Parameter Efficiency, Few-Shot, Zero-Shot, Prompting

Jonathan May

September 29, 2024

Preamble

These notes and previous notes on Pretrained LMs are possibly the most likely of all notes you'll get in class to go out of date. See that handout for details.

1 Parameter Efficiency

The models we've discussed so far follow the paradigm that, out of the box, they don't do too much, but when you expose them to some supervised data that is an exemplar of a task and fine-tune their parameters they can do the task when given more input data. One problem with this paradigm is that the base models are quite large, and then, when fine-tuned, you have another model that is as large as the base. If you have k tasks you have to store k copies of the fine-tuned base model. This is inefficient, so there have been efforts to allow the scaling to many tasks without exploding the number of models that have to be saved. This is an active area of research (as of this 2024 update), but here are a few interesting approaches to parameter efficiency.[1]

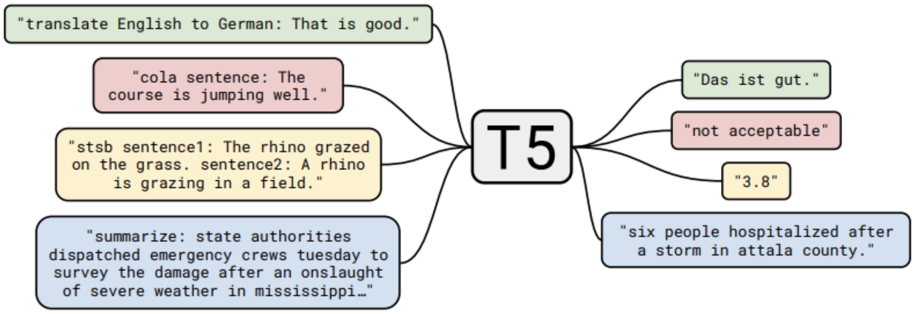
1.1 Multi-Task Learning

If you train a model to do more than one thing, then you implicitly are saving parameters.[2] T5 is one example of that approach. The pre-trained model is fine-tuned on many tasks all at once, each prepended with an instruction relevant to that task. T5 has since been further fine-tuned on more downstream tasks. A limitation of this approach is that the instructions are 'hard', i.e., you can do the tasks you're trained on, but it isn't clear you can do any other tasks.

---

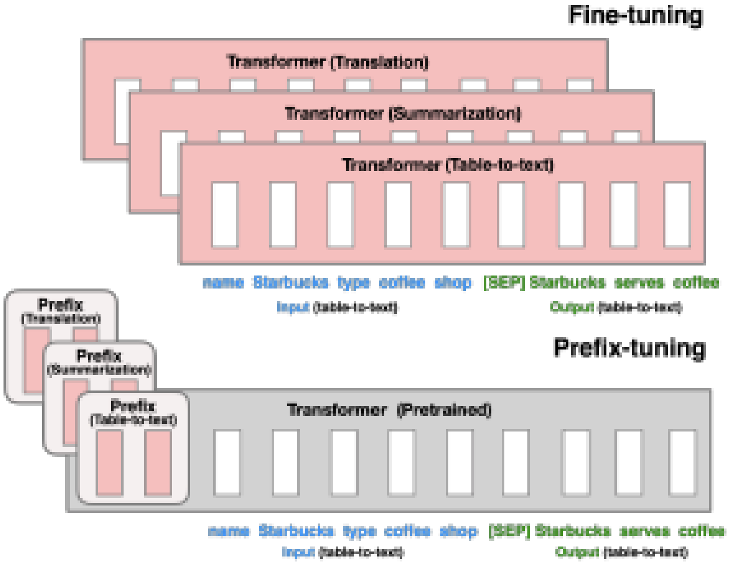
[1] Some of what follows from here: `https://github.com/allenai/acl2022-zerofewshot-tutorial`

[2] Note that the term *multi-task learning* is also applied to a case where you use a single model to do two different kinds of prediction at the same time. I'm abusing the term here.


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

1.2 Continuous Prefix Tuning

The idea behind prefix tuning[3] and a number of other related works is that, rather than imagining or declaring what the task prefix will be as in T5, it might be better to learn the prefix as well. Thus, we can imagine several otherwise unused tokens being learned when a pre-trained model is exposed to new task data; the rest of the model isn't modified! Then, instead of preceding your input with 'summarize' you precede it with '[TASK465]' or whatever that you've previously learned. You could do this for any number of tasks independently. There are a few versions of this; the one from [7] is shown below. The results on a few tasks are sometimes better than full fine-tuning but the goal is to reach parity despite only fine-tuning 0.1% of the parameters.


Fine-tuning
Transformer (Translation)
Transformer (Summarization)
Transformer (Table-to-text)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Prefix-tuning
Prefix (Translation)
Prefix (Summarization)
Prefix (Table-to-text)
Transformer (Pretrained)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)

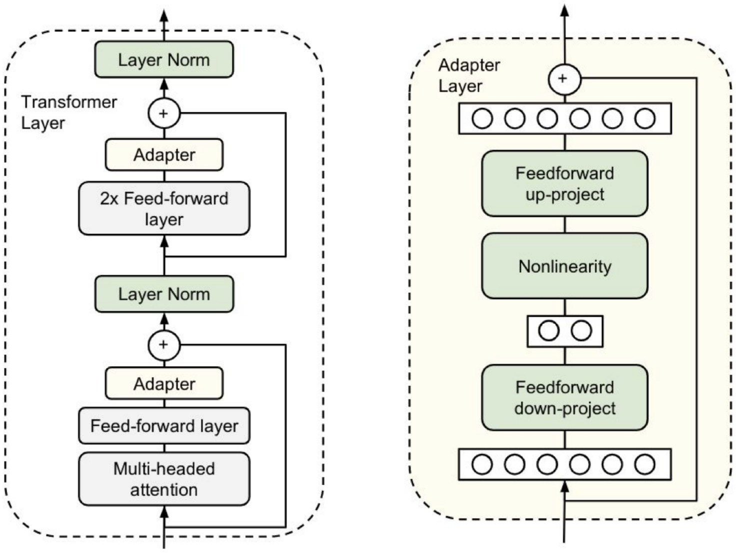
1.3 Adapters

Adapters [3] are kind of the same idea as prefix tuning but instead of adjoining to the left of the stack, they surround each layer of the transformer. The motivation for these came out slightly different from those of prefixes; before adapters, people were playing around with

---

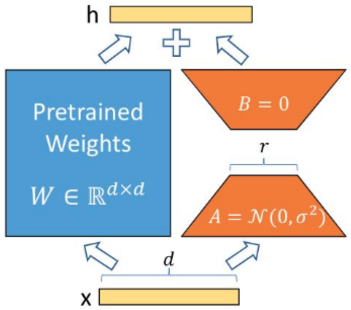
[3] `https://aclanthology.org/2021.acl-long.353/`

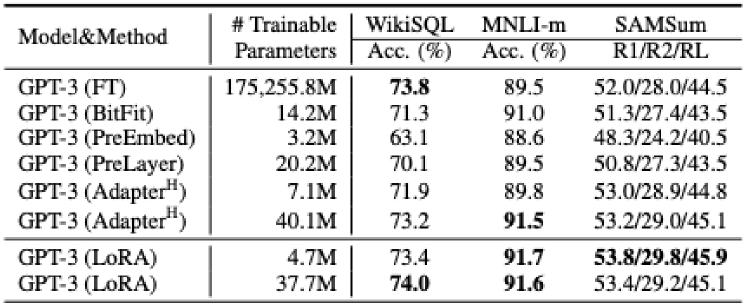
only fine-tuning some of the layers in a PTLM, and adapters were shown to be more efficient than this approach. A slightly larger footprint was claimed in the original adapters paper (3.6%) than by prefixes, but this is, of course, a hyperparameter. Below is a figure from the paper showing how adapters are added in. A goal of adapters had been to make a 'plug and play' approach to model specialization by using off-the-shelf adapters built for one task and combining them together; I haven't heard too much about this lately, probably because of in-context learning and prompting changing the paradigm somewhat.



1.4 LoRA

Instead of adding parameters around the current stack why not add them *to* the stack? That is, given $d \times d$ weight matrix W (e.g. a Q, K, V matrix), let $W' = W + W_l$ and let $W_l = W_{l1} \times W_{l2}$ where W_{l1} is $d \times r$ and W_{l2} is $r \times d$, $r << d$. How much lower? Well, it varies, but values of 4 or 8 were used in the paper [4]. A 'shadow matrix' was learned for the W matrices (sometimes only a subset of them). As few as 0.02 parameters were learned (in the case of GPT-3) with very good results! Deep in the appendix of the paper the authors noted the combination of LoRA and prefix tuning worked very well!


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

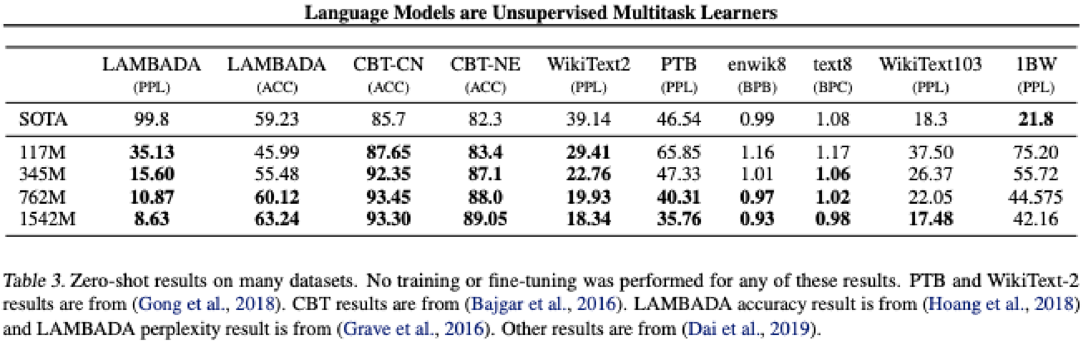


Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter^{H})	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter^{H})	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

2 Learning without Adapting

2.1 GPT-2

OpenAI, presumably not happy with being overshadowed by BERT, but not really interested in branding any better, released the paper “Language Models are Unsupervised Multitask Learners”[4] that described GPT-2 in February 2019. It was trained on 8m documents and 40GB of text sourced from outbound links from Reddit. The biggest model, at 1.5b parameters, exceeded SOTA performance considerably on a variety of LM sets even though it was not adapted to them. It was even able to do well (but not SOTA) on machine translation, question answering, and summarization tasks just by passing in natural language sequence prompts that attempted to elicit the kind of task response desired (e.g. inputting an article, then “TL;DR”, then allowing the system to generate, or inputting a few translation pair examples, then a source sentence but then generating the target). The thing that really made the news was the (cherry-picked) ‘unicorn’ story. The other thing that made the news was OpenAI’s refusal to let nearly anyone actually use the full system for the first year or so, claiming it was ‘too dangerous.’ So there was a lot of skepticism. But eventually, the models were released, and the claims were, in fact, justified.



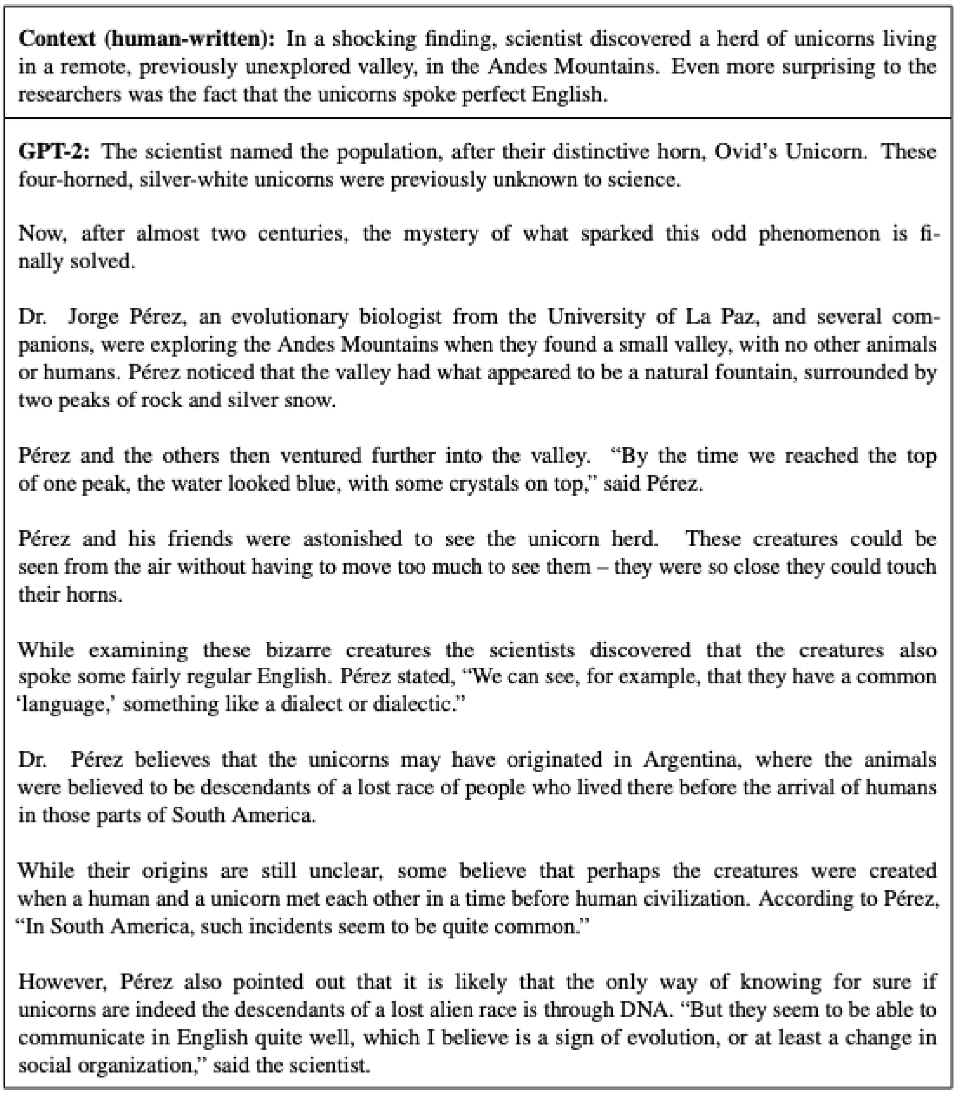
Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

---

[4] `https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`



Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Table 13. Conditional generation on an out-of-distribution context by GPT-2. Cherry pick of 10 samples generated with $k = 40$.

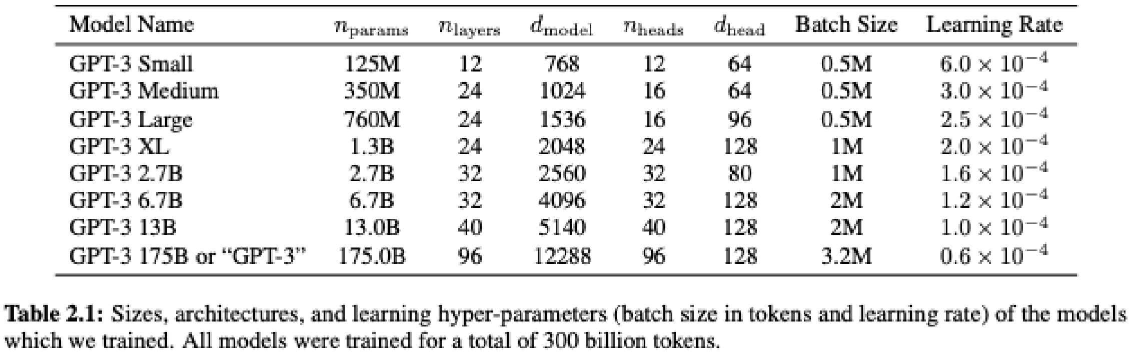
It's worth pausing to describe the two (ish) approaches that are being proposed here and why they are now so cool. The main reason they are cool is that the model parameters aren't ever changed, so manipulation of behavior occurs only at inference time.

1. **In-Context Learning** or demonstration based learning. Several (two? three? ten?) examples of desired input-output behavior are given, then one or several inputs without an output are given and the model produces outputs that follow the pattern.
2. **Instruction Learning**. A natural language description of what is wanted is produced, then an input is given, and the model produces outputs that are responsive to the instruction.

In practice, both are used together. There are also questions about the specific value of each component (see below).

2.2 GPT-3

In 2020 OpenAI released "Language Models are Few-Shot Learners"[5] which heralded the release of GPT-3 in a paper that was 72 pages long (GPT-2 paper was 24 pages long) and that described a model with 175b parameters. It was trained on 570GB of text (500b tokens) from the Common Crawl, their previous data set, and some other 'high quality' data sets. When prompted with only a few examples (and sometimes with none at all) it outperformed SOTA on a wide variety of tasks, including common sense reasoning tasks, machine translation, question answering, and others. This time OpenAI set up an API and web interface so that researchers and others could use the models. The results have been really excellent though sometimes care is needed to prompt appropriately.



Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

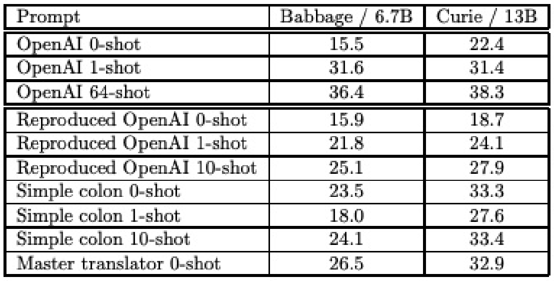
Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

2.3 Why/how does in-context learning work?

(From `https://github.com/allenai/acl2022-zerofewshot-tutorial`)

Some views (this is an area of active study, though):

- Demonstrations do not teach a new task; instead, they allow the 'locating' of an already-learned task during pretraining [12]. This was concluded after noticing that prompt-engineering zero-shot machine translation was often better than larger few-shot examples, in contrast to GPT-3 paper claims.



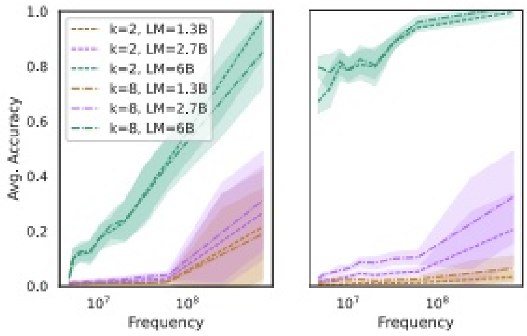
Prompt	Babbage / 6.7B	Curie / 13B
OpenAI 0-shot	15.5	22.4
OpenAI 1-shot	31.6	31.4
OpenAI 64-shot	36.4	38.3
Reproduced OpenAI 0-shot	15.9	18.7
Reproduced OpenAI 1-shot	21.8	24.1
Reproduced OpenAI 10-shot	25.1	27.9
Simple colon 0-shot	23.5	33.3
Simple colon 1-shot	18.0	27.6
Simple colon 10-shot	24.1	33.4
Master translator 0-shot	26.5	32.9

Figure 1: From [12]. Properly prompt engineering is better than few-shot.

---

[5] `https://arxiv.org/abs/2005.14165`

- In-context learning performance is highly correlated with term frequencies during pretraining [11]. This work looked at numerical reasoning tasks ("What is 14 times 30" or "What is 27 years in months?"). There was a strong correlation between frequency of seeing the operands in training (statistics were calculated on the Pile) and performance on these tasks.



(a) Arithmetic-Mult. (b) Arithmetic-Add.

Figure 6: **The effect of model size on performance** Smaller models only perform well on instances with more frequent terms in the pretraining data. *k* represents the number of shots.

Figure 2: From [11]. Performance correlates strongly with pretraining frequency

- LMs do not exactly understand the meaning of their prompt [18]. In this work the authors focused on NLI. They gave a variety of shots to smaller models like ALBERT (235M) all the way to GPT-3 (175B), varying both the number of shots and setting up the prompt in reasonable ({prem} Are we justified in saying that "{hypo}"?), misleading ({prem} Are there lots of similar words in "{hypo}"?), and irrelevant ({prem} If bonito flakes boil more than a few seconds the stock becomes too strong. "{hypo}"?) ways. In many cases none of this mattered, or didn't matter much. It also wasn't the case that models learned more slowly when given opposite prompts or misleading info; sometimes they even learned faster.

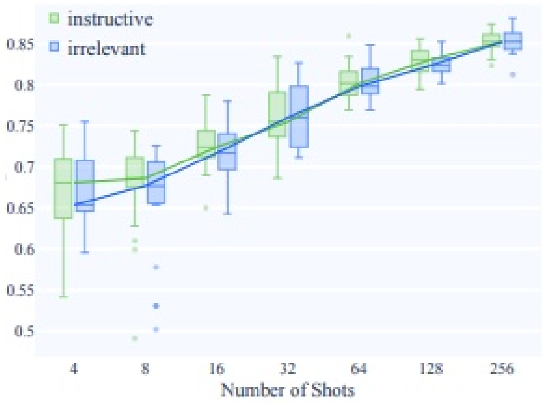
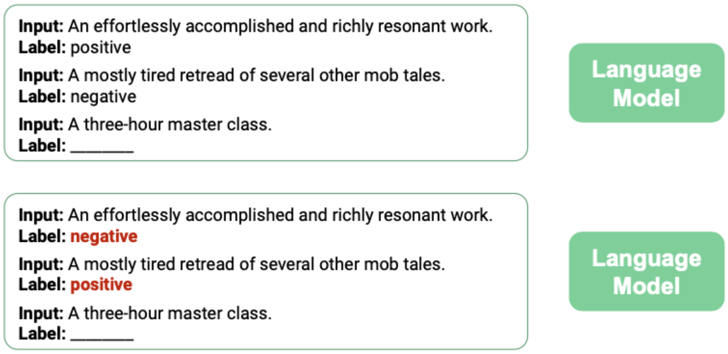


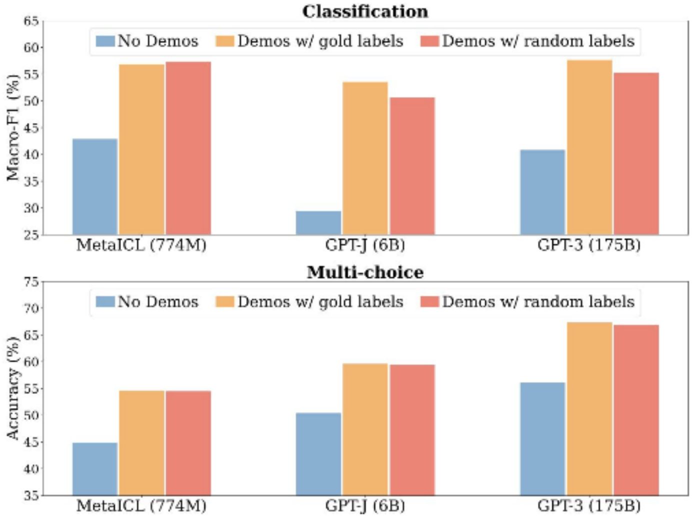
Figure 3: From [18]. The way shots are provided doesn't seem to matter much.

- LMs do not need input-label mapping in demonstrations, instead, they use the specification of the input & label distribution separately [8]. you can thus prompt with examples but *random* answers and get basically the same results as if you prompted with correct labels.

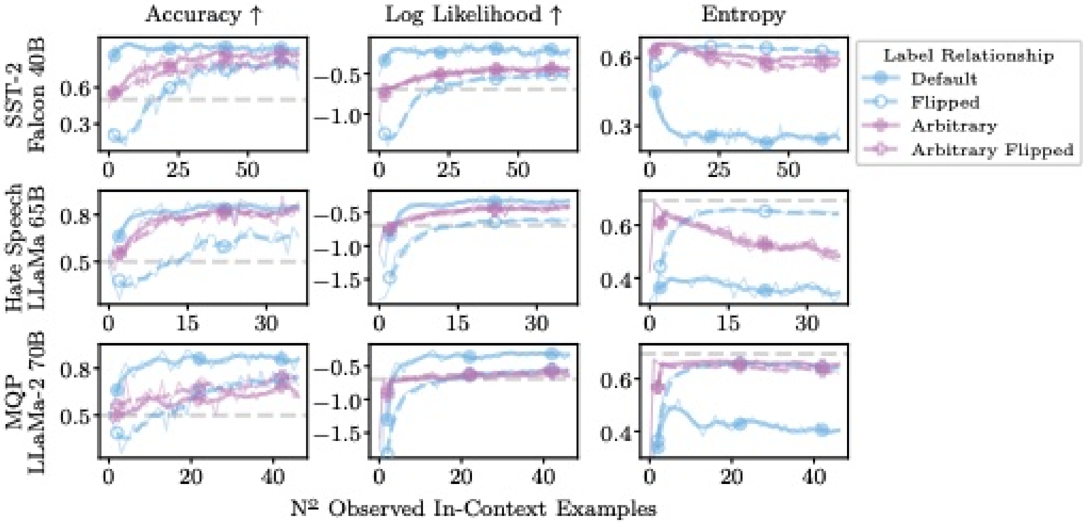


Input: An effortlessly accomplished and richly resonant work.
Label: positive
Input: A mostly tired retread of several other mob tales.
Label: negative
Input: A three-hour master class.
Label: ________

Input: An effortlessly accomplished and richly resonant work.
Label: negative
Input: A mostly tired retread of several other mob tales.
Label: positive
Input: A three-hour master class.
Label: ________

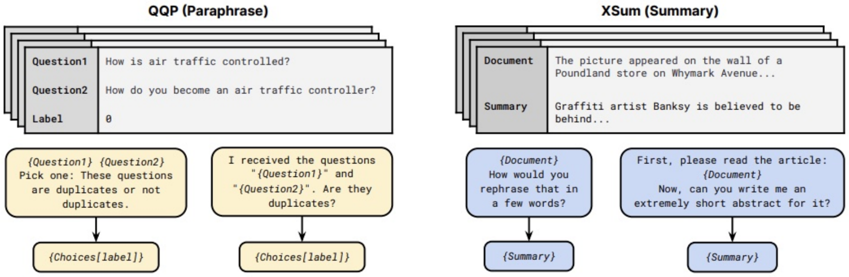


- In [6], the effect of varying ICL shot counts and model types is studied with regard to not only accuracy metrics but also model confidence (via log likelihood). They try to overcome knowledge inherent in pretraining (i.e. what would be predicted in a zero-shot case) and to learn tasks that can't be known from pretraining. Evaluation is over a suite of classification tasks, including NLI, paraphrasing, sentence similarity. Tested on LLaMa and Falcon models. When repeating the [8] experiments they notice confidence drops a lot. Accuracy isn't as good as previously noted but here larger models and larger contexts are used. Generally the models have a hard time reversing their decisions from zero-shot when ICL with consistent "wrong" labels are provided.



2.4 Instructions Only

Recall the T5 paper, which jointly trained for many tasks. Each task had its own name and style, but the point here was parameter saving. In the T0 paper (from mostly Hugging Face) [13], a T5-style model was built again but this time each of the tasks was described in a simple natural language way, like you might express the task if you were asking another human to do it. The authors solicited multiple ways to express these prompts and collected them into a free resource. There are 2,000 prompts written for 170+ datasets, expanding to 200m+ naturally prompted instances of tasks.



Question1	How is air traffic controlled?
Question2	How do you become an air traffic controller?
Label	0

{Question1} {Question2} Pick one: These questions are duplicates or not duplicates.

{Choices[label]}

I received the questions "{Question1}" and "{Question2}". Are they duplicates?

{Choices[label]}

Document	The picture appeared on the wall of a Poundland store on Whymark Avenue...
Summary	Graffiti artist Banksy is believed to be behind...

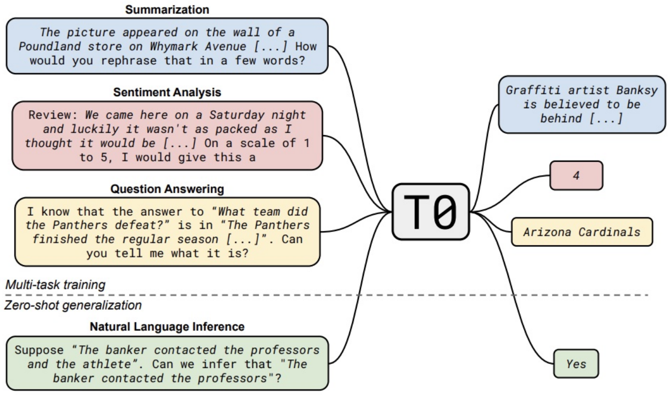
{Document} How would you rephrase that in a few words?

{Summary}

First, please read the article: {Document} Now, can you write me an extremely short abstract for it?

{Summary}

For the purposes of the paper, some tasks were held out, and then evaluated on. So the paradigm is like T5 but with a more natural input, which should in theory lead to generalization.


Summarization
The picture appeared on the wall of a Poundland store on Whymark Avenue [...] How would you rephrase that in a few words?
Sentiment Analysis
Review: We came here on a Saturday night and luckily it wasn't as packed as I thought it would be [...] On a scale of 1 to 5, I would give this a
Question Answering
I know that the answer to "What team did the Panthers defeat?" is in "The Panthers finished the regular season [...]". Can you tell me what it is?
Multi-task training
Zero-shot generalization
Natural Language Inference
Suppose "The banker contacted the professors and the athlete". Can we infer that "The banker contacted the professors"?
T0
Graffiti artist Banksy is believed to be behind [...]
4
Arizona Cardinals
Yes

The results are that adding natural prompts helps train models to generalize to new text instructions. These models had not seen the tasks they were doing explicitly but could follow human-like instructions.

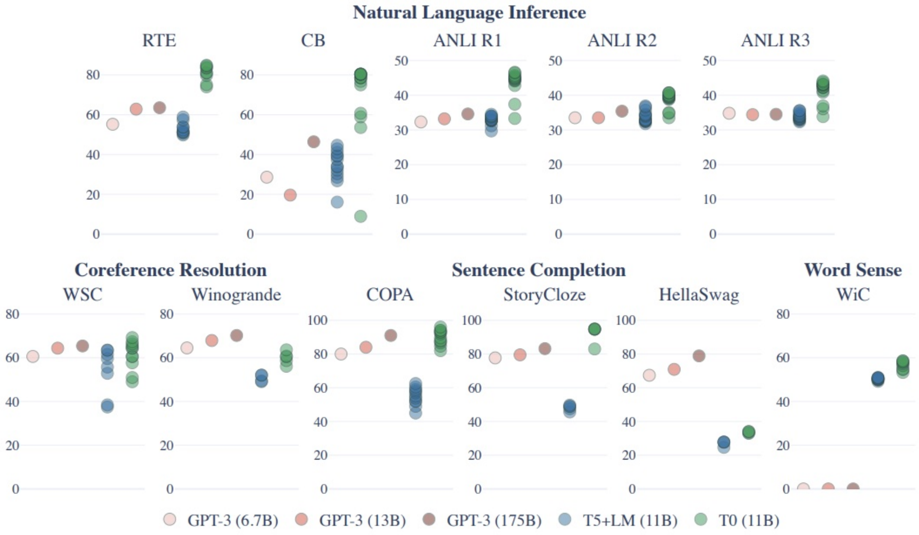


Figure 4: Results for T0 task generalization experiments compared to GPT-3 (Brown et al., 2020). Each dot is the performance of one evaluation prompt. The baseline T5+LM model is the same as T0 except without multitask prompted training. GPT-3 only reports a single prompt for each dataset.

2.5 Chat GPT and beyond

With somewhat less fanfare than GPT3, in March 2022 OpenAI released 'Training language models to follow instructions with human feedback'[6] and replaced its GPT-3 models with these. Even the smallest (1.3b param) models were shown to be preferred to GPT-3 175b by users. They were fine-tuned to respond better to user prompts using 'Reinforcement Learning with Human Feedback' (which we will cover in a separate lecture). The authors also claim these models are less toxic than previous models.

In late November 2022, OpenAI released Chat GPT, a chatbot interface that wrapped this enhanced GPT-3 in a free-to-use and widely available interface. The promotion of and

---

[6] `https://arxiv.org/abs/2203.02155`

response to ChatGPT was very very loud. Within a few months seemingly everyone, even beyond the tech world, was aware of the technology, surprised at what it could do, and possibly scared of it. This kicked off an arms race among leading tech companies to build their own models (which started being called 'large language models' despite the term being fairly vapid).

We've had a lot of models since then. Here's an attempt to cover what differentiates some of them (sometimes it's hard)

- LLaMa – from Meta/FB. Llama 1 in Feb 23 [17]. Llama 2 in Jul 23 [16]. Llama 3 in Aug 24 [1].
  - Llama 1: Normalize before sublayer instead of after. Change ReLU to SwiGLU (invented by Noam Shazeer at google and benefits attributed to "divine benevolence). Use "rotary positional embedding". 7b to 65b params. 21 days to train biggest model on 2048 A100s. 1.2T of public data (but not public preprocessing). Instruction fine-tuning helps a little but not the major part.
  - Llama 2: 7b to 70b params. 40% more data, 7b–70b params. Chat version is PPO RLHF'd to be like ChatGPT. Total of 3.3M GPU-hours of A100s used to build all models (maybe doesn't account for false starts). Safety tuning in RLHF where tasks that have safe and unsafe outcomes are used in optimization. Red teaming of early models followed by more RLHF.
  - Llama 3: much much more data (15T tokens vs 1.8T for Llama 2). 8b–405b models and instruction variants. DPO instead of PPO in RLHF. 8k context window at first, followed by 128k. More complex tasks like tool use in RLHF. Vision and speech models incorporated via paired data (image, text) and (speech, text) but multimodal parts not released. Various data filtering including dirty words, duplicates, numerical garbage, PII. A form of "grouped query attention" where key and value matrices are shared by a few query matrices (details not described here). 100k english token vocab plus 28k more for all other languages (!). Up to 126 (!!) layers. Trained on up to 16k H100 GPUs. Tool usage enables call out to API; model is trained to make calls, then calls are made and results returned as context, then inference continues
- Gemini – from Google. Gemini 1 [15] Dec 23. Gemini 1.5 [14] Mar 24.
  - Gemini 1: Sentence piece tokenizer. Built for TPUs. Has 32k context length. Post trained with RLHF. Inherently multimodal. Safety checks and Red teaming. Smallest models are 1.6B params. This is all I got out of the main paper. Compute not reported. Data size or content not reported other than it is multimodal and multilingual.
  - Gemini 1.5: Claims up to 10m tokens of context. (Long context evaluations – "needle in a haystack" – are provided). Lots of evaluation details just like with Gemini 1, but no concrete details on how the models are built or what differentiates them from other models.

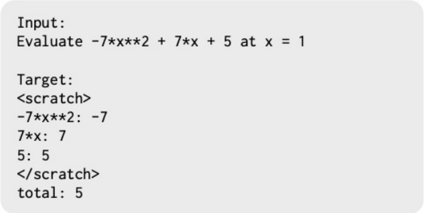
- ChatGPT and GPT-4 series – from OpenAI. There was no explicit whitepaper or research paper for the initial ChatGPT release. However there was one for GPT-4 [10] in Mar 24. On the first page they say " Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar." Cool.
- Claude – from Anthropic. Now up to version 3. I couldn't find any papers or any details...
- OLMO – from AI2. Open model! [2]. Released June 24. Complete logbooks and details provided for as much openness as possible. 1B and 7B versions. No bias terms (this is true in llama apparently but i missed that my first read through). Layer norm as described in transformer paper; other layer norms involve a linear or other transform. SwiGLU, ROPE. Overall very similar to LLaMa. Built on open Dolma dataset. DPO RLHF alignment. Trained on both NVIDIA and AMD GPUs (that's nontrivial!).

3 Introspective Model Ourobouros

Now that we have models that are starting to act a lot like humans, a recent trend has been to, when they don't get an answer right, ask them to think about their answer. This paradigm has led to a situation where we use LLMs to improve LLMs, which ultimately strikes me as rather mysterious.

3.1 Scratchpad

Scratchpad [9] was a relatively early access point into this idea. It had been long known that even RNNs were Turing-complete, as long as they had infinite thought time. Transformers were shown to also have this property, but in general we hadn't allowed models to "think" without writing anything. The idea of a scratchpad is that a model can output details about its "thought" process and better get to the correct answer. In this work training data was augmented with reasoning for math and code execution; Google was involved so they could train a 137b-param model with this kind of data.

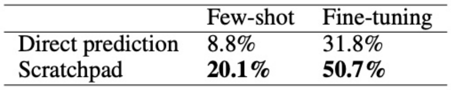


```
Input:
Evaluate -7*x**2 + 7*x + 5 at x = 1

Target:
<scratch>
-7*x**2: -7
7*x: 7
5: 5
</scratch>
total: 5
```

Figure 4: Example of polynomial evaluation with a scratchpad. Each term in the polynomial is computed separately and then added.

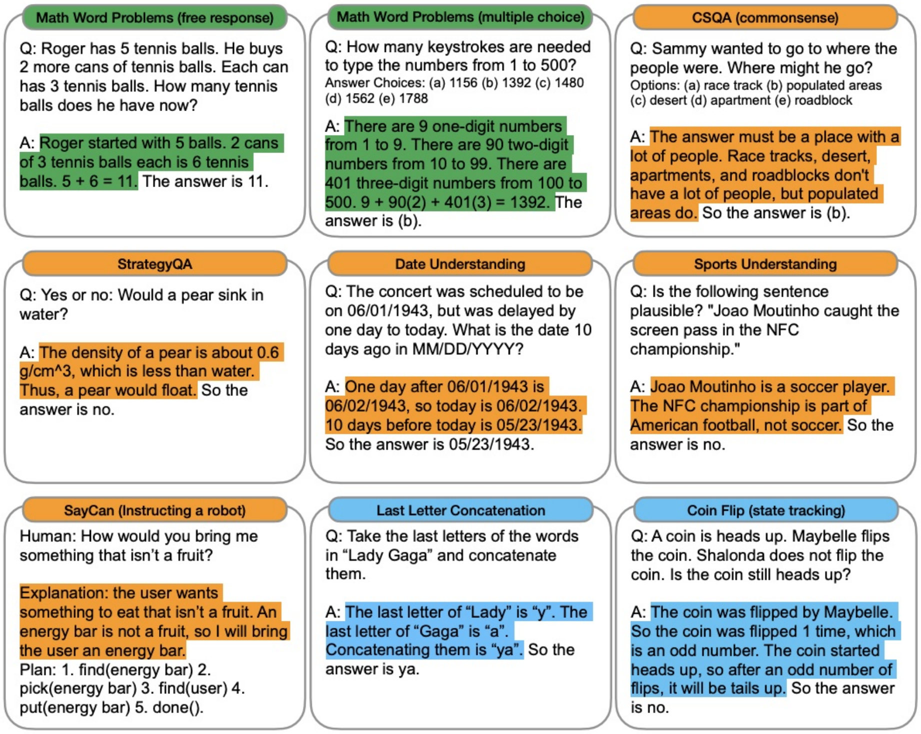
Table 1: Results for polynomial evaluation task. Scratchpad outperforms direct prediction whether using fine-tuning or few-shot.



	Few-shot	Fine-tuning
Direct prediction	8.8%	31.8%
Scratchpad	**20.1%**	**50.7%**

3.2 Chain-of-Thought

The idea behind chain of thought [19] in 2022 was to encourage the output of a prompt to also include the reasoning for that prompt. Because only a few shots are needed, the authors hand-wrote example thought processes for a variety of semantics, math, and common sense tasks. With sufficiently large models, the pattern was followed and scores strongly improved.



Math Word Problems (free response)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500? Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go? Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation

Q: Take the last letters of the words in "Lady Gaga" and concatenate them.

A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

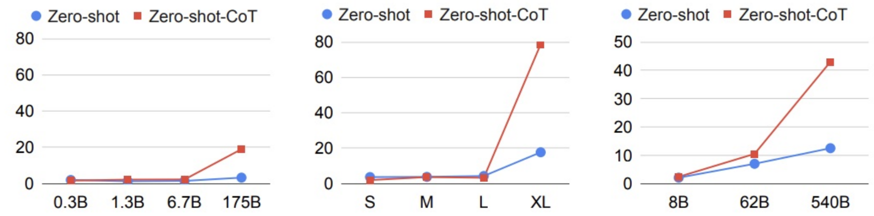
Coin Flip (state tracking)

Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

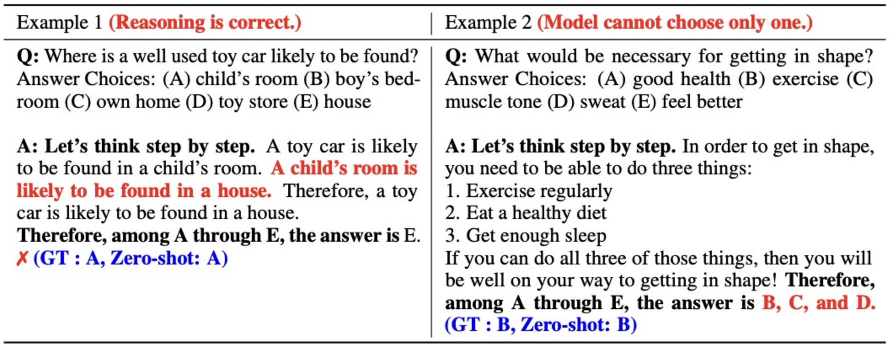
But of course we don't have to stop there, since these large models have zero-shot capability. Rather than provide exemplars, in [5], the magic command ”let's take it step by step” yields chain of thought automatically.



(a) MultiArith on Original GPT-3 (b) MultiArith on Instruct GPT-3 (c) GMS8K on PaLM

Figure 3: Model scale study with various types of models. S: text-ada-001, M: text-babbage-001, L: text-curie-001, XL: text-davinci-002. See Appendix A.3 and E for the detail.

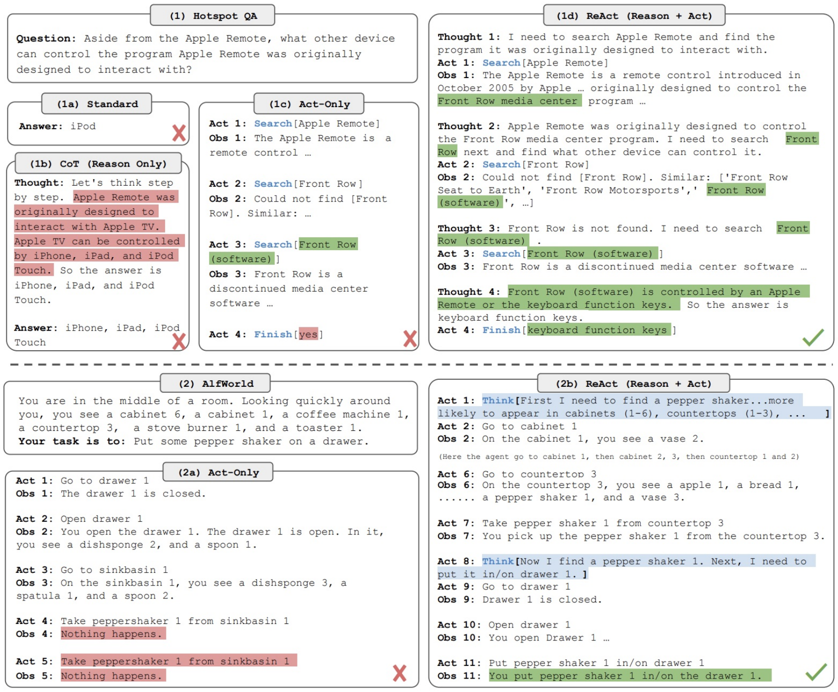
Table 3: Examples generated by Zero-Shot-CoT on CommonsenseQA for Error Analysis.



Example 1 **(Reasoning is correct.)**	Example 2 **(Model cannot choose only one.)**
Q: Where is a well used toy car likely to be found? Answer Choices: (A) child’s room (B) boy’s bedroom (C) own home (D) toy store (E) house **A: Let’s think step by step.** A toy car is likely to be found in a child’s room. **A child’s room is likely to be found in a house.** Therefore, a toy car is likely to be found in a house. **Therefore, among A through E, the answer is** E. ✗ **(GT : A, Zero-shot: A)**	**Q:** What would be necessary for getting in shape? Answer Choices: (A) good health (B) exercise (C) muscle tone (D) sweat (E) feel better **A: Let’s think step by step.** In order to get in shape, you need to be able to do three things: 1. Exercise regularly 2. Eat a healthy diet 3. Get enough sleep If you can do all three of those things, then you will be well on your way to getting in shape! **Therefore, among A through E, the answer is B, C, and D.** **(GT : B, Zero-shot: B)**

3.3 Model as Feedback

If an emergent behavior is thinking, what about criticizing others’ thinking? Or even your own thinking? The ReACT idea was to do just that; after producing an outcome, the outcome is fed back into an LLM and a next step is produced. This is particularly useful if you have tooling enabled but could in general be used in a loop for reasoning.



```
(1) Hotspot QA
Question: Aside from the Apple Remote, what other device
can control the program Apple Remote was originally
designed to interact with?
```

```
(1a) Standard
Answer: iPod
```

```
(1b) CoT (Reason Only)
Thought: Let's think step
by step. Apple Remote was
originally designed to
interact with Apple TV.
Apple TV can be controlled
by iPhone, iPad, and iPod
Touch. So the answer is
iPhone, iPad, and iPod
Touch.

Answer: iPhone, iPad, iPod
Touch
```

```
(1c) Act-Only
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is  a
remote control …

Act 2: Search[Front Row]
Obs 2: Could not find [Front
Row]. Similar: …

Act 3: Search[Front Row
(software)]
Obs 3: Front Row is a
discontinued media center
software …

Act 4: Finish[yes]
```

```
(1d) ReAct (Reason + Act)
Thought 1: I need to search Apple Remote and find the
program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in
October 2005 by Apple … originally designed to control the
Front Row media center program …

Thought 2: Apple Remote was originally designed to control
the Front Row media center program. I need to search  Front
Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row
Seat to Earth', 'Front Row Motorsports',' Front Row
(software)', …]

Thought 3: Front Row is not found. I need to search  Front
Row (software) .
Act 3: Search[Front Row (software) ]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple
Remote or the keyboard function keys.  So the answer is
keyboard function keys.
Act 4: Finish[keyboard function keys ]
```

```
(2) AlfWorld
You are in the middle of a room. Looking quickly around
you, you see a cabinet 6, a cabinet 1, a coffee machine 1,
a countertop 3,  a stove burner 1, and a toaster 1.
Your task is to: Put some pepper shaker on a drawer.
```

```
(2a) Act-Only
Act 1: Go to drawer 1
Obs 1: The drawer 1 is closed.

Act 2: Open drawer 1
Obs 2: You open the drawer 1. The drawer 1 is open. In it,
you see a dishsponge 2, and a spoon 1.

Act 3: Go to sinkbasin 1
Obs 3: On the sinkbasin 1, you see a dishsponge 3, a
spatula 1, and a spoon 2.

Act 4: Take peppershaker 1 from sinkbasin 1
Obs 4: Nothing happens.

Act 5: Take peppershaker 1 from sinkbasin 1
Obs 5: Nothing happens.
```

```
(2b) ReAct (Reason + Act)
Act 1: Think[First I need to find a pepper shaker...more
likely to appear in cabinets (1-6), countertops (1-3), ...   ]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1,
...... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: Think[Now I find a pepper shaker 1. Next, I need to
put it in/on drawer 1. ]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.
```

This general idea is probably what's in the current ChatGPT o1, and is what underlies the idea of "Reflexion" (analyze output for mistakes, then remember your mistakes), and some software packages like DSPy (for improving prompts) and Textgrad (entirely LLM-based re-evaluation). There is lots more I don't know about and probably lots more to come.

References

[1] Abhimanyu Dubey et al. *The Llama 3 Herd of Models*. 2024. arXiv: `2407.21783 [cs.AI]`. URL: `https://arxiv.org/abs/2407.21783`.

[2] Dirk Groeneveld et al. "OLMo: Accelerating the Science of Language Models". In: *ArXiv* abs/2402.00838 (2024). URL: `https://api.semanticscholar.org/CorpusID:267365485`.

[3] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *CoRR* abs/1902.00751 (2019). arXiv: `1902.00751`. URL: `http://arxiv.org/abs/1902.00751`.

[4] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *CoRR* abs/2106.09685 (2021). arXiv: `2106.09685`. URL: `https://arxiv.org/abs/2106.09685`.

[5] Takeshi Kojima et al. "Large Language Models are Zero-Shot Reasoners". In: *ArXiv* abs/2205.11916 (2022). URL: `https://api.semanticscholar.org/CorpusID:249017743`.

[6] Jannik Kossen, Yarin Gal, and Tom Rainforth. "In-Context Learning Learns Label Relationships but Is Not Conventional Learning". In: *The Twelfth International Conference on Learning Representations.* 2024. URL: `https://openreview.net/forum?id=YPIA7bgd5y`.

[7] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: `10.18653/v1/2021.acl-long.353`. URL: `https://aclanthology.org/2021.acl-long.353`.

[8] Sewon Min et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.* Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11048–11064. DOI: `10.18653/v1/2022.emnlp-main.759`. URL: `https://aclanthology.org/2022.emnlp-main.759`.

[9] Maxwell Nye et al. "Show Your Work: Scratchpads for Intermediate Computation with Language Models". In: *ArXiv* abs/2112.00114 (2021). URL: `https://api.semanticscholar.org/CorpusID:244773644`.

[10] OpenAI et al. *GPT-4 Technical Report.* 2024. arXiv: `2303.08774 [cs.CL]`. URL: `https://arxiv.org/abs/2303.08774`.

[11] Yasaman Razeghi et al. "Impact of Pretraining Term Frequencies on Few-Shot Numerical Reasoning". In: *Findings of the Association for Computational Linguistics: EMNLP 2022.* Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 840–854. DOI: `10.18653/v1/2022.findings-emnlp.59`. URL: `https://aclanthology.org/2022.findings-emnlp.59`.

[12] Laria Reynolds and Kyle McDonell. "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm". In: *CoRR* abs/2102.07350 (2021). arXiv: `2102.07350`. URL: `https://arxiv.org/abs/2102.07350`.

[13] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations.* 2022. URL: `https://openreview.net/forum?id=9Vrb9D0WI4`.

[14] Gemini Team et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.* 2024. arXiv: `2403.05530 [cs.CL]`. URL: `https://arxiv.org/abs/2403.05530`.

[15] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models.* 2024. arXiv: `2312.11805 [cs.CL]`. URL: `https://arxiv.org/abs/2312.11805`.

[16] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models.* 2023. arXiv: `2307.09288 [cs.CL]`. URL: `https://arxiv.org/abs/2307.09288`.

[17] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: `2302.13971 [cs.CL]`. URL: `https://arxiv.org/abs/2302.13971`.

[18] Albert Webson and Ellie Pavlick. "Do Prompt-Based Models Really Understand the Meaning of Their Prompts?" In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 2300–2344. DOI: `10.18653/v1/2022.naacl-main.167`. URL: `https://aclanthology.org/2022.naacl-main.167`.

[19] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2024. ISBN: 9781713871088.