EXPLAIN: Enhancing Retrieval-Augmented Generation with Entity Summary

Yaozhen Liang[1], Xiao Liu[1], Jiajun Yu[1], Zhouhua Fang[2], Qunsheng Zou[2], Linghan Zheng[2], Yong Li[2], Zhiwei Liu[2,†], Haishuai Wang[1,†]
[1] Zhejiang University [2] Ant Group
{liang.yaozhen,xiaoxiaoliu,jiajunyu,haishuai.wang}@zju.edu.cn
{fangzhouhua.fzh,zouqunsheng.zqs}@antgroup.com
{zhenglinghan.zlh,liyong.liy,biao.lzw}@antgroup.com

Abstract

Document question answering plays a crucial role in enhancing employee productivity by providing quick and accurate access to information. Two primary approaches have been developed: retrieval-augmented generation (RAG), which reduces input tokens and inference costs, and long-context question answering (LC), which processes entire documents for higher accuracy. We introduce EXPLAIN (**EX**tracting, **P**re-summarizing, **L**inking and enh**A**c**IN**g RAG), a novel retrieval-augmented generation method that automatically extracts useful entities and generates summaries from documents. EXPLAIN improves accuracy by retrieving more informative entity summaries, achieving precision comparable to LC while maintaining low token consumption. Experimental results on internal dataset (ROUGE-L from 30.14% to 30.31%) and three public datasets (HotpotQA, 2WikiMQA, and Quality, average score from 62% to 64%) demonstrate the efficacy of EXPLAIN. Human evaluation in ant group production deployment indicates EXPLAIN surpasses baseline RAG in comprehensiveness.

1 Introduction

Document question answering requires processing large volumes of text to provide precise answers to user queries. Two primary approaches address this challenge: retrieval-augmented generation (RAG) and long-context (LC) question answering.

RAG methods improve computational efficiency by retrieving relevant document segments before generating answers, thus reducing input tokens and inference costs. However, this can lead to less precise answers due to the limited context (Xu et al., 2024b; Yu et al., 2024). In contrast, LC methods achieve higher accuracy by processing entire documents, but at the cost of increased computational resources (Li et al., 2024). The main challenge is finding a balance between accuracy and computational efficiency.

Many current QA systems utilize RAG approaches with various enhancements for retrieval accuracy, but improving document understanding while maintaining low inference costs remains a significant challenge.

To address these problems, we introduce EXPLAIN (**EX**tracting, **P**re-summarizing, **L**inking and enh**A**c**IN**g RAG), which enhances the retrieval-augmented generation approach by integrating advanced extraction and summarization techniques. EXPLAIN automatically extracts potentially useful entities from documents and generates concise summaries that retain essential information, achieving precision comparable to LC methods while maintaining lower token consumption.

The EXPLAIN method first extracts entities likely relevant to the query, then pre-summarizes these entities to create a condensed version of the document. Finally, it enhances the RAG process using these summaries to generate more accurate and comprehensive answers.

We evaluate EXPLAIN using an internal dataset focused on financial and human resources services and three public datasets: HotpotQA(Yang et al., 2018), 2WikiMQA(Ho et al., 2020), and Quality(Pang et al., 2022). Experimental results demonstrate significant improvements, with EXPLAIN achieving a ROUGE-L score increase from 30.19% to 30.31% on our internal dataset and an average score increase from 62% to 64% on the public datasets.

Following deployment in a production environment in September 2024, human evaluation indicates that EXPLAIN outperforms baseline RAG approaches in terms of detail and comprehensiveness, validating its practical applicability in real-world scenarios.

Our contributions can be summarized as follows:

[†]Corresponding authors.
This work was conducted during the internships of Yaozhen Liang, Xiao Liu, and Jiajun Yu at Ant Group.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1520–1529

- We propose EXPLAIN, a retrieval-augmented method enhanced by entity summarization, improving RAG accuracy while controlling token consumption.

- We conduct experiments on three public datasets and one proprietary financial dataset, with results showing consistent performance improvements across all benchmarks.

- We demonstrate the method's effectiveness in production environments through successful deployment and positive human evaluation.

2 Related Works

2.1 Retrieval-Augmented Generation

In recent years, large language models (LLMs) have excelled in various natural language processing tasks (Achiam et al., 2023)(Dubey et al., 2024)(Yang et al., 2024), yet they often struggle with knowledge-intensive tasks that require specific domain knowledge. Retrieval-Augmented Generation (RAG) has emerged as a promising approach to address these challenges by retrieving external documents to supplement the model's knowledge (Lewis et al., 2020)(Gao et al., 2024). Recent advancements in RAG have explored the integration of summary-enhanced generation and retrieval-augmented generation in long contexts.

2.1.1 Summary Augmented Generation

Summary-enhanced generation leverages LLMs' ability to produce diverse summaries, improving comprehension and response accuracy for long documents. Methods like RECOMP (Xu et al., 2023) and Raptor (Sarthi et al., 2024) use extractive and abstractive techniques to condense documents, while GraphRAG (Edge et al., 2024) constructs entity graphs to capture semantic relationships. Inspired by these methods, our approach simplifies the process by extracting key entities and generating concise noun-based summaries and enhances the model's understanding by focusing on core content.

2.1.2 Retrieval-Augmented Generation in Long Context

With the expansion of LLMs' context lengths, models can now process entire documents in a single pass, offering a more comprehensive understanding (Achiam et al., 2023)(Dubey et al., 2024)(Yang et al., 2024). However, this also introduces challenges in efficiently integrating retrieval and generation. Approaches like OP-RAG (Yu et al., 2024) use retrieval to filter irrelevant text, maintaining accuracy while reducing inference overhead. Inspired by this, our method employs entity noun summaries to replace irrelevant text blocks, further reducing context length and improving response accuracy. By focusing on key entities, we enhance the model's ability to understand queries and contexts, offering a novel perspective on retrieval-augmented generation.

2.2 Information Extraction

Information Extraction is an important domain in Natural Language Processing (NLP) that extract structured information from plain text automatically (Xu et al., 2024a). Traditional Information Extraction method (Wang et al., 2022) (Yamada et al., 2020) (Han et al., 2020) (Lu et al., 2022) training different model using human annotate data in different format for different downstream tasks. These approaches achieve powerful performance but face difficulty in collecting large-scale and high-quality data. The lack of high quality annotated data limits the extensibility of these approaches. Recently, LLMs (Dubey et al., 2024; Achiam et al., 2023; Yu et al., 2025) achieve impressive performance in all NLP tasks. People become interested in extracting information using LLMs. OneKE (Gui et al., 2024) introduce a high-quality dataset contained 0.32B tokens to fine-tuned LLMs to adapt to the IE task. PIVOINE (Lu et al., 2023), YAYI-UIE (Xiao et al., 2024) and INSTRUCTIE (Gui et al., 2023) employ intruction-tuning of open-source LLMs which achieve notable successes on IE. (Edge et al., 2024) Use a human-written few-shot instructions to iteratively extract entities and relations from plain text. In this work, we employ LLMs to perform entity summary after entity extraction, which further aggregate information needed for question answering. Since we don't have the prior knowledge about what exactly kind of entities down stream question needed, we can just extract all possible entities that might be useful. In this case, entity extraction become entity noun extraction. In our method, we use noun extraction pipeline to extract entity.

3 Methodology

We introduce EXPLAIN, a novel RAG paradigm designed to achieve higher accuracy with lower

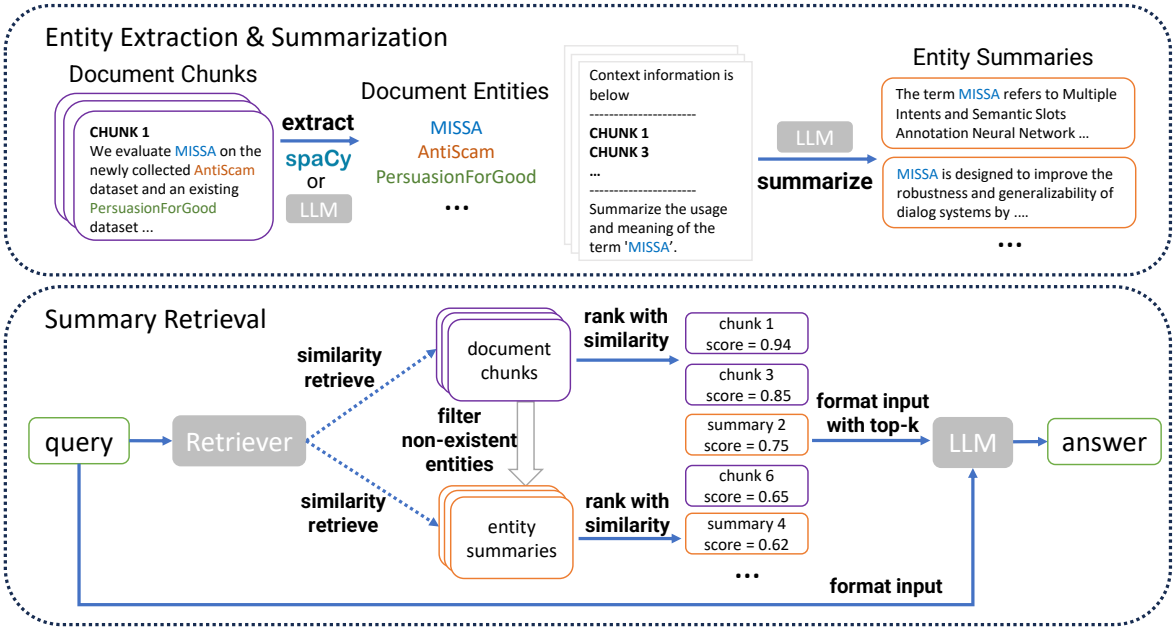

Figure 1: Main Framework of EXPLAIN.

inference consumption. As shown in Figure 1, EXPLAIN extracts entities from source documents, performs entity linking to resolve ambiguities, and generates concise summaries for these entities. When answering questions, it retrieves relevant documents and entity summaries, replacing low similarity documents with relevant entity summaries to enhance contextual information while decreasing inference consumption.

3.1 Entity Extraction

To enhance extraction rates and reduce costs, we employ a noun extraction method as a substitute for traditional entity extraction. We utilize the *en_core_web_sm*[*] pipeline in the spaCy library[†] for sentence segmentation and syntactic analysis, extracting complete nouns from sentences as entities. Given a document D divided into chunks $c_1, c_2, ..., c_n$, we extract entity nouns from each chunk to form entity sets $E_i = \{e_{i1}, e_{i2}, ..., e_{ik}\}$. We define two dictionaries: Context2Entity$(c_i) = E_i$ tracks entities in each chunk, and Entity2Context$(e_j) = \{c_k \mid e_j \in \text{Context2Entity}(c_k)\}$ records chunks containing each entity. While fast, spaCy extraction may introduce noise, so we also develop an LLM-based extraction method that produces less noise but requires more processing time.

[*] https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-3.8.0

[†] https://spacy.io/

3.2 Entity Linking

Algorithm 1 Jaccard Similarity-Based Entity Linking

Require: List of entity names `entname`; similarity threshold thr

Ensure: List of linked groups of entities `linkedgroups`

Initialize $n \leftarrow$ length of `entname`
Initialize `linkedgroups` $\leftarrow$ list containing n singleton sets: $\{\{e_1\}, \{e_2\}, \ldots, \{e_n\}\}$
Initialize a Union-Find data structure UF with elements $e_1, e_2, \ldots, e_n$
for $i = 1$ to $n - 1$ **do**
  for $j = i + 1$ to n **do**
    Calculate the Jaccard similarity $J(e_i, e_j)$ between `entname[i]` and `entname[j]`
    if $J(e_i, e_j) > T$ **then**
      UF.Union(e_i, e_j)
    end if
  end for
end for
`linkedgroups` $\leftarrow$ groups formed by UF **return** `linkedgroups`

To address the issue of entities appearing in different forms across a document, we develop an entity linking algorithm using n-gram Jaccard Sim-

ilarity:

$$J(s_1, s_2) = \frac{|N(e_1, n) \cap N(e_2, n)|}{|N(e_1, n) \cup N(e_2, n)|} \quad (1)$$

where $N(e, n)$ represents the set of n-grams extracted from entity e. As shown in In Algorithm 1, we initially assign each entity to its own distinct entity set. We then iteratively merge entity sets when their average Jaccard similarity exceeds a threshold T. For each merged set, we select the shortest entity name as the representative. After the iteration process completes, all entities with sufficient Jaccard similarity will be linked together within the same entity set.

3.3 Entity Summarization

For each entity e_i, we collect the fragments containing it using $C = \text{Entity2Context}(e_i)$ and randomly select a subset C' that fits within LLM context limits. To enhance summary completeness, we prompt the LLM to provide multiple discrete aspects of the entity's meaning and usage, citing relevant sentences before summarizing. These separate items serve as retrieval objects, improving performance over simpler summarization approaches. The prompt used for this process can be found in Appendix A.

3.4 Entity Summary Enhanced RAG

Given a question q, EXPLAIN retrieves document chunks $C = \{c_1, c_2, \ldots, c_n\}$ and extracts entity summaries E. A re-ranker orders both based on similarity to q. We replace lower-scoring chunks with higher-scoring entity summaries, using thresholds maxEntSumm and maxChunkRepl to balance entity summaries with contextual information. The final context consists of the most relevant entity summaries and document chunks, enhancing question answering quality.

4 Experiment

4.1 Datasets and Baselines

4.1.1 Datasets

We evaluate our method on three public and one private: (1) **HotpotQA** (Yang et al., 2018) is a question answering dataset featuring natural, multi-hop questions, with strong supervision for supporting facts to enable more explainable question answering systems. We use test split from LongBench (Bai et al., 2023) and report F1 score; (2) **2WikiMultihopQA** (2WikiMQA) consists of up to 5-hop questions that are synthesized using manually designed templates to ensure that they cannot be solved through shortcuts. We use test split from LongBench and report F1 score; (3) **QuALITY** (Pang et al., 2022) is a question answering dataset over stories and articles collected from Project Gutenberg and the Open American National Corpus. This is a multiple-choices dataset. The Model is required to select the correct one among four given options. Following (Xu et al., 2024b), we use official validation set as test set and report Exact Match score for QuALITY. We report Exact Match (EM) metrics, EM-V (common questions) and EM-H (hard questions), where EM-V and EM-H denote the EM scores on the common and hard question subsets of the validation set; (4) Internal QA Dataset: A Chinese QA dataset from real-world corporate scenarios containing 11,109 instances (10,000 for testing, 1,109 for validation). Performance is measured using ROUGE-L. We treat all documents as a single document for entity processing. Due to permission issues, the documents we collect in this dataset are only chunks related to the questions from the complete documents. Therefore, we are unable to test Self-Route and Long Context on this benchmark which requires full text.

4.1.2 Baselines

We implement five baselines to evaluate the effectiveness of our method: (1) **No Context**: a method that only gives LLMs input question without any documents. (2) **Standard RAG** (Lewis et al., 2020): formats input with input question and top-k retrieved document chunks. (3) **RAG+Reranker**: addtionally rerank top-k document chunks with reranker compared to Standard RAG. (4) **Long Context** (Li et al., 2024): formats input with question and full documents. (5) **Self-Route** (Li et al., 2024): let LLMs to route whether to use RAG+Reranker or Long Context according to if the retrieved document chunks can answer the question. More details of the implementation are shown in B

4.2 Main Results

The results of our offline experiments are presented in Table 1. our method, EXPLAIN, demonstrates impressive performance across all benchmarks. For the multi-hop question answering benchmarks, HotpotQA and 2WikiMQA, EXPLAIN outperforms other methods. Compared to Standard RAG and

Table 1: Main results on HotpotQA, 2WikiMQA, QuALITY and Internal QA Dataset. All results are in %. *Avg Token* denotes the average token consumption. The best result is in **bold** and the second best is underlined. ↑ denotes that a larger value is better, while ↓ denotes that a smaller value is better.

Dataset	HotpotQA		2WikiMQA		Quality			Internal QA Dataset		
Metric	F1 ↑	Avg Token ↓	F1 ↑	Avg Token ↓	EM-V ↑	EM-H ↑	Avg Token ↓	ROUGE-L ↑	F1 ↑	Avg Token ↓
No Context	9.67	**100**	20.28	**98**	34.87	26.48	**195**	7.21	1.23	**175**
Standard RAG	56.70	4380	56.38	4181	80.22	60.28	4256	30.14	20.41	1778
RAG+Reranker	56.39	4380	59.23	4181	79.53	60.66	4256	30.19	20.66	1778
Self-Route	51.30	5146	56.58	5146	80.41	59.71	4306	-	-	-
Long Context	47.75	12873	55.96	7187	**81.49**	**65.92**	5870	-	-	-
Explain (Ours)	**60.33**	4013	**62.78**	3893	80.41	60.00	3882	**30.31**	**21.05**	1738

RAG+Reranker, EXPLAIN achieves an F1 score improvement of 3.63% on HotpotQA and 3.55% on 2WikiMQA, while reducing average token usage by 135. This indicates that EXPLAIN effectively filters and utilizes relevant information, enhancing accuracy. In the Quality benchmark, where the context provided is a complete document relevant to the question, the Long Context method achieves the highest accuracy due to its comprehensive use of context. However, it also incurs the highest token consumption. EXPLAIN strikes a balance between efficiency and effectiveness, achieving near-top accuracy while using 200 fewer tokens than Standard RAG. In the Internal QA Dataset, EXPLAIN achieves a 0.39 increase in F1 score and a 0.12 increase in ROUGE-L score, with token consumption comparable to Standard RAG. This further demonstrates EXPLAIN's ability to enhance answer accuracy while maintaining low token usage.

Across all benchmarks, the 'No Context' method achieves very low scores, indicating that the questions are challenging and that the model cannot generate correct answers without external documents. In HotpotQA and 2WikiMQA, the contexts provided include both relevant documents necessary for reasoning and additional irrelevant documents. When input documents are not ranked by similarity, the model can be misled by irrelevant information, leading to decreased performance. As a result, the Long Context method underperforms on these benchmarks. Similarly, the irrelevant information confuses the selection process, resulting in lower performance of Self-route.

Overall, the experimental results indicate that EXPLAIN's entity summarization approach effectively guides the model in understanding questions, reducing interference from irrelevant information. This leads to improved accuracy and reduced token consumption, showcasing EXPLAIN's potential in complex question answering tasks.

4.3 Trade-off between inference token usage and accuracy

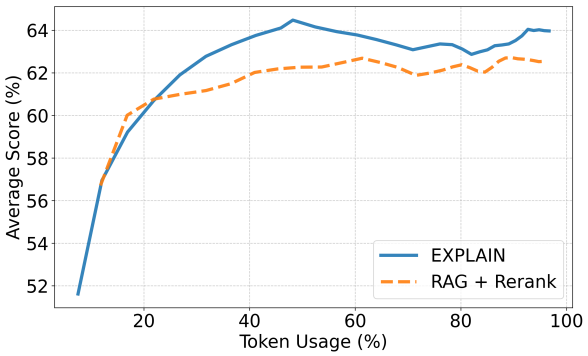

Figure 2: Token Usage(%) v.s Average Score(%) in HotpotQA, 2WikiMQA and Quality. We fix number of entity summaries to 10 and increase number of document chunks to increase token usage in each run.

In this section, we discuss the trade-off between accuracy and inference token consumption of EXPLAIN. As shown in Figure 2, we computed the average scores on three datasets: HotpotQA, 2WikiMQA, and Quality. We control token consumption by adjusting top-k for RAG+Reranker and maxChunkRepl for EXPLAIN. The token consumption percentage is computed as: (1) per-instance: the ratio between tokens consumed by inserted text chunks and tokens in the full relevant context, and (2) macro-level: the average across all instances. We plot the relationship between the average scores and this token consumption percentage. It can be observed that, in most cases, when the token usage percentage matches the baseline method RAG+Reranker, our method achieves approximately 1% to 2% higher score than the baseline. This demonstrates that our model consistently and steadily outperforms the baseline across these three benchmarks by effectively utilizing contex-

Table 2: Ablation results on HotpotQA, 2WikiMQA, and QuALITY. All results are in %. *Avg Token* denotes the average token consumption. The best result is in **bold** and the second best is underlined. ↑ denotes that a larger value is better, while ↓ denotes that a smaller value is better.

Dataset	HotpotQA		2WikiMQA		Quality		
Metric	F1 ↑	AVG Token ↓	F1 ↑	Avg Token ↓	EM-V ↑	EM-H ↑	Avg Token ↓
Explain (Default)	**60.33**	4013	**62.78**	3893	80.41	60.00	3882
w/ LLM extraction	54.95	4038	59.84	3912	**80.80**	**60.46**	3919
w/ aggregated summaries	51.67	5047	59.49	4802	79.24	59.81	5242
w/o entity linking	59.16	3929	61.10	**3852**	80.02	59.71	**3856**
w/o in-context retrieval	60.19	**3932**	62.48	3991	79.24	57.93	3868

tual information.

4.4 Ablation of EXPLAIN Components

We investigate the impact of various EXPLAIN components on model performance across HotpotQA, 2WikiMQA, and Quality datasets. Results are summarized in Table 2. We conducted ablations by modifying several key components of our system. First, we compared SpaCy versus LLM-based entity extraction methods. We also evaluated performance with and without the entity linking step. Additionally, we tested individual versus aggregated entity summary retrieval to assess granularity effects. Finally, we contrasted context-based versus full-document retrieval scopes. Our findings reveal several important insights about the system design. For entity extraction, SpaCy extracts 11.16% more entities than the LLM-based method, producing 20.26% more summaries. While this introduces some noise, the performance impact remains limited. Given SpaCy's computational efficiency, we adopt it in our final model despite the slight performance decrease. Regarding entity linking, omitting this step causes only marginal performance degradation. At a similarity threshold of 0.7, entity linking reduces entity count by 5.86%, primarily decreasing computational overhead in downstream steps without significantly affecting accuracy. The summary granularity experiments showed that aggregating all summaries of an entity into a single retrieval item significantly reduces performance while increasing token consumption. This suggests that consolidated summaries introduce irrelevant information that distracts the model from the query's focus. The impact of retrieval scope varies by dataset characteristics. For Quality, where the retriever's context already covers 72.5% of the full text, expanding to full-document retrieval has minimal effect. However, for HotpotQA and 2WikiMQA, full-document retrieval decreases performance by introducing less relevant entity summaries that confuse the model. These ablations demonstrate the robustness of EXPLAIN's design choices and highlight the importance of granular, context-relevant entity summaries in improving model performance.

4.5 Impact of maxEntSumm and maxChunkRepl on Performance

In this section, we examine the impact of the parameters $maxEntSumm$ and $maxChunkRepl$ on performance. The parameter $maxEntSumm$ determines the maximum number of entity summaries retrieved, while $maxChunkRepl$ determines the maximum number of context chunks that can be replaced by these summaries. In practice, we found that the average length of context chunks is 110 tokens, whereas entity summaries average 35 tokens. Replacing context chunks with shorter entity summaries can reduce token consumption. However, increasing $maxChunkRepl$ too much can lead to a loss of important context, as many questions are context-dependent. This often results in a decrease in accuracy that outweighs the benefits of adding more entity summaries. As shown in 3, settings with $maxChunkRepl$ of 20 and 10 generally perform worse than a setting of 5, due to excessive loss of context. On the other hand, increasing $maxEntSumm$ introduces more new information but also increases token usage. Through parameter searching, we find that setting $maxEntSumm$ to 10 provides a good balance, achieving optimal results across the datasets. This analysis highlights the importance of carefully balancing these parameters to optimize both token efficiency and model accuracy.

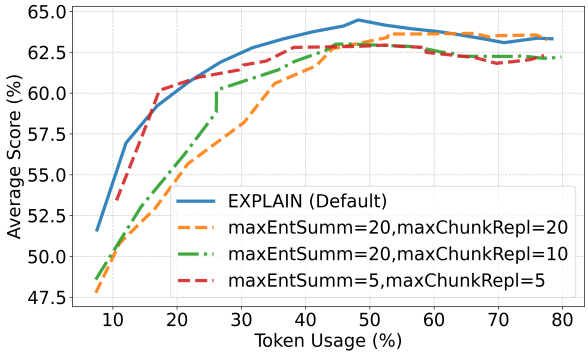

Figure 3: Token usage vs. F1 score in in HotpotQA, 2WikiMQA and Quality validation set. We increase number of contexts to increase token usage in each run.

Table 3: Vote results of online experiment.

Vote result	Accuracy	Comprehensiveness
EXPLAIN win	13.79	30.04
Tie	57.29	53.70
Baseline win	28.92	16.26

4.6 Online Experiments

We conducted a month-long online experiment involving 892 HR and financial queries handled by Ant Group's internal Q&A chatbot. Three company volunteers evaluated responses, comparing EXPLAIN against **RAG+Reranker** baseline, which has been consistently used to handle HR and financial inquiries in the ant group, on three metrics:

- **Accuracy:** The proportion of characters correctly addressing the user's question.
- **Comprehensiveness:** The extent to which the response covered all necessary information
- **Hallucination:** Instances where responses contradicted relevant documents

For each query, the evaluators were presented with the question, relevant internal documents, and two anonymized model responses. They selected which response performed better on accuracy and comprehensiveness, and mark if a response has any Hallucination. As shown in Table 3, for accuracy, EXPLAIN achieved 13.79% wins, 28.92% losses, and 57.29% ties against the baseline. Regarding comprehensiveness, EXPLAIN demonstrated a significant advantage with 30.04% wins, 16.26% losses, and 53.70% ties. For hallucinations, 2.5% of EXPLAIN's answers and 1.8% of the baseline's answers were marked, suggesting the entity summarization step does not significantly contribute to hallucination occurrence. Due to company data security policies, specific examples cannot be shared. Our analysis suggests that the lower accuracy win rate of EXPLAIN may be related to the nature of HR and financial queries, which typically require more detailed and contextualized answers than those found in public benchmarks. In these scenarios, EXPLAIN often introduces entity summaries or term definitions before providing the main answer. While this approach enhances comprehensiveness and better addresses the information needs of enterprise users, it can sometimes affect accuracy assessments. The additional contextual information may make the core answer less direct or introduce minor inaccuracies in supplementary details, which can impact strict accuracy evaluations even when the main point is correctly addressed.

5 Conclusion

In this work, we introduce EXPLAIN, a novel paradigm for document question answering based on the Retrieval-Augmented Generation framework. EXPLAIN addresses two key challenges: (1) the precision limitations of RAG-based methods due to restricted retrieved context, and (2) the high token cost of long-context-based approaches. By extracting potentially relevant entities from source documents and generating concise summaries for each, EXPLAIN enriches the information available during answer generation. These entity summaries are incorporated alongside retrieved passages, enabling the model to provide more accurate and comprehensive responses. Experimental results on public benchmarks demonstrate that EXPLAIN achieves superior inference accuracy and generation quality compared to the original RAG framework, without incurring additional real-time inference token costs. Furthermore, our month-long online experiment in a real-world corporate Q&A setting confirms that EXPLAIN significantly improves the comprehensiveness of responses to complex HR and financial queries, while maintaining a low hallucination rate. These findings highlight EXPLAIN's practical value for enterprise applications, where thorough and context-rich answers are essential.

6 Acknowledgments

This work was supported by Ant Group Research Fund.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Honghao Gui, Shuofei Qiao, Jintian Zhang, Hongbin Ye, Mengshu Sun, Lei Liang, Jeff Z Pan, Huajun Chen, and Ningyu Zhang. 2023. Instructie: A bilingual instruction-based information extraction dataset. *arXiv preprint arXiv:2305.11527*.

Honghao Gui, Hongbin Ye, Lin Yuan, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. Iepile: Unearthing large-scale schema-based information extraction corpus. *arXiv preprint arXiv:2402.14710*.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach.

Keming Lu, Xiaoman Pan, Kaiqiang Song, Hongming Zhang, Dong Yu, and Jianshu Chen. 2023. Pivoine: Instruction tuning for open-world information extraction. *arXiv preprint arXiv:2305.14898*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. 2022. Quality: Question answering with long input texts, yes!

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

Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuanjing Huang. 2022. MINER: Improving out-of-vocabulary named entity recognition from an information theoretic perspective. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5590–5600, Dublin, Ireland. Association for Computational Linguistics.

Xinglin Xiao, Yijie Wang, Nan Xu, Yuqi Wang, Hanxuan Yang, Minzheng Wang, Yin Luo, Lei Wang, Wenji Mao, and Daniel Zeng. 2024. Yayi-uie: A chat-enhanced instruction tuning framework for universal information extraction.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina

Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Retrieval meets long context large language models.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

Jiajun Yu, Yizhen Zheng, Huan Yee Koh, Shirui Pan, Tianyue Wang, and Haishuai Wang. 2025. Collaborative expert llms guided multi-objective molecular optimization. *arXiv preprint arXiv:2503.03503*.

Tan Yu, Anbang Xu, and Rama Akkiraju. 2024. In defense of rag in the era of long-context language models.

A Prompts used in EXPLAIN

A.1 summary prompt

summary prompt

our primary task is to summarize the usage and significance of the given term within the provided context. For each item in your summary, start by quoting the most relevant part of the original context using quotation marks and then provide a concise summary explaining the term's usage or significance in that context. Ensure each summary item is self-contained, capturing a complete idea or fact that can stand alone. Using '\n' to separate different items. Context information is below. **<CONTEXT>** Based on the context information, summarize the usage and significance of the term '**<ENTITY NAME>**'. For each item in your summary, start by quoting the most relevant sentence from the context using quotation marks, and then provide a concise summary explaining the term's usage or significance. Ensure that each summary item is both comprehensive and concise, and contains enough information to be understood independently, avoiding pronouns or references that rely on other sentences for context. Using '\n' to separate different items.

A.2 extract prompt

extract prompt

lease extract all the nouns and noun phrases in the context. Do not include any pronouns in your extraction. Provide the extracted nouns and noun phrases, separate them by commas, and do not provide any other text. Context: **<CONTEXT>** Please extract all the nouns and noun phrases in the Context. Do not include any pronouns in your extraction. Provide the extracted nouns and noun phrases separate them by commas and do not provide any other text.

B Experimental Settings

In our experiments, we employ the LLaMA3.1-8B-Instruct (Dubey et al., 2024) model as the foundational language model for the English dataset and the Qwen2.5-8B-Instruct (Yang et al., 2024) model for the Chinese dataset. For document preprocessing, we implement sentence-level chunking. We utilize spaCy's $en\_core\_web\_sm$ and $zh\_core\_web\_sm$ for English and Chinese sentence segmentation and respectively preprocess documents into chunks not exceeding 128 tokens. We encode and retrieve documents using the $dense\_vecs$ encoding method from BGE-m3 (Chen et al., 2024) and rerank the retrieved documents according to score from BGE-reranker-v2 (Chen et al., 2024). For entity extraction, we again utilize spaCy's $en\_core\_web\_sm$ and $zh\_core\_web\_sm$ for English and Chinese respectively and develop custom rules to extract nouns from sentences. For entity linking, we set the Jaccard similarity threshold T to 0.7. The LLaMA3.1-8B-Instruct and Qwen2.5-8B-Instruct models are employed for summarizing entities in English and Chinese. We retrieve top 40 chunks most similar to query for all baselines and EXPLAIN. We set maximum number of retrieved entity summaries $maxEntSumm$ to 10 and maximum number of document chunks that can be replaced $maxChunkRepl$ to 5 for EXPLAIN in HotpotQA and 2WikiMQA, $maxEntSumm$ to 10 and $maxChunkRepl$ to 7 in Quality and $maxEntSumm$ to 2 and $maxChunkRepl$ to 2 in Internal QA Dataset.