

ABB - Session 2

LLMs, Prompt Engineering

Shaw Talebi

Today's Session

1. Housekeeping

- 1.1. Announcements
- 1.2. Homework 1

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 (Visual) QA with Ollama



Announcements

Peer groups

Group	Name
1	Prabhakar Somu
1	Manoj Mohan
1	Vik L Vala
2	Jerry Bonner
2	Ahías Steller Mora
2	Thomas Helms
3	Kasey Markham
3	Joe Frausto
3	Cameron

Meet-and-greet the end of lecture today!

Suggestions

Share tips/resources in group chats

Share feedback when group members submit a HW project

Do weekly/bi-weekly check-ins

Let me know ASAP if you want to join a peer group

Homework 1

Shoutouts

PDF Text Search

Prabhakar Somu

GDP Scraper & Visualizer

Thomas Helms

New Sentiment Analysis

Ahías Steller Mora

Password Generator

Thierry Pienaar

Automated File Organizer

Joe Frausto

Downtrend Stock Analyzer

Krys Kuzara

Software 1.0

Rules are explicitly programmed into computer

You can do a lot with Software 1.0

But writing robust logic is hard...

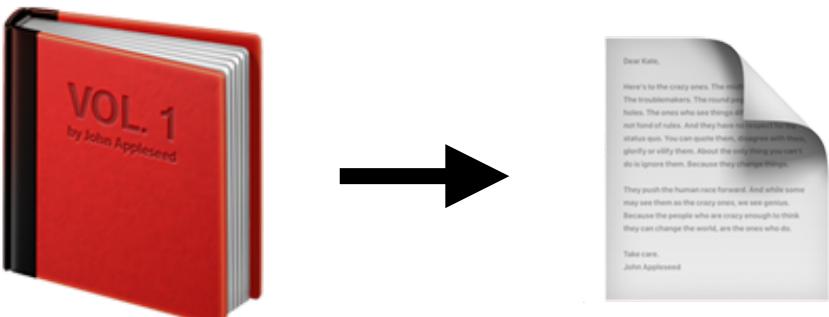
... if possible.

Software 1.0

Rules are explicitly programmed into computer

But writing robust logic is hard...

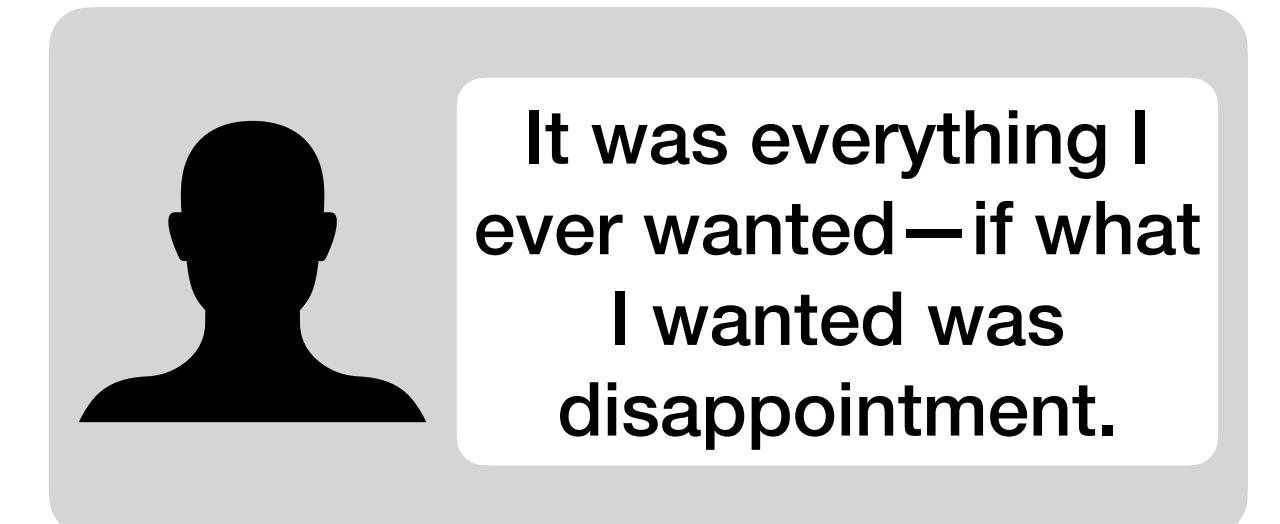
... if possible.



Summarization

What is this?

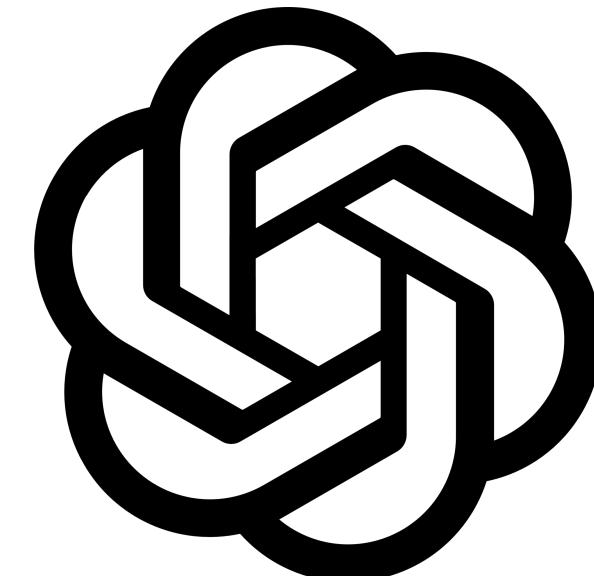
Sentiment Analysis



Software 3.0

Software 3.0

Adapting general-purpose models to specific use cases



GPT-4o

Summarize this book.

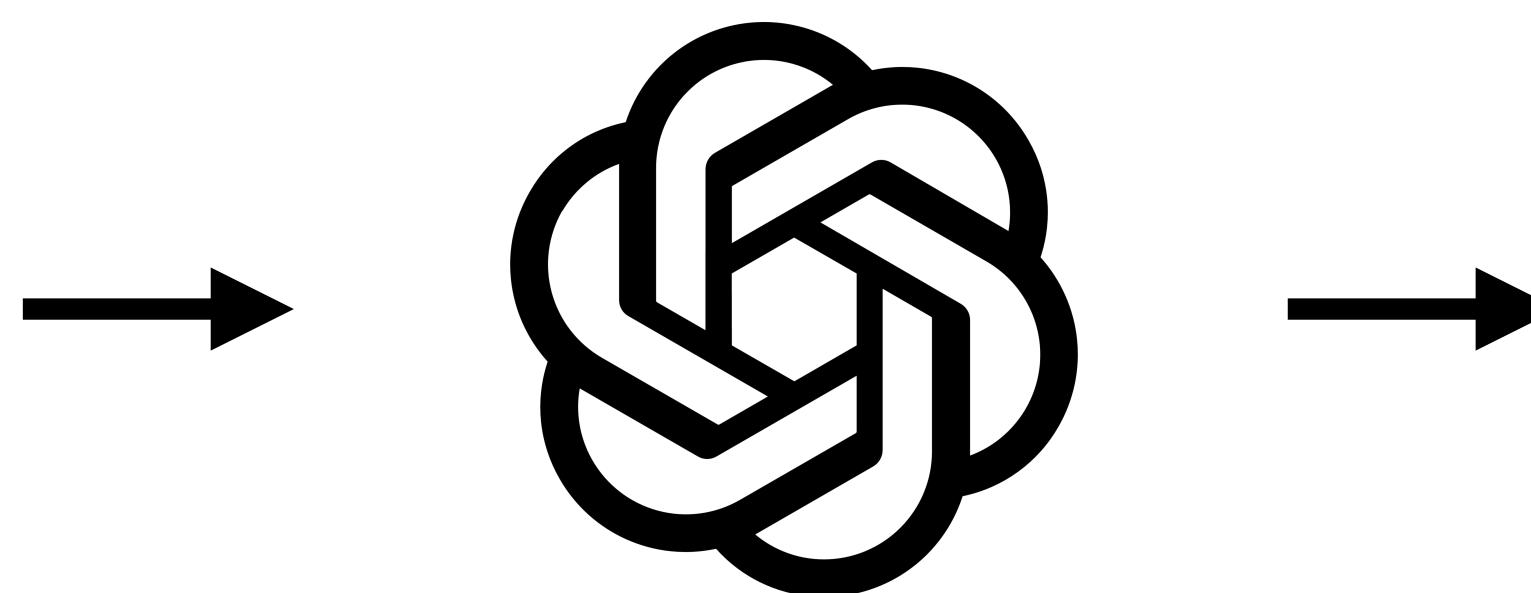
Antifragile: Things That Gain from Disorder by Nassim Nicholas Taleb is a book about systems, people, and ideas that don't just survive chaos and shocks — they *benefit* from them. Taleb introduces the concept of **antifragility** as something beyond resilience or robustness. While resilient things resist shocks and stay the same, **antifragile things grow stronger...**

Software 3.0

Adapting general-purpose models to specific use cases



What is this?

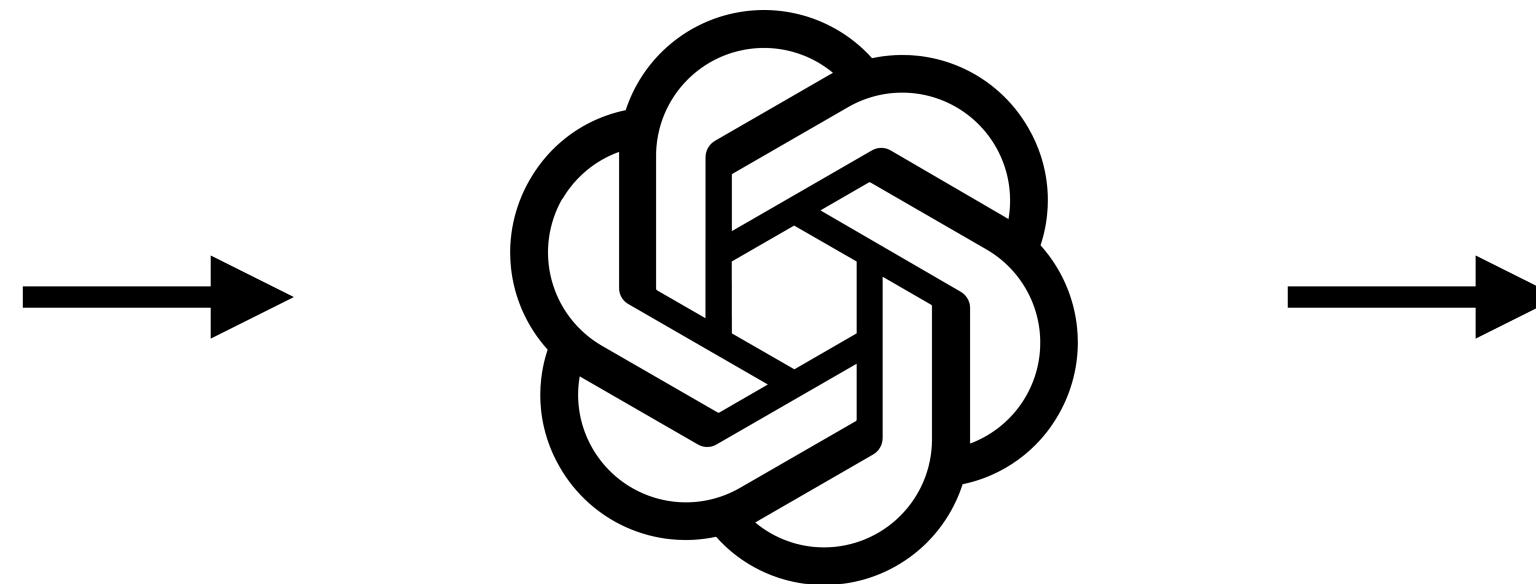
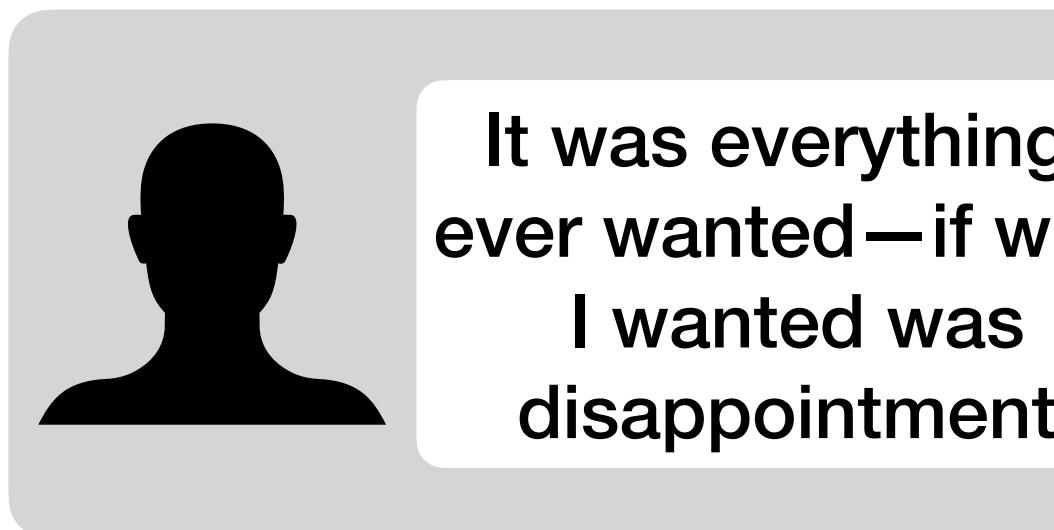


GPT-4o

This is an image of
an adorable puppy
lying on grass.

Software 3.0

Adapting general-purpose models to specific use cases



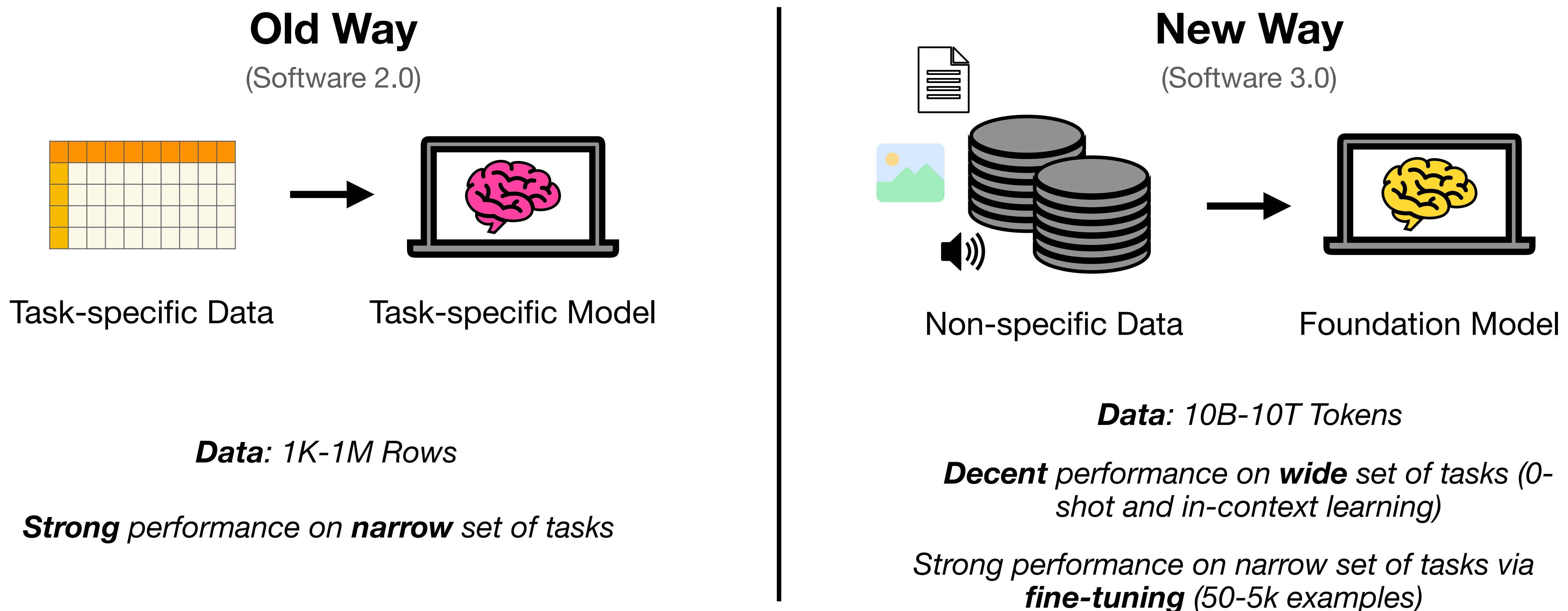
What is the sentiment of this review?

GPT-4o

The sentiment of this review is **negative**.

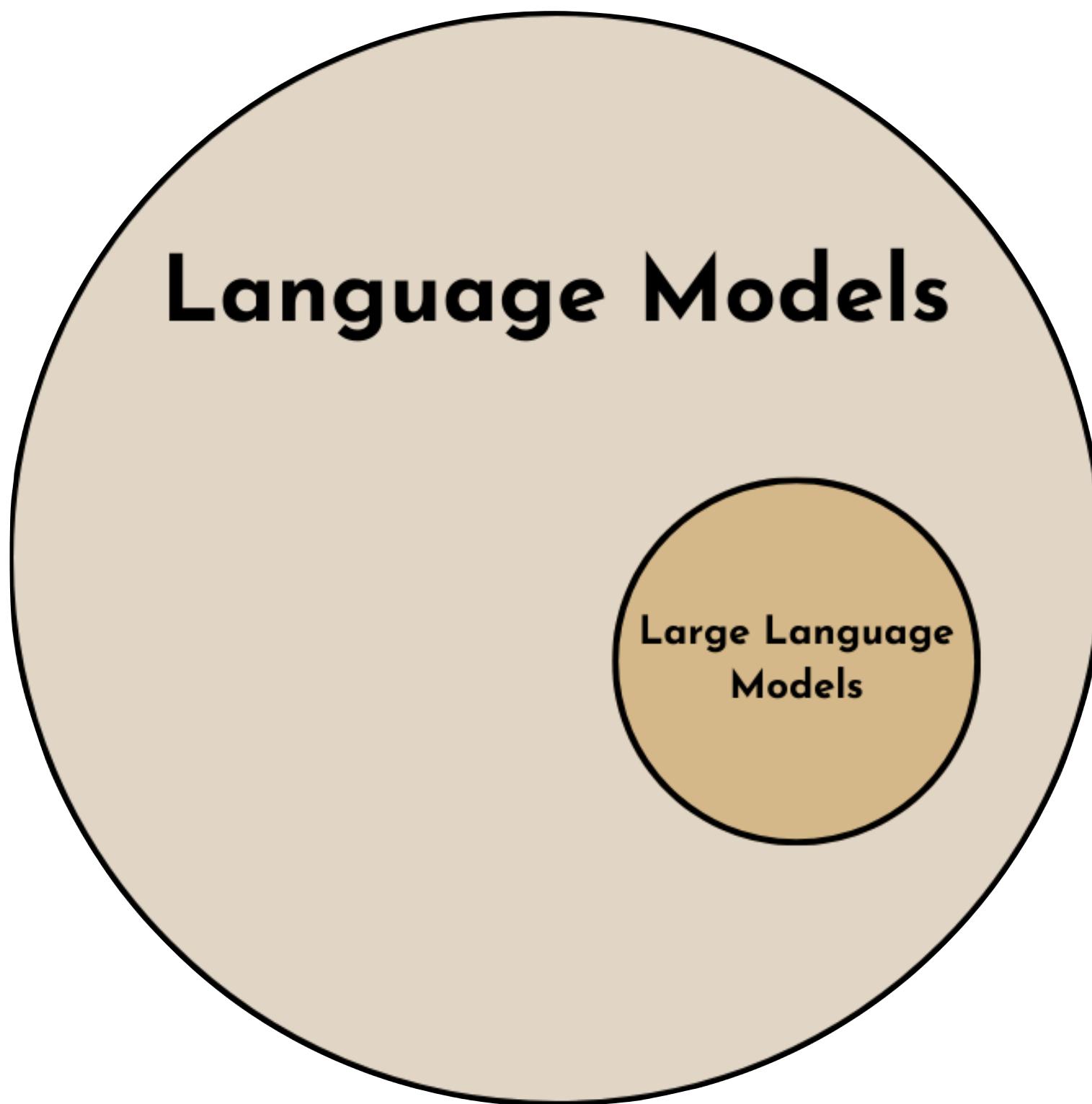
What's so different about this?

Upgrading from 2.0 to 3.0



Large Language Models (LLMs)

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



Quantitatively

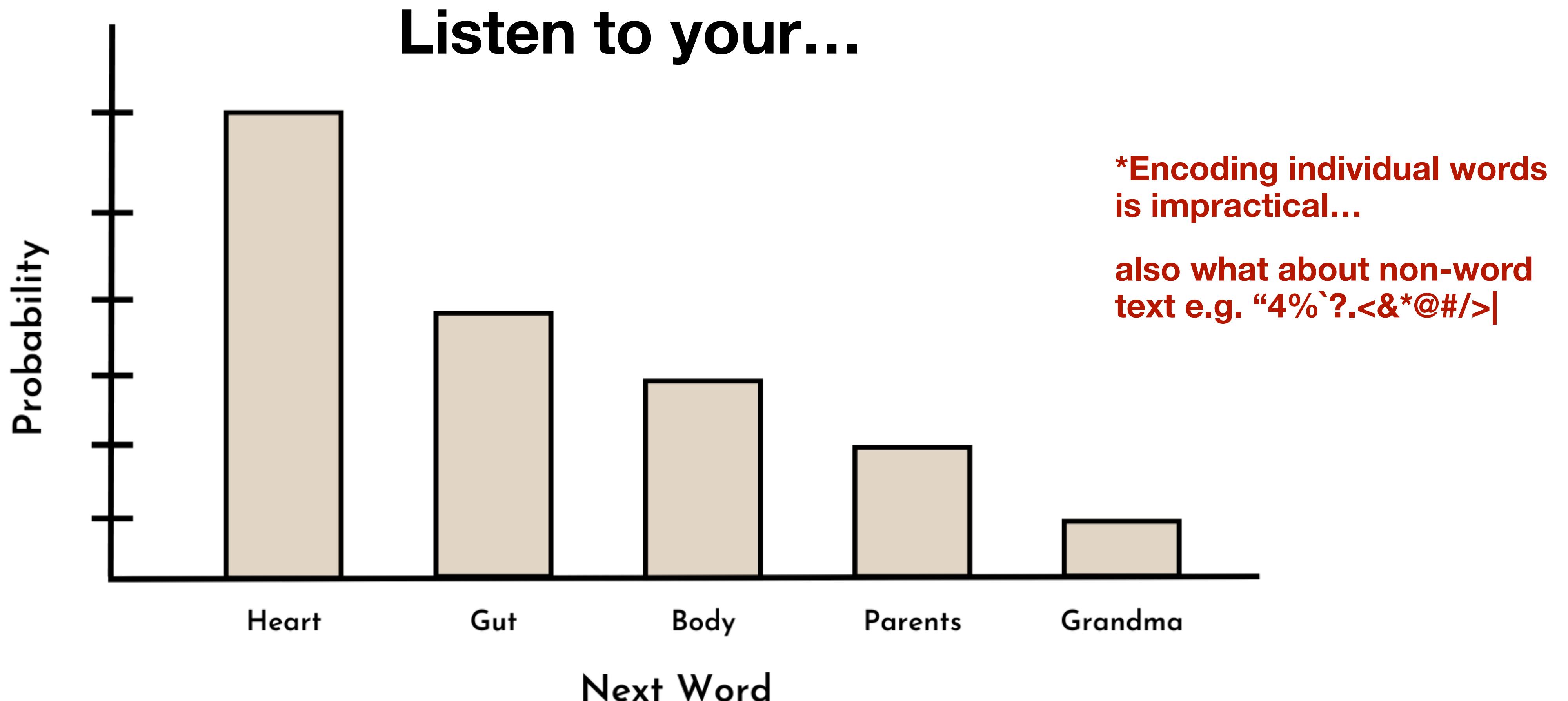
Number of model parameters
i.e. 1-100+ Billion

Qualitatively

Emergent properties
e.g. Zero-shot learning

How do LLMs work?

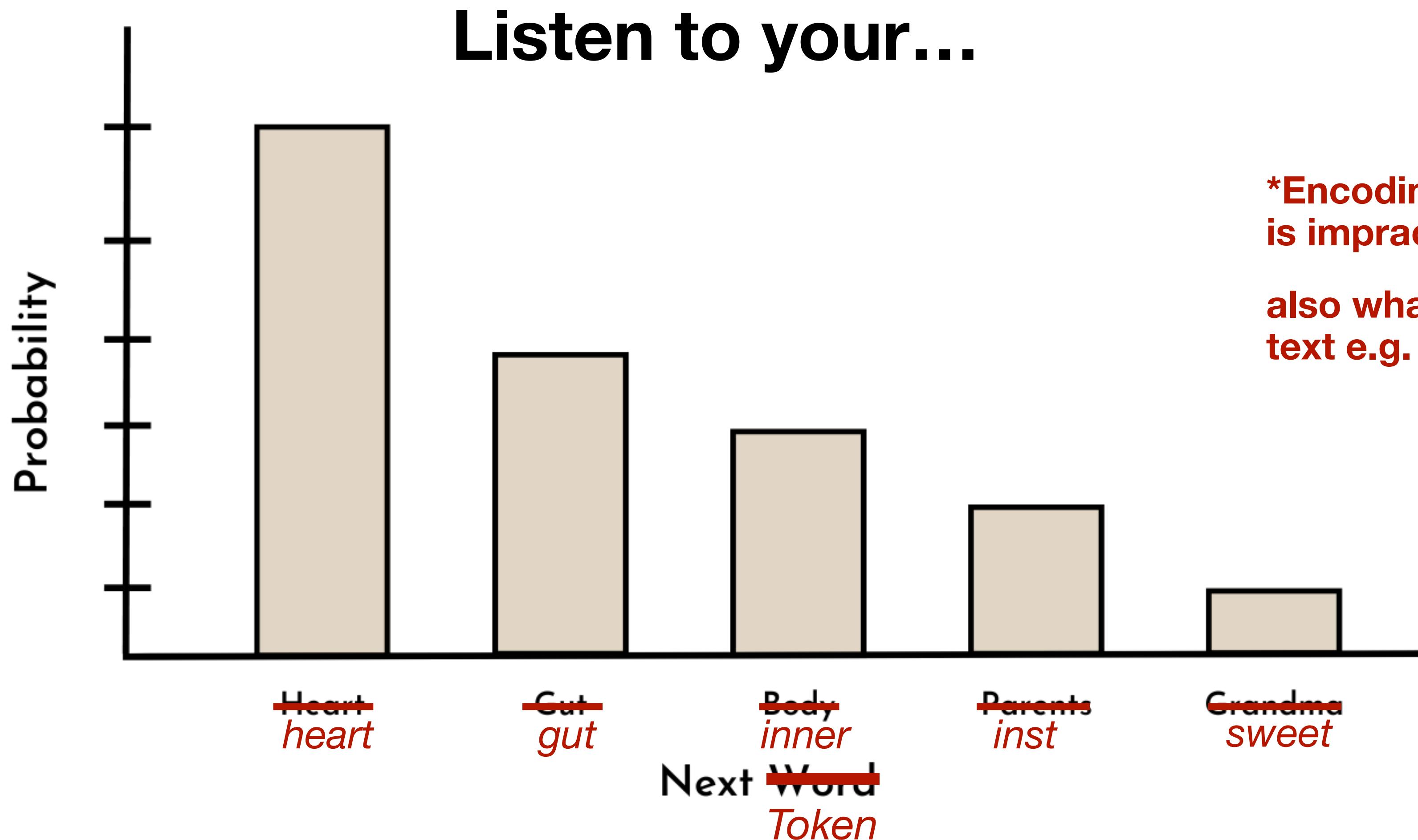
Basic idea: Next-word prediction



How do LLMs work?

Basic idea: Next-~~word~~ prediction

Token

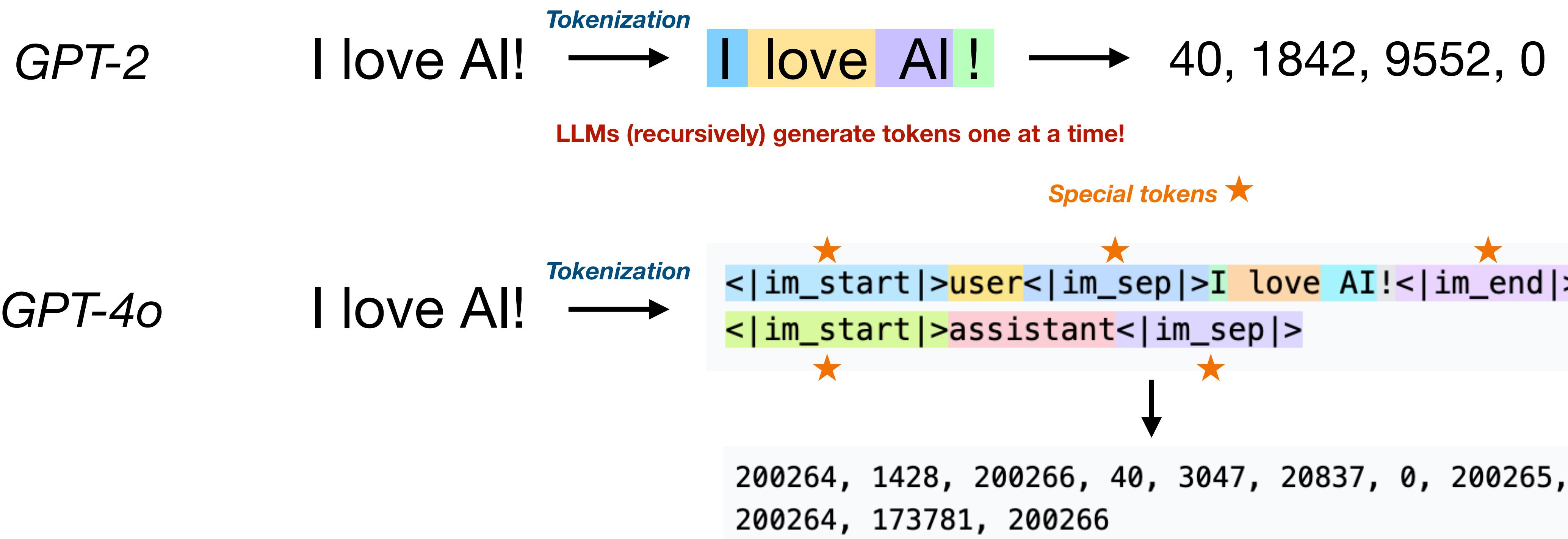


*Encoding individual words
is impractical...

also what about non-word
text e.g. “4%`?.<&*#@#/|”

Tokens

Character strings serving as basis of LLM inputs/outputs



Using Special Tokens

Help transform LLMs into helpful assistants

System

```
<|im_start|>system<|im_sep|>You are a  
helpful assistant<|im_end|>
```

User

```
<|im_start|>user<|im_sep|>What is an LLM?  
<|im_end|>
```

Assistant

```
<|im_start|>assistant<|im_sep|>An LLM is  
a Large Language Model – a type of  
artificial intelligence that's trained to  
understand and generate human language.<|  
im_end|>
```

What is an LLM?

An LLM is a Large Language Model –
a type of artificial intelligence that's
trained to understand and generate
human language.

Raw Text

(i.e. what the model sees)

Text is one continuous sequence!

UI

(i.e. what we see)

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



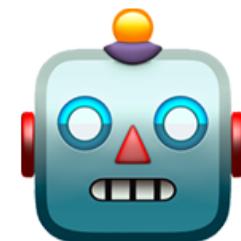
Post-training



Prompt Engineering

Adapting model behavior through (natural language) inputs

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



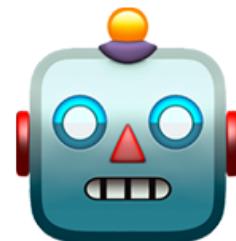
Prompt Engineering is the process of designing and optimizing input prompts to guide AI models, like ChatGPT, in producing desired and accurate outputs. It involves crafting specific instructions, context, and examples to achieve effective and reliable results.

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: Clear Instructions

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

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

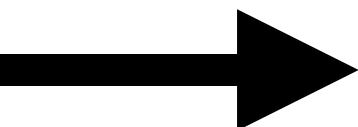
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin* ‡
illia.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.

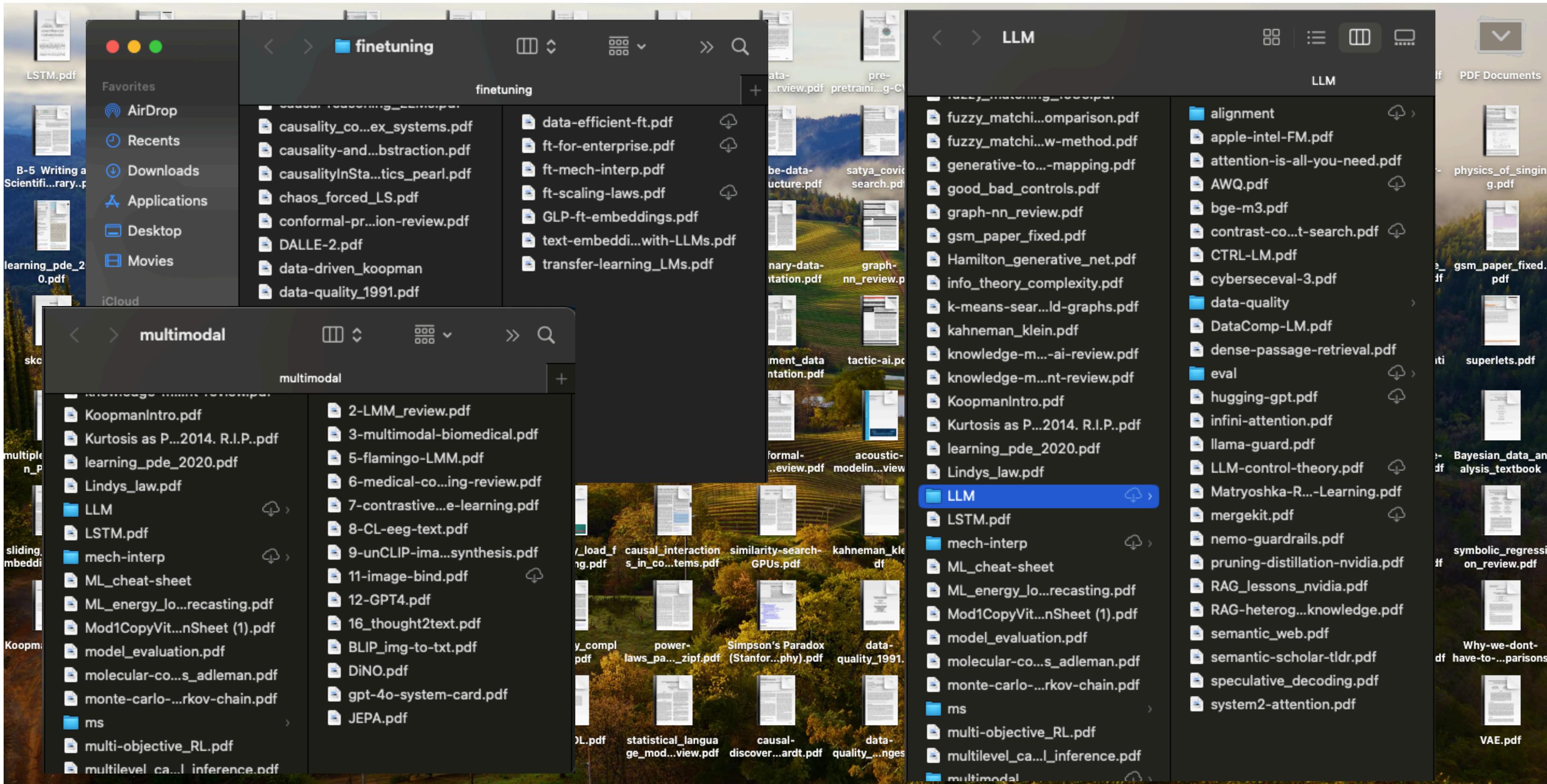


Lit Review:

Title
Authors/Affiliations
Key Concepts
Contributions

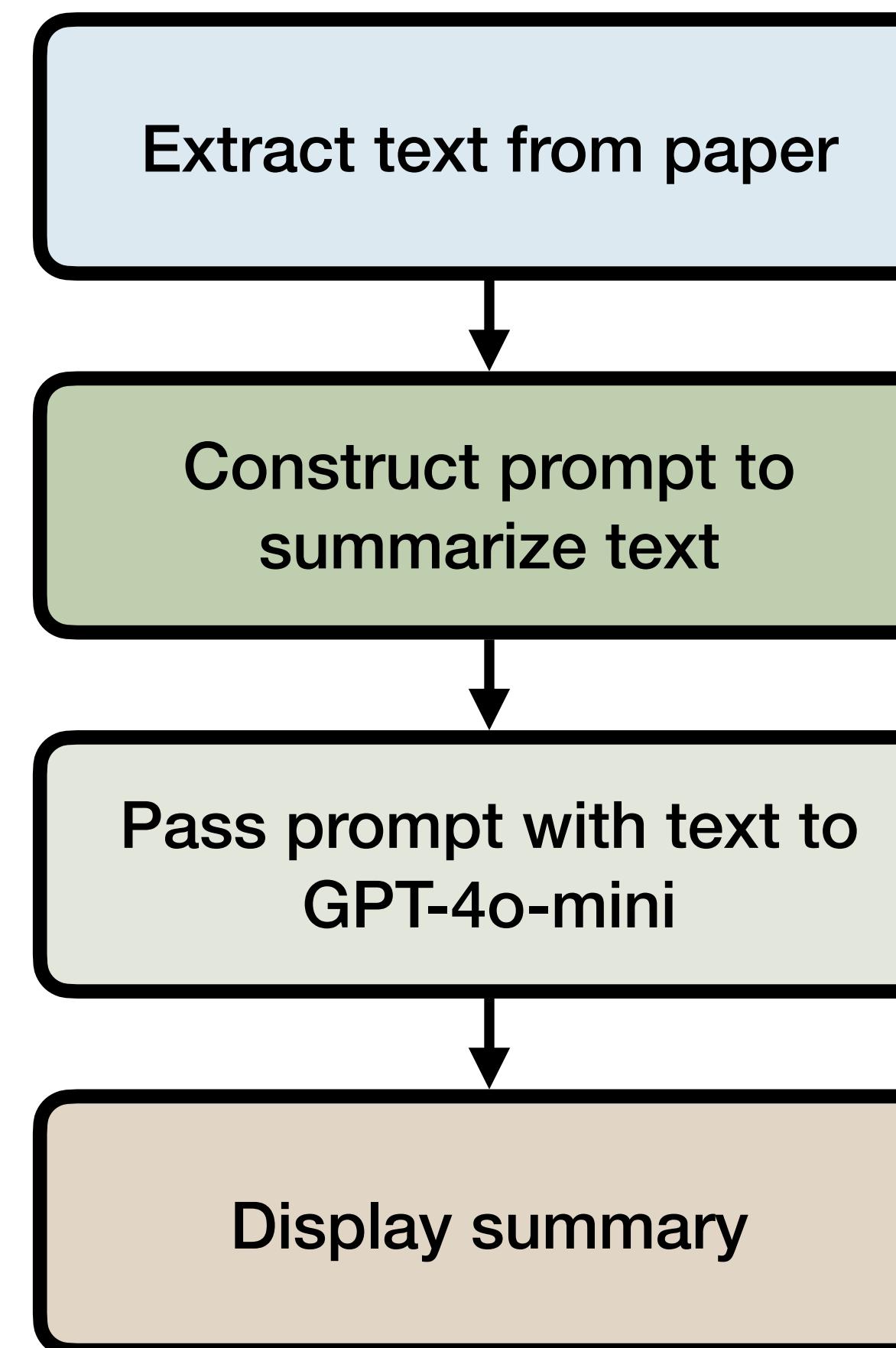
Example 1

Summarizing Research Papers with GPT-4o (Overview)

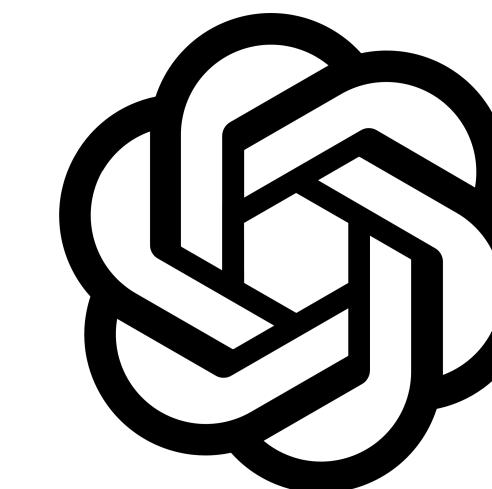


Example 1

Summarizing Research Papers with GPT-4o (Flowchart)



(From Example 1 - Session 2)



Example 1

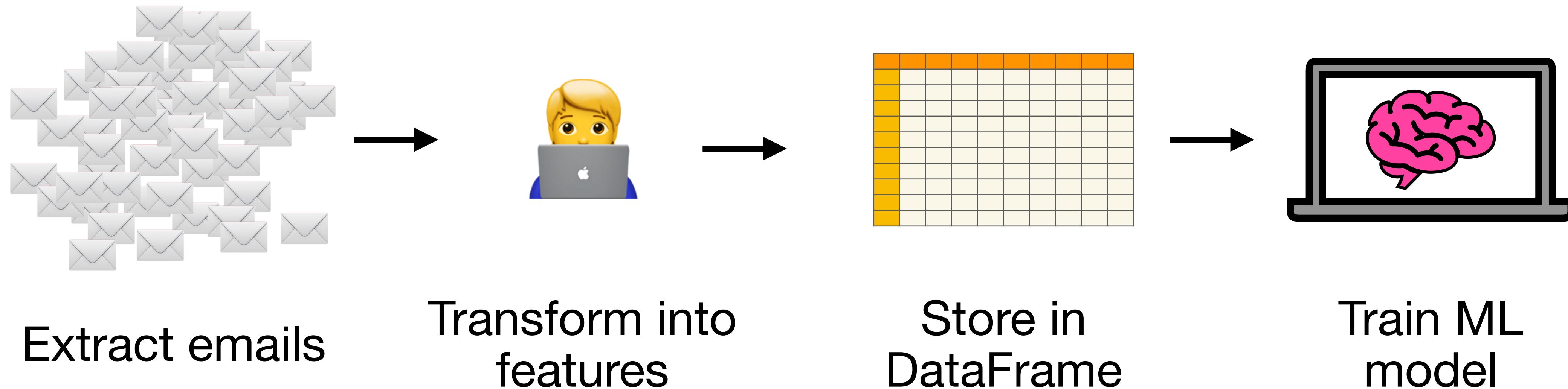
Summarizing Research Papers with GPT-4o (Code)



Example 2

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

Cohort 1, Session 2 - Examples



Example 2

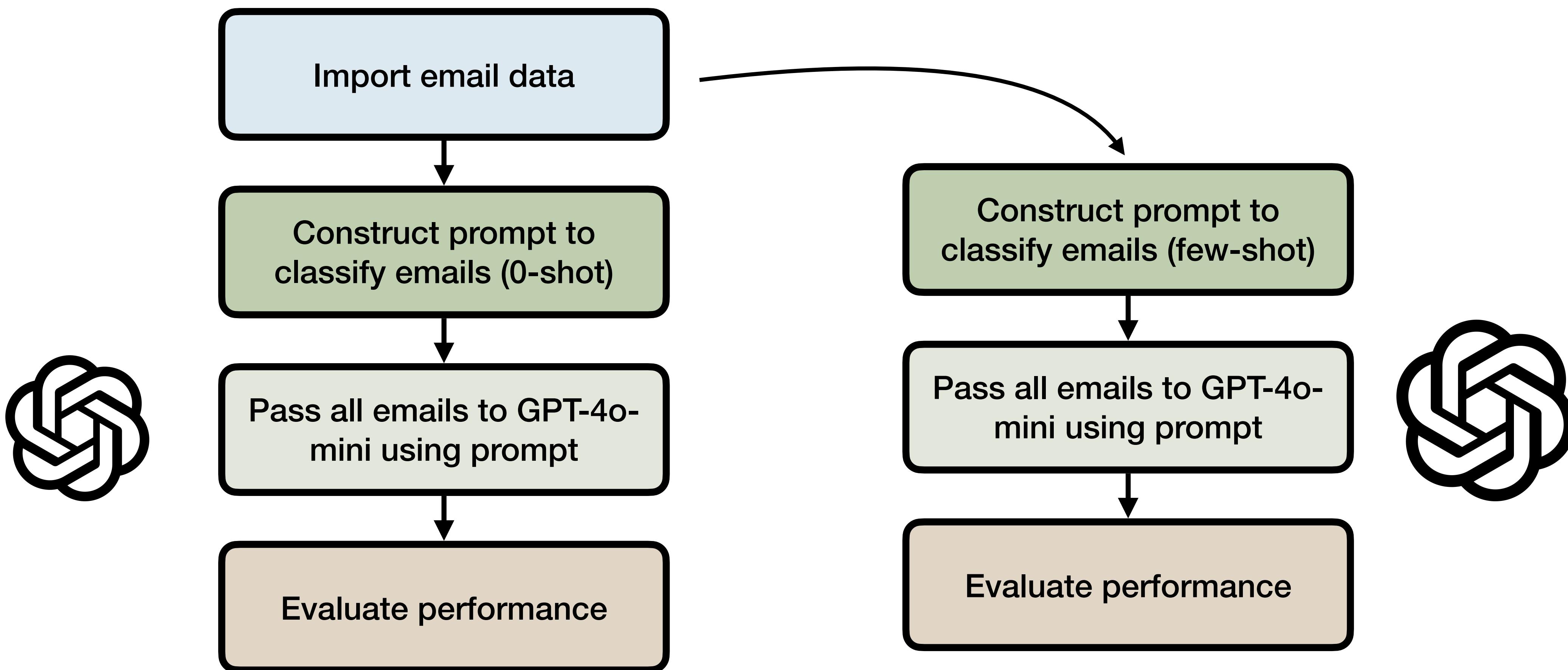
Text Classification with GPT-4o-mini (Overview)

Cohort 1, Session 2 - Examples



Example 2

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



Example 2

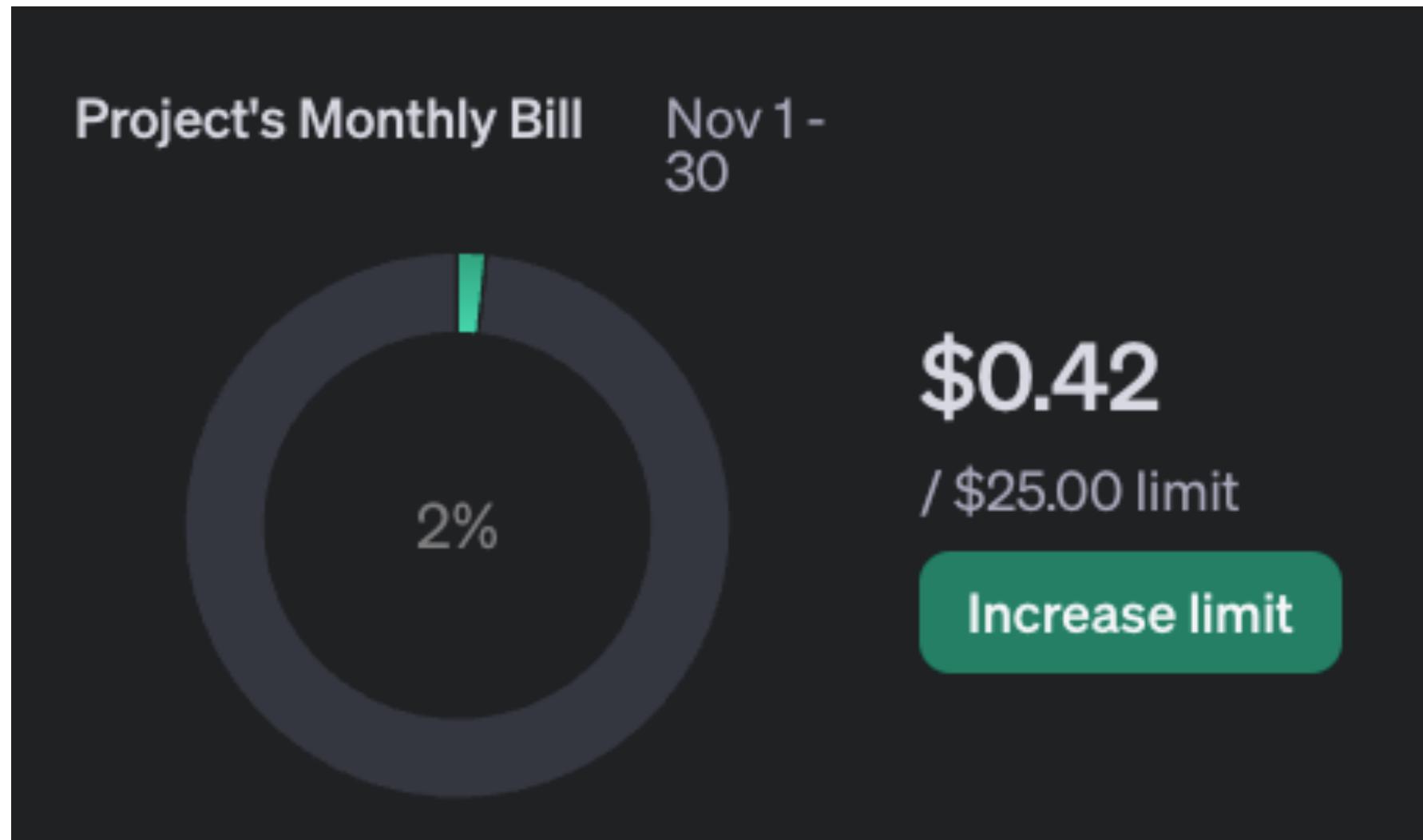
Text Classification with GPT-4o-mini (Code)



Example 3

Local (Visual) QA Bot with Ollama (Overview)

API Costs



API Limits

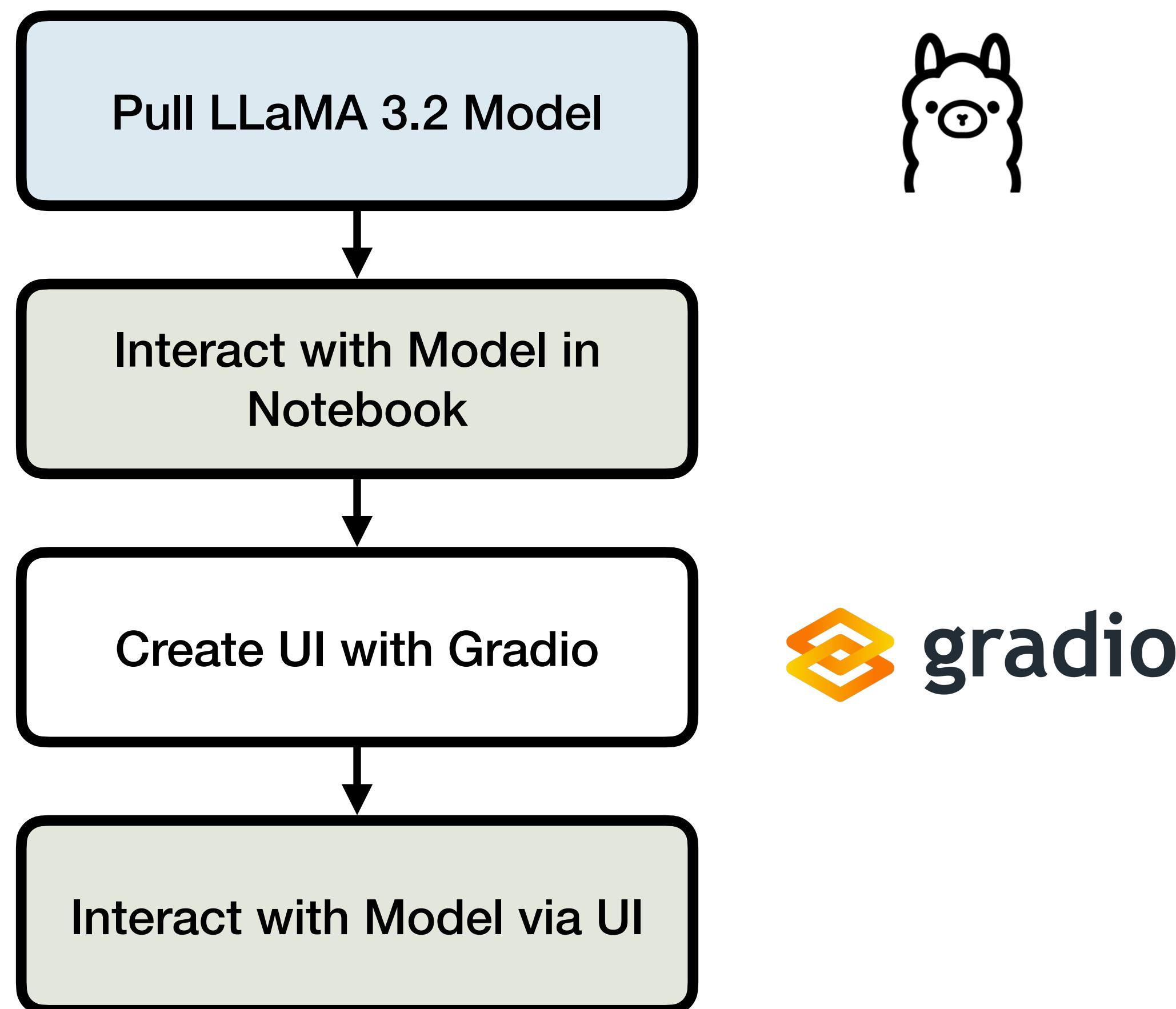
Error Code 429 - Rate limit reached for requests

Restricted Data



Example 3

Local (Visual) Question Answering with Ollama (Flowchart)



Example 3

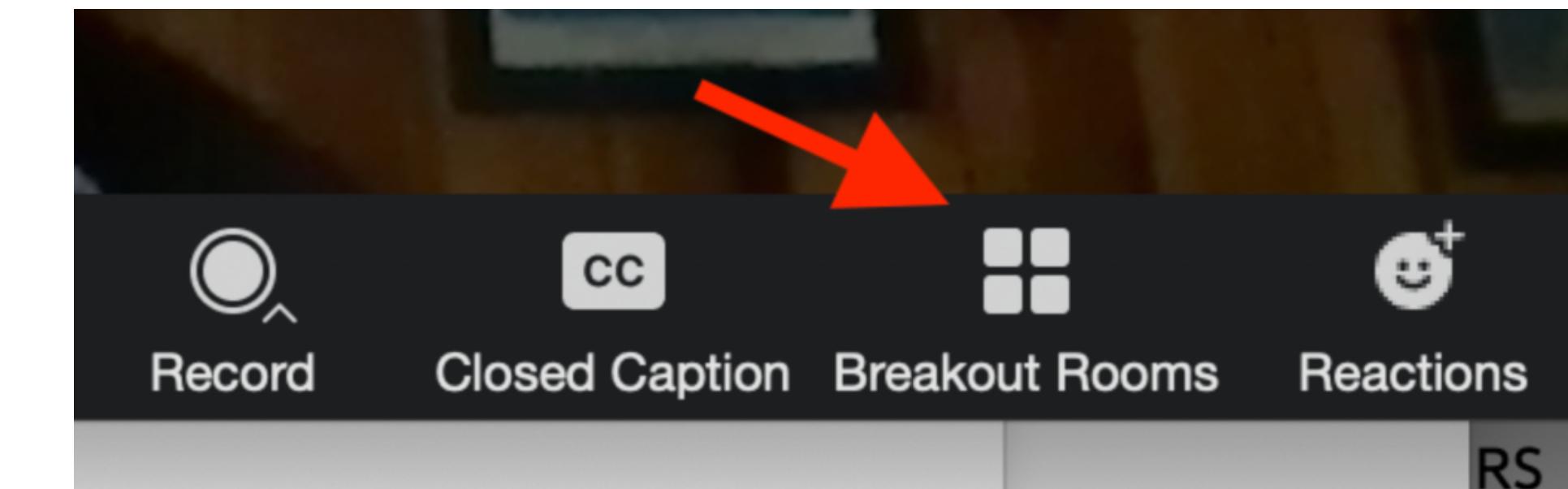
Local (Visual) Question Answering with Ollama (Code)



Peer Group Meet-and-Greet

Group	Name
1	Prabhakar Somu
1	Manoj Mohan
1	Vik L Vala
2	Jerry Bonner
2	Ahías Steller Mora
2	Thomas Helms
3	Kasey Markham
3	Joe Frausto
3	Cameron

ZOOM



15-Minute Breakout Rooms

Homework 3

Project

Build an Automation with an LLM

Pre-work

Session 3: Embedding Models

Session 3: 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](#)
- [6] [OpenAI's Prompt Engineering Guide](#)
- [7] [Anthropic's Prompt Engineering Guide](#)

