Learning to Generate Instruction Tuning Datasets for Zero-Shot Task Adaptation

Nihal V. Nayak Yiyang Nan Avi Trost Stephen H. Bach
Department of Computer Science, Brown University
{nnayak2, ynan3, atrost, sbach}@cs.brown.edu

Abstract

We introduce Bonito, an open-source model for *conditional task generation* that converts unannotated text into task-specific training datasets for instruction tuning. We aim to enable zero-shot task adaptation of large language models on users' specialized, private data. We train Bonito by fine-tuning a pretrained large language model on a new large-scale dataset with 1.65M examples created by remixing existing instruction tuning datasets into *meta-templates*. The meta-templates for a dataset produce training examples where the input is the unannotated text and the task attribute and the output consists of the instruction and the response. We use Bonito to generate synthetic tasks for seven datasets from specialized domains with unannotated text across three task types—yes-no question answering, extractive question answering, and natural language inference—and adapt language models. We show that Bonito significantly improves the average performance of pretrained and instruction tuned models over the de facto self supervised baseline. For example, adapting Mistral-Instruct-v2 and instruction tuned variants of Mistral and Llama2 with Bonito improves the strong zero-shot performance by 22.1 F1 points whereas the next word prediction objective undoes some of the benefits of instruction tuning and reduces the average performance by 0.8 F1 points. We conduct additional experiments with Bonito to understand the effects of the domain, the size of the training set, and the choice of alternative synthetic task generators. Overall, we show that learning with synthetic instruction tuning datasets is an effective way to adapt language models to new domains. The model, dataset, and code are available at https://github.com/BatsResearch/bonito.

1 Introduction

Large language models show remarkable zero-shot capabilities by simply learning to predict the next token at scale (Brown et al., 2020; Touvron et al., 2023). By fine-tuning these models on instruction tuning datasets containing many *tasks*—each comprising an input *instruction* and a desired *response*—the model generally improves in its ability to respond to unseen instructions. However, this generalization is still limited by the qualities of the instruction tuning dataset. Existing datasets like Public Pool of Prompts (P3) (Bach et al., 2022), Natural Instructions (Mishra et al., 2022; Wang et al., 2022), and Dolly-v2 (Conover et al., 2023) focus on text from the Web and classic natural language tasks so that they can serve a wide range of use cases, i.e., they are a one-size-fits-all approach. On the other hand, tasks in areas like biomedical and legal domains require specialized, often implicit, domain knowledge. We study how to adapt language models to follow instructions in specialized domains without annotated data.

The ability to follow task-specific instructions in specialized domains is important for bringing the benefits of large language models to a wider range of users. Recent evaluations—including evaluations of proprietary models—show that they often significantly underperform specialized models (Kocoń et al., 2023; Shen et al., 2023; Ziems et al., 2023), particularly in domains requiring subject matter expertise. This motivates us to investigate effective ways to provide domain knowledge to large language models.

Self supervision in the form of next word prediction on the target corpus is a simple way to teach language models about new domains (Gururangan et al., 2020). However, this approach requires an enormous amount of training to achieve strong performance (Chen et al., 2023). Further, we find that self supervision can undo the benefits of instruction tuning (see Section 5.3). Alternatively, continued training of models with instructions from specialized domains significantly improves performance (Scialom et al., 2022; Shi and Lipani, 2023;

Findings of the Association for Computational Linguistics: ACL 2024, pages 12585–12611

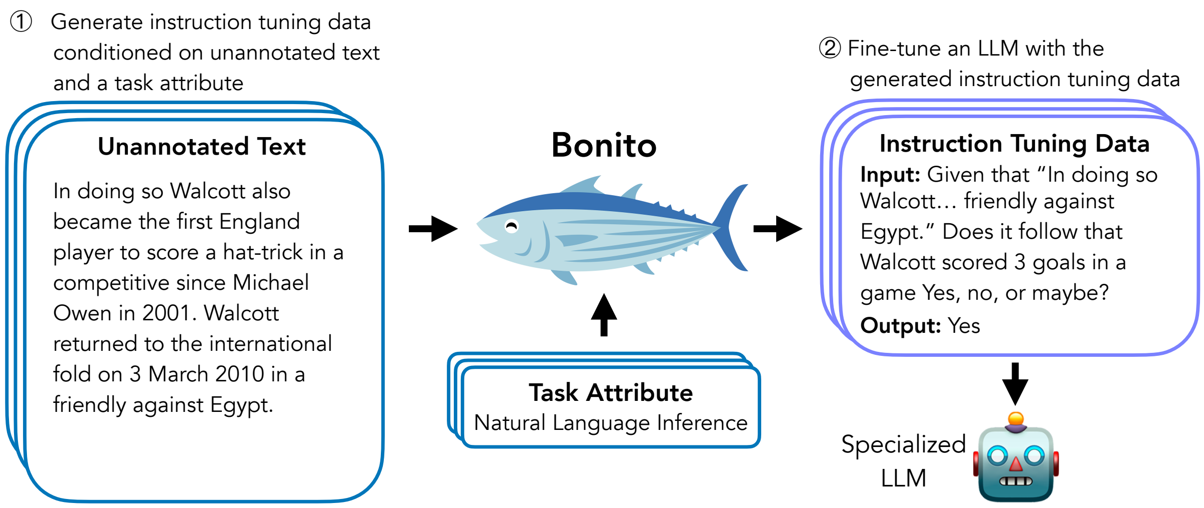

Figure 1: Bonito workflow for conditional task generation and adaptation. Bonito takes unannotated text as input, along with task attributes, to generate instruction tuning data. For each unannotated text, it generates an instruction that references the text and a target response. The instruction tuning data is then used to (further) fine-tune a language model, adapting it to the task in the specialized domain.

Yunxiang et al., 2023; Deng et al., 2023; Singhal et al., 2023a; Wu et al., 2024). However, they need to repeat the time-consuming and labor-intensive process of annotating a domain-specific dataset. Furthermore, collecting instructions in specialized domains is very expensive because they are annotated by domain experts such as scientists and researchers (Thulke et al., 2024). In this work, we automate the creation of instruction tuning datasets in specialized domains.

We create Bonito, an open-source model to convert unannotated text from specialized domains into task-specific training datasets for instruction tuning (Figure 1). We call this problem *conditional task generation*. Our key idea is to make a new large-scale dataset called Conditional Task Generation with Attributes (CTGA), to train Bonito, by reorganizing existing instruction tuning datasets (see Figure 2). Instruction tuning datasets like P3 (Bach et al., 2022) exist as templates that convert semi-structured examples of natural language tasks into a fully prompted format, in which both the input and the desired response are text strings. We focus on a subset of the templates in P3 that require a *context* or a passage to complete the task. For example, a context could be a paragraph that contains a fact or that contains the answer to a question. Then, we remix these templates to create the meta-templates. Each meta-template for a dataset produces training examples in which the input is context and a task attribute such as yes-no question answering, and the output is the entire task: the instruction (including the context) and the desired response. In this way, we can easily create abundant, diverse examples to train Bonito. After training Bonito, we can use new unannotated text from the target domain as the context to generate task-specific synthetic datasets and train specialized language models.

Bonito significantly improves over self supervision on zero-shot task adaptation of pretrained and instruction tuned models. We use Bonito to generate instruction tuning data for seven datasets across three task types—yes-no question answering (PubMedQA and Privacy Policy QA), extractive question answering (SQuADShifts-NYT, Amazon, and Reddit), and natural language inference (ContractNLI and Vitamin C)—and adapt language models. Our results show that Bonito improved Mistral-7B by 34.7 F1 points and Llama 2 7B by 31.6 F1 points over the self supervised baseline, next word prediction objective. We also consider a more practical setting where we further train Mistral-7B-Instruct-v0.2 and instruction tuned variants of Mistral-7B and Llama 2 7B trained on the T0 split of the P3 dataset. Our results show that Bonito outperforms the strong zero-shot baseline performance by an average of 22.1 F1 points across all the models. On the other hand, we find that self supervision undoes some of the benefits of instruction tuning, i.e., it leads to catastrophic forgetting, resulting in a drop in performance by an average of 0.8 F1 points across all models. Our analysis of Bonito shows that even task specialized models can be further improved by simply learning on Bonito generated tasks (see Section 6.1). We also find that training with more synthetic instructions on datasets like PubMedQA and Vitamin C improves model performance the most compared to other datasets (see Section 6.2). We perform additional experiments by prompting off-the-shelf open-source models like Zephyr-7B-β and Mistral-7B-Instruct-v0.2 and GPT-4 to generate tasks and find they can often improve the pretrained models but still struggle to increase model performance further when they are instruction tuned (see Section 7). Finally, our human evaluation of Bonito-generated tasks shows that Bonito and GPT-4o generate the same answer on 71 to 77 percent of the tasks. This indicates that Bonito generates high-quality tasks. However, there is room for improvement in generation quality to increase downstream model performance.

In summary, our main contributions are:

- We introduce Bonito, an open-source model for conditional task generation model to convert the user's unannotated text into task-specific instruction tuning datasets.
- Our experiments on zero-shot task adaptation on seven datasets across three task types show that Bonito improves over the self supervised baseline by an average of 33.1 F1 points on the pretrained models and 22.9 F1 points on the instruction tuned models.
- We analyze the effect of the domain, training size, and the choice of alternative task generators highlighting the benefits and limitations of Bonito.

2 Zero-Shot Task Adaptation

We describe the problem of zero-shot task adaptation. We have a language model, either pretrained via self supervision or further fine-tuned on a training mixture like P3 (Bach et al., 2022), along with a corpus of unannotated text from the target domain. We also know the target task type e.g., extractive question answering, natural language inference, etc. If the target task type has a fixed set of labels such as "yes" or "no" in yes-no question answering, we assume access to the label space. Our goal is to adapt the language model to follow task instructions in the target domain without human annotations, achieving zero-shot task adaptation.

3 Related Work

Instruction Tuning Multitask instruction tuning of language models dramatically improves their ability to follow instructions and generalize to new unseen tasks (Sanh et al., 2022; Wei et al., 2022; Mishra et al., 2022; Longpre et al., 2023; Chung et al., 2022; Zhou et al., 2023; Li et al., 2023). Typically, pretrained models are trained to follow instructions on large-scale training mixtures such as P3 (Bach et al., 2022) and the FLAN collection (Longpre et al., 2023). In this work, we use P3 to create meta-templates and train Bonito to generate NLP tasks in specialized domains.

Domain Adaptation Several works have adapted large language models to tasks in specialized domains (Gururangan et al., 2020; Yunxiang et al., 2023; Cui et al., 2023; Wu et al., 2023). Several works (Gu et al., 2021; Chen et al., 2023) show that self supervision or continuing the pretraining objective of the pretrained language model on the target domain corpus improves downstream performance. In this work, we find that self supervision improves the performance of pretrained models but hurts the performance of instruction tuned models (Section 5).

Recent work has adapted language models by training on large-scale in-domain datasets(Parmar et al., 2022; Gupta et al., 2022; Singhal et al., 2023b; Deng et al., 2023) or with a few examples from domain-specific tasks (Singhal et al., 2023a). However, annotating training datasets for new domains is labor-intensive and expensive (Thulke et al., 2024). We focus on generating training datasets in specialized domains and adapting language models without annotations.

Zero-shot task adaptation is closely related to unsupervised domain adaptation (Ganin and Lempitsky, 2015). In unsupervised domain adaptation, a trained model is used to generate pseudo-labels for the target unlabeled data and then trained on these labels. Naive pseudo-labeling cannot be applied to this work since we consider tasks like question answering and natural language inference tasks that require a question or a hypothesis before producing an answer in natural language. Further, popular techniques used in unsupervised domain adaptation such as choosing top-K confident classes (Huang et al., 2022; Menghini et al., 2023) cannot be easily adapted to NLP tasks as there may not be an explicit notion of classes.

There is a growing interest in using retrieval augmented generation (RAG) for domain-specific question answering (Lewis et al., 2020; Karpukhin et al., 2020; Siriwardhana et al., 2023; Zhang et al., 2024). In a RAG pipeline, given a question, the most relevant documents are retrieved before accurately producing an answer with a language model. Our work compliments the RAG pipeline as we assume access to the gold documents or paragraphs from specialized domains and improve the language model's ability to answer the questions.

Task Generation Task generation is a fast-growing area of research to adapt large language models to follow instructions (Wang et al., 2023; Taori et al., 2023; Honovich et al., 2023; Köksal et al., 2023; Liu et al., 2024). They typically condition GPT on a set of seed task demonstrations and generate new synthetic tasks (Wang et al., 2023; Honovich et al., 2023). However, task generation conditioned on the user's unannotated text has mostly been ignored. Additionally, generating with API-based models is expensive and cannot be used for proprietary or private research data (Köksal et al., 2023). On the other hand, Bonito is an open-source model that can be used to create tasks with the user's unannotated text without additional API costs.

Recently, Li et al. (2023) proposed to learn a backtranslation model, similar to Bonito, to iteratively grow and refine their instruction tuning dataset (Gulcehre et al., 2023). However, they focus on generating instructions conditioned on the unannotated text from a web corpus for long-form conversational data where the answer to the instruction is the unannotated text. In contrast, we focus on generating NLP tasks conditioned on a task type and unannotated text from a specialized domain. Further, we consider tasks such as question answering and natural language inference that require a question or a hypothesis before generating the appropriate answer.

Concurrent to this work, Yehudai et al. (2024) use in-context learning with Falcon-40B and Llama-65B to generate "grounded tasks" to adapt smaller models like FLAN-T5-XL (3B). These grounded tasks are similar to conditional tasks, except the instructions do not necessarily refer directly to the user's text. They might only be based on it, such as asking an open-ended question based on the original text. Our work goes further in several ways. First, we study how to create an open-source model for conditional task generation, as opposed to relying on prompting alone. Second, Bonito has only 7B parameters and we show that it creates data that can improve instruction tuned models of the same size and outperform even larger models like Flan-T5-XXL (11B) (see Appendix D). Third, we evaluate tasks with precise correct/incorrect answers, such as yes-no question answering and natural language inference, as opposed to tasks evaluated with similarity metrics.

Knowledge Distillation Knowledge distillation is a well-studied area (Hinton et al., 2015; Sanh et al., 2019; He et al., 2020). Typically, smaller models learn from the outputs of a larger model. Most recently, API-based models have been used to generate tasks and distilled into smaller models to mimic the abilities of the API-based models (Peng et al., 2023; Gudibande et al., 2023). In this work, we use Bonito to generate tasks based on the user's context and distill them into pretrained and instruction tuned models of the same size for zero-shot task adaptation (see Section 5).

Question Generation Several works have been proposed in question generation over the years (Mitkov and Ha, 2003; Pan et al., 2020; Lewis et al., 2021; Ushio et al., 2023). They often use heuristics such as templates (Mitkov and Ha, 2003), named entity recognition (Lewis et al., 2021), and semantic graphs (Pan et al., 2020). In our work, we train a language model without relying on task-specific heuristics. Ushio et al. (2023) is closely related to our work as they train a unified model to generate extractive questions and answers, but only focus on adapting small pretrained language models like T5-Large (770M). In contrast, Bonito can generate tasks beyond extractive question answering and enable zero-shot task adaptation on several task types with large models like Llama 2 7B and Mistral-7B.

4 Bonito: Learning to Generate Tasks

We describe the steps to create the Conditional Task Generation with Attributes (CTGA) dataset and train Bonito. Then, we briefly describe the procedure to create synthetic instruction tuning datasets with the target unannotated texts to adapt language models.

Key Properties We outline the key properties that we desire in our conditional task generation model: (1) given a corpus containing articles and paragraphs, the model should take the text as input

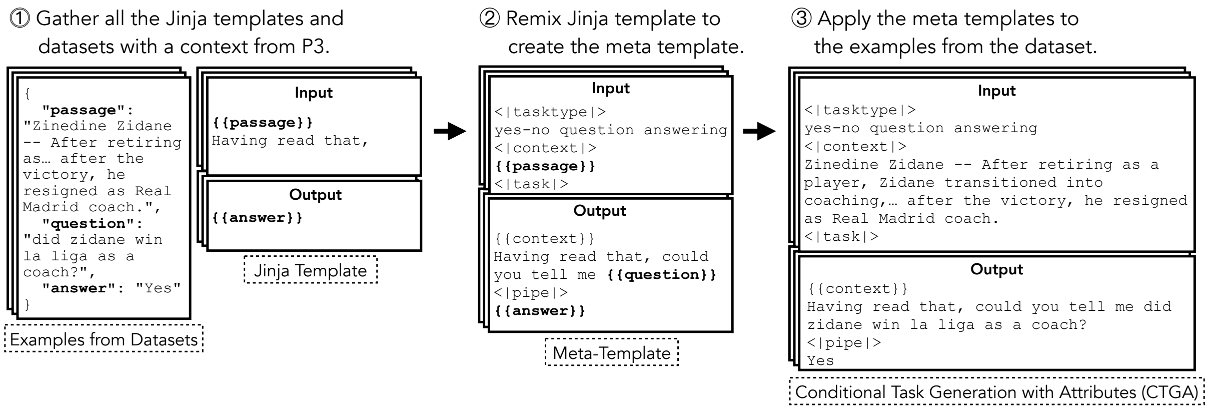

Figure 2: The high-level process of constructing the Conditional Task Generation with Attributes (CTGA) dataset.

and generate high-quality tasks that require minimal cleaning or post-processing, (2) the model should adhere to the task type like extractive question answering or natural language inference task, and (3) the model should generate diverse tasks for the exact text with varying styles.

4.1 Creating Bonito: Dataset and Training

To create a model that generates tasks conditioned on text, we create a new training dataset: Conditional Task Generation with Attributes (CTGA). The dataset contains 1.65 million examples derived from P3 (Bach et al., 2022) by annotating 323 prompt templates from 38 datasets with 16 task types (see Appendix H). Then, we train a pretrained large language model on this training dataset to create Bonito.

Constructing the Dataset Figure 2 shows the process of constructing the Conditional Task Generation with Attributes (CTGA) dataset. First, we identify datasets from P3 (Bach et al., 2022) that require a passage or a context to complete the task. For example, SQuAD (Rajpurkar et al., 2016) requires a context to answer extractive question answering tasks, whereas CommonSenseQA (Talmor et al., 2019) asks a multiple choice question without providing any relevant text. We identify a total of 38 datasets to be included in CTGA. For each dataset, we also collect the Jinja[1] templates from P3. Next, we remix the Jinja templates to create meta-templates. A meta-template is a Jinja template that includes the task attribute or the task type and the key for the context column in the input and the Jinja template for the instruction-response pair in the output with a placeholder `{{context}}` to avoid repeating the context. Since Jinja templates from P3 do not include a task type, we manually annotate them with a target task type such as yes-no question answering (see Appendix H for details). Overall, we get 323 meta-templates spanning 16 task types (See Table 13 for the list of task types). Finally, we apply the meta-templates to all the examples in a dataset to create the CTGA dataset, i.e., we replace the keys for the columns in the Jinja templates with corresponding key-value pairs from the examples. If the dataset has multiple meta-templates, we uniformly sample one meta-template per example. We limit the total number of examples per dataset to 100,000. The final training dataset is used to train Bonito.

Training the Bonito Model We train Bonito by fine-tuning Mistral-7B, an open-source decoder language model (Jiang et al., 2023), on the CTGA dataset. The model is trained by optimizing the cross entropy loss over the output tokens. We include all the hyperparameters and training details in Appendix F.1.

4.2 Adapting Models with Bonito

We use Bonito to create synthetic instruction tuning datasets for the target unannotated texts. Then, the target language model is adapted by training on the synthetic dataset to get the specialized language model.

Generating the Synthetic Dataset Figure 1 shows the inference with Bonito to generate the synthetic instruction tuning dataset. The unannotated text and the task type are passed to the Bonito model to get the synthetic instruction-response pairs. The process is repeated for all the unannotated text to get the training dataset. The generated

[1] https://jinja.palletsprojects.com/en/3.1.x/

Task	Dataset	# Unannotated
Yes-No QA	PubmedQA	211,269
	Privacy Policy QA	10,923
Extractive QA	SquadShifts-NYT	10,065
	SquadShifts-Amazon	9,885
	SquadShifts-Reddit	9,803
NLI	Contract-NLI	6,819
	Vitamin C	370,653

Table 1: Statistics of tasks and datasets used in the experiments.

pairs are then post-processed into a standardized instruction-response format for instruction tuning. In each generation, we replace `{{context}}` with the corresponding unannotated text from the input. If the generated output is not parsable due to missing `<|pipe|>`, we filter them out.

Adapting the Target Model We train the target language model on the synthetic instruction tuning dataset containing instruction-response pairs. The model is trained using a cross entropy loss over the response tokens. Additional details in Section 5.1.

5 Experiments

5.1 Experiment Setup

Target Tasks and Datasets We consider three target tasks: yes-no question answering (YNQA), extractive question answering (ExQA), and natural language inference (NLI). Table 1 shows the seven datasets across three task types and the number of unannotated text in each dataset. We use the unannotated text from the datasets to train the specialized language models. For yes-no question answering, we choose PubMedQA (Jin et al., 2019) and Privacy Policy QA (Ravichander et al., 2019). For extractive question answering, we choose the SquadShifts dataset (Miller et al., 2020) that includes splits for the New York Times (NYT), Amazon, and Reddit. Finally, for the NLI task, we choose Contract-NLI (Koreeda and Manning, 2021) and Vitamin C (Schuster et al., 2021). We provide additional details in Appendix A.

In our experiments, we focus on tasks such as question answering and natural language inference that require us to generate a question or hypothesis and an answer. Prior work generates synthetic data for tasks like summarization that do not warrant a specialized task generation model (Yehudai et al., 2024). Other work focuses on generating instructions (Li et al., 2023; Köksal et al., 2023) for long-form text generation tasks where the solution to the instruction is the unannotated text. While these long-form generative tasks are useful for applications such as code generation, domains like biomedical and legal that we consider might benefit more from traditional predictive tasks (Miller, 2024).

Baselines We consider two key baselines: zero-shot and self supervised baseline. For the zero-shot baseline, we prompt the model and run the evaluation without using any of the unannotated text from the target task (**None**). For the self supervised baseline, we use task-adaptive pretraining (**TAPT**) (Gururangan et al., 2020). The learning objective is to continue to the pretraining objective on the unannotated text in the downstream dataset. In our experiments, we use the next word prediction learning objective to fine-tune Mistral-7B and Llama 2 7B models.

Synthetic Task Generation As described in Section 4.1, we prompt Bonito with the unannotated texts and the target task type to generate the instruction tuning data. We use nucleus sampling (Holtzman et al., 2020) with a top P value of 0.95 and a temperature of 0.5, and a maximum sequence length of 256 in the vLLM framework (Kwon et al., 2023).

Models We adapt two pretrained large language models: Mistral-7B (Jiang et al., 2023) and Llama 2 7B (Touvron et al., 2023). They are decoder language models trained with the next word prediction objective on trillions of tokens. Both these models have around 7 billion parameters, with slightly different architectures optimized for sequence length and inference. For more details, see Touvron et al. (2023) and Jiang et al. (2023).

We also consider a more practical setting where we further adapt instruction tuned models to the target task. We first consider an off-the-shelf instruction tuned model: Mistral-7B-Instruct-v0.2. This model based on Mistral-7B achieves comparable performance to Llama 2 13B Chat on the MT-Bench (Zheng et al., 2023). In addition, we train Mistral-7B and Llama 2 models on the T0 split from the P3 dataset (Bach et al., 2022) and adapt them to the target tasks. We call these models Mistral-7B$_{\text{P3}}$ and Llama 2$_{\text{P3}}$. For the instruction tuning details, see Appendix F.2

Training Details We fine-tune the language models on the supervision sources—TAPT, and

Model	Supervision Source	Yes-No QA		Extractive QA			NLI		Average	Δ
		PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C		
Mistral	None	25.6 $_{2.1}$	44.1 $_{2.1}$	24.1 $_{1.6}$	17.5 $_{2.5}$	12.0 $_{2.6}$	31.2 $_{0.6}$	38.9 $_{0.6}$	27.6	-
	TAPT	27.2 $_{2.3}$	46.3 $_{1.2}$	33.5 $_{4.3}$	25.5 $_{5.9}$	22.8 $_{7.0}$	34.2 $_{0.7}$	34.7 $_{2.6}$	32.0	+4.4
	Bonito	**47.1** $_{1.0}$	**52.5** $_{3.0}$	**80.0** $_{1.0}$	**72.5** $_{1.0}$	**71.4** $_{1.6}$	**71.9** $_{0.8}$	**71.7** $_{0.2}$	**66.7**	+39.1
Llama2	None	23.7 $_{0.0}$	43.9 $_{3.0}$	20.1 $_{2.4}$	14.4 $_{2.0}$	11.0 $_{1.9}$	28.6 $_{2.2}$	22.2 $_{2.9}$	23.4	-
	TAPT	23.7 $_{0.0}$	44.1 $_{2.3}$	26.7 $_{6.6}$	25.4 $_{5.9}$	20.6 $_{6.8}$	29.8 $_{2.4}$	26.2 $_{2.0}$	28.1	+4.6
	Bonito	**26.1** $_{2.1}$	**51.4** $_{2.2}$	**75.3** $_{1.9}$	**66.5** $_{1.9}$	**63.7** $_{3.0}$	**63.9** $_{1.1}$	**70.7** $_{0.5}$	**59.7**	+36.2

Table 2: Results for zero-shot task adaptation with pretrained base models. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

Model	Supervision Source	Yes-No QA		Extractive QA			NLI		Average	Δ
		PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C		
Mistral-7B-Instruct-v0.2	None	32.8 $_{0.3}$	**57.9** $_{2.9}$	19.7 $_{2.7}$	15.8 $_{2.4}$	13.0 $_{2.2}$	55.4 $_{2.0}$	58.0 $_{1.1}$	36.1	-
	TAPT	28.3 $_{0.5}$	56.3 $_{2.4}$	37.9 $_{2.2}$	30.1 $_{2.2}$	26.3 $_{4.6}$	42.5 $_{1.8}$	49.6 $_{1.8}$	38.7	+2.6
	Bonito	**41.7** $_{0.4}$	56.2 $_{3.5}$	**80.1** $_{1.0}$	**72.8** $_{1.1}$	**71.8** $_{1.4}$	**70.9** $_{1.8}$	**72.6** $_{0.1}$	**66.6**	+30.5
Mistral-7B$_{P3}$	None	45.1 $_{1.3}$	49.9 $_{2.6}$	73.8 $_{0.8}$	61.0 $_{2.3}$	60.6 $_{2.2}$	33.3 $_{0.7}$	46.0 $_{0.6}$	52.8	-
	TAPT	**51.1** $_{2.2}$	42.8 $_{3.7}$	70.8 $_{1.7}$	59.7 $_{3.2}$	58.0 $_{2.6}$	38.1 $_{3.6}$	43.6 $_{0.4}$	52.0	-0.8
	Bonito	46.1 $_{0.5}$	**56.7** $_{4.3}$	**80.7** $_{0.7}$	**73.9** $_{0.6}$	**72.3** $_{1.1}$	**71.8** $_{0.5}$	**73.9** $_{0.1}$	**67.9**	+15.1
Llama 2$_{P3}$	None	26.0 $_{0.5}$	38.5 $_{1.9}$	64.2 $_{2.6}$	50.6 $_{3.6}$	49.4 $_{4.1}$	23.5 $_{2.6}$	44.6 $_{0.3}$	42.4	-
	TAPT	25.1 $_{0.6}$	42.0 $_{3.8}$	51.4 $_{6.7}$	47.0 $_{4.8}$	42.2 $_{5.8}$	22.6 $_{3.0}$	36.9 $_{1.7}$	38.2	-4.4
	Bonito	**27.0** $_{1.7}$	**56.9** $_{3.8}$	**77.5** $_{1.4}$	**69.6** $_{1.1}$	**68.2** $_{1.9}$	**68.5** $_{0.7}$	**73.7** $_{0.3}$	**63.1**	+20.7

Table 3: Results for zero-shot task adaptation of instruction tuned models. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

Bonito—using Q-LoRA (Dettmers et al., 2023). When further adapting Mistral-7B$_{P3}$ and Llama 2 7B$_{P3}$, we fine-tune the same Q-LoRA adapter on the supervision sources instead of merging and reinitializing the adapters. We train all the models for 1 epoch. If the dataset size is greater than 160,000 examples, then we train for 10,000 steps. We use the same hyperparameter values from Dettmers et al. (2023) to avoid additional hyperparameter tuning. Depending on the dataset, training on four GPUs takes 25 minutes to 17 hours. For more additional details, see Appendix F.5.

Evaluation We evaluate the performance of the models on the test splits of the target datasets (see Table 6 in Appendix A). To prevent "prompt hacking", following Sanh et al. (2022), we first write five prompt templates for the target datasets and then benchmark the model performance. See Appendix I for all the prompts used in our experiments. We follow standard evaluation practices and report the F1 score for all the datasets. Following Radford et al. (2019), we evaluate yes-no question answering and NLI using ranked classification, i.e., we generate the loglikelihood of all the choices and choose the sequence with the highest loglikelihood as the prediction. Following Rajpurkar et al. (2016), we evaluate models on extractive question answering by computing the SQuAD F1 score on the generated output. During evaluation, we use greedy decoding to generate the output from the model and then calculate the SQuAD F1 score for the dataset.

5.2 Adapting Pretrained Models

Table 2 shows that adapting pretrained models with synthetic instruction tuning data generated from Bonito significantly outperforms zero-shot and TAPT. Bonito improves over the zero-shot performance by an average of 37.7 F1 points across Mistral-7B and Llama 2. Although TAPT shows a nominal improvement of only 4.5 F1 points on average, we find that Bonito outperforms TAPT by an average of 33.3 F1 points across both models. This result strengthens our main claim that synthetic instruction tuning data is a much better way of providing domain knowledge compared to self supervision. Finally, we observe that the Mistral-7B shows significantly greater improvement in performance compared to Llama 2 7B suggesting that stronger pretrained models might respond better to synthetic instructions.

Model	Supervision Source	Yes-No QA		Extractive QA			NLI		Average	Δ
		PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C		
Mistral-7B-Instruct-v0.2$_{\text{special}}$	None	$47.5_{0.3}$	$59.1_{1.5}$	$\mathbf{82.6}_{0.5}$	$\mathbf{77.6}_{0.7}$	$\mathbf{75.6}_{0.8}$	$\mathbf{77.3}_{0.1}$	$70.3_{0.1}$	70.0	-
	Bonito	$47.4_{0.2}$	$\mathbf{62.3}_{0.9}$	$82.4_{0.6}$	$76.0_{0.6}$	$74.9_{0.9}$	$75.1_{1.0}$	$71.9_{0.1}$	70.0	**+0.0**
	Bonito$_{\text{special}}$	$\mathbf{50.3}_{0.1}$	$59.8_{1.3}$	$81.8_{0.7}$	$76.4_{0.8}$	$74.5_{1.0}$	$77.0_{0.4}$	**73.5**	**70.5**	**+0.5**
Mistral-7B$_{\text{special}}$	None	$36.7_{1.9}$	$54.4_{1.4}$	$\mathbf{82.6}_{0.5}$	$\mathbf{76.6}_{0.8}$	$75.0_{0.8}$	$75.1_{0.3}$	$71.8_{0.2}$	67.5	-
	Bonito	$42.7_{1.2}$	$55.1_{1.7}$	$82.5_{0.4}$	$76.1_{0.6}$	$74.3_{1.1}$	$76.7_{0.2}$	$71.4_{0.1}$	68.4	**+0.9**
	Bonito$_{\text{special}}$	$\mathbf{49.3}_{0.4}$	$\mathbf{57.2}_{1.6}$	$81.7_{0.8}$	$76.2_{0.8}$	$\mathbf{75.3}_{0.9}$	$\mathbf{76.8}_{0.2}$	$\mathbf{73.8}_{0.1}$	**70.0**	**+2.5**

Table 4: Results for adapting task-specialized models on the downstream target datasets. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

5.3 Adapting Instruction Tuned Models

Table 3 shows that Bonito improves instruction tuned models by an average of 22.1 F1 points whereas TAPT reduces the average performance by 0.8 F1 points. This is because self supervision with TAPT interferes with prior instruction tuning and leads to catastrophic forgetting (French, 1999; Kirkpatrick et al., 2017). In contrast, adapting instruction tuned models with Bonito-generated tasks further improves performance in specialized domains. We also observe that Bonito addresses the task-specific deficiencies and improves the instruction tuned models. For example, we find that Bonito significantly improves Mistral-7B-Instruct-v0.2 performance on extractive question answering as it typically generates chat-like responses for questions. Finally, adapting instruction tuned variants of Mistral-7B and Llama 2 7B achieves a higher F1 score than adapting the pretrained models (Table 2).

6 Analysis

6.1 Impact of Domain Knowledge

Here we ask a key question: are we improving the language model by learning about the domain or are we distilling instructing tuning data from a stronger to a weaker model? To answer this question, we train task-specialized instruction tuned models and then further train them on synthetic tasks generated from Bonito for the target unannotated texts. We create the task-specialized training dataset by selecting the instructions in CTGA with the target task type. We train two task-specialized models in the standard instruction-response format: Mistral-7B-Instruct-v0.2$_{\text{special}}$ and Mistral-7B$_{\text{special}}$. We also train a task-specialized Bonito $_{\text{special}}$ on the same task-specific dataset. See Appendix F.3 for training details.

Table 4 shows that further training on synthetic instructions can improve performance suggesting

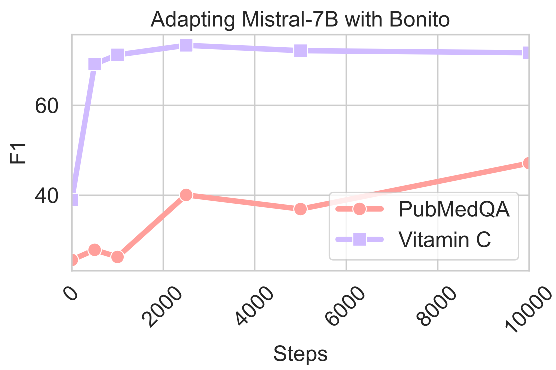

Figure 3: Adapting Mistral-7B with Bonito-generated tasks and evaluating performance after training for different number of steps.

that the model benefits from the unannotated text from the specialized domain. We find that training on Bonito tasks either slightly improves or matches the performance of task-specialized models on average. When we train on Bonito $_{\text{special}}$ tasks, we further improve task-specialized Mistral-7B-Instruct-v0.2 by 0.5 F1 points and Mistral-7B and 2.5 F1 points. We see that the model performance often reduces on extractive QA. We suspect that the model performance has saturated due to the presence of SQuAD in the task-specialized training dataset. To further improve on extractive question answering, we could benefit from having access to a few examples from the target dataset. Finally, we almost always improve performance on Vitamin C and PubMedQA datasets highlighting the importance of training on more task samples (see Section 6.2).

6.2 Effect of the Training Dataset Size

Here we study the effect of the size of the training dataset. In particular, we study how Mistral-7B performance varies when trained on different quantities of synthetic instruction tuning data for PubMedQA and Vitamin C. Figure 3 shows that training on more steps typically improves perfor-

Dataset	Match=Yes	Agreement (>=2)
PubMedQA	72%	97%
Reddit	76%	88%
ContractNLI	71%	99%

Table 5: Agreement between GPT-4o and Bonito generated answers for Bonito tasks. Agreement (>=2) is the agreement percentage when two or more annotators agree on a match or no match.

mance. We find that Bonito on PubMedQA reaches the peak performance of 47.1 F1 points after 10,000 steps but the F1 can fluctuate when trained for fewer steps. In contrast, we find that Bonito gets the highest performance of 73.3 F1 points after 2,500 points and gradually diminishes the performance to 71.7 F1 points. Finally, if available, we suggest using a validation set to select the best-performing model checkpoint.

6.3 Human Evaluation: Agreement with GPT-4o

We manually evaluate Bonito tasks by comparing the answers generated by Bonito and GPT-4o.

Setup We sample 100 unique instructions each from Bonito for PubMedQA, SQuADShifts Reddit, and ContractNLI. Next, we prompt GPT-4o with instructions generated by Bonito to produce an answer. We prefix the instructions with a simple format prompt to produce answers in the desired format with GPT-4o. Finally, we ask humans if GPT-4o and Bonito produce the same (including paraphrased) answers for the instructions. For reproducibility, we use GPT-4o-2024-05-13. We separately ask the first three authors of the paper to compare the responses from both models. We choose the final agreement if two or more annotators agree on either a match or no match.

Results Table 5 shows that Bonito and GPT-4o produce the same answer for Bonito tasks 71 to 76 percent of the time across three datasets, with high inter-annotator agreement. Each dataset reveals different patterns of disagreement. In PubMedQA, GPT-4o generates the response as "unanswerable" when the answer is not present in the passage, whereas Bonito produces either "yes" or "no", or "true" or "false". In SQuADShifts Reddit, Bonito almost always extracts the answer from the paragraph, whereas GPT-4o can generate answers with additional text that may not be present in the paragraph. In ContractNLI, both models can produce plausible answers. In one example, Bonito generates the question, 'Does this imply that "This document is confidential information"? Yes, no, or maybe?'. Bonito answers "yes", whereas GPT-4o produces the answer "maybe". In such cases, we annotate the responses as no match, reducing the agreement.

Our analysis shows that Bonito generates high-quality tasks with accurate answers. However, there is still room to improve the quality of the tasks. Improving the task quality could further increase the downstream model performance. Therefore, we believe that research on conditional task generation is an important direction for future work.

7 Additional Experiments

We briefly describe additional experiments included in Appendix B and C.

In Appendix B, we generate synthetic tasks by prompting Mistral-7B-Instruct-v0.2 and Zephyr-7B-β. Our results show that the synthetic tasks from Mistral-7B-Instruct-v0.2 and Zephyr-7B-β improve the average performance of Mistral-7B but decrease significantly when adapting Mistral$_{P3}$. This indicates that we require high-quality synthetic tasks to increase the performance of strong instruction-tuned models. In Appendix C, we generate synthetic tasks with GPT-4 for Privacy Policy QA, SQuADShifts Reddit, and ContractNLI. Our results show that GPT-4 improves Mistral$_{P3}$ on Privacy Policy QA and ContractNLI but slightly reduces performance on SQuADShifts Reddit. Finally, we analyze the generated tasks and identify common issues with both open-source models and GPT-4, such as the distribution of the label space and "chatty" responses, which potentially lead to the drop in performance.

8 Conclusion

We present Bonito, an open-source model for conditional task generation that converts unannotated text into instruction tuning datasets. We show that training with synthetic instruction tuning datasets in specialized domains is a strong alternative to self supervision. Our experiments demonstrate that Bonito-generated instructions improve pretrained and instruction tuned models on zero-shot task adaptation. Overall, Bonito enables practitioners to adapt large language models to tasks on their data without annotations.

Acknowledgements

We appreciate Yeganeh Kordi, Zheng-Xin Yong, Aidan LaBella, and members of the Brown SuperLab for their thoughtful comments and feedback on our draft. We gratefully acknowledge support from Cisco. Disclosure: Stephen Bach is an advisor to Snorkel AI, a company that provides software and services for data-centric artificial intelligence.

Limitations

Our work relies on the availability of large amounts of unannotated text. If only a small quantity of unannotated text is present, the target language model, after adaptation, may experience a drop in performance. While we demonstrate positive improvements on pretrained and instruction-tuned models, our observations are limited to the three task types considered in our experiments.

Potential Risks

Bonito poses risks similar to those of any large language model. For example, our model could be used to generate factually incorrect datasets in specialized domains. Our model can exhibit the biases and stereotypes of the base model, Mistral-7B, even after extensive supervised fine-tuning. Finally, our model does not include safety training and can potentially generate harmful content.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *ArXiv preprint*, abs/2311.16079.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *ArXiv preprint*, abs/2306.16092.

Cheng Deng, Tianhang Zhang, Zhongmou He, Qiyuan Chen, Yuanyuan Shi, Le Zhou, Luoyi Fu, Weinan Zhang, Xinbing Wang, Cheng Zhou, Zhouhan Lin, and Junxian He. 2023. Learning a foundation language model for geoscience knowledge understanding and utilization. *ArXiv preprint*, abs/2306.05064.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv preprint*, abs/2305.14314.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on*

Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1180–1189. JMLR.org.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *ArXiv preprint*, abs/2305.15717.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *ArXiv preprint*, abs/2308.08998.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint*, abs/1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Meeting of the Association for Computational Linguistics (ACL)*.

Tony Huang, Jack Chu, and Fangyun Wei. 2022. Unsupervised prompt learning for vision-language models. *ArXiv preprint*, abs/2204.03649.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartlomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, P. Milkowski, Marcin Oleksy, Maciej Piasecki, Lukasz Radliński, Konrad Wojtasik, Stanislaw Woźniak, and Przemyslaw Kazienko. 2023. ChatGPT: Jack of all trades, master of none. *Information Fusion*, page 101861.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation. *ArXiv preprint*, abs/2308.06259.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *ArXiv preprint*, abs/2301.13688.

Cristina Menghini, Andrew Delworth, and Stephen H Bach. 2023. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. *arXiv e-prints*, pages arXiv–2306.

Jeff Miller. 2024. Generative AI is hot, but predictive AI remains the workhorse — cio.com. `https://www.cio.com/article/1303984/generative-ai-is-hot-but-predictive-ai-remains-the-workhorse-2.html`. [Accessed 10-02-2024].

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *ArXiv preprint*, abs/2211.01786.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475, Online. Association for Computational Linguistics.

Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. In-BoXBART: Get instructions into biomedical multi-task learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States. Association for Computational Linguistics.

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, Zachary 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 *Advances in Neural Information Processing*

Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 8024–8035.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *ArXiv preprint*, abs/2304.03277.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. Question answering for privacy policies: Combining computational and legal perspectives. 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 4947–4958, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In ChatGPT we trust? Measuring and characterizing the reliability of chatgpt. *ArXiv preprint*, abs/2304.08979.

Zhengxiang Shi and Aldo Lipani. 2023. Don't stop pretraining? make prompt-based fine-tuning powerful learner. *ArXiv preprint*, abs/2305.01711.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023a. Large language models encode clinical knowledge. *Nature*, pages 1–9.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. 2023b. Towards expert-level medical question answering with large language models. *ArXiv preprint*, abs/2305.09617.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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

David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Rricha Jalota, Floris Fok, Michael Ramos, Ian van Wyk, Abdallah Nasir, Hayden Goldstein, et al. 2024. Climategpt: Towards ai synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*.

Together. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of lm-based question and answer generation methods. *ArXiv preprint*, abs/2305.17002.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language model with self generated instructions. In *Meeting of the Association for Computational Linguistics (ACL)*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-natural instructions: Generalization via declarative instructions on 1600+ tasks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jason Wei, Maarten Bosma, Vincent Y. 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A large language model for finance. *ArXiv preprint*, abs/2303.17564.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *ArXiv preprint*, abs/2402.01364.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofo: A benchmark to evaluate llms' format-following capability. *arXiv preprint arXiv:2402.18667*.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *ArXiv preprint*, abs/2401.14367.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. ChatDoctor: A medical chat model fine-tuned on LLaMA model using medical domain knowledge. *ArXiv preprint*, abs/2303.14070.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv preprint*, abs/2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *ArXiv preprint*, abs/2305.11206.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. Can large language models transform computational social science? *ArXiv preprint*, abs/2305.03514.

A Datasets

We briefly describe the datasets used in our experiments. We get all the datasets from the datasets library (Lhoest et al., 2021). For all the datasets, we consider five prompt templates (see Appendix I). Table 6 shows the statistics for the test splits in the evaluation datasets. Below we include details about the evaluation datasets:

- **PubMedQA** (Jin et al., 2019): The dataset asks questions about PubMed abstracts that can be answered with yes, no, or maybe. We use the abstracts without the questions as unannotated text for adaptation. During the evaluation, we provide the PubMed abstract along with the question to the model.

- **Privacy Policy QA** (Ravichander et al., 2019): The dataset consists of paragraphs from privacy policies with corresponding questions. The task involves determining the relevance of each question, formatted as a yes-no question-answering task. We use the processed test split in Privacy Policy QA from Guha et al. (2023) as unannotated text.

- **SquadShifts** (Miller et al., 2020): The dataset is designed to test the robustness of extractive question answering models. We use three of the four test sets in our work — New York Times articles, Reddit posts, and Amazon product reviews. During training, we use the articles or context from the test set without the questions and generate extractive question answering tasks with Bonito. During evaluation, we evaluate the same test set with the questions in the dataset.

- **ContractNLI** (Koreeda and Manning, 2021): ContractNLI is a natural language inference task to aid contract review. Given a hypothesis about a clause in a contract, the model predicts if the hypothesis is supported, refuted, or not mentioned.

- **Vitamin C** (Schuster et al., 2021): This dataset focuses on fact verification in Wikipedia framed as a natural language inference task. Each example consists of an evidence text from Wikipedia and a corresponding fact. The model is asked to indicate whether the fact is supported, refuted, or neutral.

Dataset	# Classes	# Test Examples
PubmedQA	3	500
Privacy Policy QA	2	10,923
SquadShifts-NYT	-	10,065
SquadShifts-Amazon	-	9,885
SquadShifts-Reddit	-	9,803
Contract-NLI	3	1,991
Vitamin C	3	55,197

Table 6: Statistics for the evaluation test sets in the datasets from our experiments. "-" in the number of classes indicates a generation task.

```
Task Type: Yes-no question answering

Prompt: Generate exactly one question that can
be answered by a yes or a no for the paragraph
below.  The question should be parsable and
enclosed in quotes ("").
<context>
```
```
Task Type: Extractive question answering

Prompt: Generate exactly one question that
can be answered by selecting 1 to 10 words
from the paragraph below.  The question should
be parsable and enclosed in quotes ("").
<context>
```
```
Task Type: Natural language inference

Prompt: Generate exactly one high-level
statement or a hypothesis for the following
paragraph.  The hypothesis about the paragraph
can be true, false, or neither.  Make sure the
output is less than 10 words.  The hypothesis
should be parsable and enclosed in quotes
("").
<context>
```

Table 7: Prompts used generated tasks with Mistral-Instruct-v0.2, Zephyr-β, and GPT-4. We replace `<context>` with the unannotated text.

B Generating Tasks with Open-Source Models

We use Mistral-Instruct-v0.2 and Zephyr-β, two popular openly available models, to generate instruction tuning data. Then, we adapt pretrained Mistral-7B and Mistral-7B-P3 on the generated tasks.

B.1 Generating Synthetic Datasets

Here we describe the process of creating synthetic datasets with Mistral-Instruct-v0.2 and Zephyr-β. We prompt these models to generate questions or hypotheses for the target unannotated text. Table 7 shows the prompts we used to generate the tasks. Creating these prompts required a tremendous amount of prompt engineering as they strug-

Model	Supervision Source	Yes-No QA		Extractive QA			NLI		Average	Δ
		PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C		
Mistral-7B	None	25.6 $_{2.1}$	44.1 $_{2.1}$	24.1 $_{1.6}$	17.5 $_{2.5}$	12.0 $_{2.6}$	31.2 $_{0.6}$	38.9 $_{0.6}$	27.6	-
	Mistral-Instruct-v0.2	29.4 $_{0.8}$	50.1 $_{5.5}$	22.3 $_{1.7}$	17.2 $_{1.9}$	13.6 $_{2.1}$	55.3 $_{1.4}$	52.2 $_{1.5}$	34.3	+6.7
	Zephyr-β	32.2 $_{1.6}$	**59.4** $_{2.3}$	20.4 $_{1.5}$	18.2 $_{1.9}$	15.0 $_{2.1}$	33.3 $_{2.9}$	51.9 $_{3.0}$	32.9	+5.3
	Bonito	**47.1** $_{1.0}$	52.5 $_{3.0}$	**80.0** $_{1.0}$	**72.5** $_{1.0}$	**71.4** $_{1.6}$	**71.9** $_{0.8}$	**71.7** $_{0.2}$	**66.7**	+39.1
Mistral-7B$_{\text{P3}}$	None	45.1 $_{1.3}$	49.9 $_{2.6}$	73.8 $_{0.8}$	61.0 $_{2.3}$	60.6 $_{2.2}$	33.3 $_{0.7}$	46.0 $_{0.6}$	52.8	-
	Mistral-Instruct-v0.2	34.1 $_{1.1}$	**62.1** $_{1.4}$	24.1 $_{1.7}$	18.8 $_{2.2}$	15.3 $_{2.2}$	53.9 $_{1.8}$	53.5 $_{1.0}$	37.4	-15.4
	Zephyr-β	38.8 $_{1.7}$	55.3 $_{3.5}$	22.2 $_{1.6}$	20.0 $_{2.0}$	16.6 $_{2.0}$	36.5 $_{5.7}$	51.6 $_{3.2}$	34.4	-18.4
	Bonito	**46.1** $_{0.5}$	56.7 $_{4.3}$	**80.7** $_{0.7}$	**73.9** $_{0.6}$	**72.3** $_{1.1}$	**71.8** $_{0.5}$	**73.9** $_{0.1}$	**67.9**	+15.1

Table 8: Results for zero-shot task adaptation with tasks generated from Mistral-Instruct-v0.2 and Zephyr-β. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

gled to follow the prompt format (Xia et al., 2024). We first generate the question or the hypothesis and then re-prompt the model to produce the answer. For question answering tasks, we prepend the question as the prompt followed by the unannotated text to generate the output. For the NLI datasets, we use five prompt templates from the ANLI dataset in Bach et al. (2022) and plug in the hypothesis and the unannotated text as the input to the model to generate the answer. We use the same input and output to adapt the pretrained and instruction tuned models. For all the generations, we use a top-P of 0.95, temperature of 0.5, and maximum token length of 256.

B.2 Results

Table 8 shows results for zero-shot task adaptation with openly available models. We see that both Mistral-7B-Instruct-v0.2 and Zephyr-7B-β improve performance over the pretrained Mistral-7B but find that they severely hurt average performance compared to Mistral-7B$_{\text{P3}}$.

We suspect that the drop in performance is due to issues related to the generated tasks. For extractive question answering, we find that Mistral-7b-Instruct-v0.2 and Zephyr-β often generate questions with multiple sub-questions that cannot be easily answered by extracting words from the context. Furthermore, the responses are "chatty" which might not be appropriate for extractive question answering. We also observe that the generated questions are often "positive", i.e., they usually have "yes" or "true" as the answer. For example, 68% of the questions generated by Zephyr-β for PubMedQA have answers starting with "yes" or "true" but only 5% of the questions have answers that start with "no" or "false". We observe a similar "positive" bias in the hypotheses generated for natural language inference datasets.

Model	Sup. src.	PrivacyQA	Reddit	ContractNLI
Mistral-7B$_{\text{P3}}$	None	49.9 $_{2.6}$	61.0 $_{2.8}$	33.3 $_{0.7}$
	GPT-4	**57.2** $_{4.8}$	52.4 $_{3.0}$	43.1 $_{0.7}$
	Bonito	56.7 $_{4.3}$	**72.3** $_{1.1}$	**71.8** $_{0.5}$

Table 9: Results for zero-shot task adaptation with task generated from GPT-4. We report the F1 and the standard error averaged across five prompts templates for all the datasets.

C Generating Tasks with GPT-4

Here we use GPT-4 to generate tasks to adapt Mistral-7B$_{\text{P3}}$.

C.1 Generating Synthetic Datasets

We prompt GPT-4 to generate tasks for Privacy Policy QA, SQuADShifts Reddit, and Contract NLI. For simplicity, we use the same prompts from Appendix B.1 to generate questions and hypotheses (see Table 7). For Privacy Policy QA, we add a simple instruction prefix to answer the question with yes or no along with the question and the context to generate the answer. For extractive question answering, we add the prefix "Extract the exact words from the paragraph for the question. If the question is not answerable, say N/A." before the question and the context and produce the answer. We use a simpler prefix "Answer the following question." when training the downstream model on SQuADShifts Reddit. Finally, for ContractNLI, we use the same prompts from Appendix B.1 to generate answers. For all the generations, we use gpt-4-0613 with a maximum token length of 256, top-P of 0.95, and temperature of 0.5.

C.2 Results

Table 9 shows that tasks generated by GPT-4 improve performance over Mistral-7BP3 on Privacy Policy QA and ContractNLI but slightly reduce performance on SQuADShifts Reddit. While GPT-4 is

Model	Yes-No QA		Extractive QA			NLI		Average
	PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C	
FLAN-T5-XXL (11B)	50.0 $_{0.4}$	**62.5** $_{2.2}$	**84.2** $_{0.2}$	72.3 $_{1.9}$	70.1 $_{3.1}$	45.4 $_{3.5}$	62.5 $_{2.7}$	63.9
FLAN-T5-XL (3B)	**52.5** $_{0.2}$	59.3 $_{1.6}$	82.1 $_{1.3}$	68.1 $_{5.4}$	67.3 $_{3.1}$	37.0 $_{0.6}$	54.7 $_{0.4}$	60.2
Mistral-7B-Instruct-v0.2 + Bonito	41.7 $_{0.4}$	56.2 $_{3.5}$	80.1 $_{1.0}$	72.8 $_{1.1}$	71.8 $_{1.4}$	70.9 $_{1.8}$	72.6 $_{0.1}$	66.6
Mistral-7B$_{\text{P3}}$ + Bonito	46.1 $_{0.5}$	56.7 $_{4.3}$	80.7 $_{0.7}$	**73.9** $_{0.6}$	**72.3** $_{1.1}$	**71.8** $_{0.5}$	**73.9** $_{0.1}$	**67.9**

Table 10: Results comparing zero-shot task adaptation of instruction tuned models with FLAN-T5 models. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

a much better task generator than the open-source models, we find that GPT-4 also suffers from a similar issue. For example, ContractNLI often has a positive hypothesis and PrivacyQA has a question with the answer yes. While GPT-4 follows the instruction to generate exactly one question for the paragraph, we find that it produces slightly longer answers to the question. The SQuAD metric penalizes if there unwanted tokens in the answers. Finally, the cost of generating tasks with GPT-4 makes it prohibitively expensive to generate tasks for larger datasets like PubMedQA and Vitamin C.

D Bonito vs. FLAN

We evaluate the zero-shot performance of FLAN-T5-XXL (11B) and FLAN-T5-XL (3B) models (Longpre et al., 2023) on the target datasets used in our experiments. Table 10 shows that Mistral-7B-Instruct-v0.2 and Mistral$_{\text{P3}}$ with Bonito-generated tasks improves over FLAN-T5-XXL (11B) by 2.7 F1 points and 4.0 F1 points. Our results also show that Mistral-7B-Instruct-v0.2 and Mistral$_{\text{P3}}$ with Bonito outperforms FLAN-T5-XL (3B) by 6.4 F1 points and 7.7 F1 points.

E Bonito with Smaller Models

We report an additional comparison with Bonito trained on Pythia (2.8B) (Biderman et al., 2023). We follow the same experimental setup used in Section 5.1.

Results Table 11 shows that Bonito improves Pythia (2.8B) by an average of 30.3 F1 points across all the datasets. We observe that Pythia (2.8B) with Bonito performs better than Mistral with TAPT and Llama 2 with TAPT despite being twice as small (See Table 2). These results show that Bonito can be used to create small but powerful specialized language models.

F Training Details

Here we provide training details for models used in the paper.

F.1 Training Bonito

We train Mistral-7B on the conditional task generation with attributes (CTGA) dataset. From the training set, we uniformly sample 10,000 examples as the validation set to monitor the loss. The rest of the dataset is used for training Bonito. We train the model using Q-LoRA (Dettmers et al., 2023) by optimizing the cross entropy loss over the output tokens. The model is trained for 100,000 steps. The training takes about 4 days on four GPUs to complete. We include all the hyperparameters in Appendix F.5.

The same training recipe can be used to train other existing language models such as Falcon (Almazrouei et al., 2023), Pythia (Biderman et al., 2023), and RedPajama (Together, 2023). While models such as Llama 2 (Touvron et al., 2023) can be trained on CTGA, their license prohibits the use of the output to enhance any other large language model.

F.2 Instruction Tuned Models

Here we describe the procedure to train Mistral-7B$_{\text{P3}}$ and Llama 2 7B$_{\text{P3}}$. We use the processed T0 dataset from Muennighoff et al. (2022). Since the dataset is large, we uniformly sample 1.6 million input-output examples and train the language model on them. Following Dettmers et al. (2023), we train the model for 10,000 steps with Q-LoRA and optimize the cross entropy loss over the output tokens. The training takes about 10 hours on four GPUs to complete. For the rest of the hyperparameters, see Appendix F.5.

F.3 Training Task-Specialized Models

To train the task-specialized Mistral-7B-Instruct-v0.2$_{\text{special}}$ and Mistral-7B$_{\text{special}}$, we create a task-specific dataset by filtering out task types from the

Model	Yes-No QA		Extractive QA			NLI		Average
	PubMedQA	PrivacyQA	NYT	Amazon	Reddit	ContractNLI	Vitamin C	
Pythia (2.8B)	23.7 $_{0.0}$	42.2 $_{1.4}$	11.9 $_{0.9}$	8.9 $_{0.5}$	8.0 $_{0.6}$	20.8 $_{3.5}$	25.4 $_{1.5}$	20.1
Pythia (2.8B) + Bonito	**25.9** $_{2.2}$	**51.6** $_{0.9}$	**59.8** $_{4.2}$	**52.2** $_{3.5}$	**51.7** $_{4.3}$	**48.4** $_{2.5}$	**63.3** $_{0.9}$	**50.4**

Table 11: Results for pretrained Pythia and Pythia adapted with Bonito. We report the F1 and the standard error averaged across five prompt templates for all the datasets.

Hyperparameters	**Values**
Q-LoRA rank (r)	64
Q-LoRA scaling factor (α)	4
Q-LoRA dropout	0
Optimizer	Paged AdamW
Learning rate scheduler	linear
Max. learning rate	$1e-04$
Min. learning rate	0
Weight decay	0
Dropout	0
Max. gradient norm	0.3
Effective batch size	16
Max. input length	2,048
Max. output length	2,048

Table 12: The hyperparameters used to train all the models in our experiments.

Task type	**# Examples**
Summarization	284,589
Sentiment	233,530
Multiple-choice question answering	229,066
Extractive question answering	222,769
Topic classification	209,980
Natural language inference	100,250
Question generation	92,847
Text generation	86,835
Question answering without choices	75,159
Paraphrase identification	47,848
Sentence completion	30,246
Yes-no question answering	25,895
Word sense disambiguation	5,428
Paraphrase generation	2,550
Textual entailment	2,490
Coreference resolution	554
Total	1,650,036

Table 13: Task distribution in the conditional task generation with attributes dataset.

CTGA dataset. We selected datasets containing templates that correspond to three task types: yes-no question answering, extractive question answering, and natural language inference. The datasets have a total of 130,703 examples for yes-no question answering, 378,167 examples for extractive question answering, and 100,250 examples for natural language inference.

To train the task-specialized Bonito $_{\text{special}}$, we convert the same task templates into meta templates. Then, we use the meta templates to generate the dataset to train the model.

For fairness, we use the same hyperparameters to train task-specialized Bonito and the task-specialized Mistral-7B-Instruct-v0.2$_{\text{special}}$ and Mistral-7B$_{\text{special}}$ models. Since the datasets have significantly fewer examples than CTGA, we train these models for at most 10,000 steps. If the training mixture has less than 160,000 examples, we train the Bonito model for 1 epoch. The training on four GPUs takes about 4 to 10 hours. For the rest of the hyperparameters, see Appendix F.5.

F.4 Software and Hardware Details

Our codebase is built using the transformers (Wolf et al., 2019) library in PyTorch (Paszke et al., 2019). We train all the models in a distributed multi-GPU environment using DeepSpeed (Rasley et al., 2020). We use the distributed data parallel in DeepSpeed to increase the effective batch size during training. For training and evaluation, we use the following GPUs depending on their availability on our compute cluster: NVIDIA GeForce RTX 3090, NVIDIA RTX A5500, NVIDIA RTX A6000, NVIDIA RTX A5000, and NVIDIA A40.

F.5 Hyperparameters

Throughout our fine-tuning experiments, unless otherwise mentioned, we use the hyperparameters from Dettmers et al. (2023). Table 12 shows the hyperparameters in our experiments. We use gradient accumulation to achieve the effective batch size of 16. We also use gradient checkpointing to train large models like Llama 2 7B and Mistral-7B.

G Use of AI Assistants

Our work used AI Assistants such as ChatGPT and Grammarly for spell-checking and fixing minor grammatical mistakes. We also use GitHub CoPilot in VSCode to write our codebase.

H Conditional Task Generation with Attributes: Datasets and Tasks

Table 13 shows the task distribution of the conditional task generation with attributes dataset. Table 14 lists all the datasets along with the task types in the dataset. The dataset includes 16 task types across 38 datasets. The task types are summarization, sentiment analysis, multiple-choice question answering, extractive question answering, topic classification, natural language inference, question generation, text generation, question answering without choices, paraphrase identification, sentence completion, yes-no question answering, word sense disambiguation, paraphrase generation, textual entailment, and coreference resolution. The difference between extractive question answering and question answering without choices is that in extractive question answering the target answer is present in the context whereas in question answering without choices, that always is not the case.

I Prompts for Evaluation

I.1 PubmedQA

Dataset from Jin et al. (2019):

- Input

```
Given a passage: {{ context.contexts | join("
") }}

Answer the question: {{question}}

Summarize the above answer as YES, NO, or
MAYBE?
```

Target

```
{{final_decision}}
```

Answer Choices

```
yes ||| no ||| maybe
```

- Input

```
I'm a doctor and I want to answer the question
"{{question}}" using The following passage:

{{ context.contexts | join(" ") }}

Summarize the above answer as YES, NO, or
MAYBE?
```

Target

```
{{final_decision}}
```

Answer Choices

```
yes ||| no ||| maybe
```

- Input

```
What is the answer to the question
"{{question}}" based on The following
passage:

{{ context.contexts | join(" ") }}

Summarize the above answer as YES, NO, or
MAYBE?
```

Target

```
{{final_decision}}
```

Answer Choices

```
yes ||| no ||| maybe
```

- Input

```
Please answer the question "{{question}}"
using The following passage:

{{ context.contexts | join(" ") }}

Summarize the above answer as YES, NO, or
MAYBE?
```

Target

```
{{final_decision}}
```

Answer Choices

```
yes ||| no ||| maybe
```

- Input

```
Given the following passage, answer the
question: "{{question}}"

            Passage: {{ context.contexts |
join(" ") }}

Summarize the above answer as YES, NO, or
MAYBE?
```

Target

```
{{final_decision}}
```

Answer Choices

```
yes ||| no ||| maybe
```

I.2 Privacy Policy QA

Dataset from Ravichander et al. (2019).

- Input

```
Given the context, is this related to the
question?
Context: {{text}}
Question: {{question}}
```

Target

```
{{answer}}
```

Answer Choices

```
Relevant|||Irrelevant
```

- Input

```
Is this question
"{{question}}"
related to this context
"{{text}}"?
```

Target

```
{% if answer == "Relevant" %} Yes {% else %}
No {% endif %}
```

Answer Choices

```
Yes|||No
```

- Input

```
Can this
"{{text}}"
help answer this question
"{{question}}"?
```

Target

```
{% if answer == "Relevant" %} Yes {% else %}
No {% endif %}
```

Answer Choices

```
Yes|||No
```

- Input

```
As a lawyer, can you answer the question
given the context?
Question: {{question}}
Context:{{text}}
```

Target

```
{% if answer == "Relevant" %} Yes {% else %}
No {% endif %}
```

Answer Choices

```
Yes|||No
```

- Input

```
Question:{{question}}
Context:{{text}}
Is the question related to the context?
```

Target

```
{% if answer == "Relevant" %} Yes {% else %}
No {% endif %}
```

Answer Choices

```
Yes|||No
```

I.3 SQuADShifts

Dataset from Miller et al. (2020).

I.3.1 NYT

- Input

```
After reading the following paragraph, please
answer this question: {{question}}

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I'm working on the final exam for my class
and am trying to figure out the answer to the
question "{{question}}" I found the following
info on New York Times and I think it has the
answer. Can you tell me the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I've always wondered: {{question}}

I searched New York Times and this is what I
found. What's the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
{{context}}

With the help of the passage, please answer
the following question:
{{question}}
```

Target

```
{{answers["text"]|choice}}
```

- Input

```
{{["Question", "Problem"]  | choice}}
{{range(1, 12) | choice}}: {{question}}

Hint: {{context}}
```

Target

```
{{answers["text"] | most_frequent | choice}}
```

I.3.2 Amazon

- Input

```
After reading the following paragraph, please
answer this question: {{question}}

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I'm working on the final exam for my class
and am trying to figure out the answer to the
question "{{question}}" I found the following
info on Amazon and I think it has the answer.
Can you tell me the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I've always wondered: {{question}}

I searched Amazon and this is what I found.
What's the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
{{context}}

With the help of the passage, please answer
the following question:
{{question}}
```

Target

```
{{answers["text"]|choice}}
```

- Input

```
{{["Question", "Problem"]  | choice}}
{{range(1, 12) | choice}}: {{question}}

Hint: {{context}}
```

Target

```
{{answers["text"] | most_frequent | choice}}
```

I.3.3 Reddit

- Input

```
After reading the following paragraph, please
answer this question: {{question}}

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I'm working on the final exam for my class
and am trying to figure out the answer to the
question "{{question}}" I found the following
info on Reddit and I think it has the answer.
Can you tell me the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
I've always wondered: {{question}}

I searched Reddit and this is what I found.
What's the answer?

{{context}}
```

Target

```
{{answers['text'] | most_frequent | choice}}
```

- Input

```
{{context}}

With the help of the passage, please answer
the following question:
{{question}}
```

Target

```
{{answers["text"]|choice}}
```

- Input

```
{{["Question", "Problem"]  | choice}}
{{range(1, 12) | choice}}: {{question}}

Hint: {{context}}
```

Target

```
{{answers["text"] | most_frequent | choice}}
```

I.4 ContractNLI

Dataset from Koreeda and Manning (2021).

- Input

```
Suppose {{premise}} Can we infer that
"{{hypothesis}}"? yes, no or maybe?
```

Target

```
{{answer_choices[label]}}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

- Input

```
{{premise}}

Question: Does this imply that
"{{hypothesis}}"? yes, no or maybe?
```

Target

```
{{answer_choices[label]}}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

- Input

```
Take the following as truth: {{premise}} Then
the following statement: "{{hypothesis}}" is
{{"true"}}, {{"false"}}, or
{{"inconclusive"}}?
```

Target

```
{{answer_choices[label]}}
```

Answer Choices

```
False ||| True ||| Inconclusive
```

- Input

```
{{premise}} Based on that information, is the
claim: "{{hypothesis}}" {{"true"}},
{{"false"}}, or {{"inconclusive"}}?
```

Target

```
{{ answer_choices[label]}}
```

Answer Choices

```
False ||| True ||| Inconclusive
```

- Input

```
{{premise}} Based on the previous passage, is
it true that "{{hypothesis}}"? Yes, no, or
maybe?
```

Target

```
 {{ answer_choices[label] }}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

I.5 Vitamin C

Dataset from Schuster et al. (2021).

- Input

```
Suppose {{evidence}} Can we infer that
"{{claim}}"? yes, no or maybe?
```

Target

```
{% if label == "REFUTES" %} No {% elif label
== "SUPPORTS" %} Yes {% else %} Maybe {%
endif %}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

- Input

```
{{evidence}}

Question: Does this imply that "{{claim}}"?
yes, no or maybe?
```

Target

```
{% if label == "REFUTES" %} No {% elif label
== "SUPPORTS" %} Yes {% else %} Maybe {%
endif %}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

- Input

```
Take the following as truth: {{evidence}}
Then the following statement: "{{claim}}" is
{{"true"}}, {{"false"}}, or
{{"inconclusive"}}?
```

Target

```
{% if label == "REFUTES" %} False {% elif
label == "SUPPORTS" %} True {% else %}
Inconclusive {% endif %}
```

Answer Choices

```
False ||| True ||| Inconclusive
```

- Input

```
{{evidence}}
Based on that information, is the claim:
"{{claim}}" {{"true"}}, {{"false"}}, or
{{"inconclusive"}}?
```

Target

```
{% if label == "REFUTES" %} False {% elif
label == "SUPPORTS" %} True {% else %}
Inconclusive {% endif %}
```

Answer Choices

```
False ||| True ||| Inconclusive
```

- Input

```
{{evidence}} Based on the previous passage, is
it true that "{{claim}}"? Yes, no, or maybe?
```

Target

```
{% if label == "REFUTES" %} No {% elif label
== "SUPPORTS" %} Yes {% else %} Maybe {%
endif %}
```

Answer Choices

```
No ||| Yes ||| Maybe
```

J Qualitatitve Examples

Table 16 shows Bonito-generated tasks for the PubMedQA, SQuADShifts Amazon, and ContractNLI.

Dataset name	Task types
adversarial_qa/dbert	Extractive question answering Question generation
adversarial_qa/dbidaf	Extractive question answering Question generation
adversarial_qa/droberta	Extractive question answering Question generation
ag_news	Topic classification
amazon_polarity	Sentiment
anli	Natural language inference
app_reviews	Multiple-choice question answering Question answering without choices Text generation
cnn_dailymail/3.0.0	Summarization Text generation
cosmos_qa	Multiple-choice question answering Question answering without choices Question generation
dbpedia_14	Topic classification
dream	Multiple-choice question answering Text generation
duorc/ParaphraseRC	Extractive question answering Question generation Summarization Text generation
duorc/SelfRC	Extractive question answering Question generation Summarization Text generation
gigaword	Summarization Text generation
glue/mrpc	Paraphrase generation Paraphrase identification
hellaswag	Sentence completion Topic classification
imdb	Sentiment
multi_newspaws/labeled_final	Paraphrase generation Paraphrase identification
qasc	Multiple-choice question answering

Table 14: Dataset names and the prompted task types in the dataset [1/2].

Dataset name	Task types
quail	Multiple-choice question answering Question answering without choices
quartz	Multiple-choice question answering
quoref	Extractive question answering Summarization
race/all	Multiple-choice question answering Question answering without choices Question generation Yes-no question answering
ropes	Extractive question answering
rotten_tomatoes	Sentiment
samsum	Summarization Text generation
social_i_qa	Multiple-choice question answering Question answering without choices Question generation Yes-no question answering
squad	Extractive question answering Question generation
super_glue/boolq	Yes-no question answering
super_glue/cb	Natural language inference
super_glue/copa	Sentence completion
super_glue/record	Extractive question answering Multiple-choice question answering
super_glue/rte	Textual entailment
super_glue/wic	Word sense disambiguation
super_glue/wsc.fixed	Coreference resolution
wiki_hop/original	Multiple-choice question answering Question answering without choices
xsum	Summarization
yelp_review_full	Sentiment

Table 15: Dataset names and the prompted task types in the dataset [2/2].

Dataset: PubMedQA **Task type:** Yes-no Question Answering **Input:** Palmitate, a saturated fatty acid (FA), is known to induce toxicity and cell death in various types of cells. Resveratrol (RSV) is able to prevent pathogenesis and/or decelerate the progression of a variety of diseases. Several in vitro and in vivo studies have also shown a protective effect of RSV on fat accumulation induced by FAs. Additionally, endoplasmic reticulum (ER) stress has recently been linked to cellular adipogenic responses. To address the hypothesis that the RSV effect on excessive fat accumulation promoted by elevated saturated FAs could be partially mediated by a reduction of ER stress, we studied the RSV action on experimentally induced ER stress using palmitate in several cancer cell lines. We show that, unexpectedly, RSV promotes an amplification of palmitate toxicity and cell death and that this mechanism is likely due to a perturbation of palmitate accumulation in the triglyceride form and to a less important membrane fluidity variation. Additionally, RSV decreases radical oxygen species (ROS) generation in palmitate-treated cells but leads to enhanced X-box binding protein-1 (XBP1) splicing and C/EBP homologous protein (CHOP) expression. These molecular effects are induced simultaneously to caspase-3 cleavage, suggesting that RSV promotes palmitate lipoapoptosis primarily through an ER stress-dependent mechanism. Moreover, the lipotoxicity reversion induced by eicosapentaenoic acid (EPA) or by a liver X receptor (LXR) agonist reinforces the hypothesis that RSV-mediated inhibition of palmitate channeling into triglyceride pools could be a key factor in the aggravation of palmitate-induced cytotoxicity.Our results suggest that RSV exerts its cytotoxic role in cancer cells exposed to a saturated FA context primarily by triglyceride accumulation inhibition, probably leading to an intracellular palmitate accumulation that triggers a lipid-mediated cell death. Additionally, this cell death is promoted by ER stress through a CHOP-mediated apoptotic process and may represent a potential anticancer strategy. Question: does resveratrol induce cell death in cancer cells Answer: **Output:** Yes
Dataset: SquadShifts Amazon **Task type:** Extractive question answering **Input:** Here is a review left by a customer on a product. Would you be able to answer the question asked by the customer about the product? Review: The storage bag I got is different from the picture. It's a mesh see-through material, where in the picture it doesn't look like it would be see-through. It still does what it's supposed to, so no complaints. I love the velcro loop at the top that I can use to hang it on the bar under the shelf in my closet. Question: How did the customer hang the storage bag? **Output:** on the bar under the shelf
Dataset: Contract NLI **Task type:** Natural language infernce **Input:** 4. Nothing in this Agreement is to be construed as granting the Recipient, by implication or otherwise, any right whatsoever with respect to the Confidential Information or part thereof. Using only the above description and what you know about the world, "The recipient is a person." is definitely correct, incorrect, or inconclusive? **Output:** Inconclusive

Table 16: Example generations from Bonito for PubMedQA, SQuADShifts Amazon, and ContractNLI.