

CS145: Introduction to Data Mining (Spring 2024)

Discussion 3: Large Language Models

Instructor: Dr. Ziniu Hu

Teaching Assistant: Yanqiao Zhu

*The Scalable Analytics Institute (ScAI)
Department of Computer Science
University of California, Los Angeles (UCLA)*

Announcement

1. HW3 has been released, due next Sunday (EOD)
2. The reader will finish grading HW1 by the end of next week
3. Baseline code has been released [here](#)
4. Each team needs to make a dummy submission by the end of next week

Recap: ML recipe

1. Given training data:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$$

2. Choose each of these:

- Decision function

$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

- Loss function

$$\ell(\hat{\mathbf{y}}, \mathbf{y}_i) \in \mathbb{R}$$

3. Define goal:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

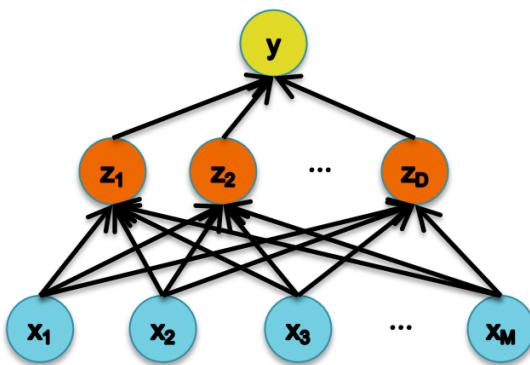
4. Train with SGD:

(take small steps
opposite the gradient)

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i)$$

Recap: Backpropagation

Example:
Neural Network



	Forward	Backward
Loss	$J = y^* \log y + (1 - y^*) \log(1 - y)$	$g_y = \frac{y^*}{y} + \frac{(1 - y^*)}{y - 1}$
Sigmoid	$y = \frac{1}{1 + \exp(-b)}$	$g_b = g_y \frac{\partial y}{\partial b}, \frac{\partial y}{\partial b} = y(1 - y)$
Linear	$b = \sum_{j=0}^D \beta_j z_j$	$g_{\beta_j} = g_b \frac{\partial b}{\partial \beta_j}, \frac{\partial b}{\partial \beta_j} = z_j$ $g_{z_j} = g_b \frac{\partial b}{\partial z_j}, \frac{\partial b}{\partial z_j} = \beta_j$
Sigmoid	$z_j = \frac{1}{1 + \exp(-a_j)}$	$g_{a_j} = g_{z_j} \frac{\partial z_j}{\partial a_j}, \frac{\partial z_j}{\partial a_j} = z_j(1 - z_j)$
Linear	$a_j = \sum_{i=0}^M \alpha_{ji} x_i$	$g_{\alpha_{ji}} = g_{a_j} \frac{\partial a_j}{\partial \alpha_{ji}}, \frac{\partial a_j}{\partial \alpha_{ji}} = x_i$ $g_{x_i} = \sum_{j=0}^D g_{a_j} \frac{\partial a_j}{\partial x_i}, \frac{\partial a_j}{\partial x_i} = \alpha_{ji}$

Today: Large language models!

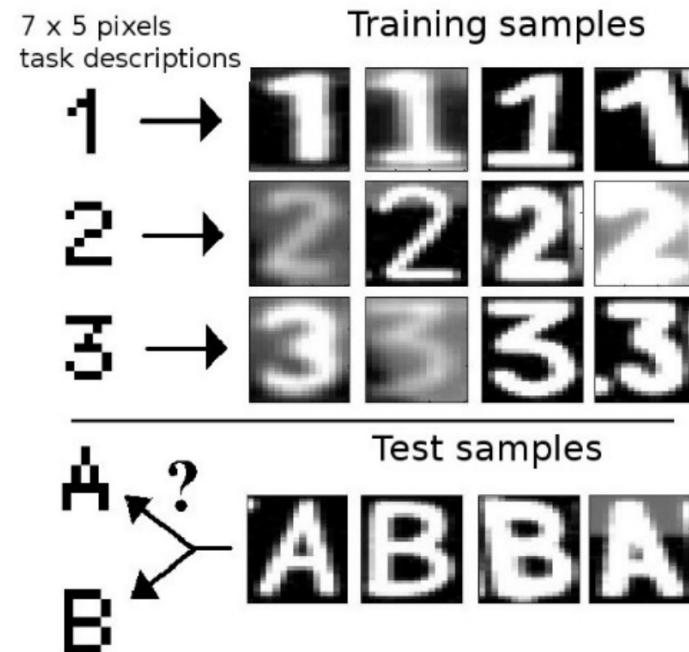
- Basic concepts
- In-context learning
- Prompting

Zero-shot learning

- Definition: in zero-shot learning we assume that training data does not contain any examples of the labels that appear in the test data

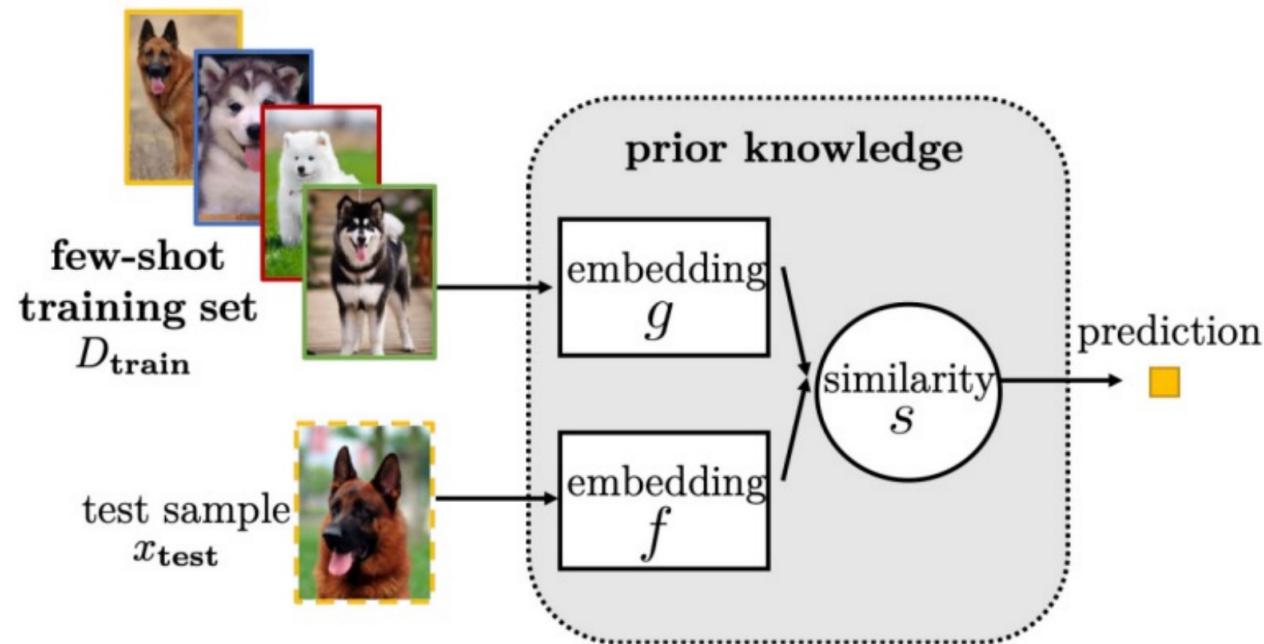
$z \rightarrow$	1	2	3	4	5
$d(z) \rightarrow$	1	2	3	A	B
x_t	y_t^1	y_t^2	y_t^3	y_t^4	y_t^5
1	1	0	0	-	-
2	0	1	0	-	-
3	0	0	1	-	-
A	-	-	-	1	0
B	-	-	-	0	1

$\underbrace{\hspace{1cm}}$ training data $\underbrace{\hspace{1cm}}$ test data



Few-shot learning

- Definition: in few-shot learning we assume that training data contains a handful (maybe two or more) examples of each label



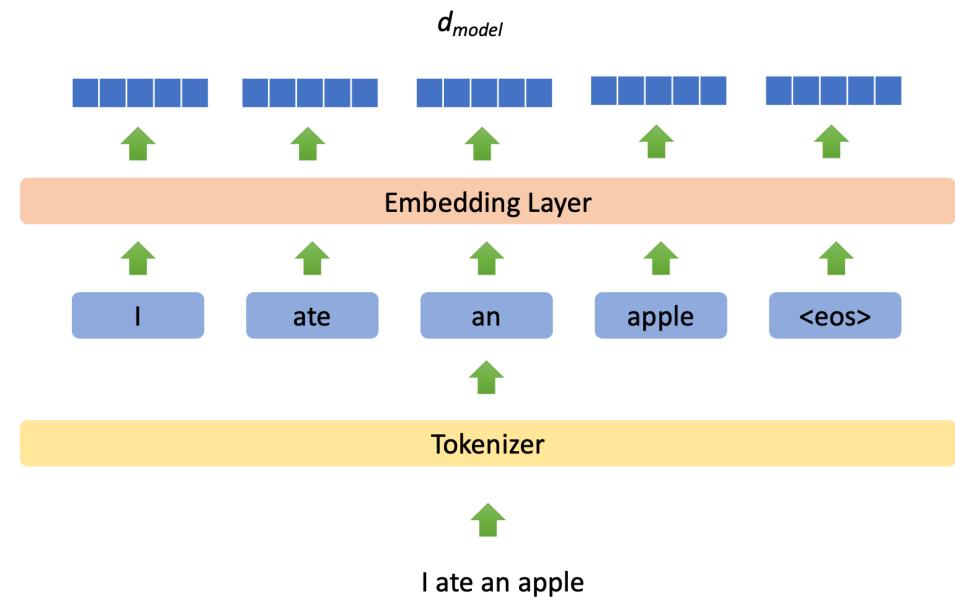
Language modeling

- Language models are trained to maximize the likelihood of sequences in the training data
- Most condition on the previous tokens to generate the next tokens
- Definition: An autoregressive language model defines a probability distribution over sequences $\mathbf{x}_{1:T}$ of the form:

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^T p(x_t \mid x_1, \dots, x_{t-1})$$

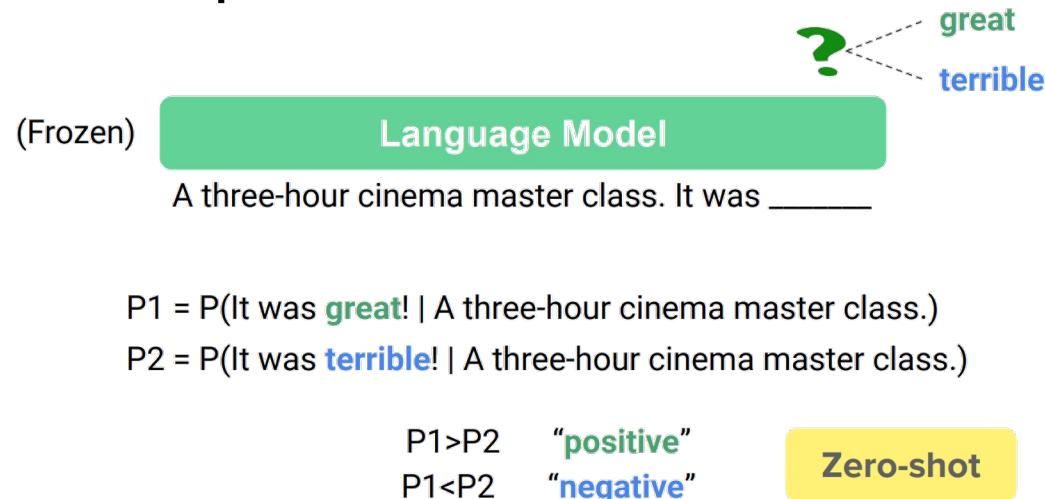
Tokenization

- Tokenization is the process of dividing a piece of text into tokens
- Tokens are meaningful units to capture the semantic and syntactic structure of the given text
- Common strategies:
 - Word-level tokenization
 - Subword-level tokenization
(e.g., using byte-pair encoding)
 - Character-level tokenization



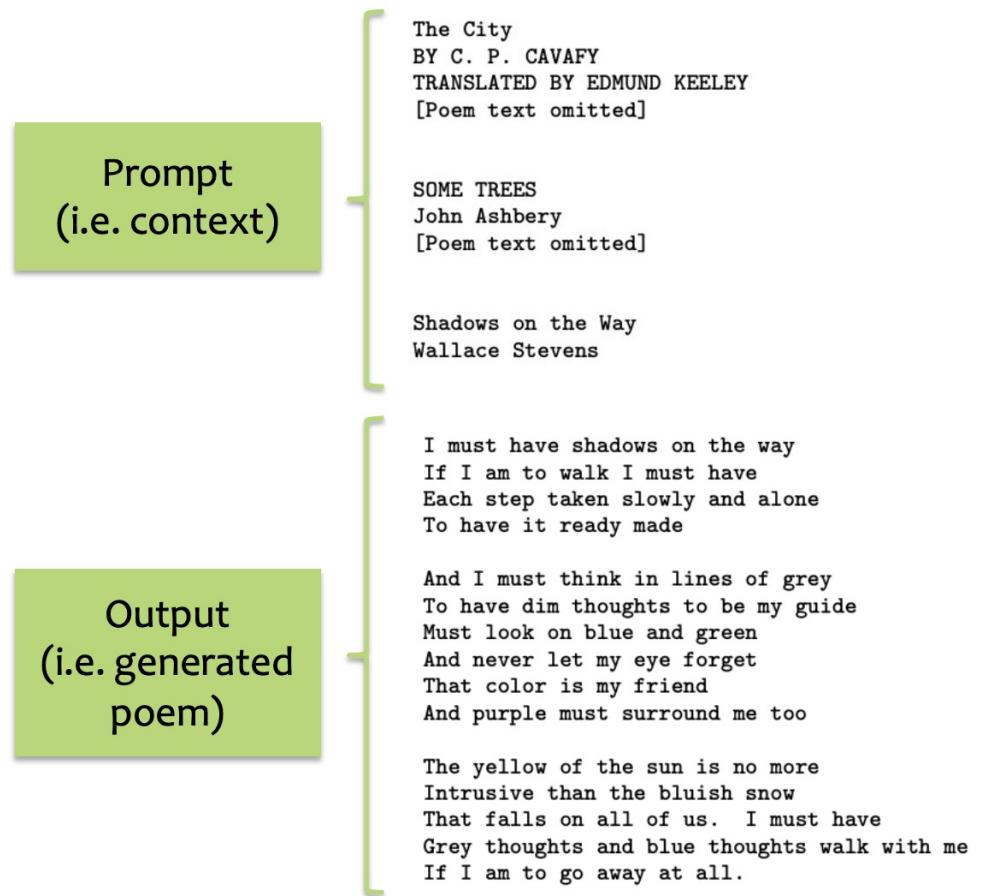
Prompting

- Using a LLM to perform a new task without gradient updates
 - Key idea behind prompting: Provide a prefix string to the such that its likely completion is the answer you want
 - Conditional sampling: To draw a sample, feed in a prompt (i.e. context) and then sample from the model conditioned on that prompt



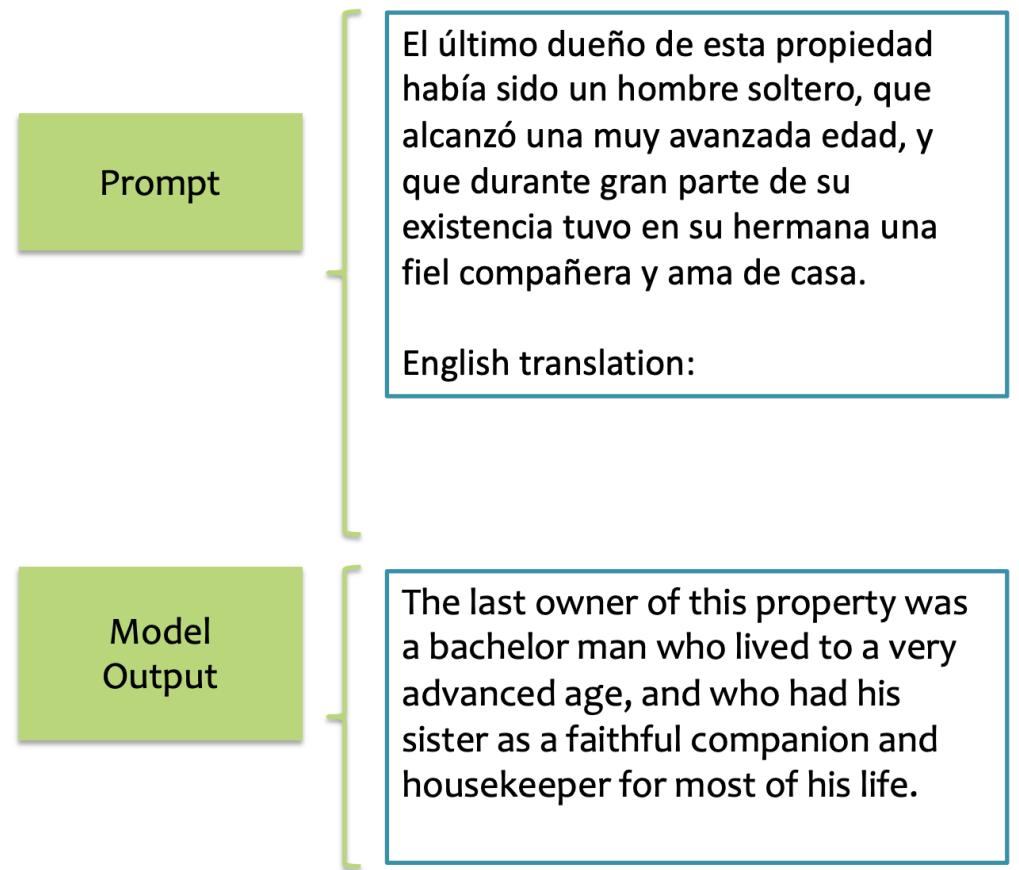
Prompting

- Example 1: text completion



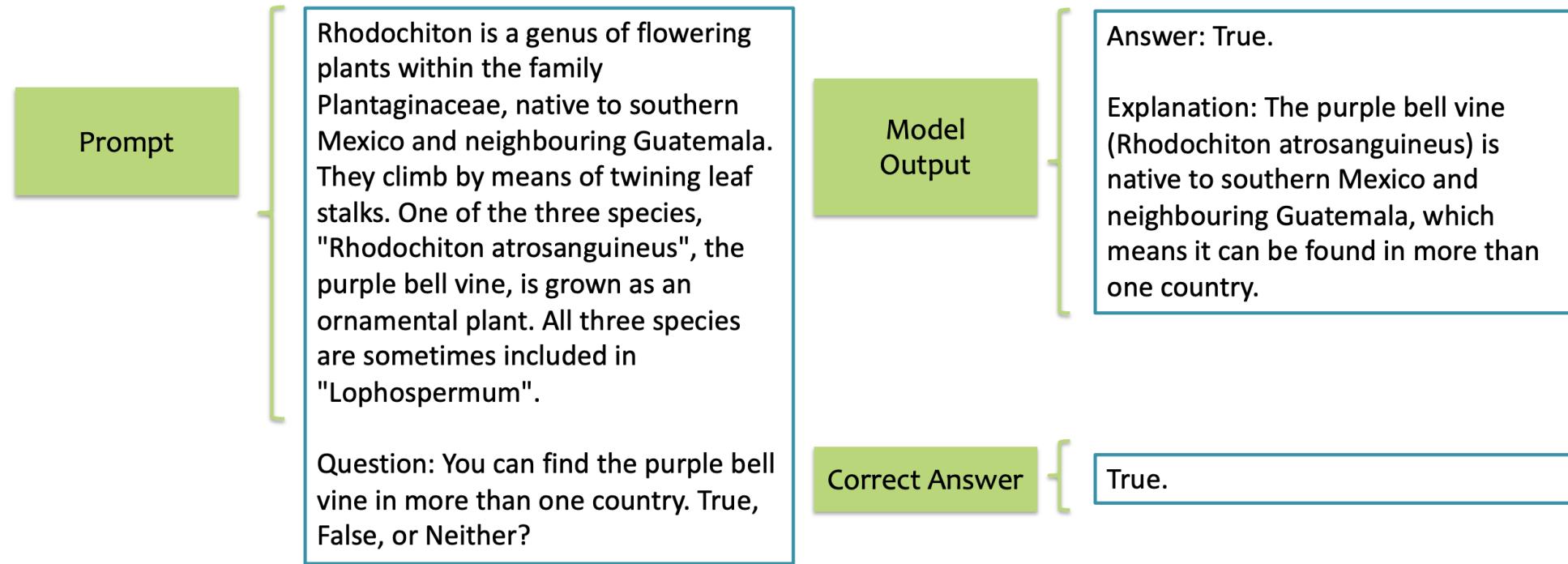
Prompting

- Example 2: translation



Prompting

- Example 3: Q&A (entailment)



What else can a large LM (attempt to) do?

Using the idea of **prompts**, we can apply LMs to a variety of different problems in natural language processing.

In the **zero-shot setting**, we simply feed the context to the model and observe how it completes the sequence. (i.e. there is no additional training)

Answer fact-based questions:

Context → Organisms require energy in order to do what?

- Correct Answer → mature and develop.
- Incorrect Answer → rest soundly.
- Incorrect Answer → absorb light.
- Incorrect Answer → take in nutrients.

Complete sentences logically:

Context → My body cast a shadow over the grass because

- Correct Answer → the sun was rising.
- Incorrect Answer → the grass was cut.

Complete analogies:

Context → lull is to trust as

- Correct Answer → cajole is to compliance
- Incorrect Answer → balk is to fortitude
- Incorrect Answer → betray is to loyalty
- Incorrect Answer → hinder is to destination
- Incorrect Answer → soothe is to passion

Reading comprehension:

Context → anli 1: anli 1: Fulton James MacGregor MSP is a Scottish politician who is a Scottish National Party (SNP) Member of Scottish Parliament for the constituency of Coatbridge and Chryston. MacGregor is currently Parliamentary Liaison Officer to Shona Robison, Cabinet Secretary for Health & Sport. He also serves on the Justice and Education & Skills committees in the Scottish Parliament.
Question: Fulton James MacGregor is a Scottish politician who is a Liaison officer to Shona Robison who he swears is his best friend. True, False, or Neither?

- Correct Answer → Neither
- Incorrect Answer → True
- Incorrect Answer → False

Prompting for instruction fine-tuned models

- Models like ChatGPT have been fine-tuned as chat assistants
- These (often) were trained with specific prompt templates that segment the prompt into different parts: (1) system, (2) assistant, and (3) user

	Llama-2 Chat	Alpaca
sys:	[INST] <>SYS>> You are a helpful AI assistant... <>/SYS>> [/INST]	### Instruction: Organisms require energy in order to do what?
asst:		
user:	mature and develop	[/INST] mature and develop
sys:	### Response: mature and develop	### Response: mature and develop
asst:		
user:		

Few-shot learning with LLMs

Suppose you have...

- a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ and N is rather small (i.e. few-shot setting)
- a very large (billions of parameters) pre-trained language model

There are two ways to “learn”

This section!



Option A: Supervised fine-tuning

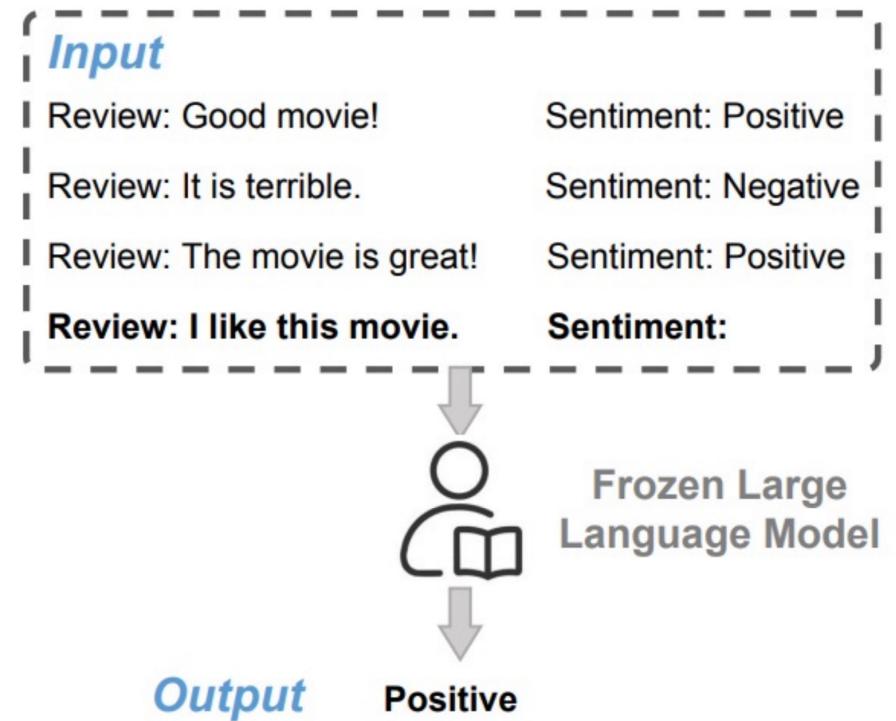
- **Definition:** fine-tune the LLM on the training data using...
 - a standard supervised objective
 - backpropagation to compute gradients
 - your favorite optimizer (e.g. Adam)
- **Pro:** fits into the standard ML recipe
- **Pro:** still works if N is large
- **Con:** backpropagation requires $\sim 3x$ the memory and computation time as the forward computation
- **Con:** you might not have access to the model weights at all (e.g. because the model is proprietary)

Option B: In-context learning

- **Definition:**
 1. feed training examples to the LLM as a prompt
 2. allow the LLM to infer patterns in the training examples during inference (i.e. decoding)
 3. take the output of the LLM following the prompt as its prediction
- **Con:** the prompt may be very long and Transformer LMs require $O(N^2)$ time/space where N = length of context
- **Pro:** no backpropagation required and only one pass through the training data
- **Pro:** does not require model weights, only API access

Few-shot in-context learning

- Few-shot learning can be done via in-context learning
- Typically, a task description is presented first
- Then a sequence of input/output pairs from a training dataset are presented in sequence



Few-shot in-context learning

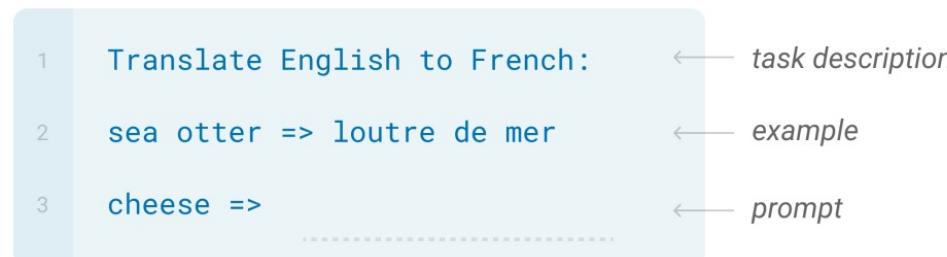
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



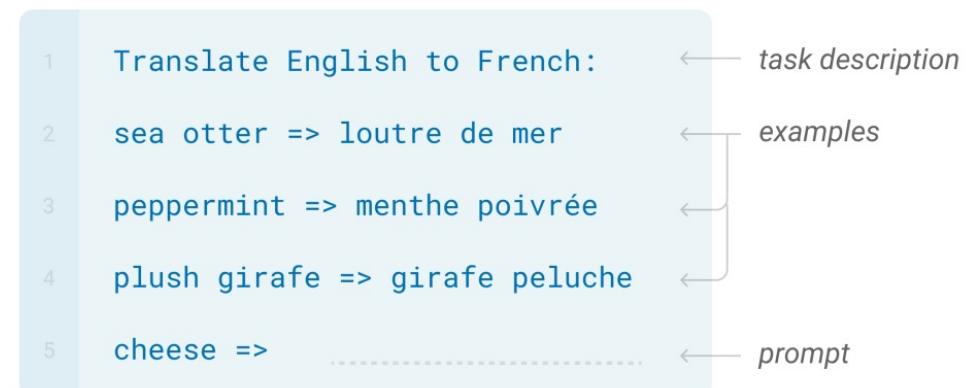
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



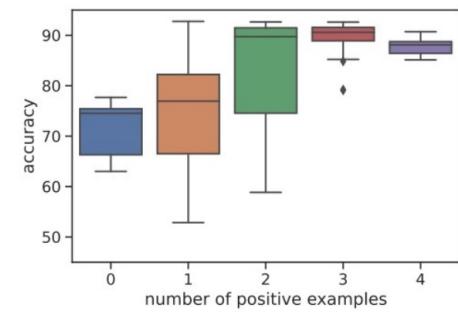
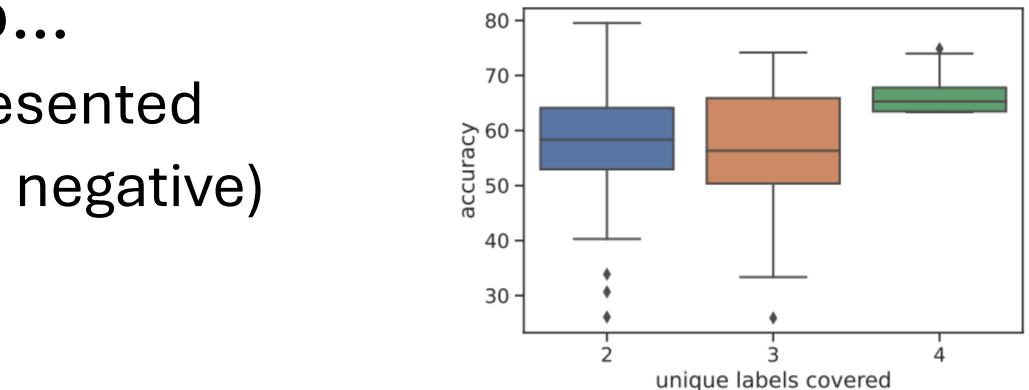
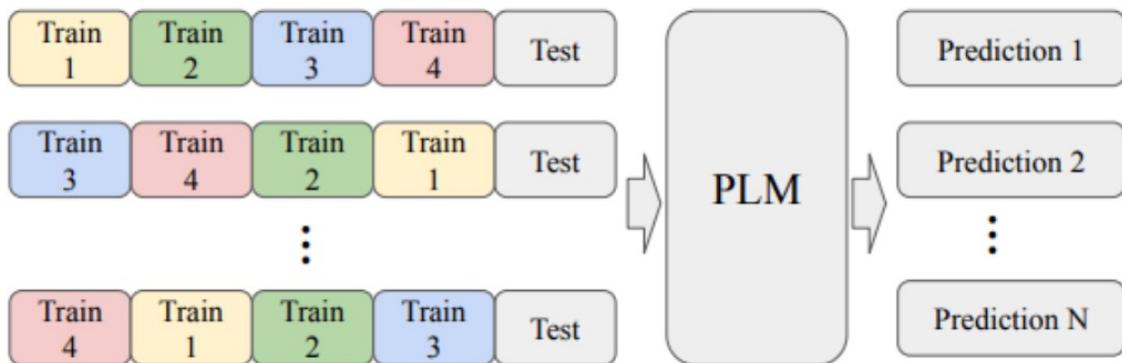
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

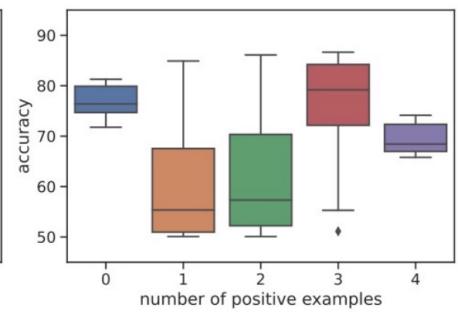


Few-shot in-context learning

- In-context learning can be sensitive to...
 - 1. the order the training examples are presented
 - 2. the balance of labels (e.g., positive vs. negative)
 - 3. the number of unique labels covered



(a) Amazon

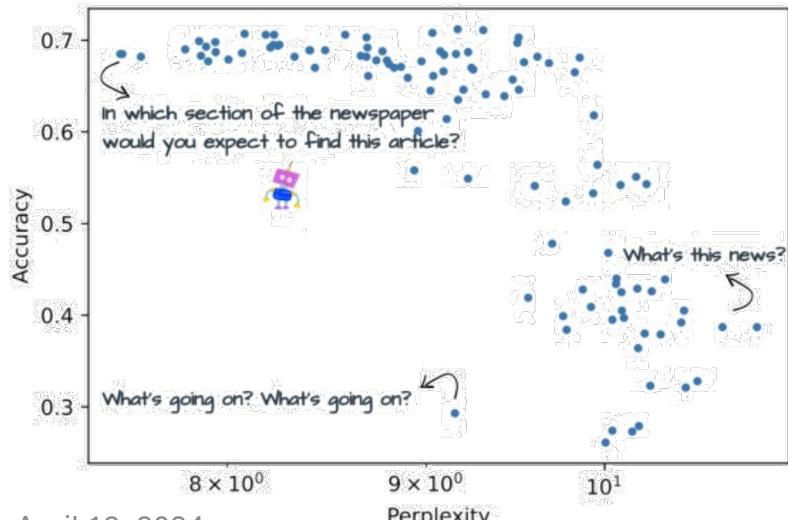


(b) SST-2

Figure 3: Accuracies of Amazon and SST-2 with varying **label balance** (number of positive examples in demonstration), across 100 total random samples of 4 demonstration examples.

Prompt engineering

- Task: News topic classification
- Dataset: AG News
- Model: OPT-175B
- Setup: zero-shot learning



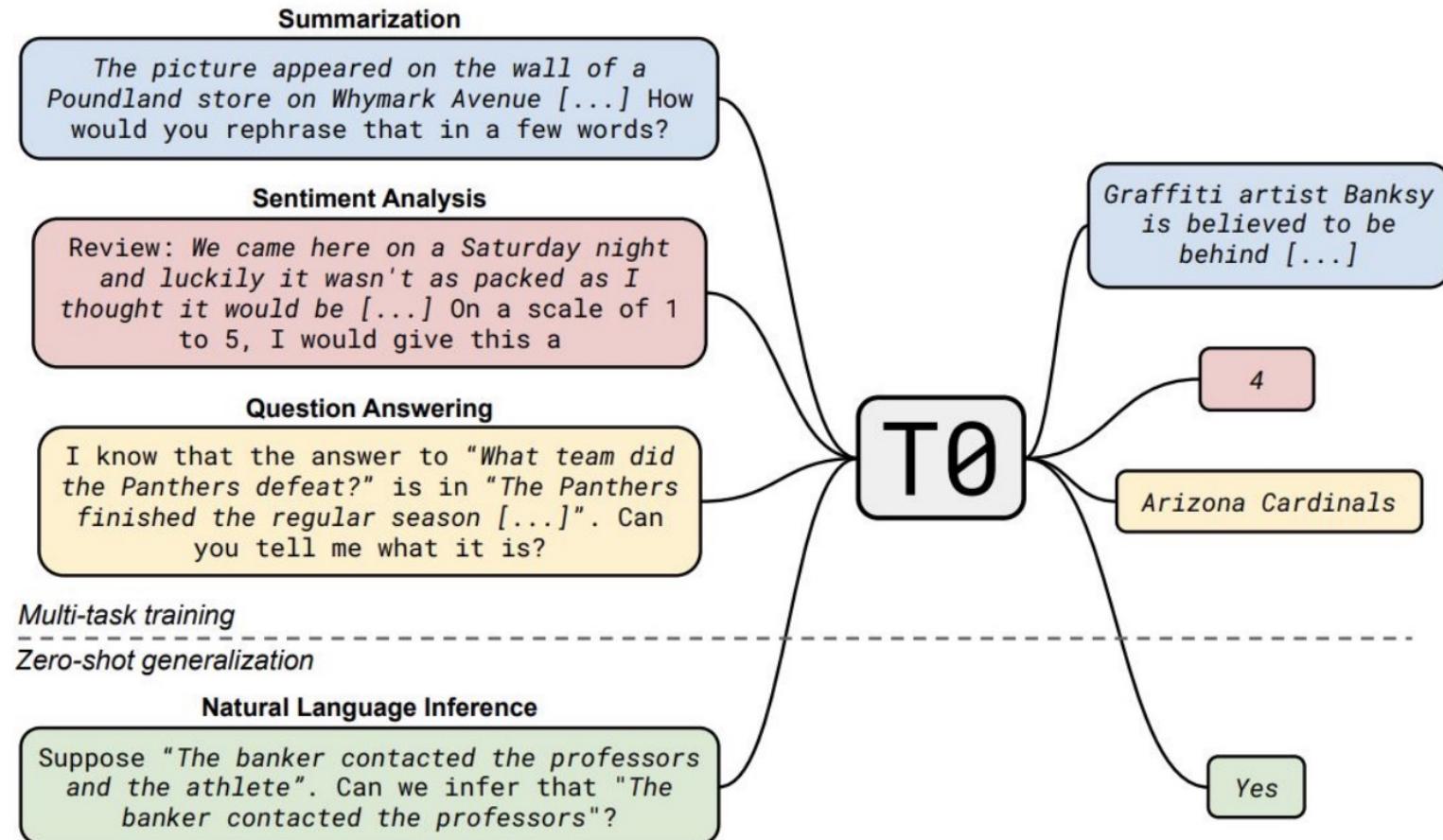
Question: if we evaluate the model multiple times keeping everything fixed except for the prompt, do we always get the same results?

Prompt	Accuracy
What is this piece of news regarding?	40.9
What is this article about?	52.4
What is the best way to describe this article?	68.2
What is the most accurate label for this news article?	71.2

Question: how can we pick a good prompt?
Answer: pick the prompt with the lowest **perplexity** under the model!

$$\text{PPL}(X) = \exp \left\{ -\frac{1}{t} \sum_i^t \log p_\theta(x_i | x_{<i}) \right\}$$

Multi-task learning for prompting



Multi-task learning for prompting

Given (x, y) , maximize $P_{LM}(v(y)|f(x))$ where...

- (x, y) is from a dataset sampled from a large collection of datasets
- f and v are sampled from a collection of different formats

- Prompt to include in-context examples
- Prompt to include natural language description about the task
- Verbalizer to include rationale about the output

An effortlessly accomplished and richly resonant work. It was great!
A mostly tired retread of several other mob tales. It was terrible!

"Identify the sentiment of this movie review."

Question: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left?

Answer: Originally, Leah had 32 chocolates and her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39 pieces left in total. The answer is 39.

Chain-of-thought prompting

- Asking the model to reason about its answer can improve its performance for few-shot in-context learning
- Chain-of-thought prompting provides such reasoning in the in-context examples

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. X

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. ✓

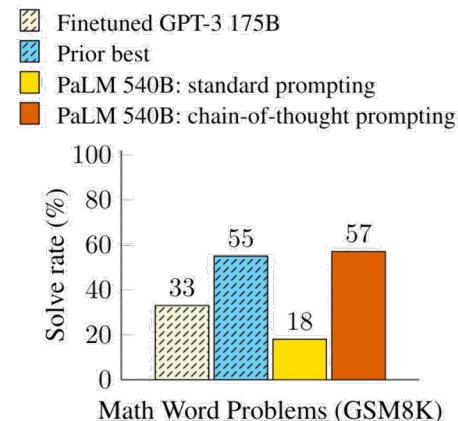


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).

Chain-of-thought prompting

- But the model does better even if you just prompt it to reason step-by-step

(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. ✓