The University of Stavanger (IAI) at the TREC 2024 Retrieval-Augmented Generation Track

Weronika Łajewska and Krisztian Balog

University of Stavanger, Stavanger, Norway
{weronika.lajewska, krisztian.balog}@uis.no

Abstract. This paper describes the participation of the IAI group at the University of Stavanger in the TREC 2024 Retrieval-Augmented Generation track. We employ a modular pipeline for Grounded Information Nugget-based GEneration of Conversational Information-Seeking Responses (GINGER) to ensure factual correctness and source attribution. The multistage process includes detecting, clustering, and ranking information nuggets, summarizing top clusters, and generating follow-up questions based on uncovered subspaces of relevant information. In our runs, we experiment with different length of the responses and different number of input passages. Preliminary results indicate that ours was one of the top performing systems in the augmented generation task.

Keywords: Conversational AI, Retrieval-Augmented Generation, TREC RAG

1 Introduction

The TREC Retrieval-Augmented Generation (RAG) track is designed to advance research and innovation in the field of retrieval-augmented generation systems. These systems combine retrieval techniques, which locate relevant information within large datasets, with large language models (LLMs) to produce accurate, relevant, and contextually appropriate content. The goal is to enhance system performance in generating high-quality outputs by leveraging both retrieval and generation capabilities.

In 2024, the TREC RAG Track contains three tasks: 1) retrieval (R) task, 2) augmented-generation (AG) task, and 3) retrieval-augmented (RAG) task. Our focus is on the second task where participants generate answers using retrieval-augmented generation systems, relying on top-k relevant segments provided by a baseline retrieval system. These segments come from the MS MARCO Segment v2.1 collection, and participants must ensure that their answers are well-supported by these retrieved segments, with proper attribution. The focus is on generating high-quality, contextually accurate answers that are backed by specific retrieved information.

The focus of our participation is to investigate the impact of the number of input passages and the length of generated response on the end-to-end system performance. We use a modular response generation pipeline that 1) ensures the grounding of the response in specific facts from the retrieved sources and 2) controls the coverage of the user's information need in the response. Our method operates on information nuggets defined as "minimal, atomic units of relevant information" of retrieved documents that have been proposed for automatic evaluation of passage relevance [19]. As a baseline for response generation, we use the approach proposed for open-domain QA in the retrieval-augmented setting with the off-the-shelf LLM without further training. We use the most recent at the time of writing snapshot of OpenAI's GPT-4 model (`gpt-4-turbo-2024-04-09`), which is the model achieving the highest scores in terms of faithfulness on the task of summary generation [27], as well as the most commonly used LLM architecture in RAG [8, 17, 22, 24, 26].

The evaluation process for the task involves assessing how well the citations support the generated sentences on a scale of 0..2: no, partial, or full support. Retrieved sentences are pooled, and nuggets—key pieces of information—are assigned to specific sentences in participants' responses. These nuggets help evaluate the accuracy and relevance of the generated content. The final scores are aggregated, considering linguistic features like fluency and answer length, ensuring the generated answers retrieve relevant data and maintain high language quality. Through this rigorous evaluation, the track aims to push the boundaries of retrieval-augmented systems, enhancing their ability to generate well-supported, coherent answers. At the time of submission of this report, we have results obtained with the AutoNuggetizer framework indicating that GINGER is the top performing system for the AG task [21] in terms of the nugget-based V_strict metric proposed in the track for response evaluation.

2 Related work

Generative retrieval, unlike traditional search engines, provides a comprehensive response by synthesizing perspectives from multiple sources, blending generative models' language fluency and world knowledge with retrieved evidence [5, 16]. In retrieve-then-generate systems, generative processes are conditioned on retrieved material by adding evidence to the prompt [9, 22, 26] or attending to sources during inference, as in models conditioned on retrieved document chunks [1]. Our focus is on the generation phase, using off-the-shelf models without altering model weights [17, 22].

Retrieved documents may contain irrelevant information, which can harm RAG systems [2], especially when evidence is added to prompts for complex queries [11]. Performance suffers when relevant information is buried within long contexts [14], making traditional retrieve-then-generate methods less effective at reducing hallucinations [11]. To address this approaches like context curation [10] and knowledge-grounded reasoning chains [4] have emerged. We enhance these efforts by curating context and decomposing the response generation process to mitigate irrelevant information.

Despite the fluency of search engine responses, they often contain unsupported claims or inaccurate citations [13]. Abstractive summaries from LLMs are prone to hallucinations and factual errors [3, 12, 28], making source attribution a crucial component to ensure information verifiability [23]. Systems with high citation precision may lack fluency, while fluent systems risk misleading users with unsupported content [13]. To address this, we propose extracting atomic statements from sources and summarizing them using LLMs, ensuring factual accuracy while maintaining fluency. Statements are referred to as “atomic/semantic content units” [15, 18] or “information nugget” in traditional IR [19, 25].

3 Approach

We introduce a modular pipeline for Grounded Information Nugget-based GEneration of Conversational Information-Seeking Responses (GINGER) that explicitly models different aspects of the query using retrieved information, producing concise responses that meet length constraints. The process for generating responses involves multiple stages: 1) identifying information nuggets in top relevant passages, 2) grouping these nuggets by different facets of the query, 3) ranking the clusters by their relevance, 4) summarizing the top clusters for the final response, and 5) refining it for fluency and coherence. This pipeline ensures that the final output remains grounded in the source material, tackling the “lost in the middle“ [14] issue common in LLMs, where the model often focuses on the beginning or end of long texts. GINGER aims at improving the response generation process and assumes ranked passages are provided as input.

3.1 Detecting Information Nuggets

We use a large language model (LLM) to detect key information nuggets in the retrieved passages automatically. The LLM is instructed to identify and annotate sections of the text that contain essential information answering the query. It does this by marking the information nuggets with specific tags while keeping the passage’s original content intact, without introducing extra symbols or altering the passage itself.

3.2 Clustering Information Nuggets

Once the information nuggets are detected, the next step is to cluster them based on the various facets of the query. Clustering serves two purposes: it helps to reduce redundancy by grouping similar nuggets that may appear across different documents, and it increases the information density of the generated response. To tackle the challenge of closely related nuggets within the same topic, we use a neural topic modeling technique called BERTopic [7], adjusting its sensitivity to capture nuanced differences between nuggets. Ideally, each cluster represents a specific facet of the answer.

Table 1. Overview of submitted runs. GINGER without the final response fluency improvement step is referred to as GINGER-fluency.

RunID	Method	Response length	# input passages	Priority
ginger_top_5	GINGER	3 sentences	5 top candidates	1
baseline_top_5	Baseline	3 sentences	5 top candidates	2
ginger-fluency_top_5	GINGER-fluency	3 sentences	5 top candidates	3
ginger-fluency_top_10	GINGER-fluency	400 words	10 top candidates	4
ginger-fluency_top_20	GINGER-fluency	400 words	20 top candidates	5

3.3 Ranking Facet Clusters

After clustering, we rank the facet clusters to determine their relevance to the query and prioritize which clusters should be included in the final response. Given the relatively small number of clusters, we use pairwise reranking techniques, such as duoT5 [20], to improve accuracy. This method compares pairs of clusters to decide which ones contain the most important information for the response, ensuring the highest-ranked clusters are prioritized.

3.4 Summarizing Facet Clusters

The final response is built from summaries of the top-ranked clusters, with the number of clusters included determined by a facet threshold that controls the response length. Each cluster is summarized independently into a single sentence, with word limits enforced to maintain brevity [6]. The summaries are concise, containing only the information provided by the nuggets. The previous steps ensure that the most relevant information from the retrieved passages is synthesized.

3.5 Improving Response Fluency

Since the response consists of independent summaries, there may be issues with fluency and coherence. To improve this, we add a final rephrasing step using an LLM. The LLM is prompted to refine the response without altering or adding any new content, ensuring that the information remains accurate while improving the overall flow of the text.

4 Submitted Runs

This section contains a high-level description of our submitted runs. which are summarized in Table 1. The differences between runs lie mainly in the number of retrieved passages used and the length of the generated response.

baseline_top_5 This run is considered as our baseline. It summarizes the top 5 candidates from the retrieval baseline provided by the organizers with GPT-4 without further training. Generation is performed on the response level. The response is limited to 3 sentences. For source attribution, each sentence in the response is accompanied by the list of document IDs of all top 5 passages.

ginger_top_5 This run uses the standard setup of GINGER to generate responses. Generation is performed on sentence level. It takes the top 5 candidates from the retrieval baseline provided by the organizers as input. The response is limited to 3 sentences. For source attribution, each generated sentence in the response is accompanied by the same set of all the document IDs that contain nuggets from the top 3 clusters.

ginger-fluency_top_5 This run uses GINGER without the last component (Improving Response Fluency) to generate responses. Generation is performed on sentence level. It takes the top 5 candidates from the retrieval baseline provided by the organizers as input. The response is limited to 3 sentences. For source attribution, each sentence in the response is accompanied by the list of document IDs for nuggets from the corresponding cluster.

ginger-fluency_top_10 This run uses GINGER without the last component (Improving Response Fluency) to generate responses. Generation is performed on sentence level. It takes the top 10 candidates from the retrieval baseline provided by the organizers as input. The response is limited to 400 words (maximum response length allowed by the organizers). For source attribution, each sentence in the response is accompanied by the list of document IDs for nuggets from the corresponding cluster.

ginger-fluency_top_20 This run is similar to **ginger-fluency_top_10** with the 20 top candidates from the retrieval baseline provided by the organizers used as input.

5 Results

This section presents the performance of our runs on the TREC RAG'24 dataset. At the time of submission, we have results in terms of V_strict score obtained with the AutoNuggetizer evaluation framework provided by the organizers [21]. The results of all the runs submitted to evaluation for all 301 topics provided in the track are presented in Table 2. As of now, response fluency scores have not been provided. We have obtained response support evaluations for two of our five systems, but the corresponding evaluation metrics are still unavailable.

6 Conclusion

This paper has described our participation in the TREC 2024 RAG track. Our submitted runs have relied on GINGER—a modular response generation pipeline that grounds responses in specific facts from retrieved sources and controls information coverage. We have manipulated the number of input passages and

Table 2. Official results of our submissions for the AG task under AutoNuggetizer evaluation. Results are cited verbatim from Pradeep et al. [21] (Table 8).

System variant	**V_strict**
ginger-fluency_top_20	**0.427**
ginger-fluency_top_10	0.369
baseline_top_5	0.247
ginger-fluency_top_5	0.213
ginger_top_5	0.211

response length to investigate their effect on system performance. Partial results available at the time of submission indicate that GINGER is the top performing AG system under AutoNuggetizer evaluation. Evaluation in terms of response fluency is not available at the time of writing—we look forward to receiving and analyzing these results in future work.

Bibliography

[1] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. V. D. Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. D. L. Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. Rae, E. Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, ICML '22, pages 2206–2240, 2022.

[2] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, and F. Silvestri. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 719–729, 2024.

[3] T. Falke, L. F. R. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 2214–2220, 2019.

[4] J. Fang, Z. Meng, and C. Macdonald. TRACE the evidence: Constructing knowledge-grounded reasoning chains for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, EMNLP '24, pages 8472–8494, 2024.

[5] L. Gienapp, H. Scells, N. Deckers, J. Bevendorff, S. Wang, J. Kiesel, S. Syed, M. Fröbe, G. Zuccon, B. Stein, M. Hagen, and M. Potthast. Evaluating generative ad hoc information retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pages 1916–1929, 2024.

[6] T. Goyal, J. J. Li, and G. Durrett. News summarization and evaluation in the era of GPT-3. *arXiv*, cs.CL/2209.12356, 2023.

[7] M. Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv*, cs.CL/2203.05794, 2022.

[8] Y. Huang and J. Huang. A survey on retrieval-augmented text generation for large language models. *arXiv*, cs.IR/2404.10981, 2024.

[9] G. Izacard and E. Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, EACL '21, pages 874–880, 2021.

[10] H. Jiang, Q. Wu, X. Luo, D. Li, C.-Y. Lin, Y. Yang, and L. Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '24, pages 1658–1677, 2024.

[11] B. Koopman and G. Zuccon. Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, EMNLP '23, pages 15012–15022, 2023.

[12] F. Ladhak, E. Durmus, H. He, C. Cardie, and K. McKeown. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '22, pages 1410–1421, 2022.

[13] N. Liu, T. Zhang, and P. Liang. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, EMNLP '23, pages 7001–7025, 2023.

[14] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[15] Y. Liu, A. Fabbri, P. Liu, Y. Zhao, L. Nan, R. Han, S. Han, S. Joty, C.-S. Wu, C. Xiong, and D. Radev. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 4140–4170, 2023.

[16] G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz, E. Grave, Y. LeCun, and T. Scialom. Augmented language models: a survey. *arXiv*, cs.CL/2302.07842, 2023.

[17] D. Muhlgay, O. Ram, I. Magar, Y. Levine, N. Ratner, Y. Belinkov, O. Abend, K. Leyton-Brown, A. Shashua, and Y. Shoham. Generating benchmarks for factuality evaluation of language models. In *Conference of the European Chapter of the Association for Computational Linguistics*, EACL '23, 2023.

[18] A. Nenkova, R. Passonneau, and K. McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2):4–es, 2007.

[19] V. Pavlu, S. Rajput, P. B. Golbus, and J. A. Aslam. IR system evaluation using nugget-based test collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 393–402, 2012.

[20] R. Pradeep, R. Nogueira, and J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv*, cs.IR/2101.05667, 2021.

[21] R. Pradeep, N. Thakur, S. Upadhyay, D. Campos, N. Craswell, and J. Lin. Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework. *arXiv*, cs.IR/2411.09607, 2024.

[22] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

[23] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, and M. Collins. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2021.

[24] R. Ren, Y. Wang, Y. Qu, W. X. Zhao, J. Liu, H. Tian, H. Wu, J.-R. Wen, and H. Wang. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv*, cs.CL/2307.11019, 2023.

[25] T. Sakai. SWAN: A generic framework for auditing textual conversational systems. *arXiv*, cs.IR/2305.08290, 2023.

[26] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, 2024.

[27] M. Subbiah, S. Zhang, L. B. Chilton, and K. McKeown. Reading subtext: Evaluating large language models on short story summarization with writers. *arXiv*, cs.CL/2403.01061, 2024.

[28] L. Tang, T. Goyal, A. Fabbri, P. Laban, J. Xu, S. Yavuz, W. Kryscinski, J. Rousseau, and G. Durrett. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 11626–11644, 2023.