Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing Text Summarization Consistency via Iterative Reset Strategies and Context Analysis in RAG-Based Summarization with Hybrid LLMs

Abdulrahman Mohsen Ahmed Zeyad[1]*, Arun Biradar[1]*
[1]*School of Computer Science and Engineering, REVA University, Bangalore, 560064, India*
*Corresponding authors: dramzeyad@gmail.com, arun.biradar@reva.edu.in

ARTICLE INFO

Received: 29 Dec 2024

Revised: 15 Feb 2025

Accepted: 24 Feb 2025

ABSTRACT

Introduction: This paper investigates the transformative impact of iterative reset strategies within a Retrieval-Augmented Generation (RAG) framework for text summarization. It examines how integrating resets into hybrid large language model (LLM) workflows can enhance summary coherence and reduce retrieval noise.

Objectives: The study aims to evaluate the effects of discrete reset methodologies on the performance of RAG-based summarization. It focuses on improving content overlap and stylistic consistency while measuring outcomes using standard metrics such as ROUGE, FactCC, and readability scores.

Methods: A comprehensive RAG pipeline is developed by combining text segmentation, semantic embedding, vector database indexing, keyword extraction, and stylistic analysis. Five reset strategies are implemented and tested on the eLife dataset, with iterative evaluations conducted to compare the resulting ROUGE and FactCC metrics alongside processing time and Flesch-Kincaid readability measures.

Results: The analysis reveals that iterative resets particularly the vector database reset (Method 3) and the full reset (Method 5) yield higher ROUGE scores (0.3716 and 0.3706, respectively) compared to baseline approaches. However, these methods also exhibit variable factual consistency, as evidenced by moderate FactCC scores.

Conclusions: The findings underscore that while iterative reset strategies significantly enhance content overlap and summary coherence in RAG frameworks, they also introduce challenges in maintaining factual accuracy. The study offers valuable insights into optimizing RAG workflows and suggests further exploration of adaptive reset mechanisms to achieve a balanced performance.

Keywords: Text Summarization Consistency, Keyword Extraction, Stylistic Analysis.

INTRODUCTION

Text summarization remains a critical task in natural language processing, particularly for distilling complex scientific literature and facilitating efficient information retrieval [1], [2]. Recent advancements have led to the development of Retrieval-Augmented Generation (RAG) frameworks that leverage both small and large language models to improve summary quality by incorporating external knowledge [1], [3]. Despite decades of research [4], a notable gap persists in systematically integrating iterative resets applied to key components such as the vector database, stylistic analysis, keyword extraction, and retrieval queries to reduce retrieval noise and better align the summarization prompt with core thematic cues [3], [5].

In this context, the central idea is that integrating iterative resets in RAG workflows coupled with stylistic analysis and keyword extraction can enhance the quality, coherence, and stylistic fidelity of text summaries.

It is posited that applying iterative resets to key components of the RAG workflow namely, the vector database, stylistic analysis, keyword extraction [6], and retrieval queries will enhance the quality, coherence, and stylistic

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

fidelity of generated summaries by reducing retrieval noise and better aligning the summarization prompt with core thematic cues. In particular, Method 5 (full reset) is expected to yield the most significant improvements. The hypothesis is testable via quantitative metrics (ROUGE-1, ROUGE-2, ROUGE-L, Factual Consistency, Flesch-Kincaid) and comparison with human reference summaries [2], [5], [7]. The contributions of this work are threefold: (i) a novel integration of discrete reset strategies within a RAG framework, (ii) a comprehensive evaluation using standard quantitative metrics alongside human reference benchmarks, and (iii) insights into the differential impact of resets on various content type in domains such as scientific research [8], [9], [10] [11].

RELATED WORK

The body of research on retrieval-augmented approaches consistently demonstrates the benefits of integrating retrieval mechanisms with language model generation. For instance, Lewis et al. [1] showed these methods excel in knowledge-intensive tasks, while Johnson and Jones [2] improved retrieval precision by incorporating vector databases and semantic embeddings. Building on these studies, later work explored multi-stage pipelines—combining text segmentation, semantic embedding, and context retrieval—where iterative resets emerged as a promising technique to refine summarization performance [3], [5].

Evaluation methods evolved beyond traditional ROUGE metrics [7] to include measures of factual consistency (FactCC) and readability (Flesch-Kincaid) [12]. Although benchmark datasets like FRANK and FactCollect provided human-labeled factuality assessments, challenges remain in aligning automatic metrics with human judgment [10], [13]. Existing datasets and metrics, designed for fine-tuned summarization models, often fail to capture the nuances of large language models in zero-shot or few-shot settings, and reference summaries themselves sometimes contain inaccuracies [4], [14], [15]. Moreover, ensuring consistency in summaries from state-of-the-art models is challenging, as many still produce factual errors [9], [16].

In response, iterative resets were employed to periodically refresh the retrieval context and recalibrate cues from stylistic analysis and keyword extraction, aiming to reduce noise and enhance both factual and stylistic accuracy [17], [18]. Empirical results indicate that while reset strategies improve content overlap as measured by ROUGE, maintaining consistent factual accuracy remains complex [9], [19]. These findings underscore the need for ongoing refinement in balancing multiple quality dimensions within summary generation.

METHODOLOGY

A Retrieval-Augmented Generation (RAG) framework was developed to summarize the eLife dataset [20]. by combining small and large language models. Figure 1 illustrates a pipeline that includes text segmentation, semantic embedding, vector database indexing, keyword extraction, stylistic analysis, context retrieval, prompt engineering, and summary generation.

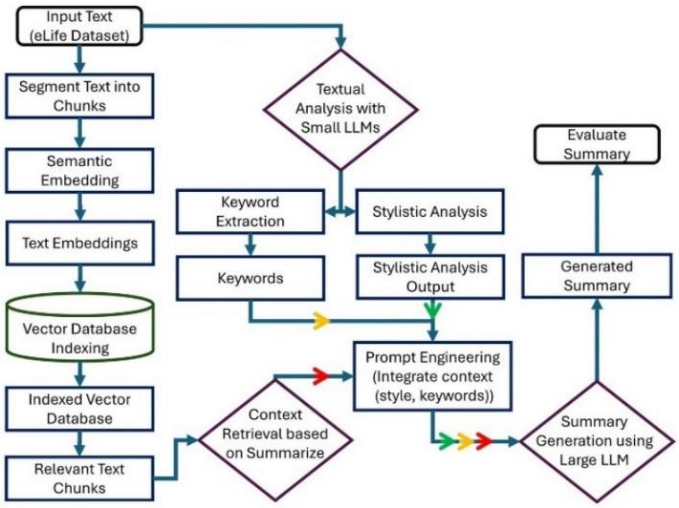

Figure 1: System Architecture for RAG-Based Summarization of the eLife Dataset.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

a. Overview of the RAG Framework

The RAG pipeline employed small language models for keyword extraction and stylistic analysis on text chunks, while a large language model generated final summaries from the retrieved context. The core steps were:

1. **Text Chunking and Embedding** [21]**:** The dataset was segmented into smaller chunks and converted into numerical embeddings.
2. **Vector Database Indexing** [22]**:** The embeddings were stored in a vector database for efficient retrieval.
3. **Textual Analysis:** Small LLMs extracted keywords and stylistic features from each chunk.
4. **Context Retrieval:** Relevant chunks were retrieved based on semantic similarity.
5. **Prompt Engineering** [23]**:** Extracted features were integrated into a prompt for the large LLM.
6. **Summary Generation:** The large language model generated the final summaries.
7. **Evaluation:** Generated summaries were assessed for performance and readability.

b. Mathematical Model

The RAG-Based Summarization Methodology and Mathematical Definitions

- The *eLife* dataset, denoted as $\mathcal{D}$, comprises N textual documents.
- Each document $d_i \in \mathcal{D}$ is divided into m_i segments, expressed as $\{c_{i,1}, c_{i,2}, \dots, c_{i,m_i}\}$.
- An embedding function, $f_{\text{embed}}(\cdot)$, is applied to each segment $c_{i,j}$, yielding a vector $\mathbf{v}_{i,j}$ in $\mathbb{R}^d$:

$$\mathbf{v}_{i,j} = f_{\text{embed}}(c_{i,j}), \tag{1}$$

where d represents the dimensionality of the embedding space.

- The resulting embeddings are stored in a vector database, $\mathcal{V} = \{\mathbf{v}_{i,j}\}$.
- A compact language model, LLM_{small}, is employed to identify keywords and assess stylistic features for each segment. The set of keywords extracted from segment $c_{i,j}$ is denoted $\mathcal{K}_{i,j}$, while the associated stylistic properties are represented as $\mathcal{S}_{i,j}$:

$$(\mathcal{K}_{i,j}, \mathcal{S}_{i,j}) = LLM_{\text{small}}(c_{i,j}). \tag{2}$$

- For summary generation, a query q is defined to reflect the user's intent or the primary topic. This query is transformed into an embedding vector $\mathbf{v}_q = f_{\text{embed}}(q)$. A similarity function, $\text{sim}(\cdot,\cdot)$, is then utilized to identify the most pertinent segments from the database:

$$\mathcal{R} = \arg\max_{\mathbf{v}_{i,j} \in \mathcal{V}} \text{sim}(\mathbf{v}_q, \mathbf{v}_{i,j}), \tag{3}$$

where $\mathcal{R}$ denotes the retrieved subset of embeddings and their associated text segments.

- A prompt p is subsequently formulated by combining the query q with the keyword sets $\{\mathcal{K}_{i,j}\}$ and stylistic attributes $\{\mathcal{S}_{i,j}\}$ derived from the retrieved segments $\mathcal{R}$:

$$p = \text{PromptEngineer}(q, \{\mathcal{K}_{i,j}\}, \{\mathcal{S}_{i,j}\}). \tag{4}$$

- Lastly, a comprehensive language model, LLM_{large}, produces the summary S based on the engineered prompt:

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$S = LLM_{\text{large}}(p). \quad (5)$$

This mathematical model encapsulates the core components of the retrieval-augmented summarization process. It integrates semantic retrieval with stylistic alignment to enhance the quality of summary generation.

RESULTS

This section presents the performance outcomes of five Retrieval-Augmented Generation (RAG)-based summarization methods and six comparison models evaluated on the eLife dataset. The evaluation utilized ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and Factual Consistency Metrics (FactCC) [24], alongside processing time and Flesch-Kincaid readability scores. Findings were organized systematically in Tables 1 through 7 to enable critical evaluation.

a. Mean Performance of RAG-Based Methods

Mean performance metrics for the RAG-based methods were calculated from Iteration 5 of five generated summaries. Table 1 presents these results.

Table 1: Mean Performance Metrics for RAG-Based Methods (Iteration 5)

Method	R-1	R-2	R-L	FactCC
Method 1: Normal RAG	0.351	0.0611	0.1636	0.0753
Method 2: Full-Featured	0.346	0.0617	0.1526	0.1142
Method 3: Vector Database Reset	0.3716	0.0718	0.1628	0.1263
Method 4: Analysis & Retrieval Reset	0.3482	0.0661	0.1573	0.1041
Method 5: Full Reset	0.3706	0.0682	0.1618	0.1186

- **Observation**: Methods 3 and 5 achieved higher R-1 scores than the baseline Method 1 and Method 2. Method 3 recorded the highest R-1 (0.3716), while Method 3 also obtained the highest FactCC (0.1263).

b. Mean Performance of Comparison Models

Mean performance metrics for the comparison models were derived from a single iteration. Table 2 presents these findings.

Table 2: Comparison Models' Mean Performance Metrics (Single Iteration)

Model	R-1	R-2	R-L	FactCC
Claude 3 Opus	0.391	0.084	0.169	0.049
Claude 3 Haku	0.374	0.075	0.168	0.129
Gemini 1.5 Pro 002	0.374	0.075	0.168	0.129
Gemini 1.5 Flash 002	0.371	0.070	0.157	0.177
GPT-4o	0.376	0.067	0.159	0.086
GPT-4o mini	0.400	0.089	0.175	0.242

- **Observation**: GPT-4o mini outperformed all models across ROUGE metrics and achieved the highest FactCC (0.242), establishing a reference standard for comparison.

c. Detailed Performance Across Iterations

R-1 scores across five iterations for each RAG-based method were analyzed to assess consistency. Table 3 presents these results.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: R-1 Scores Across Summary Iterations for RAG-Based Methods

Method	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Mean	Std Dev
Method 1	0.344	0.358	0.334	0.360	0.359	0.351	0.011
Method 2	0.348	0.293	0.370	0.358	0.361	0.346	0.030
Method 3	0.368	0.371	0.373	0.377	0.369	0.372	0.003
Method 4	0.370	0.295	0.354	0.363	0.359	0.348	0.029
Method 5	0.372	0.376	0.358	0.377	0.369	0.371	0.007

- **Observation**: Methods 3 and 5 exhibited greater consistency (standard deviations of 0.003 and 0.007) compared to Methods 2 and 4 (standard deviations of 0.030 and 0.029).

FactCC scores across five iterations for each RAG-based method were examined to evaluate factual consistency. Table 4 presents these findings.

Table 4: FactCC Scores Across Summary Iterations for RAG-Based Methods

Method	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Mean	Std Dev
Method 1	0.0751	0.0705	0.0980	0.0820	0.0490	0.0753	0.017
Method 2	0.0870	0.0640	0.0990	0.1040	0.2230	0.1142	0.060
Method 3	0.1930	0.2390	0.0950	0.1040	0.0480	0.1263	0.074
Method 4	0.1370	0.0390	0.1270	0.0900	0.1250	0.1041	0.040
Method 5	0.0340	0.1170	0.1310	0.0830	0.1920	0.1186	0.058

- **Observation**: Methods 3 and 5 demonstrated variability in FactCC scores (standard deviations of 0.074 and 0.058), with Method 3 showing a wider range of factual consistency across iterations.

d. Mean Processing Time for RAG-Based Methods

Mean processing times for generating summaries were measured in seconds and averaged across iterations. Table 5 presents these results.

Table 5: Mean Processing Time for RAG-Based Methods (in Seconds)

Method	Mean Processing Time (seconds)
Method 1: Normal RAG	246.03
Method 2: Full-Featured	690.86
Method 3: Vector Database Reset	868.28
Method 4: Analysis & Retrieval Reset	1379.00
Method 5: Full Reset	1386.40

- **Observation**: Method 1 required the least time (246.03 seconds), whereas Methods 4 and 5 demanded significantly more (1379.00 and 1386.40 seconds), reflecting the computational burden of reset strategies.

e. Readability of Generated Summaries

Flesch-Kincaid readability scores were calculated across five iterations for each RAG-based method. Table 6 presents the mean scores.

Table 6: Flesch-Kincaid Readability Metrics Across Five Summaries

Method	S1 FK	S2 FK	S3 FK	S4 FK	S5 FK	Mean	Std Dev
Method 1	15.92	14.82	15.34	15.70	15.30	15.42	0.39
Method 2	17.42	16.74	16.94	17.46	16.58	17.03	0.37

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Method 3	16.44	15.58	15.94	16.86	15.52	16.07	0.55
Method 4	17.26	16.70	16.00	17.28	16.34	16.72	0.50
Method 5	16.90	16.24	15.74	17.30	15.92	16.42	0.60

- **Observation**: Lower scores indicate higher readability. Method 1 produced the most readable summaries (mean 15.42), while Method 2 generated the least readable (mean 17.03). Methods 3 and 5 balanced readability and performance.

f. Creative Displays: Relative Performance with Readability

Table 7 compares relative performance to GPT-4o mini, incorporating Flesch-Kincaid readability scores for all methods and models.

Table 7: Relative Performance to GPT-4o mini (%) with Readability

Method/Model	R-1 (%)	FactCC (%)	Flesch-Kincaid Mean
Method 1	87.8	31.1	15.42
Method 2	86.5	47.2	17.03
Method 3	92.9	52.2	16.07
Method 4	87.1	43.0	16.72
Method 5	92.7	49.0	16.42
Claude 3 Opus	97.8	20.2	14.35
Claude 3 Haku	93.5	53.3	15.95
Gemini 1.5 Pro 002	93.5	53.3	15.95
Gemini 1.5 Flash 002	92.8	73.1	15.88
GPT-4o	94.0	35.5	16.74
GPT-4o mini	100.0	100.0	15.33

- **Observation**: Methods 3 and 5 reached over 92% of GPT-4o mini's R-1 performance but showed lower FactCC relative performance (52.2% and 49.0%). Among comparison models, Gemini 1.5 Flash 002 achieved the highest FactCC relative performance (73.1%).

g. Visual Representations

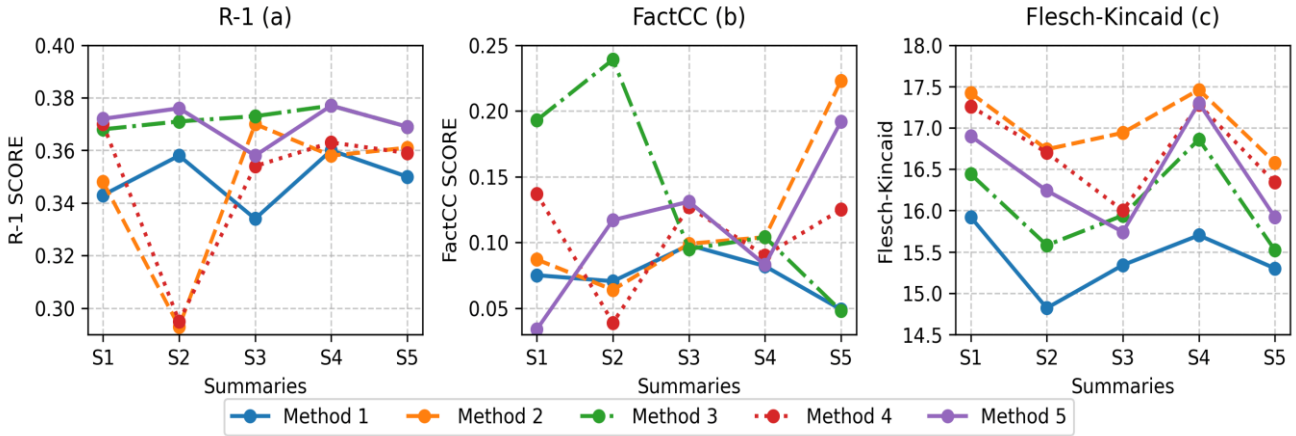

Figure 2: Line Graph of R-1 Scores (a), FactCC Scores (b), and Flesch-Kincaid Scores (c) Across Five Iterations for RAG-Based Methods

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

1. **R-1 Scores (a)** [7] "The top graph labeled 'R-1' shows performance trends across five iterations (S1 to S5) for five RAG-based methods. Methods 3 and 5 maintain consistent scores above 0.36, with peaks at 0.377, while Method 2 dips to 0.30 in S2, reflecting its higher variability (standard deviation 0.030)."
2. **FactCC Scores (b)** [24] "The middle graph labeled 'FactCC' tracks factual consistency scores across five iterations. Methods 3 and 5 show significant variability, ranging from 0.048 to 0.239 and 0.034 to 0.192, respectively, consistent with standard deviations of 0.074 and 0.058."
3. **Flesch-Kincaid Scores (c)** [12] "The bottom graph labeled 'Flesch-Kincaid' measures readability across iterations. Method 1 consistently scores around 15 (mean 15.42), indicating high readability, while Method 2 peaks at 17.46 (mean 17.03), indicating lower readability."

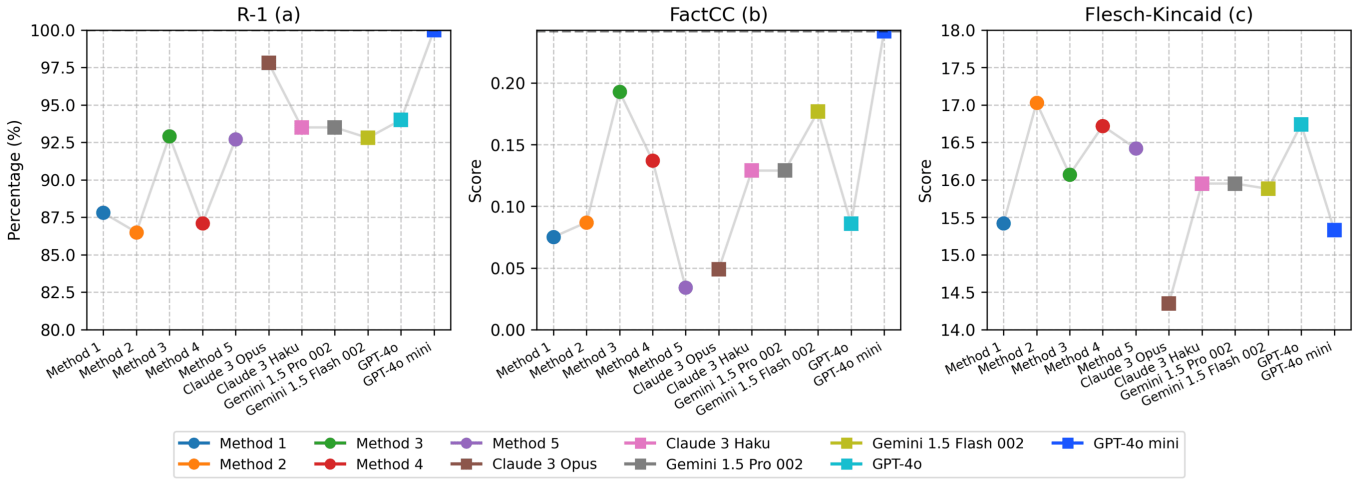

Figure 3: Line Graph of R-1 Scores (a), FactCC Scores (b), and Flesch-Kincaid Scores (c) for All Methods and Models

1. **R-1 Scores (a)** [7] The graph shows performance trends for five RAG-based methods and six comparison models. GPT-4o mini consistently scores 100%, while Methods 3 and 5 reach 92.9% and 92.7%, respectively, and Claude 3 Opus achieves 97.8%.
2. **FactCC Scores (b)** [24] This graph presents factual consistency, with GPT-4o mini leading at 0.242 (100%). Gemini 1.5 Flash 002 follows at 0.177 (73.1%), and Method 3 peaks at 0.1263 (52.2%).
3. **Flesch-Kincaid Scores (c)** [12] The graph assesses readability. Claude 3 Opus is most readable at 14.35, followed by GPT-4o mini at 15.33 and Method 1 at 15.42, while Method 2 scores highest at 17.03, indicating lower readability.

DISCUSSION

This study finds that reset strategies within RAG workflows enhance summarization quality, particularly in ROUGE metrics, yet exhibit variable factual consistency as measured by FactCC. Methods 3 (Vector Database Reset) and 5 (Full Reset) outperform the baseline, achieving R-1 scores of 0.3716 and 0.3706, respectively, although their FactCC scores (0.1263 and 0.1186) remain moderate compared to models such as GPT-4o mini (0.242). The improved performance in ROUGE likely stems from these strategies' ability to minimize irrelevant retrievals, thereby boosting content overlap. This is supported by the low standard deviations in R-1 scores (0.003 for Method 3 and 0.007 for Method 5). However, the relatively higher variability in FactCC scores (0.074 and 0.058, respectively) indicates that gains in content overlap do not fully translate to consistent factual accuracy.

In addition, reset methods incur substantially higher processing times compared to the baseline, highlighting an efficiency trade-off. Readability assessments show that while Method 1 produces the most accessible summaries, Methods 3 and 5 offer a more balanced alternative. Limitations include the exclusive reliance on the eLife dataset and a limited set of comparison models, which may affect the generalizability of the findings.

Future research should explore adaptive reset mechanisms that more effectively harmonize retrieval refinement with factual verification. Expanding evaluations to include diverse datasets and incorporating human assessments could

further refine these methods, ultimately optimizing RAG workflows for improved summarization quality and reliability.

CONCLUSION

The present study explored the impact of iterative reset strategies on Retrieval-Augmented Generation (RAG)-based text summarization. Reset methods improved summarization quality, particularly in ROUGE metrics, with Method 3 (Vector Database Reset) and Method 5 (Full Reset) achieving the highest R-1 scores (0.3716 and 0.3706, respectively). However, these methods displayed moderate factual consistency, with FactCC scores of 0.1263 and 0.1186, indicating that while resets boosted content overlap, they did not consistently ensure factual accuracy. Reset strategies also produced greater consistency in R-1 scores across iterations, as evidenced by lower standard deviations (0.003 for Method 3 and 0.007 for Method 5). Despite these improvements, variability in factual consistency remained challenging, with higher standard deviations in FactCC scores (0.074 and 0.058, respectively). Additionally, reset strategies incurred higher computational costs, with Methods 4 and 5 requiring over 1300 seconds for processing compared to 246.03 seconds for the baseline. While these strategies enhance summarization coherence and reduce retrieval noise, they involve trade-offs in factual accuracy and efficiency. Future research will investigate adaptive reset mechanisms and evaluate diverse datasets and human assessments to optimize RAG workflows.

REFERENCES

[1] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, pp. 9459–9474, 2020.

[2] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," *IEEE Trans. Big Data*, vol. 7, pp. 535–547, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:926364

[3] A. Pagnoni, V. Balachandran, and Y. Tsvetkov, "Understanding Factuality in Abstractive Summarization with {FRANK}: A Benchmark for Factuality Metrics," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 4812–4829. doi: 10.18653/v1/2021.naacl-main.383.

[4] J. Maynez and et al., "On faithfulness and factuality in abstractive summarization," *Proc. 58th Annu. Meet. Assoc. Comput. Linguist.*, 2020, [Online]. Available: https://aclanthology.org/2020.acl-main.141/

[5] D. Van Veen *et al.*, "Adapted large language models can outperform medical experts in clinical text summarization," *Nat. Med.*, vol. 30, no. 4, pp. 1134–1142, 2024.

[6] A. M. A. Zeyad and A. Biradar, "Advancements in the Efficacy of Flan-T5 for Abstractive Text Summarization: A Multi-Dataset Evaluation Using ROUGE and BERTScore," in *2024 International Conference on Advancements in Power, Communication and Intelligent Systems (APCI)*, 2024, pp. 1–5. doi: 10.1109/APCI61480.2024.10616418.

[7] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries Chin-Yew," *Assoc. Comput. Linguist.*, pp. 74–81, 2004, doi: https://aclanthology.org/W04-1013.

[8] M. Alkhalaf, P. Yu, M. Yin, and C. Deng, "Applying generative AI with retrieval augmented generation to summarize and extract key clinical information from electronic health records," *J. Biomed. Inform.*, vol. 156, p. 104662, 2024, doi: https://doi.org/10.1016/j.jbi.2024.104662.

[9] W. Kryściński, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," *EMNLP 2020 - 2020 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 9332–9346, 2020, doi: 10.18653/v1/2020.emnlp-main.750.

[10] A. Wang, K. Cho, and M. Lewis, "Asking and Answering Questions to Evaluate the Factual Consistency of Summaries," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 5008–5020. doi: 10.18653/v1/2020.acl-main.450.

[11] A. M. A. Zeyad and A. Biradar, "Assessing BigBirdPegasus and BART Performance in Text Summarization:

Research Article

Identifying Right Methods," in *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, 2023, pp. 1773–1778. doi: 10.1109/ICPCSN58827.2023.00297.

[12] M. Tahir *et al.*, "Evaluation of {Quality} and {Readability} of {Online} {Health} {Information} on {High} {Blood} {Pressure} {Using} {DISCERN} and {Flesch}-{Kincaid} {Tools}," *Appl. Sci.*, vol. 10, no. 9, 2020, doi: 10.3390/app10093214.

[13] L. Tang *et al.*, "Evaluating large language models on medical evidence summarization," *npj Digit. Med.*, vol. 6, no. 1, p. 158, 2023, doi: 10.1038/s41746-023-00896-7.

[14] R. Bommasani and C. Cardie, "Intrinsic Evaluation of Summarization Datasets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 8075–8096. doi: 10.18653/v1/2020.emnlp-main.649.

[15] P. Tejaswin, D. Naik, and P. Liu, "How well do you know your summarization datasets?," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 3436–3449. doi: 10.18653/v1/2021.findings-acl.303.

[16] Z. Luo, Q. Xie, and S. Ananiadou, "Factual consistency evaluation of summarization in the Era of large language models." 2025. [Online]. Available: https://arxiv.org/abs/2402.13758

[17] S. Gabriel, A. Celikyilmaz, R. Jha, Y. Choi, and J. Gao, "{GO} {FIGURE}: A Meta Evaluation of Factuality in Summarization," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 478–487. doi: 10.18653/v1/2021.findings-acl.42.

[18] S. Zhang, D. Wan, and M. Bansal, "Extractive is not Faithful: An Investigation of Broad Unfaithfulness Problems in Extractive Summarization," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 2153–2174. doi: 10.18653/v1/2023.acl-long.120.

[19] S. Chen, S. Gao, and J. He, "Evaluating Factual Consistency of Summaries with Large Language Models." 2023. [Online]. Available: https://arxiv.org/abs/2305.14069

[20] T. Goldsack, Z. Zhang, C. Lin, and C. Scarton, "Making Science Simple: Corpora for the Lay Summarisation of Scientific Literature," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 10589–10604. doi: 10.18653/v1/2022.emnlp-main.724.

[21] Nomic A, "Nomic A," Nomic A. [Online]. Available: https://www.nomic.ai/

[22] Q. Team, "Qdrant: High-performance vector database for the next generation of AI applications." 2023. [Online]. Available: https://qdrant.tech

[23] S. Maity, A. Deroy, and S. Sarkar, "Investigating Large Language Models for Prompt-Based Open-Ended Question Generation in the Technical Domain," *SN Comput. Sci.*, vol. 5, no. 8, p. 1128, 2024, doi: 10.1007/s42979-024-03464-2.

[24] D. Tam, A. Mascarenhas, S. Zhang, S. Kwan, M. Bansal, and C. Raffel, "Evaluating the Factual Consistency of Large Language Models Through News Summarization," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 5220–5255. doi: 10.18653/v1/2023.findings-acl.322.