

ABB - Session 3

Software 3.0, LLMs, Prompt Engineering

Shaw Talebi

Today's Session

1. Housekeeping

- 1.1. Announcements
- 1.2. Homework 2

2. Software 3.0 ↗

- 2.1. Large Language Models (LLMs)
- 2.2. Two Levels of LLM Development
- 2.3. Prompt Engineering

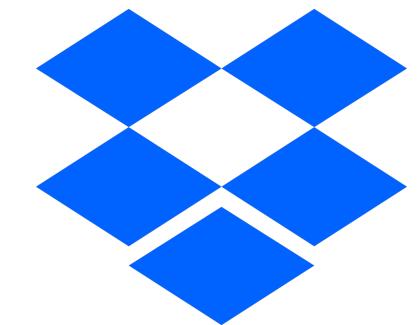
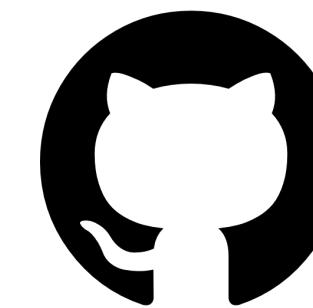
3. Examples ↗

- 3.1. Summarizing Research Papers with GPT-4o
- 3.2. Text Classification with GPT-4o-mini
- 3.3. Local Document QA with Ollama



Announcements

- 1) Pre- and post-lecture discussion (going well)
- 2) Reminder: How to upload HW



Homework 3

Shoutouts

Stock Picker

Vladimir Belony

Predicting Housing Prices

Rod Morrison

Support Ticket Classification

Serg

Predicting Natural Gas Prices

Sangeeta Bahri

Fraud Detection

Dario Zandolin

Credit Risk Modeling

Kalyan Mutyala

Reddit Post Classifier

Sanjeev NC

The Problem with Software 2.0

Getting *good* data is hard



14M images

49k human labelers

Millions of dollars invested

(deduping, quality control, remove copyrighted content)

AlexNet

**ImageNet Classification with Deep Convolutional
Neural Networks**

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

1.2M Images

2 Consumer GPUs trained for 5-6 days

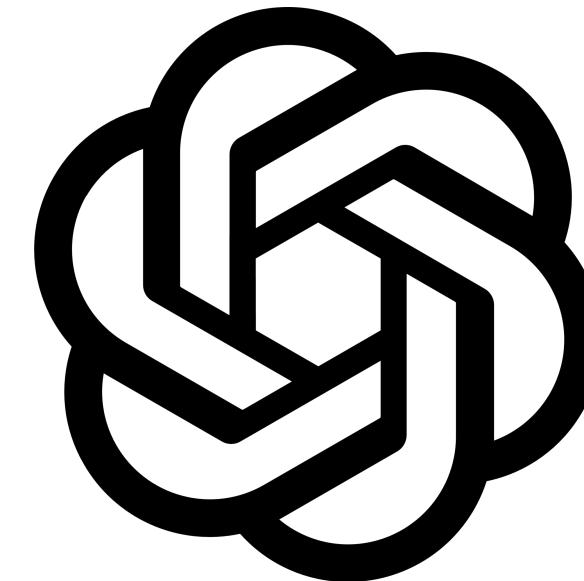
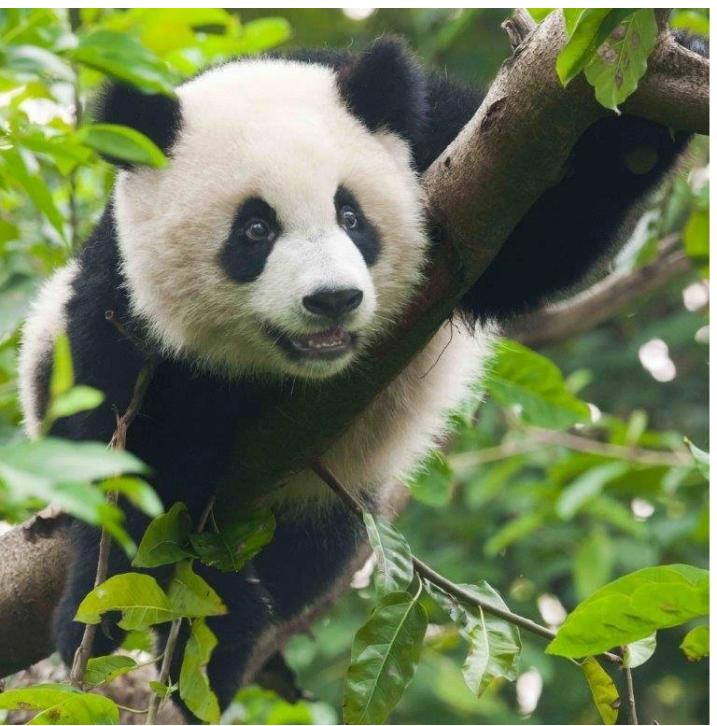
... still wrong 15% of the time.

But what if there's another way?

Software 3.0

Software 3.0

Adapting pre-trained (general-purpose) models for specific use cases



GPT-4o

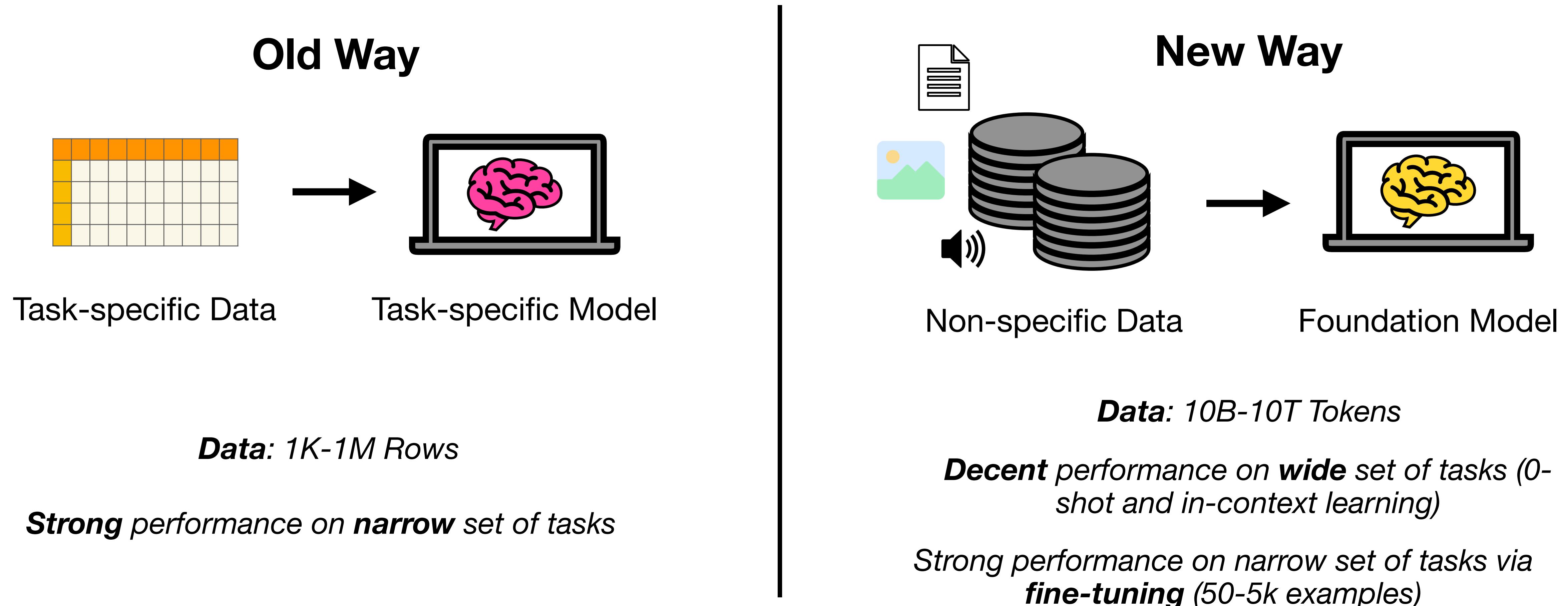
"giant panda"

n02510455

What object would this
be classified as in the
image net dataset?

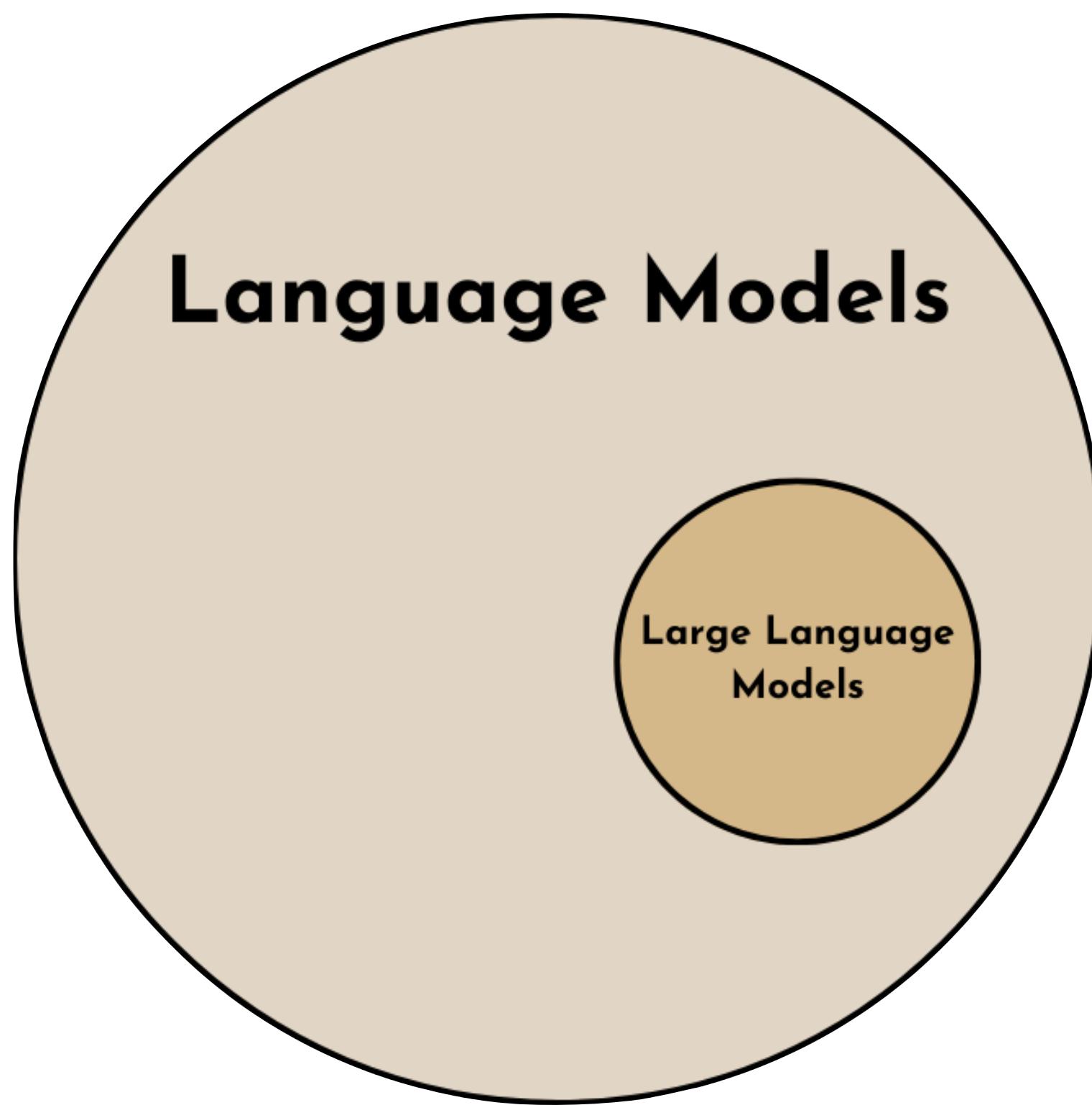
What's so different about this?

Upgrading from 2.0 to 3.0



Large Language Models (LLMs)

(Very) big models that can perform wide range of NLP tasks



Quantitatively

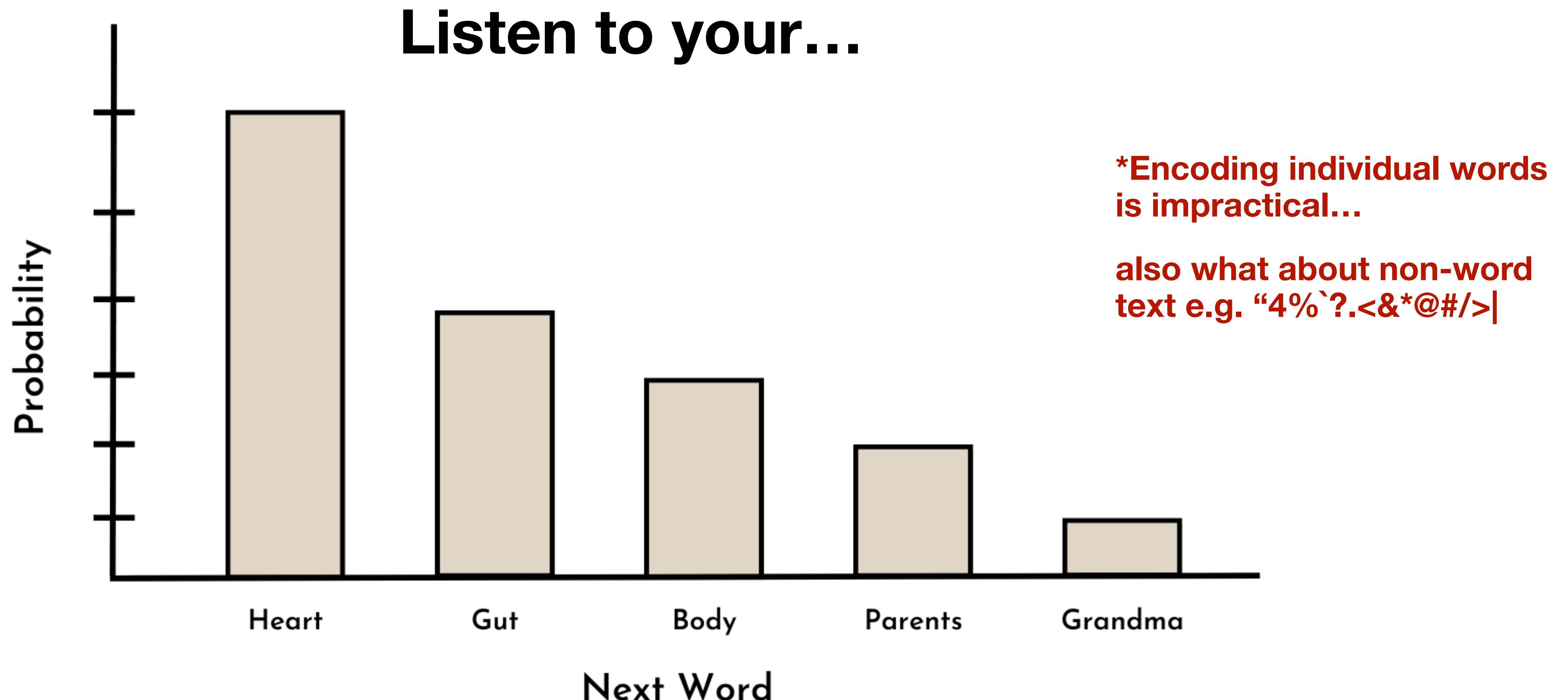
Number of model parameters
i.e. 1-100+ Billion

Qualitatively

Emergent properties
e.g. Zero-shot learning

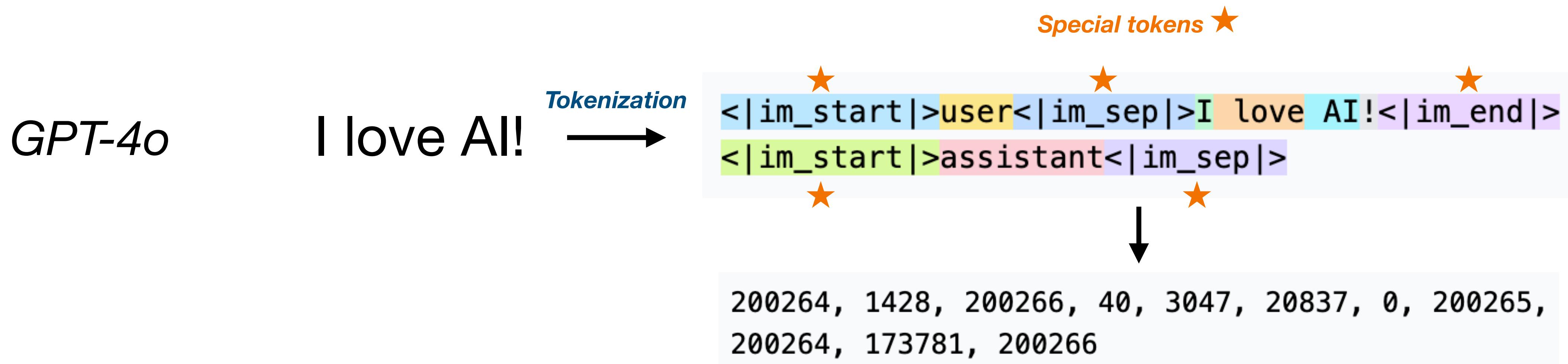
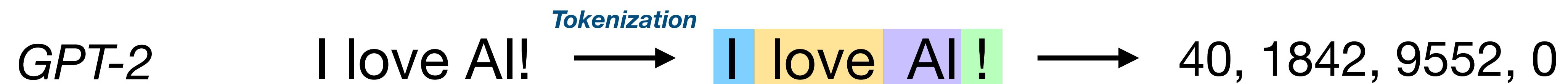
Large Language Models (LLMs)

Basic idea: Next-word prediction



Tokens

Character strings serving as basis of LLM inputs/outputs



Vocabulary = the set of all tokens used for a particular LLM

2 Levels of LLM Development

How to get LLMs to do what you want...

Level 1

Adapting models via prompts and tools

Prompt Engineering



RAG



Tool-use



Level 2

Adapting models via additional training

Fine-tuning



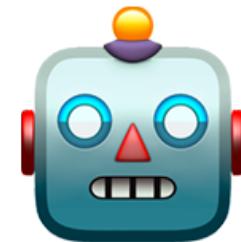
Post-training



Prompt Engineering

Adapting model behavior through (natural language) inputs

Can you write me a simple and concise definition of Prompt Engineering?



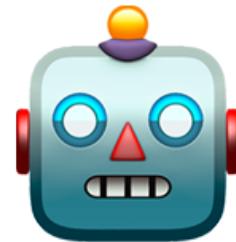
Prompt Engineering is the process of designing and optimizing input prompts to guide AI models, like ChatGPT, in producing desired and accurate outputs. It involves crafting specific instructions, context, and examples to achieve effective and reliable results.

Prompt Engineering

Adapting model behavior through (natural language) inputs

What is the tone of sentiment of the following message?

"I love prompt engineering. It's much easier than writing code :P"



The tone of the sentiment in the message is **positive** and playful. The use of "I love" conveys enthusiasm, and the emoticon ":P" adds a lighthearted and humorous touch.

Prompt Engineering

A few tricks to improve model performance

Trick 1: Be Descriptive

Summarize this paper.



Your task is to act as an academic summarizer, condensing research papers into concise, clear, and accessible summaries.

Follow these guidelines:
{Guidelines here}



Hint: using an LLM is a great starting point for prompt writing

Trick 2: Give Examples

Here are some examples of sentiment analysis:

Input: "I love the new design of this app!"

Sentiment: Positive

Input: "This service is okay, but it could be better."

Sentiment: Neutral

Input: "I'm extremely disappointed with the support team."

Sentiment: Negative

Now, analyze the sentiment of this statement:

Input: {New example}

Trick 3: Use Structured Text

Here are some examples of sentiment analysis:

1. **Input**: "I love the new design of this app!"

Sentiment: Positive

2. **Input**: "This service is okay, but it could be better."

Sentiment: Neutral

3. **Input**: "I'm extremely disappointed with the support team."

Sentiment: Negative

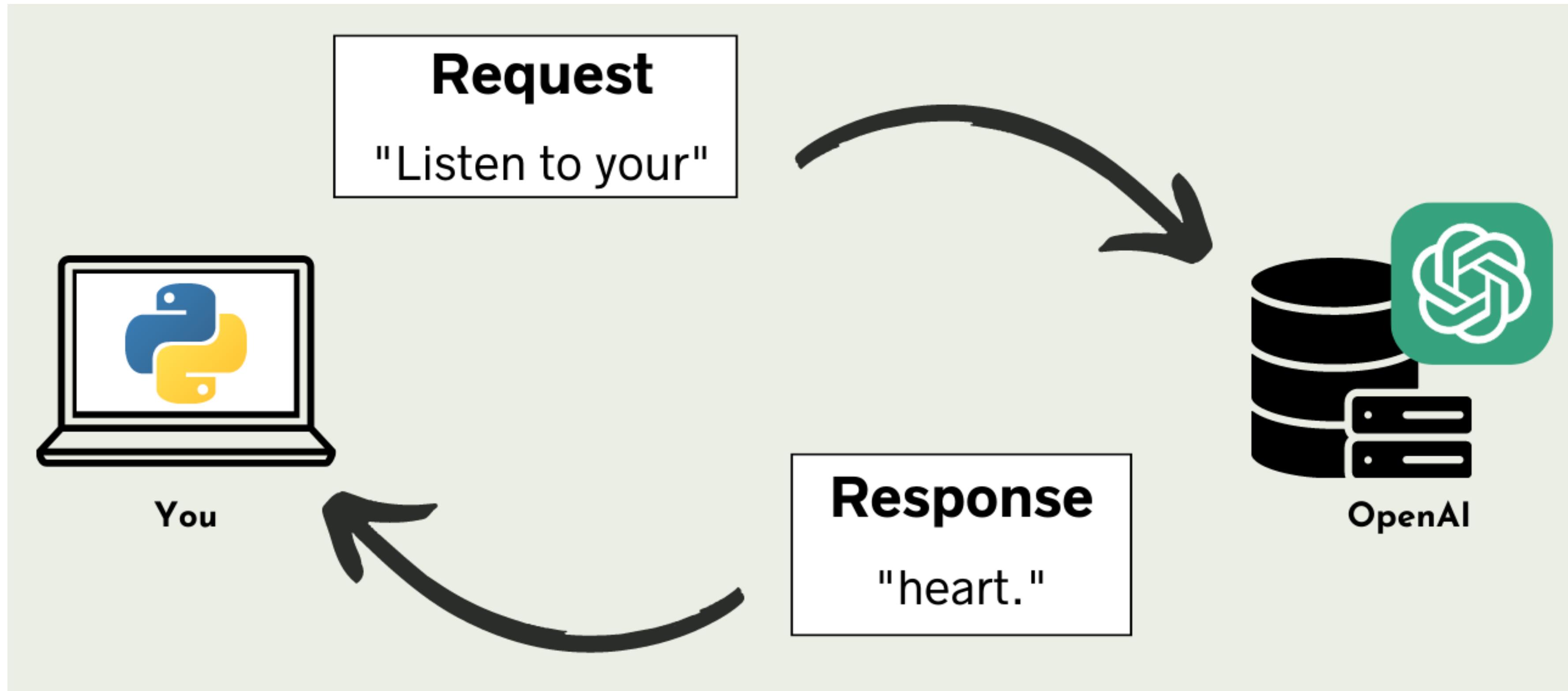
Now, analyze the sentiment of this statement:

Input: {New example}

Often better prompts are the best way to improve an LLM system

Using LLM APIs

Running model inference without GPUs



Pricing based on token count and model size

Examples

Example 1

Summarizing Research Papers with GPT-4o (Overview)

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Ilia Polosukhin* ‡
ilia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



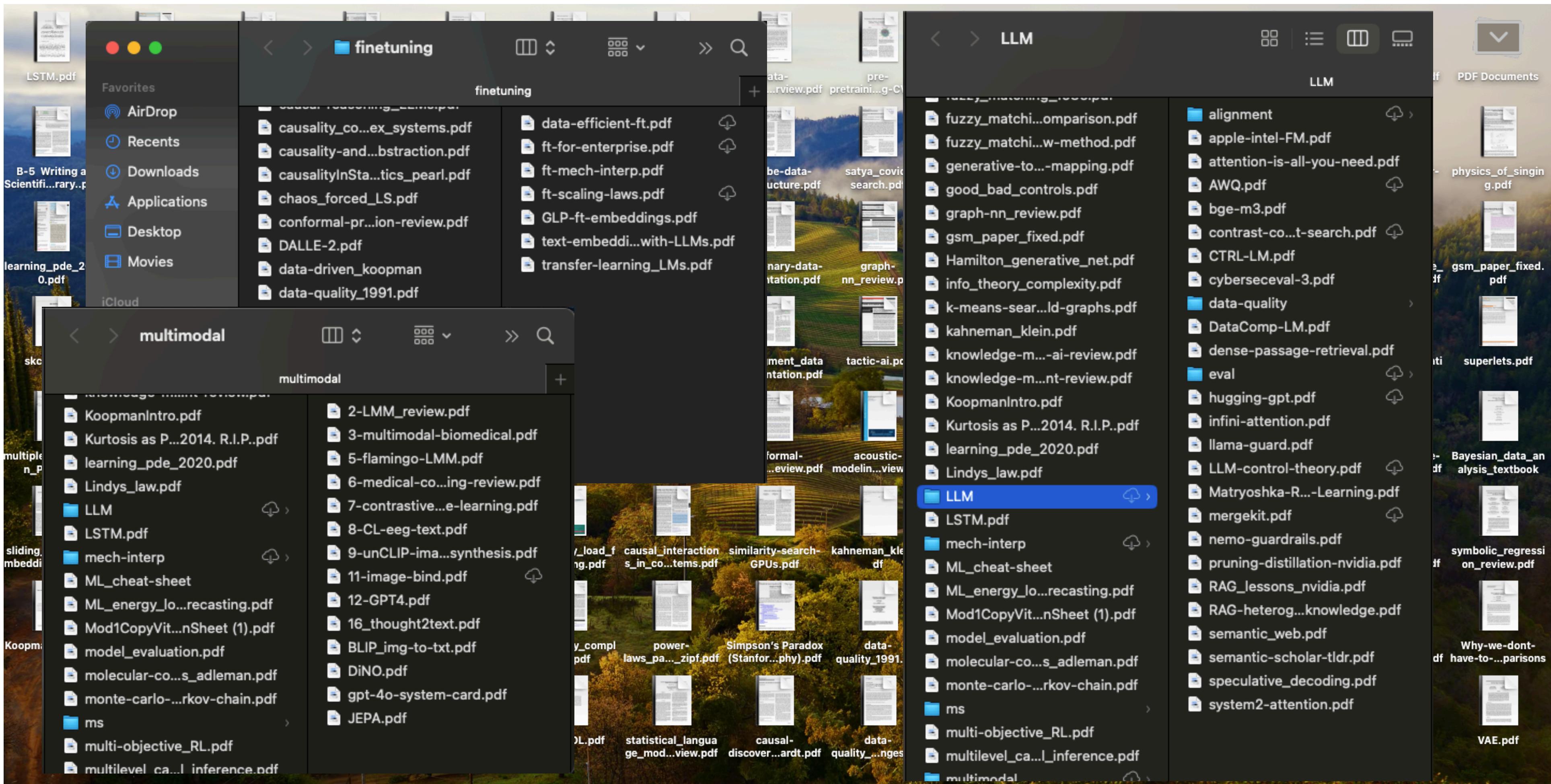
Lit Review:

Title
Authors/Affiliations
Key Concepts
Contributions



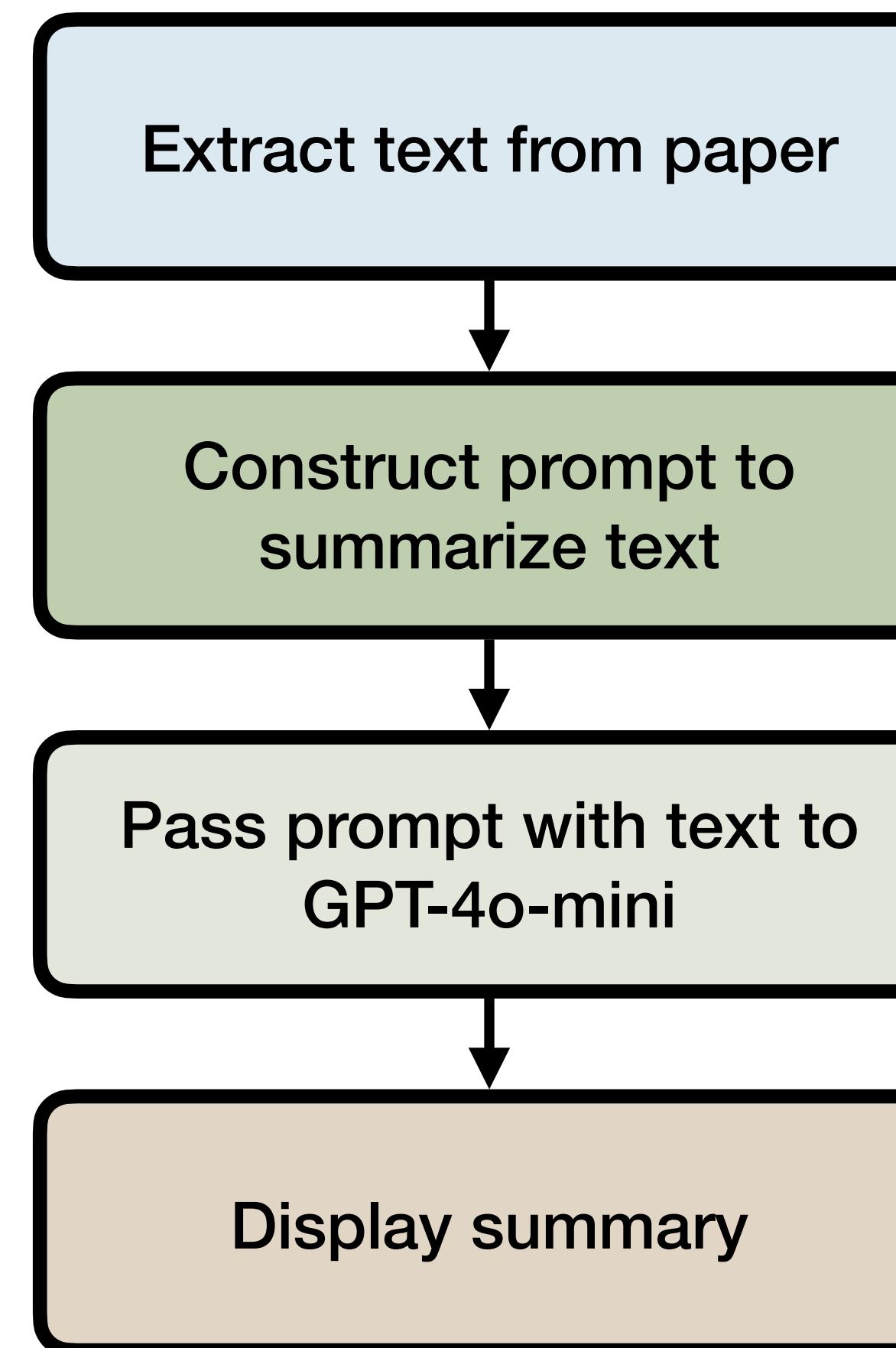
Example 1

Summarizing Research Papers with GPT-4o (Overview)

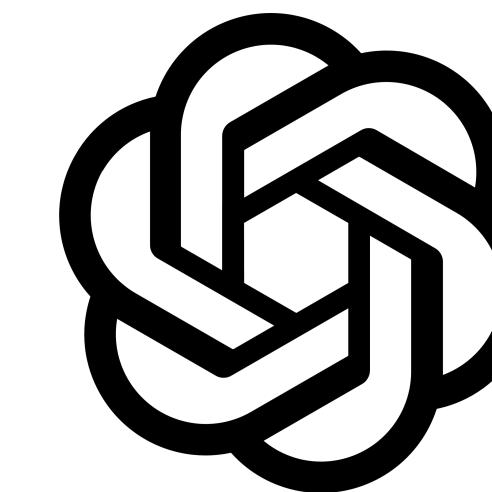


Example 1

Summarizing Research Papers with GPT-4o (Flowchart)



(From Example 1 - Session 2)



Example 1

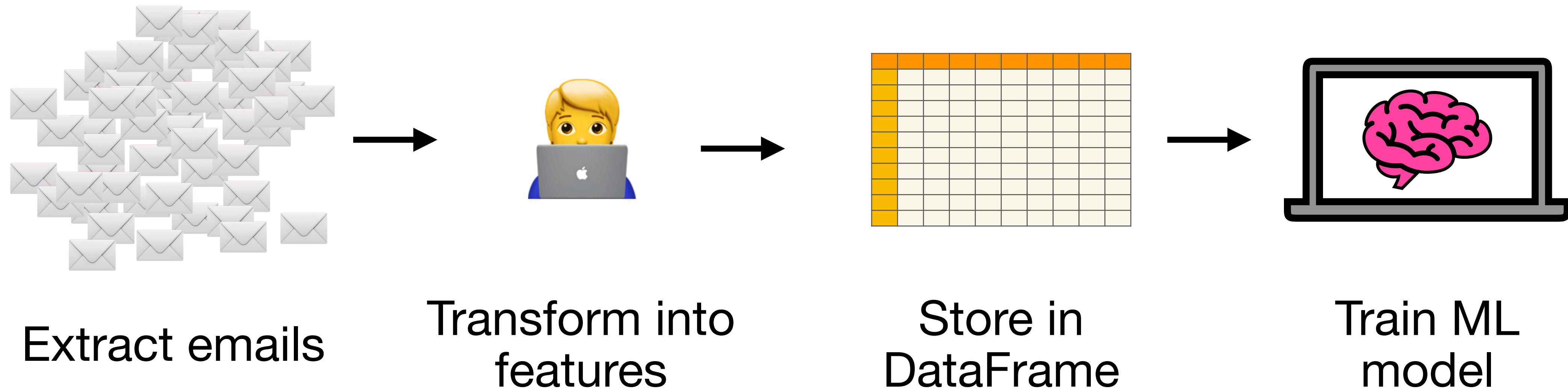
Summarizing Research Papers with GPT-4o (Code)



Example 2

Text Classification with GPT-4o-mini (Overview)

Session 2 - Examples



Example 2

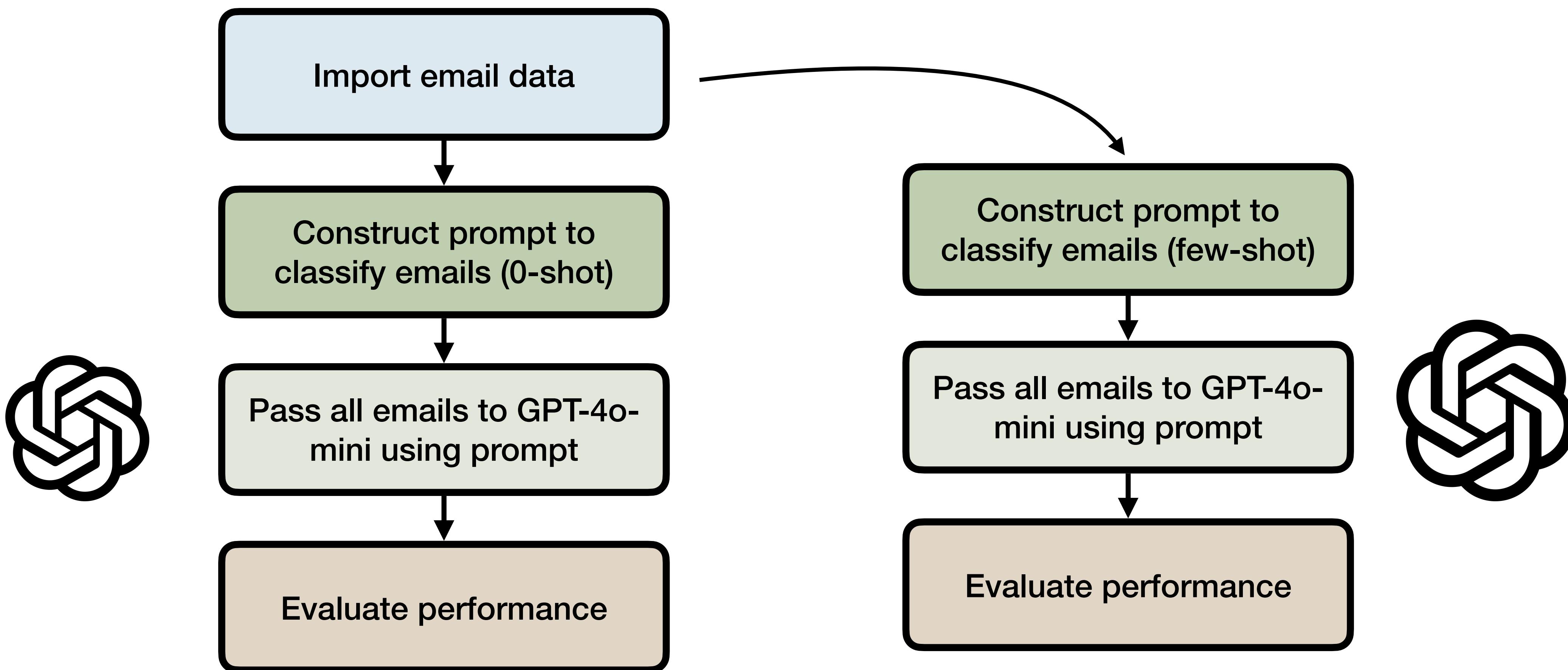
Text Classification with GPT-4o-mini (Overview)

Session 2 - Examples



Example 2

Text Classification with GPT-4o-mini (Flowchart)



Example 2

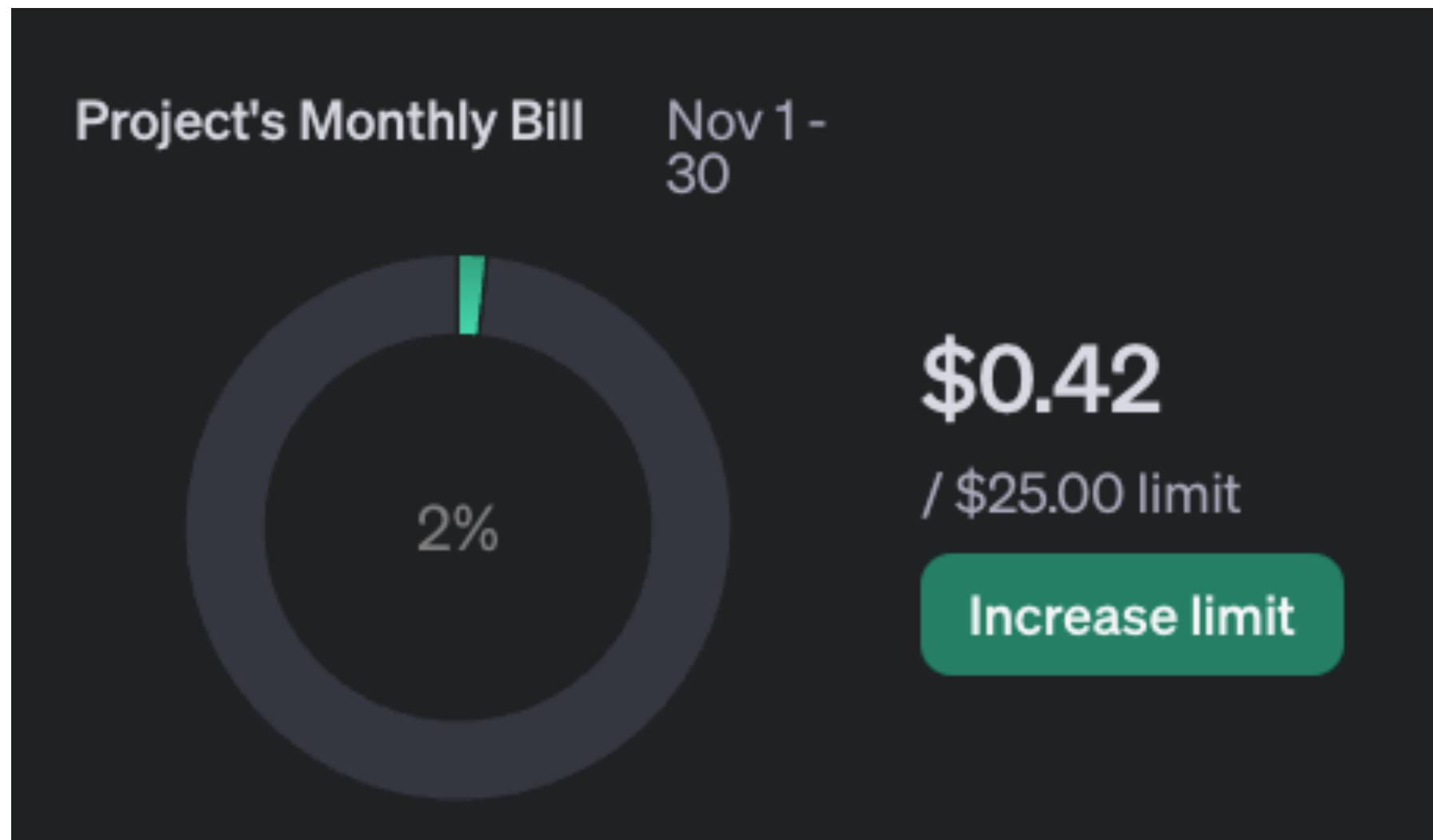
Text Classification with GPT-4o-mini (Code)



Example 3

Local Document QA Bot with Ollama (Overview)

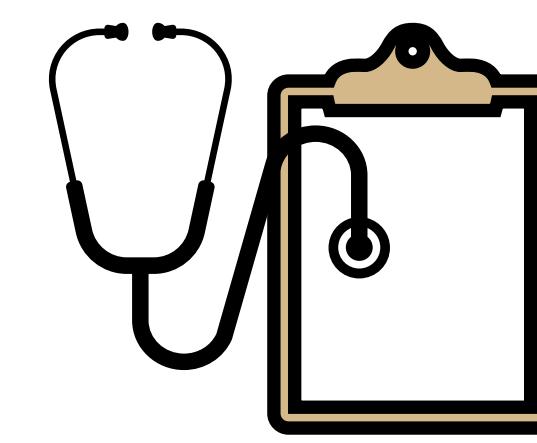
API Costs (for Example 2)



API Limits

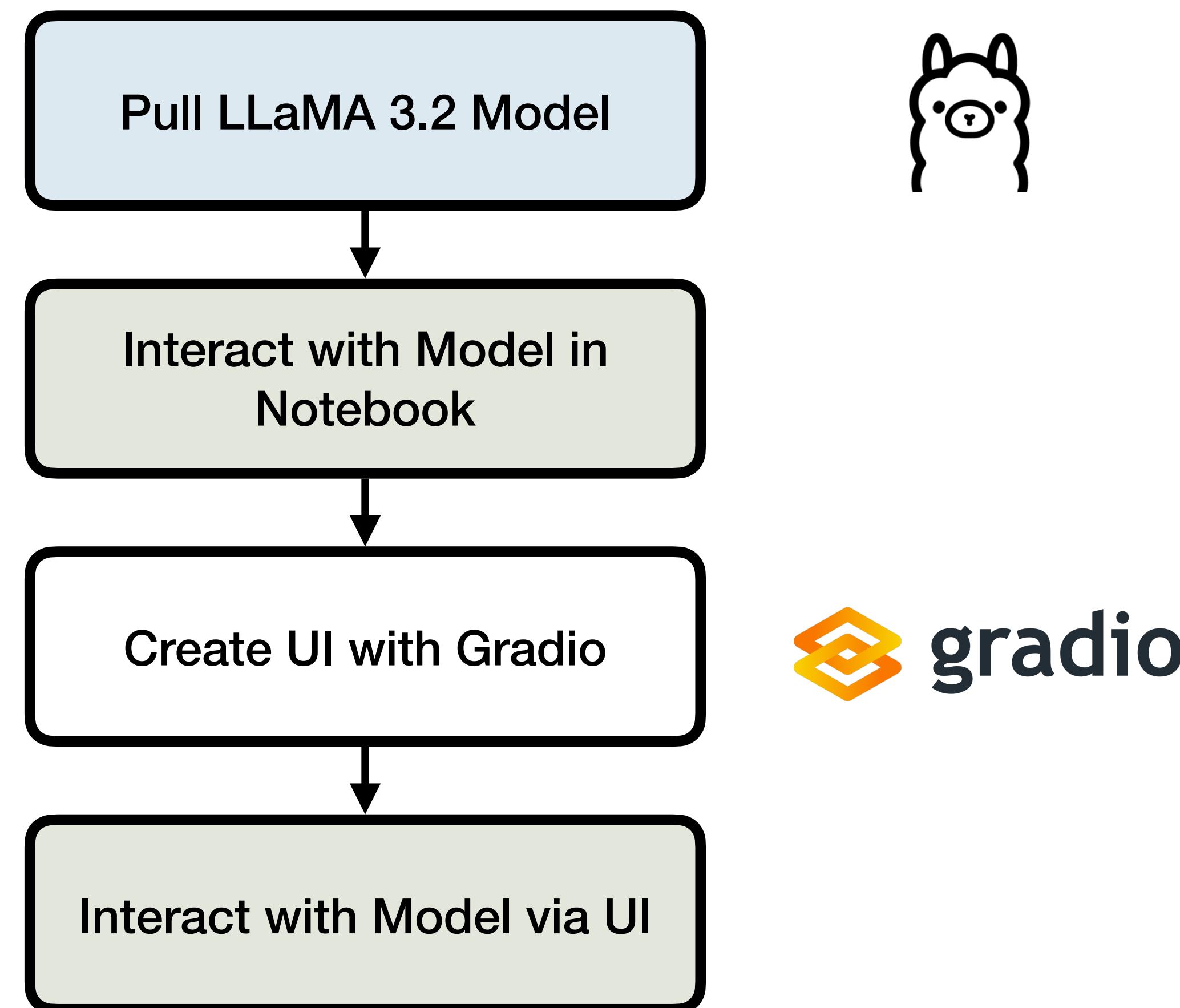
Error Code 429 - Rate limit reached for requests

PII and Sensitive Data



Example 3

Local Document Question Answering with Ollama (Flowchart)



Example 3

Local Document Question Answering with Ollama (Code)



Homework 3

Project 

Build an Automation with an LLM

Pre-work 

***No session
next week!*** 

Session 4: Embedding Models

Session 4: RAG

References

- [1] Survey of Large Language Models. [arXiv:2303.18223 \[cs.CL\]](https://arxiv.org/abs/2303.18223)
- [2] [A Practical Introduction to Large Language Models \(LLMs\)](#)
- [3] Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.
- [4] [LLM Tokenizer Demo](#)
- [5] [Prompt Engineering: How to Trick AI into Solving Your Problems](#)

