# CS 839: Foundation Models
# **Models II + Prompting Start**

Fred Sala

University of Wisconsin-Madison

**Sept. 24, 2024**

# Announcements

- **Logistics:**
  - Homework 1 is ongoing!

- Class roadmap:

| Tuesday Sept. 24 | Models II |
|---|---|
| Thursday Sept. 26 | Prompting I |
| Tuesday Oct. 1 | Prompting II |
| Thursday Oct. 3 | Specialization |
| Thursday Oct. 8 | Alignment |

Mostly Language Models

# Outline

- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models
- **Intro to Prompting**
  - Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders
- **Improving Prompting**
  - Searching for good prompts, techniques for continuous/soft prompts, ensembling

# Outline

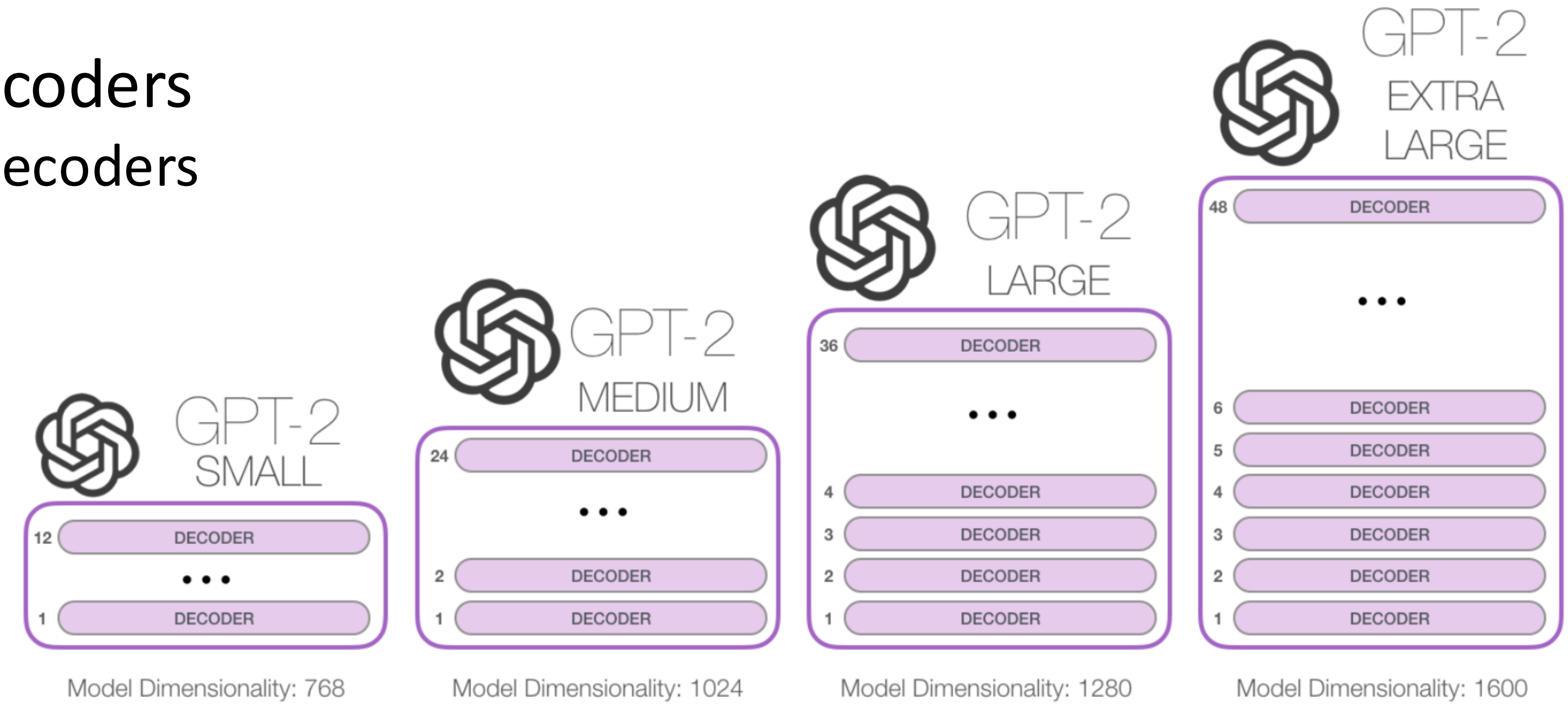
- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models
- **Intro to Prompting**
  - Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders
- **Improving Prompting**
  - Searching for good prompts, techniques for continuous/soft prompts, ensembling

# **Decoder-Only Models:** GPT

Let's get rid of the second requirement we had before,

- 1) **Outputs** in natural language
- 2) Tight alignment with **input**

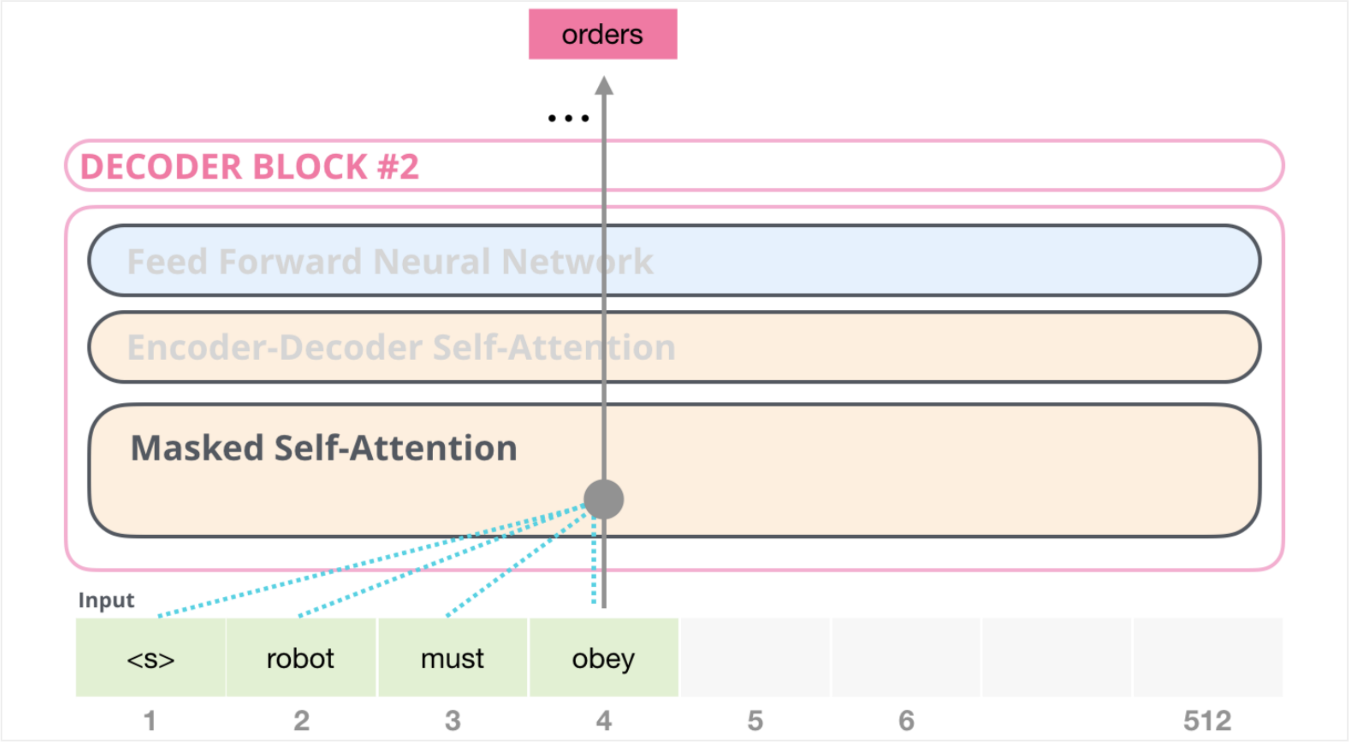
- Rip away encoders
  - Just stack decoders

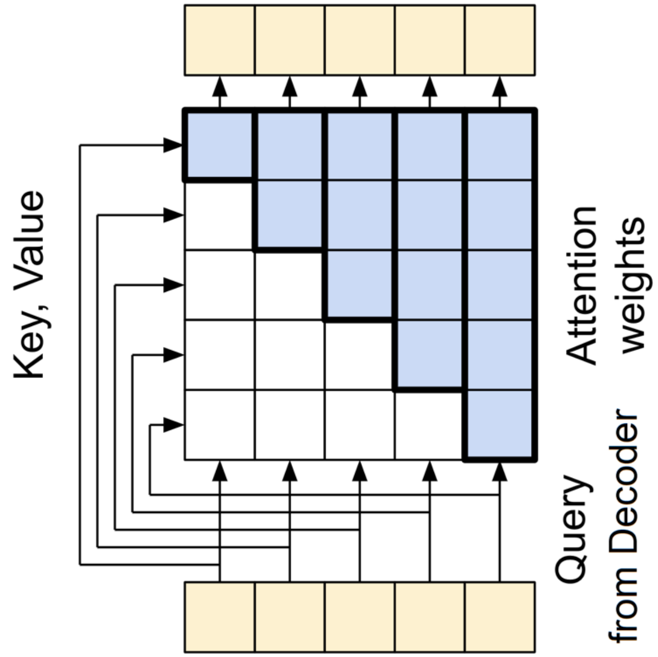


# **Decoder-Only Models:** GPT

Rip away encoders

- Just stack decoders
- Use causal masking! NB: not a *mask token* like in BERT



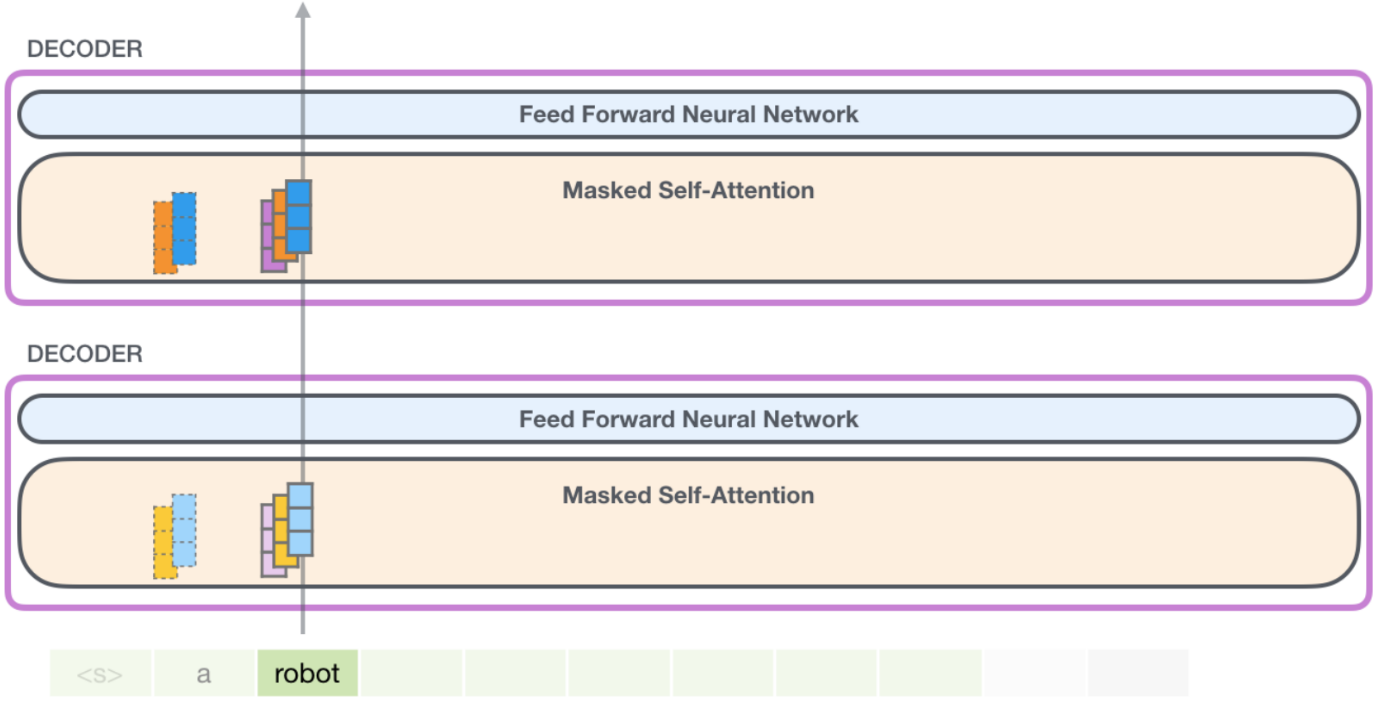



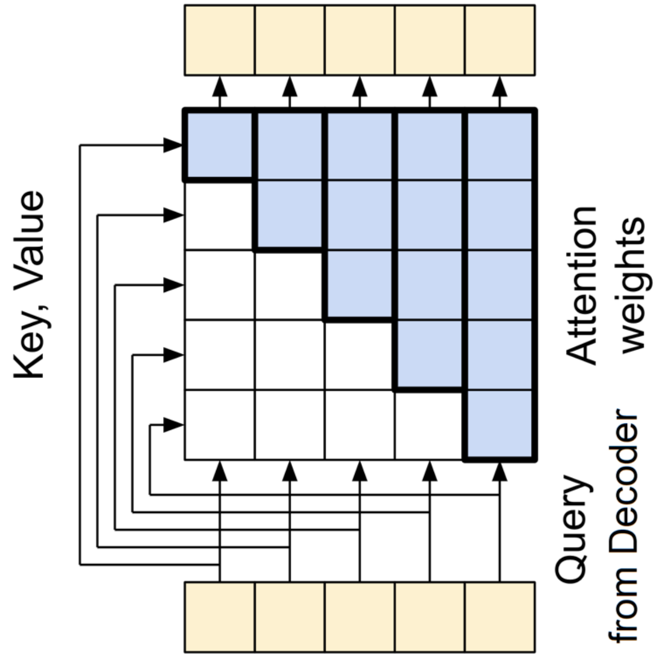

PyLessons

# **Decoder-Only Models:** GPT

Rip away encoders

- Just stack decoders
- Decoders: get rid of **encoder** aspects (masked self-attention only)




PyLessons

# From GPT2 to GPT3

Mainly make things larger!

- 96 decoder blocks (getting very tall)
- Context size: **2048**
- 175 billion parameters in total (800GB!)

## Training data:

GPT-3 training data[1]:9

| Dataset | # tokens | Proportion within training |
|---|---|---|
| Common Crawl | 410 billion | 60% |
| WebText2 | 19 billion | 22% |
| Books1 | 12 billion | 8% |
| Books2 | 55 billion | 8% |
| Wikipedia | 3 billion | 3% |

https://en.wikipedia.org/wiki/GPT-3

Brown et al '20

# Open Source: Llama 3.1

Mainly make things larger! Note: multiple model sizes:

| | 8B | 70B | 405B |
|---|---|---|---|
| Layers | 32 | 80 | 126 |
| Model Dimension | 4,096 | 8192 | 16,384 |
| FFN Dimension | 14,336 | 28,672 | 53,248 |
| Attention Heads | 32 | 64 | 128 |
| Key/Value Heads | 8 | 8 | 8 |
| Peak Learning Rate | $3 \times 10^{-4}$ | $1.5 \times 10^{-4}$ | $8 \times 10^{-5}$ |
| Activation Function | SwiGLU | | |
| Vocabulary Size | 128,000 | | |
| Positional Embeddings | RoPE ($\theta = 500,000$) | | |

Dubey et al '24

# Open Source: Llama 3.1

Some improvements for Llama 3.1:

- “We use an attention mask that **prevents self-attention between different documents** within the same sequence”
- “**grouped query attention** (GQA; Ainslie et al. (2023)) with 8 key-value heads to improve inference speed…”
- “We use a **vocabulary with 128K tokens.** Our token vocabulary combines 100K tokens from the tiktoken3 tokenizer with 28K additional tokens to better support non-English languages”

Zhao et al '21

# Break & Questions

# Outline

- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models
- **Intro to Prompting**
  - Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders
- **Improving Prompting**
  - Searching for good prompts, techniques for continuous/soft prompts, ensembling

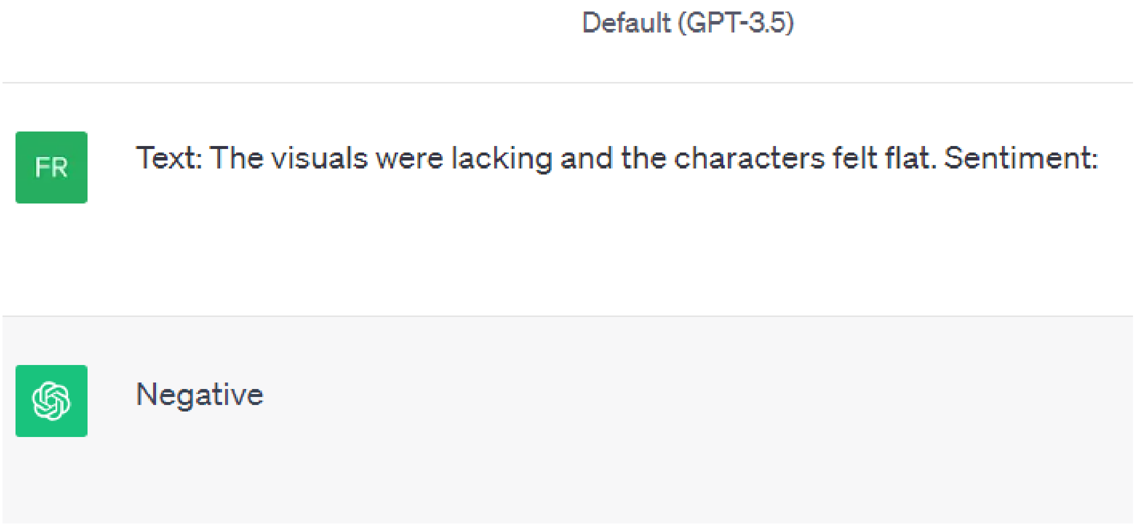
# Prompting: Ask Your Model

Essentially, ask your model to perform your goal task

**Example**: sentiment analysis task

- Prompt: “`Text: The visuals were lacking and the characters felt flat. Sentiment:`”

- Result: “`Negative`”



# Prompting: Zero-shot vs Few-shot

Terminology:

- **Zero-shot:** No "examples" provided to the model.
- **Few-shot/in-context learning**: Provide "examples"

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment:

Zhao et al '21

Positive

# Prompting: Few-shot vs. In-context learning

Terminology conflicts! Note: we have a set of labeled examples. Could **fine-tune**!

**Few-shot:** *sometimes* means fine-tune on this dataset, then prompt

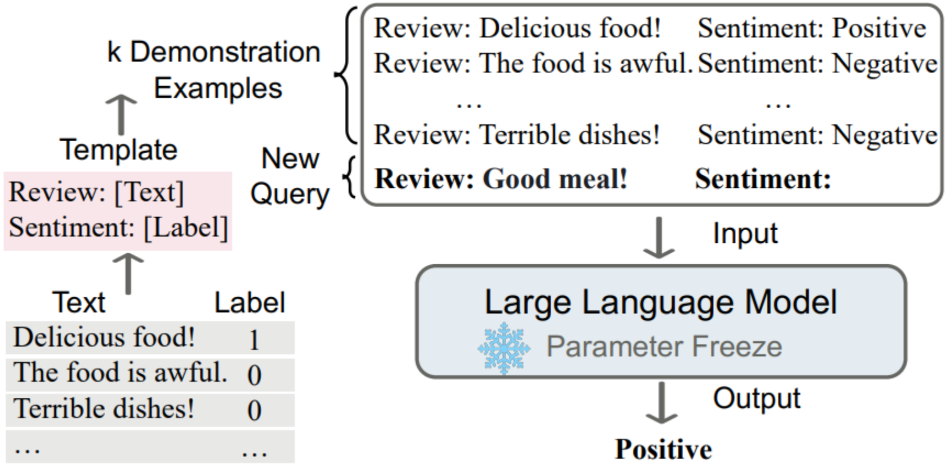
**In-context learning**: do not fine-tune. Model weights unchanged.

```
Text: (lawrence bounces) all over the stage, dancing,
Sentiment: positive

Text: despite all evidence to the contrary, this clun
Sentiment: negative

Text: for the first time in years, de niro digs deep
Sentiment: positive
```

Weng / SST

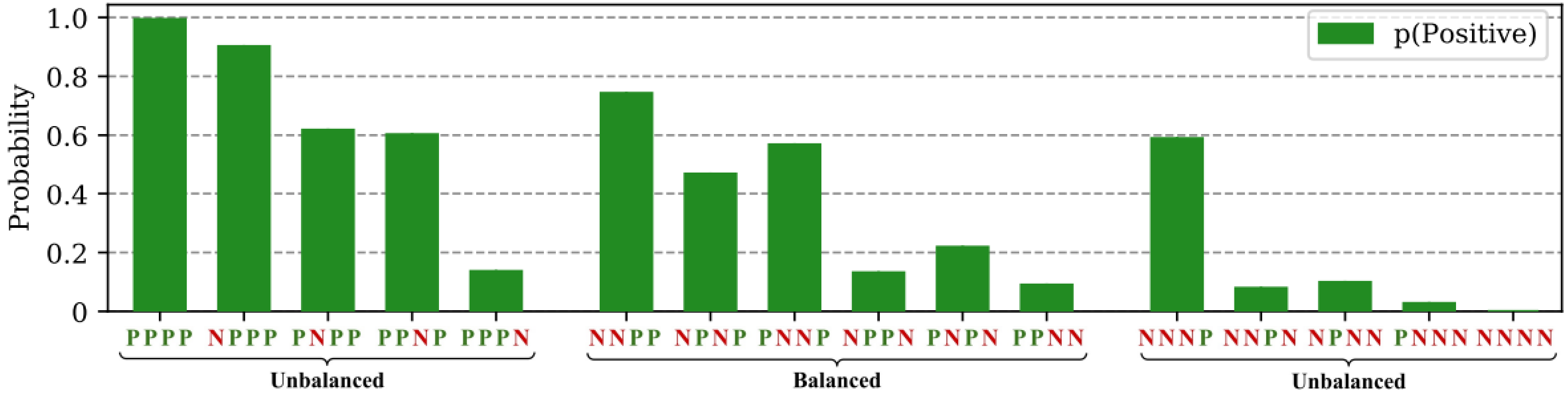



Dong et al, ‘23

# Few-Shot Choices

Examples/structure affect performance:

1. Prompt **format** (affects everything)
2. **Choice** of examples
3. **Order** of examples (permutation)




Zhao et al '21

# 1. Prompt Formats

The choice of model affects the prompt format

**Masked language model:** “Cloze”-style prompt

- “`I love this movie, it is a [Z] movie:`”

**Left-to-right language model:** prefix prompt

- “`I love this movie. What is the sentiment of this review?`”

Note: eval datasets have pre-created prompts.

- LAMA (LAnguage Model Analysis): Cloze prompts

# **1.** Prompt Formats: **Recent Models**

Modern instruction-tuned models have more complex instructions/formats

- **The good:** more natural way to tell the model what to do
- **The bad:** searching over formats/templates increasingly challenging

  - *Example: (White et al, ‘23):* `"From now on, I would like you to ask me questions to deploy a Python application to AWS. When you have enough information to deploy the application, create a Python script to automate the deployment.”`

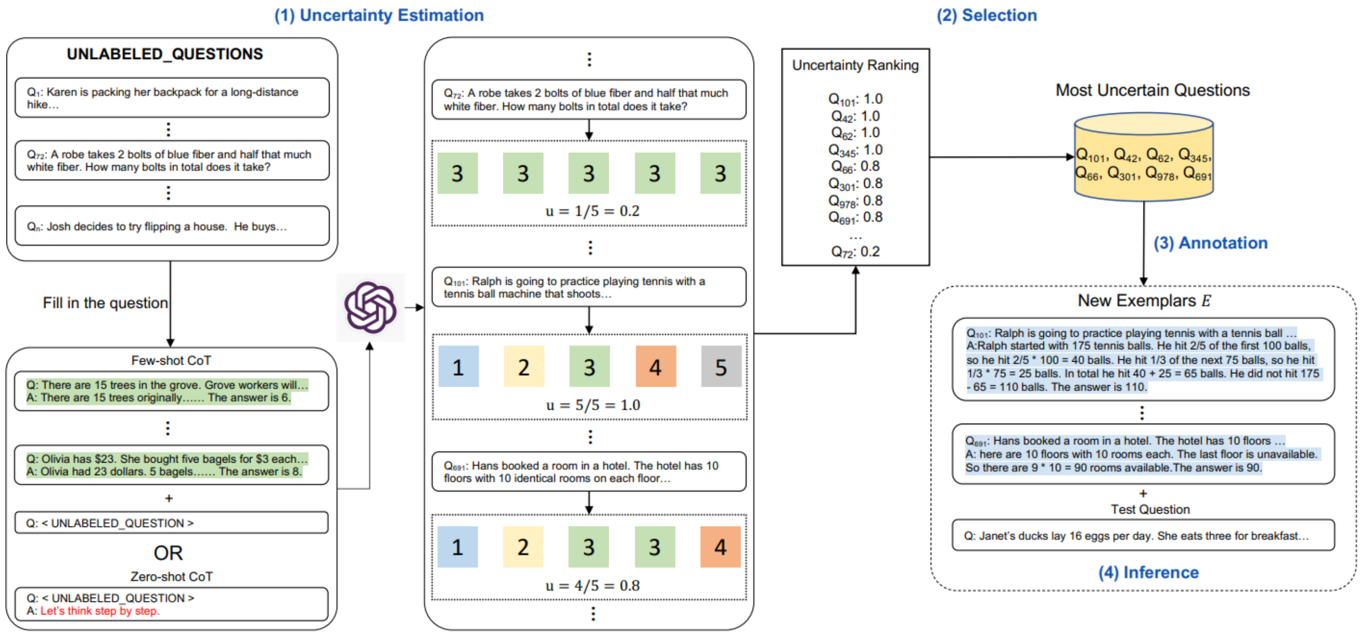
# 2. Choice of Examples

How to pick appropriate examples in few-shot?

- **Note:** only a "small' number of examples can be shown, unlike in supervised learning.

Many options. Sampling:

- Liu et al, '21: kNN in embedding space (semantic similarity)
- Su et al, '22: Encourage diversity in embeddings
- Diao et al, '23: "Active prompting"




Diao et al '23

# 3. Order of Examples

What order to show them to the model?

**Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity**

**Yao Lu**[†] **Max Bartolo**[†] **Alastair Moore**[‡] **Sebastian Riedel**[†] **Pontus Stenetorp**[†]
[†]University College London [‡]Mishcon de Reya LLP
{yao.lu,m.bartolo,s.riedel,p.stenetorp}@cs.ucl.ac.uk
alastair.moore@mishcon.com

- **Findings:**
  - Model size doesn’t guarantee low-variance
  - Adding more examples doesn’t reduce variance
  - Good prompts don’t transfer from one model to another ☹
  - Good orders don’t transfer

# Break & Questions

# Outline

- **Decoder-only Models**
  - Example: GPT, architecture, basic functionality, properties of new models
- **Intro to Prompting**
  - Terminology: zero-shot, few-shot, in-context, etc, prompt characteristics: format, examples, orders
- **Improving Prompting**
  - Searching for good prompts, techniques for continuous/soft prompts, ensembling

# Hard Prompting

Also called **zero-shot**.

- Note: terminology conflict with another area called zero-shot learning

"Hard prompt discovery is a specialized alchemy, with many good prompts being discovered by trial and error, or sheer intuition

(Wen et al '23)

- Note: not just for language models!

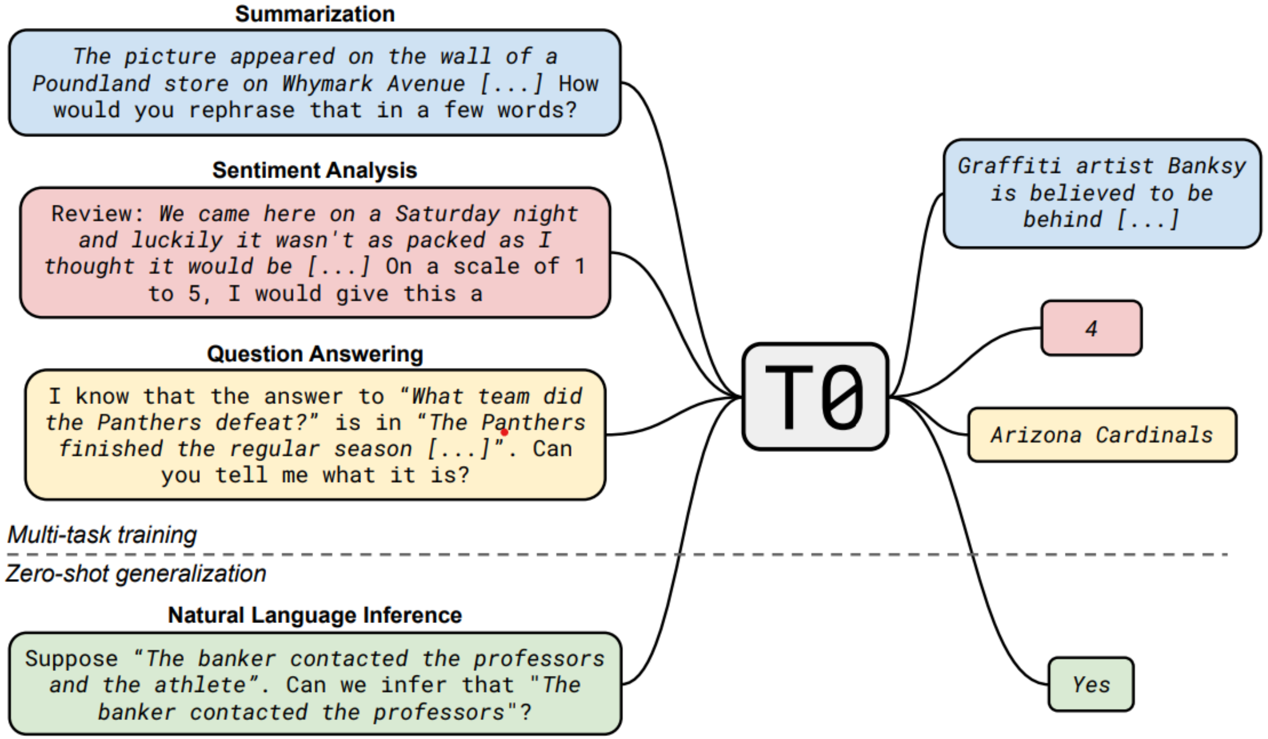
# Zero-shot Generalization

Most exciting aspect of zero-shot: don’t need to have been explicitly trained or fine-tuned.

- **Example: Multitask Prompted Training Enables Zero-Shot Task Generalization**

Recipe

- Pretrain
- Fine-tune
  - Multitask

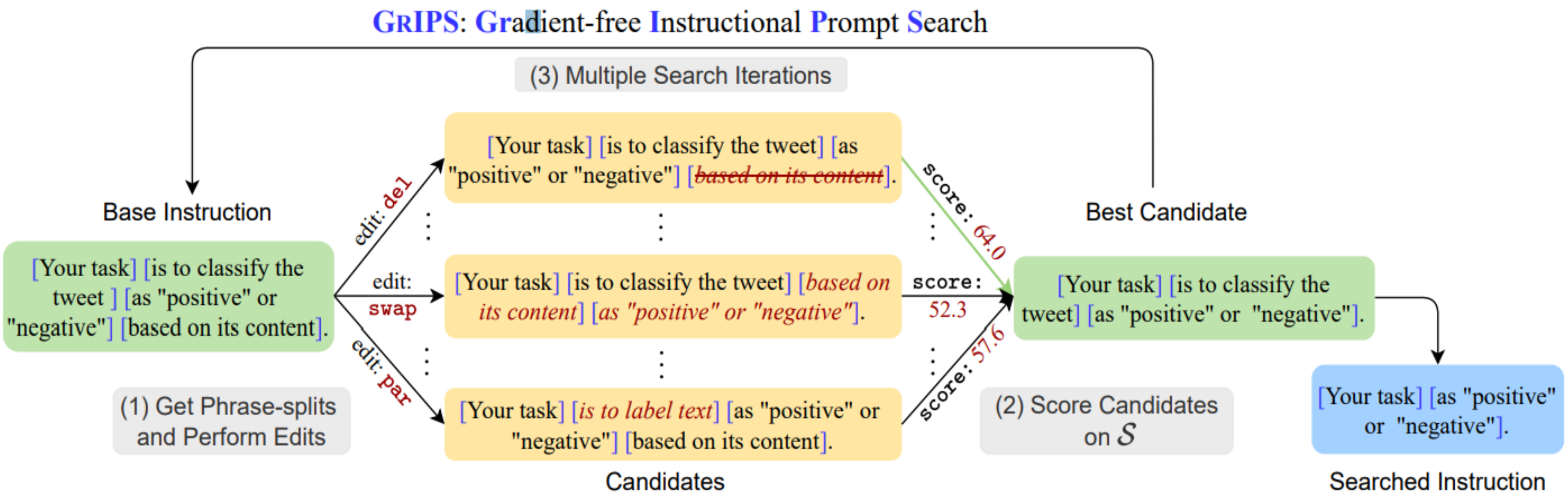



Sanh et al ‘22

# Hard Prompting: **Discrete Optimization**

Sometimes, can avoid gradients

- Random search
- Greedy




Prasad et al '23

# Soft Prompting

Also called **continuous prompting**

Basic idea: insert some (non-language) parameters into prompt

- Train these parameters
- Do not directly correspond to words in prompt

**Prefix-Tuning: Optimizing Continuous Prompts for Generation**

**Xiang Lisa Li**
Stanford University
xlisali@stanford.edu

**Percy Liang**
Stanford University
pliang@cs.stanford.edu

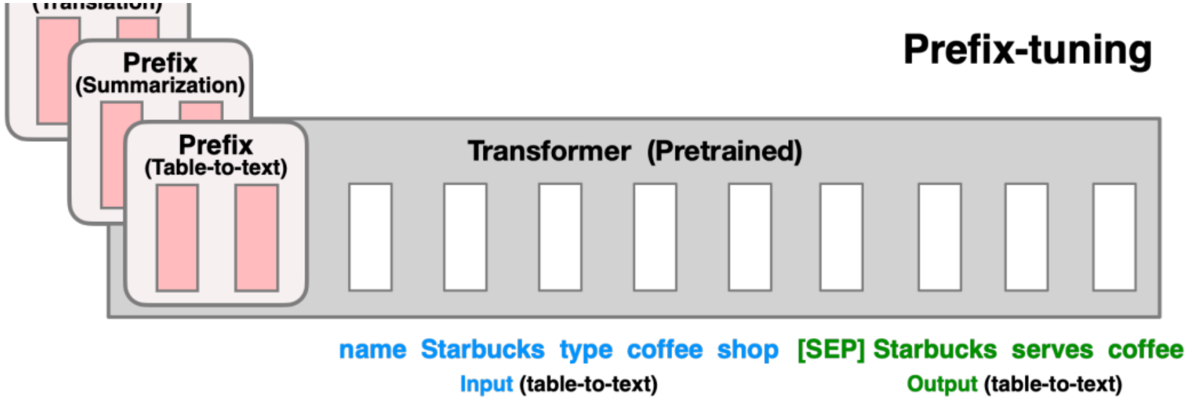
**GPT Understands, Too**

**Xiao Liu**[*1,2] **Yanan Zheng**[*1,2] **Zhengxiao Du**[1,2] **Ming Ding**[1,2] **Yujie Qian**[3] **Zhilin Yang**[4,2] **Jie Tang**[1,2]

# Soft Prompting: **Prefix-Tuning**

Goal: create prefixes that *steer* models

- Prefixes are trainable parameters
- Train one for each goal task, only store these new parameters
- Enables cheap adaptation of frozen language model

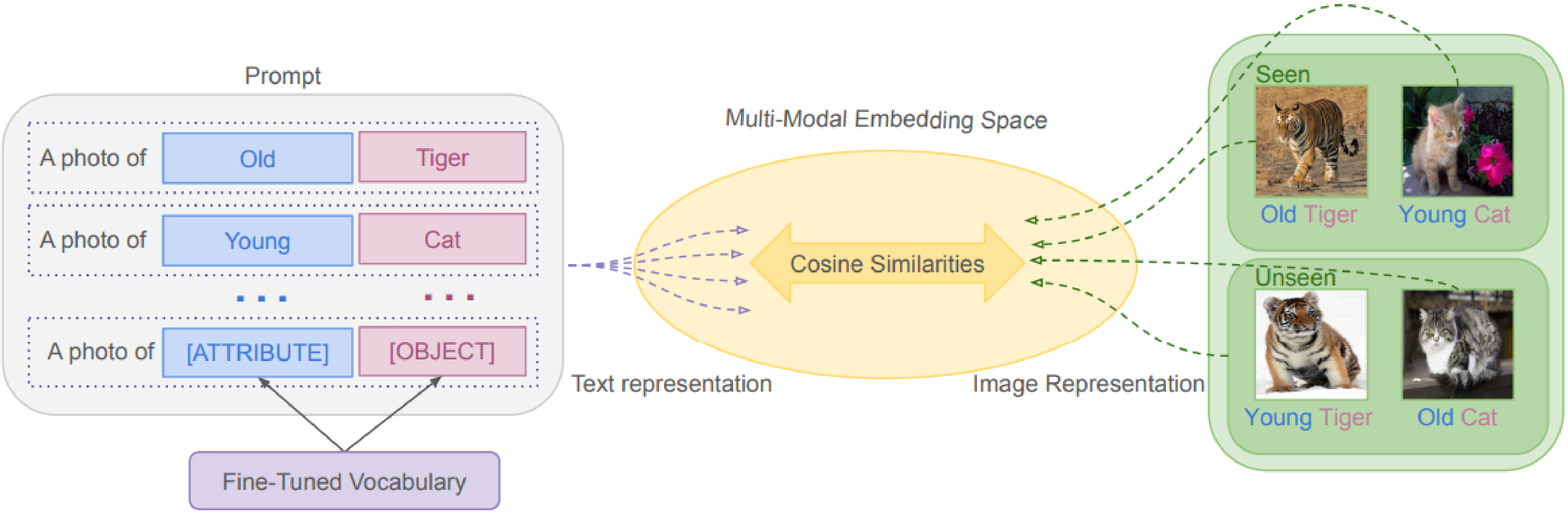



Li and Liang ‘21

# Soft Prompting: **Composing**

What about multimodal models?

- Vision-language models like CLIP
- Not great composed concepts like *old tiger*
- Tune on [attribute] [object] pairs




Nayak et al '21

# Ensembling Prompts

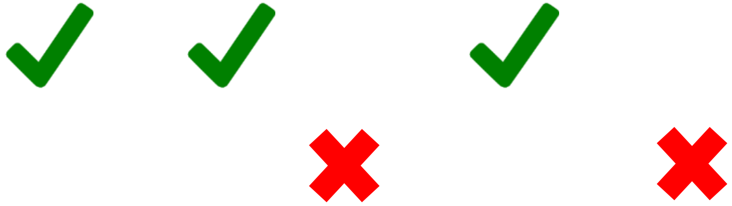
One prompt can give you an answer… but might be wrong

- One simple approach: get multiple samples
- From?
  - Change temperature parameter
  - Vary your prompts

Then, run **majority vote**

Witnesses

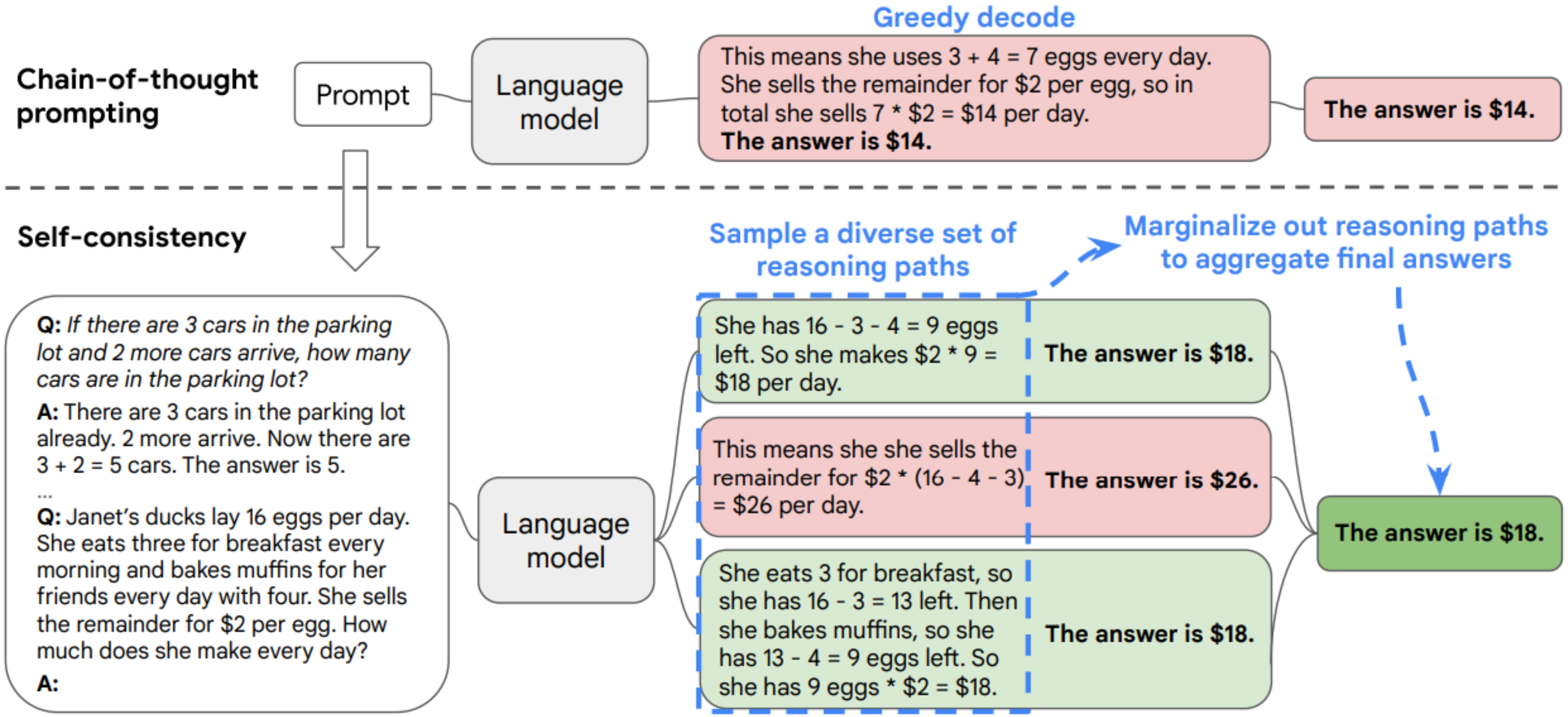
Votes



# Chain-of-Thought

A form of prompting that helps break down the problem (more in a week!)

- Produces more answers to run majority vote on

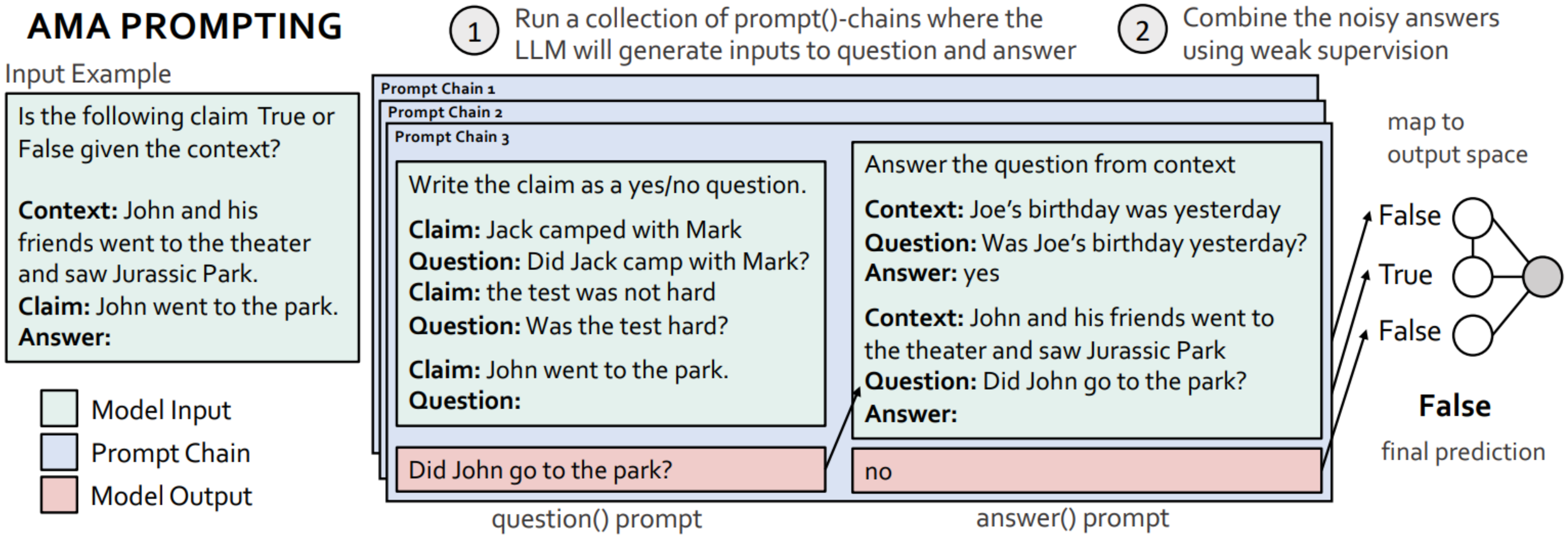



Wang et al '23

# Ensembling Prompts: Weighted Version

Downside of majority vote… most responses might be wrong

- Should weight them by how accurate they are




Chen et al, '23

# Bibliography


- Brown et al ‘20: Brown+many others, “Language Models are Few-Shot Learners” (https://arxiv.org/abs/2005.14165)
- Dubey et al ‘24: Dubey+many others, “The Llama 3 Herd of Models” (https://arxiv.org/abs/2407.21783)
- Zhao et al ’21: Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, Sameer Singh, “Calibrate Before Use: Improving Few-Shot Performance of Language Models” (https://arxiv.org/abs/2102.09690)
- Dong et al ‘23: Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, Zhifang Sui, “A Survey on In-context Learning” (https://arxiv.org/abs/2301.00234)
- Zhou et al ‘23: Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, Jimmy Ba, “Large Language Models Are Human-Level Prompt Engineers” (https://arxiv.org/abs/2211.01910)
- Yang et al ‘23: Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, Xinyun Chen, “Large Language Models as Optimizers” (https://arxiv.org/abs/2309.03409)
- Menon and Vondrick ‘23, “Visual Classification via Description from Large Language Models” (https://arxiv.org/abs/2210.07183)
- Adila ‘23: Dyah Adila, Changho Shin, Linrong Cai, Frederic Sala, “Zero-Shot Robustification of Zero-Shot Models With Foundation Models” (https://arxiv.org/pdf/2309.04344.pdf)
- Gal et al ‘22: Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, Daniel Cohen-Or, “An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion” (https://arxiv.org/abs/2208.01618)
- Zhang et al ‘23: Yuanhan Zhang, Kaiyang Zhou, Ziwei Liu, “What Makes Good Examples for Visual In-Context Learning?” (https://arxiv.org/abs/2301.13670)
- Wei et al ’22: Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, Denny Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models” (https://arxiv.org/abs/2201.11903)
- Kojima et al ‘23: Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, Yusuke Iwasawa, “Large Language Models are Zero-Shot Reasoners” (https://arxiv.org/abs/2205.11916)
- Fu et al ‘23: Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, Tushar Khot, “Complexity-Based Prompting for Multi-Step Reasoning” (https://arxiv.org/abs/2210.00720)
- Yao et al ’23: Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, Karthik Narasimhan, “Tree of Thoughts: Deliberate Problem Solving with Large Language Models” (https://arxiv.org/abs/2305.10601)

# Thank You!