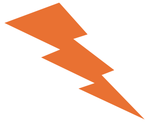
Large Language Models: the basics

Kevin Duh
Johns Hopkins University
June 2024

Today's Agenda

- 9:00-10:20: Tutorial on LLM basics
- 10:20-10:40: Break
- 10:40-12:00: Research showcase: invited talks that illustrate different research areas related to LLMs
- 12:00-13:00: Lunch
- ~~13:00-13:30: Computer lab setup~~
- 13:30-17:00: Lab

Goals of this tutorial

- Establish common terminology
- Point out standard thinking that might require re-thinking
    - items marked  = Caution: don't fully believe at face value

- Outline:
    1. Why LLMs are fundamentally different from what came before
    2. How LLMs are built
    3. Survey of popular LLM implementations
    4. Quick sampling of some advanced topics

1. Why LLMs are fundamentally different from what came before

What defines a Large Language Model (LLM)?

- Size?
- Architecture?
- Training objectives?
- Anything can be called LLM if it’s good for the press release?
- Intended Use (my preferred definition):
  - LLM are models that have emergent abilities and are intended to be used for multiple purposes

LM, PLM, & LLM

- Distinction based on intended use

- Language Model (LM)
  - use case: probability of next word
- Pre-trained Language Model (PLM) – BERT
  - use case: one NLP task after fine-tuning
- Large Language Model (LLM) – GPT-3.5
  - use case: multi-purpose & emergent ability

LM: Probability of Next Word

- LMs can be used in many applications, e.g. Speech Recognition

$$\begin{aligned} p(\vec{w}) &= p(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_1) \times p(w_{n-1} \mid w_{n-2}, \ldots, w_1) \\ &\times p(w_{n-2} \mid w_{n-3}, \ldots, w_1) \times p(w_{n-3} \mid w_{n-4}, \ldots, w_1) \\ &\times p(w_{n-4} \mid w_{n-5}, \ldots, w_1) \times \ldots \times p(w_2 \mid w_1) \times p(w_1) \end{aligned}$$

Sentence probability

Next word probability

- n-gram LM: Next word probability from counts: $p(w_2 \mid w_1) = \frac{Count(\text{“}w_1 w_2\text{”})}{Count(\text{“}w_1\text{”})}$
- neural LM: Next word probability from neural net: $p(w_i \mid w_{i-2}, w_{i-1})$

LM objective: Perplexity

- Information: Let E be an event which occurs with probability P(E). If I told you E occurred, then I've given you $I(E) = -\log_2 P(E)$ bits of info
- Entropy: suppose distribution p(x) with K possible values. What is the average amount of info?

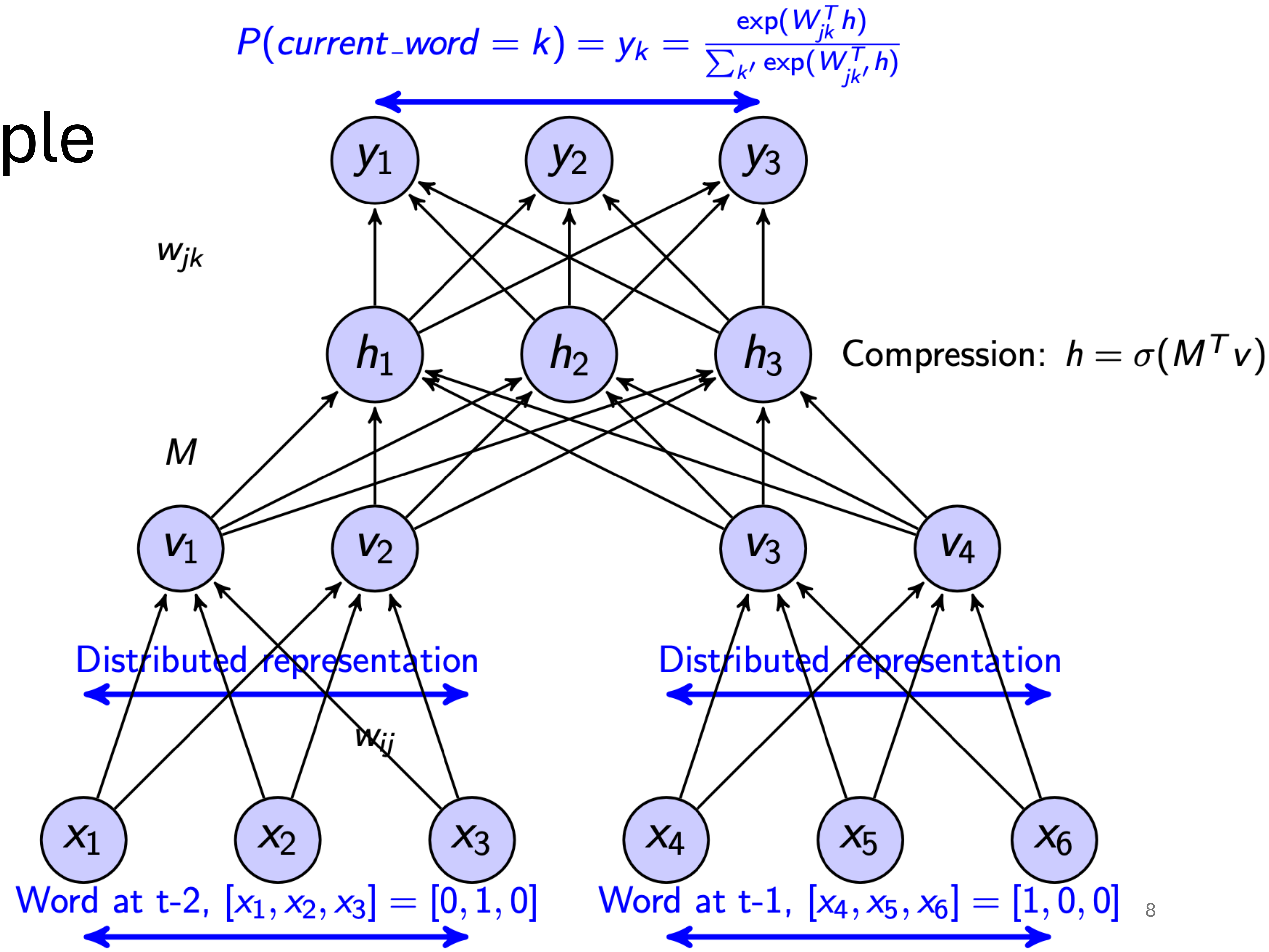
$$H(p) = \sum_{k=1}^{K} P(X = x_k) I(x_k) = \sum_{k=1}^{K} p(x_k) I(x_k) = -\sum_{k=1}^{K} p(x_k) \log_2 p(x_k)$$

- Cross-Entropy: suppose we don't know true distribution $p^*(x)$ but have a model p(x) that approximates it. How good is the model?

$$H(p^*, p) = \sum_{m=1}^{M} P^*(X = x_m) I(x_m) \approx \frac{1}{K} \sum_{k=1}^{K} I(x_m) = -\frac{1}{K} \sum_{k=1}^{K} \log_2 P(X_k = x_m)$$

- Perplexity: given a test set of K words, $PPL = 2^{-\frac{1}{K} \sum_{k=1}^{K} \log_2 P(X_k = x_m)}$

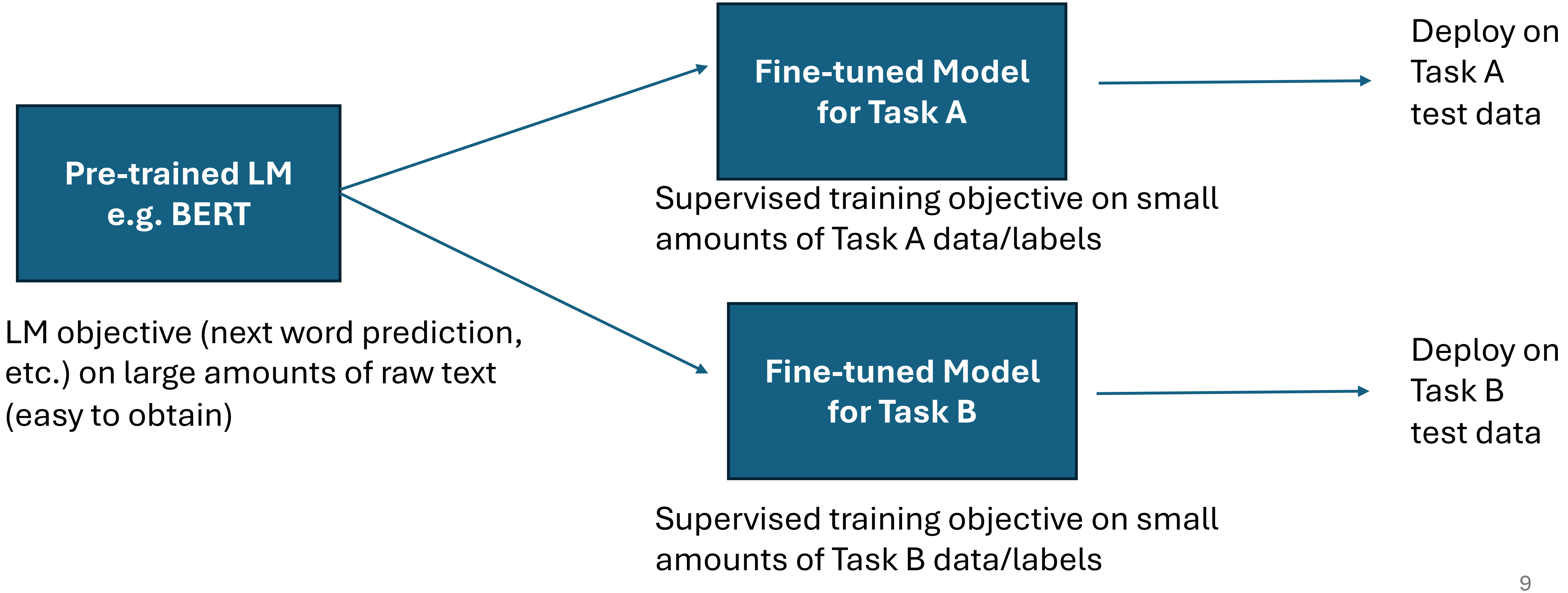
LM example



Bengio, et al. A Neural Probabilistic Language Model, JMLR 2003

PLM: Fine-tuning for one task

- Intuition: pre-training finds good “representations” of data, so only small amounts of task-specific labels are needed



BERT vs GPT-4

- Both trained with Language Model objectives but something seems fundamentally different

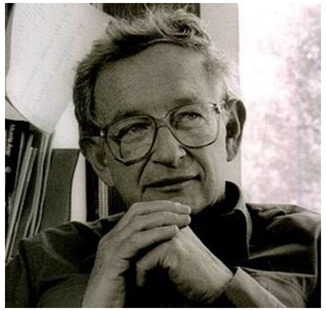
Pre-train on large data
+ Fine-tune on Task A
= Great performance on Task A

BERT

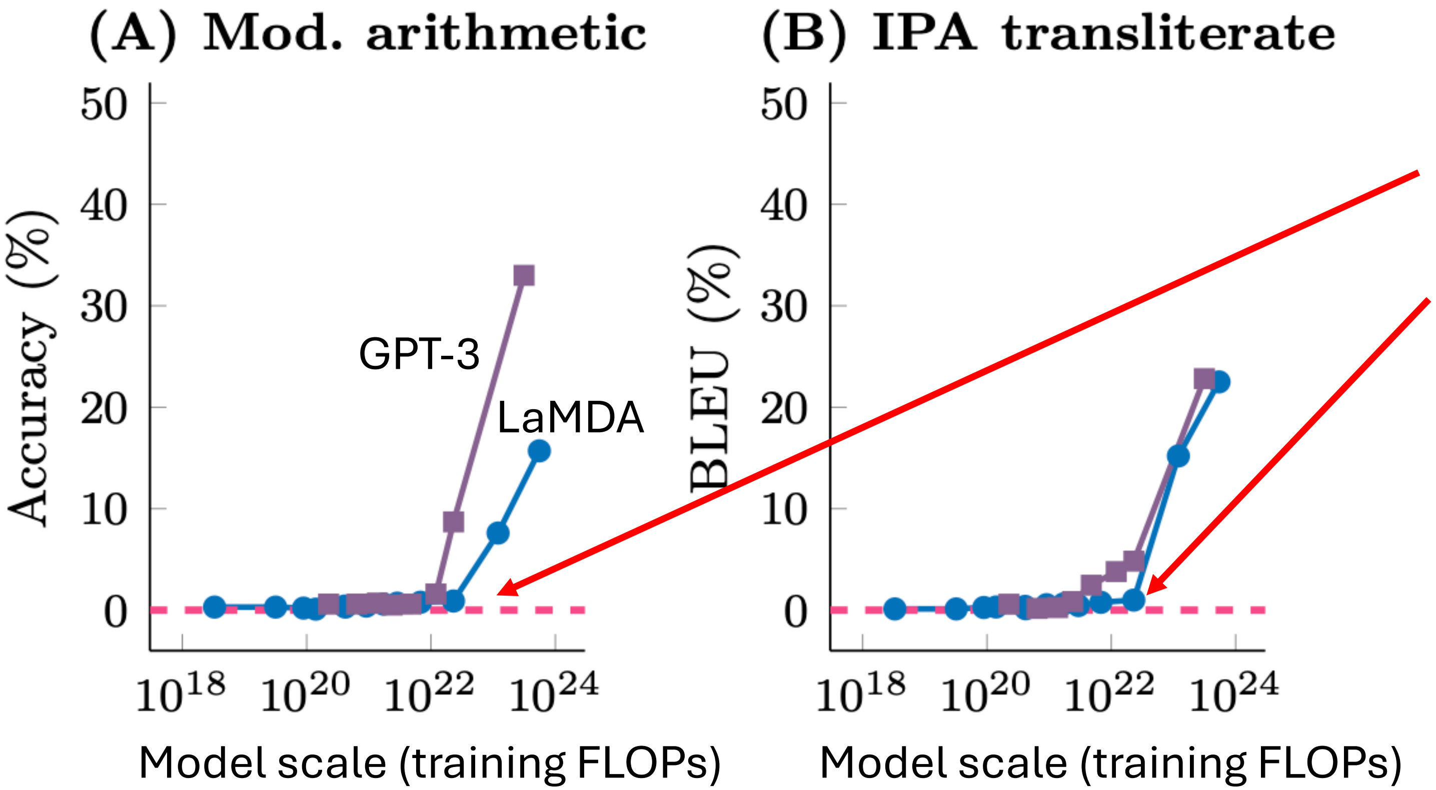
Pre-train on large data
+ Scale Up
= Emergent ability on many tasks (AGI?)

GPT-4

LLM: “Emergent” Abilities



“Emergence is when quantitative changes in a system result in qualitative changes in behavior.” – Philip Anderson (physicist), 1972

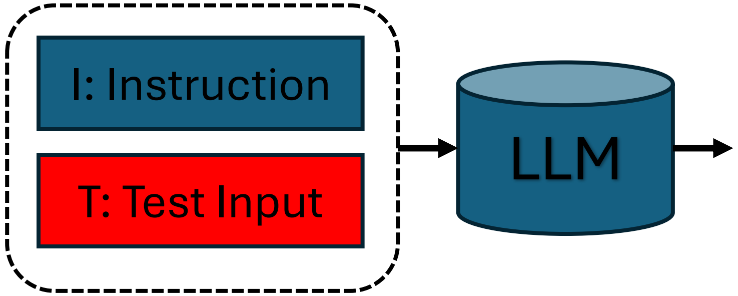


An ability is emergent if it is not present in smaller models but is present in larger models [Wei, et al (2022). Emergent Abilities of Large Language Models]

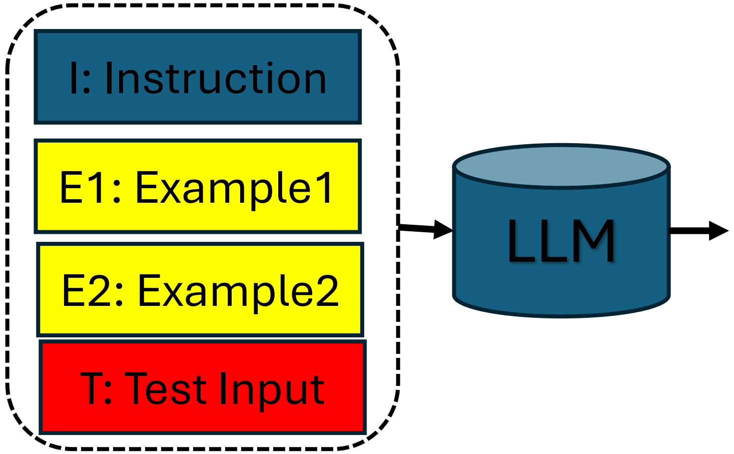
In-Context Learning (an example of emergent ability)

I: Instruction	Translate English to French
E1: Example1	[en]: A discomfort which lasts. [fr]: Un malaise qui dure
E2: Example2	[en]: HTML is a language for formatting [fr]: HTML est un langage de formatage
T: Test Input	[en]: After you become comfortable with formatting [fr]:

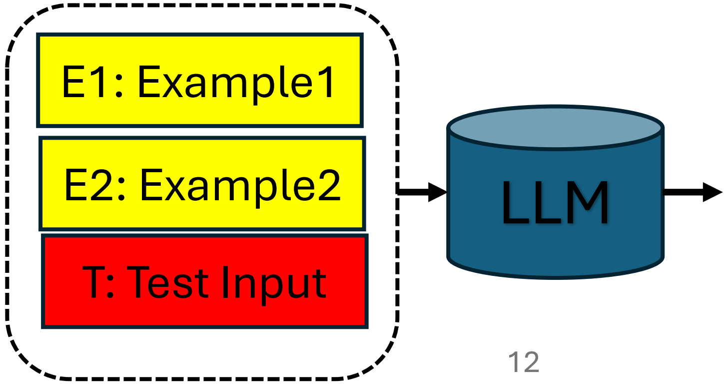
Zero Shot



Few Shot (w/ Instruction)



Few Shot (Example only)



Chain-of-Thought Prompting (also emergent)

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

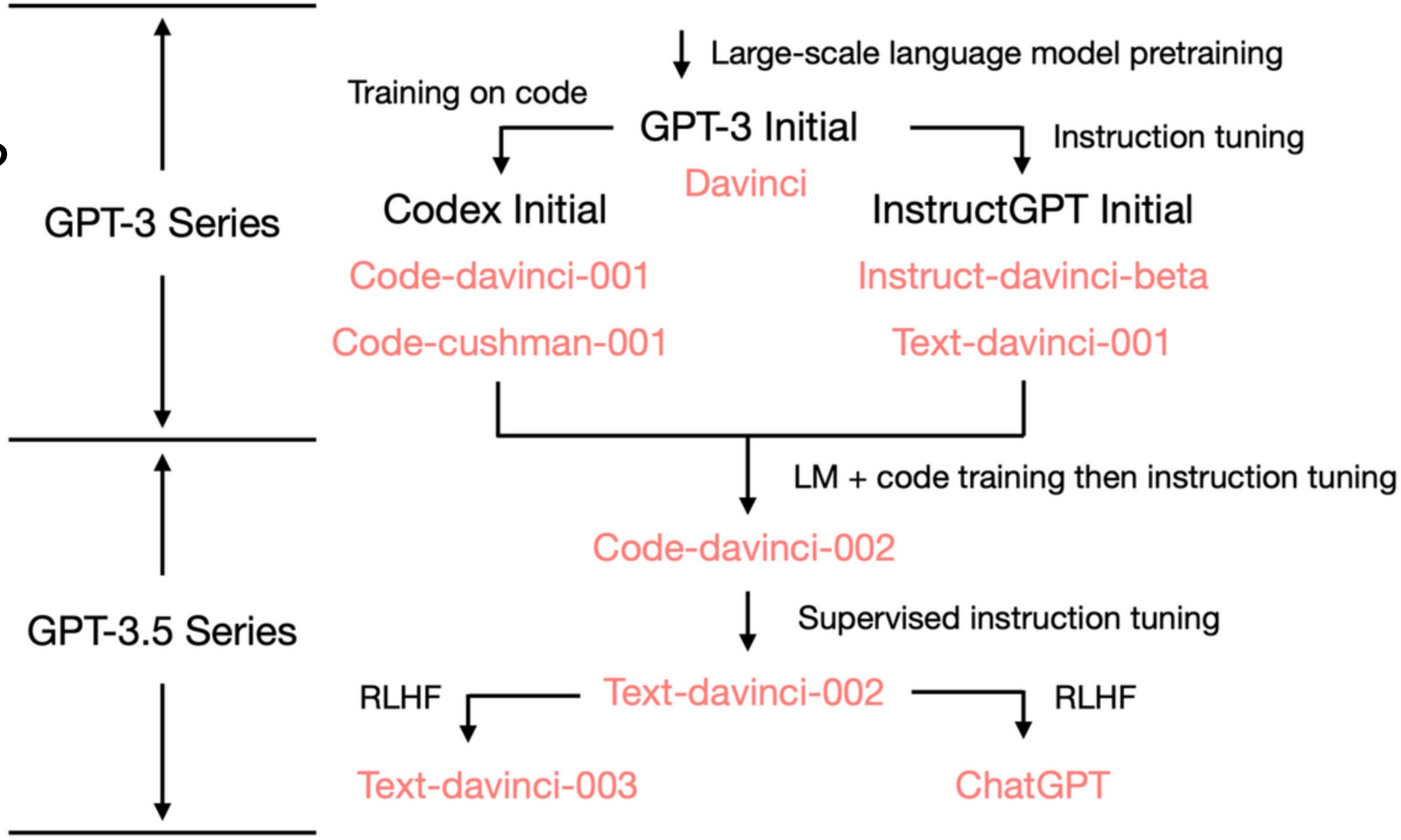
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

"There's this idea of emergence that caught me and also, I think, many researchers, by surprise – that you can just train a language model, predict the next token on a tons of raw text, and then it can answer questions, it can summarize documents, have dialogue, translate, classify text, learn all sorts of different kind of pattern manipulation, format dates, and so on. It was just really eye-opening …"

– Percy Liang (Stanford), 2022/06
https://web.stanford.edu/class/cs224u/podcast/liang/

Why do these abilities emerge? Still unknown

- Large scale?
- Overparameterization?
- Instruction tuning?
- Training on code?
- RLHF?
- Magic?



Recommended read: Blogpost by Yao Fu, Hao Peng, Tushar Khot. May 2023.
How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources

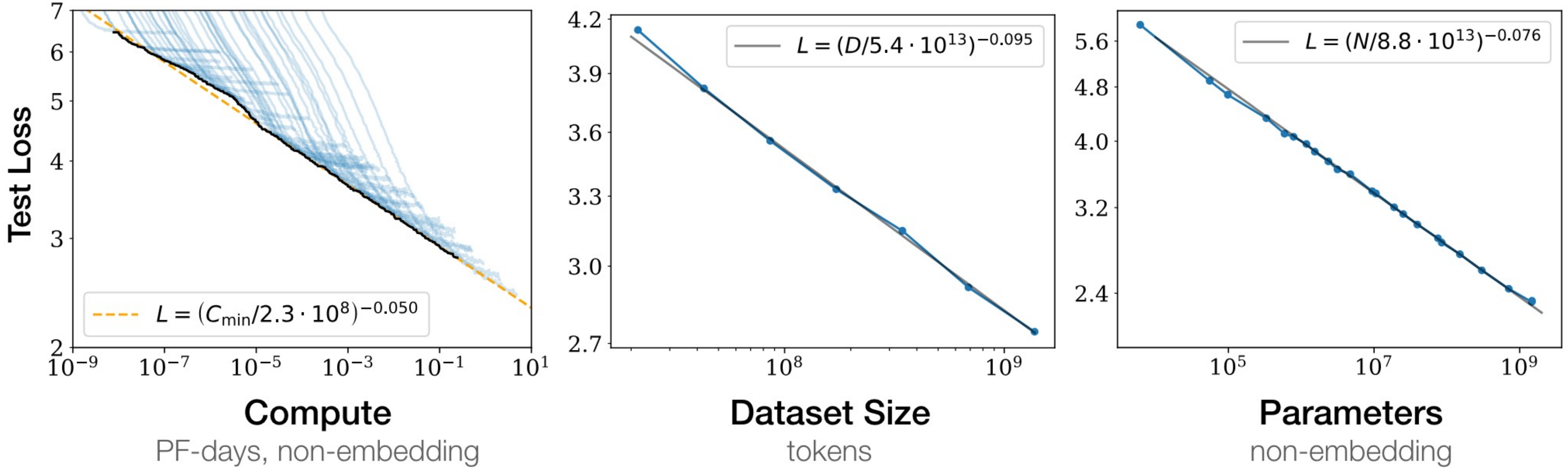
LLM's multi-purpose and emergent abilities contradict some machine learning intuitions

- No Free Lunch Theorem
  - If a method does well on certain class of problems, it must be paying for degraded performance on other problems.

- Objective function, Structural Risk Minimization
  - Generalization Error is bounded by Training Error + Capacity Term

Scaling Law

- Language modeling performance improves smoothly as we increase model size, dataset size, amount of compute for training



Kaplan, et al. (2020). Scaling Laws for Neural Language Models

Next word prediction is massively multitask?

name time

Johns Hopkins (May 19, 1795 – December 24, 1873) was an
syntax
American merchant, investor, and philanthropist. Born on a plantation, he left his home to start a career at the age of 17, and settled in Baltimore, Maryland, where he remained for most of his life.
geography

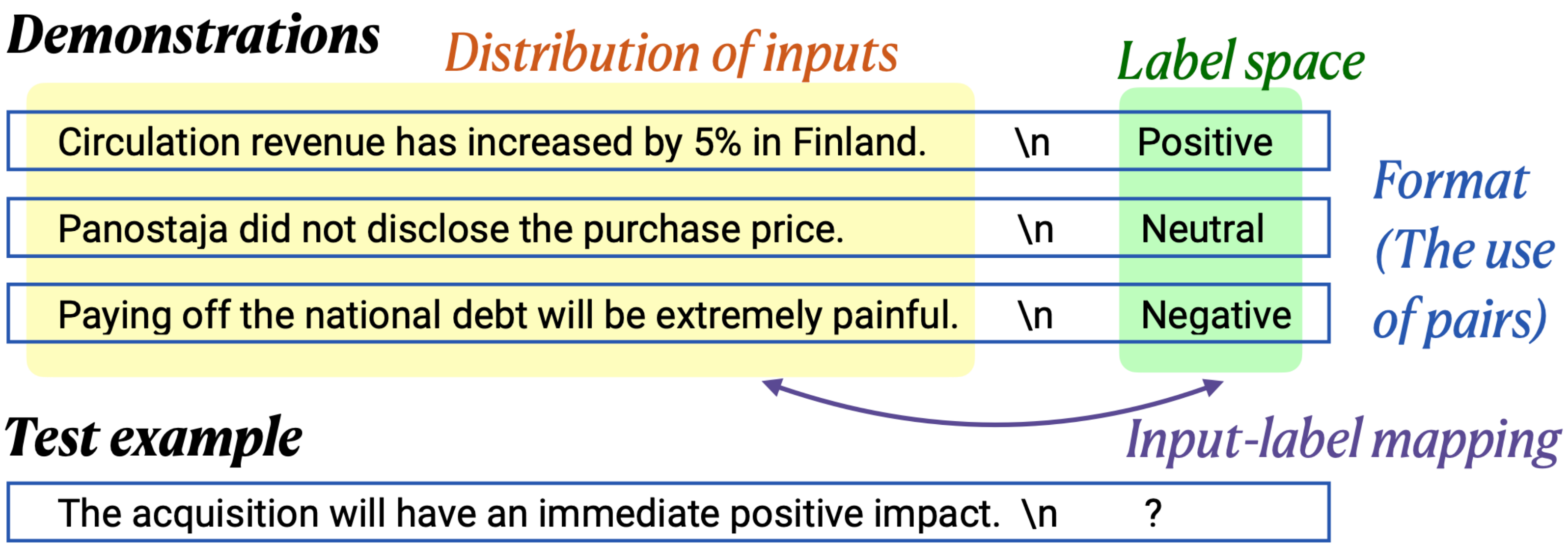
world knowledge

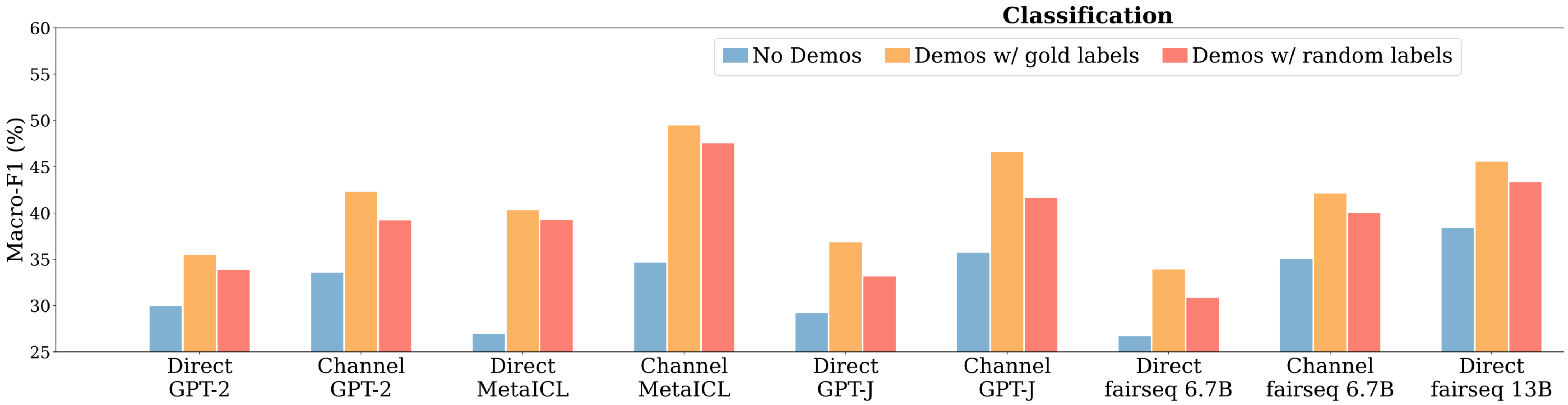
Hopkins invested heavily in the Baltimore and Ohio Railroad
syntax
(B&O), which eventually led to his appointment as finance
anaphora
director of the company. He was also president of Baltimore-based Merchants' National Bank.[a] Hopkins was a staunch supporter of Abraham Lincoln and the Union, often using his Maryland residence as a gathering place for Union strategists. He was a Quaker and supporter of the abolitionist cause.
world knowledge

Is this why LLM are multi-purpose? Small models must sacrifice long tail, whereas large models scaling up enable memorization of different knowledge

Hypotheses on the emergence of in-context learning

- Task identification?
  - Xie et al. (2021). An explanation of in-context learning as implicit Bayesian inference
  - Raventos, et al. (2023). Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression
- Some kind of ”learning” without model updates?
  - Akyurek, et al. (2024). In-context language learning: architectures and algorithms
  - von Oswald, et al. (2023). Transformers learn in-context by gradient descent
- Both?
  - Pan, et al. (2023). What in-context learning ”learns” in-context: disentangling task recognition and task learning

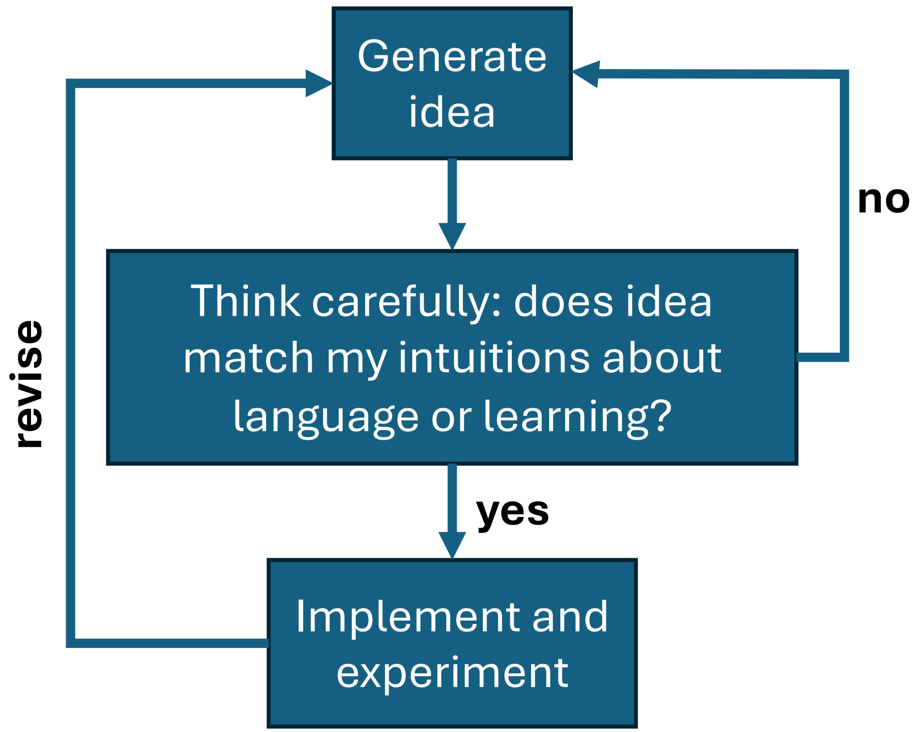




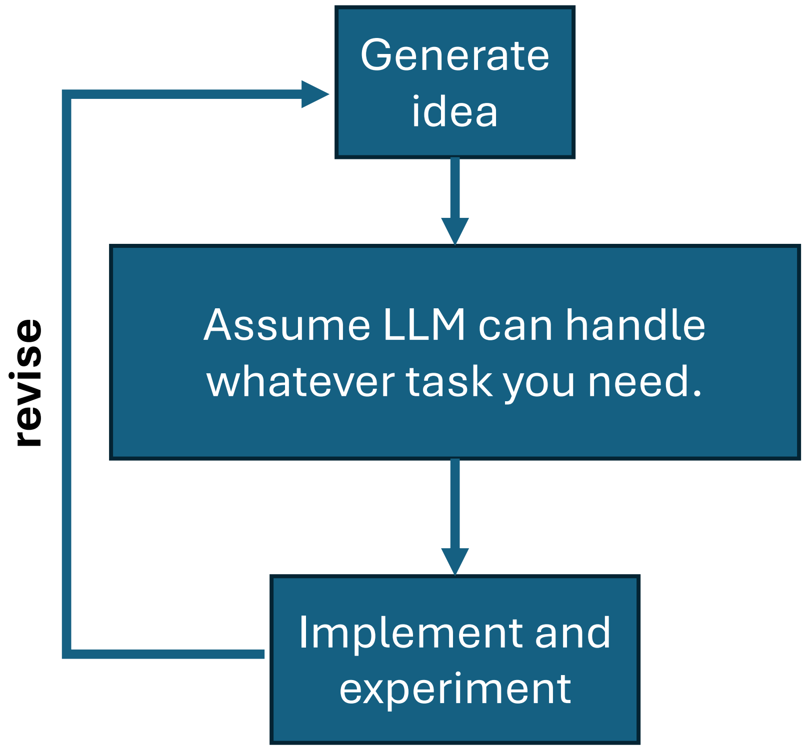
Min, et al., Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?, EMNLP 2023

On the dangers of over-trusting emergent and multi-purposes abilities

Research workflow, pre-LLM



Research workflow (for some), now



2. How LLMs are built

Main stages of building a LLM

Data Preparation

Very important.
Often under-appreciated

Pre-training

1. Architecture design
2. Grid infrastructure
3. LM objective

Fine-tuning & Alignment

1. Instruction Tuning
2. Alignment: RLHF

GPT-4
Llama
etc

Downstream Fine-Tuning

For developers building on top of LLM:
e.g. Parameter-efficient Fine-tuning (PEFT)

Data Preparation

"To spur innovation in data-centric AI approaches, perhaps it's time to hold the Code fixed and invite researchers to improve the data."

– Andrew Ng, 2021

- Nontrivial questions:
  - Optimal mix of data sources
  - What kind of cleaning
  - How to tokenize
  - How to guess the impact of all these decisions?

Data mixture (an example)

Stage 1 Pre-training

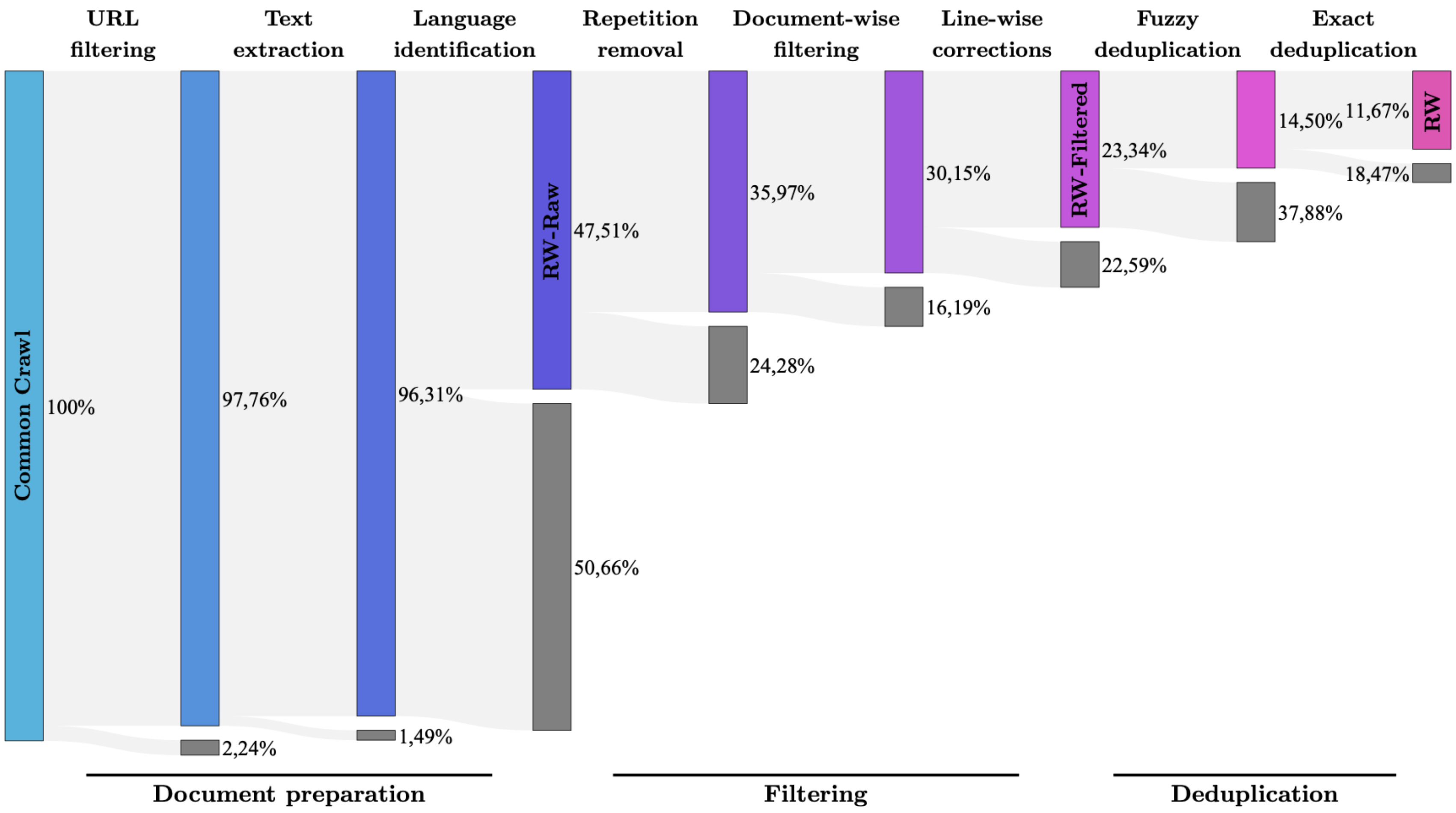
Dataset	Tokens (B)	Epochs	Sampling prop. (%)
RedPajama-CommonCrawl	879.37	1	63.98
RedPajama-GitHub	62.44	1	4.54
RedPajama-Books	65.18	2.5	4.74
RedPajama-ArXiv	63.32	2	4.61
RedPajama-StackExchange	21.38	1	1.56
C4 from 6 CC dumps (2019 - 2023)	191.50	0.2	13.93
Wikipedia-English	19.52	4	1.42
Wikipedia-21 other languages	62.04	2	4.51
Pile-DM Mathematics	7.68	2	0.56
Apex code from 6 CC dumps	2.09	1	0.15
Total	1374.52		100

Stage 2 Pre-training

Dataset	Tokens (B)	Sampling prop. (%)
Data from stage 1	55	50
BigCode Starcoderdata	55	50
Total	110	100

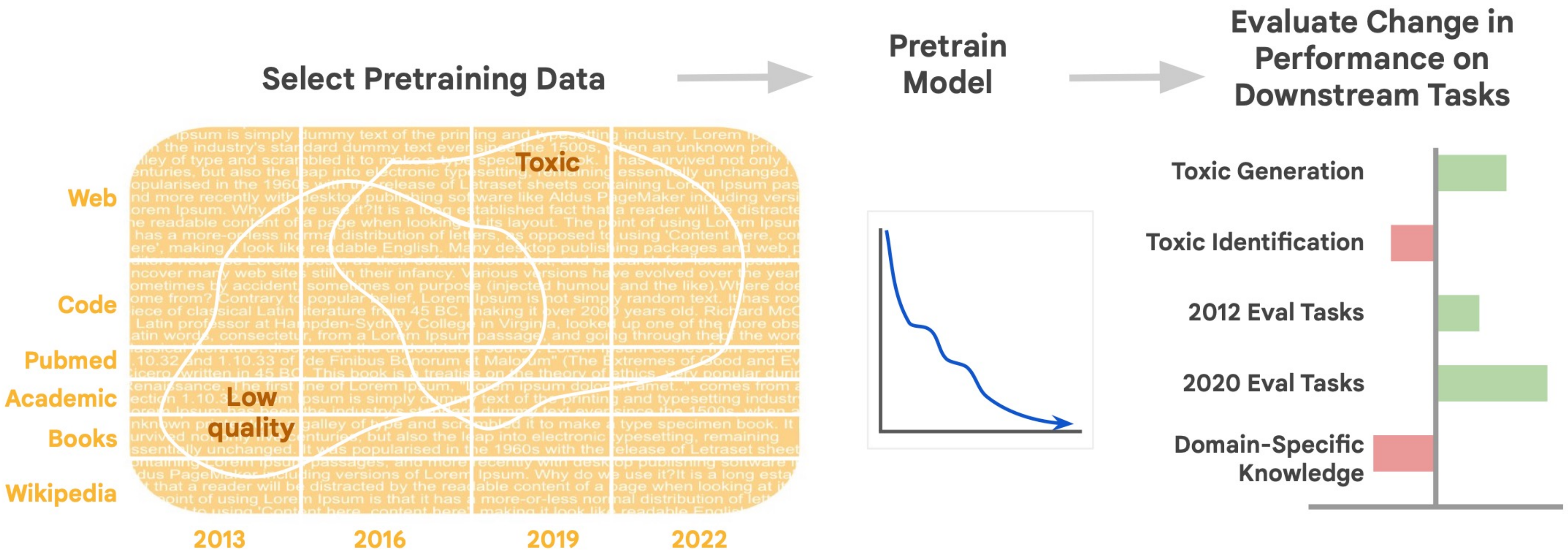
From: Shafiq Joty's CLSP Seminar (2024).
Unleash the Potential of LLMs through Task & Data Engineering
https://www.youtube.com/@jhuclsp/videos
Nijkamp (2023). XGen-7B Technical Report

Data Filtering (an example)



Penedo, et al. (2023) The Refined Web dataset for Falcon LLM

A Pretrainer's Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity [Longpre et al, NAACL 2024]

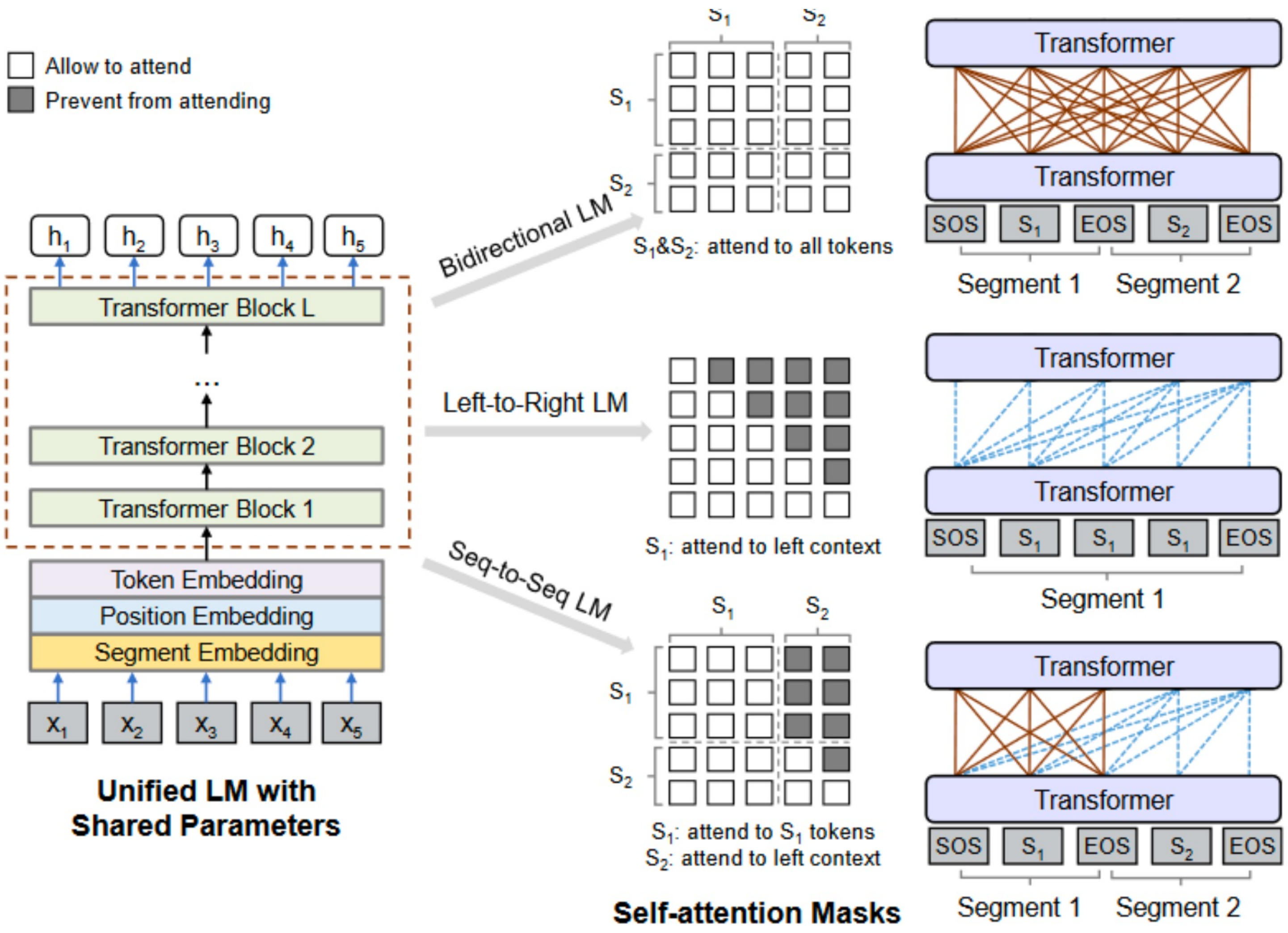


Some findings: strongly encourage to read the paper!

- "temporal shift between evaluation data and pretraining data leads to performance degradation, which is not overcome by finetuning"
- "a trade-off between performance on standard benchmarks and risk of toxic generations… there does not exist a one-size-fits-all solution to filtering."

LLM Architectures

- Decoder-only transformer is now standard
  - But still need to decide hyperparameters
  - Larger context window
- There are also architecture innovations:
  - e.g. Mixture-of-Experts, State-space models



From: Dong, et al. (2019). Unified LM Pre-training for NLU and Generation

Number of Parameters

- <1 billion
- 1-10 billion
  - Llama2-7b (7b)
  - Bloom-3b (3b)
- 10-100 billion
  - GPT-3.5-turbo (20b)
  - Alpaca (13b)
- >100 billion
  - davinci-003 (175b)
  - Claude 2 (137b)

- Impact on model size & inference
  - usu. 4 bytes per parameter:
    - Bloom-3b → 12GB on disk
  - 2 bytes per parameter (FP16):
    - Llama2-70b → 140GB on disk
- Impact on training
  - extra ~6x bytes for optimizer state, gradient, temporary activations
    - Bloom-3b → 72GB GPU RAM
  - hardware requirements:
    - Smaller models: Single or Multi-GPU training on single node (w/ 4 NVIDIA A100, 40GB RAM each)
    - Larger models: Multi-node Multi-GPU distributed training required. Fast interconnect.

Pre-training Cost

- Llama2-70b:
  - 6000 GPUs for 12 days,
  - trained on 2TB tokens of text,
  - 4k sequence length
  - 1x10^24 FLOPS → $2M

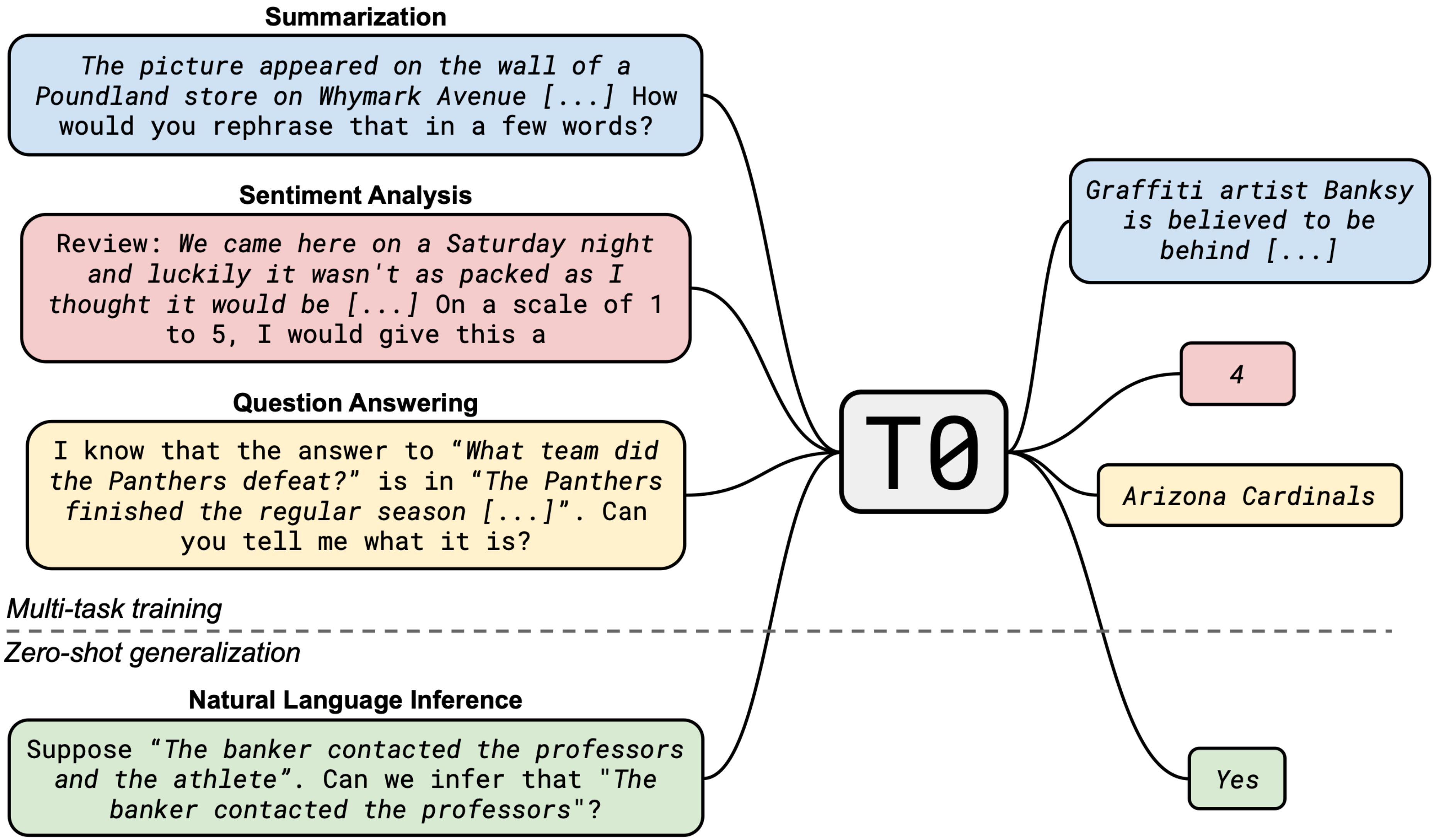
- xGen-7b:
  - trained on 1.5T tokens of text
  - 8k sequence length
  - $150k on Google Cloud TPU-v4

Fine-Tuning, Instruction Tuning, Alignment

- I’ll group everything under Fine-Tuning because they’re not all that different in my opinion.
  - Is “Alignment” really aligning models to “human values” more so than running backprop on manually created data?
- Why fine-tune?
  - Specialize to a task
  - Learn to chat
  - Get used to prompts and instructions
  - Inject more human feedback

Sanh, et al. (2022). Multitask prompted training enables zero-shot task generalization

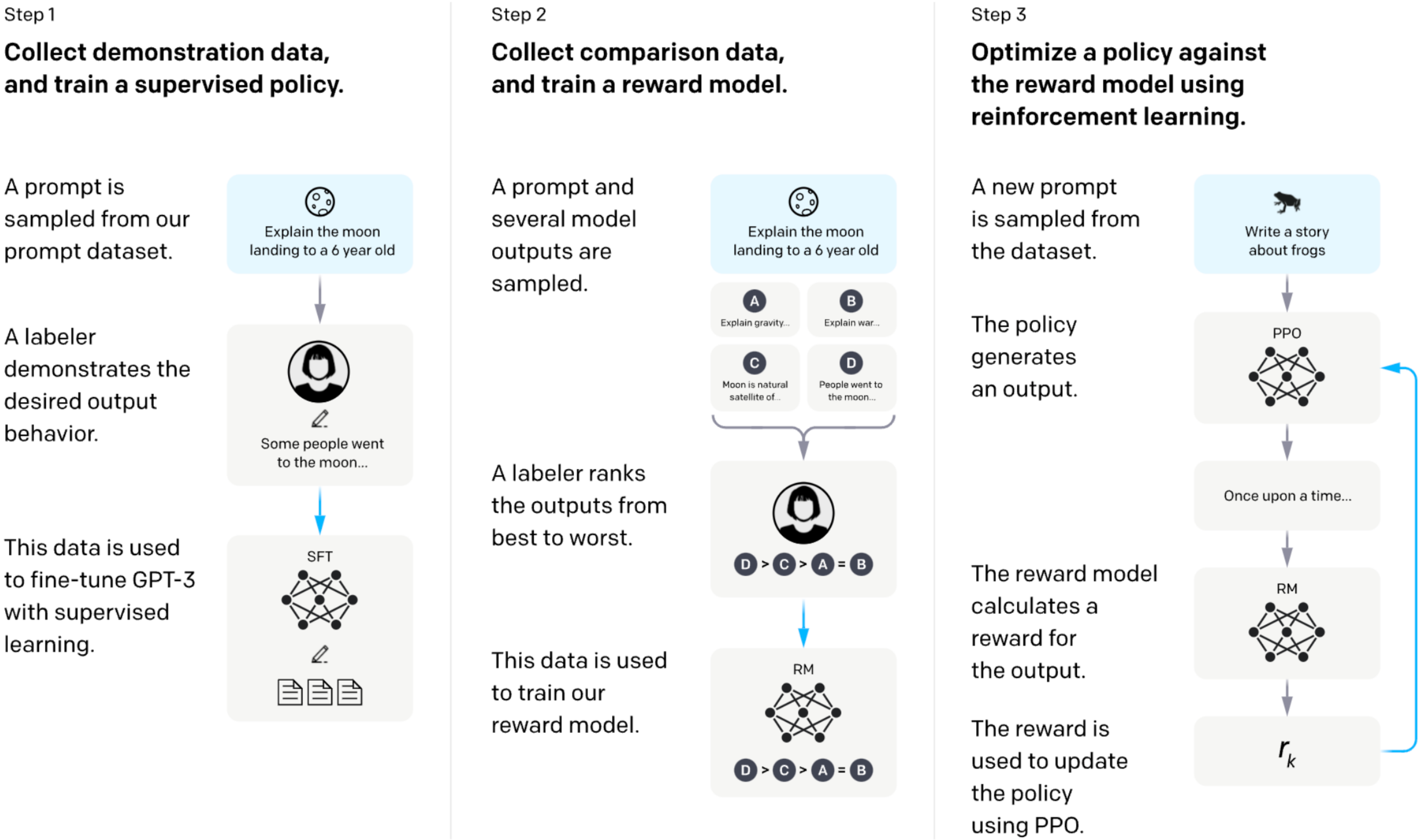
Instruction Tuning



Comparison labels

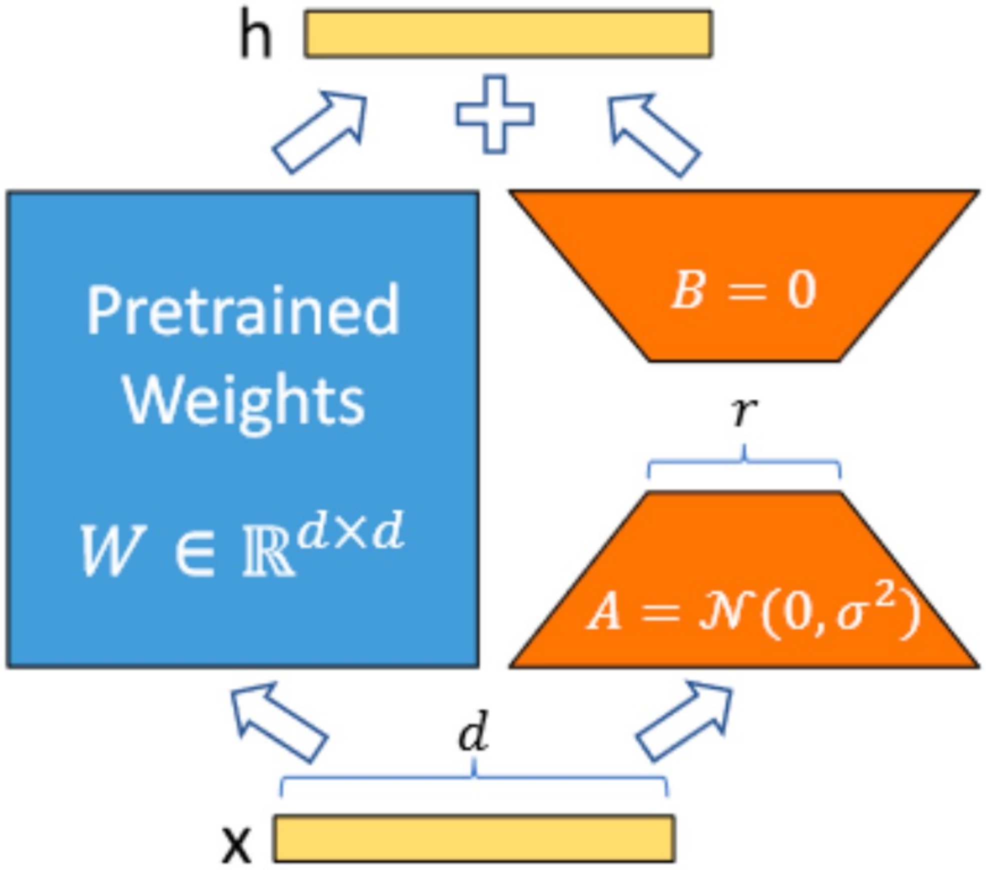
e.g. Reinforcement Learning by Human Feedback (RLHF)

- Ouyang et. al. (2022). Training language models to follow instructions with human feedback



Parameter Efficient Fine-Tuning (PEFT)

Example
LoRA: Low-rank adaptation of large language models [Hu 2021]



General-purpose LLMs → Specialized LLMs

Data Preparation	Pre-training	Fine-tuning & Alignment	Downstream Fine-Tuning

BloombergGPT (train from scratch)

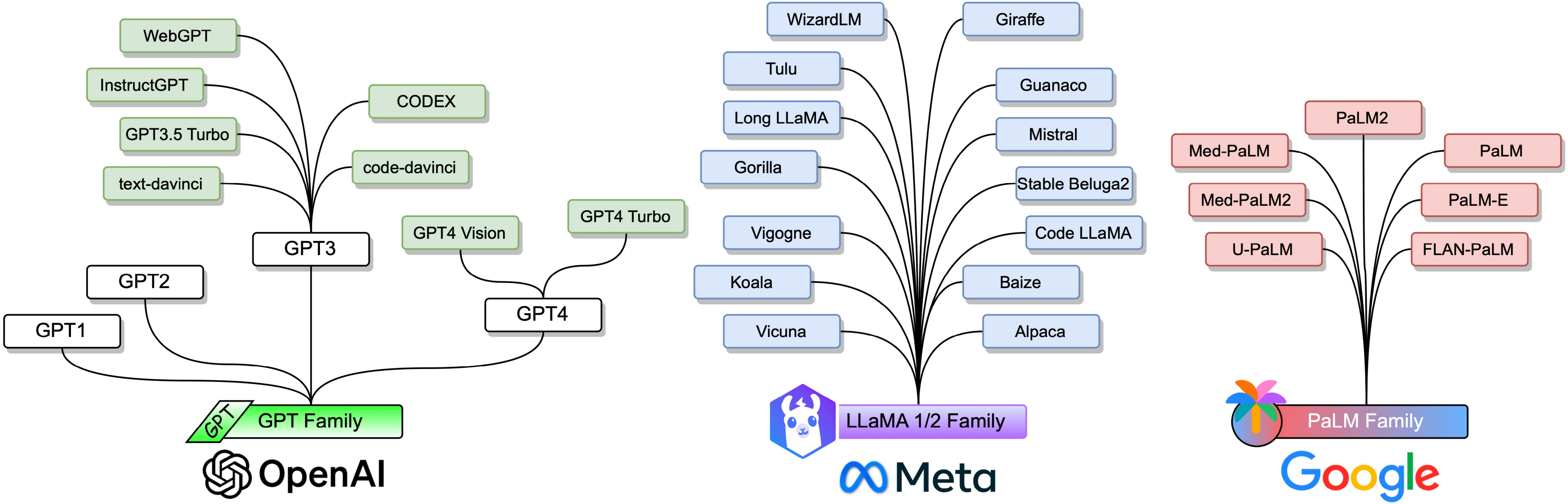
LLaMa→ AstroLlama (train on specialized data)

LLaMa → Vicuna (trained on Chat-GPT generated data)

Xuan's talk later on Machine Translation task

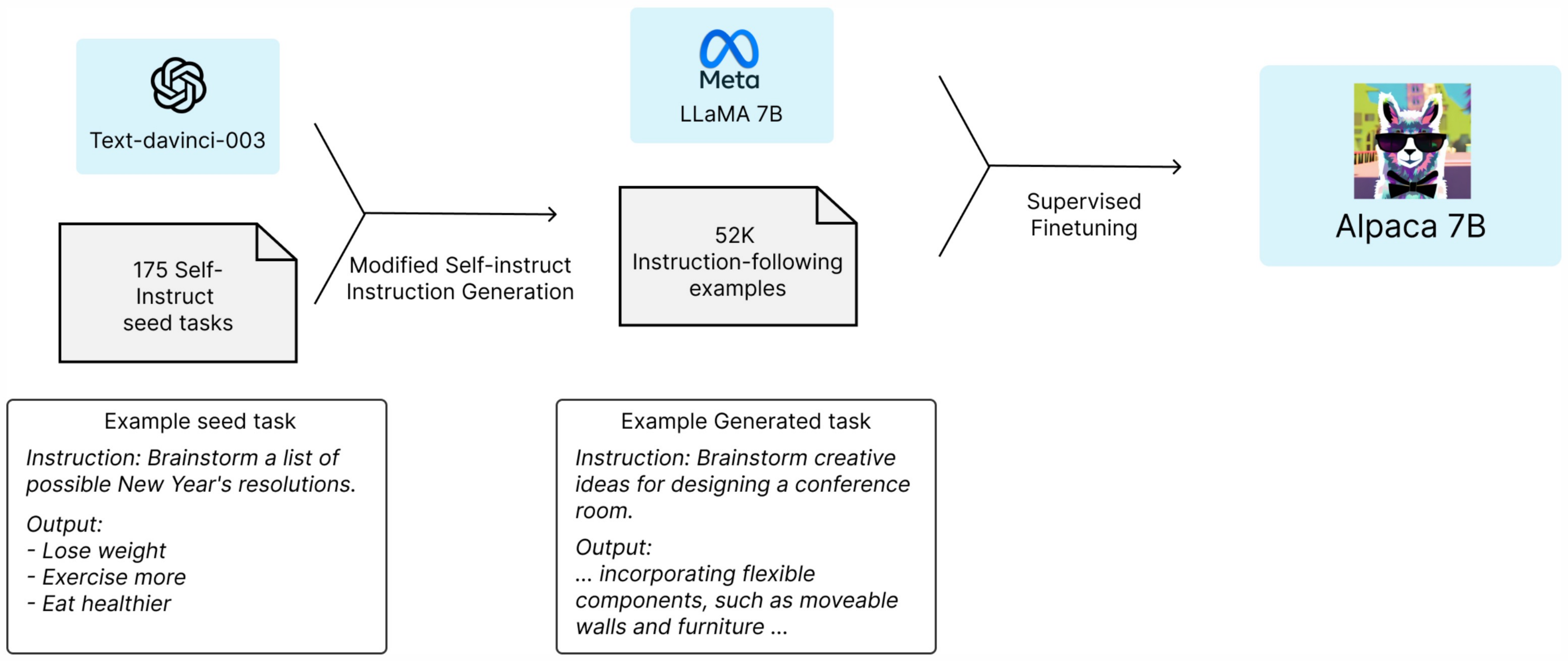
- Question: Generalized→Specialized LLM, or dedicated model from start?

3. Survey of popular LLM implementations

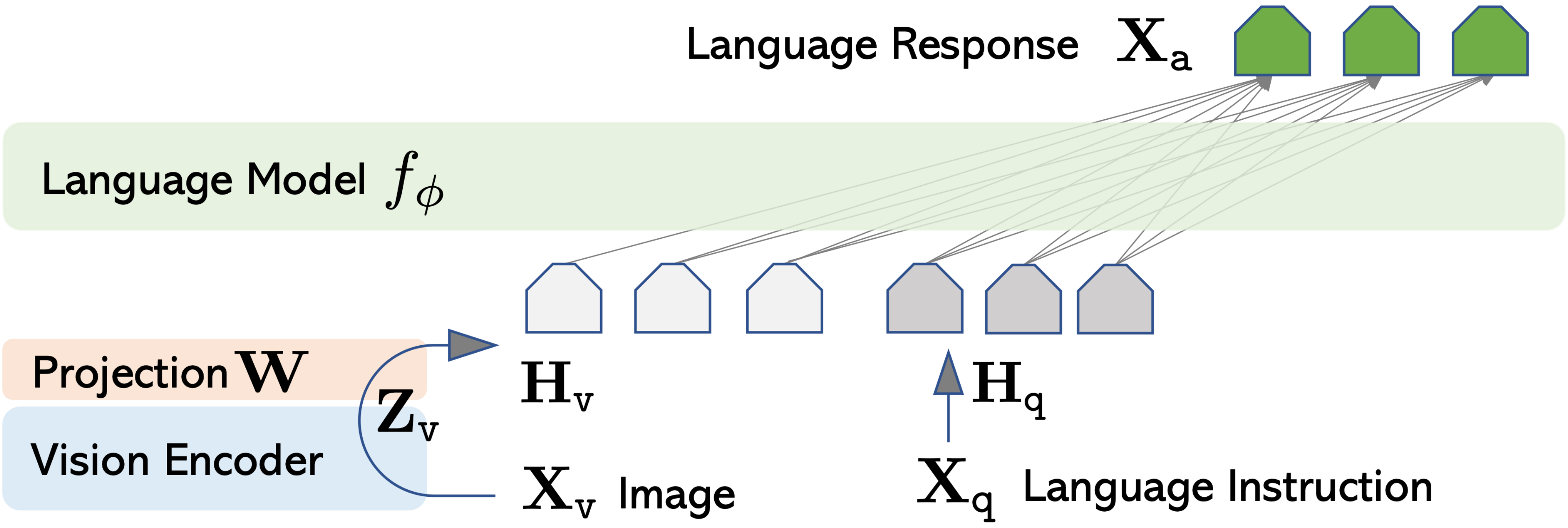


Minae, et. al. (2024). Large Language Models: A Survey

ALPACA: instruction tuning on top of LLaMa

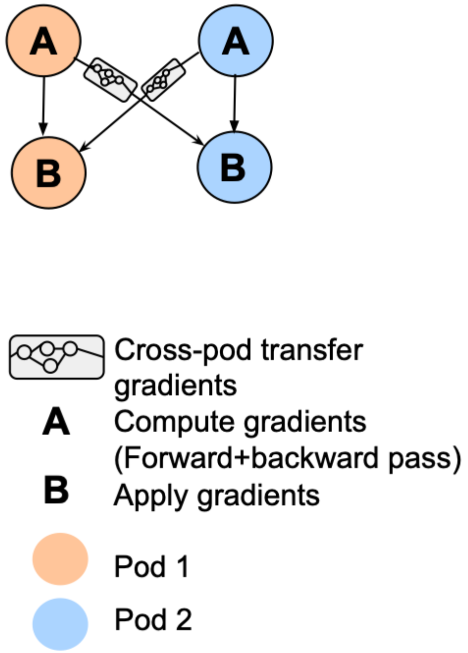


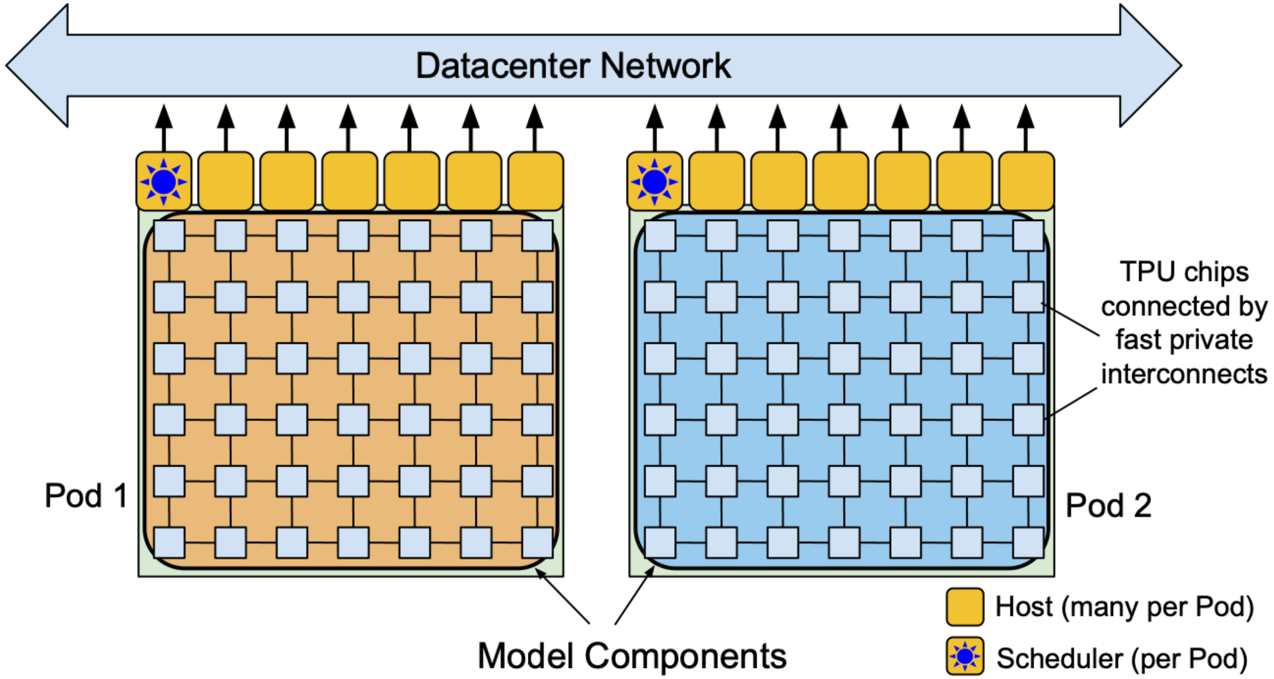
LLaVA (multimodal LLM)



PaLM

- 540b model, trained on 6144 TPU-v4 via model/data parallelism
- Illustrates growing importance of Systems work





[Chowdhery, et. al., 2022] PaLM: Scaling Language Modeling with Pathways

BLOOM (open-access model)

BLOOM: A 176B-Parameter Open-Access Multilingual Language Model

BigScience Workshop*

Major Contributors
Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Thomas Wolf, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel

Dataset
Aaron Gokaslan, Adi Simhi, Aitor Soroa, Albert Villanova del Moral, Alexandra Sasha Luccioni, Alham Fikri Aji, Amit Alfassy, Angelina McMillan-Major, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Akiki, Christopher Klamm, Colin Leong, Colin Raffel, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Hugo Laurençon, Huu Nguyen, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Lucile Saulnier, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, Margaret Mitchell, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Pawan Sasanka Ammanamanchi, Pedro Ortiz Suarez, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Roman Castagné, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Samson Tan, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Stella Biderman, Suhas Pai, Suzana Ilić, Sydney Zink, Teven Le Scao, Thomas Wang, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Yacine Jernite, Zaid Alyafeai, Zeerak Talat

Tokenization
Arun Raja, Benjamin Heinzerling, Benoît Sagot, Chenglei Si, Colin Raffel, Davut Emre Taşar, Elizabeth Salesky, Lucile Saulnier, Manan Dey, Matthias Gallé, Pedro Ortiz Suarez, Roman Castagné, Sabrina J. Mielke, Samson Tan, Teven Le Scao, Thomas Wang, Wilson Y. Lee, Zaid Alyafeai

Prompt Engineering
Abheesht Sharma, Albert Webson, Alexander M. Rush, Alham Fikri Aji, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Canwen Xu, Colin Raffel, Debajyoti Datta, Dragomir Radev, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan

Which one would you choose for research/deployments: Open or closed models?

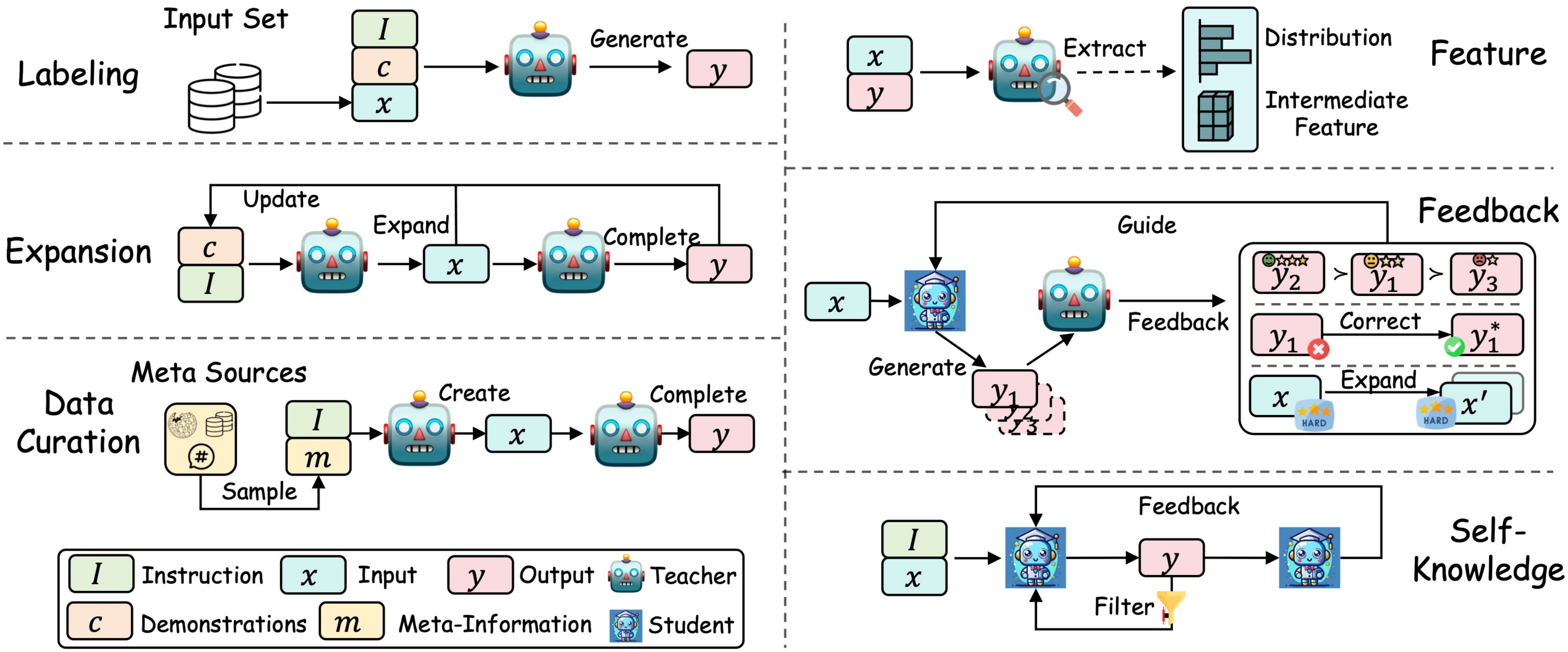
What are the important factors?

4. Quick sampling of some advanced topics

Smaller models

e.g. Distillation

Figure Xu et. al. (2024) A Survey on Knowledge Distillation of Large Language Models



e.g., Efficient memory management for LLM Serving with PagedAttention [Kwon 2023]

Efficient Inference & Serving

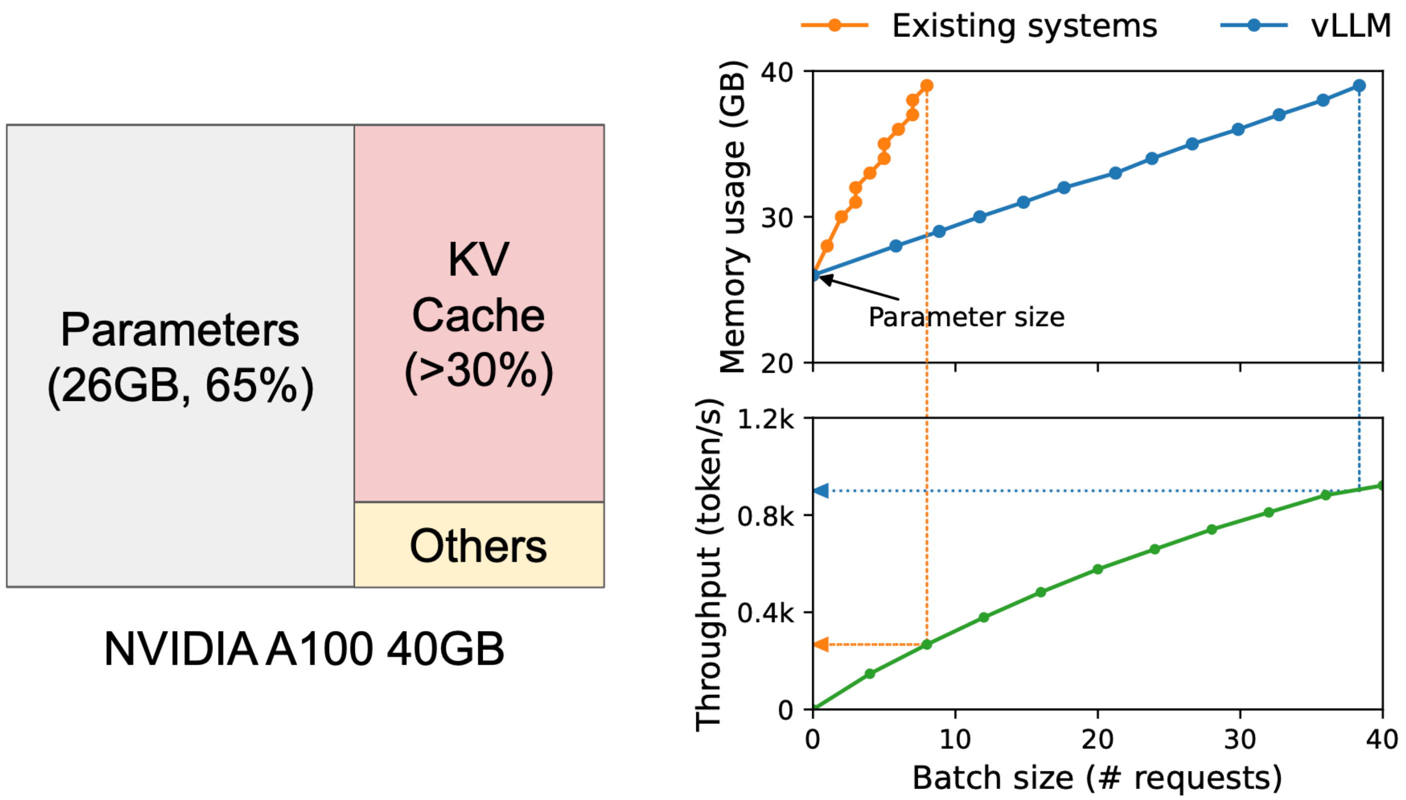
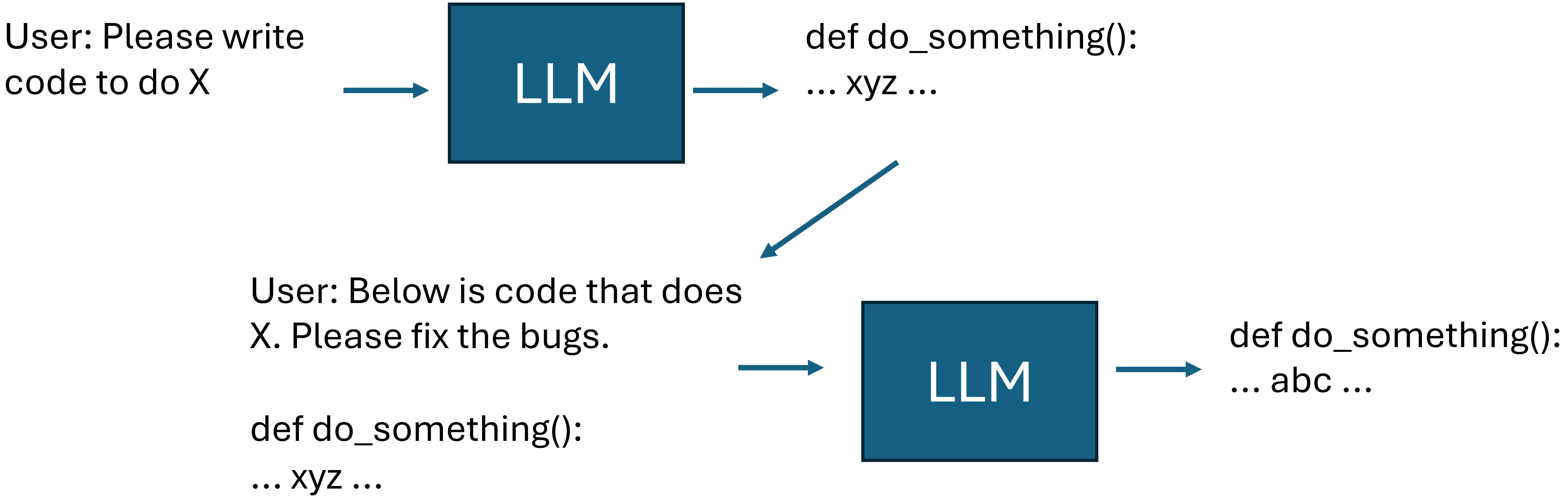


Figure 1. *Left:* Memory layout when serving an LLM with 13B parameters on NVIDIA A100. The parameters (gray) persist in GPU memory throughout serving. The memory for the KV cache (red) is (de)allocated per serving request.

Better prompts

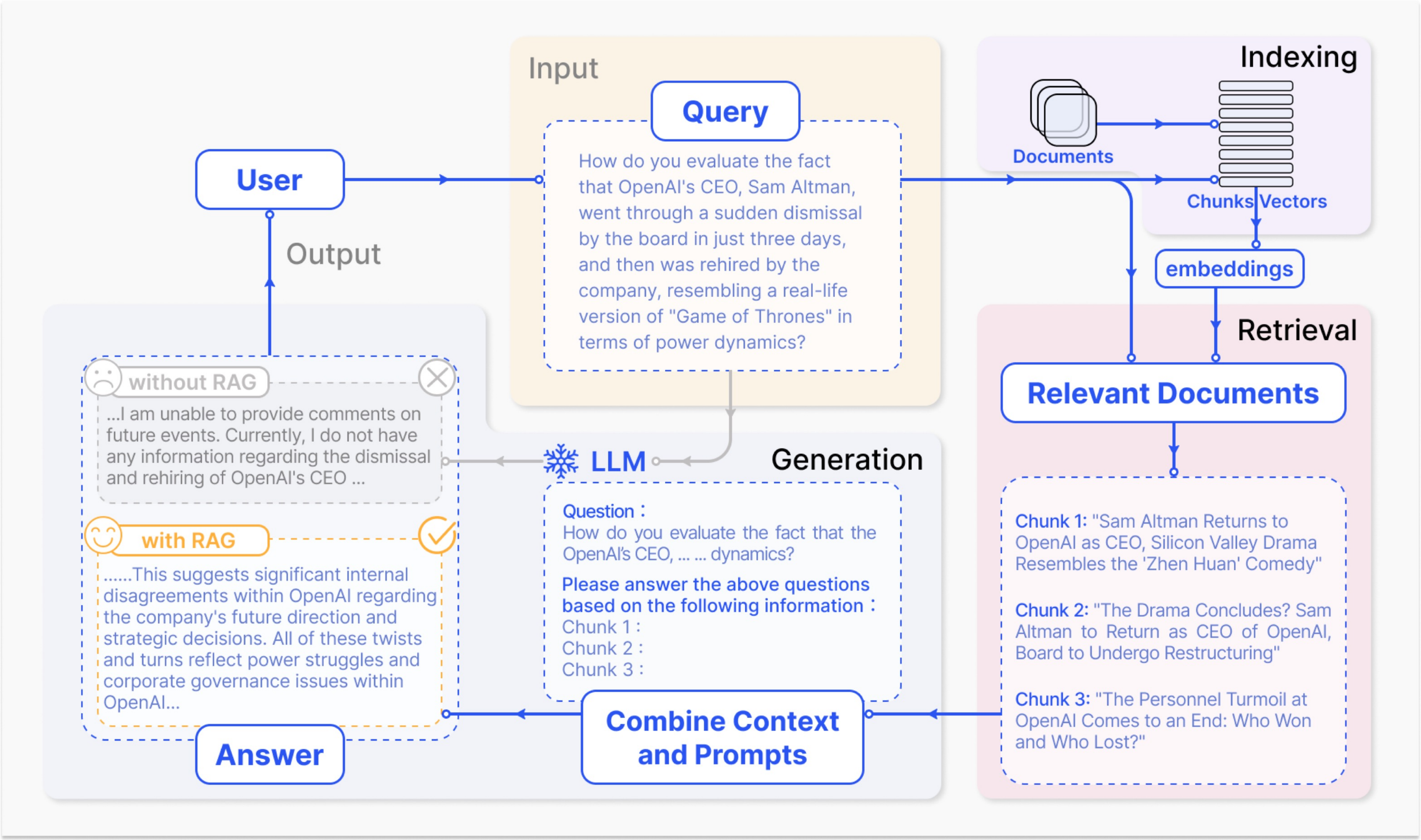
- e.g., Reflection: [Shinn et. al., (2023). Reflexion: Language Agents with Verbal Reinforcement Learning]
- (System 2 Thinking)



e.g., Retrieval Augmented Generation (RAG)

Using External Knowledge

From: Gao et. al., (2024). Retrieval-Augmented Generation for Large Language Models: A Survey



Using External Tools

- e.g., Schick et. al. (2023) Toolformer: Language Models Can Teach Themselves to Use Tools

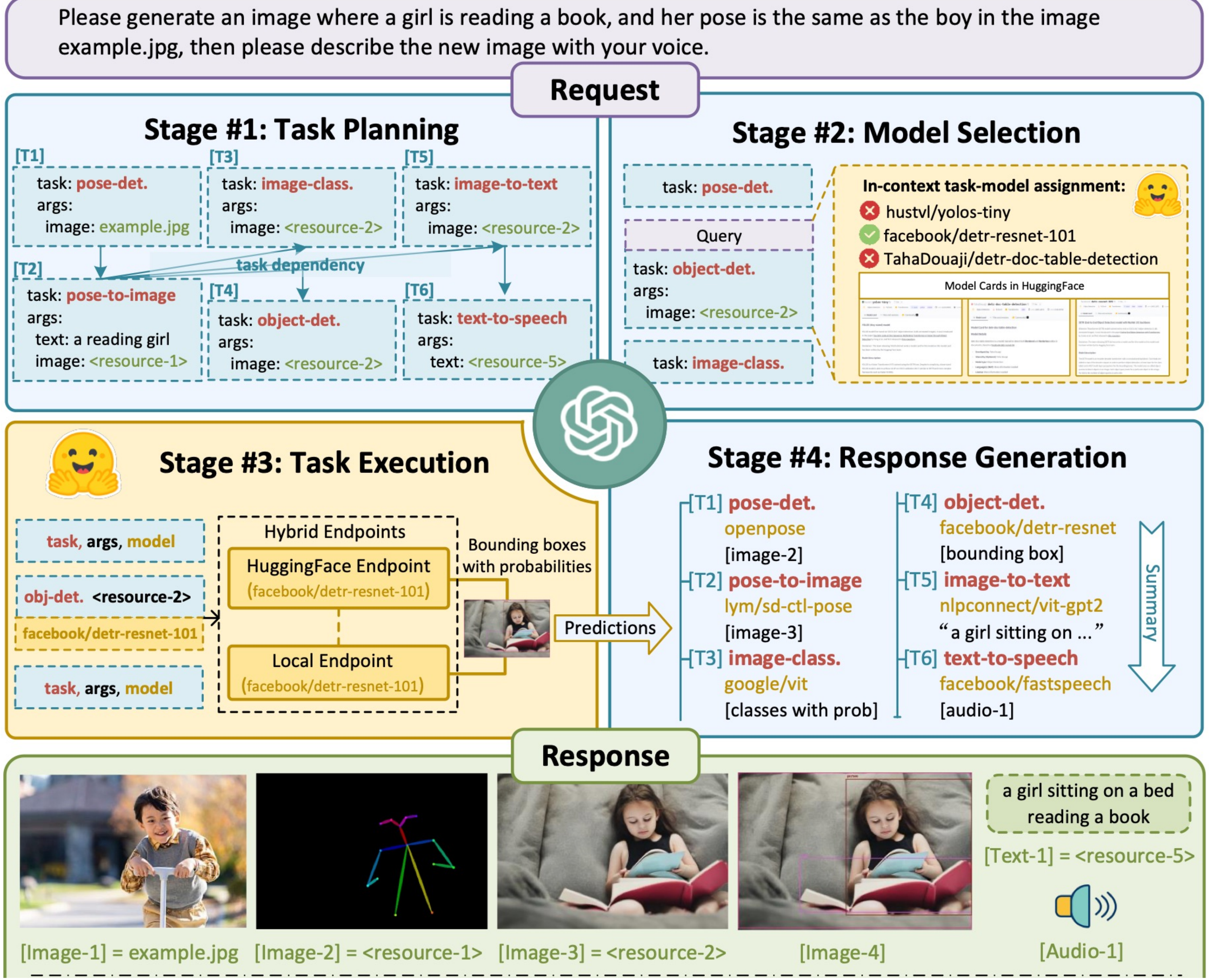
> The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

> Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

LLM Agents

The Future? Combines tool use & planning

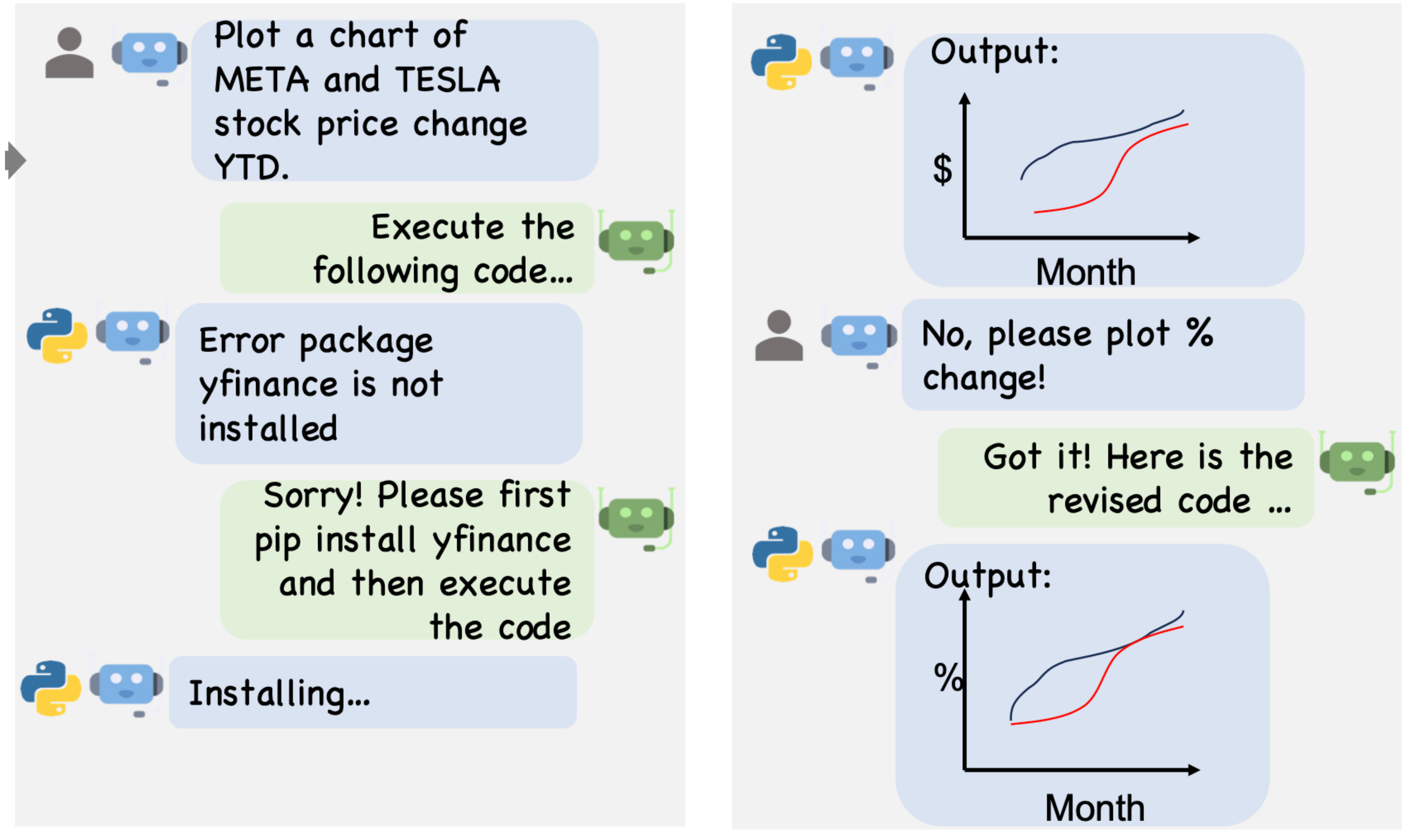
- e.g. Shen (2023) HuggingGPT



Multiple LLM Agents

The Future? LLMs working together to solve complex tasks

- e.g. Wu (2023) AutoGen



OpenAI 2024/05 demo:

Two GPT-4os interacting and singing

https://www.youtube.com/watch?v=MirzFk_DSiI

Responsible AI: broad spectrum of topics

- Reliability
  - e.g. reduce or detect hallucination
- Fairness
  - e.g. mitigate harmful bias and toxicity
- Accountability
  - e.g. design proper data governance policy
- Privacy
  - e.g. use LLM inference with confidentiality protection
- Security
  - e.g. guard against adversarial attacks on model

Summary

1. Why LLMs are fundamentally different from what came before
   → Definition by intended use: multi-purpose & emergent
2. How LLMs are built

Data Preparation	Pre-training	Fine-tuning & Alignment	Downstream Fine-Tuning

3. Survey of popular LLM implementations
4. Quick sampling of some advanced topics

Today's Agenda

- 9:00-10:20: Tutorial on LLM basics
- 10:20-10:40: Break
- 10:40-12:00: Research showcase: invited talks that illustrate different research areas related to LLMs
- 12:00-13:00: Lunch
- ~~13:00-13:30: Computer lab setup~~
- 13:30-17:00: Lab

Next up: Research Showcase

Invited talks:

- SCALE 2024 Workshop on Video-based Event Retrieval (Reno Kriz)
- Machine Translation with LLMs (Xuan Zhang)
- Continuous Training of LLMs (William Fleshman)
- LLM Performance on Challenging Analogy Tasks (Andrew Wang)
- Detection of Machine-Generated Text (Rafael Rivera Soto)
- LLMs for Hardware Design (Michael Tomlinson & Paola Vitolo)