LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions

Minghao Wu[1,2*] **Abdul Waheed**[1] **Chiyu Zhang**[1,3] **Muhammad Abdul-Mageed**[1,3] **Alham Fikri Aji**[1]
[1]Mohamed bin Zayed University of Artificial Intelligence
[2]Monash University [3]The University of British Columbia
{minghao.wu,abdul.waheed,chiyu.zhang,muhammad.mageed,alham.fikri}@mbzuai.ac.ae

Abstract

Large language models (LLMs) with instruction finetuning demonstrate superior generative capabilities. However, these models are resource intensive. To alleviate this issue, we explore distilling knowledge from instruction-tuned LLMs to much smaller ones. To this end, we carefully develop a *large* set of 2.58M instructions based on both existing and newly-generated instructions. In addition to being sizeable, we design our instructions to cover a broad set of topics to ensure. A thorough investigation of our instruction data demonstrate their *diversity*, and we generate responses for these instructions using `gpt-3.5-turbo`. We then exploit the instructions to tune a host of models, dubbed LaMini-LM, of varying sizes, both from the *encoder-decoder* as well as the *decoder-only* families. We evaluate our models both automatically (on 15 different NLP benchmarks) and manually. Results show that our proposed LaMini-LM are on par with competitive baselines while being nearly ×10 smaller in size.[1]

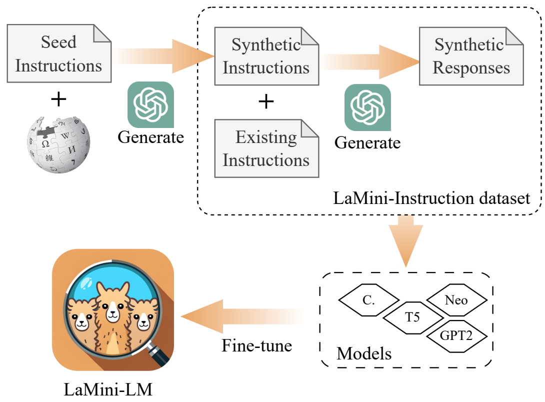


Figure 1: Overview of LaMini-LM

1 Introduction

Large language models (LLMs) with instruction tuning are capable of generating remarkable outputs for a wide range of use cases (Ouyang et al., 2022; Wei et al., 2022; Sanh et al., 2022; Chung et al., 2022; OpenAI, 2023). However, these models usually have billions of parameters, which require massive computational resources for both training and inference (Brown et al., 2020; Thoppilan et al., 2022; Hoffmann et al., 2022; Chowdhery et al., 2022). Kaplan et al. (2020) suggest that the performance of LLMs scales proportionally with model and dataset size. Consequently, scaling the models raises many issues such as those related to the energy footprint (Strubell et al., 2019). Moreover, the accessibility of large models is a real concern for many NLP practitioners due to limited access to computing resources (Nityasya et al., 2020).

In this work, we present LaMini-LM, a collection of language models that are notably smaller in size than most existing instruction-tuned models. We develop LaMini-LM models by employing sequence distillation (also known as offline distillation) (Kim and Rush, 2016) from LLMs. Although similar attempts have been made in recent work (e.g., (Taori et al., 2023; Chiang et al., 2023; Anand et al., 2023)), there are several gaps in this literature that we aim to address. Specifically, these works often (i) provide a small-scale distilled dataset (ii) that is not necessarily diverse, and a (iii) limited number of models (typically only one), (iv) without comprehensive evaluation nor analysis of the models' performance. Furthermore, many of the distilled models resulting from prior work tend to still be relatively computationally intensive. That is, parameters of these recent models usually range from 7B to 13B, making them difficult to deploy in resource-constrained settings especially for under-resourced institutions.

To alleviate these issues, we firstly generate a large-scale offline distillation dataset comprising

---

[*] work done while visiting MBZUAI

[1]Our code, model checkpoints, and dataset are available at https://github.com/mbzuai-nlp/LaMini-LM

2.58M instructions, and then fine-tune a collection of language models to obtain the LaMini-LM models, as shown in Figure 1. We collate instructions from various prior datasets such as `self-instruct` (Wang et al., 2022a), `P3` (Sanh et al., 2022), `FLAN` (Longpre et al., 2023) and `Alpaca` (Taori et al., 2023). Additionally, we use ChatGPT (`gpt-3.5-turbo`) to generate supplementary instructions, with an emphasis on diversity that adheres to the existing human-written instructions in the prompt. This approach is known as *Example-Guided Instruction Generation*. To further increase the diversity in the generated text, we also introduce the *Topic-Guided Instruction Generation* method. Subsequently, we use `gpt-3.5-turbo` to generate responses for each instruction.

After generating the dataset, we fine-tune several smaller language models with varying sizes (from 61M to 1.5B) and architectures (encoder-decoder and decoder-only). Furthermore, we compare different variations of models with the same architecture. Our work is also distinguished from previous research by providing a comprehensive evaluation of the resulting models. We assess the performance of the models on various NLP downstream tasks, in addition to manual human evaluation of the model's outputs. This analysis offers a more in-depth understanding of the models' strengths and weaknesses.

Our contributions can be summarized as follows:

1. We release a large-scale instruction dataset that contains over 2.58M examples. To the best of our knowledge, this dataset is the largest instruction dataset currently available in the NLP literature. Our instruction dataset is $\times 50$ larger than the one released by Taori et al. (2023).
2. We explore the process of distilling knowledge from LLMs to various much smaller model architectures, resulting in a family of distilled language models. Our largest model and smallest model are $\times 110$ and $\times 2800$ smaller than GPT-3 (Brown et al., 2020), respectively.
3. We conduct extensive experiments on both our proposed models and several publicly available LLMs. These experiments include automatic evaluation on 15 NLP tasks and human evaluation. Both our automatic and human evaluations show that our proposed models achieve comparable performance with Alpaca (Taori et al., 2023) while being nearly $\times 10$ smaller in size.

2 Related Work

2.1 Instruction Tuning

Instruction tuning is a emerging paradigm in NLP. This approach leverages natural language instructions along with language models to induce zero-shot performance on tasks that have not been seen before. There is a line of research, demonstrating that vanilla language models can effectively follow general language instructions when fine-tuned with the human-written instructions (Weller et al., 2020; Mishra et al., 2022; Wang et al., 2022b; Wei et al., 2022; Sanh et al., 2022; Ouyang et al., 2022; Parmar et al., 2022; Scialom et al., 2022; Chung et al., 2022; Yin et al., 2022; Gupta et al., 2022; Muennighoff et al., 2022). Recently, Wang et al. (2022a) show that the model-generated instructions can be used for instruction tuning and effectively improve the vanilla language models' capability on responding the instructions. On top of this work, there is a number of works that instruction-tune the vanilla language models on the model-generated instructions (Taori et al., 2023; Chiang et al., 2023; Anand et al., 2023; Chen et al., 2023).

In this work, we firstly generate the largest instruction dataset to date generated by `gpt-3.5-turbo`, and then fine-tune a collection of language models to obtain our LaMini-LM models.

2.2 Knowledge Distillation

Knowledge distillation is a process used to train a smaller model, referred to as the student, by learning from a larger model, known as the teacher (Hinton et al., 2015). One of the most commonly used methods of knowledge distillation involves training the student with an additional objective of matching the teacher's representation, such as logits, output probability, or intermediate activation (Sanh et al., 2019; Jiao et al., 2020; Mirzadeh et al., 2020; Wang et al., 2020; Zhao et al., 2022).

For sequence-to-sequence or generative models, Kim and Rush (2016) introduced the concept of sequence-level distillation. This approach involves generating a synthetic output by running inference with the teacher model, which is then used to train the student model. This method is more efficient as it only requires running the usually large teacher

```
<example>What are some things you can do to de-stress?</example>
<example>How can individuals and organizations reduce unconscious bias?</example>
<example>Write a program to compute the sum of integers from k to n.</example>

Generate 20 diverse examples that are similar to the provided examples.
You do not need to provide a response to the generated examples.
Each example must include an instruction.
Each generated instruction can be either an imperative sentence or a question.
Each example must start with the label "<example>" and end with the label "</example>".
```

Figure 2: An example of instruction generation prompt based on three random examples from `self-instruct`.

model once. Previous research has demonstrated the effectiveness of sequence-level distillation. For instance, Costa-jussà et al. (2022) used sequence-level distillation to reduce the size of an NLLB machine translation system to 600M parameters. Similarly, by combining sequence-level distillation with model pruning and quantization, Behnke et al. (2021); Bogoychev et al. (2020) managed to train a translation system that was approximately $\times 25$ smaller than the teacher model without a significant decrease in BLEU score.

In our work, we train our model on the output of `gpt-3.5-turbo`, which can be viewed as a sequence-level distillation approach. While other researchers also train language models based on the output of GPT models, our work is distinct in that we train our model on a considerably larger dataset and distilled it into much smaller models. Furthermore, we provide various student models.

3 Dataset Generation

We distill the knowledge from large language models by performing sentence/offline distillation (Kim and Rush, 2016). In this approach, the student model learns from the teacher's outputs. To create our dataset, we utilize several existing resources of prompts, including `self-instruct` (Wang et al., 2022a) and `Alpaca` (Taori et al., 2023) as well as random subsets of `P3` (Sanh et al., 2022) and `FLAN` (Longpre et al., 2023). Through this process, we generate a total of 2.58M pairs of instructions and responses using ChatGPT (`gpt-3.5-turbo`). We also perform an exploratory analysis of the resulting text.

3.1 Instruction Generation

We introduce two strategies to generate instructions in this section: the example-guided and the topic-guided. Moreover, we detail our approach to generating responses.

Example-Guided Instruction Generation Inspired by the works of Wang et al. (2022a) and Taori et al. (2023), we design a prompt, accompanied by a few examples and constraints, to generate instructions. We include only three random examples and a limited number of constraints in each prompt, as shown in Figure 2. Instead of providing explicit restrictions on language, output length, and instruction types, we only instruct `gpt-3.5-turbo` to generate a range of examples that conform to the given examples and adhere to the desired output format.

For `self-instruct`, we randomly sample three seed tasks from `self-instruct` and generate 20 instructions at once to reduce the generation cost and denote the generated instructions as $\widehat{\boldsymbol{X}}_{\mathrm{SI}}$. If the selected instructions are associated with the inputs, they are concatenated using a colon ":" symbol to form the format "`$instruction:$input`". For `P3` and `FLAN`, we sample three random examples from the same subset, as we observe that if the sampled examples are from different subsets, `gpt-3.5-turbo` tends to generate more instructions based on the simplest example and often fails to match the required output format. Instructions based on `P3` and `FLAN` are longer. We only generate 10 instructions at once for `P3` and `FLAN` to avoid exceeding the output length limit. We denote the original set of prompts from `P3` and `FLAN` as $\boldsymbol{X}_{\mathrm{P3}}$ and $\boldsymbol{X}_{\mathrm{FLAN}}$ respectively and then denote the generated instructions as $\widehat{\boldsymbol{X}}_{\mathrm{P3}}$ and $\widehat{\boldsymbol{X}}_{\mathrm{FLAN}}$ respectively. Furthermore, we denote the prompts from `Alpaca` as $\widehat{\boldsymbol{X}}_{\mathrm{A}}$, but they are not used in this stage because they are generated based on `self-instruct` seed tasks like $\widehat{\boldsymbol{X}}_{\mathrm{SI}}$.

Topic-Guided Instruction Generation It is of concern that `gpt-3.5-turbo` may not possess the ability to generate diverse text without explicit guidance. To address this concern, we collect several common topics from Wikipedia to guide the gen-

Dataset	# of samples	# of ins. tokens	avg. ins. len.	# of res. tokens	avg. res. len.
$\widehat{\boldsymbol{D}}_{\text{SI}}$	0.27M	3.82M	14.27	17.64M	65.90
$\widehat{\boldsymbol{D}}_{\text{t,SI}}$	0.28M	3.75M	13.26	17.61M	62.38
$\widehat{\boldsymbol{D}}_{\text{P3}}$	0.30M	14.63M	49.22	6.35M	21.34
$\widehat{\boldsymbol{D}}_{\text{FLAN}}$	0.29M	10.69M	36.37	8.62M	29.33
$\widehat{\boldsymbol{D}}_{\text{A}}$	0.05M	0.89M	17.11	2.84M	54.72
$\boldsymbol{D}_{\text{P3}}$	0.46M	39.37M	84.78	9.84M	21.19
$\boldsymbol{D}_{\text{FLAN}}$	0.93M	57.45M	61.91	21.88M	23.58
$\boldsymbol{D}_{\text{ALL}}$	2.58M	130.60M	50.62	84.78M	32.86

Table 1: Data statistics of the generated dataset. The average instruction length and average response length are measured in tokens.

eration process. We first collect a total of 2.2M categories from Wikipedia. These categories are filtered based on two requirements. Firstly, the category must consist of less than three words. Secondly, the category must comprise more than 10 sub-categories and 50 pages. Upon manual inspection, we note that lengthy category titles are more likely to be associated with specific and niche information, while a common category can be divided into several sub-categories and discussed across multiple pages. For instance, the category "machine learning" contains 35 sub-categories and 200 pages.[2] After filtering, we obtain a list of $3.5K$ categories that serve as common topics. An example of the prompt with topics is presented in Appendix A.

In this study, we generate topic-guided instructions solely from `self-instruct` seed tasks, represented as $\widehat{\boldsymbol{X}}_{\text{t,SI}}$. This decision is based on our observation that `gpt-3.5-turbo` frequently struggles to produce the appropriate context for instructions. Conversely, examples from `P3` and `FLAN` typically contain extensive contextual information. Therefore, to maintain generation quality, we limit our topic-guided instruction generation to $\widehat{\boldsymbol{X}}_{\text{t,SI}}$.

3.2 Response Generation

To perform sequence-level distillation, we generate responses from the instructions described in the previous section. We generate the responses for all the generated instructions, including $\widehat{\boldsymbol{X}}_{\text{SI}}$, $\widehat{\boldsymbol{X}}_{\text{t,SI}}$, $\widehat{\boldsymbol{X}}_{\text{P3}}$, $\widehat{\boldsymbol{X}}_{\text{FLAN}}$. As we observe that `gpt-3.5-turbo` is less capable of providing the necessary context for the instructions, we also directly generate responses for the collected instructions, including $\widehat{\boldsymbol{X}}_{\text{A}}$, $\boldsymbol{X}_{\text{P3}}$ and $\boldsymbol{X}_{\text{FLAN}}$. Hence, we denote the resulting pairs as $\widehat{\boldsymbol{D}}_{\text{SI}} = \{\widehat{\boldsymbol{X}}_{\text{SI}}, \widehat{\boldsymbol{Y}}_{\text{SI}}\}$, $\widehat{\boldsymbol{D}}_{\text{t,SI}} = \{\widehat{\boldsymbol{X}}_{\text{t,SI}}, \widehat{\boldsymbol{Y}}_{\text{t,SI}}\}$, $\widehat{\boldsymbol{D}}_{\text{P3}} = \{\widehat{\boldsymbol{X}}_{\text{P3}}, \widehat{\boldsymbol{Y}}_{\text{P3}}\}$, $\widehat{\boldsymbol{D}}_{\text{FLAN}} = \{\widehat{\boldsymbol{X}}_{\text{FLAN}}, \widehat{\boldsymbol{Y}}_{\text{FLAN}}\}$, $\widehat{\boldsymbol{D}}_{\text{A}} = \{\widehat{\boldsymbol{X}}_{\text{A}}, \widehat{\boldsymbol{Y}}_{\text{A}}\}$, $\boldsymbol{D}_{\text{P3}} = \{\boldsymbol{X}_{\text{P3}}, \boldsymbol{Y}_{\text{P3}}\}$ and $\boldsymbol{D}_{\text{FLAN}} = \{\boldsymbol{X}_{\text{FLAN}}, \boldsymbol{Y}_{\text{FLAN}}\}$.[3] The complete dataset $\boldsymbol{D}_{\text{ALL}}$ is the union of all the aforementioned instruction-response pairs.

3.3 Exploratory Data Analysis

In this section, we conduct an exploratory analysis of the generated text. Our analysis focuses on several aspects of the dataset, including basic statistics, diversity, and human evaluation.

3.3.1 Statistics

We present the dataset statistics in Table 1. As we claimed before, `gpt-3.5-turbo` often fails to provide the necessary context for the generated instruction, given that the average length of $\widehat{\boldsymbol{X}}_{\text{P3}}$ and $\widehat{\boldsymbol{X}}_{\text{FLAN}}$ is significantly short than that of $\boldsymbol{X}_{\text{P3}}$ and $\boldsymbol{X}_{\text{FLAN}}$. Another observation is that if the instructions are generated from the same source, such as `self-instruct`, the corresponding responses have a similar length.

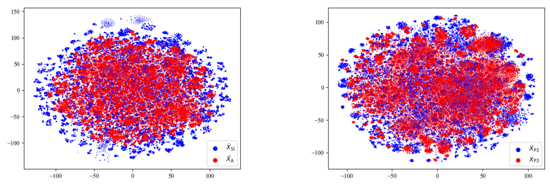
3.3.2 Diversity

Semantic Diversity To explore the semantic diversity of the generated instructions, we sample 50K instructions from $\widehat{\boldsymbol{X}}_{\text{SI}}$, $\widehat{\boldsymbol{X}}_{\text{A}}$, $\widehat{\boldsymbol{X}}_{\text{P3}}$ and $\boldsymbol{X}_{\text{P3}}$, compute their sentence embeddings using Sentence Transformer (Reimers and Gurevych, 2019),[4] and visualize the t-SNE of instruction sentence embeddings in Figure 3. We omit the comparison between $\widehat{\boldsymbol{X}}_{\text{FLAN}}$ and $\boldsymbol{X}_{\text{FLAN}}$ as it yields the same results as the comparison between $\widehat{\boldsymbol{X}}_{\text{P3}}$ and $\boldsymbol{X}_{\text{P3}}$. We observe that $\widehat{\boldsymbol{X}}_{\text{SI}}$ exhibits greater diversity than $\widehat{\boldsymbol{X}}_{\text{A}}$ as shown in Figure 3a and $\widehat{\boldsymbol{X}}_{\text{P3}}$ is slightly more diverse than $\boldsymbol{X}_{\text{P3}}$ as shown in Figure 3b. It appears

[2] https://en.wikipedia.org/wiki/Category:Machine_learning

[3] We denote the model-generated text as $\widehat{\boldsymbol{X}}_{\{\cdot\}}$ or $\widehat{\boldsymbol{Y}}_{\{\cdot\}}$ and the human-written text as $\boldsymbol{X}_{\{\cdot\}}$ or $\boldsymbol{Y}_{\{\cdot\}}$, except for $\boldsymbol{Y}_{\text{P3}}$ and $\boldsymbol{Y}_{\text{FLAN}}$ that are also generated by `gpt-3.5-turbo`.

[4] Model signature: `all-mpnet-base-v2`.



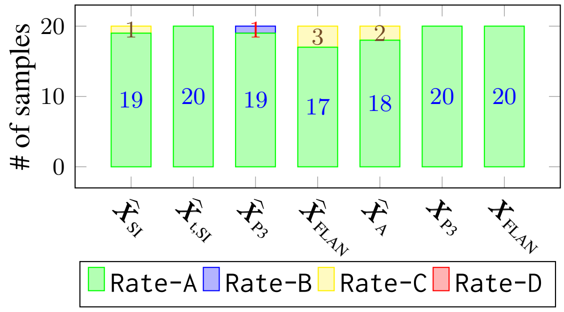
(a) The t-SNE visualization of the sentence embeddings of $\widehat{\boldsymbol{X}}_{\text{SI}}$(ours) and $\widehat{\boldsymbol{X}}_{\text{A}}$.

(b) The t-SNE visualization of the sentence embeddings of $\widehat{\boldsymbol{X}}_{\text{P3}}$(ours) and $\boldsymbol{X}_{\text{P3}}$.

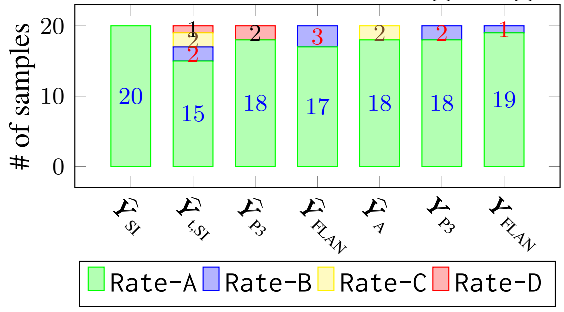
Figure 3: The t-SNE visualizations of instruction sentence embeddings.

Dataset	$\boldsymbol{X}_{\{.\}}$ or $\widehat{\boldsymbol{X}}_{\{.\}}$	$\boldsymbol{Y}_{\{.\}}$ or $\widehat{\boldsymbol{Y}}_{\{.\}}$
$\widehat{\boldsymbol{D}}_{\text{SI}}$	72.46	74.36
$\widehat{\boldsymbol{D}}_{\text{t,SI}}$	73.40	76.70
$\widehat{\boldsymbol{D}}_{\text{P3}}$	75.31	74.76
$\widehat{\boldsymbol{D}}_{\text{FLAN}}$	73.40	75.80
$\widehat{\boldsymbol{D}}_{\text{A}}$	77.00	76.20
$\boldsymbol{D}_{\text{P3}}$	77.03	74.45
$\boldsymbol{D}_{\text{FLAN}}$	76.63	76.11
$\boldsymbol{D}_{\text{ALL}}$	78.59	77.59

Table 2: MATTR (up-scaled by $\times 100$) of the generated dataset.



(a) Human evaluation for the instruction ($\boldsymbol{X}_{\{.\}}$ or $\widehat{\boldsymbol{X}}_{\{.\}}$).



(b) Human evaluation for the responses ($\boldsymbol{Y}_{\{.\}}$ or $\widehat{\boldsymbol{Y}}_{\{.\}}$).

Figure 4: Human evaluation results for the generated instruction dataset.

that this observation can be attributed to the enhanced generative capabilities of `gpt-3.5-turbo`.

Lexical Diversity We use Moving-Average Type–Token Ratio (MATTR) (Covington and McFall, 2010) to measure the lexical diversity with the window size of 50, because each subset of $\boldsymbol{D}_{\text{ALL}}$ varies in size and MATTR is free from the impact of text length. As shown in Table 2, the model-generated instructions $\widehat{\boldsymbol{X}}_{\{.\}}$ given by `gpt-3.5-turbo` are not as diverse as the human-written instructions $\boldsymbol{X}_{\{.\}}$ and $\widehat{\boldsymbol{X}}_{\text{A}}$ generated by `text-davinci-003`. It is noteworthy that $\widehat{\boldsymbol{X}}_{\text{t,SI}}$ is more diverse than $\widehat{\boldsymbol{X}}_{\text{SI}}$ and $\widehat{\boldsymbol{Y}}_{\text{t,SI}}$ is the most diverse subset of responses, which demonstrates the effectiveness of the topic-guidance. Furthermore, $\boldsymbol{D}_{\text{ALL}}$ illustrates the greatest lexical diversity, compared with all the subsets.

3.3.3 Human Evaluation

We follow the human evaluation protocol given by Wang et al. (2022a), which categorizes the quality of the generated text into four levels:

- `Rate-A`: The generated text is of high quality;
- `Rate-B`: The generated text is acceptable but has minor errors;
- `Rate-C`: The generated text has significant errors in content.
- `Rate-D`: The generated text is completely unacceptable.

More details about the human evaluation protocol are presented in Appendix C.

We randomly sample 20 examples from each subset of $\boldsymbol{D}_{\text{ALL}}$ and one of the co-authors scores the generated text. In general, both the generated instructions and the generated responses are of high quality as shown in Figure 4. During the annotation process, we observe that examples from $\widehat{\boldsymbol{X}}_{\text{P3}}$ and $\widehat{\boldsymbol{X}}_{\text{FLAN}}$ are much shorter than those from $\boldsymbol{X}_{\text{P3}}$ and $\boldsymbol{X}_{\text{FLAN}}$ and their associated context are significantly shorter and easier, which confirms our observation in Table 1. Another noteworthy observation is that `gpt-3.5-turbo` is even more prone to generated the responses with factual errors when we provide the topics.

4 Experiment

4.1 Training LaMini-LM

We present LaMini-LM, a family of language models instruction-tuned on our 2.58M instructions dataset $\boldsymbol{D}_{\text{ALL}}$. We train two types of models, encoder-decoder and decoder-only, for architectural comparison. The size for both categories of models ranges from 61M to 1.5B to facilitate size

Name	Architecture	Initialization
LaMini-T5-61M	enc-dec	T5-small
LaMini-T5-223M	enc-dec	T5-base
LaMini-T5-738M	enc-dec	T5-large
LaMini-Flan-T5-77M†	enc-dec	Flan-T5-small
LaMini-Flan-T5-248M†	enc-dec	Flan-T5-base
LaMini-Flan-T5-783M†	enc-dec	Flan-T5-large
LaMini-Neo-125M	dec-only	GPT-Neo-125M
LaMini-Neo-1.3B	dec-only	GPT-Neo-1.3B
LaMini-Cerebras-111M	dec-only	C-GPT-111M
LaMini-Cerebras-256M	dec-only	C-GPT-256M
LaMini-Cerebras-590M	dec-only	C-GPT-590M
LaMini-Cerebras-1.3B	dec-only	C-GPT-1.3B
LaMini-GPT-124M†	dec-only	GPT-2
LaMini-GPT-774M†	dec-only	GPT-2 large
LaMini-GPT-1.5B†	dec-only	GPT-2 xl

Table 3: LaMini-LM collection. Models with † are those with the best overall performance given their size/architecture, hence we recommend using them. C-GPT indicates Cerebras-GPT.

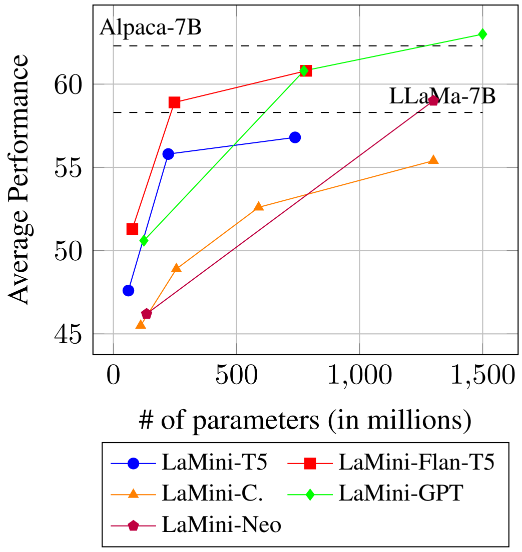


Figure 5: The performance comparison between encoder-decoder models and decoder-only models of LaMini-LM on the downstream NLP tasks. The horizontal dash lines indicate the average performance given by Alpaca-7B and LLaMa-7B.

comparison. The underlying models for initialization are from five sources, including T5 (Raffel et al., 2020), Flan-T5 (Chung et al., 2022), Cereberas-GPT (Dey et al., 2023), GPT-2 (Radford et al., 2019), and GPT-Neo (Gao et al., 2021a). The details of our LaMini-LM series are summarized in Table 3.

Experimental Setup We finetune all models over 5 epochs and a batch size of 1024. For our encoder-decoder models, we use a learning rate of 5×10^{-4} following Chung et al. (2022). For our decoder-only models, we follow the same configuration as Alpaca (Taori et al., 2023) including the learning rate of 2×10^{-5}. We use HuggingFace's transformers for training. Moreover, we use the same prompt wrapper as Alpaca (Taori et al., 2023), hence we also wrap our instruction similarly during inference. We perform all of our experiments on 8×V100 (32G) and 8×A100 (40G) GPUs. Our models are publicly available.

4.2 Model Evaluation

We then evaluate the performance based on several downstream NLP tasks as well as human evaluation on user-oriented instruction.

Automatic Evaluation on Downstream NLP Tasks We conduct a zero-shot evaluation on the downstream NLP tasks for our LaMini-LM. We use language model evaluation harness (Gao et al., 2021b) to evaluate our instruction-tuned models.[5] We select 15 diverse NLP tasks, covering QA, sentiment analysis, paraphrase identification, natural language inference, coreference resolution, word sense disambiguation, and sentence completion. The details for these NLP tasks can be found in Appendix D.

Human Evaluation on User-Oriented Instructions The NLP tasks in Appendix D are designed for academic-oriented tasks, and are focused on classification. To complete the evaluation, we additionally evaluate the practicality of both our LaMini-LM and our baseline models by utilizing the user-oriented instructions from Wang et al. (2022a), which consists of 252 instructions covering 71 commonly used apps use-cases. In contrast with downstream NLP tasks, there is no single gold answer for many of these questions, therefore manual human evaluation is needed to benchmark the performance. We follow the guideline as in Appendix C for measuring the model's response quality.

To reduce the annotation cost yet ensure the instruction diversity, we keep no more than 2 instructions for each app and manually filter out those instructions that are already covered in downstream NLP tasks, such as natural language inference, sen-

[5] https://github.com/EleutherAI/lm-evaluation-harness

	T5	LaMini-T5	F-T5	LaMini-F-T5	C-GPT	LaMini-C	GPT-2	LaMini-GPT	LLaMa	Alpaca
#params.	738M		783M		1.3B		1.5B		7B	
OpenBookQA	32.8	36.0	31.2	34.0	29.0	34.0	32.0	39.8	42.4	43.2
SciQ	82.4	84.5	93.8	86.7	73.0	79.4	76.1	80.4	66.3	69.6
RACE	31.5	32.6	40.9	32.8	30.3	32.9	33.1	39.1	39.9	42.2
ARC	25.4	29.0	30.7	31.8	25.3	30.3	28.5	35.8	41.4	41.8
PIQA	55.9	67.2	72.2	70.6	66.8	66.9	70.5	71.3	77.5	76.0
ReCoRD	73.1	68.7	76.7	70.4	75.0	66.3	84.4	78.5	91.4	87.4
SST	50.2	90.3	94.0	93.1	51.3	90.3	49.1	93.5	53.0	85.8
MRPC	34.3	71.1	82.6	77.9	68.4	71.3	63.2	76.0	68.4	74.3
RTE	79.8	57.0	87.4	65.0	53.1	65.7	52.3	67.9	53.4	67.1
MultiNLI	61.3	54.7	72.4	61.4	35.2	47.4	36.5	67.5	34.4	38.8
MultiNLI (mis)	63.1	55.8	72.0	61.0	35.4	49.2	37.0	69.3	35.6	39.6
WSC	60.4	59.0	66.7	64.1	62.3	57.1	73.3	69.6	80.6	77.3
WinoGrande	55.2	54.9	59.9	56.0	51.9	51.8	58.3	56.0	67.0	65.7
WiC	49.4	50.5	64.7	63.8	50.2	50.2	49.8	52.4	50.0	57.5
HellaSwag	38.9	40.6	48.7	43.7	38.4	38.7	50.9	48.3	73.0	68.7
Average	52.9	**56.8**	**66.3**	60.8	49.7	**55.4**	53.0	**63.0**	58.3	**62.3**

Table 4: Automatic evaluation results of selected language models on 15 NLP tasks. "Average" indicates the micro-average of the individual task results. The best average results are highlighted in **bold**. F-T5 and LaMini-F-T5 indicate Flan-T5 and LaMini-Flan-T5 respectively. C-GPT and LaMini-C indicate Cerebras-GPT and LaMini-Cerebras respectively.

timent analysis, and summarization. Finally, we obtain a test set for human evaluation with 114 instructions. we organize a team of 8 human experts for human evaluation, with each expert responsible for evaluating the responses to 15 instructions across all chosen models. Arguably, human annotation is subjective. Thus, to ensure consistency, all model responses from the same instruction are scored by the same annotator, as the scores for that particular instruction is based on the same standard.

5 Result and Discussions

In this section, we provide evaluation result and discussion of LaMini-LM for both the downstream NLP tasks and human evaluation on user-oriented instruction. For NLP downstream task, larger models yield better average performance, as seen in Figure 5. Therefore to save space, we present the broken-down results given by the largest models in each group (Table 4). We also compare their performance with LLaMa-7B (Touvron et al., 2023) and Alpaca-7B (Taori et al., 2023). Surprisingly, we also observe that the instruction-tuned models, including ours and Alpaca, always underperform their baselines on the ReCoRD benchmark. We leave the further investigation of this observation to future work. Breakdown results of other models can be found in Appendix E.

We present the human evaluation results in Figure 6. Similar to downstream NLP performance, larger models generally perform better. Interestingly, encoder-decoder models from T5 are performing exceptionally well, given their rather small size.

Encoder-Decoder vs. Decoder-Only The encoder-decoder LaMini language models (LaMini-T5 series and LaMini-Flan-T5 series) outperform the decoder-only LaMini language models (LaMini-GPT series) when the number of parameters is limited (less than 500M parameters). LaMini-Flan-T5-248M even outperforms LLaMa-7B on downstream NLP tasks. When the model size is higher, LaMini-Flan-T5 is comparable to LaMini-GPT. Yet, both LaMini-Flan-T5 and LaMini-T5 demonstrate strong human evaluation results for user-oriented instructions, despite their relatively small size. Especially, T5-based models of 200M parameters is competitive against LaMini-GPT-1.5B for human evaluation result. We recommend further exploration of the encoder-decoder architecture for language models, given their potential, as demonstrated in our experiments.

GPT-2 vs. Cerebras-GPT Among all the decoder-only models that we fine-tune, we observe a performance discrepancy among models that are of comparable size. Based on the results in Table 4 and Figure 5, LaMini-GPT series is significantly superior on downstream NLP tasks compared to LaMini-Cerebras, despite both having similar architecture and size. Even more, LaMini-GPT of

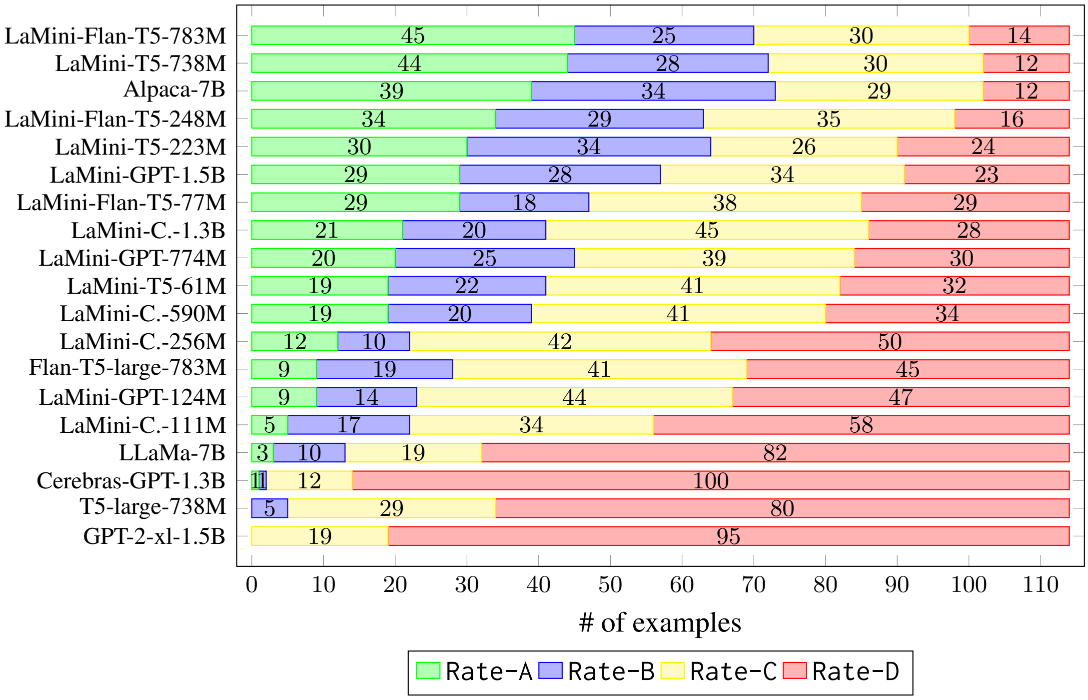


Figure 6: Human evaluation results of the selected models on our 114 user-oriented instructions.

774M is outperforms LaMini-Cerebras-1.3B despite being half in size. We also observe similar results on human evaluation.

Generally, vanilla GPT-2 also outperforms Cerebras-GPT models of comparable size on downstream tasks, as shown in Appendix E. Therefore, there may be a correlation between the initial model's performance and the performance achieved after instruction tuning.

T5 vs. Flan-T5 LaMini-Flan-T5 series exhibit better performance than LaMini-T5 in the downstream NLP tasks. Considering the original Flan-T5 is already instruct-tuned, this result is not surprising. But, the original Flan-T5 perform poorly under general user-oriented instruction, hence improvement on that aspect is possible by further fine-tuning the model with a diverse set of instruction. After fine-tuning with our dataset, both LaMini-Flan-T5 and LaMini-T5 achieve comparable performance according to human evaluation.

Qualitative Analysis We present a comparison of model responses based on user-oriented human evaluation in Table 5 and Table 6. Our analysis reveals that the responses generated by LaMini-LM tend to be shorter in length when compared to those generated by the Alpaca-7B model. This phenomenon can be attributed to the fact that we have imposed a constraint on the `gpt-3.5-turbo` model to ensure that its responses are as concise as possible during the generation process described in Section 3.2. As shown in Table 5, LaMini-LM correctly respond to the instruction and generate coherent responses with minor errors, while Alpaca fails to respond the instruction. However, LaMini-LM hallucinate when responding the instruction, while Alpaca generates the response with accurate information. From both examples, we conclude that current language models are still prone to generate hallucinated and nonfactual information. We present more discussions on the limitations of LaMini-LM in Section 6.

6 Limitations

In this paper, we explore instruction tuning on various small-size language models and performe evaluation across multiple benchmarks. However, our work still has some limitations:

- **Model Variations**: Compared with prior work on which they only provide 1 model and sometimes without any evaluation, our work excels at providing analysis across different models with different configurations. However, our model set is still fairly limited. We only used T5, GPT-2, Cerebras GPT, and GPT-Neo as our base models.
- **Model Size**: Currently, we only explore models up to around 1B parameters in size. It

Model	Response	Rate
Alpaca-7B	Key Points to Remember About History Of The United States 1607- present	D
LaMini-Flan-T5-783M	History of the USA: - The United States as it was founded on July 4, 1776. It is known for its unique culture, economy and lifestyle, technological advancements, political system such as democracy, nationalism and civil rights movements leading up to these events in particular. - This early American history begins with European colonization by Native Americans who established their own independent nations around America after crossing this waterway over thousands of years ago.	B
LaMini-T5-738M	Important study notes for the history of the USA include: - The US was created in 1776, three decades after the invention of the printing press. - The United States became an independent nation on April 15, 1787. - In the early days of the Republic, the US played a role in colonization and commerce. - The American Revolution took place during the Revolutionary War, from 1775 to 1803, and it marked the beginning of the end of slavery and the end of democracy. - The Civil War is characterized by the draft, which led to the expansion of the military and the a	B

Table 5: Model responses to the instruction "`Include important study notes and key points that someone should know about the given subject: 'history of the USA'`", where Alpaca-7B fails but LaMini language models manage to respond. The high-quality contents are highlighted in blue. The errors are highlighted in red.

Model	Response	Rate
Alpaca-7B	Netflix's The Witcher tells the story of the monster slayer Geralt, played by Henry Cavill in this dark fantasy action-adventure TV show based on Andrzej Sapkowski's best selling novels and CD Projekt Red video games from 2007 to date	A
LaMini-Flan-T5-783M	"The Witcher" is a German crime drama film directed by Christopher Nolan, first released in 2019. The movie follows the story of Christian Fenker, who becomes obsessed with killing Frank Zappa and becoming the next prime suspect after being tricked into selling voodoo souls on black market exchanges. As he races against time to bring down these criminal masterminds and make peace with humanity, he begins to have personal struggles that threaten both his own existence as well as those around him.	C
LaMini-T5-738M	"The Witcher" is a 2019 film that follows the story of a former witch who is now a powerful witch and embarks on a perilous adventure through a magical world filled with dangerous creatures.	C

Table 6: Model responses to the instruction "`Write a short description about the given movie or series: "The Witcher (2019)"`", where LaMini language models fails but Alpaca-7B manages to respond. The high-quality contents are highlighted in blue. The errors are highlighted in red.

would be beneficial to additionally explore larger models to further see the performance trend and to better compare with prior work.

- **Single Turn Dialog**: While our training data, as well as our user-oriented evaluation, consist of "dialog-like" instruction, our models are not designed to handle multi-turn dialog.
- **Error Propagation**: Our models have been trained using distilled knowledge extracted from `gpt-3.5-turbo`, thus inheriting all the potential risks associated with it. As shown in Table 5 and Table 6, LaMini-LM models also exhibit the issue of hallucination. Additionally, during the human evaluation process, it was observed that LaMini-LM models demonstrate inadequate performance when tasked with coding, mathematical problems, and instructions requiring logical reasoning.

We leave these limitations to be addressed to the future work.

7 Ethical Consideration

We demonstrate that training small language models on large-scale instruction can significantly enhance their performance on downstream NLP tasks, as well as in human evaluation. These instruction-tuned models exhibit superior performance compared to significantly larger models and are particularly adept at engaging in open-ended conversation. Despite these advantages, it is important to acknowledge that these instruction-tuned models are not fully aligned with human objectives. They may frequently generate discriminatory responses and propagate biases or other forms of discrimination originating from the teacher model. Moreover, as we detail in Section 6, these models often generate false information, which may have unintended consequences.

To mitigate any potential harm arising from the use of these models, we intend to minimize the risks associated with their use in future research. We advocate for the responsible use of our models to prevent any harm.

8 Conclusion

In this work, we release a large-scale instruction dataset distilled from ChatGPT with more than 2.58M examples. To the best of our knowledge, this dataset is currently the largest dataset of its kind. We explore distilling knowledge from LLMs to various smaller and more efficient model architectures. We refer to the resulting family of language models as LaMini, which includes 6 encoder-decoder models and 9 decoder-only models with varying model sizes. We also conduct a comprehensive evaluation in this work, including the automatic evaluation of the downstream NLP tasks and human evaluation. Both evaluation strategies highlight that our proposed models achieve comparable performance with Alpaca (Taori et al., 2023) while is nearly $\times 10$ smaller in size. This work sheds light on distilling knowledge from LLMs to much smaller model architectures and demonstrates the potential of training efficient yet effective language models.

References

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. `https://github.com/nomic-ai/gpt4all`.

Maximiliana Behnke, Nikolay Bogoychev, Alham Fikri Aji, Kenneth Heafield, Graeme Nail, Qianqian Zhu, Svetlana Tchistiakova, Jelmer van der Linde, Pinzhen Chen, Sidharth Kashyap, and Roman Grundkiewicz. 2021. Efficient machine translation with model pruning and quantization. In *Proceedings of the Sixth Conference on Machine Translation*, pages 775–780, Online. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Nikolay Bogoychev, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2023. Phoenix: Democratizing chatgpt across languages. *arXiv preprint arXiv:2304.10453*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Michael A. Covington and Joe D. McFall. 2010. Cutting the gordian knot: The moving-average type-token ratio (MATTR). *J. Quant. Linguistics*, 17(2):94–100.

Nolan Dey, Gurpreet Gosal, Zhiming, Chen, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. 2023. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021a. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021b. A framework for few-shot language model evaluation.

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.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5191–5198. AAAI Press.

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.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning. *CoRR*, abs/2211.01786.

Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Radityo Eko Prasojo, and Alham Fikri Aji. 2020. No budget? don't flex! cost consideration when planning to adopt NLP for your business. *CoRR*, abs/2012.08958.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

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

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, 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.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. 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`.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *CoRR*, abs/2201.08239.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language model with self generated instructions. *CoRR*, abs/2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. ConTinTin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3062–3072, Dublin, Ireland. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11943–11952. IEEE.

A Prompt with Topics

For the prompts with topics, besides three random examples, we sample three random topics from the common topic list and present an example in Figure 7.

B Response Generation

The Python code used to generate the response can be found in Figure Figure 8. Before asking `gpt-3.5-turbo` to generate responses, we firstly send a message as the "system" that requires `gpt-3.5-turbo` to respond the instructions as concise as possible to avoid the overly lengthy responses.

C Human Evaluation Protocol

We present the human evaluation protocol as well as the corresponding example for each rating level in Table 7.

D Automatic Evaluation Datasets

We present the details of 15 downstream NLP tasks, including the number of test examples and the corresponding evaluation metrics, in Table 8.

E Automatic Evaluation Results

The breakdown results given by LaMini-T5, LaMini-Flan-T5, LaMini-Neo, LaMini-Cerebras and LaMini-GPT are presented in Table 9,Table 10,Table 11,Table 12 and Table 13 respectively.

```
<example>Try coming up with a creative way to stay motivated during a workout.</example>
<example>In your opinion, what are the qualities of an effective sports coach?</example>
<example>Return the SSN number for the person: "Yann LeCun"</example>

Generate 20 diverse examples that are similar to the provided examples with the topics "Design bureaus, Conidae, Infantry".
You do not need to provide a response to the generated examples.
Each example must include an instruction.
Each generated instruction can be either an imperative sentence or a question.
Each example must start with the label "<example>" and end with the label "</example>".".
```

Figure 7: An example of instruction generation prompt based on three random examples from `self-instruct` and three random topics.

```python
import openai
def send_request(instruction):
    response = openai.ChatCompletion.
    create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content"
    : "You are a helpful assistant, but
    you must respond the provided
    instructions as concise as possible.
    "},
            {"role": "user", "content":
    instruction}
        ]
    )
    return response
```

Figure 8: The Python code of sending request via OpenAI API to generate the response for an instruction.

<table>
<tr><th>Level</th><th>Criteria</th><th>Example</th></tr>
<tr><td>Rate-A</td><td>• Valid, acceptable and satisfying (subject to the annotator) response;
• Accurate in terms of facts, yet comparable to human standards;
• The response meets the required criteria, but it may not be in the expected format.</td><td>• Instruction: Make a list of the pros and cons of the given decision: "Implementing a Remote Working Policy"
• Response: Pros of implementing a remote working policy include improved productivity, reduced commuting time to work remotely, access to healthcare services for patients with chronic illnesses or injuries from their employment activities, potential cost savings by avoiding long commutes between workers instead of relying solely on trains, and better employee job security since individuals are constantly recharging into each other's networks. However, it may still face challenges in terms of accessibility due to restrictions imposed upon those living within certain walking distance limits and areas without reliable transportation infrastructure such as high density housing units or public transit systems.</td></tr>
<tr><td>Rate-B</td><td>• The response is acceptable but has minor errors that can be improved;
• Mirror errors include out-of-context content, minimal factual errors, partially responding to the instruction, etc.</td><td>• Instruction: Make a list of the pros and cons of the given decision: "Implementing a Remote Working Policy"
• Response: Pros - 1) Increased flexibility in work schedule, allowing for better job satisfaction with fewer commuting obligations 2) Improved workplace culture by offering opportunities to connect from another location or country leading off site; this can lead to more diverse employees 3) Reduced turnover rate as companies save costs associated with office space expenses and maintain separate offices which may attract new talent looking for quality Workforces due to remote working</td></tr>
<tr><td>Rate-C</td><td>• The response is relevant and responds to the instruction, but it has significant errors in the content.</td><td>• Instruction: Make a list of the pros and cons of the given decision: "Implementing a Remote Working Policy"
• Response: Effective Remote Working Policy A Remote Working Policy could significantly reduce the level of work done as a result of a saht process, including remotely working, or a project being carried out on the client computer without the physical workplace equipment being available.</td></tr>
<tr><td>Rate-D</td><td>• Invalid and unacceptable response.</td><td>• Instruction: Make a list of the pros and cons of the given decision: "Implementing a Remote Working Policy"
• Response: Implementing a Remote Working Policy</td></tr>
</table>

Table 7: Human evaluation protocol with examples.

Task Category	Dataset	Size	Metric
Multiple-Choice QA	OpenBookQA (Mihaylov et al., 2018)	500	Acc_{norm}
	SciQ (Welbl et al., 2017)	1,000	Acc_{norm}
	RACE (Lai et al., 2017)	1,045	Acc
	ARC (Clark et al., 2018)	1,172	Acc_{norm}
	PIQA (Bisk et al., 2020)	1,838	Acc_{norm}
Extractive QA	ReCoRD (Zhang et al., 2018)	10,000	F_1
Sentiment Analysis	SST (Socher et al., 2013)	872	Acc
Paraphrase Identification	MRPC (Dolan and Brockett, 2005)	408	Acc
Natural Language Inference	RTE (Wang et al., 2019)	277	Acc
	MultiNLI (Williams et al., 2018)	9,815	Acc
	MultiNLI (mis) (Williams et al., 2018)	9,832	Acc
Coreference Resolution	WSC273 (Levesque et al., 2012)	273	Acc
	WinoGrande (Sakaguchi et al., 2020)	1,267	Acc
Word Sense disambiguation	WiC (Pilehvar and Camacho-Collados, 2019)	638	Acc
Sentence Completion	HellaSwag (Zellers et al., 2019)	10,042	Acc_{norm}

Table 8: Details of 15 downstream NLP tasks. Acc_{norm} indicates the output probability used for computing the accuracy is normalized by the target sequence length.

	T5	LaMini-T5	T5	LaMini-T5	T5	LaMini-T5
# of params.	61M		223M		738M	
OpenBookQA	30.2	31.8	34.8	32.0	32.8	36.0
SciQ	58.0	69.7	71.7	82.9	82.4	84.5
RACE	26.4	29.0	31.1	32.6	31.5	32.6
ARC	22.7	23.0	24.4	26.5	25.4	29.0
PIQA	55.3	59.0	55.7	64.0	55.9	67.2
ReCoRD	53.4	51.7	64.6	59.1	73.1	68.7
SST	71.0	76.8	57.3	91.2	50.2	90.3
MRPC	48.0	68.4	31.6	73.5	34.3	71.1
RTE	53.4	52.7	61.4	71.5	79.8	57.0
MultiNLI	35.4	36.3	56.7	54.7	61.3	54.7
MultiNLI (mis)	35.2	36.2	57.1	55.5	63.1	55.8
WSC273	50.9	52.7	53.8	54.2	60.4	59.0
WinoGrande	48.9	49.3	50.4	51.9	55.2	54.9
WiC	50.0	50.0	52.0	56.0	49.4	50.5
HellaSwag	26.8	27.9	31.0	32.0	38.9	40.6
Average	44.4	47.6	48.9	55.8	52.9	56.8

Table 9: Automatic evaluation results of LaMini-T5 language models and their baselines on 15 NLP tasks. “Average” indicates the micro-average of the individual task results.

	Flan-T5	LaMini-Flan-T5	Flan-T5	LaMini-Flan-T5	Flan-T5	LaMini-Flan-T5
# of params.	77M		248M		783M	
OpenBookQA	27.0	30.0	28.8	33.0	31.2	34.0
SciQ	89.0	79.4	93.0	86.2	93.8	86.7
RACE	29.7	28.9	35.9	34.4	40.9	32.8
ARC	22.3	24.0	25.1	27.3	30.7	31.8
PIQA	61.9	61.9	67.0	65.7	72.2	70.6
ReCoRD	57.7	53.8	68.2	61.3	76.7	70.4
SST	87.3	85.7	92.3	92.2	94.0	93.1
MRPC	63.2	58.6	71.3	74.8	82.6	77.9
RTE	60.3	56.3	78.7	66.1	87.4	65.0
MultiNLI	42.4	53.2	66.7	66.6	72.4	61.4
MultiNLI (mis)	42.5	53.2	66.9	66.8	72.0	61.0
WSC273	53.1	54.6	57.5	60.4	66.7	64.1
WinoGrande	50.0	50.1	54.2	53.0	59.9	56.0
WiC	51.3	50.8	52.7	60.8	64.7	63.8
HellaSwag	29.1	28.6	36.4	34.6	48.7	43.7
Average	51.1	51.3	59.7	58.9	66.3	60.8

Table 10: Automatic evaluation results of LaMini-Flan-T5 language models and their baselines on 15 NLP tasks. "Average" indicates the micro-average of the individual task results.

	GPT-Neo	LaMini-Neo	GPT-Neo	LaMini-Neo
# of params.	135M		1.3B	
OpenBookQA	26.2	31.6	33.6	36.4
SciQ	68.8	66.8	77.1	84.2
RACE	27.6	28.7	34.1	34.3
ARC	23.1	24.2	25.9	32.9
PIQA	62.5	63.5	71.1	71.7
ReCoRD	65.6	62.1	81.4	75.2
SST	53.9	52.2	65.7	91.2
MRPC	68.4	64.2	68.4	70.3
RTE	54.9	53.1	60.3	71.1
MultiNLI	35.5	31.9	35.8	49.3
MultiNLI (mis)	35.4	32.0	36.2	49.7
WSC273	55.3	52.7	75.1	66.7
WinoGrande	50.4	50.6	54.9	54.8
WiC	50.0	50.0	50.0	50.2
HellaSwag	30.4	29.9	48.9	47.5
Average	47.2	46.2	54.6	59.0

Table 11: Automatic evaluation results of LaMini-Neo language models and their baselines on 15 NLP tasks. "Average" indicates the micro-average of the individual task results.

	C-GPT	LaMini-C	C-GPT	C-GPT	C-GPT	LaMini-C	C-GPT	LaMini-C
# of params.	111M		256M		590M		1.3B	
OpenBookQA	29.6	30.8	25.4	30.6	28.0	33.0	29.0	34.0
SciQ	52.8	60.0	65.7	68.8	68.2	71.7	73.0	79.4
RACE	25.6	27.1	27.5	27.1	28.4	29.0	30.3	32.9
ARC	22.9	23.3	21.9	26.1	23.5	26.9	25.3	30.3
PIQA	58.4	60.3	61.4	61.4	62.8	63.2	66.8	66.9
ReCoRD	52.4	51.6	61.2	58.6	67.2	63.6	75.0	66.3
SST	60.1	61.2	49.8	76.9	56.0	85.8	51.3	90.3
MRPC	68.4	68.4	68.4	68.4	68.4	68.4	68.4	71.3
RTE	53.1	49.8	52.3	55.6	52.3	60.6	53.1	65.7
MultiNLI	35.1	34.4	35.2	39.0	35.0	49.0	35.2	47.4
MultiNLI (mis)	35.0	35.2	35.1	40.3	35.1	50.8	35.4	49.2
WSC273	51.3	54.2	54.6	49.5	61.9	54.2	62.3	57.1
WinoGrande	50.2	49.3	51.3	52.0	49.8	50.9	51.9	51.8
WiC	50.0	50.0	50.0	50.0	50.0	50.0	50.2	50.2
HellaSwag	26.4	27.2	28.6	29.3	32.3	32.3	38.4	38.7
Average	44.8	45.5	45.9	48.9	47.9	52.6	49.7	55.4

Table 12: Automatic evaluation results of LaMini-Cerebras language models and their baselines on 15 NLP tasks. "Average" indicates the micro-average of the individual task results. C-GPT and LaMini-C indicate Cerebras-GPT and LaMini-Cerebras respectively.

	GPT-2	LaMini-GPT	GPT-2	LaMini-GPT	GPT-2	LaMini-GPT
# of params.	124M		774M		1.5B	
OpenBookQA	28.2	30.4	31.2	37.0	32.0	39.8
SciQ	66.1	64.4	69.4	78.3	76.1	80.4
RACE	28.7	31.8	31.6	37.6	33.1	39.1
ARC	23.3	26.4	25.1	30.6	28.5	35.8
PIQA	61.2	62.4	69.2	69.9	70.5	71.3
ReCoRD	70.7	66.8	81.9	77.5	84.4	78.5
SST	52.8	84.5	49.4	91.5	49.1	93.5
MRPC	67.6	68.4	65.2	70.6	63.2	76.0
RTE	54.2	55.2	52.7	74.4	52.3	67.9
MultiNLI	35.6	38.9	35.9	62.5	36.5	67.5
MultiNLI (mis)	35.1	40.2	36.0	65.6	37.0	69.3
WSC273	55.7	57.1	72.5	68.1	73.3	69.6
WinoGrande	51.5	51.9	55.3	54.7	58.3	56.0
WiC	50.0	50.0	49.7	50.0	49.8	52.4
HellaSwag	30.8	30.7	45.3	43.5	50.9	48.3
Average	47.4	50.6	51.4	60.8	53.0	63.0

Table 13: Automatic evaluation results of LaMini-GPT language models and their baselines on 15 NLP tasks. "Average" indicates the micro-average of the individual task results.