Distilling Step-By-Step With RAG

Anirudh Kanneganti* Christopher B. Nalty* Devin Crowley* Logeswaran Sivakumar*
Oregon State University
{kannegaa, naltyc, crowleyd, sivakuml}@oregonstate.edu

Abstract

Large language models (LLMs) provide impressive capabilities but present challenges in training and deployment due to the sheer size and inference time of modern architectures. To combat this, there is a push towards training smaller LLMs specialized to particular tasks. These can achieve comparable performance on those tasks to larger, more general LLMs. Methods to achieve this include fine-tuning and teacher-student distillation, but these require large amounts of data to train. Prior work introduced the technique of distilling step-by-step, which uses rationales for the models' answers to provide a richer learning signal and reduce the data requirement. We systematically improve the quality of these rationales by augmenting the technique with retrieval-augmented generation (RAG), additionally circumventing the need for human-generated rationales in distilling step-by-step, and test our variation using a significantly smaller teacher model. Ultimately we demonstrate comparable to incrementally superior performance despite using a teacher model that is 67 times smaller, and better performance in the low data setting.

The code is available on our repository.

1. Introduction

In recent years, Large Language Models (LLMs) like Llama3 have demonstrated remarkable performance across a wide range of natural language processing (NLP) tasks. However, their deployment in real-world applications is often constrained by substantial computational and memory requirements. LLMs are powerful, general models that are able to tackle a wide variety of natural language tasks, such as summarization, question answering, and coding. However, deploying such large models can be costly and at times unnecessary; there is no need to use a model capable of many tasks if you only have one target task. Smaller models are usually unable to capture the wide variety of tasks that a larger model can, but they have the ability to perform strongly on a single task. Smaller models allow for deployment on edge devices, such as mobile phones, and provide cheaper and quicker inference over their larger counterparts. To address these limitations, we propose a novel method called **Distilling Step-by-Step with RAG** (Retrieval-Augmented Generation) to train smaller, task-specific models that achieve comparable or even superior performance to their larger counterparts. Our method extends from the previous work on step-by-step distillation [3].

Our approach leverages the strengths of both retrieval-based methods and LLMs by distilling the knowledge from a powerful teacher model (Llama3) into a more compact student model (T5). The Llama3 model [5] is a state-of-the-art large-scale language model optimized for chat-based question answering, containing approximately 8 billion parameters. On the other hand, T5 [8] (Text-To-Text Transfer Transformer) is a versatile LLM developed by Google, with the base variant consisting of 220 million parameters. The process begins by vectorizing the dataset and using a retrieval mechanism to gather the most relevant context for each query. This enriched context is then used to generate rationales and labels through the teacher model, which serve as informative supervision for training the student model.

The key innovation in the step-by-step method we build upon lies in the distillation of both rationales and labels, allowing the student model to learn not only the final predictions but also the underlying reasoning processes. This multi-task learning framework enhances the student model's capability to generalize from fewer examples, significantly reducing the data and computational resources required for training.

We demonstrate the benefit of our approach through a series of experiments, showing that the student model trained with Distilling Step-by-Step with RAG context can achieve performance on par with or better than the teacher model while being substantially smaller and more efficient. Our results indicate a promising direction for making advanced NLP capabilities more accessible and practical for a broader range of applications.

In the following sections, we detail our methodology, ex-

*Equal contribution

perimental setup, and results, highlighting the advantages and limitations of our proposed approach and we also discuss potential future directions.

2. Related Work

There are many methods available to produce a relatively small model with strong task-specific capabilities. In today's ecosystem it is largely unnecessary to train a small model from scratch when it can be bootstrapped from a pretrained model; certainly this reduces the amount of data necessary. In general there are two main approaches for doing this: fine-tuning and teacher-student distillation.

2.1. Fine-Tuning

Fine-tuning is the process of using labeled data to further train a pretrained model to improve its performance on the distribution from that labeled dataset. It is essentially the same process as training a model from scratch, but beginning with pretrained weights. Unfortunately, labeled data is often scarce and is often insufficient to prevent overfitting. Labeled data is particularly scarce if we are interested in data of a particular form; this is sometimes desirable when training LLMs, as in step-by-step distillation, discussed in Section 2.3.

2.2. Model Distillation

Model-distillation is similar to fine-tuning except it leverages a larger "teacher" model to generate labels with which to train a smaller "student" model. In contrast to fine-tuning, model-distillation therefore does not require labeled data, bypassing some key issues with fine-tuning: unlabeled data is much more abundant than labeled data, and the form of the labels can be molded as desired at generation.

Model distillation suffers a computational overhead compared to fine-tuning since inference on the large teacher model to generate labels is expensive. However, the teacher model outputs soft labels (probability distributions over possible outputs) which provide a richer signal than hard labels, potentially reducing the amount of data needed. Despite this, quite a lot of unlabeled data is still required.

2.3. Distilling Step-by-Step

Distilling step-by-step [3] is a technique to make model distillation more efficient and robust. This technique uses chain-of-thought (CoT) [10] prompting to elicit responses from the teacher model that additionally contain rationales. These rationales explain the teacher's answers to the queries from the dataset. It does this by prepending a series of demonstrative examples to each original query before providing it as input to the teacher. Then, step-by-step distillation separately queries the student model for a rationale and again for a label and trains it on the similarity of these outputs to those generated by the teacher.

This approach trains the student in a multi-task fashion wherein it is asked to provide either a rationale or a label by prepending "[rationale]" or "[label]" to the original query. Training the student model to produce rationales acts as supervision and has the benefit of regularizing its behavior because it must be able to output intermediate reasoning. This also improves the richness of the learning signal, and significantly reduces the amount of data required to achieve the same performance, or better [3].

Signalling to the student whether to output either a rationale or an label creates no computational overhead to this approach at test time because the student can be queried just for labels without having to produce rationales as it does in training. However, this work [3] has the downside that it uses (a small number of) CoT example prompts and corresponding rationales hand-made by human researchers tailored to the dataset being learned upon.

2.4. Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) [4] is a technique designed to help LLMs generate correct answers by giving them relevant context to the task at hand. This is done by keeping a database of information that may be useful for the model. Choices for the type of information can range from reference documents to example questions and answers. This is usually a vector database, where each element of the database is encoded into a vector. At runtime the query is also encoded into a vector and the database is searched for the top k most similar vectors. These k most similar vectors are then retrieved from the database and added to the prompt given to the LLM. This has been shown to be a powerful method for improving answer quality and preventing hallucinations.

3. Methodology

Our method adapts step-by-step distillation [3] to circumvent the need for human-generated rationales for each dataset, and potentially improve the quality of the training data produced by the teacher. Rather than prepending a handful of human-generated rationales in the CoT prompts that augment inputs for the teacher, we instead use RAG [4] to find the k most similar queries and corresponding ground truth labels to a given query and its label. There are other possible choices here, such as the k most similar queries and corresponding un-augmented teacher outputs, but in our implementation we choose to leverage the labels.

3.1. Models

In our experiments we use two teacher models: 8B Llama3, with 8 billion parameters; 540B PaLM, with 540 billion parameters; and 220M T5, with 220 million parameters. Llama3 and PaLM serve as the teacher model in

our experiments, and in all cases we use T5 as the student model.

3.2. Datasets

3.2.1 CommonsenseQA (CQA)

The CommonsenseQA [9] is a benchmark dataset designed to evaluate natural language understanding models in commonsense reasoning tasks. It comprises over 12,000 question-answer pairs sourced from crowd-workers, where each question challenges models to infer likely outcomes and make decisions based on everyday scenarios. Each question has five possible answers, including one correct answer and four distractors, aiming to test the model's ability to discern plausible but incorrect choices. Evaluated based on accuracy, CommonsenseQA assesses a model's ability in understanding and reasoning about implicit knowledge and relationships, serving as a critical tool for advancing the capabilities of natural language understanding systems.

Below is an example:

Question: What might someone do if they accidentally cut themselves?

Answer Choices: ✓ put on a bandage, × tie their shoes, × wash their hair, × brush their teeth, × feed their pet

3.2.2 Adversarial NLI (ANLI)

The Adversarial NLI dataset (ANLI) [6], introduced by facebook research, extends traditional Natural Language Inference (NLI) evaluation by providing challenging examples derived from heuristic-based transformations of existing datasets. Each example consists of a premise (sentence A) and a hypothesis (sentence B), where the goal is to determine the logical relationship between them: entailment (A entails B), contradiction (A contradicts B), or neutral (no inferential relationship). These pairs are carefully constructed to maintain the original label while subtly altering surface-level features, thereby challenging models that rely on shallow linguistic cues. The dataset serves as a critical benchmark for assessing the robustness and generalization capabilities of NLI models, fostering the development of systems with deeper semantic understanding and enhanced performance on real-world language tasks.

Below is an example:

Premise: A cat is chasing a mouse.

Hypothesis: A mouse is being pursued by a feline.

Label: Entailment

3.2.3 Simple Variations on Arithmetic Math word Problems (SVAMP)

SVAMP (Simple Variations on Arithmetic Math word Problems) [7] is a meticulously curated dataset aimed at evaluating models' capabilities in solving elementary-level arithmetic word problems (MWPs). Derived from seed examples selected from the ASDiv-A dataset for their higher quality and difficulty, SVAMP introduces variations categorized into Question Sensitivity, Reasoning Ability, and Structural Invariance. These variations test models across different aspects of problem-solving, ensuring robust evaluation beyond superficial patterns. SVAMP comprises one-unknown arithmetic problems solvable with expressions using up to two operators, maintaining a focus on fundamental problem-solving skills. By prioritizing clarity and fairness in example selection and evaluation, SVAMP sets a rigorous standard for assessing the depth of mathematical reasoning and natural language understanding in AI models.

Below is an example:
Body: Robin has 28 packages of gum and 14 packages of candy. There are 6 pieces in each package.
Question: How many pieces does Robin have?
Equation: ((28.0 + 14.0) / 6.0)
Answer: 7

3.3. Pseudocode Walkthrough

Pseudocode for Step-By-Step Distillation With RAG is provided in Algorithm 1. Lines 1-6 set up the RAG database and create the data to train the student on, and lines 7-14 describe the training procedure.

3.3.1 Preparing the RAG database

Line 1 vectorizes each ground-truth query and label (Q^*&L^*) using BERT embeddings [2]. Line 2 finds the (indices of the) top-k nearest vectors for each vector using simple L2 distance in the embedded space.

Line 4 creates the prompt to infer the teacher with. This has 4 components.

1. A system prompt, which provides context for the following CoT structure. For SVAMP, it is: *"System: This is a chat between a user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions based on the context. The assistant should make use of the context to logically answer the questions"*

2. The CoT RAG context, consisting of k=10 query-label pairs of the form *"User: <query>\nAssistant: <label>"*

3. The pre-question, which requests output of a particular form. This varies slightly between tasks but for

SVAMP it is this: *"Now carefully observe how assistant logically answered above questions and answer the user question along with a rationale on how the answer is achieved. Provide me output in this form 'Rationale: ,hence the answer is Answer: '"*

4. The actual query, of the form: *"User: <query>\n\nAssistant:\n"*

These are modified on a task-by-task basis, requiring special treatment for multiple-choice and hypothesis entailment datasets. See [3] for further details.

Lines 5 & 6 infer the teacher model on the above prompt and parse its output into a rationale and label using regular expressions.

3.3.2 Training the Student

The student model receives a much simpler input prompt compared to the teacher, consisting of the original query with "[rationale]" or "[label]" prepended to indicate whether it should output a rationale or label, as described in Section 2.3. Finally, on line 13 the loss is computed on the similarity between the student's labels and rationales versus the ground truth label and the teacher's rationale (it is also possible to use the teacher's label but we leverage the labels from the dataset). The similarity metric used is mean square error (MSE). The hyperparameter λ determines the relative importance of the rationale in this overall loss, although in practice the loss is a convex combination of these two. We follow [3] and use an equal weighting between the two.

Algorithm 1: Distilling Step-By-Step With RAG

Input: Dataset $D = \{(Q_1^*, L_1^*), \ldots (Q_n^*, L_n^*)\}$ of queries & labels, the RAG context size k, the rationale-importance λ, the teacher model T, and the student model S

▷Prepare RAG database
1 $D_{vec} \leftarrow$ vectorize(D)
2 $M_{idx} \leftarrow$ get_match_idxs(D_{vec}, k)
3 **for** $i \leftarrow 1$ ***to*** n **do**
4 $\quad P \leftarrow$ create_prompt$(D[i], D[M_{idx}[i]])$
5 $\quad O_i \leftarrow T(P)$ ▷Infer the teacher
6 $\quad R_i^T, L_i^T \leftarrow$ parse_output(O_i)
▷Train student
7 **while** *not done* **do**
8 $\quad$ **for** *each mini-batch of* Q^*, R^T, L^T **do**
$\quad\quad$ ▷Query student separately for rationales and labels
9 $\quad\quad P \leftarrow$ [rationale]$+Q^*$
10 $\quad\quad R^S \leftarrow S(P)$
11 $\quad\quad P \leftarrow$ [label]$+Q^*$
12 $\quad\quad L^S \leftarrow S(P)$
$\quad\quad$ ▷Compute loss and train
13 $\quad\quad loss \leftarrow$ MSE$(L^S, L^*) + \lambda$ MSE(R^S, R^T)
14 $\quad\quad$ train$(S, loss)$

4. Results

In investigating the benefits of step-by-step distillation with RAG we performed a series of simple experiments. First, we trained our method along with two baseline methods to compare their performances across multiple datasets, presented in Figure 1. Our method, step-by-step distillation with RAG, is labeled as Llama3 With Top-10 RAG since it uses the comparatively small 8B Llama3 teacher model. The first baseline is Llama3 With Random RAG, differentiated by using randomly chosen RAG context rather than the most similar query-label pairs from the RAG database. The second baseline is PaLM With Human CoT. This uses the 540B PaLM model [1] for the teacher and uses the same human-generated CoT context provided in [3].

We find similar performance across all methods, noting that the reported accuracy is literally the token-for-token accuracy. Exact values can be found in Table 1. No method is clearly dominant. This tells us a few things. First, because Llama3 With Top-10 RAG only slightly outperformed Llama3 With Random RAG, it appears that the selection of query-label pairs from the RAG database by proximity in the embedded space is not as helpful as we might have expected. This could be confounded by the consistency of the datasets however – when the data is sufficiently regular, there is less value in choosing the most similar examples for context. It is possible that we would see more value on more varied datasets.

The other main takeaway is that we were able to achieve similar performance using the much smaller 8B Llama3 as the teacher model compared to 540B PaLM – the difference being our use of RAG context rather than the human-generated CoT context of prior work [3].

We also present the performance of these methods on the CQA dataset in Figure 2 using different percentages of the data to explore their scaling properties and how well they handle low data regimes. Exact values can be found in Table 2. Although no method is strictly dominant, our Llama3 With Top-10 RAG method and the Llama3 with Random RAG baseline outperform Palm With Human CoT by a few percentage points on 3 out of 4 of the data sizes. The difference isn't dramatic but recall again that the Llama3 teacher models are using less than 1.5% as many parameters as the PaLM model, and still outperform down to 25% of the CQA data set.

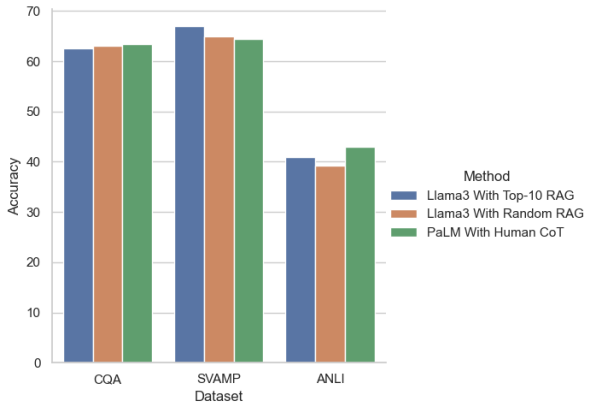

Figure 1. Comparison of methods on the CQA, SVAMP, and ANLI datasets.

Model	T5 Base		
Method/Dataset	**CQA**	**SVAMP**	**ANLI**
Llama3 with Top-10 RAG	63.39	64.50	42.90
Llama3 with Random RAG	63.06	65.00	39.20
PaLM with Human CoT	62.57	67.00	40.90

Table 1. Comparison of methods on CQA, SVAMP and ANLI datasets using T5 Base model.

5. Conclusion

In this work we explore the impact of using RAG with distilling step-by-step. We find that for two of three datasets tested the use of RAG improves the final accuracy of distilled models. We also show that with recent advancements in LLM techniques strong performance is retained even when using a greatly reduced teacher size. Our method improves performance using an 8-billion parameter model, compared to a 540-billion parameter model used in previous work. This improvement does not require human generated rationales, showing that strong rationales can be inferred with simple prompting.

Our work is limited in a few ways that could be approved upon in future work. Firstly, our method currently only retrieves examples from the dataset as context. In knowledge intensive tasks this may not provide enough information to the teacher to properly respond to the query. This could be addressed by adding a database of RAG information as context to the teacher. Our work is also limited in that it makes adding human rationales difficult. For human rationales to be used as demonstrations along with our method they would have to be produced for all queries in the dataset. Finally, with our limited resources we were not able to explore the effect of using larger teachers, or how our method scales with larger student models.

Future work on Distilling Step-By-Step with RAG can

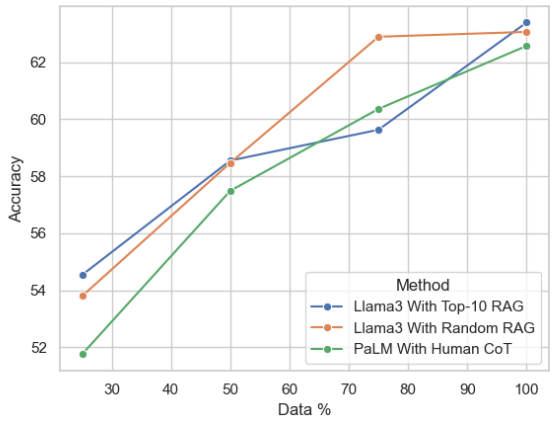

Figure 2. Comparison of methods on CQA dataset using various percentages (25%, 50%, 75% and 100%) of the dataset.

Dataset: CQA				
Method	**25%**	**50%**	**75%**	**100%**
Llama3 with Top-10 RAG	54.54	58.55	59.63	63.39
Llama3 with Random RAG	53.80	58.47	62.89	63.06
PaLM with Human CoT	51.76	57.49	60.36	62.57

Table 2. Performance on CQA dataset with varying percentages of training data.

expand in several impactful directions. One key area is enhancing the retrieval mechanism and exploring more advanced models to improve the quality of the retrieved context, potentially boosting the student model's performance. This could involve multi-hop reasoning or integrating external knowledge sources to enrich the rationale generation process. Applying this method to a wider range of datasets, including those in different languages or with varied linguistic structures, can further validate the generalizability of our approach. Investigating dynamic rationale generation techniques and developing more complex rationales that capture deeper reasoning processes are also promising avenues for enhancing the student model's capabilities.

Conducting longitudinal studies to monitor the student models' performance and adaptability over time is essential for understanding how they cope with evolving data distributions and language trends. Addressing ethical considerations and mitigating biases in both the training data and models is crucial. Developing methods to detect, evaluate, and reduce these biases will ensure fairness and transparency in model predictions, supporting their responsible and equitable use in real-world applications.

References

[1] 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. Palm: Scaling language modeling with pathways, 2022. 4

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 3

[3] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. 1, 2, 4

[4] 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. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. 2

[5] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa: Surpassing gpt-4 on conversational qa and rag, 2024. 1

[6] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding, 2020. 3

[7] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems?, 2021. 3

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020. 1

[9] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2018. 3

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 2