Retrieval Augmented Generation of Summarized Answers on Visually-Rich Documents for Trend and Risk Analysis

Retrieval Augmented Generation of Summarized Answers on Visually-Rich Documents for Trend and Risk Analysis

Giuseppe Gallipoli[1,*], Luca Cagliero[1], Alessandro Mosca[1], Arianna Miola[2,3] and Daniele Borghi[2]

[1]Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy
[2]Intesa Sanpaolo Innovation Center, Corso Inghilterra 3, 10138 Turin, Italy
[3]Università degli Studi di Milano-Bicocca, Piazza dell'Ateneo Nuovo 1, 20126 Milan, Italy

Abstract

The analysis of Visually-Rich Documents (VRDs) is crucial in the banking sector to support Trend and Risk Analysis (TRA) as financial TRA documents are multimodal to a large extent. Recently, Retrieval Augmented Generation (RAG) systems have enabled the effective use of Large Language Models (LLMs) to answer questions related to multimodal content. However, the inherent verbosity and complexity of financial documents could degrade the quality of the generated answers. In this work, we explore the use of text summarization techniques to condense the information retrieved from TRA-related VRDs. We analyze the level of synthesis of the original RAG answers, both with and without cascading an ad hoc summarization step. We apply summarization performance measures to compare standard RAG answers with the summarization outputs achieved on the retrieved passages directly. The results show that proprietary LLMs (GPT-4o) significantly improve the RAG's ability to sum up the retrieved passages, whereas integrating open-source LLMs or traditional summarizers turns out to be not beneficial even while applying the summarization step on top of the RAG answer.

Keywords

Visually-Rich Documents, Trend and Risk Analysis, Large Language Models, RAG Systems, Text Summarization

1. Introduction

Visually-Rich Documents (VRDs) are types of documents that are commonly used in the banking sector to perform Trend and Risk Analysis (TRA). They consist of visual and textual elements such as charts, diagrams, textual paragraphs, and tables. Multimodal elements collectively refer to semantic entities whose identification, comprehension, and elaboration are crucial to solve advanced reasoning tasks such as Visual Question Answering [1], Entity Linking [2], and Key Information Extraction [3].

In the banking sector, analysts of TRA units often need to query financial VRDs to gain insights into the latest advancements in economic and technological fields. To support this time-consuming activity, the use of Large Language Models has become increasingly appealing [4]. Specifically, Retrieved Augmented Generation (RAG) systems combine the effectiveness of Information Retrieval modules, which extract passages relevant to the analyst-generated question, with the generative capabilities of LLMs [5]. Existing RAG applications to financial documents mainly focus on textual reports [6, 7], with limited research devoted to multimodal sources [8, 9], which are, however, of major interest for TRA banking units. Although RAG answers produced by LLMs are expected to be relevant to the input question, their conciseness and non-redundancy are usually not guaranteed by design. However, especially when dealing with financial VRDs, the multimodal content and its textual reformulations are often characterized by a fairly high level of verbosity, making the generated answers not sufficiently focused.

To deal with this issue, we first design and implement a RAG system to manage the financial VRDs provided by three separate TRA units of a primary banking institution. Next, given the textual passages shortlisted by the multimodal retrieval step, we compare the level of synthesis of the RAG outputs produced by three alternative strategies:
(S1) Classical RAG: The LLM is prompted with the content of the retrieved passages without explicitly enforcing any summarization constraints;
(S2) Summarization: The retrieved passages are summarized by an ad hoc summarization module;
(S3) Cascade of RAG and Summarization: The output of S1 is summarized by an ad hoc summarization module.

We compare the summarization performance achieved by the above-mentioned strategies S1-S3 against a human-generated ground truth. The goal is to address the following research questions:
(Q1) *Are LLMs effective in summarizing TRA document passages?*
(Q2) *To what extent are RAG outputs less similar than summarizers' outputs to ground truth summaries?*
(Q3) *Is it beneficial to apply text summarization on top of the Classical RAG answers?*

The experimental results show that the answers provided by Classical RAG to TRA-related questions are inherently redundant, calling for ad hoc summarization strategies. While proprietary LLMs excel at generating concise summaries of the retrieved passages, the level of synthesis of open-source models (including LLMs) is, in general, not satisfactory. Furthermore, cascading RAGs with summarization modules (regardless of the approach used) turns out to be not beneficial.

2. Problem statement

Given a set of Visually-Rich Documents $\mathcal{D}$ related to Trend, Innovation, and Risk Analysis in the banking sector and a textual question $t \in \mathcal{T}$ on $\mathcal{D}$, the RAG system returns the answers to each question t by performing the following steps: (1) *Document chunking and encoding*: it recognizes and splits the visual and textual elements in the documents' content, generates alternative textual descriptions of the visual elements, and encodes the text corresponding to each element separately; (2) *Passage retrieval*: it encodes the ques-

Published in the Proceedings of the Workshops of the EDBT/ICDT 2025 Joint Conference (March 25-28, 2025), Barcelona, Spain
*Corresponding author.
giuseppe.gallipoli@polito.it (G. Gallipoli); luca.cagliero@polito.it (L. Cagliero); alessandro.mosca@studenti.polito.it (A. Mosca); arianna.miola@intesasanpaolo.com (A. Miola); daniele.borghi@intesasanpaolo.com (D. Borghi)
0009-0003-1744-6674 (G. Gallipoli); 0000-0002-7185-5247 (L. Cagliero); 0000-0001-6695-8877 (A. Miola)

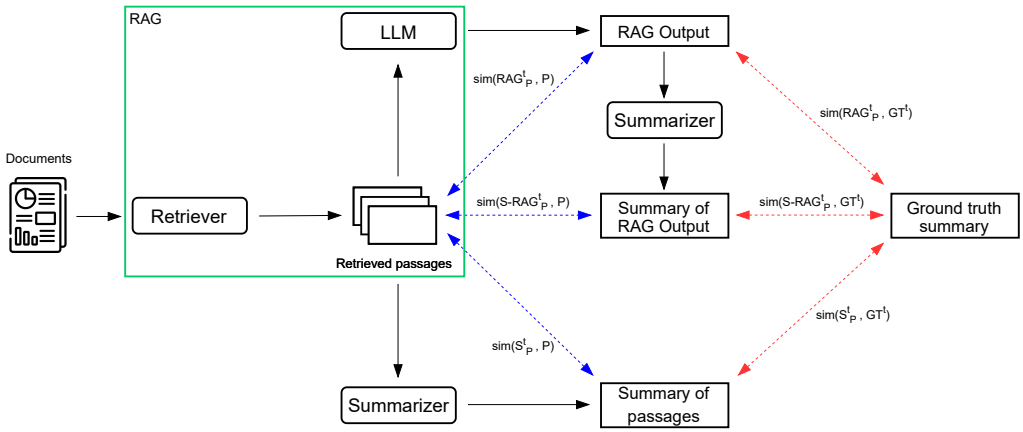

Figure 1: Sketch of our research scenario. Given a RAG system and an external summarizer, we analyze the similarities between the outputs of the RAG system and the summarizer with both the retrieved passages (shown as blue dashed lines) and the ground truth summaries (shown as red dashed lines).

tion t and retrieves the top-k passages P^t from $\mathcal{D}$ that are most relevant to t; (3) *LLM prompting*: it prompts the LLM with both the question t and the retrieved passages P^t. Note that Classical RAG prompts are designed for Question Answering and do not include any explicit summarization step.

We aim to analyze the level of synthesis of the RAG output RAG^t_P. Specifically, given an input question $t \in \mathcal{T}$, we compare the corresponding passages P^t and the (human-generated) ground truth summary GT^t_P of P^t with the following outputs:
(1) **Classical RAG**: The final output of the RAG, denoted by RAG^t_P;
(2) **Summarizer**: The output of an external summarizer that takes as input P^t, denoted by S^t_P;
(3) **Cascade of RAG+Summarizer**: The output of an external summarizer that takes as input the RAG output RAG^t_P, denoted by $S\text{-}RAG^t_P$.

The diagram in Figure 1 shows the scenario under analysis, where the similarities between the retrieved passages and the outputs are depicted using blue dashed lines, whereas those between the outputs and the ground truth summaries are depicted using red dashed lines. Note that the summarizer module is not necessarily integrated into the RAG system, as we explore various summarization approaches and models, including abstractive summarization (using both LLMs and non-LLM models) and hybrid strategies combining extractive and abstractive methods.

3. Settings of RAG and Summarizers

We implement the RAG system using the LangChain framework.

Document chunking and encoding We detect the VRD elements using the proprietary Document Intelligence service provided by the Azure AI platform [10] and generate alternative textual descriptions of visual contents using the Multimodal LLM GPT-4o [11]. To encode the VRD elements, we use the OpenAI `text-embedding-ada-002` embedding model.

Passage retrieval We retrieve passages via textual semantic similarity. Specifically, we retrieve the textual content associated with the k elements in $\mathcal{D}$ whose embeddings maximize the cosine similarity with the t's encoding.

LLM prompting We consider the proprietary LLM GPT-4o and use the following prompt:

```
You are a virtual assistant that can do Q&A. Try to
answer without using bullet points. Given the
following context, try to write a text that
highlights the topics discussed in the question.
If any of the context elements are not useful, ignore
them. If you don't know the answer, just say you don't
know, don't try to invent an answer, but say that the
documents you have can't satisfy the request.
[context]
[question]
```

where `[context]` and `[question]` are the retrieved passages and the current question, respectively.

External summarizers We conduct experiments with traditional Transformer-based models, i.e., LED [12], which is suited to long documents, PEGASUS [13], BART [14], and T5 [15], three open-source LLMs, i.e., Llama3-Instruct 8B [16], Zephyr 7B [17], and Mistral-Instruct 7B [18] and one proprietary LLM, i.e., GPT-4o [11]. For LLM-based summarization, we use the following prompt:

```
Summarize the following text.
Focus on the topic of [keyword]: [to_summarize]
```

where we replace `[keyword]` with the question expressed as a keyword and `[to_summarize]` with the corresponding retrieved passages to summarize. We also test two hybrid strategies combining extractive summarization using graph-based (TextRank, LexRank) or clustering (K-Means) methods with an LLM-based generative step.

4. Strategies for similarity computation

We evaluate the pairwise textual similarities between passages, LLM answers, and summaries using the following strategies:

Table 1
Human evaluation of summaries generated by the best-performing open-source LLM and GPT-4o. **Bold** denotes the best score for each metric.

	ICT Risk Analysis		Innovation Analysis		Trend Analysis	
	Llama3-Instruct	GPT-4o	Zephyr	GPT-4o	Zephyr	GPT-4o
Grammaticality	4.08±1.21	**4.44**±0.67	4.35±0.53	**4.53**±0.45	4.06±0.91	**4.23**±0.75
Usefulness	3.23±1.64	**3.40**±1.34	3.20±1.72	**3.40**±1.79	2.99±1.65	**3.76**±1.56
Coherence	3.55±1.19	**3.75**±1.34	3.70±0.71	**3.87**±1.10	3.66±1.14	**4.01**±1.21
Non-Redundancy	3.00±2.58	**4.15**±1.35	3.78±0.76	**3.91**±1.13	3.44±1.74	**3.90**±1.20
Overall Quality	3.02±1.64	**3.59**±1.38	3.17±1.55	**3.36**±1.78	2.96±1.71	**3.56**±1.62

Table 2
Similarity results between RAG and summarizers' outputs with the ground truth summaries. **Bold** denotes the best score for each metric. * and † denote results for which $p < 0.05$ with respect to the outputs of Classical and Cascade RAG+Summarizer.

dataset	model	R1	R2	RL	BS	keyword F1	Δ token
ICT Risk Analysis	GPT-4o	**31.2***†	**12.2***†	**18.1***†	**85.2***†	10.0*†	501†
	Llama3-Instruct	29.5†	9.4	16.9†	83.0*	7.4	160*†
	LED large	22.1*	11.1	15.7	81.2*†	**12.0***†	492†
	TextRank + Llama3-Instruct	29.9†	9.7	17.2†	82.8*	8.4†	170*†
	K-Means + GPT-4o	23.8*	9.0	13.4*	85.0*†	7.1	629*†
	Output of Classical RAG	29.2	9.4	16.2	84.4	7.1	504
	Output of Cascade RAG+Summarizer	24.6	7.4	13.9	84.2	5.6	573
Innovation Analysis	GPT-4o	**33.6***†	14.4†	21.2†	**86.0***†	10.5†	-67
	Zephyr	29.7	9.9	17.0*	84.4	15.0†	68
	BART large	26.5*	**18.7**†	**23.7**†	85.1	15.0†	175
	LexRank + Mistral-Instruct	32.4†	14.5†	15.4*	85.0	**20.0***†	65
	TextRank + GPT-4o	28.1	7.8	16.2*	85.4	7.5	124
	Output of Classical RAG	31.2	13.7	22.4	84.6	10.0	-44
	Output of Cascade RAG+Summarizer	26.5	5.4	14.9	83.9	5.0	39
Trend Analysis	GPT-4o	**42.8***†	**20.6***†	**25.1***†	**87.0***†	**18.5***†	245†
	Zephyr	31.5†	13.1†	18.3†	85.8†	18.1*†	538
	LED large	31.0†	16.8*†	20.4†	85.5	12.7	559
	LexRank + Llama3-Instruct	36.3†	16.3*†	20.1†	84.9	15.4†	-98*†
	TextRank + GPT-4o	27.7	10.7	15.9	86.0†	14.4	621*
	Output of Classical RAG	31.4	9.0	16.3	85.1	13.8	515
	Output of Cascade RAG+Summarizer	24.3	6.4	12.9	84.3	10.0	609

Syntactic similarity We compute the ROUGE-1/2/L (R1/2/L) F1-score [19], which indicates the unit overlap between the generated text and the ground truth in terms of unigrams, bigrams or longest common subsequence.

Semantic similarity We employ the BERTScore (BS) F1-score [20], which leverages BERT to compare the contextualized embeddings of the generated text and the ground truth.

Keyword-based similarity We first adopt KeyBERT [21] to extract keywords from both the generated text and the ground truth and then compute the corresponding F1-score.

LLM-based similarity We prompt GPT-4o with the generated texts and the ground truth summary, asking it to identify which model's answer is better.

5. Experimental results

Open-source models are accessed via the Hugging Face Transformers library and the proprietary GPT-4o (`gpt-4o-2024-05-13`) model [11] using the OpenAI API. We run experiments on a machine equipped with Intel® Core™ i9-10980XE CPU, 1 × NVIDIA® RTX A6000 48GB GPU, 128 GB of RAM running Ubuntu 22.04 LTS.

Datasets We analyze three proprietary collections of English VRDs provided by the following TRA units of a leading banking institution: (1) *ICT Risk*: 10 documents related to cyber risk, Distributed Ledger Technology, and AI in the ICT Risk area. They contain 2800 textual elements and 45 visual ones; (2) *Innovation*: 3 documents related to embedded finance/insurance, digital players, and Digital Wealth Management. They contain 82 textual elements and 32 visual ones; (3) *Trend*: 5 documents related to specific technologies and technological fields such as hydrogen economy. They mainly contain visual elements (232).

We ask TRA units' experts to generate questions corresponding to distinct keywords (72 for *ICT Risk*, 19 for *Innovation*, and 11 for *Trend*). For each question, ground truth summaries are manually annotated by at least 3 units' experts.

Human evaluation of generated summaries To answer Q1, we conduct a human validation of the passage summaries generated by the best-performing open-source LLM according to the automatic evaluation metrics and GPT-4o. TRA units' experts evaluated each output as *Very bad*, *Bad*, *Moderate*, *Good*, or *Very good* according to the following facets: *Grammaticality*, *Usefulness*, *Coherence*, *Non-Redundancy*, and *Overall Quality* [22]. The results reported in Table 1 highlight GPT-4o's superior summarization capabilities and, conversely, the limitations of open-source sum-

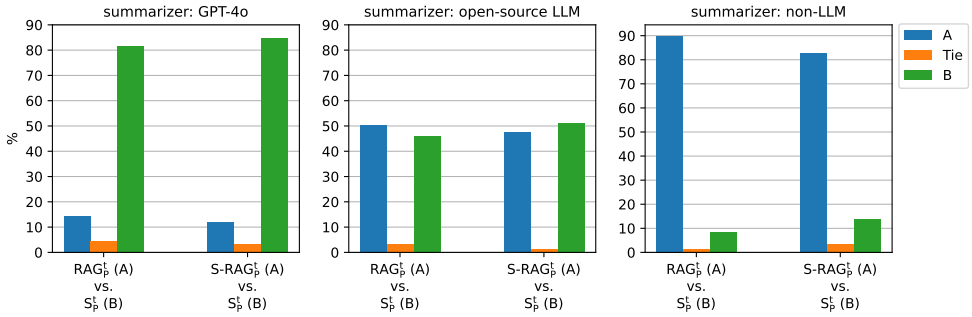

Figure 2: A/B tests using GPT-as-an-expert between (1) Left-hand side bars: Classical RAG output RAG^t_P (A) vs. summary output S^t_P (B); (2) Right-hand side bars: Cascade RAG+Summarizer output $S\text{-}RAG^t_P$ (A) vs. summary output S^t_P (B).

marizers on TRA-related VRDs. Notably, GPT-4o demonstrates significant improvements in both Non-Redundancy (e.g., 3.00 vs. 4.15 in ICT Risk Analysis) and Overall Quality (e.g., 2.96 vs. 3.56 in Trend Analysis) criteria.

Comparison between RAG and summarizers' outputs with respect to ground truth summaries To address Q2, we evaluate the similarities between the ground truth summaries and the outputs of (1) The classical RAG (see line *Output of Classical RAG* in Table 2), and (2) The best configurations for each dataset of the different summarizers employed (see lines *GPT-4o*, *Llama3-Instruct*, *Zephyr*, *LED large*, *BART large*, and hybrid strategies in Table 2). The results indicate that GPT-4o outperforms Classical RAG in terms of coherence with the ground truth summaries, whereas all the other summarizers, including the open-source LLMs and hybrid approaches, generally perform on par with or even worse than Classical RAG.

Effect of cascading RAG and Summarizer To answer Q3, the line *Output of Cascade RAG+Summarizer* in Table 2 reports the summarization performance of the approach based on applying summarization on top of the RAG output. In this case, we always use the best-performing model (i.e., GPT-4o) as the summarizer. The comparison with Classical RAG shows that cascading is never beneficial, even when employing the most effective summarizer, consistently resulting in performance degradation. These findings suggest that, since RAG answers are not specifically designed for the summarization task, concise answers condensing the relevant retrieved information can be produced more effectively by applying an explicit summarization step directly to the retrieved passages. Notably, summarizing the RAG output generated in the previous step fails to achieve the same quality as a direct summarization step, leading to even lower performance.

LLM-based similarity We carry out an A/B test using GPT-as-an-expert to compare Classical RAG against summarizers' outputs (see the left-hand side bars of the plots in Figure 2) and Cascade RAG+Summarizer against summarizers' outputs (see the right-hand side bars). We consider as summarizer GPT-4o, the best-performing open-source LLM and traditional Transformer-based model separately for each dataset. The results align with the automatic evaluation: GPT-4o as a summarizer outperforms both Classical and Cascade RAG+Summarizer (>80% vs. <20%), open-source LLMs perform comparably with them (both around 50%), whereas non-LLM summarizers demonstrate worse performance (<20% vs. >80%).

Comparison between RAG and summarizers' outputs with respect to retrieved passages Instead of evaluating the similarities between the generated outputs and the ground truth summaries (see Table 2), in this analysis we consider the retrieved passages as references. Specifically, we report in Table 3 the results of the comparisons between the retrieved passages and the outputs of (1) The classical RAG (see line *Output of Classical RAG*), (2) The cascade of RAG and summarizer (see line *Output of Cascade RAG+Summarizer*), and (3) The best configurations for each dataset of the different summarizers tested (see lines *GPT-4o*, *Llama3-Instruct*, *Zephyr*, *LED large*, *BART large*, and hybrid strategies).

In most cases, GPT-4o outperforms all the other approaches including open-source LLMs, traditional Transformer-based models, and hybrid strategies. Notably, it achieves significantly higher performance compared to both types of RAG outputs (i.e., Classical and Cascade RAG+Summarizer). Similar to the previous analysis, applying an additional summarization step on top of the RAG output proves to be detrimental, leading to even lower scores across all metrics. The results are aligned with those obtained using ground truth summaries as references. However, here we focus on a different aspect. Higher similarity with respect to the retrieved passages denotes better attribution of the generated text to the source passages. This is particularly relevant in TRA domains, where maintaining a high level of accountability to the document sources is critical. In conclusion, both sets of results indicate that GPT-4o, when used as a summarizer directly applied to the retrieved passages, excels at generating summaries that align well not only with the ground truth summaries but also with the source documents. In contrast, the other summarizers, and in particular the two types of RAG outputs considered, demonstrate lower performance.

6. Conclusions

This paper explored methods for summarizing passages retrieved by a RAG system indexing financial VRDs related to Trend and Risk Analysis in the banking sector. Employing proprietary LLMs as summarizers enhances the level of synthesis of classical RAG outputs, whereas open-source LLMs or traditional summarizers do not achieve significant performance improvements due to the inherent complexity of multimodal, domain-specific sources. Notably, applying

Table 3
Similarity results between RAG and summarizers' outputs with the retrieved passages. **Bold** denotes the best score for each metric. * and † denote results for which $p < 0.05$ with respect to the outputs of Classical and Cascade RAG+Summarizer.

dataset	model	R1	R2	RL	BS	keyword F1	Δ token
ICT Risk Analysis	GPT-4o	**45.5***†	27.2*†	**35.4***†	**88.7***†	**36.6***†	433†
	Llama3-Instruct	37.6	22.4	22.8	84.5*†	21.1	50*†
	LED large	33.2	**28.5***†	27.0†	84.4	32.6*†	473†
	TextRank + Llama3-Instruct	36.8	21.4†	22.9	84.6*†	23.1	41*†
	K-Means + GPT-4o	36.1†	21.6†	23.8	88.1*†	29.3*†	593*†
	Output of Classical RAG	33.5	17.8	23.9	86.5	20.8	494
	Output of Cascade RAG+Summarizer	28.2	12.9	19.5	85.9	17.9	557
Innovation Analysis	GPT-4o	**41.8***†	21.8*†	32.7*†	87.9*†	31.0*†	228
	Zephyr	37.6†	22.6*†	27.9*†	86.7*†	25.8*†	230
	BART large	39.6†	**34.8***†	**37.2***†	**88.3***†	**42.7***†	338*†
	LexRank + Mistral-Instruct	35.8†	19.5†	20.0	86.4	26.8*†	224
	TextRank + GPT-4o	39.9†	19.7†	30.1†	87.9*†	32.5*†	304
	Output of Classical RAG	31.4	13.6	21.2	85.2	14.5	210
	Output of Cascade RAG+Summarizer	27.1	8.7	17.1	84.9	15.7	256
Trend Analysis	GPT-4o	**42.5***†	**28.9***†	**32.7***†	**89.2***†	**36.7***†	2133*†
	Zephyr	27.7†	21.1*†	20.6*†	86.8†	25.9*†	2416
	LED large	26.3†	24.4*†	23.1*†	85.7	27.1*†	2407
	LexRank + Llama3-Instruct	40.2*†	25.2*†	22.1*†	85.4	28.5*†	1764*†
	TextRank + GPT-4o	24.6†	15.3†	17.1†	87.0*†	24.8†	2462*
	Output of Classical RAG	24.5	11.8	15.8	85.8	16.5	2394
	Output of Cascade RAG+Summarizer	18.7	8.2	11.9	85.0	12.8	2471

summarizers directly to the retrieved passages has shown to be more effective than cascading RAGs with an additional summarization step.

As future work, we plan to generate summarized answers using RAG systems with different characteristics and evaluate them on existing benchmarks, as well as using sequence-to-sequence models specialized on languages other than English [23]. We also aim to generate explanations highlighting the weaknesses of RAG outputs.

Limitations

We identify the following limitations of our work:

Open-source LLMs Due to computational constraints, we have currently tested the 8B parameter version of Llama3-Instruct. As a future extension, we plan to evaluate a broader suite of open-source LLMs with varying levels of complexity. It is worth noting that, despite their significantly smaller number of parameters, the open-source LLMs considered show fairly good performance, in some cases comparable to that of larger, proprietary ones.

Model fine-tuning Currently, both LLMs and traditional Transformer-based model versions we employ are not specialized on domain-specific data. We plan to fine-tune a selection of models to generate more domain-aware summaries.

RAG architecture For visual elements, the retrieval module currently relies on semantic similarity between textual descriptions of multimodal elements generated using GPT-4o. We aim to explore the use of different Multimodal LLMs that also capture layout information (e.g., LayoutLLM [24]) and test them in combination with various document retrieval strategies.

Ethics statement

Proprietary LLMs are, by design, non-transparent, therefore the reproducibility of their results is limited and the motivations behind their weaknesses remain partly obscure.

LLMs are also known to suffer from bias issues. Although we verified the coherence of the summarization outputs with the help of domain experts, we cannot guarantee that the summaries produced by generative AI models in a real-world scenario are entirely free of hallucination. This could be mitigated by introducing ad hoc approaches or models to evaluate the factuality of the generated answers, enabling to filter out or possibly improve non-factual outputs.

The proprietary data were selected by banking units' experts who were explicitly trained on how to properly sample the document collections under analysis. While the overall quality of the data sampling is above average, we cannot exclude the possibility that the collection may contain outliers or minor errors.

Data and code availability

Documents cannot be disclosed due to confidentiality and copyright restrictions. Code could be made available upon request to the authors.

Credits to financial institution

Intesa Sanpaolo is a leading banking group in the Eurozone and the largest one in Italy. Intesa Sanpaolo Innovation Center is part of ISP group, and its mission is exploring business models of the future to discover new assets and skills that support the long-term competitiveness of ISP group and its customers. ISP has established the Innovation Center Labs to respond to the complex needs of the bank and the market, determined by the evolution of market trends and exponential technology growth.

Acknowledgments

The authors would like to thank Giorgio Bella, Anna Polise, Stefania Vigna, Laura Li Puma, Chiara Napione, Giovanni Troiano, Patrizio Paolo Dionisi, Maura Bertaglia, Carla Monferrato, and Simone Scarsi for their useful comments. They would also like to thank Luigi Ruggerone for supporting the research team.

The work by Giuseppe Gallipoli was carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004). This study was also partially carried out within the FAIR (Future Artificial Intelligence Research) and received funding from Next-GenerationEU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1555.11-10-2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, ACM, 2020, p. 1192–1200. URL: http://dx.doi.org/10.1145/3394486.3403172. doi:10.1145/3394486.3403172.

[2] Y.-M. Chen, X.-T. Hou, D.-F. Lou, Z.-L. Liao, C.-L. Liu, Damgcn: Entity linking in visually rich documents with dependency-aware multimodal graph convolutional network, in: G. A. Fink, R. Jain, K. Kise, R. Zanibbi (Eds.), Document Analysis and Recognition - ICDAR 2023, Springer Nature Switzerland, Cham, 2023, pp. 33–47.

[3] Y. Ding, L. Vaiani, C. Han, J. Lee, P. Garza, J. Poon, L. Cagliero, 3MVRD: Multimodal multi-task multi-teacher visually-rich form document understanding, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 15233–15244. URL: https://aclanthology.org/2024.findings-acl.903. doi:10.18653/v1/2024.findings-acl.903.

[4] J. Lee, N. Stevens, S. C. Han, M. Song, A survey of large language models in finance (finllms), 2024. URL: https://arxiv.org/abs/2402.02315. arXiv:2402.02315.

[5] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T. Chua, Q. Li, A survey on RAG meeting llms: Towards retrieval-augmented large language models, in: R. Baeza-Yates, F. Bonchi (Eds.), Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024, ACM, 2024, pp. 6491–6501. URL: https://doi.org/10.1145/3637528.3671470. doi:10.1145/3637528.3671470.

[6] A. J. Yepes, Y. You, J. Milczek, S. Laverde, R. Li, Financial report chunking for effective retrieval augmented generation, 2024. URL: https://arxiv.org/abs/2402.05131. arXiv:2402.05131.

[7] Y. Zhao, P. Singh, H. Bhathena, B. Ramos, A. Joshi, S. Gadiyaram, S. Sharma, Optimizing LLM based retrieval augmented generation pipelines in the financial domain, in: Y. Yang, A. Davani, A. Sil, A. Kumar (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 279–294. URL: https://aclanthology.org/2024.naacl-industry.23. doi:10.18653/v1/2024.naacl-industry.23.

[8] G. Gallipoli, S. Papicchio, L. Vaiani, L. Cagliero, A. Miola, D. Borghi, Keyword-based annotation of visually-rich document content for trend and risk analysis using large language models, in: Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 130–136. URL: https://aclanthology.org/2024.finnlp-1.13.

[9] S. Xue, T. Chen, F. Zhou, Q. Dai, Z. Chu, H. Mei, Famma: A benchmark for financial domain multilingual multimodal question answering, 2024. URL: https://arxiv.org/abs/2410.04526. arXiv:2410.04526.

[10] Microsoft Azure, Azure AI Document Intelligence, 2024. URL: https://azure.microsoft.com/en-us/products/ai-services/ai-document-intelligence/.

[11] OpenAI, OpenAI GPT-4o, 2024. URL: https://openai.com/index/gpt-4o-system-card/.

[12] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, 2020. URL: https://arxiv.org/abs/2004.05150. arXiv:2004.05150.

[13] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, Pegasus: pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[14] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020).

[16] Llama Team, The Llama 3 Herd of Models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[17] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, T. Wolf, Zephyr: Direct distillation of lm alignment, 2023. arXiv:2310.16944.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. arXiv:2310.06825.

[19] C.-Y. Lin, ROUGE: A package for automatic evaluation

of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[20] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[21] M. Grootendorst, KeyBERT: Minimal keyword extraction with BERT, 2020. URL: https://doi.org/10.5281/zenodo.4461265. doi:10.5281/zenodo.4461265.

[22] N. Iskender, T. Polzehl, S. Möller, Reliability of human evaluation for text summarization: Lessons learned and challenges ahead, in: A. Belz, S. Agarwal, Y. Graham, E. Reiter, A. Shimorina (Eds.), Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval), Association for Computational Linguistics, Online, 2021, pp. 86–96. URL: https://aclanthology.org/2021.humeval-1.10.

[23] M. La Quatra, L. Cagliero, Bart-it: An efficient sequence-to-sequence model for italian text summarization, Future Internet 15 (2023). URL: https://www.mdpi.com/1999-5903/15/1/15. doi:10.3390/fi15010015.

[24] C. Luo, Y. Shen, Z. Zhu, Q. Zheng, Z. Yu, C. Yao, Layoutllm: Layout instruction tuning with large language models for document understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 15630–15640.