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

Ingo Ziegler[◇⋆] **Abdullatif Köksal**[†‡⋆] **Desmond Elliott**[◇] **Hinrich Schütze**[†‡]

[◇]Department of Computer Science, University of Copenhagen
[†]Center for Information and Language Processing (CIS), LMU Munich
[‡]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 these examples, CRAFT uses 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, medicine, and commonsense question-answering (QA), as well as summarization. Our experiments show that CRAFT-based models outperform or match general LLMs on QA tasks, while exceeding models trained on human-curated summarization data by 46 preference points. CRAFT outperforms other synthetic dataset generation methods such as Self- and Evol-Instruct, and remains robust even when the quality of the initial few-shots varies.

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). 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. Moreover, CRAFT not only outperforms other fully synthetic data generation methods, such as Self-Instruct (Wang et al., 2023) and Evol-Instruct (Xu et al., 2023), but also exhibits robustness to variations in the quality of the initial few shots. 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

2.1 Optimizing LLMs for Specific Tasks

Prompting: Prompts are added to the input to provide additional context that guides the computation and output of a model (Gu et al., 2023). A prompt usually takes the form of a natural language instruction (Radford et al., 2019; Brown et al., 2020). Prompting is commonly used with instruction-tuned models to define tasks and extract responses from language models, using natural language, without gradient updates.

Zero-Shot Inference: Originally discovered in the vision domain, zero-shot inference (Larochelle et al., 2008) is a technique that allows models to generalize their learned knowledge from pre-training to previously unseen classes, tasks, or sample instance variants at inference time without gradient updates. Pre-training LLMs on large corpora produces semantic representations that are generally applicable to multiple downstream tasks. GPT-2 (Radford et al., 2019) demonstrated that the acquired capabilities can then be activated by prompting a new task in natural language. However, zero-shot inference often falls short of the performance achieved by few-shot learning (Brown et al., 2020).

Few-Shot Learning: In few-shot learning, the model is provided with a small number of task-specific examples at inference time. The few-shot examples are given to the model in the prompt, in a technique known as in-context learning (Brown et al., 2020). While full fine-tuning generally requires a substantial amount of labeled data, few-shot learning offers an inexpensive alternative to adapt a model to a new task with a limited number of examples (Dong et al., 2024). Nonetheless, few-shot learning faces several challenges, including inaccurate assessment of the underlying data distribution (Song et al., 2023), biases related to small sample sizes (Song et al., 2023), and sensitivity to shot length (Liu et al., 2024), shot quality and noise (Perez et al., 2021; Chang et al., 2021; Chen et al., 2022).

Full Fine-Tuning: During full fine-tuning, all model parameters are updated on a large dataset with the goal of adapting the model to a domain, task or dataset (Howard and Ruder, 2018). This approach usually provides the best performance by learning task-specific patterns and relationships that may not be captured by pre-training and zero- or few-shot learning alone. However, it requires a dataset of appropriate size.

Instruction Tuning: Instruction tuning (Wei et al., 2022) is a type of full fine-tuning that optimizes a model to produce more relevant answers to questions or instructions (Leike et al., 2018; Askell et al., 2021). This approach enables language models to understand and follow user intents rather than simply continuing the input text. Instruction-tuned models produce answers preferred by humans for tasks ranging from question-answering to summarization (Ouyang et al., 2022). The challenge is obtaining a large high-quality dataset that is both task-specific and in the desired instruction-output format.

Low-Rank Adaptation: Full fine-tuning may be too expensive for LLMs but the difference between pre-trained weights and their fine-tuned counterparts often has low rank (Li et al., 2018; Aghajanyan et al., 2021). Low-Rank Adaptation (Hu et al., 2021, LoRA) approximates these low-rank matrices during fine-tuning, and is efficient because it freezes the full model and only learns the low-rank matrices, which typically results in learning the equivalent of 2% of the model's parameters.

2.2 Synthetic Data Generation

Synthetic data refers to artificially generated data that mimics the characteristics of real-world data (Little et al., 1993). It can be generated us-

ing 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. Methods such as Self-Instruct (Wang et al., 2023) and Evol-Instruct (Xu et al., 2023) generate general-purpose datasets by prompting LLMs to create task instructions and corresponding outputs. Other approaches focus on generating task-specific fine-tuning data through the rephrasing of existing datasets (Yin et al., 2023; Gandhi et al., 2024) or on synthesizing pre-training data from general-purpose corpora (Maini et al., 2024). When applied to fine-tuning, these methods are either implemented via complex, resource-intensive multi-agent workflows (Mitra et al., 2024) or are constrained to a narrow set of tasks because the generation process relies on models fine-tuned for those specific tasks (Nayak et al., 2024).

Two primary challenges of current approaches to fully synthetic data generation are repetition and inconsistent data quality. Evaluations have indicated that many samples in fully synthetic datasets tend to exhibit high similarity to one another or to the seed samples (Honovich et al., 2023; Wang et al., 2023). Moreover, reported correctness rates for these datasets suggest that a substantial portion of the generated samples may not meet desired correctness standards (Chen et al., 2024a). These observations highlight areas where fully synthetic data generation can be refined.

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. There are also approaches where, conversely, the output is synthetically generated from human-curated input samples. Such methods employ distillation to extract patterns from larger models to teach those patterns and skills to smaller models (Mukherjee et al., 2023; Mitra et al., 2023).

Partially synthetic data generation can alleviate some of the quality and diversity concerns inherent to fully synthetic approaches. However, using a raw corpus document as the output may introduce noisy or unnecessary information (Agarwal et al., 2007). Data augmentation has been shown to mitigate these issues for pre-training data generation (Maini et al., 2024). When applied to fine-tuning data, though, such augmentation typically requires a powerful model (e.g., GPT-4 (OpenAI, 2023)) to construct an intermediate synthetic dataset for fine-tuning a sample creator model (Chen et al., 2024b). This multi-stage approach can lead to a sample creator model that essentially distills the larger model's knowledge, potentially limiting task flexibility based on the synthesized training data. Alternatively, some methods have started to use retrieval to enhance diversity: SynthesizRR (Divekar and Durrett, 2024) uses knowledge distillation from larger models while incorporating retrieved documents as in-context examples. RADA (Seo et al., 2024) relies on already structured input-output pairs from other existing datasets and retrieves them to improve the generation prompt for low-resource tasks.

In contrast, CRAFT streamlines this process by operating directly on unstructured, free-text corpora. Our approach 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. CRAFT 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 col-

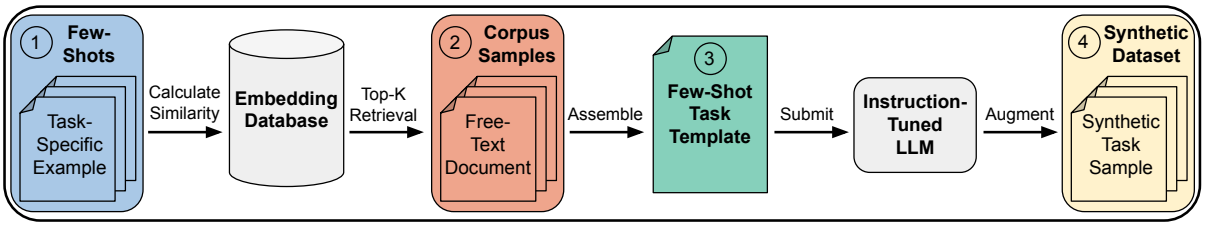

Figure 1: Synthetic dataset generation process. Given a small set of task-specific few-shots (1), we retrieve the top-k most similar free-text documents (2) from an embedding database. Each document is then integrated into a task template (3) alongside original few-shots and an instruction prompt. An 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 Figure 2.

lection 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 A.

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 E.

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.

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.

(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, and [...]

Answer: A.

(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.

(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 of [...] `[/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 along [...]
Proteins are involved in the formation of higher-ordered chromosome structures, [...] `[/INST]`

(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 2: Step-by-step synthetic task sample generation process for BioQA. The color coding indicates where each section is reused throughout the process. For readability, we shorten text sections in this figure, indicated by "[...]". (1) Few-shot design: the layout of a user-written few-shot sample that is used to guide the retrieval and task sample creation process. (2) Corpus sample: a retrieved free-text document from the embedding database based on cosine similarity to the user-written few-shot. (3) 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. (4) Synthetic task sample: this is an actual synthetic task sample that is generated from the corpus sample (2) using the few-shot task template (3).

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 the 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 free-text documents into task-specific training data while eliminating noise. A few-shot task template consists of three elements: (i) one or more few-shots, (ii) a corpus sample, and (iii) a brief instruction for the model to generate instruction-output pairs from the corpus sample. The template structures all information from the instruction, the few-shot examples, and the retrieved corpus sample, which together serves as input for the model that generates synthetic task samples.

Figure 2, step 3, shows an example of how these templates guide the model in augmenting the corpus samples into synthetic task samples. This augmentation step not only rephrases the text but also condenses the retrieved document down to the essential information required for the task. This step produces final synthetic instruction-output pairs that can be used to fine-tune a language model. Figure 2, step 4, shows an actual example output from the generated pool of synthetic training samples, and Appendix E provides an overview of length statistics from the stages of corpus retrieval up to the synthesized input-output pairs.

4 Experimental Setup

This section summarizes how we implemented the CRAFT pipeline, the tasks on which we evaluate, dataset details for both CRAFT and the human-annotated baselines, model baselines and training details, as well as the evaluation metrics.

4.1 CRAFT Implementation Overview

We construct the embedding database from four large-scale corpora: C4 (Raffel et al., 2020), English Wikipedia, Stack Exchange (Flax Sentence Embeddings Team, 2021), and WikiHow (Koupaee and Wang, 2018). After filtering documents by length (200-25,000 characters), this collection comprises 383 million documents. We generate 384D embeddings for each document using SentenceTransformers (Reimers and Gurevych, 2019) with MiniLM (Wang et al., 2020) version `multi-qa-MiniLM-L6-cos-v1`.

During the retrieval phase, cosine similarity is computed between each few-shot example and the documents in each corpus shard (approx. 350K documents), retaining the top 5% from each shard. Then, a global top-k selection was performed over the retained candidates. We employ a mixed strategy to balance topic coverage and specificity: 50% of documents are retrieved based on their top-k cosine similarity to individual few-shot examples, while the other 50% are retrieved based on similarity to the averaged embedding of all few-shots.

For task samples synthesis, we use Mistral 7B Instruct v0.2 (Jiang et al., 2023) with temperature=0.7, top-k=40 (Fan et al., 2018), and top-p=0.9 (Holtzman et al., 2020). Input prompts used three randomly sampled few-shots. To ensure data quality and structural integrity, we generate outputs as JSON objects, which allows for format validation. Finally, we enhance dataset diversity

by performing a deduplication step, filtering out generated samples that are highly similar to the seed few-shots or to each other using fuzzy string matching (Ranjith et al., 2022) with a token set similarity ratio of 0.85. Further implementation details and filtering statistics are provided in Appendices A and D, respectively.

4.2 Tasks and Datasets

To test the performance of the CRAFT pipeline, we evaluate it on five different tasks: multiple-choice (MC) biology QA, MC commonsense QA, MC medicine QA, summarization, and recipe generation. We first describe the details of CRAFT and then introduce the datasets for the human-annotated baselines.

4.2.1 CRAFT Datasets

We generate datasets for all tasks with sizes of 100, 500, 5,000, and 25,000. We refer to the few-shot datasets (with 8 or 32 examples) as size XS, and to the datasets with 100, 500, 5,000, and 25,000 examples as sizes S, M, L, and XL, respectively. For few-shot curation, we do not refer to any existing datasets; instead, the authors manually curate examples from corpora (see Appendix A.1 for sources used).

Multiple-Choice QA: We generate three synthetic QA datasets in the domains of biology (BioQA), commonsense (CSQA), and medicine (MedQA). All datasets follow the MMLU multiple-choice format (Hendrycks et al., 2021), where each question is accompanied by several answer options. Exactly one of these options is correct. The task is to output the correct answer, the letter label corresponding to the correct option.

Generation: We develop two synthetic datasets for the generation tasks of recipe generation (RecipeGen) and summarization. The goal of summarization is to convey accurate and concise information, while recipe generation focuses on creating coherent and structured text that adheres to specific formatting and stylistic conventions (Wang et al., 2022).

To build a synthetic summarization dataset, we first select a corpus sample and instruct the model to extract an extended section of text. In the second step, the extracted section is transformed into a summary, *optionally incorporating elements from the raw text* (e.g., 'abstract:', 'conclusion:', or 'TLDR'). This approach avoids using full corpus samples as the text to be summarized, as they can be lengthy and overly broad, potentially resulting in uninformative summaries. For recipe generation, our goal is to generate a list of ingredients and cooking steps for a specific recipe.

4.2.2 Human-Annotated Datasets

We select human-annotated datasets as optimal baselines for each task to compare the performance of models trained on CRAFT. For evaluation, we use the test split of these datasets, as detailed in §4.5; therefore, the human-annotated datasets can also be considered in-domain.

BioQA: We use the biology subset of ScienceQA (Lu et al., 2022) with 1,192 training samples without images. ScienceQA sourced expert-curated question-answer pairs from online learning platforms, ensuring high quality and accuracy. The dataset's answer options range from two to five, include a single correct answer per question, and are randomized to prevent pattern recognition.

MedQA: We use MedMCQA (Pal et al., 2022) and randomly select 25,000 samples from the training split. The dataset comprises entrance exam questions from two of India's postgraduate institutions. All samples come from preparation materials or real exams created by medical professionals, with each question containing four answer options and one correct answer.

CSQA: We use CommonsenseQA 2.0 (Talmor et al., 2021) and select 9,264 samples from the training split. The dataset was generated through a gamified yet controlled question generation process where players earned points by designing challenging yes/no questions that outperform an AI model. Generated questions were validated by other players, independent validators, and another model to ensure they were well-formed, answerable, and representative of common sense.

RecipeGen: We use RecipeNLG (Bień et al., 2020) and select 25,000 samples from the training split. The recipes were scraped from cooking websites and post-processed through fine-grained cleaning and formatting to ensure correctness. Each recipe includes a title, ingredient list, and cooking steps. We exclude samples present in C4 based on provided URLs.

Summarization: We use CNN-DailyMail (See et al., 2017) and randomly select 25,000 samples from the training split. This dataset is commonly used for summarization tasks as its CNN and DailyMail articles have highlights presented in abstract or bullet point formats.

4.3 Baselines

We compare CRAFT models trained on synthetic data against multiple baselines:

Few-shot is fine-tuned only on XS-size CRAFT datasets containing human-curated few-shot examples. This serves as our primary baseline, representing all human-curated data in our pipeline.

Instruct: Mistral 7B Instruct v0.2 (Jiang et al., 2023) is instruction-tuned on proprietary instruction-following datasets. This provides a meaningful comparison to a similarly-sized instruction-tuned model, though trained on undisclosed data of unknown quality and quantity. Surpassing this baseline would indicate CRAFT's ability to generate high-quality synthetic data.

In-Domain (ID) is fine-tuned on human-curated training splits from evaluation datasets. This baseline represents the optimal performance achievable with human quality datasets.

We additionally compare CRAFT against two popular synthetic data generation methods:

Self-Instruct (Wang et al., 2023) distills new task samples from instruction-tuned models, typically using fewer than 10 seed examples per task.

Evol-Instruct (Xu et al., 2023) generates new samples by rewriting seed examples to be more complex, to incorporate more domain-specific concepts, or to add step-by-step reasoning.

We initialize these methods with 8 few-shot examples to follow Self-Instruct's setup and also generate four dataset sizes (S: 100, M: 500, L: 5,000, XL: 25,000) to follow CRAFT's approach. We use each method's original hyperparameters and the same seed examples.

4.4 Training and Optimization

We fine-tune base models using either CRAFT datasets or human-annotated datasets. Models fine-tuned with human-annotated datasets are denoted with the suffix -ID (in-domain), while models fine-tuned with CRAFT datasets are labeled as -CRAFT$_{\#}$, where # represents the training split size (XS, S, M, L, or XL). Our primary experimental setting uses Mistral 7B v0.2 as the base model with 32 few-shot examples in CRAFT. Additional experiments may use 8-shot settings or another base model, Llama 3 8B, specified explicitly when used. We also evaluate (as baselines) instruct versions of base models without fine-tuning.

For all experiments, low-rank adaptation (Hu et al., 2021, LoRA) fine-tuning is performed using 16-bit BrainFloat (Abadi et al., 2016) as the computation type. All implementations use PyTorch (Paszke et al., 2019) and HuggingFace libraries (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%. Models are fine-tuned for three epochs across all tasks and dataset sizes for CRAFT. When training only on human-curated few-shots, we adopt a batch size of 2; otherwise, we use batch size 16 or equivalent gradient accumulation steps. Following Dettmers et al. (2024), we apply LoRA adapters to every linear layer (query-, key-, value-, output-, gate-, up- and down-projection matrices) with rank 64 and α 64 and 0.1 dropout rate. Bias terms in update matrices are deactivated. This configuration adds 2.3% (160 million parameters) to the base model's 7 billion parameters as LoRA adapters. Frozen base parameters remain unchanged during training, with updated parameters merged post-training.

4.5 Evaluation

4.5.1 Metrics

QA Tasks: We evaluate multiple-choice QA tasks using *accuracy*, following MMLU's approach of assessing logarithmic probabilities for vocabulary tokens corresponding to answer labels (Hendrycks et al., 2021). We perform greedy decoding without temperature scaling across answer choices ranging from A-B to A-E.

Generation Tasks: While automated metrics like ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) are efficient, their reliance on n-gram overlap limits effectiveness for generative tasks (Barbella et al., 2021). Reference text quality issues and length disparities further reduce reliability (Graham, 2015; Sai et al., 2019; Celikyilmaz et al., 2020). For example, Sottana et al. (2023) found human reviewers often rank benchmark answers among the worst options.

We instead evaluate generations using *LLMs as judges* (Eldan and Li, 2023), where models provide binary preference scores for output pairs, yielding win rates as metrics (Chiang et al., 2024). This approach shows high inter-rater reliability comparable to human annotators (Hackl et al., 2023; Sottana et al., 2023; Liu et al., 2023).

For general-purpose outputs, we use the popular

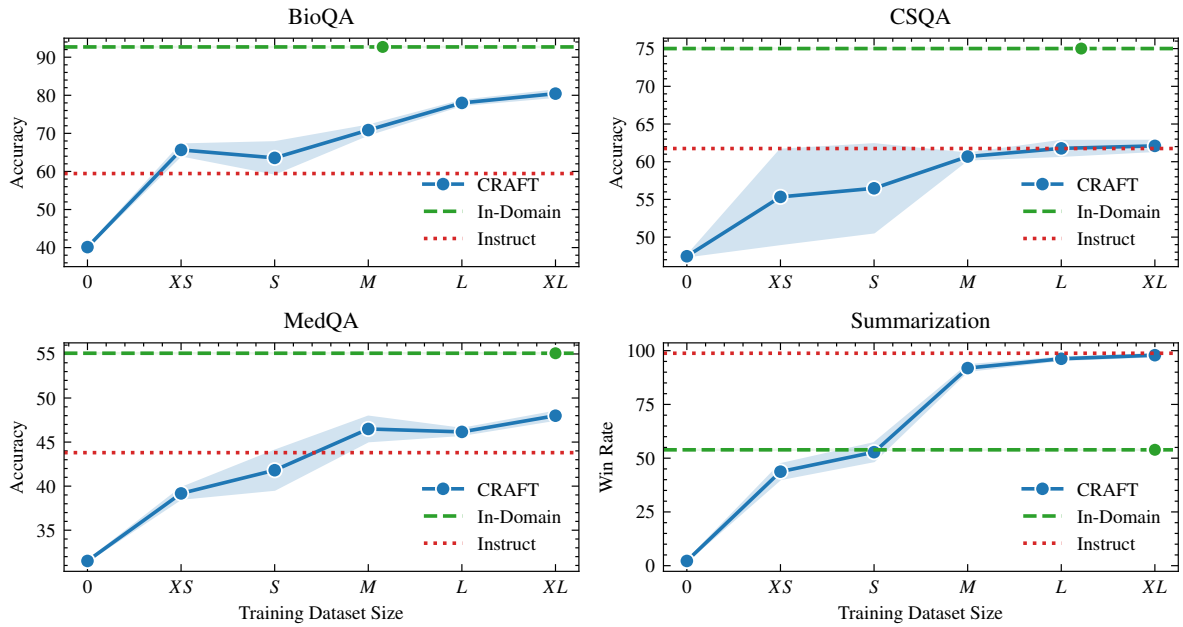

Figure 3: Performance scaling with increasing data size across multiple tasks using CRAFT with 32 few-shot examples. Graphs demonstrate consistent improvements as training data grows from few-shot (XS) to 25,000 synthetic samples (XL). CRAFT models consistently match or exceed Instruct performance (dotted red line). Shaded regions indicate standard deviation across three runs.

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 January 2025, Llama 3 70B ranks 4th in human agreement with a score of 67.5, close to customized GPT-4 versions at 69.2.

4.5.2 Datasets

We benchmark all models on the original test splits of the human-annotated datasets used for the in-domain (ID) baselines:

BioQA: 397 test samples without images from ScienceQA's biology subset (Lu et al., 2022).

MedQA: 4,183 validation samples from MedMCQA (Pal et al., 2022).

CSQA: 2,541 validation samples from CommonsenseQA 2.0 (Talmor et al., 2021).

RecipeGen: 1,000 high-quality samples from RecipeNLG (Bień et al., 2020).

Summarization: 1,000 test samples from CNN-DailyMail (See et al., 2017).

5 Results

5.1 Scaling the Data

Figure 3 shows performance improvements from data scaling, reporting means and standard deviations over three random seeds. We observe consistent gains across four tasks with increasing data size relative to the few-shot baseline: 22% (from 65.7 to 80.4), 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, from XS to XL. These results demonstrate CRAFT's effectiveness across diverse tasks starting from minimal curated examples. Models show appropriate scaling from 100 to 25,000 synthetic samples. Additionally, models trained with fewer examples (32, 100) exhibit higher variance than those trained with 5,000 and 25,000 examples, as indicated by the shaded regions that visualize the standard deviation in the plots.

Notably, CRAFT matches or exceeds Instruct performance across all tasks except RecipeGen.[1] It is worth noting that CRAFT uses an LLM in a limited way (to restructure and rewrite existing corpora) that seems to exclude the possibility that distillation has played a role. However, even if distillation were to be considered the reason for good CRAFT performance, the results indicate otherwise: we use the same model as Instruct, Mistral 7B Instruct v0.2, to paraphrase existing corpora examples but achieve even stronger results.

Finally, we observe that CRAFT models out-

[1]We further investigate the RecipeGen results in §5.5.

Dataset	In-Domain (ID)	CRAFT_{XL}
ScienceQA (In-Domain)	**92.7**	80.4
$\text{MMLU}_{\text{Medical Genetics}}$	**67.0**	66.0
$\text{MMLU}_{\text{Anatomy}}$	58.5	**60.0**
$\text{MMLU}_{\text{High School Biology}}$	68.1	**69.0**
$\text{MMLU}_{\text{College Biology}}$	69.4	**70.8**
MMLU-Avg	65.8	**66.5**

Table 1: Out-of-domain performance of in-domain (ID) vs CRAFT_{XL} models on biology QA tasks. While ID outperforms on its in-domain ScienceQA test set, CRAFT_{XL} shows better generalization to three of four OOD biology subsets from MMLU.

perform the In-Domain (ID) baseline of 25,000 samples in summarization. For other tasks, while we observe lower performance than ID with sample numbers between 1,192 for BioQA, 9,264 for CSQA, and 25,000 for MedQA, we speculate that this could be due to in-domain evaluation for 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.

5.2 Data Contamination and OOD Generalization

For the experiments in Figure 3, we investigate potential data contamination between test and training examples. We conduct 5-gram weighted Jaccard similarity analyses between CRAFT or in-domain (ID) datasets and the test dataset. 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/ID and test datasets, where n-grams receive weight proportional to their frequency.

This analysis reveals that all CRAFT datasets have less than 0.4% similarity with the test sets, whereas the in-domain datasets show much higher similarities: BioQA (16.6%, 1,192 samples), CSQA (4.4%, 9,264 samples), MedQA (1.1%, 25,000 samples), and Summarization (0.3%, 25,000 samples), indicating some overlap between train and test splits. However, the substantial overlap in in-domain datasets suggests that their reported performance might benefit from train-test similarity.

To isolate generalization capabilities, we evaluate the in-domain (ID) baseline and CRAFT models on four out-of-domain (OOD) biology QA tasks from MMLU (Hendrycks et al., 2021). Table 1 shows that while the ID model outperforms CRAFT_{XL} by 12.3 percentage points on its in-domain test set, CRAFT_{XL} performs better on three of the four OOD biology tasks. This indicates that CRAFT's training datasets offer bet-

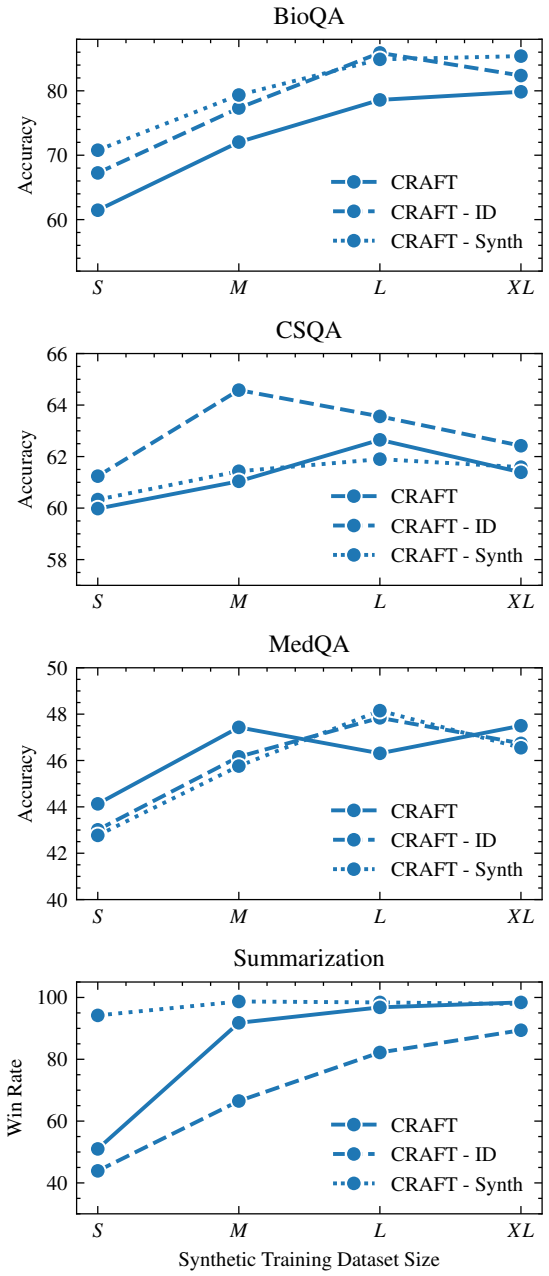

Figure 4: Performance comparison when CRAFT's retrieval process is initiated with standard human-curated (CRAFT), in-domain (-ID) and purely synthetic (-Synth) few-shots. As the dataset size increases, performance converges across the different few-shot sources, indicating that the retrieval and augmentation framework of CRAFT effectively abstracts away the variability in the quality of the initial few-shots.

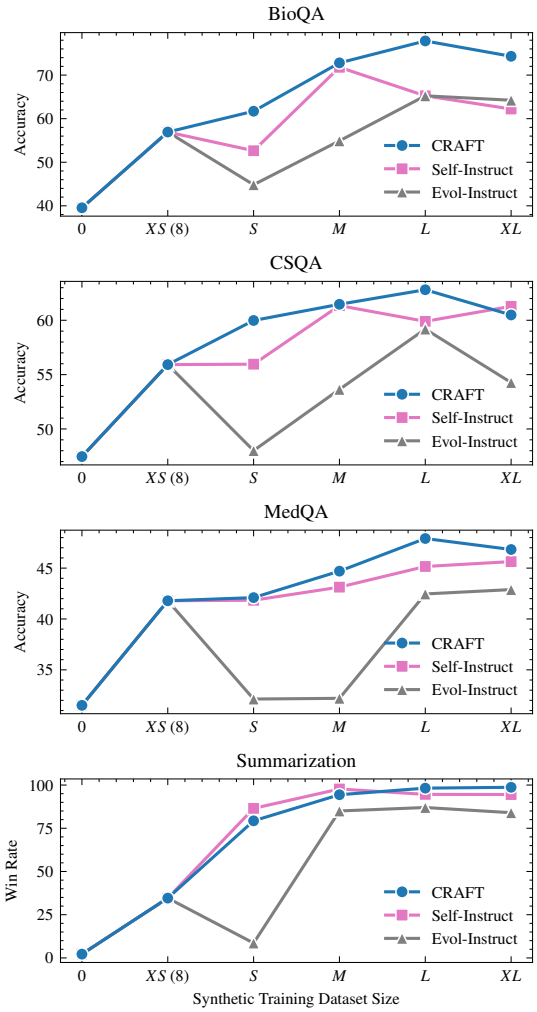

Figure 5: Performance of CRAFT versus Evol-Instruct and Self-Instruct across tasks and dataset sizes with 8 few-shots. CRAFT shows better scaling and higher accuracy than both baselines in most settings.

ter generalization capability and robustness across different domains than human-annotated datasets.

5.3 Sensitivity Analyses

We investigate the sensitivity of the CRAFT pipeline based on different setups: the source of few-shots, the number of few-shots, and the base model used for fine-tuning.

5.3.1 Few-Shot Source

To assess the sensitivity of CRAFT to the source of the initial few-shots, we evaluate two additional variants alongside our standard human-curated few-shots. In the in-domain (**ID**) setting, we directly select few-shots from the training splits of our evaluation datasets and manually pair them with appropriate web texts following the design illustrated in Figure 2. This setup represents an optimal distribution match to the test set. The synthetic (**Synth**) setting employs zero-shot prompts using Mistral 7B Instruct v0.2 to generate both the few-shots and the corresponding web texts without further curation. Although this approach minimizes human effort, it carries the risk of producing repetitive or lower-quality examples.

Num FS	BioQA	CSQA	MedQA	Summ
8	74,31	60,49	46,83	98,70
32	79,85	61,39	47,50	98,40

Table 2: Accuracy and win rate of CRAFT_{XL} when initiated with 8 and 32 human-curated few-shots. Using just 8 few-shots to initialize CRAFT achieves comparable performance in three of four tasks.

We run the full CRAFT pipeline using these few-shot variants, fine-tuning Mistral 7B v0.2 and evaluating performance on our standard benchmarks. As shown in Figure 4, most tasks yield very similar performance regardless of the few-shot source. For CSQA and MedQA, performance varies by at most 1.02 and 0.96 percentage points (pp.), respectively, confirming the stability of our approach. Summarization exhibits higher variance across few-shot sources, with a 9 pp. difference favoring manually curated CRAFT few-shots. However, when scaled to CRAFT_{XL}, different few-shot sources lead to more similar results, despite the initially higher variability. Overall, these findings highlight the robustness of CRAFT, demonstrating that even with synthetic few-shots, it consistently generates high-quality, task-specific datasets.

5.3.2 Number of Few-Shots

To better understand the impact of initial human effort required on our synthetic data generation process, we investigate the sensitivity of CRAFT to the number of few-shot examples provided.

Table 2 compares the performance of CRAFT_{XL}, which generates 25,000 synthetic samples, when our retrieval started with 8 versus 32 human-curated few-shots. The results indicate that, while increasing the number of few-shots can yield notable improvements (e.g., BioQA increases from 74.31 to 79.85), the overall performance across three of four tested tasks remains largely comparable. In commonsense QA, medicine QA, and summarization, differences are minimal, with performances varying below 1 percentage point or even matching performance

Task	CRAFT_{XS}	CRAFT_{S}	CRAFT_{M}	CRAFT_{L}	CRAFT_{XL}
BioQA	61.5	64.7	68.8	73.0	78.3
CSQA	55.8	54.8	58.6	60.4	61.4
MedQA	49.5	49.6	52.2	51.5	53.2
Summ	37.3	32.7	86.6	96.8	96.9

Table 3: Accuracy and win rate of CRAFT when Llama 3 8B is used as base model. Datasets generated using Mistral 7B v0.2 can further improve a stronger baseline model such as Llama 3 8B.

between the two settings. These findings suggest that initializing CRAFT with as few as 8 few-shots is a valid and cost-effective option, significantly reducing the human effort required while still producing high-quality synthetic datasets. Detailed sensitivity results for all dataset sizes (from XS to XL) are provided in Appendix B.

5.3.3 Base Model

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 (Dubey et al., 2024) 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 3. Notably, the datasets created with Mistral 7B Instruct v0.2 are also effective in improving a model with a much stronger baseline performance, such as Llama 3 8B. This finding underscores that our framework leverages the sample-generating LLM only in a reduced capacity – only to augment the retrieved documents – without relying heavily on its inherent performance.

5.4 Synthetic Data Generation Comparisons

Figure 5 shows the results when CRAFT is compared against other synthetic data generation methods. CRAFT consistently outperforms Evol-Instruct (EI) across all tasks and dataset sizes. While Self-Instruct (SI) sometimes matches CRAFT's performance at smaller scales (S/M), CRAFT always achieves higher performance than SI at larger scales (L/XL), except for CSQA at the XL size. Moreover, CRAFT scales more consistently while performing better than the other methods. We believe that a fully distillation-based technique like SI could be competitive with CRAFT in some cases, especially if the number of few-shot examples were increased; however, the available corpus samples in CRAFT improve quality and diversity even with a small number of few-shots.

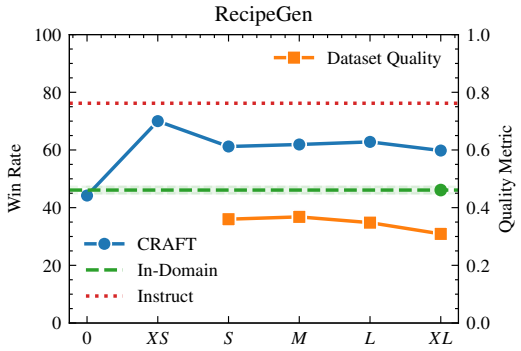

Figure 6: Non-scaling behavior in Recipe Generation: Dataset quality declines with increasing size (100 to 25K examples), showing an inverse scaling trend. This anomaly reflects diminishing data quality at scale.

Importantly, we observed significant differences in method sensitivity across few-shot sources. When running the methods with three different few-shot sources (human-curated, in-domain, and purely synthetic examples), CRAFT demonstrated notably lower variability. At the XL model size, CRAFT achieved an average standard deviation of 1.57 across all five tasks, compared to 3.53 for EI and 4.31 for SI, while also achieving better performance.

5.5 Negative Results: Recipe Generation

Out of our five tasks, we observe non-scaling behavior in one: 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 6. 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.

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 Llama 3 8B Instruct to answer the instructions. Then, using Llama 3 70B Instruct as a judge, we compared win rates, i.e., which output the model preferred: the gold output in the CRAFT datasets 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.

Generation Method	Generated Sample
Human (ID)	**Sample 1:** Select the temperate deciduous forest ecosystem. A. This ecosystem has: long, cold winters and short, cool summers; many evergreen trees; soil that is poor in nutrients B. This ecosystem has: year-round rain and warm temperatures; soil that is poor in nutrients; many different types of organisms C. This ecosystem has: warm, wet summers and cold, wet winters; soil that is rich in nutrients; only a few types of trees **Answer:** C **Sample 2:** Which organ controls the function of other body organs? A. skeleton B. lungs C. heart D. brain **Answer:** D
CRAFT	**Sample 1:** Which organ plays the primary role in excreting acid and regulating electrolyte balance in the body? A. Lungs B. Skin C. Kidneys D. Liver **Answer:** C **Sample 2:** Which simple, single-celled organism formed the first stromatolites on Earth approximately 2 billion years ago? A. Cyanobacteria B. Algae C. Fungi D. Protozoa **Answer:** A
Self-Instruct (SI)	**Sample 1:** Which of the following is a characteristic feature of prokaryotic cells? A. Nucleus and mitochondria are present in the cell. B. The cell membrane is not present. C. The DNA is located in the nucleolus. D. The cell wall is absent. **Answer:** D **Sample 2:** Which of the following is a function of the endoplasmic reticulum (ER)? A. It acts as a site for protein synthesis and folding. B. It functions in the storage and transport of lipids. C. It plays a role in the breakdown of complex carbohydrates. D. It is involved in the process of cell division. **Answer:** A
Evol-Instruct (EI)	**Sample 1:** In the early stages of cellular evolution, how did organisms with rudimentary structures develop a complex and dynamic [. . .] In the pursuit of resource optimization, how did these ancient organisms establish an intricate signaling network, enabling [. . .] **Sample 2:** As you delve deeper into the intricate diaphyseal region of a human right femur, you'll encounter various fascinating [. . .] Further studies on the presence, distribution, and function of bone marrow [. . .]

Table 4: Qualitative comparison of BioQA samples. For each source, we show two examples to illustrate strengths or shortcomings. CRAFT's outputs are well-formed and comparable to the human baseline. Self-Instruct maintains the format, but Sample 1 contains a factual error, and Sample 2 includes multiple correct options, making Option B ambiguous in a single-answer context. Evol-Instruct deviates from the task, failing to produce usable QA samples.

We observe a decline in data quality when scaling to 25,000 examples: The 100- and 500-example sets achieve win rates of around 0.4, while the 25K set drops to 0.3. This degradation likely causes the performance decline during scaling. We also observe a similar trend in other synthetic data generation methods like Self-Instruct and Evol-Instruct. While the final CRAFT dataset remains practical (outperforming the baseline using official human data), future work should incorporate stopping criteria or more quality validation.

6 Qualitative Analysis

To better understand the quantitative results presented in Section 5, we perform qualitative case studies by inspecting synthesized samples from multiple tasks and data generation methods. This analysis provides intuition as to *why* certain methods perform better than others and illustrates the style and patterns in generated samples.

6.1 Case Study Design

Our analysis includes both QA and generative tasks across samples generated by CRAFT, Self-Instruct (SI), Evol-Instruct (EI), and human annotators. We use BioQA as the representative for QA tasks, but analyze samples from summarization and recipe generation separately due to format differences. After manually inspecting a wide range of outputs, we selected representative examples that either adhere to or violate the following three criteria: (i) alignment with the expected task format, (ii) content quality and factuality, as well as (iii) stylistic appropriateness. This setup illustrates common advantages and shortcomings.

6.2 Question-Answering Observations

Adherence to format: As Table 4 shows, both CRAFT and SI produce well-formed QA pairs in a concise format, matching the human in-domain data. EI often drifts into free-form essays, bundles multiple questions into one instruction, and often reaches maximum generation length before completing a sample. The method's goal to evolve samples by increasing complexity leads to a breakdown in task adherence, which observably resulted in decreased performance in Section 5.

Domain fidelity and correctness: CRAFT leverages retrieved corpus passages to embed precise terminology ("*stromatolites*", "*electrolyte balance*") and produces correct QA pairs. Distractor options are semantically proximate yet unambiguously wrong, mirroring expert-curated style. SI's Sample 1 is factually incorrect, stating that the cell wall is absent in prokaryotes. This exemplifies the risk of ungrounded generation: the model produces plausible-sounding samples at the cost of factual integrity. Furthermore, in SI's Sample 2, the distractor option B is also correct, creating annotation noise. EI's questions are verbose and often include multiple sub-questions, making it unclear which question should or will be answered.

6.3 Generative Tasks Observations

Recipe Generation: The human-annotated in-domain recipes are terse, often just listing steps without providing descriptive details.

> *Add all dry ingredients together and mix well. Add remaining ingredients stirring only until moistened. [. . .]* (Human (ID), Banana Nut Muffin)

In contrast, CRAFT can produce specific, multi-step recipes with headings and detailed directions.

> ***Cook the apples:*** *In a saucepan, cook the sliced apples with sugar over medium heat until softened and caramelized. [. . .]*
> ***Add apple filling:*** *Place the apple filling in a line down the middle of the pastry, leaving a 2-inch border on all sides.*
> ***Brush the edges:*** *Brush the edges of the pastry with the beaten egg and water mixture.*
> *[. . .]* (CRAFT, Apple Strudel)

CRAFT's ability to generate more descriptive content stems from grounding the sample synthesis in more elaborate retrieval documents such as cooking blogs. However, as CRAFT retrieves less relevant documents at larger scales, its outputs can degrade. Instructions can become vague (e.g., *"Prepare a dish using items from your well-stocked pantry, freezer, and fridge."*) or convoluted, possibly from combining multiple separate recipes from a retrieved document into one (e.g., *"Create a salad or bowl by combining greens, proteins, vegetables, cheeses, fruits, and nuts with the specified dressing."*, resulting in an ingredient list that starts with greens but ends with *"assorted fruits"* and a *"scoop of ice cream"*. Complete recipe samples are shown in Appendix C.1.

Summarization: The human-annotated summaries are written in a headline-driven style, presenting information as extractive bullet points (e.g., *"NEW: California Public Utilities Commission passed ban Thursday."*. While factually grounded, this format is limited in stylistic variety and can lack context as well as narrative coherence. CRAFT learns to generate more abstractive, prose-style summaries that are tailored to specific instructional details by being exposed to varying retrieval contexts. The full sample is provided in Appendix C.2.

> ***Instruction:*** *Summarize the text below, focusing on Bill Ackman's investment in Procter & Gamble and his intentions for the company.* ***Text:*** *[153 word text ...].* ***Summary:*** *Bill Ackman, a hedge fund manager, believes that Procter & Gamble's (P&G) stock could be worth more than its current price due to poor marketing and pricing strategies. He holds around 1% of the company's shares and is actively pushing for changes. [48 more words . . .]*

SI also demonstrates task adherence and is likewise able to adjust its style toward varying instructions (e.g., summarizing a text *"in a tweet"*), but its outputs tend to be generic and lack the specificity seen in CRAFT's document-grounded summaries. EI again introduces superficially complex phrasing, such as *"Unraveling the Perplexing Enigma of Age-Related Pituitary [. . .]"*, or *"In the intricately woven narrative of Afghanistan's rich and complex history [. . .]"*.

7 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 and maintained robustness to variations in the quality of the initial few-shot examples. Furthermore, while some fully synthetic methods such as Self-Instruct produce competitive results, CRAFT outperforms these approaches overall, offering a more scalable and reliable dataset generation framework.

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.

Acknowledgements

We thank the action editor, Tao Ge, and the anonymous reviewers at TACL for their helpful comments during the review process. 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 acknowledges the EuroHPC Joint Undertaking for awarding access to MareNostrum5, hosted at Barcelona Supercomputing Center (BSC), Spain, under proposal No. EHPC-DEV-2024D12-031. IZ was also supported by a G-Research Travel Grant to present this work at the ELLIS Doctoral Symposium 2024 in Paris.

References

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. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint, arXiv:1603.04467*.

Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. 2007. 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.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328.

Bruce Alberts. 2017. *Molecular Biology of the Cell*, 5th edition. Garland Science.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

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

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. 2020. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28, Dublin, Ireland. Association for Computational Linguistics.

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

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

Ernie Chang, Xiaoyu Shen, Alex Marin, and Vera Demberg. 2021. The selectgen challenge: Finding the best training samples for few-shot neural text generation. In *Proceedings of the 14th*

International Conference on Natural Language Generation, pages 325–330.

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

Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. 2024b. 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.

Zhikui Chen, Tiandong Ji, Suhua Zhang, and Fangming Zhong. 2022. Noise suppression for improved few-shot learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1900–1904.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Forty-first International Conference on Machine Learning*.

Zhuyun Dai, Arun Tejasvi Chaganty, Vincent Y Zhao, Aida Amini, Qazi Mamunur Rashid, Mike Green, and Kelvin Guu. 2022. Dialog inpainting: Turning documents into dialogs. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4558–4586. PMLR.

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

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

Abhishek Divekar and Greg Durrett. 2024. SynthesizRR: Generating diverse datasets with retrieval augmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19200–19227, Miami, Florida, USA. Association for Computational Linguistics.

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. 2024. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

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. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

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

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How small can language models be and still speak coherent English? *arXiv preprint arXiv:2305.07759*.

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

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

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

Yvette Graham. 2015. 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.

Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. 2023. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*.

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. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Veronika Hackl, Alexandra Elena Müller, Michael Granitzer, and Maximilian Sailer. 2023. Is GPT-4 a reliable rater? evaluating consistency in GPT-4's text ratings. *Frontiers in Education*, 8.

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. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

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

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

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

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. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

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

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. LongForm: Optimizing instruction tuning for long text generation with corpus extraction. *arXiv preprint arXiv:2304.08460*.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, volume 1, pages 646–651.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*.

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

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B.

Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Roderick JA Little et al. 1993. Statistical analysis of masked data. *Journal of Official Statistics - Stockholm*, 9:407–407.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. 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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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

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

Umar Maqsud. 2015. 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.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

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. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

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

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*.

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

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arxiv:2303.08774*.

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. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

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. 2019. PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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

Nils Reimers and Iryna Gurevych. 2019. 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.

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

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

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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. Association for Computational Linguistics.

Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. 2024. Retrieval-augmented data augmentation for low-resource domain tasks. *arXiv preprint arXiv:2402.13482*.

Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. 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.

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

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

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. 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)*.

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

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. 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.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. 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.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. 2023. 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.

A Implementation Details

A.1 Few-Shot Design

BioQA few-shot texts were drawn from diverse sources, including textbooks (Alberts, 2017; Malmquist and Prescott, 2022; Wilkin and Brainard, 2016; Rye et al., 2016), Encyclopedia Britannica, and openly accessible materials. **MedQA** examples were based on public content from health-related websites (e.g., NIH, NHS, FDA, Mayo Clinic, Cleveland Clinic). **CSQA** few-shots were compiled from blogs, articles, and other topical online sources. **Recipe generation** few-shots were sourced from blogs and public recipe sites, typically structured as an instruction or question with lists of ingredients and steps. If recipes were in continuous prose, the authors added structured elements manually. To ensure diverse retrieval, a wide range of dishes and vocabulary was included. **Summarization** few-shots were built from texts across websites, blogs, magazines, and GitHub issues. Each example included a full input text, a summarization instruction, and a corresponding summary. Where summaries were not available, they were authored to match realistic format and content.

A.2 Corpora

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

C4 (Raffel et al., 2020) provides a broad web-crawled dataset. We use a 305GB filtered subset of the original 750GB corpus, excluding not-safe-for-work content. **Wikipedia** offers high-quality encyclopedic content. We use the English Wikipedia dump from January 2, 2024, processed with WikiExtractor (Attardi, 2015) to extract clean text. **Stack Exchange** (Flax Sentence Embeddings Team, 2021) includes QA-format documents across 173 communities, combining title, body, and top-voted answer. It spans technical and non-technical domains. **WikiHow** (Koupaee and Wang, 2018) features instructional content in a step-by-step format, useful for generative tasks such as recipe generation or summarization.

After filtering documents to lengths between 200 and 25,000 characters, we retain 362M documents from C4, 10.5M from Wikipedia, 9.5M from Stack Exchange, and 190K from WikiHow. Combined, these 383M documents occupy 247GB when GZIP-compressed (Deutsch, 1996) and stored as 16-bit NumPy arrays (Harris et al., 2020).

A.3 Document Retrieval

We use a two-step retrieval strategy to approximate global similarity search over 383M embedded documents, avoiding expensive full pairwise comparisons.

First, the embedding database is split into sequential shards of 350K documents. For each shard, cosine similarity to the few-shot samples is computed, and the top 5% most similar documents are retained, reducing the candidate pool to 19M documents. Secondly, we perform a second round of cosine similarity and standard top-k retrieval on this reduced set to obtain the final document matches.

To avoid redundancy or topical overfitting in retrieval, we combine two strategies: (i) 50% of samples are retrieved via top-k similarity to each few-shot individually, and (ii) 50% via similarity to the averaged embedding of all few-shots to capture latent task structure.

A.4 Task Sample Synthesis

Synthetic task samples are generated via in-context learning (Brown et al., 2020), using three randomly sampled few-shot examples per prompt. These are interleaved with the instruction and a retrieved corpus document, following the template shown in Table 2, step 3. Because each few-shot includes a long text, input prompts often exceed 10,000 tokens (up to 20,000), requiring long-context models.

We use Mistral 7B Instruct v0.2 (Jiang et al., 2023) with vLLM (Kwon et al., 2023) for generation. Sampling is performed with temperature=0.7, top-k=40 (Fan et al., 2018), and top-p=0.9 (Holtzman et al., 2020). Maximum output lengths are capped at 256 tokens for QA, 1280 for recipes, and 1536 for summarization, based on empirical tuning.

To enable quality control, all outputs are formatted as JSON with fixed keys. We discard any samples with malformed structure, missing fields, or insufficient length. For QA tasks, we validate that answer

options are complete and contain a valid label. To reduce redundancy, we filter out samples with high similarity to few-shots or other outputs using fuzzy string matching (Ranjith et al., 2022) with a token set ratio > 0.85. We recommend retrieving roughly twice the desired number of corpus samples to account for filtering. Task-specific filtering statistics are detailed in Appendix D.

B Additional Results: Varying the Number of Few-Shots

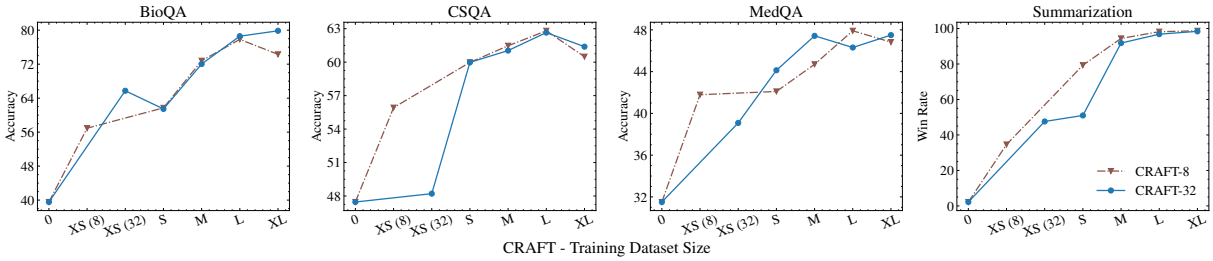

Figure 7: Performance between CRAFT when initiated with 8 and 32 human-curated few-shot examples.

C Full-Length CRAFT-Generated Examples

C.1 Recipe Generation

Instruction: Prepare apple strudel using puff pastry and apples with sugar and raisins.
Ingredients: 5 medium apples, peeled, cored, and sliced; 1/2 cup granulated sugar; 1/2 cup raisins; 1 pound puff pastry, thawed; 1 egg, beaten; 1 tbsp water; powdered sugar, for dusting.
Steps: Cook the apples: In a saucepan, cook the sliced apples with sugar over medium heat until softened and caramelized. **Combine apples and raisins:** Transfer the cooked apples to a bowl and mix in the raisins. **Prepare the pastry:** Unroll the puff pastry on a lightly floured surface and place it on a baking sheet. **Add apple filling:** Place the apple filling in a line down the middle of the pastry, leaving a 2-inch border on all sides. **Brush the edges:** Brush the edges of the pastry with the beaten egg and water mixture. **Fold the pastry:** Fold the edges of the pastry over the apple filling, sealing it tightly. **Bake the strudel:** Brush the top with the egg mixture and bake at 375°F for 30–35 minutes, until golden brown. **Cool and serve:** Let cool slightly before dusting with powdered sugar and serving.

Instruction: Prepare a dish using items from your well-stocked pantry, freezer, and fridge.
Ingredients: Honey, Chipotle peppers in adobo sauce, Assorted olives, jalapenos, pepperoncinis etc., Flour (all-purpose, whole wheat, self-rising), Sugar, brown sugar, Powdered sugar, Baking powder, baking soda, extracts, yeast, Masa, Potatoes, onions, garlic, Dried beans, Stocks and broths (chicken, beef, vegetable), Jarred pesto and specialty relishes, chutneys, etc., Panko breadcrumbs, Kosher salt, black pepper, Worcestershire, Tabasco, olive oil, different vinegars, soy sauce, Longer lasting cheeses (cheddar, Parmesan, feta), Bacon, Corn tortillas, Beef, sausage (breakfast sausage, Italian sausage, chorizo), Dinner rolls, Pie crust, Frozen fruits (peaches, berries, cherries, etc.), Frozen vegetables (green beans, peas, lima beans, carrots, corn), Ready-made recipes, Chili, Homemade pumpkin puree
Steps: Choose a recipe or decide on a dish to make using the available ingredients. **Prepare and gather ingredients** from the pantry, freezer, and fridge. **Cook the dish** according to the recipe or your preferred method. **Serve and enjoy!**

Instruction: Create a salad or bowl by combining greens, proteins, vegetables, cheeses, fruits, and nuts with the specified dressing.
Ingredients: Chopped cucumber, Tomato, Bell pepper, Red onion, Feta cheese, Kalamata olives, Chopped green kale, Shredded chicken, Black beans, Crumbled cojita, Diced avocado, Spicy crispy pepitas, Lemon-cumin vinaigrette, Chopped Napa cabbage, Rice noodles, Julienned red bell peppers, Scallions, Shredded carrots, Cilantro, Peanuts, Zingy miso ginger vinaigrette, Crispy noodles, Sweet

corn, Diced bell pepper, Tomato, Avocado, Red onion, Cilantro, Toasted pine nuts, Blue cheese vinaigrette, Dried cherries, Mild feta cheese, Swiss chard, Peppery baby arugula, Toasted pecans, Champagne vinaigrette, Fresh baby lettuce, Sweet green peas, Crunchy bacon shards, Savory white mushroom sauté, Asiago cheese, Tart vinaigrette, Grilled Bosc pears, Bacon crumbles, Toasted pecan halves, Tangy Maytag blue cheese, Tender greens, Ruby port vinaigrette, Frisée greens, Poached egg, Bacon lardons, Warm shallot vinaigrette, Baby spinach, Sautéed mushrooms, Bacon, Hardboiled egg pieces, Slivered red onion, Honey balsamic vinaigrette, Red cabbage, Romaine, Tomato, Corn, Green and red onion, Seasoned beef and chorizo, Cheddar and jack cheese, Cumin honey-lime vinaigrette, Duck confit, Hoisin and hot sauce, Peanuts, Cabbage, Scallions, Carrots, Crisp jicama shells, Assorted fruits, Scoop of ice cream
Steps: Chop or prepare all vegetables, fruits, and proteins as specified in the text. **Combine the greens** in a large bowl or on individual plates. **Add proteins, vegetables, fruits, cheeses, and nuts** to the greens. **Drizzle with the specified dressing** or vinaigrette. **Toss gently** to evenly coat the salad or bowl. **Serve and enjoy!**

C.2 Summarization

Instruction: Summarize the text below, focusing on Bill Ackman's investment in Procter & Gamble and his intentions for the company.
Input Text: Bill Ackman, a well-known hedge fund manager, believes that Procter & Gamble (P&G) could be worth more than its current share price due to poor marketing and pricing strategies. Ackman holds around 1% of the company's shares and has been actively advocating for changes. During a presentation, he mentioned that he spends 25% of his time on his investment in P&G. Ackman has been critical of the present management and its financial performance since disclosing his stake last year. It remains to be seen whether the company will welcome Ackman's involvement more enthusiastically than in the past. In the past, P&G has been targeted by activists after an event-driven turn in the share price. Ackman has not specified who he would like to see as the new CEO but has mentioned that potential changes could lead to savings, marketing efficiencies, and operating leverage. He did not mention any particular candidates during his presentation.
Generated Summary: Bill Ackman, a hedge fund manager, believes that Procter & Gamble's (P&G) stock could be worth more than its current price due to poor marketing and pricing strategies. He holds around 1% of the company's shares and is actively pushing for changes. Ackman has been vocal about his dissatisfaction with the current management and has spent significant time advocating for improvements. He has not specified who he would like to see as the new CEO but has mentioned that potential changes could lead to savings, marketing efficiencies, and operating leverage.

D Filtering Statistics

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 5: 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	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 6: 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	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 7: 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	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 8: 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	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

Table 9: Summarization corpus and task sample filtering process. Corpus samples are turned into task samples after filtering for excessive length.

E Dataset Statistics

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

Table 10: Dataset 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.