



Natural Language Processing

Info 159/259

Lecture 26: Review (Nov 29, 2018)

David Bamman, UC Berkeley

Announcements

- Final exam: 12/14 7-10pm, this room (LeConte 4)
- Cheat sheet (two 8.5x11" pages, double-sided)

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

1. Language has structure

“... 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

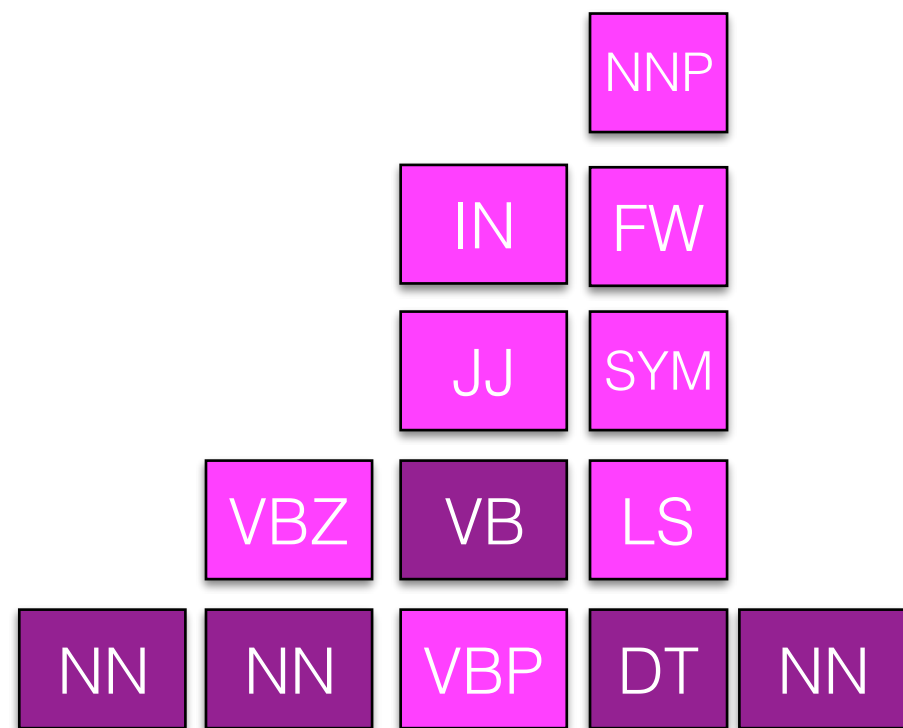
	Apocalypse now	North
the	1	1
of	0	0
hate	0	9
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
...		

Language model

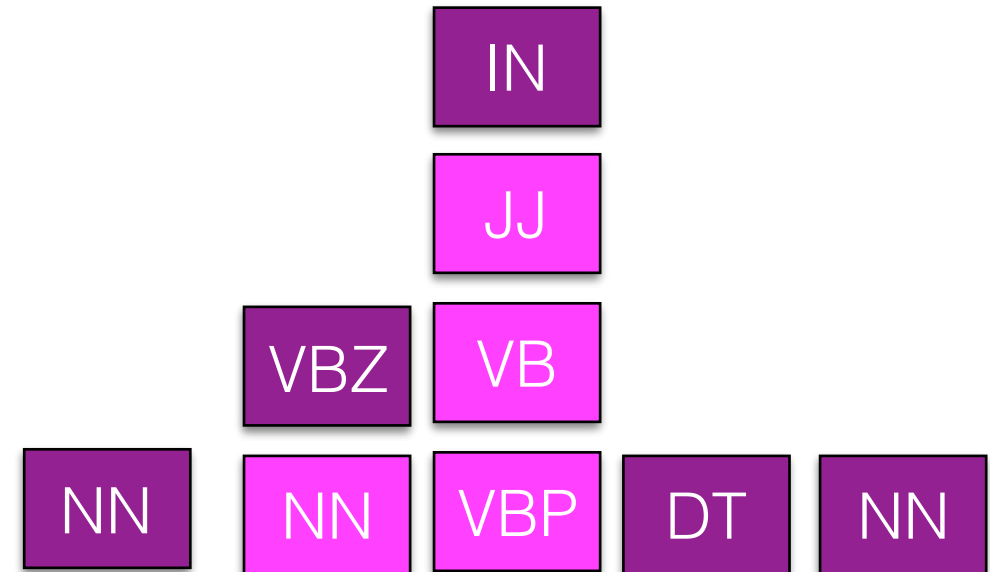
“Hillary Clinton seemed to add Benghazi to her already-long list of culprits to blame for her upset loss to Donald _____”

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 urge¹ that the city² take¹ step¹ to remedy¹ this problem²
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

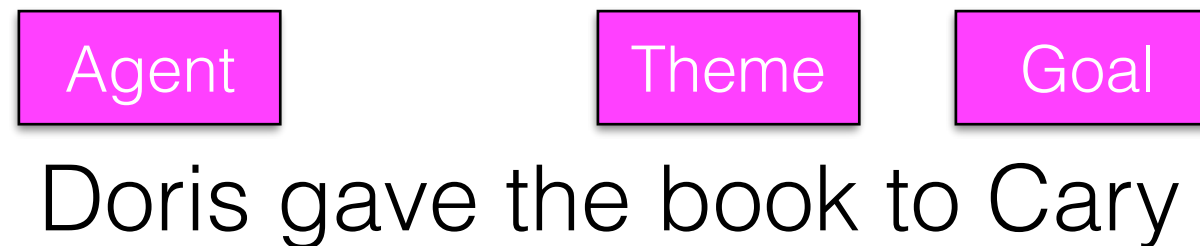
location

location

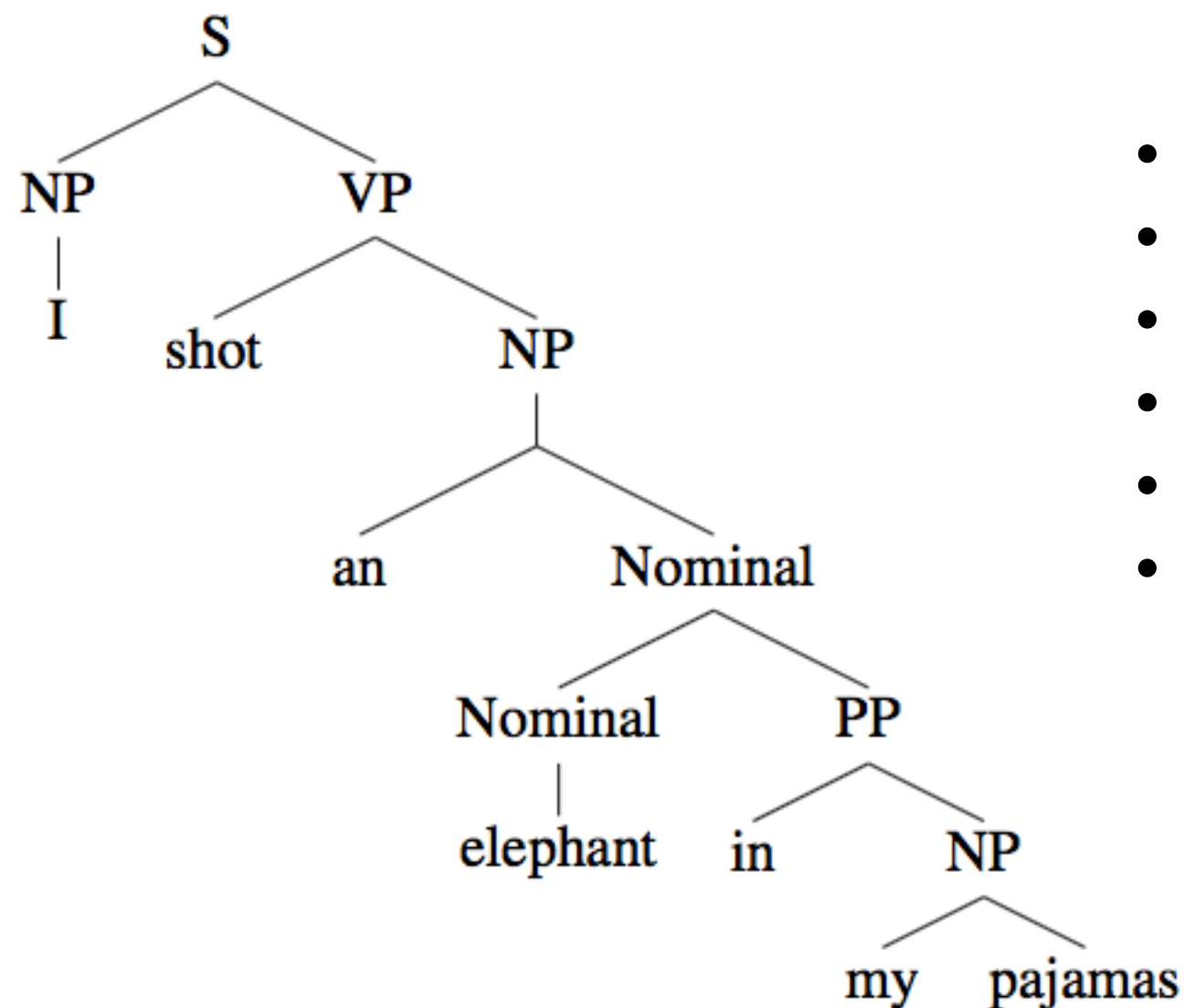
that coursed through the west campus.

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:



Phrase-structure syntax

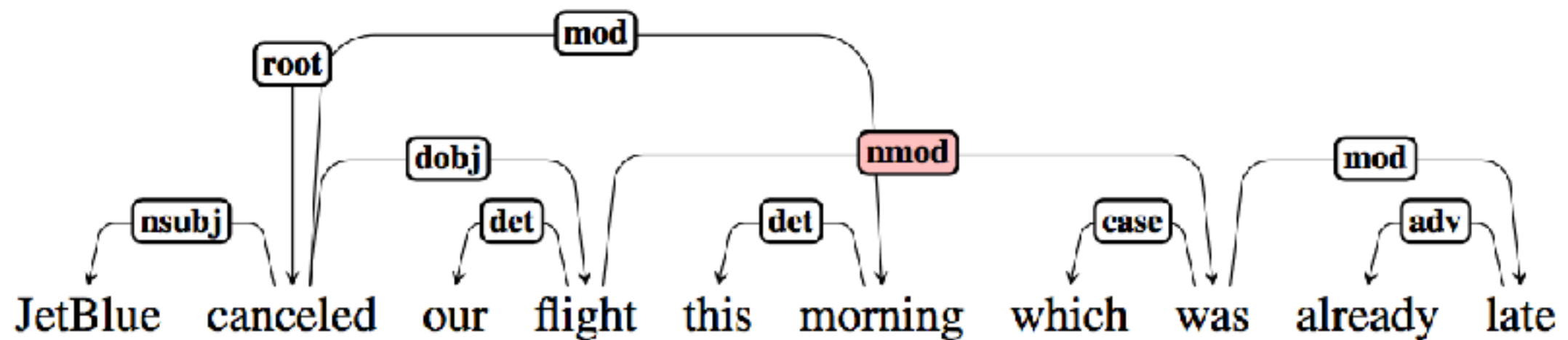


Every internal node is a phrase

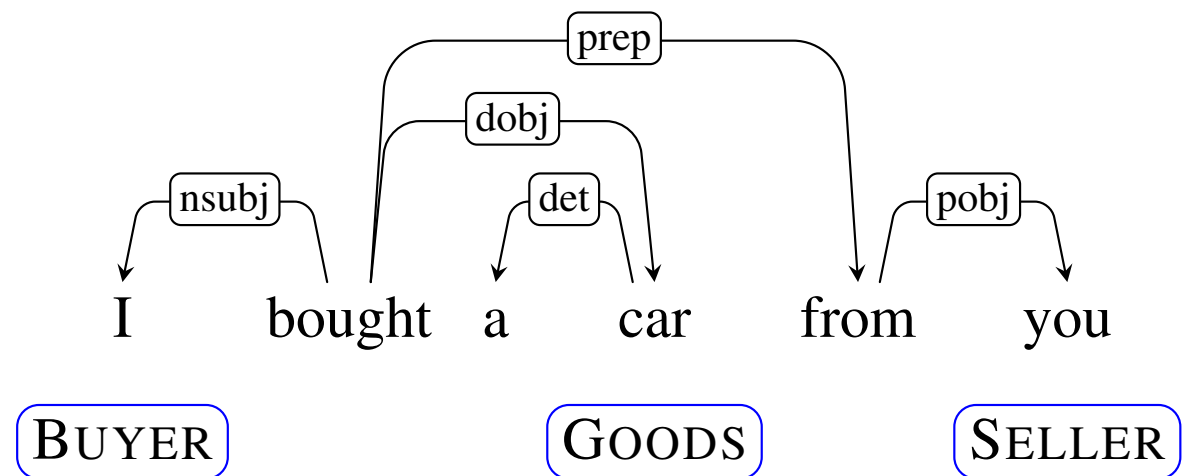
- my pajamas
- in my pajamas
- elephant in my pajamas
- an elephant in my pajamas
- shot an elephant in my pajamas
- I shot an elephant in my pajamas

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

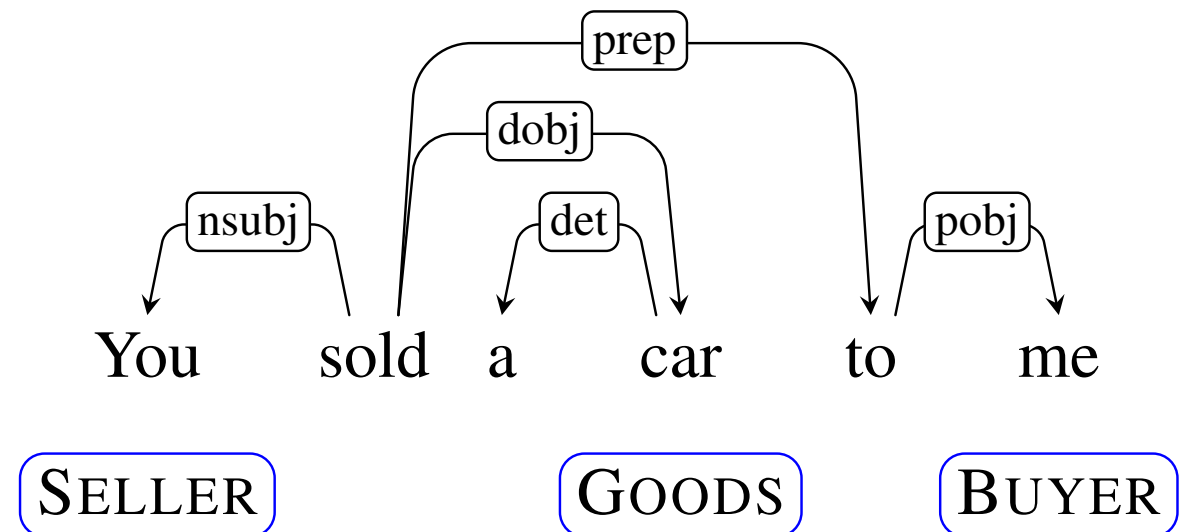
Dependency syntax



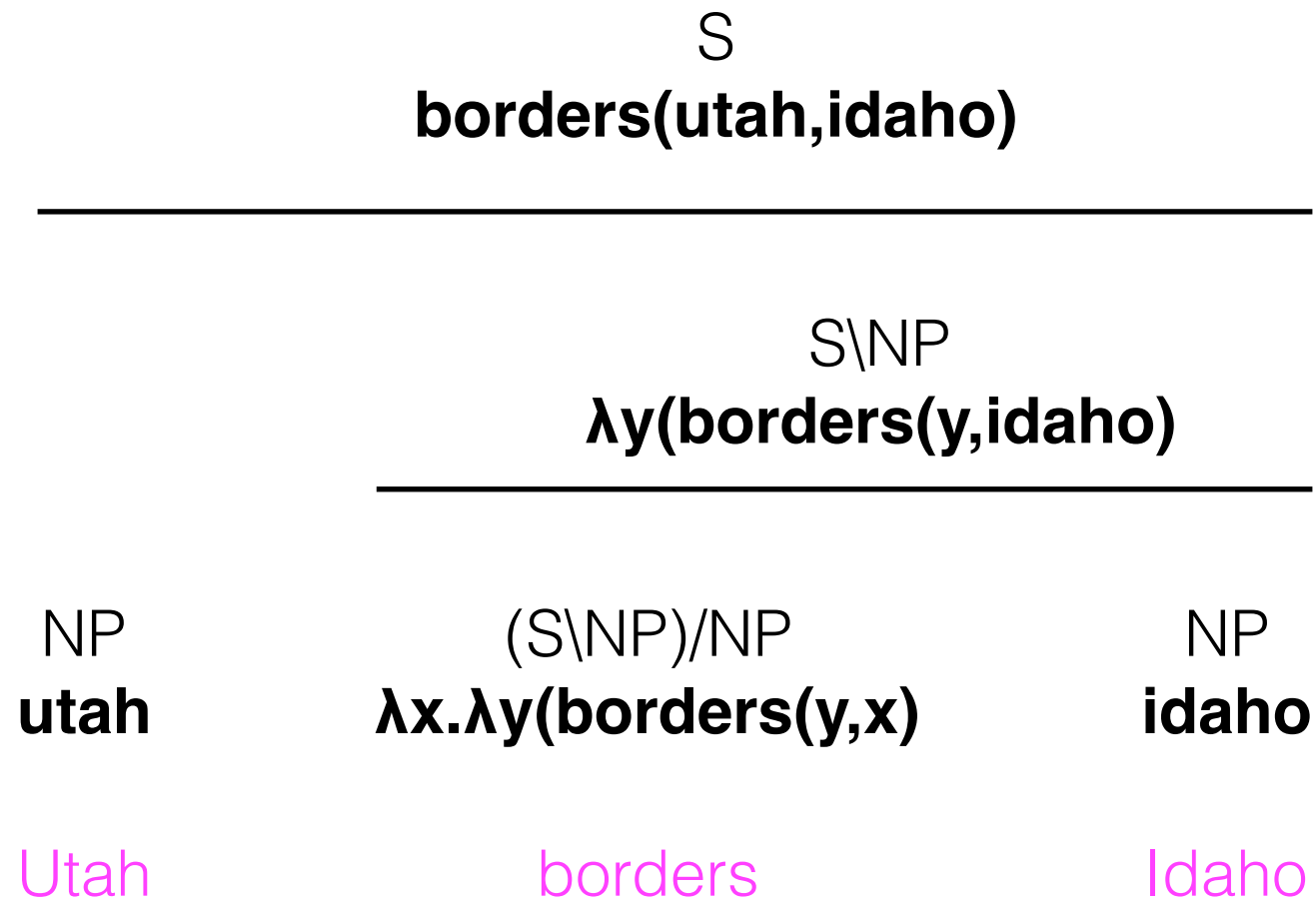
Semantic Frame



Two different perspectives on a commercial transaction



Compositional semantics





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**

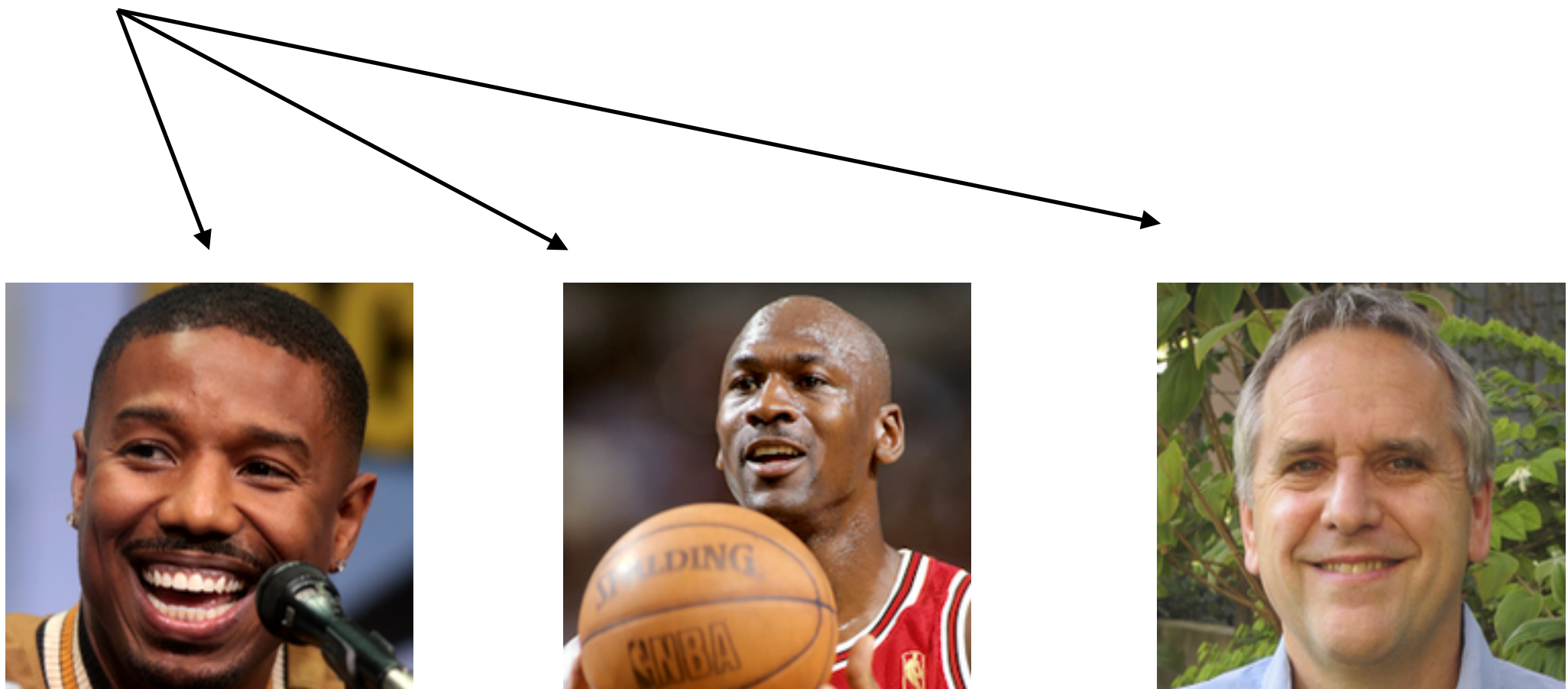
LUKE
No. No. That's not true!
That's impossible!

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

LUKE
No! No! No!

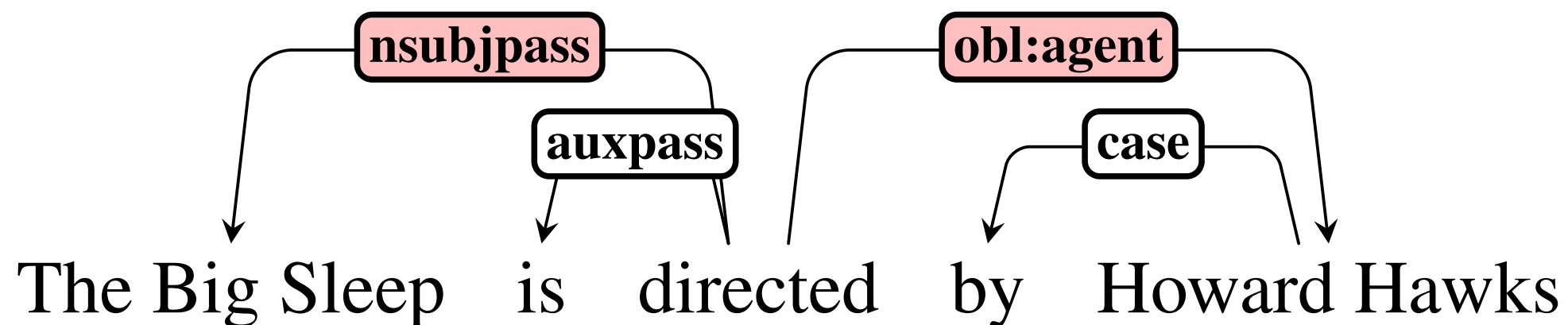
Entity linking

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} $\leftarrow$ *nsubjpass* directed $\rightarrow$ *obl:agent* [Howard Hawks]_{m2},

$m1 \leftarrow$ *nsubjpass* $\leftarrow$ directed $\rightarrow$ *obl:agent* $\rightarrow$ $m2$

Problem

Design a question answering system that takes as input a document collection and a natural language question and returns an answer.



Who wrote *White Noise*?

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 $Y = \text{positive}$
(before you see any data)

Likelihood of “really really the
worst movie ever”
given that $Y = \text{positive}$

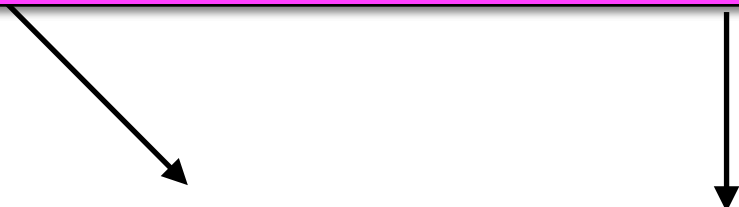
$$P(Y = y | X = x) = \frac{P(Y = y)P(X = x | Y = y)}{\sum_y P(Y = y)P(X = x | Y = y)}$$

Posterior belief that $Y = \text{positive}$ given that
 $X = \text{“really really the worst movie ever”}$

This sum ranges over
 $y = \text{positive} + y = \text{negative}$
(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:

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

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

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
<i>BIAS</i>	1

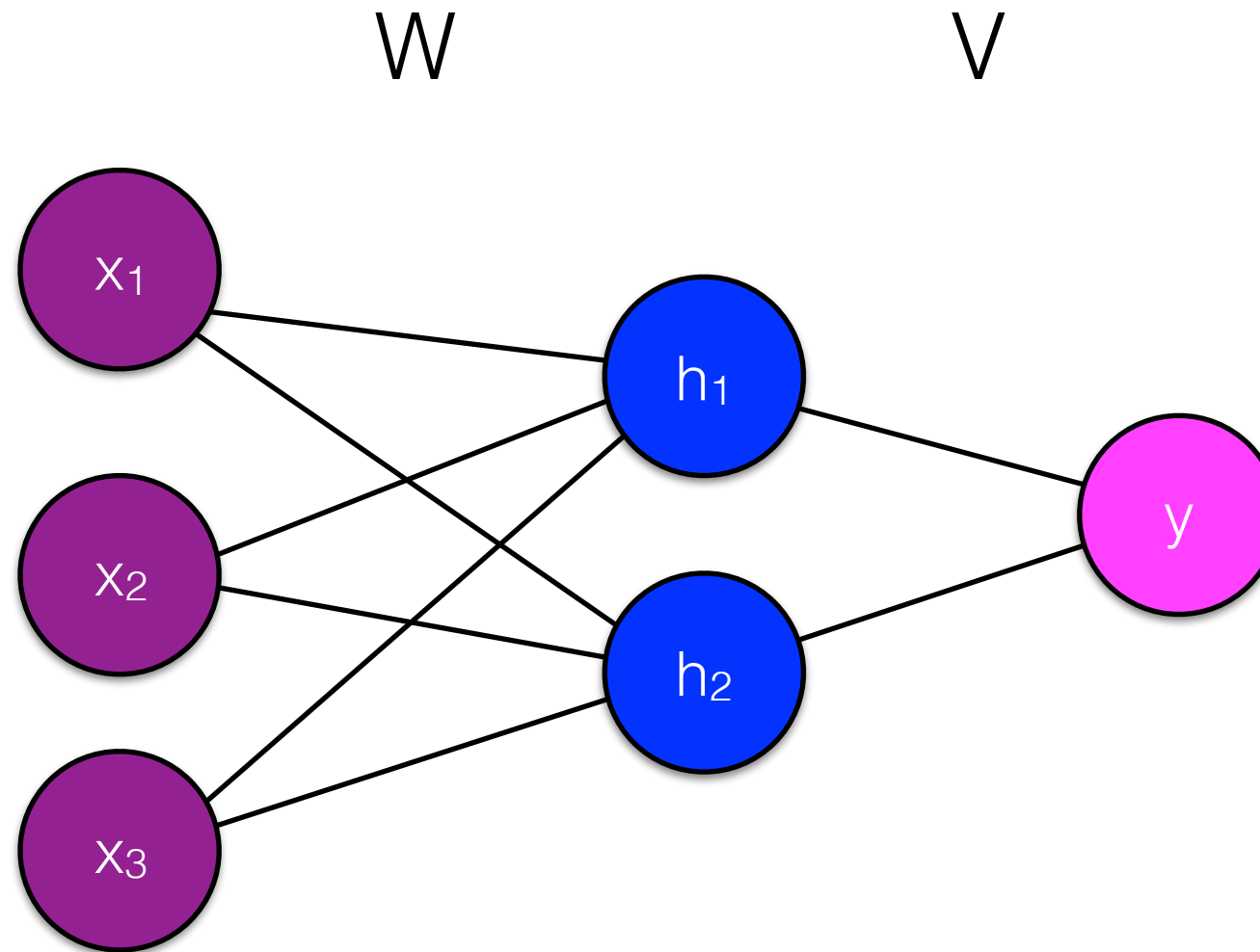
β = coefficients

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
<i>BIAS</i>	-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**.

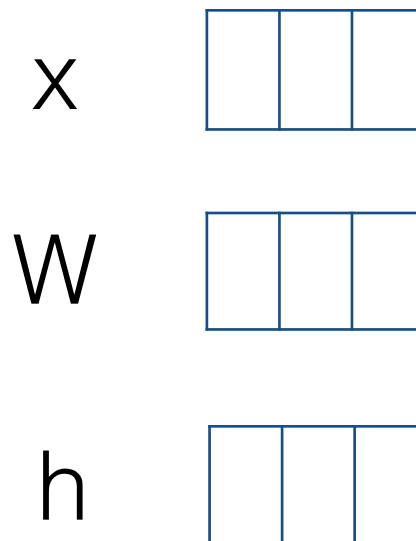
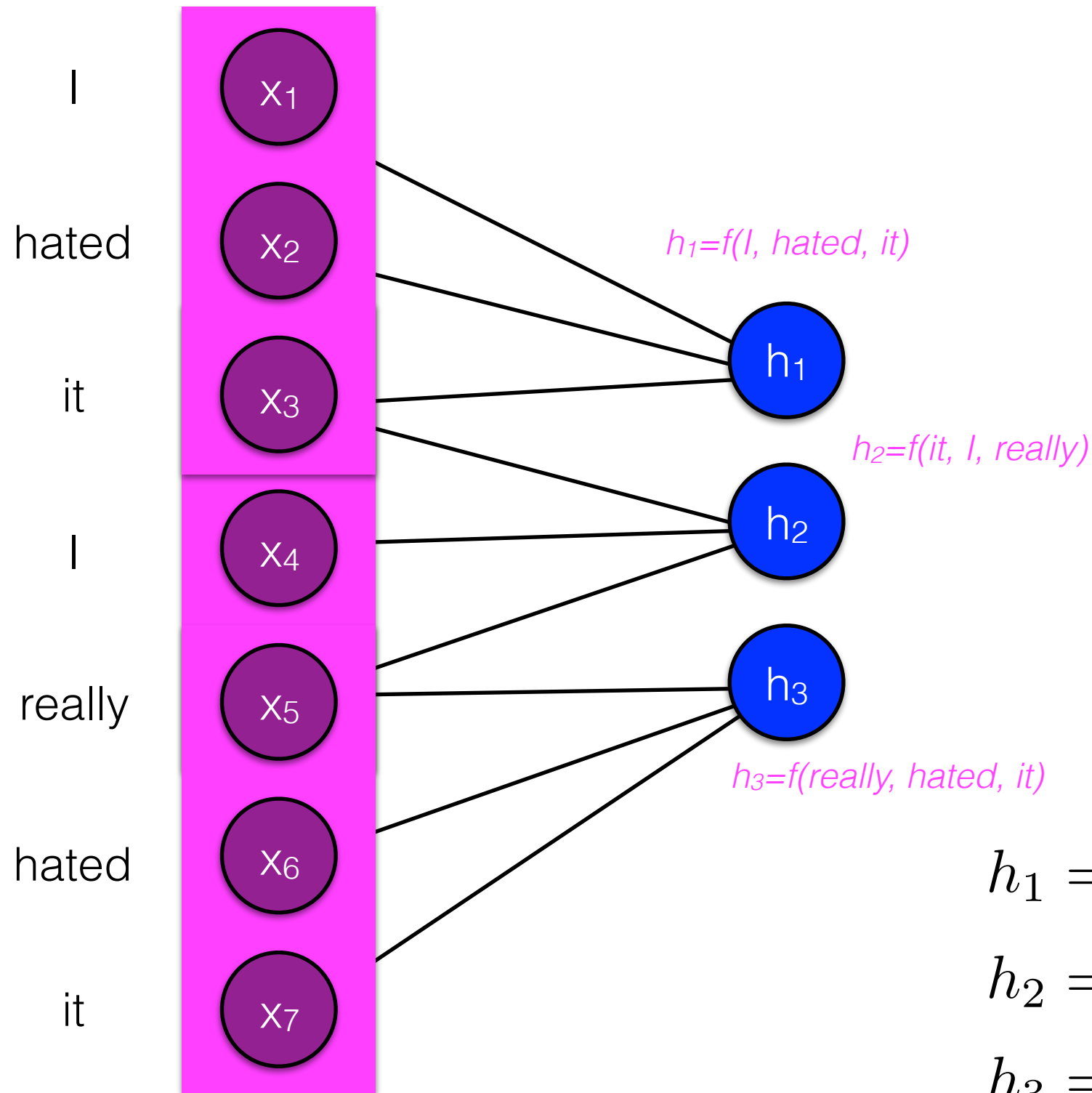
features
contains like
has word that shows up in positive sentiment dictionary
review begins with "I like"
at least 5 mentions of positive affectual verbs (like, love, etc.)



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

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

Convolutional networks

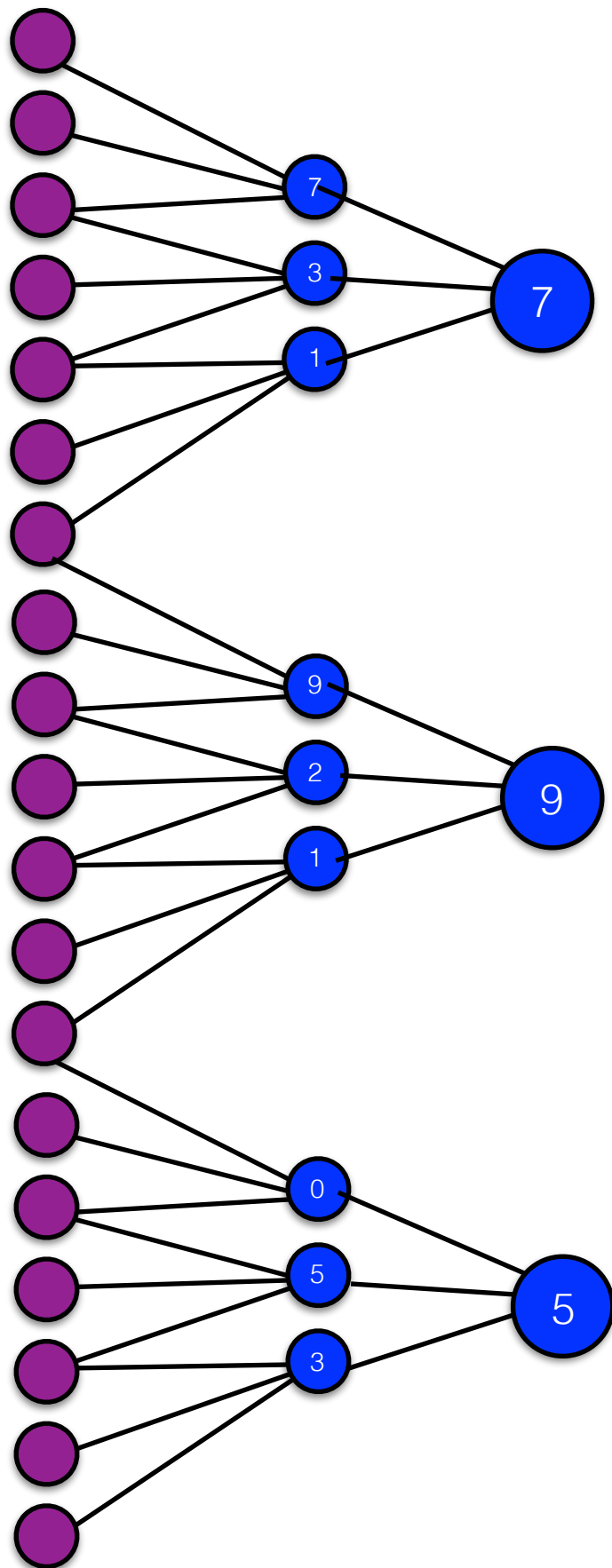


$$h_1 = \sigma(x_1 W_1 + x_2 W_2 + x_3 W_3)$$

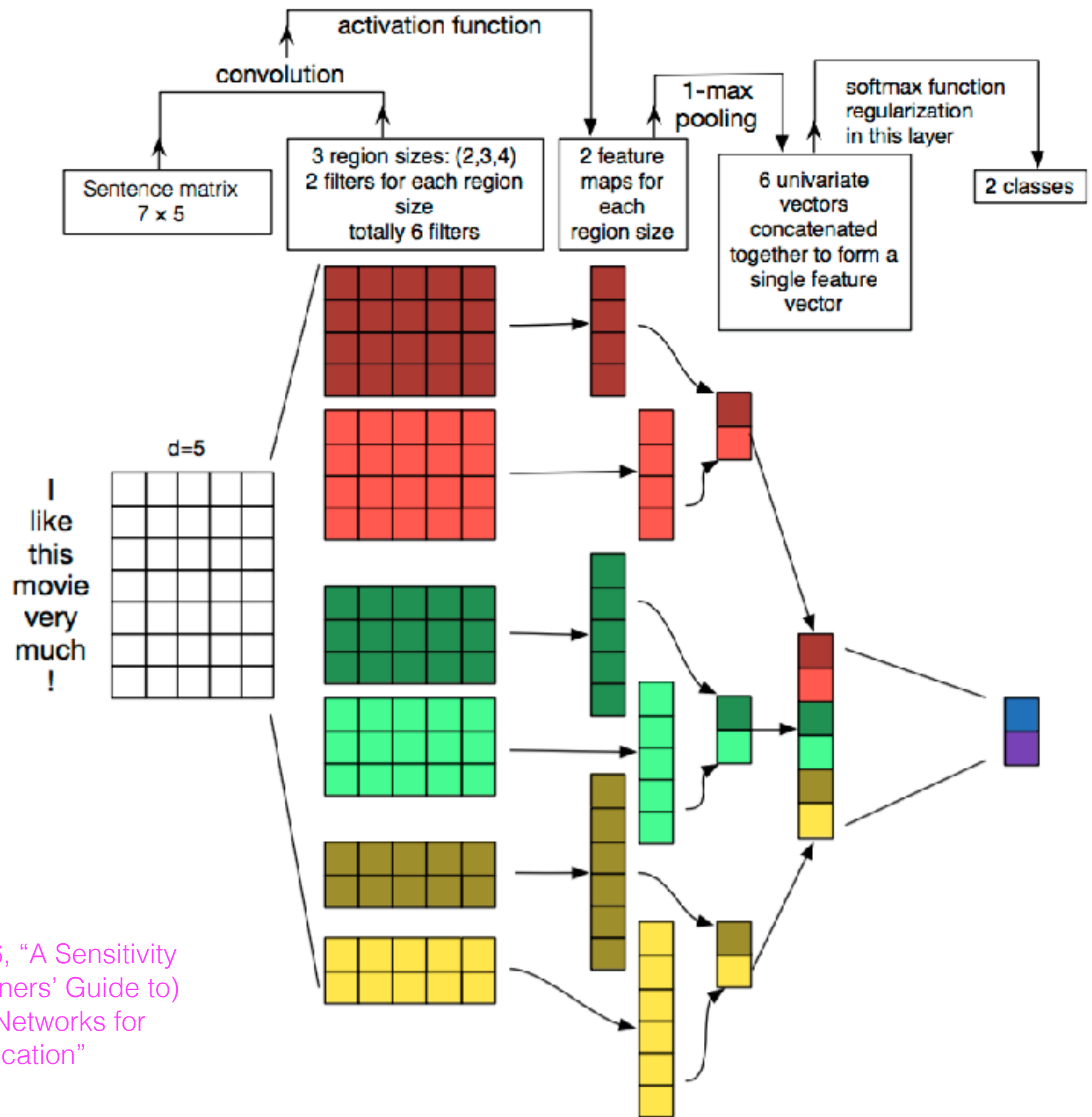
$$h_2 = \sigma(x_3 W_1 + x_4 W_2 + x_5 W_3)$$

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

Pooling



- Down-samples a layer by selecting a single point from some set
- **Max-pooling** selects the largest value



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

sequences

Sequence labeling

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

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

- 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

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

Hidden Markov Model

Prior probability of label sequence

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

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

- 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

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

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

- 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

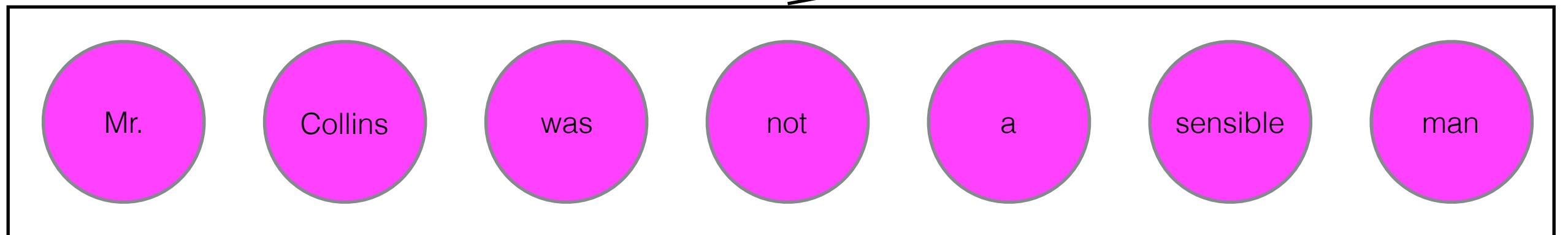
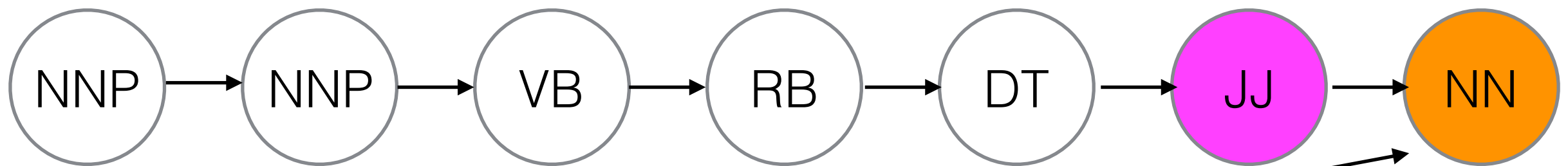
General maxent form

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

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

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

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

feature	example
$x_i = \text{man}$	1
$t_{i-1} = \text{JJ}$	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

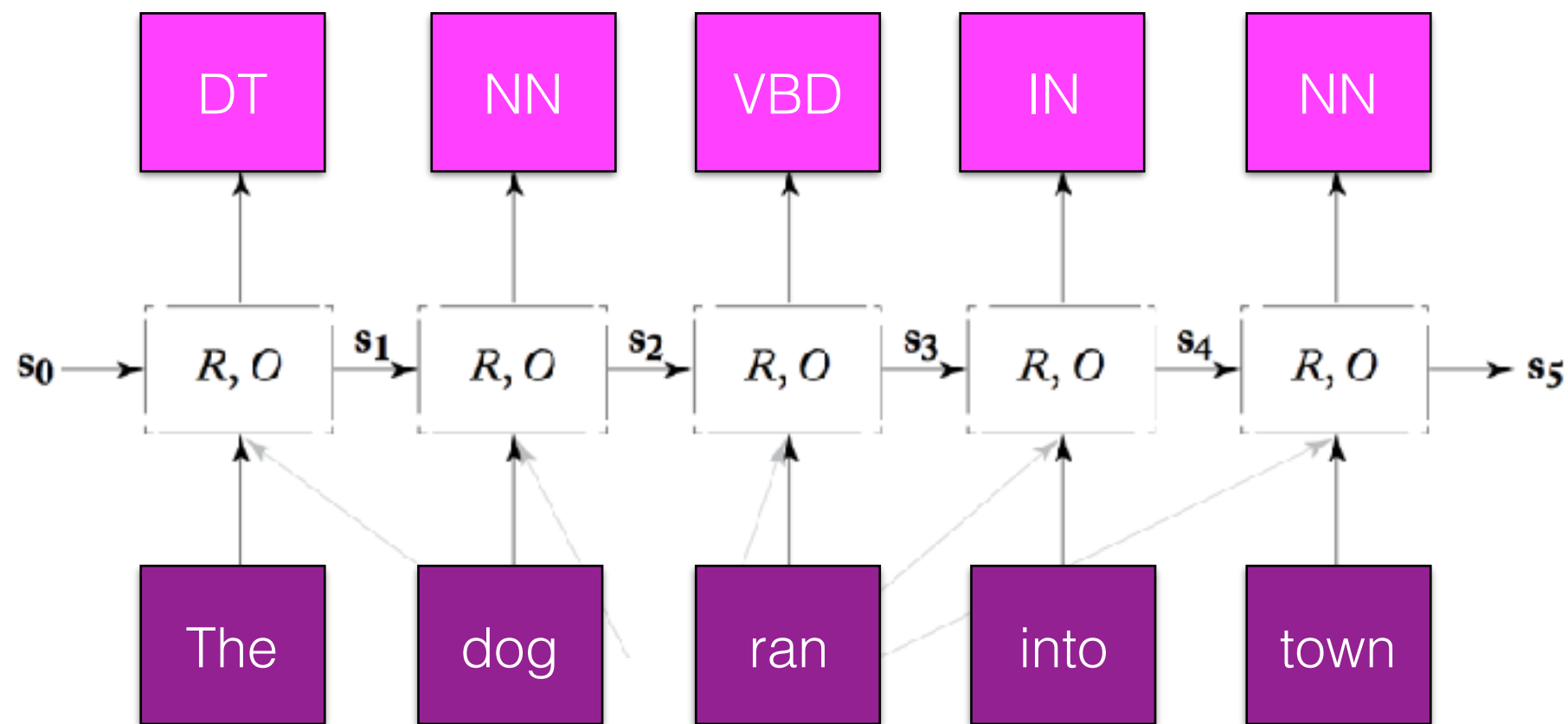
$$P(y \mid x, \beta) = \prod_{i=1}^n P(y_i \mid y_{i-1}, x, \beta)$$

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

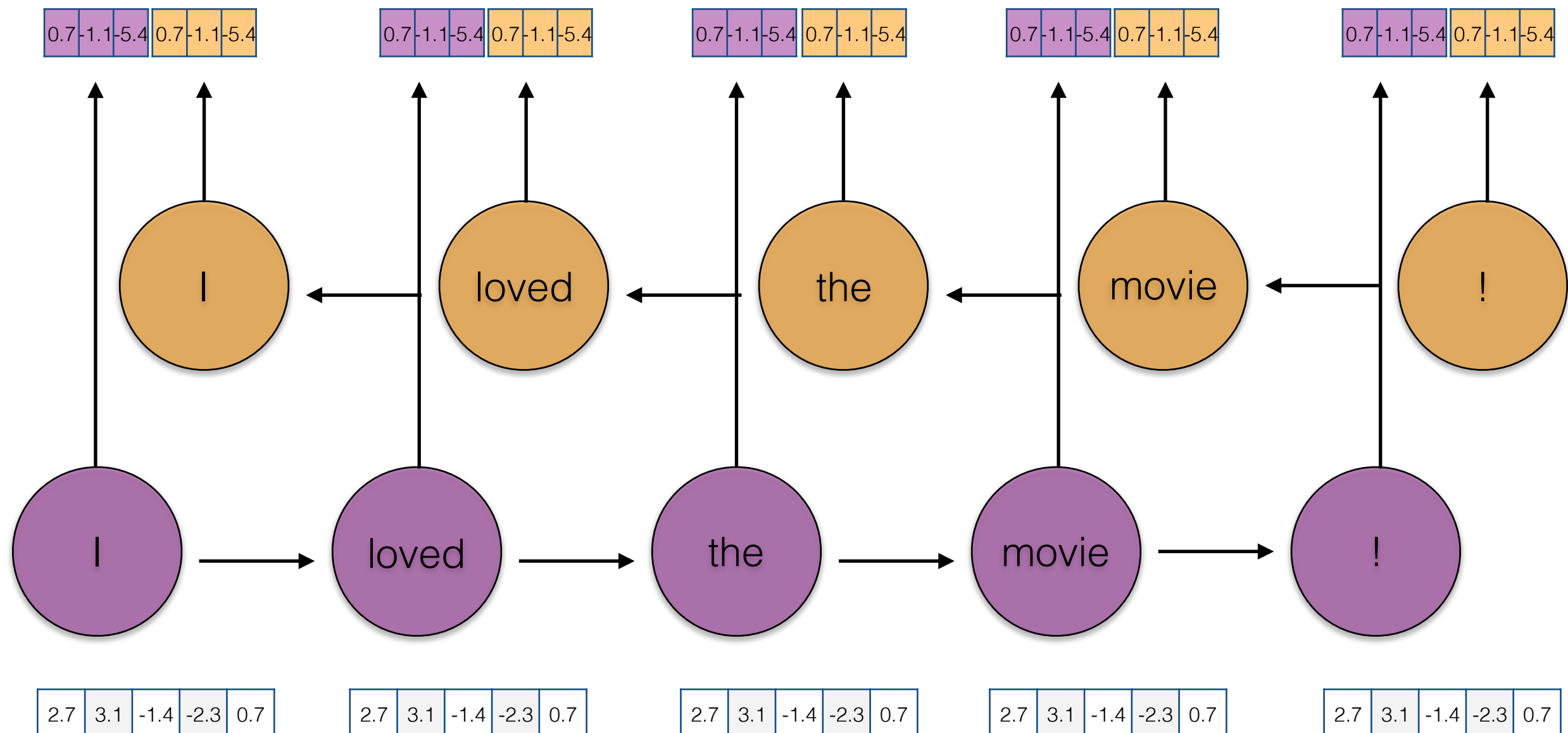
- For POS tagging, predict the **tag** conditioned on the context



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

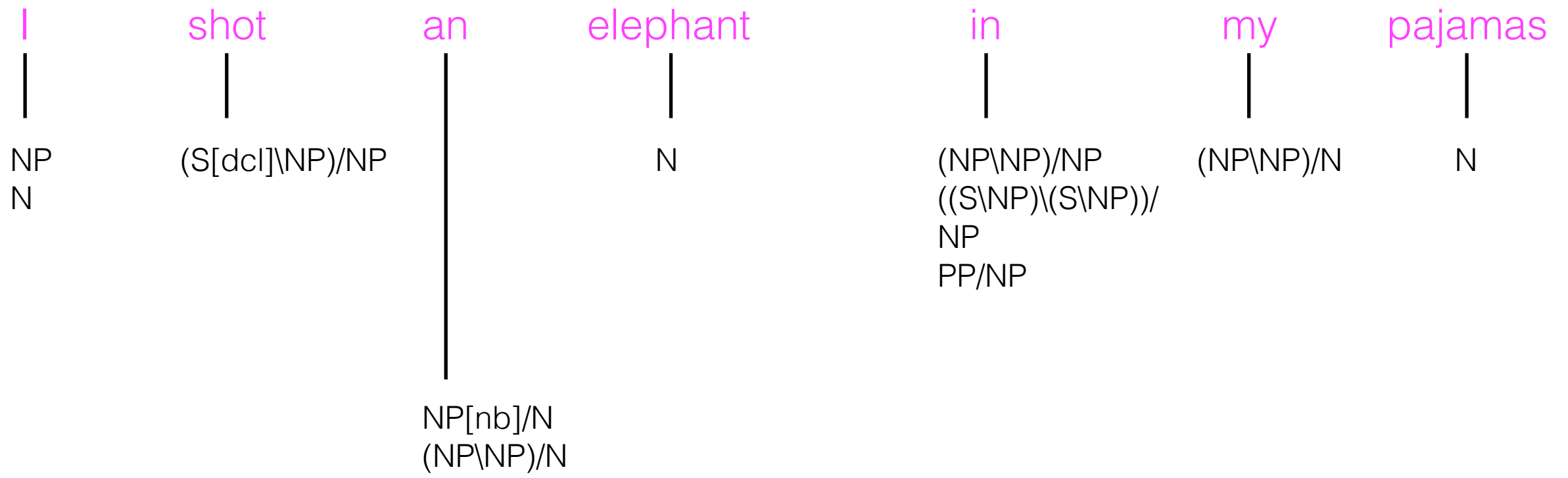
B-PER	I-PER	O	O	O	O	B-ORG
Tim	Cook	is	the	CEO	of	Apple

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

Supertagging

- The CCG lexicon + very small set of combinators dictates the overall parse for a sentence.
- The base categories in CKY for syntactic parsing are POS tags (~45 in PTB)
- The base categories in CKY for CCG parsing are the lexical rules (~1,363 in CCGBank)
- This blows up the complexity of parsing; **supertagging** reduces the lexical categories to a much smaller set by predicting the likeliest tags **in context**.

I	shot	an	elephant	in	my	pajamas
NP	N		N	(NP\NP)/NP	(NP\NP)/N	N
N	(S[dcI]\NP)/NP		N/N	((S\NP)\(S\NP))/	(S/S)\(S/S)	
N/N	S[dcI]\NP			NP	N/N	
N\N	S[pss]\NP			PP/NP	NP[nb]/N	
	(S[pss]\NP)/NP			(S/S)/NP	NP\NP	
				(S\NP)\(S\NP)		
				((S\NP)/(S\NP))/		
				NP		
				((S\NP)\(S\NP))/		
				(S[ng]\NP)		
				((S\NP)\(S\NP))\		
				((S\NP)\(S\NP)))\		
				NP		
				(NP\NP)/(S[ng]		
				\NP)		
				((S[adj]\NP)\		
				(S[adj]\NP))/NP		
				PP/(S[ng]\NP)		
				(S\S)/NP		
				(PP\PP)/NP		
				((S\NP)\(S\NP))/		
				(S[to]\NP))/N		
				S[adj]\NP		
				((NP\NP)\		
				(NP\NP))/NP		
				(NP\NP)/(S[adj]		
				\NP)		
				...		



Supervised WSD

- We have labeled training data; let's **learn** from it.
- Decision trees (Yarowsky 1994)
- Naive Bayes, log-linear classifiers, support vector machines (Zhong and Ng 2010)
- Bidirectional LSTM (Raganato et al. 2017)

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 $A \rightarrow \beta$):

$$P(T, S) = \prod_i^n P(\beta \mid A)$$

Estimating PCFGs

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

(equivalently)

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

A		β	$P(\beta \mid \text{NP})$
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

Output space $\mathbf{y} =$

- 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?

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

Use any multiclass classification model

- Logistic regression
- SVM
- NB
- Neural network

feature	example	β
stack ₁ = me	1	0.7
stack ₂ = book	1	1.3
stack ₁ POS = PRP	1	6.4
buffer ₁ = the	1	-1.3
buffer ₂ = morning	1	-0.07
buffer ₁ = today	0	0.52
buffer ₁ POS = RB	0	-2.1
stack ₁ = me AND stack ₂ =	1	0
stack ₁ = PRP AND stack ₂ =	1	-0.1
iobj(book,*) in arcs	0	3.2

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)

Learning from logical forms

- We can train a semantic parser in a number of ways:
 - Full derivational trees (CCGBank)
 - Logical forms (Zettlemoyer and Collins 2005)
 - Denotations (Berant et al. 2013)

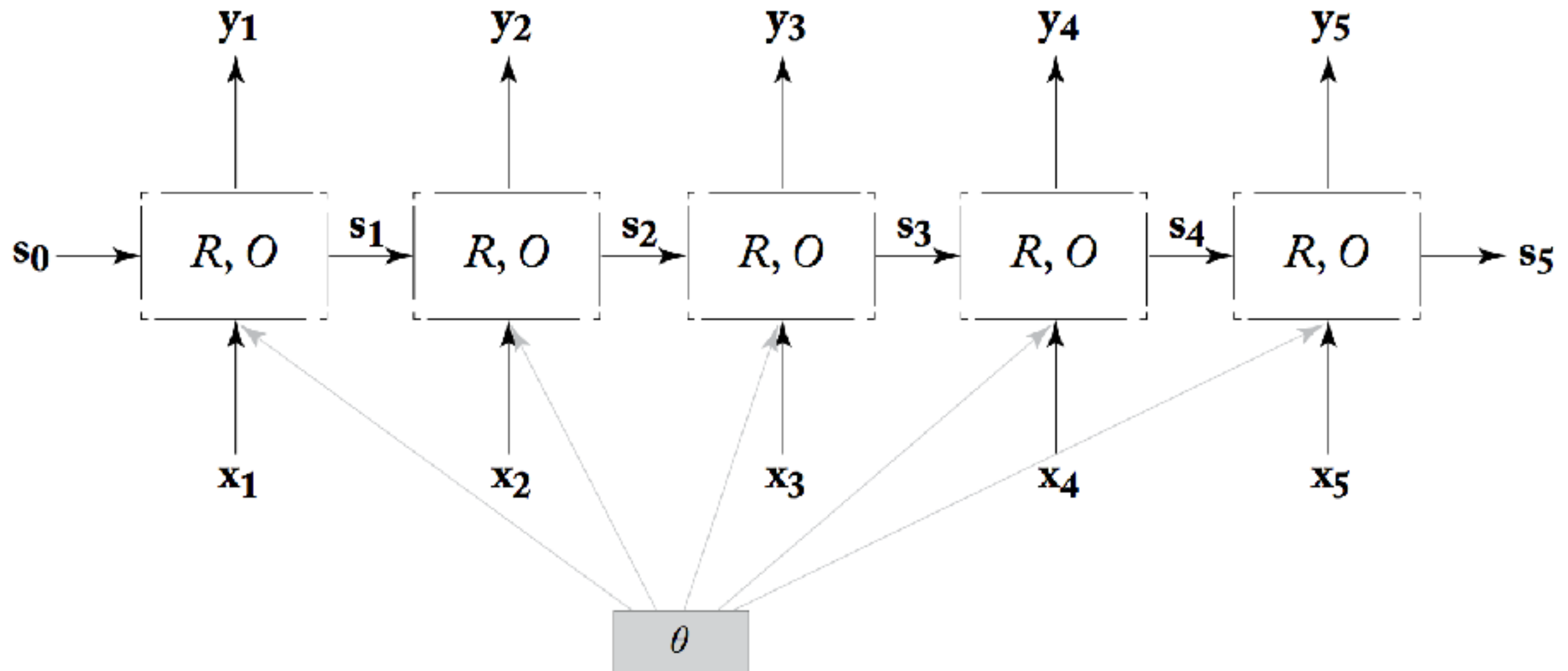
natural language generation

Language Model

Language modeling is the task of estimating $P(w)$

- Count and normalize
- Featurized
- Neural (RNN)

Recurrent neural network

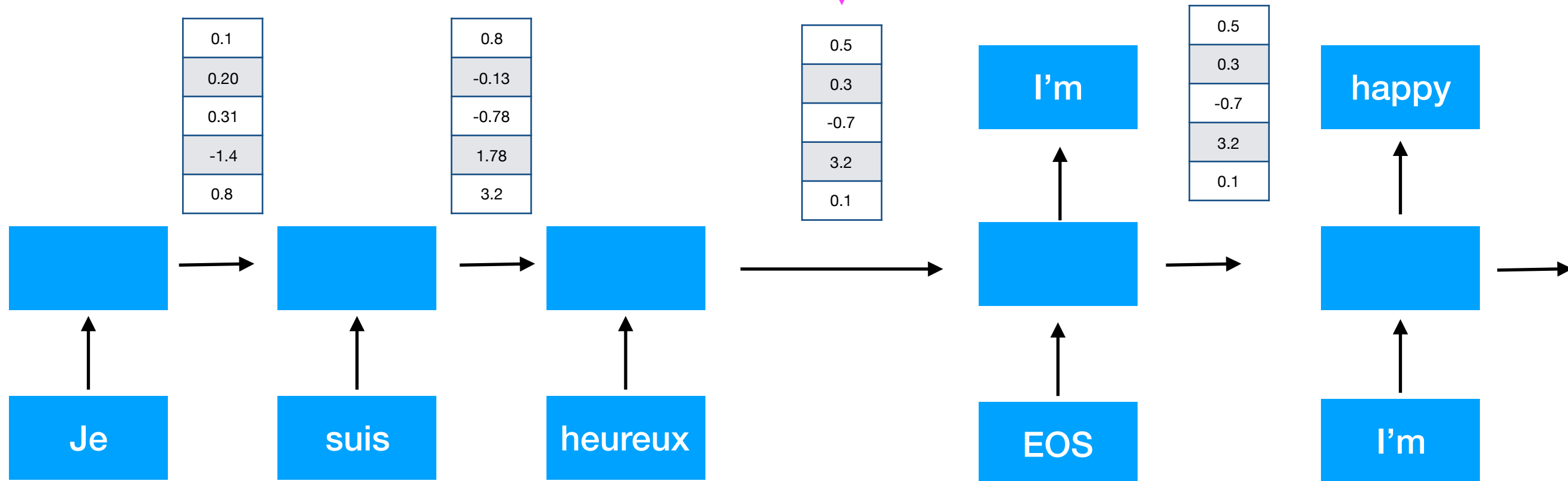


Encoder-decoder framework

- 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



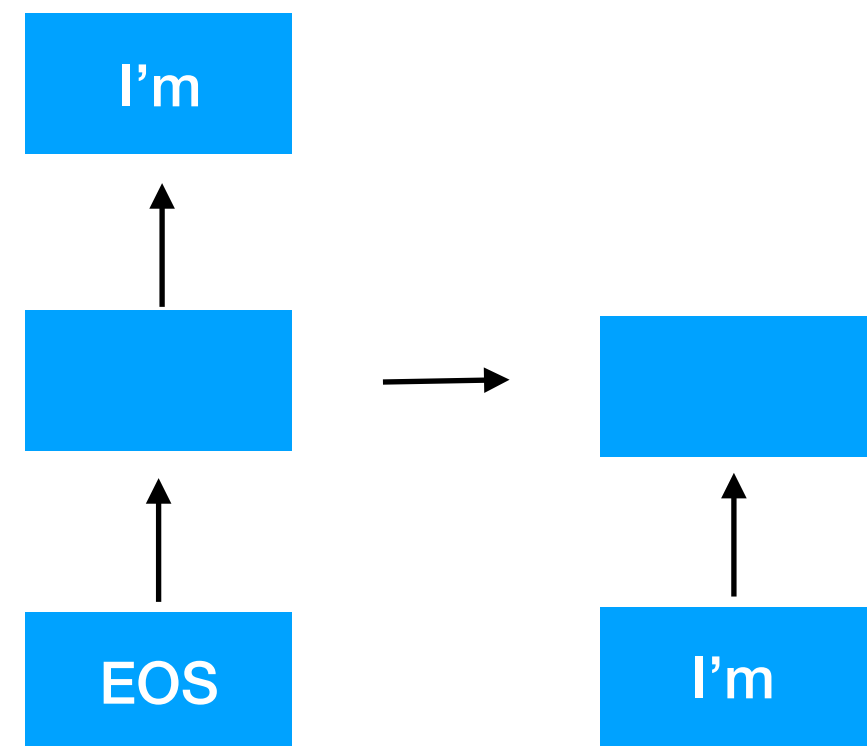
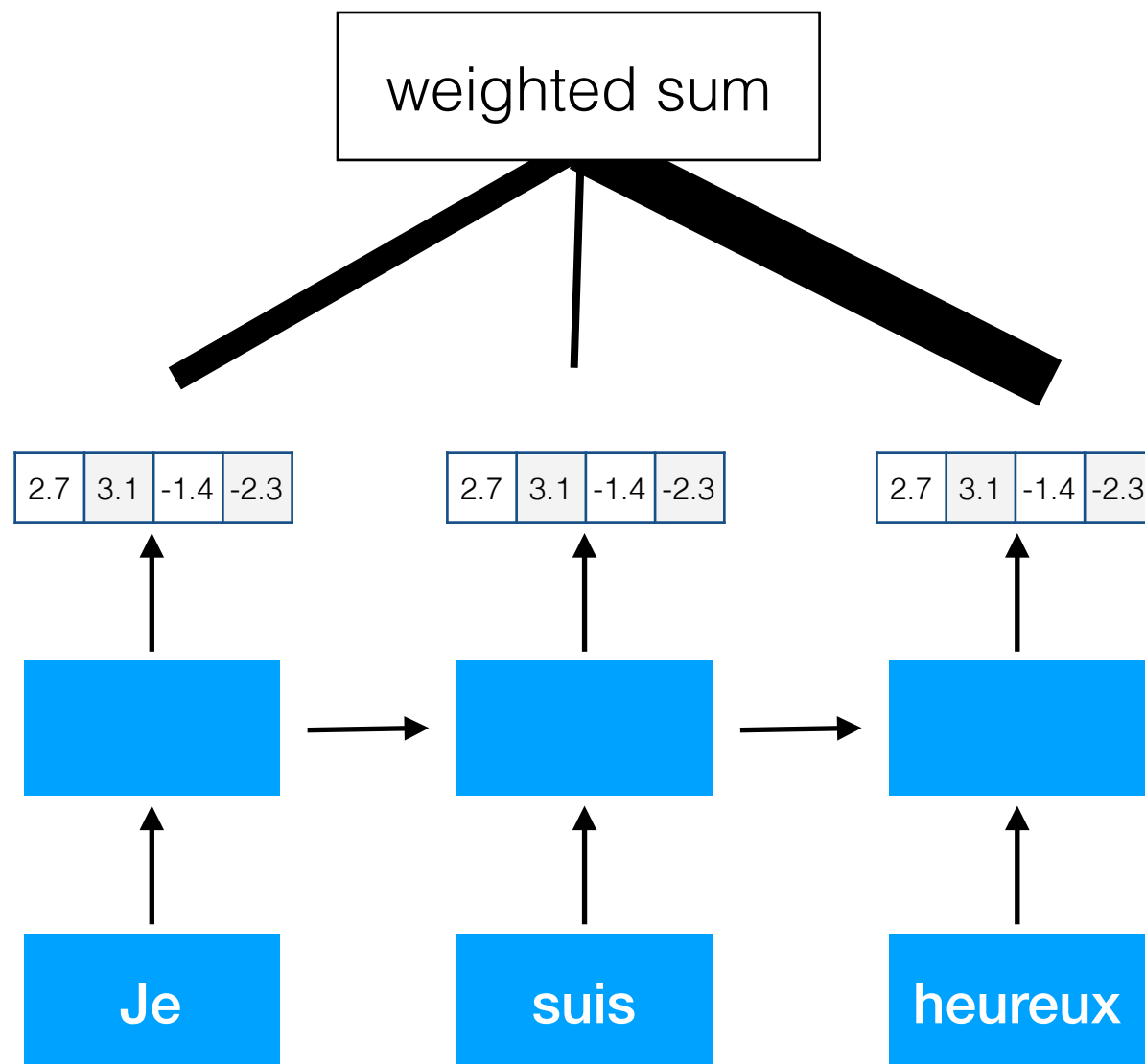
The decoder state depends just on the previous *state* and the previous *output*

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

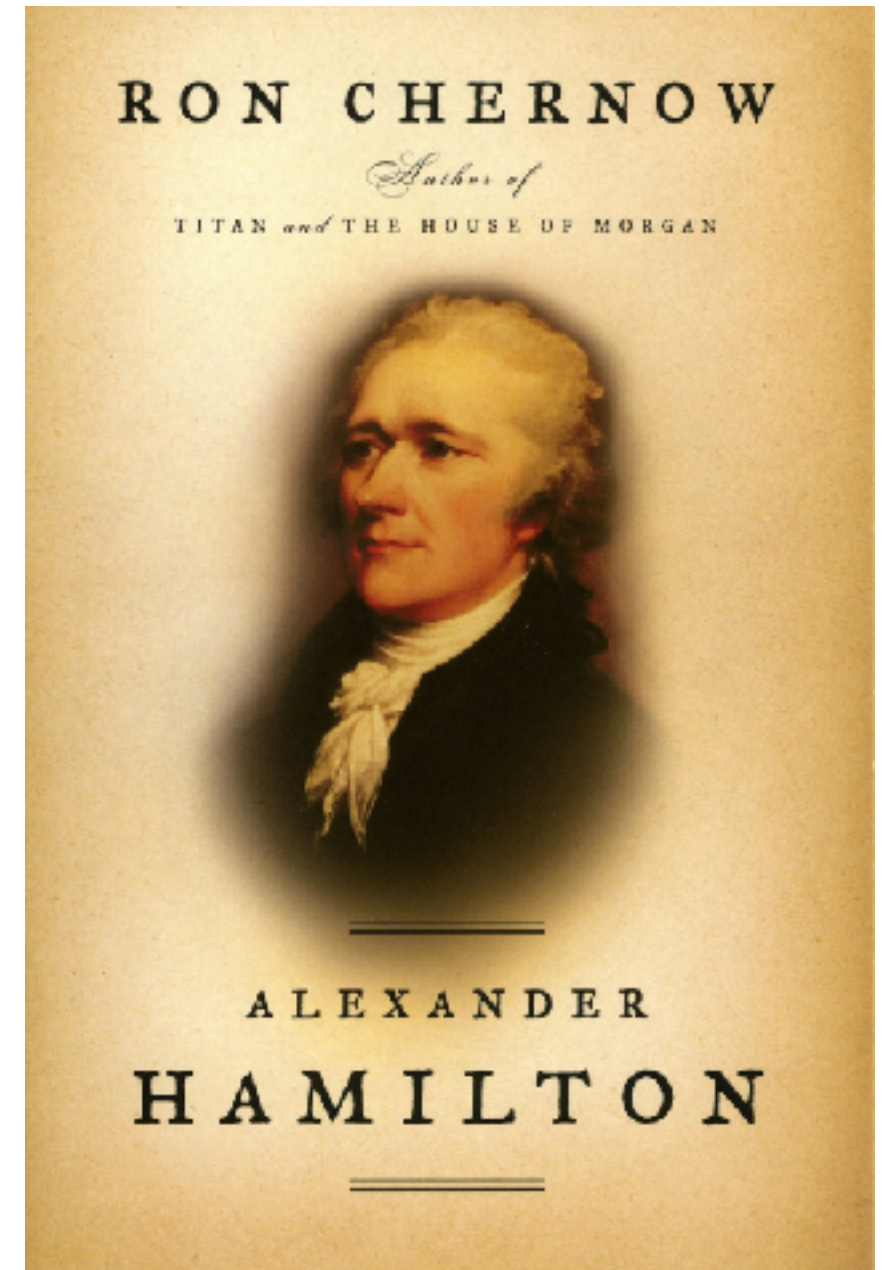
Encoder-decoder with attention

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

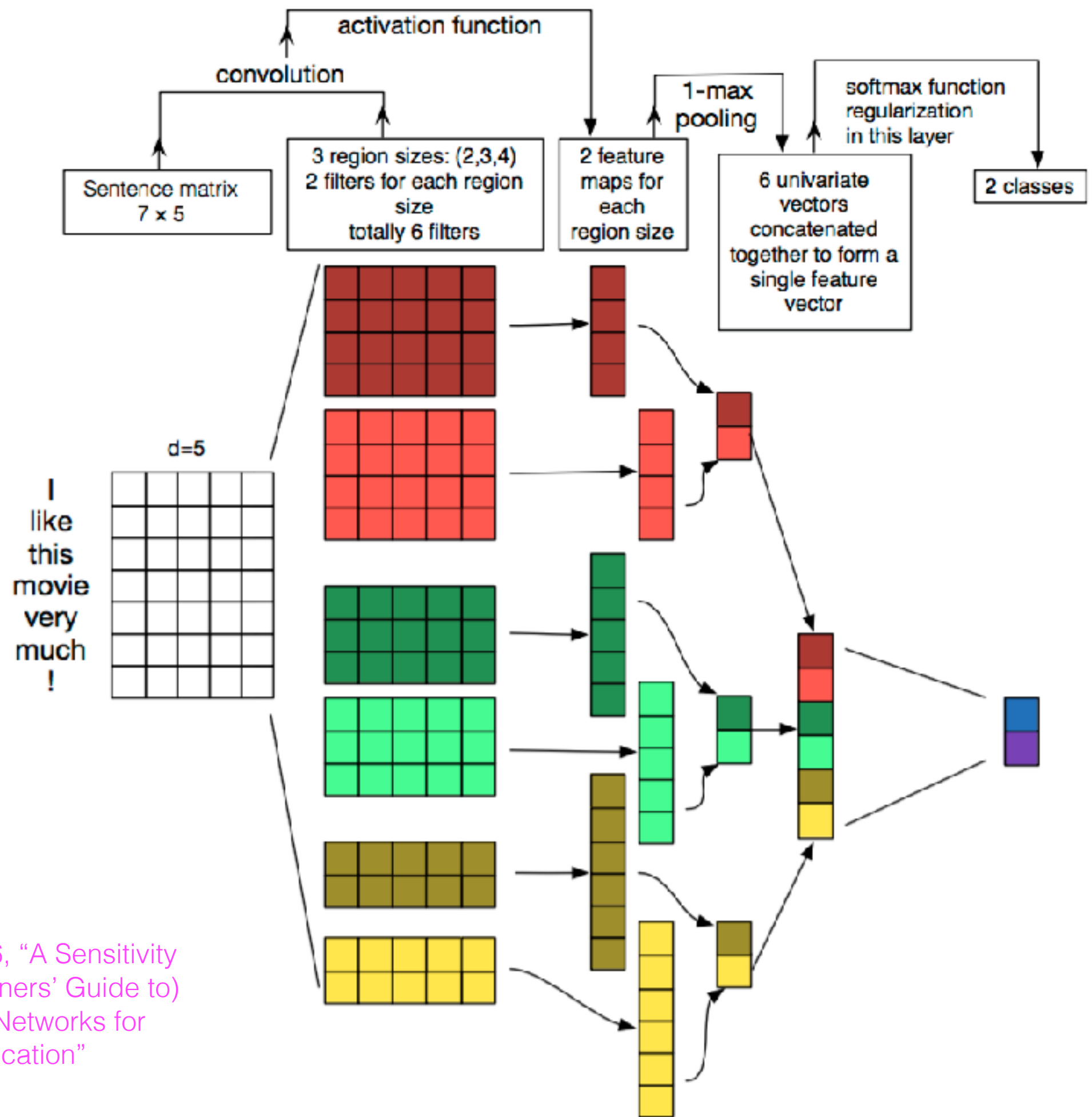
$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$



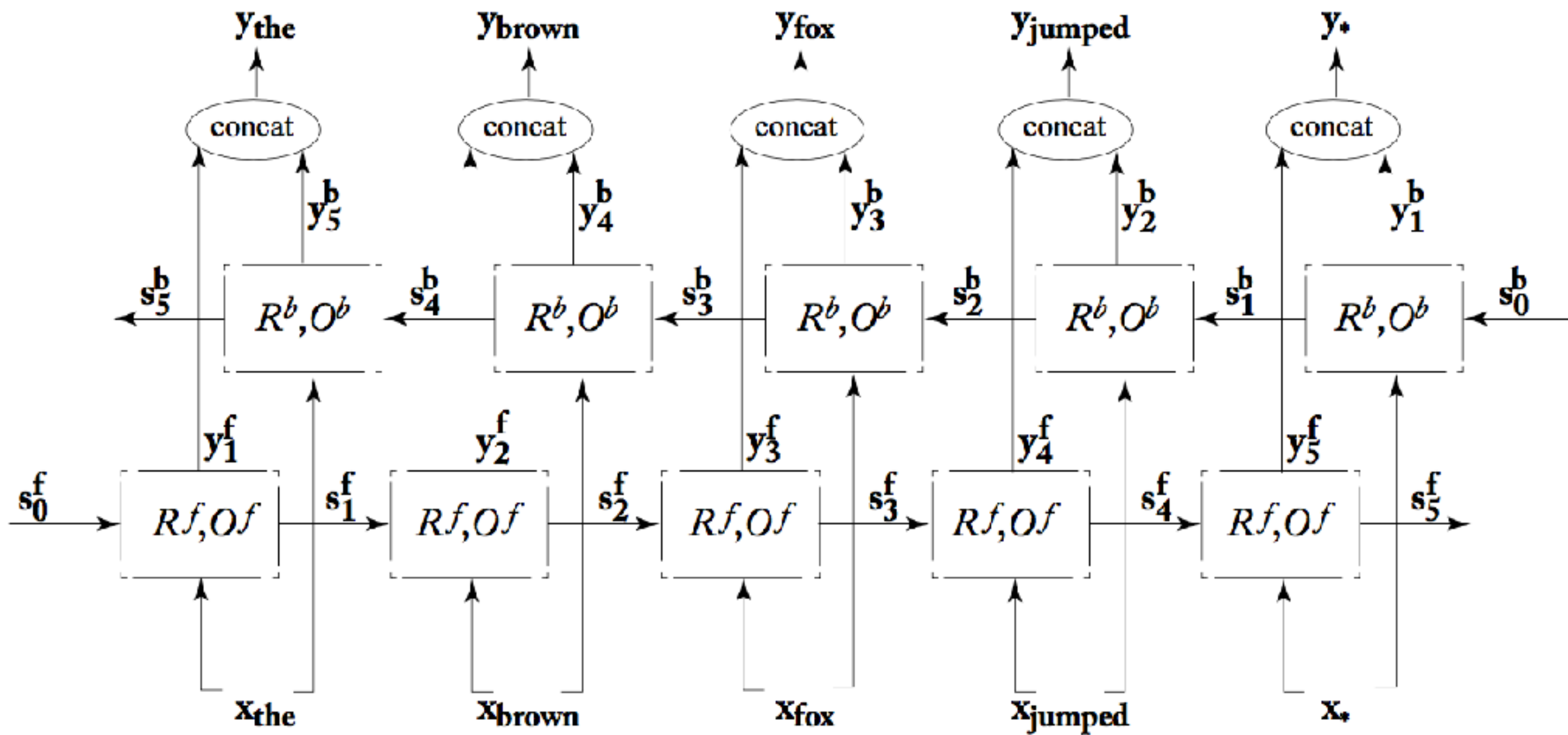
Summarization



3. Neural methods are generally* better



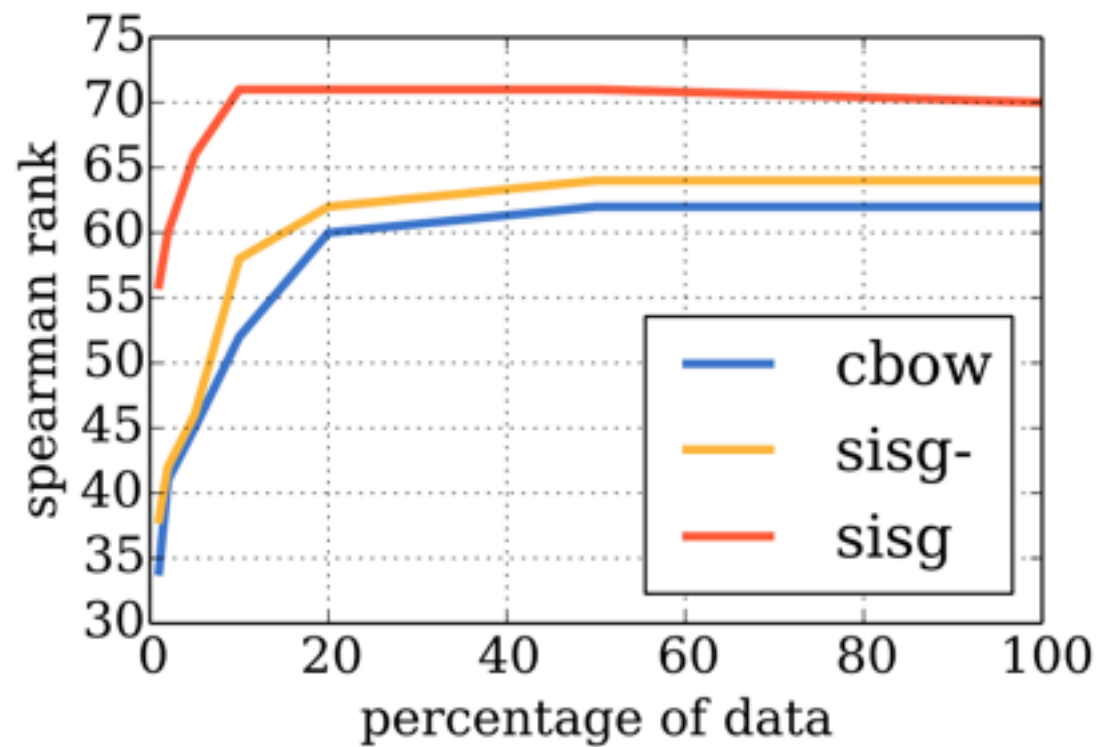
Zhang and Wallace 2016, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification"



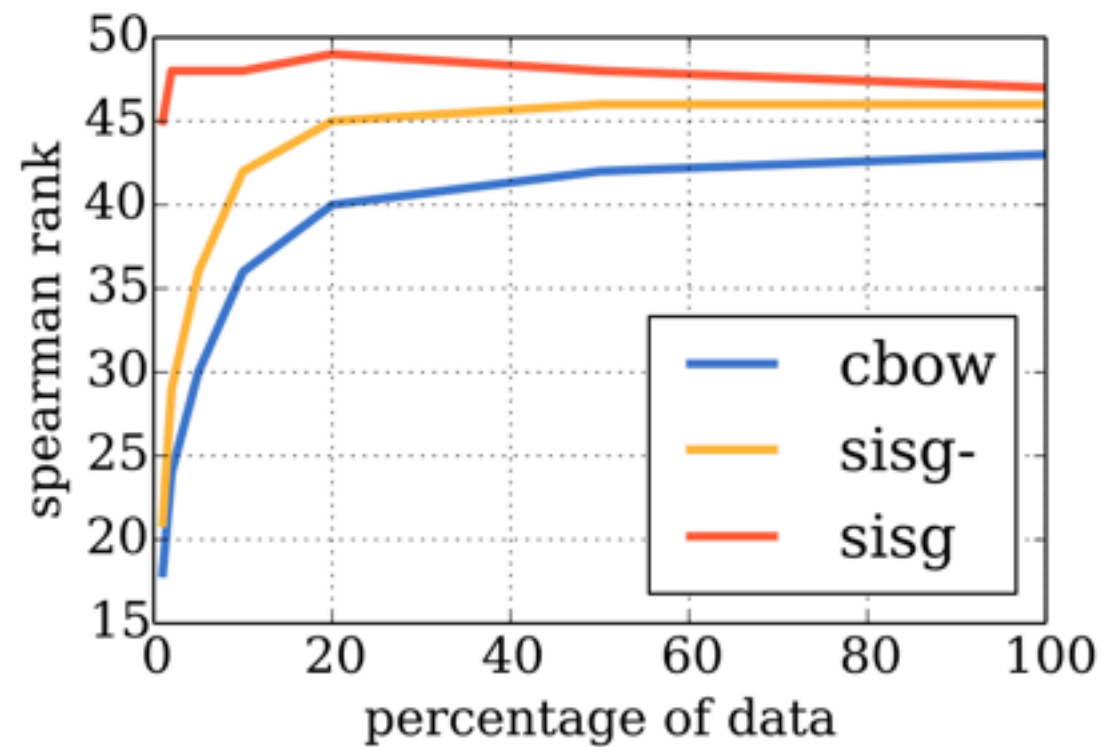
Subword models

- 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



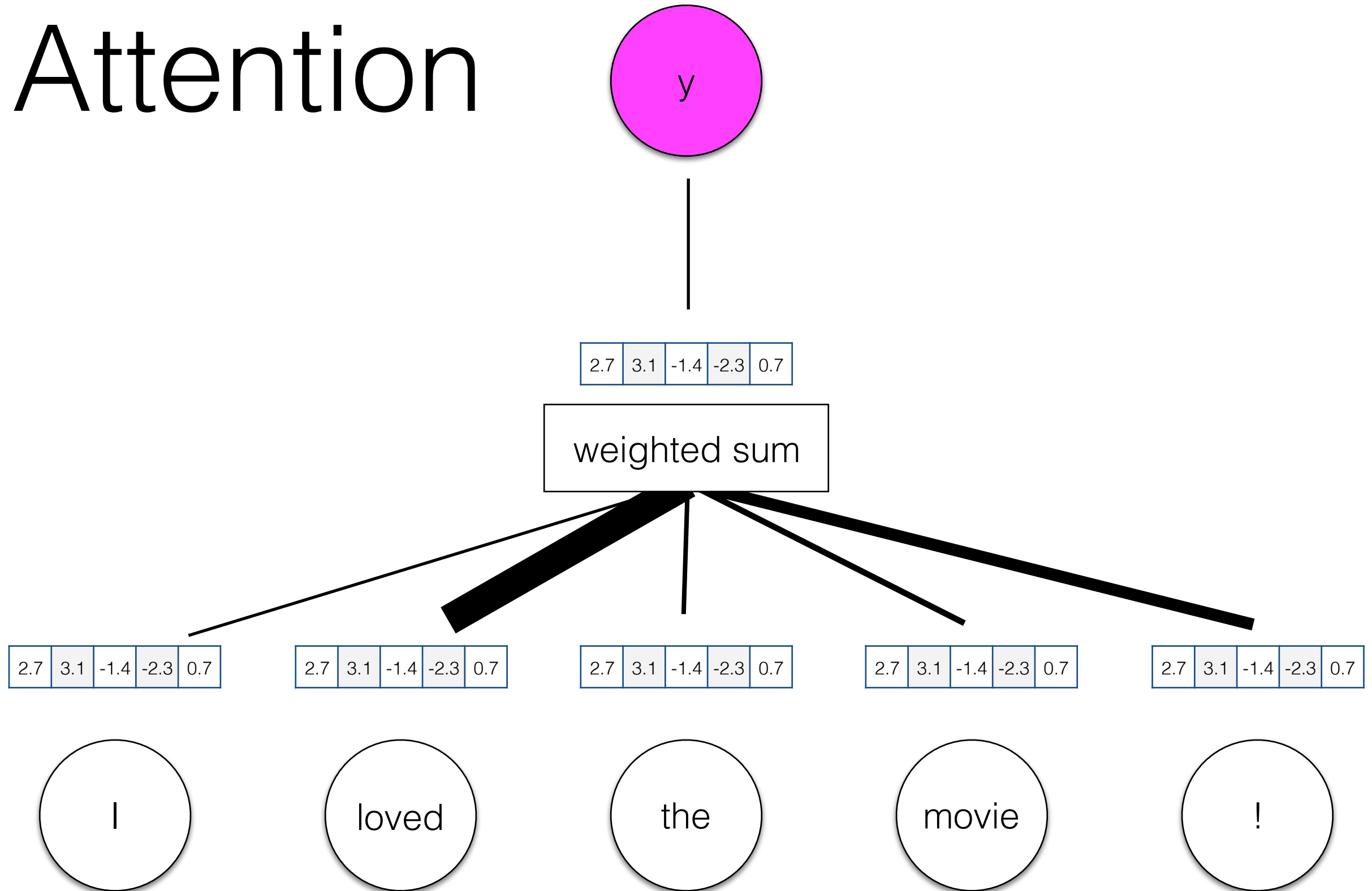
(b) EN-RW

1% =
~20M tokens

100% =
~1B tokens

- Subword models need less data to get comparable performance.

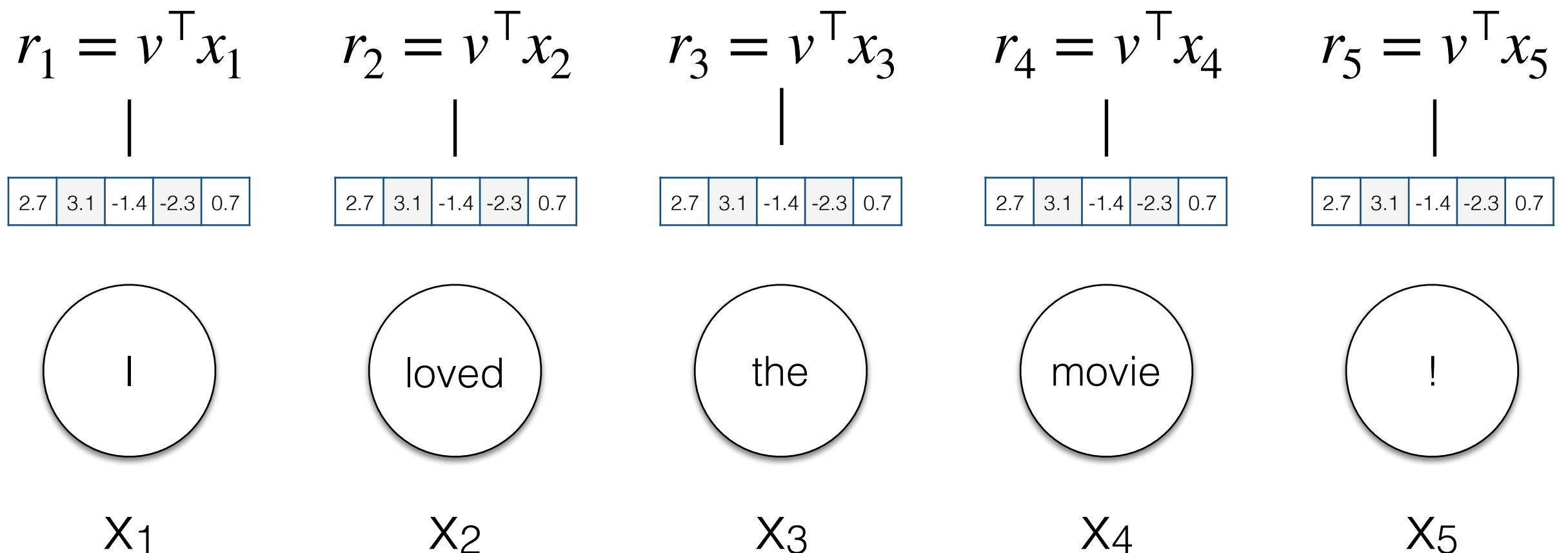
Attention

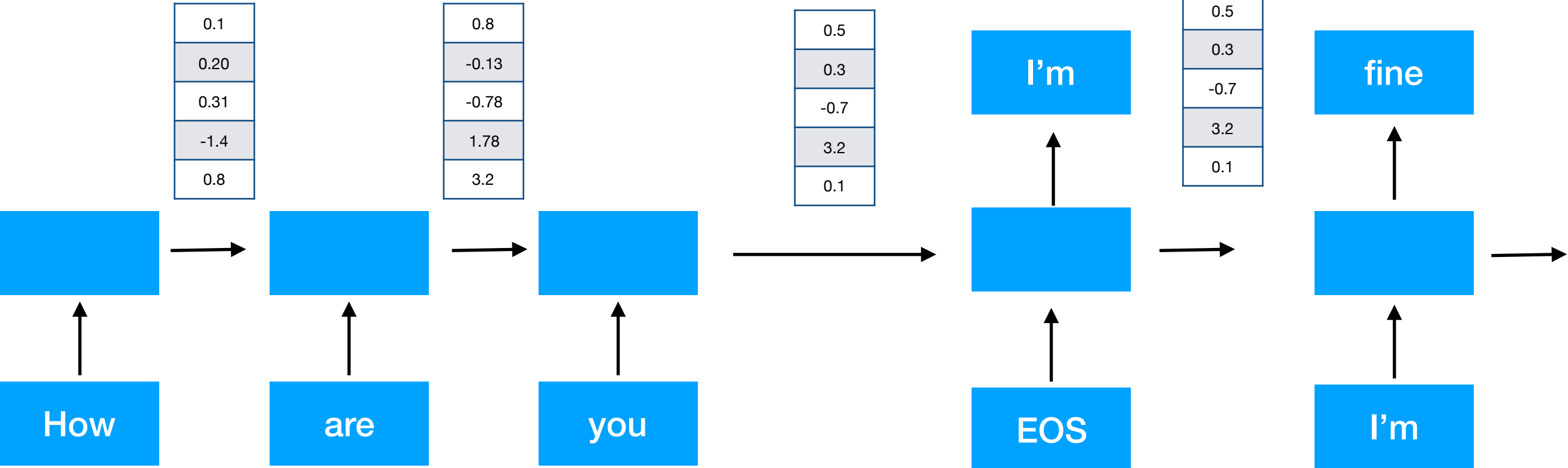


$$v \in \mathcal{R}^H$$

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





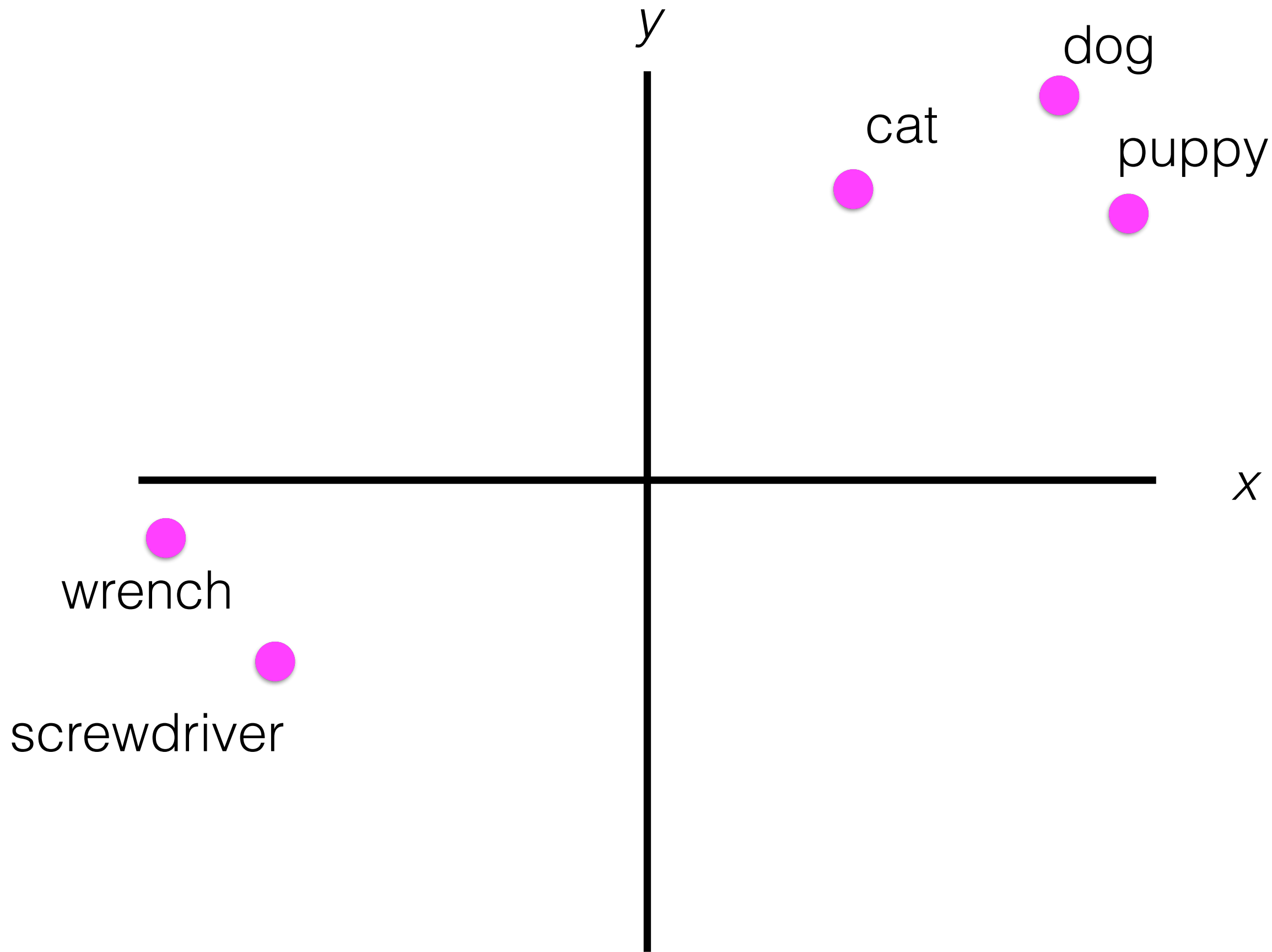
Lexical semantics

“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**).



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.

4. Evaluation is critical

Interannotator agreement



annotator A

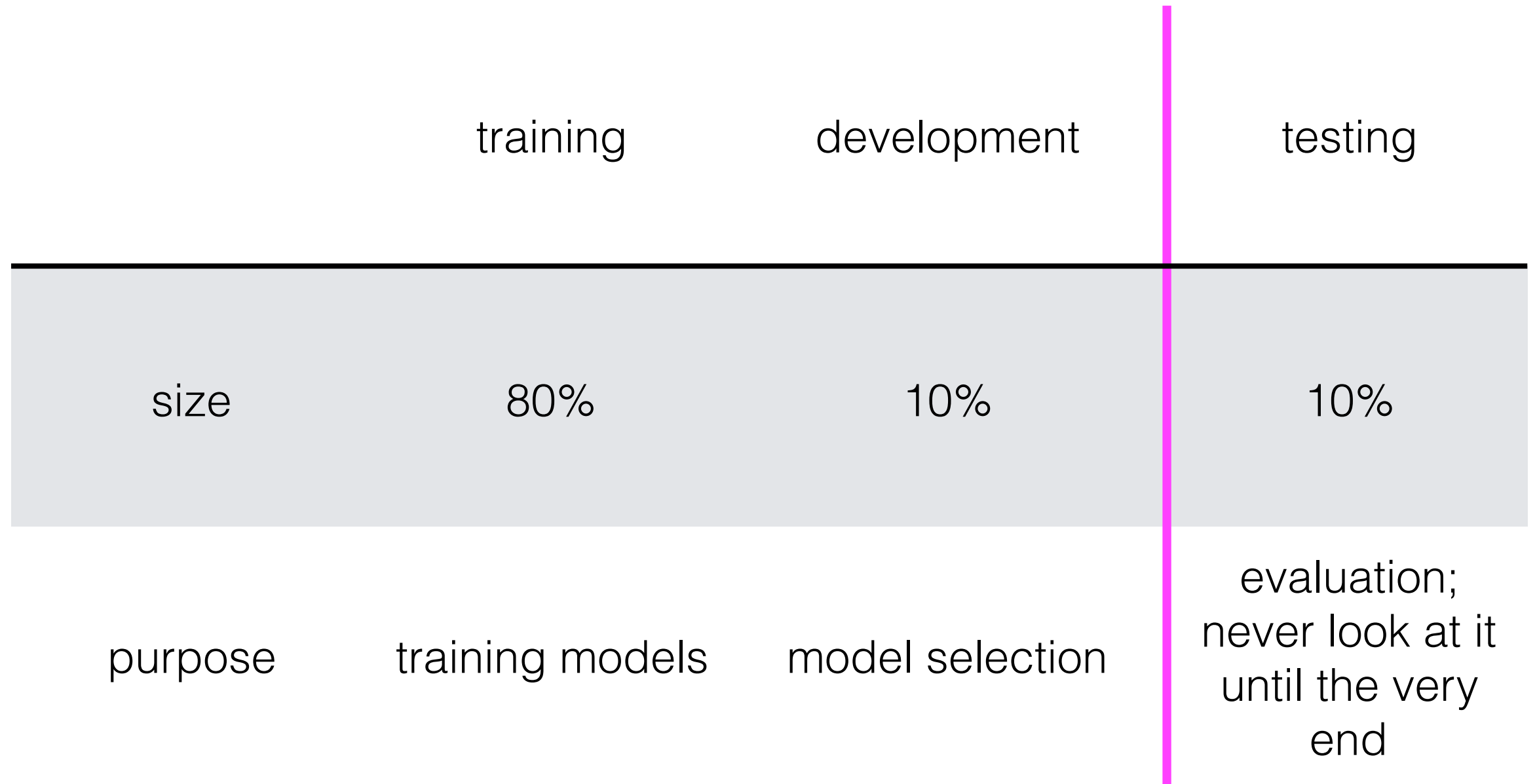
puppy fried
chicken

annotator B

	puppy	fried chicken
puppy	6	3
fried chicken	2	5

observed agreement = $11/16 = 68.75\%$

Experiment design



Metrics

- Perplexity
- Accuracy
- Precision/recall/F1
- Parseval (P/R/F1 over labeled constituents)
- Correlation with human judgments
- B³
- BLEU

Stance detection

- What is the stance of some text with respect to some target?

Target: legalization of abortion (1)

Tweet: *The pregnant are more than walking incubators, and have rights!*

Stance detection

- What is the stance of some text with respect to some target?

Topic	Post and Threading Variables					Normalized LIWC Variables				
	Posts	Rebuttals	P/A	A > 1p	PL	Pro	WPS	6LTR	PosE	NegE
Cats v. Dogs	148	40%	1.68	26%	242	3.30	-1.95	-2.43	1.70	.30
Firefox vs. IE	218	40%	1.28	16%	167	-0.11	-0.84	0.53	1.23	-0.81
Mac vs. PC	126	47%	1.85	24%	347	0.52	0.28	-0.85	-0.11	-1.05
Superman/Batman	140	34%	1.41	21%	302	-0.57	-1.78	-0.43	1.21	.99
2nd Amendment	134	59%	2.09	45%	385	-1.38	1.74	0.58	-1.04	0.38
Abortion	594	70%	2.82	43%	339	0.63	-0.27	-0.41	-0.95	0.68
Climate Change	202	69%	2.97	40%	353	-0.74	1.23	0.57	-1.25	-0.63
Communism vs. Capitalism	212	70%	3.03	47%	348	-0.76	-0.15	1.09	0.39	-0.55
Death Penalty	324	62%	2.44	45%	389	-0.15	-0.40	0.49	-1.13	2.90
Evolution	798	76%	3.91	55%	430	-0.80	-1.03	1.34	-0.57	-0.94
Exist God	844	77%	4.24	52%	336	0.43	-0.10	0.34	-0.24	-0.32
Gay Marriage	505	65%	2.12	29%	401	-0.13	.86	.85	-0.42	-0.01
Healthcare	110	80%	3.24	56%	280	0.28	1.54	.99	0.14	-0.42
Marijuana Legalization	214	52%	1.55	26%	423	0.14	0.37	0.53	-0.86	0.50

5. Text is data.





ΙΜΙΑΛΛΟΣ

ΑΛΦΑ

ἡμιν ἄλλοις ἀνταρθεῖν

ἡμιν ἄλλοις ἀνταρθεῖν

οὐ μόνον ἡμιν. ἡμιν ἄλλοις ἀνταρθεῖν

οὐ μόνον ἡμιν. ἡμιν ἄλλοις ἀνταρθεῖν

πολλὰς δὲ φθιμώνας τυχεῖν

πολλὰς δὲ φθιμώνας τυχεῖν

ἡμιν ἄλλοις ἀνταρθεῖν

ἡμιν ἄλλοις ἀνταρθεῖν

οἱ ἡμῖν σὺν τοῖς ἀνταρθεῖν

οἱ ἡμῖν σὺν τοῖς ἀνταρθεῖν

οἱ ἡμῖν σὺν τοῖς ἀνταρθεῖν



Donald J. Trump  @realDonaldTrump · Aug 25



I am pleased to inform you that I have just granted a full Pardon to 85 year old American patriot Sheriff Joe Arpaio. He kept Arizona safe!



80K



34K



141K



Donald J. Trump  @realDonaldTrump · 18 Jun 2016



How can the NY **Times** show an empty room hours before my speech even started when they knew it was going to be packed? So totally dishonest!



3.4K



7.4K



25K







Thanks!

Course evaluations!

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