CS 333: Natural Language Processing

Fall 2025

Prof. Carolyn Anderson
Wellesley College

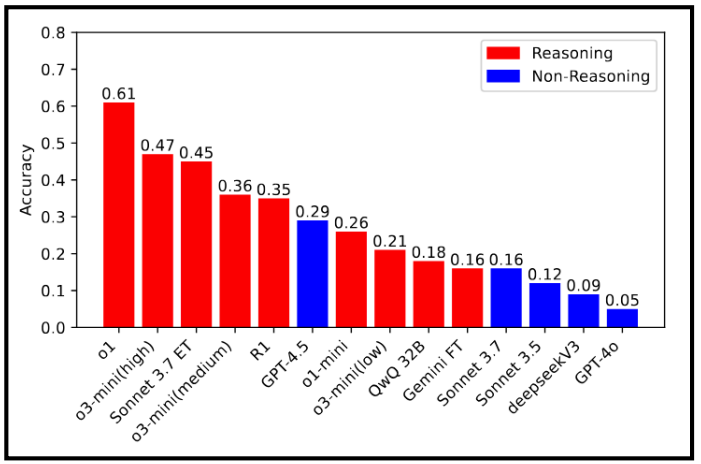
Recent Work by Wellesley Alums

ReasoningWeekly: A General Knowledge and Verbal Reasoning Challenge for Large Language Models

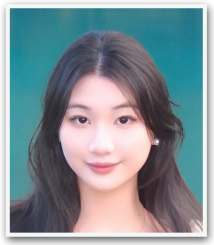
Zixuan Wu, Francesca Lucchetti, Aleksander Boruch-Gruszecki, Jingmiao Zhao, Carolyn Jane Anderson, Joydeep Biswas, Federico Cassano, Arjun Guha

28 Jul 2025 (modified: 14 Oct 2025) ACL ARR 2025 July Submission
July, Senior Area Chairs, Area Chairs, Reviewers, Authors, Commitment Readers Revisions

Abstract:
Existing benchmarks for frontier models often test specialized, "PhD-level" knowledge that is difficult for non-experts to grasp. In contrast, we present a benchmark with 613 problems based on the NPR Sunday Puzzle Challenge that requires only general knowledge. Our benchmark is challenging for both humans and models; however correct solutions are easy to verify, and models' mistakes are easy to spot. As LLMs are more widely deployed in society, we believe it is useful to develop benchmarks for frontier models that humans can understand without the need for deep domain expertise.



Class of 2024



Class of 2023 (took CS 333 in Fall 2023!)

Example item:

Challenge: Think of a common greeting in ~~another~~ **a** country **that is not the United States**. You can rearrange its letters to get the capital of a country that neighbors the country where this greeting is commonly spoken. What greeting is it?
Ground Truth Answer: Ni hao –> Hanoi

Accepted to the International Joint Conference on Natural Language Processing & Asia-Pacific Chapter of the Association for Computational Linguistics, 2025 (AACL)

Major NLP Conferences

ACL – main rotating conference

AACL – Asia ACL

NAACL – Nations of Americas ACL

EACL – Europe ACL

EMNLP – Empirical Methods in NLP

CoLM – Conference on Language Modeling

ML: ICLR NeurIPS

ACL Anthology

Reminders

- Midterm 2 is on Friday! Basic calculator OK
  - The exam is open-note but closed-device
  - List of topics posted on the course website.
- Tuesday -> Monday cycle for final two assignments:
  - HW 7 will be released Friday but not due until Monday, 11/17
  - HW 8 will be released on Tuesday, 11/18 and due on Monday, 11/24
- My next help hours: Thursday 4-5

NEW RESEARCH IN RELIGIOUS STUDIES

Hinduism Online: Mechanized Murti and Automated Adoration

Presented by Dr. Holly Walters

Department of Anthropology

SCAN QR CODE
TO RSVP

Tuesday, November 4th at 5pm
FND 120

WELLESLEY

exp: 11/5
?: mk110@wellesley.edu

ADR:
accessibility@wellesley.edu

Recorded?

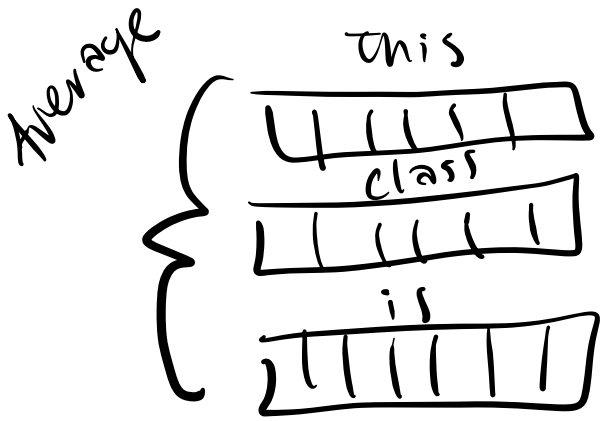
Lindsey Cameron Colloquium

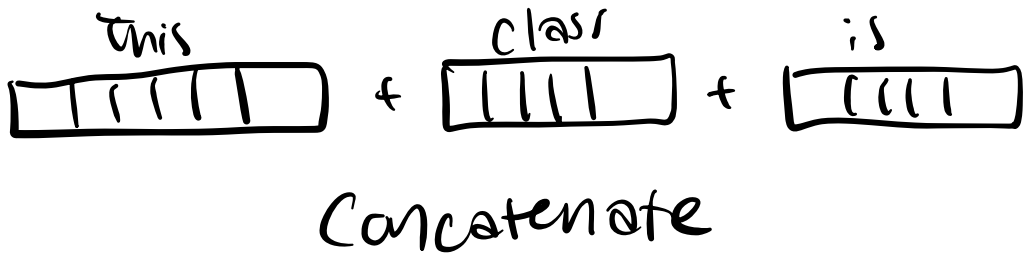
November 14th, 3:45-5pm in H105

Title: Scalable Subjugation: The Myth of Geographic Scalability in the Gig Economy and How Workers Reconstitute Platforms

HW 5 Curiosity Points

- Model changes and tweaks:
  - Implemented weighting to address class imbalance
  - Exploration of lexical diversity-based features
  - Extension to a neural network model
  - Visualization of feature correlations
  - Data analysis functions to aid in feature engineering
- Task extensions:
  - Experiments on different artists
  - Naive Bayes comparison experiment
  - Binary classification experiment
  - Swift album classifier
- Research literature exploration





Recap: Recurrent Neural Networks

A RNN Language Model

output distribution

$$\hat{y} = \text{softmax}(W_2 h^{(t)})$$

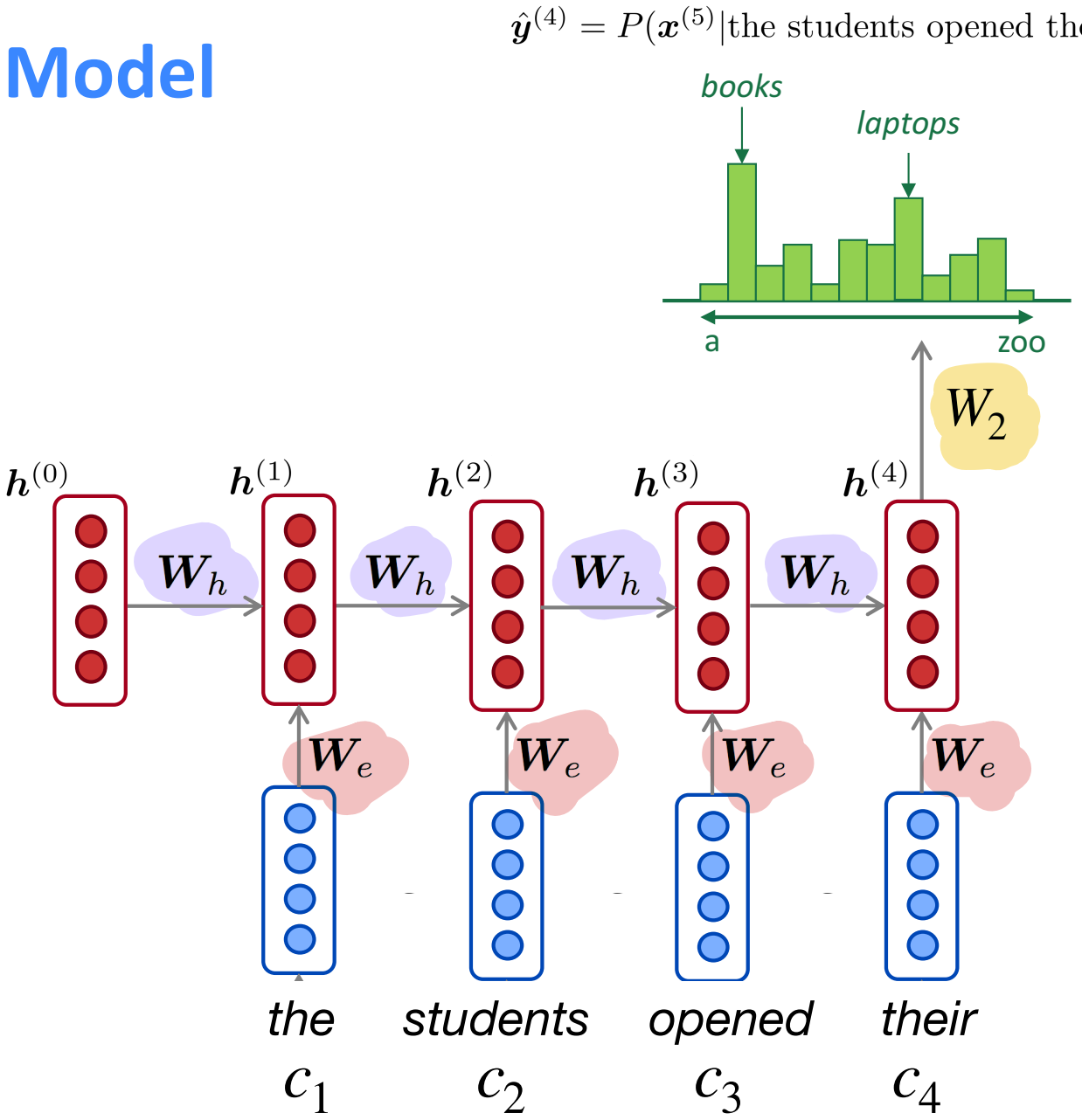
hidden states

$$h^{(t)} = f(W_h h^{(t-1)} + W_e c_t)$$

h(0) is initial hidden state!

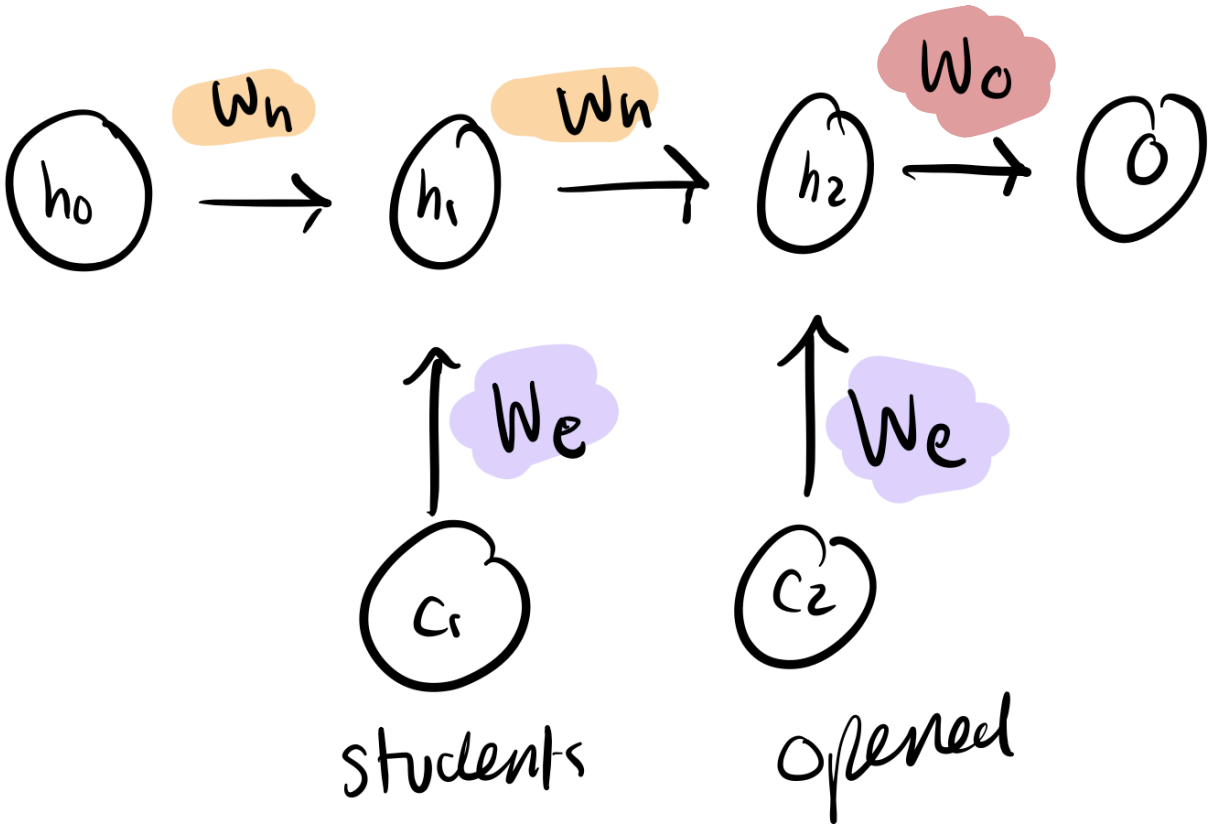
word embeddings

$$c_1, c_2, c_3, c_4$$



Training a Recurrent Neural Network

Training an RNN



$$h_1 = \tanh(W_e C_1 + W_h h_0)$$

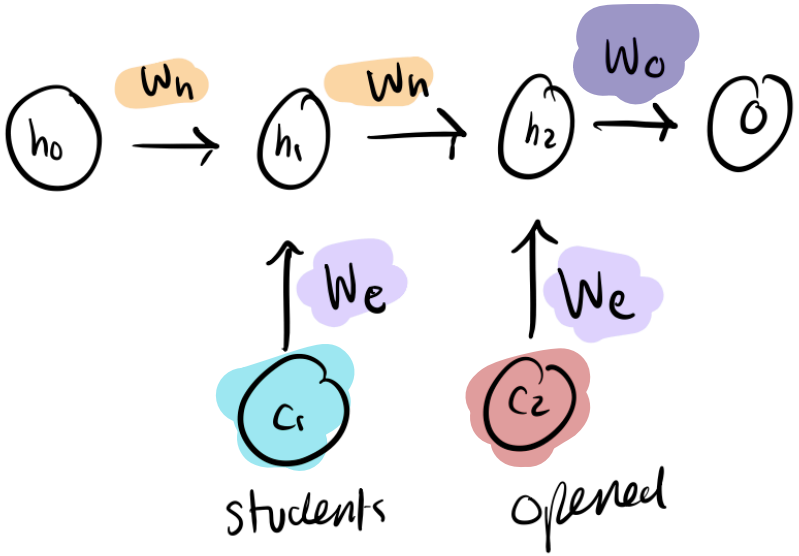
$$h_2 = \tanh(W_e C_2 + W_h h_1)$$

Square Loss

$$L = \frac{1}{2}(y - o)^2$$

$$o = W_o \cdot h_2$$

Key Question: what are the parameters?



W_o

W_h

W_e

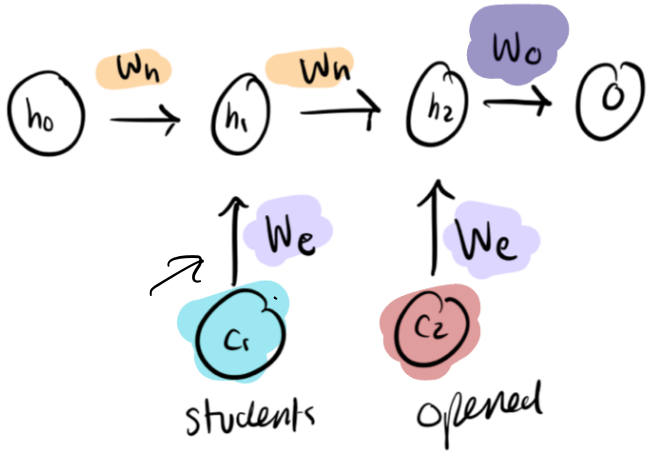
c_1

c_2

$$L = \frac{1}{2}(y - o)^2$$

$$o = W_o \cdot h_2$$

Gradients



$$L = \frac{1}{2}(y - o)^2$$

$$o = W_o \cdot h_2$$

$$h_1 = \tanh(W_e C_1 + W_h h_0)$$

$$h_2 = \tanh(W_e C_2 + W_h h_1)$$

$$\frac{\partial L}{\partial W_o}, \quad \frac{\partial L}{\partial W_e}, \quad \frac{\partial L}{\partial W_h}, \quad \frac{\partial L}{C_2}$$

Gradient of w_0

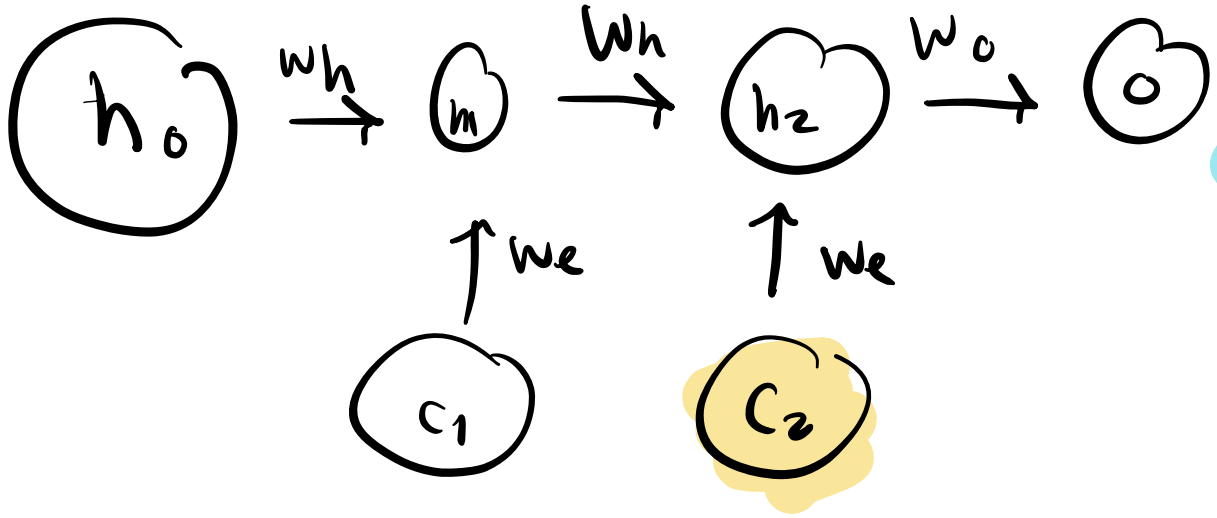
$$L = \frac{1}{2}(y-o)^2 \qquad o = w_0 h_2$$

$$\frac{\partial L}{\partial w_0} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial w_0} = -(y-o) \cdot h_2$$

$$\frac{\partial L}{\partial o} = -(y-o) \qquad \frac{\partial o}{\partial w_0} = h_2$$

Gradient of c_2

$$h_2 = \tanh\left(\overset{a}{w_e c_2} + \overset{b}{w_h \cdot h_1}\right)$$



$$a = w_e \cdot c_2$$

$$\frac{\partial L}{\partial c_2} = \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial h_2} \cdot \frac{\partial h_2}{\partial a} \cdot \frac{\partial a}{\partial c_2}$$

$$-(y - o) \quad W_o\left(1 - h_2^2\right) \quad W_e$$

Gradient of w_e

$$L = \frac{1}{2}(y-o)^2 \qquad o = w_o h_2$$

Goal:

$$\frac{\partial L}{\partial w_e}$$

w_e is used twice

so we need gradients from those two time steps

$$h_2 = \tanh(a+b)$$

$$a = w_e c_2$$

$$b = w_h \cdot h_1$$

$$h_1 = \tanh(\underbrace{w_e c_1}_{a_2} + w_h \cdot \underbrace{h_0}_{b_2})$$

$$a_2 = w_e \cdot c_1$$

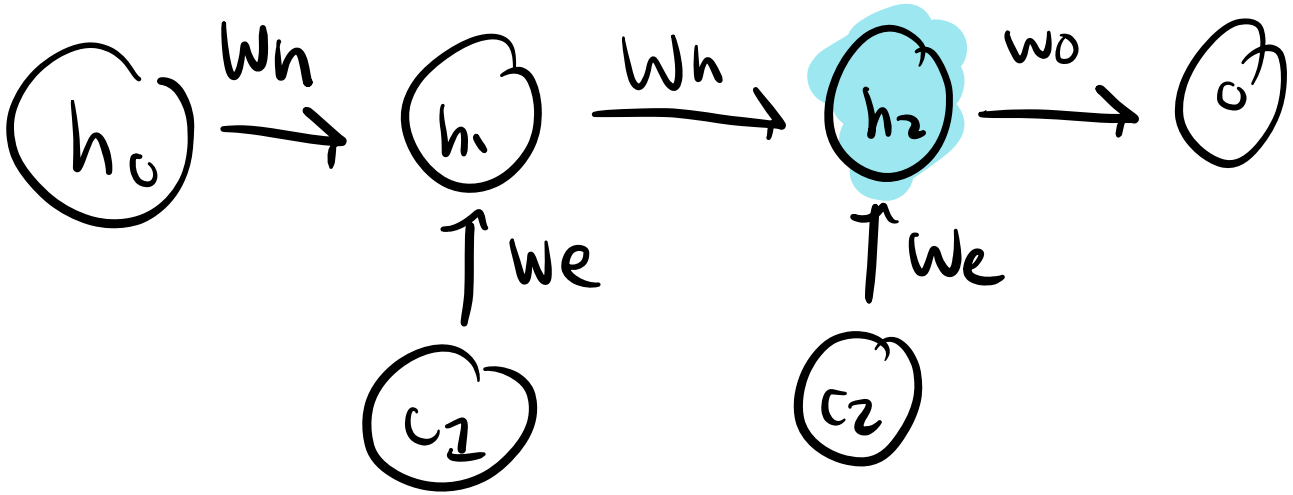
$$\frac{\partial L}{\partial w_e} = \text{time step 2} + \text{time step 1}$$

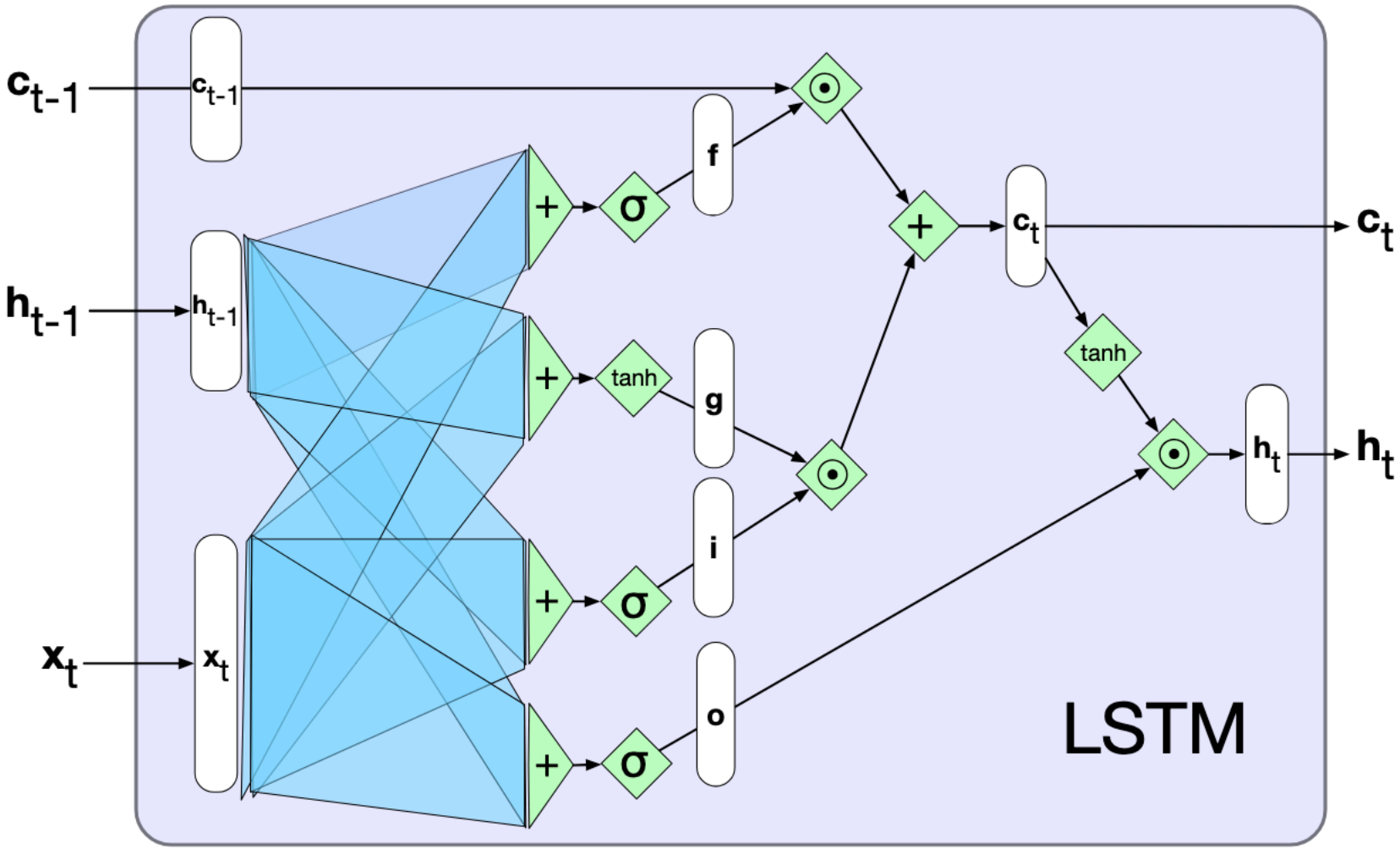
$$\frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial h_2} \cdot \frac{\partial h_2}{\partial a} \cdot \frac{\partial a}{\partial w_e} + \frac{\partial L}{\partial o} \cdot \frac{\partial o}{\partial h_2} \cdot \frac{\partial h_2}{\partial b} \cdot \frac{\partial b}{\partial h_1} \cdot \frac{\partial h_1}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_e}$$

$$-(y-o)\ w_o\ (1-h_2^2)\ c_2 \qquad -(y-o)\ w_o\ (1-h_2^2)\ w_h\ (1-h_1^2)\ c_1$$

Vanishing Gradients Problem

- It's very easy for the gradients in an RNN to zero out since the dependencies between them are so complex.
- There are very few "pipes" for the information to flow through.





A single LSTM unit displayed as a computation graph.

Inputs: the current input, x, the previous hidden state, h_{t-1}, and the previous context, c_{t-1}.

Outputs: a new hidden state, h_t and an updated context, c_t.

Components:

- add gate/input gate (i): selects the information to add to the current context
- forget gate (f): delete information from the context that is no longer needed
- output gate (o): decides what information is required for the current hidden state

RNN Advantages and Limitations

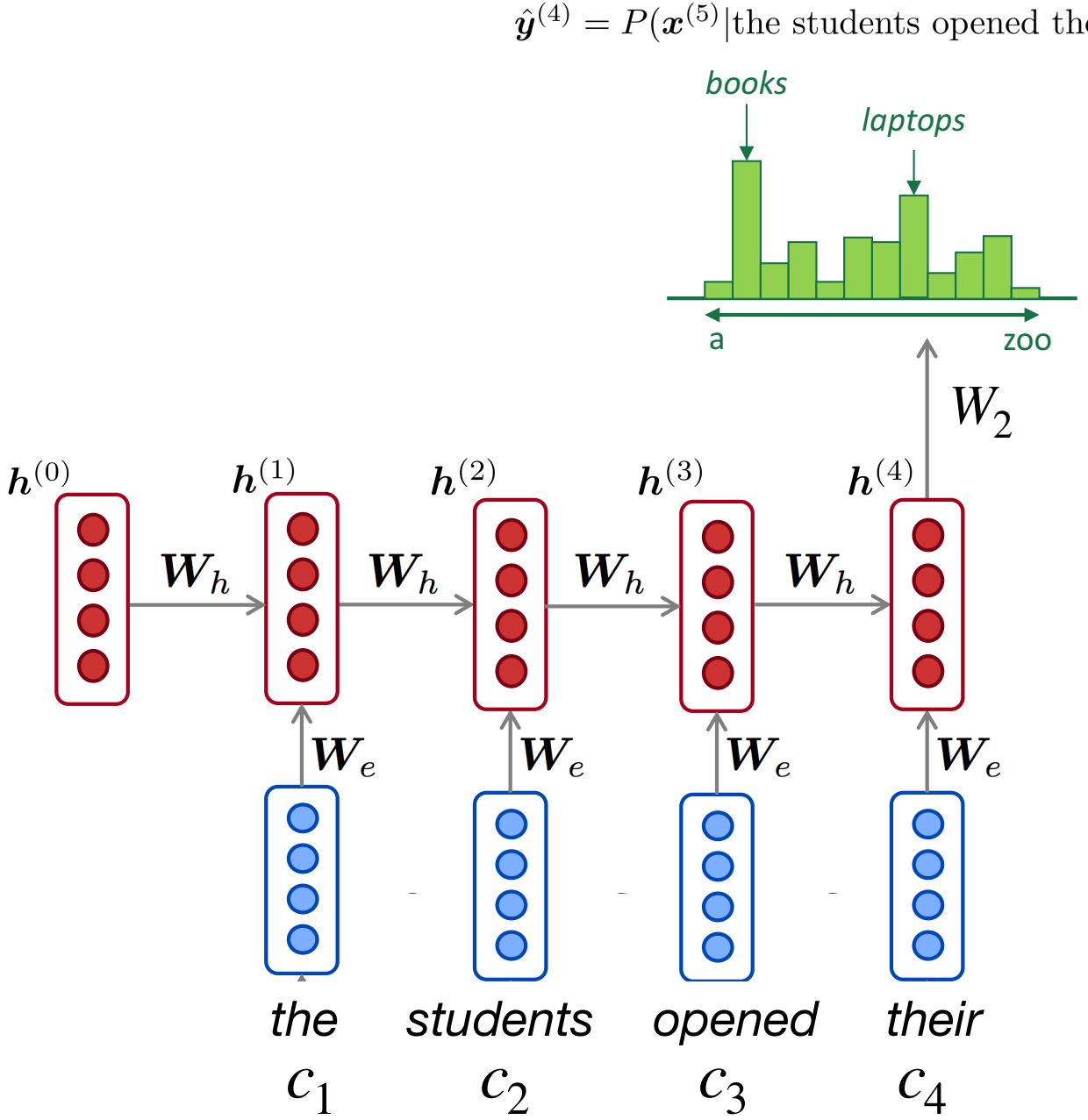
why is this good?

RNN **Advantages**:

- Can process any length input
- Model size doesn't increase for longer input
- Computation for step *t* can (in theory) use information from many steps back
- Weights are shared across timesteps → representations are shared

RNN **Disadvantages**:

- Recurrent computation is slow
- In practice, difficult to access information from many steps back

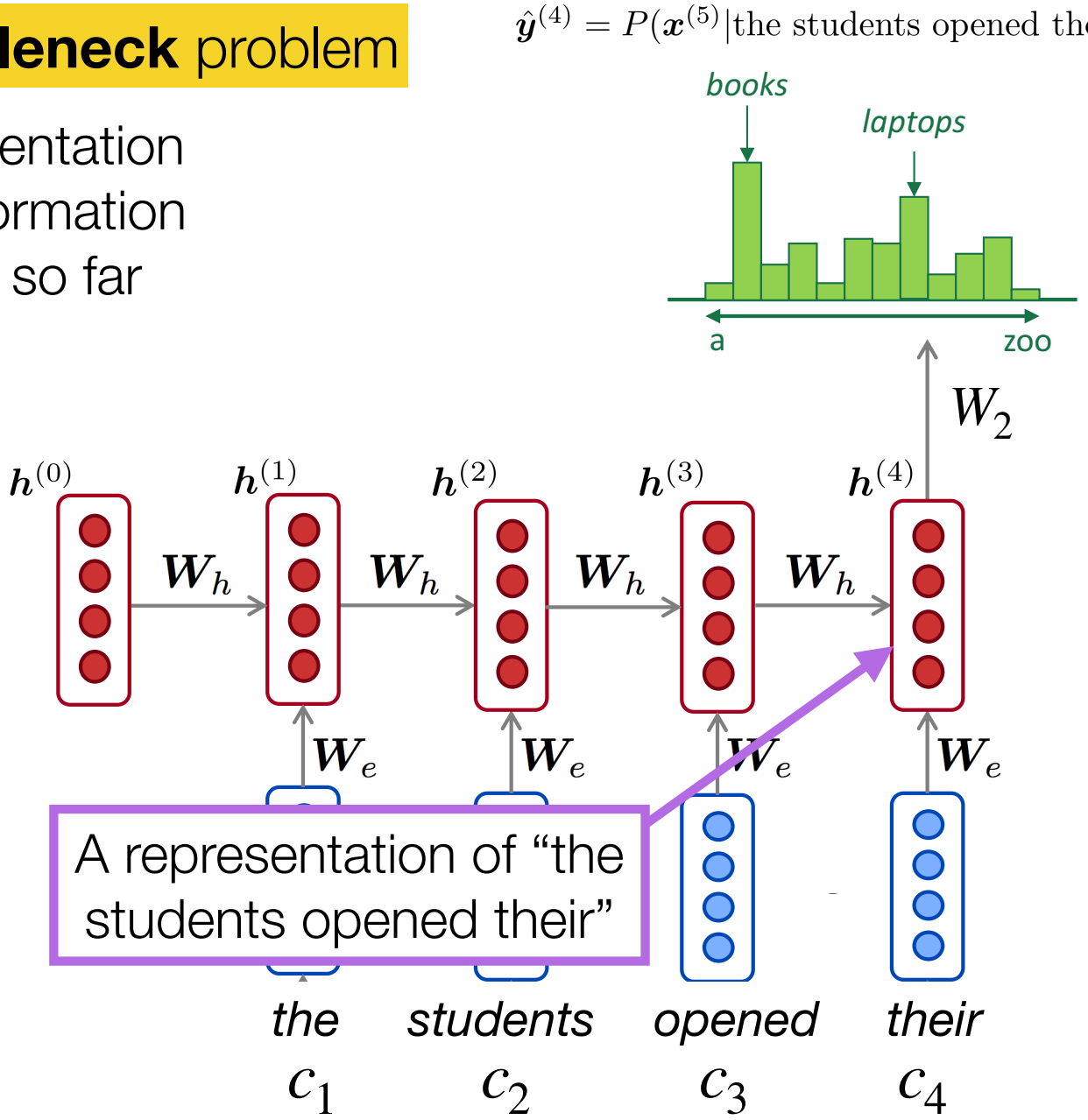


Slides adapted from Mohit Iyyer

RNNs suffer from a **bottleneck** problem

The current hidden representation must encode all of the information about the text observed so far

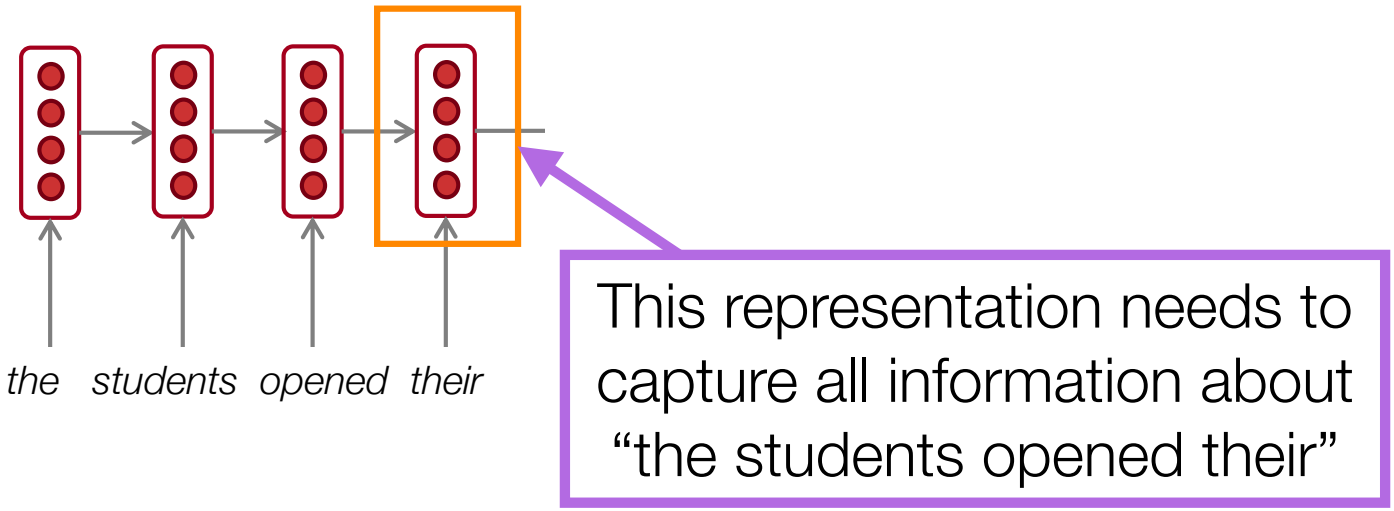
This becomes difficult especially with longer sequences



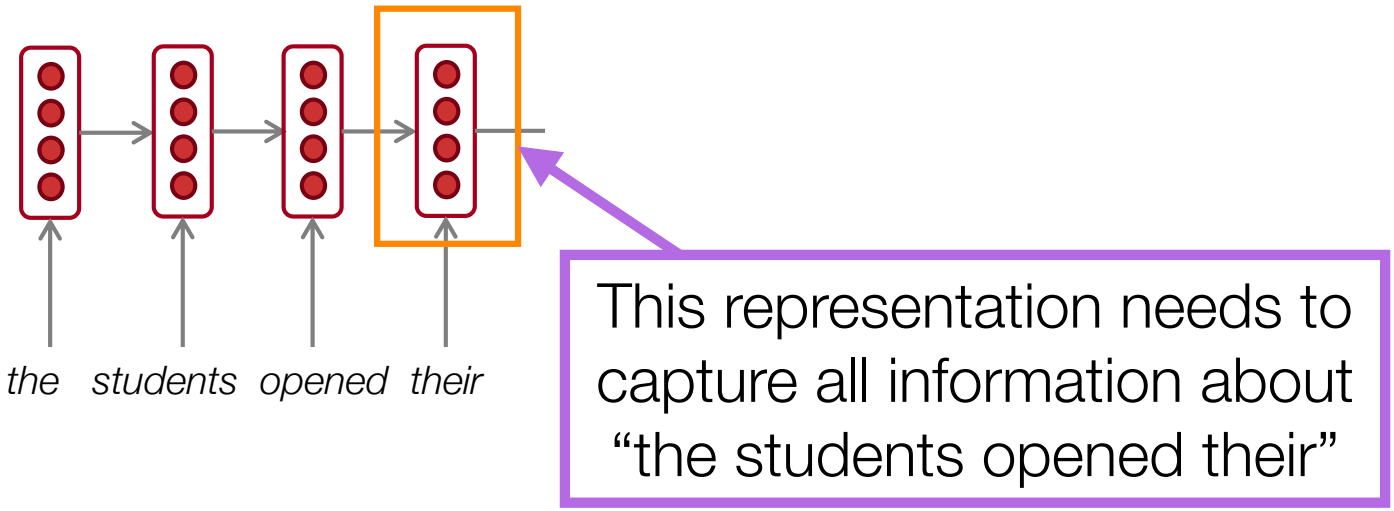
“you can’t cram the meaning of a whole %&@#&ing sentence into a single $*(&@ing vector!”

— Ray Mooney (NLP professor at UT Austin)

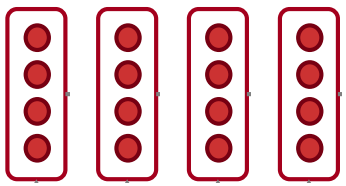
idea: what if we use multiple vectors?



idea: what if we use multiple vectors?



Instead of this, let's try:

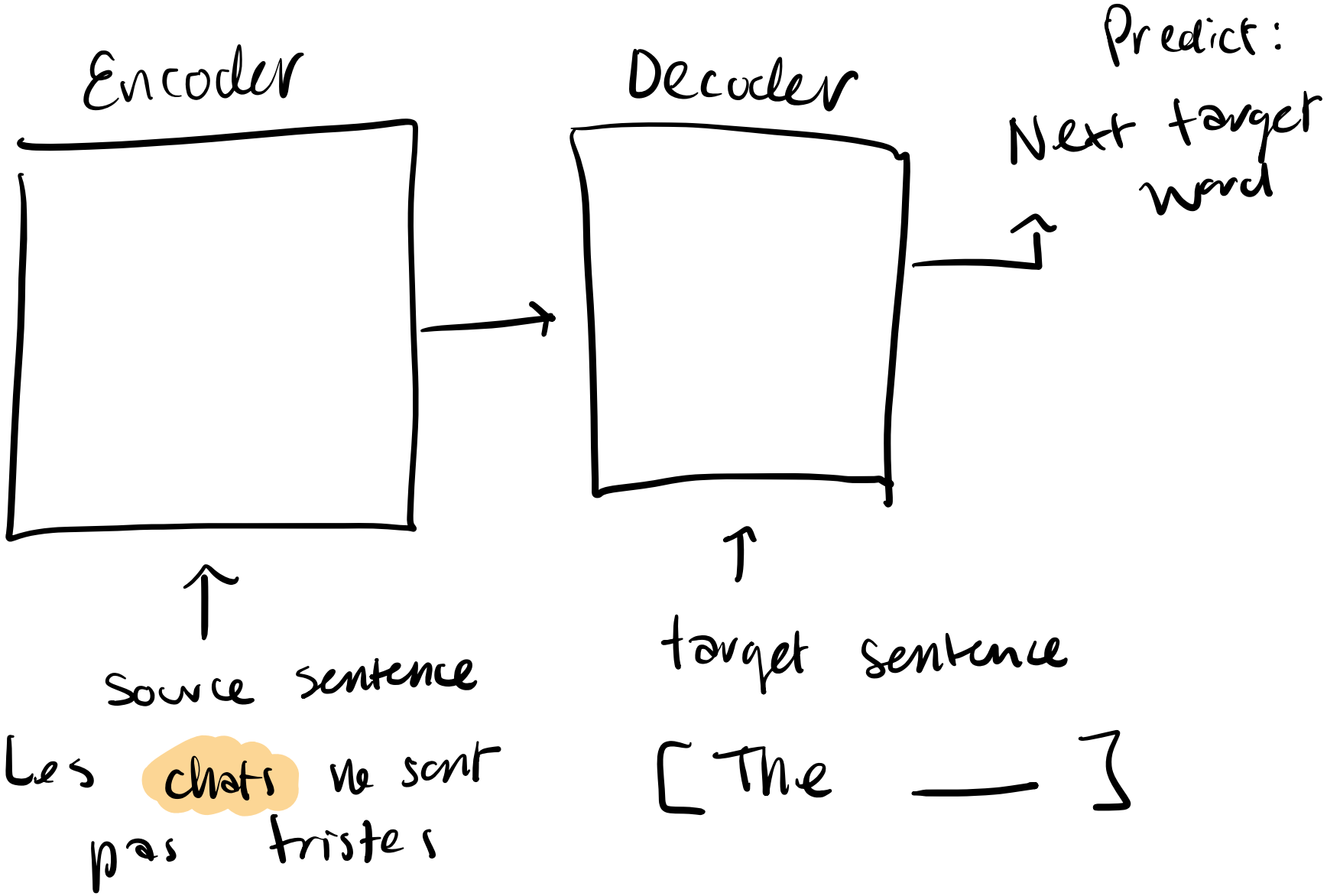
the students opened their =  **(all 4 hidden states!)**

The solution: **attention**

- **Attention mechanisms** (Bahdanau et al., 2015) allow language models to focus on a particular part of the observed context at each time step
  - Originally developed for machine translation, and intuitively similar to *word alignments* between different languages

Slides adapted from Mohit Iyyer

Encoder-Decoder Models of Machine Translation



Attention

How does it work?

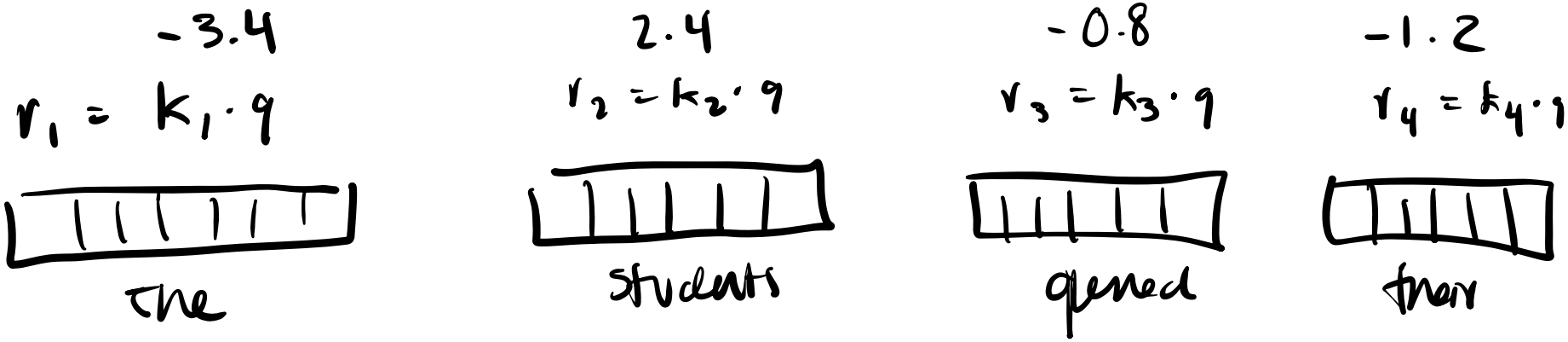
- in general, we have a single *query* vector and multiple *key* vectors. We want to score each query-key pair

in a neural language model, what are the queries and keys?

What Is Attention?

Query: as a task-specific vector that is "searching" for important things from the past context.

Step 1) Measure the similarity (dot product) between query & key

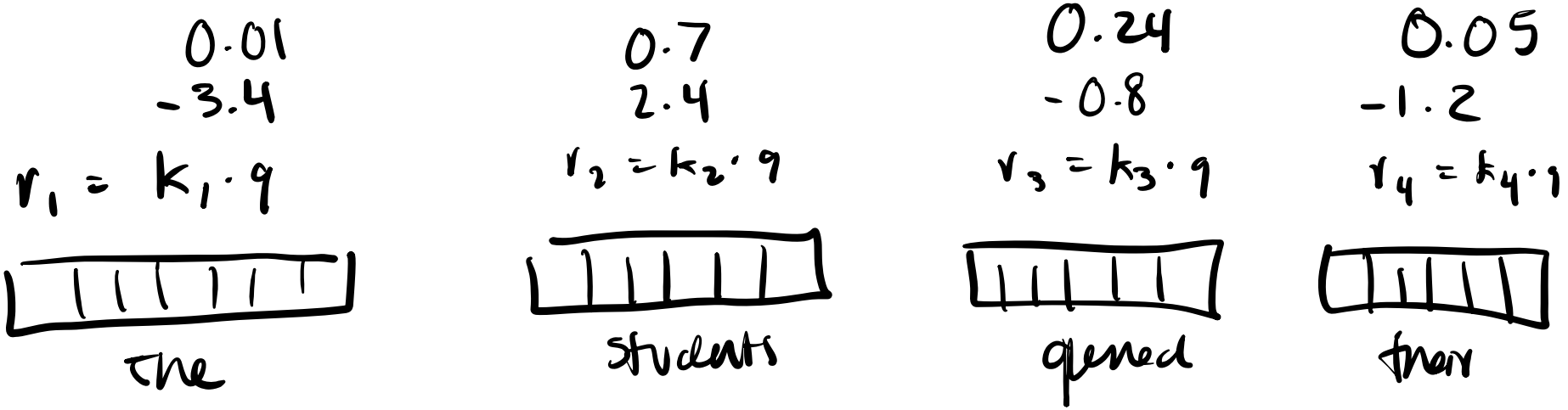


Keys: representations of past context

What Is Attention?

Query: as a task-specific vector that is "searching" for important things from the past context.

Step 2: softmax Scale scores to between 0-1



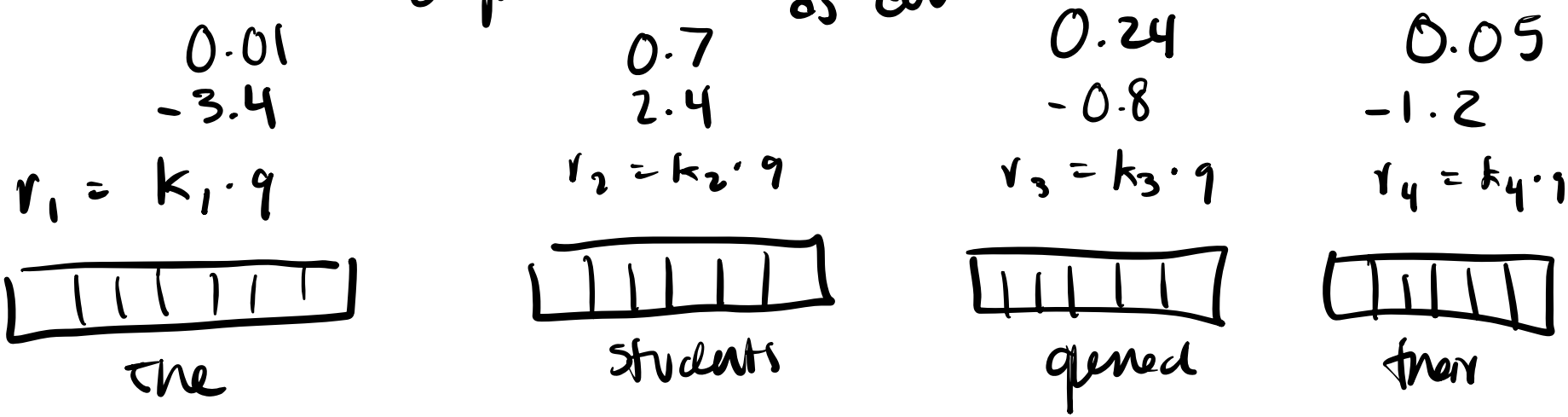
Keys: representations of past context

What Is Attention?

Query: as a task-specific vector that is "searching" for important things from the past context.

Step 3: Compute a weighted average of embeddings

Output: a vector of the same dimension as our word embedding.



Keys: representations of past context


Vicki
@vboykis
They don’t tell you this in the paper (well they do but you have to read it like 15 times)
Multiplying
a lot of vectors
a lot of times
with scaled softmax
Attention
imgflip.com
6:20 PM · Feb 22, 2023 · 88.1K Views