

GINGER: Grounded Information Nugget-Based Generation of Responses

Weronika Łajewska
University of Stavanger
Stavanger, Norway
weronika.lajewska@uis.no

Krisztian Balog
University of Stavanger
Stavanger, Norway
krisztian.balog@uis.no

Abstract

Retrieval-augmented generation (RAG) faces challenges related to factual correctness, source attribution, and response completeness. To address them, we propose a modular pipeline for grounded response generation that operates on information nuggets—minimal, atomic units of relevant information extracted from retrieved documents. The multistage pipeline encompasses nugget detection, clustering, ranking, top cluster summarization, and fluency enhancement. It guarantees grounding in specific facts, facilitates source attribution, and ensures maximum information inclusion within length constraints. Experiments on the TREC RAG’24 dataset, using the AutoNuggetizer framework, demonstrate that GINGER achieves state-of-the-art performance on this benchmark.

CCS Concepts

- **Computing methodologies** → **Natural language generation;**
- **Information systems** → **Information extraction.**

Keywords

Retrieval-augmented generation; Grounding; Source attribution

ACM Reference Format:

Weronika Łajewska and Krisztian Balog. 2025. GINGER: Grounded Information Nugget-Based Generation of Responses. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3726302.3730166>

1 Introduction

We observe an increasing reliance on conversational assistants, such as ChatGPT, for a variety of information needs, involving open-ended questions, indirect answers that require inference, and complex queries with partial answers spread over several passages [2, 6, 46]. Generating responses poses challenges related to factual correctness [15, 17, 39], source attribution [32], information verifiability [20], consistency, and coverage [8]. Recently proposed retrieval-augmented generation (RAG) models aim to address these issues by conditioning the generative processes on retrieved sources, yet ensuring response transparency and source attribution remain open challenges [8, 12, 19]. Current commercial generative search

engines appear informative but often contain unsupported statements and inaccurate citations, further highlighting the difficulty of grounding responses [20]. Injecting evidence in LLM prompts can reduce hallucinations [17]. However, redundant information and overly long contexts can lead to the “lost in the middle” problem, where models struggle to retrieve relevant information from the middle of long contexts [21]. Consequently, a post-retrieval refinement step is recommended to retain only essential information and prevent key details from being diluted [7].

To address the above challenges, we present a modular pipeline for Grounded Information Nugget-Based Generation of Responses (GINGER). The main novelty of our approach compared to existing RAG approaches [31, 36] is that it operates on *information nuggets*, which are atomic units of relevant information [27]. Given a set of passages retrieved in response to a user query, our approach identifies information nuggets in top passages, clusters them by query facet, ranks clusters by relevance, summarizes the top ones, and refines the response for fluency and coherence. GINGER uniquely models query facets to maximize the unique information provided in the answer. This approach can significantly improve the user experience by ensuring the grounding of the final response in the source passages and enabling easy verifiability of source attribution.

We evaluate our system on the augmented generation task within the recently launched Retrieval-Augmented Generation track at Text Retrieval Conference (TREC RAG’24). We report on official evaluation results based on AutoNuggetizer [30] framework piloted in this track for variants of our pipeline submitted to TREC RAG’24. For other system configurations, we utilize a reimplementations of this evaluation framework (using the same prompts) to score responses. The results demonstrate that GINGER achieves performance on par with top-performing TREC submissions and significantly outperforms two strong baselines: (1) a grounded generation for open-domain QA [33] and (2) chain-of-thought generation with in-context learning demonstration [43]. We show that GINGER successfully filters out and synthesizes information from relevant sources, and its performance increases with the amount of information provided to the system. Finally, an ablation study reveals that the core strength of GINGER lies in its operation on information nuggets, rather than the individual performance of its components.

In summary, the main contributions of this work are twofold: (1) A novel modular response generation pipeline that operates on information nuggets, enabling precise grounding and facilitating source verification. (2) Extensive experimental evaluation of the proposed approach on the TREC RAG’24 augmented generation task using the AutoNuggetizer framework. All the resources developed in this paper, along with additional results and analysis are made available at: <https://github.com/iai-group/ginger-response-generation>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR ’25, Padua, Italy

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1592-1/2025/07

<https://doi.org/10.1145/3726302.3730166>

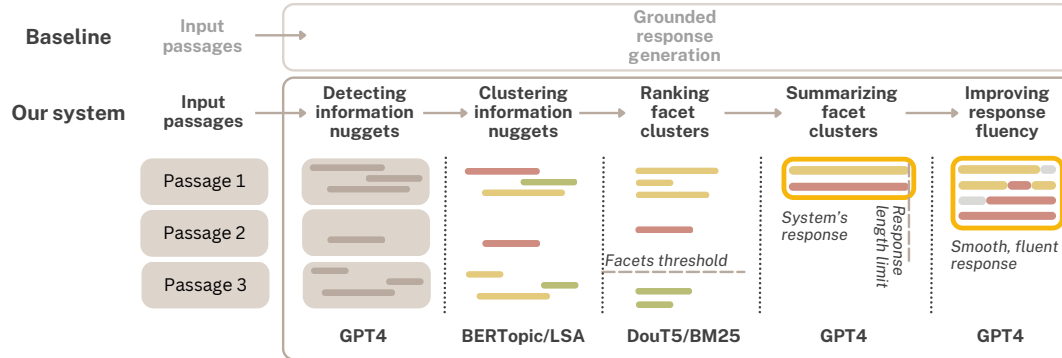


Figure 1: High-level overview of our nugget-based response generation pipeline (GINGER).

2 Related Work

Retrieval-Augmented Generation. Unlike traditional search engines that return a ranked list of documents, RAG systems generate a single, coherent response by synthesizing diverse perspectives from retrieved sources, conditioning the generation process on retrieved evidence either through prompt injection or direct attention during inference [8, 13, 23, 31, 36]. RAG approaches include conditioning models on retrieved document chunks [3] or combining parametric and non-parametric memory [12, 19]. In RAG, retrieving documents based on partial relevance can introduce irrelevant or loosely related text, which may hinder the quality of the generated response [4]. Also, generation performance can significantly drop when relevant information is not at the beginning or end of long contexts [21]. These findings suggest that basic retrieve-then-generate pipelines may not effectively reduce hallucinations [17]. Several methods have been proposed for context curation for LLMs, such as removing unimportant tokens [16], information aggregation [47], or training information extractors and condensers [44, 45]. Generative responses often exhibit high fluency and perceived utility but frequently contain unsupported statements or inaccurate citations [20]. Despite advancements in LLMs, abstractive summaries still suffer from hallucinations and factual errors [5, 15, 17, 18, 39, 40]. Source attribution and verifiability, which require in-line citations to support generated statements, are key to addressing these challenges [20, 32, 35]. To ensure high citation precision while maintaining fluency, we propose to extract and group atomic statements (referred to as “atomic/semantic content units” [22, 25] or “information nugget” in traditional IR [27, 34]) from sources and summarize them with LLMs.

TREC RAG’24. The RAG track at TREC has been launched in 2024 with a focus on combining retrieval methods for finding relevant information within large corpora with LLMs to enhance the ability of systems to produce relevant, accurate, and contextually appropriate content [30]. The track is divided into three tasks: Retrieval (R), which involves ranking and retrieving the most relevant segments from the corpus; Augmented Generation (AG), which requires generating RAG answers using top-k relevant segments from a baseline retrieval system provided by organizers; and Retrieval-Augmented Generation (RAG), where participants generate RAG answers with attributions using their retrieval system and chunking technique. Our work focuses on the augmented generation task,

similar to in-context retrieval augmented language models without modifying model weights [24, 31].

A trend toward retrieval context curation is evident in TREC RAG’24 submissions. A common approach to AG task involves generating responses based on the top 20 documents, often in a single step using proprietary models, with an optional post-processing phase. Several submissions adopt a multi-step approach, like segment clustering followed by extracting, combining, and condensing relevant information. Similarly, some approaches emphasize verifying key facts across multiple documents, eliminating redundant content, prioritizing facts by relevance, and enhancing clarity and coherence. In this work, we take this a step further by not only curating the LLM context but also decomposing the response generation process to mitigate the negative effects of irrelevant and redundant information by operating on atomic pieces of information.

3 Nugget-Based Multi-Step Response Generation

We present a novel method for generating grounded conversational responses by operating on information nuggets. It explicitly models various facets of the query based on the retrieved information and generates a concise response that adheres to length constraints.

Generating grounded responses is a multistage process, that includes: (1) detecting information nuggets in top relevant passages, (2) clustering detected nuggets, corresponding to different facets of the query, (3) ranking the clusters with respect to their relevance to the query, (4) summarizing the top-ranked clusters to be included in a final response, and (5) refining the response to improve its fluency and coherence (see Fig. 1). Steps 1-3 aim at curating the context for response generation, while Steps 4-5 focus on synthesizing and refining the response. By operating on information nuggets in all intermediate components of the pipeline we ensure the grounding of the final response in the source passages, ensuring that all information in the final response is entailed by the source [5].

Detecting Information Nuggets We automatically detect information nuggets by prompting an LLM¹ to annotate input passages with information nuggets containing the key information that answer the query. Specifically, it is instructed to copy the passage’s text and place the annotated nuggets between specific tags, without modifying the passage content or adding any extra symbols.

¹The prompts used by different components of the system are made available in the online appendix.

Table 1: Response evaluation with AutoNuggetizer. TREC scores are provided for TREC RAG'24 AG runs. The remaining scores are based on the framework reimplementaion.

Method	V_strict				
	TREC	GPT4o	Gemini	Claude	avg LLM
baseline-top5	0.247	0.332	0.468	0.525	0.442
baseline_CoT-top5	—	0.332	0.452	0.500	0.428
Webis	0.357	—	—	—	—
TREMA	0.261	—	—	—	—
GINGER-top20 wo/ rewriting	0.427	0.500	0.543	0.659	0.568
GINGER-top10 wo/ rewriting	0.369	0.423	0.502	0.582	0.502
GINGER-top5 wo/ rewriting	0.213	0.263	0.392	0.431	0.362
GINGER-top5	0.211	0.279	0.400	0.451	0.377

Clustering Information Nuggets Next, we cluster the detected information nuggets with respect to different facets of the query topic. This clustering step serves two purposes. First, it addresses the problem of information redundancy, stemming from the fact that nuggets and their variants can appear in multiple documents in different forms but still convey the same information [27]. Second, by clustering redundant information nuggets, we attempt to increase the density of the generated information [1]. Nugget clustering is challenging due to the semantic closeness of nuggets within the same topic. We address this by employing a neural topic modeling technique, BERTopic [10], and adjusting its sensitivity on the validation partition of the dataset. Ideally, information nuggets in each cluster represent specific facets of the answer to the query.

Ranking Facet Clusters This step in the pipeline is responsible for the ranking of facet clusters with respect to the input query to determine which clusters are most important and should be prioritized for inclusion in the response, and which may be skipped [7, 21]. Given the relatively low number of facet clusters we observe in practice, we can employ more expensive reranking techniques relying on pairwise comparisons to maximize effectiveness. Specifically, we employ pairwise reranking using duoT5 [29] by joining the nuggets in clusters and treating them as individual passages.

Summarizing Facet Clusters The response is made up of the summaries of the top n clusters, where n is the facet threshold that controls the desired response length and may be adjusted based on the information need, task context, or user preferences. Each cluster of information nuggets is summarized independently as a single sentence with the maximum number of words specified in the prompt, to stop the LLM from generating very long sentences [9]. We follow the prompt design used for short story summarization [37] to generate summaries that are concise and only contain the information provided. Previous steps in the pipeline ensure that the most relevant information from retrieved passages is synthesized and that the generated summaries are attributed to the sources. This allows summarization to operate in a shorter but more relevant context.

Improving Response Fluency Our modular approach results in a response that is a concatenation of independent summaries of facet clusters, that may lack fluency and consistency. To mitigate this shortcoming, we include an additional step to rephrase the generated response with the help of an LLM. The LLM is prompted not to modify the provided information, nor include any additional content.

Table 2: AutoNuggetizer evaluation of responses generated with different variants of GINGER without the fluency enhancement step and with top 20 passages provided as input.

Clustering	Ranking	V_strict (avg LLM)
BERTopic	DuoT5	0.568
BERTopic	BM25	0.554
LSA	DuoT5	0.521
LSA	BM25	0.551

4 Experimental Setup

Dataset We base our evaluation on the Augmented Generation task of TREC RAG'24 [30] comprising 301 queries. This work focuses on the problem of response generation and assumes a ranked list of passages to be provided, along with the query, as input. In all experiments, we utilize the top relevant passages (from MS MARCO V2.1 segment collection) from a fixed list of 100 retrieved results provided by the organizers. This setup represents the real-world case with the top passages retrieved by a competitive passage ranker. To account for the amount of input information, we consider three sizes of input rankings containing 5, 10, or 20 passages.

TREC Evaluation We use the AutoNuggetizer framework proposed for RAG evaluation and validated during TREC RAG'24 [30]. AutoNuggetizer comprises two steps: nugget creation and nugget assignment. In nugget creation, nuggets are formulated based on relevant documents and classified as either “vital” or “okay” [42]. The second step, nugget assignment, involves assessing whether a system’s response contains specific nuggets from the answer key. The score V_{strict} for system’s response is defined as follows:

$$V_{strict} = \sum_i ss_i^v / |n^v|$$

where n^v represents the subset of the vital nuggets; ss_i^v is 1 if the response supports the i -th nugget and is 0 otherwise. The score of a system is the mean of the scores across all queries.

LLM-based Evaluation We reimplemented the AutoNuggetizer evaluation framework to compute the V_{strict} measure, adhering to the original prompts from Pradeep et al. [30]. To make the evaluation more robust and mitigate any potential bias of using the same LLM for response generation and judging, we use the average of the scores generated by three different LLMs (*gpt-4o-2024-08-06*, *claude-3-5-haiku-20241022*, *gemini-1.5-flash*) as a final score (as opposed to original TREC RAG scores which are based solely on GPT-4o). We validate our implementation of the evaluation framework by comparing the results for our submitted runs with the official numbers reported by track organizers [30]. Even though our V_{strict} scores are higher than the scores reported in the TREC RAG track, the relative ordering of the systems remains the same.

Baseline To compare our method with models using external knowledge in generation, we focus on grounded response generation with a fixed retriever. This excludes standard end-to-end RAG models [11, 14, 19] and LLM approaches relying solely on internal knowledge [38]. In pursuit of a strong baseline, we explored grounded text generation, Chain-of-Thought (CoT) prompting, and open-domain QA models using samples from the TREC

CAsT'22 datasets [26]. We found that a pre-trained model proposed for grounded text generation in dialogues relying on external knowledge [28] tended to copy directly from the source text, showing extractive behavior. Using an off-the-shelf LLM proposed for a retrieval-augmented setting for open-domain QA showed notably better coherence and naturalness without additional training [24, 31, 33]. Therefore, our first baseline is OpenAI's GPT-4 model, prompted similarly to retrieval-augmented QA [33] using the top 5 relevant passages (*baseline-top5*). The length of the generated summary is limited to around 100 words (3 sentences), controlled via the task model prompt [9]. The second baseline uses GPT-4 with CoT prompting [43] and one ICL demonstration created manually based on TREC CAsT'22 dataset [26] (*baseline_CoT-top5*).

Implementation Details The modular design of GINGER allows for independent implementation of individual pipeline components. The detection of information nuggets is performed with GPT-4, with the query and passage as input. Information nuggets clustering is based on BERTopic, with parameters set experimentally on samples from TREC CaST'22.² The ranking of information nugget clusters is done using a duoT5 model,³ implemented based on the HuggingFace transformers library. The top 3 ranked information nuggets clusters are passed to query biased summarization [41] performed with GPT-4. The length of each cluster summary is limited to one sentence and around 35 words and specified in the prompt [9]; the output length limit is specified in the model parameters.

5 Results and Analysis

We seek to measure GINGER's capability to generate high-quality responses. Table 1 presents results for the two baseline systems, as well as for our nugget-based response generation pipeline (with or without response fluency improvement) using different numbers of retrieved passages. Responses from the top 5 passages are limited to 3 sentences, while those from the top 10 or 20 passages have a 400-word limit. TREC scores are reported for system configurations where we have official evaluation results.

Does GINGER improve response generation performance over the baselines? The best-performing variant of our system outperforms both baseline approaches. Even prompting the model to divide the task into several steps using CoT and providing a ground-truth response as an example does not help in the response generation process. This implies that given the complexity of the queries and the amount of input context to be taken into account, the LLM needs further guidance to generate an accurate answer.

How does GINGER perform in comparison to other systems submitted to TREC RAG'24? According to the initial results provided by TREC RAG organizers [30], our best system (*GINGER-top20 wo/ rewriting*) was the top performing AG submission. Even though several other systems decompose response generation into multi-step process, GINGER shows higher performance. For reference, we include two other competitive AG submissions in the results table (*Webis.webis-ag-run0-taskrag* and *TREMA-UNH.Enhanced_Iterative_Fact_Refinement_and_Prioritization*).

²If fewer than four information nuggets are identified in the top n passages, we skip clustering; instead, we treat each nugget as an independent cluster and proceed with standard ranking and summarization.

³<https://huggingface.co/castorini/duot5-base-msmarco>

How does the amount of input information affect GINGER's performance? By operating on information nuggets throughout the pipeline, GINGER grounds responses in specific facts and synthesizes information from the top passages. Responses generated from more passages are of higher quality, indicating that the additional context is indeed utilized in the system output (*GINGER-top10 wo/ rewriting* vs. *GINGER-top20 wo/ rewriting*). This suggests that splitting response generation into several independent steps and increasing information granularity mitigates the information loss to which LLMs are prone. Even with potential information redundancy, responses of the same length limit score higher with more input context, implying that they include more vital facts.

How much do individual pipeline components contribute to overall system performance? GINGER's modular architecture allows us to evaluate individual components and their impact on the final responses. We experiment with traditional Latent Semantic Analysis (LSA) for nugget clustering and BM25 for cluster ranking, as alternatives. Following our best-performing setup, we provide 20 relevant passages as input and limit responses to 400 words. Table 2 presents the results of this ablation study. We find that modifying specific components does not strongly impact response quality. Therefore, we conclude that operating on information nuggets, rather than the effectiveness of individual components, is the primary factor contributing to GINGER's performance. This highlights the importance of higher information granularity and reduced redundancy in response generation.

Does LLM-based fluency enhancement reduce grounding? Since the responses returned by our method are concatenations of independent facet cluster summaries, the final response may lack fluency and coherence. However, the difference between the responses generated with the fluency improvement and without this step (*GINGER-top5* vs. *GINGER-top5 wo/ rewriting*) is not significant. This indicates that LLMs can be used to refine response fluency without sacrificing quality or grounding.

6 Conclusions

We have introduced an approach for generating grounded responses that uses information nuggets from the top retrieved passages and employs a multi-stage process (clustering, reranking, summarization, fluency enhancement) to generate concise, information-rich, non-redundant text. Our approach offers several key advantages: maximizing the information contained within response length limits and ensuring grounding in input context. Automatic evaluation with AutoNuggetizer shows that GINGER outperforms two strong baselines and achieves top performance on the TREC RAG'24 Augmented Generation task. Our modular method lays a strong foundation for future research on constraint-based response generation. This could include investigating the impact of various response lengths (e.g., based on user preferences), controlling response completeness, or developing strategies to manage redundancy in multi-turn conversations by considering previously discussed facets.

Acknowledgments

This research was supported by the Norwegian Research Center for AI Innovation, NorwAI (Research Council of Norway, nr. 309834).

References

- [1] Griffin Adams, Alex Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting. *Proceedings of the 4th New Frontiers in Summarization Workshop* (2023), 68–74.
- [2] Valeriia Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. 2023. WikiHowQA: A Comprehensive Benchmark for Multi-Document Non-Factoid Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (ACL '23). 5291–5314.
- [3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassiere, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *Proceedings of the 39th International Conference on Machine Learning (ICML '22)*. 2206–2240.
- [4] Florin Cucenasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonello, and Fabrizio Silvestri. 2024. 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)*. 719–729.
- [5] Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. 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)*. 2214–2220.
- [6] Matteo Gabburo, Nicolaas Paul Jedema, Siddhant Garg, Leonardo F. R. Ribeiro, and Alessandro Moschitti. 2024. Measuring Retrieval Complexity in Question Answering Systems. In *Findings of the Association for Computational Linguistics: ACL 2024 (ACL '24)*. 14636–14650.
- [7] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]
- [8] Lukas Gienapp, Harrison Scells, Niklas Deckers, Janek Bevendorff, Shuai Wang, Johannes Kiesel, Shahbaz Syed, Maik Fröbe, Guido Zuccon, Benno Stein, Matthias Hagen, and Martin Potthast. 2024. Evaluating Generative Ad Hoc Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. 1916–1929.
- [9] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2023. News Summarization and Evaluation in the Era of GPT-3. arXiv:2209.12356 [cs.CL]
- [10] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. (2022). arXiv:2203.05794 [cs.CL]
- [11] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning (ICML '20)*. 3929–3938.
- [12] Yizheng Huang and Jimmy Huang. 2024. A Survey on Retrieval-Augmented Text Generation for Large Language Models. arXiv:2203.05794 [cs.CL]
- [13] Gautier Izacard and Edouard Grave. 2021. 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)*. 874–880.
- [14] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dvivedi-Yu, Armand Joulin, and Sebastian Riedel. 2023. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *Journal of Machine Learning Research* 24 (2023), 1–43.
- [15] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12 (2023), 1–38.
- [16] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. 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)* (ALC '24). 1658–1677.
- [17] Bevan Koopman and Guido Zuccon. 2023. Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness. In *Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP '23)*. 15012–15022.
- [18] Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. 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). 1410–1421.
- [19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. 9459–9474.
- [20] Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating Verifiability in Generative Search Engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP '23)*. 7001–7025.
- [21] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.
- [22] Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. 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). 4140–4170.
- [23] Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dvivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented Language Models: a Survey. arXiv:2302.07842 [cs.CL]
- [24] Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. Generating Benchmarks for Factuality Evaluation of Language Models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL '23)*.
- [25] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing* 4, 2 (2007), 4–es.
- [26] Paul Owoicho, Jeffrey Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R Trippas, and Svitlana Vakulenko. 2022. TREC CAsT 2022: Going Beyond User Ask and System Retrieve with Initiative and Response Generation. In *The Thirty-First Text Retrieval Conference Proceedings (TREC '22)*.
- [27] Virgil Pavlu, Shahzad Rajput, Peter B. Golbus, and Javed A. Aslam. 2012. IR system evaluation using nugget-based test collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*. 393–402.
- [28] Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. GODEL: Large-Scale Pre-Training for Goal-Directed Dialog. arXiv:2206.11309 [cs.CL]
- [29] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. arXiv:2101.05667 [cs.LR]
- [30] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial Nugget Evaluation Results for the TREC 2024 RAG Track with the AutoNuggetter Framework. arXiv:2411.09607 [cs.LR]
- [31] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.
- [32] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, and Michael Collins. 2021. Measuring Attribution in Natural Language Generation Models. *Computational Linguistics* 49, 4 (2021), 777–840.
- [33] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation. arXiv:2307.11019 [cs.CL]
- [34] Tetsuya Sakai. 2023. SWAN: A Generic Framework for Auditing Textual Conversational Systems. arXiv:2305.08290 [cs.LR]
- [35] Tal Schuster, Adam D. Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William W. Cohen, and Donald Metzler. 2023. SEMQA: Semi-Extractive Multi-Source Question Answering. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1 (2023), 1363–1381.
- [36] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. 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)* (NAACL-HLT '24). 8371–8384.
- [37] Melanie Subbiah, Sean Zhang, Lydia B. Chilton, and Kathleen McKeown. 2024. Reading Subtext: Evaluating Large Language Models on Short Story Summarization with Writers. arXiv:2403.01061 [cs.CL]
- [38] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-Augmented Language Models. In *The Eleventh International Conference on Learning Representations (ICLR '23)*.
- [39] Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. 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). 11626–11644.
- [40] Xiangru Tang, Alexander Fabbri, Haoran Li, Ziming Mao, Griffin Adams, Borui Wang, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. Investigating Crowdsourcing Protocols for Evaluating the Factual Consistency of Summaries. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL-HLT '22). 5680–5692.
- [41] Anastasios Tombros and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '98)*. 2–10.
- [42] Ellen M. Voorhees. 2004. Overview of the TREC 2003 Question Answering Track. *NIST* (2004).
- [43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*.
- [44] Fanguan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation. arXiv:2310.04408 [cs.CL]
- [45] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting Black-Box Large Language Models for Retrieval Question Answering via Pluggable Reward-Driven Contextual Adapter. In *Findings of the Association for Computational Linguistics: EMNLP 2023 (EMNLP '23)*. 5364–5375.
- [46] Hamed Zamani, Johanne R. Trippas, Jeff Dalton, and Filip Radlinski. 2023. Conversational Information Seeking. *Foundations and Trends in Information Retrieval* 17, 3-4 (2023), 244–456.
- [47] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arik. 2024. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. arXiv:2406.02818 [cs.CL]