessment of our LLM evaluators indicates that they significantly surpass prompting-based LLMs in evaluation abilities, demonstrating 74.4% accuracy.

Our evaluation experiments are conducted on various retrievers, including BGE-M3 (Chen et al., 2024b), BGE-large-zh (Xiao et al., 2023a), GTE-Qwen2-1.5b (Li et al., 2023), and jina-zh (Günther et al., 2023), and diverse open-resource LLMs, *i.e.*, Qwen2.5-72B-Instruct (Team, 2024), Llama3.1-70B-Instruct (Dubey et al., 2024), Deepseek-v2-chat (DeepSeek-AI, 2024), and Yi15-34B (Young et al., 2024). The experimental results reveal that RAG performance varies across different topics and tasks. Moreover, there remains a large space to improve RAG systems in vertical domains.

2 Related Work

2.1 RAG Benchmarks

With the rapid development of RAG investigation, existing QA datasets and evaluation metrics are limited to providing advanced evaluation re-

[*]Note that the automatically generated examples are extensible by prompting GPT-4 (OpenAI, 2023), we currently provide this amount of examples due to the limited budgets.

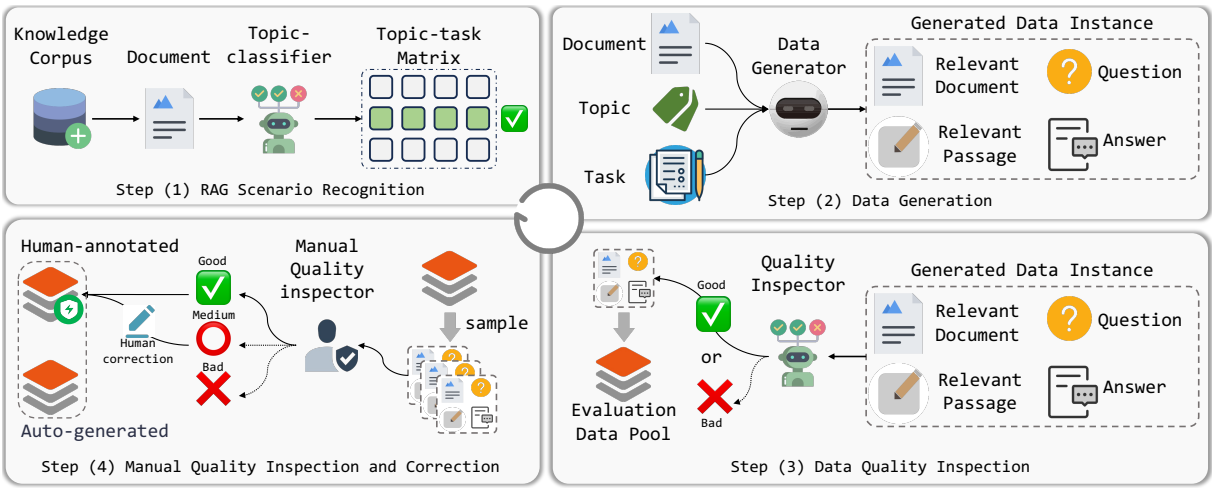

Figure 2: The visualization of the multi-dimensional and auto-chained data generation pipeline.

Datasource	Data Type	Doc Number	Length Sum
BSCF-DB	DB - JSON	193,774	23,631,875
BSCF-PDF	PDF - TXT	3,082	10,587,648
FinGLM	PDF - TXT	55,595	97,296,690
Wiki-Fin	JSON	3,367	5,679,758
BAAI-Fin	JSON	48,124	70,014,858
Official Web	JSON	58,616	45,837,298

Table 2: Statistics of our data sources. "Doc" and "Sum" are short for "Document" and "Summation".

sults. Therefore, various researchers (Chen et al., 2024c; Liu et al., 2023; Xiong et al., 2024; Saad-Falcon et al., 2024; Yu et al., 2024; Lyu et al., 2024; Wang et al., 2024a) concentrate on building comprehensive and reliable RAG benchmarks. The early study, RGB (Chen et al., 2024c), focuses on the advanced abilities of RAG models, such as noise robustness and information integration. ARES (Saad-Falcon et al., 2024) automatically builds a RAG benchmark with the support of LLMs, including automatically generating data instances and automatically judging responses. Beyond open-domain QA, some studies (Xiong et al., 2024; Wang et al., 2024a) also constructed domain-specific RAG benchmarks to evaluate the abilities of RAG systems in vertical domains.

2.2 LLM Evaluation in Financial Domains

In practice, finance is one of the most widespread vertical domains, comprising a wealth of professional knowledge. Therefore, evaluating LLMs in the financial domain is critical for assessing their expertise in vertical domains. Some studies (Shah et al., 2022; Xie et al., 2023, 2024; Li et al., 2024; Chen et al., 2023) collect existing financial QA datasets (Thakur et al., 2021; Sinha and Khandait, 2020; Salinas Alvarado et al., 2015; Chen et al., 2021, 2022; Soun et al., 2022) to build benchmarks, thereby assessing LLMs' understanding of financial knowledge. Recently, Xie et al. (2023) further develops instruction-tuning financial benchmarks by writing instructions for various financial tasks. Beyond assessing LLMs alone, AlphaFin (Li et al., 2024) also introduces RAG tasks to judge RAG models on financial scenarios. However, it primarily focuses on the quality of final responses, neglecting the retrieval performance. In this paper, we construct an omnidirectional and automatic RAG evaluation benchmark that automatically generates evaluation datasets and omnidirectionally assesses RAG systems, leading to comprehensive profiles for them. We compare our benchmark to existing financial LLM benchmarks in Table 1 to demonstrate our advantages.

3 Construction Pipeline of OmniEval

We introduce the construction pipeline of our benchmark alongside the following steps: First, we demonstrate the collection of knowledge corpus in Section 3.1. Next, the generation of evaluation instances is illustrated in Section 3.2. Finally, in Section 3.3, we introduce the evaluation of RAG models. The details are demonstrated below.

3.1 Construction of Knowledge Corpus

To build a wide coverage and diverse financial document corpus, we collect our knowledge corpus from various data sources, including two open-source financial challenges, BS Challenge Financial (BSCF for short) and FinGLM; finance-related web pages from wikipedia-zh; open-source finan-

cial pretraining dataset; BAAI IndustryCorpus Finance (zh) (BAAI-Fin for short); and crawled financial web pages from the official Chinese agency websites. Since these external documents have various formats, such as PDF and SQLite, we use LlamaIndex*, which is compatible with various data formats, to build our retrieval corpus. Specifically, we first transfer SQLite data to the JSON format, then utilize the LlamaIndex toolkit to split all documents into passages with the length set as 2048 and the overlap as 256. The statistical information of our data resources is shown in Table 2, where "document" denotes the LlamaIndex node.

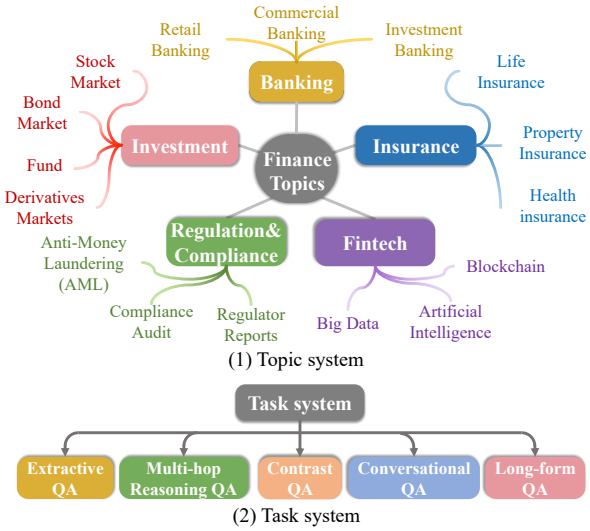

Figure 3: Topic & task systems of our benchmark.

3.2 Generation of Evaluation Instances

Given the knowledge corpus with abundant domain-specific information, we devise a multi-dimensional and auto-chained data generation pipeline (MADGEP), which enables it to adapt well to the constantly updated corpus.

RAG Scenario Recognition To construct matrix-based RAG evaluation scenarios that reflect real-world RAG applications, we classify our RAG evaluation scenarios from two orthogonal perspectives: domain topics and RAG tasks.

From the topic perspective, we categorize RAG scenarios by domain topics related to user queries, such as the stock market and investment banks. Our topic system is initially generated from GPT-4, and we subsequently prune it according to the topic frequency. This approach enables seamless adaptation of our data generation method to other domains, significantly improving its versatility. From the task perspective, we adopt five common and important RAG tasks, following existing studies (Wang et al., 2024a): Extractive QA: Answers to queries can be extracted from the relevant documents without additional reasoning. Multi-hop reasoning QA: It requires multi-hop reasoning as answers are not explicitly stated in external documents. Contrast QA: It involves comparing two objects, requiring multi-aspect external knowledge to produce the final answer. Long-form QA: The queries demand detailed and comprehensive answers, which are usually long-form. Conversational QA: Answering the current question needs to consider the context of conversation histories.

The Cartesian product of these two perspectives forms an RAG scenario matrix, where each element represents a specific topic-task scenario. The topic and task systems are presented in Figure 3. With the pre-defined topic-task matrix (T^2M), we develop a *topic classifier* powered by GPT-4. This classifier receives a sampled document from the knowledge corpus and then classifies the most relevant domain topic. This process locates a specific "row" in T^2M. Subsequently, given the document and its topic, we will traverse all pre-defined RAG tasks to generate associated data instances for each RAG scenario within T^2M elements. The generation approaches are demonstrated below.

Data Generation Leveraging LLMs for automatic data generation has been proven to be effective and reliable, significantly reducing the cost of human annotation (Tan et al., 2024). In this context, we utilize GPT-4 to build a *data generator*, thereby automatically creating data instances for our various RAG scenarios. Specifically, given a document, its domain topic, and a task description, we input these into the data generator to synthesize a QA pair. This pair is required to align with the task requirements and remain relevant to the topic. The input document is viewed as the relevant document for this QA pair. Additionally, to address the challenge of lengthy documents with noisy information, we instruct the generator to extract the most relevant passage within the document, hence precisely locating the valuable content. Finally, each sample comprises a user question, its answer, the relevant document, and a relevant passage.

Data Quality Inspection To ensure the generation quality, we develop a *quality inspector* to filter out low-quality examples. The rationale behind this approach is that judging the instance quality is

*https://www.llamaindex.ai/

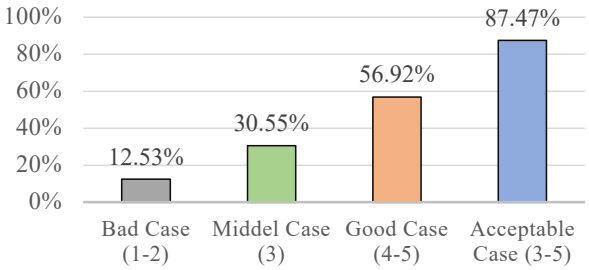

Figure 4: Statistical information of manual inspection.

Setting	Base Model	κ	Accuracy
Prompting	Llama3.1-8B-Inst	39.70	55.60
Prompting	Llama3.1-70B-Inst	54.14	66.40
Prompting	Qwen2.5-7B-Inst	48.05	62.00
Prompting	Qwen2.5-32B-Inst	61.44	71.60
Prompting	Qwen2.5-72B-Inst	55.38	67.20
Lora-FT	Llama3.1-8B-Inst	48.63	62.80
Lora-FT	Qwen2.5-7B-Inst	**64.86**	**74.40**

Table 3: Experimental results of model-based evaluator.

generally easier than generating high-quality data from scratch. Therefore, the inspection process could potentially improve the quality of the filtered dataset. This inspector treats the generated instance as input and predicts whether it contains meaningful information and meets the task requirement. We only retain the instances that the quality inspector identifies as high-quality ones.

All used GPT prompts are shown in Appendix D.

Manual Quality Inspection and Correction Besides automatic quality inspection, we employ annotators to perform data quality inspection and correction, leading to a high-quality evaluation dataset and enhanced reliability of our benchmark.

We first sample a subset from generated instances for each T^2M element. Annotators are then requested to check the following aspects of the data: Does the generated question meet the *task requirements*? Is the question *related to the given topic*? Is the question *semantically complete*? Is the *answer correct and complete*?. Are the *extracted passages accurate and complete*? The annotation follows a five-point scale from 1 to 5, where 1 and 2 indicate low data quality, suggesting that the instance should be discarded; 3 signifies the data contains some human-fixable defects; and 4 or 5 denotes good to excellent data quality. The number of labeled data instances is 910.

We present the statistical results of the inspection in Figure 4. The findings reveal that the acceptance rate of our auto-generated cases is 87.47%, potentially confirming the effectiveness and usability of MADGEP. Annotators are also tasked with correcting instances labeled as 3 to create high-quality human-annotated data. Through these inspection and correction steps, we establish a reliable human-annotated dataset, significantly enhancing the robustness of our benchmark. Finally, we create two datasets: one auto-generated and the other human-annotated. We further split the auto-generated ones into train and test datasets to facilitate related investigations based on our benchmark.

The data amounts of these datasets are shown in Appendix A and the instructions we used for GPT-4 and annotators are shown in Appendix D.

3.3 Evaluation of RAG Models

To comprehensively and accurately assess RAG baselines, we integrate two types of metrics: rule-based metrics and model-based metrics.

Rule-based Metrics Given the widespread usage and stability of rule-based metrics, we use Rouge-L and F1 to provide basic evaluations for generated responses.[*] We also adopt ranking metrics, MAP and MRR, to assess the performance of retrievers within RAG systems. This combination facilitates a holistic evaluation of the entire RAG pipeline. There calculations are shown in Appendix B.

Model-based Metrics Given the flexibility and diversity of AI chatbot responses, rule-based metrics often struggle to provide semantic evaluations. To solve it, we devise five high-level metrics implemented based on fine-tuned LLMs:

Accuracy (ACC). LLMs often generate responses that are correct in content but poorly matched in wording. Therefore, we introduce a model-based 3-point accuracy metric in semantics: 1 = poor quality; 2 = average quality; 3 = good quality.

Completeness (COM). Long-form QA usually requires LLM to provide comprehensive answers that address various aspects of the question (Wang et al., 2024b). To assess completeness, we use a four-point metric: 1 = no relevant aspects addressed; 2 = partially addressed; 3 = comprehensively addressed; and -1 = completeness measurement is not applicable for the input QA scenario.

Hallucination (HAL). It assesses hallucinations in generated responses: HAL is 0 if the response is correct, or incorrect but derived from retrieved

[*] https://pypi.org/project/rouge-chinese/

Models	MAP ↑	MRR ↑	Rouge-L ↑	F1 ↑	ACC ↑	HAL ↓	COM ↑	UTL ↑	NAC ↑
Auto-generated evaluation set									
Jina-zh	0.3395	0.3469	0.1662	0.2553	0.3908	0.0794	0.5981	0.5078	0.2837
BGE-large-zh	0.3777	0.3865	0.1693	0.2541	0.4080	0.0597	0.6048	0.5194	0.3124
BGE-M3	0.3961	0.4057	0.1746	**0.2593**	0.4091	0.0634	0.6092	0.5203	0.3060
GTE-Qwen2-1.5B	**0.4370**	**0.4491**	**0.1778**	0.2563	**0.4326**	**0.0467**	**0.6256**	**0.5613**	**0.3293**
Human-annotated evaluation set									
Jina-zh	0.3458	0.3533	0.2341	0.3821	0.4089	0.0886	0.5930	0.5163	0.3073
BGE-large-zh	0.4153	0.4252	0.2435	0.3870	0.4325	0.0718	**0.6224**	0.5367	0.3545
BGE-M3	0.4152	0.4236	0.2517	0.3913	0.4450	0.0709	0.6208	0.5410	0.3472
GTE-Qwen2-1.5B	**0.4443**	**0.4574**	**0.2528**	**0.3919**	**0.4476**	**0.0618**	0.6190	**0.5576**	**0.3595**

Table 4: The overall results of retrieval models with the generator being set as Qwen2.5-72B.

Retriever	Generator	Rouge-L ↑	F1 ↑	ACC ↑	HAL ↓	COM ↑	UTL ↑	NAC ↑
Auto-generated evaluation set								
Close-Book	Yi15-34B	0.0326	0.0673	0.1573	-	0.5063	-	0.0693
Close-Book	Deepseek-v2-chat	0.1861	0.3709	0.3587	-	0.5755	-	0.1121
Close-Book	Qwen2.5-72B	0.1607	0.3222	0.3788	-	0.6017	-	0.1256
Close-Book	Llama3.1-70B-Instruct	0.1993	0.3989	0.3238	-	0.5284	-	0.0677
GTE-Qwen2-1.5B	Yi15-34B	0.0593	0.0958	0.3402	0.0597	0.5778	0.4229	0.1682
GTE-Qwen2-1.5B	Deepseek-v2-chat	0.2279	0.3300	0.4099	0.0634	0.6072	0.5197	0.3175
GTE-Qwen2-1.5B	Qwen2.5-72B	0.1778	0.2563	0.4326	**0.0467**	**0.6256**	**0.5613**	**0.3293**
GTE-Qwen2-1.5B	Llama3.1-70B-Instruct	**0.3235**	**0.4810**	**0.4398**	0.0792	0.5926	0.4754	0.3088
Human-annotated evaluation set								
Close-Book	Yi15-34B	0.0497	0.1161	0.1461	-	0.4987	-	0.0749
Close-Book	Deepseek-v2-chat	0.2250	0.4353	0.3306	-	0.5541	-	0.1153
Close-Book	Qwen2.5-72B	0.2082	0.4191	0.3405	-	0.5754	-	0.1241
Close-Book	Llama3.1-70B-Instruct	0.2195	0.4183	0.2859	-	0.5133	-	0.0659
GTE-Qwen2-1.5B	Yi15-34B	0.0887	0.1583	0.3366	0.0648	0.5821	0.4234	0.1856
GTE-Qwen2-1.5B	Deepseek-v2-chat	0.2916	0.4353	0.4234	0.0750	0.6006	0.5160	0.3213
GTE-Qwen2-1.5B	Qwen2.5-72B	0.2528	0.3919	**0.4476**	**0.0618**	**0.6190**	**0.5576**	**0.3595**
GTE-Qwen2-1.5B	Llama3.1-70B-Instruct	**0.3390**	**0.5042**	0.4433	0.1131	0.5745	0.4764	0.3268

Table 5: The overall evaluation results on final responses of RAG models.

documents; HAL is 1 if the response is incorrect and unsupported; and HAL is -1 if not applicable.

Utilization (UTL). Assesses whether the LLM effectively uses retrieved documents and if the answer is traceable. Its scale is similar to ACC.

Numerical accuracy (NAC). For queries requiring financial calculations, NAC = 1 for correct answers, 0 for incorrect answers, and -1 for non-numerical responses.

Finally, all metrics are normalized into [0,1], and samples evaluated as -1 will not be considered for the specific metrics. For our evaluation, each matrix item has its own independent subset, and every question is evaluated using all seven metrics.

SFT of LLM evaluator To ensure the reliability of our LLM evaluator, we conduct human annotation on a subset of generated responses for the five metrics, creating a labeled dataset for training stable evaluators. With the high cost of manual annotation, we randomly sample 127 cases and produce 635 examples by aggregating all five metrics. We divide it into training, validation, and test sets in a ratio of 5:1:4.

Leveraging the robust zero-shot capabilities of LLMs, which are presented in the second to sixth lines in Table 3), we achieve promising performance of our evaluator by few-shot Lora fine-tuning, even with limited training data. It proves both the annotation quality and reliability of our LLM-based evaluators. Finally, we build our evaluator as the fine-tuned Qwen-2.5-7B-Instruct with the best performance.

4 Experiment

We conduct our experiments on various open-resource retrievers and LLMs. For **retrievers**, we select GTE-Qwen2-1.5B (Li et al., 2023), BGE-large-zh (Xiao et al., 2023b), BGE-M3 (Xiao et al., 2023b), and Jina-zh (Mohr et al., 2024).

For **LLMs**, we ues Qwen2.5-72B-Instruct (Team, 2024), Deepseek-v2-chat (DeepSeek-AI, 2024), Yi15-34b (Young et al., 2024), and Llama3.1-70B-Instruct (Dubey et al., 2024). We set the retrieved document number as 5 to ensure a fair comparison.

4.1 Comparison Experiments of Retrievers

Our experiments aim to assess the entire pipeline of RAG systems, including both retrievers and generators (LLMs). First, we present the experimental results on retrievers using our two evaluation datasets, the auto-generated set and the human-annotated set, with the generator set as Qwen2.5-72B.

The main results are displayed in Table 4. According to the results shown, GTE-Qwen2-1.5B demonstrates the best retrieval performance across most retrieval and generation metrics. We attribute this superiority to two factors: (1) Model parameters: GTE-Qwen2-1.5B encompasses the most model parameters among all baselines, significantly enhancing its performance upper bound. (2) Fine-tuning from LLM: It is continuously fine-tuned from the LLM, Qwen2-1.5B, which is pre-trained using a large-scale corpus. This strategy equips it with extensive world knowledge, providing better prior knowledge compared to retrievers that are pre-trained from scratch.

4.2 Comparison Experiments of Generators

We then evaluate the generators' abilities to solve expert problems. Given the superiority of GTE-Qwen2-1.5B in the retrieval task, we choose it as our retriever and compare the response quality of LLMs. The main results are presented in Table 5. "Close-Book" indicates that responses are generated solely by LLMs without incorporating retrieved external knowledge. Since HAL and UTL metrics are conditioned on the retrieved results, there are no corresponding results in close-book settings.

Based on these, we conclude the following findings: (1) We notice that LLMs typically yield better results when equipped with retrievers compared to close-book settings. It proves that in domain-specific scenarios, it is essential for LLMs to retrieve external expert knowledge, thereby enhancing the reliability of generated responses. (2) There remains significant potential for existing retrievers and LLMs to enhance RAG abilities in financial domains. Even with the RAG systems, performance is still lacking across all retriever and LLM configurations. This indicates the difficulty of our evaluation datasets, which involve expert and reasoning

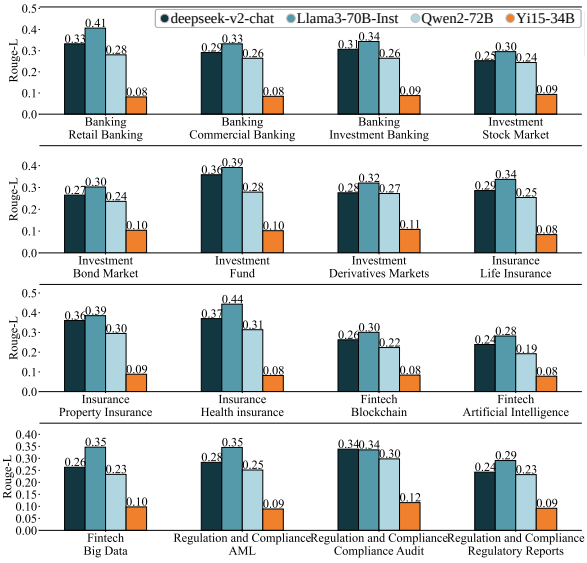

Figure 5: Rouge-L scores of generators on topic-specific human-annotated subsets.

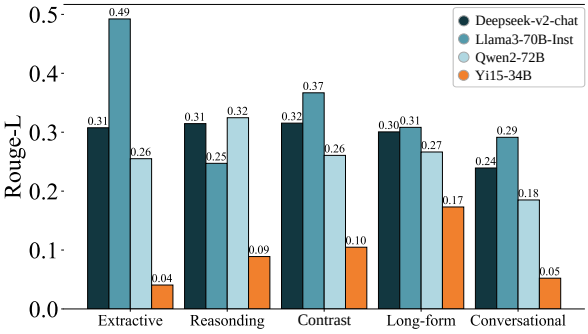

Figure 6: Rouge-L scores of generators on task-specific human-annotated subsets.

financial tasks. Additionally, it confirms that our benchmark introduces new challenges for existing RAG systems, potentially driving further investigation into RAG models in domain-specific scenarios.

4.3 Topic and Task-specific Experiments

Utilizing our T^2M-based evaluation subsets, we further compare RAG models across different task and topic evaluation sets, assessing their abilities on different evaluation views. The results are illustrated in Figures 5 and 6. Due to limited space, we present the topic-specific results on auto-generated sets in Appendix C, *i.e.*, Figures 8 and 9.

We notice that the same RAG model exhibits varying performance across different task or topic scenarios, indicating an imbalance in their capabilities to solve different query scenarios. We analyze the main reasons as three-fold: (1) The availability of accessible documents varies across topics, leading to significant distribution differences in the pre-trained corpora of LLMs. This uneven expo-

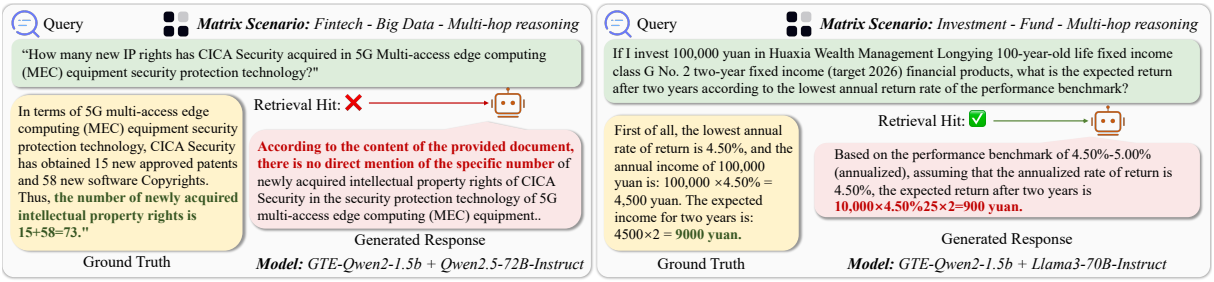

Figure 7: Failure cases of evaluated RAG models.

sure results in varying performance across different topics; (2) different LLMs are trained on distinct corpora, which influences their domain-specific competencies and leads to varying performance across models for the same task; and (3) tasks inherently differ in difficulty. For instance, extractive QA primarily requires RAG models to retrieve relevant documents and extract correct answers, whereas multi-hop reasoning tasks demand both precise retrieval and strong reasoning abilities to navigate complex questions.

4.4 Matrix-based Visualization of Results

As we mentioned earlier, our matrix-based evaluation scenarios offer a comprehensive ability profile for the evaluated RAG model, distinctly revealing their performance on specific topic-task scenarios. Accordingly, we present a representative matrix-based visualization of GTE-Qwen2-1.5B+Llama3.1-70B-Instruct on human-annotated subsets, which is shown in Figure 1. Due to limited spaces, we show matrix-based results of other models in Appendix C, *i.e.*, Figures 13, 14, 15, 16 17, 18, and 19.

This method demonstrates the abilities of RAG models more clearly than simply averaging all results, allowing for more detailed and fine-grained analyses. For example, in Figure 1, which presents the results of GTE-Qwen2-1.5B+Deepseek-v2, it is evident that this RAG model excels in the extractive QA task with the "Fund"-related topic. However, there remains significant room for improvement in the conversational QA task with the "AI"-related topic. This visualization provides a novel approach to analyzing RAG performance in different scenarios, allowing targeted strategies to address localized limitations of RAG models.

4.5 Case Analyses

To prove the reliability of our benchmark, we further visualize some failure cases of evaluated RAG models in Figure 7. The first case highlights a scenario where the retriever fails to retrieve relevant information, likely due to the long-tail nature of the question topic. As a result, the LLM lacks the necessary expert knowledge to provide an accurate response. This underscores the critical role of high-quality, domain-specific retrievers for effective in-domain RAG applications. The second case demonstrates a scenario where the retriever successfully retrieves relevant content, yet the generator fails to correctly perform the necessary financial calculations. This highlights the difficulty of our evaluation dataset, which requires RAG models to possess not only retrieval accuracy but also strong reasoning and numerical computation capabilities to handle complex financial queries.

5 Conclusion

In this study, we propose an automatic and omnidirectional RAG benchmark in a vertical domain *i.e.*, finance. We first identify diverse query scenarios via a matrix-based method, which considers two orthogonal perspectives, topics, and tasks. This approach allow us to assess RAG systems comprehensively and finely by simulating diverse practical RAG scenarios. We develop an auto-chained generative assessment pipeline to create our evaluation datasets. Through rigorous model-based and manual quality inspections, we derive three datasets: an auto-generated training set, an auto-generated test set, and a human-annotated test set. The high acceptance of auto-generated data confirms the reliability of our data generation methods. Our experimental results illustrate that there is still a significant improvement space for existing RAG models in vertical domains. In addition, RAG systems exhibit varying performance across diverse query scenarios, highlighting new challenges and investigation directions for RAG studies.

Limitations

In this study, we develop an omnidirectional and automated RAG benchmark specifically tailored for the finance domain. Our benchmark is featured by its matrix-based RAG evaluation scenarios, multi-dimensional data generation approaches that combine automatic and manual methods, a multi-stage evaluation pipeline, and a multi-dimensional evaluation system. However, we acknowledge several limitations that warrant further investigation:

First, despite our efforts to collect a diverse data corpus, the distribution remains somewhat limited. This limitation arises primarily from challenges related to accessibility and the open licensing of data resources. As a result, there is a risk of introducing potential biases into our datasets, which could affect the generalizability of our benchmark findings. However, considering OmniEval is designed to be highly flexible, allowing seamless expansion of the knowledge corpus and the generation of additional evaluation data sets. We'd like to evolve our benchmark over time, further enhancing its generalizability and robustness. Second, we recognize that the costs associated with human annotation have led to a limited amount of collected human evaluation data for training our LLM evaluators, which may impact the performance of LLM evaluators. In future studies, we plan to gather a more extensive set of human evaluation data. This enhancement aims to boost the accuracy and reliability of our LLM evaluators, ultimately leading to a more effective benchmark.

Ethical Statements

In this paper, we collect our document corpus from various sources, where BSCF, FinGLM, wikipedia-zh, and BAAI-Fin are publicly available. Note that the financial web pages are crawled from the official agency websites and have passed the judgment of legal personnel.

References

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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6025–6047. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024c. Benchmarking large language models in retrieval-augmented generation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press.

Wei Chen, Qiushi Wang, Zefei Long, Xianyin Zhang, Zhongtian Lu, Bingxuan Li, Siyuan Wang, Jiarong Xu, Xiang Bai, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-finllm: A chinese financial large language model based on multiple experts fine-tuning. *CoRR*, abs/2310.15205.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data. *Proceedings of EMNLP 2021*.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *Proceedings of EMNLP 2022*.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi,

Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *CoRR*, abs/2310.19923.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *CoRR*, abs/2311.11944.

Xiang Li, Zhenyu Li, Chen Shi, Yong Xu, Qing Du, Mingkui Tan, and Jun Huang. 2024. Alphafin: Benchmarking financial analysis with retrieval-augmented stock-chain framework. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 773–783. ELRA and ICCL.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. RECALL: A benchmark for llms robustness against external counterfactual knowledge. *CoRR*, abs/2311.08147.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2024. CRUD-RAG: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *CoRR*, abs/2401.17043.

Isabelle Mohr, Markus Krimmel, Saba Sturua, Mohammad Kalim Akram, Andreas Koukounas, Michael Günther, Georgios Mastrapas, Vinit Ravishankar, Joan Fontanals Martínez, Feng Wang, et al. 2024. Multi-task contrastive learning for 8192-token bilingual text embeddings. *arXiv preprint arXiv:2402.17016*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: an automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 338–354. Association for Computational Linguistics.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.

Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. WHEN FLUE MEETS FLANG: benchmarks and large pre-trained language model for financial domain. *CoRR*, abs/2211.00083.

Ankur Sinha and Tanmay Khandait. 2020. Impact of news on the commodity market: Dataset and results. *CoRR*, abs/2009.04202.

Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. 2022. Accurate stock movement prediction with self-supervised learning from sparse noisy tweets. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1691–1700. IEEE Computer Society.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Shuting Wang, Jiongnan Liu, Shiren Song, Jiehan Cheng, Yuqi Fu, Peidong Guo, Kun Fang, Yutao Zhu, and Zhicheng Dou. 2024a. Domainrag: A chinese benchmark for evaluating domain-specific retrieval-augmented generation. *CoRR*, abs/2406.05654.

Shuting Wang, Xin Yu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. 2024b. Richrag: Crafting rich responses for multi-faceted queries in retrieval-augmented generation. *CoRR*, abs/2406.12566.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023a. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023b. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. The finben: An holistic financial benchmark for large language models. *CoRR*, abs/2402.12659.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A large language model, instruction data and evaluation benchmark for finance. *CoRR*, abs/2306.05443.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6233–6251. Association for Computational Linguistics.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *CoRR*, abs/2403.04652.

Xiaodong Yu, Hao Cheng, Xiaodong Liu, Dan Roth, and Jianfeng Gao. 2024. Reeval: Automatic hallucination evaluation for retrieval-augmented large language models via transferable adversarial attacks. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1333–1351. Association for Computational Linguistics.

A Statistical Information of Our datasets

In this section, we provide the detailed statistical information of our three datasets, including auto-generated training set, auto-generated test set, and human-annotated test set, in Figure 10, 11, and 12.

B Calculation of Rule-based Evaluation

We provide detailed calculation functions for our utilized rule-based metrics in this section, including

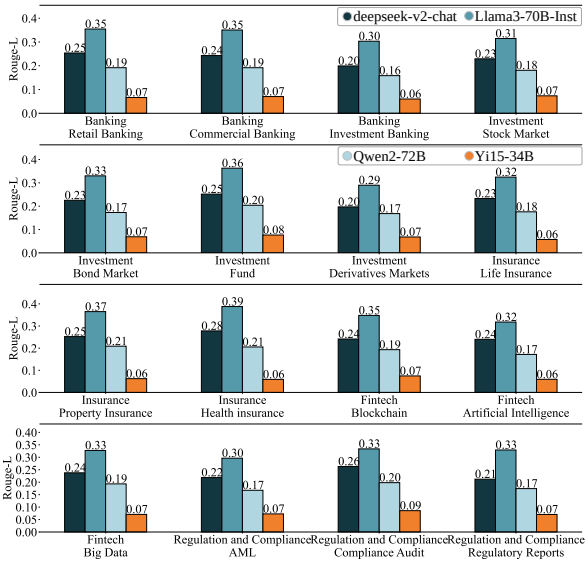

Figure 8: Rouge-L scores of generators on topic-specific auto-generated subsets.

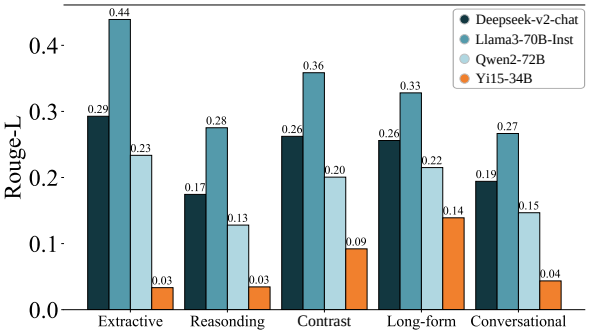

Figure 9: Rouge-L scores of generators on task-specific auto-generated subsets.

Rouge-L, F1, MAP, and MRR. The calculation of Rouge-L (F1 setting) is presented below:

$$\text{Rouge} - \text{L} = \frac{2 * R_{lcs} * P_{lcs}}{R_{lcs} + P_{lcs}}, \quad (1)$$

$$R_{lcs} = \frac{\text{LCS}(X, Y)}{\text{Length}(Y)}, \quad (2)$$

$$P_{lcs} = \frac{\text{LCS}(X, Y)}{\text{Length}(X)}, \quad (3)$$

where X, Y denote the generated and referenced texts, $\text{LCS}()$ is the function to compute the longest common subsequence between two input sentences, and $\text{Length}()$ returns the length of the input sentence. The computation of F1 is shown as follows:

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (6)$$

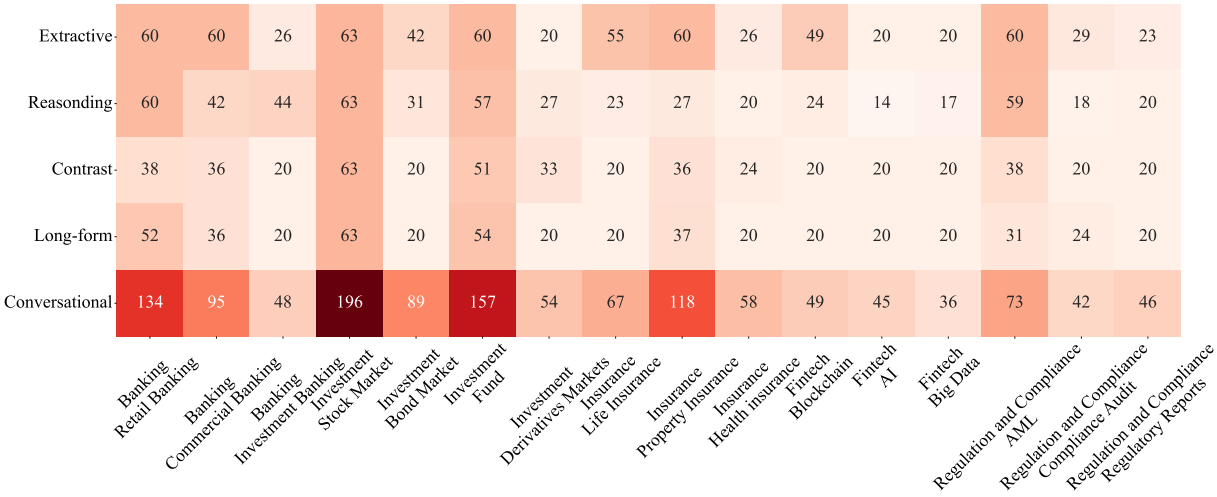

Figure 10: Data amount of the auto-generated training set.

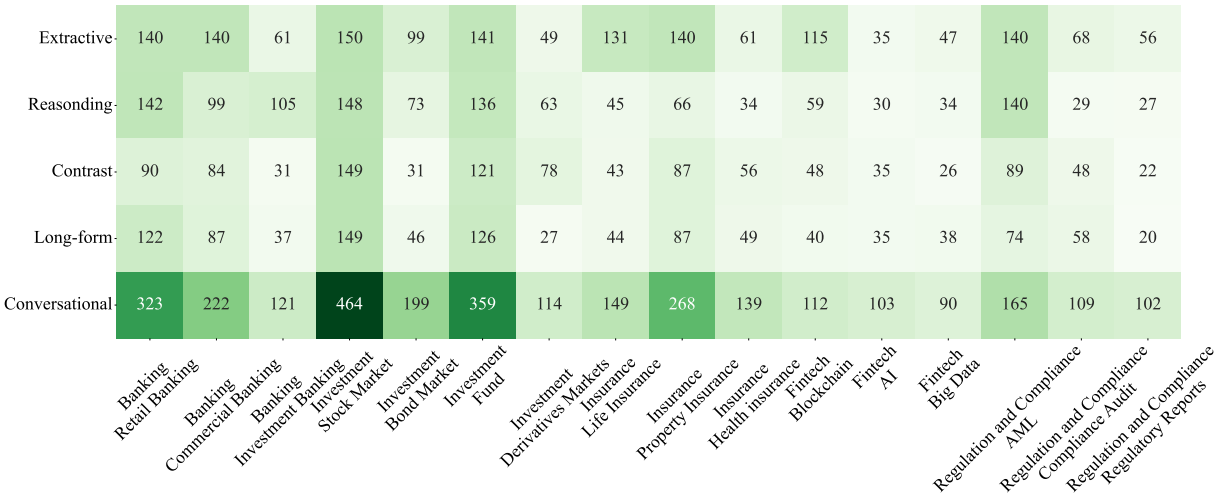

Figure 11: Data amount of the auto-generated test set.

where TP is the number of matched words between the generated response and the golden answer, FP is the number of mismatched words in the generated response, and FN is the number of mismatched words in the golden answer.

MAP and MRR are calculated as follows:

$$\text{MAP} = \frac{\sum_q = 1^Q}{\text{AP@}k_q}, \tag{7}$$

$$\text{AP@}k = \frac{\sum_{i=1}^{k} P(i) * rel(i)}{\text{Number of relevant documents}}, \tag{8}$$

$$\text{MRR} = \frac{\sum_q = 1^Q}{\text{RR}_q}, \tag{9}$$

$$\text{RR} = \frac{1}{\text{FRP}}. \tag{10}$$

where Q is the number of all queries. $P(i)$ indicates the number of relevant documents up to the i-th ranking position and $rel(i)$ denotes the relevance of the i-th ranked document. FRP represents the ranking position of the first relevant document.

C Supplementary Visualization Results

In this section, we present the supplementary matrix-based visualization results of our RAG models in Figures 13, 14, 15, 16, 17, 18, and 19.

D Human and GPT Instructions

In this section, we provide detailed instructions we used for human annotation and GPT generation, including the topic-tree generation (Box 3), automated data generation (Boxs 23, 24, and 25), automated data quality inspection (Box 27), and human annotation and correction (a flow chart, shown in Figure 20). We also show detailed task requirements which support the GPT generation and human annotation in Tables 6 and 7.

Task	Requirement
Extractive QA	This task is designed to evaluate the ability of retrieving enhanced financial large language models to answer one-hop questions. That is, the user's question does not need to do multi-hop thinking, and the answer to the question can be directly found in the search document and extracted as an answer. - Please note the distinction between this task and multi-hop inference problems.
Multi-hop Reasoning	This task aims to evaluate the ability of a retrieve-enhanced financial grand language model to answer questions involving multi-hop reasoning. That is, the answer cannot be found directly in the external document retrieved, and **the model needs to do at least two hops of reasoning** to arrive at the final answer according to the external information provided by the document or its own knowledge. - Do not generate questions that can be answered with one-hop reasoning. - Evaluation data generation to evaluate multi-hop inference capability mainly includes the following two categories: 1. First identify the "entity-relationship" link composed of multiple entities with information progressive relationship in the document, and then generate multi-hop inference data according to the relationship link. That is, there should be at least two unknown information points in the proposed question (**and the unknown information in the middle node is necessary for solving the final question**). To solve the final answer, the LLM to be evaluated needs to perform information retrieval and reasoning on the previously unknown information points to obtain the dependency information for solving the final answer, and then solve the final answer. Trying to satisfy the content of the question is a more obvious need for multi-hop reasoning. 2. If you need to perform financial calculations based on the information provided in the document, ensure that the questions and answers are accurate. - If I provide one piece of document data, generate the second type of multi-hop inference data, which is the problem that requires financial calculation based on the information provided in the document. - If I provide multiple document data, generate the first type of multi-hop inference data. That is to identify the "entity-relationship" link composed of multiple entities with information transfer relationship in the document, and ensure that the "entity-relationship" link is through all the provided documents, and then generate multi-hop inference data according to the relationship link. Please ensure that the generated multi-hop inference problem cannot be solved by only one document content, ensure that all documents provided are valuable for solving the generated inference problem. - Be careful not to directly write out the complete content of each step of information transmission in the question, especially do not say that the middle answer is written in the question, otherwise the multi-hop reasoning problem will degenerate into a one-hop reasoning problem.

Table 6: Requirements of tasks for human and GPT generation – Part 1.

Task	Requirement
Contrast QA	This task is designed to evaluate the ability of a retrieve-enhanced financial large language model to answer questions involving contrast classes. That is, the question involves comparing two aspects of the transaction, and the corresponding answer needs to provide a correct and comprehensive comparison and summary of results. - When I provide multiple document data, please ensure that the generated question-answer data is cross-document, *i.e.*, the need to answer the question requires the help of all the provided document data. Based on only one or a few of them can lead to incomplete answers.
Long-form QA	This task is designed to evaluate the ability to retrieve enhanced financial large language models when answering questions with longer answers. Such as introducing classes and summarizing class problems. - Ensure that the answers to the generated data are comprehensive enough to cover all aspects of the user's questions. - When I provide multiple document data, please ensure that the generated question-answer data is cross-document, *i.e.*, the need to answer the question requires the help of all the provided document data. Based on only one or a few of them can lead to incomplete answers.
Conversation QA	This task is designed to evaluate the ability to retrieve enhanced financial large language models to do multiple rounds of conversations. That is, the generated data should be in the form of multiple rounds of conversations. - Therefore, the document is required to be rich enough in contextual information to support the generation of multiple rounds of conversations. - Take care to ensure the dependency between the generated multiple rounds of dialogue, especially the dependency of the content of the question, that is, the subject of the question in the second and later rounds is missing, or is a pronoun, resulting in ambiguous semantics. Understanding the full intent of subsequent rounds of questions requires a full understanding of what was said in previous rounds. - The generated data should be stored as a JSON list for multiple rounds of Q&A information. - I may provide multiple document data, in this case, please ensure that the generated multi-round conversation data is cross-document and able to use all the content of the provided document.

Table 7: Requirements of tasks for human and GPT generation – Part 2.

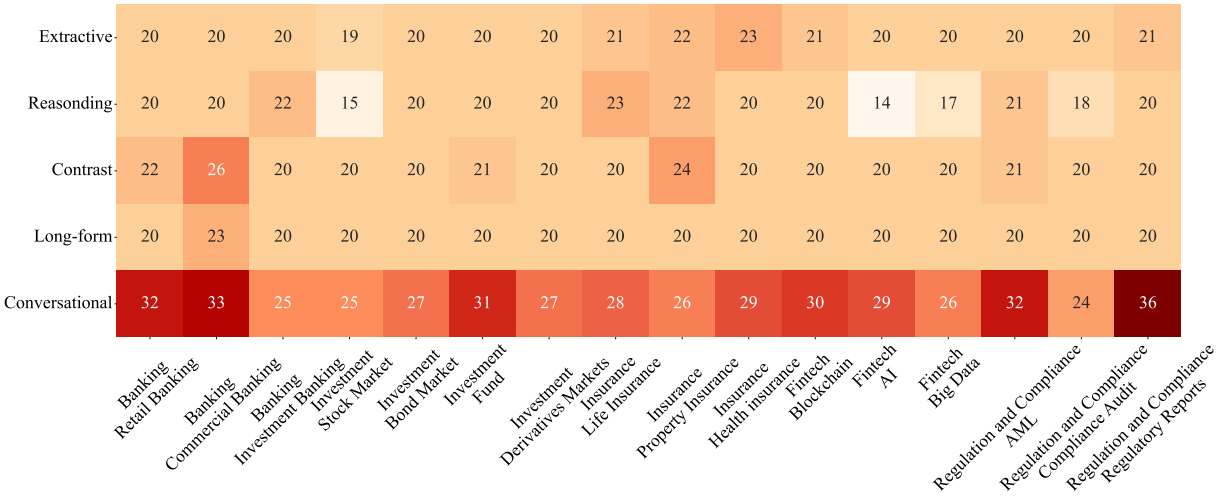

Figure 12: Data amount of the human-annotated test set.

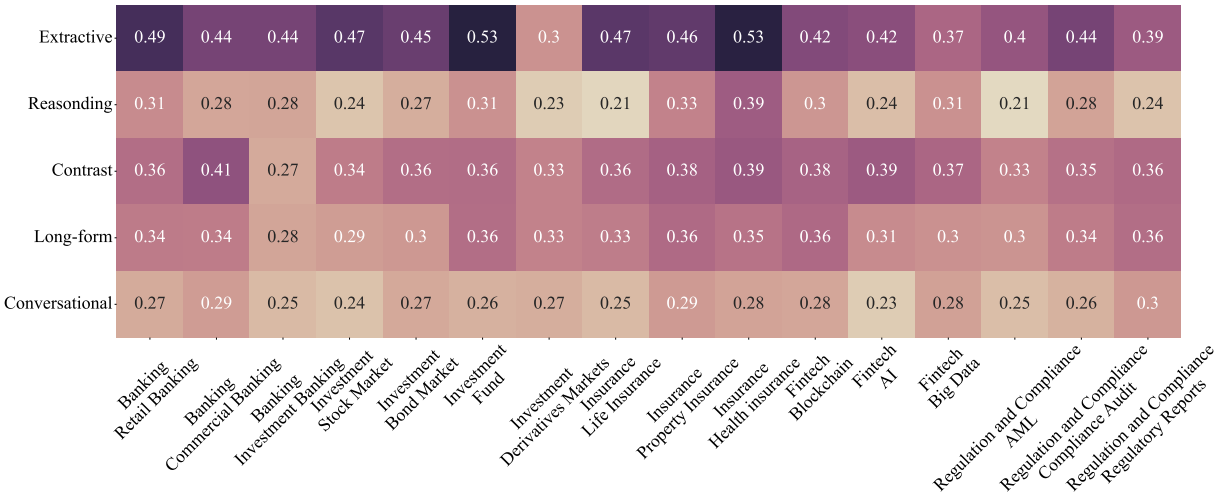

Figure 13: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Qwen2-72b on auto-generated subsets.

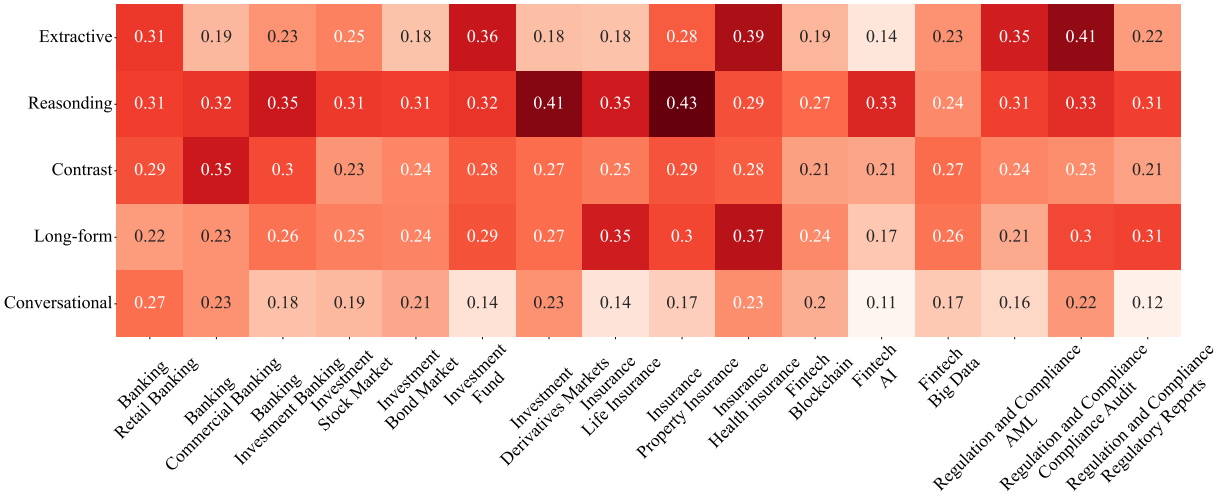

Figure 14: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Qwen2-72b on human-annotated subsets.

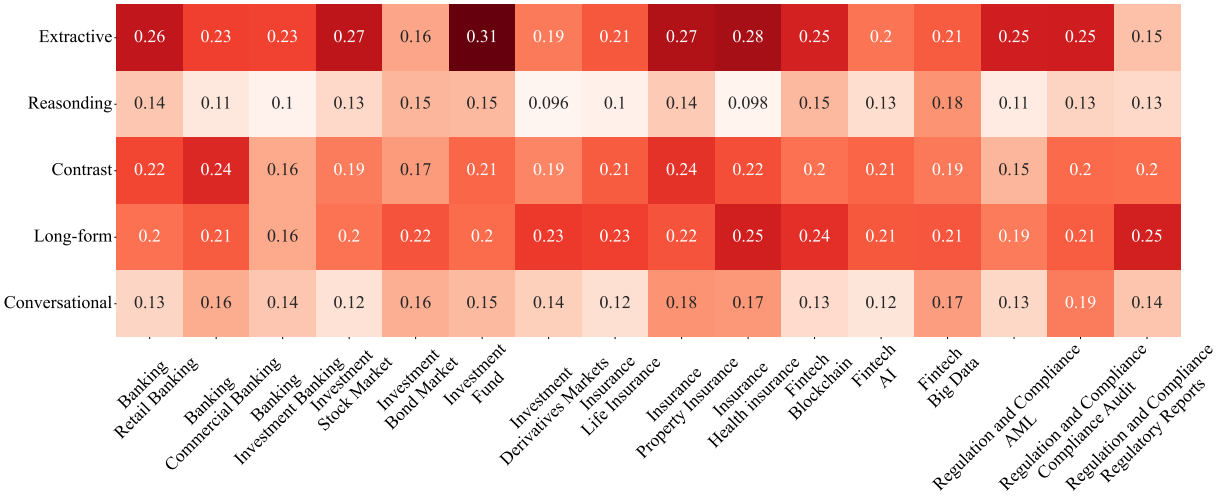

Figure 15: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Qwen2-72b on auto-generated subsets.

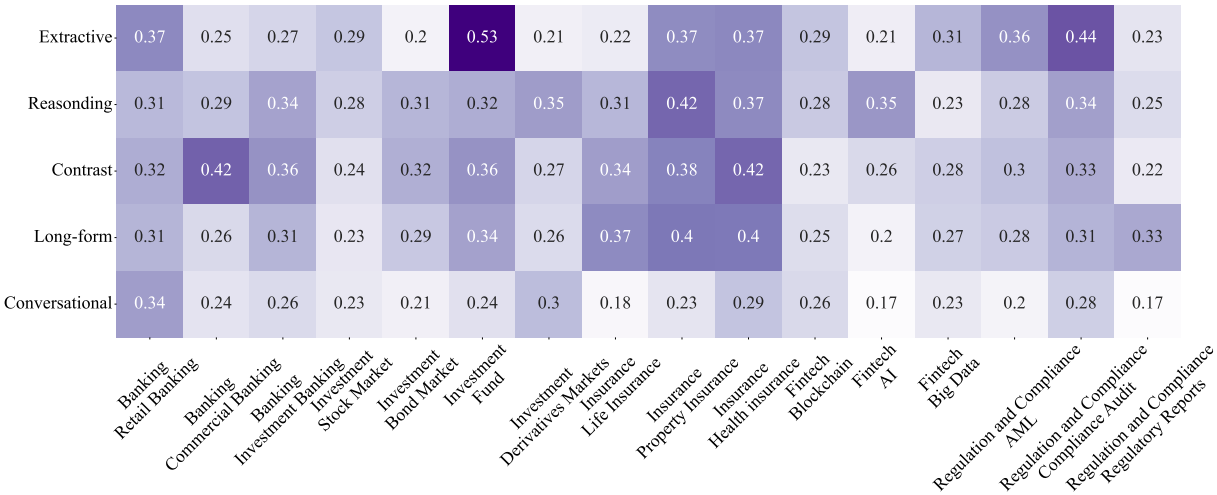

Figure 16: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+deepseek-v2-chat on human-annotated subsets.

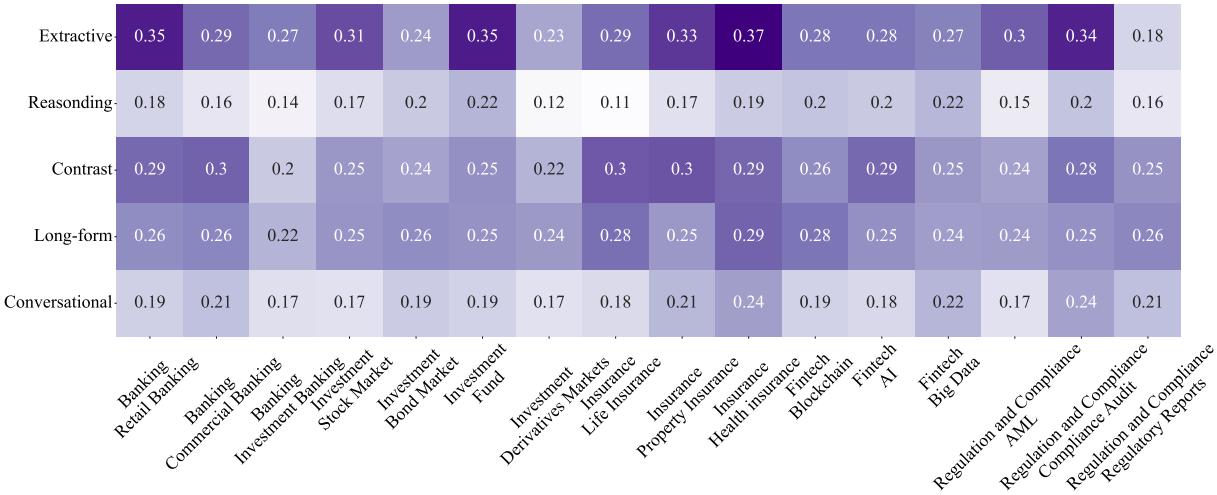

Figure 17: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+deepseek-v2-chat on auto-generated subsets.

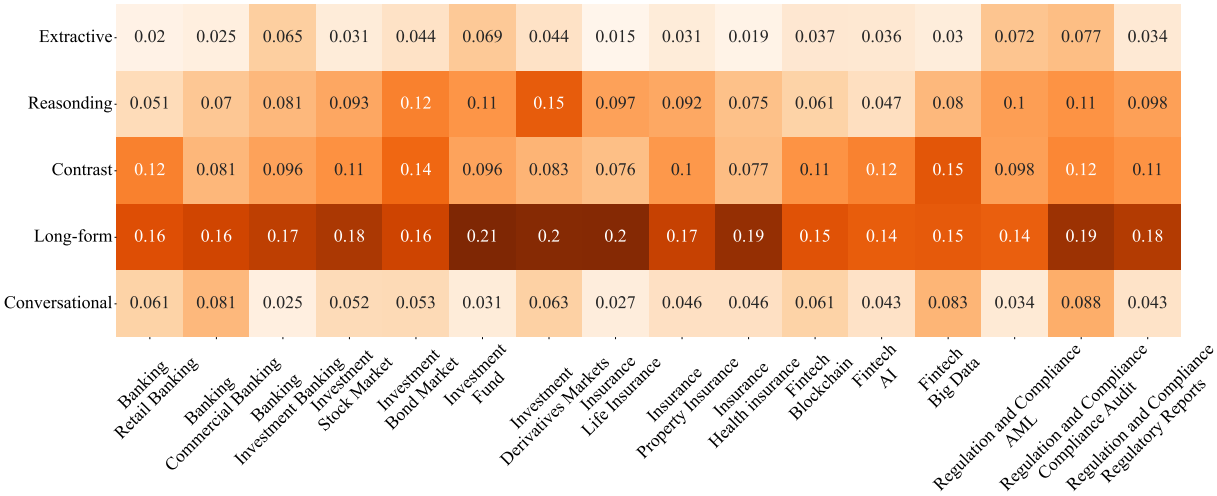

Figure 18: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Yi15-34B on human-annotated subsets.

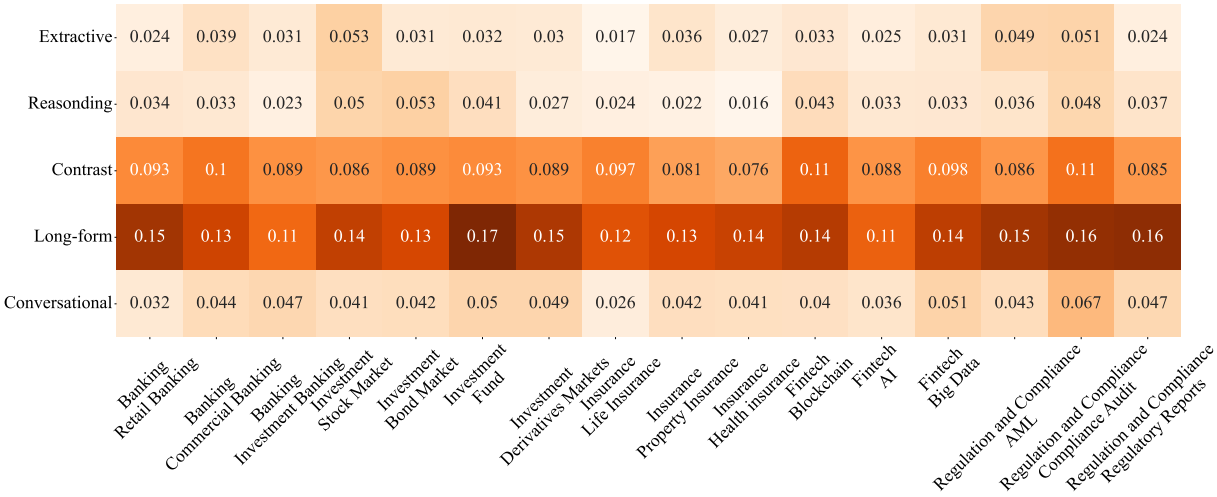

Figure 19: Rouge-L of matrix-based results of GTE-Qwen2-1.5B+Yi15-34B on auto-generated subsets.

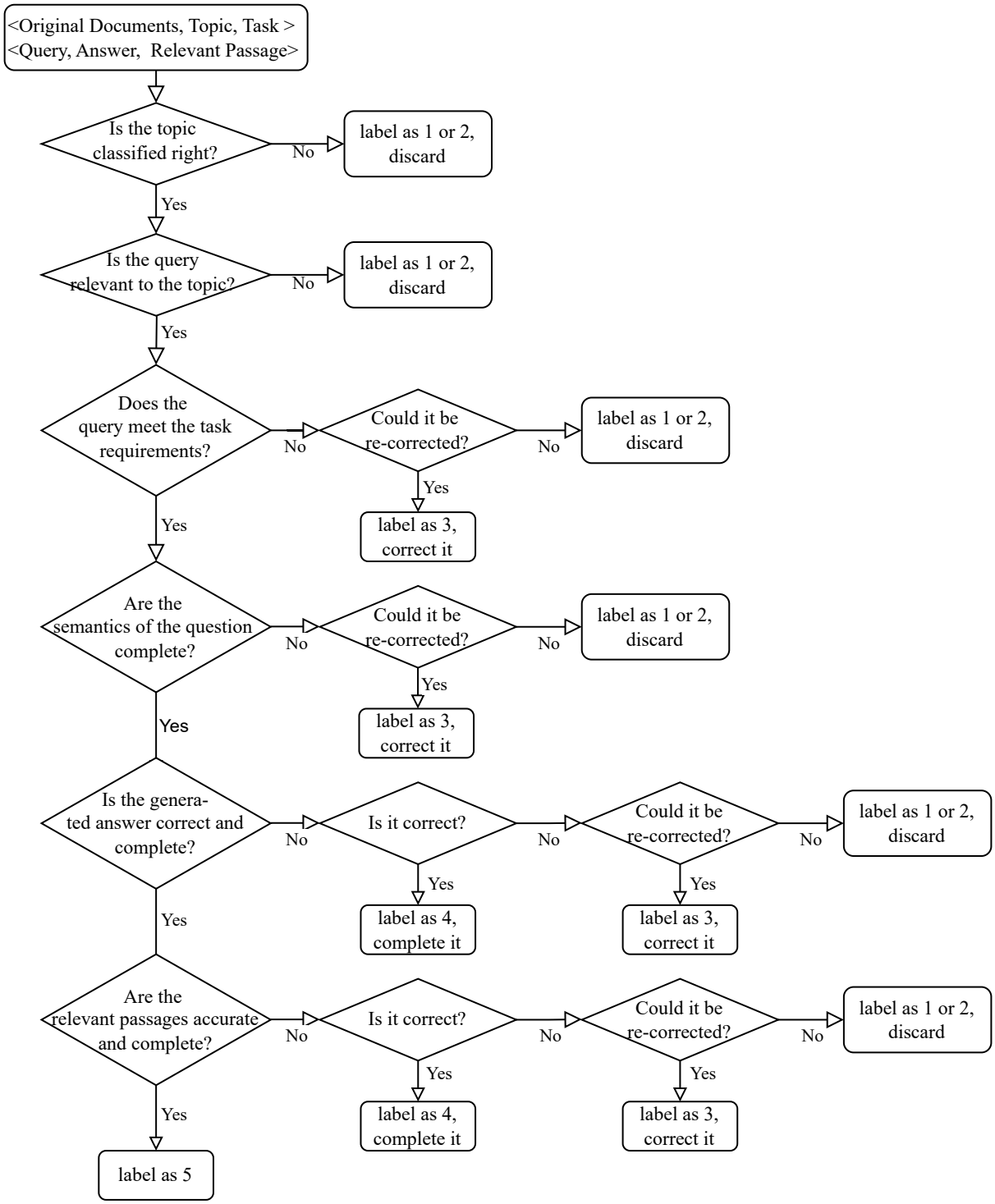

Figure 20: The pipeline of human annotation and correction for automatically generated data instances.

Instructions for GPT-4 to generate a topic tree for the specific domain.

Background
You are a professional domain subcategory tree builder. I will provide you with the name of the root node for the domain type, and you should generate a comprehensive and diverse subcategory tree under that domain.
The output should be returned in JSON format. This JSON should include the following two properties:
- topic_name: Represents the category name of the current tree node.
- sub_topics: Represents the subcategory tree of the current tree node, which is a list of JSON data for that subcategory tree. If the current node is a leaf node (i.e., it has no subcategories), this property will be an empty list.
The data format requirements are as follows:
{
"topic_name": The name of the category for this node,
"sub_topics": A list of JSON data for the subcategory tree under this node, with each item being JSON data of a subtree that also contains the "topic_name" and "sub_topics" properties.
}
Name of the Root Node for the Domain Type
domain_name

Figure 21: Instructions for GPT-4 to generate a topic tree for the specific domain.

Instructions for GPT-4 to classify the domain topic for the input document.

Background
You are an intelligent document topic classification assistant. I am generating retrieval-augmented financial model multi-task evaluation data. This evaluation data is automatically generated by a large language model. I will provide the large language model with the following content: [financial subcategories of interest for the evaluation data, task description for the evaluation, documents in the knowledge base]. I need the large language model to generate: [user questions that align with the task description, corresponding correct answers, and document fragments that support those answers] based on the provided documents. I will provide you with a knowledge base document, and I need you to first classify whether the document falls within the scope of the financial domain, and if so, which topic subcategory it belongs to.
Data Input Format
The input consists of the following two parts:
- Subcategory list: A list format of data, where each item in the list is JSON data representing a financial subcategory. This data includes the following attributes:
    - id: An integer value representing the id of the financial topic subcategory. Your classification result should return only the subcategory id, not the subcategory name.
    - topic_name: A string representing the name of the financial topic subcategory.
- Document content to be classified: A JSON formatted data, containing the following attributes:
    - title: A string representing the document title.
    - content: A string representing the document content.
Generated Data Format
You need to generate the value of the financial topic subcategory id that is most relevant to the document.
If the document content is unrelated to finance, or does not relate to any provided financial topic subcategory, please return 0.
Generate in JSON format, with the following data format:
{
    "topic_id": An integer value indicating the most relevant financial topic subcategory id for the document. If the document is unrelated to finance, please return 0.
}
Note to generate only JSON formatted data, and do not generate any other characters.
Subcategory List
topics_str
Document Content to be Classified
{
    "title": title,
    "content": content,
}
Most Relevant Subcategory ID for the Document

Figure 22: Instructions for GPT-4 to classify the domain topic for the input document.

Instructions for GPT-4 to automatically generate data instances.

Background
You are an intelligent evaluation data generation assistant. I am generating retrieval-augmented financial model multi-task evaluation data. I require you to automatically generate evaluation data that is strongly relevant to the evaluation tasks. I will provide the following content: [financial topic subcategories of interest for the evaluation data, task descriptions and requirements, documents in the knowledge base]. I need you to generate evaluation data that is strongly relevant to the provided financial topic area and meets the evaluation task requirements. The evaluation data includes the following content:
- User questions that align with the topic requirements and task descriptions
- Corresponding correct answers
- Document passages extracted from the original text that support those answers

Quality Requirements for Data Generation
...(see details in Boxs 24 and 25)

Data Generation Process:
1. First, determine if the document is a high-quality document. If the document is not closely relevant to the provided financial subtopic, has low informational content, is incomplete, has mixed formats, or does not meet the above requirements, then it is unsuitable for generating evaluation data. If the document is not suitable for generating domain-knowledge-related evaluation data, please return an empty list.
2. If the document is high-quality, further assess whether it is suitable for generating relevant data for the provided evaluation task. If it is not suitable, please return an empty list.
3. If the document is suitable for generating evaluation data relevant to the provided evaluation task and financial subtopic, please generate high-quality evaluation data.

Generated Data Format Requirements
The generated data should be returned in the form of a JSON data list, formatted as follows:
[
    {
        "thought_process": A Chinese string representing your thought process while generating this data entry,
        "question": A Chinese string representing the question posed by the user,
        "answer": A list of strings representing all possible forms of the answer to that question,
        "relevant_passage": A list of Chinese strings representing relevant content excerpts from the original document that help answer the question. Please ensure the completeness of the extracted passages' information,
    },
    ...
]

Financial Subcategories of Interest for Evaluation Data
{topic_name}

Task Description and Requirements
Task Name
{task_name}
Task Requirements
{task_require}

Provided Document
{doc_str}

List of Generated Data

Figure 23: Instructions for GPT-4 to automatically generate data instances.

Quality requirements for data generation – Part 1

- Quality Requirements for Documents:
  - First, determine whether the document is relevant to the domain being evaluated (financial subdomain). If it is not relevant, do not generate data.
  - The content used to generate evaluation data should not involve any personal privacy of users, such as names, phone numbers, ID numbers, home addresses, etc. If the provided document contains private information, please return an empty list.
  - The content used to generate evaluation data must be rigorous and of high quality; do not generate evaluation samples based on low-quality documents.
  - If you believe the document is unsuitable for generating evaluation data for the provided task, please return an empty list.
- Quality Requirements for Question Generation:
  - User questions should be as realistic as possible, simulating what users genuinely care about when applying large language models for knowledge Q&A in the financial domain.
  - Questions must be semantically complete and unambiguous. The user's intent should be clear from the question content alone. Questions that rely on the content of the provided document to complete the context are strictly prohibited.
  - Note that only when generating evaluation data for multi-turn dialogue capabilities should subsequent questions be ambiguous and dependent on previous dialogue content to clarify their semantics. In this case, subjects may be omitted or replaced with pronouns in later questions.
  - Users do not provide documents when asking real questions; they only ask questions. Therefore, real user questions will not involve phrases like "according to the given document...". Such questions are strictly prohibited.
  - The types of generated questions must strictly match the description of the evaluation task.
  - The generated questions must be strongly relevant to the provided financial subtopic.
  - Ensure the solvability of the generated questions. The answers in the generated data must be meaningful, and prohibited answers include "none", "empty", "unable to answer based on the retrieved document", etc.
- Quality Requirements for Answer Generation:
  - Only generate knowledge-rich data samples; the answers must contain substantial valuable information. Avoid generating vague or generic Q&A pairs, especially answers like "positive impact", "beneficial effect", etc., which lack actual meaning.
  - Answers must be consistent with the content of the provided document and should not contain factual inaccuracies or hallucinations.
  - Ensure the accuracy and factual validity of the generated answers. The answers in the generated data must be meaningful; prohibited answers include "none", "empty", "unable to answer based on the retrieved document", etc.
  - The format of answers can vary (*e.g.*, numeric in Arabic or Chinese characters, various date formats), and please provide all possible forms of the answer in a string list format.

Figure 24: Quality requirements for data generation – Part 1.

Quality requirements for data generation – Part 2

- Quality Requirements for Relevant Passage Extraction:
    - Must accurately provide document passages that support the answer; these passages must come from the original text of the provided document and cannot be altered.
    - The extracted relevant passage content must be complete and coherent, without missing contextual meaning.
- Overall Quality Requirements for Generated Evaluation Samples:
    - Please strictly follow the evaluation task requirements to generate evaluation data that corresponds to that task's capabilities; for instance, multi-hop reasoning tasks must generate questions that require multiple inferences from the retrieved documents to answer, rather than being answerable in a single reading.
    - The question-answer pairs generated must be answerable based on the content of the document, meaning understanding the document content is crucial to answering the question, and the role of the reference document cannot be ignored in the dialogue.
    - Multiple high-quality evaluation data entries can be generated, but the high quality of the generated data must be guaranteed.
    - Ensure precision in generated data rather than recall; only generate data that fully meets requirements, prohibiting data with low confidence.
    - Generated data must meet task requirements and be strongly relevant to the target task and financial domain. If the document cannot generate any task-related data, please return an empty list.
    - Ensure diversity in the generated data; do not generate multiple identical or closely similar evaluation data entries.

Figure 25: Quality requirements for data generation – Part 2.

Instructions for GPT-4 to inspect the quality of the generated instance – Part 1

Background
You are a professional data quality evaluator and corrector. I will provide you with evaluation data generated by a large language model (related to the financial domain), and your task is to assess the quality of this generated data and make corrections when necessary. The quality of the generated data is classified into three levels:
- 0: The quality of the generated data is very poor, and it cannot be suitably corrected to become high-quality data.
- 1: The quality of the generated data is average; the generated questions, answers, or extracted relevant passages do not meet the requirements, but they can be corrected to become high-quality data.
- 2: The quality of the generated data is very high and does not require correction.
Background Knowledge – Data Generation Process:
...(summarization of data generation process)
Input Content for Data Quality Evaluation Task:
1. A long document in the financial domain used for generating data.
2. The financial subtopic that the generated data should conform to.
3. The description and requirements of the evaluation subtask to which the generated data belongs.
4. The evaluation data generated by the large language model is to be assessed. The format of this data is a JSON list containing:
[
{
"thought_process": A Chinese string representing the thought process of the large language model when generating this data entry.
"question": A Chinese string representing the question posed by the user,
"answer": A list of strings representing all possible forms of the answer to that question.
"relevant_passage": A list of Chinese strings representing relevant content excerpts from the original document that help answer the question. Please ensure the completeness of the extracted passages' information.
},
...
]

Figure 26: Instructions for GPT-4 to inspect the quality of the generated instance – Part 1.

Instructions for GPT-4 to inspect the quality of the generated instance – Part 2

Data Quality Evaluation Requirements
1. Determine whether the generated questions are related to the provided financial subtopic.
2. Assess whether the generated questions meet the requirements of the evaluation subtask, paying particular attention to whether questions for multi-hop reasoning tasks require multi-hop reasoning.
3. Check if the answers to the generated questions are correct and whether they can be fully answered based on the provided long document.
4. Evaluate whether the extracted relevant passages from the original text are complete and sufficiently support the full answer to the generated questions.
Output Requirements and Format for Evaluation and Correction Results
Only when you assess the quality of the data as 1 should you make corrections; no corrections are needed for 0 or 2.
During the data quality evaluation process, pay special attention to the following key points:
- For questions of the form “yes or no” where the answer is usually “yes” or similar affirmative responses, please mark the quality as 0. This is because it is generally impossible to generate data pairs with a “no” answer, and such generated data would bias our dataset; therefore, please remove this type of generated data.
- For multi-hop reasoning questions, pay special attention to whether the question requires multi-hop reasoning, meaning the (retrieval-augmented) large language model needs to engage in at least two steps of “thinking-answering” reasoning to fully resolve the issue. If the question only adds complex conditions but can still be solved with a single inference, the quality of such generated data should be marked as 0 or 1. If it can be corrected based on the original document, mark it as 1 and correct it. If it cannot be corrected, mark it as 0.
The evaluation results should be returned in JSON format, with the specific format and requirements as follows:
{
"evaluation": An integer value indicating the assessment result of the generated data quality, with values in [0, 1, 2].
"corrected_result": A JSON list format of the corrected results for data assessed as quality 1, making them high-quality evaluation data. If the evaluation quality is 0 or 2, this attribute should be None. Note: The data format and types should be completely consistent with the input evaluation data generated by the large language model; only the contents of the internal attributes are corrected.
}
Long Document in the Financial Domain Used for Data Generation
{doc_str}
Financial Subtopic that the Generated Data Should Conform to
{topic_name} ## Description and Requirements of the Evaluation Task to Which the Generated Data Belongs
Task Name
{task_name}
Task Requirements
{task_require}
Evaluation Data Generated by the Large Language Model
{gen_datas}
Evaluation and Correction Results

Figure 27: Instructions for GPT-4 to inspect the quality of the generated instance – Part 2.