



Multi-Step Semantic Reasoning in Generative Retrieval

Steven Dong^(✉), Yubao Tang, and Maarten de Rijke

University of Amsterdam, Amsterdam, The Netherlands
steven.dong@student.uva.nl, {y.tang3,m.derijke}@uva.nl

Abstract. Generative retrieval (GR) models encode a corpus within model parameters and generate relevant document identifiers directly for a given query. While this paradigm shows promise in retrieval tasks, existing GR models struggle with complex queries in numerical contexts, such as those involving semantic reasoning over financial reports, due to limited reasoning capabilities. This limitation leads to suboptimal retrieval accuracy and hinders practical applicability. We propose ReasonGR, a framework designed to enhance multi-step semantic reasoning in numerical contexts within GR. ReasonGR employs a structured prompting strategy combining task-specific instructions with step-wise reasoning guidance to better address complex retrieval queries. Additionally, it integrates a reasoning-focused adaptation module to improve the learning of reasoning-related parameters. Experiments on the FinQA dataset, which contains financial queries over complex documents, demonstrate that ReasonGR improves retrieval accuracy and consistency, indicating its potential for advancing GR models in reasoning-intensive retrieval scenarios.

Keywords: Generative retrieval · Financial information retrieval · Semantic reasoning · Chain-of-thought prompting

1 Introduction

Generative retrieval (GR) has emerged as a promising paradigm in information retrieval. Unlike traditional dense retrieval methods that rely on explicit index-retrieve pipelines [10, 17, 26], GR models learn to encode the entire corpus within their parameters and directly generate relevant document identifiers (docids) in response to queries [35]. This approach reduces memory storage requirements and enables more integrated, end-to-end retrieval.

While GR has demonstrated notable results in tasks such as question answering and fact verification [3–5, 18], it still faces challenges in retrieval tasks that require complex inference or aggregation of dispersed evidence, exemplified by financial queries over intricate reports in the FinQA dataset [6]. We refer to this challenge as multi-step semantic reasoning in numerical contexts, which focuses on interpreting relevant numerical information in complex documents [11].



Fig. 1. ReasonGR performing multi-step semantic reasoning on a FinQA query. The model extracts key info and locates relevant report sections to generate the docid, formed by the company name and report year.

Current GR methods struggle to effectively address these tasks, mainly due to two factors: (i) GR models often treat retrieval as a black-box operation, encoding it implicitly in a single forward pass without explicit, interpretable reasoning steps [28]. (ii) Most models are trained with maximum likelihood estimation (MLE) [13] on surface-level query-document pairs, which does not explicitly foster multi-step or logical reasoning capabilities [1, 19].

To address these challenges, we propose ReasonGR, a framework designed to enhance multi-step semantic reasoning capabilities of GR models in numerical contexts. ReasonGR employs structured prompting that combines task-specific instructions with chain-structured reasoning prompts to guide step-by-step semantic inference. Moreover, we introduce reasoning adapter modules that enable the model to better capture reasoning-related parameters tailored to the task.

We evaluate ReasonGR on the FinQA dataset [6], which contains financial queries over complex documents. Our experiments show that ReasonGR improves retrieval accuracy over baseline methods. We also report that the introduced adaptation modules yield efficiency benefits in training.

2 Related Work

Traditional Information Retrieval (IR). Traditional IR systems typically follow a “index-retrieve” pipeline [22]. Early approaches such as BM25 [25] rely on exact term matching, while dense retrieval methods [17] use neural networks to capture semantic similarity by encoding queries and documents into dense vectors.

Generative Retrieval. GR learns to index the corpus and then directly generate docids given a query [28, 32, 33, 35]. This paradigm internalizes corpus information in model parameters, enabling end-to-end retrieval without an external index. A number of related studies have advanced this line of work [21, 23, 27, 29–31, 34]. While these methods achieve promising retrieval performance, their ability to perform complex reasoning, particularly multi-step semantic reasoning required by tasks such as financial document retrieval, remains limited [16]. Addressing this reasoning gap is essential for improving the applicability of GR in such domain.

3 Methodology

3.1 Task Definition

Given a corpus $\mathcal{D} = \{d_1, \dots, d_N\}$ with size N , and a set of queries $\mathcal{Q} = \{q_1, \dots, q_M\}$ with size M , each document d_i is associated with a unique docid j_i , forming the set \mathcal{J} . The task is to generate the relevant docid j_i for a given query q_i using a generative model P_θ , which models the conditional probability, $P_\theta(j_i|q_i) = \prod_{p=0}^{|j_i|} P_\theta(j_i^p|q_i, j_i^{<p})$, where j_i^p is the p -th token of the docid sequence j_i , and $j_i^{<p}$ denotes all previously generated tokens.

3.2 Model Architecture

Our model adopts a transformer-based encoder-decoder architecture, following the GR paradigm [35]. The backbone encodes the input query and autoregressively decodes the corresponding docid.

To learn reasoning-specific capabilities, we introduce Low-rank adaptation (LoRA) modules [15] into the backbone. LoRA freezes the original model weights θ and injects trainable low-rank matrices $\Delta\theta$ to adapt the model to the reasoning-intensive retrieval task: $\theta' = \theta + \Delta\theta$, where $\Delta\theta = AB^\top$, $A \in \mathbb{R}^{d \times r}$, $B \in \mathbb{R}^{k \times r}$, with rank $r \ll \min(d, k)$. This decomposition allows efficient learning of new parameters dedicated to reasoning, while keeping the number of trainable parameters small. Furthermore, we apply QLoRA [9] to quantize the frozen backbone weights to 4-bit precision, reducing memory and computational costs with minimal performance loss. LoRA and QLoRA are used jointly in our framework.

3.3 Reasoning-Guided Training Framework

Beyond naive optimization with MLE [35], our framework improves the model’s reasoning ability through both input construction and training objectives.

Reasoning-enhanced Prompting. To help the model to structure its generation in a way that encourages multi-step logical inference, we design our prompts by combining two sets of templates: task-specific instructions and Chain-of-Thought (CoT) additions, as shown in Fig. 1. During training, one of five task templates is randomly selected and combined with one of five CoT instructions to form diverse and informative prompts. The five task templates are:

- Answer the query with a document ID.*
- Generate the document ID that answers the question.*
- Based on the question, predict the document ID.*
- Retrieve a document ID that fits the query.*
- Using the question, find the document ID.*

The CoT additions encourage stepwise reasoning, examples of which include:

Use step-by-step reasoning.
You need to explain your answer.
Think this through carefully.
Let's think step-by-step.
Explain your reasoning before answering.

These instructions are appended to the task templates to form full prompts. For CoT prompts, the expected output includes intermediate reasoning steps before the final answer.

Training Tasks. To train the retrieval model, we have two tasks: (i) To memorize the corpus, we learn the mapping from documents to docids using the MLE loss [5, 35]. Since these documents often contain tables with numerical data, which are crucial for retrieval generation, we also incorporate learning from the tables by splitting them and attaching each entry with its corresponding column and row headers. (ii) To learn relevance, the model takes queries as input and generates correct multi-step reasoning traces along with the corresponding docids as output. To better approximate the distribution of real-world queries, training data is augmented with multiple pseudo-queries generated per document using a query generation model [37].

Adaptive Penalty Scaling Loss. We design a custom loss function that provides fine-grained supervision by incorporating multiple evaluation signals, including exact match (EM), partial match (PM), set match (SM), and structural similarity (S-Score). The penalty factor scales the cross-entropy loss to emphasize errors on incorrect or incomplete docid predictions proportionally, $\text{Loss}(\hat{y}, y) = \text{CE}(\hat{y}, y) \times P_{\text{loss}}(\hat{y}, y)$, where P_{loss} is a weighted combination of penalties derived from these metrics. y and \hat{y} are the model's predicted sequence, and corresponding ground truth sequence, respectively.

4 Experimental Setup

Dataset and Metrics. We conduct experiments on the FinQA dataset [6], which contains financial queries requiring complex multi-step numerical reasoning over financial reports and associated tabular data. We use each document's filename as a unique docid. The dataset includes 8,281 samples, partitioned into training, validation, and test sets with splits of 75%, 10%, and 15%, respectively. Performance is evaluated using several standard metrics: (i) Exact Match accuracy (EM), measuring complete sequence correctness; (ii) Part Match accuracy (PM), accounting for ordered token-level correctness; (iii) Set Match accuracy (SM), assessing unordered token overlap; (iv) Hits@1 and Hits@10; (v) MRR@10; and (vi) NDCG@10. We compare against BM25 [20], a sparse retrieval approach, and DSI [35], a GR method.

Table 1. Retrieval performance on the test set. Bold indicates best performance. PM and SM are not applicable to BM25.

Model	EM	PM	SM	Hits@1	Hits@10	MRR@10	NDCG@10
BM25	0.619	–	–	0.619	0.956	0.743	0.795
DSI	0.578	0.654	0.659	0.578	0.747	0.573	0.688
ReasonGR (Zero)	0.588	0.741	0.758	0.588	0.837	0.684	0.722
ReasonGR (CoT)	0.586	0.743	0.763	0.586	0.823	0.674	0.712
ReasonGR	0.624	0.761	0.780	0.624	0.846	0.709	0.743

Model Variants. We evaluate three models: our main method, ReasonGR, and two variants. (i) ReasonGR combines few-shot prompting with 2 examples, CoT prompting with 2 examples, LoRA adapters, and 4-bit quantization applied on a frozen backbone. (ii) ReasonGR (Zero) removes prompt training. (iii) ReasonGR (CoT) extends ReasonGR by modifying the outputs to reflect CoT prompting.

Implementation Details. Our backbone model is FLAN-T5 base [7]. Docids are formed by concatenating keywords extracted via KeyBERT [12], with company and year prepended, joined by hyphens. For each document, we adopt 10 pseudo-queries generated by docT5query [24]. LoRA weights are initialized with OLoRA [2] and optimized with LoRA+ [14] based on an 8-bit AdamW optimizer [8]. Training uses a batch size of 32 for 100 epochs with early stopping based on PM improvement. All experiments are performed on a single NVIDIA H100 GPU. The code is publicly available on <https://github.com/SmallNub/bachelor-thesis>

5 Experimental Results

5.1 Main Results

Table 1 presents retrieval performance of our ReasonGR variants alongside traditional baselines BM25 and DSI on both evaluation and test sets. We observe the following: (i) ReasonGR variants consistently outperform DSI across all metrics, indicating the benefit of reasoning-enhanced prompting and parameter-efficient adaptation over the vanilla GR approach. (ii) Compared to DSI, ReasonGR models show improvements in PM and SM metrics, reflecting better token-level and unordered component matching. The full ReasonGR model achieves the highest scores on PM and SM, and also improves hits@1 relative to BM25, showing increased accuracy in generating exact docids. (iii) Among ReasonGR variants, prompt training proves important: ReasonGR (Zero), without prompt training, performs worse than the full ReasonGR model, confirming the effectiveness of the combined prompting strategies. (iv) ReasonGR (CoT), which applies additional CoT prompting, achieves intermediate results, suggesting CoT contributes positively but suffers from increased complexity in predictions. (v) The similar or slightly better results on the test set compared to evaluation indicate stable training and reasonable generalization. The relatively lower EM scores compared

Table 2. GPU memory usage and training efficiency of different models.

Model	GPU Memory (MiB)		Training		
	Model only	Training (median)	Epochs	Time	Samples (mean)
ReasonGR (Zero)	653.875	8524.56	40	03:42:57	260.292
ReasonGR (CoT)	653.875	25188.56	51	05:50:15	165.684
ReasonGR	653.875	18308.56	55	05:14:19	184.630

to PM and SM highlight the challenge of exact docid generation, leaving room for future improvements. (vi) In terms of MRR and NDCG, GR methods underperform BM25, which highlights the challenges of applying generative paradigm to this task.

5.2 Training Efficiency

Memory Usage. Table 2 shows that the baseline model memory is similar across ReasonGR variants. Training memory varies notably: ReasonGR (Zero) uses about 8.5 GiB, ReasonGR with CoT prompting uses the most, and full ReasonGR is in between. The higher memory for CoT variants likely comes from longer or more complex input and output sequences.

Training Time. ReasonGR (Zero) trains 40 epochs in almost 4 h, processing 260 samples/sec. ReasonGR (CoT) takes nearly 6 h for 51 epochs with 166 samples/sec throughput. Full ReasonGR trains for about 5 h over 55 epochs at 185 samples/sec. CoT prompting adds computational cost, increasing training time and reducing throughput, while the Zero variant is more efficient but less reasoning-capable.

6 Discussion

The few-shot model with LoRA performs best overall, with the CoT model also outperforming the baseline, showing that prompting improves generative retrieval. However, limitations include: (i) the dataset was adapted from non-IR tasks, which may affect generalizability; (ii) the 512-token input limit is shorter than the average document length, and the FLAN-T5-base model may be too small for effective CoT reasoning [36], future work could explore larger or decoder-only models or add a classification head to separate reasoning from retrieval; and (iii) using sample-level loss may reduce fine-grained feedback, so token-level loss could better focus on relevant tokens and reduce noise from reasoning text.

7 Conclusion

We have presented ReasonGR, a generative information retrieval framework that enhances the reasoning ability through guided prompting and efficient fine-tuning. By combining few-shot and CoT prompting with LoRA and an adaptive loss, ReasonGR improves retrieval accuracy on complex numerical reasoning tasks, as shown on the FinQA dataset. Our results demonstrate that reasoning-enhanced prompting boosts performance while keeping training efficient. Though limited by model size and dataset scope, this work shows the promise of reasoning-driven generative retrieval. Future work will explore larger models, finer loss designs, and broader evaluation benchmarks.

Acknowledgements. We thank our reviewers for their feedback. This work is funded by Ahold Delhaize, through AIRLab Amsterdam, by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and by the European Union under grant agreements no. 101070212 and 101201510. We thank SURF for support in using the Dutch national supercomputer Snellius. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Disclosure of interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bose, J., Monti, R.P., Grover, A.: Controllable generative modeling via causal reasoning. *Trans. Mach. Learn. Res.* (2022)
2. Büyükkayyız, K.: OLoRA: orthonormal low-rank adaptation of large language models. arXiv preprint [arXiv:2406.01775](https://arxiv.org/abs/2406.01775) (2024)
3. Chen, J., Zhang, R., Guo, J., Fan, Y., Cheng, X.: GERE: generative evidence retrieval for fact verification. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2184–2189 (2022)
4. Chen, J., Zhang, R., Guo, J., Liu, Y., Fan, Y., Cheng, X.: CorpusBrain: pre-train a generative retrieval model for knowledge-intensive language tasks. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 191–200 (2022)
5. Chen, J., et al.: A unified generative retriever for knowledge-intensive language tasks via prompt learning. In: *46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1448–1457 (2023)
6. Chen, Z., et al.: FinQA: a dataset of numerical reasoning over financial data, pp. 3697–3711 (2021)
7. Chung, H.W., et al.: Scaling instruction-finetuned language models. *J. Mach. Learn. Res.* **25**(70), 1–53 (2024)
8. Dettmers, T., Lewis, M., Shleifer, S., Zettlemoyer, L.: 8-bit optimizers via block-wise quantization. In: *9th International Conference on Learning Representations*, ICLR (2022)

9. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: QLoRA: efficient fine-tuning of quantized LLMs. *Adv. Neural. Inf. Process. Syst.* **36**, 10088–10115 (2023)
10. Gao, L., Callan, J.: Unsupervised corpus aware language model pre-training for dense passage retrieval. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 2843–2853 (2022)
11. Ginting, M.F., Kim, S.K., Fan, D.D., Palieri, M., Kochenderfer, M.J., Agha-Mohammadi, A.a.: Seek: semantic reasoning for object goal navigation in real world inspection tasks. *arXiv preprint arXiv:2405.09822* (2024)
12. Grootendorst, M.: KeyBERT: minimal keyword extraction with bert (2020). <https://doi.org/10.5281/zenodo.4461265>
13. Hao, Y., Liu, Y., Mou, L.: Teacher forcing recovers reward functions for text generation. In: *Advances in Neural Information Processing Systems*. vol. 35, pp. 12594–12607 (2022)
14. Hayou, S., Ghosh, N., Yu, B.: LoRA+: efficient low rank adaptation of large models. In: *International Conference on Machine Learning*, pp. 17783–17806. PMLR (2024)
15. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. In: *International Conference on Learning Representations*. vol. 1, p. 3 (2022)
16. Jiang, Z., Araki, J., Ding, H., Neubig, G.: Understanding and improving zero-shot multi-hop reasoning in generative question answering. In: *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 1765–1775 (2022)
17. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781 (2020)
18. Li, X., Dou, Z., Zhou, Y., Liu, F.: CorpusLM: towards a unified language model on corpus for knowledge-intensive tasks. In: *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval* (2024)
19. Liu, Y., Cao, Y., Wang, S., Wang, Q., Bi, G.: Generative models for complex logical reasoning over knowledge graphs. In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 492–500 (2024)
20. Lù, X.H.: BM25S: orders of magnitude faster lexical search via eager sparse scoring (2024), <https://arxiv.org/abs/2407.03618>
21. Mekonnen, K.A., Tang, Y., de Rijke, M.: Lightweight and direct document relevance optimization for generative information retrieval. In: *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1327–1338 (2025)
22. Nogueira, R., Cho, K.: Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085* (2019)
23. Nogueira, R., Lin, J.: From doc2query to DocTTTTTquery. *An MS MARCO Passage Retrieval Task Publication* (2019), university of Waterloo
24. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to doctttttquery. *Online preprint* **6**(2) (2019)
25. Robertson, S., Zaragoza, H., et al.: The probabilistic relevance framework: BM25 and beyond. *Found. Trends® Inf. Retrieval* **3**(4), 333–389 (2009)
26. Tang, H., Sun, X., Jin, B., Wang, J., Zhang, F., Wu, W.: Improving document representations by generating pseudo query embeddings for dense retrieval. In: *ACL-JCNLP*, pp. 5054–5064 (2021)
27. Tang, Y., et al.: Semantic-enhanced differentiable search index inspired by learning strategies. In: *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4904–4913 (2023)

28. Tang, Y., Zhang, R., Guo, J., de Rijke, M.: Recent advances in generative information retrieval. In: Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, pp. 294–297 (2023)
29. Tang, Y., Zhang, R., Guo, J., de Rijke, M., Chen, W., Cheng, X.: Generative retrieval meets multi-graded relevance. In: Globerson, A., et al., (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 72790–72817. Curran Associates, Inc. (2024)
30. Tang, Y., Zhang, R., Guo, J., de Rijke, M., Chen, W., Cheng, X.: Listwise generative retrieval models via a sequential learning process. *ACM Trans. Inf. Syst.* (2024)
31. Tang, Y., Zhang, R., Guo, J., de Rijke, M., Fan, Y., Cheng, X.: Bootstrapped pre-training with dynamic identifier prediction for generative retrieval. In: Findings of the Association for Computational Linguistics ACL 2024, pp. 10303–10317 (2024)
32. Tang, Y., et al.: Generative retrieval for book search. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, vol. 1, pp. 2606–2617 (2025)
33. Tang, Y., Zhang, R., Ren, Z., Guo, J., de Rijke, M.: Recent advances in generative information retrieval. In: Goharian, N., et al. (eds.) Advances in Information Retrieval, pp. 363–368. Springer Nature Switzerland, Cham (2024)
34. Tang, Y., Zhang, R., Sun, W., Guo, J., de Rijke, M.: Recent advances in generative information retrieval. In: Companion Proceedings of the ACM Web Conference 2024, pp. 1238–1241. The Web Conference 2024, Association for Computing Machinery, New York (2024). <https://doi.org/10.1145/3589335.3641239>
35. Tay, Y., et al.: Transformer memory as a differentiable search index. *Adv. Neural. Inf. Process. Syst.* **35**, 21831–21843 (2022)
36. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural. Inf. Process. Syst.* **35**, 24824–24837 (2022)
37. Zhuang, S., et al.: Bridging the gap between indexing and retrieval for differentiable search index with query generation. In: Gen-IR@SIGIR 2023: The First Workshop on Generative Information Retrieval (2023)