LLM Emergent Abilities

22-808: Generative models
Sharif University of Technology
Fall 2025

Fatemeh Seyyedsalehi

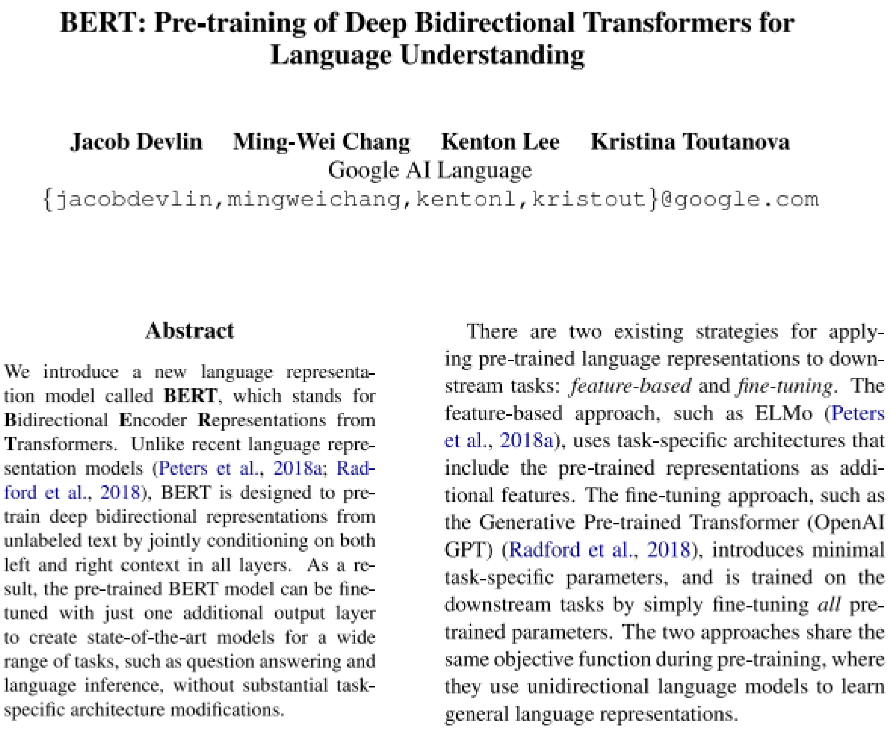
Paradigm shift

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

Encoder Only Model: Bert



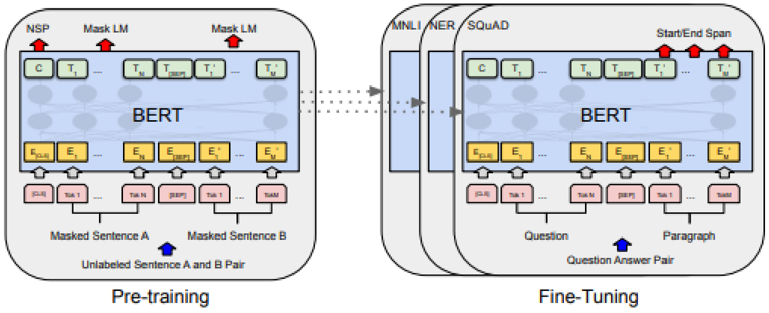
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

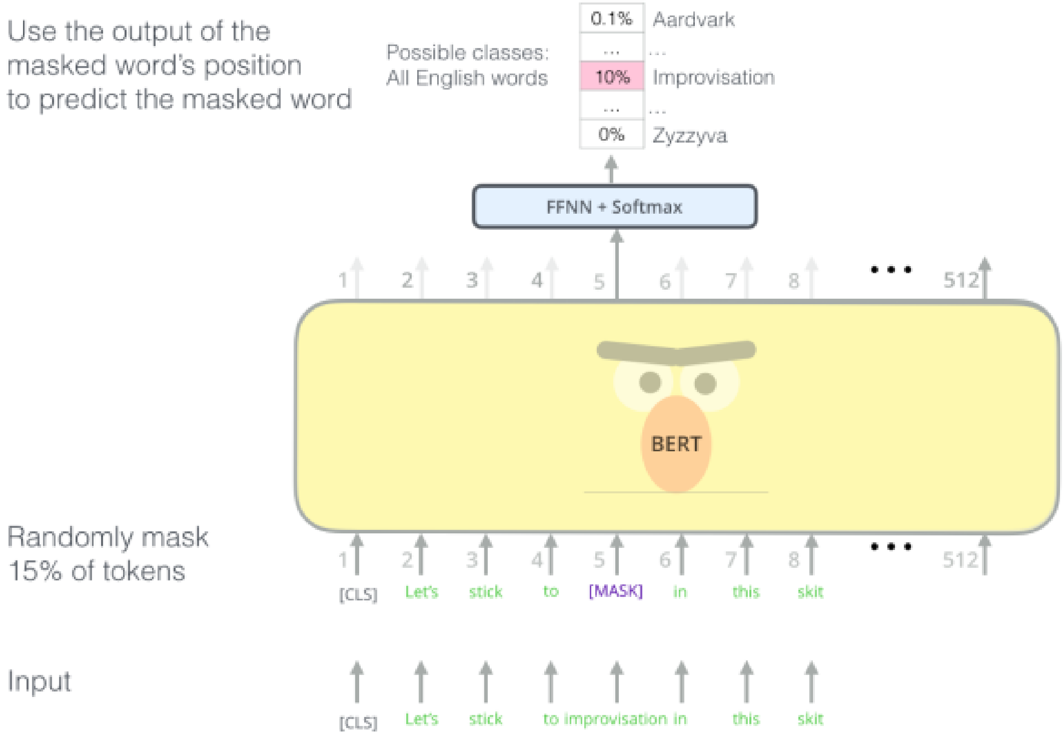
Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.



Pretraining Task 1: masked words



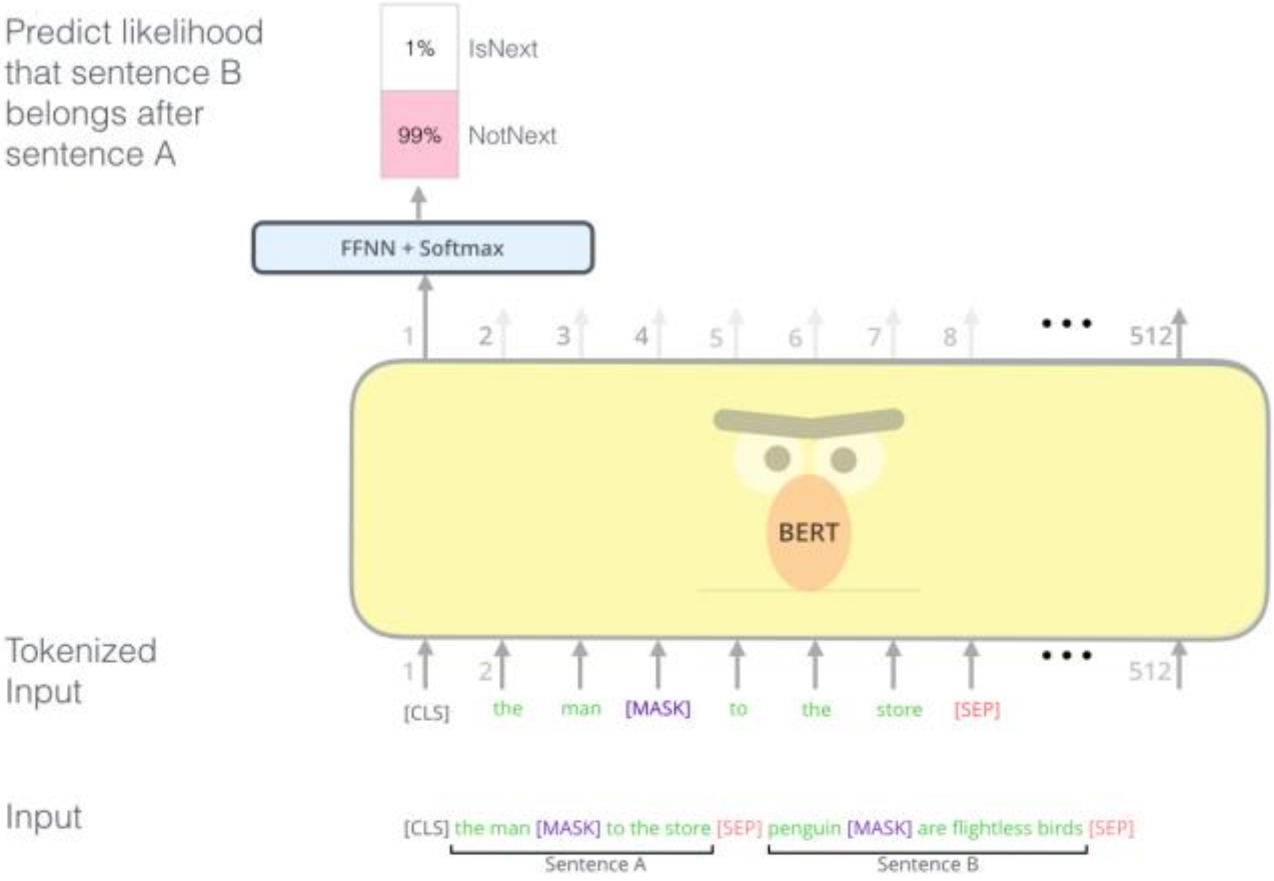
Use the output of the masked word's position to predict the masked word

Randomly mask 15% of tokens

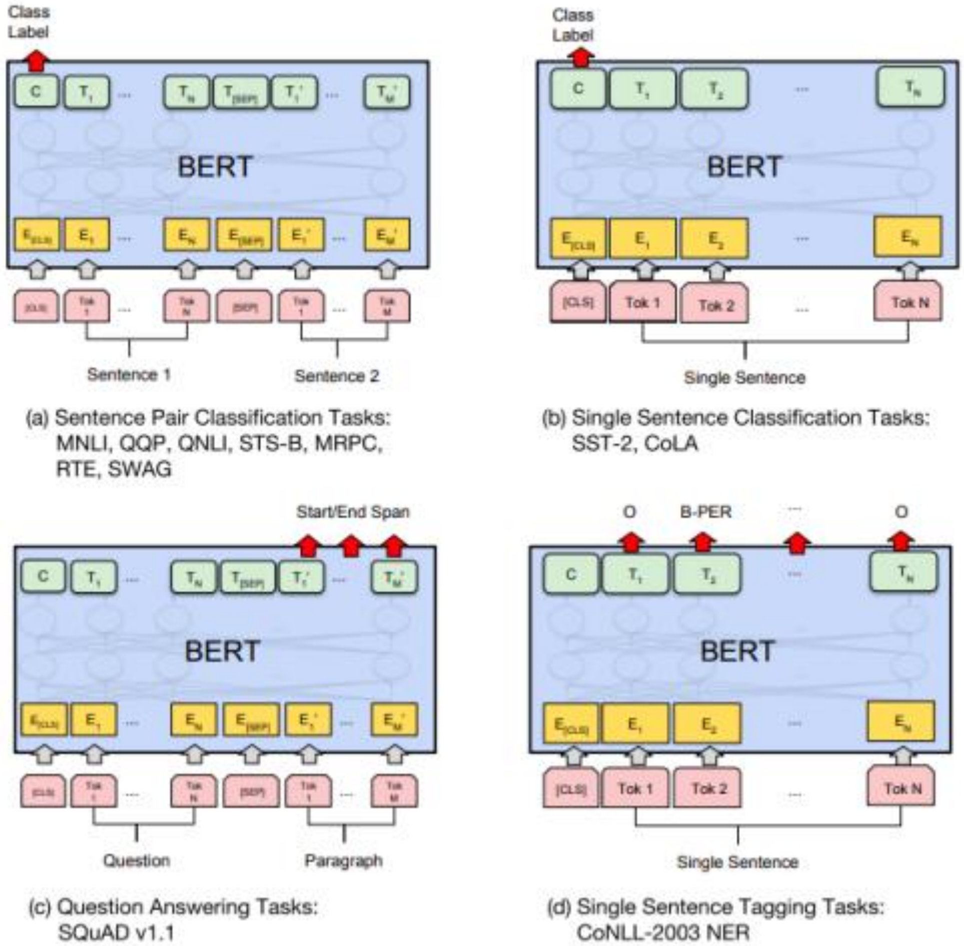
Input

Out of this 15%,
80% are [Mask],
10% random words
10% original words

Pretraining Task 2: two sentences


Predict likelihood that sentence B belongs after sentence A
1%
IsNext
99%
NotNext
FFNN + Softmax
1
2
3
4
5
6
7
8
...
512
BERT
Tokenized Input
1
2
512
[CLS]
the
man
[MASK]
to
the
store
[SEP]
...
Input
[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]
Sentence A
Sentence B

Fine-tuning BERT for other specific tasks



(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

(b) Single Sentence Classification Tasks:
SST-2, CoLA

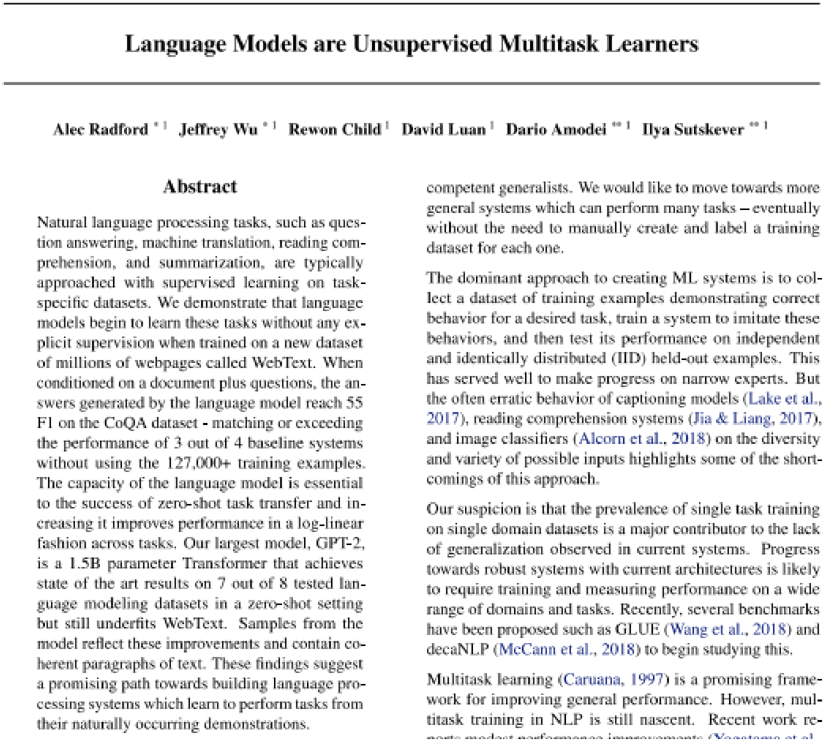
(c) Question Answering Tasks:
SQuAD v1.1

(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Traditional way: Fine-tuning

- Fine-tune the parameters of the pre-trained model for a specific downstream task using a large (thousands to hundreds of thousands) corpus of labeled data.
- Keep training the model via repeated gradient updates.
- Strong performance on many benchmarks.
- Need a new large dataset for each task.
- Potential for poor out-of-distribution generalization.
- Potential to explore spurious features of the data

Decoder Only Model: GPT2



Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al.,

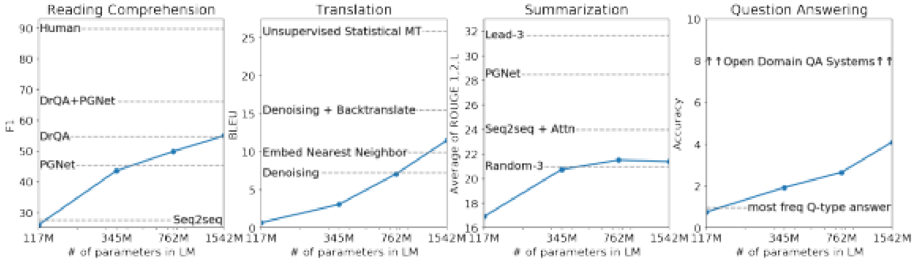
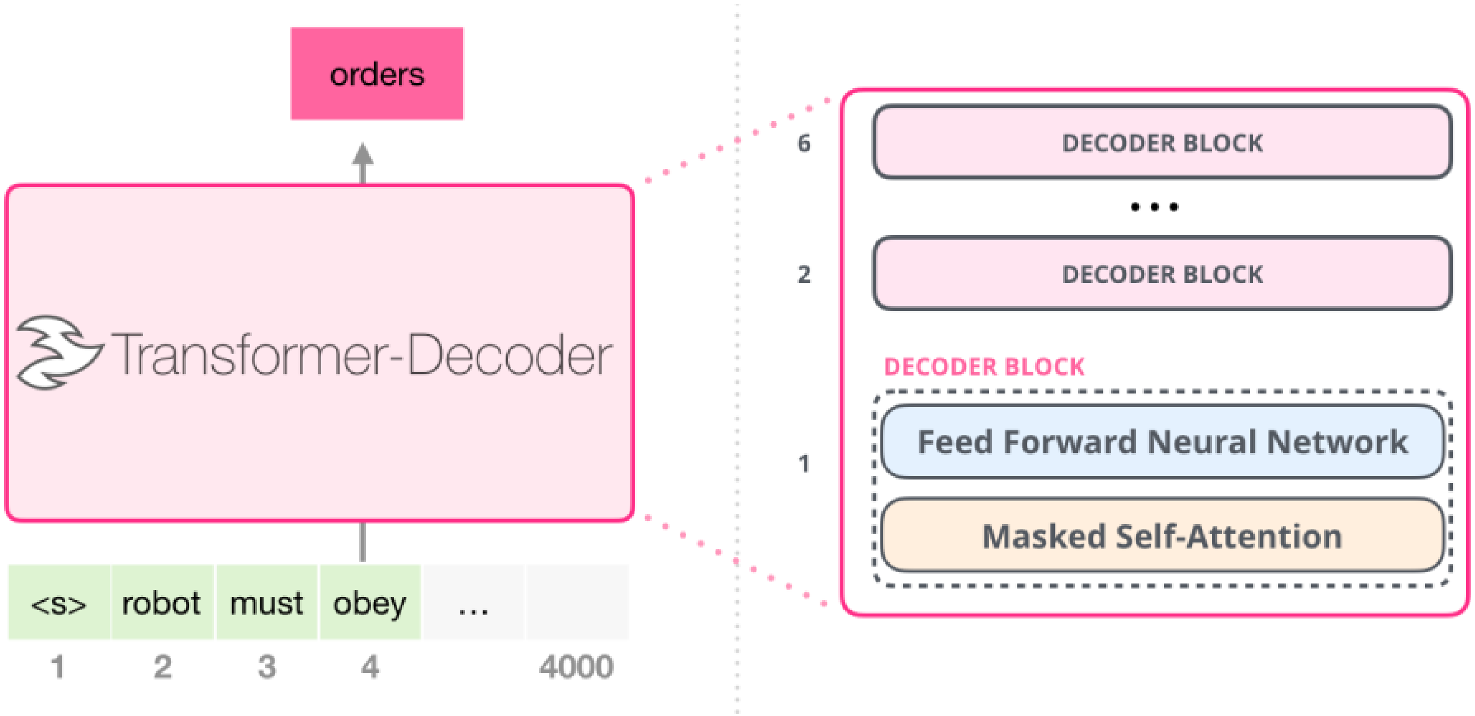


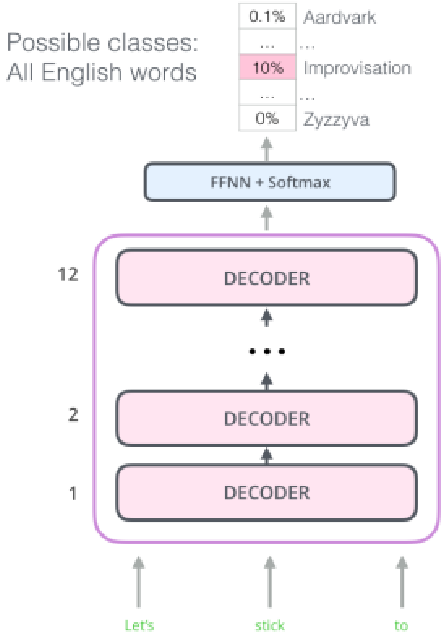
Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

Pretraining Task: next token prediction



Re-use previous computation results: at any step, only need to results of q, k , v related to the new output word, no need to re-compute the others. Additional computation is linear, instead of quadratic.

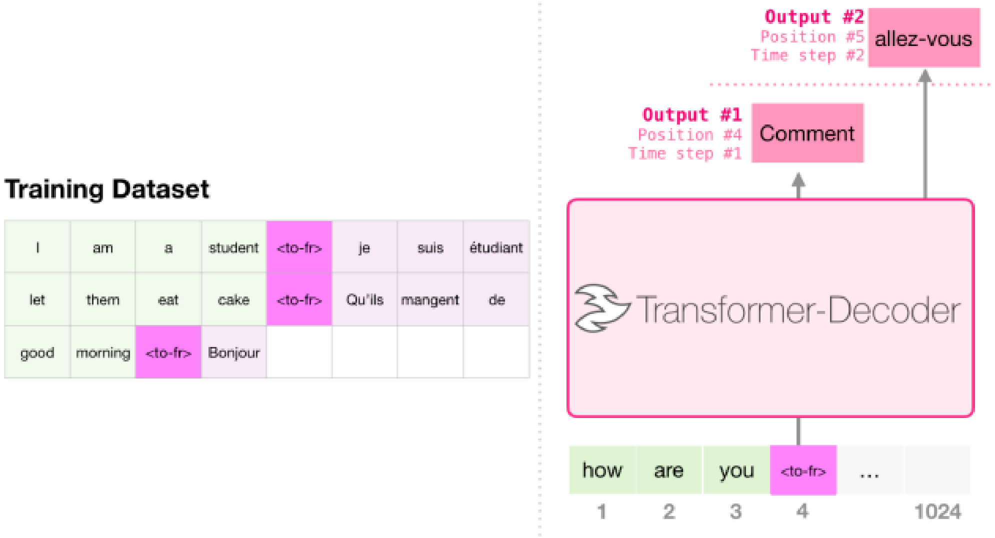
Pretraining Data



GPT-2 uses unsupervised learning approach to training the language model.

- "Our approach motivates building as large and diverse a dataset as possible in order to collect natural language demonstrations of tasks in as varied of domains and contexts as possible."
- Over 8 million documents for a total of 40 GB of text.

Downstream Tasks



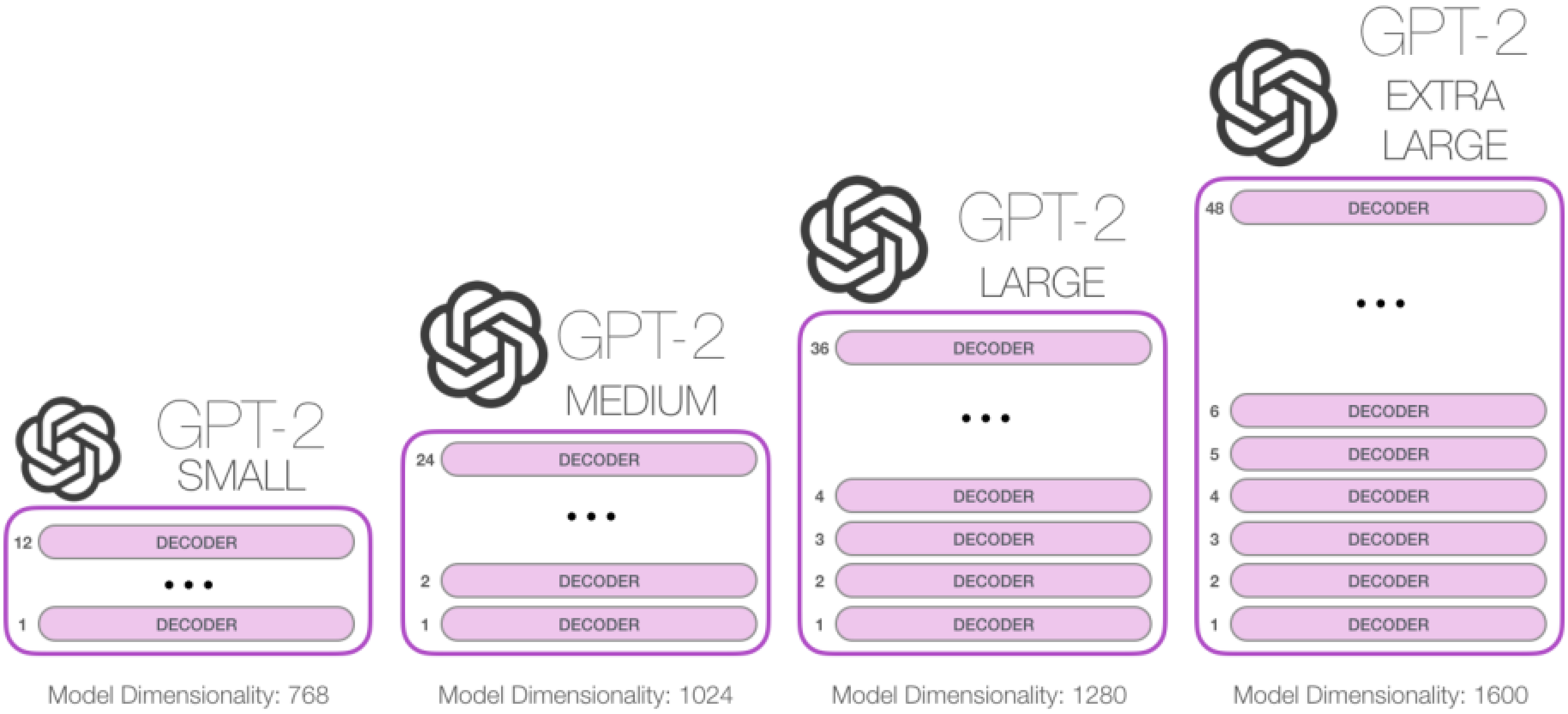
Training Dataset

I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				

There is no custom training for GPT-2, no separation of pre-training and fine-tuning like BERT.

- Translation
- QA
- Summarization
- Reading Comprehension
- Language Modeling

What makes it work? Scaling



117M parameters	345M	762M	1542M

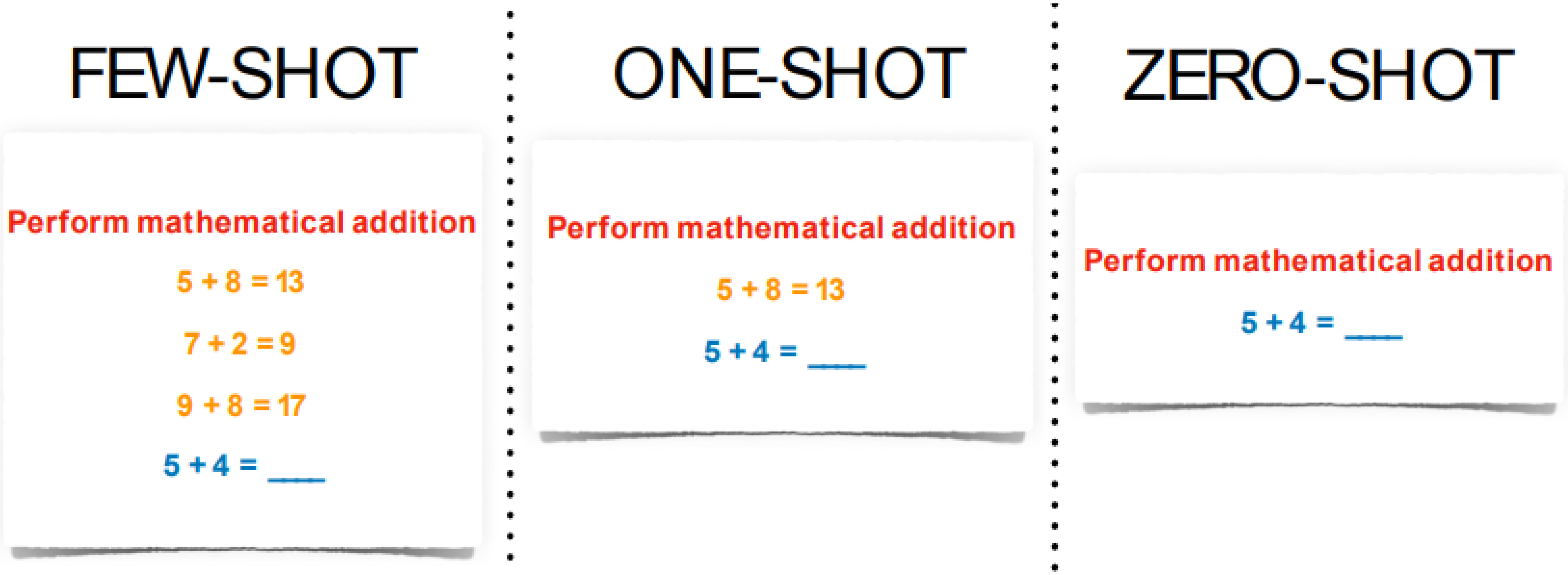
Terminology

- In-context learning: A frozen LM performs a task only by conditioning on the prompt text.
- Few-shot in-context learning: (1) The prompt includes examples of the intended behavior, and (2) no examples of the intended behavior were seen in training.
- Zero-shot in-context learning: (1) The prompt includes no examples of the intended behavior (but it can contain other instructions), and (2) no examples of the intended behavior were seen in training.
- Emergence: when quantitative changes in a system result in qualitative changes in behavior.
- Emergent behaviors: abilities that larger models have and smaller models don’t

In-context learning

- No training or optimization of the model parameters in the “adaptation step”
- Simply give the model a task description as well as none/one/few examples as the input at inference time.
- No gradient updates are performed.

Example



FEW-SHOT

Perform mathematical addition

5 + 8 = 13

7 + 2 = 9

9 + 8 = 17

5 + 4 = ____

ONE-SHOT

Perform mathematical addition

5 + 8 = 13

5 + 4 = ____

ZERO-SHOT

Perform mathematical addition

5 + 4 = ____

Contrast

- Zero-shot in-context learning

  - Provides maximum convenience (no task-specific example needed)

  - Potential for robustness

  - Potential for avoidance of spurious correlations

  - Most challenging

  - Even for humans, it is often hard to understand a task without an example.

- Few-shot in-context learning

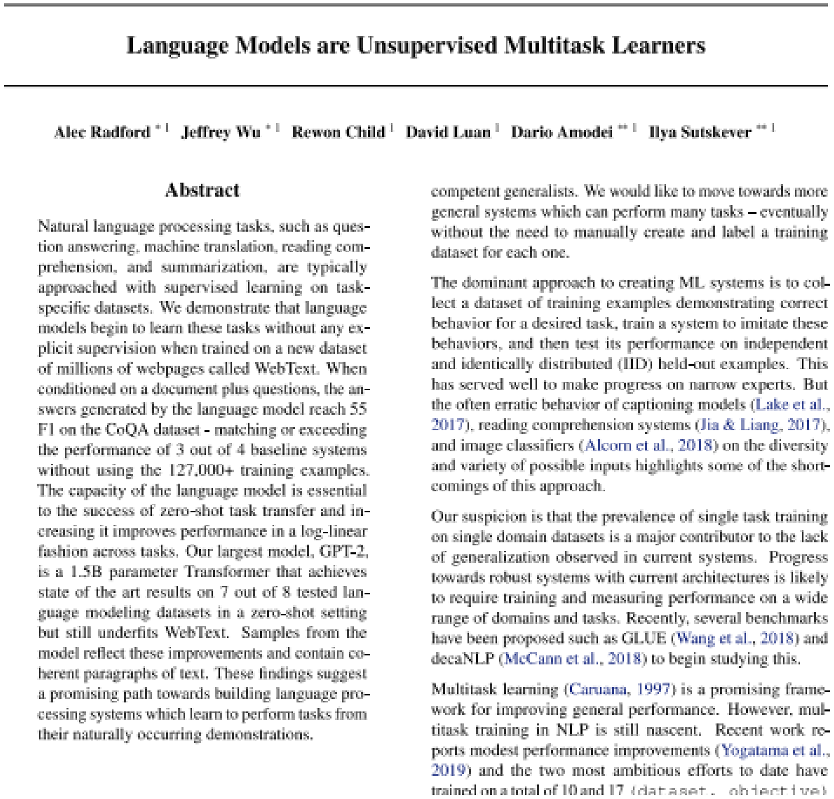
  - Major reduction in the need for task-specific data.

  - Reduced potential to learn an overly narrow distribution from a large but narrow fine-tuning dataset.

  - Still not as good as the fine-tuning SOTA, but competitive (GPT-3).

  - Still need a few task-specific data

GPT2 – GPT3



Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Abstract

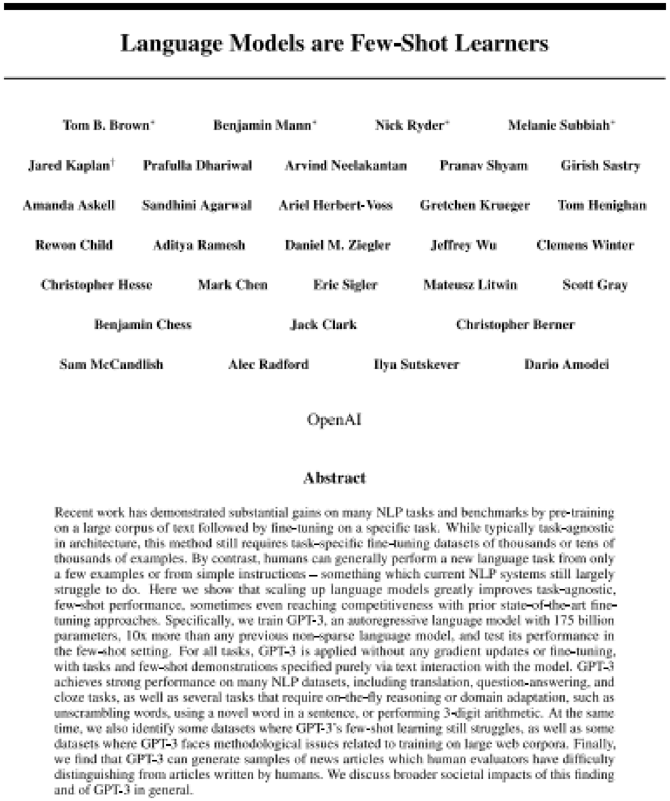
Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have trained on a total of 10 and 17 (`dataset, objective`)



Language Models are Few-Shot Learners

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 Dario Amodei

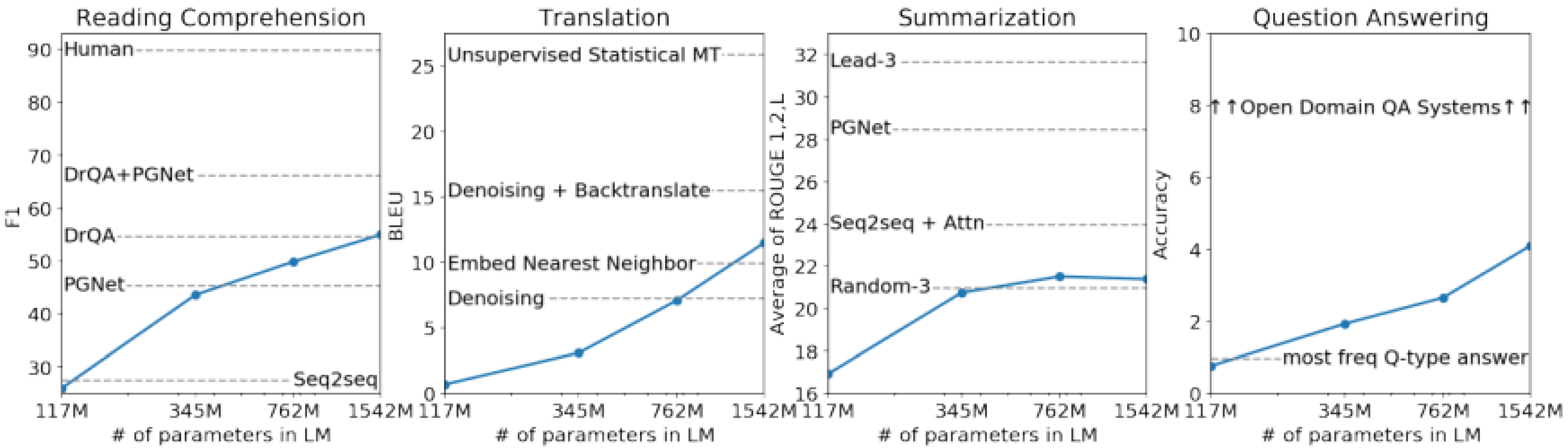
OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

GPT2 – GPT3

- The authors “demonstrate that language models begin to learn [question answering, machine translation, reading comprehension, and summarization] tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText.”



GPT2 – GPT3

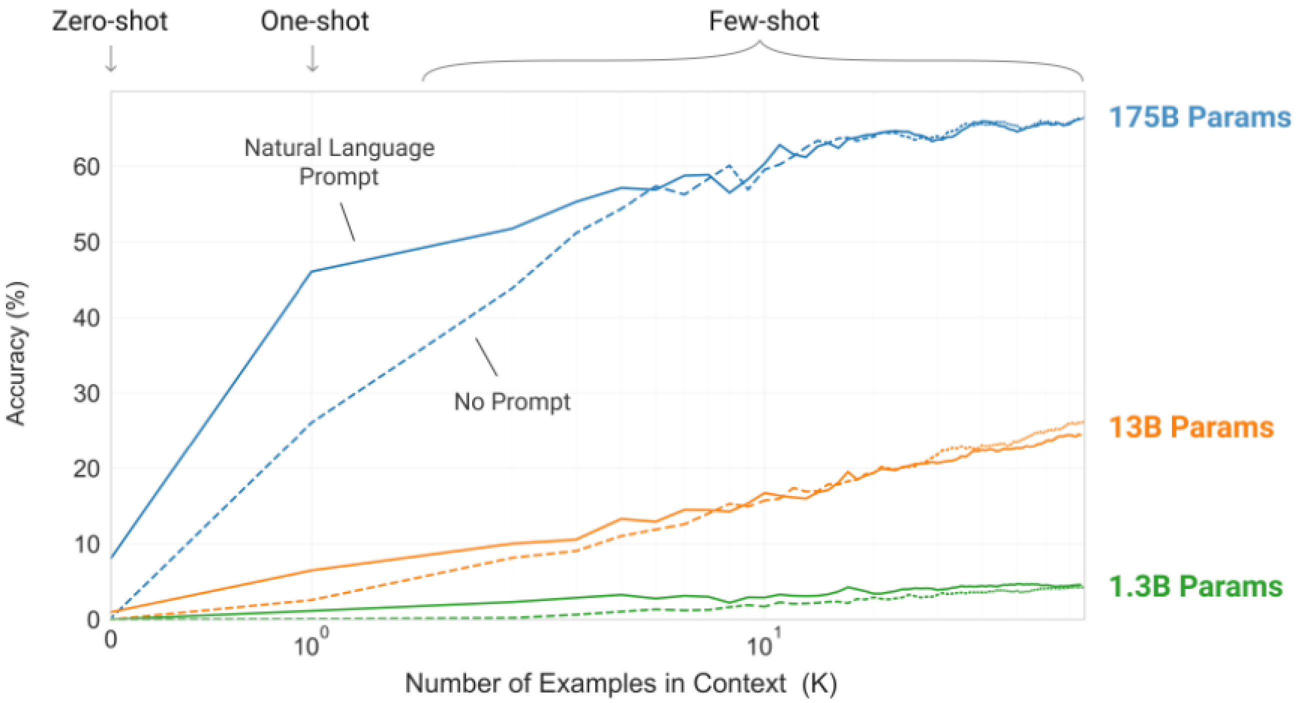


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

GPT2 – GPT3

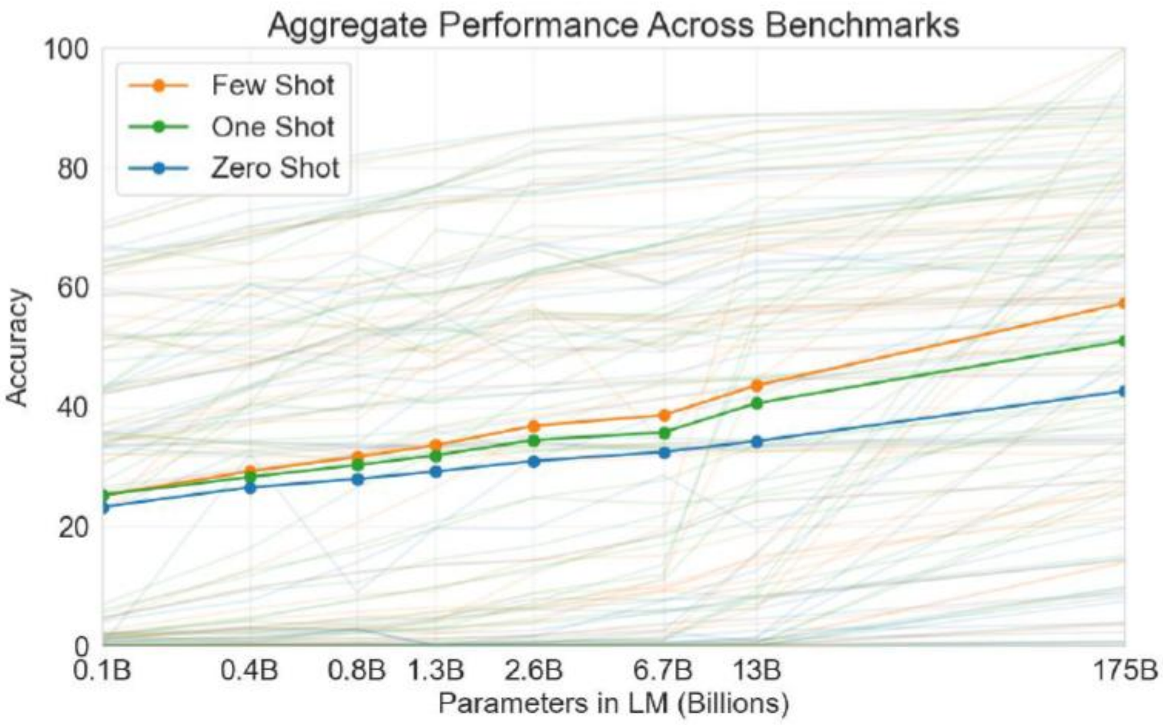
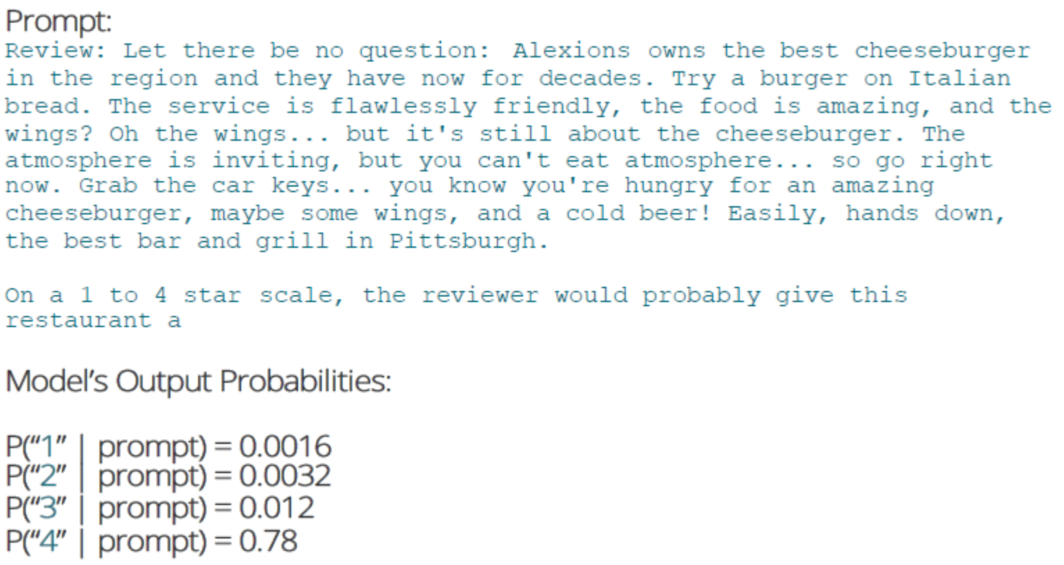


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Zero-Shot Learning (Sentiment Classification)



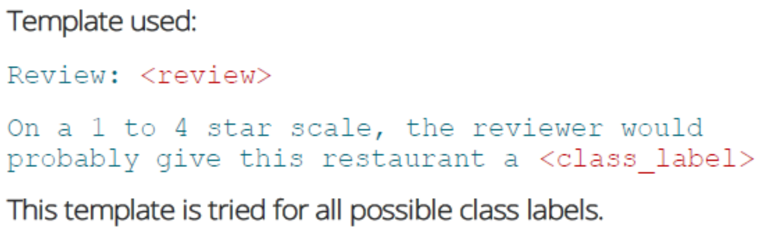
Prompt:

Review: Let there be no question: Alexions owns the best cheeseburger in the region and they have now for decades. Try a burger on Italian bread. The service is flawlessly friendly, the food is amazing, and the wings? Oh the wings... but it's still about the cheeseburger. The atmosphere is inviting, but you can't eat atmosphere... so go right now. Grab the car keys... you know you're hungry for an amazing cheeseburger, maybe some wings, and a cold beer! Easily, hands down, the best bar and grill in Pittsburgh.

On a 1 to 4 star scale, the reviewer would probably give this restaurant a

Model's Output Probabilities:

P("1" | prompt) = 0.0016
P("2" | prompt) = 0.0032
P("3" | prompt) = 0.012
P("4" | prompt) = 0.78



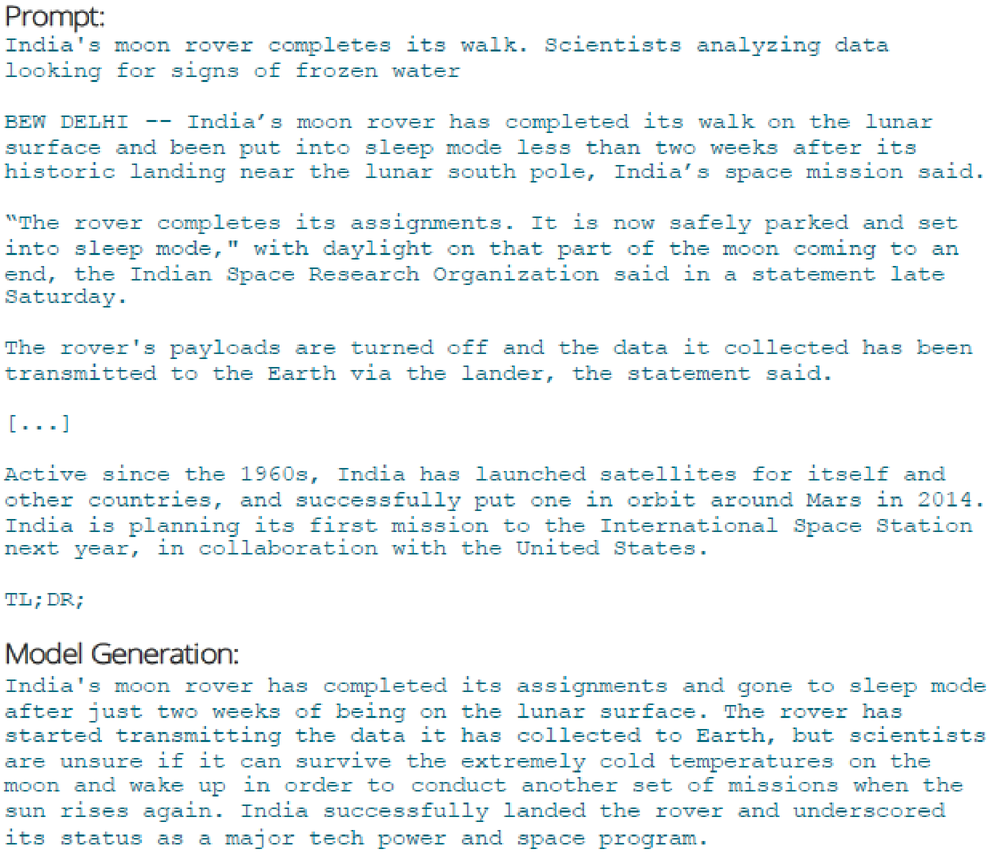
Template used:

Review: <review>

On a 1 to 4 star scale, the reviewer would probably give this restaurant a <class_label>

This template is tried for all possible class labels.

Zero-Shot Learning (Summarization)



Prompt:

India's moon rover completes its walk. Scientists analyzing data looking for signs of frozen water

BEW DELHI -- India’s moon rover has completed its walk on the lunar surface and been put into sleep mode less than two weeks after its historic landing near the lunar south pole, India’s space mission said.

“The rover completes its assignments. It is now safely parked and set into sleep mode," with daylight on that part of the moon coming to an end, the Indian Space Research Organization said in a statement late Saturday.

The rover's payloads are turned off and the data it collected has been transmitted to the Earth via the lander, the statement said.

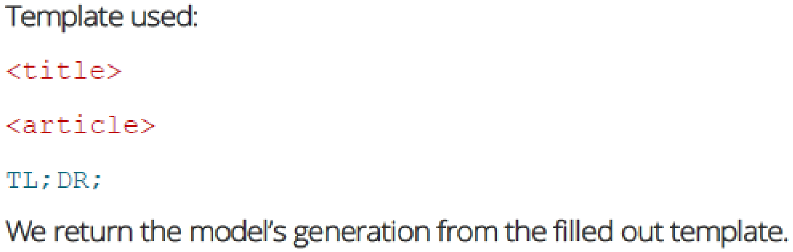
[...]

Active since the 1960s, India has launched satellites for itself and other countries, and successfully put one in orbit around Mars in 2014. India is planning its first mission to the International Space Station next year, in collaboration with the United States.

TL;DR;

Model Generation:

India's moon rover has completed its assignments and gone to sleep mode after just two weeks of being on the lunar surface. The rover has started transmitting the data it has collected to Earth, but scientists are unsure if it can survive the extremely cold temperatures on the moon and wake up in order to conduct another set of missions when the sun rises again. India successfully landed the rover and underscored its status as a major tech power and space program.



Template used:

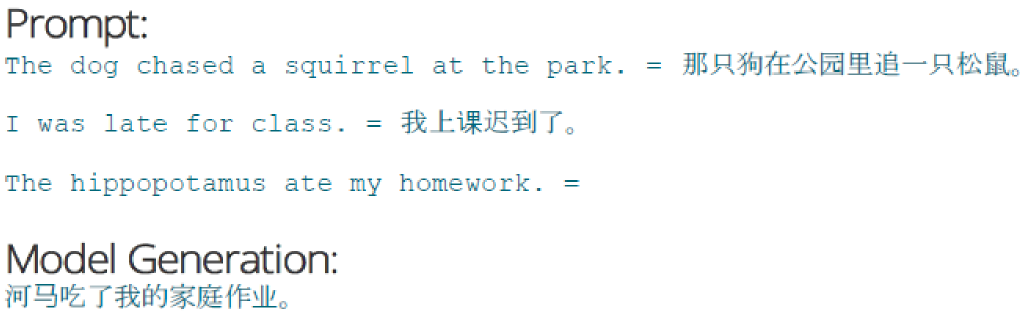
<title>

<article>

TL;DR;

We return the model’s generation from the filled out template.

Few-Shot Learning (Machine Translation)



Prompt:

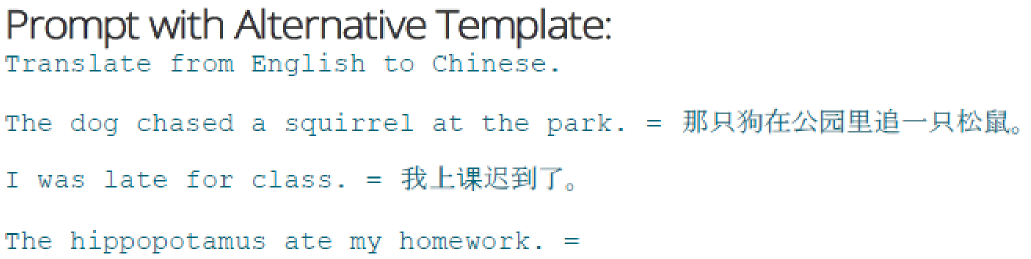
```
The dog chased a squirrel at the park. = 那只狗在公园里追一只松鼠。

I was late for class. = 我上课迟到了。

The hippopotamus ate my homework. =
```

Model Generation:

河马吃了我的家庭作业。



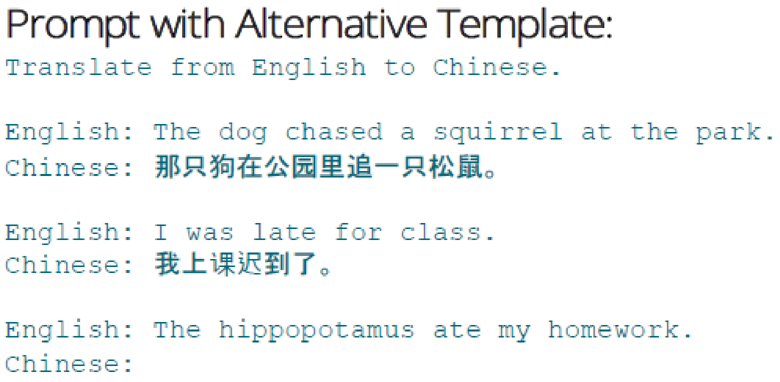
Prompt with Alternative Template:

```
Translate from English to Chinese.

The dog chased a squirrel at the park. = 那只狗在公园里追一只松鼠。

I was late for class. = 我上课迟到了。

The hippopotamus ate my homework. =
```



Prompt with Alternative Template:

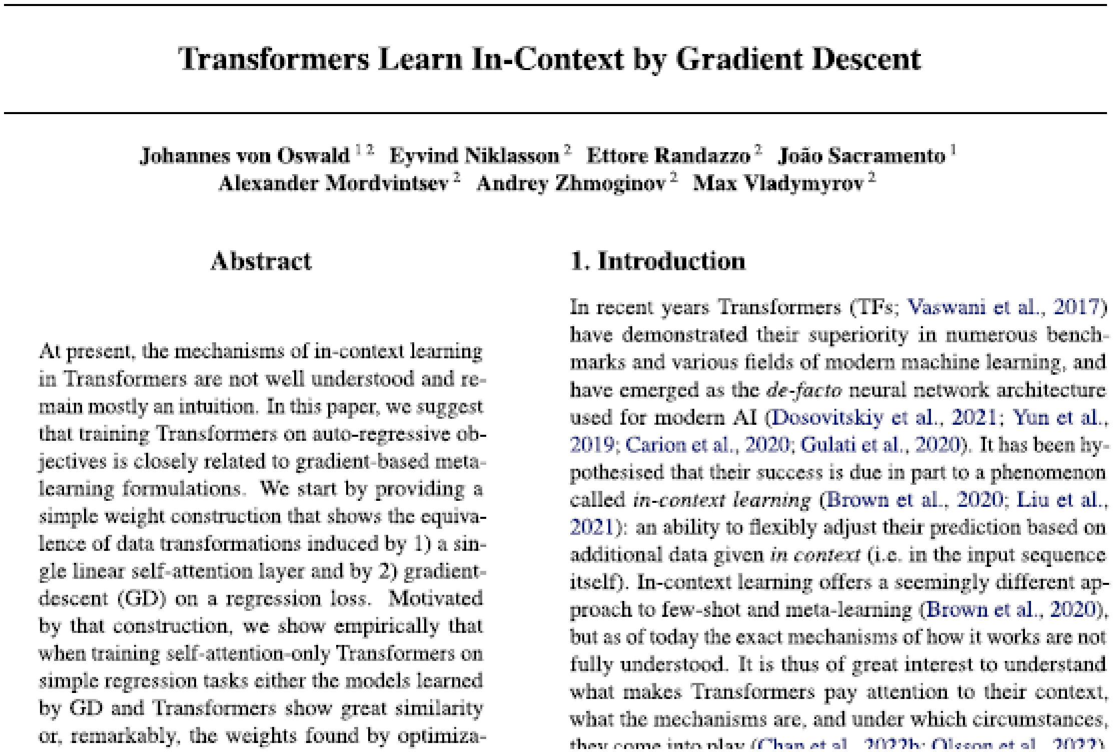
```
Translate from English to Chinese.

English: The dog chased a squirrel at the park.
Chinese: 那只狗在公园里追一只松鼠。

English: I was late for class.
Chinese: 我上课迟到了。

English: The hippopotamus ate my homework.
Chinese:
```

Why does in-context learning work



Transformers Learn In-Context by Gradient Descent

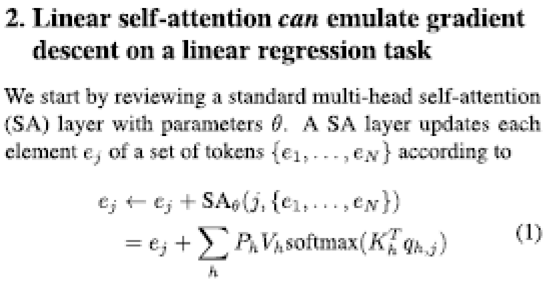
Johannes von Oswald [1 2] Eyvind Niklasson [2] Ettore Randazzo [2] João Sacramento [1]
Alexander Mordvintsev [2] Andrey Zhmoginov [2] Max Vladymyrov [2]

Abstract

At present, the mechanisms of in-context learning in Transformers are not well understood and remain mostly an intuition. In this paper, we suggest that training Transformers on auto-regressive objectives is closely related to gradient-based meta-learning formulations. We start by providing a simple weight construction that shows the equivalence of data transformations induced by 1) a single linear self-attention layer and by 2) gradient-descent (GD) on a regression loss. Motivated by that construction, we show empirically that when training self-attention-only Transformers on simple regression tasks either the models learned by GD and Transformers show great similarity or, remarkably, the weights found by optimiza-

1. Introduction

In recent years Transformers (TFs; Vaswani et al., 2017) have demonstrated their superiority in numerous benchmarks and various fields of modern machine learning, and have emerged as the *de-facto* neural network architecture used for modern AI (Dosovitskiy et al., 2021; Yun et al., 2019; Carion et al., 2020; Gulati et al., 2020). It has been hypothesised that their success is due in part to a phenomenon called *in-context learning* (Brown et al., 2020; Liu et al., 2021): an ability to flexibly adjust their prediction based on additional data given *in context* (i.e. in the input sequence itself). In-context learning offers a seemingly different approach to few-shot and meta-learning (Brown et al., 2020), but as of today the exact mechanisms of how it works are not fully understood. It is thus of great interest to understand what makes Transformers pay attention to their context, what the mechanisms are, and under which circumstances, they come into play (Chan et al., 2022b; Olsson et al., 2022)

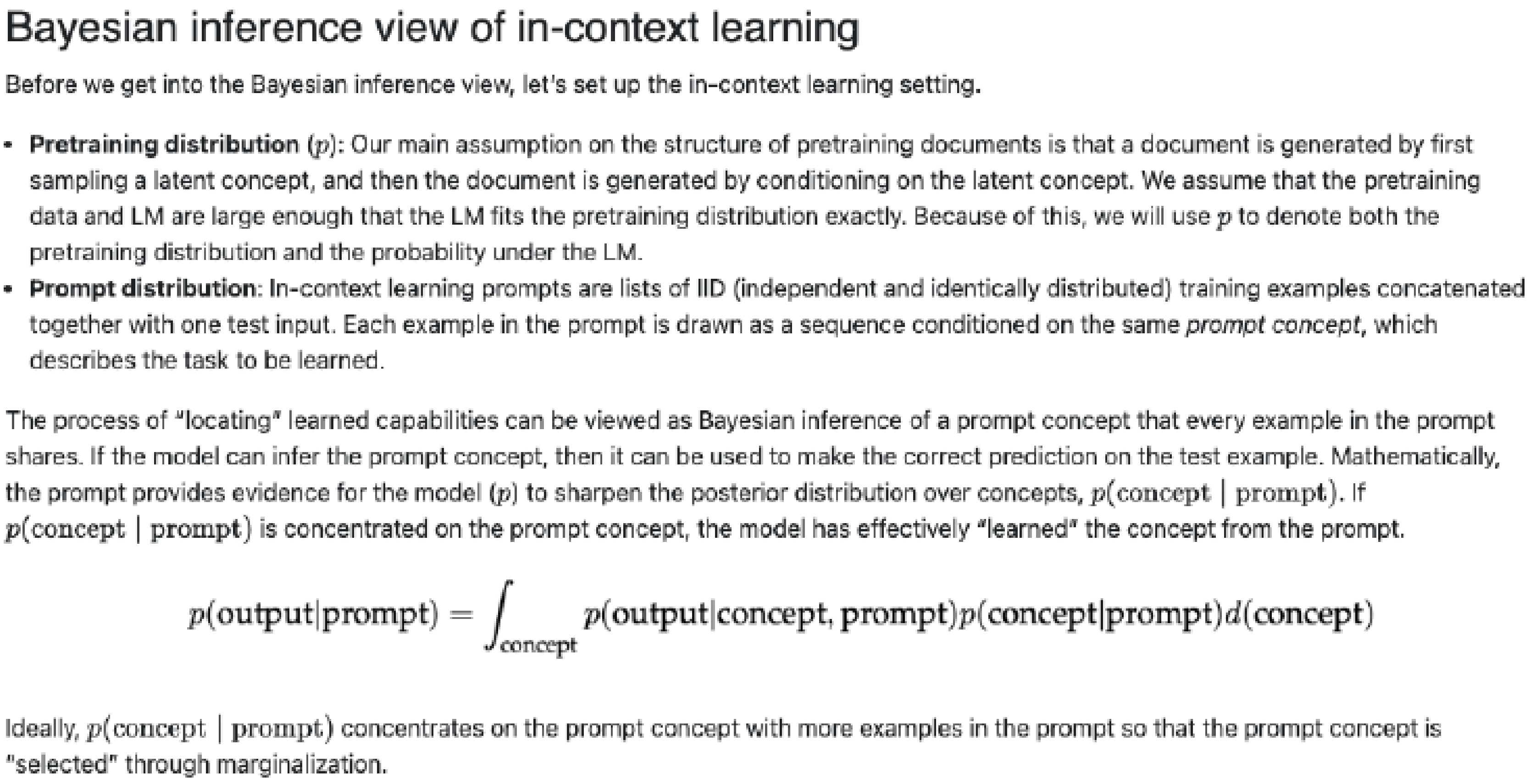


2. Linear self-attention *can* emulate gradient descent on a linear regression task

We start by reviewing a standard multi-head self-attention (SA) layer with parameters θ. A SA layer updates each element e_j of a set of tokens $\{e_1, \dots, e_N\}$ according to

$$\begin{aligned} e_j &\leftarrow e_j + \text{SA}_\theta(j, \{e_1, \dots, e_N\}) \\ &= e_j + \sum_h P_h V_h \text{softmax}(K_h^T q_{h,j}) \end{aligned} \tag{1}$$

Why does in-context learning work



Bayesian inference view of in-context learning

Before we get into the Bayesian inference view, let's set up the in-context learning setting.

- **Pretraining distribution** (p): Our main assumption on the structure of pretraining documents is that a document is generated by first sampling a latent concept, and then the document is generated by conditioning on the latent concept. We assume that the pretraining data and LM are large enough that the LM fits the pretraining distribution exactly. Because of this, we will use p to denote both the pretraining distribution and the probability under the LM.
- **Prompt distribution**: In-context learning prompts are lists of IID (independent and identically distributed) training examples concatenated together with one test input. Each example in the prompt is drawn as a sequence conditioned on the same *prompt concept*, which describes the task to be learned.

The process of "locating" learned capabilities can be viewed as Bayesian inference of a prompt concept that every example in the prompt shares. If the model can infer the prompt concept, then it can be used to make the correct prediction on the test example. Mathematically, the prompt provides evidence for the model (p) to sharpen the posterior distribution over concepts, $p(\text{concept} \mid \text{prompt})$. If $p(\text{concept} \mid \text{prompt})$ is concentrated on the prompt concept, the model has effectively "learned" the concept from the prompt.

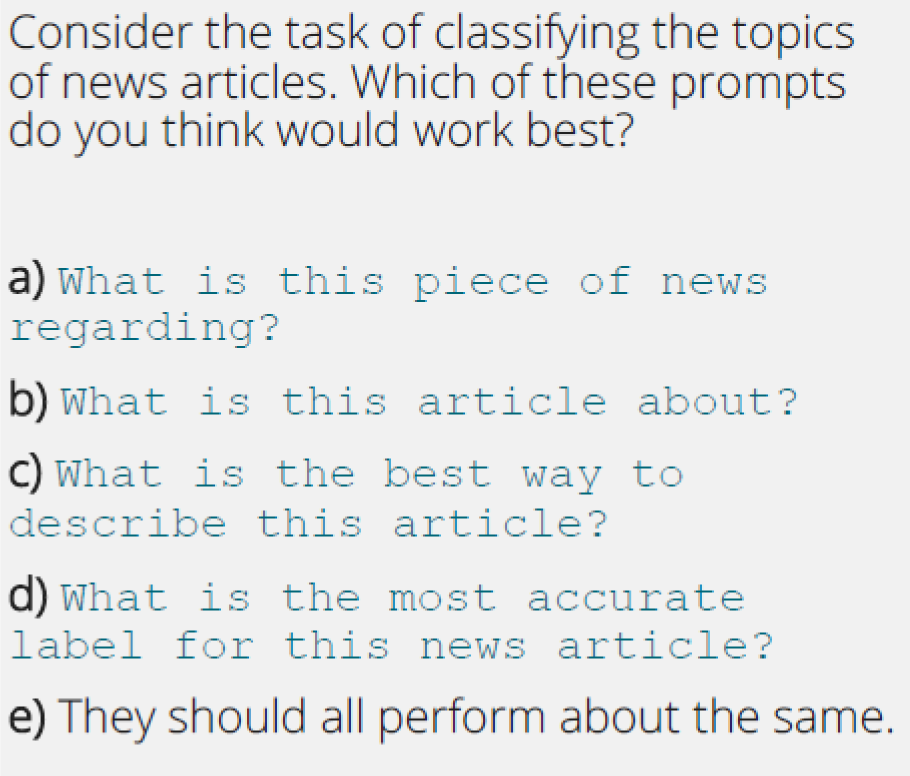
$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt})p(\text{concept}|\text{prompt})d(\text{concept})$$

Ideally, $p(\text{concept} \mid \text{prompt})$ concentrates on the prompt concept with more examples in the prompt so that the prompt concept is "selected" through marginalization.

Why does in-context learning work

- Instances of the task exist in the pre-training data.
  - Example: “TL;DR” is a well used string on Reddit.
  - Example: Translation data on the internet
- The few-shot examples “teach” the LLM what format to expect.

Prompt Engineering



Consider the task of classifying the topics of news articles. Which of these prompts do you think would work best?

a) What is this piece of news regarding?

b) What is this article about?

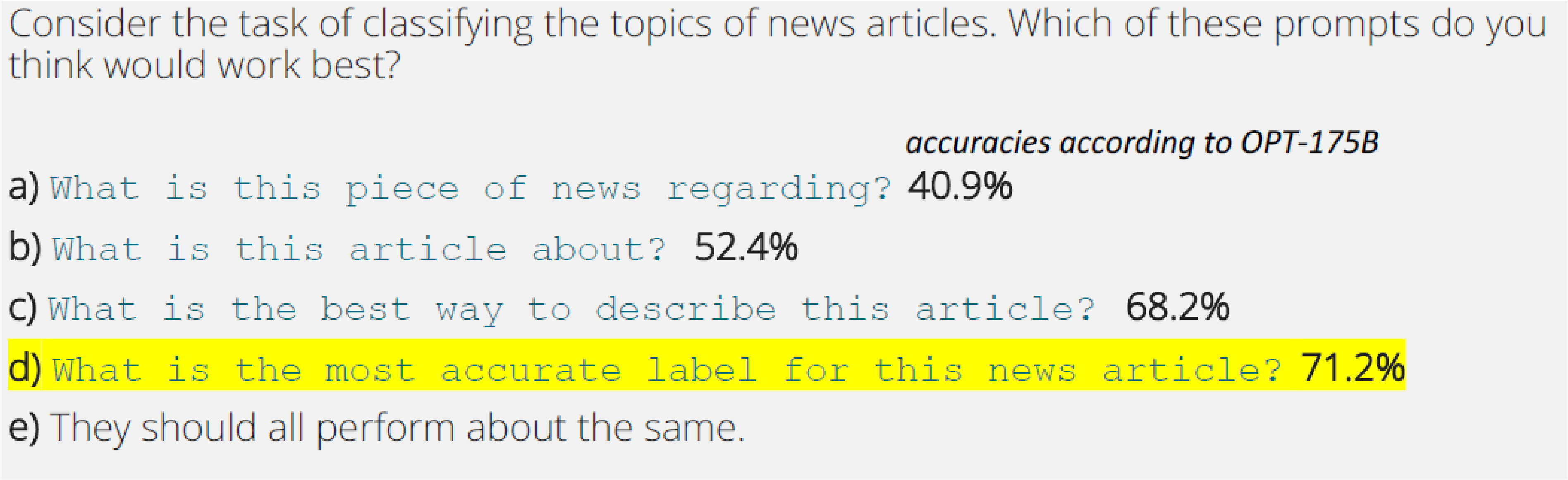
c) What is the best way to describe this article?

d) What is the most accurate label for this news article?

e) They should all perform about the same.

Prompt Engineering

- Prompts which are perceptually equivalent to humans can result in radically different performance.



Consider the task of classifying the topics of news articles. Which of these prompts do you think would work best?

accuracies according to OPT-175B

a) `What is this piece of news regarding?` 40.9%

b) `What is this article about?` 52.4%

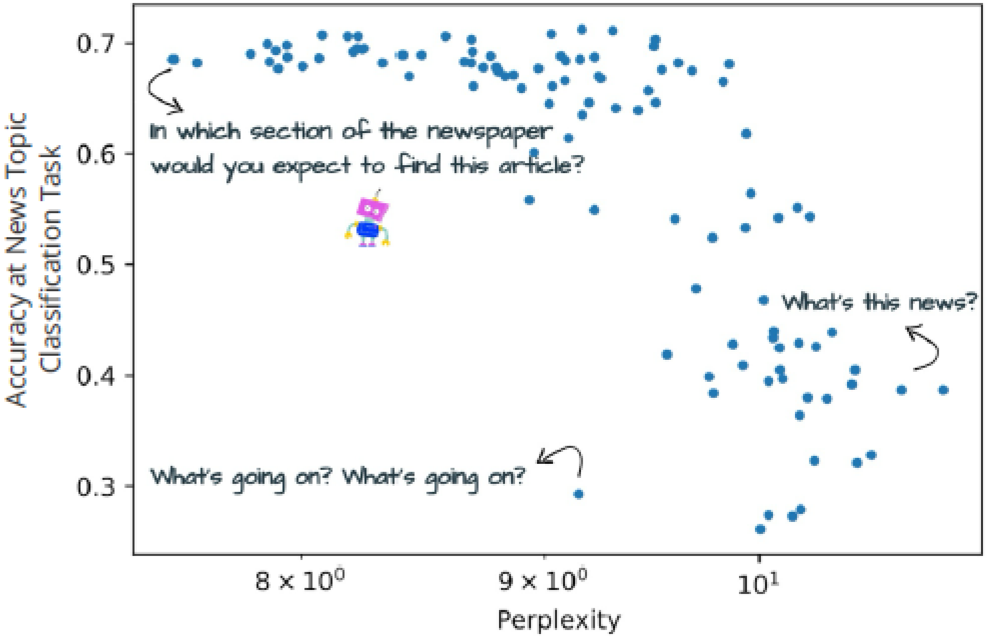
c) `What is the best way to describe this article?` 68.2%

d) `What is the most accurate label for this news article?` 71.2%

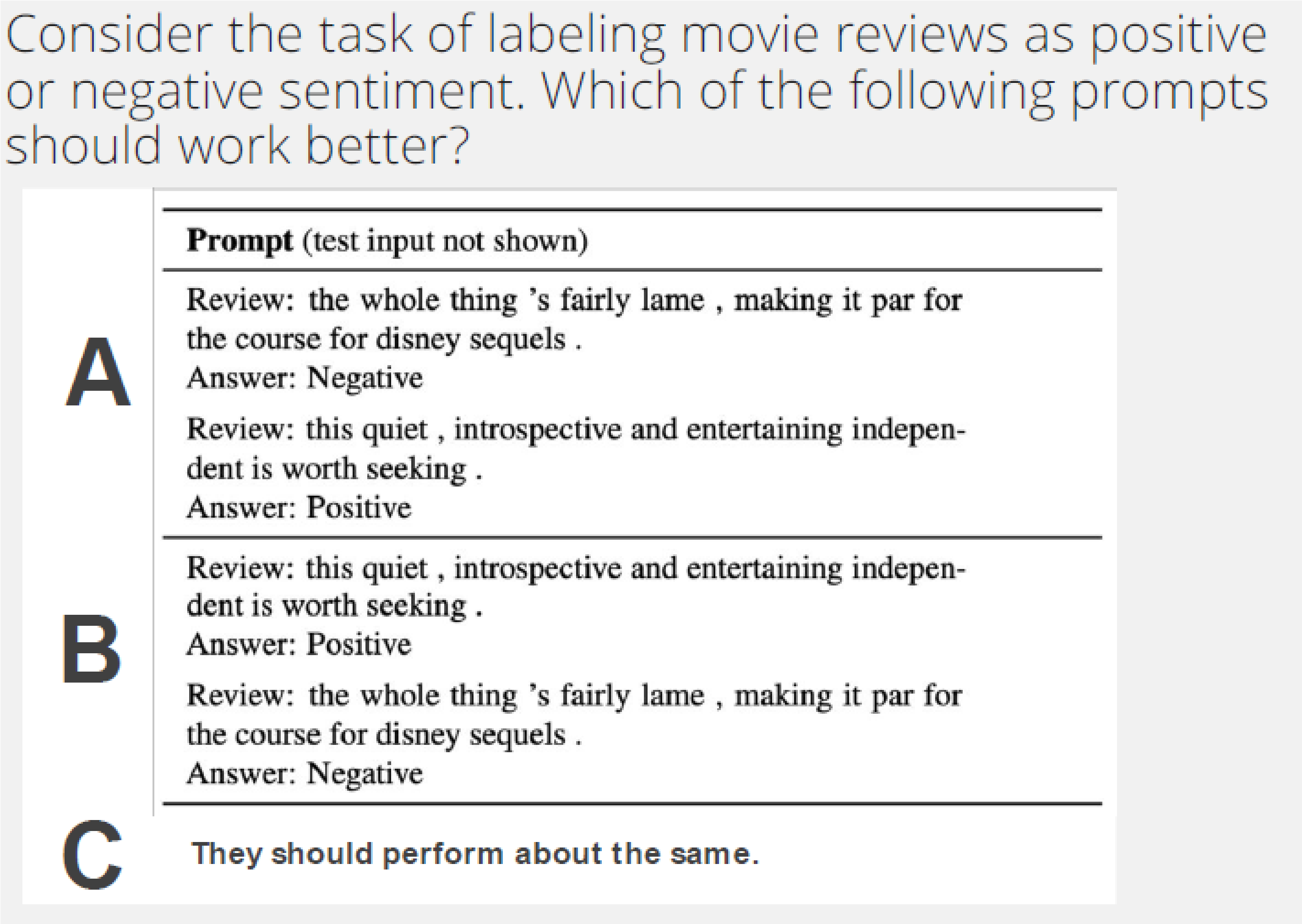
e) They should all perform about the same.

What matters in prompt selection?

- Prompts which are perceptually equivalent to humans can result in radically different performance.
- Prompt performance is correlated with the extent to which the model is familiar with the language the prompt contains.



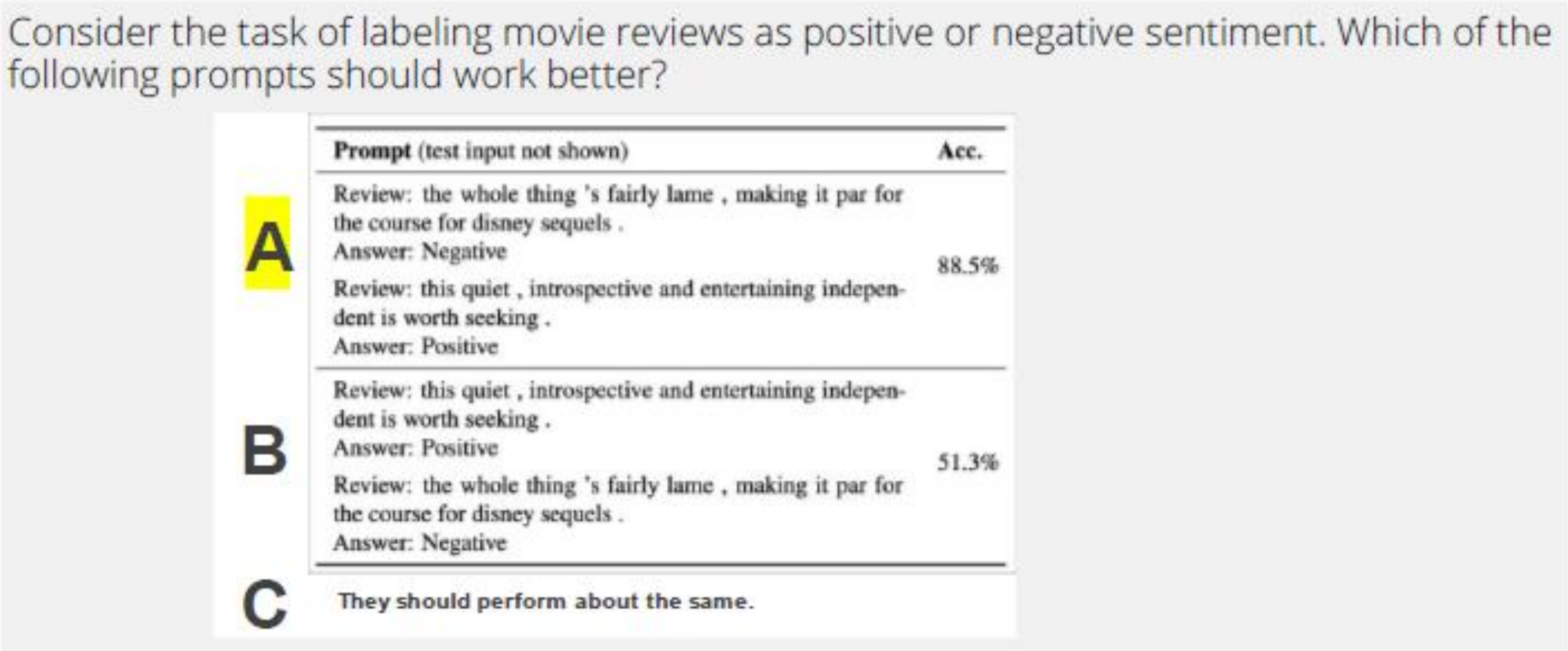
What matters in prompt selection?



Consider the task of labeling movie reviews as positive or negative sentiment. Which of the following prompts should work better?

	Prompt (test input not shown)
A	Review: the whole thing 's fairly lame , making it par for the course for disney sequels . Answer: Negative Review: this quiet , introspective and entertaining independent is worth seeking . Answer: Positive
B	Review: this quiet , introspective and entertaining independent is worth seeking . Answer: Positive Review: the whole thing 's fairly lame , making it par for the course for disney sequels . Answer: Negative
C	They should perform about the same.

What matters in prompt selection?



Consider the task of labeling movie reviews as positive or negative sentiment. Which of the following prompts should work better?

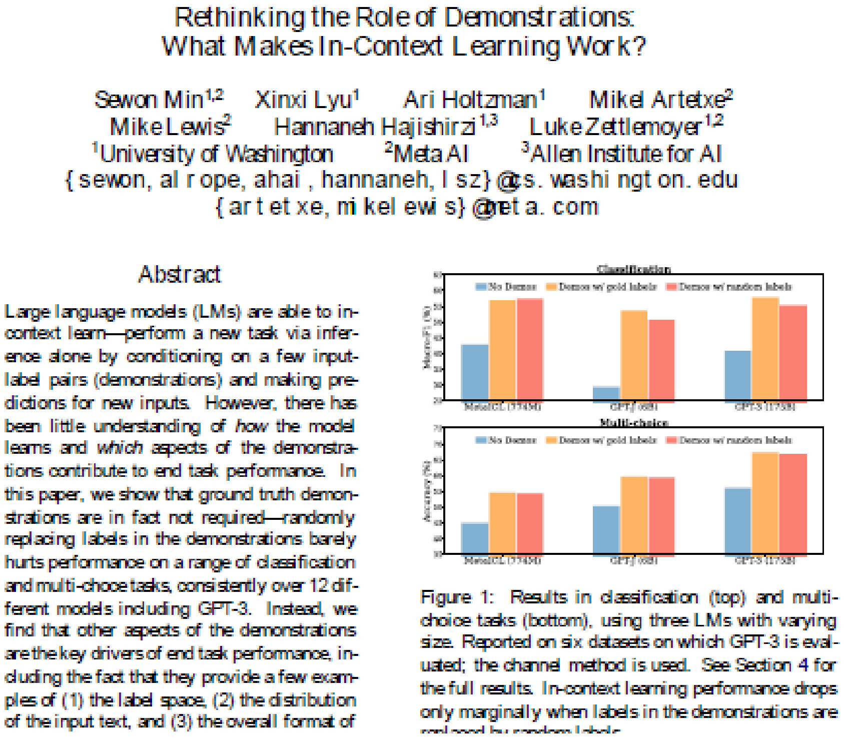
	Prompt (test input not shown)	**Acc.**
A	Review: the whole thing 's fairly lame , making it par for the course for disney sequels . Answer: Negative Review: this quiet , introspective and entertaining independent is worth seeking . Answer: Positive	88.5%
B	Review: this quiet , introspective and entertaining independent is worth seeking . Answer: Positive Review: the whole thing 's fairly lame , making it par for the course for disney sequels . Answer: Negative	51.3%

C They should perform about the same.

What matters in prompt selection?

- Prompts which are perceptually equivalent to humans can result in radically different performance.
- Prompt performance is correlated with the extent to which the model is familiar with the language the prompt contains.
- Few-shot example choice and ordering make a huge difference in performance.
- LLMs can be biased toward answers which occur more frequently in the prompt.
- Labels can be wrong and it doesn’t matter.

What matters in prompt selection?



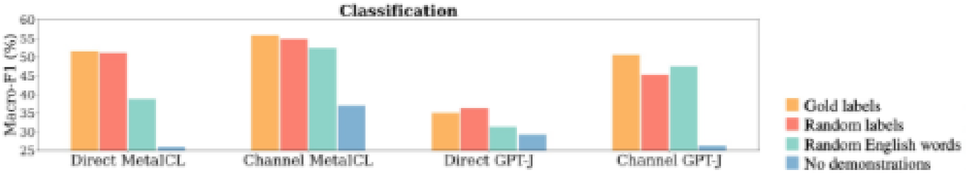
Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

Sewon Min[1,2] Xinxi Lyu[1] Ari Holtzman[1] Mikel Artetxe[2]
Mike Lewis[2] Hannaneh Hajishirzi[1,3] Luke Zettlemoyer[1,2]
[1]University of Washington [2]Meta AI [3]Allen Institute for AI
{sewon, alrope, ahai, hannaneh, lsz}@cs.washington.edu
{artetxe, mikelewis}@meta.com

Abstract

Large language models (LMs) are able to in-context learn—perform a new task via inference alone by conditioning on a few input-label pairs (demonstrations) and making predictions for new inputs. However, there has been little understanding of *how* the model learns and *which* aspects of the demonstrations contribute to end task performance. In this paper, we show that ground truth demonstrations are in fact not required—randomly replacing labels in the demonstrations barely hurts performance on a range of classification and multi-choce tasks, consistently over 12 different models including GPT-3. Instead, we find that other aspects of the demonstrations are the key drivers of end task performance, including the fact that they provide a few examples of (1) the label space, (2) the distribution of the input text, and (3) the overall format of

Figure 1: Results in classification (top) and multi-choice tasks (bottom), using three LMs with varying size. Reported on six datasets on which GPT-3 is evaluated; the channel method is used. See Section 4 for the full results. In-context learning performance drops only marginally when labels in the demonstrations are replaced by random labels



How much does having more exemplars help?

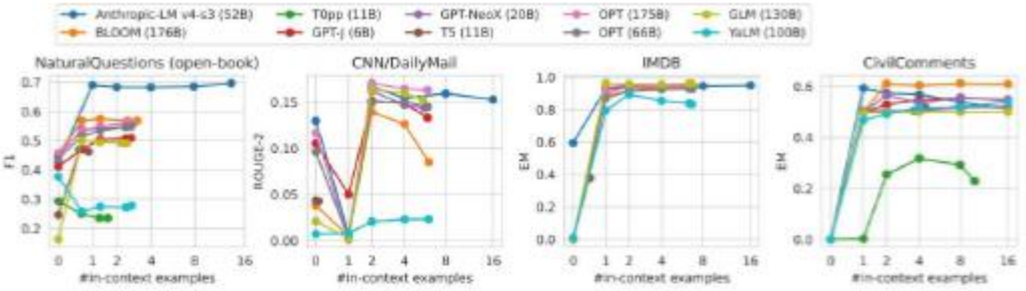
Holistic Evaluation of Language Models

Percy Liang[1], Rishi Bommasani[1], Tony Lee[1], Dimitris Tsipras[2], Dilara Soylu[2], Michihiro Yasunaga[2], Yian Zhang[2], Deepak Narayanan[2], Yuhuai Wu[2], Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, Yuta Koreeda
pliang@cs.stanford.edu, nlprishi@stanford.edu, tonyhlee@stanford.edu
Center for Research on Foundation Models (CRFM)
Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

Reviewed on OpenReview: https://openreview.net/forum?id=iO4LZibEqW

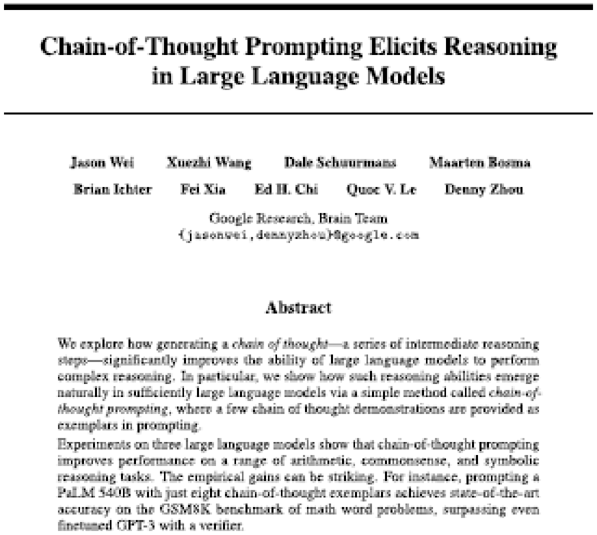
Abstract

Language models (LMs) are becoming the foundation for almost all major language technologies, but their capabilities, limitations, and risks are not well understood. We present Holistic Evaluation of Language Models (HELM) to improve the transparency of language models. First, we taxonomize the vast space of potential scenarios (i.e. use cases) and metrics (i.e. desiderata) that are of interest for LMs. Then we select a broad subset based on coverage and feasibility, noting what's missing or underrepresented (e.g. question answering for neglected English dialects, metrics for trustworthiness). Second, we adopt a multi-metric approach: We measure 7 metrics (accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency) for each of 16 core scenarios to the extent possible (87.5% of the time), ensuring that metrics beyond accuracy don't fall to the wayside, and that trade-offs across models and metrics are clearly exposed. We also perform 7 targeted evaluations, based on 26 targeted scenarios, to more deeply analyze specific aspects (e.g. knowledge, reasoning, memorization/copyright, disinformation). Third, we conduct a large-scale evaluation of 30 prominent language models (spanning open, limited-access, and closed models) on all 42 scenarios, including 21 scenarios that were not previously used in mainstream LM evaluation. Prior to HELM, models on average were evaluated on just 17.9% of the core HELM scenarios, with some prominent models not sharing a single scenario in common. We improve this to 96.0%: now all 30 models have been densely benchmarked on a set of core scenarios and metrics under standardized conditions. Our evaluation surfaces 25 top-level findings concerning the interplay between different scenarios, metrics, and models. For full transparency, we release all raw model prompts and completions publicly[1] for further analysis, as well as a general modular toolkit for easily adding new scenarios, models, metrics, and prompting strategies.[2] We intend for HELM to be a living benchmark for the community, continuously updated with new scenarios, metrics, and models.



Chain-of-Thought Prompting

- **Intuition:** An LLM will be better able to perform tasks (especially reasoning-based ones) if it is made to break down the task into multiple small steps.
- **Main idea:** each of the exemplars in your few-shot prompt contains logic showing how to solve the task.



Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

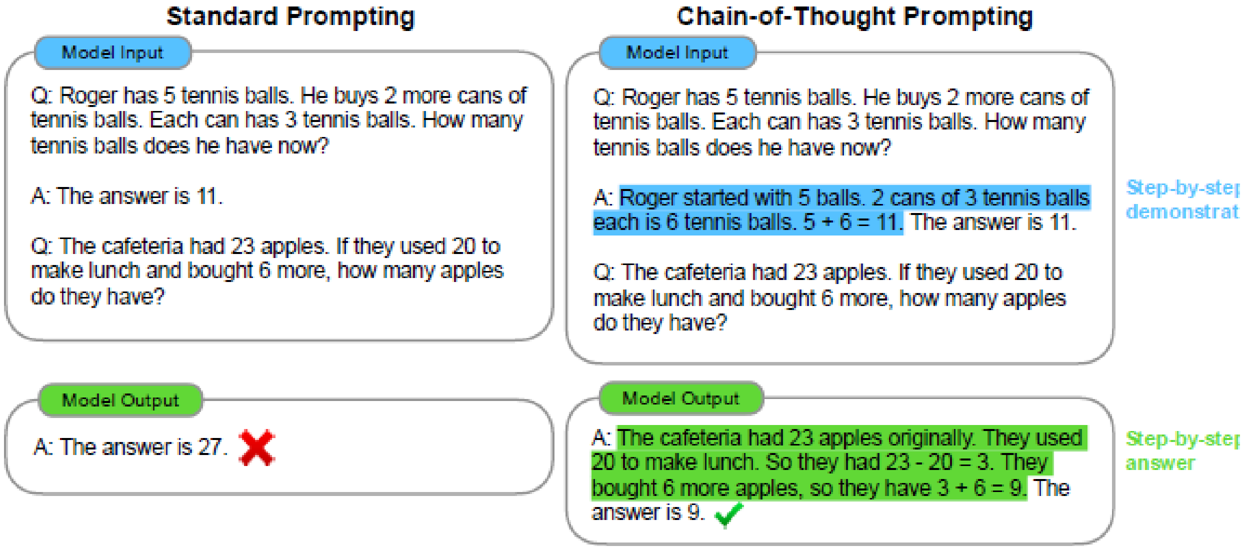
Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves *state-of-the-art* accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.



Standard Prompting

Model Input

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: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

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.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Zero-Shot Chain-of-Thought Prompting

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

Abstract

Pretrained large language models (LLMs) are widely used in many sub-fields of natural language processing (NLP) and generally known as excellent *few-shot* learners with task-specific exemplars. Notably, chain of thought (CoT) prompting, a recent technique for eliciting complex multi-step reasoning through step-by-step answer examples, achieved the state-of-the-art performances in arithmetics and symbolic reasoning, difficult *system-2* tasks that do not follow the standard scaling laws for LLMs. While these successes are often attributed to LLMs' ability for few-shot learning, we show that LLMs are decent *zero-shot* reasoners by simply adding "Let's think step by step" before each answer. Experimental results demonstrate that our Zero-shot-CoT, using the same single prompt template, significantly outperforms zero-shot LLM performances on diverse benchmark reasoning tasks including arithmetics (MultiArith, GSM8K, AQUA-RAT, SVAMP), symbolic reasoning (Last Letter, Coin Flip), and other logical reasoning tasks (Date Understanding, Tracking Shuffled Objects), without any hand-crafted few-shot examples, *e.g.* increasing the accuracy on MultiArith from 17.7% to 78.7% and GSM8K from 10.4% to 40.7% with large-scale InstructGPT model (text-davinci-002), as well as similar magnitudes of improvements with another off-the-shelf large model, 540B parameter PaLM. The versatility of this single prompt across very diverse reasoning tasks hints at untapped and understudied fundamental *zero-shot* capabilities of LLMs, suggesting high-level, multi-task broad cognitive capabilities may be extracted by simple prompting. We hope our work not only serves as the minimal strongest zero-shot baseline for the challenging reasoning benchmarks, but also highlights the importance of carefully exploring and analyzing the enormous zero-shot knowledge hidden inside LLMs before crafting finetuning datasets or few-shot exemplars.

Main idea: We don't need any exemplars! Just append the string "Let's think step by step." to the end of the prompt.

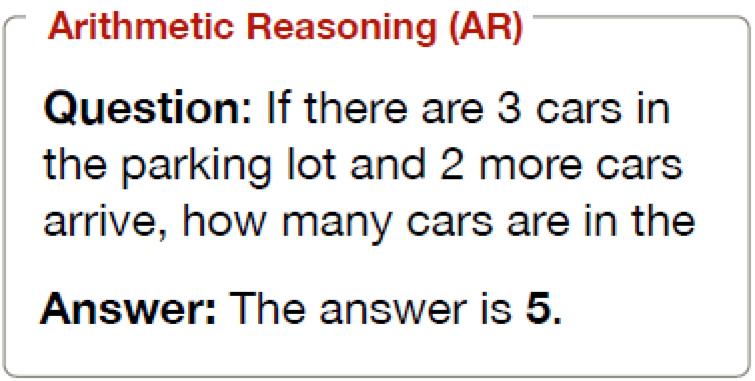
Reasoning Problems

- Multi-step reasoning is often seen as a weakness in NLP models.

- There is former research on reasoning in small language models through fully-supervised finetuning on specific datasets. However,
  - Creating a dataset containing explicit reasoning can be difficult and time-consuming.
  - training on a specific dataset limits application to a specific domain

- Reasoning ability may emerge in language models at a certain scale, such as models with over 100 billion parameters (Wei et al., TMLR 2022)

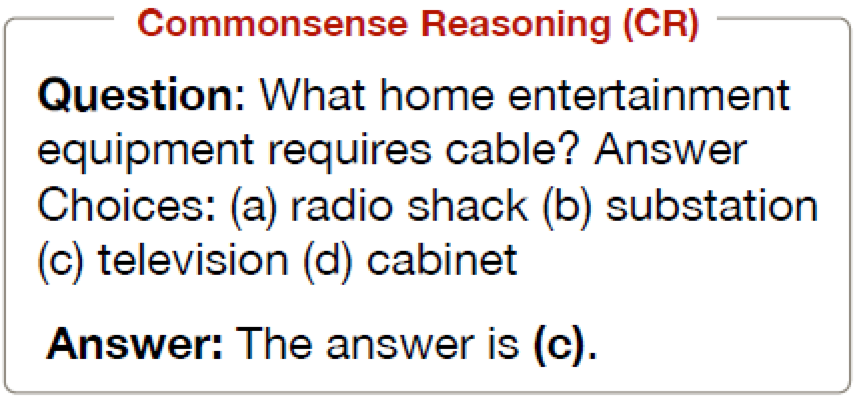
Reasoning Datasets



Arithmetic Reasoning (AR)

Question: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the

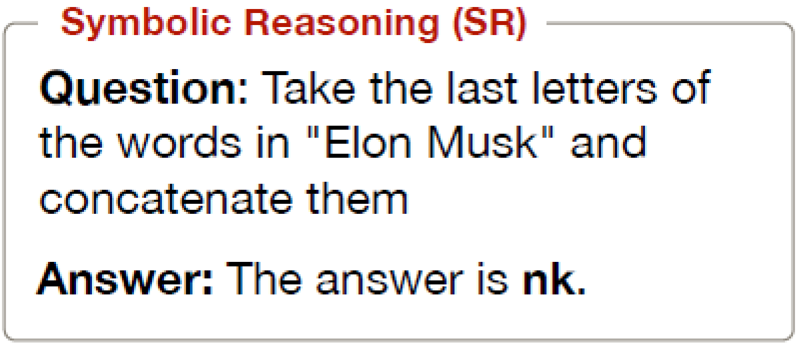
Answer: The answer is **5.**



Commonsense Reasoning (CR)

Question: What home entertainment equipment requires cable? Answer Choices: (a) radio shack (b) substation (c) television (d) cabinet

Answer: The answer is **(c).**



Symbolic Reasoning (SR)

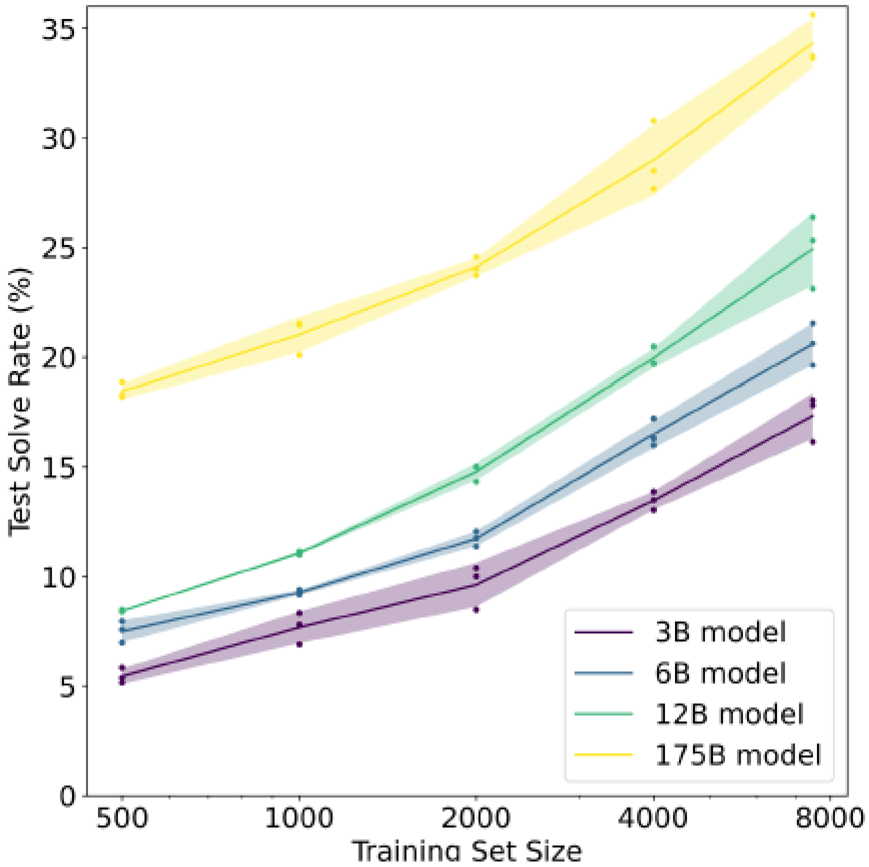
Question: Take the last letters of the words in "Elon Musk" and concatenate them

Answer: The answer is **nk.**

Reasoning Problems

- **Conjecture**: to achieve > 80%, needs 100 times more fine-tuning data for 175B model.

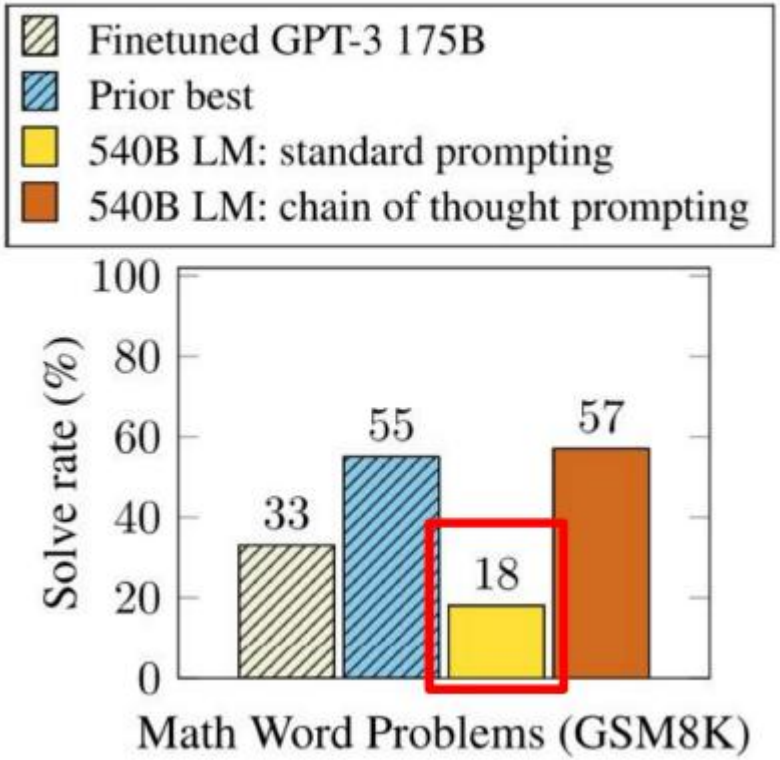
Fine-tune GPT-3 on GSM8K (arithmetic):



Reasoning Problems

- **Few-shot standard prompting** with even larger model (PaLM 540B) also does not work well.

GSM8K (arithmetic):



Chain-of-Thought Prompting

- **Definition:**
  - A chain of thought is **a series of intermediate natural language reasoning steps** that lead to the final output.
  - <**input, output**> demonstrations are replaced with <**input, chain of thought, output**>

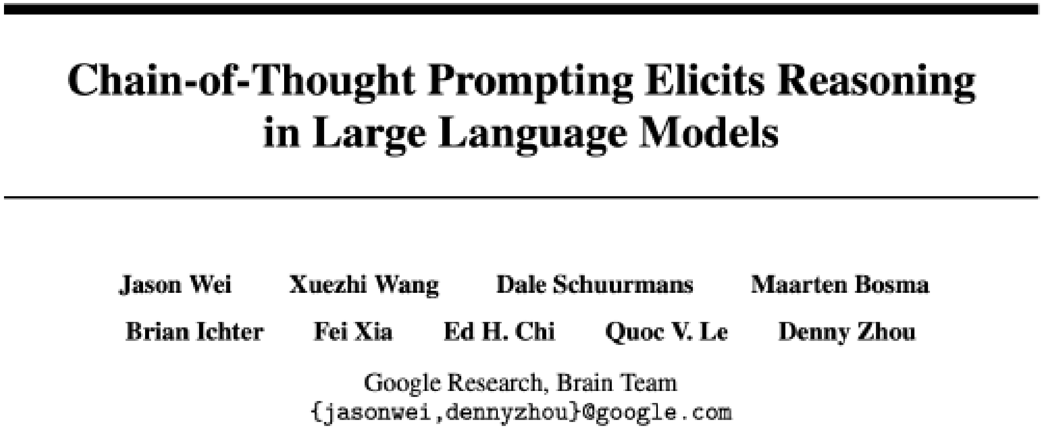
Compositionality of Language

- **Compositionality of languages**
  - Compositional out-of-distribution generalization: ability to understand novel composition of known concepts

- **Problem decomposition**
  - Can help decompose multi-step reasoning into intermediate steps

Chain-of-Thought(CoT) Prompting

Few-Shot CoT

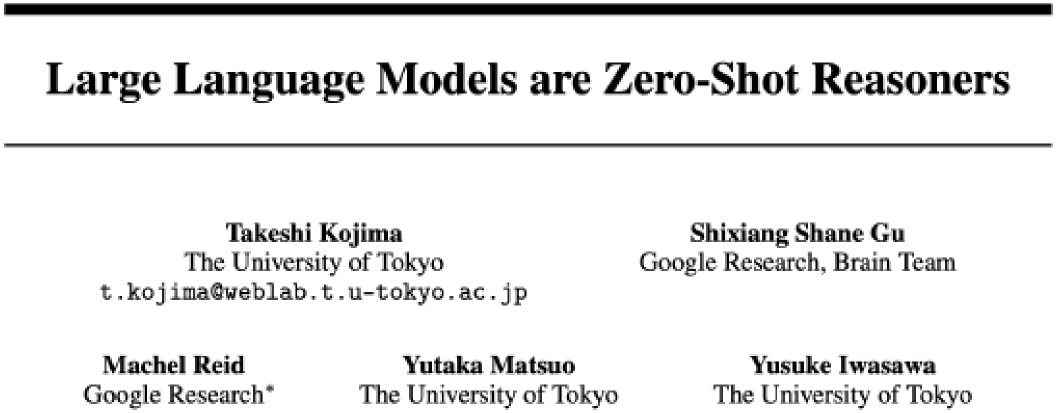


Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

Zero-Shot CoT



Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

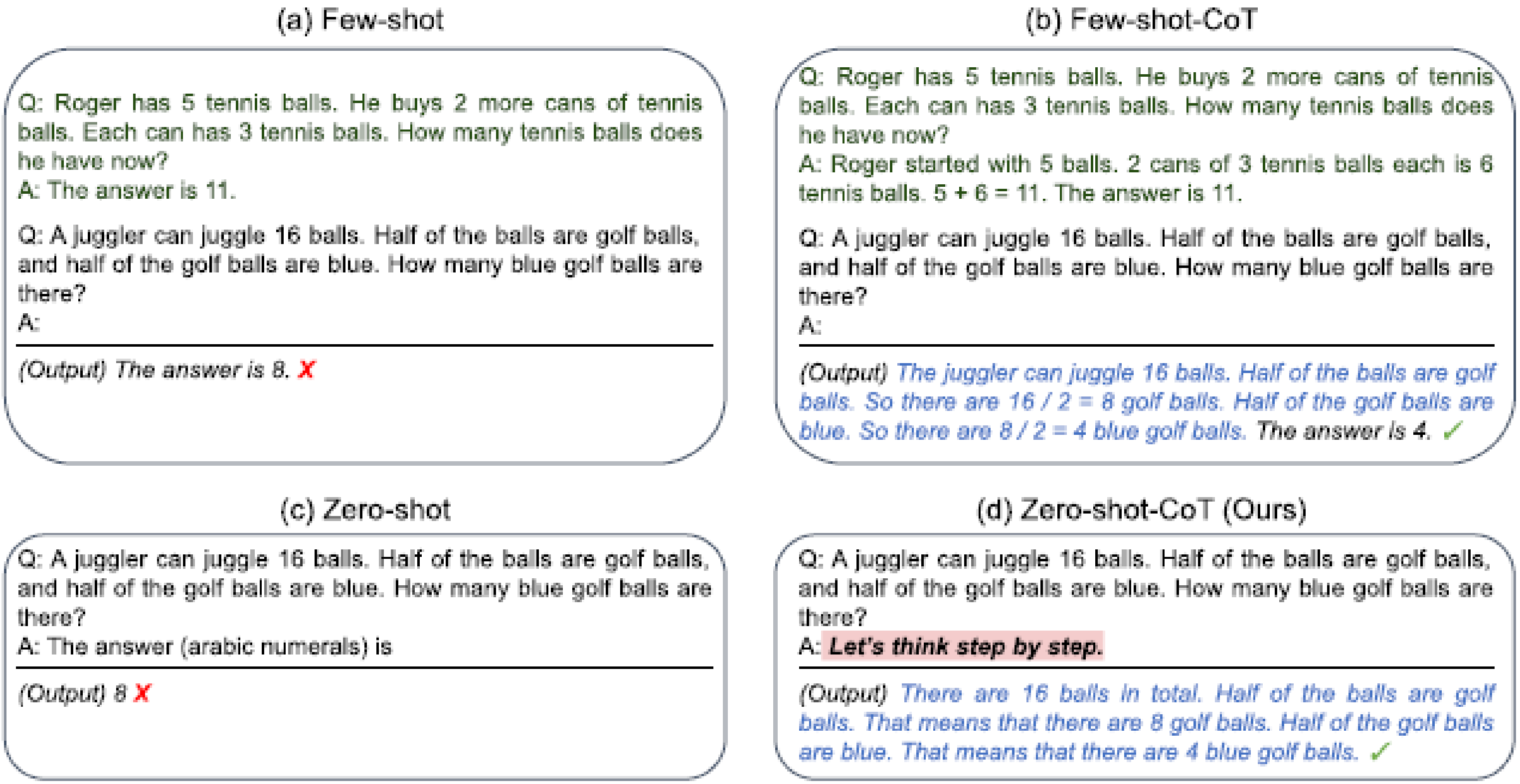
Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

Both papers will appear in **NeurIPS'22!**

Chain-of-Thought Prompting

- Few-shot vs. Zero-shot



(a) Few-shot

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: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

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.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

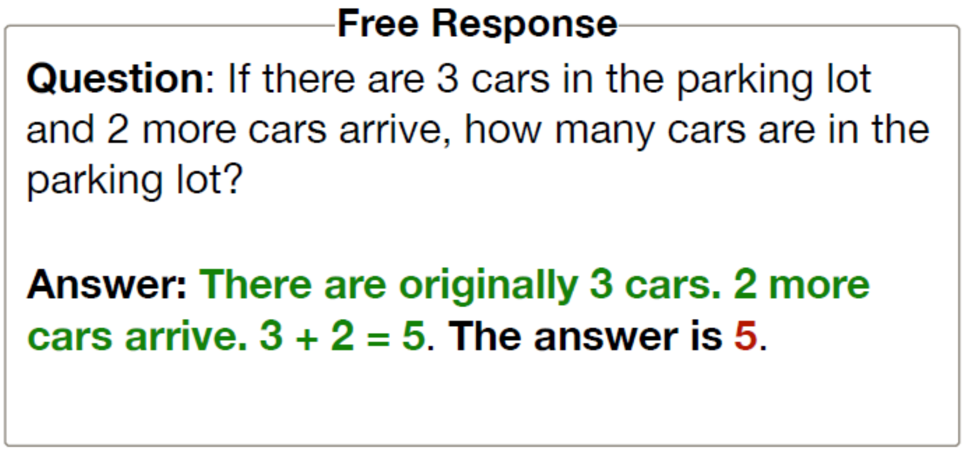
(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Chain-of-Thought Prompting

- Free Response vs. Multiple Choice



Free Response

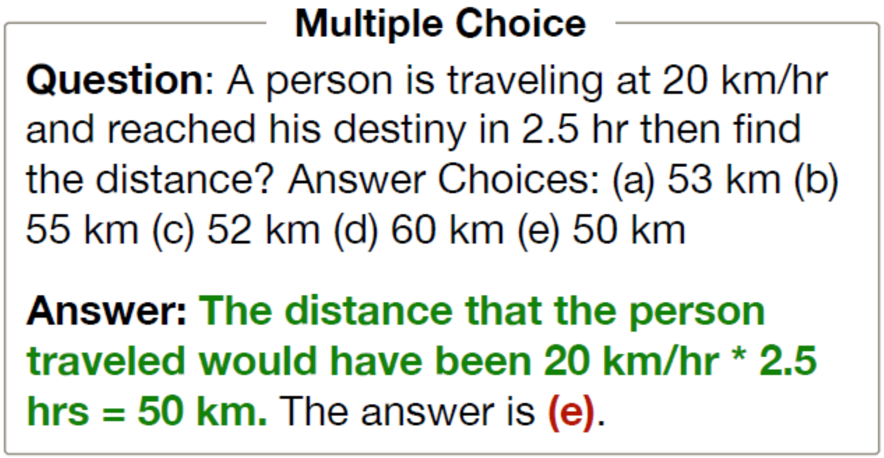
Question: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

Answer: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5.

- **Manually** composed **8** exemplars
- All contains equations with flexible formats
- Benchmarked on:
  - **GSM8K** (Cobbe et al. 2021)
  - **SVAMP** (Patel et al., 2021)
  - **MAWPS** (Koncel-Kedziorski et al., 2016)

Chain-of-Thought Prompting

- Free Response vs. Multiple Choice



Multiple Choice

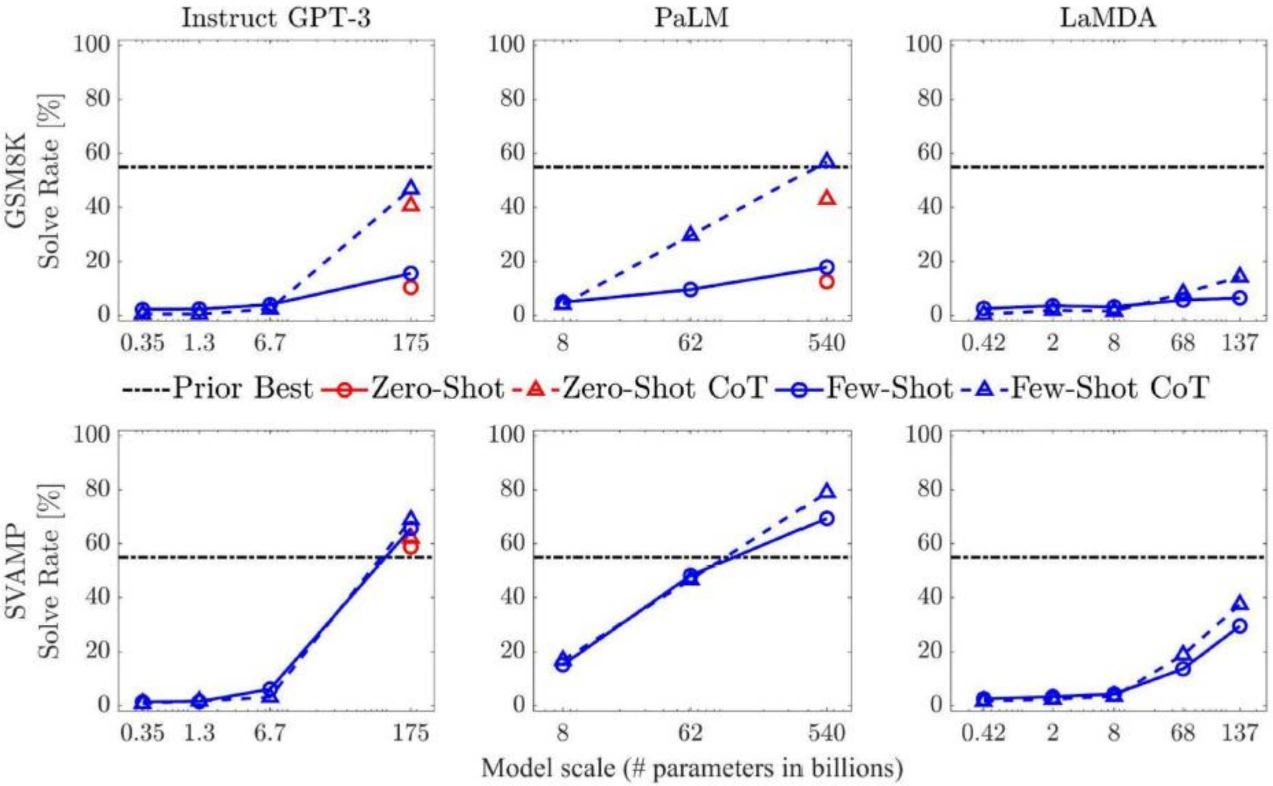
Question: A person is traveling at 20 km/hr and reached his destiny in 2.5 hr then find the distance? Answer Choices: (a) 53 km (b) 55 km (c) 52 km (d) 60 km (e) 50 km

Answer: The distance that the person traveled would have been 20 km/hr * 2.5 hrs = 50 km. The answer is **(e)**.

GSM8K (Cobbe et al. 2021)

- 4 exemplars, whose questions, intermediate reasoning, and answers are from AQuA-RAT's **training set**
- Exemplars have flexible formats
- Benchmarked on **AQuA-RAT** (Ling et al., 2017)

Arithmetic Reasoning - Results



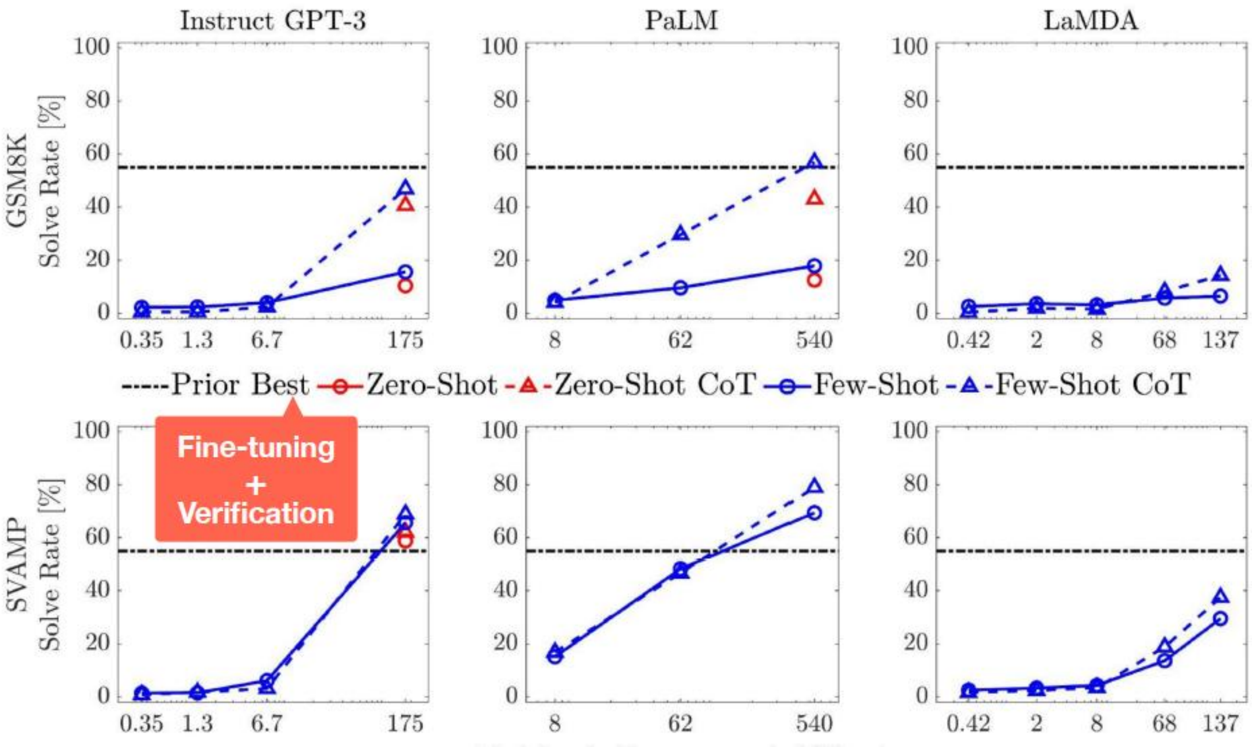
GSM8K

Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,00 in repairs. This increased the value of the house by 150%. How much profit did he make?

SVAMP

Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?

Arithmetic Reasoning - Results



Fine-tuning + Verification

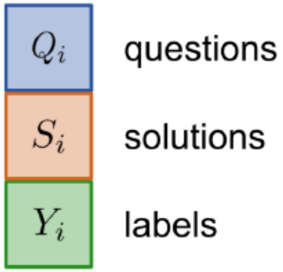
GSM8K

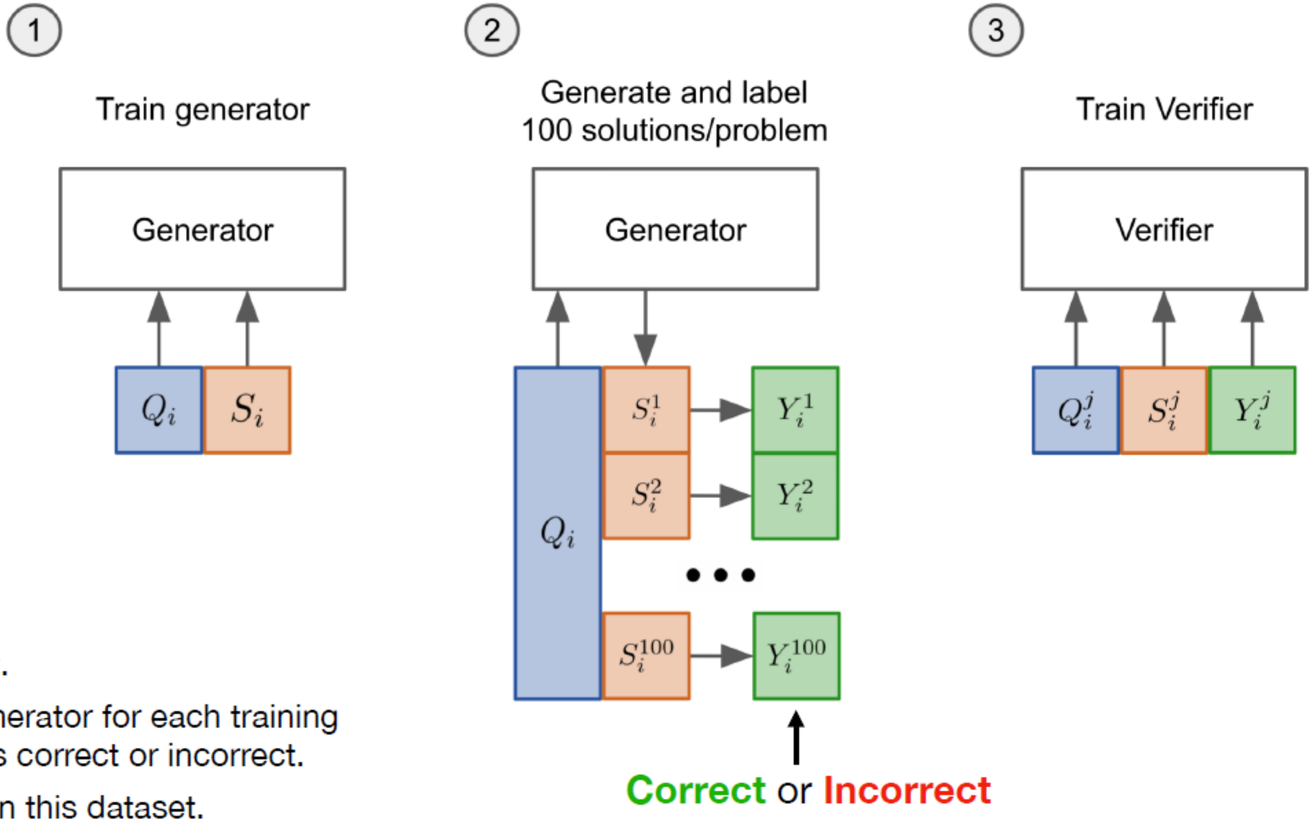
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,00 in repairs. This increased the value of the house by 150%. How much profit did he make?

SVAMP

Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?

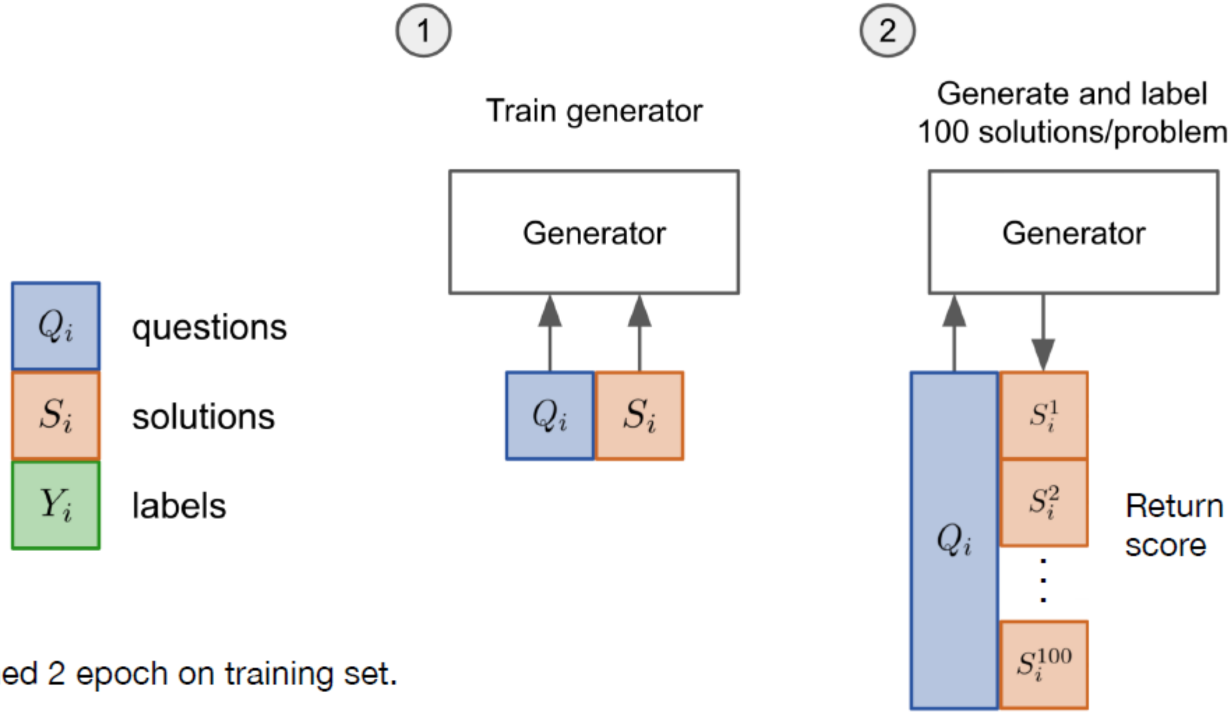
Prior Best – Fine-tuning + Verification





1. Fine-tuned 2 epoch on training set.
2. Sample 100 solutions from the generator for each training problem and label each solution as correct or incorrect.
3. Train a verifier for a single epoch on this dataset.

Prior Best – Fine-tuning + Verification



Q_i questions

S_i solutions

Y_i labels

1 Train generator

Generator

Q_i S_i

2 Generate and label 100 solutions/problem

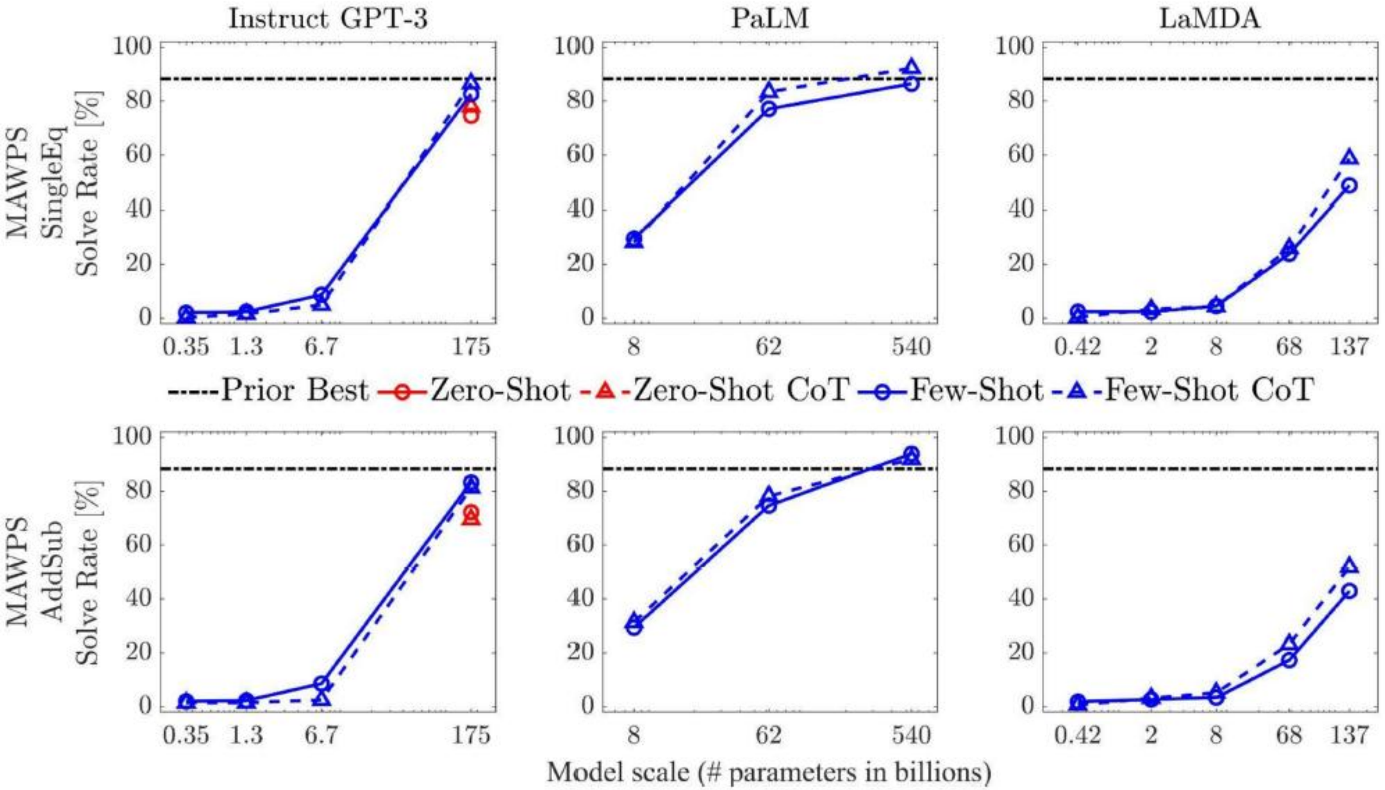
Generator

Q_i S_i^1 S_i^2 S_i^{100}

Return the one with the highest verifier score

1. Fine-tuned 2 epoch on training set.

Arithmetic Reasoning - Results



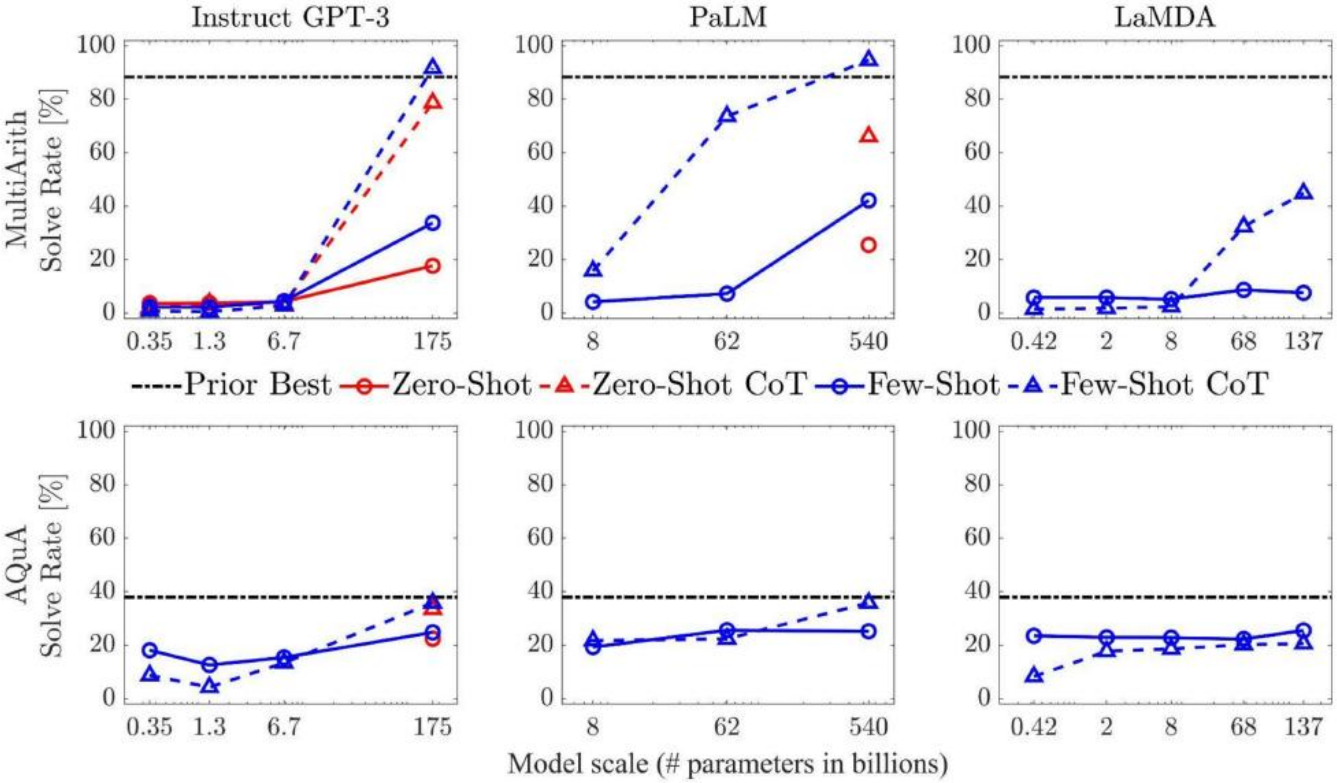
MAWPS - SingleEq

If there are 7 bottle caps in a box and Linda puts 7 more bottle caps inside, how many bottle caps are in the box?

MAWPS - AddSub

There were 6 roses in the vase. Mary cut some roses from her flower garden. There are now 16 roses in the vase. How many roses did she cut?

Arithmetic Reasoning - Results



MAWPS - MultiArith

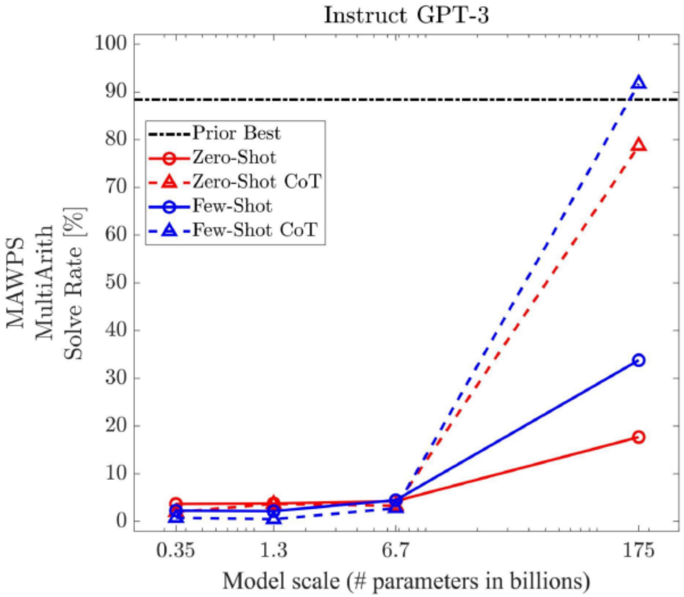
The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

AQuA-RAT

A person is traveling at 20 km/hr and reached his destiny in 2.5 hr then find the distance? Answer Choices: (a) 53 km (b) 55 km (c) 52 km (d) 60 km (e) 50 km

Arithmetic Reasoning - Results

- Both zero-shot and few-shot CoT promptings are emergent **abilities of model scale.**
- Do not positively impact performance for small models
  - start to yield performance gains when used with models with more than ~100B parameters.
- Few-shot CoT achieves **better** performance on LLM than zero-shot CoT.


Instruct GPT-3
MAWPS MultiArith Solve Rate [%]
Prior Best
Zero-Shot
Zero-Shot CoT
Few-Shot
Few-Shot CoT
Model scale (# parameters in billions)