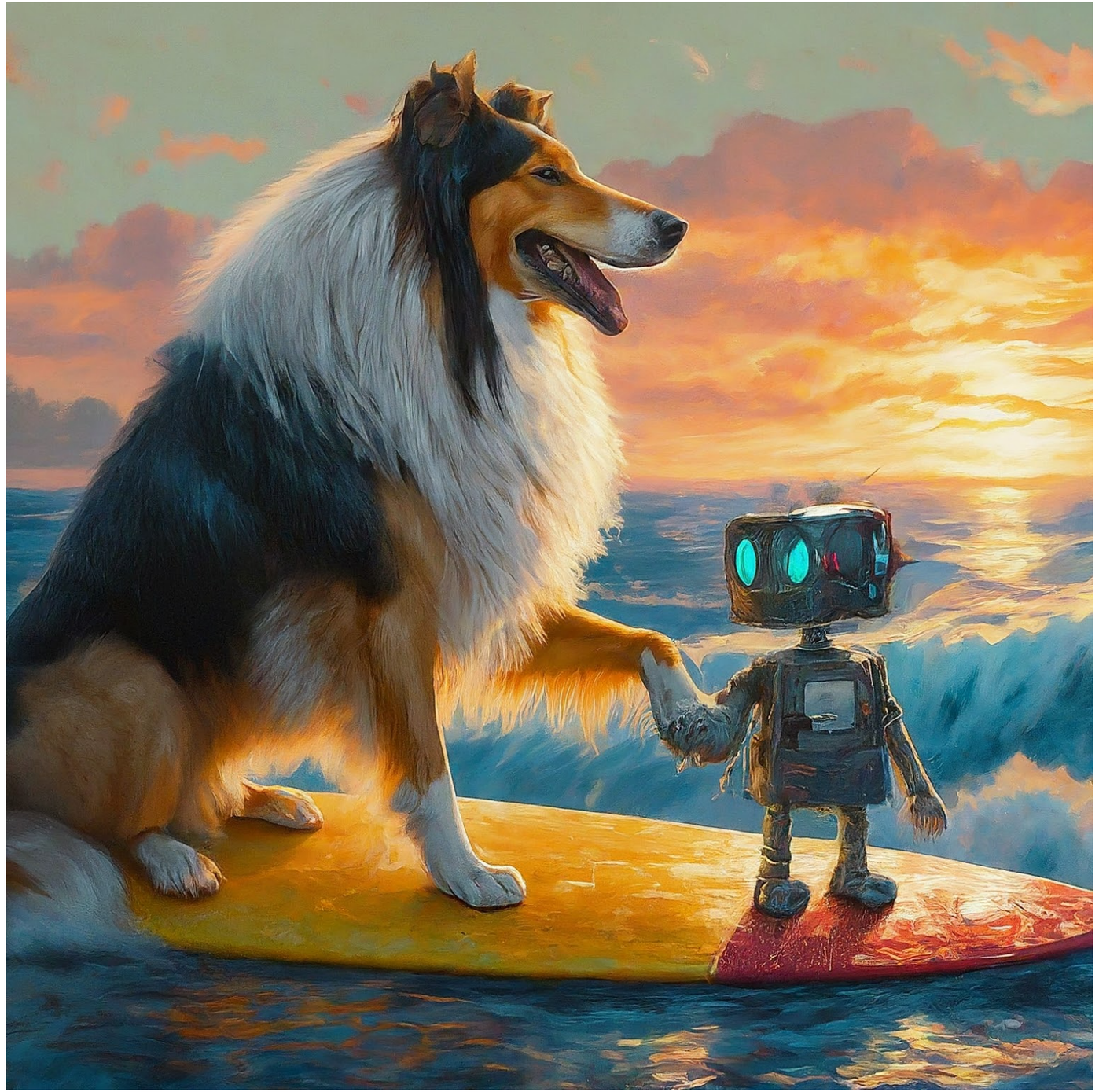
Large Language Models

Class 01: Overview

CSCE 689 :: Fall 2024
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee



Welcome to LLMs!

This is a research-oriented special topics course!

(This is not an applied, build LLM demos course.)

No homework, no exams.

But, lots of papers to read, questions to ask, opportunities to dive deep!

We should all view this as a ***collaborative learning experience***

There are many other special topics courses this semester

Deep Learning and LLMs (Galanti)

Programming LLMs (Jeff Huang)

Generative AI (Tu)

Multi-Agent Reinforcement Learning (Zhou)

Vision Foundation Models (Zhang)

Trustworthy Natural Language Processing (K-H Huang)

Big Data Systems (Nguyen)

Algorithms for Big Data (Crawford)

Who am I?

Prof. James Caverlee (call me “Cav”)

Faculty member in CS here at TAMU (since 2007)

Visiting Researcher at Google DeepMind

- Working on LLMs + personalization; LLMs and data/model efficiency

Who are you?

Course logistics

Be here on time: we will start at 12:45pm sharp everyday

We will finish at 2pm sharp

There is a big waitlist to get into this class, so take this opportunity to really engage

> What I mean: come to class, ask questions, go above just chasing a grade

Guest Lectures

I am working on scheduling guest lectures from folks in industry working at the forefront of LLMs

Stay tuned ...

Course Assignments / Grading

Grading

10% Engagement

30% Class Presentation

10% Lecture Feedback

50% Project

Engagement

Since this is a small-ish course, I will know everyone's name and face. You are expected to come to all class meetings and participate in the discussions. For each student-led presentation, you are encouraged to ask questions aloud or post questions to the course polling system.

Part of your engagement grade is class attendance. **You are expected to attend every class, but you may miss up to three class meetings without providing an excuse.**

Class Presentation

A majority of this course will be student-led.

Students will work in pairs to read recent research papers, create slides, co-present to the class, and lead the discussion. We will set the course schedule in the first week (or so) of the course so you will know when you are to present.

Class Presentation

Your slides will be due to me ~~one week~~ 2 days before your in-class presentation. For example, if your in-class presentation is on September 24, then you must submit a link to your slides here on Canvas (see the assignment) by September ~~17~~22 at 11:59pm. These should be 95% finalized slides. I will provide feedback so you have time in the ensuing ~~week~~ two days to sharpen the slides for class.

Class Presentation

You must use the Google Slides template that we provide:

https://docs.google.com/presentation/d/1mLwE_T2zUSa5qntn5mkCQraBWN8vLBI7ezxypIVWDI4/edit?usp=sharing

We're using the "Simple Light" theme; do not change this!

Class Presentation

What do we expect? You should provide the appropriate background and in-depth treatment of at least 2-3 papers that we provide. You are welcome to supplement with additional readings and other materials. Your goal is to level up the class on your topic. You should assume that everyone has at least skimmed the papers that you will present. Your job is to go deeper and help us appreciate the contributions. Budget plenty of time for question-and-answers. This should be an interactive discussion, not just a boring lecture.

Class Presentation

For the slide content, you may not re-use existing slides. You may not also just copy-paste formulas and figures from the paper and be done (though some amount of this is not avoidable). We expect you to create your own figures, your own animations, your own examples, etc. to help communicate the material.

Lecture Feedback

Over the course of the semester, each student will be responsible for providing **written feedback on three in-class student-led presentations**.

Your feedback should be around 1 page long in PDF, submitted here on Canvas. Feedback will be due no later than one week after the in-class presentation (e.g., if you providing feedback for a presentation on September 17, then you must submit your feedback by 11:59pm on September 24).

Lecture Feedback

What do we expect in your feedback? You should comment on the content itself, the quality of the slides, clarity of the presentation (including Q-and-A), completeness of the materials, etc. You can use bullet points; this is not an essay. I will provide the feedback to the presenters, so your tone should be professional and your criticisms should be constructive.

Project

The course project is a major semester-long effort to advance research in large language models. **You will work in a team of 2 or 3 students** to go deep into some aspect of LLMs. Your final deliverable will be an in-class talk (around 10-15 minutes) plus a final paper in the style of a top conference publication.

Project

Proposal (due Sept 19): this is a 1-page proposal describing your effort. I will give feedback to help you shape your project.

In-class briefing slides (Oct 2): you will provide 2-3 slides giving a high-level overview of the project.

In-class briefing (Oct 3): we will reserve one class period for all teams to brief us on their projects.

Final presentation (Nov 19/21)

Project report (Nov 26)

Topics

https://docs.google.com/document/d/1hAyM6gwjacm-SEPSoctxafZFuKlC_kE4J7jEjrtw260/edit

CSCE 689 LLMs

(this is a tentative list of topics and papers; feel free to add additional papers and/or topics but be sure to use a highlighter color so I will know which ones have been added)

Transformers and New Directions (Linear Attention, Linear RNNs, State Space Models)

- Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention
- Longformer: The Long-Document Transformer
- Generating Long Sequences with Sparse Transformers
- Linformer: Self-Attention with Linear Complexity
- Efficiently Modeling Long Sequences with Structured State Spaces
- Mamba: Linear-Time Sequence Modeling with Selective State Spaces
- RWKV: Reinventing RNNs for the Transformer Era
- Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models

Parameter-Efficient Tuning, Compression

- LoRA: Low-Rank Adaptation of Large Language Models
- Controlling Text-to-Image Diffusion by Orthogonal Finetuning

Our plan

Today (Aug 20): Cav

Thurs (Aug 22): Cav

Tues (Aug 27): no class

Thurs (Aug 29): Cav

Tues (Sept 3): first student-led presentation!

Your job this week:

Add topics / papers that you are excited about

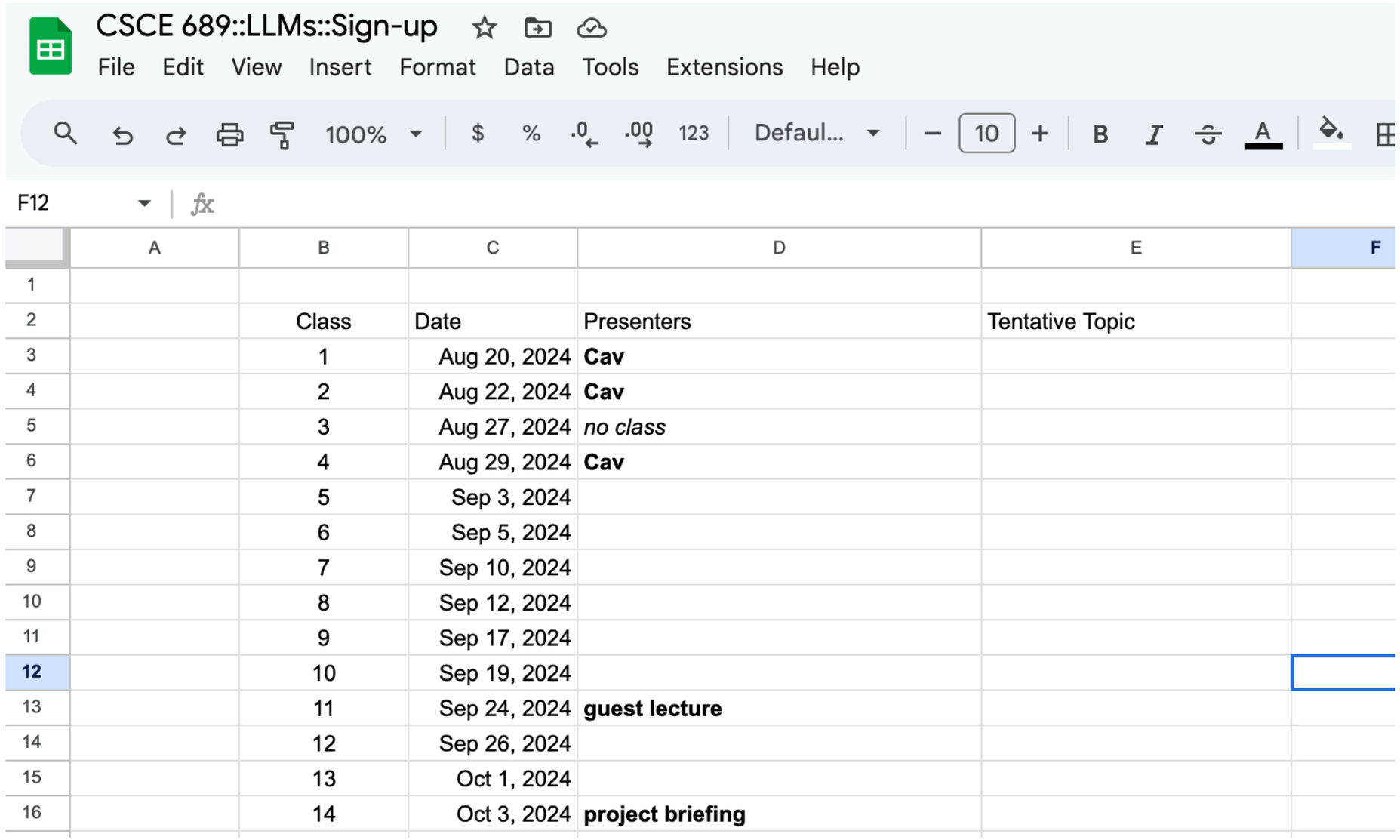
(Use a highlighter color so I know what has been added)

Sign-up for a slot (find a partner)

First student-led lecture is Sept 3 (two weeks from now): need to lock the first two (Sept 3/5) ASAP

(There is a chance I may have to move a few lectures around depending on our guest lecture schedule)

https://docs.google.com/spreadsheets/d/1nLeSmyM3QhRyexU99QuGIWW-RT5JMCY1C1FzJtvduCs/edit?usp=sharing



CSCE 689::LLMs::Sign-up

File Edit View Insert Format Data Tools Extensions Help

100% $ % .0 .00 123 Defaul... 10 B I A

F12

	A	B	C	D	E	F
1						
2		Class	Date	Presenters	Tentative Topic	
3		1	Aug 20, 2024	**Cav**		
4		2	Aug 22, 2024	**Cav**		
5		3	Aug 27, 2024	*no class*		
6		4	Aug 29, 2024	**Cav**		
7		5	Sep 3, 2024			
8		6	Sep 5, 2024			
9		7	Sep 10, 2024			
10		8	Sep 12, 2024			
11		9	Sep 17, 2024			
12		10	Sep 19, 2024			
13		11	Sep 24, 2024	**guest lecture**		
14		12	Sep 26, 2024			
15		13	Oct 1, 2024			
16		14	Oct 3, 2024	**project briefing**		

Questions / Concerns / Comments?

Let’s get started

Large language models (LLMs) refer to Transformer language models that contain hundreds of billions (or more) of parameters, which are trained on massive text data. LLMs exhibit strong capacities to understand natural language and solve complex tasks (via text generation).

https://arxiv.org/abs/2303.18223

What is a Language Model?

The probability of a sentence or a sequence of words:
$P(W) = P(w_1,w_2,w_3,w_4,w_5...w_n)$

Probability of an upcoming word: $P(w_5|w_1,w_2,w_3,w_4)$

A model that computes either of these:

$P(W)$ or $P(w_n|w_1,w_2...w_{n-1})$

is called a language model.

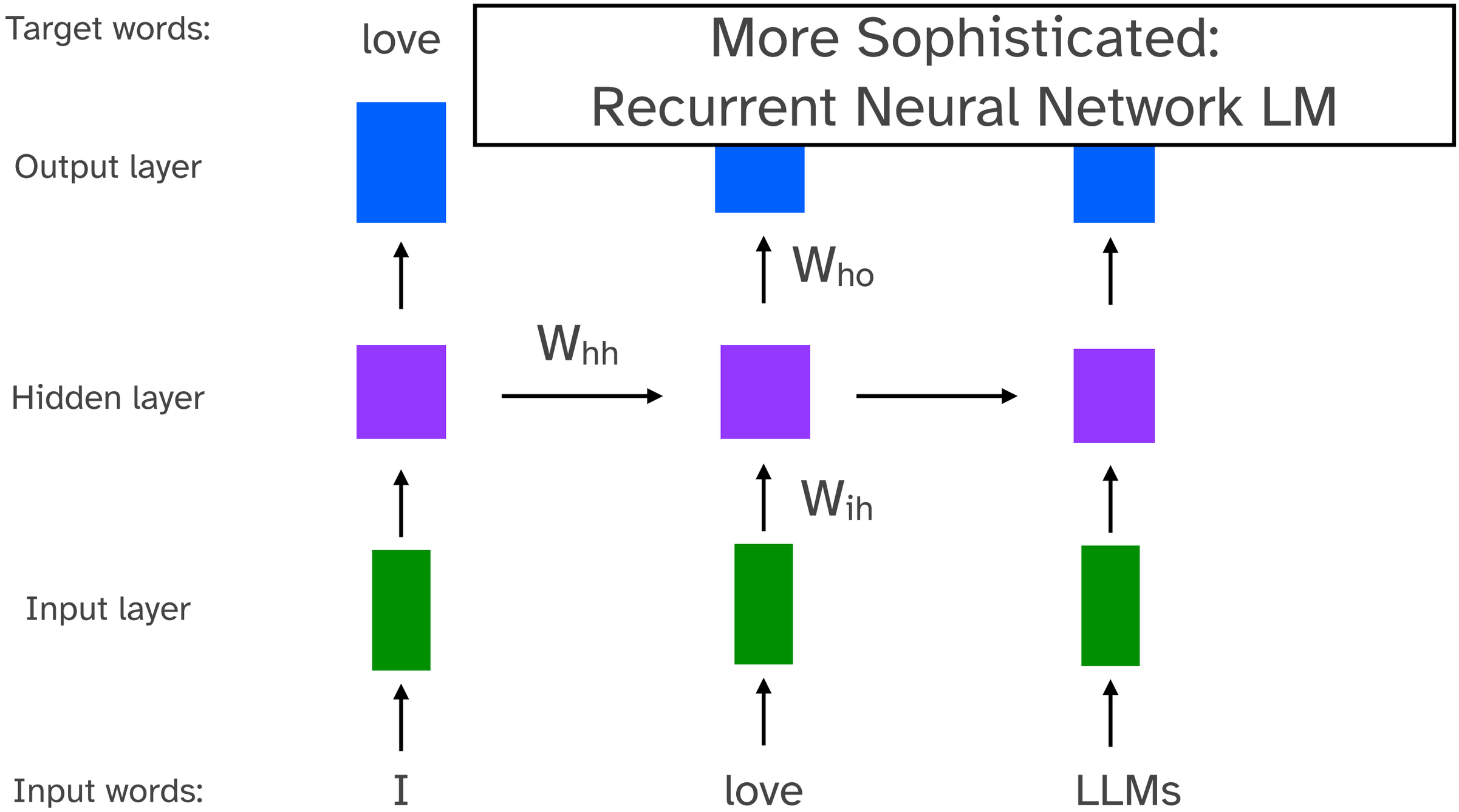
https://web.stanford.edu/~jurafsky/slp3/

Simple LM: Bigram model

Condition on the previous word

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

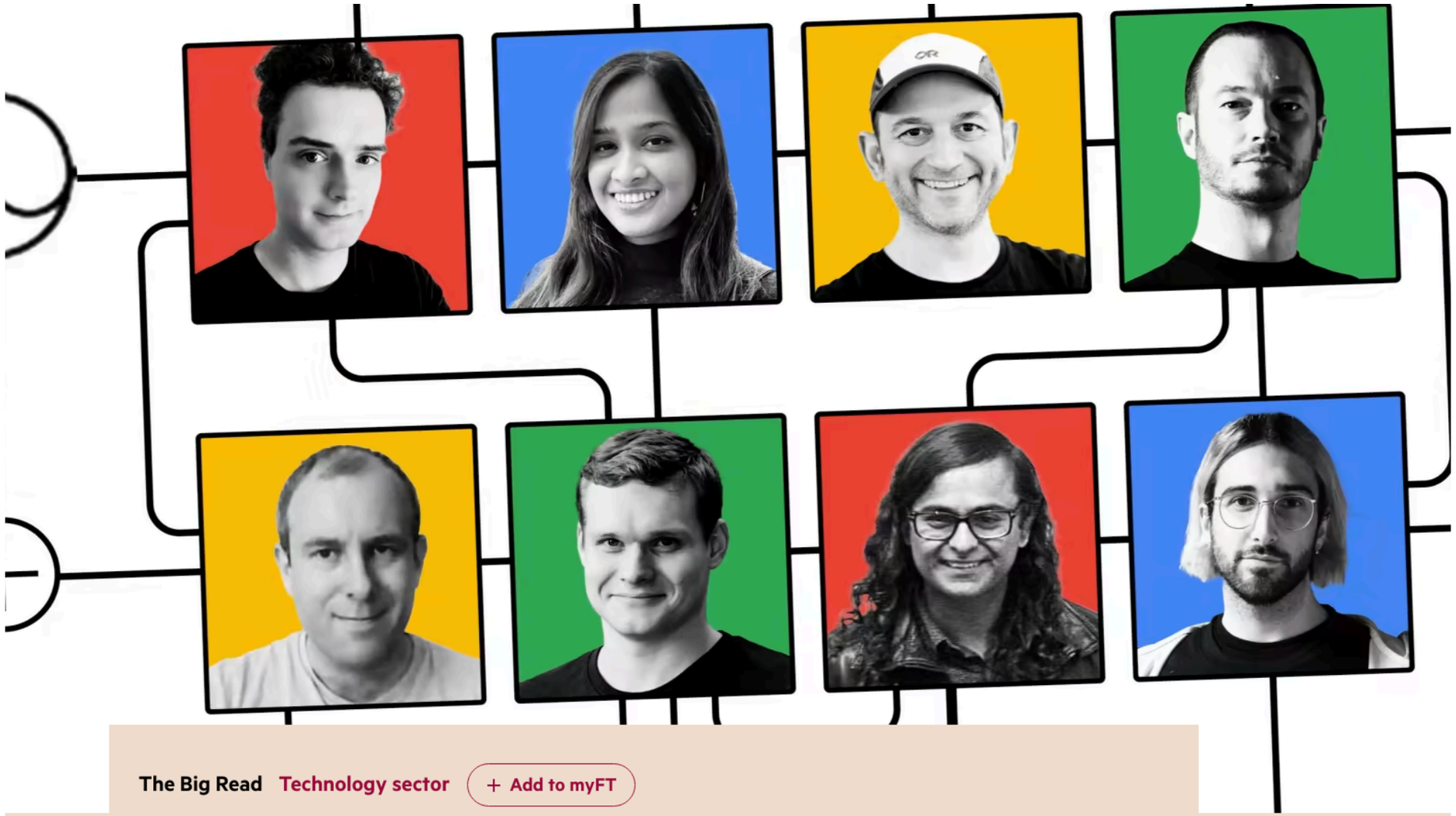
texaco, rose, one, in, this, issue, is, pursuing, growth, in, a, boiler, house, said, mr., gurria, mexico, 's, motion, control, proposal, without, permission, from, five, hundred, fifty, five, yen


Target words:
love
More Sophisticated:
Recurrent Neural Network LM
Output layer
W_{ho}
W_{hh}
Hidden layer
W_{ih}
Input layer
Input words:
I
love
LLMs

Transformers: Special Neural Network Model

Advantages:

- Can process entire input sequences in parallel (unlike sequential processing in RNNs) —> much faster training and inference
- Self-attention helps transformers focus on long-range dependencies in sentences (unlike RNNs which have trouble due to vanishing gradients), leading to better models of language
- Empirically: super strong performance!



The Big Read Technology sector + Add to myFT

Transformers: the Google scientists who pioneered an AI revolution

The eight research scientists wo
'transformer' model © FT monta

Their paper paved the way for the rise of large language models. But all have since left the Silicon Valley giant

Published at NeurIPS 2017

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

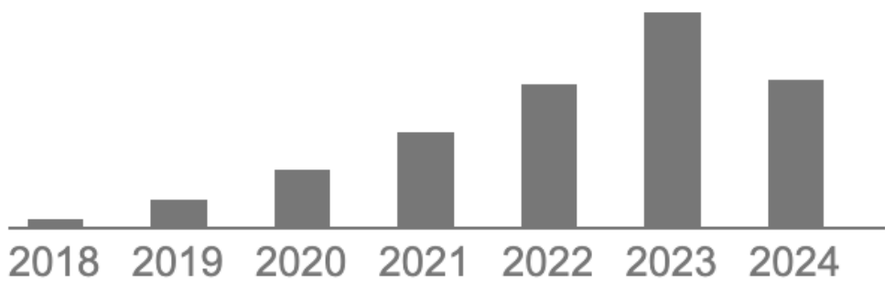
Llion Jones* Google Research llion@google.com

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

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Cited by 133199



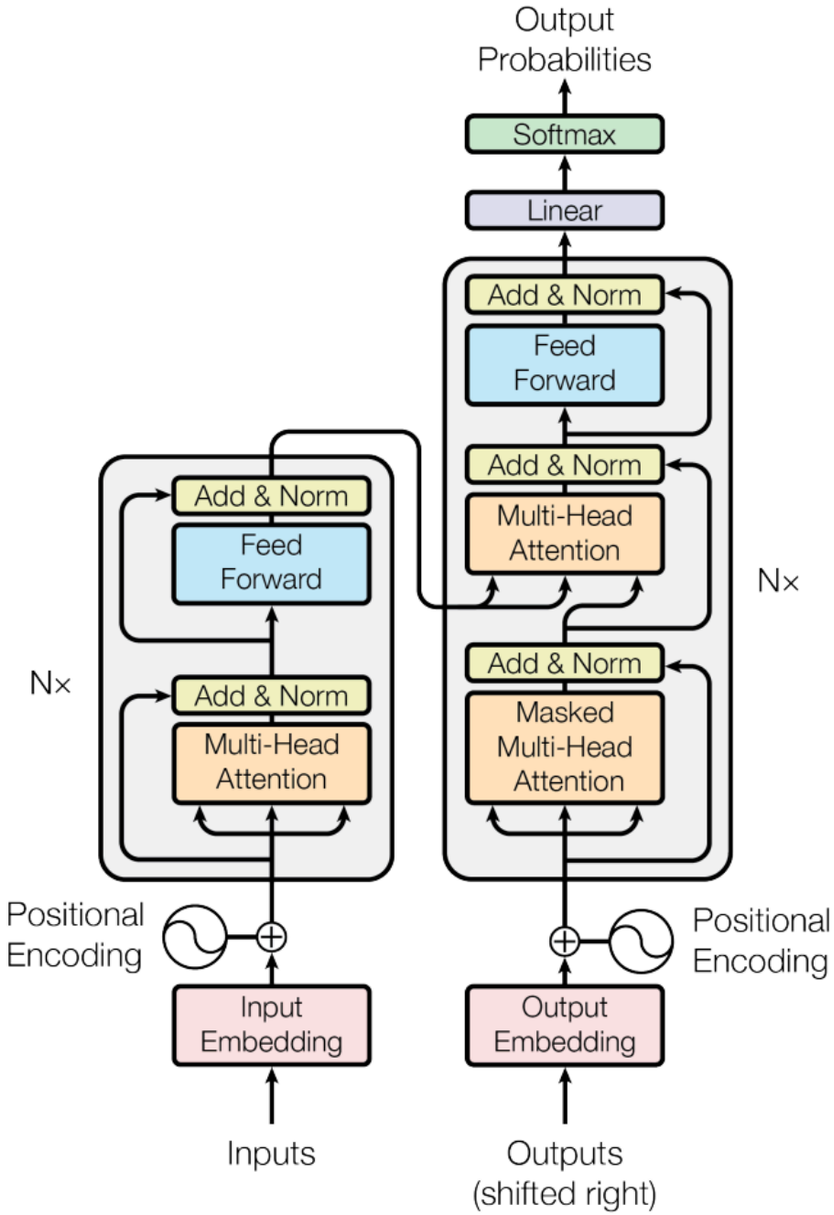
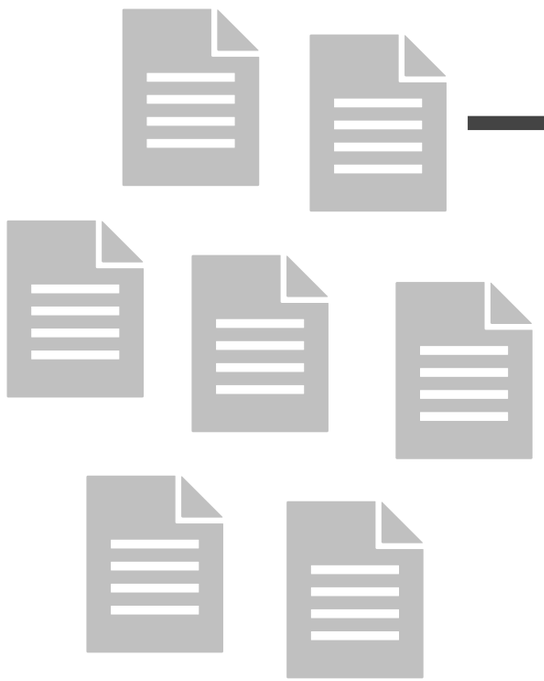


Figure 1: The Transformer - model architecture.

Typical Training Task: next token prediction

Huge scale: trillions of words



Texas A&M University (Texas A&M, A&M, or TAMU) is a public, land-grant, research university in College Station, Texas. It was founded in 1876 and became the flagship institution of the Texas A&M ...

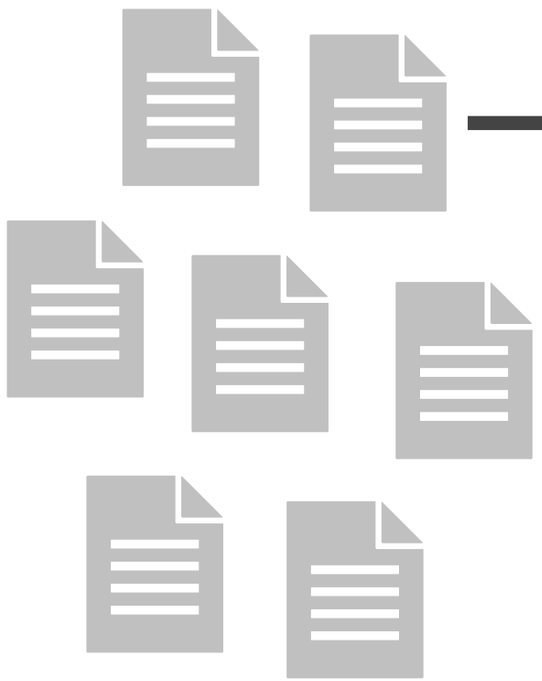
Large Language Model
(1b-1t parameters)

Texas

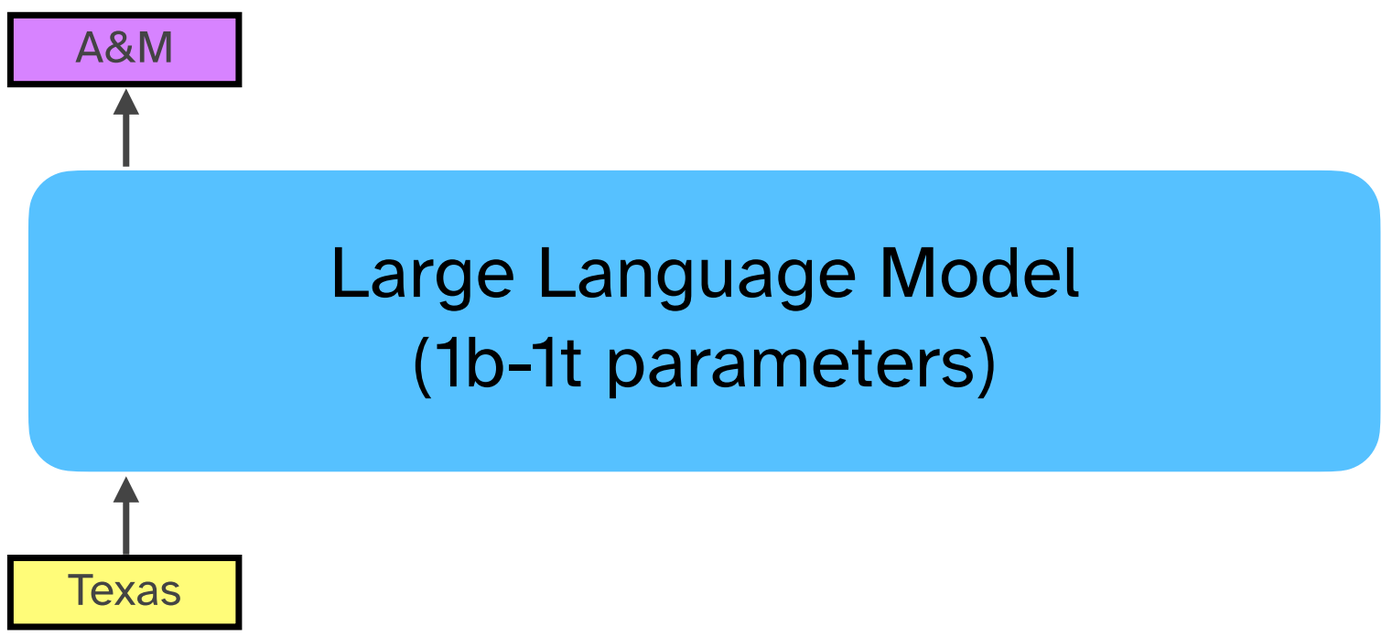
Web pages, books,
Wikipedia, Arxiv articles,
Code examples, ...

Typical Training Task: next token prediction

Huge scale: trillions of words



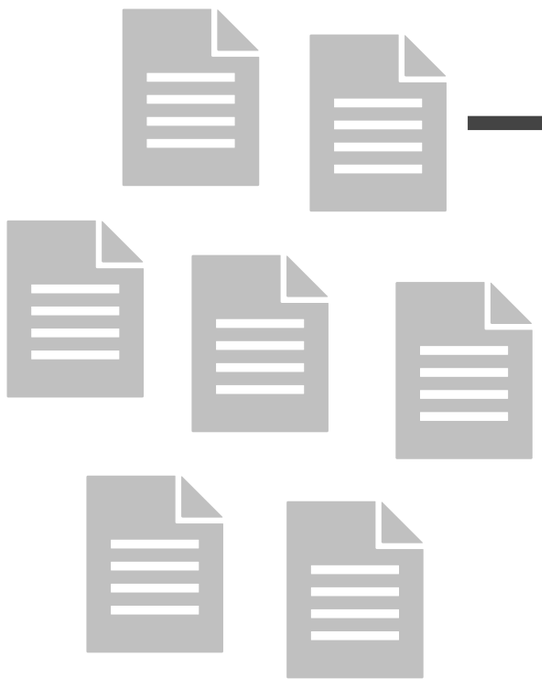
Texas A&M University (Texas A&M, A&M, or TAMU) is a public, land-grant, research university in College Station, Texas. It was founded in 1876 and became the flagship institution of the Texas A&M ...



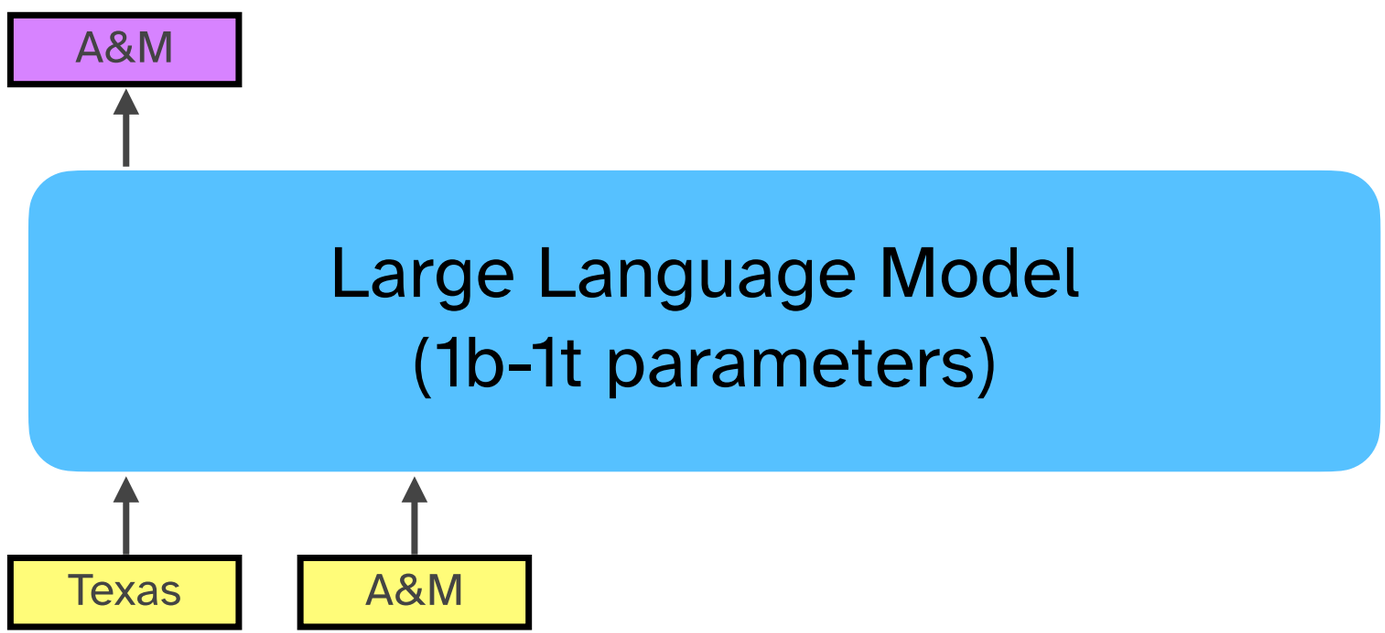
Web pages, books,
Wikipedia, Arxiv articles,
Code examples, ...

Typical Training Task: next token prediction

Huge scale: trillions of words



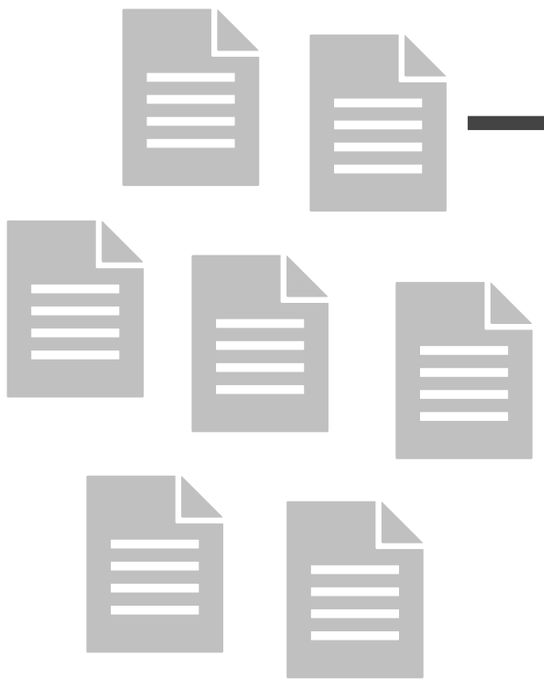
Texas A&M University (Texas A&M, A&M, or TAMU) is a public, land-grant, research university in College Station, Texas. It was founded in 1876 and became the flagship institution of the Texas A&M ...



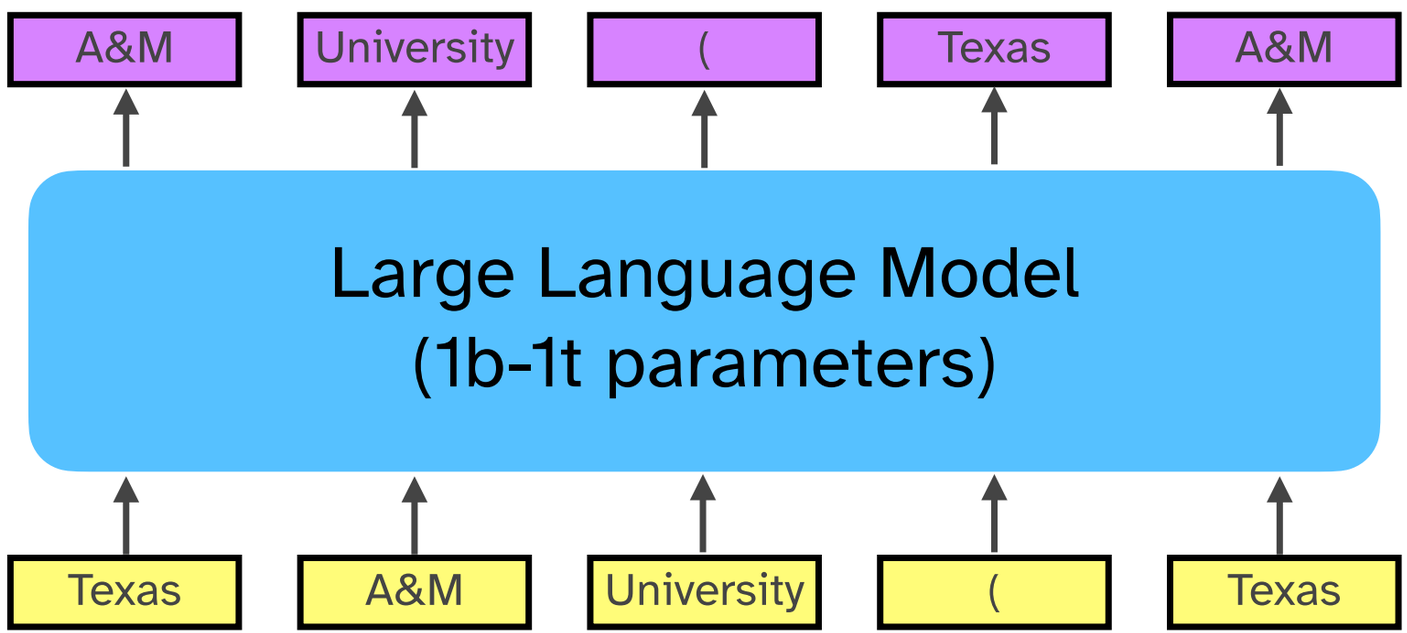
Web pages, books,
Wikipedia, Arxiv articles,
Code examples, ...

Typical Training Task: next token prediction

Huge scale: trillions of words



Texas A&M University (Texas A&M, A&M, or TAMU) is a public, land-grant, research university in College Station, Texas. It was founded in 1876 and became the flagship institution of the Texas A&M ...



Web pages, books,
Wikipedia, Arxiv articles,
Code examples, ...

So far …

Objective is next-word/token prediction

But what we really want is the model to align with the user's goal of completing a task or following instructions

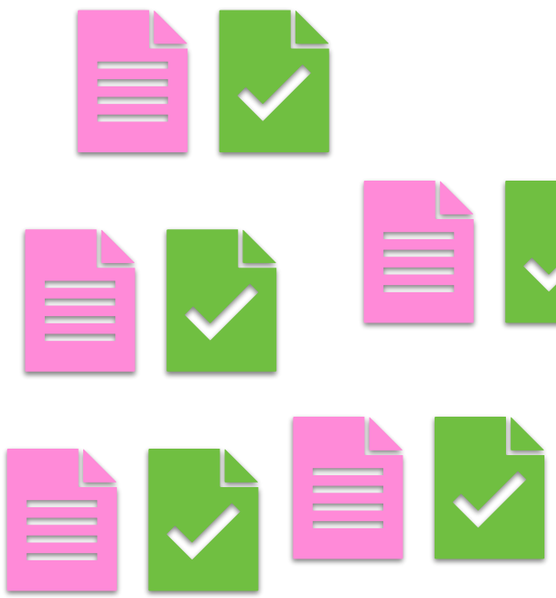
So, what if we give the model examples of successful instructions?

[INST] Write a python function to output 'Hello World' [\INST]

```
def hello_world():
  """Prints the classic 'Hello, world!' message."""
  print("Hello, world!")

# Call the function to print the message
hello_world()
```

Instruction tuning

Lots of pre-defined task/ response pairs (~100k)



Natural language inference, Summarization, code generation, Sentiment analysis, …

[INST] Write a python function to output ‘Hello World’ [\INST]

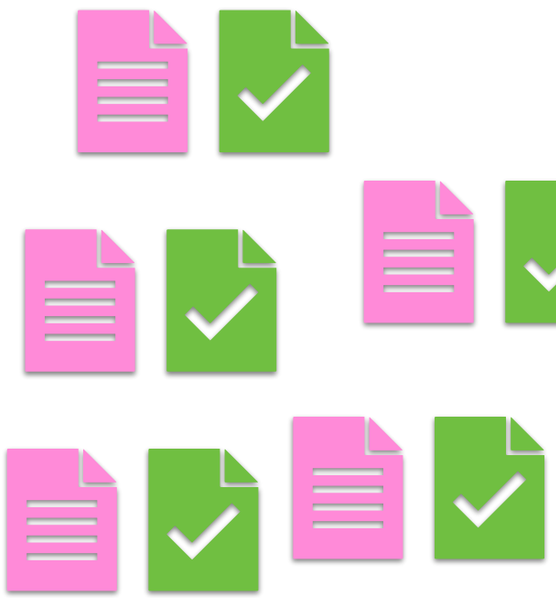
```
def hello_world():
  """Prints the classic 'Hello, world!' message."""
  print("Hello, world!")

# Call the function to print the message
hello_world()
```

Pretrained Large Language Model (1b-1t parameters)

[INST]

Instruction tuning

Lots of pre-defined task/ response pairs (~100k)



Natural language inference, Summarization, code generation, Sentiment analysis, …

[INST] Write a python function to output ‘Hello World’ [\INST]

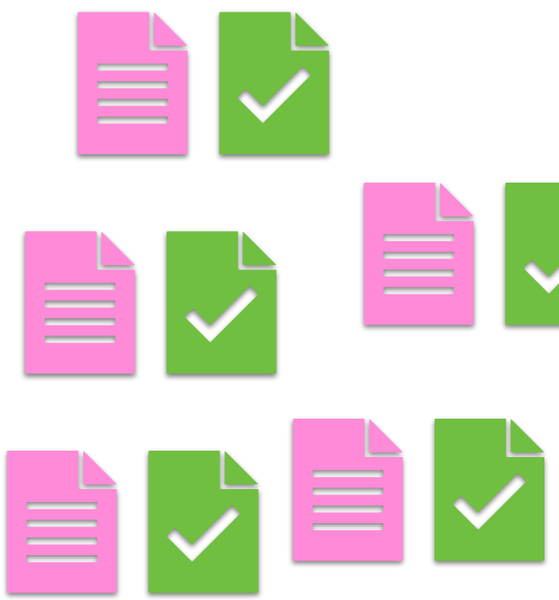
```
def hello_world():
  """Prints the classic 'Hello, world!' message."""
  print("Hello, world!")

# Call the function to print the message
hello_world()
```

Pretrained Large Language Model (1b-1t parameters)

[INST] Write … [\INST]

Instruction tuning

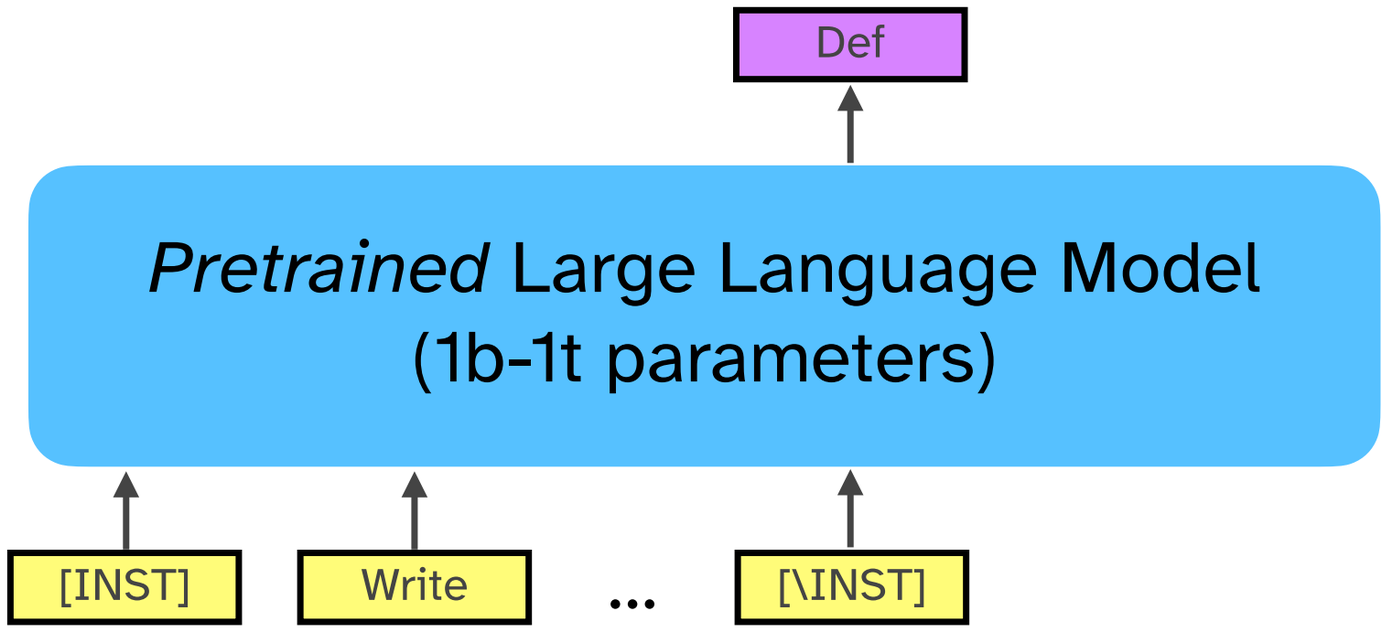
Lots of pre-defined task/ response pairs (~100k)



Natural language inference, Summarization, code generation, Sentiment analysis, …

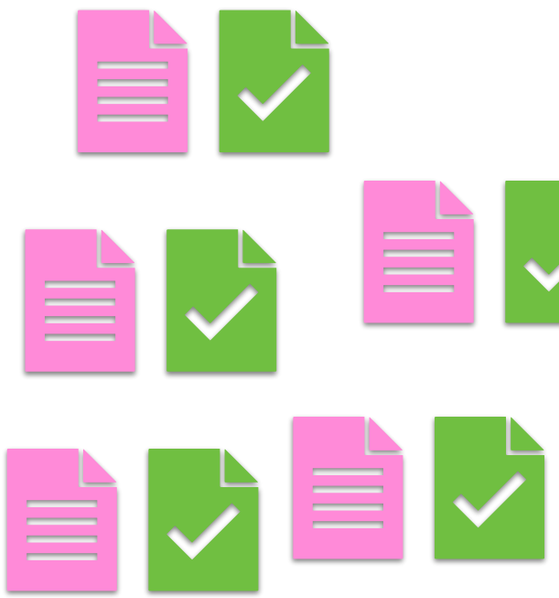
[INST] Write a python function to output ‘Hello World’ [\INST]

```
def hello_world():
  """Prints the classic 'Hello, world!' message."""
  print("Hello, world!")

# Call the function to print the message
hello_world()
```



Instruction tuning

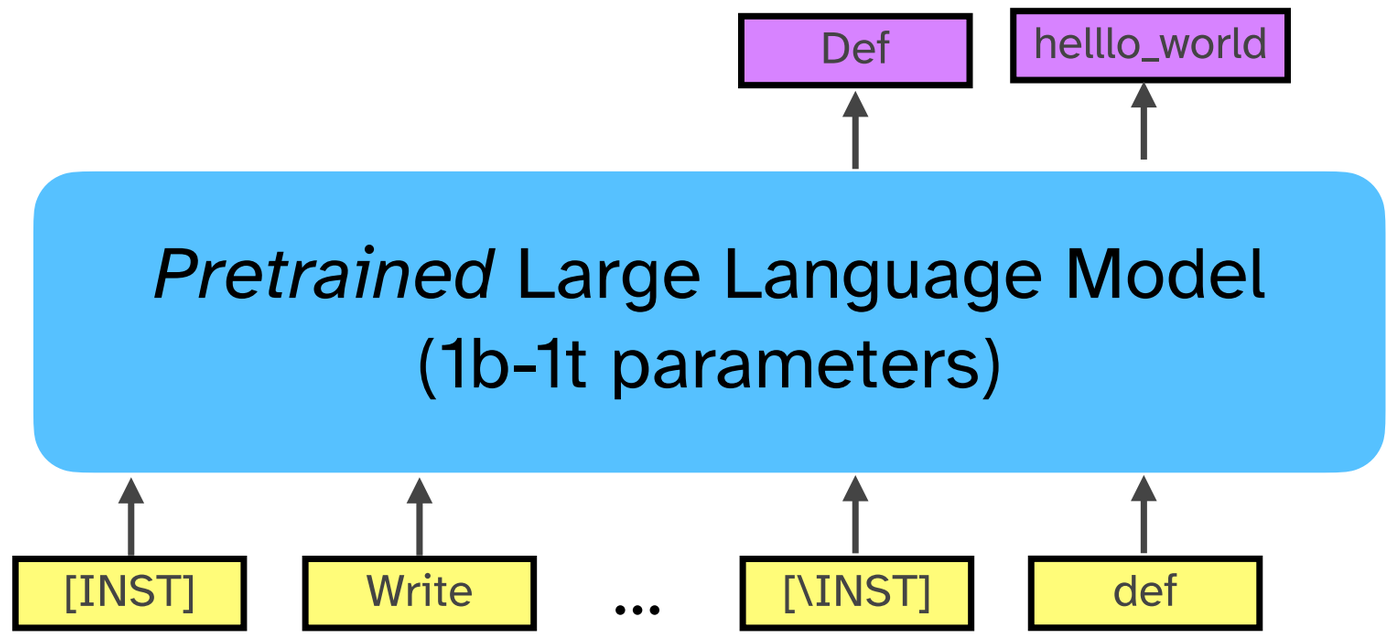
Lots of pre-defined task/ response pairs (~100k)



Natural language inference, Summarization, code generation, Sentiment analysis, …

[INST] Write a python function to output ‘Hello World’ [\INST]

```
def hello_world():
  """Prints the classic 'Hello, world!' message."""
  print("Hello, world!")

# Call the function to print the message
hello_world()
```



We’d also like to **align** with human preferences

“Write a one stanza poem about Texas A&M traditions in the style of Ogden Nash.”

The Aggies march, a sight to see,
With "Gig 'em" yells, so wild and free,
Bonfires blaze, reaching for the sky,
While Muster's call, makes spirits fly,
From Silver Taps, to Midnight Yell,
Texas A&M, traditions swell!

Response 1

In College Station where the Aggies dwell,
Traditions reign, so stories tell,
Of bonfires high, and yells so grand,
A sea of maroon, across the land,
With Muster solemn, memories they hold,
Texas A&M, a story to be told.

Response 2

Can extend to any preferences ...

Which response is funnier?

... safer?

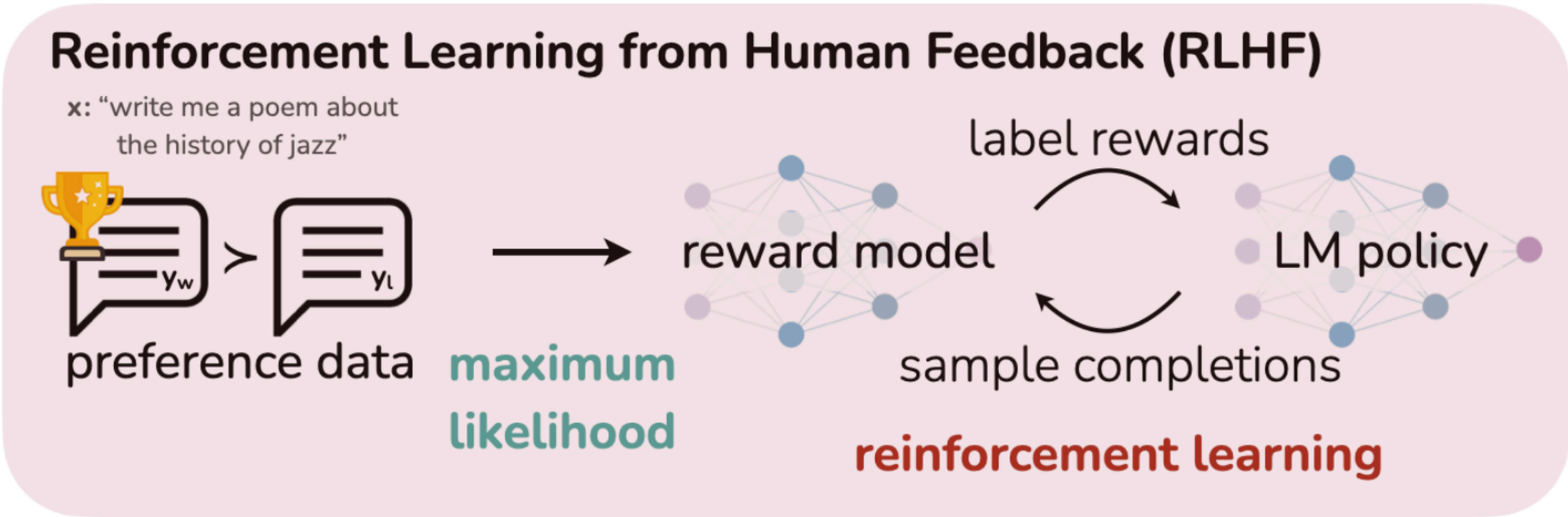
... more concise?

... more engaging?

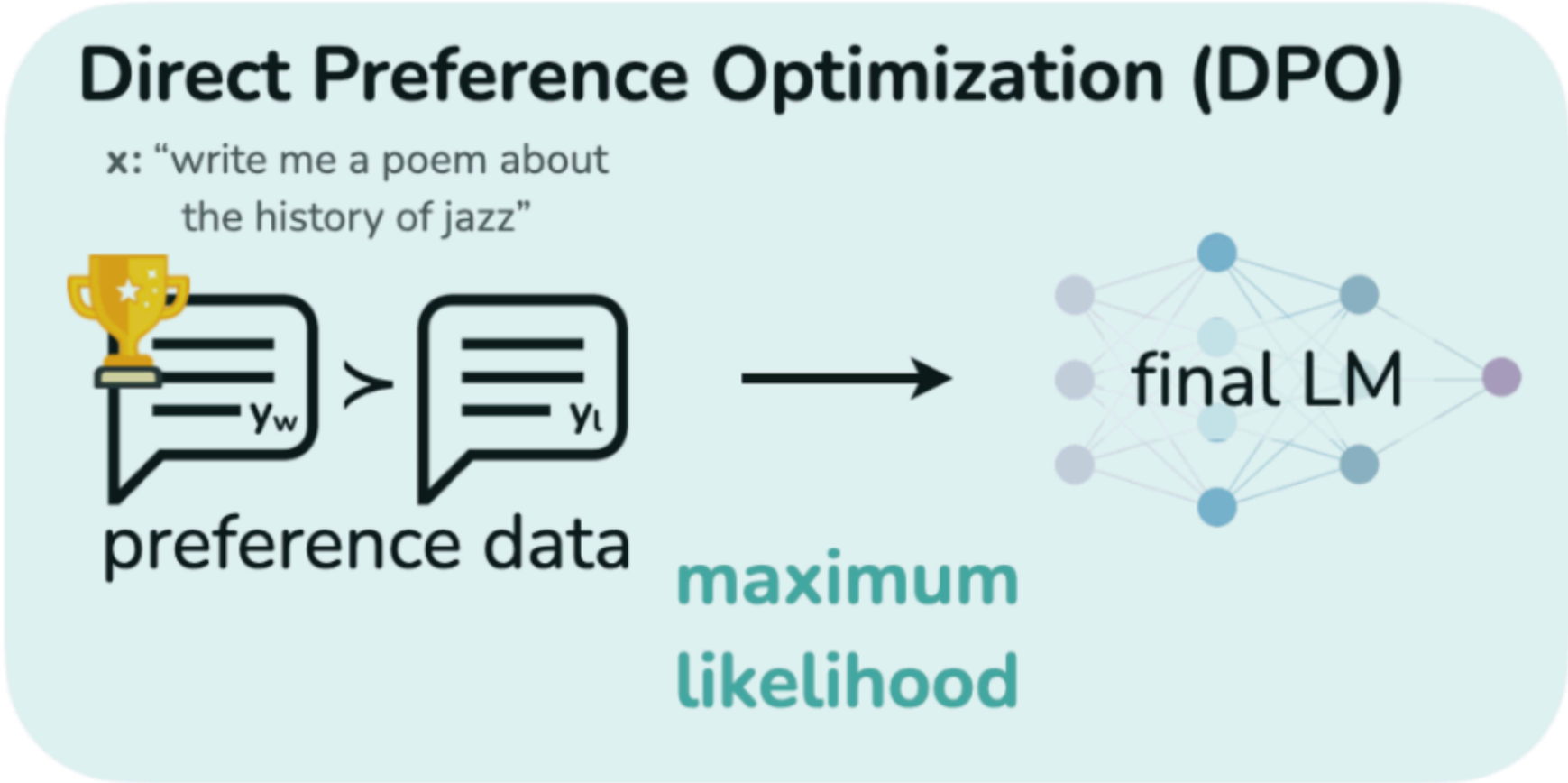
... more trustworthy?

... ?

RLHF (2017)


Reinforcement Learning from Human Feedback (RLHF)
x: "write me a poem about the history of jazz"
y_w
y_l
preference data
maximum likelihood
label rewards
reward model
LM policy
sample completions
reinforcement learning

More recently: Direct Preference Optimization (DPO)



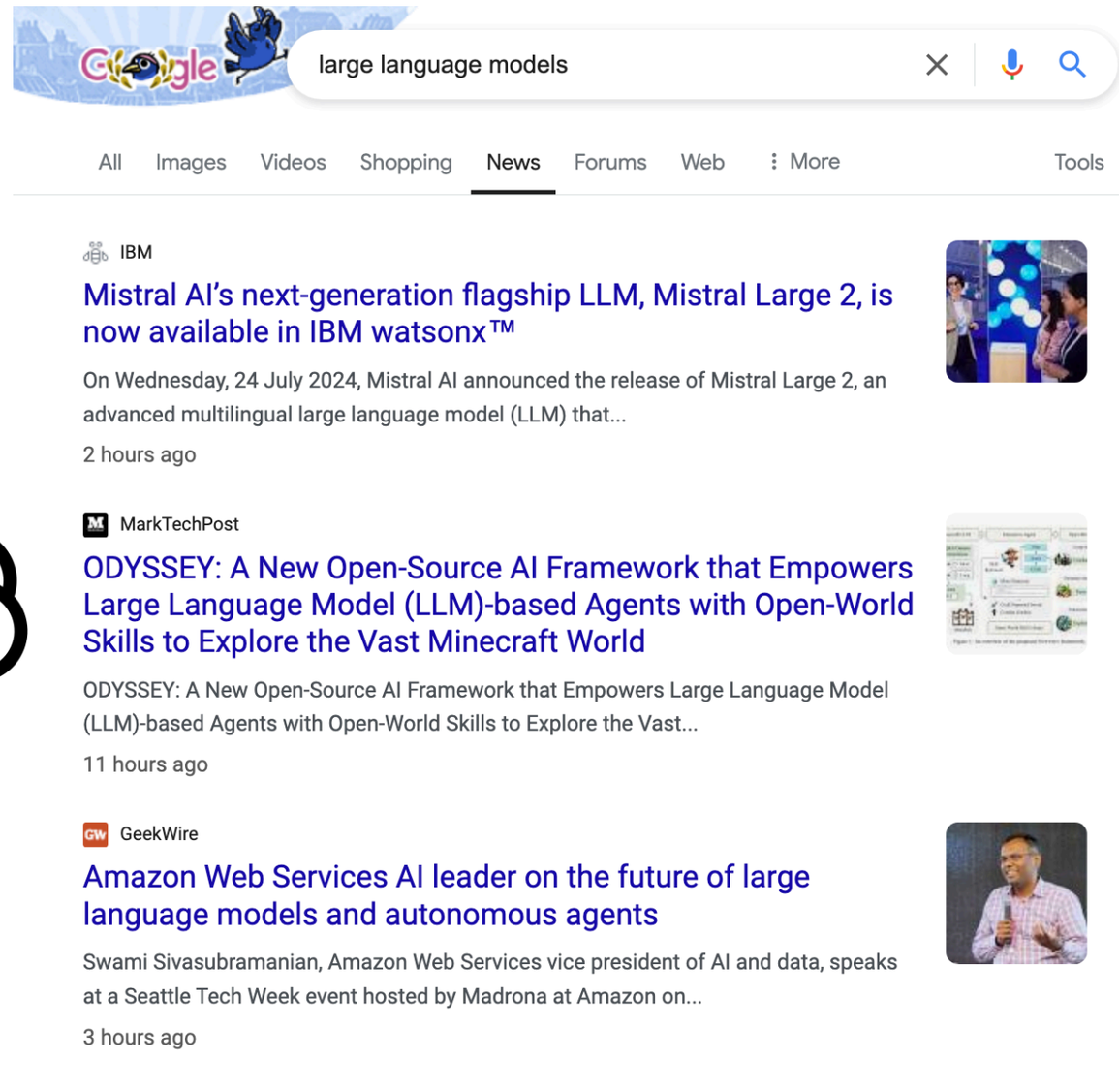
https://arxiv.org/abs/2305.18290

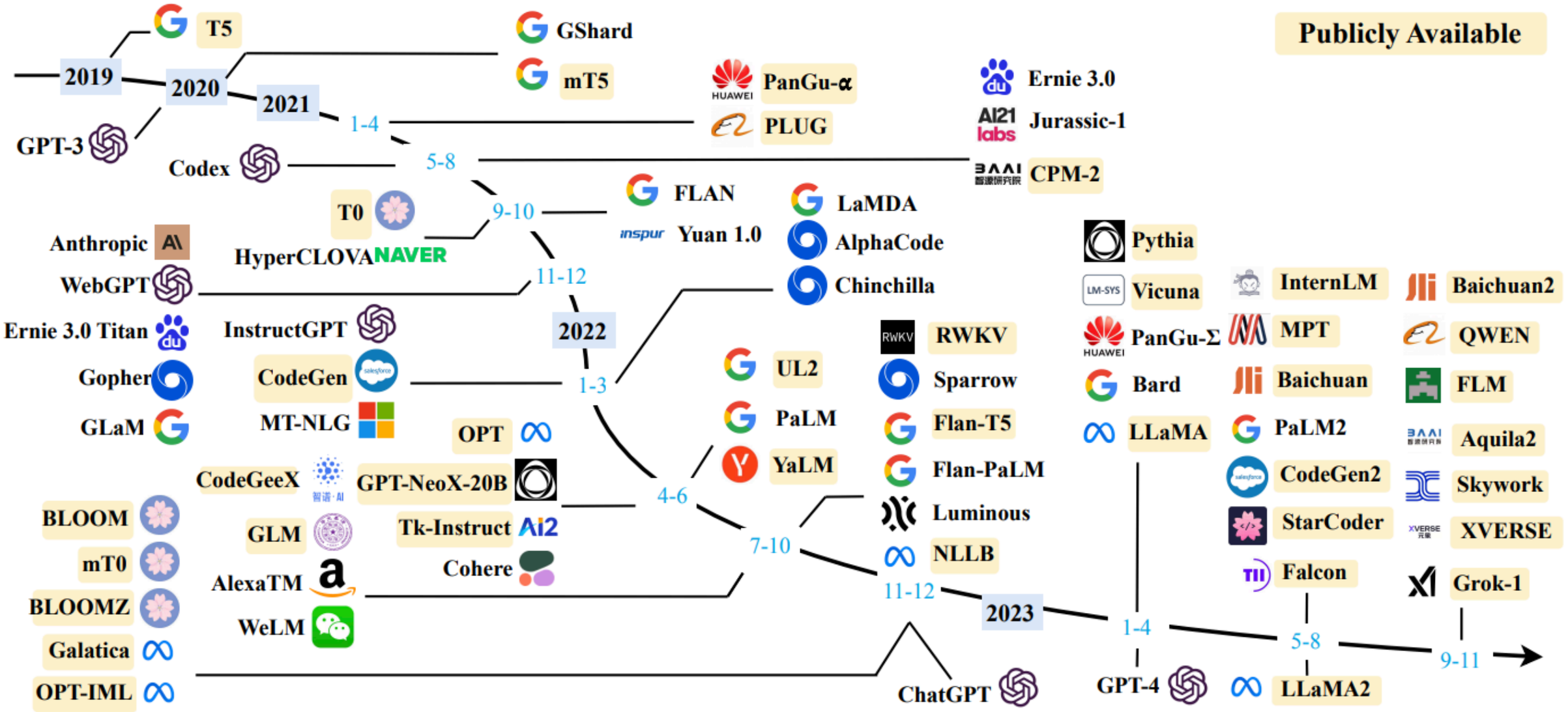
Large Language Models

In the news, with updates coming fast and furious

Launched into popular consciousness with release of ChatGPT in November 2022

Daily (hourly!) news to track




Publicly Available
2019
2020
2021
2022
2023
T5
GPT-3
GShard
mT5
PanGu-α
PLUG
Ernie 3.0
Jurassic-1
CPM-2
Codex
T0
HyperCLOVA
Anthropic
WebGPT
Ernie 3.0 Titan
Gopher
GLaM
FLAN
Yuan 1.0
LaMDA
AlphaCode
Chinchilla
InstructGPT
CodeGen
MT-NLG
OPT
GPT-NeoX-20B
Tk-Instruct
Cohere
CodeGeeX
GLM
AlexaTM
WeLM
BLOOM
mT0
BLOOMZ
Galatica
OPT-IML
UL2
PaLM
YaLM
RWKV
Sparrow
Flan-T5
Flan-PaLM
Luminous
NLLB
ChatGPT
Pythia
Vicuna
PanGu-Σ
Bard
LLaMA
InternLM
MPT
Baichuan
PaLM2
CodeGen2
StarCoder
Falcon
Baichuan2
QWEN
FLM
Aquila2
Skywork
XVERSE
Grok-1
GPT-4
LLaMA2
1-4
5-8
9-10
11-12
1-3
4-6
7-10
11-12
1-4
5-8
9-11

Large Language Models

ChatGPT sparked huge investment by all the major players (e.g., Google, Amazon, Meta, …)

Plus kickstarted a frenzy of venture investments in generative AI startups

Google Calls In Help From Larry Page and Sergey Brin for A.I. Fight

A rival chatbot has shaken Google out of its routine, with the founders who left three years ago re-engaging and more than 20 A.I. projects in the works.

Research

Introducing LLaMA: A foundational, 65-billion-parameter large language model

February 24, 2023

Large Language Models
showcase exciting new capabilities ...

Activity Time!

Activity 1

Part 0: Pair up

Introduce yourself to your partner

Share an interesting fact

Part 1: Work together to identify an interesting capability of LLMs