GPT-3 scores in the top 7% of students in the United States on the ACT Reading Section

Oliver Brady
Stanford / Computer Science
oqbrady@stanford.edu

Andrew Hojel
Stanford / Computer Science
ahojel@stanford.edu

Salvatore Spina
Stanford / Computer Science
salspina@stanford.edu

Abstract

This paper investigates the ability of available GPT-3 models to take the ACT Reading section and the affects of finetuning GPT-3 with a dataset of 440 ACT Reading section questions. We found that using just one-shot prompts with the most powerful off-the-shelf version of GPT-3, which was recently finetuned to better follow directions, is able to achieve a remarkable average scale score on the ACT Reading section of 33.66 out of 36 over 3 test exams, which means it answered 112 out of 120 questions correctly (placing it in the top 93-96% of students in the US). In addition, we find that finetuning can drastically improve task-specific performance, evidenced by our custom finetuned GPT-3 model achieving the best zero-shot performance of a scale score of 30.66 over the same three tests.

1 Introduction

The goal of our project is to evaluate GPT-3's performance on a multiple choice standardized test. For our test metric we chose the reading section of the ACT, a common test students take in order to apply for college (ACT). We structure this task in a few different ways. First we evaluated GPT-3's ability to take the ACT as a zero shot task, meaning each prompt was just the passage and question with a blank spot for GPT-3 to fill the answer. Zero-shot means the model has seen no examples of the task. We then evaluate performance as a one-shot and two-shot task, meaning we gave an example question and answer or two before asking GPT-3 to answer another question about the passage. Finally, we fine tuned GPT-3 on a number of ACT tests to help it understand the task we were asking it to do.

Our motivation for this project was twofold. First, because of how widespread the ACT is, it is a helpful contextualization of GPT-3's performance in a way that people can easily understand. Second, the ACT breaks down the reading section into multiple reporting categories, which test for different skills. By analyzing model performance by section we can gain insight into where GPT-3 struggles and thrives.

2 Prior Literature

(Brown et al., 2020) introduced GPT-3, OpenAI's 175B parameter natural language processing model. The paper explains that previous NLP tasks have made substantive leaps through pretraining on a large corpus of text and then finetuning before being able to make use of task-agnostic model. The paper explores one-shot and two-shot learning where no gradient updates are performed on the model, unlike traditional finetuning. OpenAI explains that with significant parameter scaling, GPT-3 is able to use zero, one, and few shot learning to beat other traditionally fine tuned models in standard benchmark tasks. For tasks including (but not limited to) unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic, GPT-3 achieves strong performance on many NLP datasets without any gradient updates or fine-tuning, only natural language interaction with the model. OpenAI recently released an API for finetuning the base GPT-3 models.

There is little existing literature on the effects of finetuning GPT-3 for specific tasks.

However, (Wei et al., 2021) from Google Research investigated the effects of finetuning the 137B parameter FLAN (Finetuned LAnguage Net) model. They take 60 common NLP task datasets and create a new process that they call "instruction tuning." Whereas traditional finetuning involves tuning on task A and then testing on task A, they instead tune on a variety of other tasks and then test on the unseen task cluster. They compare this performance to both GPT-3 few-shot learning and GPT-3 zero-shot learning. They then tested FLAN on 25 language tasks and found that it surpasses GPT-3 zero-shot on 20 of 25 tasks. It also outperforms GPT-3 few-shot on 10 of the 25 tasks. This is despite FLAN never having seen examples of task it is being asked to solve and also starting from a smaller base model compared to the 175B parameters included in GPT-3. This paper shows how beneficial finetuning can be on a variety of tasks. We explored attempting instruction-tuning on GPT-3, but decided to finetune on a small dataset of real ACT problems given the prohibitively high per-token finetune and inference costs. A potential dataset for instruction tuning for the ACT Reading section is RACE++ (Liang et al., 2019). RACE++ is a dataset of around 110k focused multiple-choice machine reaching comprehension of varying difficulty built from English comprehension tests in China. In addition, (Wei et al., 2021) find that finetuning large LMs is more about teaching models how to properly retrieve and format information for a specific task than teaching the model the necessary background knowledge to perform the task.

This assertion is corroborated by (Reynolds and McDonell, 2021) who investigated how different styles of prompts can affect the performance of large pretrained language models such as GPT-3. (Reynolds and McDonell, 2021) argued that when deciding between finetuning, few-shot, and zero-shot prompts, zero-shot prompts may be the most effective because the purpose of the prompt is not to teach the model how to address that type of prompt, but instead, the goal of a prompt is to show the task location within the model's space of learned tasks. (Reynolds and McDonell, 2021) argue that it is worth noting that GPT-3 learns the autoregressive task of predicting the next word given a context; therefore, when writing a prompt it can be valuable to take into account the generation of a context that may encourage high performance on a task. For example, a prompt could create the context between a student and a teacher if explanation of reasoning behind a response is important. In addition, prompts must be constrained in attempt to make undesirable behavior / answers to a prompt more unlikely.

(Ouyang et al., 2022) attempted to address one of the major problems facing GPT-3. The problem is that GPT-3 is a model trained on generating the next word or phrase based on the prompt at hand, but most commercial applications are asking GPT-3 to perform specific tasks or follow specific instructions. An example of this is that if you asked GPT-3 "where is the moon?", it might auto-complete "where is the sun?" which is a logical statement to follow the first if you are not interpreting the first question as a direction. They decided to address this issue by curating a dataset built off of requests from their clients and having expert labelers answer the questions. They collected 13k prompt and answer pairs and then finetuned GPT-3 on that data. The result was a supervised fined tuned model, labeled SFT that outperformed GPT-3 on a all tasks according to the labelers.

While the SFT model was already outperforming GPT-3, OpenAI then took the process a step further and trained a reward function and optimal policy creator to select the best output from a handful of generated answers. They created this dataset by asking the labelers to rank multiple outputs from the SFT model. They called these new models InstructGPT. They found that this step improved the output of GPT-3 so drastically that the outputs from the 1.3B parameter InstructGPT-3 were preferred by the labelers over the outputs from the 175B parameter standard GPT-3.

(Ouyang et al., 2022) provided valuable information on the scale of data needed for finetuning, given that only around 13k data points were needed to finetune SFT, and then two more batches of roughly 30k labels were used to create InstructGPT and completely change the capabilities of the GPT-3.

3 Data

3.1 Overview

Given the high cost per token for training and running inference using the 175B parameter GPT-3 model, our primary goal for preparing data for finetuning was to prepare the highest quality small-scale dataset.

Although we investigated non-ACT datasets to prepare for the ACT Reading section, such as RACE++, developed by (Liang et al., 2019), we concluded that it did not provide high enough quality data to finetune for ACT. In addition, for the conditional generation tasks, the the OpenAI finetuning documentation recommends around 500 training examples (Ope). Thus, we decided compile our own dataset of ACT Reading sections.

We collected the ACT Reading section from 14 different ACT exams (each with 40 questions in the Reading section), resulting in 560 total questions. We then set aside 3 of these exams to use for model testing and evaluation, resulting in a finetuning dataset of 460 questions. These results are summarized in Table 1. In the following section, we will detail how this ACT data was collected.

The reading section of the ACT consists of four passages, each with 10 questions. One of the four passages is commonly a multi-passage, meaning it has two sub-passages (A and B) and then generally three questions about A, three about B, and then four questions about both. This passage type requires GPT-3 to differentiate between two different passages and synthesize information about both passages.

Data Split	# ACT Exams	# Individual Questions
Train	11	440
Test	3	120

Table 1: Dataset Breakdown

3.2 Collection

With the goal of getting the highest quality ACT data, we sourced only ACT exams that were administered to students or practice exams released by the ACT organization. These exams were sourced from (ACT), (Cra), and (McE). All of the exams we found were in PDF file format.

Given that all the PDFs we generated by the ACT organization, they had a standardized format. We used (McKie, 2022) to parse each PDF then had to manually remove certain artifacts that were not consistent for every page. In addition, we had to mark where the reading passage begins and the length of the passage in lines to properly parse the exam. After parsing the passage, answers, and questions for a reading passage within an exam, we would manually inspect the results to ensure the information was parsed properly (including cross referencing line numbers and corresponding lines, looking at each question, and ensuring the answers were properly parsed). We also implemented small automated tests to ensure that the PDF was parsed properly. Therefore, there is extremely high consistency across the parsed exam problems. See Table 2 to see a breakdown of every exam used.

4 Prompt Format

4.1 Overview

To standardize our results, we attempted to use prompts with very similar formats across different prompts and models. Although OpenAI fine-tuning documentation suggests the removal of any instructions, we decided keep the zero-shot prompt nearly identical across base and finetuning.

Data Split	Exam	Year
Test	Form E23	December 2021
Test	Form E25	April 2022
Test	Form Z08	April 2022
Train	Form D06	June 2021
Train	Form D05	April 2021
Train	Form Z04	April 2021
Train	Form D03	December 2020
Train	Preparing for ACT	2021-2022
Train	Preparing for ACT	2019-2020
Train	Preparing for ACT	2015-2016
Train	Preparing for ACT	2013-2014
Train	Preparing for ACT	2011-2012
Train	Preparing for ACT	2008-2009
Train	Preparing for ACT	2005-2006

Table 2: ACT Exams

In Table 3, we define a few terms that will be used consistently across prompts.

Section	Value
Header	"After reading a passage, choose the best answer to each question"
Passage Reference	"This questions asks about Passage *{"A", "B", or "A and B"}*."

Table 3: Prompt Keys (will be shown in blue in the Figures below)

4.2 Passage Section

The way that we displayed the information about a passage from the ACT Reading section and the passage itself was consistent across zero, one, and two-shot prompts. However, the prompt was formatted differently in the case of single-passage and multi-passage sections (as explained in 3.1)

Figure 1 displays how the passage section is formatted when there is only one passage.

```
Header\n
Passage Introduction\n
(1) line one of passage\n
(2) line two of passage\n
...
(90) line ninety of passage\n
```

Figure 1: Single-Passage Prompt (will be shown in orange in the Figures below)

Figure 2 displays how the passage section is formatted when there are two passages (A and B).

```
Header\n
Passage Introduction\n
Passage A\n
(1) line one of passages\n
(2) line two of passages\n
...
Passage B\n
(42) line forty-two of passages\n
...
(90) line ninety of passage\n
```

Figure 2: Multi-Passage Prompt (will be shown in orange in the Figures below)

4.3 Zero-Shot Prompt

The zero-shot prompt remained identical across base and finetuned models. For the finetuning dataset, an ending character of ` ###' was added to the desired completions. To address this, ` ###' was set as the stop token for the model during inference for the finetuned model. The zero-shot prompt can be found in Figure 3.

```
(Single/Multi)Passage-Prompt\n
Passage Reference (if multi-passage)\n
1. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer:\n
```

Figure 3: Zero-Shot Prompt

4.4 One-Shot Prompt

The one-shot prompt was never used for finetuning because given the nature of finetuning, OpenAI recommends that only zero-shot prompting is used for finetuning and inference on finetuned models. The one-shot prompt is very similar to the zero-shot prompt (Figure 3), but it includes an example of a question answered properly. The one-shot prompt can be found in Figure 4. To keep our token usage as short as possible, we selected the shortest question by word count from the passage as the example to show GPT-3. When we were predicting the answer for the shortest question itself, we provide the second shortest question as the example.

4.5 Two-Shot Prompt

The two-shot prompt was also not used in finetuning given the recommendations discussed in the previous section. The two-shot prompt follows a very similar format to the zero-shot and one-shot prompt, yet it includes two examples questions (the two shortest questions). The two-shot prompt can be found in Figure 5.

```
(Single/Multi)Passage-Prompt\n
Passage Reference (if multi-passage)\n
1. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer: correct answer\n
Passage Reference (if multi-passage)\n
2. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer:\n
```

Figure 4: One-Shot Prompt

```
(Single/Multi)Passage-Prompt\n
Passage Reference (if multi-passage)\n
1. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer: correct answer\n
Passage Reference (if multi-passage)\n
2. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer: correct answer\n
Passage Reference (if multi-passage)\n
3. Question\n
A. answer A\n
B. answer B\n
C. answer C\n
D. answer D\n
Answer:\n
```

Figure 5: Two-Shot Prompt

5 Models

5.1 Overview of Models

We evaluated the performance of three different versions of GPT-3 on all of our tasks.

GPT-3 is a 175B parameter auto regressive language model. The model architecture is heavily based on the Transformer architecture proposed by (Vaswani et al., 2017). Essentially, it is block of stacked transformers with multihead attention. The primary differentiation of the model is through scaling up the size and number of parameters. A more detailed description of the GPT-3 model architecture can be found in (Brown et al., 2020). OpenAI has multiple versions of GPT-3 released: ada, babbage, curie, and davinci. They increase in size and cost from ada to davinci. While we were originally planning on finetuning curie as our end model, initial tests with all four models revealed that davinci was the only model able to interpret the task in such a way that it would have any sort of diversity in its answers. The other three models simply guessed the same letter for all 40 questions on the test. Thus, due to cost, we decided to decrease the amount we were able to experiment with our models and stick to davinci based models the entire time.

5.2 InstructGPT Davinci (text-davinci-002)

The model text-davinci-002 is the most recent and powerful version of GPT-3 available through OpenAI's API. As mentioned in the prior literature section, it is a finetuned version of the original davinci, and was built to better follow human instructions. GPT-3 was originally trained to predict the word that is going to appear next in the text, but many customers and researchers prefer if the model is specialized to follow instructions. By finetuning davinci with human-instruction-based tasks, OpenAI created a powerful model that outperforms the original davinci on instruction based tasks.

The text-davinci-002 model also has the most recent training data, with information as recent as June 2021 included in the models training set. Due to this fact we chose the three most recent ACT's in our dataset to make up the test set. We have one test from December 2021 and two from April 2022 to make sure that the answers have not already been seen by GPT-3 during training.

5.3 Original Davinci (davinci)

The original davinci was the best and most powerful model when OpenAI first released GPT-3. It has now since been replaced by text-davinci-002, but it is still the best model available for finetuning through OpenAI's API. We planned to finetune this model, so we wanted to evaluate davinci on all of the tasks so that we had a baseline to compare to our finetuned model.

5.4 Finetuned Davinci (davinci-fintuned)

OpenAI allows users to create custom datasets in the format {prompt: completion} and finetune the original davinci model through their API. GPT-3 is known to change it answers quite drastically based on very small shifts in prompt format, and finetuning helps eliminate some of that variance from the answers and increase performance. Finetuning can be done with hundreds or a few thousand examples, and it is incredibly helpful in teaching GPT-3 how to navigate a task. Due the the cost of finetuning davinci on OpenAI, we unfortunately were not able to experiment with different prompt formats or hyperparamenters, and thus just have one finetuned model (for which we followed the suggested best practices from OpenAI).

6 Methods

6.1 Metrics

Given that GPT-3 is taking a standardized test, there exist many metrics provided by the ACT organization that we can use to analyze our model / prompt performance. The ACT Reading section contains 40 questions, and one metric that we report is number correct out of these 40 questions. However, each exam has a conversion chart from the raw reading score (out of 40) to a scale score in a range of 1-36. The scale score is calculated by looking at the performance of students on that specific exam and placing it on a distribution that matches other ACT exams. In other words, scale scores should have the same meaning for all the different forms of the ACT test, no matter which date a test was taken. Given that scale scores are the same across exams, we can analyze how the model performs across student averages over America (through the percentile metric).

In addition, we are able to analyze Reporting Category results. There are three Reporting Categories for the ACT Reading section (as reported by (ACT)):

Key Ideas and Details (KID)

- Determine central ideas and themes
- Summarize information and ideas accurately
- Make logical inferences
- Understand sequential, comparative, and cause-effect relationships

Craft and Structure (CS)

- Determine the meaning of words and phrases
- Analyze text structure and an author's word choice rhetoriclaly

- Understand authorial purpose and perspective
- Analyze characters' POV
- Differentiate between various perspective / sources of info

Integration of Knowledge and Ideas (IKI)

- Understand author's claims
- Differentiate facts and opinions
- Use evidence to make connections between different texts related by topic
- Analyze how authors construct arguments
- Evaluate reasoning and evidence from many sources

In Table 4, we show the breakdown of these different reporting categories in the ACT Reading section.

Category	# of Questions	% of Test
KID	22-24	55-60%
CS	10-12	25-30%
IKI	6-7	15-18%

Table 4: Breakdown of Reporting Categories in ACT Reading section

6.2 Experimental Approach

We kept zero-shot, one-shot, and two-shot prompts consistent across all models and across finetuning/testing. This was done to ensure consistency when evaluating the different models. We tested the zero-shot prompt with every model (davinci, davinci-finetune, and text-davinci-002). We tested one-shot on davinci and text-davinci-002 because the OpenAI finetuning documentation recommends to finetune only with zero-shot prompts. Finally, we were only able to test the two-shot prompt on text-davinci-002 because it has a a higher maximum prompt size than the davinci model, for which the two-shot prompt was too long.

The hyperparameters used for finetuning can be found in Table 7, and the hyperparameters used for inference across all models can be found in Table 8.

We chose the finetuning hyperparameters based on the recommendations of OpenAI for conditional generation, which stated that for a small dataset of around 500 examples, one should use a low learning rate multiplier and only train for 1-2 epochs.

We would have loved to perform hyperparameter sweeps / optimization, but were constrained by the cost of training and inference.

7 Results

In Table 5, we present the average results over all three ACT exams in the test set for different prompts and the models. For a more detailed breakdown of the results by exam, please see Table 9 in the Appendix.

In Table 6, we investigate the performance of the various models and prompts on different Reporting categories provided by the ACT exam grading rubric.

8 Analysis

Our primary takeaway from the results of this experiment is that text-davinci-002 is shockingly impressive in its ability to learn how to answer ACT reading questions after just one example, and analyze a fairly long passages to extract key information. A significant number of ACT reading questions refer to specific line numbers, yet the model has very little trouble localizing and extracting information. There are also multi-passage sections on most ACT reading tests, accompanied by questions that ask the test taker to compare and contrast concepts across passages. The text-davinci-002 model's ability to answer questions across passages and perform better than 93-96% of students shows the power of the model.

We were also surprised by how much better text-davinci-002 performed compared to the original davinci model. While we knew from (Ouyang et al., 2022) that human evaluators preferred outputs from the intructGPT models, we expected our finetuned davinci model to perform at least on par with text-davinci-002. Though, due to our limited budget and thus constrained ability to hyperparameter tune, this experiment does not show that finetuneing could not perform as well as text-davinci-002 on this task. We were too constrained in our experimentation to be confident that our finetuned model is close to the ceiling of how good a finetuned davinci model could perform.

One of our motivations for testing on the ACT was the fact that the ACT labels each question with the specific skill that it is testing. We thought this would be an exciting chance to see what GPT-3's strengths and weaknesses are. The results in table 5 are relatively intuitive in terms of the order of

Model	Prompt	Avg Correct (out of 40)	Avg Score (out of 36)	Percentile Range
davinci	zero-shot	12	13	15%
davinci	one-shot	22.33	21	56%
davinci-finetune	zero-shot	30.66	27	79%
text-davinci-002	zero-shot	27.33	23.66	67-71%
text-davinci-002	**one-shot**	**37.33**	**33.66**	**93-96%**
text-davinci-002	**two-shot**	**37.33**	**33.66**	**93-96%**

Table 5: Average Results

Model	Prompt	KID	CS	IKI
davinci	zero	33.3%	25.7%	23.1%
davinci	one	63.9%	48.6%	53.8%
davinci-finetune	zero	77.8%	85.7%	46.2%
text-davinci-002	zero	70.8%	71.4%	46.2%
text-davinci-002	one	**95.8%**	91.4%	**84.6%**
text-davinci-002	two	**95.8%**	**94.3%**	76.9%

Table 6: Average Reporting Category Results

difficulty of questions for the model. The models were best at questions about key ideas and details, then craft and structure, and finally the integration of knowledge and ideas.

9 Conclusion

We set out to analyze how GPT-3 performs on the ACT Reading section and to investigate how finetuning can affect performance. A clear finding is that InstructGPT-3 (text-davinci-002) performed remarkably well using single and two-shot promtps and achieved our best aggregate results of a 33.66 scale score over three different ACT exams, which places it in the top 93-96% of students in the United States. In addition, we found that our finetuned base GPT-3 model (davinci-finetune) achieved the highest zero-shot performance, achieving an average scale score of 30.66 (top 79% of students in the US). This was a dramatic improvement over the near random performance of davinci zero-shot. Therefore, we can see that finetuning can drastically improve zero-shot, task-specific performance, yet the InstructGPT model shattered expectations when given an example of how to respond. Interestingly, inference for text-davinci-002 was less expensive than davinci-finetuned making a strong case for using text-davinci-002 with one or two-shot prompts on novel tasks. Yet, finetuning with more data and more extensive hyperparameter tuning is needed to back up this hypothesis.

There are a lot of exciting experiments that could be carried out with more funds and higher spending limits, such as extensive hyperparameter tuning and attempting to take other sections of the ACT.

Known Project Limitations

Due to the overhead of parsing data and training/-tuning this model, we have a few known project limitations. One limitation is that a test set of three ACTs is not as large of a sample size as we would ideally have. Another limitation is that because we only had one shot at finetuning due to the cost, we never had the opportunity to do any sort of extensive hyperparameter tuning. A final limitation of our project is that due to the size of GPT-3 and the fact that we are unable to access the model directly, we have no real ability to see how it reasons. Without the ability to analyze metrics directly about the model, all we are able to do is observe its results and the log probabilities of different tokens.

Authorship Statement

Andrew Hojel:

- Built the PDF parser for ACT exams
- Converted the parsed ACT PDFs into .jsonl files for finetuning
- Designed and structured prompts

Oliver Brady:

- Ran the inference steps for all of the models
- Analyzed the ACT test results by sub-categories

Sal Spina:

- Sourced online ACTs for parsing
- Conducted first set of davinci vs. curie tests
- Assisted Andrew with cleaning the data after the PDFs were parsed

Acknowledgements

We would like to thank our mentor Sterling F Alic III and professor Christopher Potts for their guidance on this project.

References

50 official sat pdfs and 87 official act pdf practice tests (free).

Act downloads real act tests download.

Act: Our only measure of success is yours.

Openai fine-tuning api.

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

Yichan Liang, Jianheng Li, and Jian Yin. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Asian Conference on Machine Learning*, pages 742–757. PMLR.

Jorj X. McKie. 2022. Pymupdf. https://github.com/pymupdf/pymupdf.

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. 2022. Training language models to follow instructions with human feedback.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

A Example Appendix

Model	Batch Size	LR Multiplier	# Epochs
davinci-finetune	1	0.025	1

Table 7: Finetuning Hyperparameters

temperature	top-p	max_tokens
0	0.95	1

Table 8: Inference Hyperparameters (used across all models)

Model	Exam	Prompt	Correct (out of 40)	Score (out of 36)	Percentile
davinci	Form E23 (Dec 2021)	zero-shot	13	14	21%
davinci	Form E25 (Apr 2022)	zero-shot	11	12	11%
davinci	Form ZO8 (Apr 2022)	zero-shot	12	13	15%
davinci	Form E23 (Dec 2021)	one-shot	25	22	62%
davinci	Form E25 (Apr 2022)	one-shot	25	22	62%
davinci	Form ZO8 (Apr 2022)	one-shot	20	19	50%
davinci-finetune	Form E23 (Dec 2021)	zero-shot	30	27	79%
davinci-finetune	Form E25 (Apr 2022)	zero-shot	34	30	86%
davinci-finetune	Form ZO8 (Apr 2022)	zero-shot	28	24	71%
text-davinci-002	Form E23 (Dec 2021)	zero-shot	28	24	71%
text-davinci-002	Form E25 (Apr 2022)	zero-shot	26	23	67%
text-davinci-002	Form ZO8 (Apr 2022)	zero-shot	28	24	71%
text-davinci-002	Form E23 (Dec 2021)	one-shot	37	34	96%
text-davinci-002	Form E25 (Apr 2022)	one-shot	38	34	96%
text-davinci-002	Form ZO8 (Apr 2022)	one-shot	37	33	93%
text-davinci-002	Form E23 (Dec 2021)	two-shot	37	34	96%
text-davinci-002	Form E25 (Apr 2022)	two-shot	37	33	93%
text-davinci-002	Form ZO8 (Apr 2022)	two-shot	38	34	96%

Table 9: Total Results Results