RADCoT: Retrieval-Augmented Distillation to Specialization Models for Generating Chain-of-Thoughts in Query Expansion

Sung-Min Lee*, **Eunhwan Park***, **DongHyeon Jeon**, **Inho Kang**, **Seung-Hoon Na**†
Jeonbuk National University, Buzzni AI Lab, NAVER
{cap1232, nash}@jbnu.ac.kr,
jude@buzzni.com,
{donghyeon.jeon, once.ihkang}@navercorp.com

Abstract

Large language models (LLMs) have demonstrated superior performance to that of small language models (SLM) in information retrieval for various subtasks including dense retrieval, reranking, query expansion, and pseudo-document generation. However, the parameter sizes of LLMs are extremely large, making it expensive to operate LLMs stably for providing LLM-based retrieval services. Recently, retrieval-augmented language models have been widely employed to significantly reduce the parameter size by retrieving relevant knowledge from large-scale corpora and exploiting the resulting "in-context" knowledge as additional model input, thereby substantially reducing the burden of internalizing and retaining world knowledge in model parameters. Armed by the retrieval-augmented language models, we present a retrieval-augmented model specialization that distills the capability of LLMs to generate the chain-of-thoughts (CoT) for query expansion – that is, injects the LLM's capability to generate CoT into a retrieval-augmented SLM – referred to as **RADCoT**. Experimental results on the MS-MARCO, TREC DL 19, 20 datasets show that RADCoT yields consistent improvements over distillation without retrieval, achieving comparable performance to that of the query expansion method using LLM-based CoTs. Our code is publicly available at https://github.com/ZIZUN/RADCoT.

Keywords: Retrieval-augmented distillation, chain-of-thoughts, query expansion

1. Introduction

Large language models (LLMs) have achieved drastic improvements in various natural language understanding and generation tasks, owing to their knowledge manipulation and reasoning capabilities (Brown et al., 2020; Chowdhery et al., 2022), invoking the breakthrough AI services of ChatGPT and Bard. Recently, LLMs have also been successfully extended to information retrieval, including dense retrieval (Neelakantan et al., 2022; Ni et al., 2022), reranking (Sun et al., 2023; Saad-Falcon et al., 2023), query expansion (Jagerman et al., 2023), and document expansion (Yu et al., 2023; Ren et al., 2023; Wang et al., 2023b). These observable yet unpredictable capabilities of LLMs are often referred to as *emergent* abilities, and are usually not present in small language models (SLMs) (Wei et al., 2022). Even SLMs that have shown to perform some difficult tasks are characterized by qualitatively different behaviors and abilities than those of LLMs (Wei et al., 2023).

In information retrieval applications, however, it is impractical to exploit the large model parameters of LLMs for improving the retrieval performance, as most retrieval applications must be performed in real time for hundreds of millions of users simultaneously. Because the parameters of LLMs are necessary to retain world knowledge, *retrieval-augmented language models* have been recently explored as an alternative approach that stores world knowledge in external texts, rather than memorizing the knowledge in the model parameters (Izacard et al., 2023; Khandelwal et al., 2020; Lewis et al., 2020; Zhong et al., 2022; Borgeaud et al., 2022a; Wang et al., 2023a). As exemplified by RETRO (Borgeaud et al., 2022a), retrieval-augmented language models have demonstrated stable generation performance using much less parameters, comparing to those of LLMs.

In this paper, we aim to induce specialized SLMs that improve retrieval performance by generating a chain-of-thought (CoT) for query expansion. Recently, CoTs have been shown to be effective for query expansion under various sizes of language models (Jagerman et al., 2023), particularly LLMs. Towards designing effective SLMs for query expansion, motivated by the efficiency on the retrieval-augmented language models, we present a retrieval-augmented model specialization that distills the capability of LLMs to generate CoTs for query expansion, thereby injecting the LLM's capability of generating CoT into a retrieval-augmented SLM, referred to as the Retrieval-Augmented Distillation to Specialization Models for Generating Chain-of-Thoughts (**RADCoT**). Using

* Equal contribution.
† Corresponding author

LREC-COLING 2024, pages 13514–13523
20-25 May, 2024.

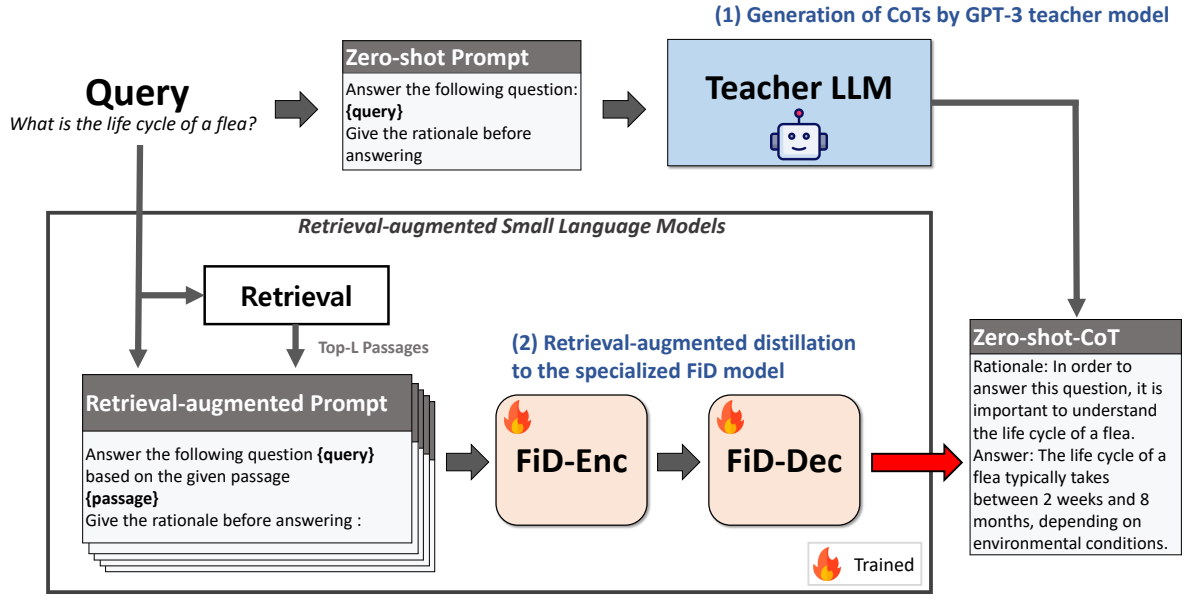

Figure 1: Overall architecture of RADCoT: 1) The generation step of CoTs by the GPT3 teacher model produces a set of queries and corresponding pseudo ground-truth CoTs without retrieval using Eq. (1); 2) Retrieval-augmented distillation, wherein the generated dataset of query—relational pairs is used as the training dataset to train the specialized 'retrieval-augmented' FiD model by adopting the top-retrieved passages as additional context, using retrieval-aware prompting and the loss function in Eq. (4). During the inference process, the student model takes the top-retrieved passages and retrieval-prompting method to generate a CoT using Eq. (5).

GPT3 as the teacher model and the small fusion-in-decoder (FiD) model (Izacard and Grave, 2021) as the student model, the knowledge distillation of RADCoT consists of the following steps:

1. **Generation of CoTs by GPT3 teacher model**: Similar to the insights of (Wang et al., 2023b; Jagerman et al., 2023), for each query, we pass a manually-designed zero-shot prompt as input to the GPT3 teacher model, which then generate CoT rationales. Given a dataset, the resulting pairs of queries and their CoT rationales are used as a set of pseudo-training examples which are learned by the student model.

2. **Retrieval-augmented distillation to specialized FiD model**: For each query, we first retrieve the top K passages and finetune the specialized FiD student model using the previously generated CoT by GPT3 as a pseudo-ground-truth CoT based on the autoregressive loss function. Our underlying assumption is that the use of the retrieval augmentation for distillation is largely helpful to substantially reduce the model size by minimizing the burden of storing knowledge explicitly in the model parameters, as in (Kang et al., 2023; Borgeaud et al., 2022a). With retrieval augmentation, it is expected the SLMs to prioritize reading and finding related spans in a retrieved context, as well as revising mentioned spans, not paying attention to storing the relevant knowledge.

The contributions of our work can be briefly summarized as follows. (1) We introduce a novel retrieval-augmented model specialization that distills the capabilty of LLMs to generate CoTs, employing their strengths to enhance the reasoning abilities of SLMs for query expansion. (2) Using models with significantly fewer parameters compared to than LLMs, we obtain the substantial performance improvements on various datasets.

2. Proposed Method: RADCoT

2.1. Task Definition

Our goal aims to distilling the capability of CoT generation by LLMs into a specialized SLM for query expansion.

2.2. Generation of CoTs by GPT3 Teacher Model

To prepare pseudo-training dataset for learning the SLM, we first generate a *pseudo-CoT rationale* using GPT3, thereby constructing a pair of query–rationale. Given LLM $\mathcal{M}$ and query q, the CoT is generated by the prompting method of (Wang et al., 2023b; Jagerman et al., 2023), described as

follows:

$$\mathsf{CoT}_{\mathsf{rationale}} = \mathcal{M}(\mathsf{Prompt}_{\mathsf{CoT}}(q)), \quad (1)$$

where $\mathcal{M}$ refers to GPT3 (text-davinci-003) and $\mathsf{Prompt}_{\mathsf{CoT}}$ indicates the CoT prompting, as illustrated in Figure 1. Using this generation process, we construct a query–rationale pair dataset across publicly available QA datasets including TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), and SQuAD (Rajpurkar et al., 2016).

2.3. Retrieval-augmented distillation to specialized FiD model

Given the generated CoT dataset in Section 2.2 and the set of top-k retrieved textual blocks $\mathcal{D}$, the next step is to train the retrieval-augmented FiD student model (Izacard and Grave, 2021).

More specially, the top K passages, $p_1, \cdots, p_L$ are first retrieved using a seperate retrieval module (i.e., BM25 or dense retrieval) for a given query q. The student model is designed a retrieval-augmented manner by placing a question-related passage at the CoT prompt as follows:

$$\begin{aligned} \tilde{q}_i &= \mathsf{Prompt}_{\mathsf{RADCoT}}(q, p_i) \\ \boldsymbol{T}_i &= \text{T5-Enc}(\tilde{q}) \in \mathbb{R}^{|\tilde{q}_i| \times d_{model}} \end{aligned} \quad (2)$$

where $\mathsf{Prompt}_{\mathsf{RADCoT}}$ indicates the retrieval-aware CoT prompting template used to generate a CoT rationale by referring to a retrieved set of passages as additional context, as illustrated in Figure 1, and $\boldsymbol{T}_i$ is the top-ith encoded retrieval-augmented prompt.

The resulting retrieval-augmented prompts of Eq. (2) are concatenated and fed into the decoder of FiD, as follows:

$$\boldsymbol{T} = [\boldsymbol{T}_1; \cdots; \boldsymbol{T}_L] \in \mathbb{R}^{(|\tilde{q}_i| \times L) \times d_{model}} \quad (3)$$

where $\boldsymbol{T}_i$ is the top-ith encoded retrieval-augmented prompt.

For training, given that $\boldsymbol{T}$ is used as the retrieval-augmented context, the FiD model is fitted to generate the pseudo target CoT, $\mathsf{CoT}_{\mathsf{rationale}}$ as follows:

$$Loss = -\log \text{T5-Dec}_{prop}(\mathsf{CoT}_{\mathsf{rationale}} \mid [\text{PAD}], \boldsymbol{T}) \quad (4)$$

where $\text{T5-Dec}_{prop}(x|y)$ is the generative probability of x under the condition of y, and [PAD] is the padding token located at the starting position in the decoder.

During the inference process, the FiD model uses $\boldsymbol{T}$ and [PAD] again to generate $\mathsf{CoT}^{(\mathsf{Small})}_{\mathsf{output}}$ as follows:

$$\mathsf{CoT}^{(\mathsf{Small})}_{\mathsf{output}} = \text{T5-Dec}([\text{PAD}], \boldsymbol{T}) \quad (5)$$

where $\text{T5-Dec}(x, y)$ is the decoder function in which x is the prefix of the decoder, and y is the encoder representation.

Finally, the expanded query is obtained by appending $\mathsf{CoT}^{(\mathsf{Small})}_{\mathsf{output}}$ N times to the original query q, formulated as follows:

$$q^+ = [\underbrace{q, \cdots, q}_{N}, \mathsf{CoT}^{(\mathsf{Small})}_{\mathsf{output}}] \quad (6)$$

Although prior studies used the fixed values of N, we note that when the length of the expanded terms significantly exceeds the original query q, the importance of q becomes undesirably small. To address this problem, we propose an adaptive method for determining N, namely the *repetition count*:

$$\begin{aligned} R &= \lfloor |\mathsf{CoT}^{(\mathsf{Small})}_{\mathsf{output}}| \;/\; |q| \rfloor \\ N &= \begin{cases} N_c, & \text{if } R \leq N_c \\ R, & \text{otherwise} \end{cases} \end{aligned} \quad (7)$$

where N_c is a fixed value.

3. Experiments

3.1. Experimental Setup

The details of the implementation and experiment setup are presented in Appendix A.

3.2. Baselines

In the experiment, we compare the performance of RADCoT to that of the following baselines:

- **BM25 + RM3** (Abdul-Jaleel et al., 2004) combines BM25 with the RM3 query expansion technique. BM25 measures document-query similarity, while RM3 uses the top-retrieved documents from initial search results to expand the original query.
- **BM25 + Query2Doc** (Wang et al., 2023b) employs an LLM to generate pseudo-documents by few-shot prompting, and then expands the query with the generated pseudo-documents.
- **BM25 + GRF-CoT-Keywords, Entities** (Mackie et al., 2023) builds probabilistic feedback models from long-form text generated from LLMs, consisting of generation of queries, entities, facts, news, documents and essays.
- **BM25 + CoT** employs CoT, which prompts LLMs for query expansion.

3.3. Main Results

Table 1 presents the performance results of RADCoT on MRR@K, recall@K, and nDCG@K over the MS MARCO dev and TRECL DL19, 20 datasets.

As seen in Table 1, even though RADCoT uses sorely SLMs, its performance is often comparable

Method	params	MS MARCO dev			TREC DL 19	TREC DL 20
		MRR@10	R@50	R@1k	nDCG@10	nDCG@10
Sparse retrieval						
BM25	✗	18.40	58.12	85.26	50.58	47.96
+ RM3	✗	15.66	56.43	86.06	52.16	48.96
+ query2doc	175B	21.40	65.30	91.80	66.20	62.90
+ GRF-CoT-Keywords	175B	-	-	-	55.00	54.20
+ GRF-CoT-Entities	175B	-	-	-	56.30	55.20
+ CoT (ours)	175B	21.19	64.19	89.65	60.76	55.74
+ $\text{RADCoT}_{\text{small}}$	80M	19.45	59.82	87.29	56.05	50.56
+ $\text{RADCoT}_{\text{base}}$	250M	20.64	60.93	88.12	56.41	51.57
+ $\text{RADCoT}_{\text{large}}$	780M	20.69	61.63	88.78	57.22	53.00

Table 1: Main results of RADCoT across various model sizes, comparing to other existing and baseline methods on MS-MARCO passage ranking and TREC datasets.

Method	params	MS MARCO dev			TREC DL 19	TREC DL 20
		MRR@10	R@50	R@1k	nDCG@10	nDCG@10
Sparse retrieval						
BM25	✗	18.40	58.12	85.26	50.58	47.96
+ RADCoT (p:0, trained:0)	250M	18.79	59.14	86.68	50.10	51.96
+ RADCoT (p:0, trained:50)	250M	19.08	59.09	87.12	50.52	50.72

Table 2: Comparison of RADCoT variants on MS-MARCO passage ranking and TREC datasets, where p:0 indicates that retrieval is not used (i.e., L is 0) for FiD decoding and trained:x indicates that x retrieved passages is used to distill the FiD model.

Method	nDCG		
	@1	@5	@10
CoT (ours)	67.44	62.81	60.76
RADCoT	66.67	**63.69**	59.80
RADCoT + CoT (ours)	**69.77**	62.75	**60.98**

Table 3: RADCoT results comparing to the run of using CoT by GPT3 and the combination of CoTs generated by RADCoT and LLM on TREC DL 19 dataset.

to that of LLM-based approaches. In particular, RADCoT exhibits slightly superior performance to that of GRF-CoT-Keywords and CRF-CoT-Entities on TREC DL 19. Overall, the performance of RADCoT increases with model size.

3.4. Ablation studies

Impact of retrieval component. Table 2 shows that RADCoT achieves superior performance to that of its variants without retrieval, confirming that the retrieval component is quite important for significantlly improving the distillation effect.

Impact of combining LLM's CoT. Table 3 shows that RADCoT yields further improvements when combining the run using CoT generated by LLMs, suggesting that the ensemble of SLMs and LLMs is promising, so being valuable to be explored further in future studies.

Impact of query2doc distillation. Noting that **query2doc** exhibits the state-of-the-art retrieval performance (Wang et al., 2023b), we further extensively apply RADCoT for distilling query2doc, resulting in **RADCoT-query2doc**, shortly referred to as the *query2doc distillation*, being distinguished from the original *CoT distillation*, where a pseudo CoT rationale in Eq. (1) is replaced with a pseudo-document generated by query2doc, i.e., $\text{CoT}_{\text{rationale}} = \mathcal{M}(\text{Prompt}_{\text{query2doc}}(q))$ where $\mathcal{M}(\text{Prompt}_{\text{query2doc}}(q))$ is exactly the same method of few-shot prompting LLMs of query2doc. Equipping with query2doc as a teacher model, a FiD-based student model is then trained using the same manner of the retrieval-augmented distillation in Section 2.3.

Table 4 shows the comparision results of RADCoT-query2doc and RADCoT. As expected from the superiority of query2doc, RADCoT-query2doc consistently improves the original RADCoT, across all test collections. Comparing to RADCoT, the overall performance gap between the student and teacher models tends to be larger in RADCoT-querydoc, thus still having rooms to make improvements.

3.5. Case studies

Table 5 presents illustrated examples of comparing generation results of the CoT-based teacher model and the RADCoT-distilled student model, for two different queries, where top retrieved passages and ground truth are presented for additional reference.

Method	params	MS MARCO dev			TREC DL 19	TREC DL 20
		MRR@10	R@50	R@1k	nDCG@10	nDCG@10
Sparse retrieval						
BM25	✗	18.40	58.12	85.26	50.58	47.96
+ CoT	175B	21.19	64.19	89.65	60.76	55.74
+ query2doc	175B	21.40	65.30	91.80	66.20	62.90
+ RADCoT	250M	20.46	61.30	88.39	59.80	53.94
+ RADCoT - query2doc	250M	21.63	63.61	89.47	62.31	58.15

Table 4: The comparison results of RADCoT-query2doc and the original RADCoT on the MS-MARCO passage ranking and TREC datasets, where RADCoT-query2doc is the student model distilled by RADCoT under query2doc as a teacher model.

Query	who sings monk theme song
LLM generation (CoT)	Rationale: **The theme song** for the television show **Monk** is an instrumental piece of music composed by Jeff Beal. Answer: Jeff Beal.
RADCoT generation	Rationale: **The Monk theme song is** a popular **song** from the television show **Monk**. Answer: **The Monk theme song is** performed by **Randy Newman**.
Retrieved passages	**Top-1**: Incoming search terms: 1 who sings the real theme song. ... **Top-2**: Who Sings The Theme Song For The Show The OC Phantom Planet sing the song California the theme song for the show The OC. ... **Top-3**: Related Who Sings Questions. What band sings the theme song for Smallvile The theme song for Smallville is called Save Me by the band Remy Zero. ... **Top-4**: Who sings the theme song to WWE Superstars? The artist that sings the theme song for WWE Superstars ... **Top-5**: Who sings John Cena theme song? john cena sings his own theme song john cena sings his own theme song ...
Ground truth	exists and is an alternate of. **The Monk theme song is** It's a Jungle Out There **by Randy Newman**. **The Monk theme song is** It's a Jungle Out There **by Randy Newman**.
Query	antibiotics for what kind of infection
LLM generation (CoT)	The answer to this question depends on the type of infection that is present. Antibiotics are used to treat bacterial infections, so if the infection is caused by bacteria, then antibiotics may be an appropriate treatment. However, antibiotics are not effective against viral infections, so if the infection is caused by a virus, then antibiotics would not be an appropriate treatment. Therefore, it is important to determine the type of infection before deciding whether or not antibiotics are an appropriate treatment.
RADCoT generation	Rationale: **Antibiotics** are medications **used to treat** a variety of **infections**, including **bacterial**, viral, and skin **infections.** Answer: **Antibiotics** are used to **treat** a variety of **infections**, including **bacterial, viral**,
Retrieved passages	**Top-1**: In most cases of antibiotic use, a doctor must choose an antibiotic based on the most likely cause of the infection. For example, if you have an earache, the doctor knows what kinds of bacteria cause most ear infections. ... **Top-2**: 1. Get a prescription for antibiotics from your doctor. Antibiotics will not help an ear infection that is caused by a virus, so doctors do not always prescribe antibiotics for ear infections. ... **Top-3**: Antibiotics are not recommended to help treat many ear infections. Your healthcare professional will be able to determine what kind of ear infection you or your child has and if antibiotics would help. ...
Ground truth	**Antibiotics** are **used to treat** or prevent some types of **bacterial infection**. They kill **bacteria** or prevent them from reproducing and spreading. **Antibiotics** aren't effective against **viral infections.**

Table 5: Case study for RADCoT on TREC DL 2020 dataset, comparing to the generation outputs of the teacher and student models, given top-retrieved documents and ground truths.

For both queries, RADCoT successfully generates relevant terms and a proper answer which is well-overlapped with the ground truth, where LLM suffers from the hallucination in some extent so causing to generate irrelevant or unnecessary terms, resulting in the incorrect answers.

4. Conclusion

In this paper, we proposed the use of the retrieval-augmented model specialization to generate CoTs in query expansion, namely RADCoT, in order to reduce the model size of the distilled model, motivated by the recently-reported effectiveness of the retrieval-augmented language models and the knowledge-augmented distillation. The experiment results on various standard datasets show that the proposed RADCoT led to consistent improvements over the distillation without the retrieval, showing the comparable performances to those using LLM-based CoTs. In the future work, we would like to investigate on jointly distilling the retrieval capability from LLM as well as query expansion under the unified FiD framework.

Acknowledgements

This work was supported by NAVER Corporation. We would like to thank all anonymous reviewers for their valuable comments and suggestions.

5. Bibliographical References

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George 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 Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022a. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022b. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

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*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. 2023. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. *CoRR*, abs/2305.18395.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. 2023. Chain of knowledge: A framework for grounding large language models with structured knowledge bases. *arXiv preprint arXiv:2305.13269*.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. *arXiv preprint arXiv:2304.13157*.

Donald Metzler and W Bruce Croft. 2007. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large language models are effective text rankers with pairwise ranking prompting.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6102–6114, Toronto, Canada. Association for Computational Linguistics.

Joseph John Rocchio. 1971. The smart retrieval system: Experiments in automatic document processing, chapter relevance feedback in information retrieval. *Englewood Cliffs, NJ*.

Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. 2023. Udapdr: Unsupervised

domain adaptation via llm prompting and distillation of rerankers.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, Anima Anandkumar, and Bryan Catanzaro. 2023a. Shall we pretrain autoregressive language models with retrieval? A comprehensive study. *CoRR*, abs/2304.06762.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian-guang Lou. 2023. Re-reading improves reasoning in language models. *arXiv preprint arXiv:2309.06275*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators.

Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training language models with memory augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5657–5673, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Category	Number
SQuAD v1.1 (Rajpurkar et al., 2016)	10,570
WebQuestions (Berant et al., 2013)	11,313
TriviaQA (Joshi et al., 2017)	2,032
Pseudo Query-Rationale pair Dataset	23,915

Table 6: The detailed statistics for the datasets.

A. Implementation Details.

As mentioned in Section 2.2, we utilize publicly available QA datasets, TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013), and SQuAD (Rajpurkar et al., 2016) to generate a pseudo CoT rationale using GPT3, enabling us to distill the capability of generating CoTs. Table 6 shows more detailed statistics for the dataset. To train and evaluate the performance of query expansion task, we employ MS-MARCO dev dataset (Nguyen et al., 2016), TREC DL 19, 20 dataset (Craswell et al., 2020) for training and MRR, Recall@K, and nDCG for evaluation, following prior works (Wang et al., 2023b; Mackie et al., 2023). For training small LM, we use flan-t5-base [1] (Chung et al., 2022) as backbone model. All trainings were conducted with AdamW optimizer with learning rate 1e-4 for 5 epochs, and random seed was fixed for reproducing the results. Batch size was set to 4 with 2 accumulation. Training was conducted for 7 hours with 4 NVIDIA RTX A6000. For generation, greedy decoding method was exploited. N_c was fixed to 5 and passage numbers were fixed to 50 by default.

B. Impact of N_c value.

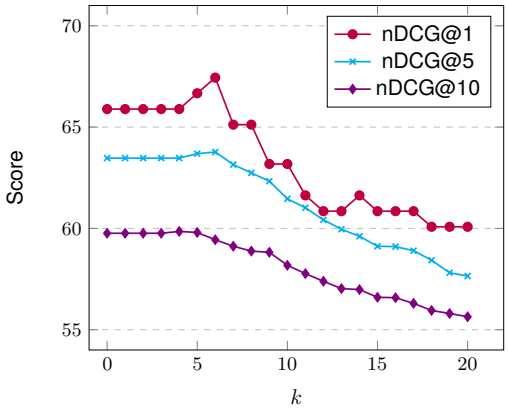

Figure 2: The retrieval performance changes based on the N_c value.

Figure 2 shows the retrieval performance changes based on the N_c value. This value is a

[1] https://huggingface.co/google/flan-t5-base

parameter that determines the frequency of query repetition, irrespective of the text length of the query and CoT. It appears that an increase in this value leads to a decrease in performance, suggesting that the continuous repetition of a query reduces the significance of CoT, thereby diminishing performance.

C. Impact of passages number.

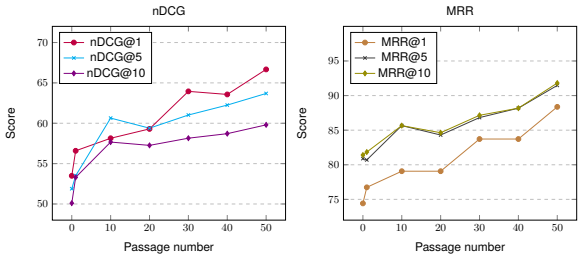

Figure 3: The impact of the number of passages on the retrieval performance of our method during training and inference.

Figure 3 illustrates the impact of the number of passages on the retrieval performance of our method during training and inference. It is evident that an increase in the usage of passages correlates with enhanced performance, indicating that knowledge distillation is more effective when more passages are utilized.

Query	query
Zero-shot Prompt	Answer the following question: {query} Give the rationale before answering
Retrieval-augmented Prompt	Answer the following question {query} based on the given passage {passage} Give the rationale before answering

Table 7: The prompt example used for RADCoT

Query	query
Few-shot Prompt	**Write a passage that answers the given query:** **Query:** what state is this zip code 85282 **Passage:** Welcome to TEMPE, AZ 85282. 85282 is a rural zip code in Tempe, Arizona. The population is primarily white, and mostly single. At $200,200 the average home value here is a bit higher than average for the Phoenix-Mesa-Scottsdale metro area, so this probably isn't the place to look for housing bargains.5282 Zip code is located in the Mountain time zone at 33 degrees latitude (Fun Fact: this is the same latitude as Damascus, Syria!) and -112 degrees longitude. **Query:** why is gibbs model of reflection good **Passage:** In this reflection, I am going to use Gibbs (1988) Reflective Cycle. This model is a recognised framework for my reflection. Gibbs (1988) consists of six stages to complete one cycle which is able to improve my nursing practice continuously and learning from the experience for better practice in the future.n conclusion of my reflective assignment, I mention the model that I chose, Gibbs (1988) Reflective Cycle as my framework of my reflective. I state the reasons why I am choosing the model as well as some discussion on the important of doing reflection in nursing practice. **Query:** what does a thousand pardons means **Passage:** Oh, that's all right, that's all right, give us a rest; never mind about the direction, hang the direction - I beg pardon, I beg a thousand pardons, I am not well to-day; pay no attention when I soliloquize, it is an old habit, an old, bad habit, and hard to get rid of when one's digestion is all disordered with eating food that was raised forever and ever before he was born; good land! a man can't keep his functions regular on spring chickens thirteen hundred years old. **Query:** what is a macro warning **Passage:** Macro virus warning appears when no macros exist in the file in Word. When you open a Microsoft Word 2002 document or template, you may receive the following macro virus warning, even though the document or template does not contain macros: C:\<path>\<file name>contains macros. Macros may contain viruses. **Query:** {query} **Passage:**
Retrieval-augmented Prompt	Answer the following question {query} based on the given passage {passage} Give the rationale before answering

Table 8: The prompt example used for RADCoT - query2doc