CS11-711 Advanced NLP

Introduction to Natural Language Processing

Sean Welleck

Carnegie Mellon University
Language Technologies Institute

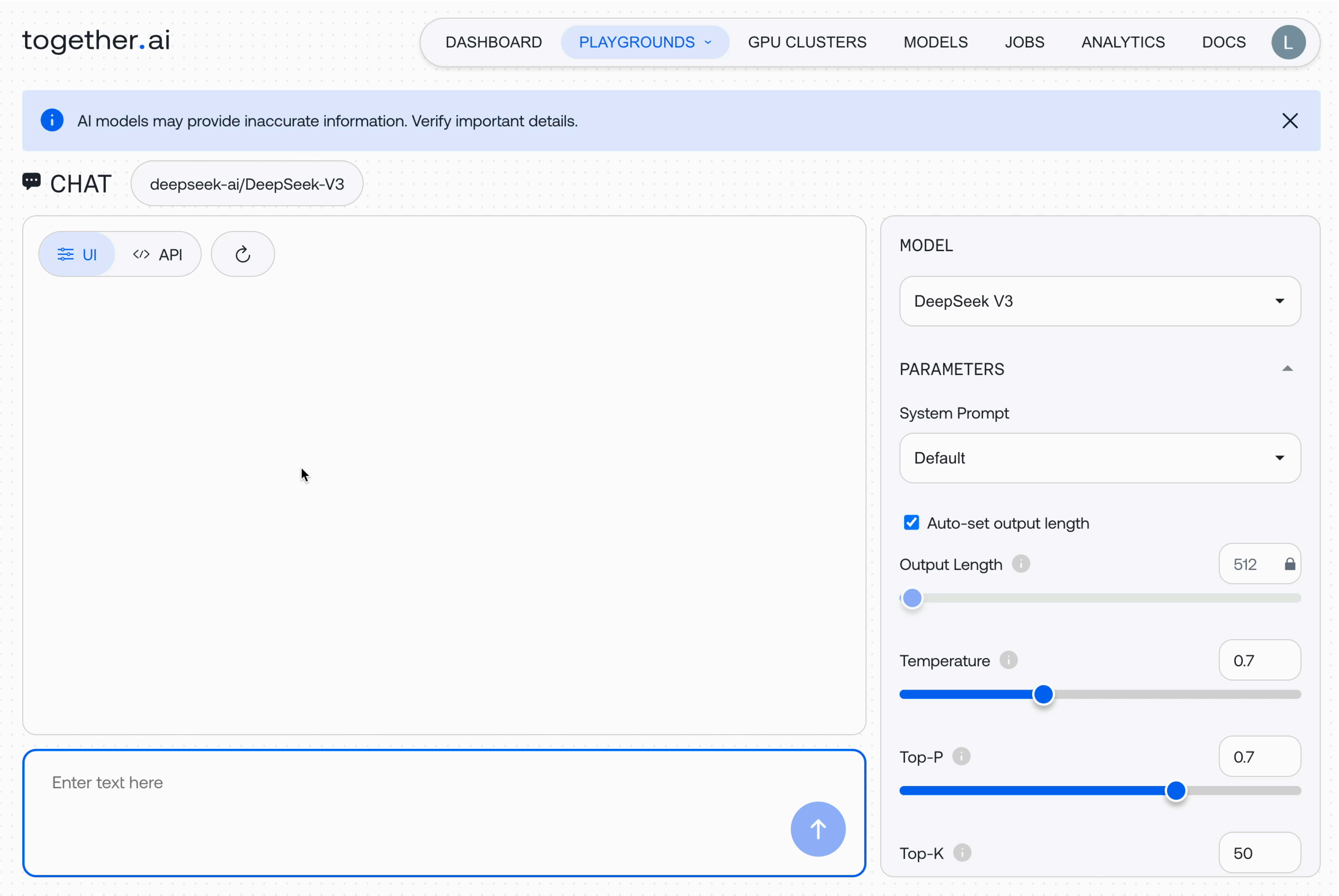
https://cmu-l3.github.io/anlp-spring2025/

Many slides by Graham Neubig

What is Natural Language Processing (NLP)?

- Technology that enables computers to process, generate, and interact with language (e.g., text). Some key aspects:
  - **Learn useful representations**: capture meaning in a structured way that can be used for downstream tasks (e.g., embeddings used to classify a document)
  - **Generate language**: create language (e.g., text, code) for tasks like dialogue, translation, or question answering.
  - **Bridge language and action**: Use language to perform tasks, solve problems, interact with environments (e.g., a code IDE)

Today’s NLP



AI models may provide inaccurate information. Verify important details.

deepseek-ai/DeepSeek-V3

Enter text here

MODEL

DeepSeek V3

PARAMETERS

System Prompt

Default

☑ Auto-set output length

Output Length 512

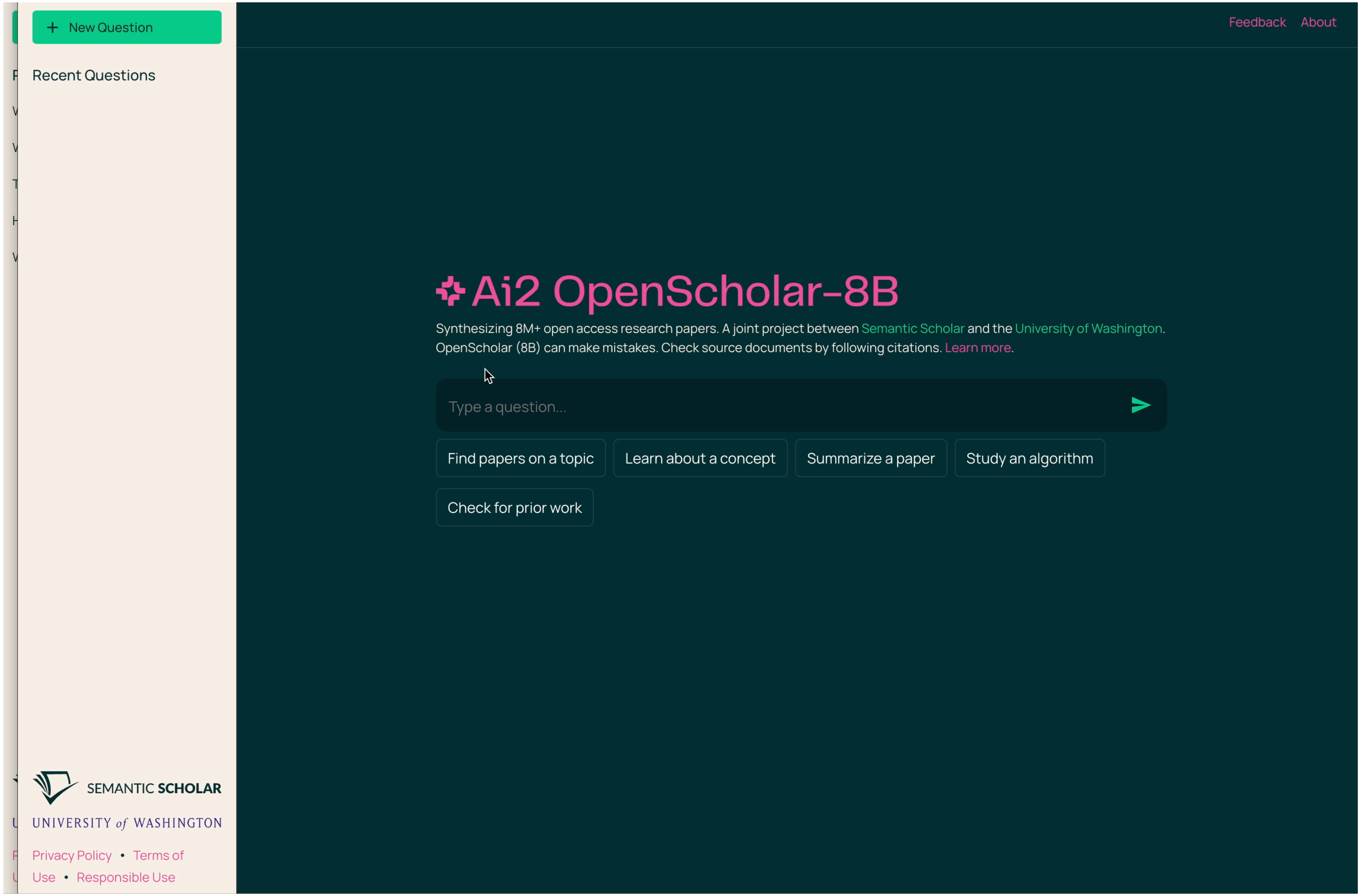
Temperature 0.7

Top-P 0.7

Top-K 50

DeepSeek-V3 on Together.ai, Generated Jan 8, 2025

Today's NLP

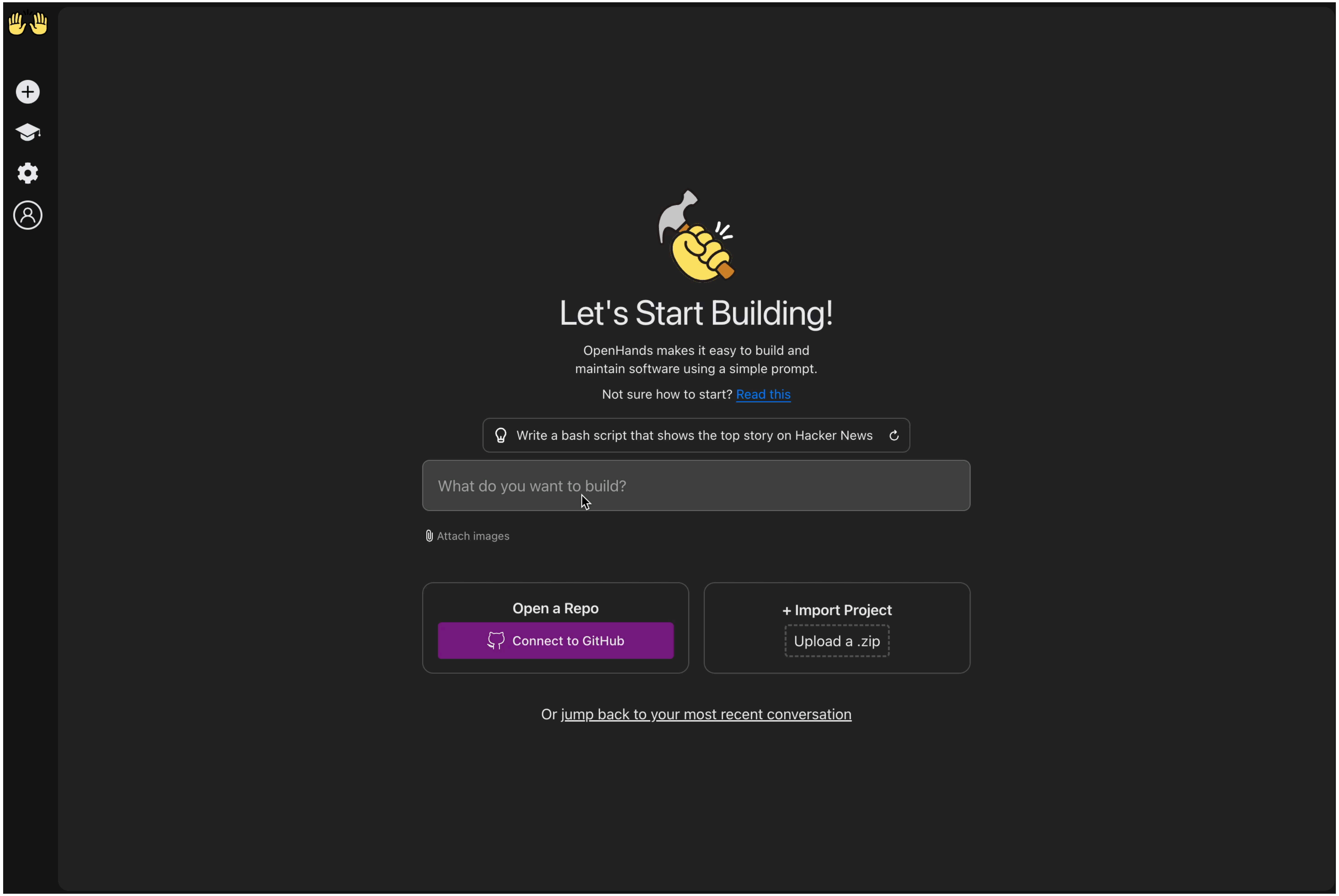


Ai2 OpenScholar-8B

Synthesizing 8M+ open access research papers. A joint project between Semantic Scholar and the University of Washington. OpenScholar (8B) can make mistakes. Check source documents by following citations. Learn more.

https://openscholar.allen.ai/, Generated Jan 8, 2025

Today's NLP



Let's Start Building!

OpenHands makes it easy to build and maintain software using a simple prompt.

Not sure how to start? Read this

Write a bash script that shows the top story on Hacker News

What do you want to build?

Or jump back to your most recent conversation

https://github.com/All-Hands-AI/OpenHands , generated Jan 8 2025

In this class, you’ll learn the fundamental concepts and practical techniques underlying systems like these!

In this class, we ask:

- What goes into building the state-of-the-art NLP systems that **work well at some tasks**?
- Where and why do current state-of-the-art NLP systems still **struggle**?
- How can we **make appropriate improvements** and **achieve whatever we want to do with NLP**?

NLP System Building Overview

A General Framework for NLP Systems

- Create a function to map an input *X* into an output *Y*, where *X* and/or *Y* involve language.

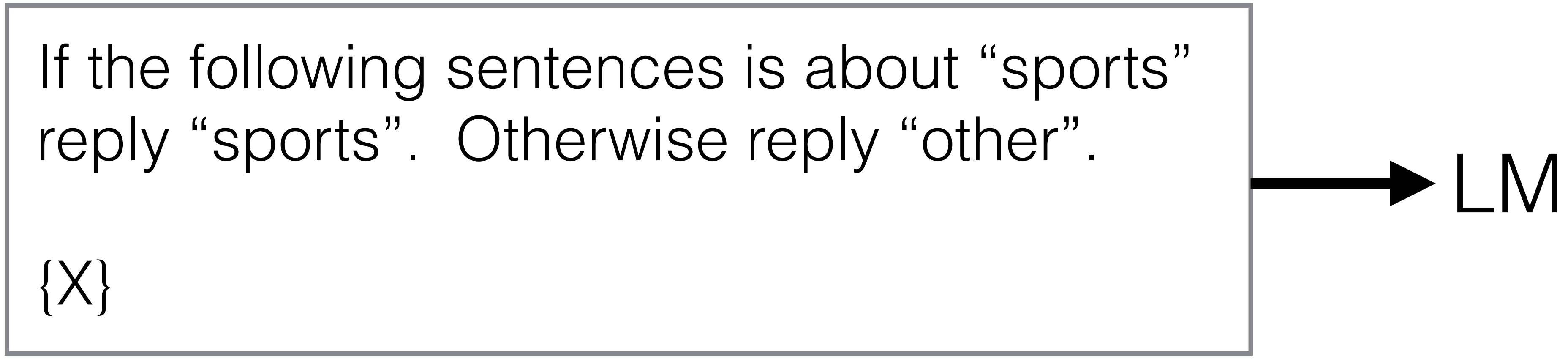
Input *X*	Output *Y*	Task
Text	Continuing Text	Language Modeling
Text	Text in Other Language	Translation
Text	Label	Text Classification
Text	Linguistic Structure	Language Analysis
Image	Text	Image Captioning

Common Methods for Creating NLP Systems

- **Rules:** Manual creation of rules

```
def classify(x: str) -> str:
    sports_keywords = ["baseball", "soccer", "football", “tennis"]
    if any(keyword in x for keyword in sports_keywords):
        return "sports"
    else:
        return "other"
```

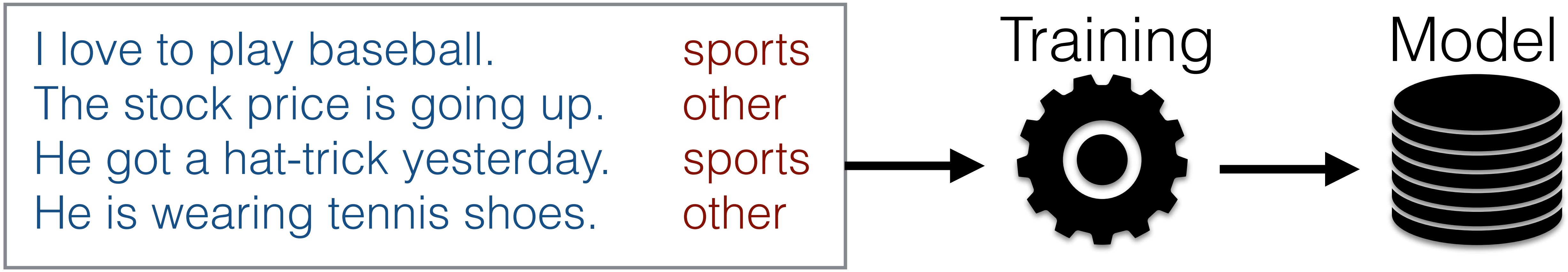
- **Prompting:** Prompting a language model w/o training



If the following sentences is about “sports” reply “sports”. Otherwise reply “other”.

{X}

- **Fine-tuning:** Machine learning from paired data <*X*, *Y*>



I love to play baseball.	sports
The stock price is going up.	other
He got a hat-trick yesterday.	sports
He is wearing tennis shoes.	other

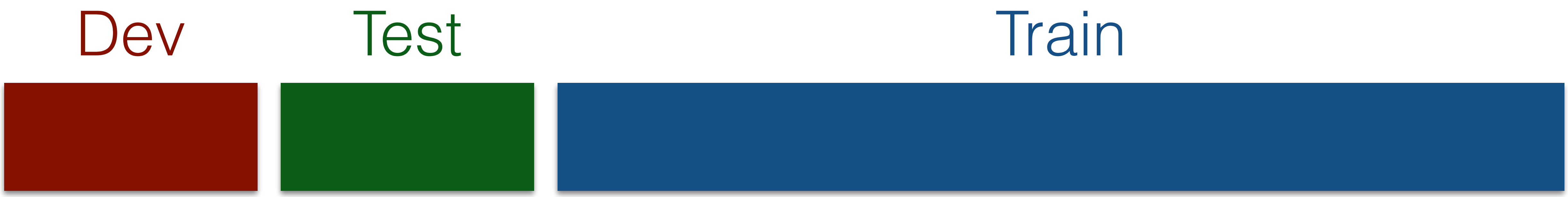
Data Requirements for System Building

- **Rules/prompting based on intuition:**
  No data needed, but also no performance guarantees

- **Rules/prompting based on spot-checks:**
  A small amount of data with input X only

- **Rules/prompting with rigorous evaluation:**
  Development set with input X and output Y (e.g. 200-2000 examples). Additional held-out test set also preferable.

- **Fine-tuning:**
  Additional train set. More is often better — constant accuracy increase when data size doubles.

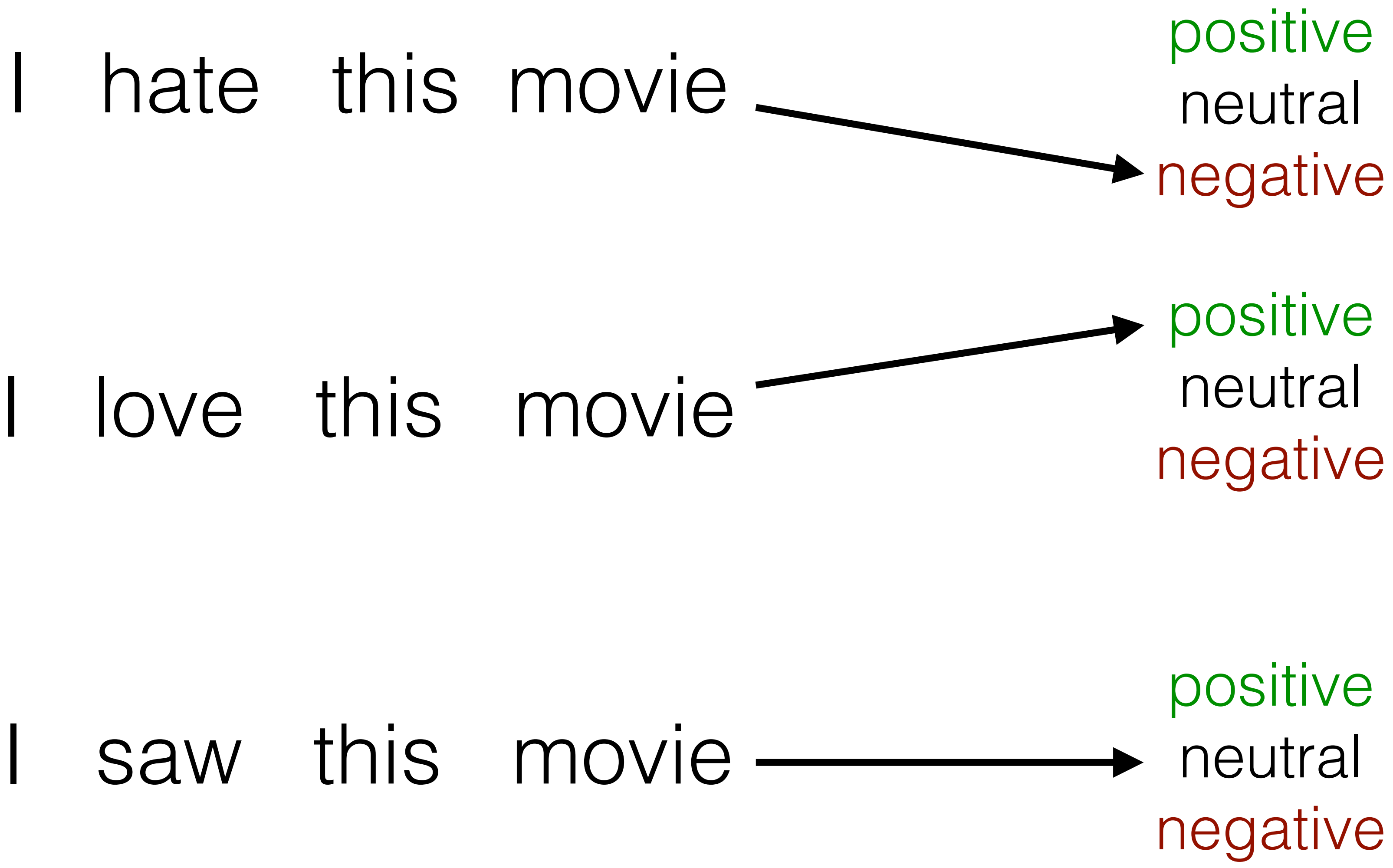


Dev Test Train

Let's Try to Make a Rule-Based NLP System!

Example Task:
Review Sentiment Analysis

- Given a review on a reviewing web site (*X*), decide whether its label (*Y*) is positive (1), negative (-1) or neutral (0)



I hate this movie

positive
neutral
negative

I love this movie

positive
neutral
negative

I saw this movie

positive
neutral
negative

A Three-step Process for Making Predictions

- **Feature extraction:** Extract the salient features for making the decision from text
- **Score calculation:** Calculate a score for one or more possibilities
- **Decision function:** Choose one of the several possibilities

Formally

- **Feature Extraction:** $\mathbf{h} = f(x)$
- **Score Calculation:** binary, multi-class

$$s = \mathbf{w} \cdot \mathbf{h} \quad \mathbf{s} = W\mathbf{h}$$

- **Decision:** $\hat{y} = g(s)$

Sentiment Classification Code Walkthrough

https://github.com/cmu-l3/anlp-spring2025-code/blob/main/01_intro/rule_based_classifier.ipynb

- See code for all major steps:
  1. Featurization
  2. Scoring
  3. Decision rule
  4. Accuracy calculation
  5. Error analysis

Now Let's Look at Data

https://github.com/cmu-l3/anlp-spring2025-code/tree/main/01_intro/data

cmu-l3 / anlp-spring2025-code

- Remember: don’t look at “test”!

Now Let's Improve

1. What's going wrong with my system?
   → Look at error analysis

2. Modify the system (featurization, scoring function, etc.)

3. Measure accuracy improvements, accept/reject change

4. Repeat from 1

5. Finally, when satisfied with dev accuracy, evaluate on test

Some Difficult Cases

Low-frequency Words

The action switches between past and present , but the material link is too **tenuous** to anchor the emotional connections that **purport** to span a 125-year divide .

negative

Here 's yet another studio horror franchise **mucking** up its storyline with **glitches** casual fans could correct in their sleep .

negative

Solution?: Keep working until we get all of them? Incorporate external resources such as sentiment dictionaries?

Conjugation

An operatic , sprawling picture that 's **entertainingly** acted , **magnificently** shot and gripping enough to sustain most of its 170-minute length .

positive

It 's basically an **overlong** episode of Tales from the Crypt .

negative

Solution?: Use the root form and part-of-speech of word?

Note: Would require morphological analysis.

Negation

This one is not nearly as dreadful as expected .

positive

Serving Sara does n't serve up a whole lot of laughs .

negative

Solution?: If a negation modifies a word, disregard it.

Note: Would probably need to do syntactic analysis.

Metaphor, Analogy

Puts a human face on a land most Westerners are unfamiliar with.

positive

Green might want to hang onto that ski mask , as robbery may be the only way to pay for his next project .

negative

Has all the depth of a wading pool .

negative

Solution?: ???

Other Languages

見事に視聴者の心を掴む作品でした。

positive

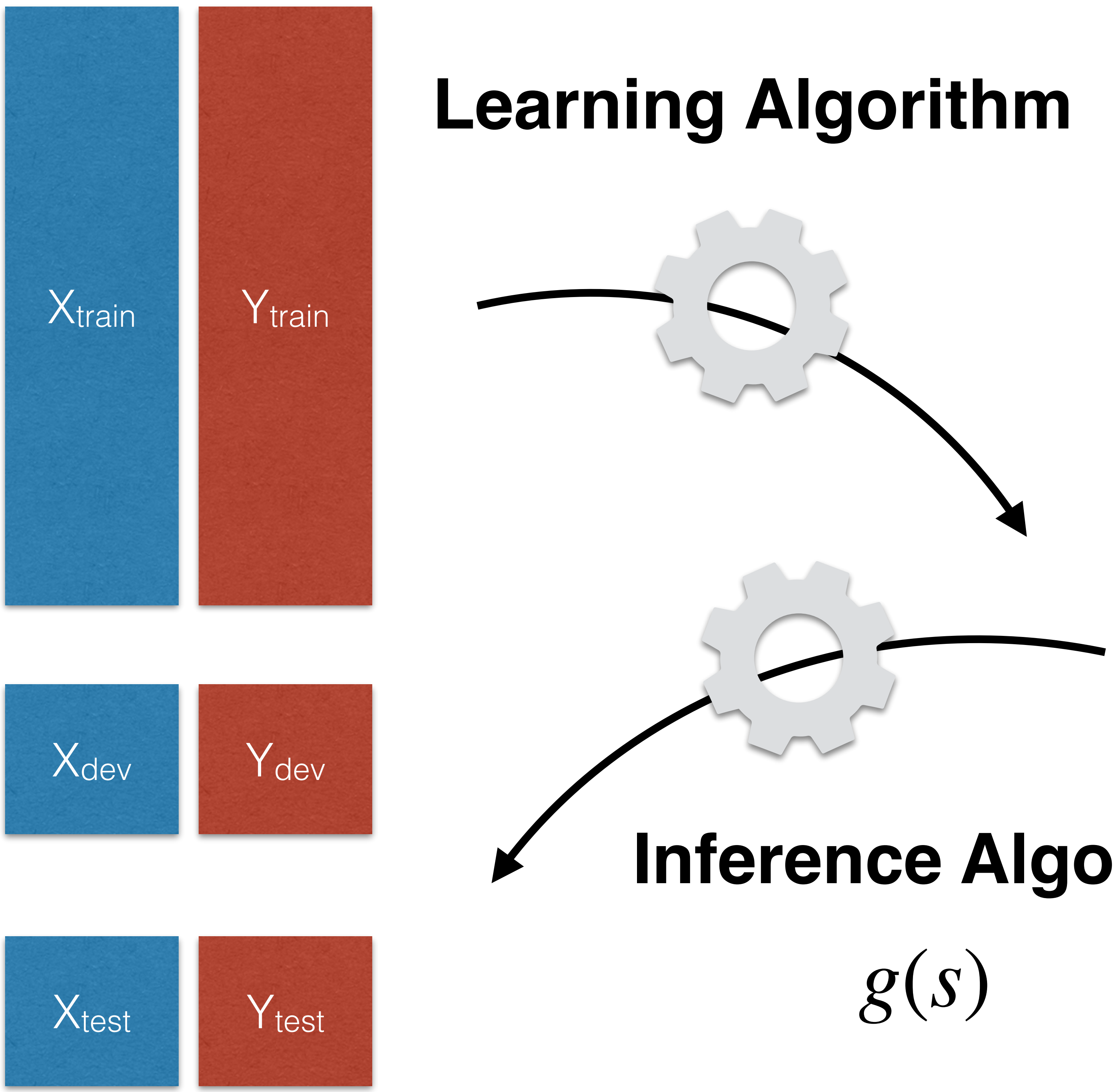
モンハンの名前がついてるからとりあえずモンハン要素をちょこちょこ入れればいいだろ感が凄い。

negative

Solution?: Learn Japanese and re-do all the work?

Machine Learning Based NLP

Machine Learning



Learning Algorithm

Learned
Feature Extractor *f*
Weights *w*

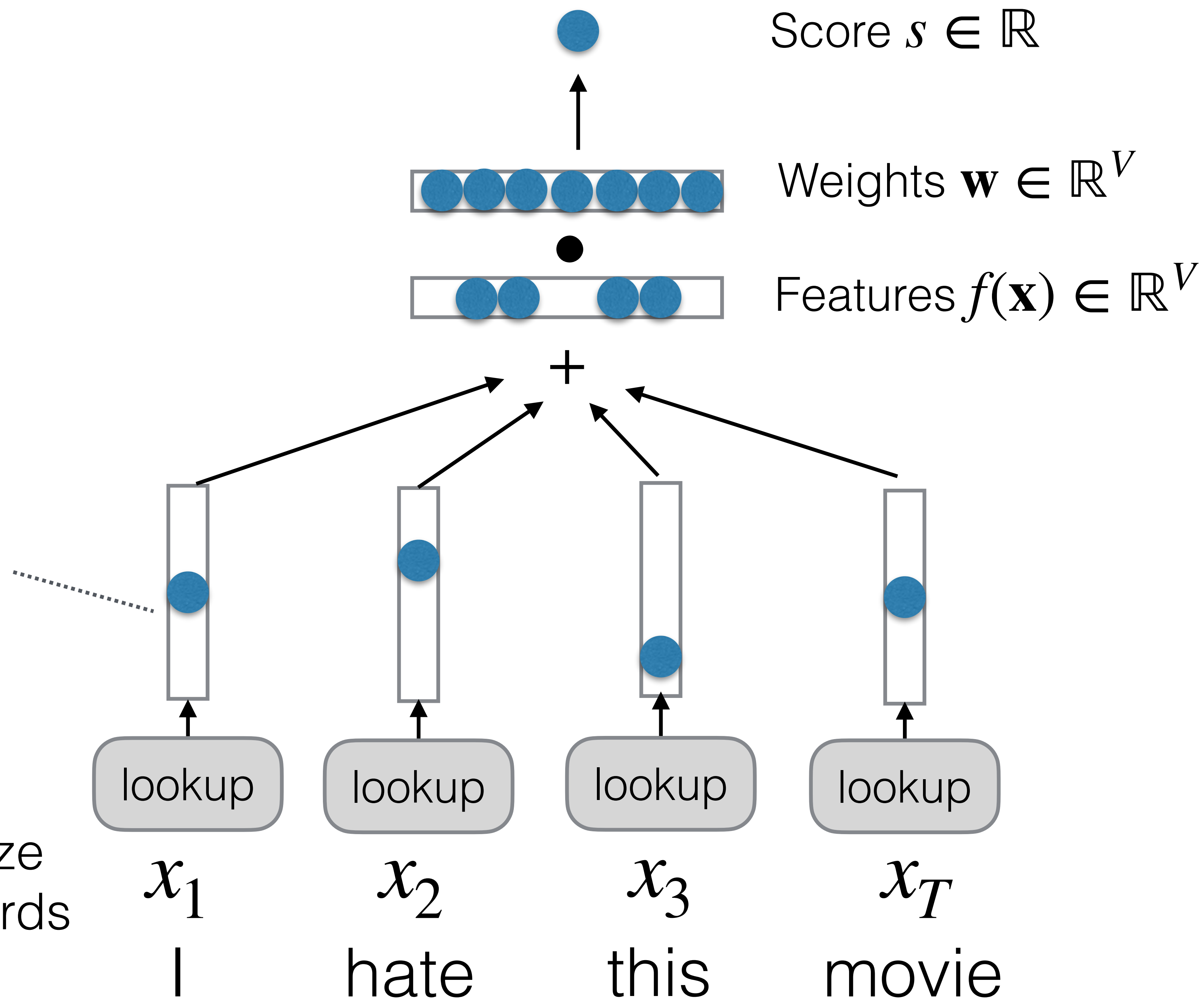
$$\mathbf{h} = f(\mathbf{x})$$

$$s = \mathbf{w} \cdot \mathbf{h}$$

Inference Algorithm

$$g(s)$$

Attempt 1: Bag of Words (BoW)

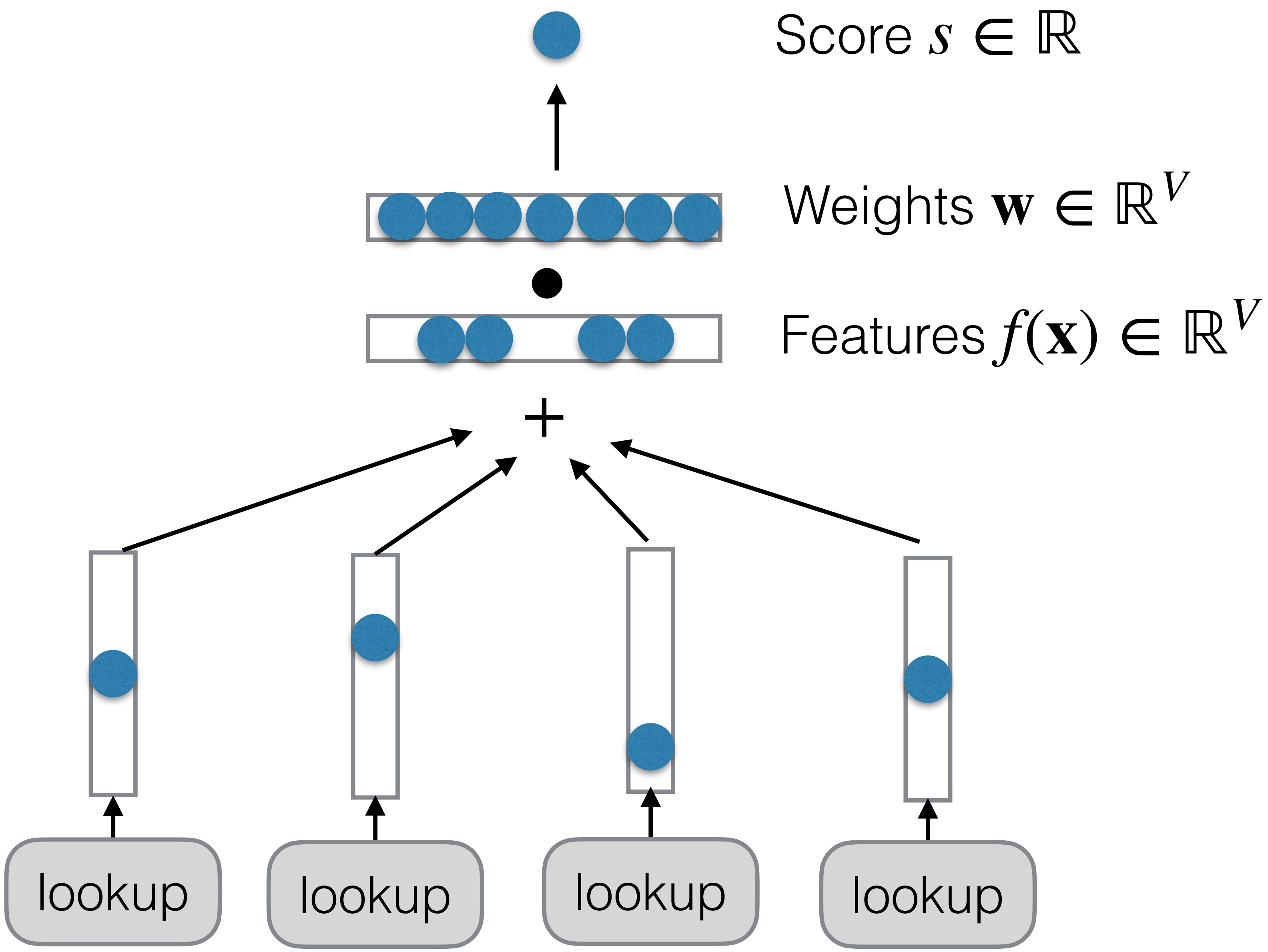


$$\begin{bmatrix} 0 & \text{a} \\ 0 & \text{aardvark} \\ \vdots & \vdots \\ 1 & \text{I} \\ \vdots & \vdots \\ 0 & \text{zebra} \end{bmatrix}$$

$$x_t \in \{0,1\}^V$$

V: Vocabulary size
(e.g., set of all words
in training set)

Attempt 1: Bag of Words (BoW)


Score $s \in \mathbb{R}$
Weights $\mathbf{w} \in \mathbb{R}^V$
•
Features $f(\mathbf{x}) \in \mathbb{R}^V$
+
lookup
lookup
lookup
lookup

Features *f* are based on word identity, weights *w* learned

Which problems mentioned before would this solve?

What do Our Vectors Represent?

- **Binary classification:** Each word has a single scalar, positive indicating “yes” and negative indicating “no”

- **Multi-class classification:** Each word has its own 5 elements corresponding to [very good, good, neutral, bad, very bad]

Binary

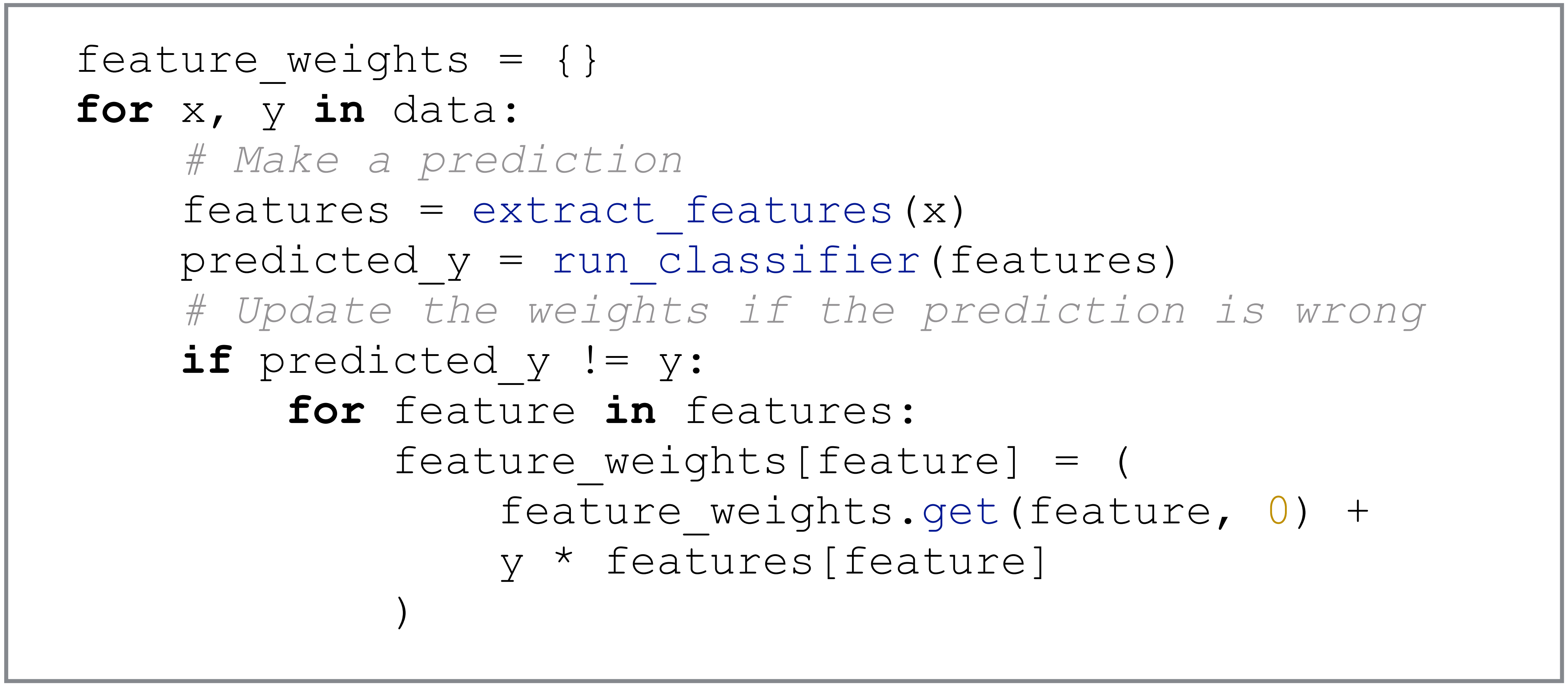
	$\mathbf{w} \in \mathbb{R}^V$
love	2.4
hate	-3.5
nice	1.2
no	-0.2
dog	-0.3
…	…

Multi-class

$\mathbf{W} \in \mathbb{R}^{V \times K}$ $K = 5$	v. positive	positive	neutral	negative	v. negative
love	2.4	1.5	-0.5	-0.8	-1.4
hate	-3.5	-2.0	-1.0	0.4	3.2
nice	1.2	2.1	0.4	-0.1	-0.2
no	-0.2	0.3	-0.1	0.4	0.5
dog	-0.1	0.3	0.6	0.2	-0.2
…			…		

Simple Training of BoW Models

- Use an algorithm called “structured perceptron”



```
feature_weights = {}
for x, y in data:
    # Make a prediction
    features = extract_features(x)
    predicted_y = run_classifier(features)
    # Update the weights if the prediction is wrong
    if predicted_y != y:
        for feature in features:
            feature_weights[feature] = (
                feature_weights.get(feature, 0) +
                y * features[feature]
            )
```

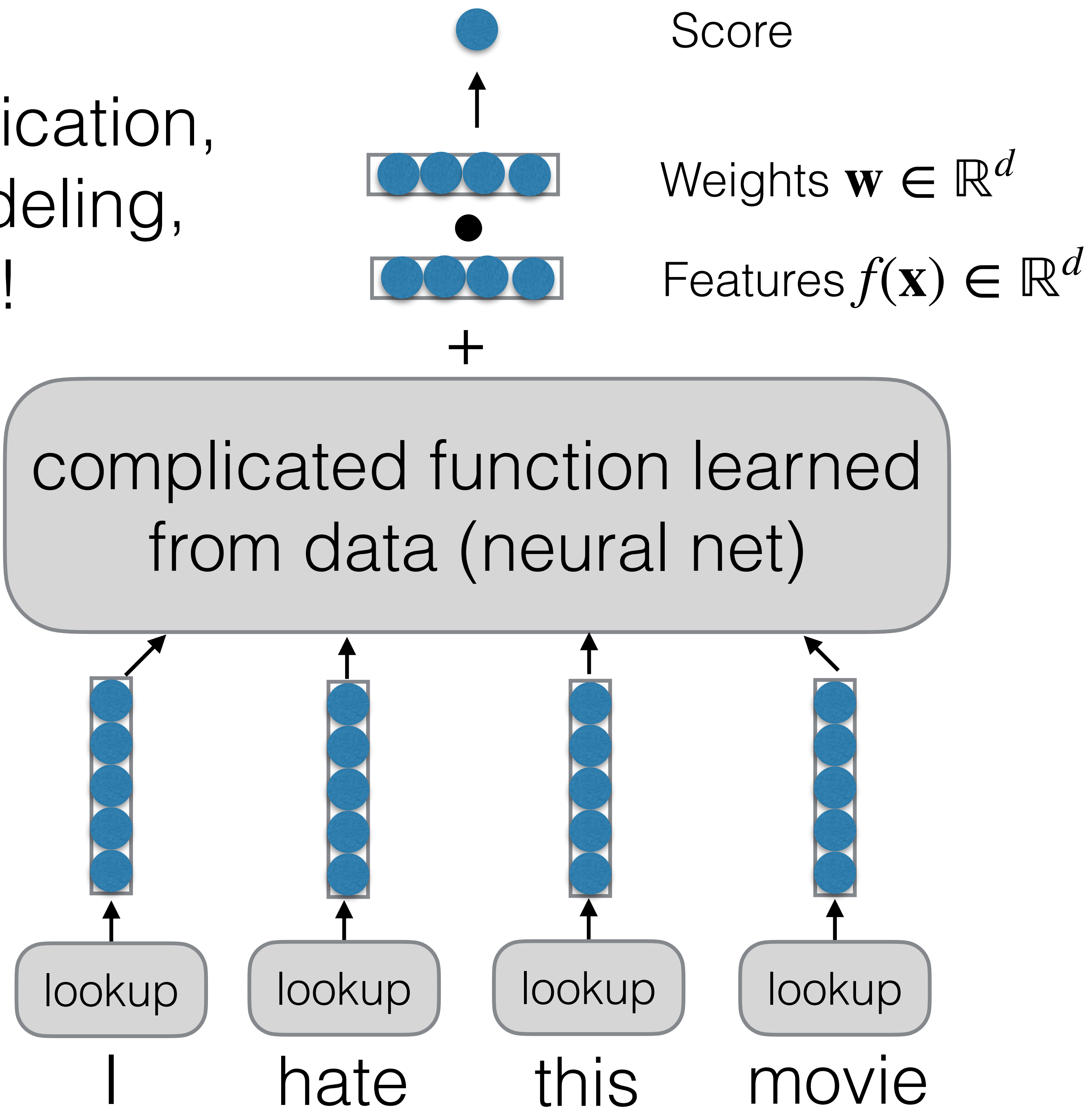
Full Example:
https://github.com/cmu-l3/anlp-spring2025-code/blob/main/01_intro/trained_bow_classifier.ipynb

What’s Missing in BoW?

- Handling of *conjugated or compound words*
  - I **love** this move -> I **loved** this movie
- Handling of *word similarity*
  - I **love** this move -> I **adore** this movie
- Handling of *combination features*
  - I **love** this movie -> I **don’t love** this movie
  - I **hate** this movie -> I **don’t hate** this movie
- Handling of *sentence structure*
  - It has an interesting story, **but** is boring overall

A Better Attempt: Neural Networks

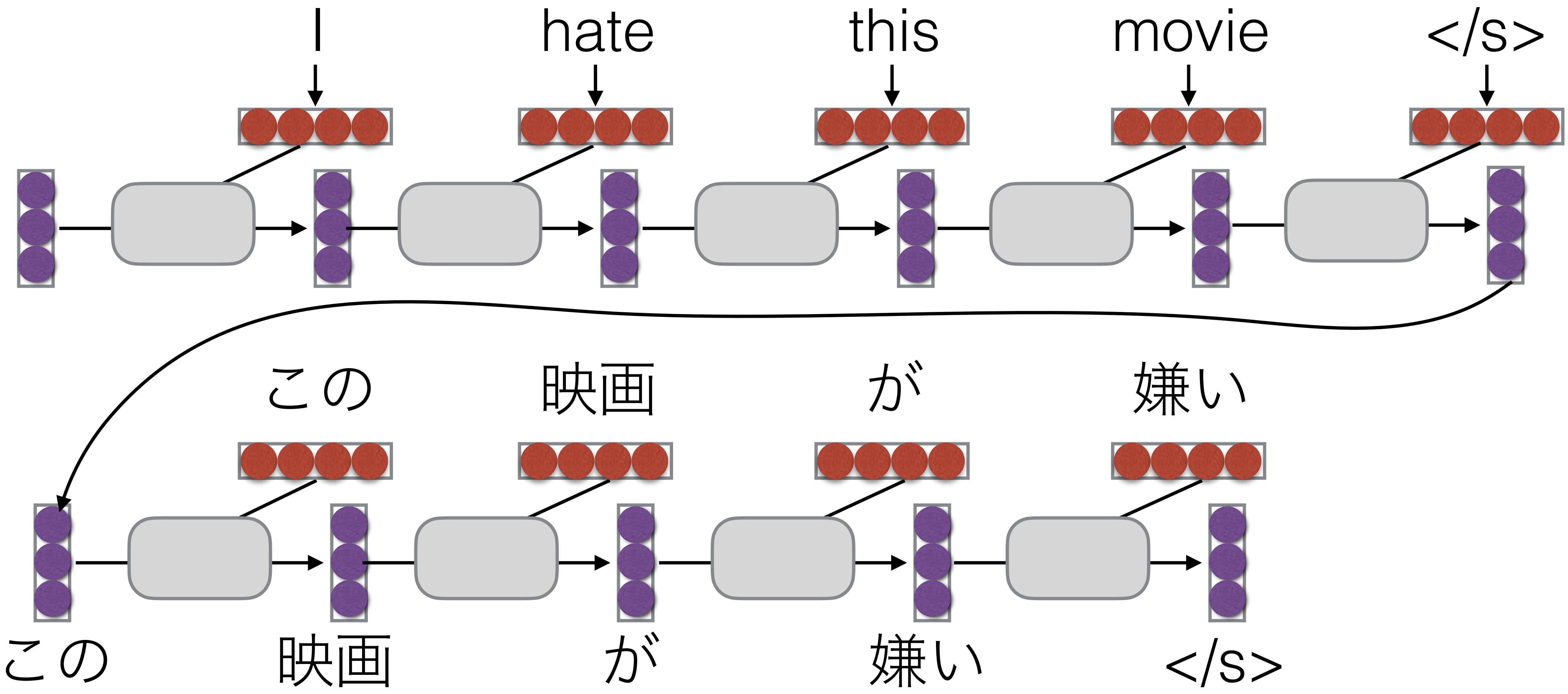
Perform classification,
language modeling,
any task!



Score
Weights $\mathbf{w} \in \mathbb{R}^d$
Features $f(\mathbf{x}) \in \mathbb{R}^d$
+
complicated function learned from data (neural net)
lookup
lookup
lookup
lookup
I
hate
this
movie

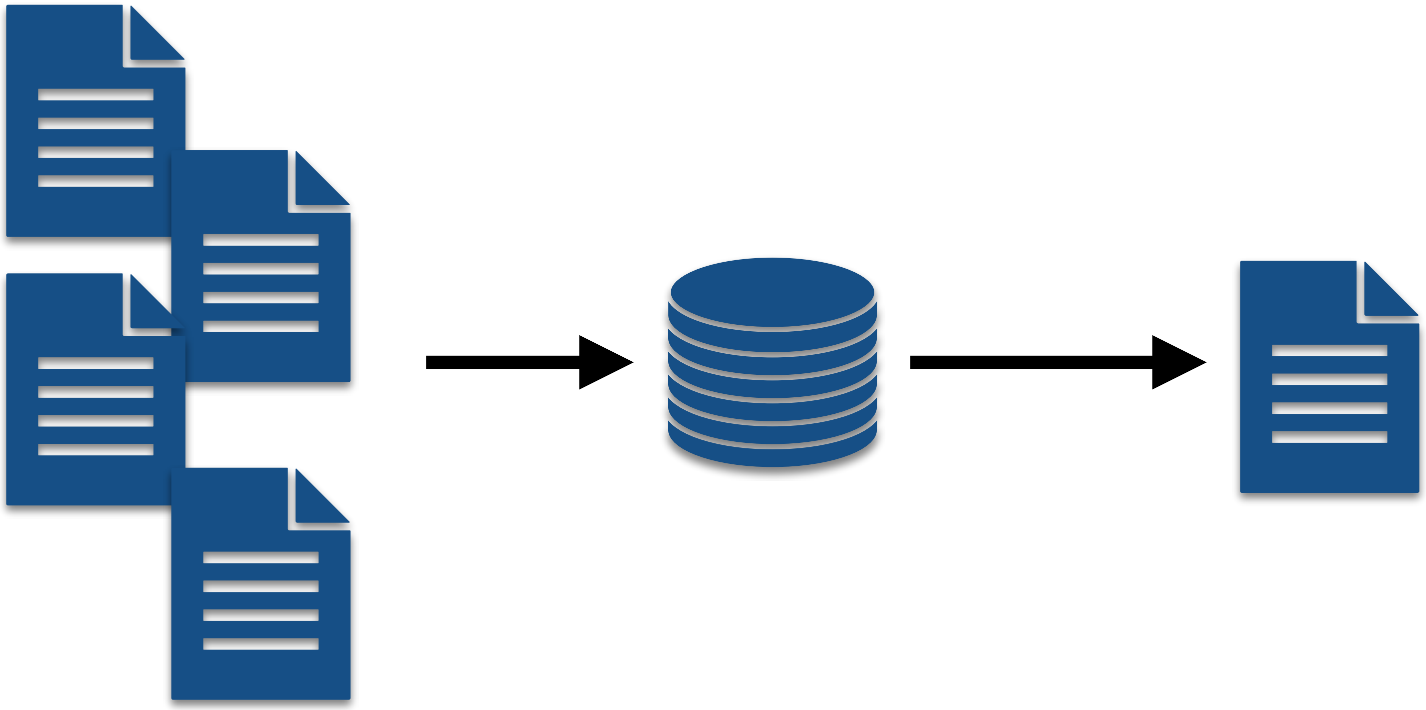
Roadmap Going Forward

Topic 1: Language Modeling Fundamentals



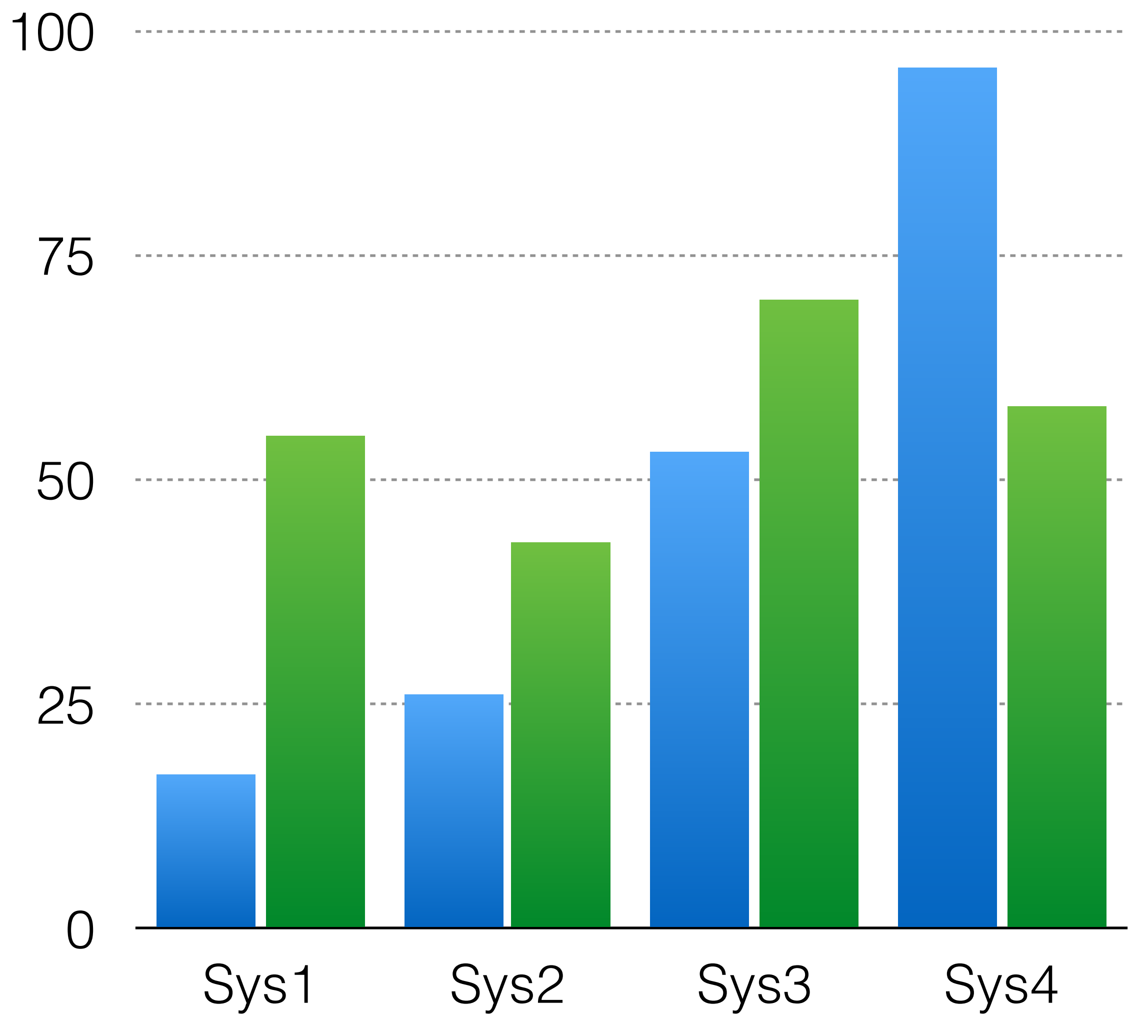
- Representing words
- Language modeling
- Sequence modeling architectures

Topic 2: Training and Inference Methods



- Decoding and Generation Algorithms
- In-context learning
- Pre-training
- Fine-tuning
- Reinforcement Learning

Topic 3: Evaluation and Experimental Design

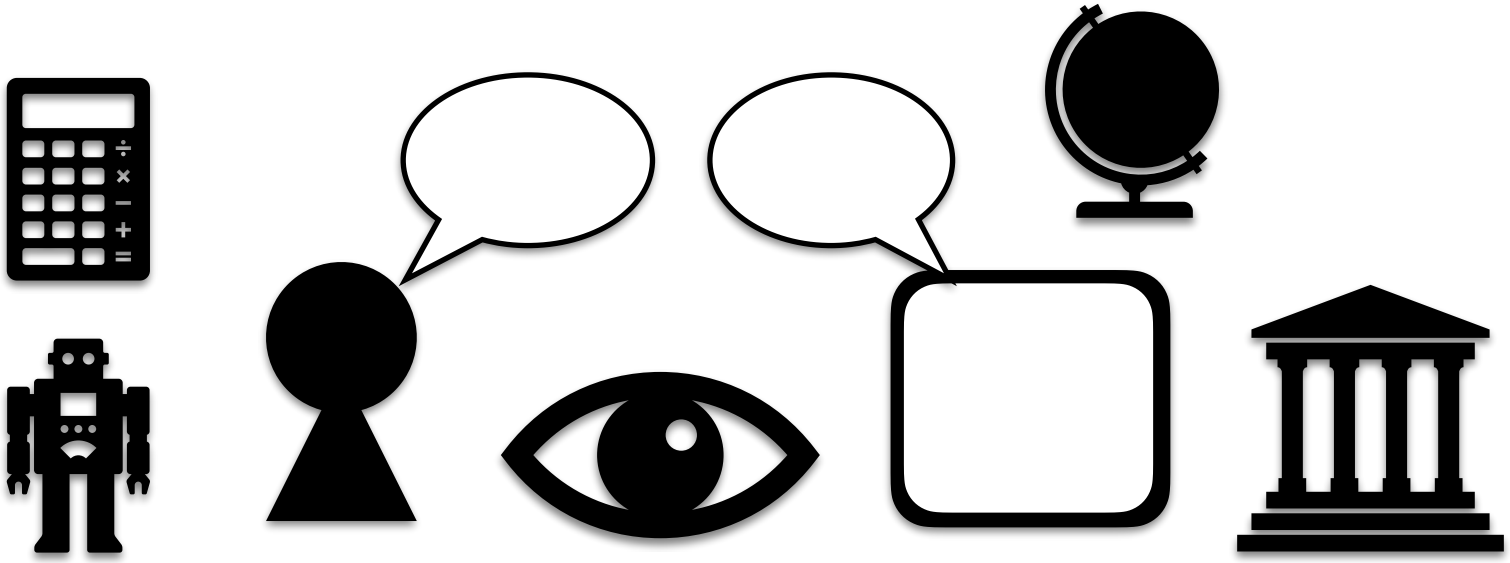


- Evaluating Language Generators
- Experimental Design
- Human Annotation
- Debugging/Interpretation Techniques

Topic 4:
Advanced Algorithms and Architectures

- Advanced Pretraining, Post-Training, and Inference
- Retrieval and Retrieval-augmented Generation
- Long Sequence Models
- Distillation and Quantization
- Ensembling and Mixture of Experts

Topic 5: NLP Applications and Society



- Complex Reasoning Tasks
- Language Agents
- Multimodal NLP
- Multilingual NLP
- Bias and Fairness

Class Format/Structure

Class Content

- Learn in detail about **building NLP systems from a research perspective**

- Learn basic and advanced topics in **machine learning approaches** to NLP and language models

- See several case studies of **NLP applications** and learn how to identify unique problems for each

- Learn how to debug **when and where NLP systems fail**, and build improvements based on this

Class Format

- **Before class:** For some classes, do recommended reading
- **During class:**
    - *Lecture/Discussion:* Go through material and discuss
    - *Code/Data Walk:* The TAs (or instructor) will sometimes walk through some demonstration code, data, or model predictions
- **After class:** Do quiz about class or reading material

Assignments

- **Assignment 1 - Build-your-own LLaMa:** *Individually* implement LLaMa model loading and training
- **Assignment 2 - NLP Task from Scratch:** *In a team,* perform data creation, modeling, and evaluation for a specified task
- **Assignment 3 - SOTA Survey / Re-implementation:** Survey literature, re-implement and reproduce results from a recently published NLP paper
- **Assignment 4 - Final Project:** Perform a unique project that either (1) improves on state-of-the-art, or (2) applies NLP models to a unique task. Present a poster and write a report.

Teaching Team and Resources

- **Instructor:** Sean Welleck
- **TAs:** Darsh Agrawal, Hugo Contant, Alex Fang, Akshita Gupta, Trisha Sarkar, Manan Sharma, Sanidhya Vijayvargiya
- **Piazza:** https://piazza.com/cmu/spring2025/11711
- **Website:** https://cmu-l3.github.io/anlp-spring2025/
- **Code:** https://github.com/cmu-l3/anlp-spring2025-code

Thanks, Any Questions?