CRAFT Your Dataset: Task-Specific Synthetic Dataset Generation Through Corpus Retrieval and Augmentation

Ingo Ziegler[1*] **Abdullatif Köksal**[2,3*] **Desmond Elliott**[1] **Hinrich Schütze**[2,3]

[1]Department of Computer Science, University of Copenhagen
[2]Center for Information and Language Processing (CIS), LMU Munich
[3]Munich Center for Machine Learning (MCML)
[*]Shared first authorship
`inzi@di.ku.dk`, `akoksal@cis.lmu.de`

Abstract

Building high-quality datasets for specialized tasks is a time-consuming and resource-intensive process that often requires specialized domain knowledge. We propose Corpus Retrieval and Augmentation for Fine-Tuning (CRAFT), a method for generating synthetic datasets, given a small number of user-written few-shots that demonstrate the task to be performed. Given the few-shot examples, we use large-scale public web-crawled corpora and similarity-based document retrieval to find other relevant human-written documents. Lastly, instruction-tuned large language models (LLMs) augment the retrieved documents into custom-formatted task samples, which then can be used for fine-tuning. We demonstrate that CRAFT can efficiently generate large-scale task-specific training datasets for four diverse tasks: biology question-answering (QA), medicine QA and commonsense QA as well as summarization. Our experiments show that CRAFT-based models outperform or achieve comparable performance to general LLMs for QA tasks, while CRAFT-based summarization models outperform models trained on human-curated data by 46 preference points.

1 Introduction

Large language models (LLMs) demonstrate strong generalization capabilities across diverse tasks Dubey et al. [2024], Ouyang et al. [2022], but optimizing these models for specific tasks remains a considerable challenge. Although zero-shot and few-shot prompting methods provide some degree of adaptability Dong et al. [2024], task-specific fine-tuning generally delivers better performance, particularly for specialized and out-of-domain tasks Liu et al. [2022]. A key challenge for effective fine-tuning is obtaining high-quality task-specific examples at large scale.

Traditionally, creating high-quality datasets for specific tasks involves a time-consuming and resource-intensive process, often requiring extensive manual curation and annotation (e.g. Marcus et al. [1993]). This challenge is particularly acute for low-resource domains or novel tasks where existing datasets may be limited or non-existent.

On the other hand, "raw" (i.e., unannotated, free-text) web-crawled corpora are known for their diversity and potential utility for various tasks Maini et al. [2024]. Prior work has used raw data by targeted crawling of recipe websites [Bień et al., 2020] or word-specific filtering of crawling metadata to gather examples from pre-training corpora for sentiment analysis and summarization tasks via ratings [Maas et al., 2011] and bullet point summaries found in news articles [See et al., 2017].

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

These approaches either rely on a predefined task definition based on keywords, or on the targeted crawling of websites which are expected to contain the desired content. This reliance hinders the generalization of these methods to tasks where such prior knowledge is unavailable, difficult to define, or highly context-dependent.

In this work, we propose Corpus Retrieval and Augmentation for Fine-Tuning (CRAFT) to curate task-specific samples from raw data for a wide variety of tasks. CRAFT only requires a small set of few-shot examples from a user to initiate the process of crawling and structuring task examples. CRAFT first detects relevant corpus examples from large-scale unannotated corpora using similarity-based retrieval. Then it uses LLMs to structure these examples into a proper task format, effectively transforming free-text documents into custom-formatted task samples for fine-tuning.

We demonstrate the effectiveness of CRAFT on four diverse tasks: three QA tasks – in biology, medicine and commonsense – as well as a text summarization generative task. Our results show that models fine-tuned on CRAFT-generated datasets achieve performance that is either better than or comparable to instruction-tuned LLMs. This holds across diverse tasks, LLMs, and dataset sizes, highlighting the effectiveness of our approach. We publicly release the code to craft datasets for other tasks as well as all datasets and checkpoints at github.com/ziegler-ingo/CRAFT.

2 Related work

Synthetic data refers to artificially generated data that mimics the characteristics of real-world data [Little et al., 1993]. It can be generated using statistical [Sue, 1987, Maqsud, 2015] or deep neural approaches [Sutskever et al., 2011] with the aim of replicating the patterns, distributions, and structures found in real-world datasets.

Fully synthetic data generation A dataset is fully synthetic if the question or instruction, the possible context, as well as the answers are generated synthetically. For instance, Self-Instruct [Wang et al., 2023a], Unnatural Instructions [Honovich et al., 2023], and Alpaca [Taori et al., 2023] are examples of fully synthetic general-purpose data generated by LLMs. More focused approaches for task-specific fine-tuning data generation have also been proposed, especially based around the rephrasing of already existing task-specific datasets [Yin et al., 2023, Gandhi et al., 2024]. Recently proposed methods that generate pre-training data from general-purpose corpora [Maini et al., 2024] have been adapted for fine-tuning data generation, but these adaptations either require complex multi-agent workflows [Mitra et al., 2024] or are restricted to a small set of tasks, as the generation process relies on a model that has been fine-tuned for those tasks [Nayak et al., 2024].

The two greatest drawbacks of current approaches to fully synthetic data generation are repetition and low quality. Unnatural Instructions reported that a majority of their samples have a BERTScore [Zhang et al., 2020] of above 45% when compared to other samples in the generated dataset. Self-Instruct faces similar issues, with generated instructions often having ROUGE-L scores [Lin, 2004] greater than 0.4 compared to the provided seed instructions. Both approaches also only contain about 54%-56.5% correct samples, while the correctness rate in Alpaca is as low as 17% [Chen et al., 2024a]. This suggests that a large portion of the samples in these datasets may not be useful for fine-tuning models.

Partially synthetic data generation In partially synthetic data generation, a portion of the input, context, or output is generated synthetically, while the remaining portion is human-curated. It is distinct from approaches that combine fully synthetic and purely human-curated samples at the dataset level, such as Phi [Gunasekar et al., 2023].

One recent approach is reverse instruction generation [Köksal et al., 2023], where a language model, provided with a human-curated output in context, generates the instruction that would have prompted this output. This produces more coherent and correct input-output pairs because the LLM does not need to generate the longer and more complex component of the data sample. Conversely, when the output is synthetically generated from human-curated input samples, these methods employ distillation techniques to extract patterns from larger, more capable models to teach those patterns and skills to smaller models [Mukherjee et al., 2023, Mitra et al., 2023].

Partially synthetic data generation alleviates some of the quality and diversity concerns of fully synthetic data generation. However, taking a raw corpus document as the output can result in

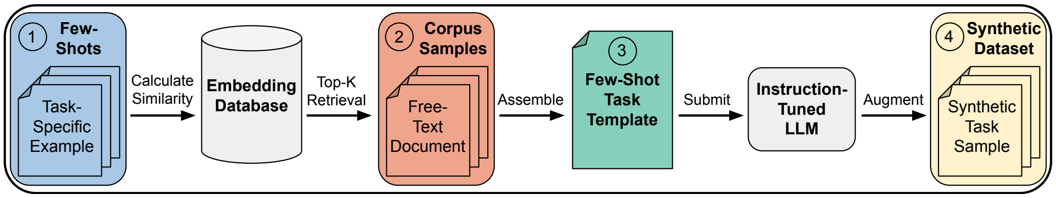

Figure 1: Synthetic dataset generation process. Given a small set of task-specific few-shots ①, we retrieve the top-k most similar free-text documents ② from an embedding database. Each document is then integrated into a task template ③ alongside original few-shots and an instruction prompt. An instruction-tuned LLM generates new synthetic task samples ④ by augmenting the content of the corpus samples to mimic the style of the few-shots. The transformation process for each numbered step is illustrated with example documents in Appendix A, Figures 4 – 7.

generating noisy or unnecessary information [Agarwal et al., 2007], but data augmentation can mitigate these concerns [Maini et al., 2024]. Augmentation techniques were already applied to generating fine-tuning data, but it required GPT-4 [OpenAI, 2023] to build an intermediate synthetic dataset to fine-tune a model that then creates the synthetic samples [Chen et al., 2024b]. This approach can result in a sample creator model that is a distilled version of the larger model's knowledge and data, while also limiting the model's task flexibility, depending on the synthesized training data.

In contrast, CRAFT produces fully synthetic data but leverages the quality and diversity advantages of human-written documents from partially synthetic data generation approaches while removing noise through augmentation. Our approach does not require intermediate datasets, nor a separately fine-tuned model, nor knowledge distillation from a larger model; instead, it relies only on a small number of human-curated examples, retrieval, and in-context learning.

3 The CRAFT approach

3.1 Architecture overview

CRAFT is used to fine-tune language models by generating task-specific synthetic datasets, given a few human-curated examples of the task. During CRAFT (see Figure 1), we retrieve human-written, free-text documents from a large collection of corpora by calculating their similarity to the provided few-shots and transforming them into the task-specific format through augmentation. The only human effort required is in writing a small number of high-quality examples of the target task. CRAFT has two phases: In the initial phase, an embedding database is created from large corpora. While this phase can be resource-intensive, its cost is incurred only once for all subsequent tasks, and it can be easily expanded with new corpora. In the second phase, the user-generated, task-specific few-shot examples are embedded, enabling the retrieval of relevant documents by calculating similarity measures between few-shots and corpus documents. Once relevant documents are retrieved, an instruction-tuned LLM is used to augment the retrieved free-text documents into a task-specific design, generating synthetic task samples in the layout that is needed for instruction-tuning (illustrated in Figure 1). Finally, the synthetic dataset is used to fine-tune a task-specific language model. We report implementation details for the whole CRAFT framework in Appendix B.

3.2 Few-shot examples

A small number of human-curated few-shots serve as the "definition" of the task, i.e., they indicate how the task is to be performed. The few-shot samples consist of three elements: (i) a long text that mirrors in language, content, and accuracy what a high-quality corpus sample from the web should look like, (ii) a natural language instruction for the task to be performed, which can take the form of a direct instruction or a question about the text, and (iii) an output that satisfies the instruction or answers the question the way that the final model should later respond. Length statistics for texts, instructions, and outputs of our few-shots can be found in the XS row of Appendix D.

We note that the task does not need to be explicitly specified. For example, there is no need to state the task as "biology question-answering"; it is sufficient for the human-curated few shots to focus on QA in the domain of biology. If multiple-choice questions or single-letter outputs are in the few-shots, this will result in a corresponding dataset and fine-tuned model behavior. These examples show that CRAFT is highly customizable: few-shot examples enable users to tailor the model's behavior to specific formats, use cases, or domains. Users can create few-shots with unique terminology, style preferences, or domain-specific constraints, optimizing the retrieval and the final model's performance for particular tasks. Appendix A, Figure 4 shows an example few-shot.

3.3 Corpora and embedding database

The embedding database is a key element of CRAFT as it provides, for all corpora, embeddings of human-written documents that should be retrievable for task-specific augmentation. It is, therefore, important that the embedding database encompasses a wide variety of linguistically and semantically diverse documents. This diversity can be achieved by including corpora that exhibit different writing styles, tones, and vocabularies. Task-specific, task-agnostic, public, and also private documents can provide a comprehensive coverage of relevant information. The more varied the documents in the embedding database, the better the coverage will be for diverse or rare tasks. Notably, CRAFT can also handle sensitive company data, as the encoding, storage, and retrieval can be performed on-site.

3.4 Document retrieval

Our retrieval system is task agnostic, both in terms of domain and complexity, in contrast to previous approaches [Ein-Dor et al., 2020, Dai et al., 2022, Lewis et al., 2021]. The CRAFT approach relies on human-curated few-shot examples as query documents and can dynamically retrieve any document of the base corpora. As the few-shot samples include a text containing the domain, the instruction or question, as well as the output, the resulting embedding representation of the sample contains contextualized [Reimers and Gurevych, 2019] semantic information about both the domain and the nature of task to be performed. Relevant text documents that contain similar latent features as the few-shots are retrieved from the corpora by calculating similarity scores based on the embedded few-shots and corpus samples.

As corpus size increases, the risk of retrieving redundant or similar corpus samples also increases. This is partly due to the growing volume of documents, but also because the diversity of documents within the corpora may plateau, resulting in a higher proportion of similar documents. Designing few-shots that are sufficiently diverse in topic may alleviate this issue. For example, when creating few-shots for biology question-answering, various subtopics of biology, such as genetics, anatomy, or physiology, should be covered to broaden the range of retrieved documents.

3.5 Task sample synthesis

The retrieved documents naturally contain noise [Agarwal et al., 2007] and lack the formatting required for fine-tuning. Therefore, it is necessary to convert these free-text documents into appropriate task samples by removing noise and undesired sections.

To address this, we utilize instruction-tuning prompt templates [Sanh et al., 2022, Maini et al., 2024] to augment the free-text documents into task-specific training data while simultaneously eliminating noise. A few-shot task template consists of three elements: (i) one or more few-shots, (ii) a corpus sample, (iii) and a brief instruction for the model to generate instruction-output pairs from the content of the corpus sample. Aside from the brief instruction, it is easy to assemble these templates from material we already have. The template only structures all information from the instruction, the few-shots, and the retrieved corpus samples to generate one continuous string that serves as input for the model generating the synthetic task samples. In this setup, the contents of the few-shots serve as in-context examples for the completion of the instruction. Appendix A, Figure 6, shows an example of how these templates guide the model in augmenting the corpus samples into synthetic task samples. Any instruction-tuned language model can be used for this purpose. This augmentation step not only rephrases the text but also condenses the retrieved document down to the essential information required for the task. The result of this step produces the final synthetic instruction-output pairs that can be used to fine-tune a language model. Appendix A, Figure 7 shows an actual example output from the generated pool of synthetic training samples, and Appendix D provides an overview of length statistics from the stages of corpus retrieval up to the synthesized input-output pairs.

4 Experimental setup

4.1 Tasks

We generate datasets for all tasks in sizes 100, 500, 5000, and 25,000. We refer to the few-shots as a dataset of size XS, and to the sizes ranging from 100 to 25,000 as S, M, L, and XL, respectively. Implementation details related to fine-tuning can be found in Appendix B.6.

We generate three synthetic QA datasets for biology (BioQA), medicine (MedQA), and commonsense (CSQA). The multiple-choice QA datasets follow the MMLU format [Hendrycks et al., 2021], where the task is to identify the correct answer from a set of options. Additionally, we create datasets for two generative tasks: summarization, which aims to convey accurate and concise information from a longer text, and recipe generation, which requires creating coherent and structured text that adheres to specific formatting and stylistic conventions [Wang et al., 2022].

4.2 Evaluation

4.2.1 Metrics

All multiple-choice QA tasks are evaluated using accuracy. We follow the evaluation approach of MMLU and assess the logarithmic probabilities for the vocabulary options corresponding to the letter labels of the answer choices [Hendrycks et al., 2021]. Accordingly, greedy decoding without temperature scaling is performed. For generative tasks, we do not use automated metrics like ROUGE [Lin, 2004] and METEOR [Banerjee and Lavie, 2005], which rely heavily on n-gram overlap and may not accurately reflect the true quality of generated text [Barbella et al., 2021, Graham, 2015, Sai et al., 2019, Sottana et al., 2023, Celikyilmaz et al., 2020].

As an alternative, we opt to evaluate generations using LLMs as a judge [Eldan and Li, 2023]. In this setup, the LLM effectively acts as a human annotator, providing a binary preference score for each pair of outputs, resulting in a win rate as the final metric [Chiang et al., 2024]. This approach has become increasingly common: LLMs have been successfully employed as annotators while demonstrating high inter-rater reliability [Hackl et al., 2023, Sottana et al., 2023, Liu et al., 2023].

For general-purpose outputs, we use the popular Alpaca-Eval benchmark [Li et al., 2023] that evaluates multiple LLMs on about 650 human-curated questions [Dubois et al., 2023]. We select Llama 3 70B [Dubey et al., 2024] as our annotator model due to its open nature and cost-efficiency for high-volume experiments. As of July 2024, Llama 3 70B ranks 4th in human agreement with a score of 67.5, close to customized GPT-4 versions at 69.2.

4.2.2 Datasets

To assess the quality of our synthetically generated datasets, we compare the performance of models trained on them to those trained on human-curated datasets.

We use five datasets: The biology subsection (800 samples) from ScienceQA [Lu et al., 2022], MedMCQA (4183 samples) [Pal et al., 2022], CommonsenseQA 2.0 (2541 samples) [Talmor et al., 2021], RecipeNLG (1000 samples) [Bień et al., 2020], and CNN-DailyMail (1000 samples) [See et al., 2017]. Specifically, BioQA and MedMCQA feature expert-curated question-answer pairs, while CSQA was generated through a standardized gamified and controlled process. RecipeNLG was post-processed using a fine-grained cleaning and formatting process to ensure correctness. The CNN-DailyMail dataset consists of news articles and their corresponding abstract or bullet-points summaries, making it a suitable benchmark for summarization tasks.

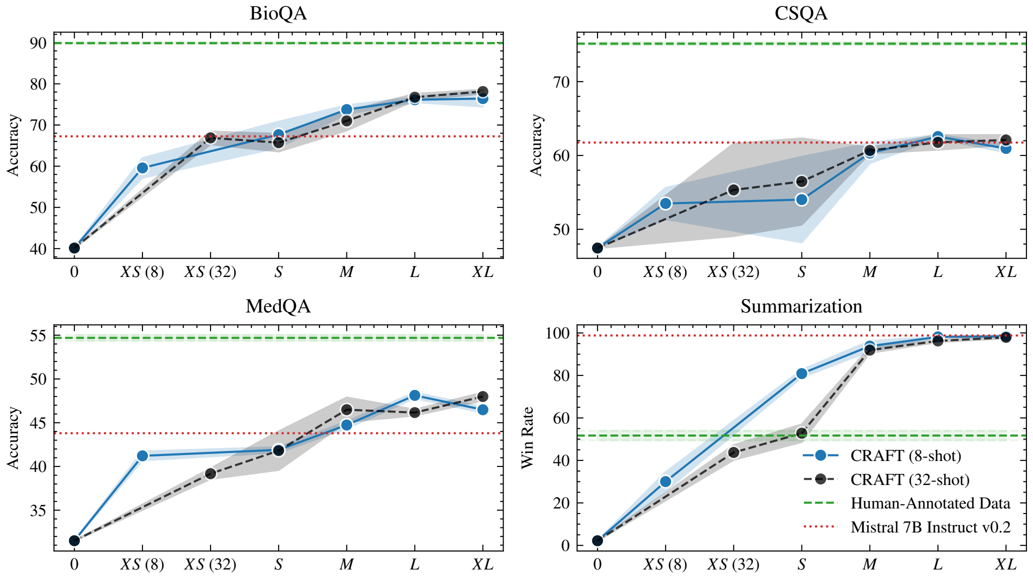

Figure 2: Performance scaling with increasing data size across multiple tasks when running CRAFT with 8 (solid blue line) and 32 (dashed black line) few-shots. The graphs illustrate consistent improvement as the training data size increases from few-shots (XS) to 25,000 synthetic task samples (XL). CRAFT-trained models consistently outperform or match the Mistral 7B Instruct v0.2 baseline (dotted red line). Shaded regions represent the standard deviation across three runs.

5 Results

5.1 Baselines

We compare the performance of CRAFT, trained on our synthetic datasets, against three baselines. The few-shot baseline is a model fine-tuned only on the XS size CRAFT dataset, with human-curated few-shots. It serves as the primary baseline since this model uses all human-curated data available in our pipeline.

The second baseline is the instruction-tuned model, Mistral 7B Instruct v0.2 [Jiang et al., 2023], which has been fine-tuned on proprietary instruction-following datasets that mix various tasks and sources. This baseline provides a meaningful comparison, as it is similar in size and instruction-tuned like CRAFT models, though it is trained on undisclosed datasets of unknown quality and quantity. Thus, matching or exceeding the performance of instruction-tuned models with our synthetic data would indicate that CRAFT can produce high-quality datasets.

The upper bound of expected performance is fine-tuning the models on the in-domain human-curated training splits from the chosen evaluation datasets. This baseline represents the optimal performance achievable with human-quality datasets.

5.2 Scaling the data

We report the performance gain when scaling up our training data in Figure 2. We report the mean and standard deviation across three seeds. We observe consistent improvements across four tasks as we increase the data size. Relative to the baseline models trained with only few-shot examples, we see improvements of 17% (from 66.9 to 78.1), 12% (from 55.3 to 62.1), 23% (from 39.1 to 48.0), and 124% (from 43.7 to 97.9) for BioQA, CSQA, MedQA, and Summarization, respectively. This shows that CRAFT can be used for diverse tasks, starting with just a few curated examples. We also find appropriate scaling for each set of examples, ranging from 100 to 25,000 across all tasks. Additionally, models trained with fewer examples (32, 100) exhibit much more variance than those trained with 5,000 and 25,000 examples, as indicated by the gray regions in the plots that visualize the standard deviation.

For all tasks, we achieve results that are clearly better or comparable to Mistral Instruct. It is worth noting that CRAFT uses an LLM in a limited way (to restructure and rewrite existing corpora) that seem to exclude the possibility that distillation may have played a role here. However, even if distillation were to be considered the reason for good CRAFT performance, the results indicate otherwise: we use the same model (Mistral Instruct) to paraphrase existing corpora examples but achieve even stronger results. Finally, we observe that CRAFT models outperform those trained with official human-curated large-scale data in Summarization. For other tasks, while we observe lower performance than with official data, we speculate that this could be due to in-domain evaluation for official human-curated data. We use their test split to evaluate our models, which may give these models an unfair advantage. We investigate this further in the next section.

5.3 OOD generalization and data contamination

We now report experiments to understand the level of data contamination or similarity between test and training examples in the experiments introduced in Figure 2. We conduct 5-gram weighted Jaccard similarity analyses between CRAFT datasets and the test data. For each sample, we combine the instruction and output and gather 5-gram frequencies for the whole dataset. We then calculate the Jaccard similarity between the 5-gram frequency distributions of the respective CRAFT and test dataset, where 5-grams receive weight proportional to their frequency.

This analysis shows that all CRAFT datasets have less than 0.4% similarity with the task test set, whereas the original human-authored datasets show much higher similarities: BioQA (17.9%); CSQA (4.4%), MedQA (1.1%), and Summarization (0.3%); this may indicate some overlap between train and test splits. Since we have curated the few-shots manually, rather than copying from existing datasets, our low overlap (0.4%) may be expected. To further investigate CRAFT's performance improvement, we select four out-of-domain datasets for the biology question answering task and compare CRAFT with the human-curated baseline.

Table 1: OOD evaluation of baseline and CRAFT$_{\mathrm{XL}}$.

Dataset	Baseline	CRAFT$_{\mathrm{XL}}$
In-domain	**89.9**	78.1
MMLU$_{\text{Medical Genetics}}$	60.0	**69.0**
MMLU$_{\text{Anatomy}}$	55.6	**57.0**
MMLU$_{\text{High School Biology}}$	**69.3**	67.4
MMLU$_{\text{College Biology}}$	66.7	**74.3**
MMLU-Avg	62.9	**66.9**

In Table 1, we compare the human-curated baseline with CRAFT on the in-domain dataset for the baseline and four out-of-domain (OOD) datasets selected from MMLU Hendrycks et al. [2021]. Although the baseline outperforms CRAFT by more than 11% in the biology subset of the ScienceQA test set (in-domain for the baseline), CRAFT outperforms the baseline on 3 out of 4 OOD datasets. On average, CRAFT outperforms the baseline by 4%.

5.4 Negative results: recipe generation

Out of the five tasks we selected, we observe non-scaling behavior in one task: Recipe Generation. While our manually curated few-shots are of high quality, we see a drop when scaling from 32 to 25,000 examples, as illustrated in Figure 3. CRAFT's performance is still better than the baseline with official human data, which means that the final dataset is usable. However, we explore why this reverse scaling occurs and examine the drop in performance.

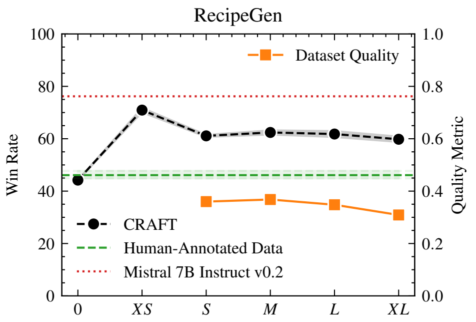

Figure 3: Non-scaling behavior observed in the RecipeGen. The graph shows an inverse relationship between dataset size and quality. Unlike the general trend, data quality decreases as synthetic samples increase.

An initial analysis suggested that the CRAFT pipeline tends to find less relevant examples over time. We conducted automated data quality analysis to analyze this on a larger scale. For 500 randomly sampled instructions from different sizes of CRAFT datasets (i.e., the training sets), we used the Llama 3 8B Instruct model to answer the instructions. Then, using

Table 2: Performance comparison of Llama and Mistral across different tasks and few-shot settings. Values show accuracy ± standard deviation, with relative improvement in parentheses compared to the CRAFT_{XS} model.

Task	**Model**	**CRAFT_{XS}**	**CRAFT_{S}**	**CRAFT_{M}**	**CRAFT_{L}**	**CRAFT_{XL}**
BioQA	Mistral	66.88±1.24	65.71±1.74 (-1.74%)	71.00±1.87 (6.17%)	76.75±0.64 (14.76%)	78.13±0.35 (16.82%)
	Llama	64.38±1.98	69.09±0.78 (7.32%)	70.38±0.57 (9.32%)	73.38±1.24 (13.99%)	76.63±1.22 (19.03%)
CSQA	Mistral	55.33±5.09	56.47±4.76 (2.06%)	60.68±0.31 (9.67%)	61.77±0.80 (11.63%)	62.10±0.54 (12.23%)
	Llama	56.72±0.82	54.60±0.44 (-3.74%)	59.57±0.85 (5.02%)	60.45±0.40 (6.58%)	61.64±0.64 (8.67%)
MedQA	Mistral	39.17±0.48	41.81±1.79 (6.72%)	46.49±1.13 (18.68%)	46.16±0.27 (17.83%)	47.99±0.36 (22.52%)
	Llama	49.65±0.35	49.70±0.07 (0.10%)	52.38±0.11 (5.49%)	52.07±0.42 (4.86%)	52.82±0.29 (6.37%)
Summ	Mistral	43.73±2.78	52.80±3.35 (20.73%)	91.90±1.02 (110.14%)	96.23±0.40 (120.05%)	97.90±0.37 (123.86%)
	Llama	31.70±3.97	28.20±3.28 (-11.04%)	88.27±1.18 (178.44%)	96.63±0.24 (204.84%)	97.17±0.38 (206.52%)

Llama 3 70B Instruct as a judge, we compared win rates, i.e., which output the model preferred: the gold output in the CRAFT dataset or the output generated by Llama 3 8B Instruct. We report the average win rate against the Llama outputs as the data quality metric. Higher scores indicate that the pipeline created higher quality output than Llama 3 8B Instruct's answers.

We observe that data quality drops when scaling up to 25,000 examples. The 100 and 500 example sets have win rates around 0.4, while the win rate for 25K drops to 0.3. We believe this is the cause of the performance drop with scaling. While the final dataset is still useful (it outperforms the baseline with official human data), the next version of CRAFT should include either effective stopping criteria or additional validators for quality.

5.5 Base model comparison

In previous sections, we fine-tuned CRAFT models using the pretrained Mistral 7B model. Now, we repeat the experiments using the pretrained Llama 3 8B model. We observe similar trends across all tasks, and the relative improvement is comparable when scaling up from few-shots to 25,000 examples, as illustrated in Table 2.

In all experiments, we manually curated 32-shot examples and expanded our synthetic data examples from that point. However, even curating 32 examples can be time-consuming. We can limit the number of few-shots to just eight examples to bootstrap the CRAFT process. Figure 2 shows the results compared to the CRAFT pipeline with 32-shots. While the final results with 25,000 examples are slightly lower for 8-shots, the trend is similar for both 8- and 32-shot examples. We observe that the model trained with 25,000 samples, based on running CRAFT with 8 few-shots, significantly outperforms the model trained with only 32 few-shots (i.e., no extra synthetic data). This suggests that if there are time and resource limitations, using CRAFT with fewer initial examples leads to better models than trying to curate and train only on more human-curated few-shot examples.

6 Conclusion

In this work, we introduced CRAFT (Corpus Retrieval and Augmentation for Fine-Tuning), a framework for generating task-specific synthetic datasets grounded in text corpora. CRAFT requires only a small set of human-curated few-shot examples to bootstrap the creation of large-scale training data by leveraging existing corpora and instruction-tuned language models. Our experiments across multiple tasks, including biology, medicine, and commonsense question-answering, as well as summarization, demonstrate that models fine-tuned on CRAFT-generated datasets can match or outperform strong baselines, including instruction-tuned models and those trained on human-curated datasets. Notably, CRAFT-based models showed better generalization capabilities on out-of-domain datasets compared to models trained on human-curated data, highlighting the robustness of our approach.

While CRAFT shows promising results for most tasks, we also identified limitations in scaling performance for recipe generation, emphasizing the need for careful quality control and potential stopping criteria in future iterations. Nevertheless, the overall success of CRAFT in producing high-quality synthetic datasets with minimal human effort opens up new possibilities for efficient and adaptable model fine-tuning across a wide range of domains and tasks.

Acknowledgments and Disclosure of Funding

IZ and DE have been supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 101135671 (TrustLLM). AK and HS have been funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project SCHU 2246/14-1. IZ was also supported by a G-Research Travel Grant to present this work at the ELLIS Doctoral Symposium 2024 in Paris.

References

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024. URL `https://arxiv.org/abs/2301.00234`.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf`.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL `https://aclanthology.org/J93-2004`.

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Data Problems for Foundation Models Workshop at ICLR*, 2024. URL `https://arxiv.org/abs/2401.16380`.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada, editors, *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28, Dublin, Ireland, December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.inlg-1.4. URL `https://aclanthology.org/2020.inlg-1.4`.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1015`.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL `https://www.aclweb.org/anthology/P17-1099`.

Roderick JA Little et al. Statistical analysis of masked data. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 9:407–407, 1993.

Leurgans Sue. Linear models, random censoring and synthetic data. *Biometrika*, 74(2):301–309, 1987.

Umar Maqsud. Synthetic text generation for sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 156–161, 2015.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024, 2011.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023a.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806. URL `https://aclanthology.org/2023.acl-long.806`.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4031–4047, 2023.

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets. *arXiv preprint arXiv:2404.14361*, 2024.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, et al. Agentinstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024.

Nihal V Nayak, Yiyang Nan, Avi Trost, and Stephen H Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv:2402.18334*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-1013`.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*, 2024a.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. Longform: Optimizing instruction tuning for long text generation with corpus extraction, 2023.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason, 2023.

Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. How much noise is too much: A study in automatic text classification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 3–12. IEEE, 2007.

OpenAI. Gpt-4 technical report, 2023.

Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. Dog-instruct: Towards premium instruction-tuning data via text-grounded instruction wrapping. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4125–4135, 2024b.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, et al. Corpus wide argument mining—a working solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7683–7691, 2020.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. Dialog inpainting: Turning documents into dialogs. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4558–4586. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/dai22a.html`.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115, 2021. doi: 10.1162/tacl_a_00415. URL `https://aclanthology.org/2021.tacl-1.65`.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Hao Wang, Guosheng Lin, Steven CH Hoi, and Chunyan Miao. Learning structural representations for recipe generation and food retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3363–3377, 2022.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://aclanthology.org/W05-0909`.

Marcello Barbella, Michele Risi, and Genoveffa Tortora. A comparison of methods for the evaluation of text summarization techniques. In *DATA*, pages 200–207, 2021.

Yvette Graham. Re-evaluating automatic summarization with bleu and 192 shades of rouge. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 128–137, 2015.

Ananya B Sai, Mithun Das Gupta, Mitesh M Khapra, and Mukundhan Srinivasan. Re-evaluating adem: A deeper look at scoring dialogue responses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6220–6227, 2019.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. Evaluation metrics in the era of gpt-4: Reliably evaluating large language models on sequence to sequence tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8776–8788, 2023.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*, 2020.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

Veronika Hackl, Alexandra Elena Müller, Michael Granitzer, and Maximilian Sailer. Is gpt-4 a reliable rater? evaluating consistency in gpt-4's text ratings. *Frontiers in Education*, 8, 2023. ISSN 2504-284X. doi: 10.3389/feduc.2023.1272229. URL `https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2023.1272229`.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, 2023.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL `https://proceedings.mlr.press/v174/pal22a.html`.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv*, 2023.

Bruce Alberts. *Molecular biology of the cell*. Garland science, 5th edition, 2017.

Sarah Malmquist and Kristina Prescott. *Human Biology*. Pressbooks, 2nd edition, 2022.

Douglas Wilkin and Jean Brainard. *Communication Behavior in Animals - Advanced*. CK-12, aug 2016.

Connie Rye, Robert Wise, Vladimir Jurukovski, Jean DeSaix, Jung Choi, and Yael Avissar. *Behavioral Biology: Proximate and Ultimate Causes of Behavior*. OpenStax, Houston, Texas, oct 2016.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

The HDF Group. Hierarchical Data Format, version 5, 2002. URL `https://github.com/HDFGroup/hdf5`.

Giuseppe Attardi. Wikiextractor. `https://github.com/attardi/wikiextractor`, 2015.

Flax Sentence Embeddings Team. Stack exchange question pairs. `https://huggingface.co/datasets/flax-sentence-embeddings/`, 2021.

Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset, 2018.

Peter Deutsch. Gzip file format specification version 4.3, 1996. URL `https://www.rfc-editor.org/rfc/rfc1952.html`.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

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

Deren Lei, Yaxi Li, Mingyu Wang, Vincent Yun, Emily Ching, Eslam Kamal, et al. Chain of natural language inference for reducing large language model ungrounded hallucinations. *arXiv preprint arXiv:2310.03951*, 2023.

Yanshan Wang, Shyam Visweswaran, Sumit Kappor, Shravan Kooragayalu, and Xizhi Wu. Chatgpt, enhanced with clinical practice guidelines, is a superior decision support tool. *medRxiv*, pages 2023–08, 2023b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, 2018.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

V Ranjith, MK Dhananjaya, P Yamini Sahukar, M Akshara, and Partho Sharothi Biswas. A review of deduplicate and significance of using fuzzy logic. *ICT Analysis and Applications*, pages 281–287, 2022.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

A Synthetic task sample generation process

① Few-Shot Design

Text: However, it has become clear that human chromosomes also carry a great deal of information that is epigenetic, and not contained in the sequence of the DNA itself. Imprinting is one example. Another is seen in the phenomenon of mono-allelic expression, in which only one of the two copies of certain human genes is expressed [. . .].

Question: What is epigenetic inheritance, and what are two examples of epigenetic changes?

Options: A. Epigenetic inheritance signifies any heritable difference in the phenotype [. . .]
B. Epigenetic inheritance refers to inheriting variations in the number of chromosomes [. . .]
C. Epigenetic inheritance implies inheriting acquired traits during lifetime, whereas two [. . .]
D. Epigenetic inheritance denotes acquiring beneficial mutations via natural selection, [. . .]

Answer: A.

Figure 4: Few-shot design. The layout of a user-written few-shot sample that is used to guide the retrieval and task sample creation process. The color coding indicates where the section is reused throughout the process in Figure 6. For readability, we shorten text sections in this figure, indicated by "[. . .]".

② Corpus Sample

Text: Proteins are involved in the formation of higher-ordered chromosome structures, such as chromosome loops. Some proteins, including special AT-rich sequence-binding protein-1 (SATB1), CCCTC-binding factor (CTCF) and cohesin, play key roles in disease development and recovery.

Figure 5: Corpus sample. A retrieved free-text document from the embedding database based on cosine similarity to the user-written few-shot. The color coding indicates where the section is reused throughout the process in Figure 6. For readability, we shorten text sections in this figure, indicated by "[. . .]".

③ Few-Shot Task Template

`<s>[INST]` Please carefully read the text below. Then, generate exactly one question along with four answer choices designated as A, B, C, and D based on the provided text. Then, respond to the question with the correct answer using only the corresponding letter label. Return the output only as a JSON structure in this format: `{"question": [...], "options": [...], "answer": [...]}`
However, it has become clear that human chromosomes also carry a great deal [. . .] `[/INST]`
`{"question": [...], "options": [...], "answer": [...]}</s>`

Repeat for randomly sampled few-shots 2 and 3

`<s>[INST]` Please carefully read the text below. Then, generate exactly one question [. . .]
Proteins are involved in the formation of higher-ordered chromosome structures [. . .] `[/INST]`

Figure 6: Few-shot task template. The prompting template that is used to augment the retrieved corpus sample into a synthetic task sample by using multiple few-shots as in-context examples. The color coding indicates where previous sections from Figures 4 and 5 as well as current sections are reused throughout the process. For readability, we shorten text sections in this figure, indicated by "[. . .]".

④ Synthetic Task Sample

Question: Which proteins play key roles in the formation of higher-ordered chromosome structures and disease development?

Options: A. SATB1, CTCF, and cohesin
B. Histone proteins only
C. Transcription factors and co-TFs
D. RNA polymerase II and transcription factors

Answer: A.

Figure 7: Synthetic task sample. This is an actual synthetic task sample that is generated from the corpus sample in Figure 5 using the few-shot task template of Figure 6.

B Implementation details

B.1 Few-shot design

The BioQA few-shot text samples were compiled from a diverse range of sources, including textbooks [Alberts, 2017, Malmquist and Prescott, 2022, Wilkin and Brainard, 2016, Rye et al., 2016], openly accessible materials, and Encyclopedia Britannica. For MedQA, we primarily drew upon openly accessible websites, such as the National Institutes of Health, National Health Service, Food and Drug Administration, Mayo and Cleveland Clinics, and other medicine-related websites, to generate our few-shot text samples. The CSQA few-shot text samples were sourced from a variety of online resources, including blogs, articles, and other websites tailored to the specific topic at hand. From sources that are not openly accessible through websites, continuous text snippets were directly extracted and used as texts, while all other material was shortened, rephrased, or restructured by the authors. This process ensures that articles which may have been crawled through C4 [Raffel et al., 2020] do not produce exact matches at retrieval time. Since none of these texts have direct question-answer pairs available, the question, answer options, and the answer were generated by the authors for each sample. Figure 4 shows an example.

The recipe generation few-shots were taken from blogs and other openly accessible websites. Their design always includes a simple instruction or question to cook a meal, as well as bulleted or numbered lists of ingredients and steps to produce the meal. Sometimes, the recipes on websites were described only as continuous text. In such cases, the authors added the list of ingredients and steps to create the few-shots. Similar to the QA few-shots, it is desirable to increase the vocabulary diversity for retrieval. Therefore, we made sure to cover a wide range of recipes in the few-shots. To design summarization few-shots, we collected a wide variety of texts from websites, blogs, and magazines, as well as specialized sources such as GitHub issues. Here, the few-shots feature a text, an instruction to create a summary, as well as the summary output. On some websites, summarized versions of the main text are given, while in other cases continuous or bulleted summaries were created by the authors.

B.2 Embedding database

To create the embeddings, SentenceTransformers [Reimers and Gurevych, 2019] is used, specifically a MiniLM [Wang et al., 2020] version (`multi-qa-MiniLM-L6-cos-v1`) optimized for cosine-similarity search between two document pairs. This model creates 384D embeddings, which are stored in an HDF5 database [The HDF Group, 2002], that allows for native storage and retrieval of array-like data.

B.3 Corpora

To enable retrieving human-written documents for general-purpose as well as specialized domains, we include four large corpora.

The Colossal Clean Crawled Corpus [Raffel et al., 2020, C4] dataset consists of approximately 750GB of English data, pre-cleaned by the creators to exclude non-informative texts. We use a 305GB subset of the dataset, which excludes not-safe-for-work or offensive content.

The English Wikipedia corpus comprises a diverse and high-quality collection of textual information on a wide range of topics. We use the January 2nd, 2024 dump of English Wikipedia, which we pre-process using WikiExtractor [Attardi, 2015] to extract clean text documents.

The Stack Exchange corpus [Flax Sentence Embeddings Team, 2021] features a structured format with title, body, and best-voted answer collected from the Stack Exchange network. The dataset encompasses 173 distinct sub-communities, covering both technical and non-technical topics.

The WikiHow corpus [Koupaee and Wang, 2018] presents information in a step-by-step instructional layout, making it a valuable resource for tasks such as summarization or recipe generation. Each document consists of a title, clear instructions, and accompanying text.

After filtering out documents with fewer than 200 characters or more than 25,000 characters, the resulting datasets contain approximately 362 million documents from C4, 10.5 million documents from English Wikipedia, 9.5 million documents from Stack Exchange, and 190,000 documents from WikiHow. The resulting 383 million documents take up approximately 247GB of storage when GZIP compressed [Deutsch, 1996] and represented as 16-bit NumPy arrays [Harris et al., 2020].

B.4 Document retrieval

To retrieve relevant human-written samples, we employ a two-step process that approximates a global similarity search. Due to the large size of the embedding database (383 million documents), a direct global pair-wise comparison is impractical.

In the first step, we divide the embedding database sequentially into subsections of approximately 350,000 documents each. We calculate cosine similarity between each subsection and the few-shot samples, and store the top 5% most similar documents in memory. This reduces the number of documents considered from 383 million to approximately 19 million. In the second step, we recalculate cosine similarity and perform traditional top-k similarity retrieval on the filtered documents. This yields the indices of the top-k most similar documents to the few-shot examples.

Optimizing solely for top-k similarity between individual few-shot samples and the embedding database may lead to limited variation and high similarity among the retrieved documents. Conversely, optimizing for similarity by averaging the embedding representations of all few-shot samples may prioritize a single dominant topic, potentially biasing the selection. To balance these approaches, we adopt a mixed similarity retrieval strategy, combining two methods: (i) 50% of samples are retrieved based on individual top-k similarity to each few-shot sample, ensuring a minimum number of similar samples for each few-shot and mitigating topic dominance, and (ii) the remaining 50% of samples are retrieved based on fully averaged similarity, which aims to help uncover more latent topics.

B.5 Task sample synthesis

To facilitate the creation of high-quality synthetic task samples, we align the text generation process to the few-shot design through in-context learning [Brown et al., 2020]. This approach helps reduce issues like hallucinations [Lei et al., 2023, Wang et al., 2023b] and formatting errors, although it does not entirely prevent them. Specifically, three few-shot examples from the human-curated few-shots are dynamically and randomly sampled for each input prompt and corpus sample, interleaving them with the instruction to augment the text. Since each few-shot example includes a text similar in length to corpus samples, the three-shot prompting technique can result in input prompts that often exceed 10,000 tokens, sometimes reaching 20,000 tokens. For optimal performance, it is recommended to use models offering long context lengths.

In our experiments, all task samples are created using Mistral 7B Instruct v0.2 [Jiang et al., 2023] along with the few-shot task template shown in Figure 6. During the generation process, vLLM [Kwon et al., 2023] is used with temperature 0.7, top-k [Fan et al., 2018] and top-p [Holtzman et al., 2020] sampling at 40 and 0.9, respectively. The maximum generation length was determined based on preliminary experiments with the goal of accommodating full-length sample generation. For all QA datasets we limited samples to 256 tokens, whereas recipe generation and summarization were limited at 1280 and 1536 tokens, respectively.

To enable quality control checks such as checking whether the QA datasets have the desired number of answer options, or whether the samples have been fully generated, we instruct the model to produce all task samples as valid JSON objects with a set number of keys and proper naming. We perform checks on each JSON object to ensure that the questions, instructions, and generative task answers exceed a reasonable minimum length, as well as that the answer option generated for the QA tasks contain valid letters. If no valid JSON is found, or keys are missing, or any formatting check fails, we discard the sample. Furthermore, we filter task samples that are either too similar to the few-shots or to other generated task samples using fuzzy string matching [Ranjith et al., 2022] at a token set similarity ratio of 0.85. To account for the removal of samples during deduplication, we recommend retrieving approximately twice as many corpus samples as the number of desired final synthetic task samples. A detailed breakdown of the retrieval and deduplication quantities for each task is provided in Appendix C.

B.6 Training and optimization

For all experiments, low-rank adaptation [Hu et al., 2021, LoRA] fine-tuning is performed with 16-bit BrainFloat [Abadi et al., 2016] as the computation data type. All experiments are implemented using PyTorch [Paszke et al., 2019] and the Hugging Face packages [Wolf et al., 2020]. For optimization, the adaptive momentum optimizer with decoupled weight decay [Loshchilov and Hutter, 2019] of 5% and a learning rate of 1×10^{-4} is employed. A linear learning rate schedule is applied with a warm-up ratio of 10%. Fine-tuning is performed for 3 epochs across all tasks and dataset size variations. When fine-tuning on only the human-curated few-shots, a batch size of 2 is adopted. In all other scenarios, fine-tuning is performed with a batch size of 16 or the equivalent with gradient accumulation steps. Following Dettmers et al. [2024], LoRA adapters are used on every linear layer, specifically on the query-, key-, value-, output-, gate-, up-, and down-projection matrices. We use a rank and α parameter of 64, the adapter's dropout rate is set to 0.1, and the bias terms of the update matrices are deactivated. In total, this adds 2.3%, or about 160 million parameters, of the model's 7 billion base parameters as LoRA adapters. The rest of the model remains frozen during fine-tuning, and the updated parameters will be merged with the base model after fine-tuning.

C Filtering statistics

Table 3: BioQA corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

Filter Criteria	**S**	**M**	**L**	**XL**
Retrieved Corpus Samples	240	1,200	12,000	60,000
Exact duplicates	25	37	819	8,551
Excessive length	2	14	266	1,632
Format errors	10	40	466	2,174
Similarity to few-shots	0	1	22	45
Similarity to other task samples	9	117	1,469	5,961
Available synthetic task samples	194	991	8,958	41,637

Table 4: CSQA corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

Filter Criteria	**S**	**M**	**L**	**XL**
Retrieved Corpus Samples	240	1,200	12,000	60,000
Exact duplicates	24	30	165	1,348
Excessive length	2	8	64	307
Format errors	5	30	364	1,879
Similarity to few-shots	11	19	141	410
Similarity to other task samples	14	129	2,655	17,749
Available synthetic task samples	184	984	8,611	38,307

Table 5: MedQA corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

Filter Criteria	**S**	**M**	**L**	**XL**
Retrieved Corpus Samples	240	1,200	12,000	60,000
Exact duplicates	24	24	50	773
Excessive length	1	10	141	890
Format errors	15	36	540	2,911
Similarity to few-shots	0	10	55	204
Similarity to other task samples	3	40	813	5,221
Available synthetic task samples	197	1,080	10,401	50,001

Table 6: RecipeGen corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

Filter Criteria	**S**	**M**	**L**	**XL**
Retrieved Corpus Samples	240	1,200	12,000	60,000
Exact duplicates	24	24	28	620
Excessive length	1	1	20	54
Format errors	87	417	4,035	18,684
Similarity to few-shots	6	18	111	389
Similarity to other task samples	0	7	473	3,711
Available synthetic task samples	122	733	7,333	36,542

Table 7: Summarization corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

Filter Criteria	**S**	**M**	**L**	**XL**
Retrieved Corpus Samples	240	1,200	12,000	60,000
Exact duplicates	24	24	19	101
Excessive length	34	189	1,793	8,964
Format errors	55	336	3,119	14,803
Similarity to few-shots	21	28	99	379
Similarity to other task samples	1	1	32	394
Available synthetic task samples	105	622	6,938	35,359

D Dataset token count statistics

Table 8: Dataset token count statistics. TS is short for task sample. For summarization, the instruction includes the model-generated long but cleaned text augmentation from a corpus sample that will subsequently be summarized.

Dataset	**Size**	**Corpus Samples**		**TS Instruction**		**TS Output**	
		Mean	Median	Mean	Median	Mean	Median
BioQA	XS	1,109	1,088	93	91	1	1
	S	1,786	1,170	83	77		
	M	1,858	1,093	76	64		
	L	2,033	1,038	80	69		
	XL	2,122	972	86	77		
CSQA	XS	1,496	1,444	25	26	1	1
	S	1,265	851	25	25		
	M	1,399	884	26	25		
	L	1,324	864	26	25		
	XL	1,300	848	27	26		
MedQA	XS	1,755	1,815	117	118	1	1
	S	1,612	1,203	85	77		
	M	1,577	1,053	79	67		
	L	1,599	1,013	78	68		
	XL	1,691	1,001	81	71		
RecipeGen	XS	1,277	1,223	16	16	593	528
	S	1,168	823	20	19	433	363
	M	1,107	807	24	22	369	326
	L	1,058	786	24	23	355	319
	XL	1,005	754	24	23	345	316
Summarization	XS	1,595	734	1,019	690	82	61
	S	1,442	829	612	442	107	92
	M	1,440	852	471	366	116	106
	L	1,396	880	432	358	122	110
	XL	1,369	882	433	355	117	107