Carnegie Mellon
School of Computer Science

Language Grounding to Vision and Control

Introduction

Katerina Fragkiadaki

Course logistics

- This is a seminar course. There will be no homework.
- Prerequisites: Machine Learning, Deep Learning, Computer Vision, Basic Natural Language Processing (and their prerequisites, e.g., Linear Algebra, Probability, Optimization).
- Each student presents 2-3 papers per semester. Please add your name in that doc: https://docs.google.com/document/d/1JNd4HS-RxR_hVZ3egUtx6xelqLiMQTgA1cEB43Mkyac/edit?usp=sharing. Next, you will be added to a doc with list of papers. Please add your name next to the paper you wish to present in the shared doc. You may add a paper of your preference in the list. FIFS. Papers with no volunteers will be either discarded or presented briefly in the introductory overview in each course.
- Final project: An implementation of language grounding in images/videos/ simulated worlds and/or agent actions, with the dataset/supervision setup of your choice. There will be help on the project during office hours.

Overview

- Goal of our work life
- What is language grounding
- What NLP has achieved w/o explicit grounding (supervised neural models for reading comprehension, syntactic parsing etc.)+ quick overview of basic neural architectures that involve text
- Neural models VS child models
- Theories of simulation/imagination for language grounding
- What is the problem with current vision-language models?

Goal of our work life

- To solve AI: build systems that can see, understand human language, and act in order to perform tasks that are useful.

- Task examples: book appointments/flights, send emails, question answering, description of a visual scene, summarization of activity from NEST home camera, holding a coherent situated dialogue etc.

- **Q:** Is it that Language Understanding is harder than Visual Understanding and thus should be studied after Visual Understanding is mastered?

  - Potentially no. NLP and vision can go hand in hand. In fact, language has tremendously helped Visual Understanding already. Rather than easy or hard senses (vision, NLP etc), there are easy and hard examples within each: e.g., detecting/understanding nouns is EASIER than detecting/ understanding complicated noun phrases or verbal phrases. Indeed, Imagenet classification challenge is a great example of very successful object label grounding.

How language helps action/behavior learning

Many animals can be **trained** to perform novel tasks. E.g., monkeys can be trained to harvest coconuts; after training, they climb on trees and spin them till they fall off.

Training is a torturous process: they are trained by imitation and trial and error, through reward and punishment.

The hardest part is conveying the goal of the activity

Language can express a novel goal effortlessly and succinctly!

Consider the simple routine of looking both ways when crossing a busy street —a domain ill suited to trial and error learning. In humans, the objective can be programmed with a few simple words (“Look both ways before crossing the street”).

How language helps action/behavior learning

*``Many animals can be **trained** to perform novel tasks. People, too, can be trained, but sometime in early childhood people transition from being trainable to something qualitatively more powerful—being **programmable**. …available evidence suggests that facilitating or even enabling this programmability is the learning and use of language."*

How language programs the mind, Lupyan and Bergen

How language helps Computer Vision

- **Explanation based learning**: For a complex new concept, e.g., burglary, instead of collecting a lot of positive and negative examples and training concept classifier, as purely statistical models do, we can define it based on simpler concepts (explanations) that are already grounded.

- E.g., ``*a burglary involves entering from smashed window, the person often wears a mask and tries to take valuable things from the house, e.g. TV*”

- In Computer Vision, simplified explanations are known as attributes.

What is Language Grounding?

Connecting linguistic symbols to perceptual experiences and actions.

Examples:

- *Sleep* (v)

- *Dog reading newspaper* (NP)

- *Climb on chair to reach lamp (VP)*

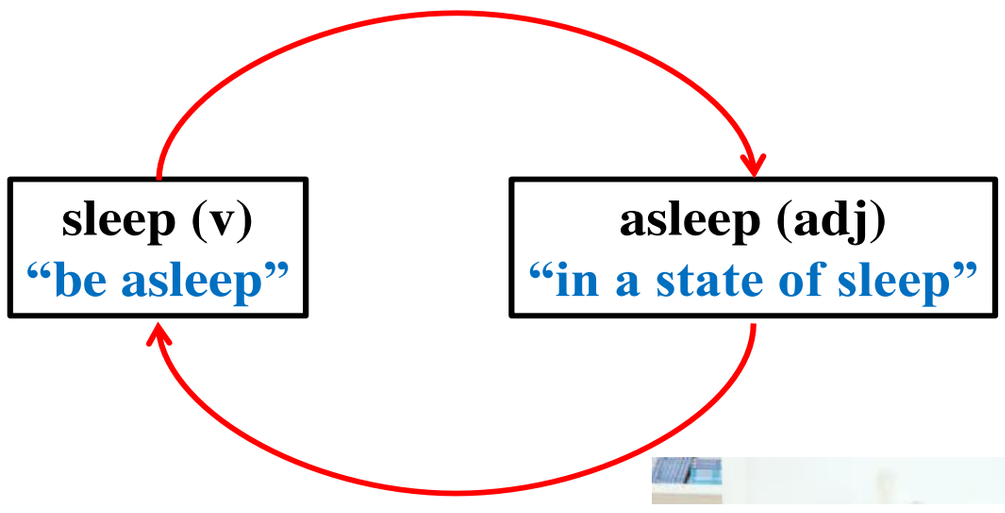
Google didn't find something sensible here, which is why we have the course

What is not Language Grounding?

Not connecting linguistic symbols to perceptual experiences and actions, but rather connecting linguistic symbols to other linguistic symbols.

Example from Wordnet:

- ``Sleep" means ``be asleep"



sleep(n): ``a natural and periodic state of rest during which consciousness of the world is suspended"

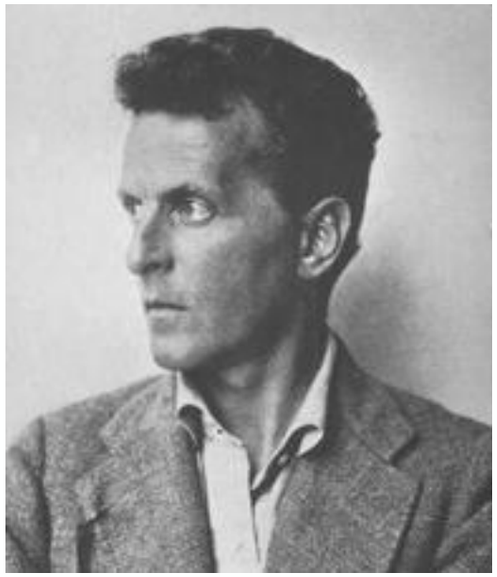
This results in circular definitions

Slide adapted from Raymond Mooney

Historical Roots of Ideas on Language Grounding

Meaning as Use & Language Games

Wittgenstein (1953)



Symbol Grounding

Harnad (1990)

"Without grounding is as if we are trying to learn Chinese using a Chinese-Chinese dictionary"

Slide adapted from Raymond Mooney

Bypassing explicit grounding

Task: Learn Word Vector Representations
(in an unsupervised way) from large text corpora

- Input: the one hot encoding of a word (long sparse vector, as long as the vocabulary size) **hotel = [0 0 0 … 1 … 0]**
- Output: a low dimensional vector **hotel = [0.23 0.45 -2.3 … -1.22]**
- Supervision: No supervision is used, no annotations

Q: Why such low-dim representation is worthwhile?

From Symbolic to Distributed Representations

- Its problem, e.g., for web search
  - If user searches for [*Dell notebook battery size*], we would like to match documents with *"Dell laptop battery capacity"*
  - If user searches for [*Seattle motel*], we would like to match documents containing *"Seattle hotel"*
- But:

  motel [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]^T

  hotel [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0] = 0
  - Our query and document vectors are orthogonal
  - There is no natural notion of similarity in a set of one-hot vectors
- Could deal with similarity separately; instead we explore a direct approach, where vectors encode it.

Slide adapted from Chris Manning

Distributional Similarity Based Representations

You can get a lot of value by representing a word by means of its neighbors:

"You shall know a word by the company it keeps."

(J. R. Firth 1957: 11)

One of the most successful ideas of modern statistical NLP.

government debt problems turning into	banking	crises as has happened in
saying that Europe needs unified	banking	regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Slide adapted from Chris Manning

Word Meaning is Defined in Terms of Vectors

We will build a dense vector for each word type, chosen so that it is good at predicting other words appearing in its context

...those other words also being represented by vectors... it all gets a bit recursive

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Slide adapted from Chris Manning

Basic Idea of Learning Neural Network Word Embeddings

- We define a model that aims to predict between a center word wt and context words in terms of word vectors:

  $$p(\text{context} \mid w_t) = \ldots$$

- which has a loss function, e.g.:

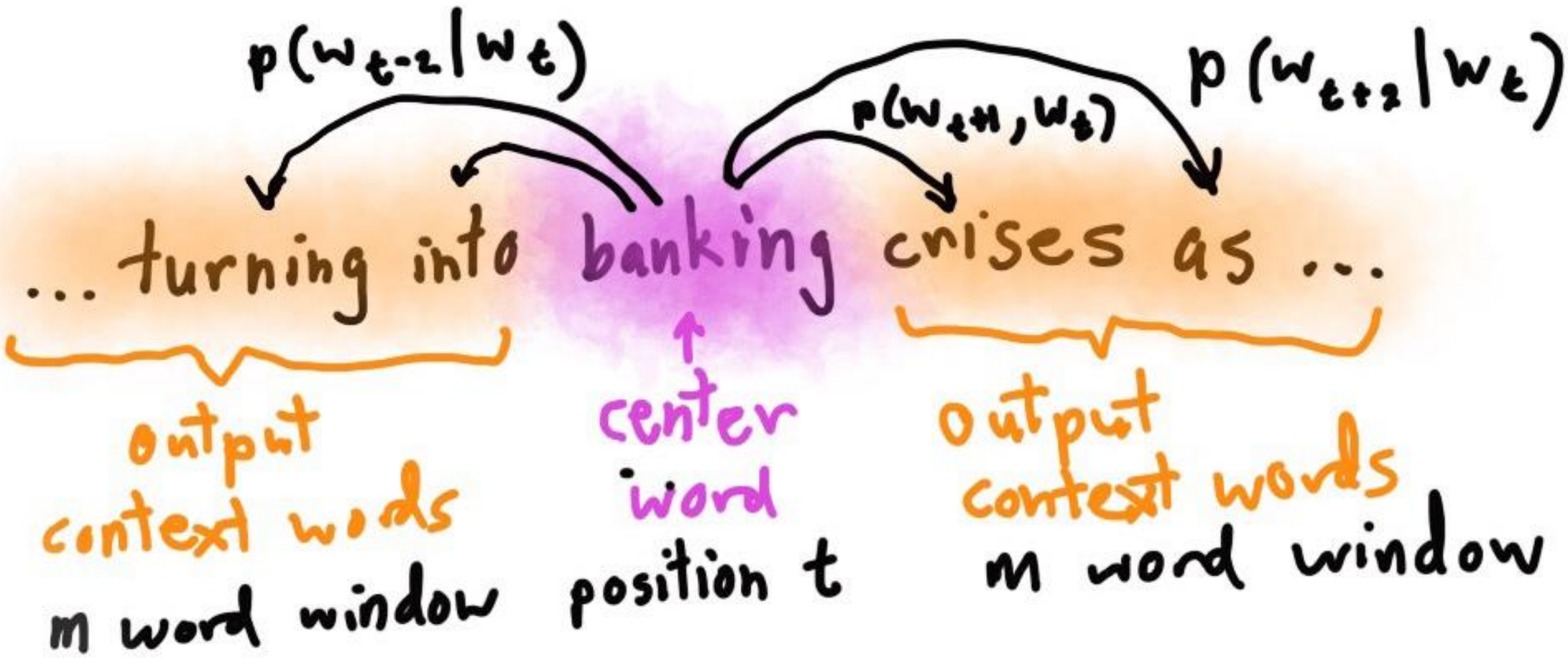
  $$J = 1 - p(w_{-t} \mid w_t)$$

- We look at many positions t in a big language corpus.

- We keep adjusting the vector representations of words to minimize this loss.

Slide adapted from Chris Manning

Skip Gram Predictions



Slide adapted from Chris Manning

Details of word2vec

- For each word t = 1, ... , T, predict surrounding words in a window of "radius" *m* of every word.
- Objective function: Maximize the probability of any context word given the current center word.

$$J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \neq 0}} p(w_{t+j} \mid w_t ; \theta)$$

Negative Log Likelihood

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log p(w_{t+j} \mid w_t)$$

Where theta represents all variables we will optimize

Slide adapted from Chris Manning

Details of word2vec

- Predict surrounding words in a window of radius m of every word
- For $p(w_{t+j} \mid w_t)$ the simplest first formulation is:

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

Where o is the outside (or output) word index, c is the center word index, v_c and u_o are "center" and "outside" vectors of indices c and o

- Softmax using word c to obtain probability of word o

Slide adapted from Chris Manning

Skipgram

$V \times 1$ $d \times V$ $d \times 1$

W

w_t $v_c = W w_t$

one hot word symbol ↑ word

Looks up column of word embedding matrix as representation of center word

$V \times d$

W'

Output word representation

$V \times 1$ $W' v_c = [u_x^T v_c]$ $V \times 1$ $p(x|c) = \text{softmax}(u_x^T v_c)$ $V \times 1$ Truth

softmax → softmax → softmax →

W_{t-3} W_{t-2} W_{t-1}

Actual context words

Softmax

$$p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Slide adapted from Chris Manning

Details of word2vec

- The normalization factor is too computationally expensive.

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

Instead of exhaustive summation in practice we use negative sampling

Slide adapted from Chris Manning

Details of word2vec

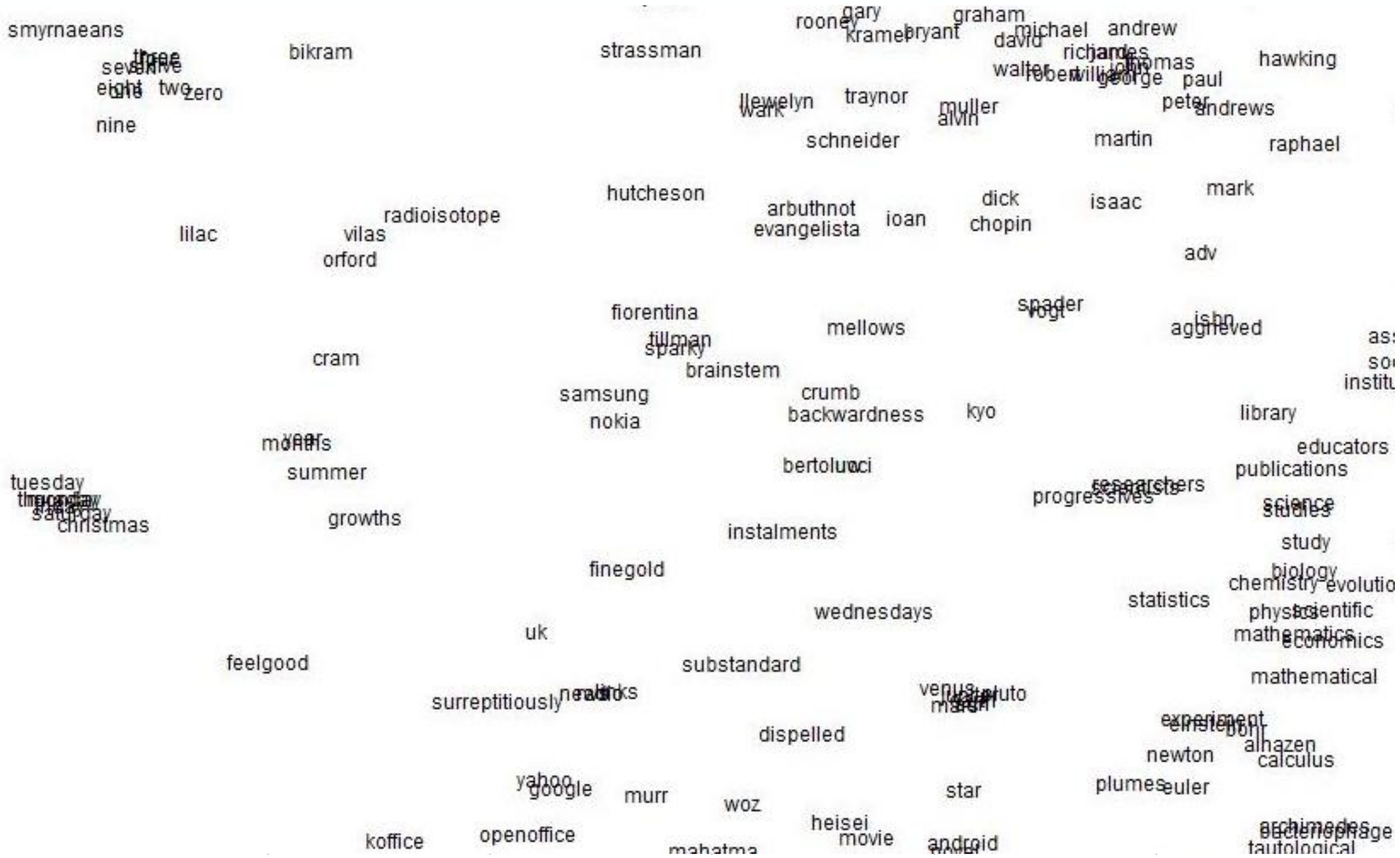
- From paper: "Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al. 2013)
- Overall objective function: $J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J_t(\theta)$

$$J_t(\theta) = \log \sigma\left(u_o^T v_c\right) + \sum_{i=1}^{k} \mathbb{E}_{j\sim P(w)}\left[\log \sigma\left(-u_j^T v_c\right)\right]$$

- P(w): background word probabilities (obtained by counting). We use U^{3/4} to boost probabilities of very infrequent words.

Slide adapted from Chris Manning

word2vec Improves Objective Function by Putting Similar Words Nearby in Space



Slide adapted from Chris Manning

Learning word vectors by counting co-occurrences and SVD

- With a co-occurrence matrix X:
  - Two options: full document vs. windows
  - Word-document co-occurrence matrix will give general topics (all sports teams will have similar entries) leading to "Latent Semantic Analysis)
- Instead: Similar to word2vec, use window around each word --> captures both syntactic (POS) and semantic information

Slide adapted from Richard Socher

Window Based Co-Occurrence Matrix

Example Corpus

- I like deep learning.
- I like NLP.
- I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Slide adapted from Richard Socher

Problems with Simple Co-Occurrence Vectors

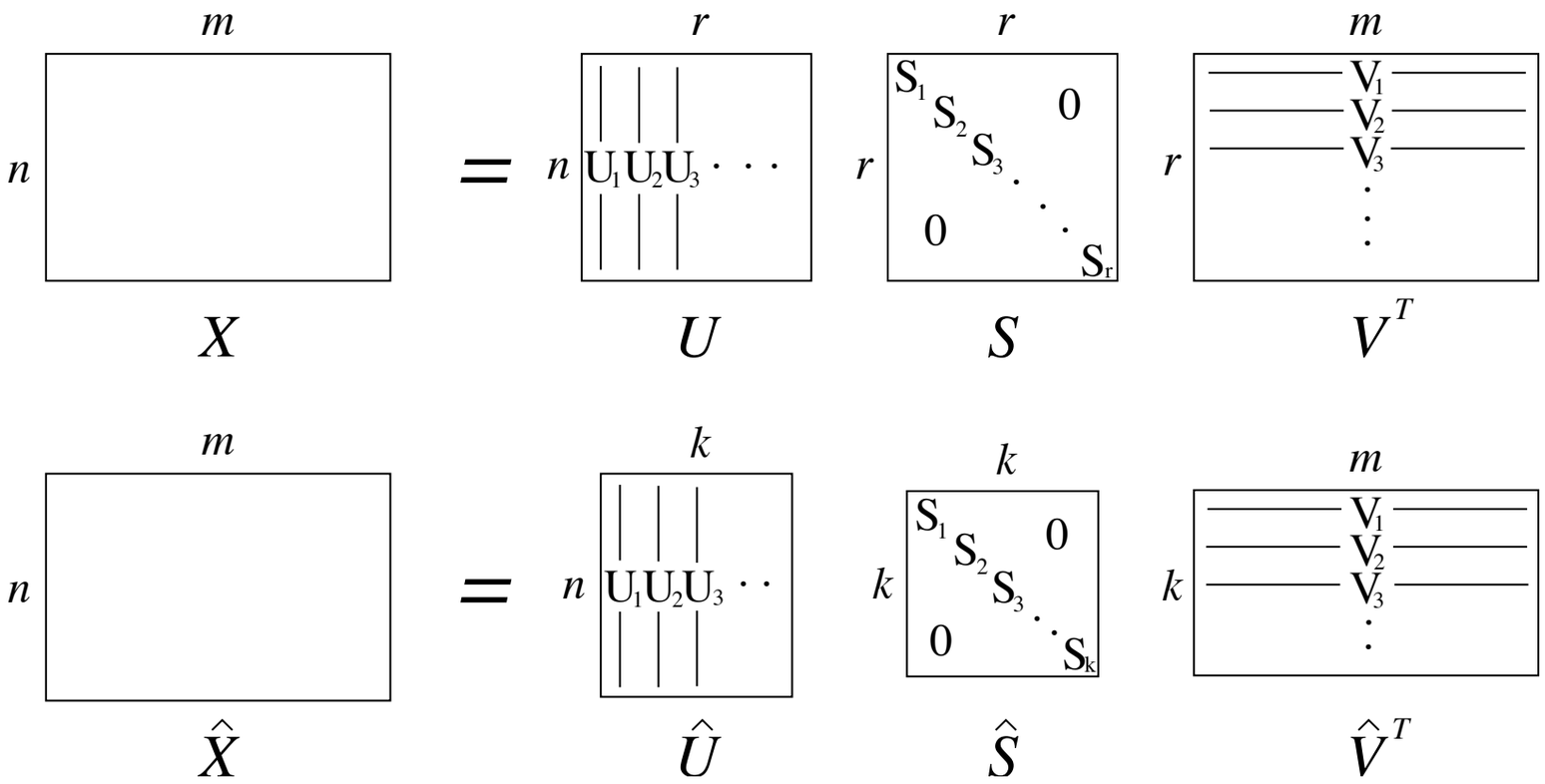
Same problems as one hot word representations:

- Increase in size with vocabulary
- Very high dimensional: require a lot of storage
- Subsequent classification models have sparsity issues
- Models are less robust

Slide adapted from Richard Socher

Reduce Dimensionality

Singular Value Decomposition of co-occurrence matrix *X*.

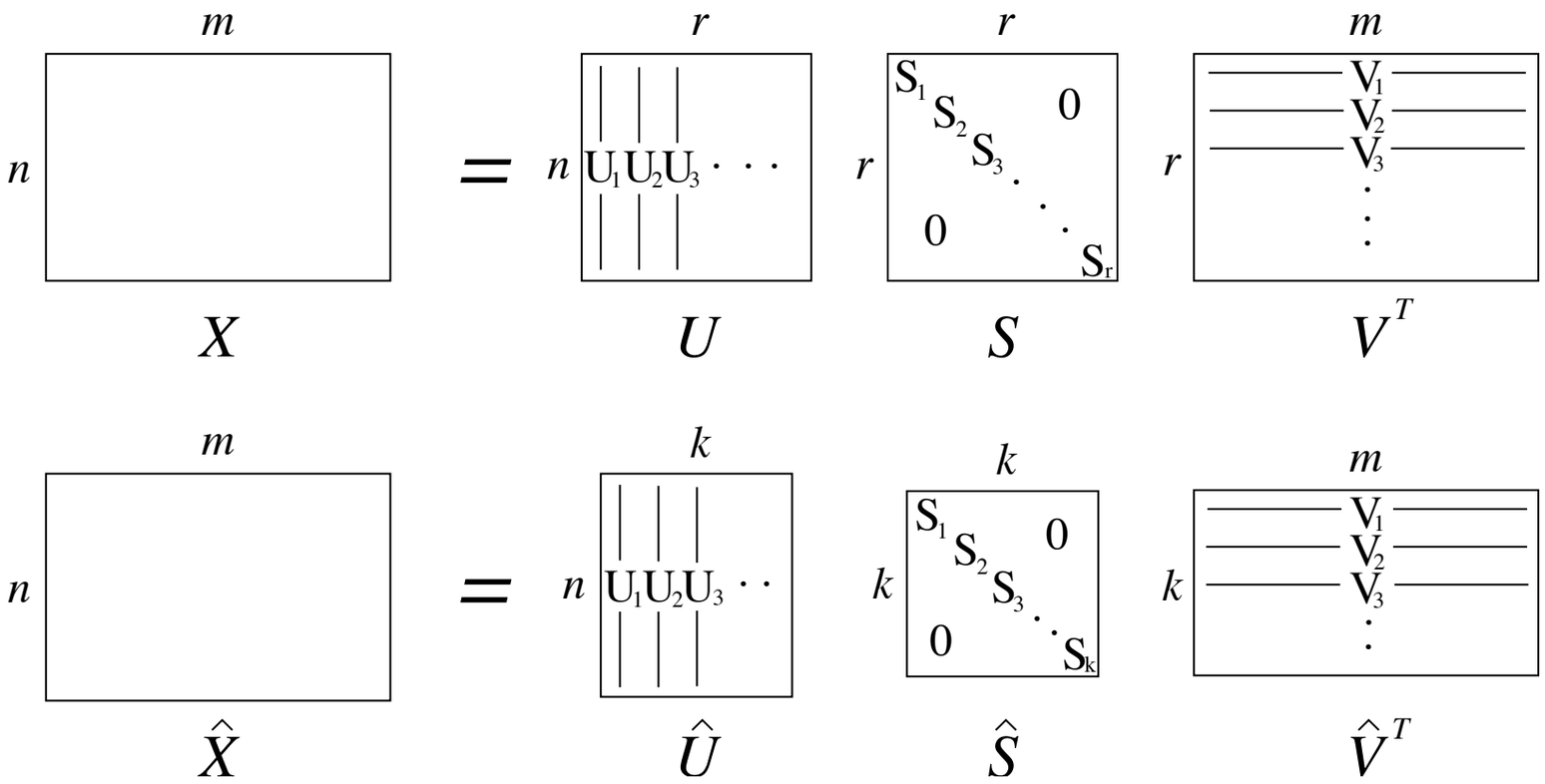


$\hat{X}$ is the best rank *k* approximation to *X* , in terms of least squares.

Slide adapted from Richard Socher

Reduce Dimensionality

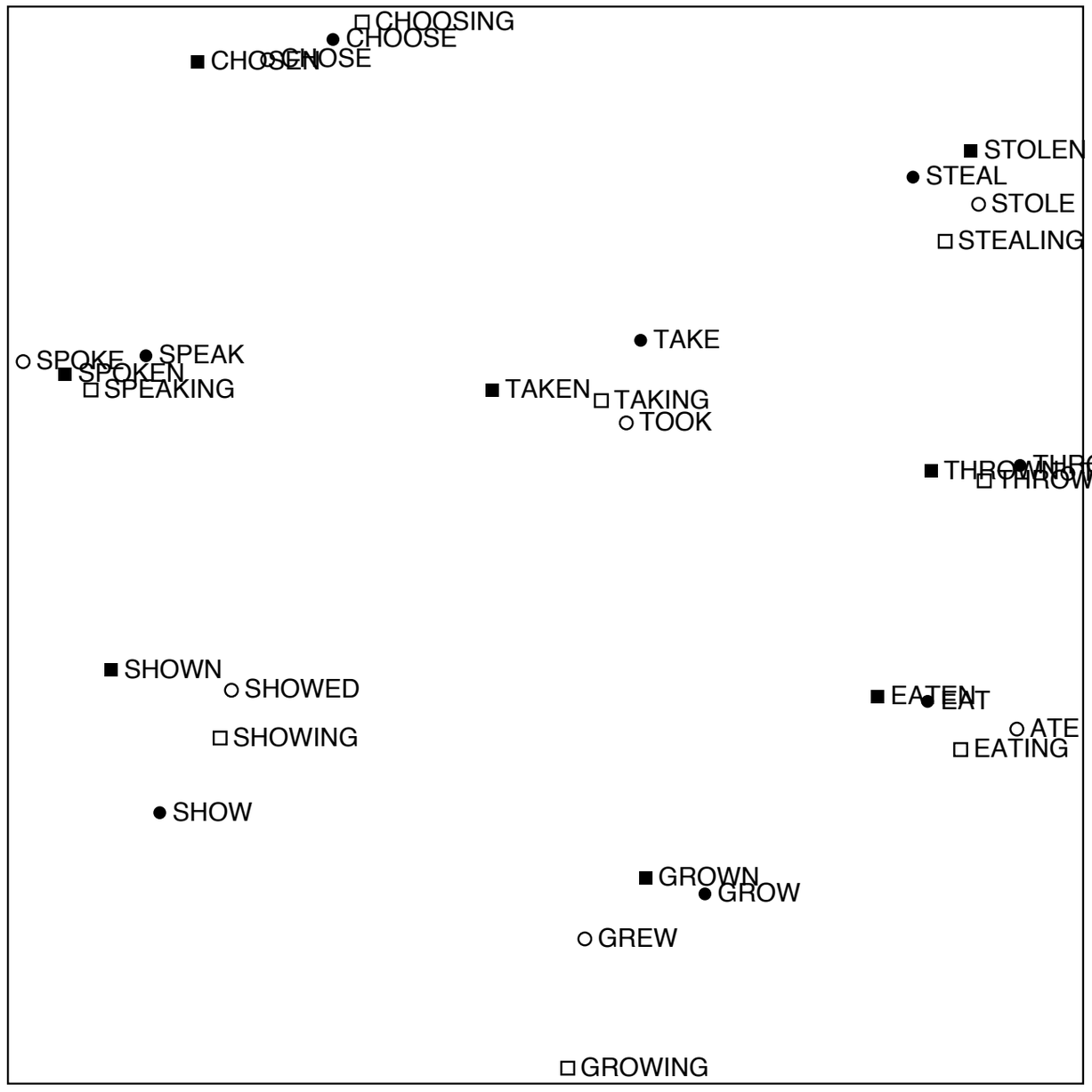
Singular Value Decomposition of co-occurrence matrix *X*.



$\hat{X}$ is the best rank *k* approximation to *X* , in terms of least squares.

Slide adapted from Richard Socher

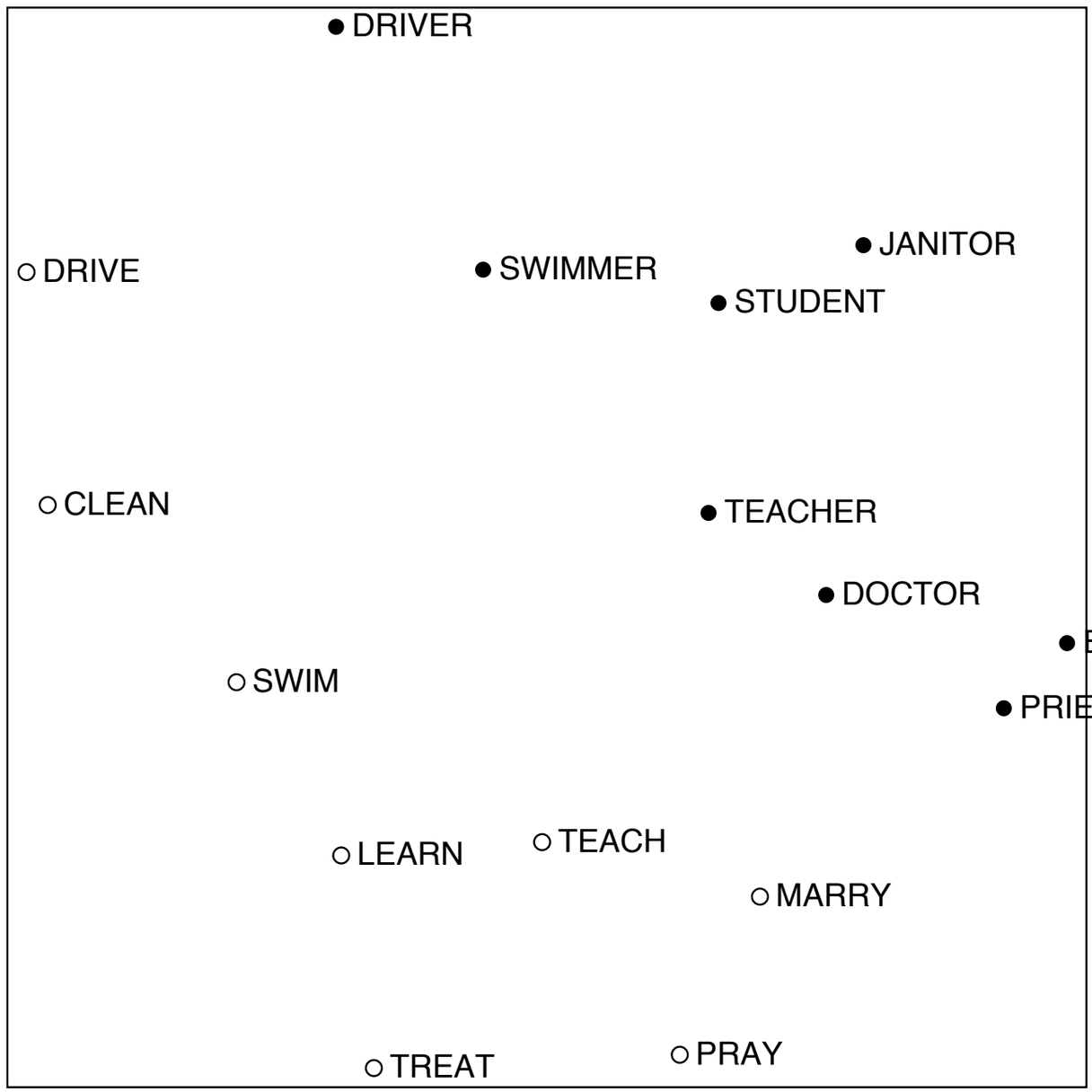
Interesting Semantic Patterns Emerge in the Vectors



An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence
Rohde et al. 2005

Slide adapted from Richard Socher

Interesting Semantic Patterns Emerge in the Vectors



An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence
Rohde et al. 2005

Slide adapted from Richard Socher

Interesting Semantic Patterns Emerge in the Vectors

Nearest words to frog:

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

litoria

leptodactylidae

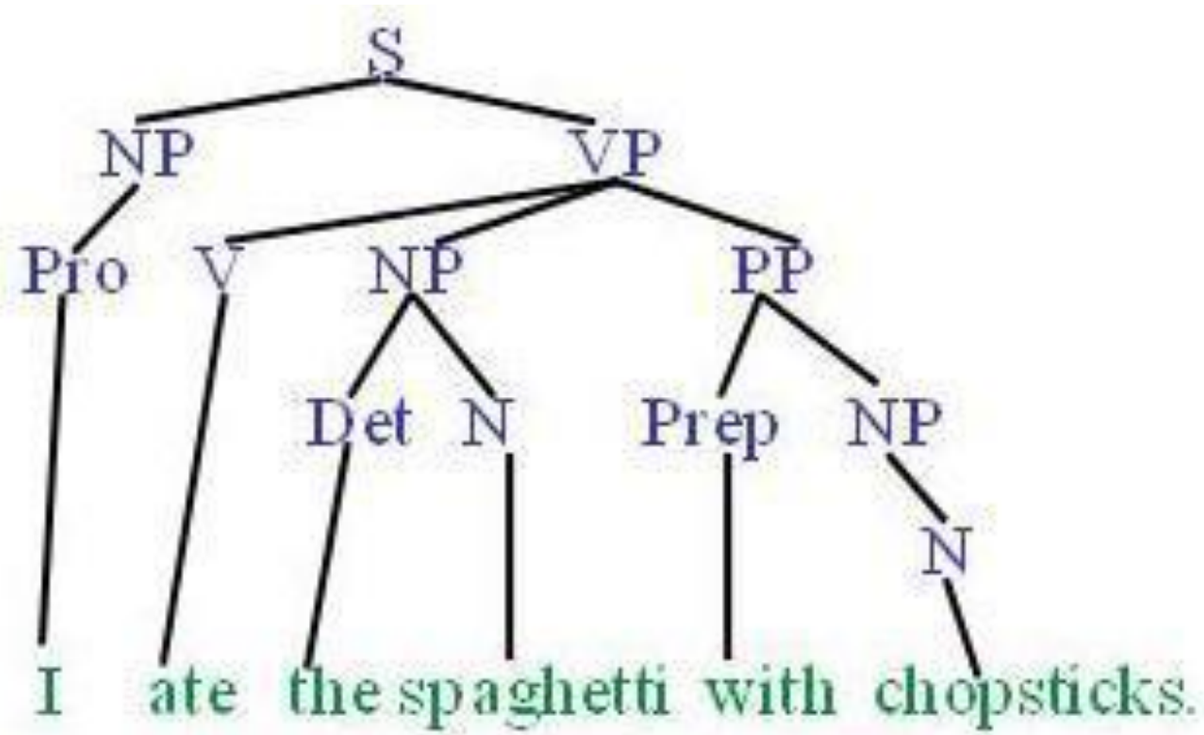
rana

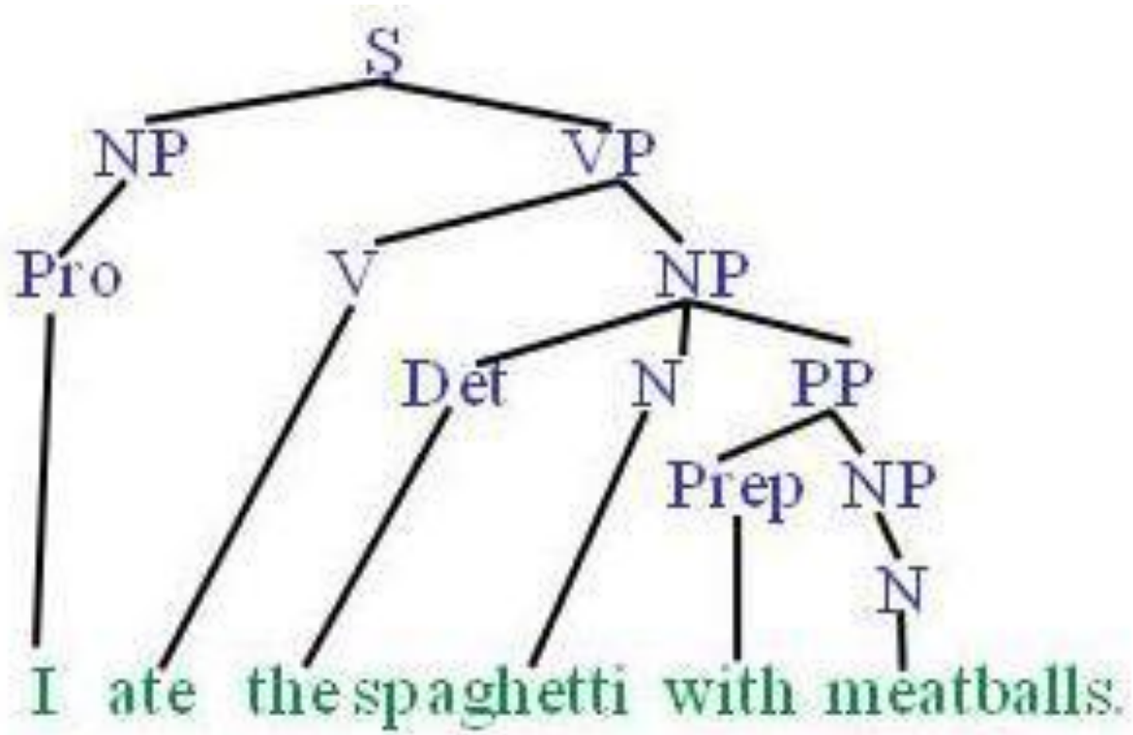
eleutherodactylus

Slide adapted from Richard Socher

Bypassing explicit grounding

Task: Generate the correct syntactic tree of a sentence





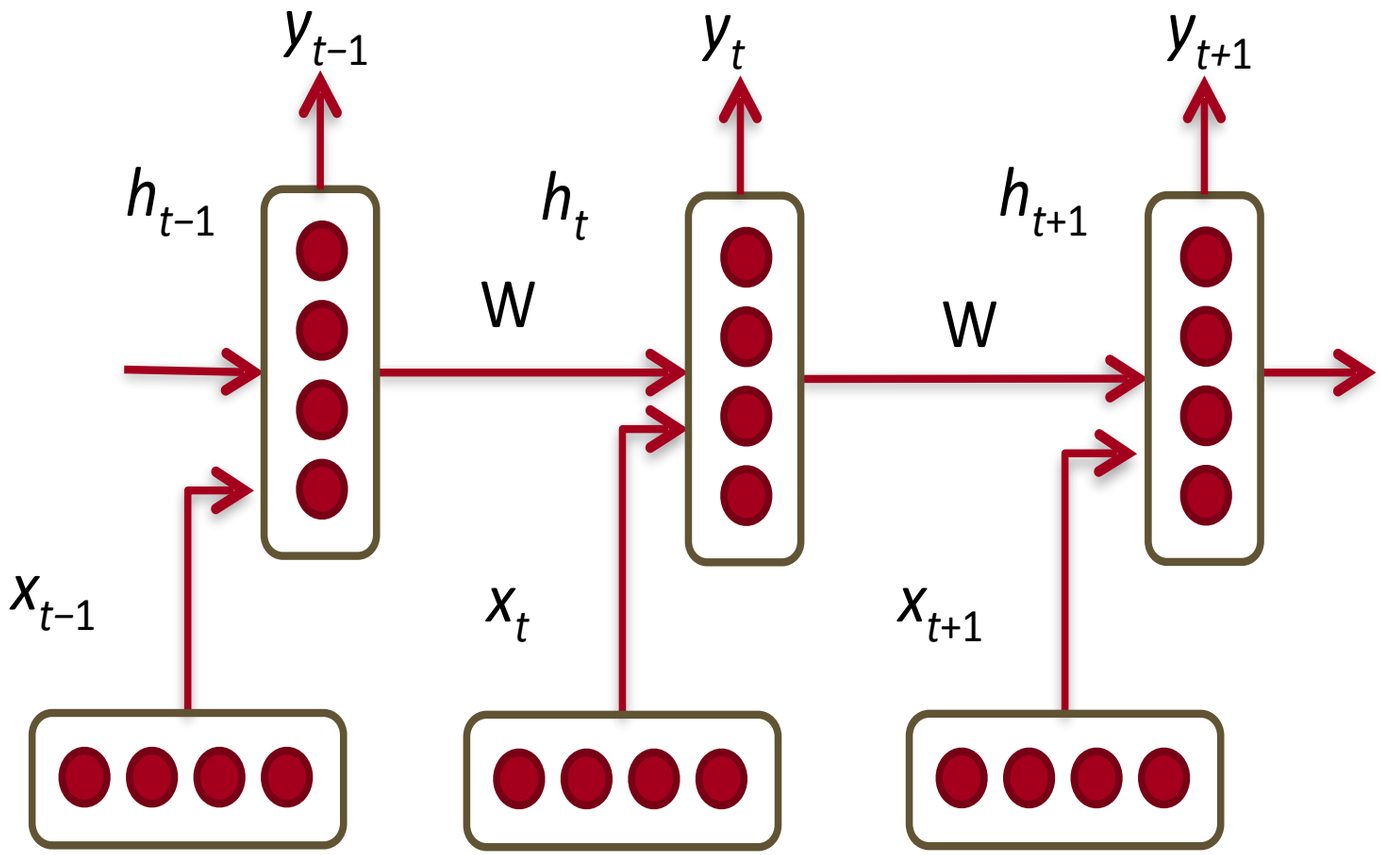
Bypassing explicit grounding

Task: Generate the correct syntactic tree of a sentence

- Input: A sentence
- Output: The syntactic tree
- Supervision: Large scale corpora human annotated with syntactic parse trees, e.g., Penn Treebank
- Model examples: Neural syntactic parsers, e.g., Grammar as a Foreign Language, where an attention based seq-to-seq model maps a sentence to each syntactic tree, expressed in a DFS format

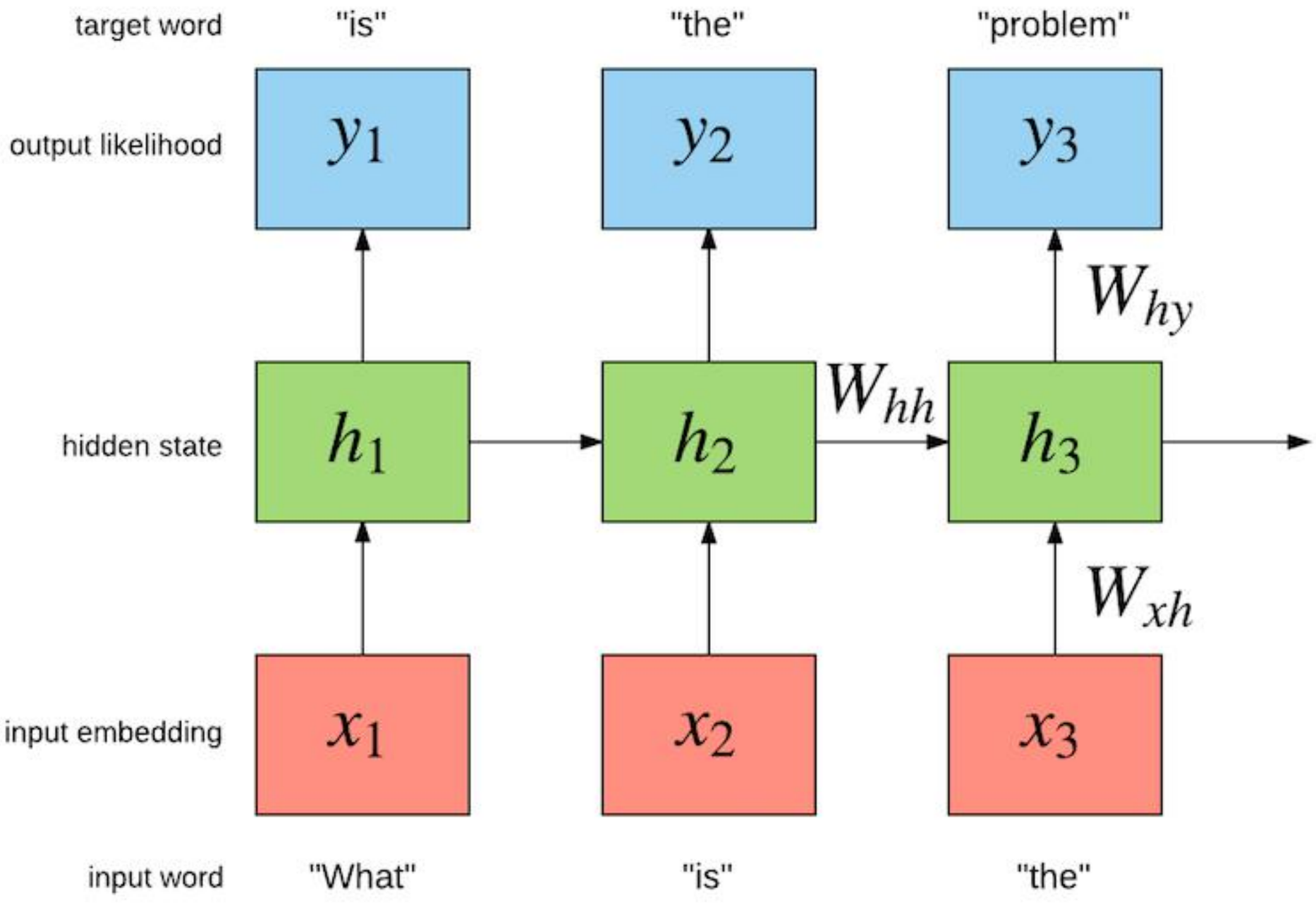
Recurrent Neural Networks!

- RNNs tie the weights at each time step
- Condition the neural network on all previous words



Slide adapted from Richard Socher

RNN language model



Recurrent Neural Network Language Model

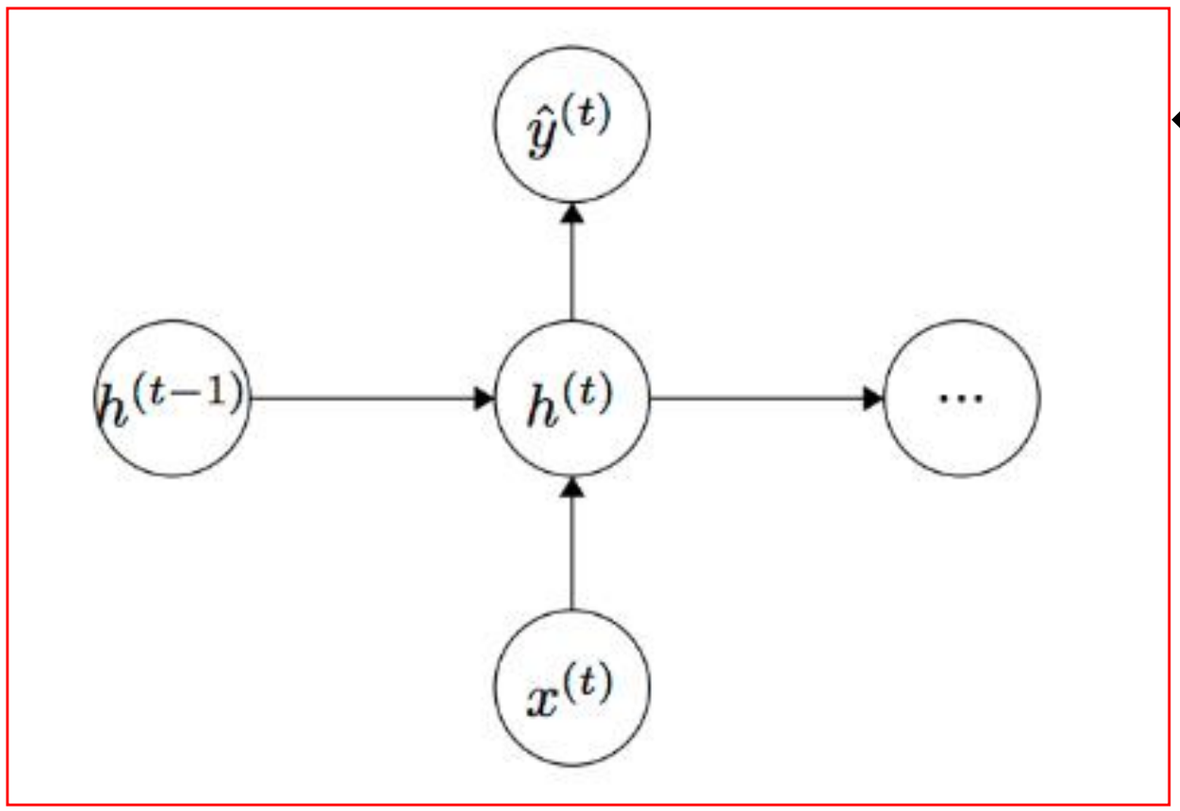
Given list of word **vectors**: $x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$

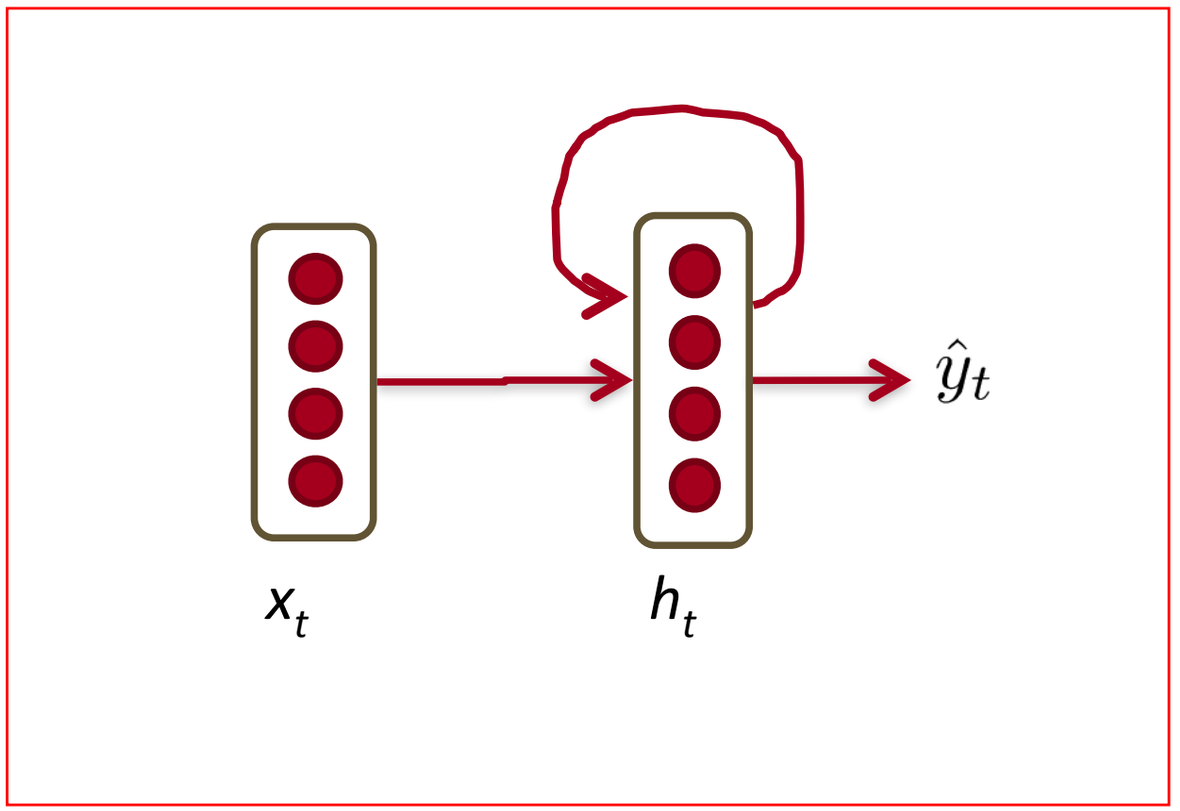
At a single time step:

$$h_t = \sigma\left(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]}\right)$$

$$\hat{y}_t = \text{softmax}\left(W^{(S)} h_t\right)$$

$$\hat{P}(x_{t+1} = v_j \mid x_t, \ldots, x_1) = \hat{y}_{t,j}$$





Slide adapted from Richard Socher

Recurrent Neural Network Language Model

Main idea: we use the same set of W weights at all time steps!

Everything else is the same:

$$h_t = \sigma\left(W^{(hh)}h_{t-1} + W^{(hx)}x_{[t]}\right)$$

$$\hat{y}_t = \text{softmax}\left(W^{(S)}h_t\right)$$

$$\hat{P}(x_{t+1} = v_j \mid x_t, \ldots, x_1) = \hat{y}_{t,j}$$

$h_0 \in \mathbb{R}^{D_h}$ is some initialization vector for the hidden layer at time step 0

$x_{[t]}$ is the column vector of L at index [t] at time step t

$W^{(hh)} \in \mathbb{R}^{D_h \times D_h}$ $W^{(hx)} \in \mathbb{R}^{D_h \times d}$ $W^{(S)} \in \mathbb{R}^{|V| \times D_h}$

Slide adapted from Richard Socher

Recurrent Neural Network Language Model

$\hat{y} \in \mathbb{R}^{|V|}$ is a probability distribution over the vocabulary

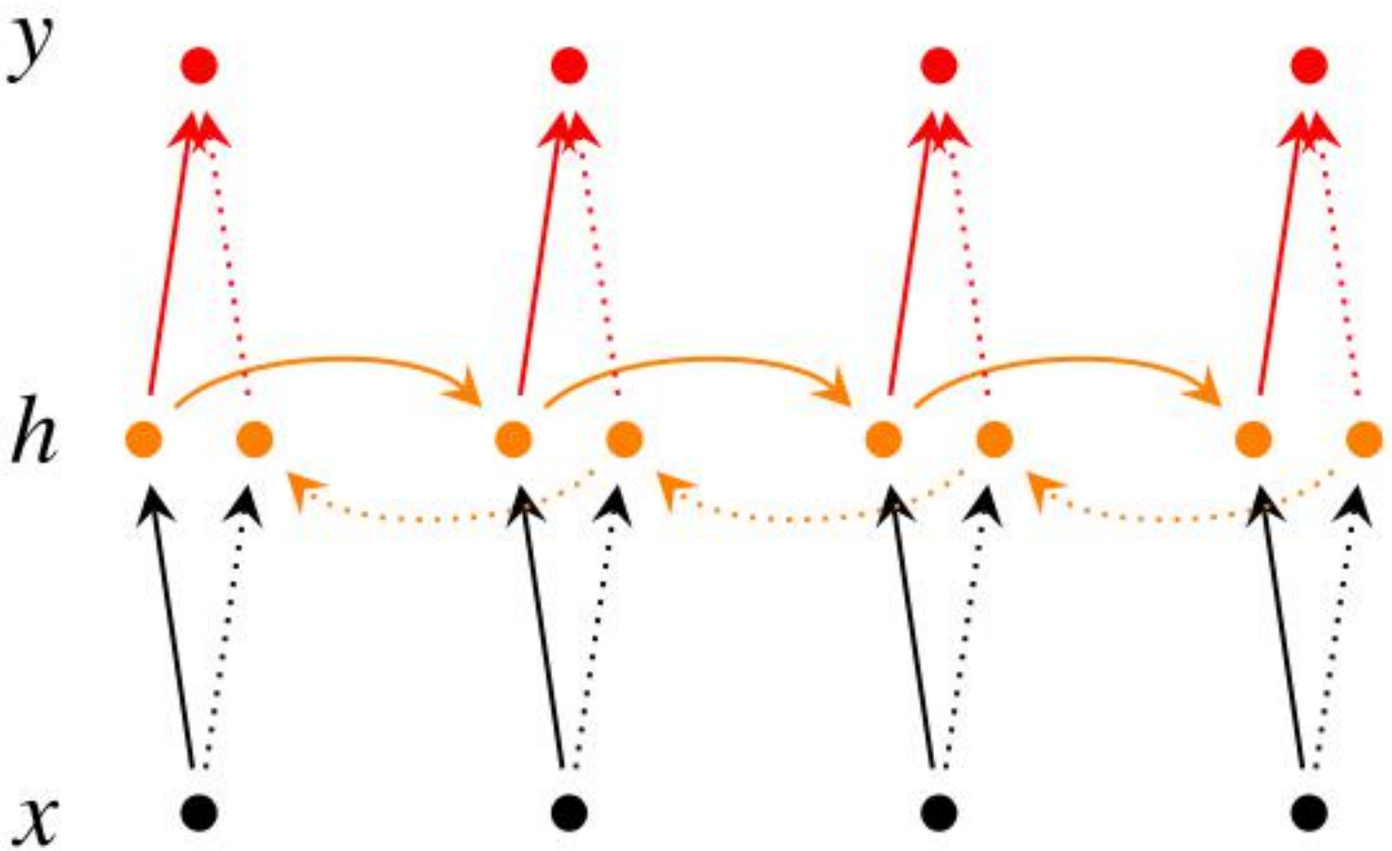
Same cross entropy loss function but predicting words instead of classes

$$J^{(t)}(\theta) = -\sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

Slide adapted from Richard Socher

Bidirectional RNNs

Problem: For classification you want to incorporate information from words both preceding and following



$$\vec{h}_t = f(\vec{W}x_t + \vec{V}\vec{h}_{t-1} + \vec{b})$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t+1} + \overleftarrow{b})$$

$$y_t = g(U[\vec{h}_t; \overleftarrow{h}_t] + c)$$

$h = [\vec{h}; \overleftarrow{h}]$ now represents (summarizes) the past and future around a single token.

Slide adapted from Richard Socher

GRUs

- Standard RNN computes hidden layer at next time step directly: $$h_t = f\left(W^{(hh)}h_{t-1} + W^{(hx)}x_t\right)$$
- GRU first computes an update **gate** (another layer) based on current input word vector and hidden state

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$

- Compute reset gate similarly but with different weights

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$

Slide adapted from Richard Socher

GRUs

- Update gate $z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$
- Reset gate $r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$
- New memory content: $\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$

  If reset gate unit is ~0, then this ignores previous memory and only stores the new word information

- Final memory at time step combines current and previous time steps: $h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$

Slide adapted from Richard Socher

GRU Intuition

- If reset is close to 0, ignore previous hidden state
  → Allows model to drop information that is irrelevant in the future

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$
$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$
$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

- Update gate z controls how much of past state should matter now.
  - If z close to 1, then we can copy information in that unit through many time steps! **Less vanishing gradient**!
- Units with short-term dependencies often have reset gates very active

Slide adapted from Richard Socher

Long-short-term-memories (LSTMs)

- We can make the units even more complex
- Allow each time step to modify
  - Input gate (current cell matters) $i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right)$
  - Forget (gate 0, forget past) $f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right)$
  - Output (how much cell is exposed) $o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right)$
  - New memory cell $\tilde{c}_t = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right)$
- Final memory cell: $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$
- Final hidden state: $h_t = o_t \circ \tanh(c_t)$

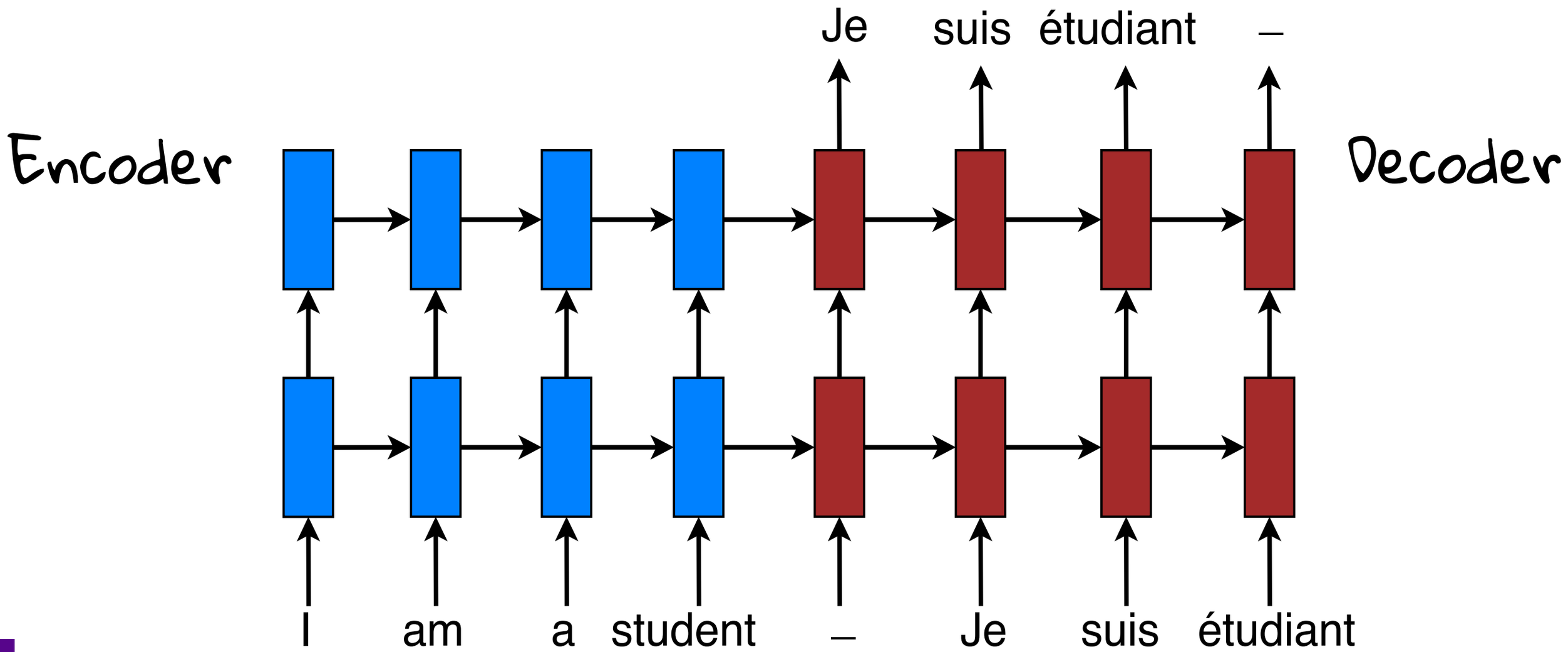
Slide adapted from Richard Socher

RNNs

- In the models we have seen so far, the output labels align with the input (word) sequence.
- What if the inout and output sequences have different lengths? **Q:** example tasks?

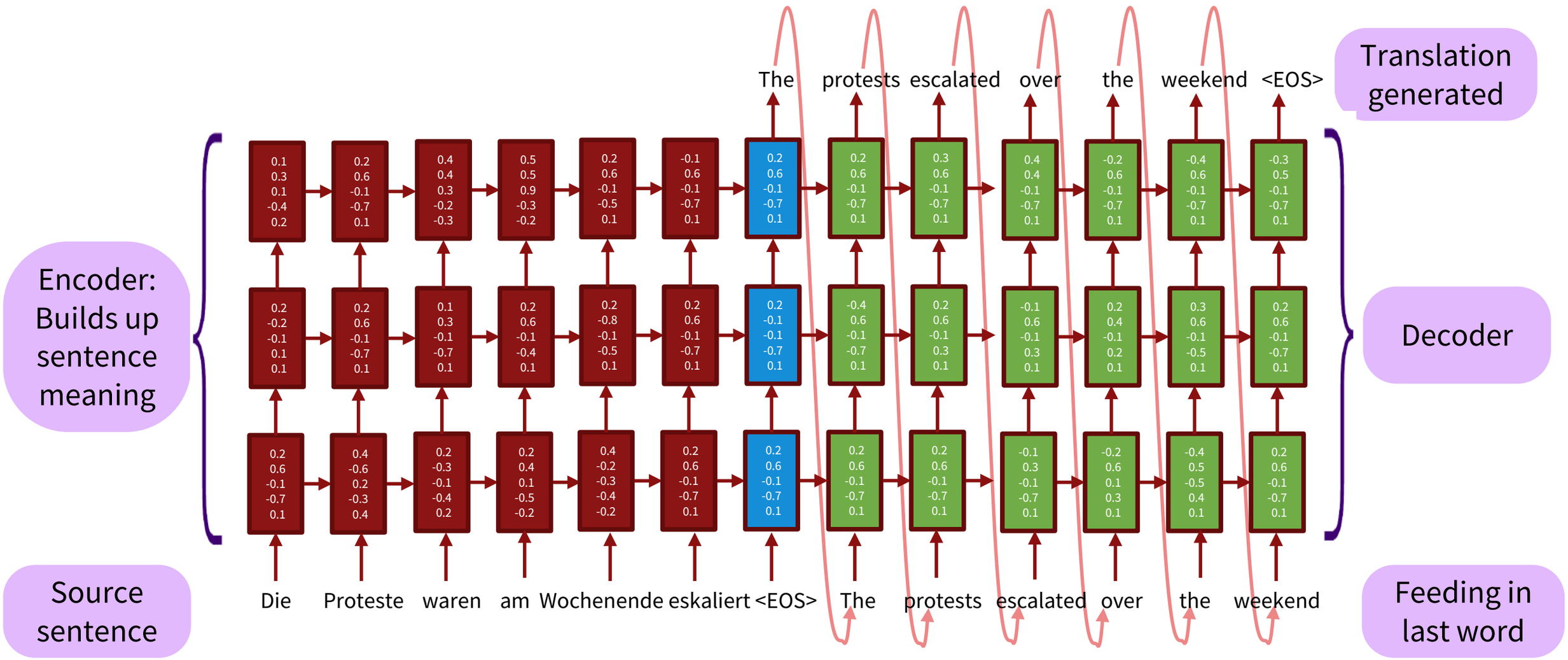
Modern Sequence Models for NMT

Sutskever et al. 2014, fc. Bahdanau et al. 2014, et seq.



Modern Sequence Models for NMT

Sutskever et al. 2014, fc. Bahdanau et al. 2014, et seq.



A deep recurrent neural network

Conditional Recurrent Language Model

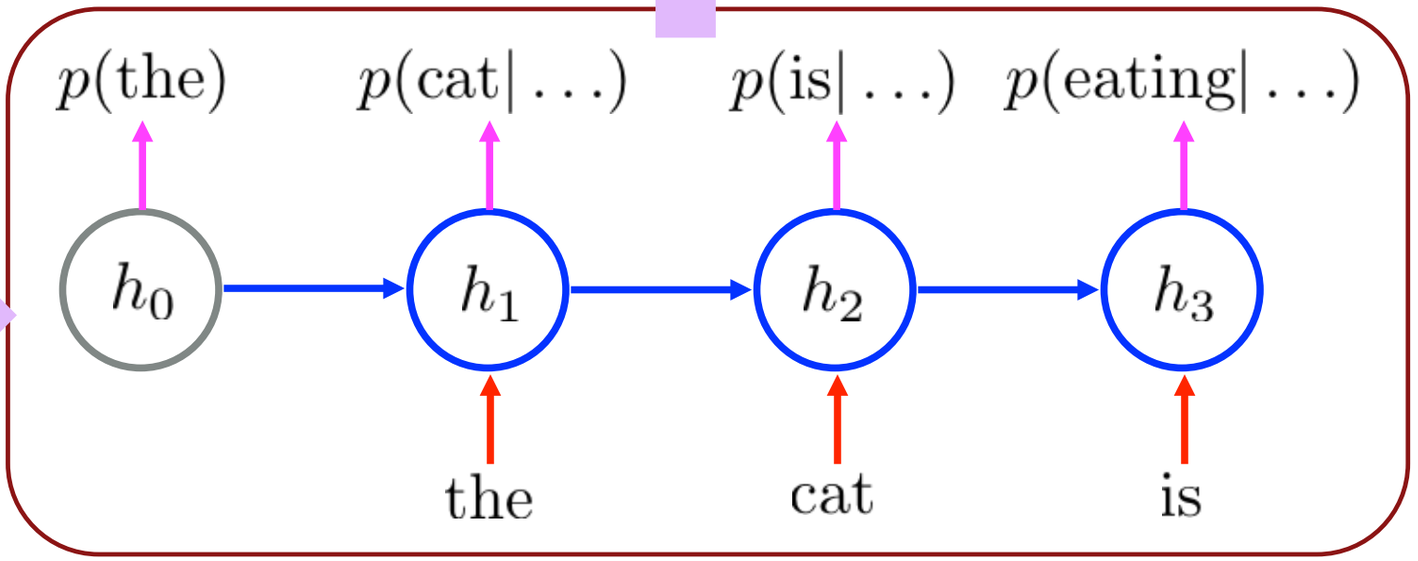
Le chat assis sur le tapis.

?

The cat sat on the mat.

Encoder

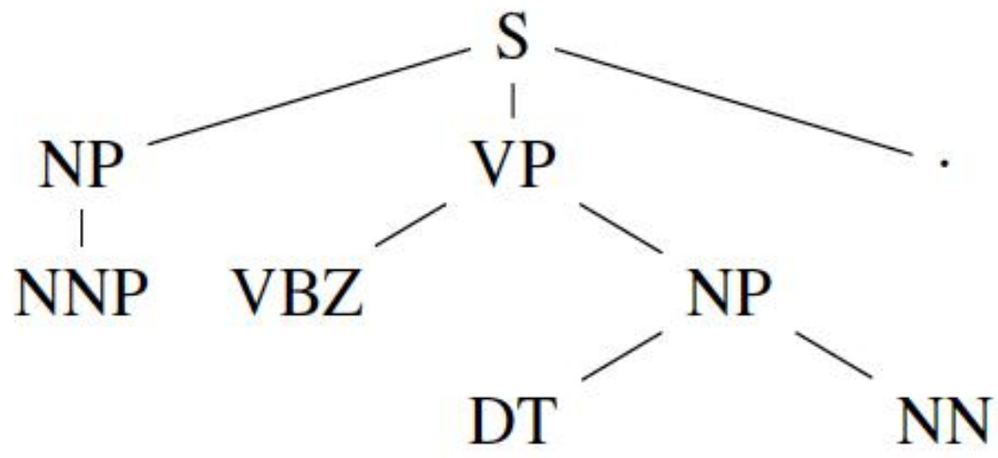
Y



Bypassing explicit grounding

Task: Generate the correct syntactic tree of a sentence

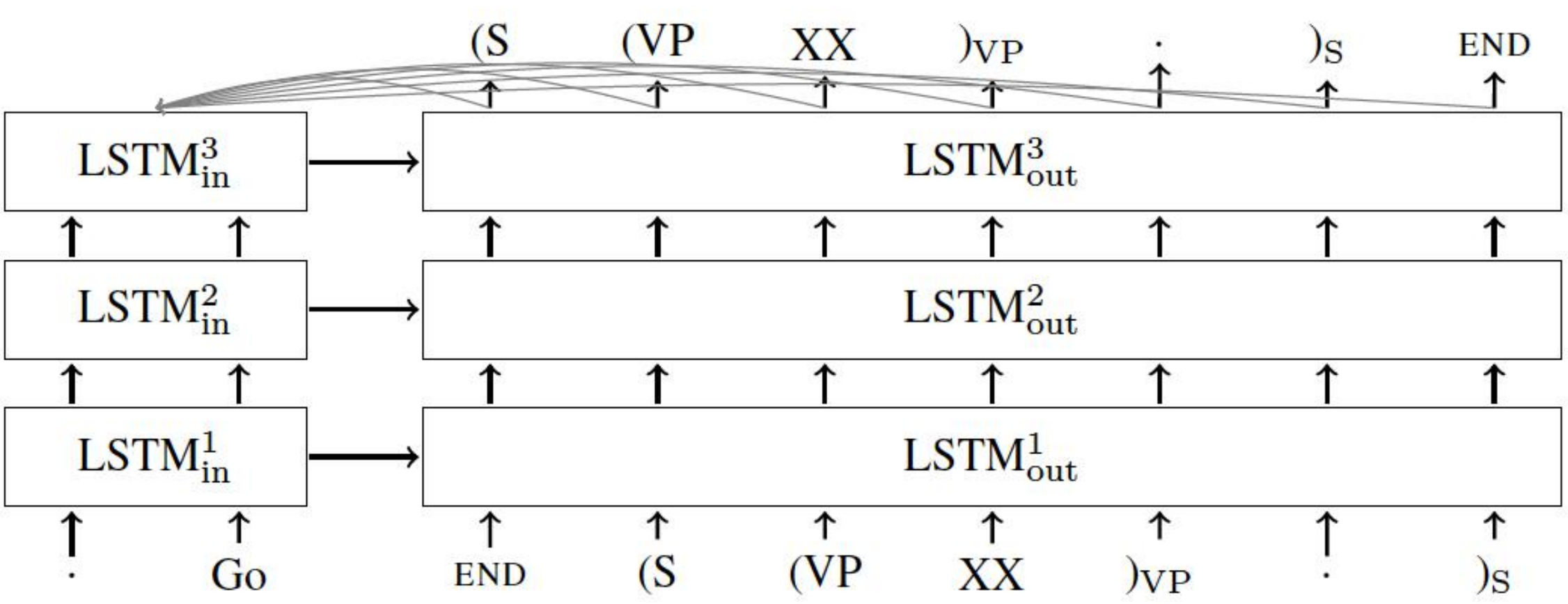
First we convert the syntactic tree into a sequence

John has a dog . → 

John has a dog . → (S (NP NNP)$_{\text{NP}}$ (VP VBZ (NP DT NN)$_{\text{NP}}$)$_{\text{VP}}$.)$_{\text{S}}$

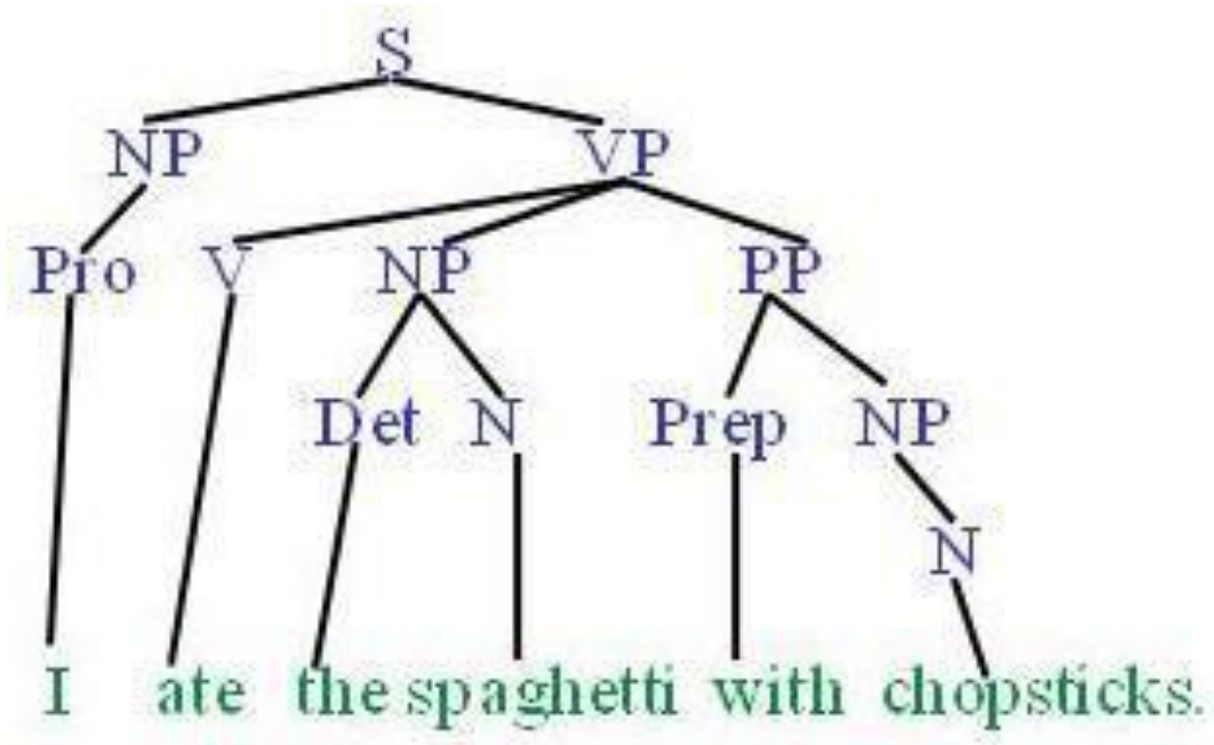
Bypassing explicit grounding

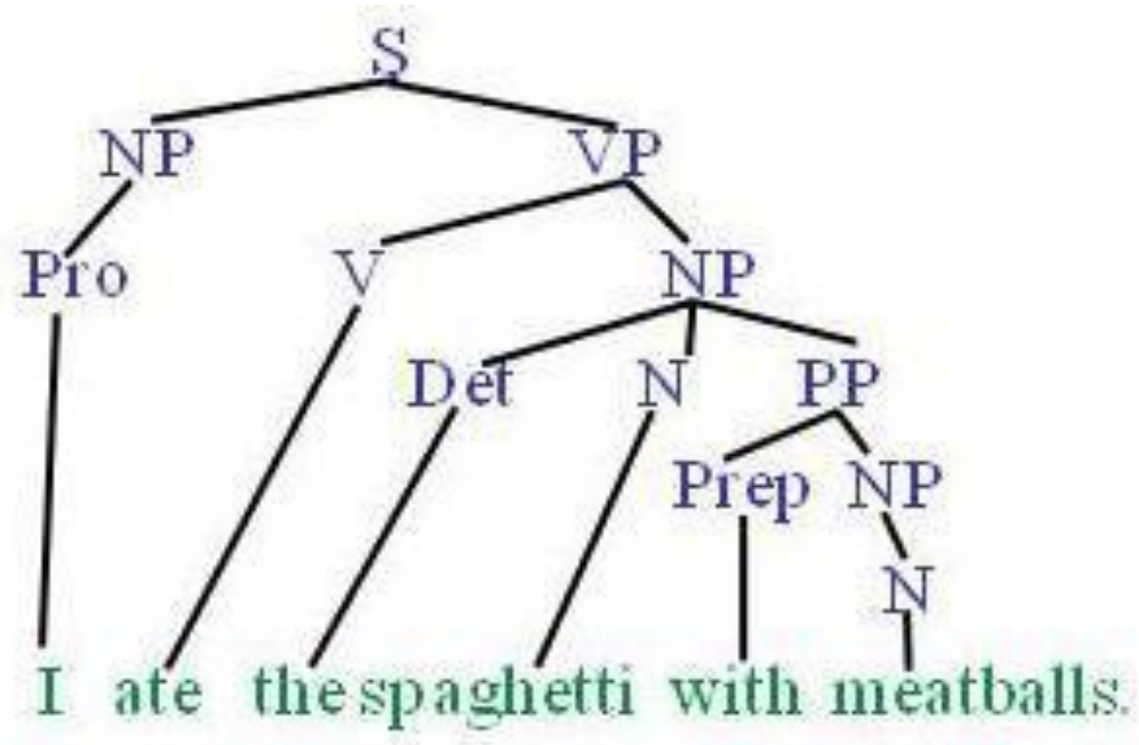
Task: Generate the correct syntactic tree of a sentence



Bypassing explicit grounding

Task: Generate the correct syntactic tree of a sentence





These models do not explicitly know how meatballs and chopsticks look like, or their explicit affordances, however, they do learn implicitly their affordances, by looking at large amounts of text. Is such implicit understanding enough?

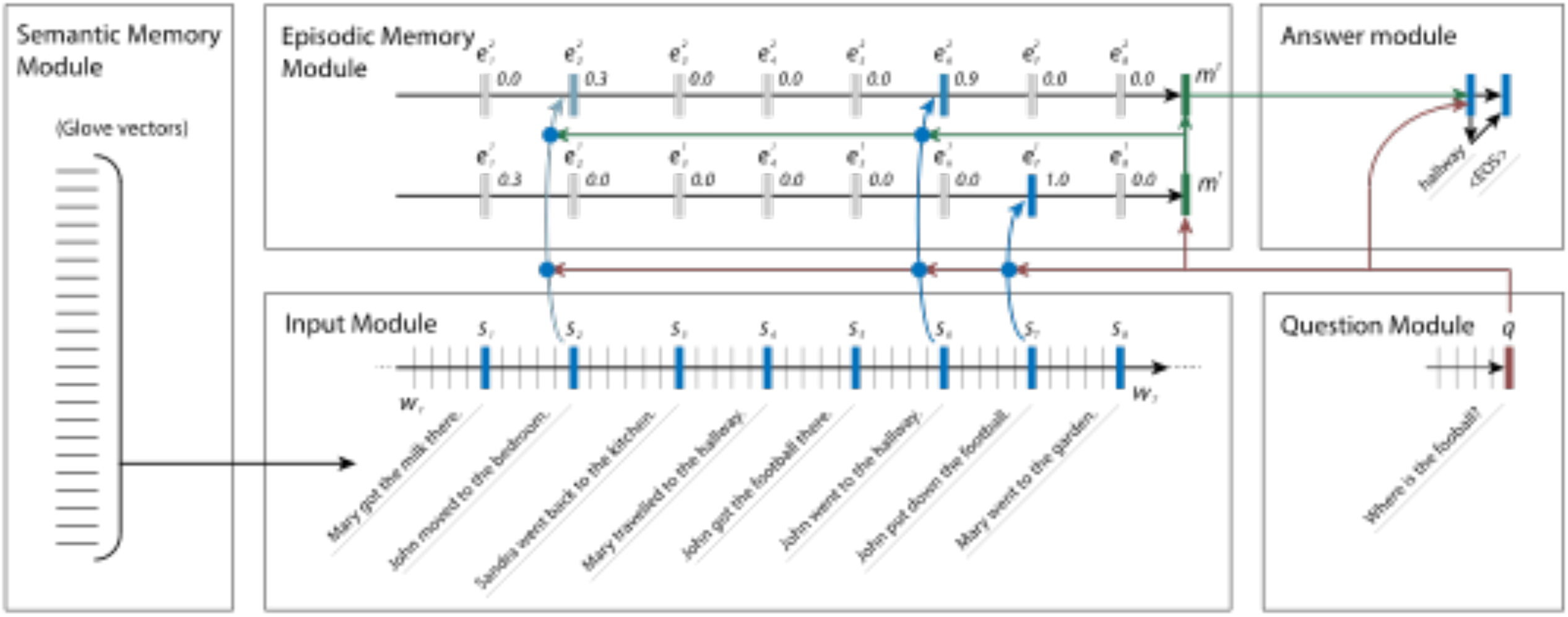
Bypassing explicit grounding

Task: Reading Comprehension

Mary moved to the bathroom. John went to the hallway. Daniel went back to the hallway. Sandra moved to the garden.
Q: Where is Mary? A: bathroom
Q: Where is Daniel? A: hallway

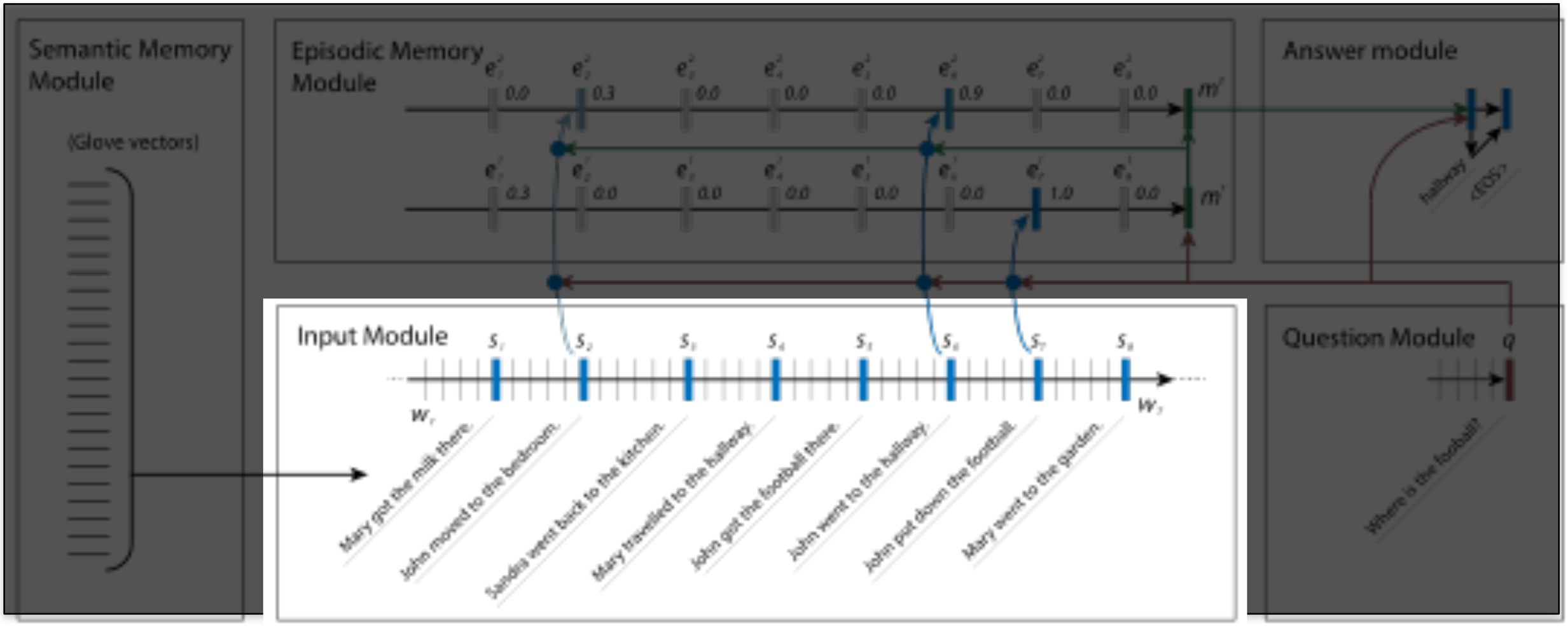
- Input: A passage of text, and a set of questions regarding the passage
- Output: the desired answers
- Supervision: from pairs of (passage+questions, ground truth answers)
- Model examples: Memory networks, dynamic memory networks, (gated) attention readers etc.

Dynamic Memory Network



Slide adapted from Richard Socher

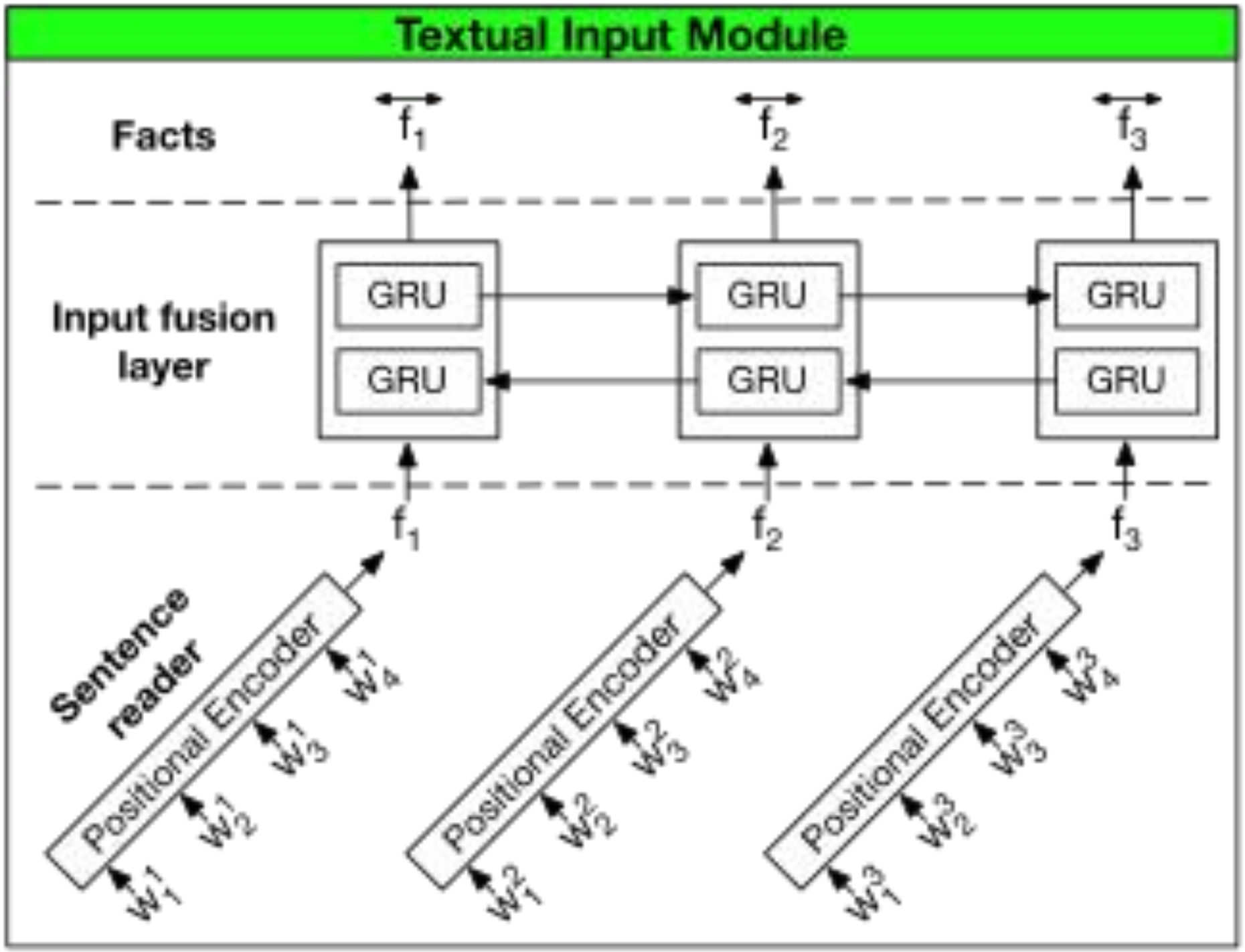
The Modules: Input



Standard GRU. The last hidden state of each sentence is accessible.

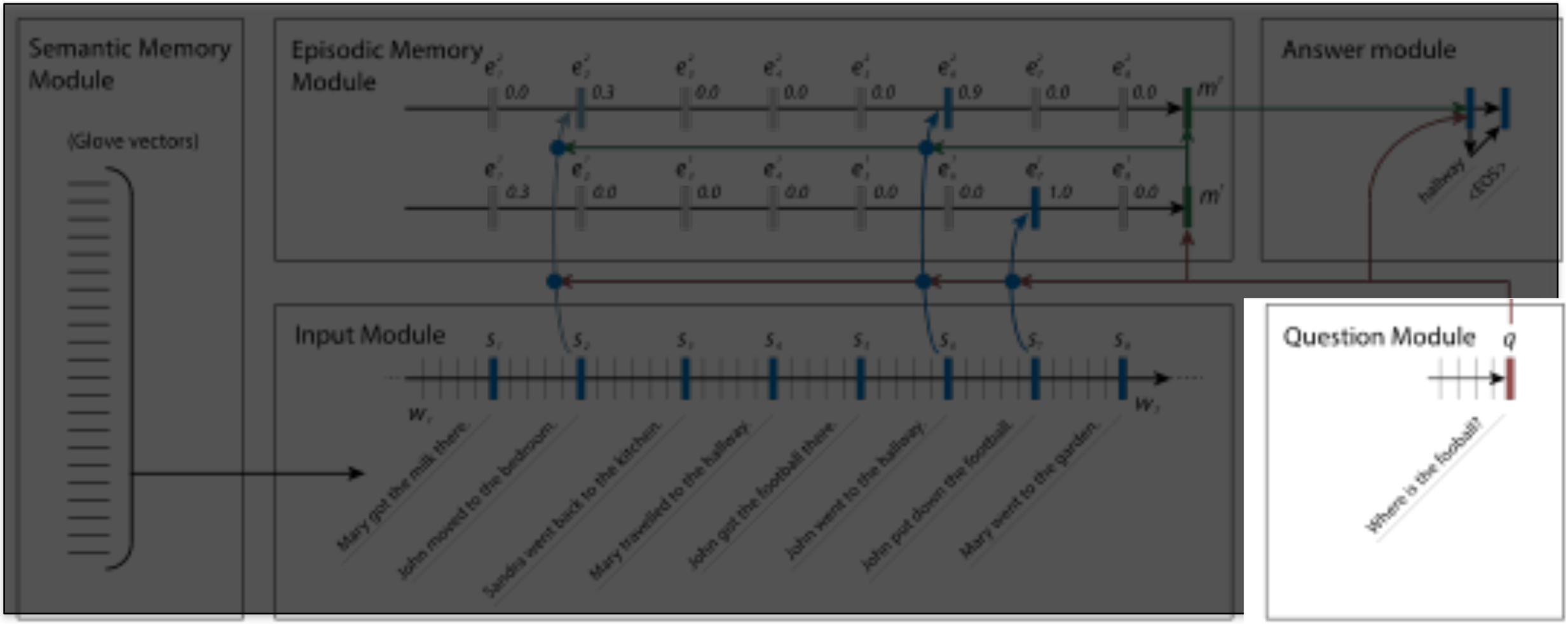
Slide adapted from Richard Socher

Further Improvement: BiGRU



Slide adapted from Richard Socher

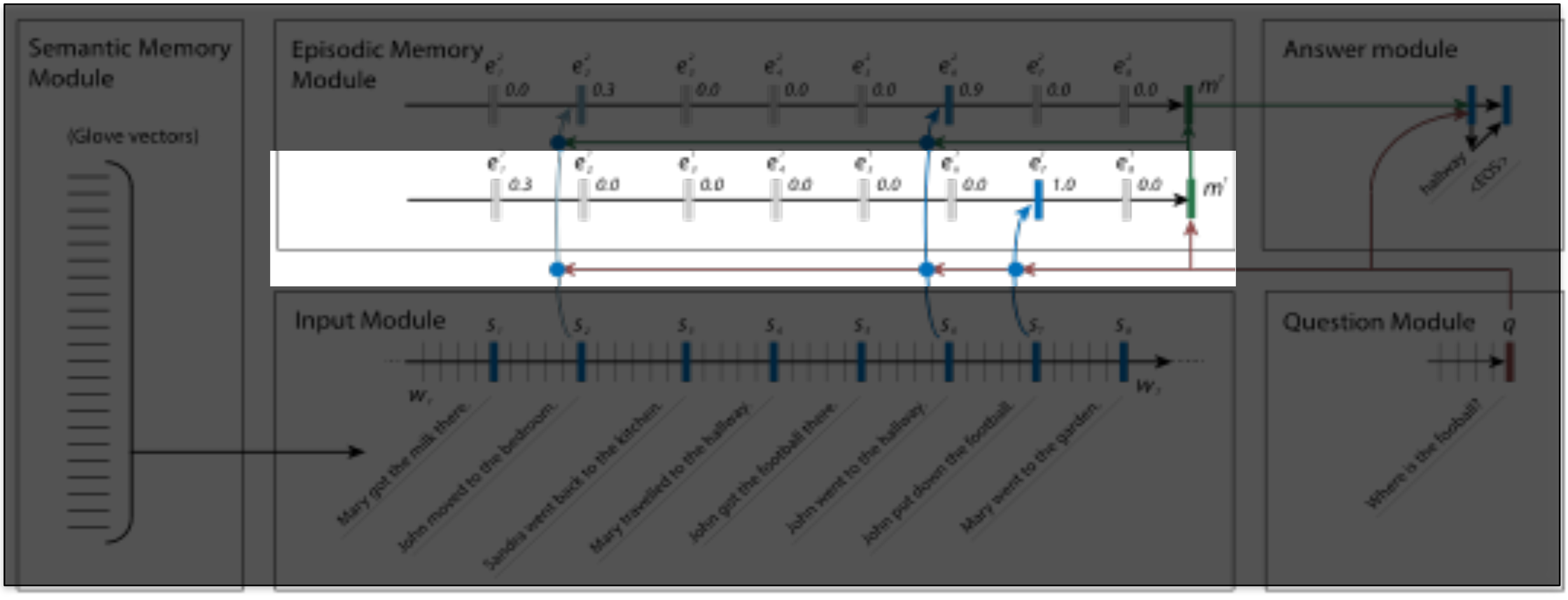
The Modules: Question



$$q_t = GRU(v_t, q_{t-1})$$

Slide adapted from Richard Socher

The Modules: Episodic Memory



$$h_i^t = g_i^t GRU\left(s_i, h_{i-1}^t\right) + \left(1 - g_i^t\right) h_{i-1}^t$$

Last hidden state: m^t

Slide adapted from Richard Socher

The Modules: Episodic Memory

- Gates are activated in sentence relevant to the question or memory.

$$z_i^t = [s_i \circ q \; ; s_i \circ m^{t-1}; |s_i - q| \; ; |s_i - m^{t-1}|]$$

$$Z_i^t = W^{(2)} \tanh\left(W^{(1)} z_i^t + b^{(1)}\right) + b^{(2)}$$

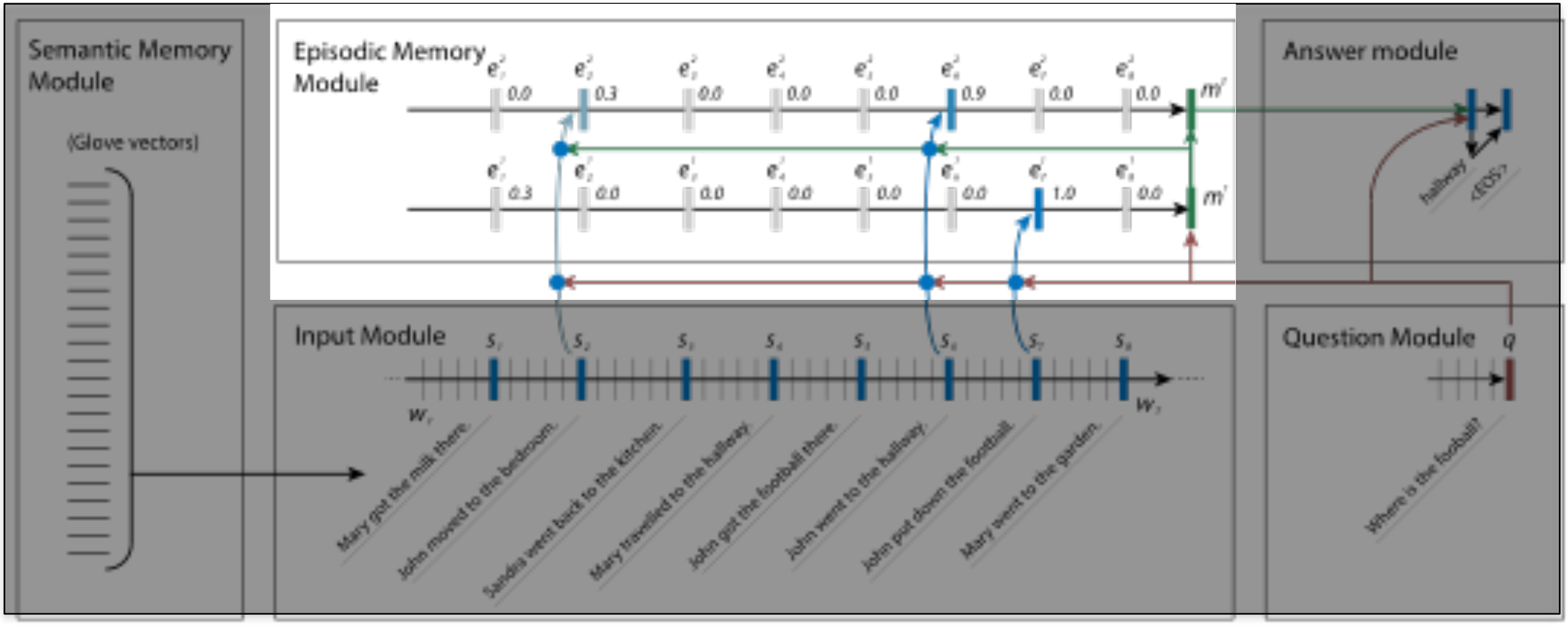
$$g_i^t = \frac{\exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

- When the end of the input is reached, the relevant facts are summarized in another GRU

Slide adapted from Richard Socher

The Modules: Episodic Memory

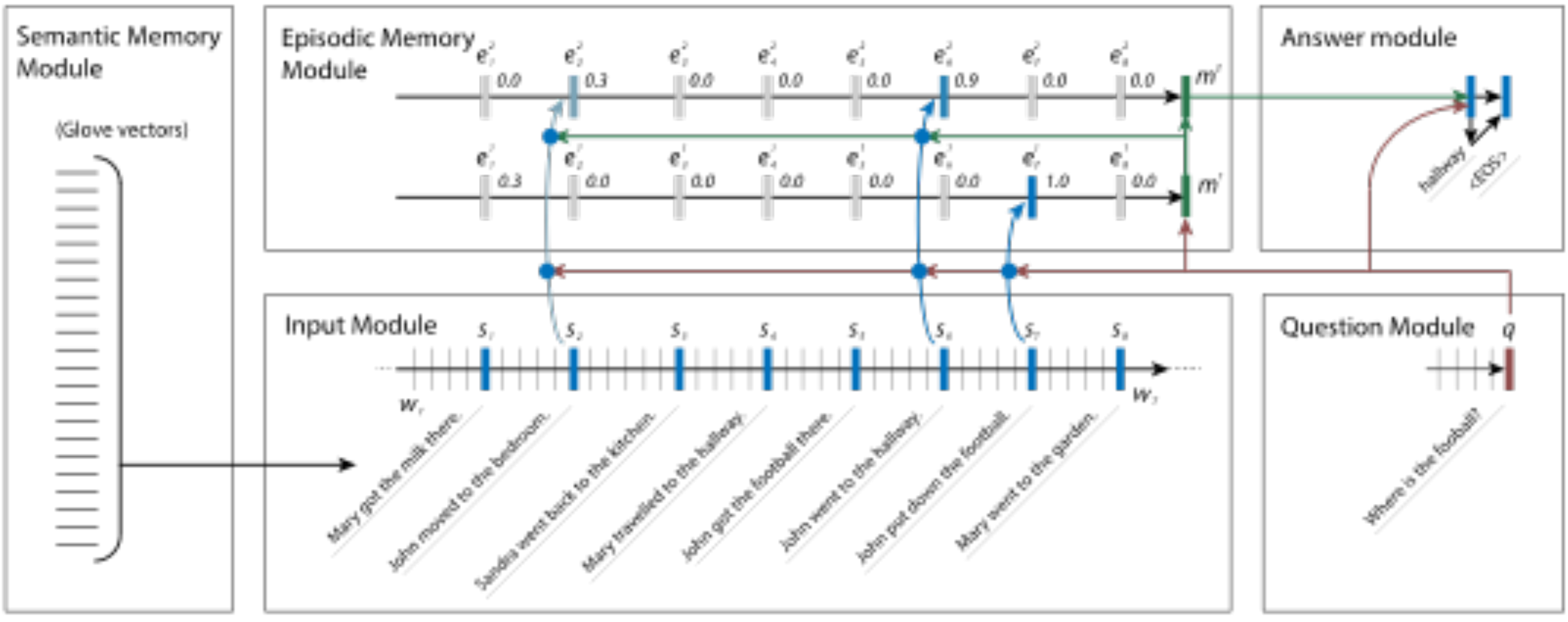
If summary is insufficient to answer the question, repeat sequence over input.



Slide adapted from Richard Socher

The Modules: Answer

$$a_t = \texttt{GRU}([y_{t-1}, q], a_{t-1}), \qquad y_t = softmax(W^{(a)} a_t)$$



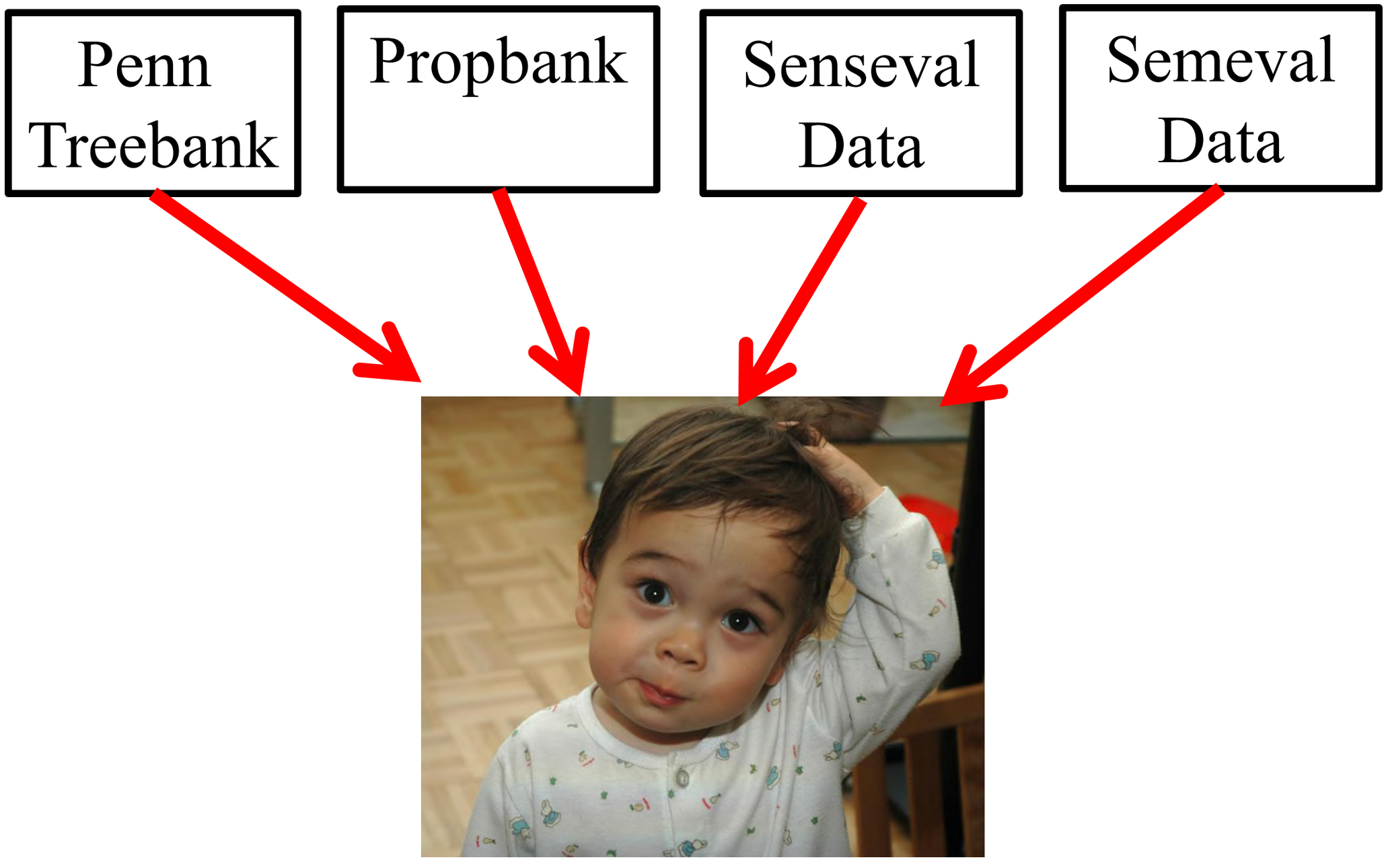
Slide adapted from Richard Socher

Bypassing explicit grounding

We circumvent grounding by using large amounts of supervised training data.

Not only for syntactic parsing, reading comprehension, but for sense disambiguation, for POS tagging, semantic role labelling etc.

Children Do Not Learn Language from Supervised Data



Slide adapted from Raymond Mooney

Children Do Not Learn Language from Raw Text

Unsupervised language learning is difficult and not an adequate solution since much of the requisite semantic information is not in the linguistic signal.

Children do not learn language from television

Simply aligned visual and linguistic representations do not support learning in infants. Yet, all image captioning and visual question answering models work learn from dataset of such aligned representations.

Learning Language from Perceptual Context

- The natural way to learn language is to perceive language in the context of its use in the physical and social world.
- This requires inferring the meaning of utterances from their perceptual context.

Slide adapted from Raymond Mooney

Problem: Dataset collection!

- Supervision is the bottleneck! Is much harder to have robots wondering around interacting with things and humans giving them sparse linguistic rewards.

- We will visit in the course many efforts/shortcut/solutions to supervision and models researchers have come up thus far. We definitely do not need to follow the embodiment solution, if we can do without it.

What is wrong with ungrounded language?

1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。

In 1519, six hundred Spaniards landed in Mexico to conquer the Aztec Empire with a population of a few million. They lost two thirds of their soldiers in the first clash.

translate.google.com (2009): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of soldiers against their loss.

translate.google.com (2011): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the initial loss of soldiers, two thirds of their encounters.

translate.google.com (2013): 1519 600 Spaniards landed in Mexico to conquer the Aztec empire, hundreds of millions of people, the initial confrontation loss of soldiers two-thirds.

translate.google.com (2014/15/16): 1519 600 Spaniards landed in Mexico, millions of people to conquer the Aztec empire, the first two-thirds of the loss of soldiers they clash.

translate.google.com (2017): In 1519, 600 Spaniards landed in Mexico, to conquer the millions of people of the Aztec empire, the first confrontation they killed two-thirds.

What is wrong with ungrounded language?

Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning

Arthur M. Glenberg and David A. Robertson

University of Wisconsin—Madison

…Because the symbols are ungrounded, they cannot, in principle, capture the meaning of **novel** situations. In contrast, (human) participants in three experiments found it trivially easy to discriminate between descriptions of sensible novel situations (e.g., using a newspaper to protect one's face from the wind) and nonsense novel situations (e.g., using a matchbook to protect one's face from the wind). These results support the Indexical Hypothesis that the meaning of a sentence is constructed by (a) indexing words and phrases to real objects or perceptual, analog symbols; (b) deriving affordances from the objects and symbols; and (c) meshing (coordinating) the affordances under the guidance of syntax.

Cosine vector similarities of sentences, defined as the avg of the vectors of their word constituents, failed to detect coherent versus not coherent stories, while humans succeeded.

What is wrong with ungrounded language?

	LSA cosines	
	Sentence to setting	Central to distinguishing
Setting: Marissa forgot to bring her pillow on her camping trip.		
Afforded: As a substitute for her *pillow*, she filled up an old sweater with **leaves.**	.58	.08
Nonafforded: As a substitute for her *pillow*, she filled up an old sweater with **water.**	.55	.06
Related: As a substitute for her *pillow*, she filled up an old sweater with **clothes.**	.63	.24
Setting: Mike was freezing while walking up State Street into a brisk wind. He knew that he had to get his face covered pretty soon or he would get frostbite. Unfortunately, he didn't have enough money to buy a scarf.		
Afforded: Being clever, he walked into a store and bought a **newspaper** to cover his *face*.	.38	.06
Nonafforded: Being clever, he walked into a store and bought a **matchbook** to cover his *face*.	.42	.03
Related: Being clever, he walked into a store and bought a **ski-mask** to cover his *face*.	.41	.46

Note. Central concepts are italicized; distinguishing concepts are in boldface.

Word meaning in a grounded Language (Glenberg and Robertson 1999)

The **meaning** of a particular situation for a particular animal is the coordinated set of actions available to that animal in that situation.

For example, a chair affords sitting to beings with humanlike bodies, but it does not afford sitting for elephants. A chair also affords protection against snarling dogs for an adult capable of lifting the chair into a defensive position, but not for a small child.

The set of actions depends on the individual's learning history, including personal experiences of actions and learned cultural norms for acting.

Thus, a chair on display in a museum affords sitting, but that action is blocked by cultural norms.

Third, the set of actions depends on the individual's **goals** for action.

A chair can be used to support the body when resting is the goal, and it can be used to raise the body when changing a light bulb is the goal.

Word meaning in a grounded Language (Glenberg and Robertson 1999)

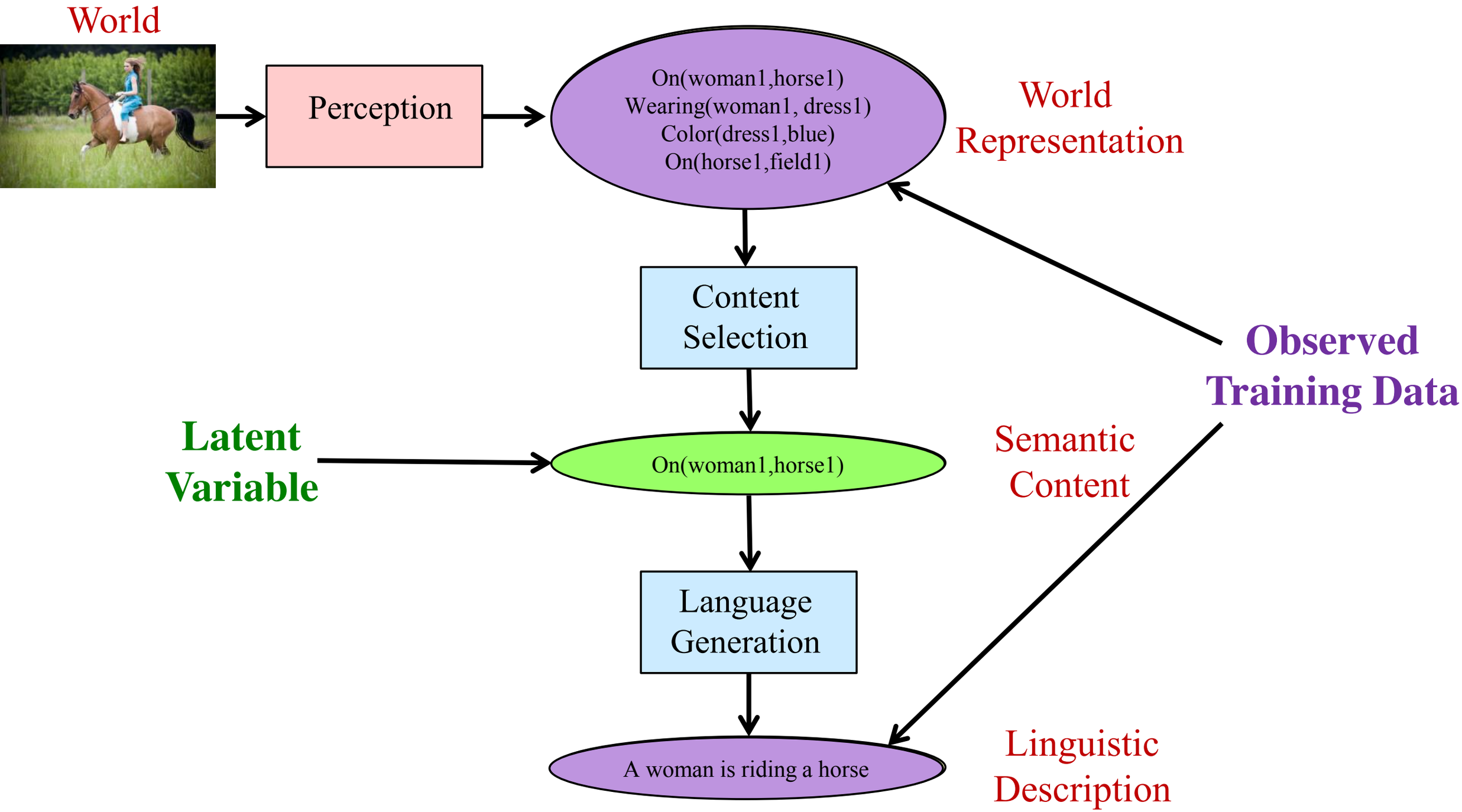
The meaning of a word is not given by its relations to other words and other abstract symbols. Instead, the meaning of words in sentences is emergent: Meaning emerges from the mesh of affordances, learning history,

and goals. Thus the meaning of the word “chair” is not fixed: A chair can be used to sit on, or as a step stool, or as a weapon. Depending on our learning histories, it might also be useful in a balancing act or to protect us from lions in a circus ring. A newspaper can be read, but it can also serve as a scarf.

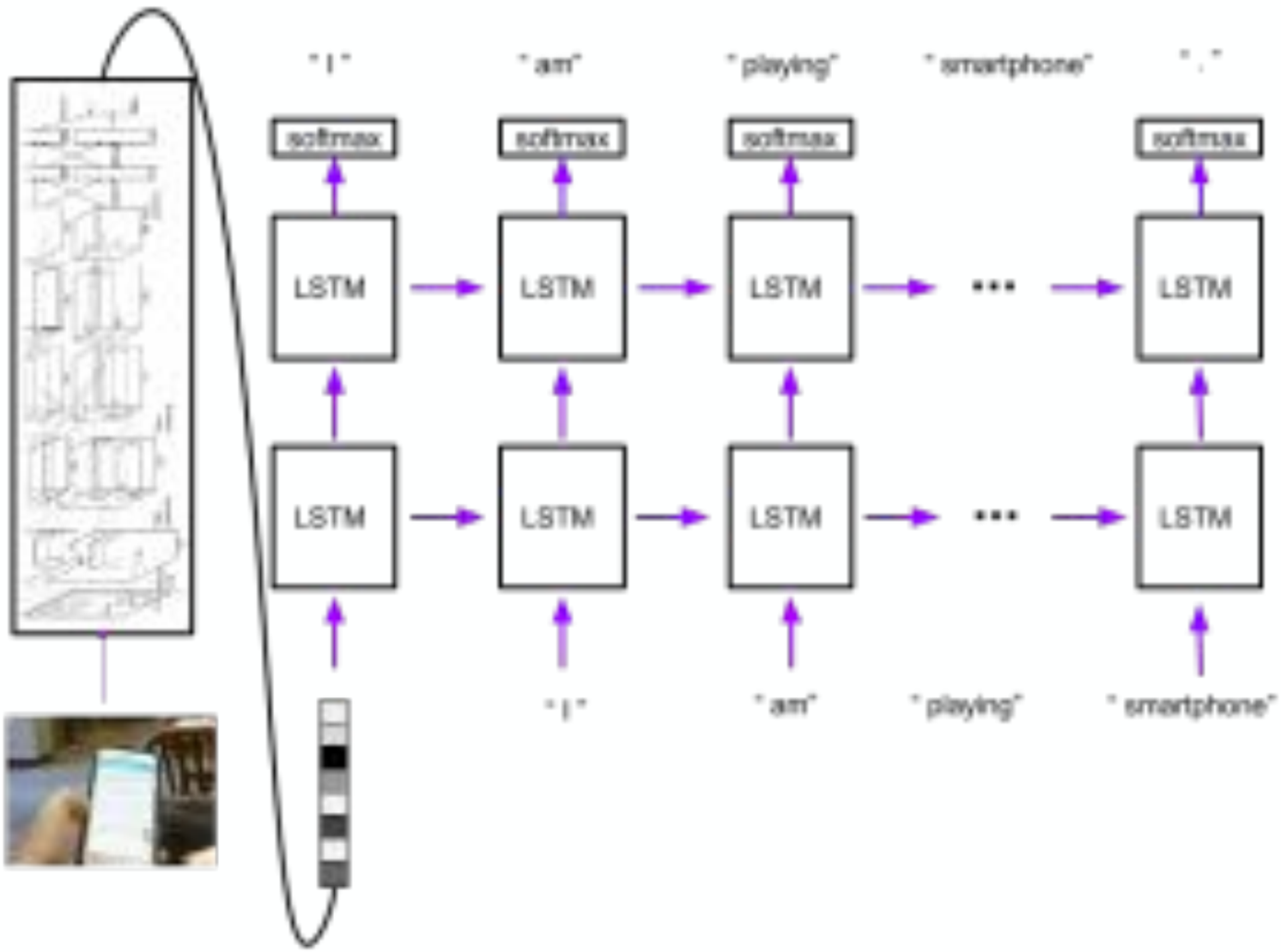
Thus, language comprehension according to this theory, is closely connected to learning affordances and Physics of the world.

What is wrong with current Visual-language Models

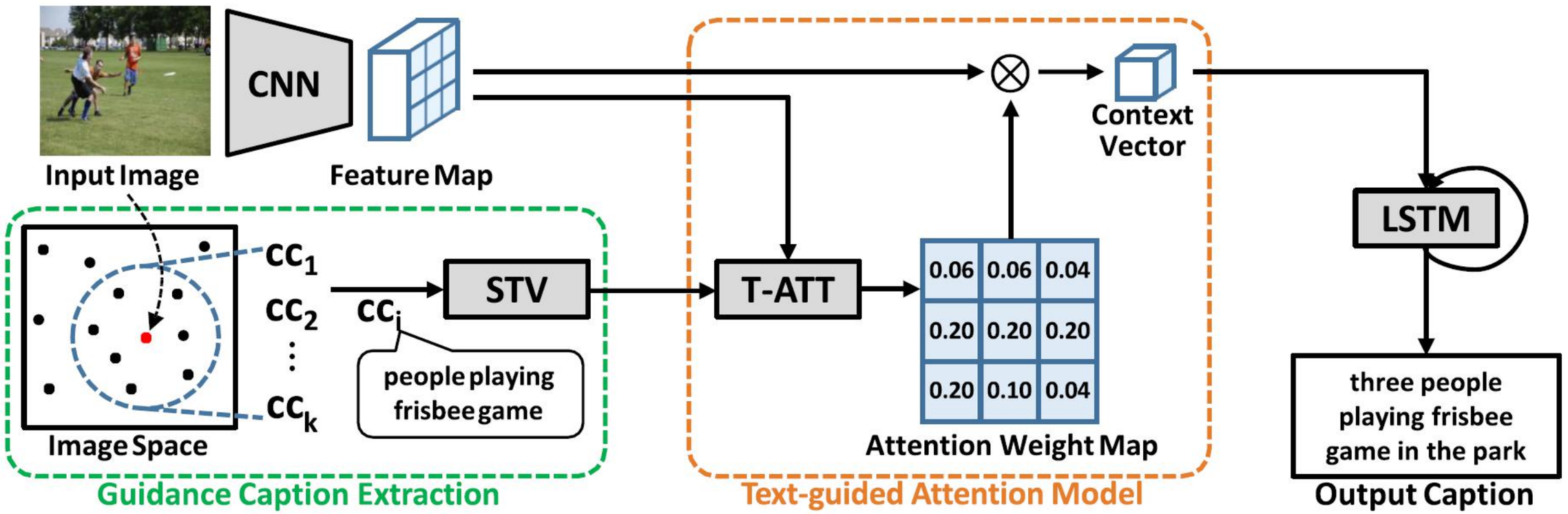
Pre-deep era



What is wrong with current Visual-language Models

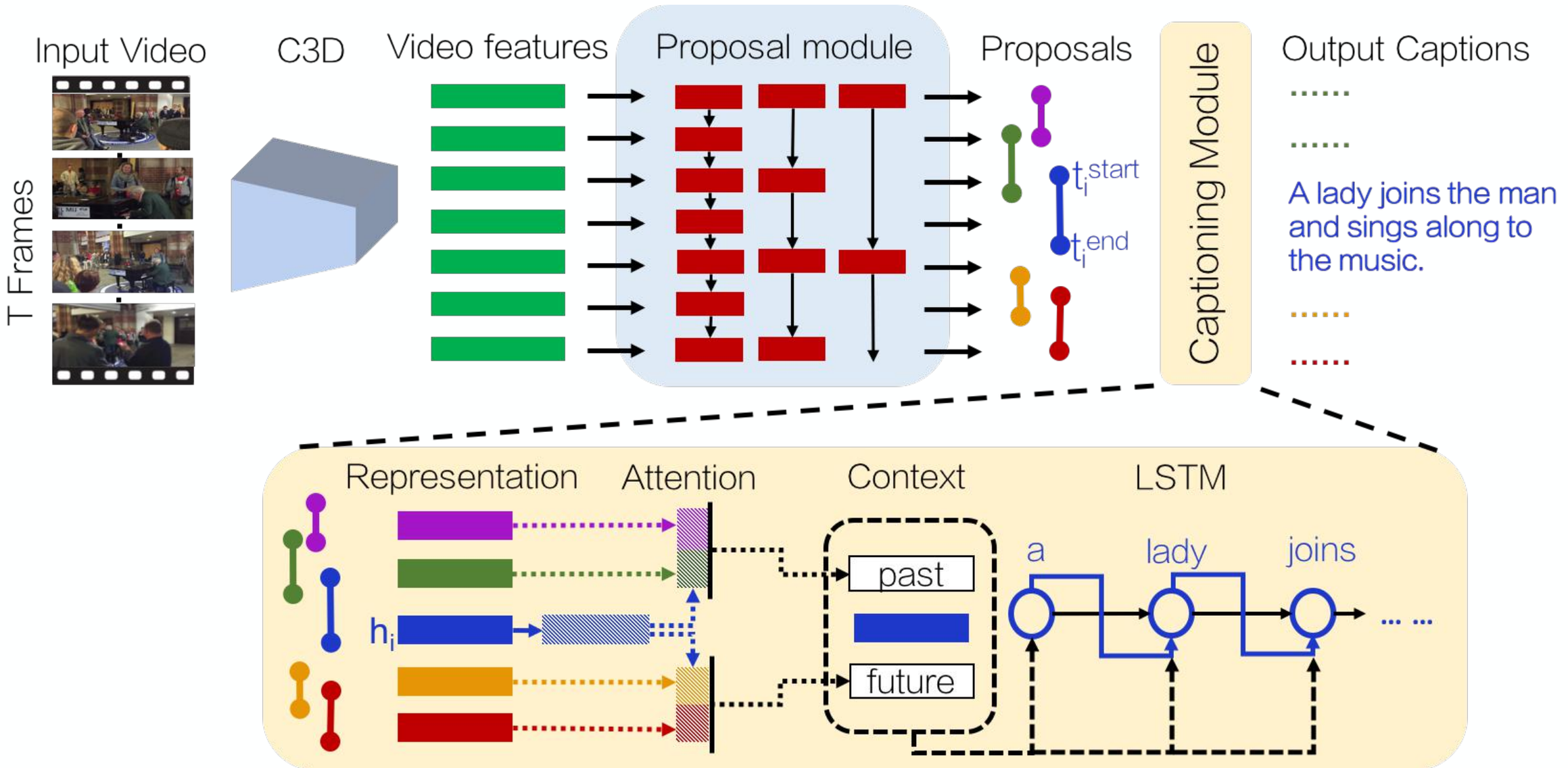


What is wrong with current Visual-language Models



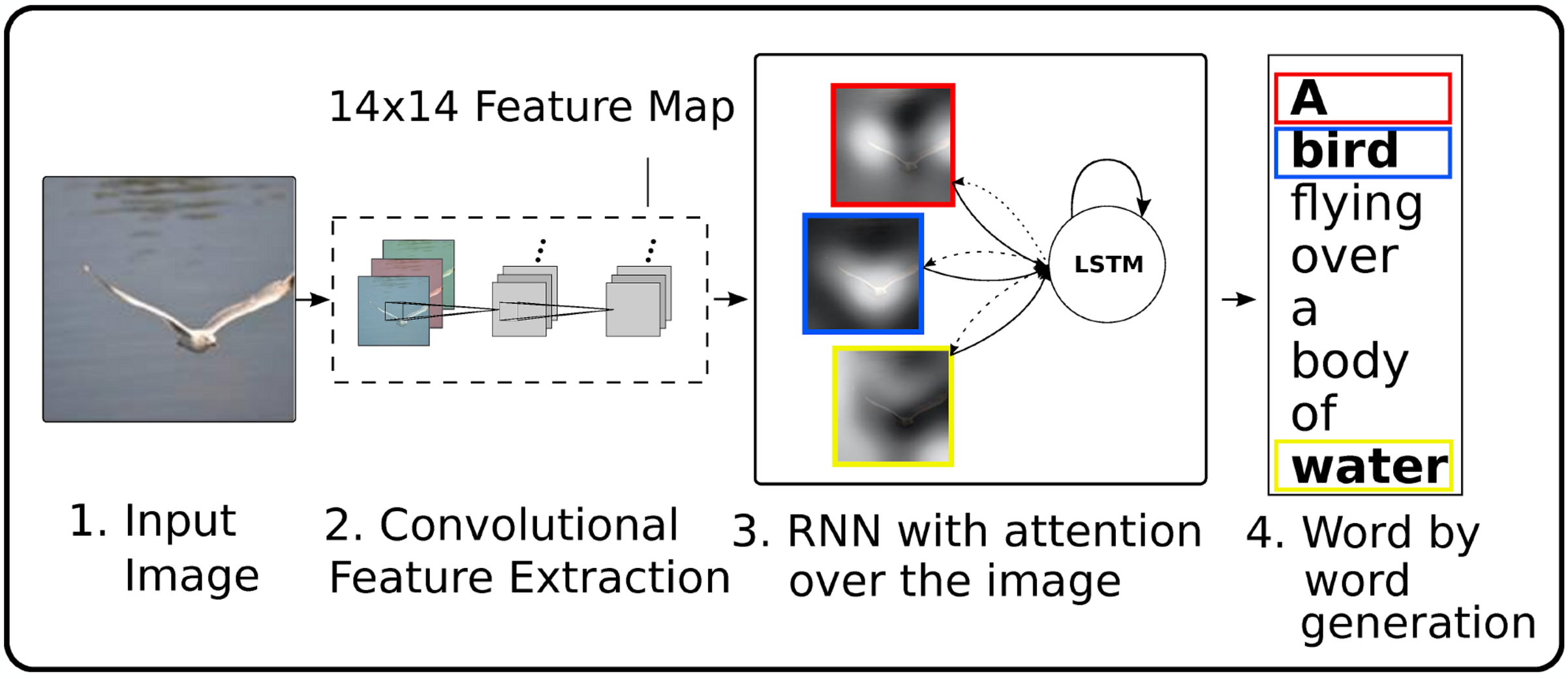
Text-guided attention models for image captioning

What is wrong with current Visual-language Models



Dense captioning events in Videos

What is wrong with current Visual-language Models



Show attend and tell: Neural Image Caption Generation with Visual Attention

What is wrong with current Visual-Language Models

Current Visual-Language models still do not reason about affordances and Physics.

They do not easily generalize to ``novel" situations.

Their success depends on similarity of the test set to the training data.

Language grounding as mental simulation

Comprehending a word like “eagle” activates visual circuits that

capture the implied shape (Zwaan, Stanfield, & Yaxley, 2002) canonical location (Estes, Verges, & Barsalou, 2008), and other visual properties of the object, as well as auditory information. Words denoting actions like stumble engage motor, haptic, and affective circuits (Glenberg & Kaschak, 2002).

We now know that the neural mechanisms underlying imagining a red circle are similar in many respects to the mechanisms that underlie seeing a red circle (Kosslyn, Ganis, & Thompson, 2001). Thus maybe vision provides the simulation of thought.

On this view, there is no need for a language of thought. It’s not that we think “in” language. Rather, language directly interfaces with the mental representations, helping to form the (approximately) compositional, abstract representations. Mental simulations for “You handed Andy the pizza” and “Andy handed you the pizza” are measurably different even though they contain the same words (Glenberg & Kaschak, 2002).