Natural Language Processing

Info 159/259

Lecture 26: Review (April 28, 2020)

David Bamman, UC Berkeley

How do you go out and solve new problems involving text?

1. Language has structure

Sentiment Analysis

“… is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola's genius”

Roger Ebert, Apocalypse Now

“I hated this movie. Hated hated hated hated hated this movie. Hated it. Hated every simpering stupid vacant audience-insulting moment of it. Hated the sensibility that thought anyone would like it.”

Roger Ebert, North

Bag of words

Representation of text only as the counts of words that it contains

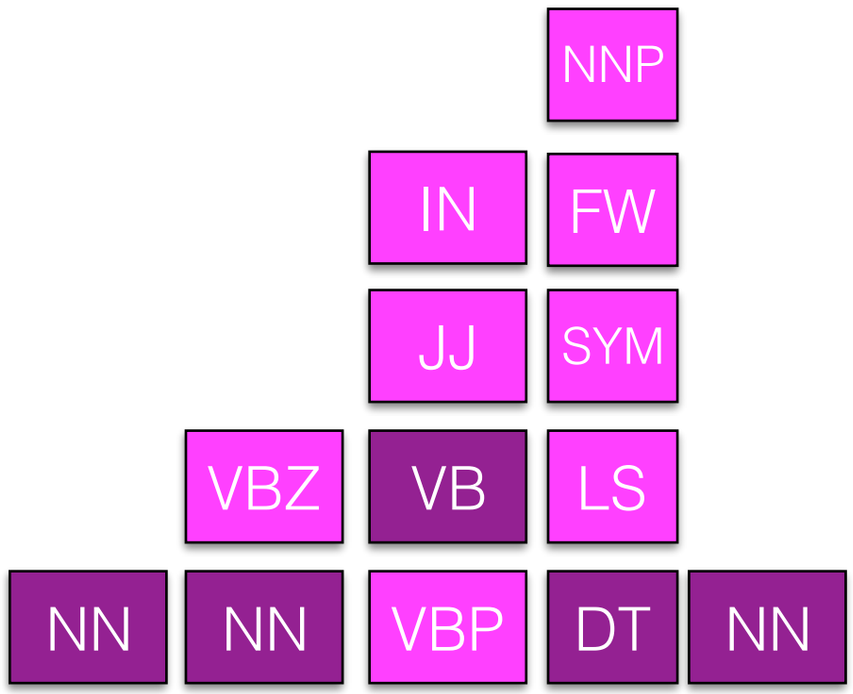
	Apocalypse now	North
the	1	1
of	0	0
hate	0	9
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

“WASHINGTON — Senior Justice Department officials intervened to overrule front-line prosecutors and will recommend a more lenient sentencing for Roger J. Stone Jr., convicted last year of impeding investigators in a bid to protect his longtime friend President _____”

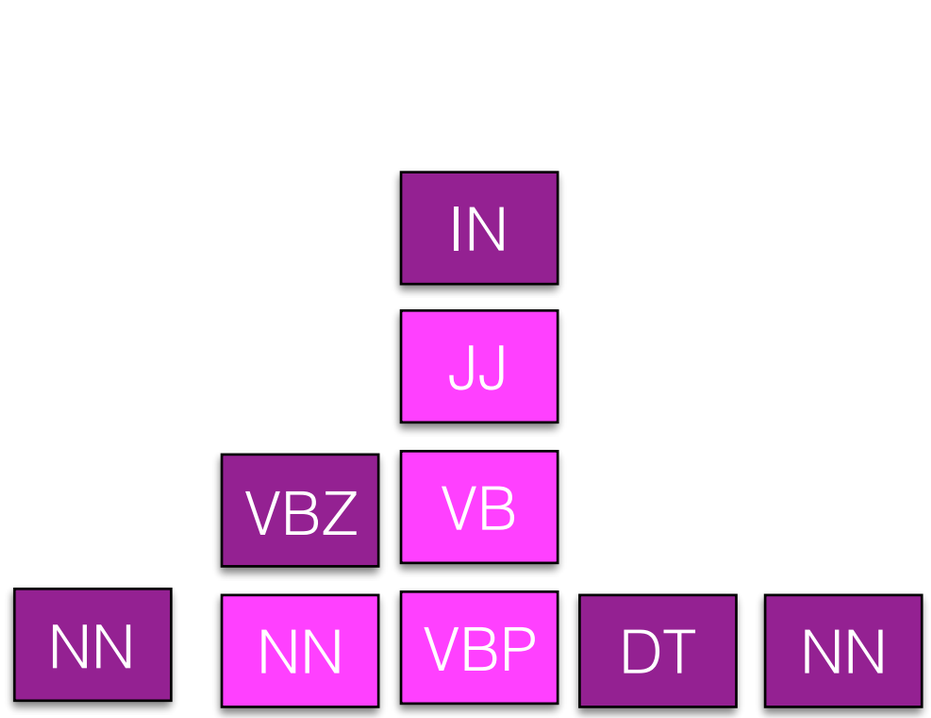
https://www.nytimes.com/2020/02/11/us/politics/roger-stone-sentencing.html

POS tagging

Labeling the tag that's correct for the context.



Fruit flies like a banana



Time flies like an arrow

(Just tags in evidence within the Penn Treebank — more are possible!)

Word senses

original	It urged that the city take steps to remedy this problem
lemma sense	It urge[1] that the city[2] take[1] step[1] to remedy[1] this problem[2]
synset number	It urge[2:32:00] that the city[1:15:01] take[2:41:04] step[1:04:02] to remedy[2:30:00] this problem[1:10:00]

Supersense tagging

artifact artifact motion time group

The station wagons arrived at noon, a long shining line

motion location location

that coursed through the west campus.

Noun supersenses (Ciarmita and Altun 2003)

Thematic roles

The thematic roles for verbs generally are predictable by the syntactic position of the argument (specific to each verb class). Some allow for consistent alternations:

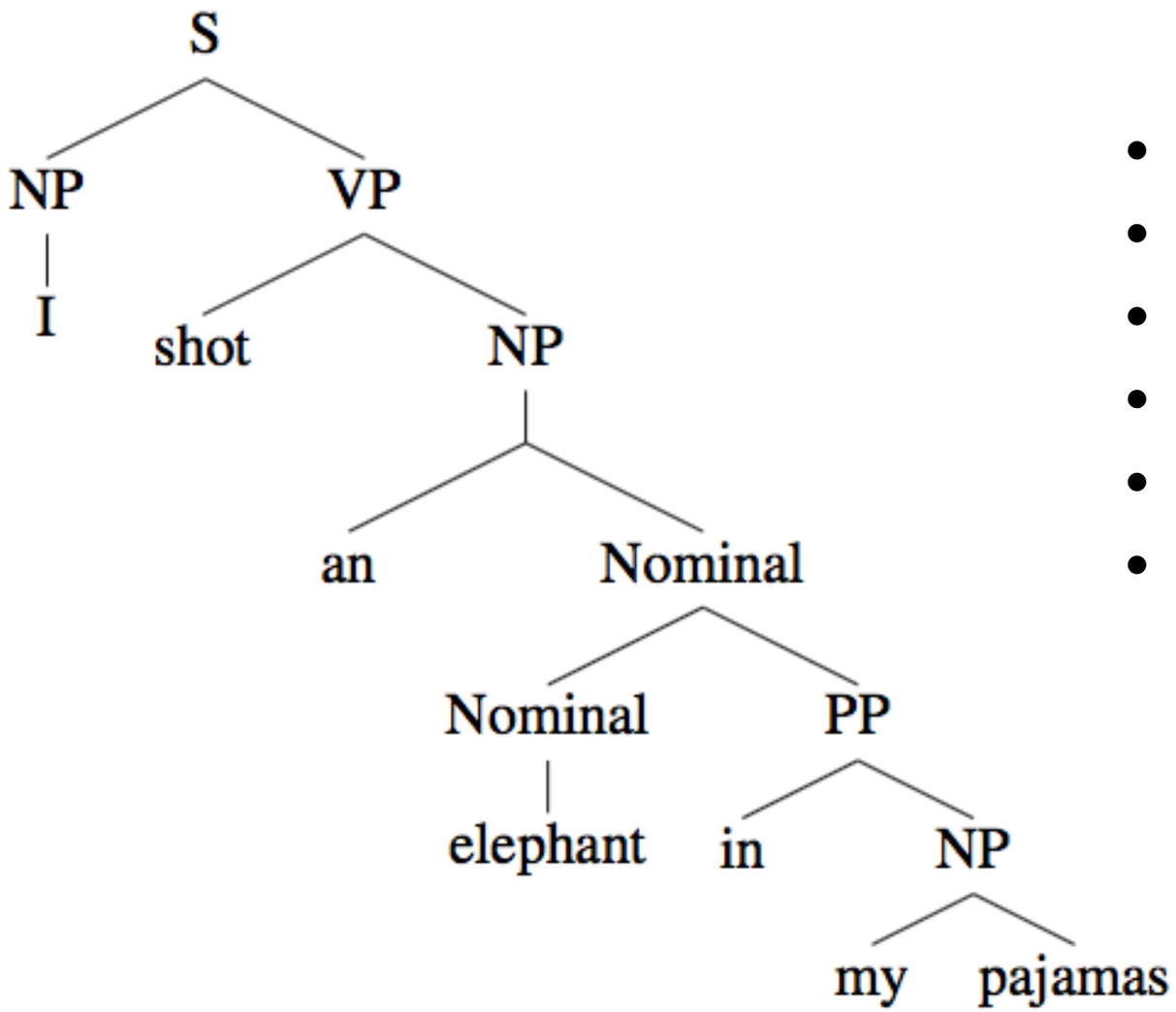
Agent Theme Goal

Doris gave the book to Cary

Agent Goal Theme

Doris gave Cary the book

Phrase-structure syntax

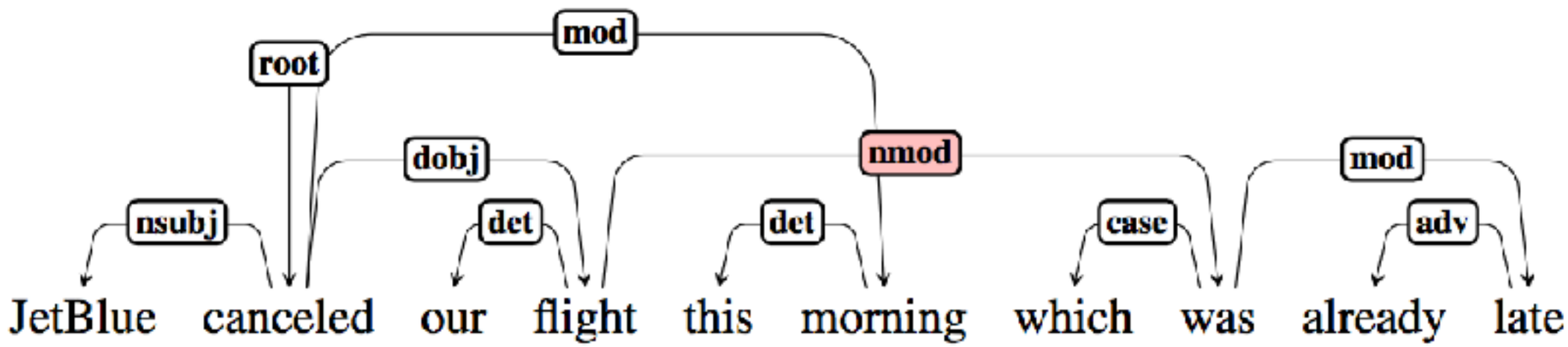


Every internal node is a phrase

- my pajamas
- in my pajamas
- elephant in my pajamas
- an elephant in my pajamas
- shot an elephant in my pajamas
- I shot an elephant in my pajamas

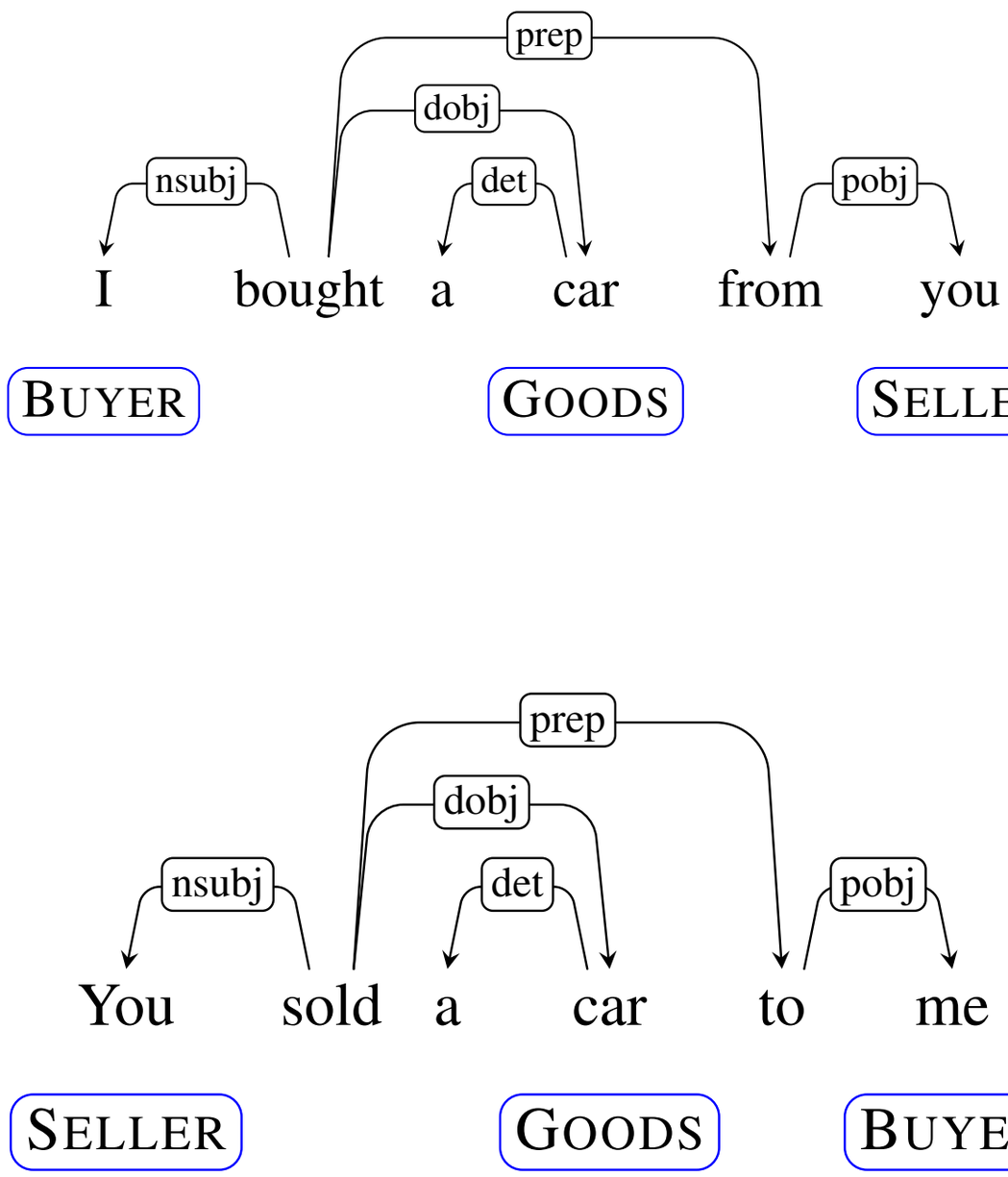
Each phrase could be replaced by another of the same type of constituent

Dependency syntax



Semantic Frames

Two different perspectives on a commercial transaction



Compositional semantics

S
borders(utah,idaho)

S\NP
λy(borders(y,idaho)

NP	(S\NP)/NP	NP
utah	**λx.λy(borders(y,x)**	**idaho**
Utah	borders	Idaho

Coreference

LUKE
I'll never join you!

VADER
If you only knew the power of the dark side. Obi-Wan never told you what happened to your father.

LUKE
He told me enough! It was you who killed him.

VADER
No. I am your father.

LUKE
No. No. That's not true! That's impossible!

VADER
Search your feelings. You know it to be true.

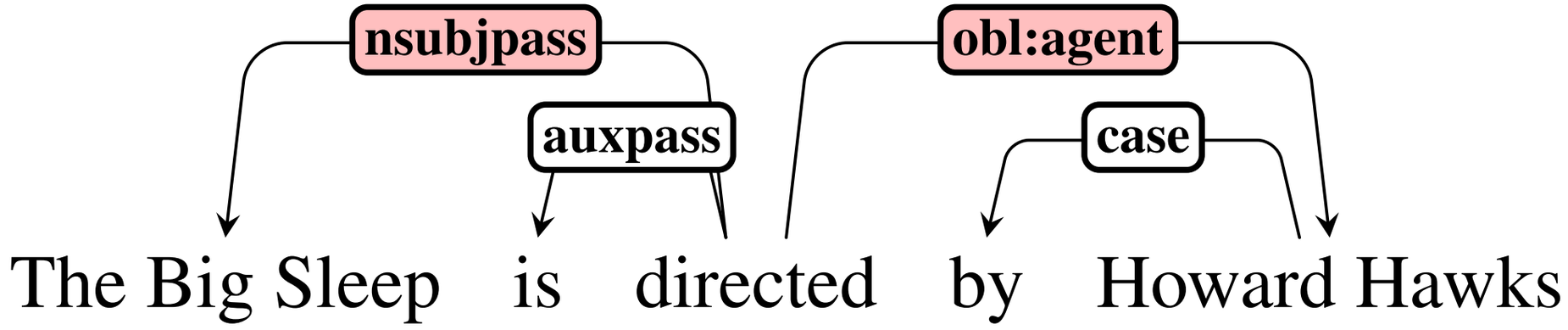
LUKE
No! No! No!

Entity linking

Michael	Jordan	can	dunk	from	the	free	throw	line
B-PER	I-PER							

Supervised relation extraction

[The Big Sleep]$_{m1}$ is a 1946 film noir directed by [Howard Hawks]$_{m2}$, the first film version of Raymond Chandler's 1939 novel of the same name.



[The Big Sleep]$_{m1}$ ←*nsubjpass* directed→*obl:agent* [Howard Hawks]$_{m2}$,

m1←*nsubjpass* ← directed→*obl:agent* → m2

Problem

Design a system that takes in two sentences and judges whether they are paraphrases of each other.

Should I learn Java or Python first?	If I had to choose between learning Java and Python, which one should I learn first?	1
How do you start a bakery?	How can one start a bakery business?	1
Which pizzas are the most popularly ordered on the Domino’s menu?	How many calories does a Dominos pizza have?	0

2. Most new problems can be solved with a familiar class of algorithms

- Classification
- Sequence labeling
- Trees
- Graphs

- Counting and normalizing (NB, PCFG, HMM)
- Loglinear (logistic regression, MEMM, CRF)
- Neural (CNN, RNN, LSTM, seq2seq, attention)

classification

Bayes' Rule

Prior belief that Y = positive
(before you see any data)

Likelihood of "really really the worst movie ever"
given that Y= positive

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{\sum_y P(Y = y)P(X = x|Y = y)}$$

Posterior belief that Y=positive given that X="really really the worst movie ever"

This sum ranges over
y=positive + y=negative
(so that it sums to 1)

Naive Bayes training

Training a Naive Bayes classifier consists of estimating these two quantities from training data for all classes y

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{\sum_y P(Y = y)P(X = x|Y = y)}$$

At test time, use those estimated probabilities to calculate the posterior probability of each class y and select the class with the highest probability

Independence Assumption

really really the worst movie ever

We will assume the features are independent:

$$P(x_1, x_2, x_3, x_4, x_6, x_7 \mid c) = P(x_1 \mid c)P(x_2 \mid c) \ldots P(x_7 \mid c)$$

$$P(x_i \ldots x_n \mid c) = \prod_{i=1}^{N} P(x_i \mid c)$$

Logistic regression

$$P(y = 1 \mid x, \beta) = \frac{1}{1 + \exp\left(-\sum_{i=1}^{F} x_i \beta_i\right)}$$

output space $\mathcal{Y} = \{0, 1\}$

x = feature vector

Feature	Value
the	0
and	0
bravest	0
love	0
loved	0
genius	0
not	0
fruit	1
BIAS	1

β = coefficients

Feature	β
the	0.01
and	0.03
bravest	1.4
love	3.1
loved	1.2
genius	0.5
not	-3.0
fruit	-0.8
BIAS	-0.1

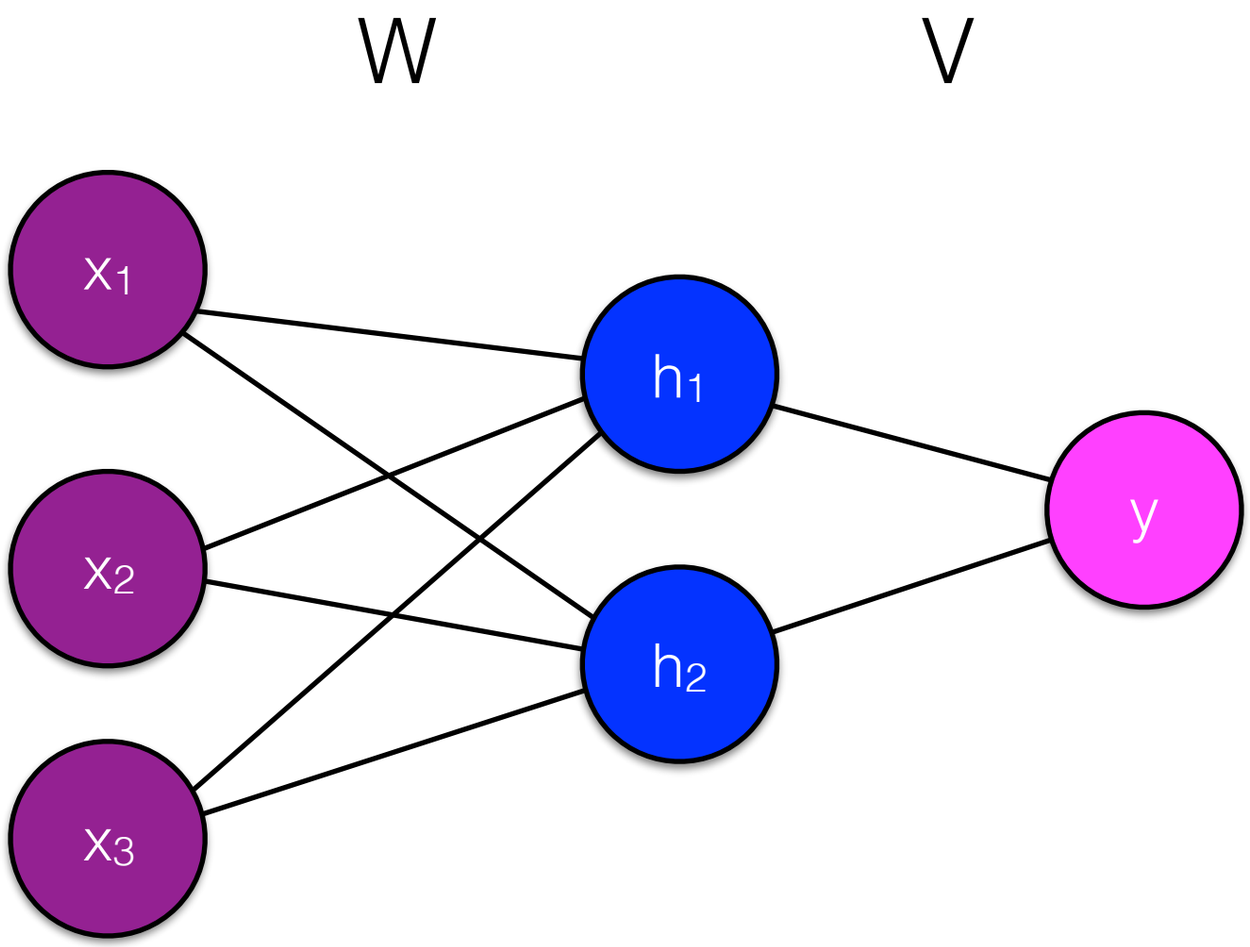
Features

- As a discriminative classifier, logistic regression doesn't assume features are independent like Naive Bayes does.

- Its power partly comes in the ability to create richly expressive features with out the burden of independence.

- We can represent text through features that are not just the identities of individual words, but any feature that is scoped over the entirety of the input.

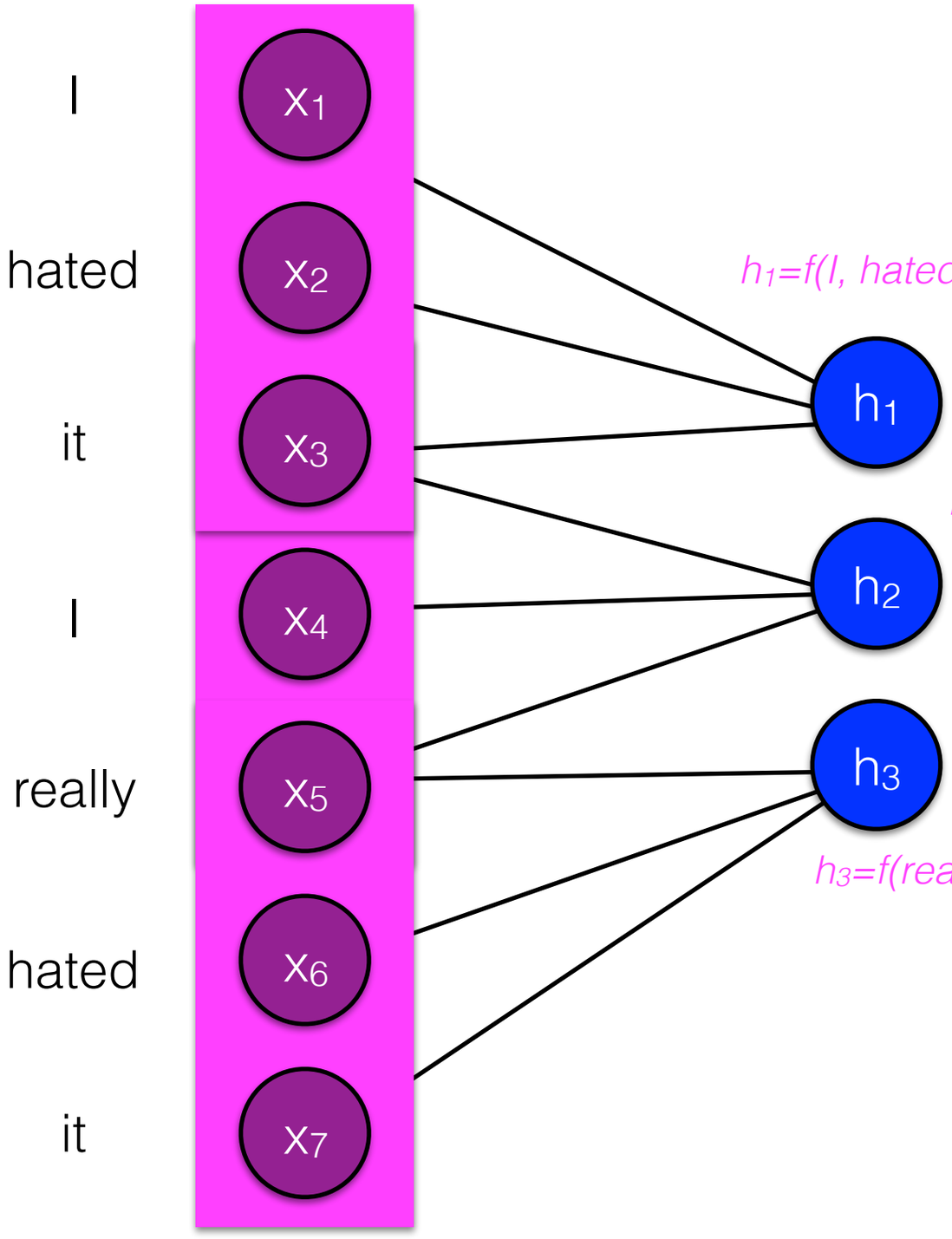
features
contains like
has word that shows up in positive sentiment dictionary
review begins with "I like"
at least 5 mentions of positive affectual verbs (like, love, etc.)

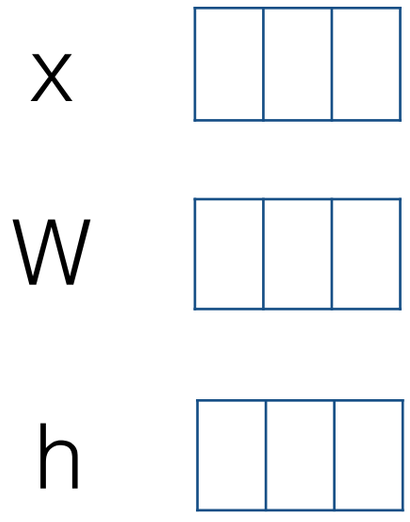


$$\hat{y} = \sigma\left[V_1\left(\sigma\left(\sum_i^F x_i W_{i,1}\right)\right) + V_2\left(\sigma\left(\sum_i^F x_i W_{i,2}\right)\right)\right]$$

we can express y as a function only of the input x and the weights W and V

Convolutional networks

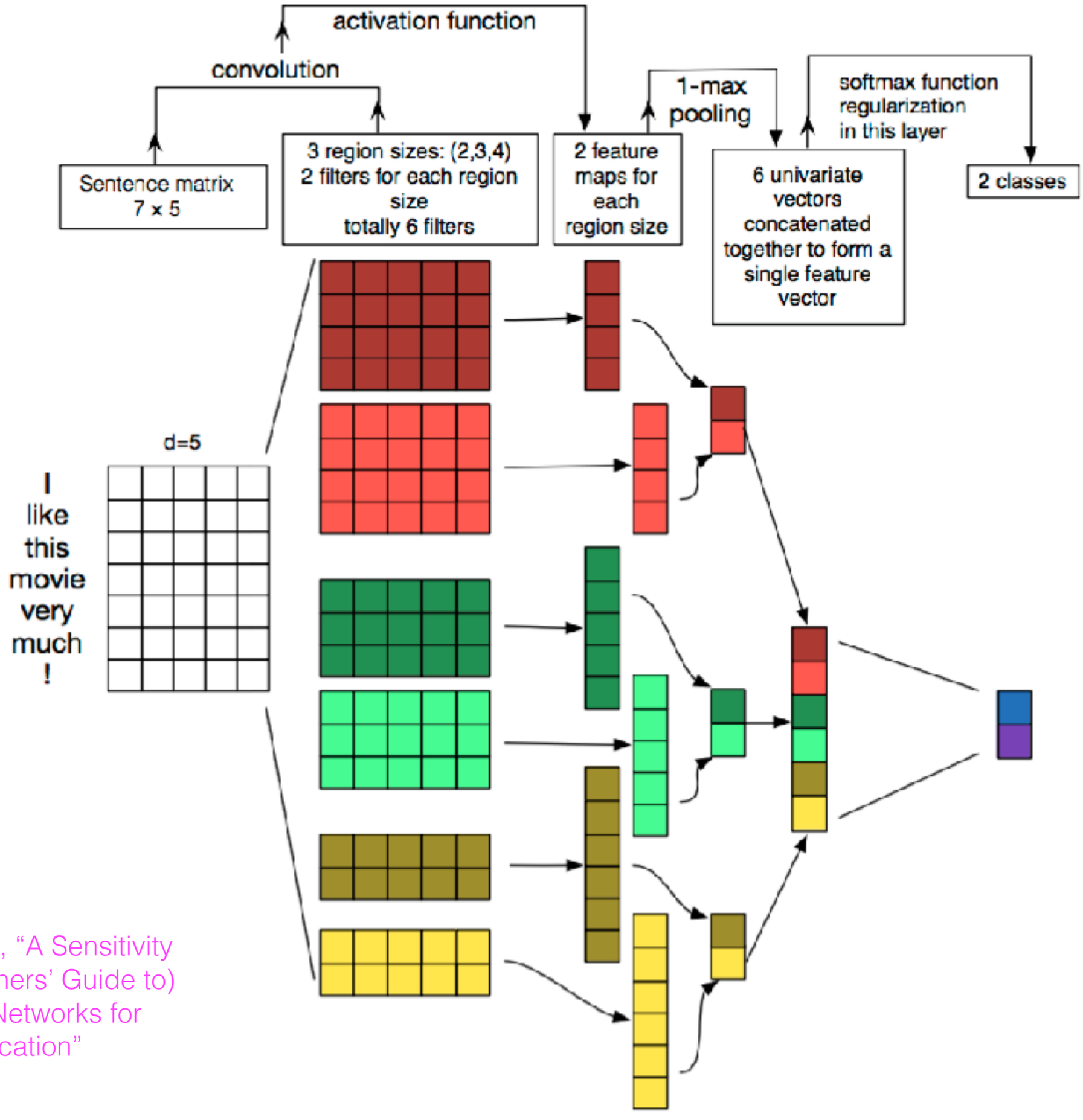




$$h_1 = \sigma(x_1 W_1 + x_2 W_2 + x_3 W_3)$$

$$h_2 = \sigma(x_3 W_1 + x_4 W_2 + x_5 W_3)$$

$$h_3 = \sigma(x_5 W_1 + x_6 W_2 + x_7 W_3)$$



Zhang and Wallace 2016, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification”

sequences

Sequence labeling

$$x = \{x_1, \ldots, x_n\}$$

$$y = \{y_1, \ldots, y_n\}$$

- For a set of inputs x with n sequential time steps, one corresponding label y_i for each x_i
- Model the structure that exists between within y

HMM

$$P(x_1, \ldots, x_n, y_1, \ldots, y_n) \approx \prod_{i=1}^{n+1} P(y_i \mid y_{i-1}) \prod_{i=1}^{n} P(x_i \mid y_i)$$

Hidden Markov Model

Prior probability of label sequence

$$P(y) = P(y_1, \ldots, y_n)$$

$$P(y_1, \ldots, y_n) \approx \prod_{i=1}^{n+1} P(y_i \mid y_{i-1})$$

- We'll make a first-order Markov assumption and calculate the joint probability as the product the individual factors conditioned only on the previous tag.

Hidden Markov Model

$$P(x \mid y) = P(x_1, \ldots, x_n \mid y_1, \ldots, y_n)$$

$$P(x_1, \ldots, x_n \mid y_1, \ldots, y_n) \approx \prod_{i=1}^{N} P(x_i \mid y_i)$$

- Here again we'll make a strong assumption: the probability of the word we see at a given time step is only dependent on its label

MEMM

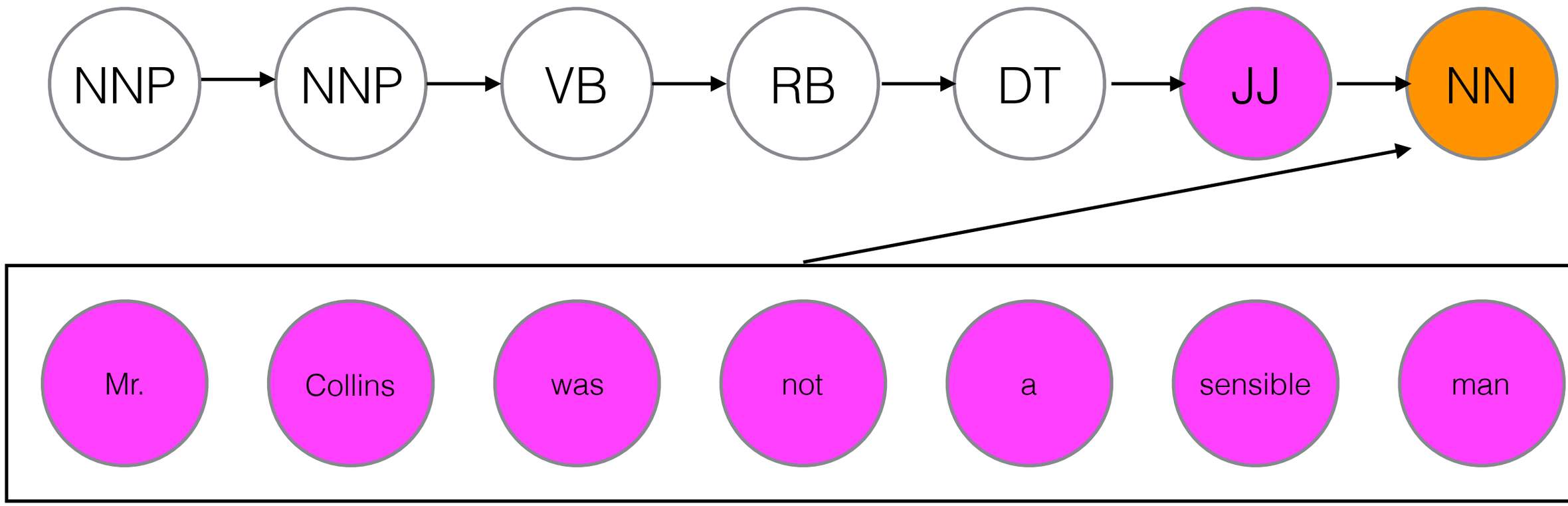
General maxent form

$$\arg\max_y P(y \mid x, \beta)$$

Maxent with first-order Markov assumption: Maximum Entropy Markov Model

$$\arg\max_y \prod_{i=1}^{n} P(y_i \mid y_{i-1}, x)$$

MEMM



Features

$$f(t_i, t_{i-1}; x_1, \ldots, x_n)$$

Features are scoped over the previous predicted tag and the entire observed input

feature	example
x_i = man	1
t_{i-1} = JJ	1
i=n (last word of sentence)	1
x_i ends in -ly	0

MEMM Training

$$\prod_{i=1}^{n} P(y_i \mid y_{i-1}, x, \beta)$$

Locally normalized — at each time step, each conditional distribution sums to 1

Conditional random fields

- We can solve this problem using global normalization (over the entire sequences) rather than locally normalized factors.

MEMM

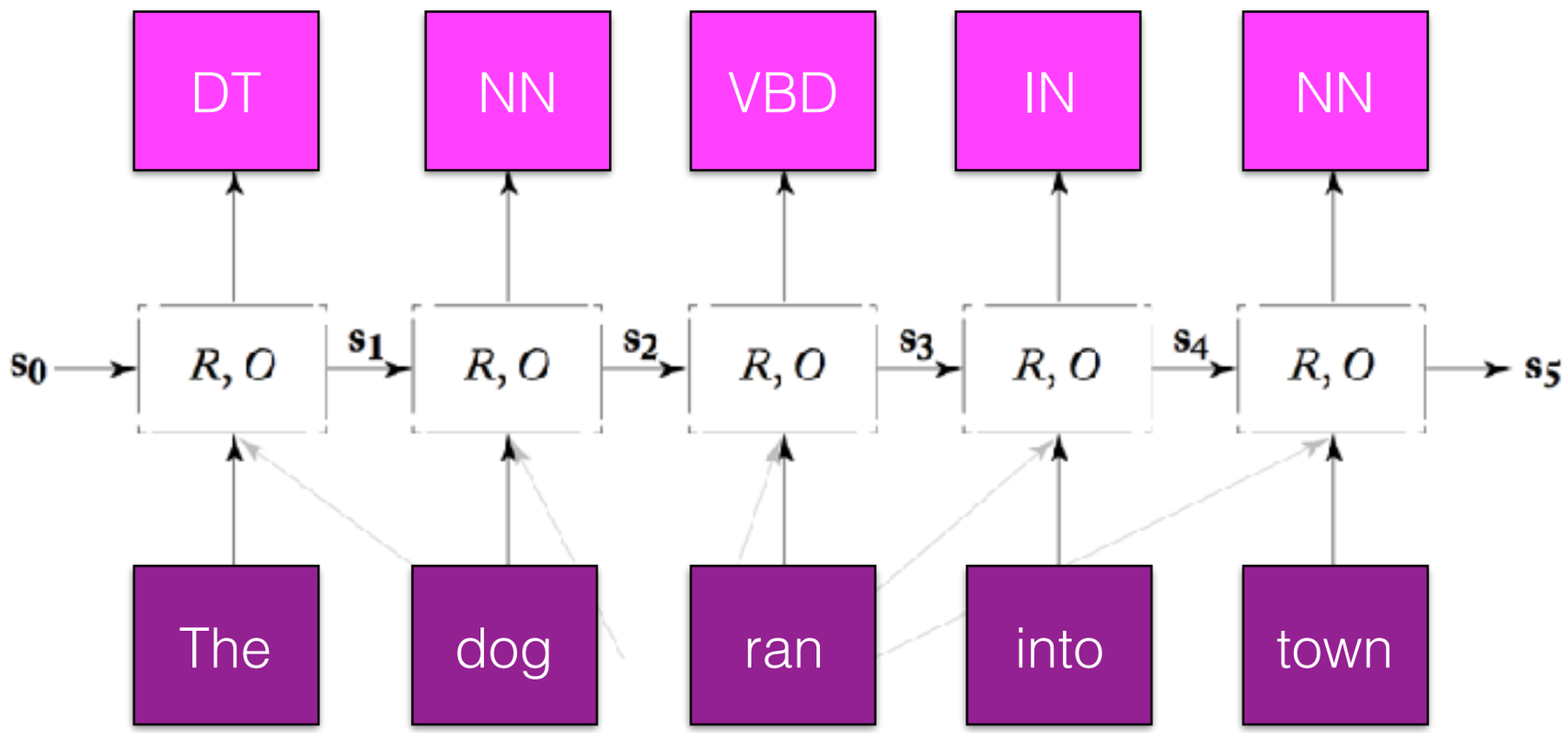
$$P(y \mid x, \beta) = \prod_{i=1}^{n} P(y_i \mid y_{i-1}, x, \beta)$$

CRF

$$P(y \mid x, \beta) = \frac{\exp(\Phi(x, y)^\top \beta)}{\sum_{y' \in \mathcal{Y}} \exp(\Phi(x, y')^\top \beta)}$$

Recurrent neural network

- For POS tagging, predict the tag conditioned on the context



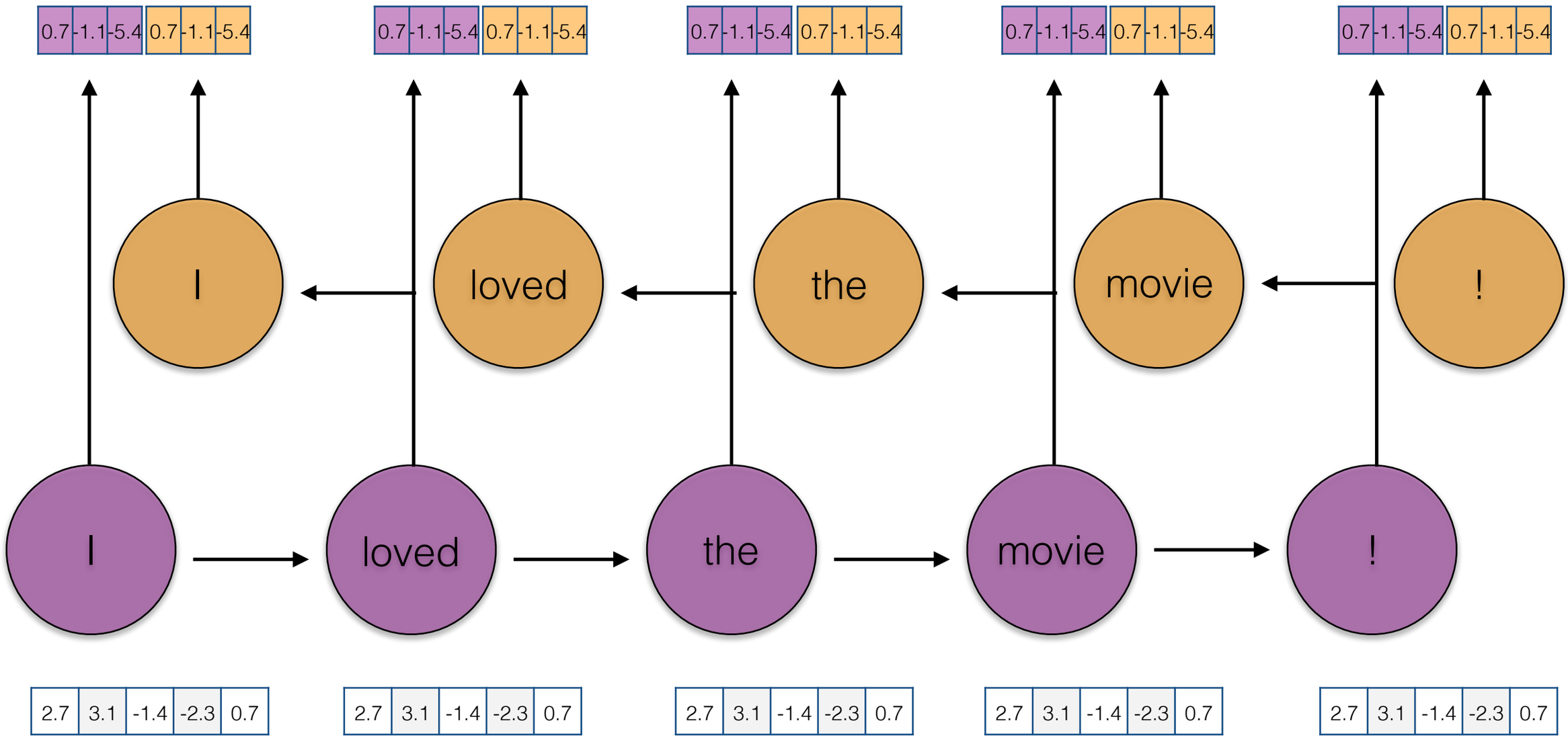
Bidirectional RNN

- A powerful alternative is make predictions conditioning both on the past and the future.

- Two RNNs
    - One running left-to-right
    - One right-to-left

- Each produces an output vector at each time step, which we concatenate

Bidirectional RNN



sequence labeling problems?

Sequence labeling

- Sequence labeling problems make a labeling decision at each timestep

B-PER	I-PER	O	O	O	O	B-ORG
Tim	Cook	is	the	CEO	of	Apple

0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
#	b	l	a	c	k	l	i	v	e	s	m	a	t	t	e	r

trees

PCFG

- Probabilistic context-free grammar: each production is also associated with a probability.

- This lets us calculate the probability of a parse for a given sentence; for a given parse tree T for sentence S comprised of n rules from R (each A → β):

$$P(T, S) = \prod_{i}^{n} P(\beta \mid A)$$

Estimating PCFGs

$$\sum_{\beta} P(\beta \mid A) = \frac{C(A \to \beta)}{\sum_{\gamma} C(A \to \gamma)}$$

(equivalently)

$$\sum_{\beta} P(\beta \mid A) = \frac{C(A \to \beta)}{C(A)}$$

A		β	P(β | NP)
NP	→	NP PP	0.092
NP	→	DT NN	0.087
NP	→	NN	0.047
NP	→	NNS	0.042
NP	→	DT JJ NN	0.035
NP	→	NNP	0.034
NP	→	NNP NNP	0.029
NP	→	JJ NNS	0.027
NP	→	QP -NONE-	0.018
NP	→	NP SBAR	0.017
NP	→	NP PP-LOC	0.017
NP	→	JJ NN	0.015
NP	→	DT NNS	0.014
NP	→	CD	0.014
NP	→	NN NNS	0.013
NP	→	DT NN NN	0.013
NP	→	NP CC NP	0.013

Dependency parsing

- Transition-based parsing
  - O(n)
  - Only projective structures (pseudo-projective [Nivre and Nilsson 2005])
- Graph-based parsing
  - $O(n^2)$
  - Projective and non-projective trees

Transition-based parsing

- Basic idea: parse a sentence into a dependency by training a local classifier to predict a parser's next action from its current configuration.

Configuration

- Stack
- Input buffer of words
- Arcs in a parsed dependency tree
- Parsing = sequences of transitions through space of possible configurations

- This is a multi class classification problem: given the current configuration — i.e., the elements in the stack, the words in the buffer, and the arcs created so far, what's the best transition?

Output space $\boldsymbol{y}$ =

Shift
LeftArc(nsubj)
RightArc(nsubj)
LeftArc(det)
RightArc(det)
LeftArc(obj)
RightArc(obj)
...

Features are scoped over the stack, buffer, and arcs created so far

stack

me

book

buffer

the morning flight

arc

feature	example
$stack_1$ = me	1
$stack_2$ = book	1
$stack_1$ POS = PRP	1
$buffer_1$ = the	1
$buffer_2$ = morning	1
$buffer_1$ = today	0
$buffer_1$ POS = RB	0
$stack_1$ = me AND $stack_2$ = book	1
$stack_1$ = PRP AND $stack_2$ = VB	1
iobj(book,*) in arcs	0

Training

We’re training to predict the parser action (Shift, RightArc, LeftArc) given the featurized configuration

Configuration features	Label
<stack1 = me, 1>, <stack2 = book, 1>, <stack1 POS = PRP, 1>, <buffer1 = the, 1>,	Shift
<stack1 = me, 0>, <stack2 = book, 0>, <stack1 POS = PRP, 0>, <buffer1 = the, 0>,	RightArc(det)
<stack1 = me, 0>, <stack2 = book, 1>, <stack1 POS = PRP, 0>, <buffer1 = the, 0>,	RightArc(nsubj)

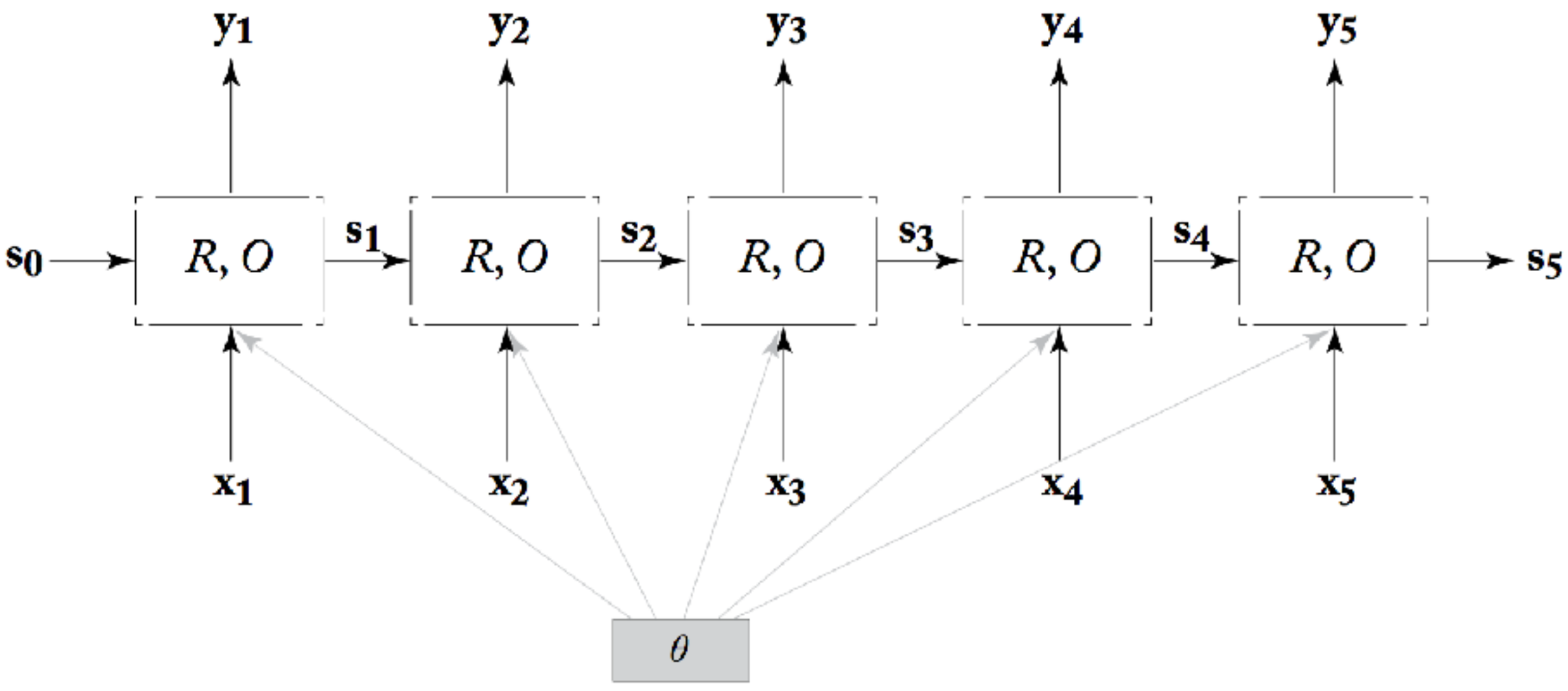
natural language generation

Language Model

Language modeling is the task of estimating P(w)

- Count and normalize
- Featurized
- Neural (RNN)

Recurrent neural network



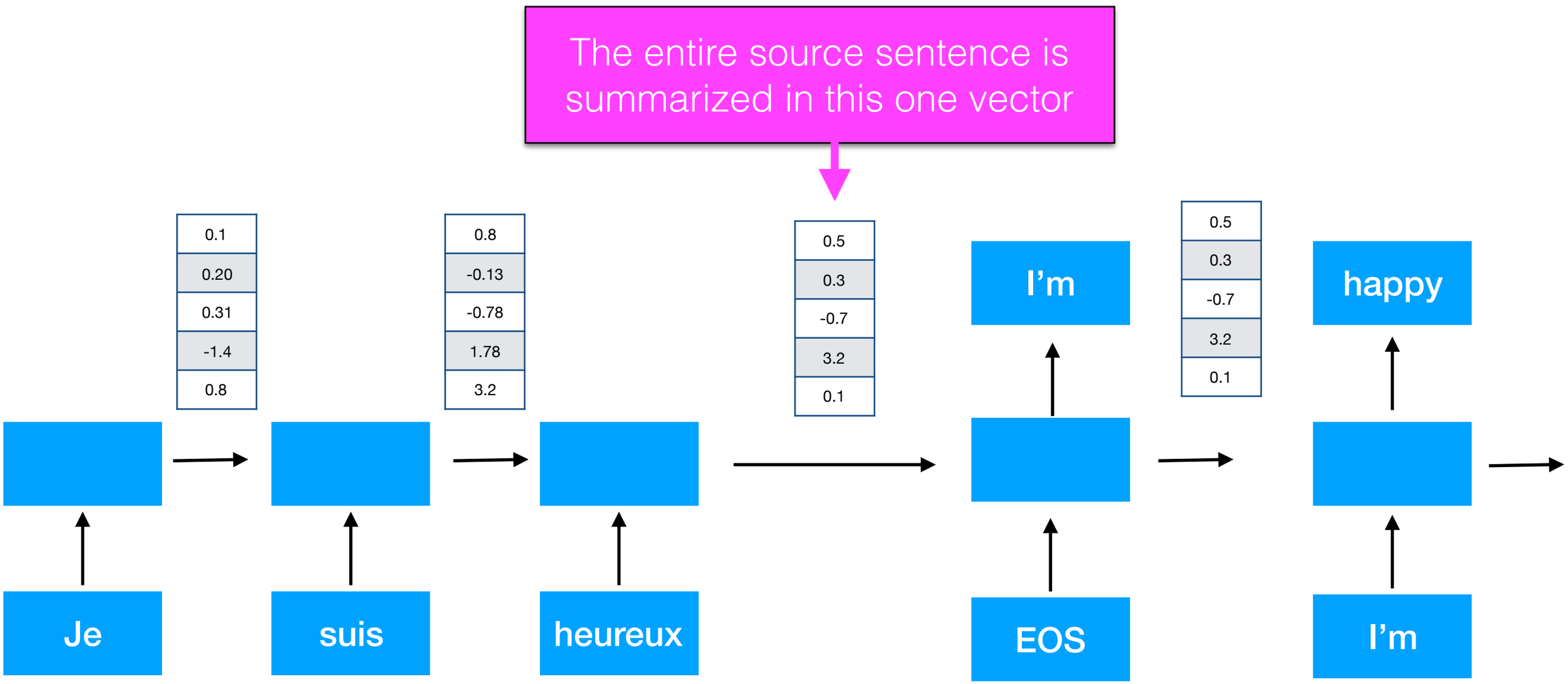
Goldberg 2017

Encoder-decoder framework

- Language modeling: predict a word given its left context
- Conversation: predict a word given its left context and the dialogue context.
- Machine translation: predict a word given its left context and the full text of the source.
- Basic idea: encode some context into a fixed vector; and then decode a new sentence from that embedding.

Encoder-decoder

The entire source sentence is summarized in this one vector



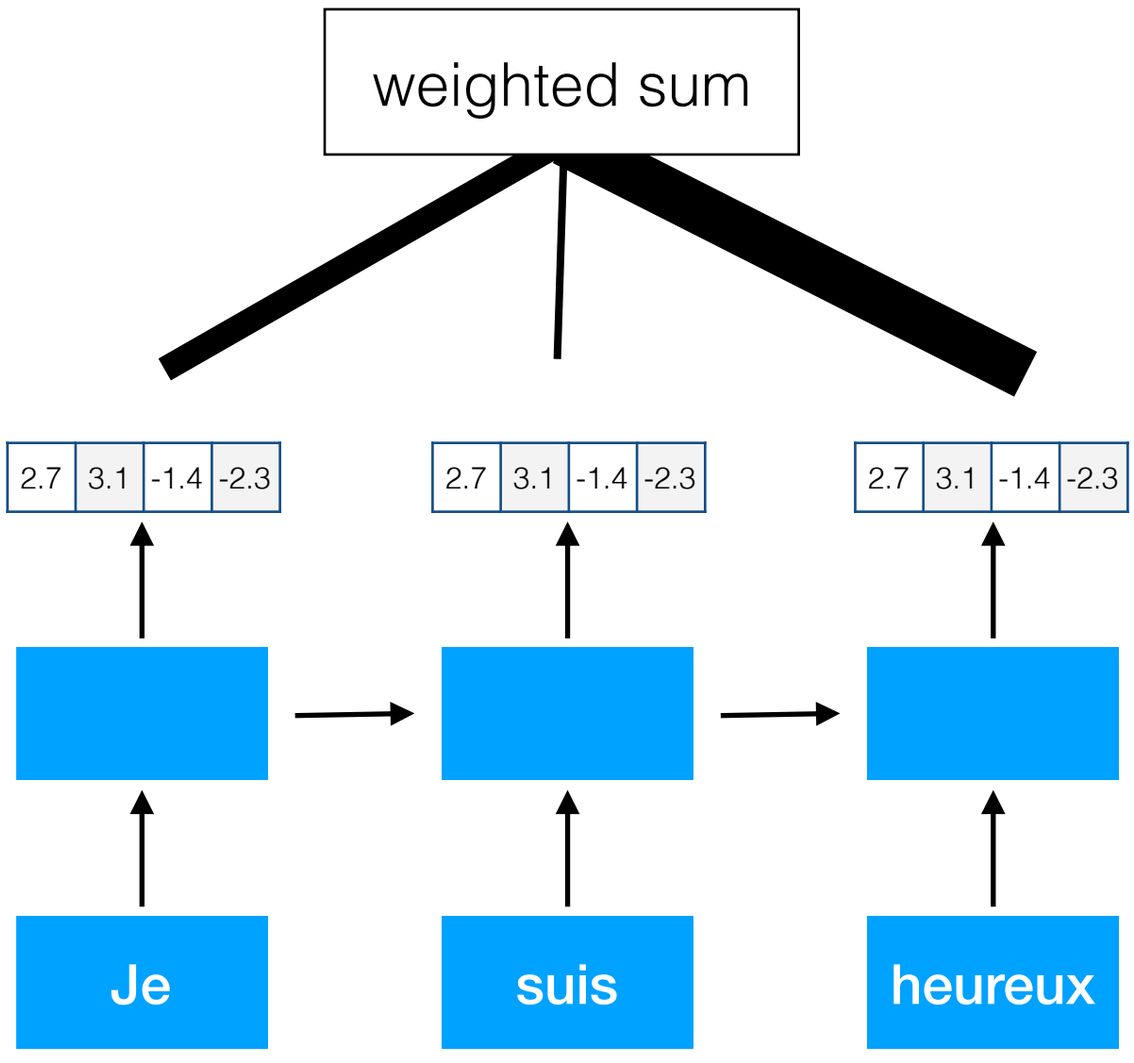
The decoder state depends just on the previous *state* and the previous *output*

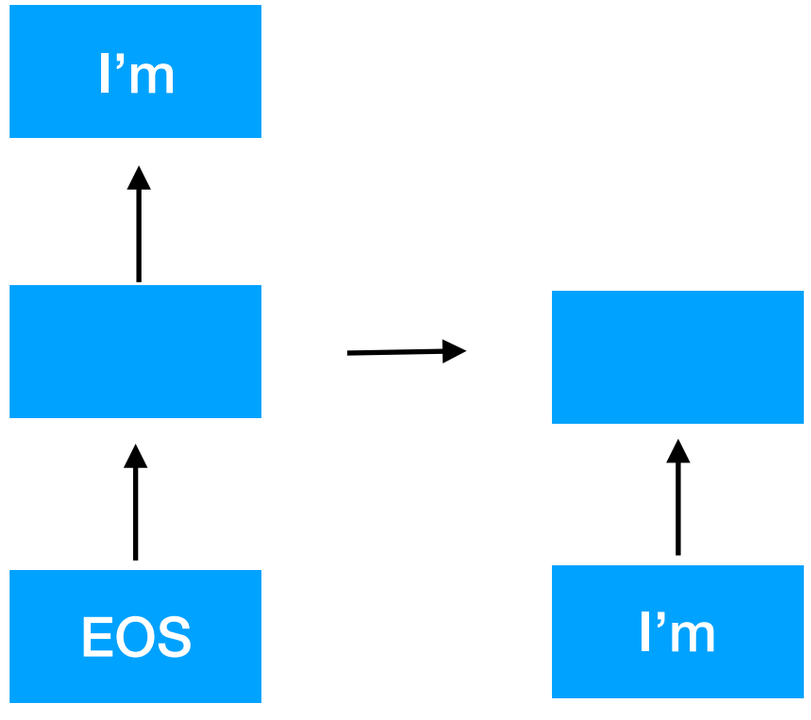
$$s_i = f(s_{i-1}, y_{i-1})$$

Encoder-decoder with attention

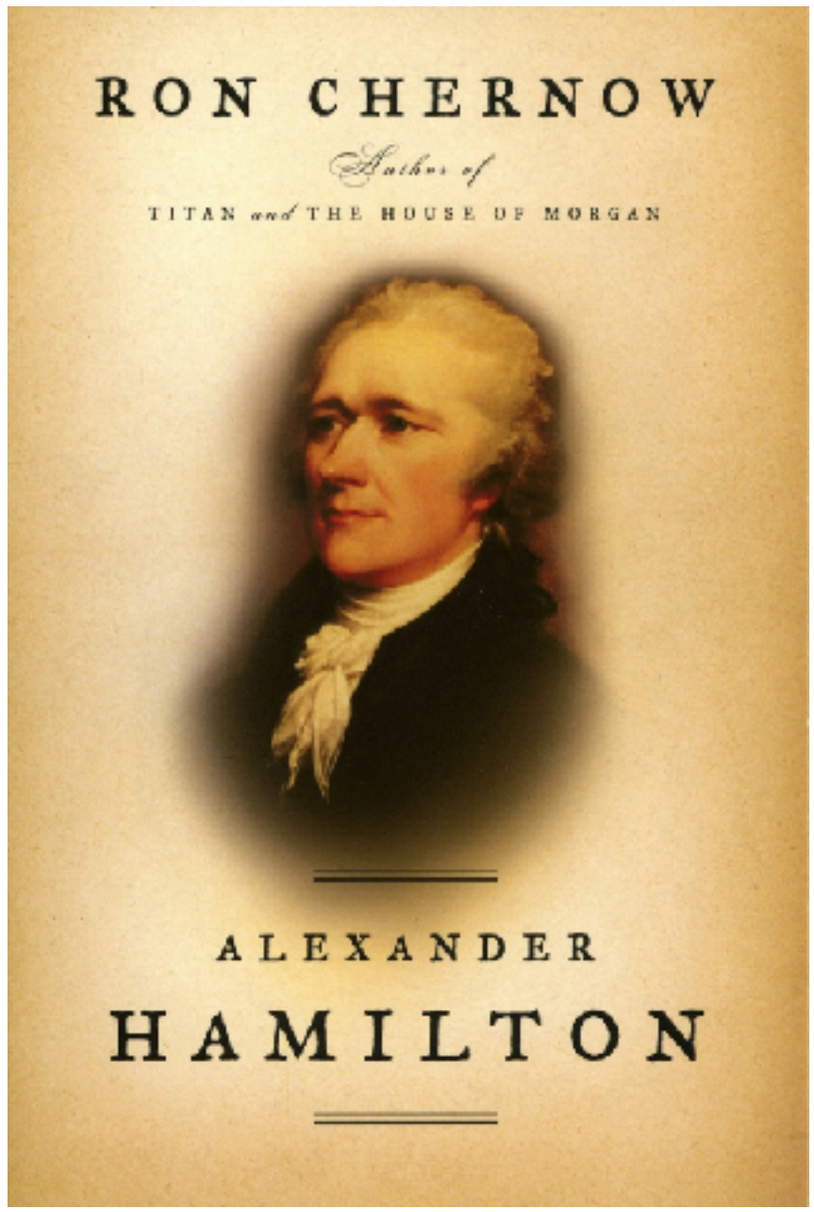
$$c = h_1 a_1 + h_2 a_2 + h_3 a_3$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

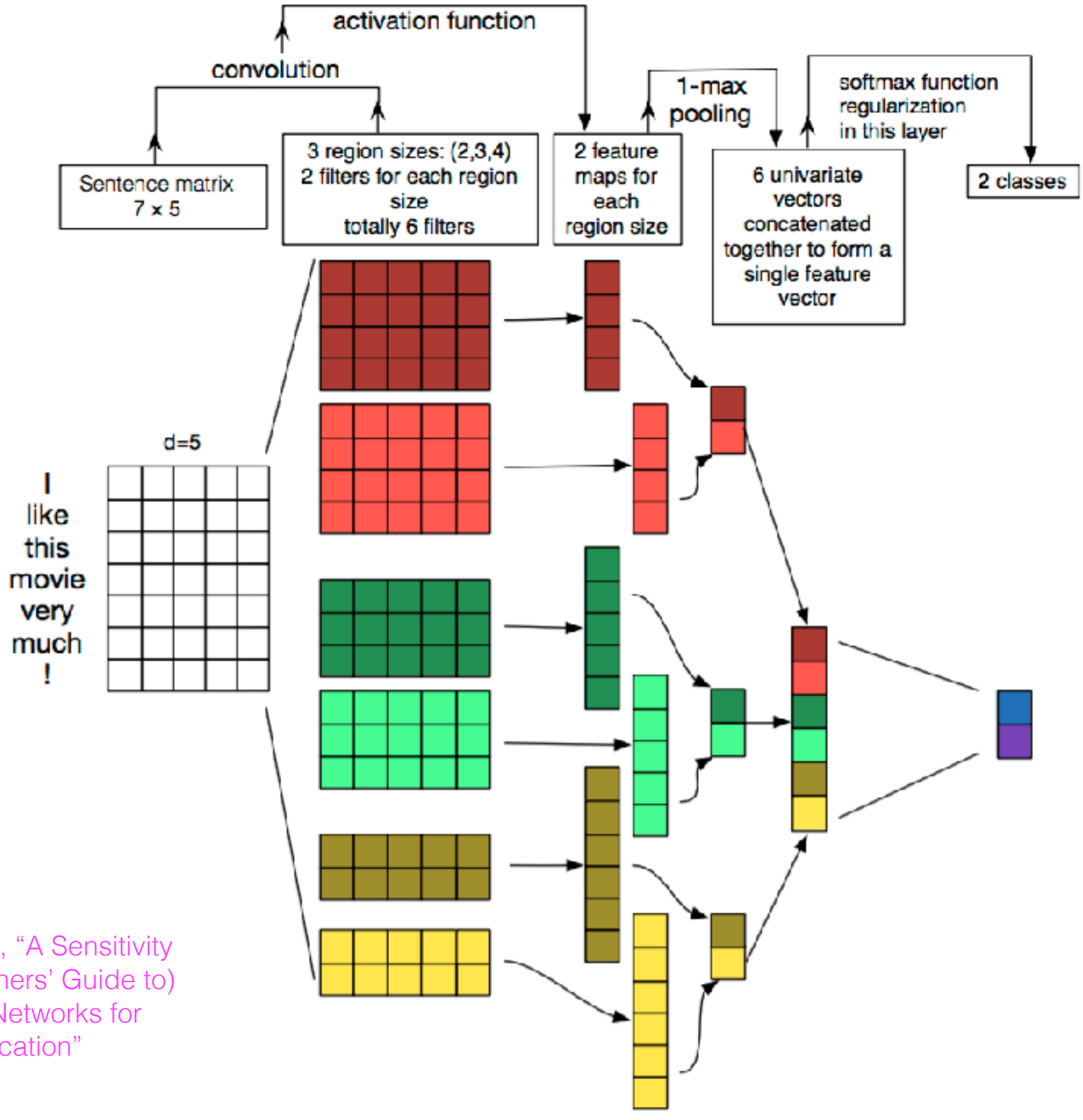




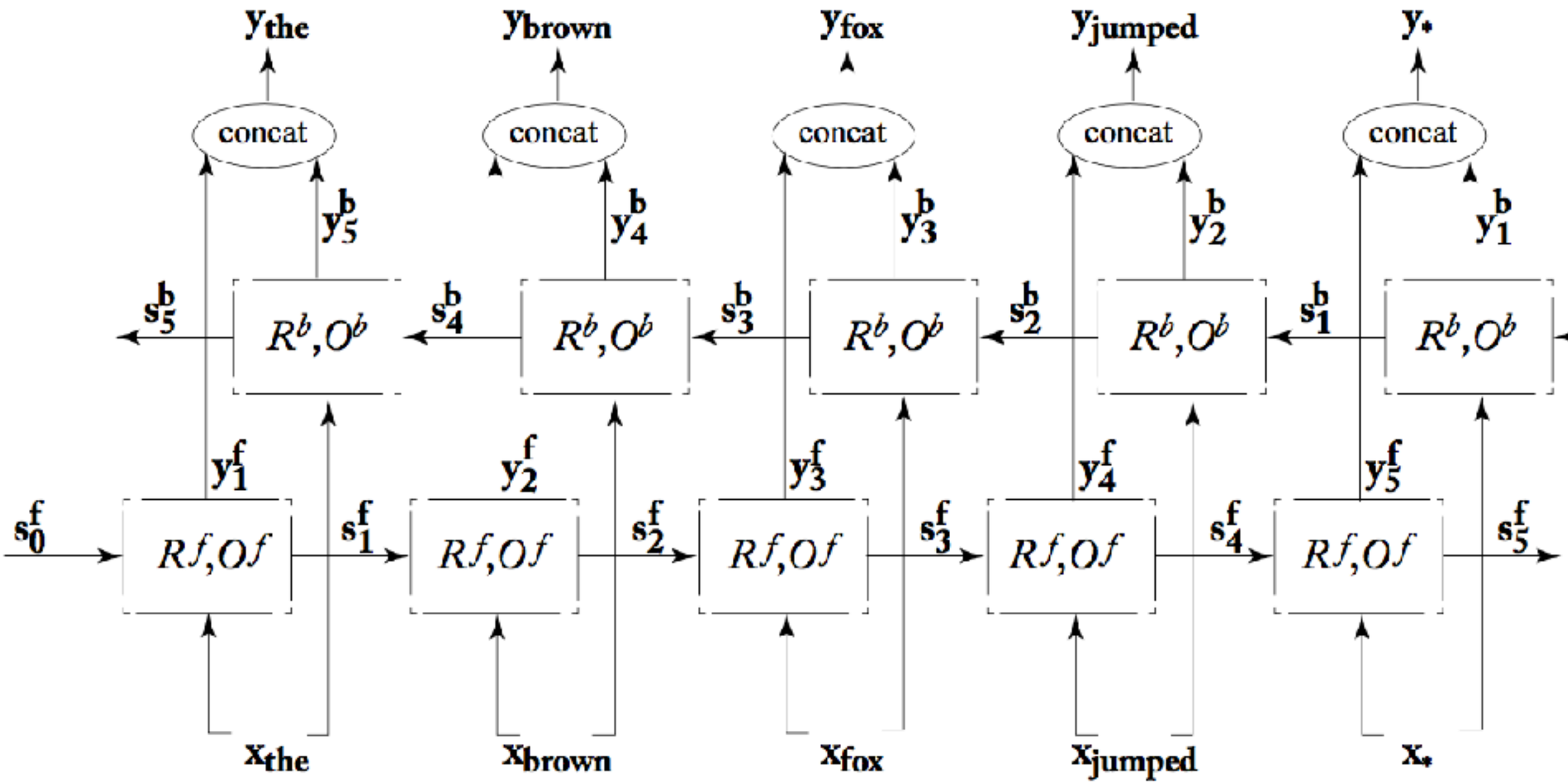
Summarization



3. Neural methods are generally* better



Zhang and Wallace 2016, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification”



Goldberg 2017

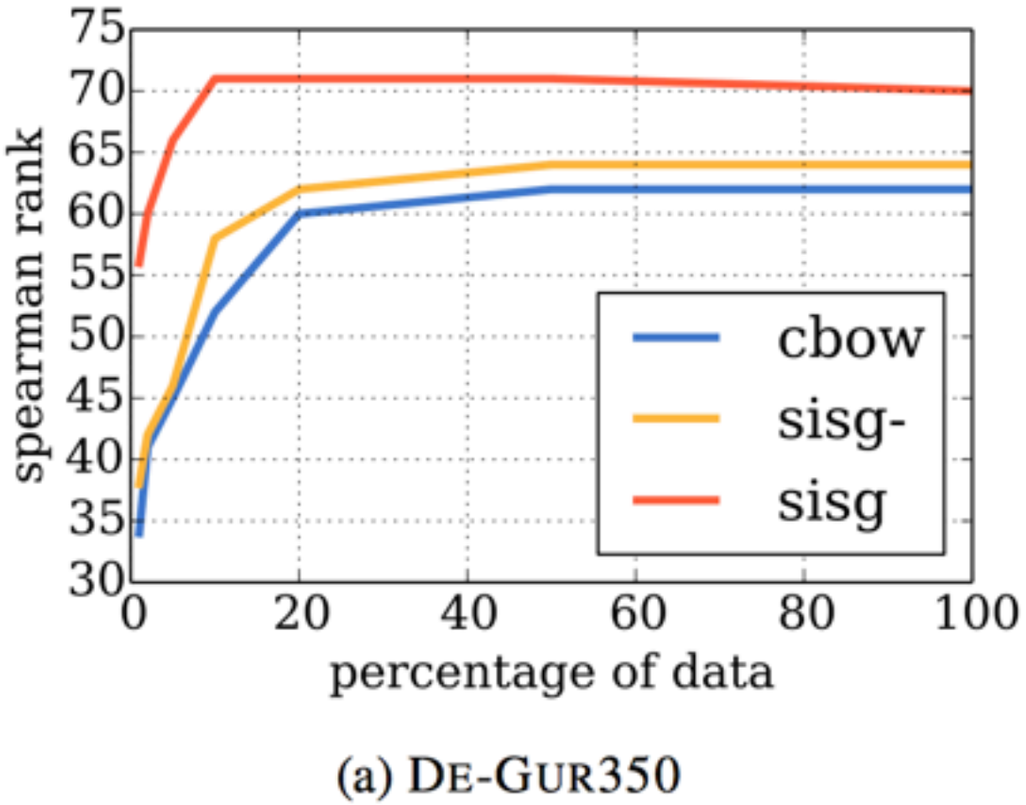
Subword models

- Rather than learning a single representation for each word type w, learn representations z for the set of ngrams $\mathcal{G}_w$ that comprise it [Bojanowski et al. 2017]

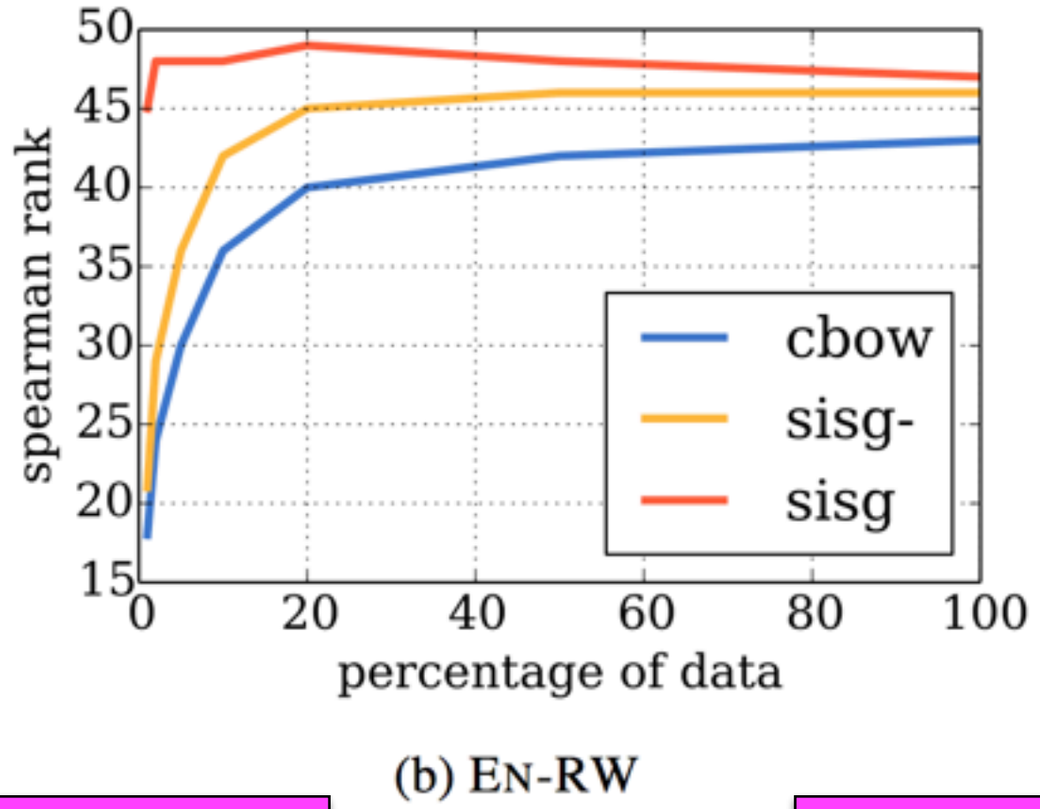
- The word itself is included among the ngrams (no matter its length).

- A word representation is the sum of those ngrams

$$w = \sum_{g \in \mathcal{G}_w} z_g$$



(a) DE-GUR350



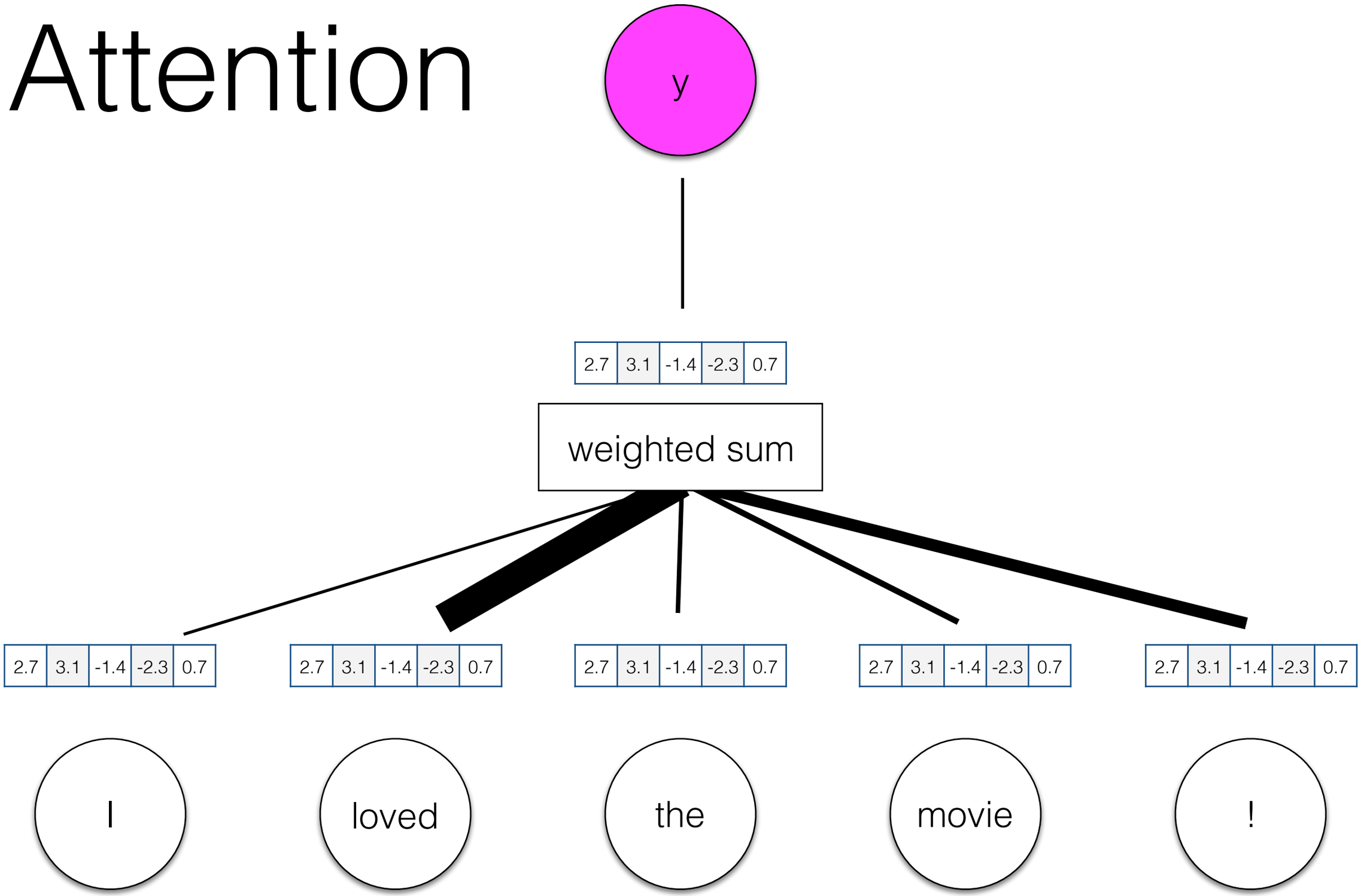
(b) EN-RW

1% =
~20M tokens

100% =
~1B tokens

- Subword models need less data to get comparable performance.

Attention


y
2.7 3.1 -1.4 -2.3 0.7
weighted sum
2.7 3.1 -1.4 -2.3 0.7
2.7 3.1 -1.4 -2.3 0.7
2.7 3.1 -1.4 -2.3 0.7
2.7 3.1 -1.4 -2.3 0.7
2.7 3.1 -1.4 -2.3 0.7
I
loved
the
movie
!

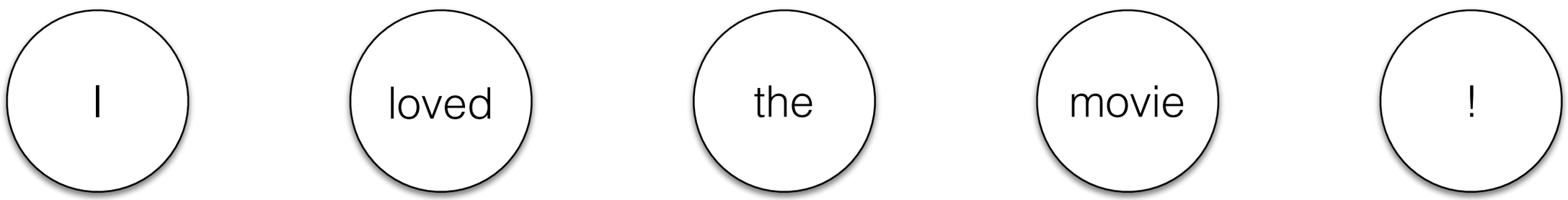
$$v \in \mathscr{R}^H$$

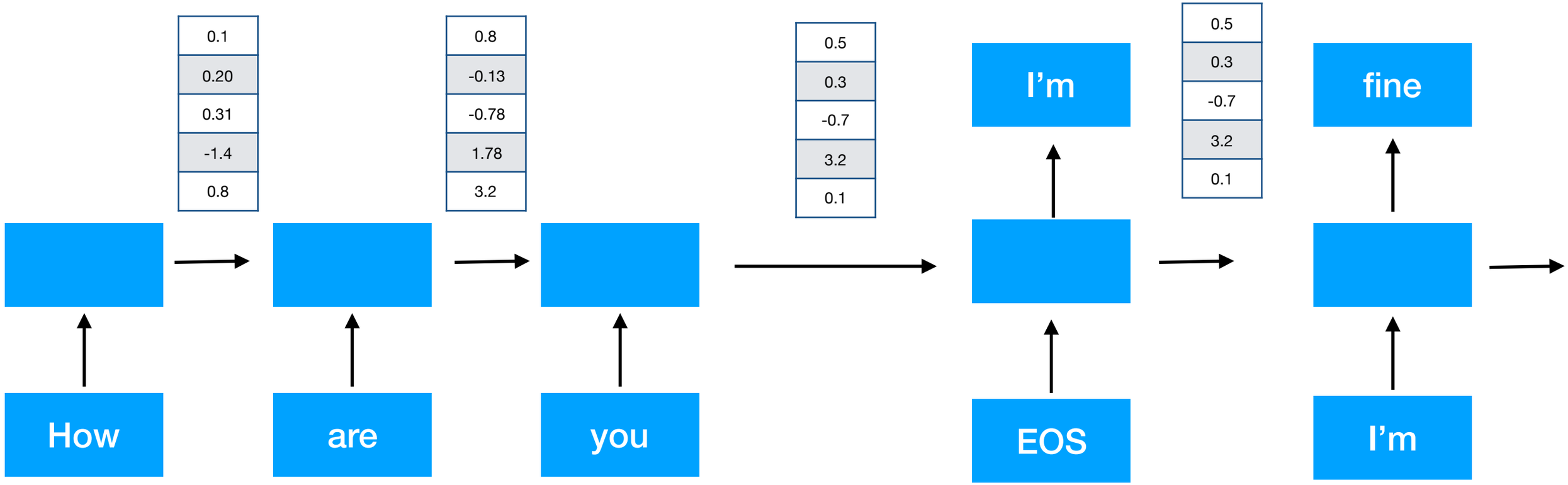
2.7	3.1	-1.4	-2.3	0.7

Define v to be a vector to be learned; think of it as an "important word" vector. The dot product here measures how similar each input vector is to that "important word" vector

$$r_1 = v^\top x_1 \qquad r_2 = v^\top x_2 \qquad r_3 = v^\top x_3 \qquad r_4 = v^\top x_4 \qquad r_5 = v^\top x_5$$

2.7	3.1	-1.4	-2.3	0.7




0.1
0.20
0.31
-1.4
0.8
0.8
-0.13
-0.78
1.78
3.2
0.5
0.3
-0.7
3.2
0.1
I’m
0.5
0.3
-0.7
3.2
0.1
fine
How
are
you
EOS
I’m

Lexical semantics

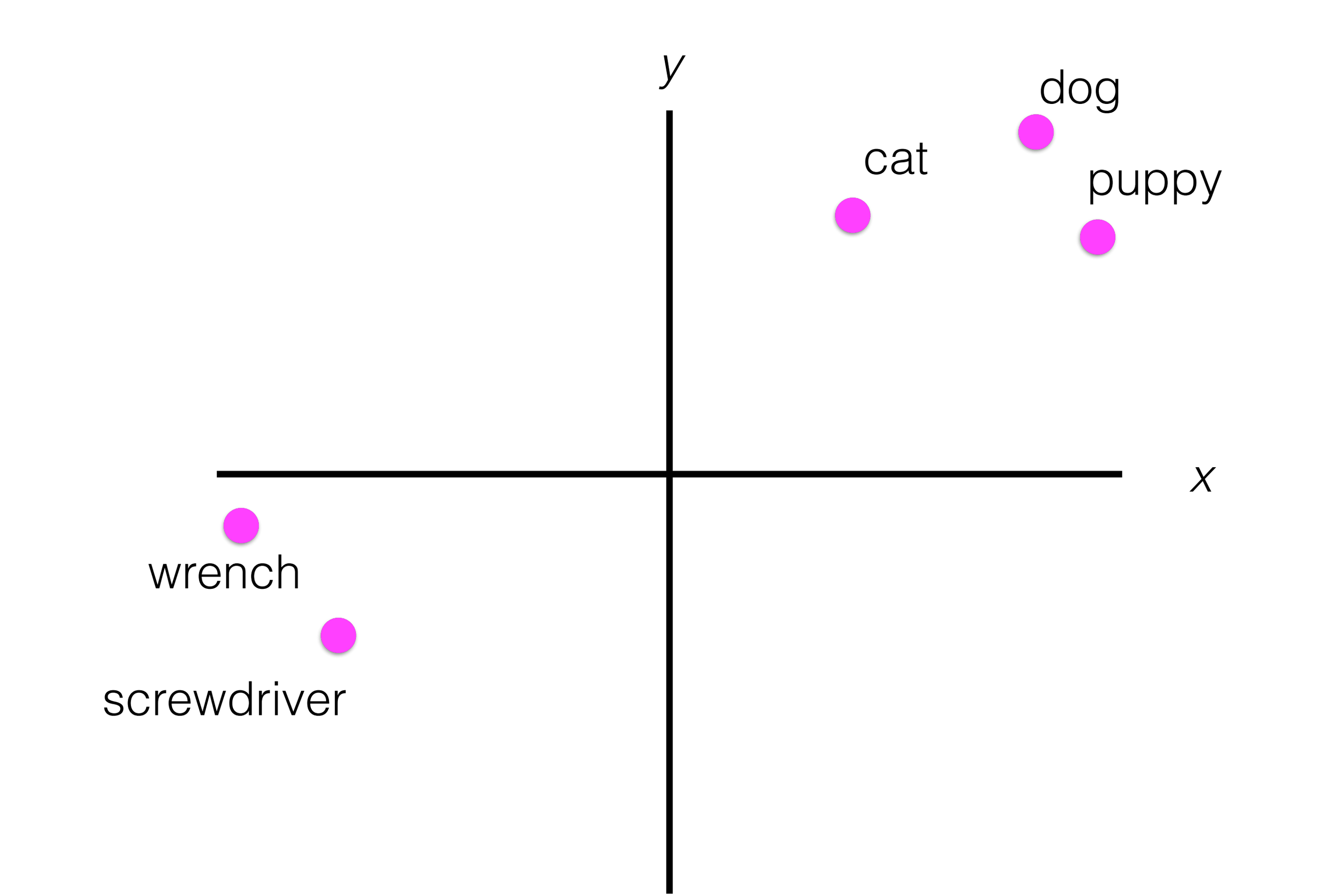
“You shall know a word by the company it keeps”

[Firth 1957]

Distributed representation

- Vector representation that encodes information about the distribution of contexts a word appears in

- Words that appear in similar contexts have similar representations (and similar meanings, by the distributional hypothesis).

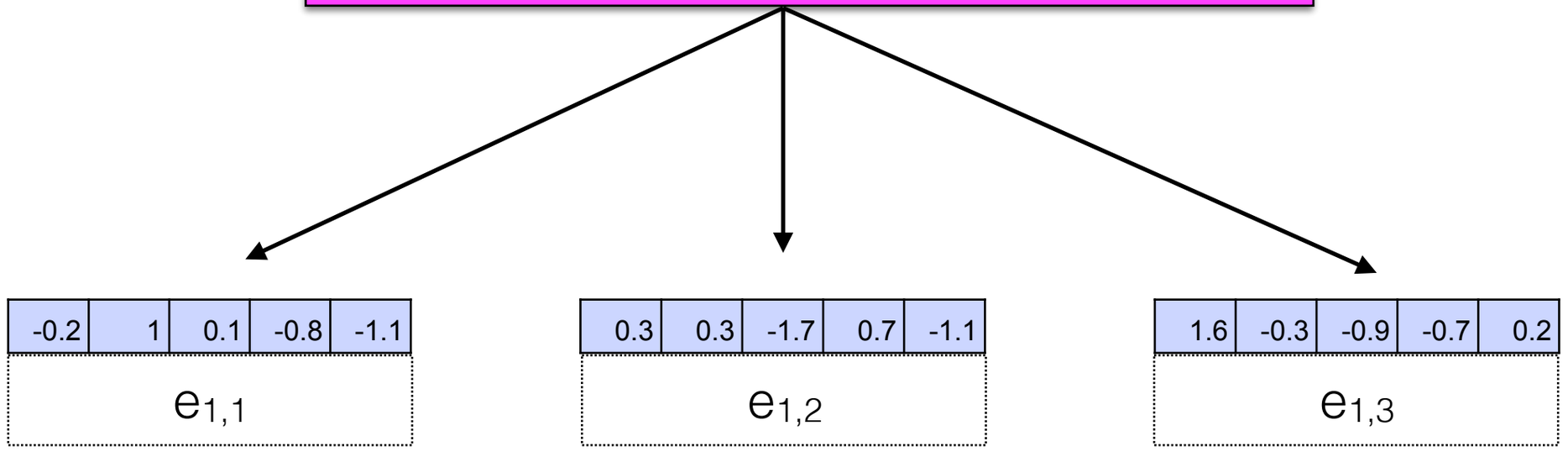

y
dog
cat
puppy
x
wrench
screwdriver

BERT

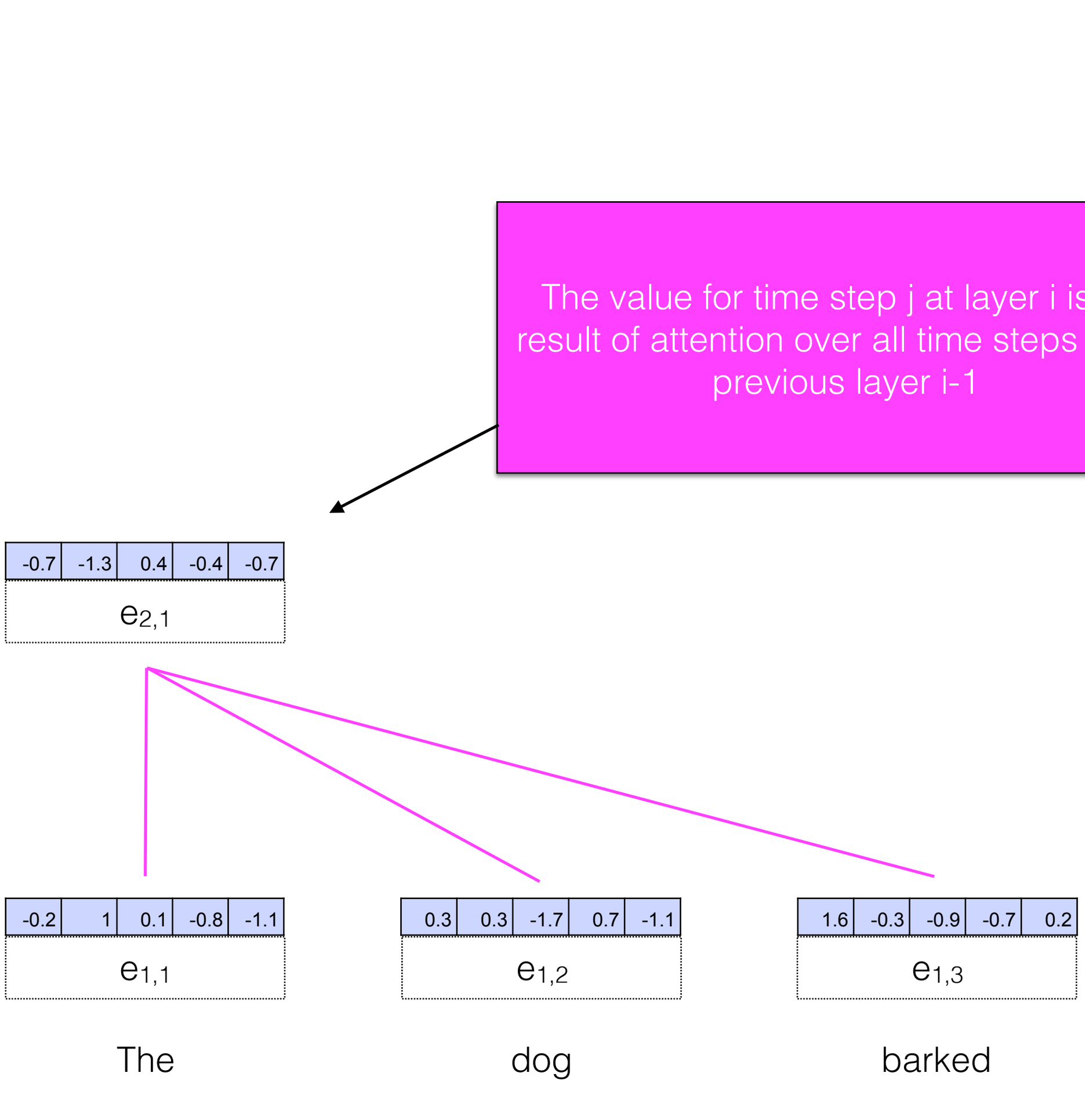
- Transformer-based model (Vaswani et al. 2017) to predict masked word using bidirectional context + next sentence prediction.

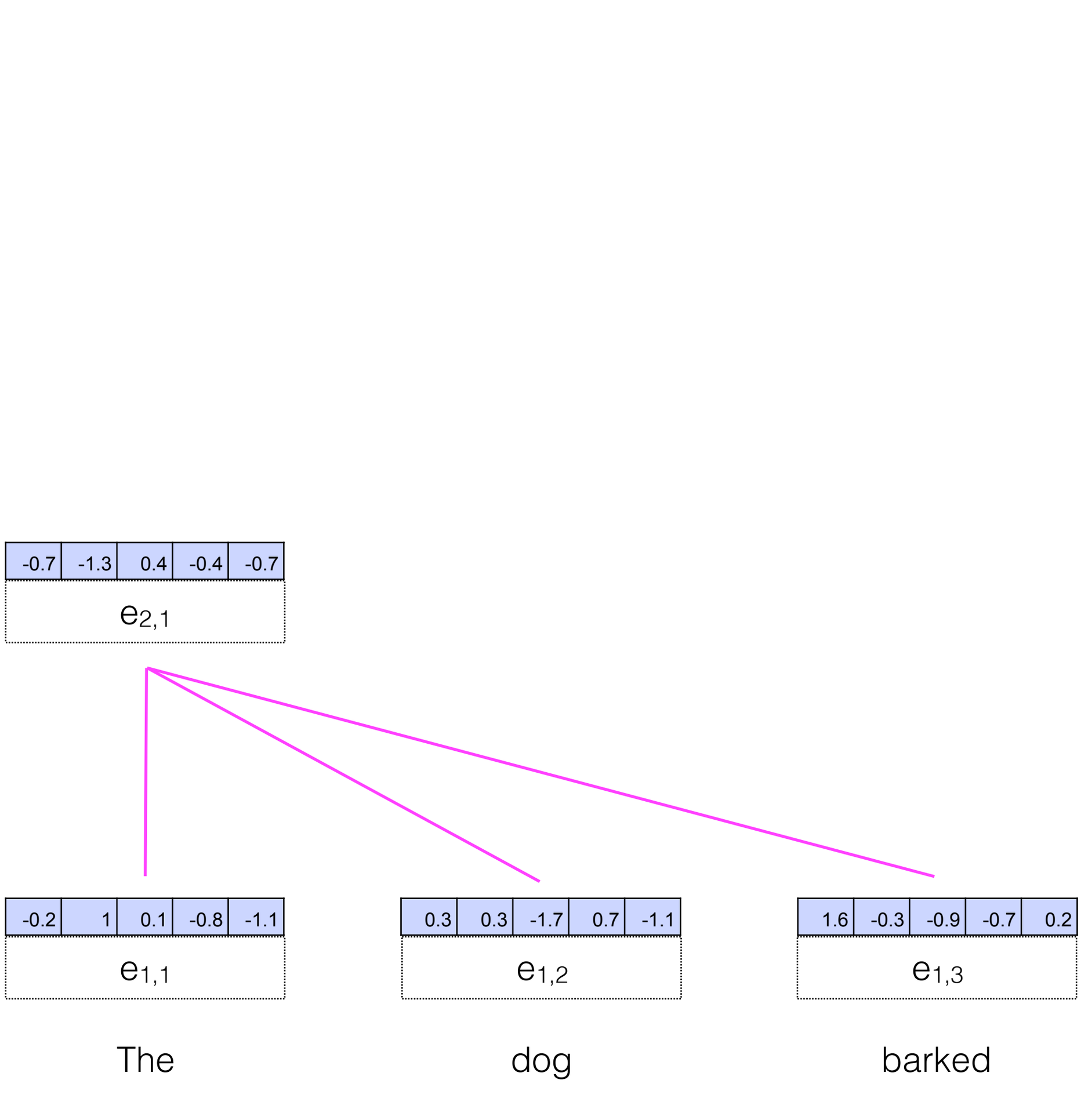
- Generates multiple layers of representations for each token sensitive to its context of use.

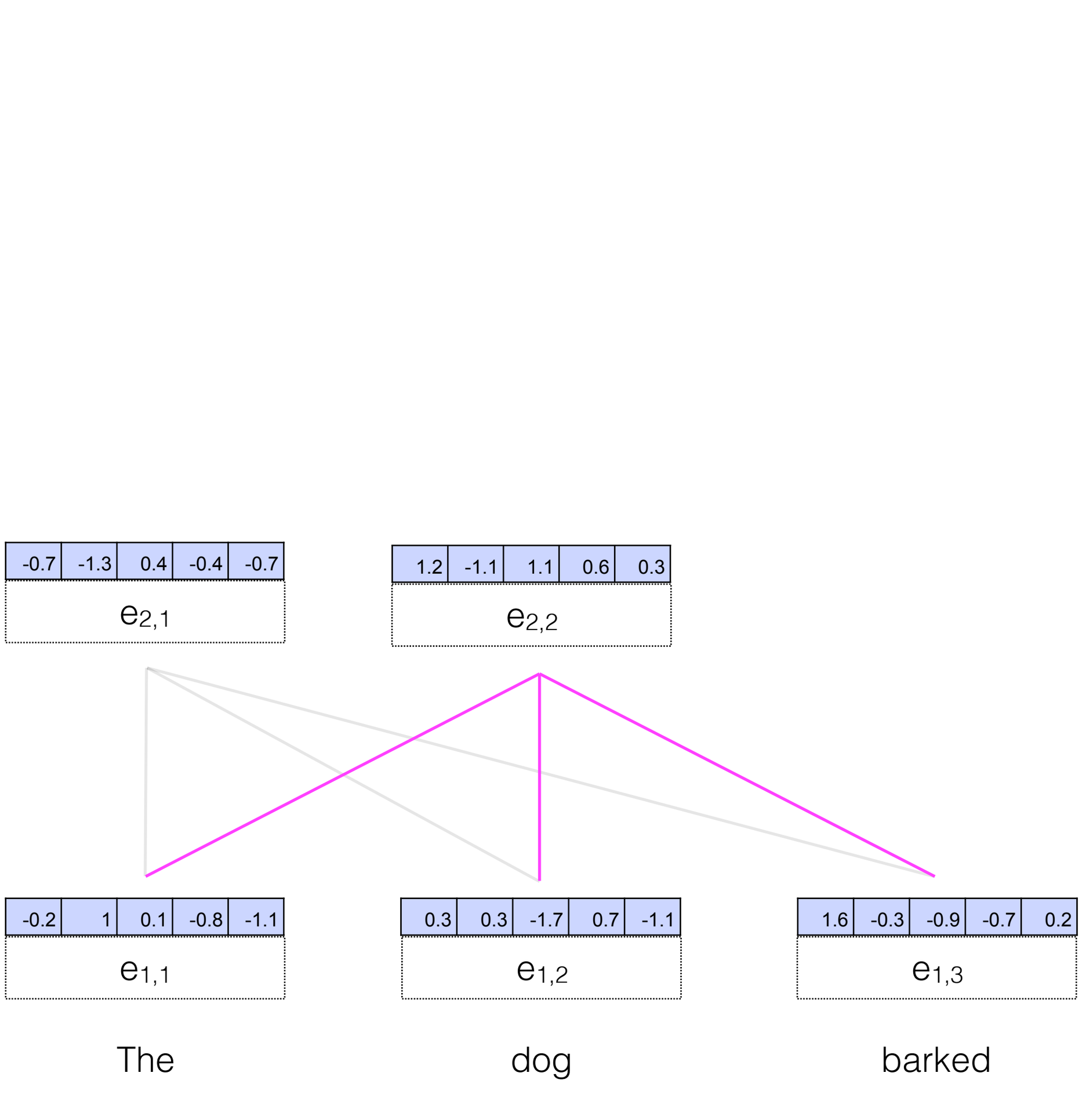
Each token in the input starts out represented by token and position embeddings

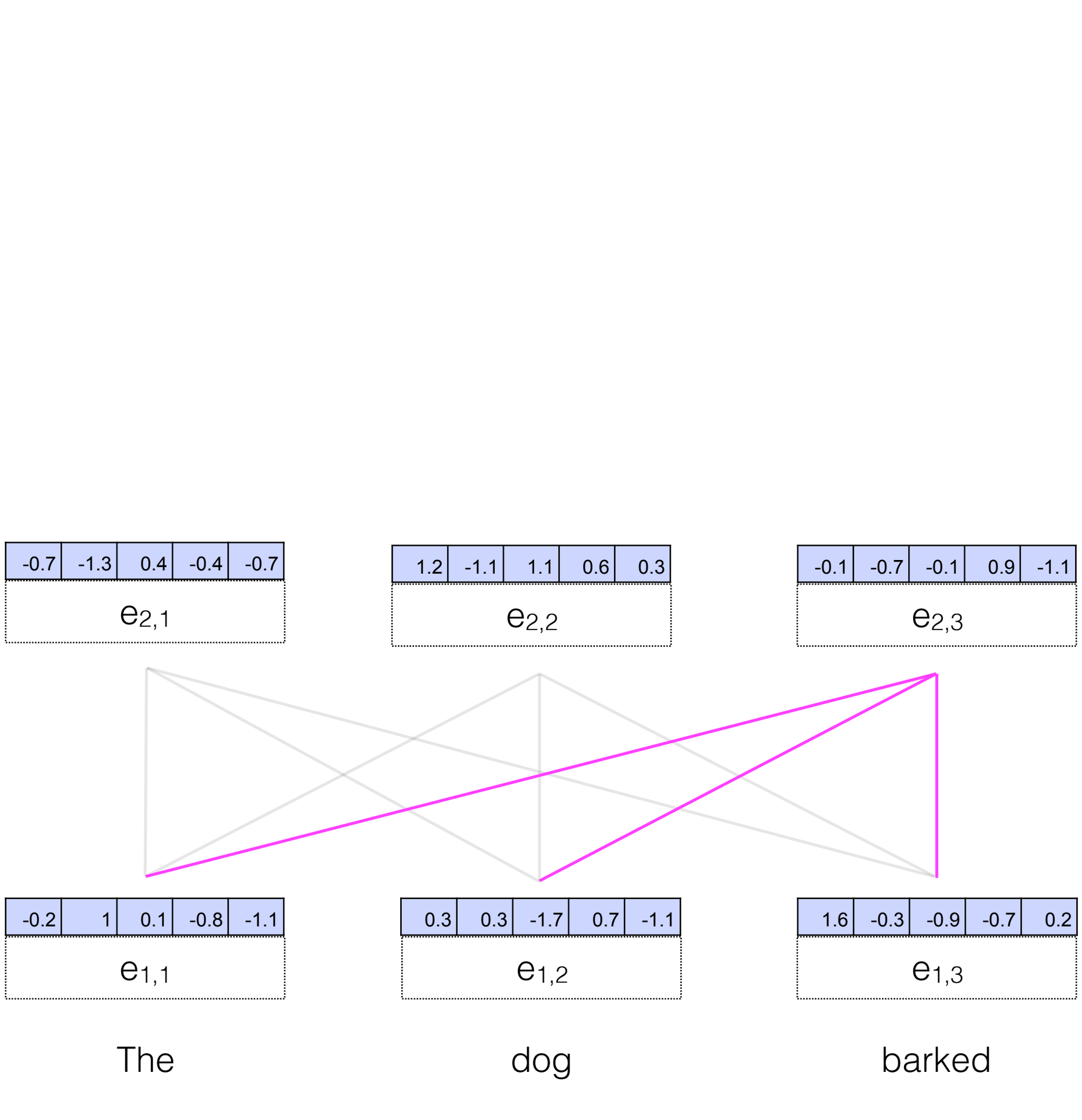


The value for time step j at layer i is the result of attention over all time steps in the previous layer i-1




-0.7 -1.3 0.4 -0.4 -0.7
e2,1
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked


-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

-0.2	0.3	2.1	1.2	0.6

$e_{3,1}$

-0.7	-1.3	0.4	-0.4	-0.7

$e_{2,1}$

1.2	-1.1	1.1	0.6	0.3

$e_{2,2}$

-0.1	-0.7	-0.1	0.9	-1.1

$e_{2,3}$

-0.2	1	0.1	-0.8	-1.1

$e_{1,1}$

0.3	0.3	-1.7	0.7	-1.1

$e_{1,2}$

1.6	-0.3	-0.9	-0.7	0.2

$e_{1,3}$

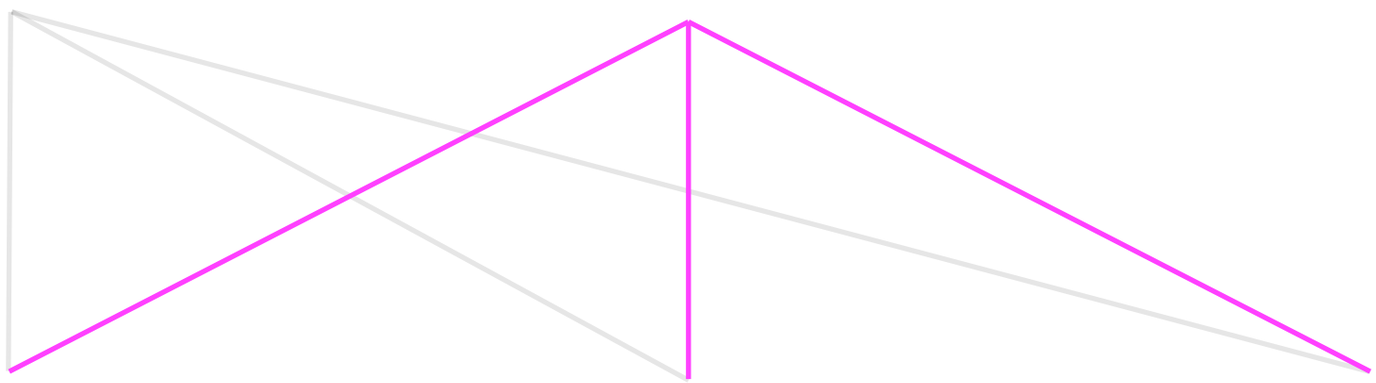
The dog barked

-0.2	0.3	2.1	1.2	0.6

$e_{3,1}$

-1.8	-0.2	-2.4	-0.2	-0.1

$e_{3,2}$



-0.7	-1.3	0.4	-0.4	-0.7

$e_{2,1}$

1.2	-1.1	1.1	0.6	0.3

$e_{2,2}$

-0.1	-0.7	-0.1	0.9	-1.1

$e_{2,3}$

-0.2	1	0.1	-0.8	-1.1

$e_{1,1}$

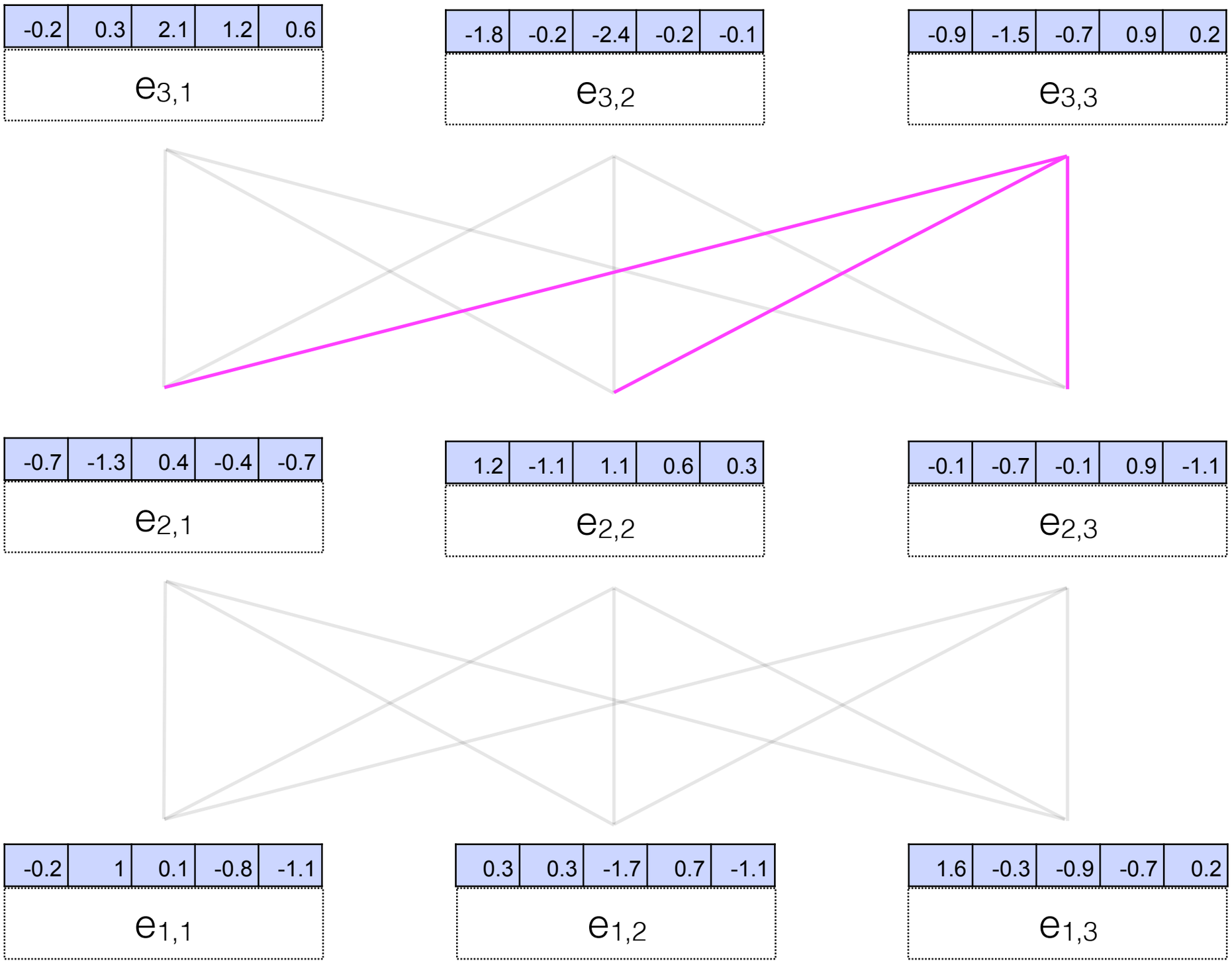
0.3	0.3	-1.7	0.7	-1.1

$e_{1,2}$

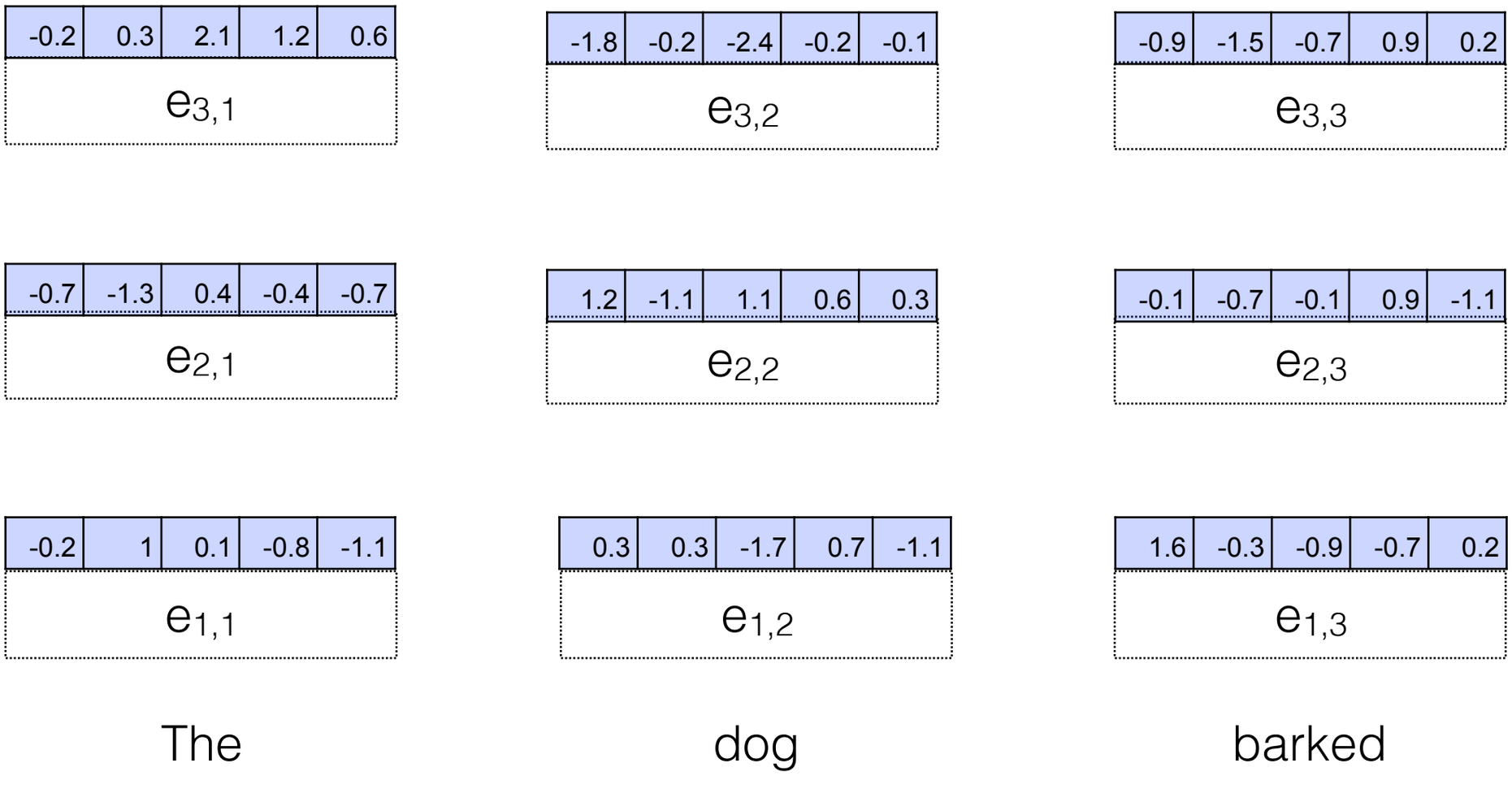
1.6	-0.3	-0.9	-0.7	0.2

$e_{1,3}$

The dog barked


-0.2 0.3 2.1 1.2 0.6
e3,1
-1.8 -0.2 -2.4 -0.2 -0.1
e3,2
-0.9 -1.5 -0.7 0.9 0.2
e3,3
-0.7 -1.3 0.4 -0.4 -0.7
e2,1
1.2 -1.1 1.1 0.6 0.3
e2,2
-0.1 -0.7 -0.1 0.9 -1.1
e2,3
-0.2 1 0.1 -0.8 -1.1
e1,1
0.3 0.3 -1.7 0.7 -1.1
e1,2
1.6 -0.3 -0.9 -0.7 0.2
e1,3
The
dog
barked

At the end of this process, we have one representation for each layer for each token



Natural language inference

A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

Bowman et al. 2016

4. Evaluation is critical

Interannotator agreement

annotator A

annotator B

	puppy	fried chicken
puppy	6	3
fried chicken	2	5

observed agreement = 11/16 = 68.75%

https://twitter.com/teenybiscuit/status/705232709220769792/photo/1

Experiment design

	training	development	testing
size	80%	10%	10%
purpose	training models	model selection	evaluation; never look at it until the very end

Metrics

- Perplexity
- Accuracy
- Precision/recall/F1
- Parseval (P/R/F1 over labeled constituents)
- Correlation with human judgments
- B^3
- BLEU

5. Text is data.

OAKLAND POLICE
1770
5005

Twitter Safety
@TwitterSafety

We have broadened our guidance on unverified claims that incite people to engage in harmful activity, could lead to the destruction or damage of critical 5G infrastructure, or could lead to widespread panic, social unrest, or large-scale disorder.

Coronavirus: Staying safe and informed on Twitter
blog.twitter.com

563 12:35 PM - Apr 22, 2020

Thanks!

Course evaluations!

https://course-evaluations.berkeley.edu/berkeley/