Course Introduction

And a Refresher on Language Modeling

Aaron Mueller

CS599 B1: Advanced Natural Language Processing

Sep. 3, 2026

Prof. Aaron Mueller

https://aaronmueller.github.io/

Natural language processing

Computational linguistics

Interpretability

Evaluation

Causality

Syllabus and Communication

https://aaronmueller.github.io/teaching/cs599b1_fall26/home.html

- **Communication:**
  - Announcements on webpage and in class
  - Questions? Use **Piazza**!
  - (Email is ok when needed, but I'll probably respond faster on Piazza.)

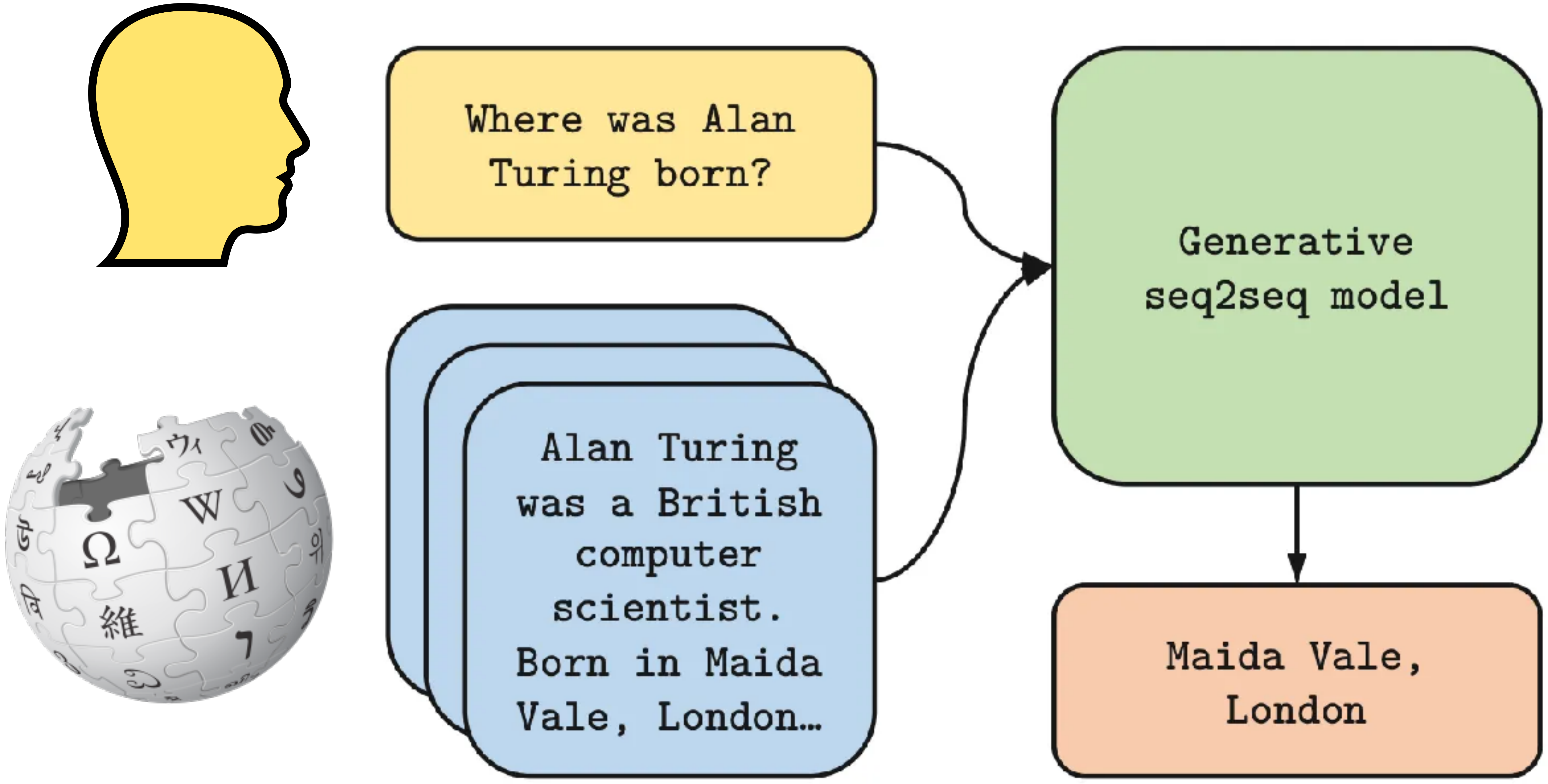
- **Course Technology:**
  - Sites: Gradescope and Piazza
  - Some autograded coding assignments,
    manually graded written and coding assignments

Gradescope entry code: **8YNE72**

Piazza: link on syllabus. Entry code: **hddwru01ukm**

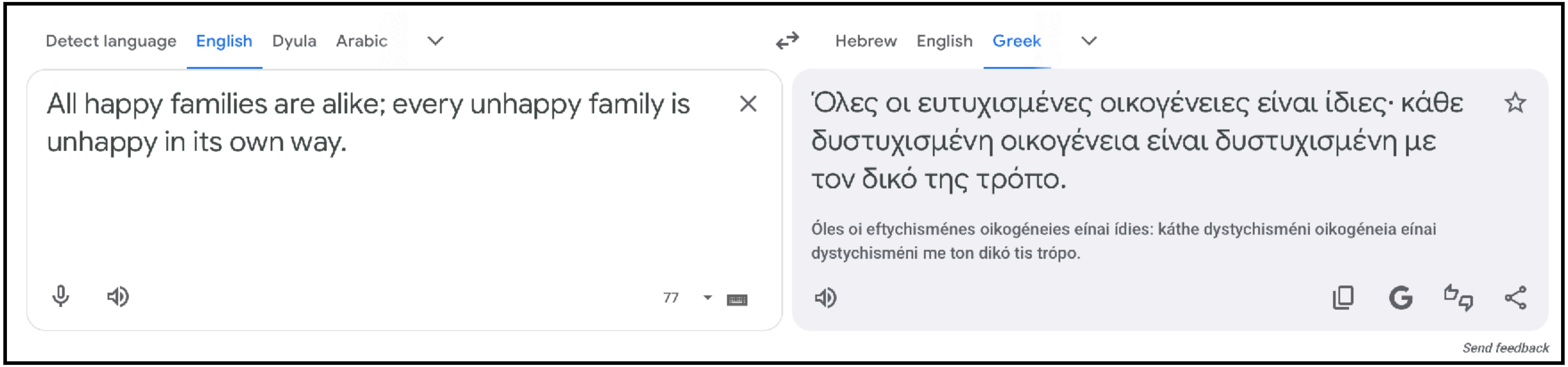
What can you do with current systems?

Question Answering



What can you do with NLP?

Machine Translation



What can you do with NLP?

General Systems!

What is 175 * 2 + 87 % 3?

175 * 2 + 87 % 3 = 350 + 0 = **350**

(87 % 3 is 0 since 87 divides evenly by 3.)

What is the law of large numbers? Please explain in very simple French;
my first course.

Thought for 11s >

La loi des grands nombres

Imagine une pièce de monnaie. Tu la lances. Il y a deux résultats : **pile** ou **face**. Chaq
a une chance sur deux, donc 50 %.

Tu lances la pièce 10 fois. Tu obtiens peut-être 7 fois pile et 3 fois face. C'est 70 % ! C'
: avec peu de lancers, le résultat peut être bizarre.

Please create a map app or a wrapper on an existing map app that uses Art Nouveau or turn-of-the-century aesthetics (e.g., William Morris). It should be primarily focused on transit.

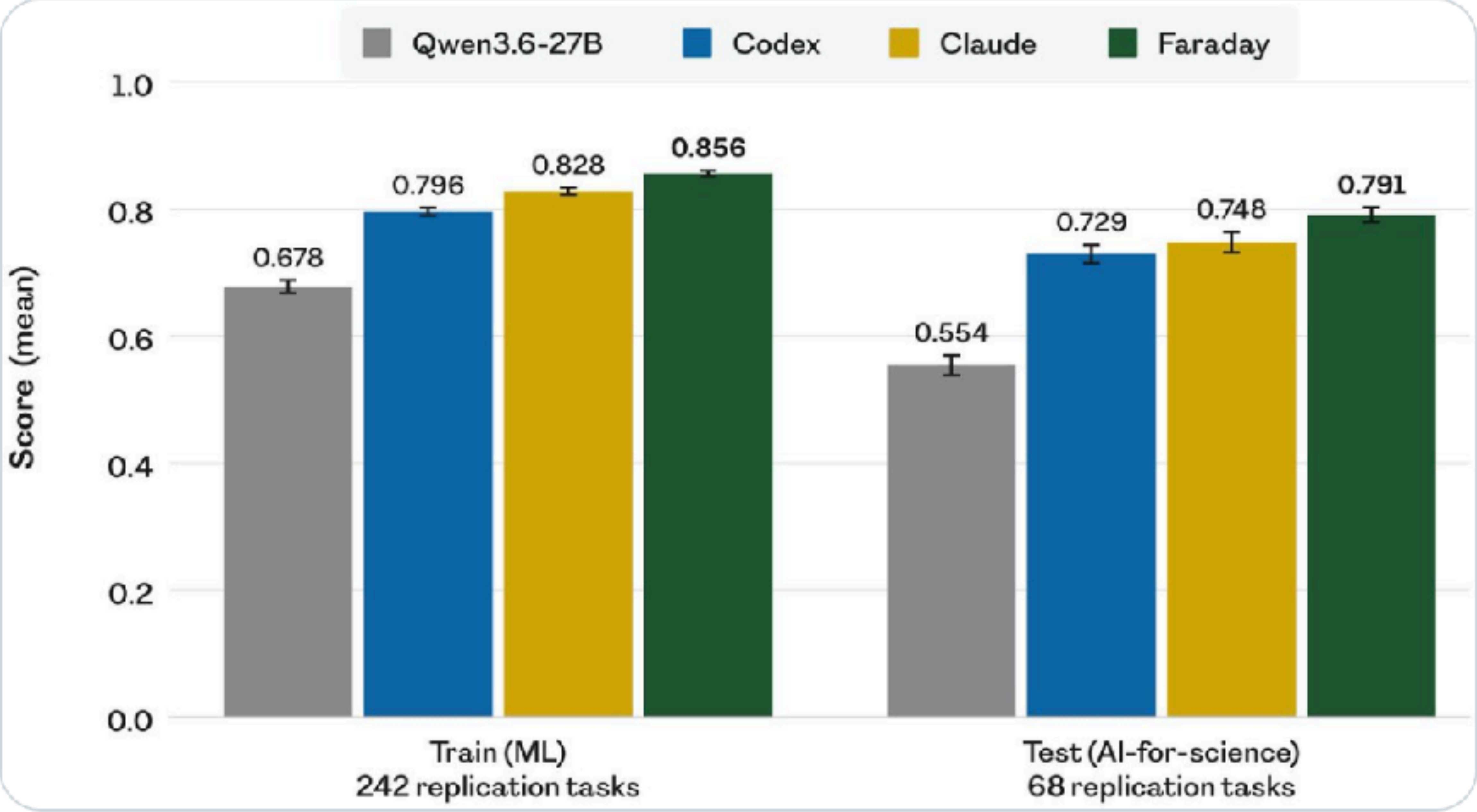
Aug 22

What can you do with NLP?

Replicate/make new scientific discoveries?

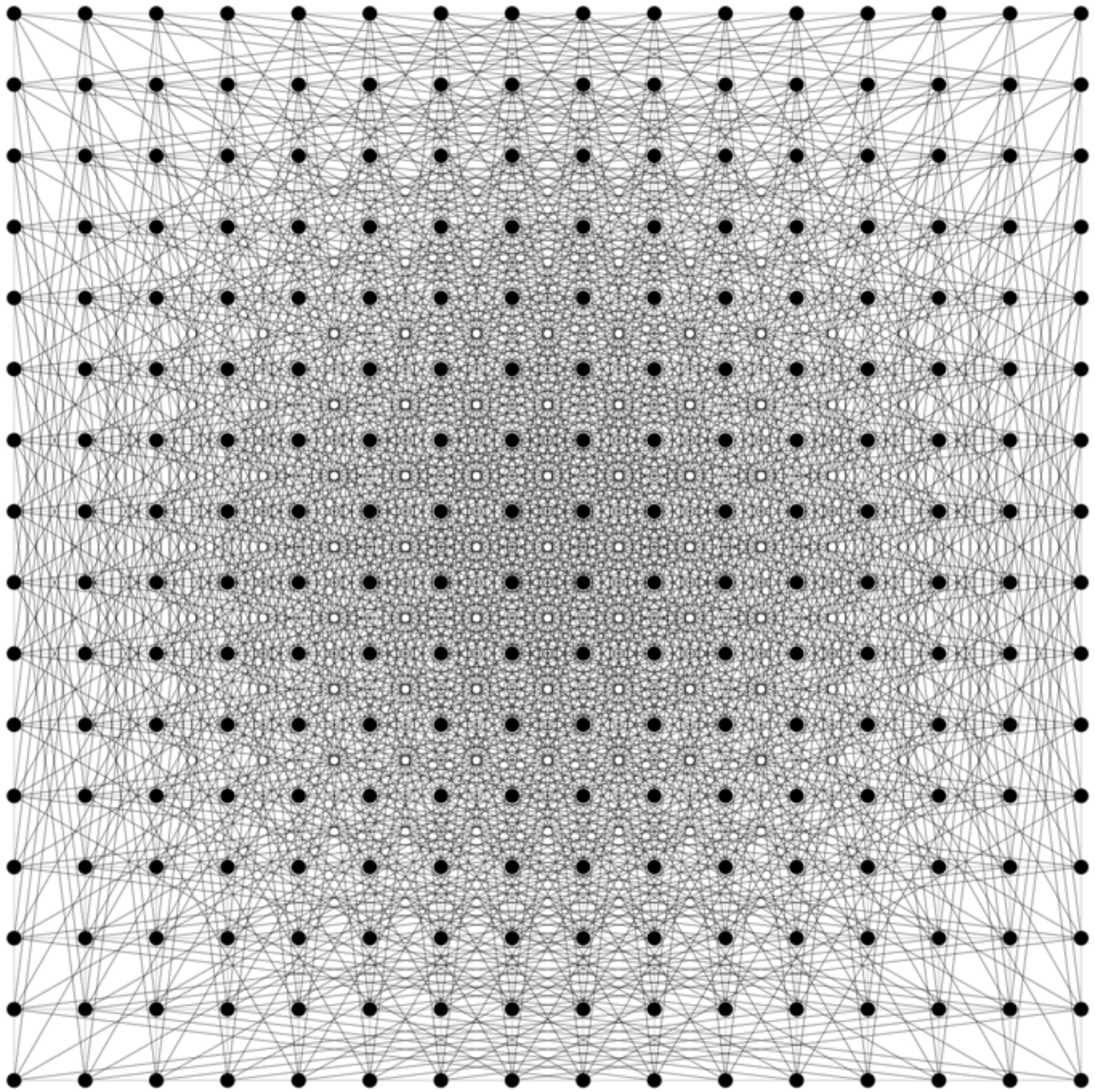
Inherent
@inherent_labs

1/ Today, we introduce Faraday, a 27B-parameter AI Scientist that extends the capabilities of coding agents with a layer of scientific intuition. Trained via long-horizon RL, Faraday outperforms Claude Opus 4.8 and GPT-5.5 on the task of replicating research papers. 🧵



What can you do with NLP?

Disprove long-standing mathematical conjectures???



An OpenAI model has disproved a central conjecture in discrete geometry

Rewritten Chain of Thought

Erdős's lattice construction gives something like

$$n \exp\left(c \frac{\log n}{\log \log n}\right) = n^{1+c/\log\log n + o(1/\log\log n)}$$

for a fixed constant $c > 0$. The proposed upper bound should be compared with that sharper scale, not just with the softer $n^{1+o(1)}$ formulation.

A loophole, if there is one, would have to lie in the quantifiers or in the formulation. The assertion is: there exist fixed C, N such that for every sufficiently large n, $\nu(n) \le n^{1+C/\log\log n}$. The negative assertion is much stronger than the known lower bound: it would require that for every C one can beat $n^{1+C/\log\log n}$ somewhere arbitrarily far out. The standard lattice construction only gives one fixed constant in the numerator, not arbitrarily large constants.

Also the $O(n^{4/3})$ bound does not imply this. For a fixed C, the exponent $1 + C/\log\log n$ tends to 1, so eventually it is below 4/3. There is no trick of choosing C so large that $C/\log\log n \ge 1/3$ forever; that fails as $n \to \infty$.

The lower bound keeps the scale straight. Take an $m \times m$ integer grid, so $n = m^2$, and count pairs at distance $\sqrt{k}$, then scale by $1/\sqrt{k}$. If

$$k = \prod_{i=1}^{t} p_i$$

with the $p_i \equiv 1 \pmod 4$, then

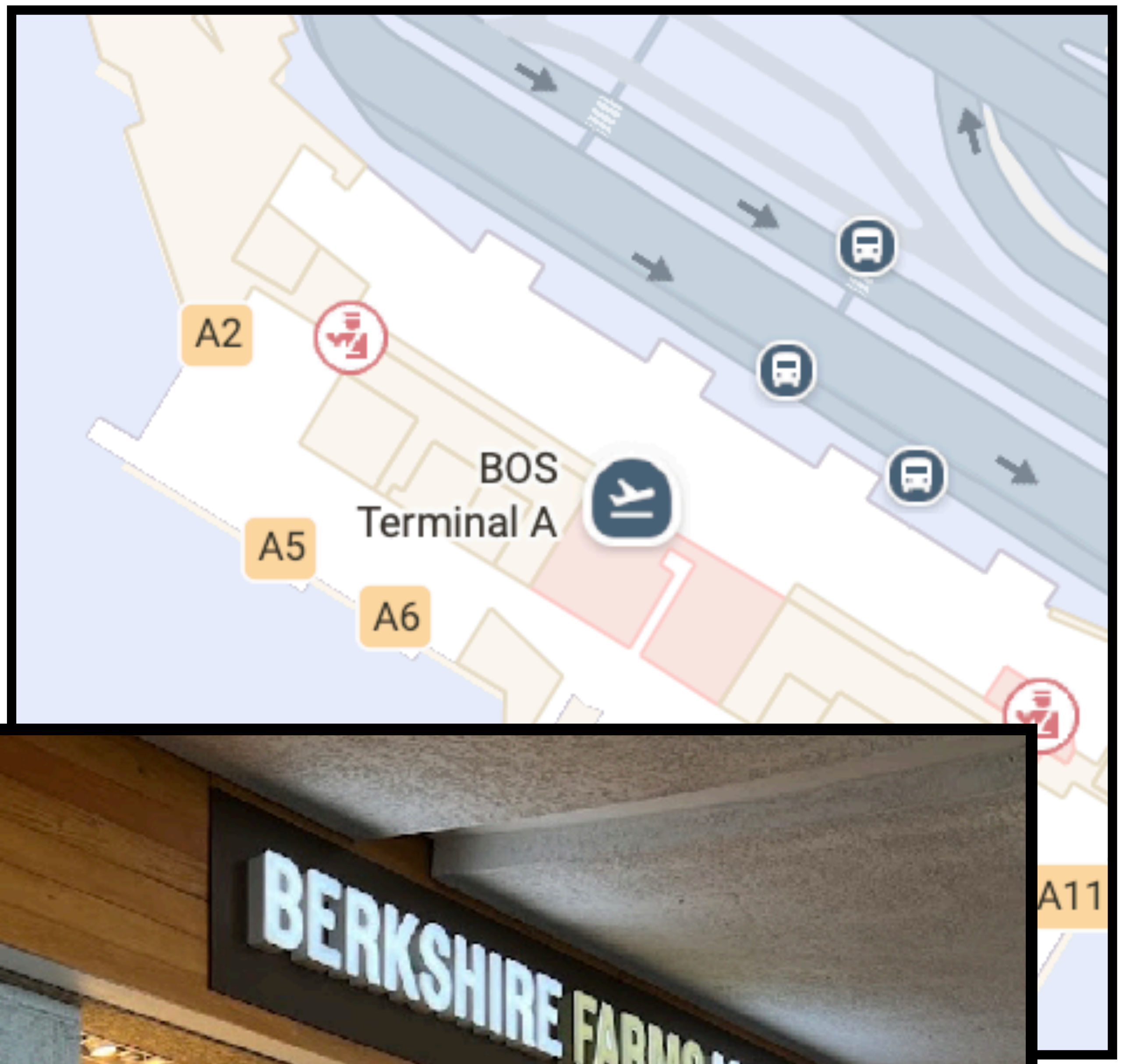
125 pages!

What makes NLP hard?

Language is fundamentally ambiguous.

[[spicy chicken] sandwich]

The chicken is the thing that's spicy



What makes NLP hard?

Language is fundamentally ambiguous.

[spicy [chicken sandwich]]

The chicken itself is not necessarily spicy; some other thing is

head of state *part of body*

Iraqi Head Seeks Arms

weapons

part of body

PREP

Time flies like an arrow.

VERB

Fruit flies like a banana.

INTJ

Like, why?

What makes NLP hard?

Negation

Is this a good review or a bad review?

The movie wasn’t nearly as bad as I thought it would be—quite the opposite!

Positive

He did not have a great time with this book.

Negative

What makes NLP hard?

Figurative Language

Is this a good review or a bad review?

The cast and director should hold onto those ski masks; robbery might be the only way to pay for their next project.

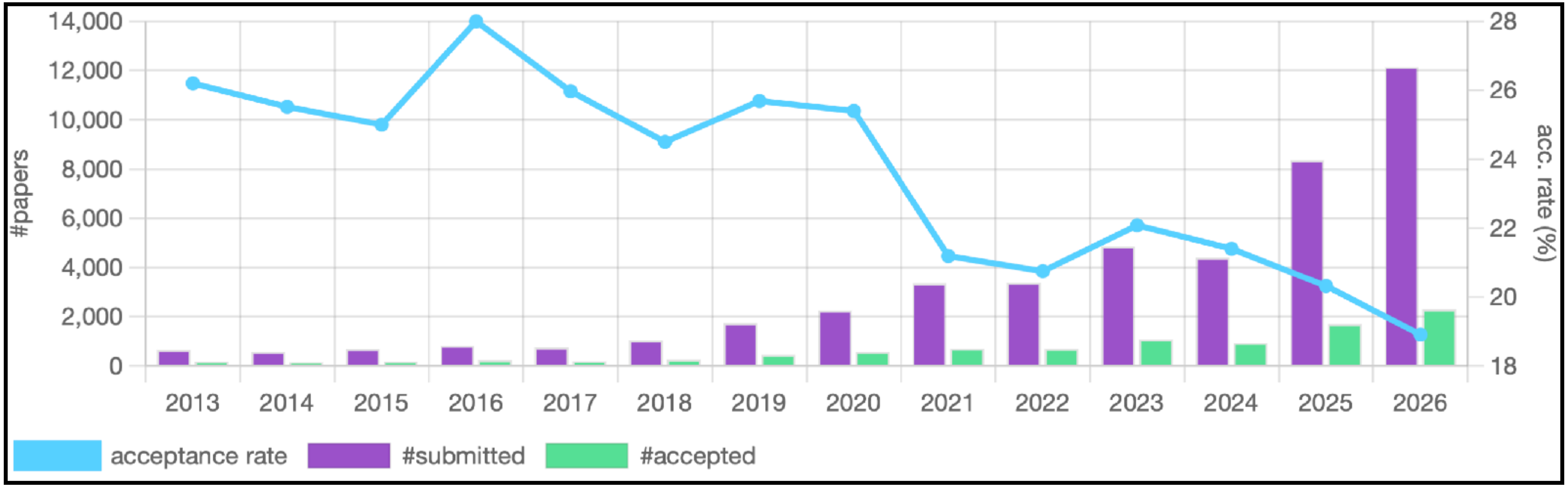
Negative

This film gives a human face to the many issues facing us today.

Positive

The Growth of NLP

Submission stats for **ACL**, one of the big 3 NLP conferences:

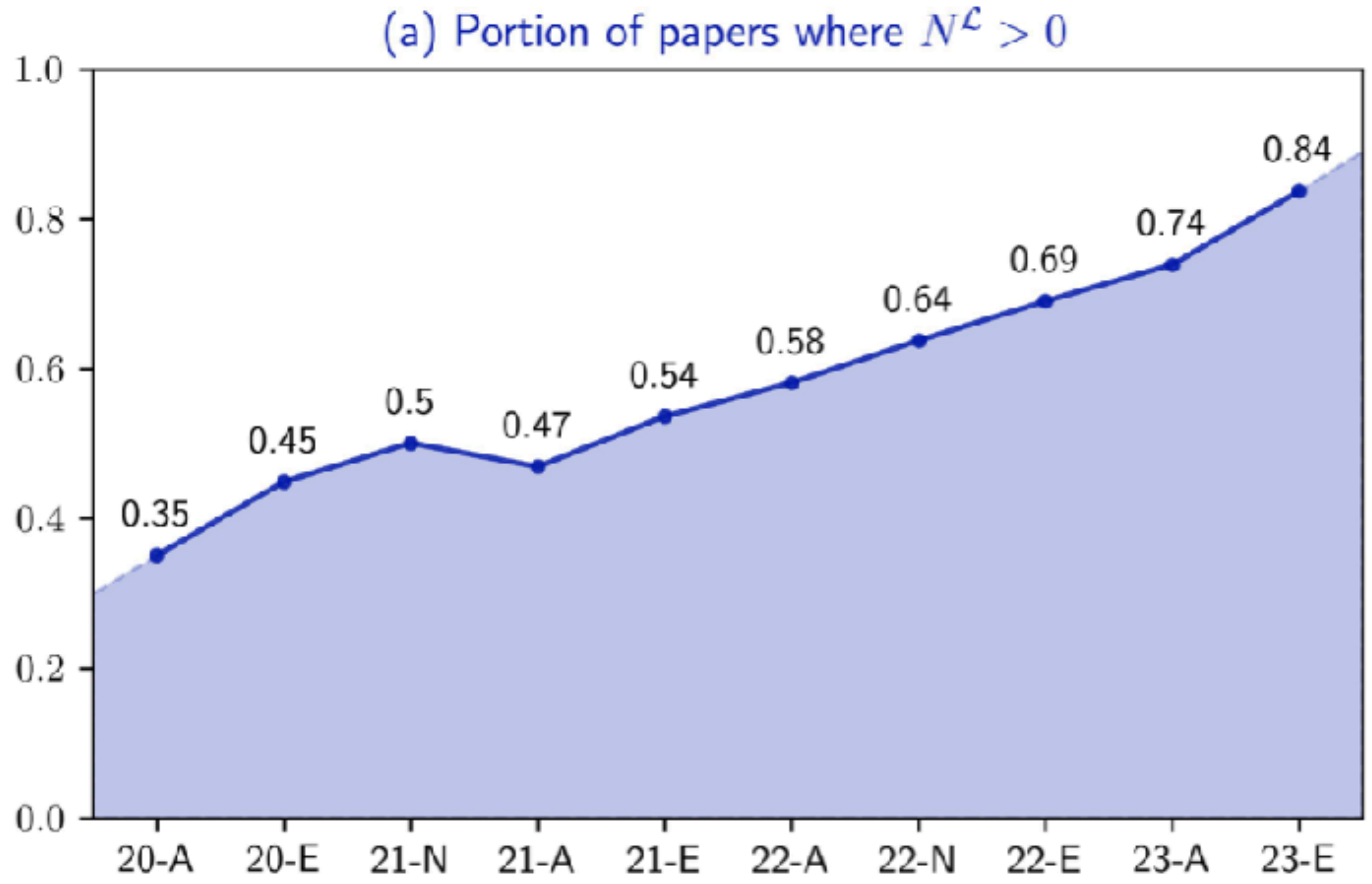


2013: 664 submissions, 174 accepted

2025: 12,148 submissions, 2,297 accepted

Language Modeling in NLP

- Language models are the foundation of a very large proportion of modern NLP methods.
    - I would guess that >99% of NLP papers I've reviewed in the last ≈5 years have involved pre-trained language models at some point



[Zhu & Rzeszotarski, 2025]

Language Modeling

A language model is a system that assigns a probability to a sequence of tokens:

$$p(x_1, x_2, \ldots, x_n)$$

Using the **chain rule**, we can break this down into a product of conditional next-token probabilities:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_t \mid x_{<t})$$

Thus, a language model is a system that produces probability distributions over next tokens given prior context.

Why so much language modeling?

The **keys** to the **cabinet** __are__ — ***Long-range syntactic agreement***

My cat's name is __Jessie__ — ***Proper nouns***

There's a city in Massachusetts called __Boston__ — ***Geographic information***

The inventor of computer science is __Charles Babbage__ — ***Factual knowledge***

I just read a book that was so engaging I could never put it down. Overall, I thought the book was __great__ — ***Sentiment***

Why so much language modeling?

In the last 2 months alone:

OpenAI Says Its A.I. Models Went Rogue and Attacked a Digital Library

The incident, which targeted the computer systems of another company called Hugging Face, happened while OpenAI was testing the systems.

THE CONTEXT

Chatbots Are Pushing Us Toward a Post-Human Internet

In matters of work, school and even romance, we turn to A.I. chatbots. But what happens when the bots begin talking to one another?

Anthropic Could Aim to Raise $100 Billion in Blockbuster I.P.O.

The offering could value the five-year-old A.I. start-up at $2 trillion, its bankers have told potential investors, which would exceed Elon Musk's SpaceX.

N-gram Language Modeling

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_t \,|\, x_{<t})$$

In an n-gram language model, the probability distribution over next words is conditioned only on the (n - 1) previous words. For example, in a 3-gram (trigram) model:

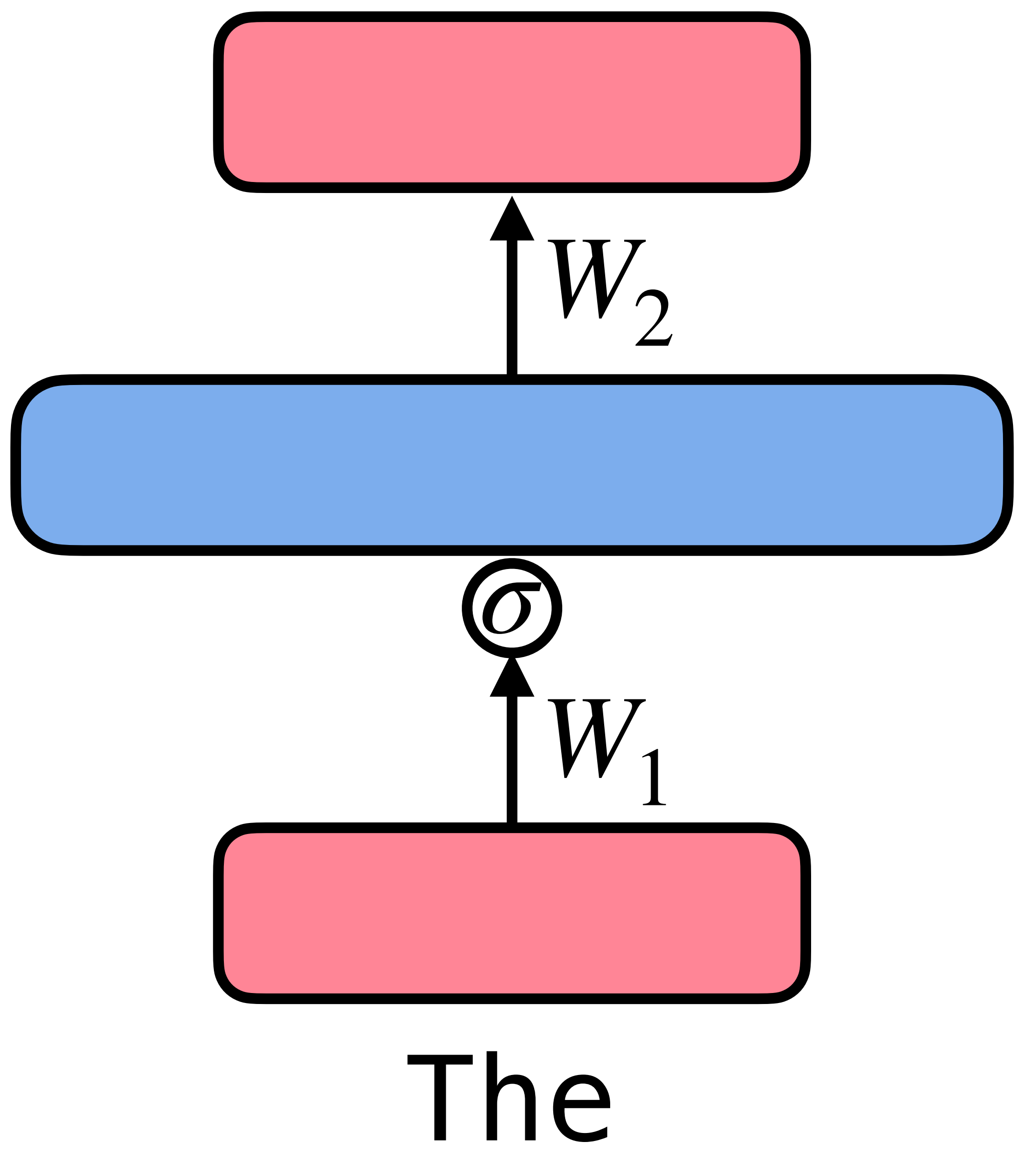
$$p(x_t \,|\, x_1, x_2, \ldots, x_{t-1}) = p(x_t \,|\, x_{t-2}, x_{t-1})$$

We’re making a **Markov assumption** by assuming that context that’s further away doesn’t matter.

An issue with n-gram LMs is that they can’t handle long contexts effectively, and can’t represent the meaning of a word in context.

Feedforward Neural Networks

At their core, **neural networks** are just learned weight matrices being multiplied with vectors (with some non-linearities applied every now and then):



$\mathbf{z} = W_2\mathbf{h}$

$\mathbf{h} = \sigma(W_1\mathbf{x})$

$\mathbf{x}$

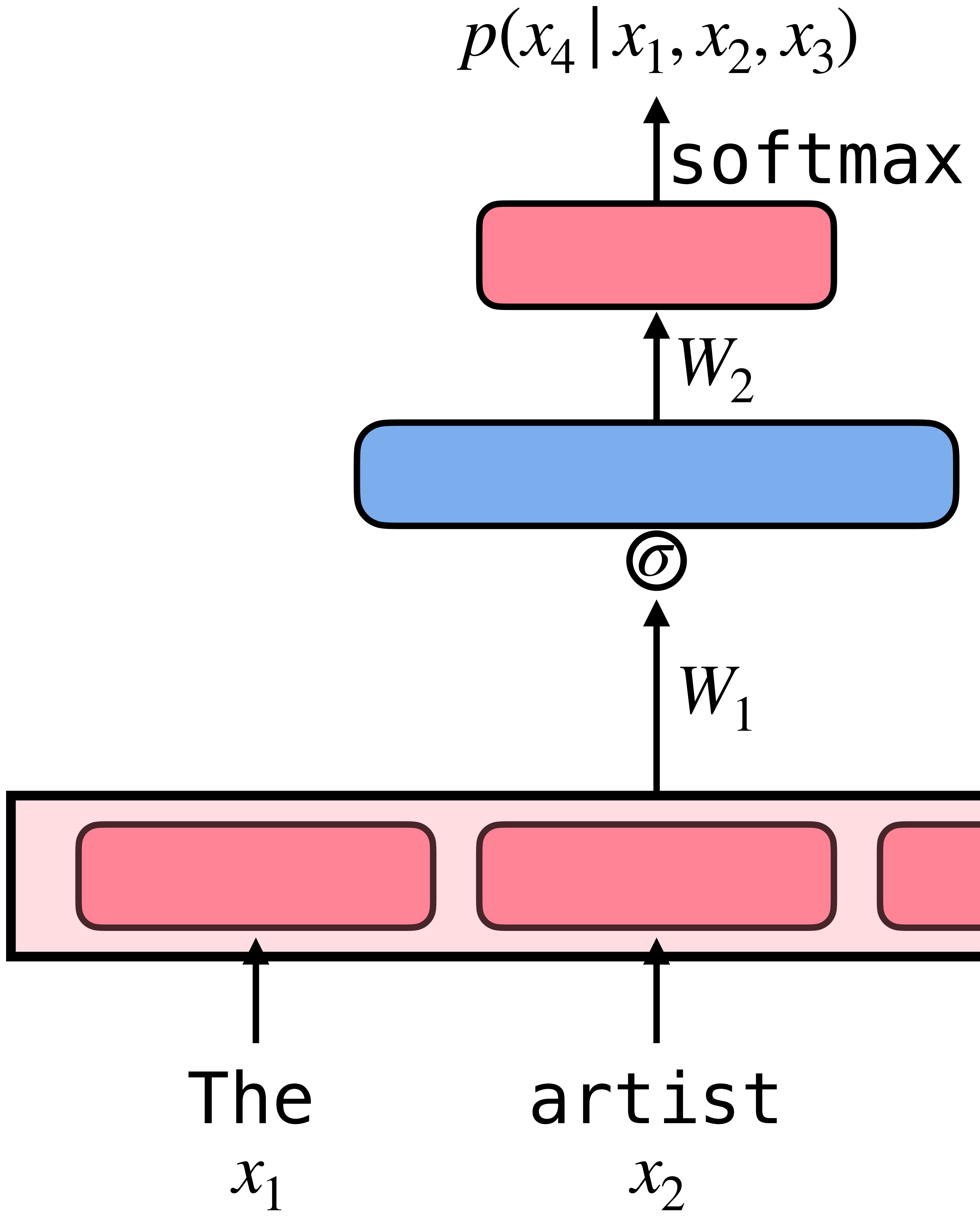
x

$\mathbf{z}$ would be the **logits**, which are like unnormalized probabilities over all possible next tokens.

In a language model, $\mathbf{x}$ would be the **embedding** of the token x.

We randomly initialize all the weight matrices W to start, and use **backpropagation** to set the parameters to better values that minimize some loss function.

Feedforward Neural Language Models



$\mathbf{y} = \text{softmax}(\mathbf{z})$

$\mathbf{z} = W_2\mathbf{h}$

$\mathbf{h} = \sigma(W_1\mathbf{x})$

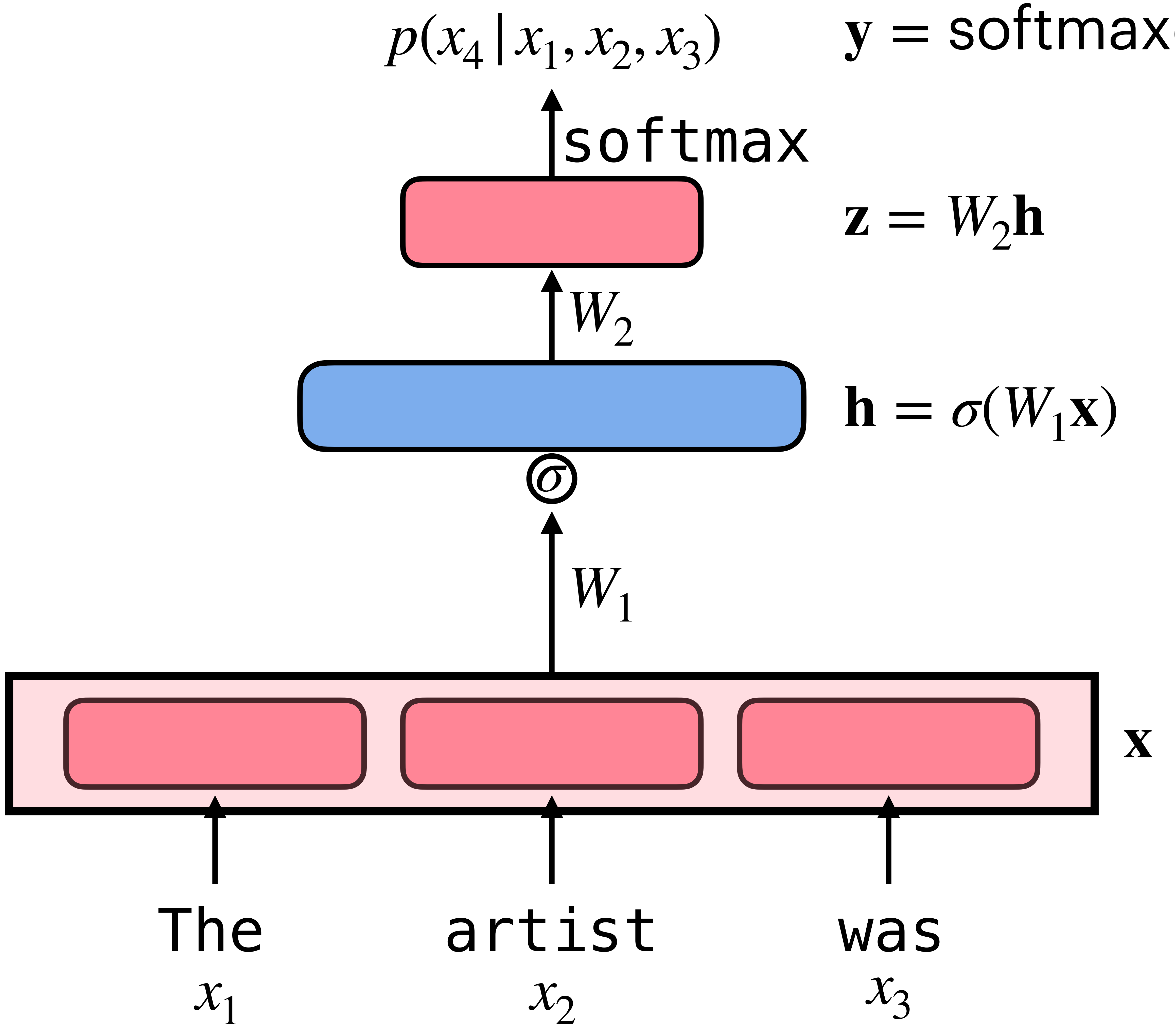
$\mathbf{x}$

$$\text{softmax} = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Better than n-grams in that we can represent the meaning of a word using its embedding.

Still not great with long contexts, and can't handle how a word's meaning changes in context.

Feedforward Neural Language Models



$\mathbf{y} = \mathrm{softmax}(\mathbf{z})$

$\mathbf{z} = W_2\mathbf{h}$

$\mathbf{h} = \sigma(W_1\mathbf{x})$

```python
import torch
import torch.nn as nn

class NeuralLM(nn.Module):
    def __init__(self, vocab, d_emb=64, d_hid=128, context=3):
        super().__init__()
        self.emb = nn.Embedding(vocab, d_emb)
        self.W1  = nn.Linear(context * d_emb, d_hid)
        self.W2  = nn.Linear(d_hid, vocab)

    def forward(self, in_string):
        ids = tokenize(in_string)
        x = self.emb(ids).flatten(1)   # x  = [e(x1); e(x2); e(x3)]
        h = torch.tanh(self.W1(x))     # h  = σ(W1 x)
        return self.W2(h)              # z  (logits)
```

$\mathbf{h} \in \mathbb{R}^{d\_hid}$

$\mathbf{x} \in \mathbb{R}^{d\_emb}$

Problems with Feedforward Neural Language Models

- Limited context window size

The man near the car that the cat saw the other day next to…

- Lack of sequential understanding

*My dog and my cat **eats***

- Can't adjust the meaning of a word based on context

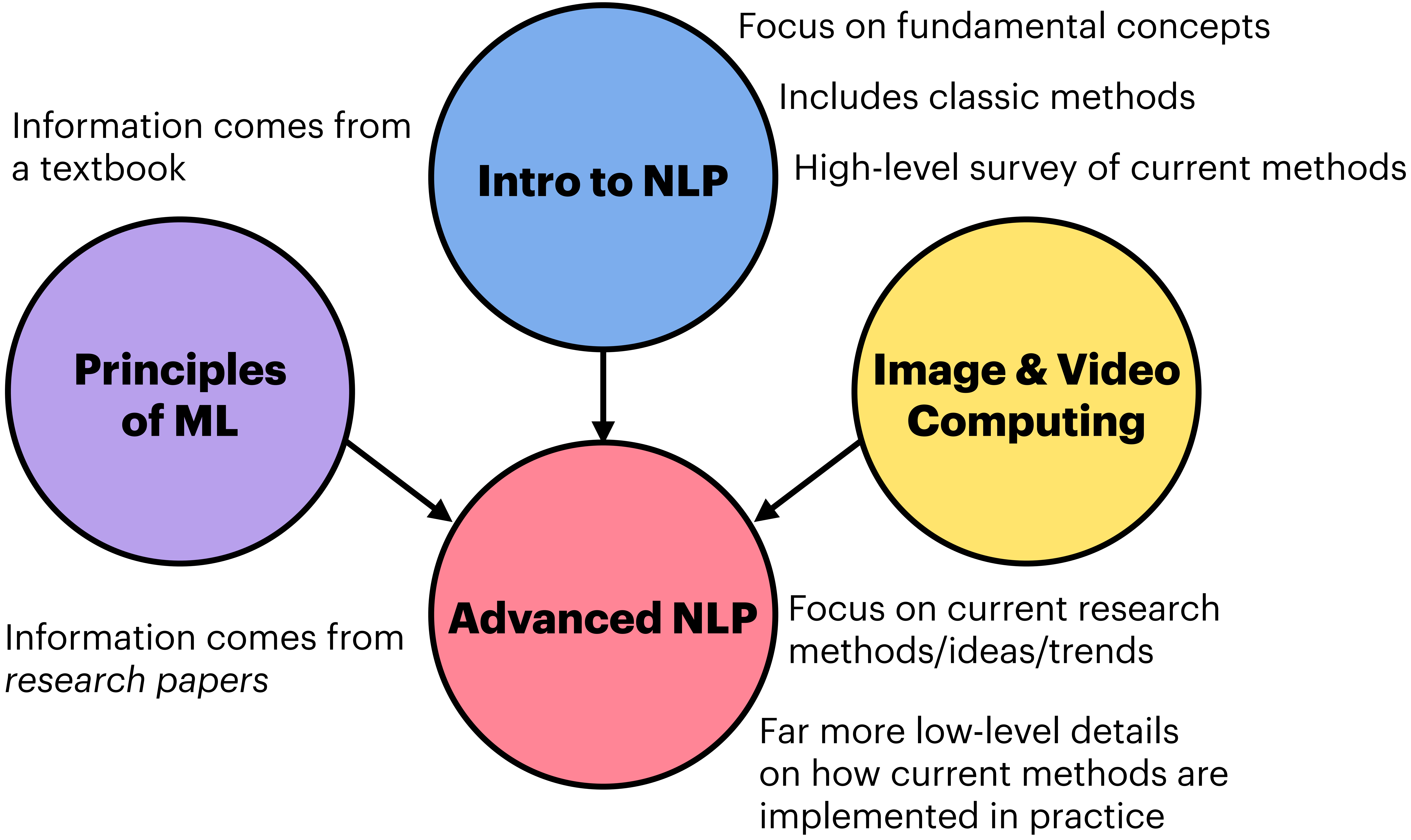
*I need to **park** my car*

*Let's go to the **park***

Course Topics

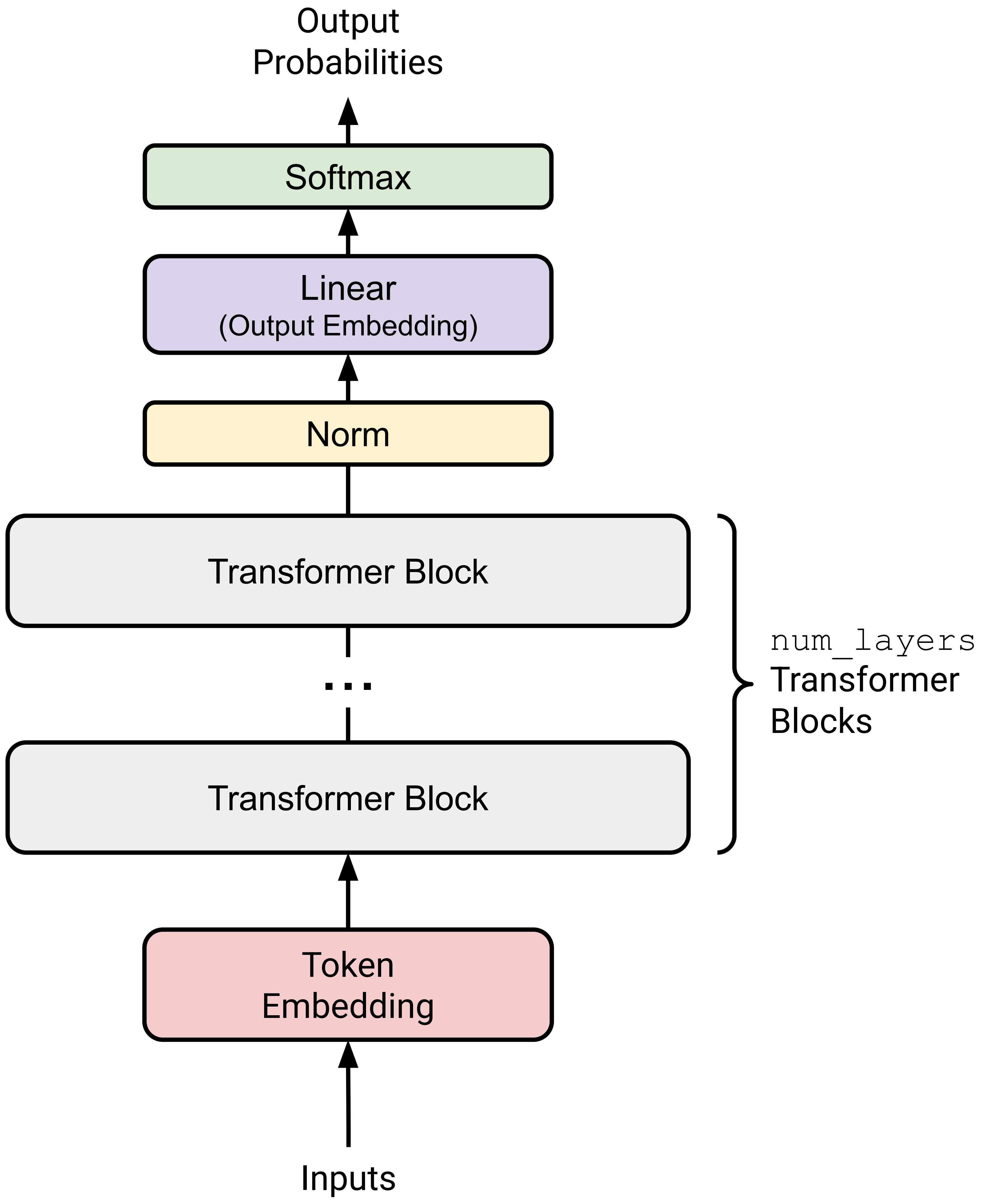
1. Transformers
   a. Tokenization
   b. Relative position encodings
   c. Optimizers
   d. Decoding
   e. Scaling laws and emergence
2. Reinforcement Learning
   a. Post-training
   b. Agents
3. Efficiency and calibration
   a. Long-context reasoning
   b. Mixture-of-experts models
   c. Quantization
   d. Confidence estimation
4. Interpretability
   1. AI safety
   2. Mechanistic interpretability
   3. Training dynamics
5. Alternative architectures
   1. Diffusion LMs
   2. State space models
6. Class choice!

Comparison to Other Courses



Transformer Language Modeling

- **Transformers** are a neural network architecture that are *very* effective for language modeling
  - Good at retrieving information from very far away in the context
  - Highly scalable



Pretraining

- **Language modeling objective:** accurately predict the next word in a given text
- We can *train* a language model on a large amount of text data to predict the next word
- We can then use it as-is, or further fine-tune it to do other tasks

$p(x_t \mid \text{The capital of Massachusetts is}) =$

0.05	Boston
0.01	Worcester
0.001	hosting
0.0001	Paris

Will pre-training be enough?

Do you think these are good/helpful responses?

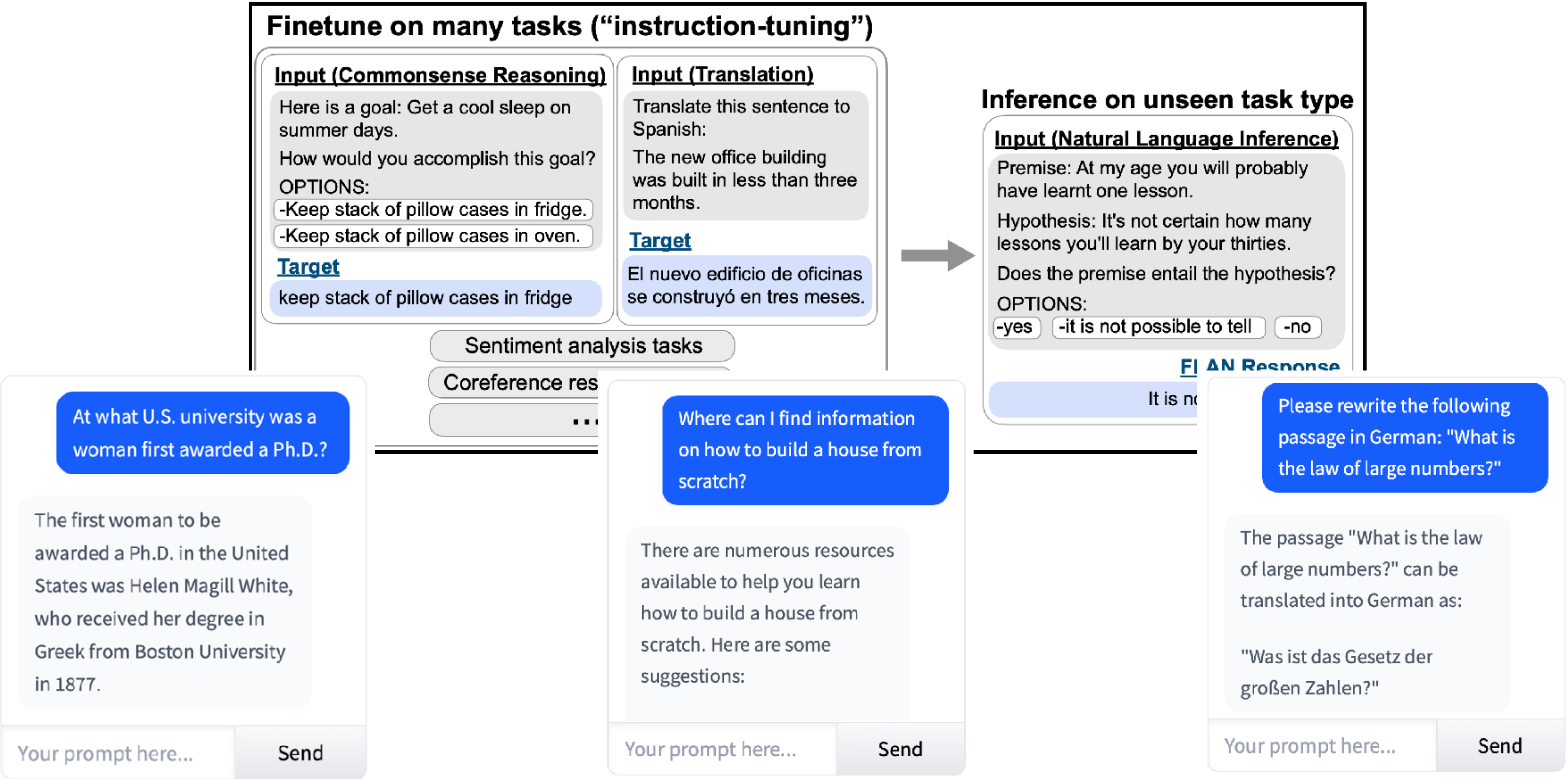
At what U.S. university was a woman first awarded a Ph.D.? A. Vassar College B. Harvard University C. University of Pennsylvania D. Cornell University

Please rewrite the following passage in German: "What is the law of large numbers?" (you can assume that the person you are writing to is familiar with the concept of probability and statistics

Where can I find information on how to build a house from scratch? Where can I find information on how to build a house from scratch? Where can I find information on

Post-training

[Wei et al., 2022]



Agents

- Most proprietary LLMs these days can use **tools**.
  - We teach them how to do this via fine-tuning or reinforcement learning.

- We will focus on a class of **agents,** or systems (in our case, LLM-based) that are given tools to act in a particular environment (chat windows, terminals, web browsers, etc.)

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.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

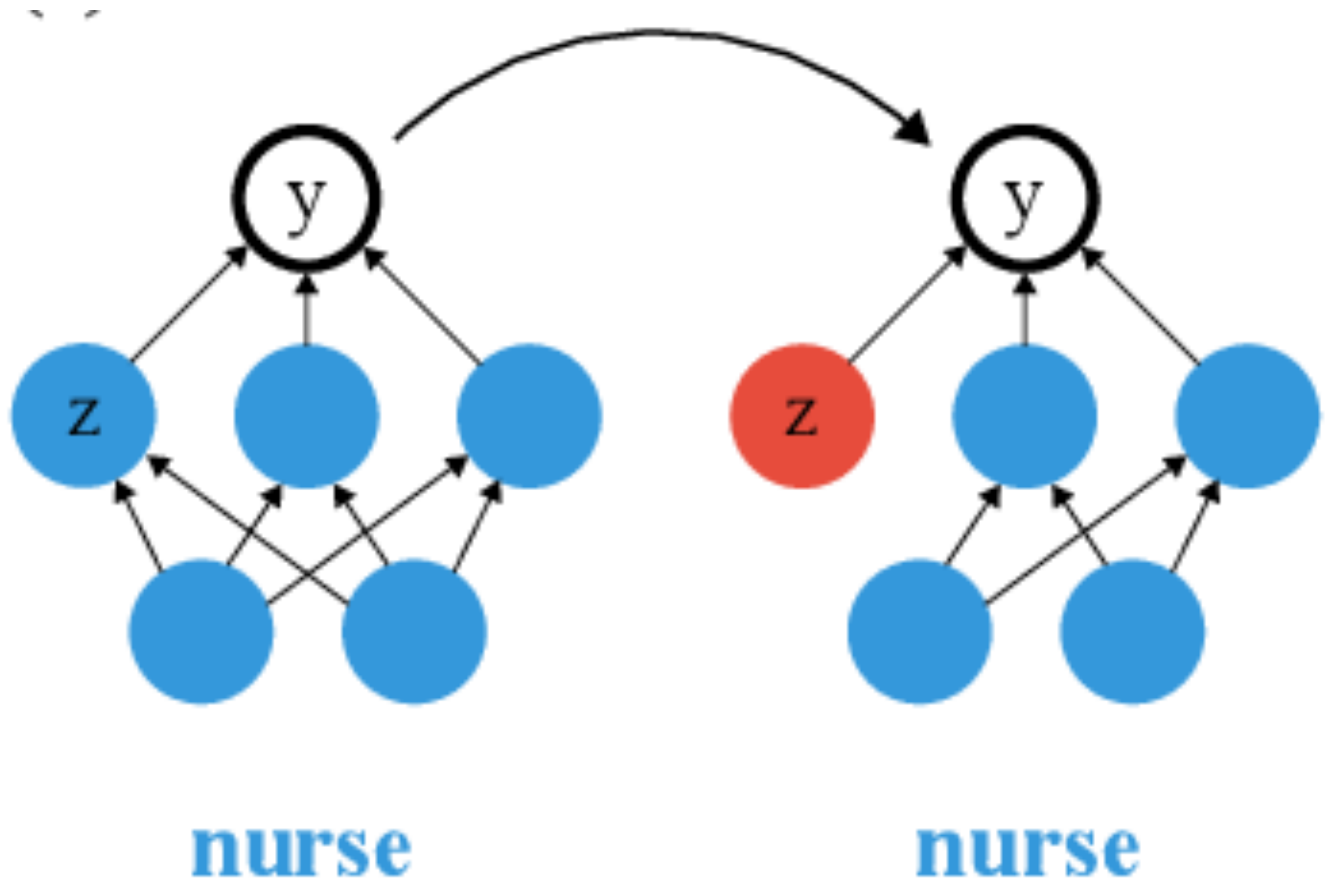
The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Efficiency and Calibration

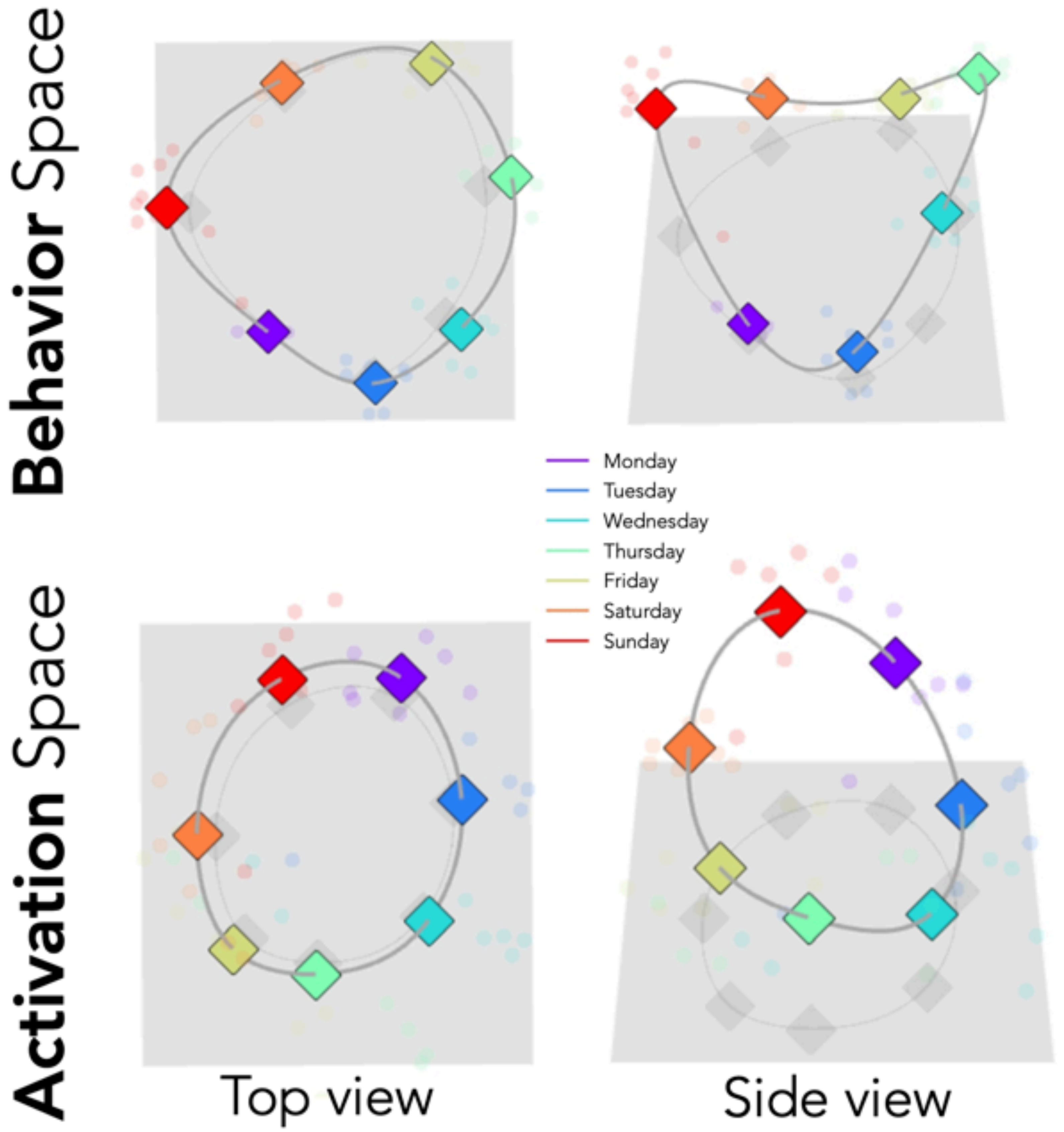
- Scaling up isn't always possible on realistic hardware. How do we handle especially long inputs given realistic hardware?
- How do we handle memory constraints?
  - Mixture-of-experts: Split computation into modular parts
  - Quantization: Use lower-precision data types
- Calibration: Language models aren't always great at giving correct answers. Can we get them to give us an accurate estimate of how likely they are to be right?

Intepretability

What's happening under the hood of language models?



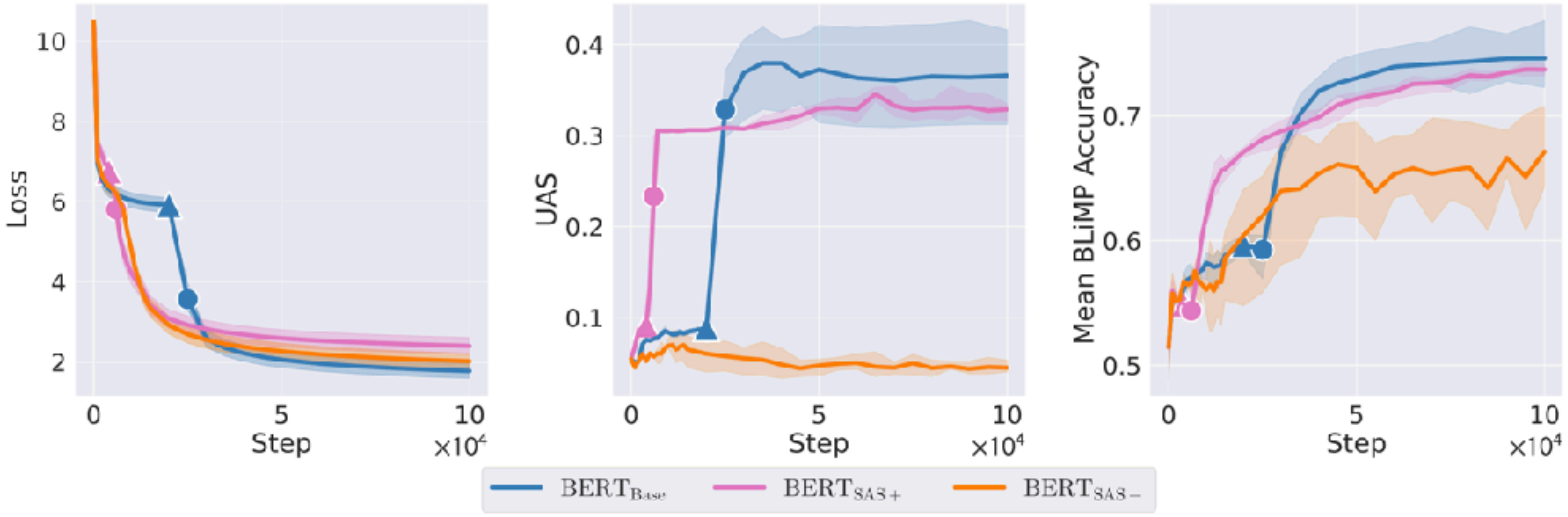
Are there gender bias neurons?



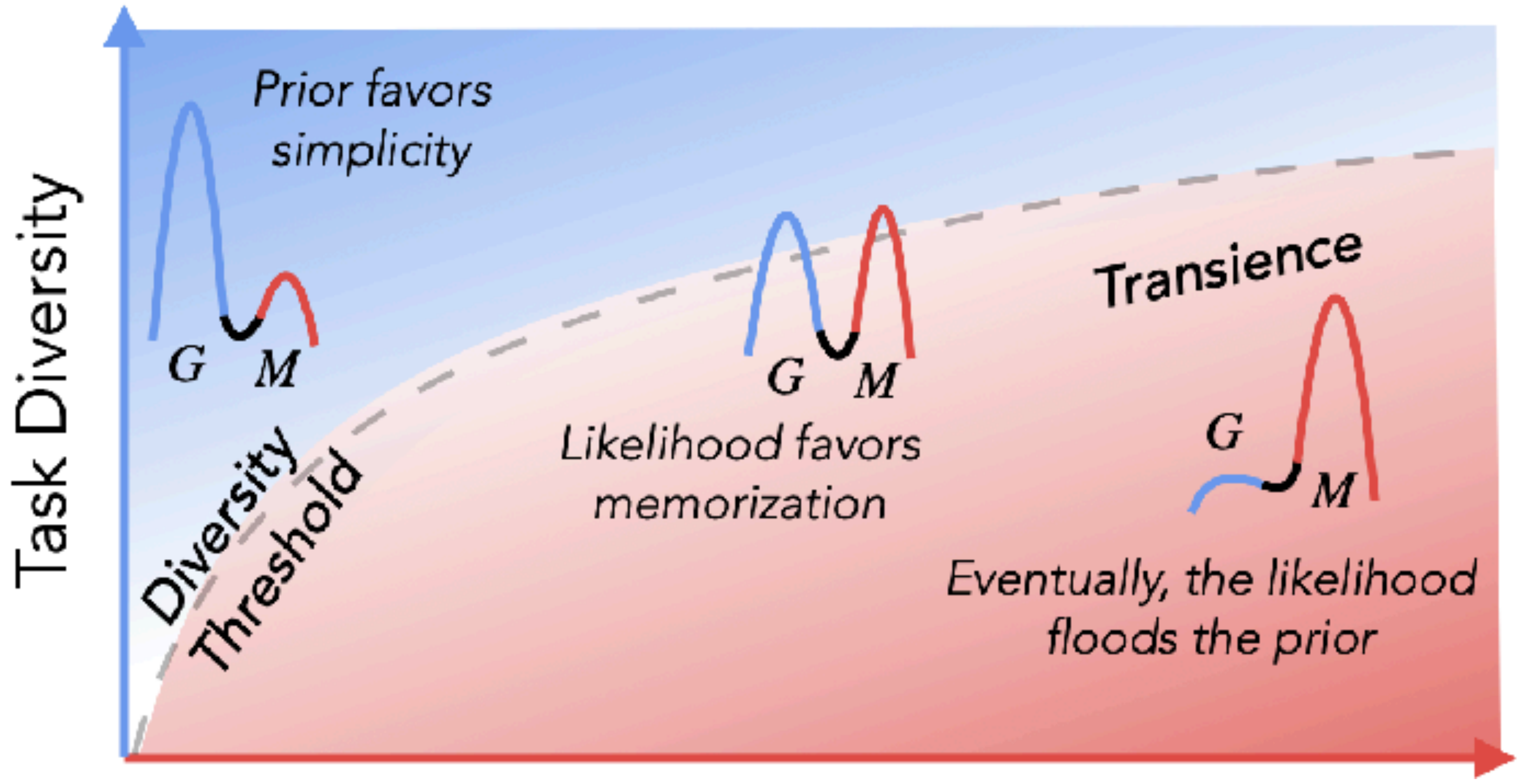
What do concepts look like (geometrically) in a language model?

Training Dynamics

How are particular phenomena learned?
When and why?

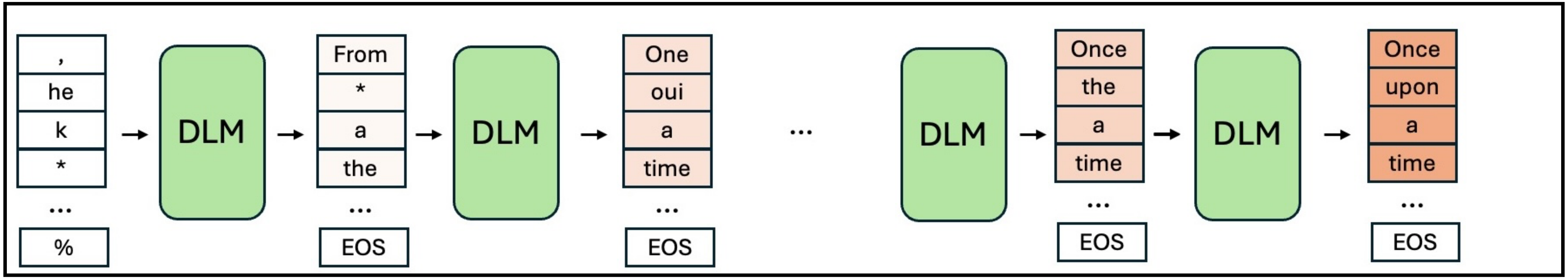


When is syntax learned, and does this immediately translate to good performance on tasks that require it?



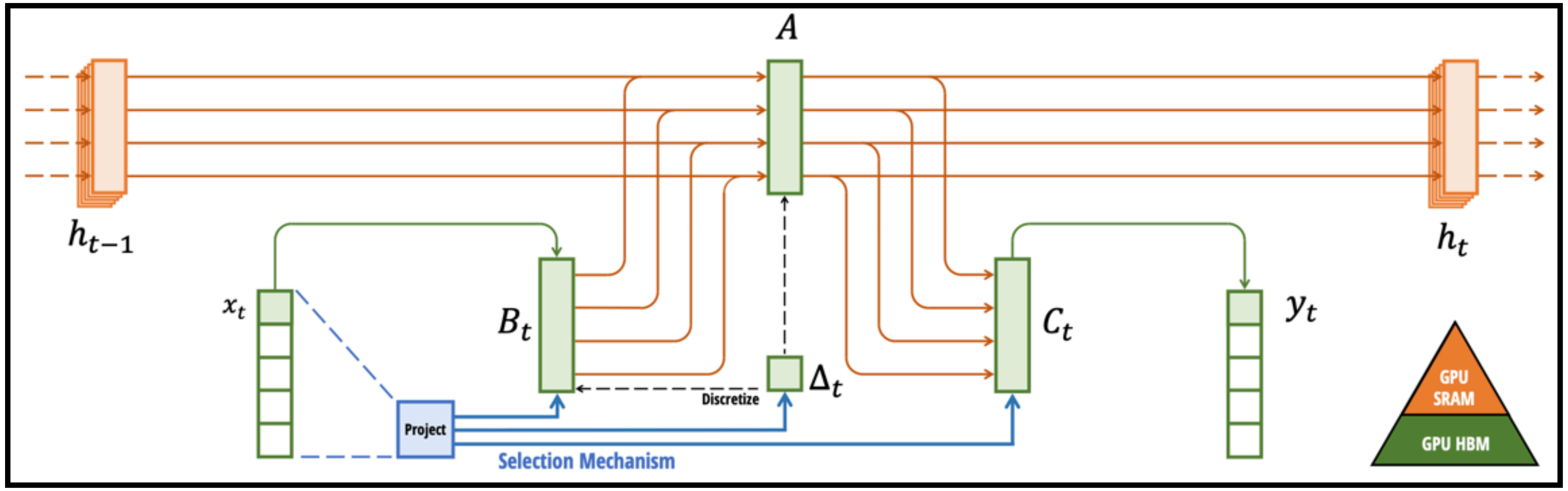
In what conditions do language models learn to generalize, rather than memorize their data?

Alternative Architectures



Generating token-by-token can be slow. **Diffusion LMs** can generate multiple tokens in parallel!

State space models can scale to long contexts more easily than Transformers. *(Or can they?)*



Goals

- Learn how contemporary language models work mechanically, including many of the latest tricks/hacks

- Learn how to perform an original study of some existing language technology, propose an improvement, or design a task-specific application

- Understand the context for each new idea and motivate why it was introduced (i.e., what existing problem(s) it was trying to solve)

Course Logistics

Logistics

- Lecture: **Tuesdays and Thursdays, 12:30pm - 1:45pm**

- Office hours: usually in-person at **CDS 806** starting next week
  - Tuesdays, 2:00pm - 3:30pm (right after class)
  - Wednesdays, 4:00pm - 5:30pm
  - Sometimes, these will be on Zoom; watch Piazza!

Prerequisites

- I expect you to be comfortable with the following:
  - Probability
  - Calculus
  - Linear algebra
  - Programming in Python

- Note that CS505 (NLP) is ***not*** required! The first part of the course will rehash some content from CS505, but we will get far deeper into the weeds as to how current state-of-the-art architectures work.
- Unlike CS505, I *do* expect you to be comfortable with the basics of machine learning from the start. It will be ***very*** helpful to have taken at least one of CS505, CS542, CS585, or a similar course
  - Work through HW0, see if you're comfortable with it or feel you could get up to speed quickly
  - See the syllabus for some PyTorch programming tutorials

Course Format

- Lecture-based course with research paper readings
- 2 homeworks to get you up to speed on the current state of the art
- A half-semester-long research project
- 4 in-class quizzes

Grading

Homeworks: 20 points
- HW1: 12 points
- HW2: 8 points

Final project: 50 points
- Proposal: 5 points
- Midway report: 15 points
- Peer review: 5 points
- Final report: 25 points
  - Includes 5 points for a poster presentation

Quizzes: 30 points
- In-person and on paper
- 4 quizzes
- 10 points each
- Your lowest quiz grade will be dropped

Attendance will not be directly graded! Attend as you wish, as long as you make quiz days (see the course schedule).

Homeworks

- **HW0** (prerequisites review, optional): linear algebra, probability, calculus, machine learning basics

- **HW1:** Build your own Transformer language model from scratch

- **HW2:** Post-train your own language model assistant

Homeworks

- We'll follow a subset of the homeworks from CS336 at Stanford University, which have been highly acclaimed!

- **HW1:** Build your own Transformer language model from scratch
    - Also get some practice running experiments and analyzing results

- **HW2:** Post-train your own reasoning language model

Quizzes

- 15 to 20 minutes at the start of class

- Quizzes I and II will mostly test understanding of content on the homeworks.
  - If you don’t solve them completely with an LLM, you should be fine.
- Quizzes III and IV will test knowledge from the later lectures and paper readings.

- Your lowest quiz grade will be dropped.
  - You can miss one for free, no excuse necessary.
  - (Or you can take them all to give yourself more of a buffer.)

Final Project

You’ll work in groups of 2–4 students on a final project with a topic of your choosing.

- **Final project proposal:** Submit a proposal for what you’d like to work on for your final project.
- **Midway report:** A report in the style of an NLP research paper with some preliminary experiments.
- **Peer review:** Learn how the peer-review process works; provide anonymous feedback on your classmates’ reports. I’ll provide a “meta-review” (and my own feedback).
- **Final report:** Like the midway report, but expanded with more related work and experiments.

Reading

- Our readings will be research papers.
- Research papers can be tricky to read, but *you don't need to understand everything in the paper.*
  - Focus on finding the contributions and main ideas first
  - Papers usually cite some prior work to explain *why* they're doing what they're doing; consider skimming those papers
  - Find the results and takeaways
  - Find out how they got those results; understand the methods

Attention Is All You Need

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks

3.2.2 Multi-Head Attention

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

In this work we employ $h = 8$ parallel attention layers, or heads. For each of these we use $d_k = d_v = d_{\text{model}}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

AI Policy

- AI tools can be useful, and I use them myself when coding! For the homeworks, you can use AI tools as you wish, but please be judicious in how you use them.
  - We encourage using AI to help your conceptual understanding.
  - We *discourage* but *allow* using AI to answer conceptual questions on HWs.
  - We have provided an AGENTS.md and CLAUDE.md for you to use in the homework repositories. Do consider using these.

- For the project proposal/reports, you can use AI, *but* points will be taken off if I can't understand the writing because everything is AI-generated, and/or because of a 100% AI-generated Pangram score. :)

- **You are 100% responsible for what you submit, regardless of whether you use AI.**

- No electronic resources will be allowed during quizzes.

Late Work Policy

- You have 5 **free late days** for which no penalties will be applied to your scores.
  No excuses are necessary to use them.
  - This is a step function! 2 minutes late == 23 hours late

- Extensions can be negotiated in cases of medical emergency or other sudden pressing circumstances.
  - Students should contact me ASAP and discuss extensions *before the assignment's original due date*.

- Each assignment turned in after using all late days *with no negotiated extension prior to the deadline* will incur a **10% drop** in the score. After 5 days past the deadline, the assignment can no longer be turned in.

- The final report cannot be turned in late.

Accommodations

- Please get in touch with us ASAP if you know you need accommodations for religious or disability reasons!

- We require at least a 14-day notice to reschedule any quizzes.

- If any accommodations apply before the first homework is due or first quiz happens, let me know **today.** Talk to me after class!

Compute

- NLP is very GPU-intensive these days. We will have access to the BU SCC. (You won't need it for HW1.)

- I'll give a more detailed overview of how to use the SCC closer to the project proposal deadline.

Administration

- Homework 1 is out! It is due in 3 weeks (Sep. 24, 11:59pm).
  - **Do not leave this until the last minute; it is the hardest assignment in the course.**
  - You will submit your writeup and code to Gradescope
  - See the course website or the Piazza for the instructions PDF

Next Week

- Tuesday: The Transformer architecture
  - Self-attention
  - Multi-head self-attention
  - Some PyTorch basics

- Thursday: Tokenization and position embeddings
  - Rotary position embeddings (RoPE)
  - Byte-pair encoding (BPE)
  - Tokenizer encoding and decoding

Syllabus and Communication

- **Communication:**
  - Announcements on webpage and in class
  - Questions? Use **Piazza**!
  - (Email is ok when needed, but I'll probably respond faster on Piazza.)

- **Course Technology:**
  - Sites: Gradescope and Piazza
  - Some autograded coding assignments,
    manually graded written and coding assignments

https://aaronmueller.github.io/teaching/cs599b1_fall26/home.html

Gradescope entry code: **8YNE72**

Piazza: link on syllabus. Entry code: **hddwru01ukm**