Generalizability of FLAN-T5 Model Using Composite Task Prompting

CSCI 5541: NLP Project Final Report
Amitabha Deb, Issac Seuer Blaine, Peter Oritz, Srijan Pal
Team - Iron Code Benders
University of Minnesota, Twin Cities

1 Introduction

Natural Language Processing can help create solutions that facilitate access to data for all people. But it can be often intimidating for many people because it is hidden behind a wall of technical jargon. This requires some amount of technical savvy in order to use many of the methods that could be beneficial to people of all industries and careers. By having a model that is generalized for a variety of unseen tasks, a service could be created that would be applicable to many fields.

Instead of having to train a new model every time you need to ship out a solution for a specific industry, you could give the generalized model this new situation. This would remove unnecessary work and save a user from adapting to a new model every time.

There has been a recent rise in the value of the generalizability of language models with the introduction of ChatGPT by OpenAI. We have witnessed the power of generative models but on the other hand, we have limited information on how such models work.

This project aims to contribute to the field of natural language processing by evaluating the effectiveness of a widely used model "google/flan-t5" and exploring its potential for handling multiple tasks. We hope that the findings of this study will provide valuable insights and pave the way for further research in this area.

2 Literature Survey

Prior to Cross-Task Generalization via Natural Language Crowdsourcing Instructions[1], there was an absence of a large public benchmark dataset that could be used for task generalization purposes.

In the work[1], they have introduced NATURAL INSTRUCTIONS, a dataset of human-authored instructions curated from existing well-known datasets mapped to a unified schema, providing training and evaluation data for learning from instructions.

In the SUPER-NATURALINSTRUCTIONS[2] paper, a meta-dataset (i.e., a dataset of datasets; Triantafillou et al., 2019) was formed that consisted of a wide variety of NLP tasks. Each task has an instruction, input, and desired output mapped to it. Some of the tasks include classification, sequence tagging, question answering, etc.

Features	Value (s)
No of Tasks	1616
No of Task Types	76
No of Language	55
No of Domains	33
No of Instance	5M

Table 1: Features of SUP-NATINST

Also, the model being trained was able to perform new tasks based on given instructions. This proposed model, tk-INSTRUCT, outperformed InstructGPT[3] by a margin of 9 percent. It was based on the T5 model (Raffel et al., 2020). The T5 is basically a model trained on the multi-task dataset given their in-context instructions and was evaluated on unseen tasks, and was having 11B parameters compared to the 175B parameter InstrcutGPT[3].

In the paper, Training language models to follow instructions with human feedback[3], OpenAI has introduced IntructGPT. A set of labeler-written prompts and prompts submitted through the OpenAI API and a dataset of labeler demonstrations of the desired model behavior were used to fine-tune GPT-3 using supervised learning. Furthermore, a dataset of rankings of model outputs was used to fine-tune this supervised model using reinforcement learning from human feedback. The models were mainly evaluated by having the labelers rate the quality of model outputs on our test set, consisting of prompts from held-out customers. They also conducted automatic evaluations on a range of public NLP datasets. In human evaluations on the prompt distribution, outputs from the 1.3B parameter InstructGPT model were more preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters.

In the paper "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"[4], they have implemented transfer learning techniques for NLP by introducing a unified framework that converts all text-based language problems into a text-to-text format. A variety of tasks are cast as feeding the model text as input and training it to generate some target text. This allowed them to use the same model, loss function, hyperparameters, etc. across all the diverse sets of tasks. They have referred to their model and framework as the “Text-to-Text Transfer Transformer” hence the name “T5”.

In the paper "Scaling Instruction-Finetuned Language Models"[5], they show that finetuned language models have better performance and generalization to unseen tasks. he finetuning has been applied to various families of models such as T5[4], PaLM [6], and U-PaLM [7]. Fine-tuning has been done on a collection of datasets phrased as instructions and also on chain-of-thought data. Here they use 1,836 finetuning tasks with a combination of four mixtures from prior work based on the principle that the generalization would be better with scale. The chain of thought finetuning helps to generate the reasoning behind the answer and make it more convincing. T

ROUGE-L[8] has been widely used to evaluate the quality of generated text by comparing it with human-written text. For this, the longest common subsequence is computed from two sequences which are used to measure the correlation between the two. The ROGUE-L metric score for tk-INSTRUCT is 62.0 for English tasks. The baseline testing in the paper [2] and [5] shows that the models specifically trained to leverage instructions outperform the other models.

3 Methodology

In this project, we aim to explore the generalizability of the "google/flan-t5-large"[Link] model. Our goal was to test the model performance on individual tasks as well as composite tasks. We began by identifying three important NLP tasks for our analysis.

3.1 Identifying Natural Language Processing Tasks

We identified 3 tasks that we are interested in and could benefit from the task generalization:

Text Classification - This is a fundamental task in natural language processing. There any many models available publicly which have shown great performance for text classification.

Text Generation - This task is poised at generating coherent and meaningful text resembling a human-generated text.

Text Summarization - This involves condensing a long document into a shortened version that retains the most important information.

3.2 Implementing the Baseline Model

We implemented a baseline model for each chosen task. HuggingFace has been used as a primary resource for sourcing dataset/model matchups.

To assess the baseline performance of the models, we used various datasets for each task:
Sentiment analysis: "IMDb [Link]
Text generation: "amazon reviews multi" [Link]
Text summarization: "Samsum" [Link]

Using a test set of 70 samples provided meaningful results.

3.3 Design Composite Prompts

Once we had baseline metrics for each task, we used compositions of tasks to experiment with the generalizability of the model. We created the three following compositions of tasks:
Text summarization/Sentiment analysis (TS/SA) - 25 Prompts,
Text generation/Sentiment analysis (TG/SA) - 25 Prompts.
Text generation/Text summarization (TG/TS) - 21 Prompts,

For each prompt, we calculated the perplexity, likelihood, and confidence score of the generated text or output. We also performed human analysis on each response, dissecting where it may have gone wrong or generated an expected response correctly. Further, we analyzed the structure of our prompts, experimented with different parameters of the model, and analyzed the different scores obtained for successful and successful task performance by the model.

For prompt engineering, we referred to examples provided by SageMaker documentation [link]. There are provided parameters that lead to the best

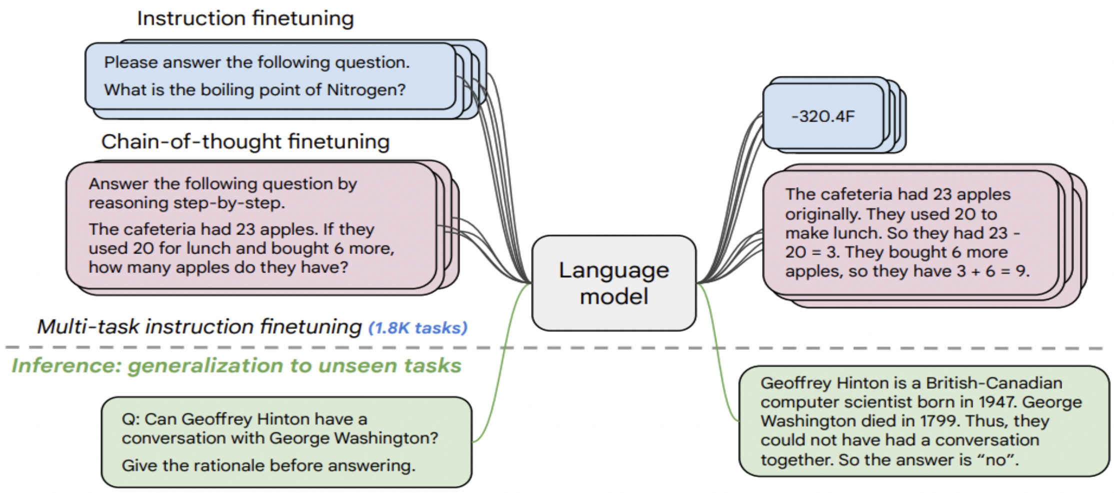

Figure 1: Finetuning Techniques applied to flan-t5 model[5]

results when each task is being run separately, but combining tasks also meant combining parameters. Further, the model should be able to understand the multiple tasks it has been asked to perform which seemed to be different from human understanding.

We landed on a set of parameters to use for each composition. Using the same parameters across the board allowed us to see which task was generalized better across the different compositions.

After passing each prompt through the model, we performed human analysis to further analyze the correctness of each output and which task performed best.

4 Experiment and Results

After the baseline scores were obtained and the prompts were designed, we generated the responses from our model. The Baseline scores obtained on the "flan-t5-large" model have been shown in Table 2. The prompts were individually analyzed with the help of different computational metrics as well as through human evaluation.

Dataset	Avg. Perplexity	Avg. Likelihood (s)
IMDb	239818.447	0.564
amazon-reviews-multi	15888.715	0.280

Table 2: Baseline perplexity and likelihood scores

4.1 Results and Analysis

The two major things we were hoping to understand were the success rates of the tasks while using generalized prompting and the methodologies for prompting that yielded better results with FLAN-T5. Due to the nature of our project, we relied heavily on human evaluation to determine the effectiveness of the prompting. We were able to draw from an AWS SageMaker blog post [link] for ideas on effectively prompting the FLAN-T5 model. By combining a number of strategies, we were able to identify a number of factors that led to consistent generalized results. Our findings showed that Top-p sampling was the most effective method for text generation involving multiple tasks. A greedy search was able to generate baseline answers but failed to elaborate on the prompt.

For prompts asking the model to perform text generation and sentiment analysis, we found that separating the tasks with "\n" greatly improved the model's ability to differentiate between the two tasks it was being asked to perform. It was also essential to isolate the title with "\" or else it would ignore the first task, identifying it as part of the title. In addition to this, a number of words influenced the effectiveness of the prompt. Certain words such as ‘imagine’, ‘continue’, ‘start’, and ‘fiction’ caused the model to respond unexpectedly, i.e. ‘start’ had to be replaced with ‘beginning’ or the model would fall into repetitive patterns.

The model correctly identified the sentiment of its generated text at a rate of 82 percent when given basic commands. It also proved effective at gener-

	Sentiment Analysis/ Text Generation	Text Summarization/ Sentiment Analysis	Text Generation/ Text Summarization
Input	Title: \”World War II.“\\nGiven the above title of an non-fiction story, write the story.\n Is this story negative or positive?	Input: Shelly: I'm looking for a fall coat, are there any good ones in today? Porter: We got some really nice wool ones in today. Come take a look! Shelly: I will! Be there on my lunch! Porter: You better take an early lunch, they will go fast! Shelly: Ooh! Okay! Return both tasks: Task 1: Summarize the input. Task 2: Analyze the overall sentiment of the input.	Short Story: \"The world ended with a suddenness that no one had predicted. There was no warning, no time to prepare. It began with a flash of light, followed by a deafening explosion. Buildings crumbled, trees snapped, and the ground shook with fury. In the days that followed, there were fires and floods and a relentless, choking dust that filled the air. People tried to flee, but there was nowhere to go. Those who survived huddled in the ruins, scavenging for food and water, waiting for a rescue that never came. They were alone in a wasteland, surrounded by death and despair. And as they looked around at the devastation, they wondered if there was any hope for a future in this new, shattered world.\ \n Generate a short conversation of 2 people discussing the summary of the given short story\n
Output	Negative. Because of the indiscriminate fighting, many people died. The death toll from the war was higher than expected. It was also the deadliest conflict on Earth. There were over a million killed. The average lifespan of a soldier in the war was about 10 years.	Positive. The wool coats will sell out quickly. They are for women. The shop is open until 6pm. It's very early. The coats are very nice. They are very affordable. The shop is open until 6pm. It's very early.	M: It's been a long time since we heard about the earthquake.W: Yeah, it's been a while, wasn't it?M: Yeah, I think it was about 5,000 years ago.W: That's really scary.M: It really was, wasn't it?W: Yeah, but it was an earthquake, and it was a deafening one, so we were all in shock and we didn't know what was going to happen.M: How long was it?W: About five hours.M: Then, what about the people who survived the earthquake?W: They scavenged for food and water, and waiting for a rescue that never came. They were alone in a wasteland, surrounded by death and despair.

Figure 2: Composite Prompt Examples

ating a text on a topic given a certain sentiment, i.e. The prompt: “Write a positive paragraph about artificial intelligence”, yielded the response “Artificial intelligence is a growing field that has the potential to transform the way we think and behave.” However, in cases where the prompts were made more complicated, such as asking the model to identify the sentiment of each sentence it generated, it did not provide any sentiment analysis whatsoever. Eg. "Prompt: \”Write a story about space“ \n Given the above prompt, write a story.\n Lable each sentence either positive or negative" did not generate any sentiment labels.

For prompts asking the model to respond to text summarization and sentiment analysis, we experimented with different types of input such as “dialogue”, “article”, “review”, and “input”. The two that performed the best for both tasks were “input” and “dialogue”. The model seemed to skew more towards text generation when prompted with the other two input types.

The model also performed both tasks at a higher success rate when explicitly told to complete both tasks: “return both tasks: task 1: analyze the overall sentiment of the dialogue. task 2: briefly summarize the dialogue.” compared to when both tasks were combined into one statement: “output: analyze the sentiment of the input and provide a brief summary of the input.”

For text generation and summarization, we were unsuccessful in cases where we asked the model to generate some text and then summarise it. The model doesn’t seem to understand the two tasks it has to perform and summarises the prompt. For example, "Prompt: Humans have reached Jupiter \n For the given prompt, Task 1: generate a fictional story.\Task 2: summarize the story in 2 sentences." generates a decent story but no summary is generated.

4.2 Model Evaluation

The averages of the computed scores from the model are given in Table 3.

Scores	TG/SA	TS/SA	TG/TS (s)
Avg. Perplexity	270.64	674.14	3837.16
Avg. Likelihood	0.58	0.57	0.53
Avg. Confidence	0.16	0.26	0.33

Table 3: Matrix evaluation of Composite Prompts

The frequency plots of the different tasks shown in Figure 3 describe how many times the model is

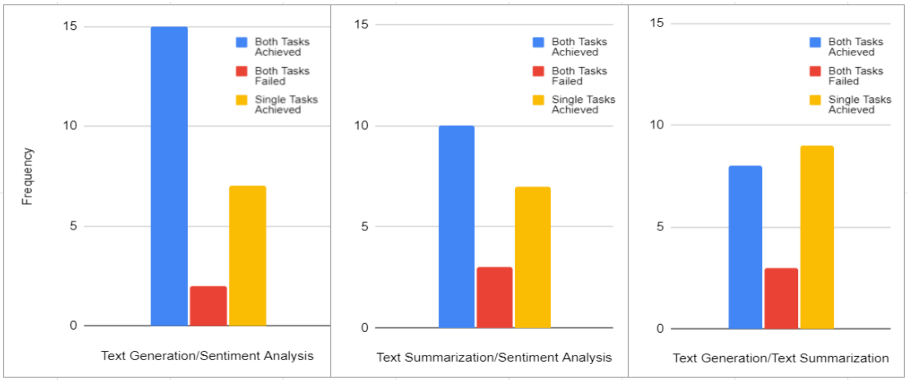

Figure 3: Composite Tasks Success Counts

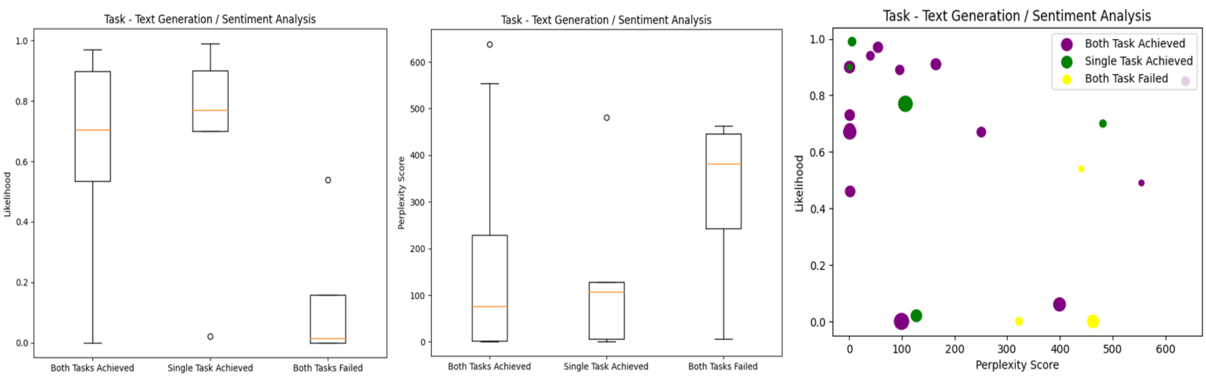

Figure 4: Text Generation/Sentiment Analysis Plots

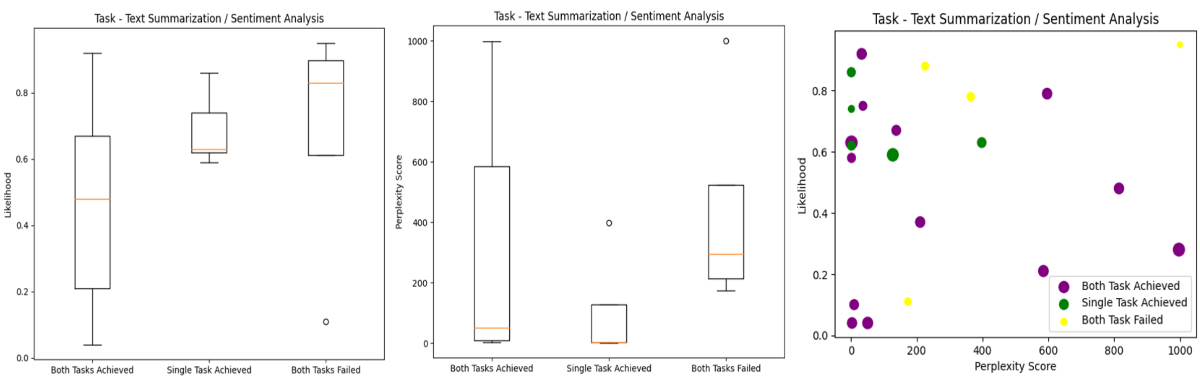

Figure 5: Text Summarization/Sentiment Analysis Plots

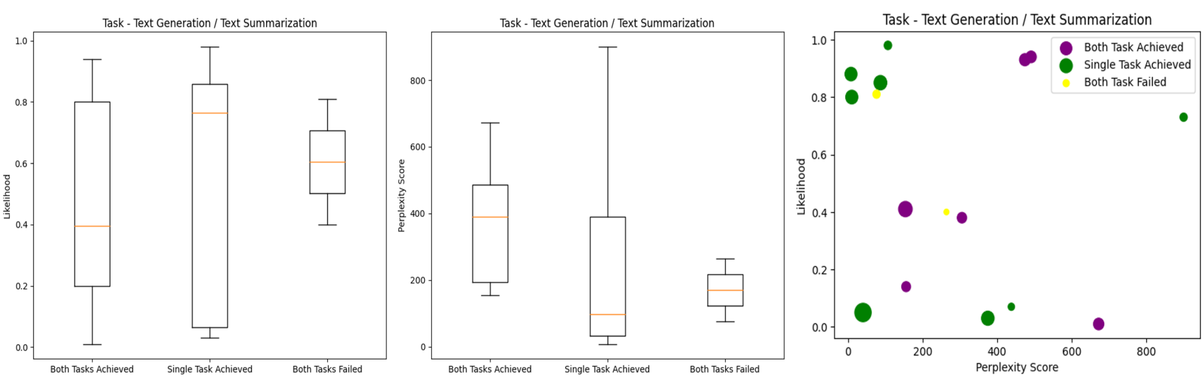

Figure 6: Text Generation/Text Summarization Plots

able to perform both of the composite tasks successfully or failed in both tasks or successfully complete a single task. This data is produced by human evaluation of the response corresponding to the prompt given. Apart from the Text Generation/Text Summarization case, the model was able to perform both the given tasks from the composite prompt in most cases. While the number of cases in which the model was able to achieve a single task was highest in the Text Generation/Text Summarization task. The frequency of failure of both tasks seems to be almost the same for all three composite tasks.

In Figures 4,5 and 6, a critical analysis of the different tasks is shown which combines data from both human evaluations of task success and also the computed metric scores(Likelihood, Perplexity, and Confidence). The box plots in Figure 4, represents the likelihood and perplexity score distribution of the model with respect to its success in performance. It can be seen that the median of the Likelihood scores of both single task successful and both task successful cases are much higher compared to both tasks failed cases, which is intuitive. Similarly anticipating the perplexity, the median is low for both single task successful and both task successful cases and is high for the counterpart. The scatterplot between the perplexity and likelihood is able to depict that the model is able to perform both tasks successfully when the likelihood score is high and the perplexity is low forming a cluster of violet points in the top left. The radius of the scattered points corresponds to their confidence score value.

In the Figure 5, the median of the Likelihood score of both task-failed cases is higher than the other two cases which shows that although the model is able to generate text with high likelihood scores, it is failing to perform a composite task of Text summarization and Sentiment analysis together sucessfully. The median of the perplexity score is also not that high for both task-failed cases compared to the other two. The following scatterplot finally depicts that we can not rely on the Likelihood and perplexity score for Text Summarization and Sentiment Analysis task as the violet points are almost scattered randomly forming no cluster.

Likewise in Figure 6, the model does not produce an intuitive graph for the Likelihood and Perplexity score for the Text Generation/Text Summarization task. Although the green and violet points (depicting single and both task successful cases respectively) are mostly present in the regions of high likelihood and low perplexity values. The radius of the scattered points corresponds to their confidence score value.

Thus from the overall analysis, it is seen that the model performs best for the Text Generation/Sentiment Analysis task. The parameters we found to be most effective across a variety of different prompts were top-p, min-length, and max-length. Changing these parameters heavily affected the model performance in all the composite tasks. Compared to the baseline performance, we see that the likelihood and perplexity scores of composite tasks are quite comparable to individual tasks. Further, we see a tradeoff between likelihood and perplexity scores in our results.

4.3 Report Findings

We have identified the following key takeaways from our results.

- The model has the potential to generalize tasks but is constrained by prompt structure and several keywords.
- The model performed both tasks at a higher success rate when explicitly told to complete both tasks.
- For prompts asking the model to perform text generation and sentiment analysis, we found that separating the tasks with "\n" greatly improved the model's ability to differentiate between the two tasks it was being asked to perform.
- Certain words such as 'imagine', 'continue', 'start', and 'fiction' caused the model to respond unexpectedly, i.e. 'start' had to be replaced with 'beginning' or the model would fall into repetitive patterns.
- The model was unable to summarise the text that it generated from a prompt.

5 Discussion

Task generalization has been growing steadily since last year with the introduction of ChatGPT. A better understanding of the prompts/instructions and identifying the patterns of failure will help us create robust NLP systems for the future.

5.1 Replicability

The results we have obtained are by using the google/flan-t5-large model which is available at hugging face. This model can be easily accessed by others. We have also uploaded our spreadsheet of prompts and responses to Github. By using the model and the same prompts similar results can be achieved.

5.2 Dataset

We have formed a spreadsheet of the prompts, and responses, along with their computed scores and human evaluation. This can be analyzed further by researchers to find more insights and trends which can be used as a reference when designing new models and can be further used as a baseline to test the generalizability of new models.

5.3 Ethics

One of the concerns is the presence of bias in the model response. The model is limited to the data it has been trained on and contamination of such data can result in a biased model. The data should be clean and diverse for the model to generate ethical responses.
Another concern that we have seen is that the model prioritizes certain tasks over others which can cause issues while dealing with critical tasks such as healthcare etc.
A big concern with such a multi-task-capable model is unemployment. These models may eliminate the need for workers in several industries. This would result in the increase of differences among different sections of society. Steps should be taken to ensure proper upskilling and a smooth transition of these workers to new roles. We must remember that AI models are made to improve human lives, and proper care must be taken to ensure employment for all.

5.4 Limitations and Future Work

This work can be extended to more tasks to further analyze the capacity of the model. Moreover, a comparison study can be done on multiple models. By fine-tuning some models for a certain task and testing their performance on other tasks, we would be able to better understand the relationship between different tasks. This could be used to reduce redundancy when training models.

6 Contribution

Peter Oritz - Baseline Implementation and Prompt Design, Amitabha Deb - Report Writing and Analysis, Issac Blaine Seuer - Prompt Design and Visualizations, Srijan Pal - Visualization and Prompt Analysis

The code and the spreadsheet for the project can be found at GitHub.

References

[1] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions, 2021.

[2] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022.

[3] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[5] 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 Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton,

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

[7] Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q. Tran, David R. So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc V. Le, and Mostafa Dehghani. Transcending scaling laws with 0.1 per cent extra compute, 2022.

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