



SI 630

Natural Language Processing: Algorithms and People

Lecture 5: Sequence Labeling
Feb. 5, 2019

David Jurgens
jurgens@umich.edu

Today's Overview

Today's Overview

- Part of Speech Tagging

Today's Overview

- Part of Speech Tagging
- Mini-lab on the Viterbi Algorithm

Today's Overview

- Part of Speech Tagging
- Mini-lab on the Viterbi Algorithm
- More Neural Networks

Meta-Discussion of 630



Meta-Discussion of 630



I'm the NLP
Class Train! Choo
Choo!

Meta-Discussion of 630

Fundamentals +
Machine Learning



I'm the NLP
Class Train! Choo
Choo!

Meta-Discussion of 630

Fundamentals +
Machine Learning



I'm the NLP
Class Train! Choo
Choo!



Meta-Discussion of 630

Fundamentals + Structure in
Machine Learning Language



I'm the NLP
Class Train! Choo
Choo!

Meta-Discussion of 630

Fundamentals + Structure in Semantics + NLP
Machine Learning Language Applications

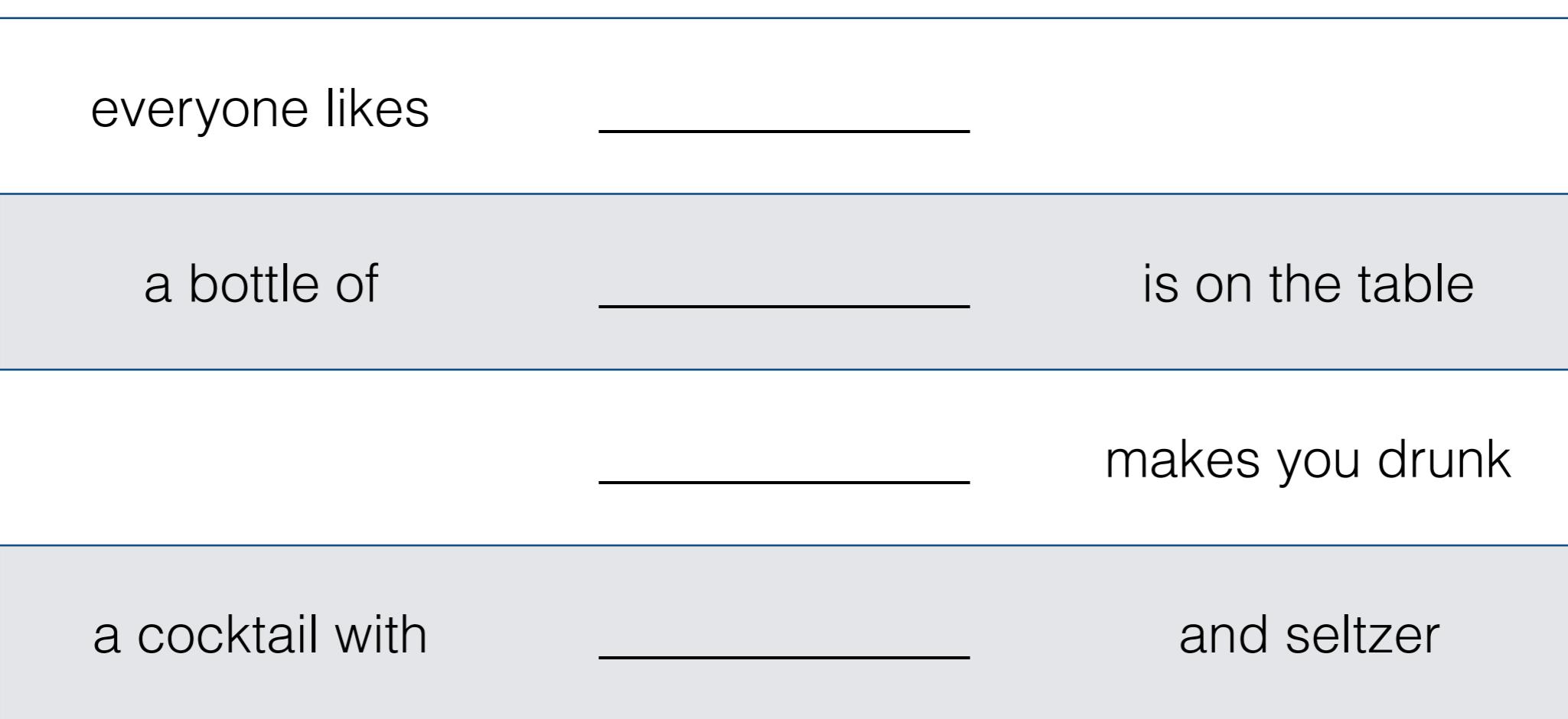


I made you a cookie...



but I eated it.

context



from last time

Distribution

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

from last time

Parts of speech

- Parts of speech are categories of words defined **distributionally** by the morphological and syntactic contexts a word appears in.

Morphological distribution

POS often defined by distributional properties; verbs
= the class of words that each combine with the
same set of affixes

	-s	-ed	-ing
walk	walks	walked	walking
slice	slices	sliced	slicing
believe	believes	believed	believing
of	*ofs	*ofed	*ofing
red	*reds	*redded	*reding

Morphological distribution

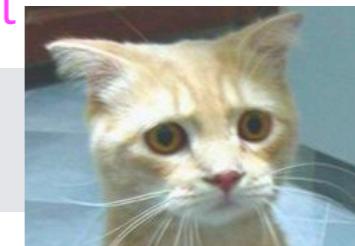
We can look to the function of the affix (denoting past tense) to include irregular inflections.

	-s	-ed	-ing
walk	walks	walked	walking
sleep	sleeps	slept	sleeping
eat	eats	ate	eating
give	gives	gave	giving

Morphological distribution

We can look to the function of the affix (denoting past tense) to include irregular inflections.

	-s	-ed	-ing
walk	walks	walked	walking
sleep	sleeps	slept	sleeping
eat	eats	ate	eating
give	gives	gave	giving



Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Kim saw the elephant before we did

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Kim saw the elephant before we did

dog

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Kim saw the elephant before we did

dog

idea

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Kim saw the elephant before we did

dog

idea

*of

Syntactic distribution

- Substitution test: if a word is replaced by another word, does the sentence remain **grammatical**?

Kim saw the elephant before we did

dog

idea

*of

*goes

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the elephant before we did

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the

elephant

before we did

*Sandy

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the

elephant

before we did

*Sandy

both nouns but
common vs. proper

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the

elephant

before we did

*Sandy

both nouns but
common vs. proper

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the elephant before we did

*Sandy

both nouns but
common vs. proper

Kim *arrived the elephant before we did

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the elephant before we did

*Sandy

both nouns but
common vs. proper

Kim *arrived the elephant before we did

Syntactic distribution

- These can often be too strict; some contexts admit substitutability for some pairs but not others.

Kim saw the

elephant

before we did

*Sandy

both nouns but
common vs. proper

Kim *arrived the

elephant

before we did

both verbs but
transitive vs. intransitive

Nouns

People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesterday”)

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesterday”)
Determiner	Mark the beginning of a noun phrase (“a dog”)

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesterday”)
Determiner	Mark the beginning of a noun phrase (“a dog”)
Pronouns	Refer to a noun phrase (he, she, it)

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesterday”)
Determiner	Mark the beginning of a noun phrase (“a dog”)
Pronouns	Refer to a noun phrase (he, she, it)
Prepositions	Indicate spatial/temporal relationships (on the table)

Nouns	People, places, things, actions-made-nouns (“I like swimming”). Inflected for singular/plural
Verbs	Actions, processes. Inflected for tense, aspect, number, person
Adjectives	Properties, qualities. Usually modify nouns
Adverbs	Qualify the manner of verbs (“She ran downhill extremely quickly yesterday”)
Determiner	Mark the beginning of a noun phrase (“a dog”)
Pronouns	Refer to a noun phrase (he, she, it)
Prepositions	Indicate spatial/temporal relationships (on the table)
Conjunctions	Conjoin two phrases, clauses, sentences (and, or)

Closed class

Open class

Open class

Nouns

affluenza, subtweet, bitcoin, cronut, emoji, listicle,
dumpster fire, mocktail, glamping, skort, avo

Closed class

Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram

Open class

Closed class

Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
Adjectives	crunk, amazeballs, post-truth, woke, bougie

Closed class

Open class

Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
Adjectives	crunk, amazeballs, post-truth, woke, bougie
Adverbs	hella, wicked

Closed class

Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
Adjectives	crunk, amazeballs, post-truth, woke, bougie
Adverbs	hella, wicked
Determiner	

Open class	Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
	Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
	Adjectives	crunk, amazeballs, post-truth, woke, bougie
	Adverbs	hella, wicked
	Determiner	
	Pronouns	

Closed class

Open class	Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
	Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
	Adjectives	crunk, amazeballs, post-truth, woke, bougie
	Adverbs	hella, wicked
Closed class	Determiner	
	Pronouns	
	Prepositions	English has a new preposition, because internet [Garber 2013; Pullum 2014]

Open class	Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
	Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
	Adjectives	crunk, amazeballs, post-truth, woke, bougie
	Adverbs	hella, wicked
	Determiner	
	Pronouns	
	Prepositions	English has a new preposition, because internet [Garber 2013; Pullum 2014]
	Conjunctions	

Open class	Nouns	affluenza, subtweet, bitcoin, cronut, emoji, listicle, dumpster fire, mocktail, glamping, skort, avo
	Verbs	text, chillax, manspreading, photobomb, unfollow, google, instagram
	Adjectives	crunk, amazeballs, post-truth, woke, bougie
	Adverbs	hella, wicked
	Determiner	OOV? Guess Noun
	Pronouns	
	Prepositions	English has a new preposition, because internet [Garber 2013; Pullum 2014]
	Conjunctions	

POS tagging

Labeling the tag that's correct
for the context.

Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

NN

Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

VBZ

NN

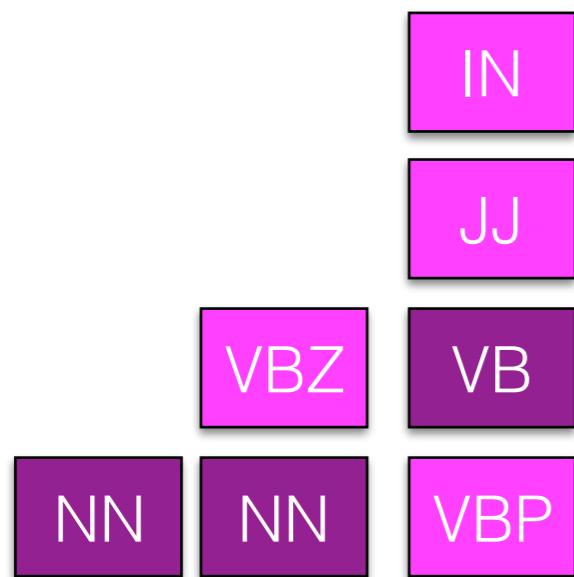
NN

Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

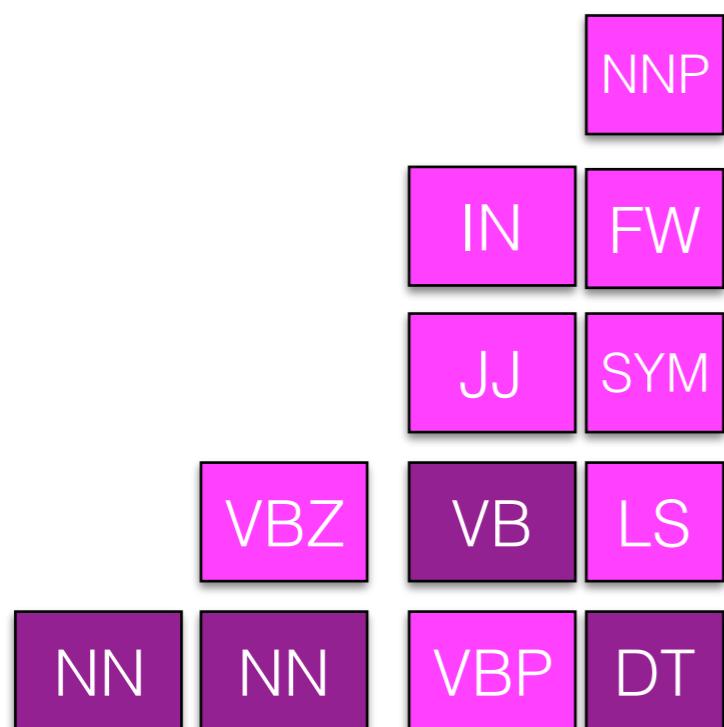


Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

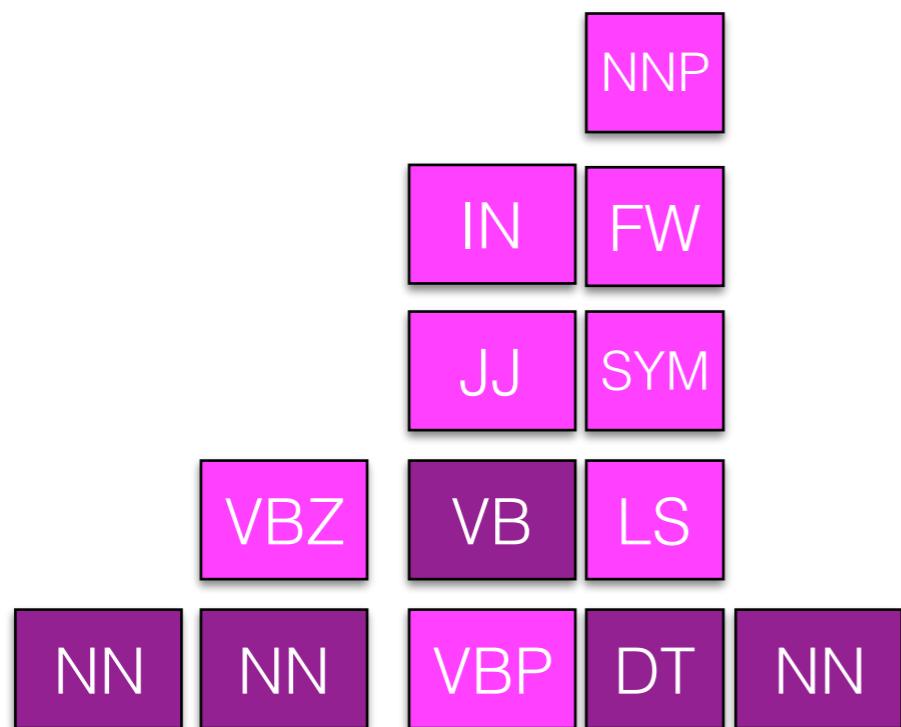


Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

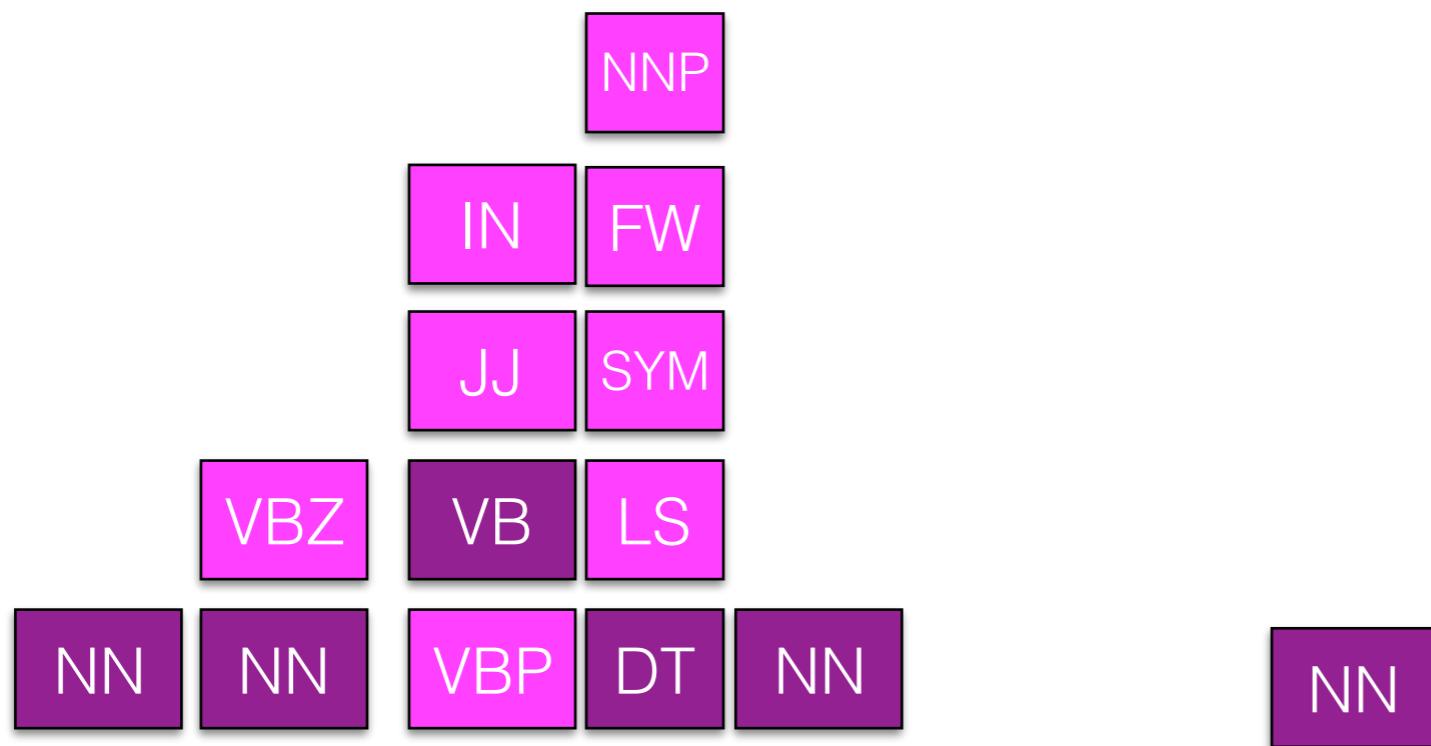


Fruit flies like a banana

Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.

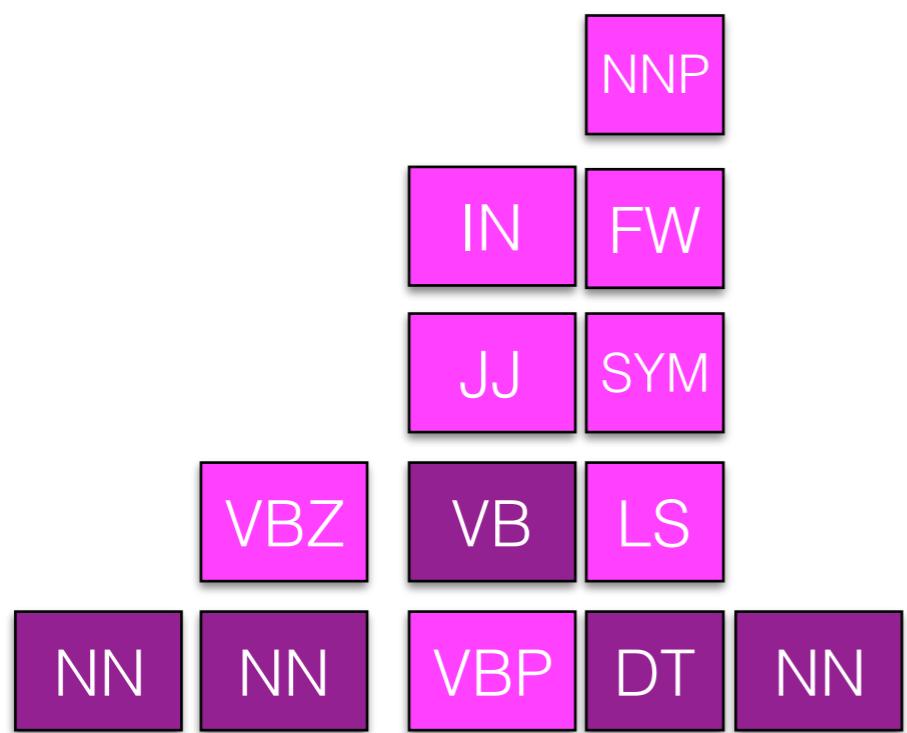


Fruit flies like a banana

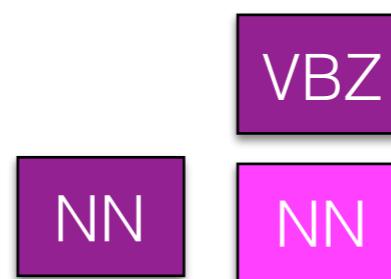
Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.



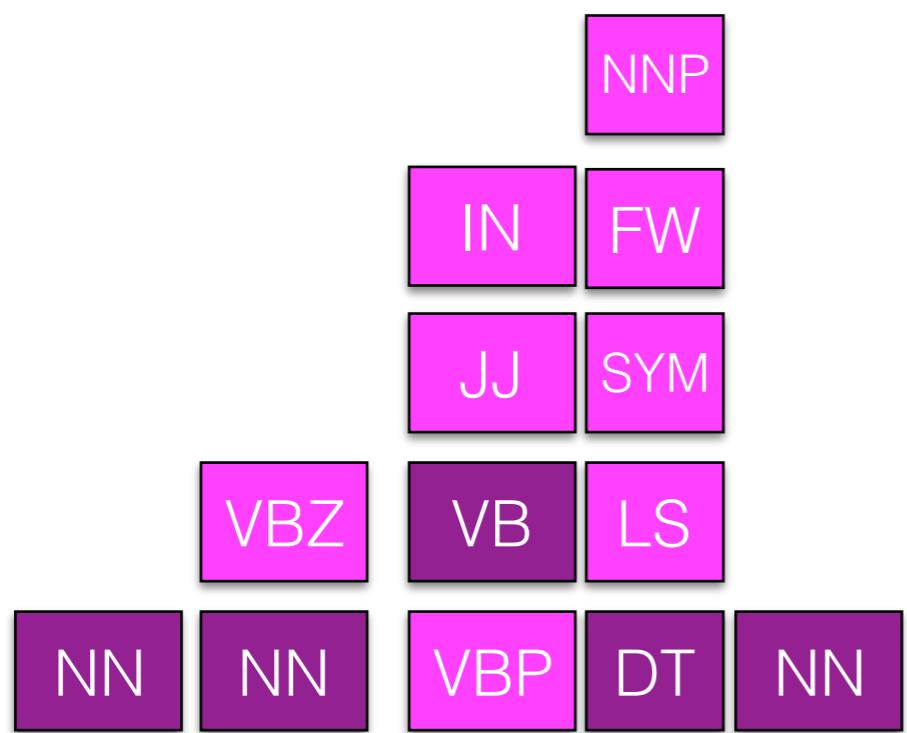
Fruit flies like a banana



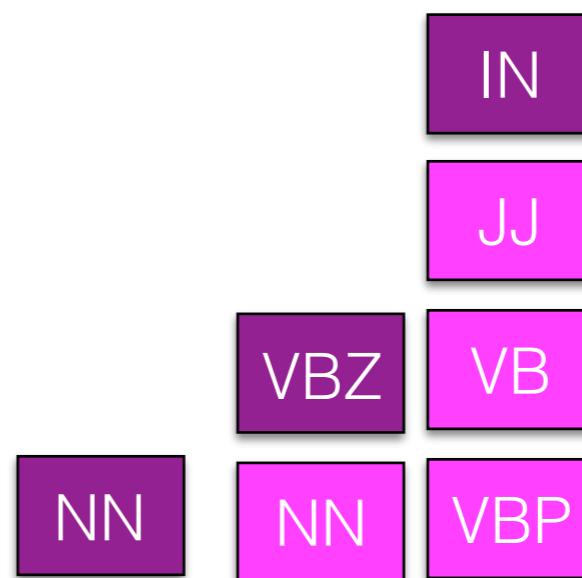
Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.



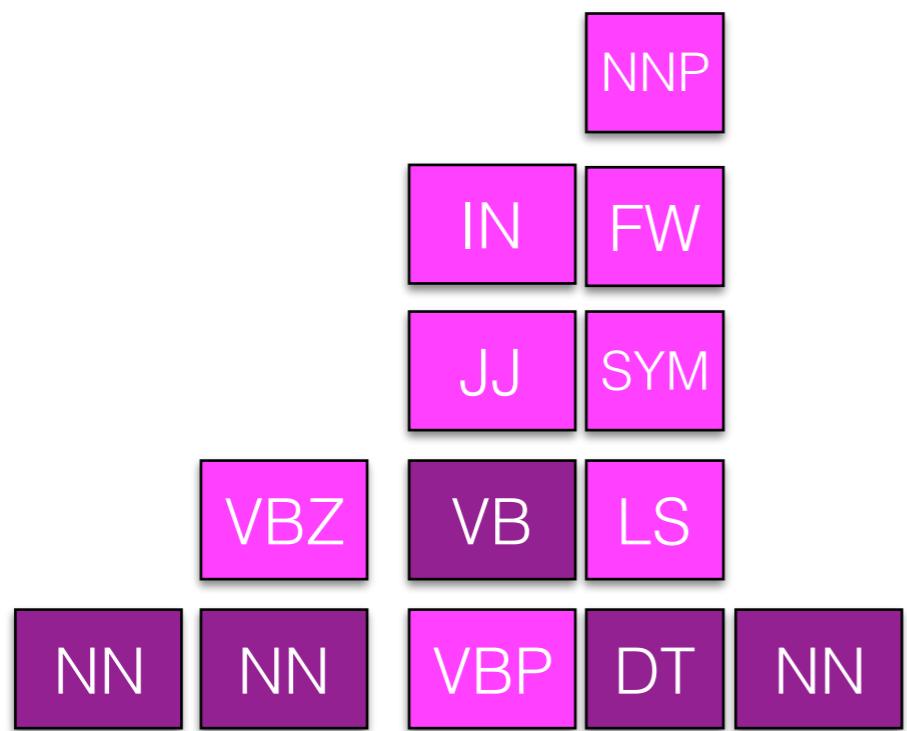
Fruit flies like a banana



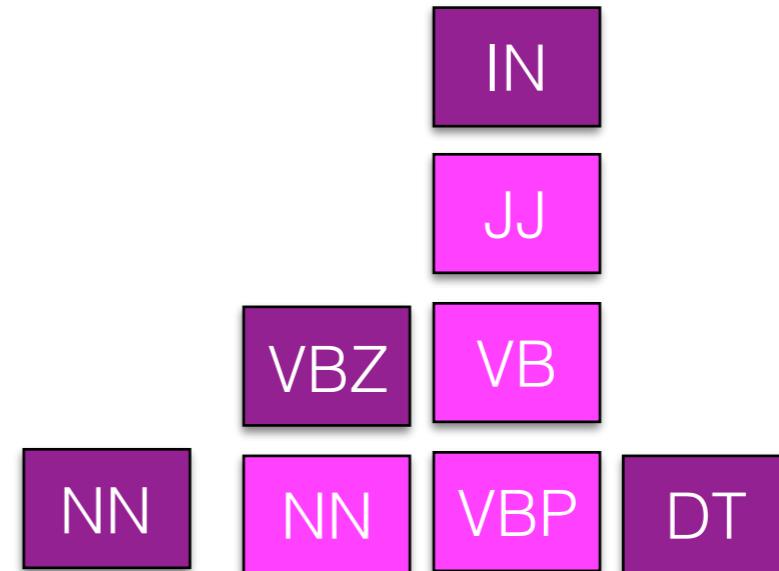
Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.



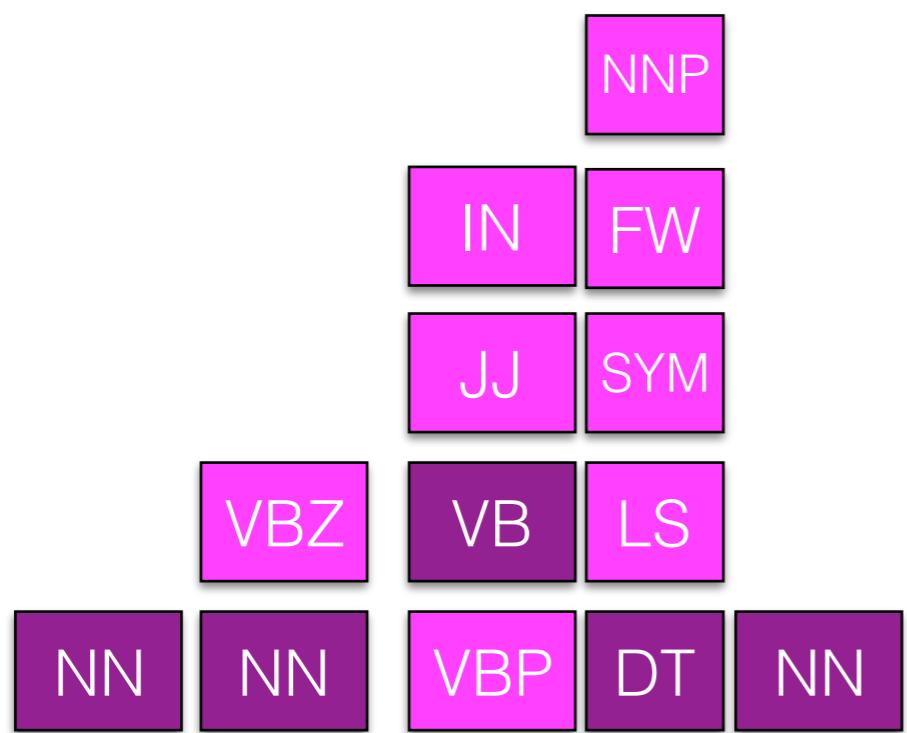
Fruit flies like a banana



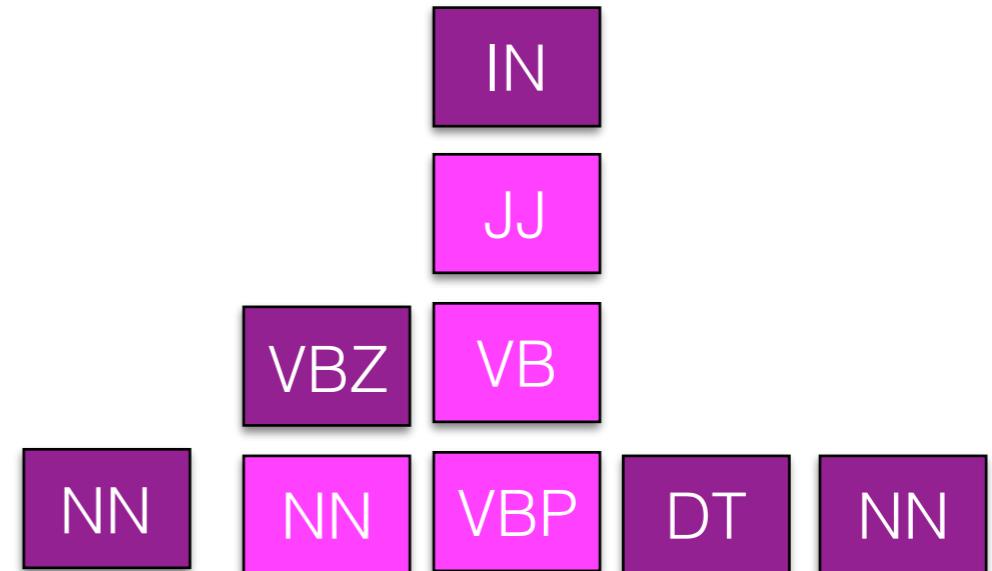
Time flies like an arrow

POS tagging

Labeling the tag that's correct
for the context.



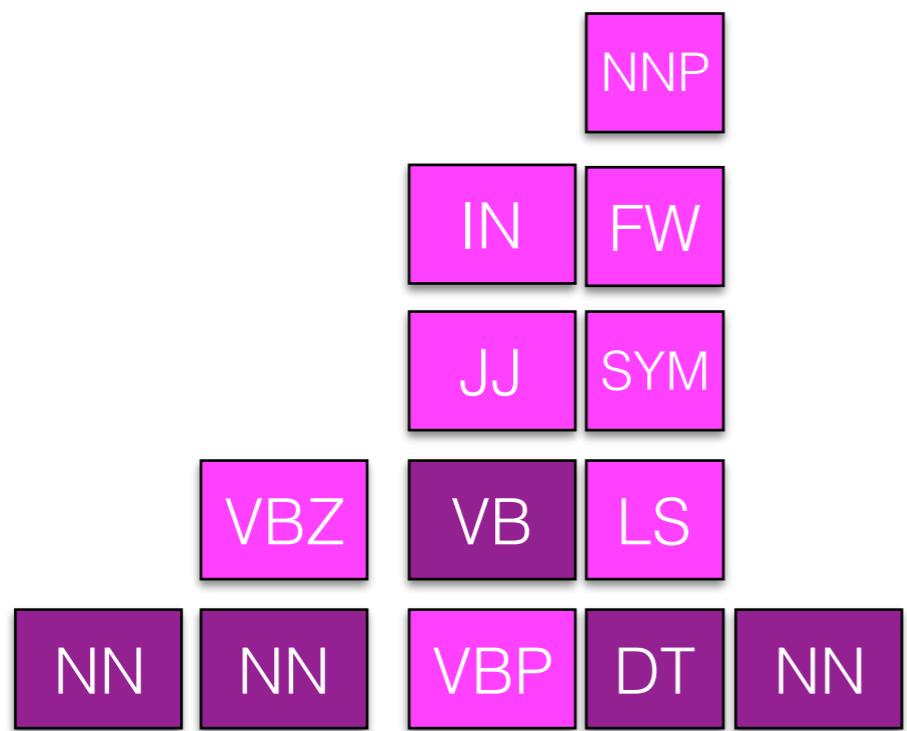
Fruit flies like a banana



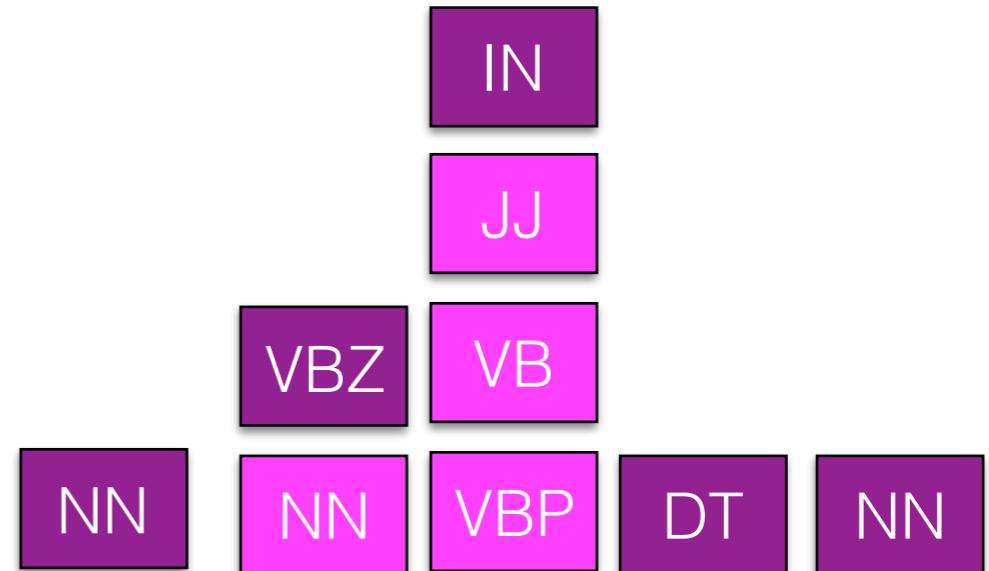
Time flies like an arrow

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!)

State of the art

Manning 2011

State of the art

- Baseline: Most frequent class = 92.34%

State of the art

- Baseline: Most frequent class = 92.34%

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
 - Substantial drop across domains (e.g., train on news, test on literature)

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
 - Substantial drop across domains (e.g., train on news, test on literature)

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
 - Substantial drop across domains (e.g., train on news, test on literature)

State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
 - Substantial drop across domains (e.g., train on news, test on literature)
- Whole sentence accuracy: 55%

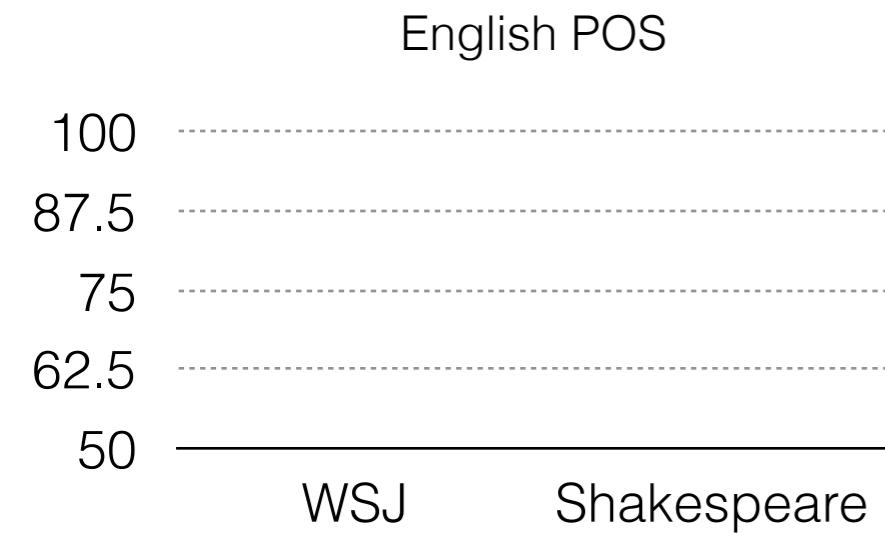
State of the art

- Baseline: Most frequent class = 92.34%
- Token accuracy: 97% (English news)
[Toutanova et al. 2003; Søgaard 2010]
 - Optimistic: includes punctuation, words with only one tag (deterministic tagging)
 - Substantial drop across domains (e.g., train on news, test on literature)
- Whole sentence accuracy: 55%

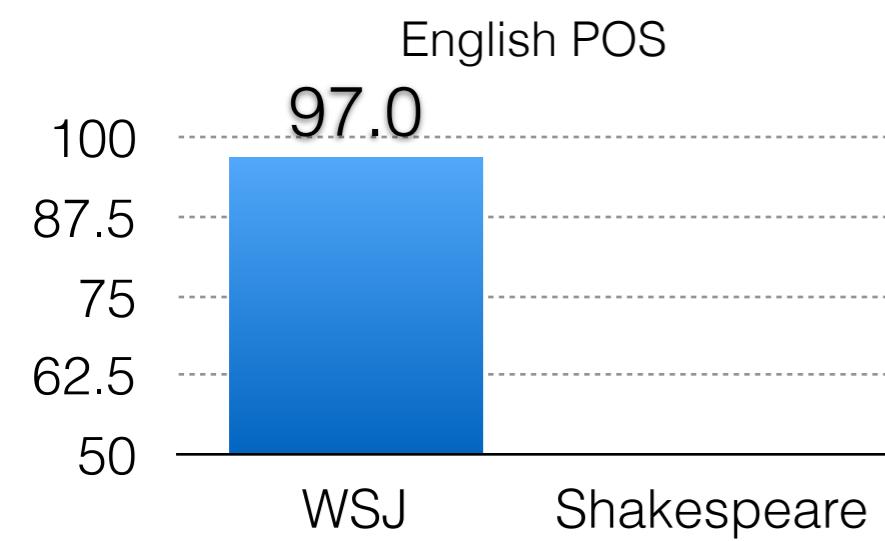
Motivation for why we
need to compare
against baselines!

Domain difference

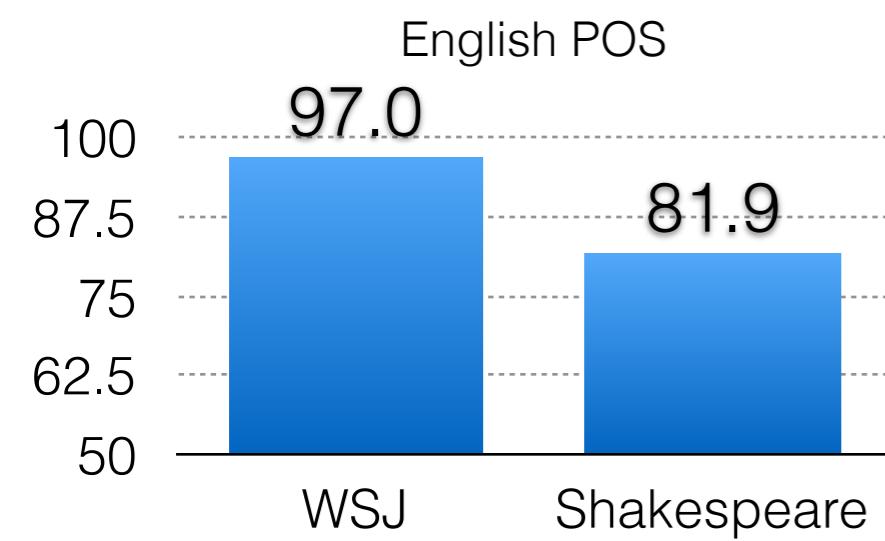
Domain difference



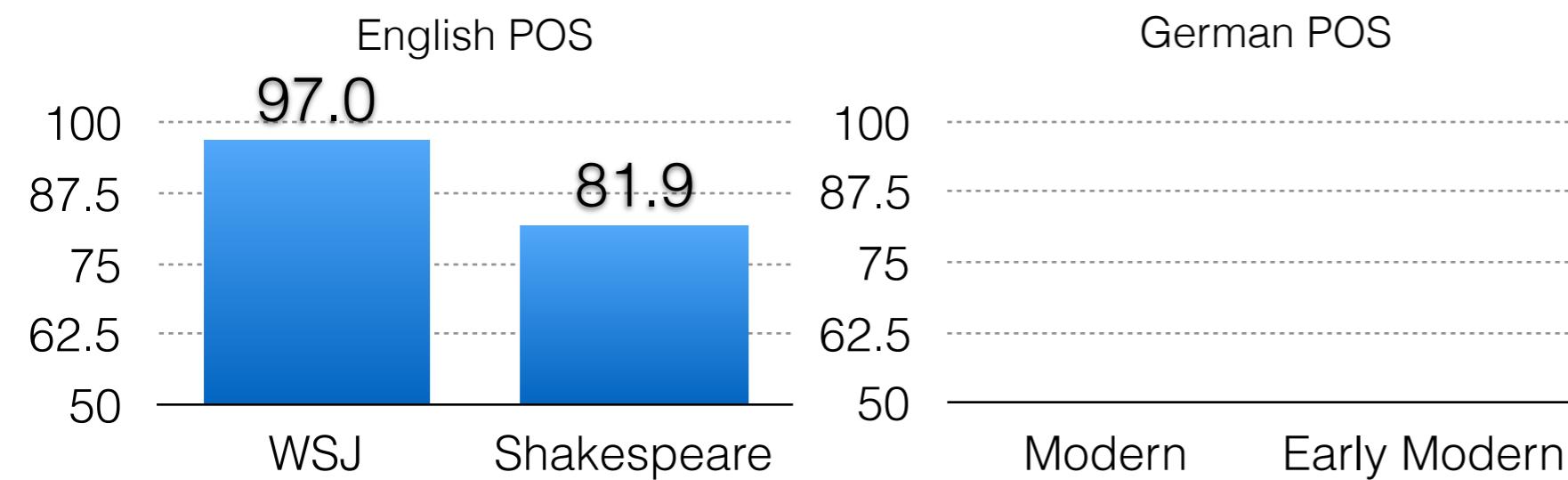
Domain difference



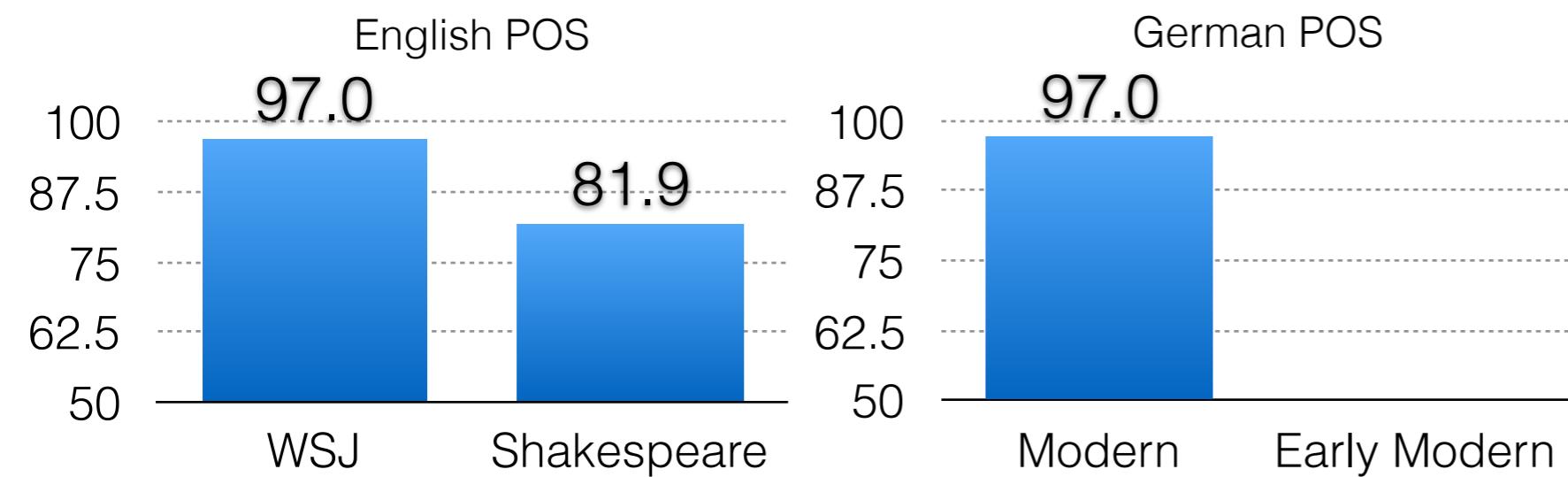
Domain difference



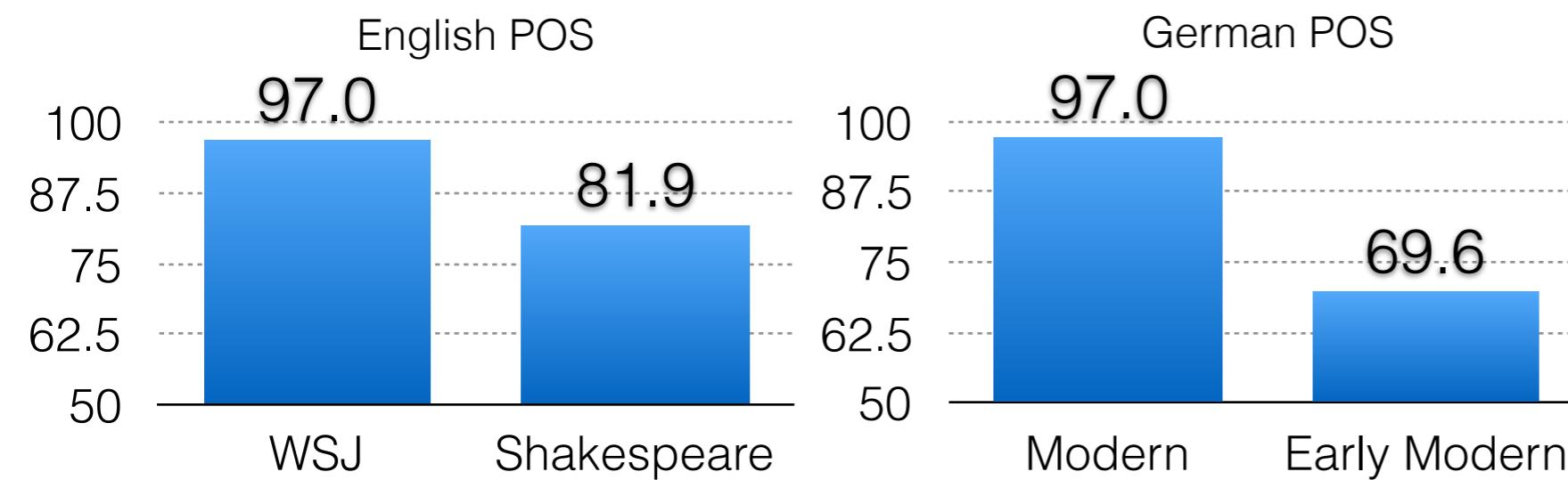
Domain difference



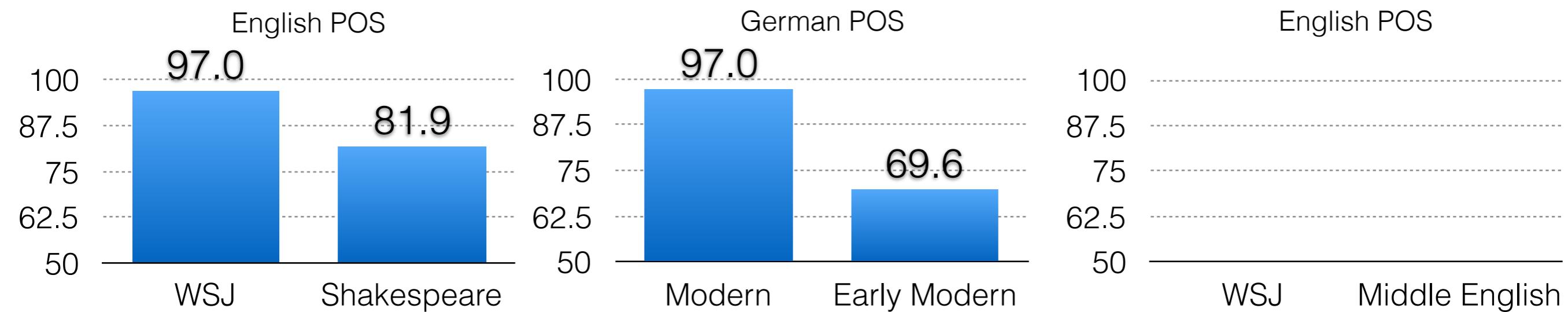
Domain difference



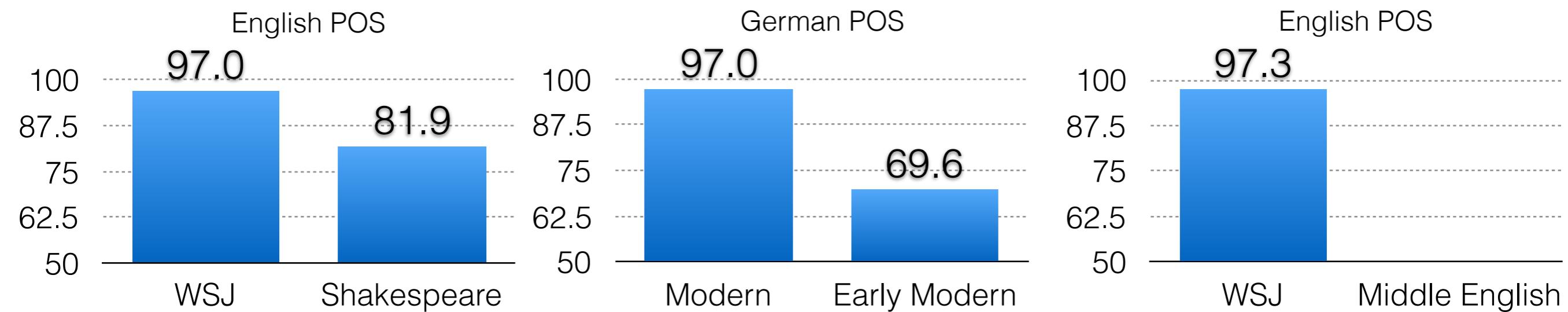
Domain difference



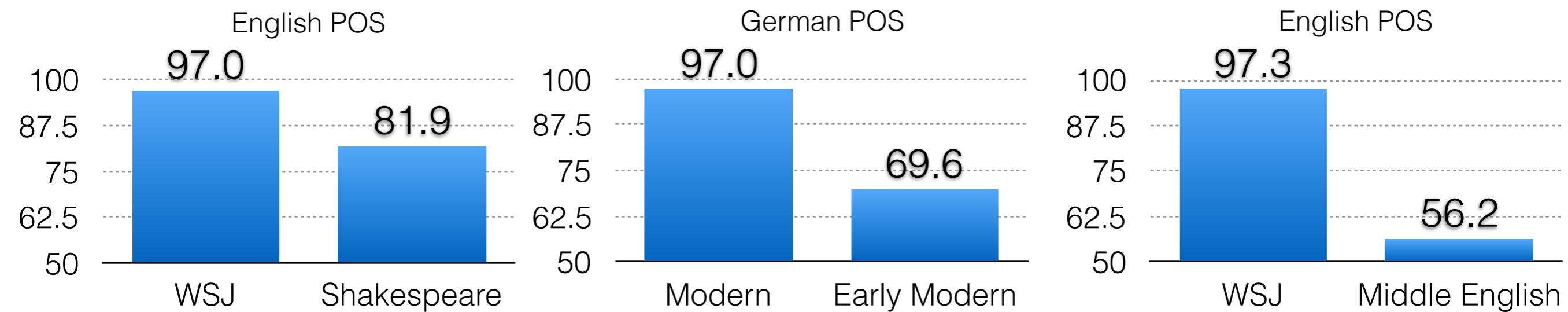
Domain difference



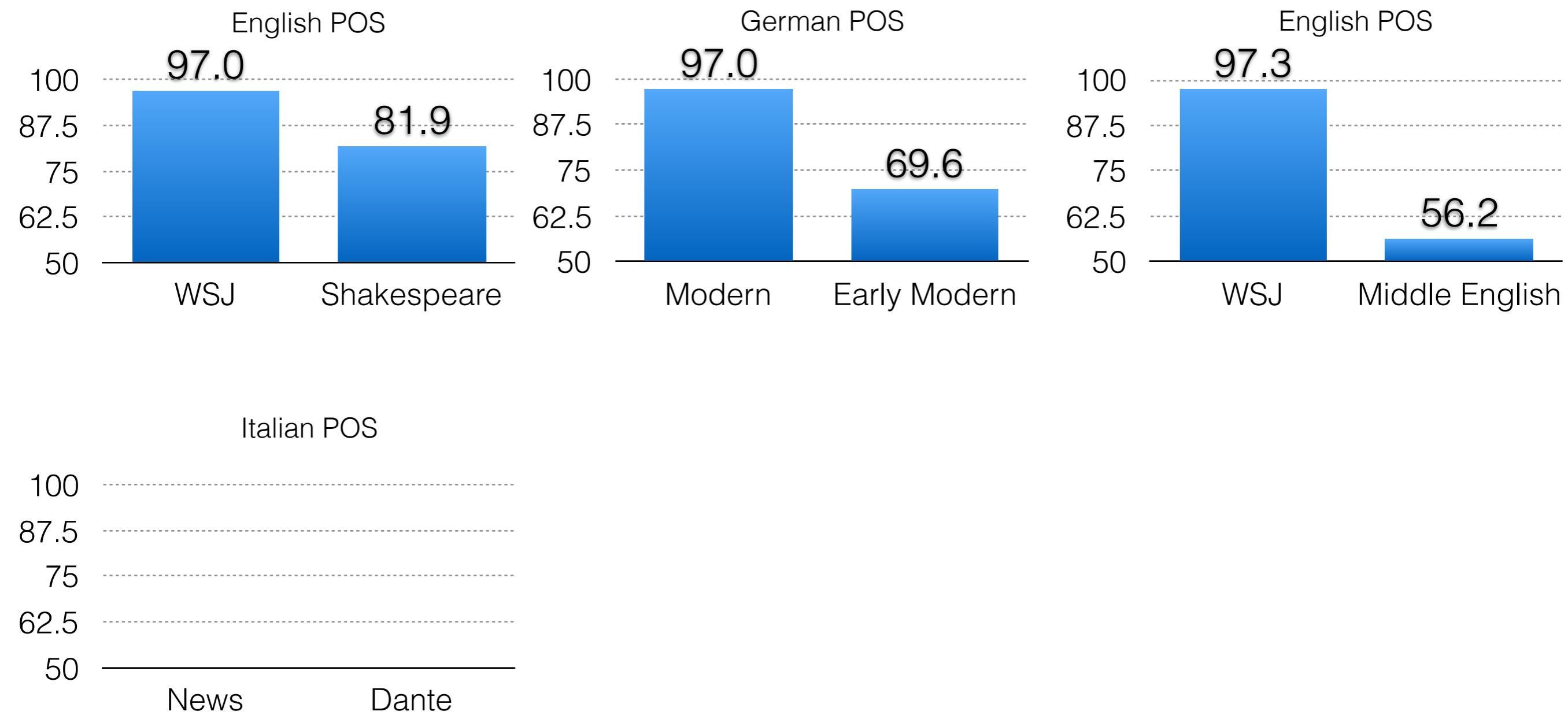
Domain difference



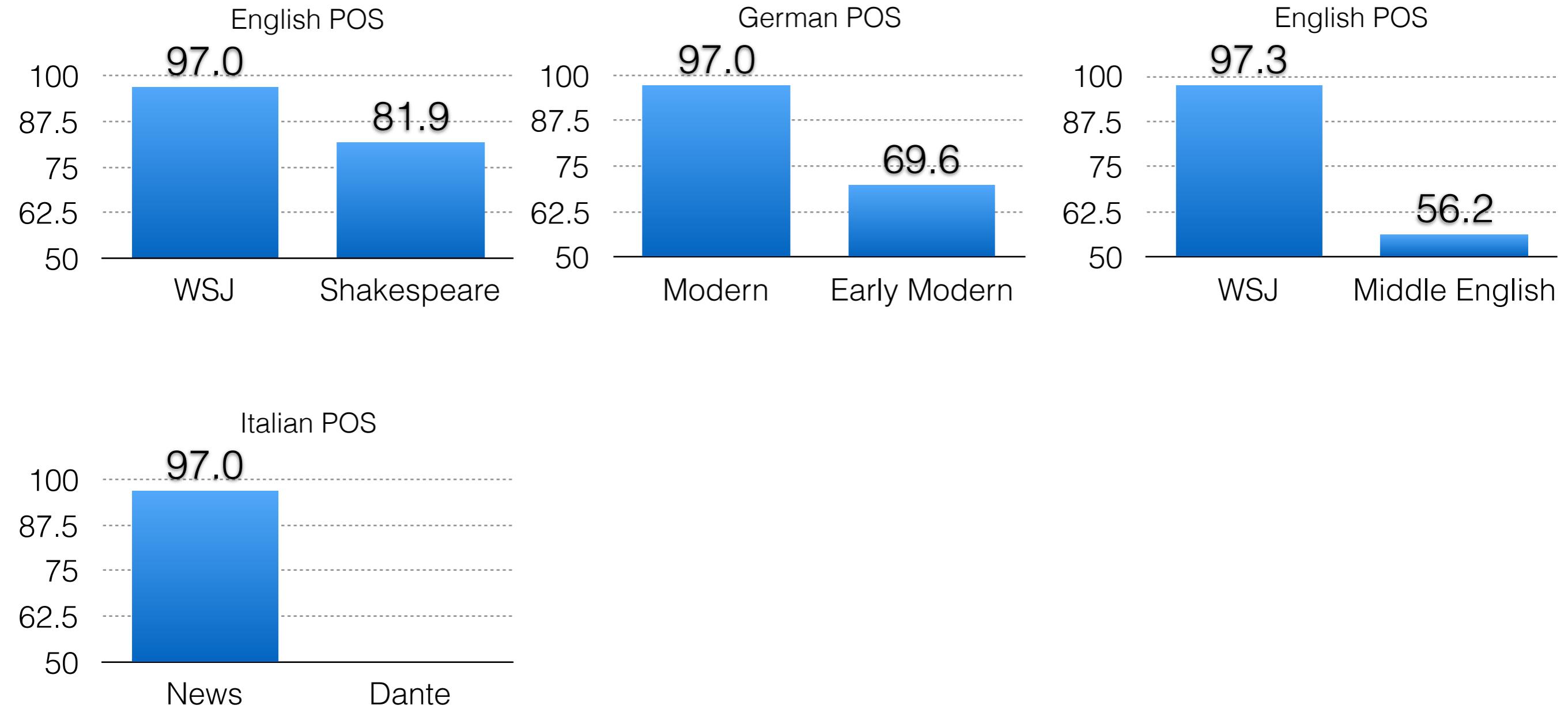
Domain difference



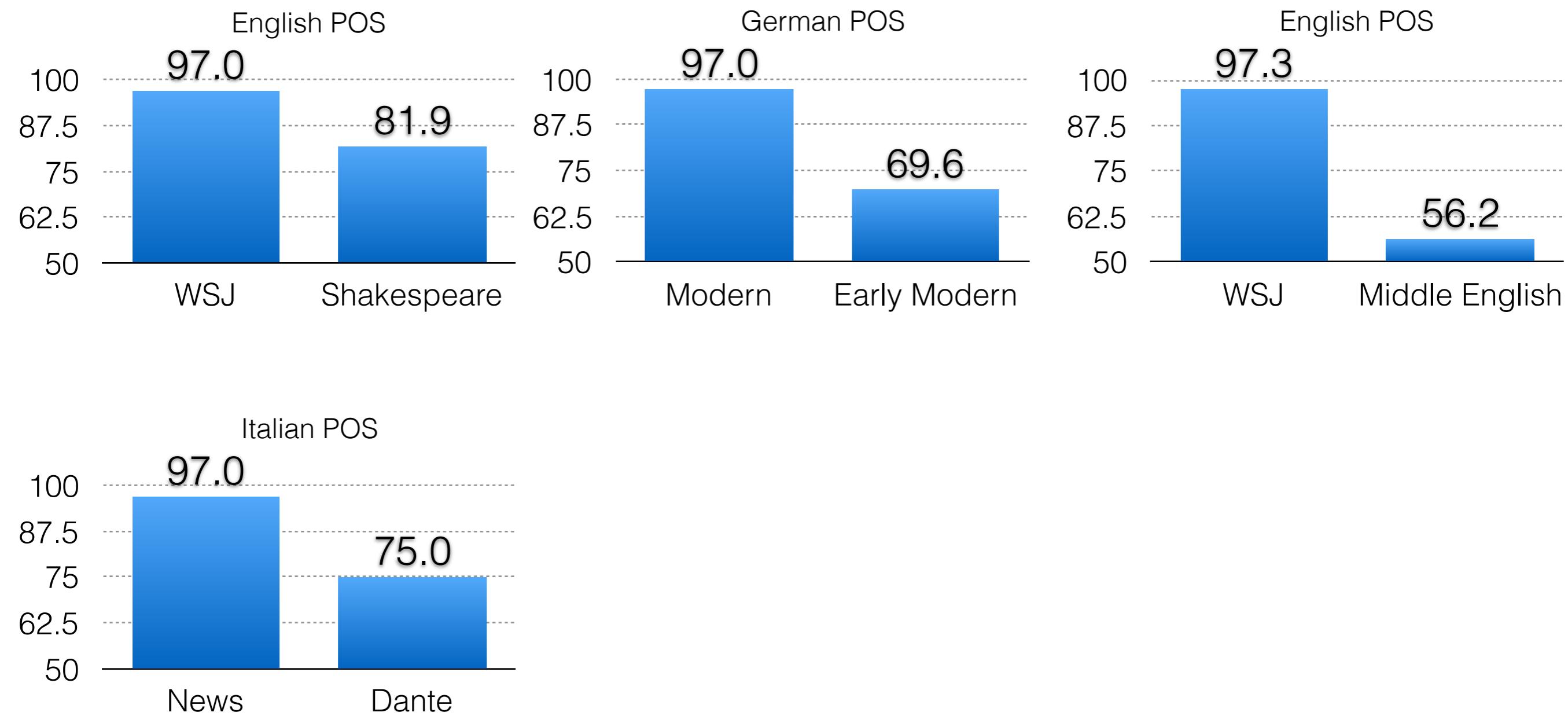
Domain difference



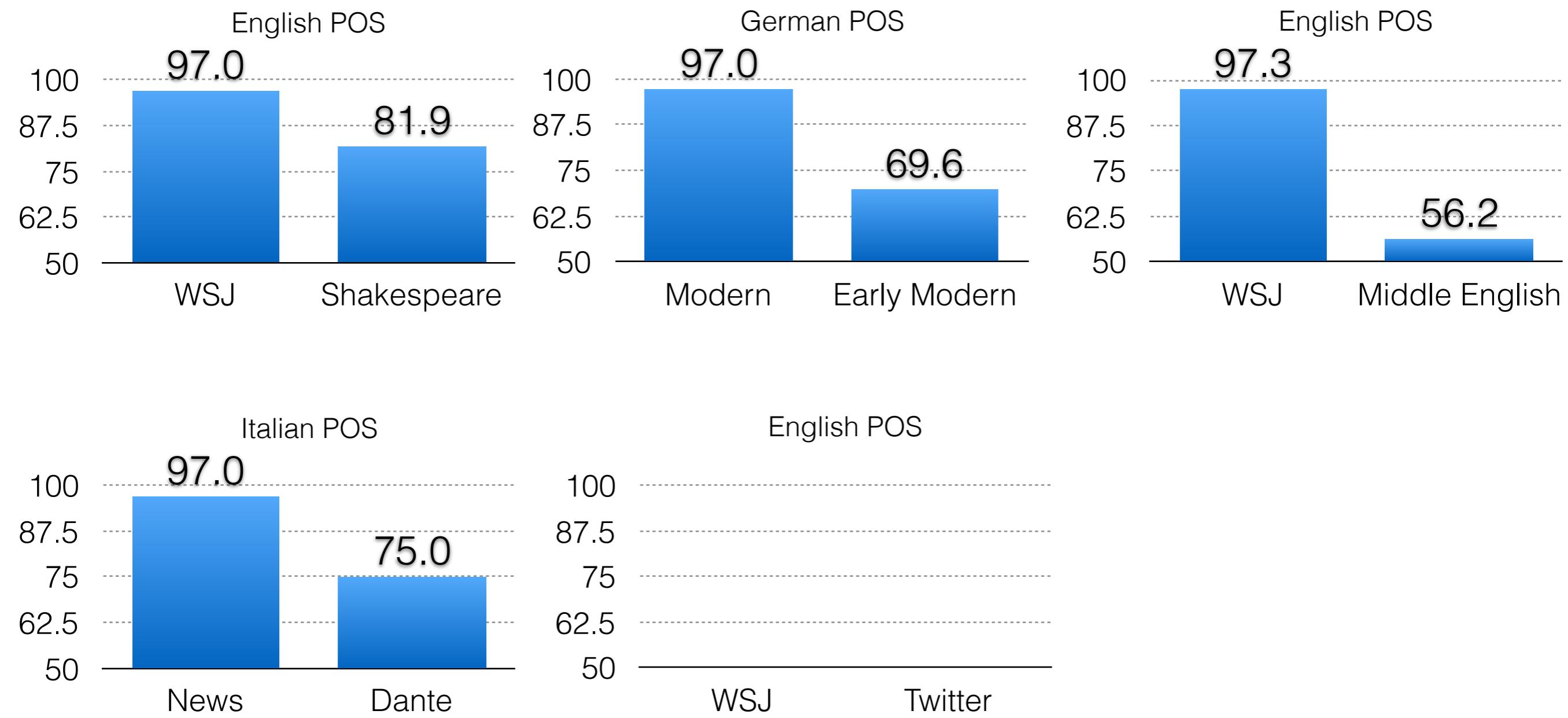
Domain difference



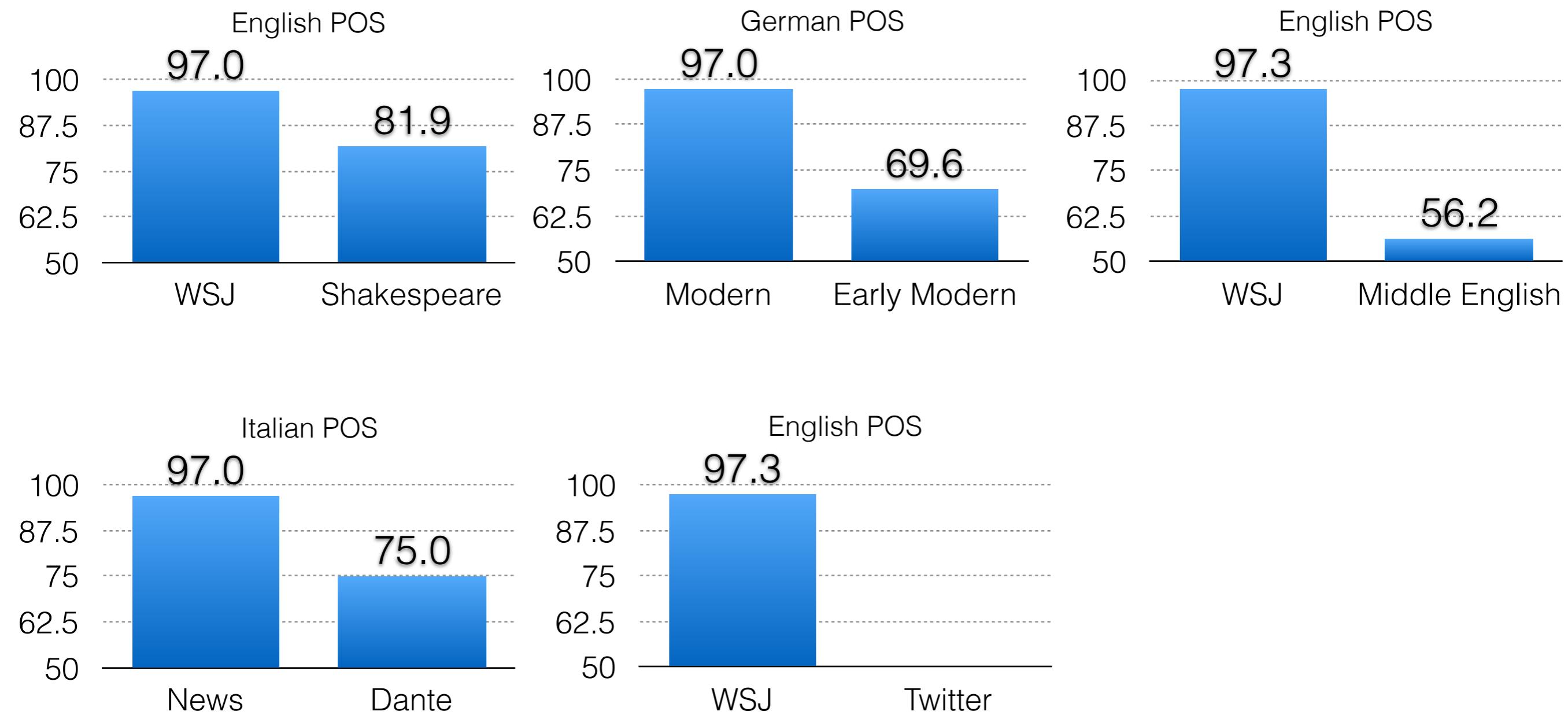
Domain difference



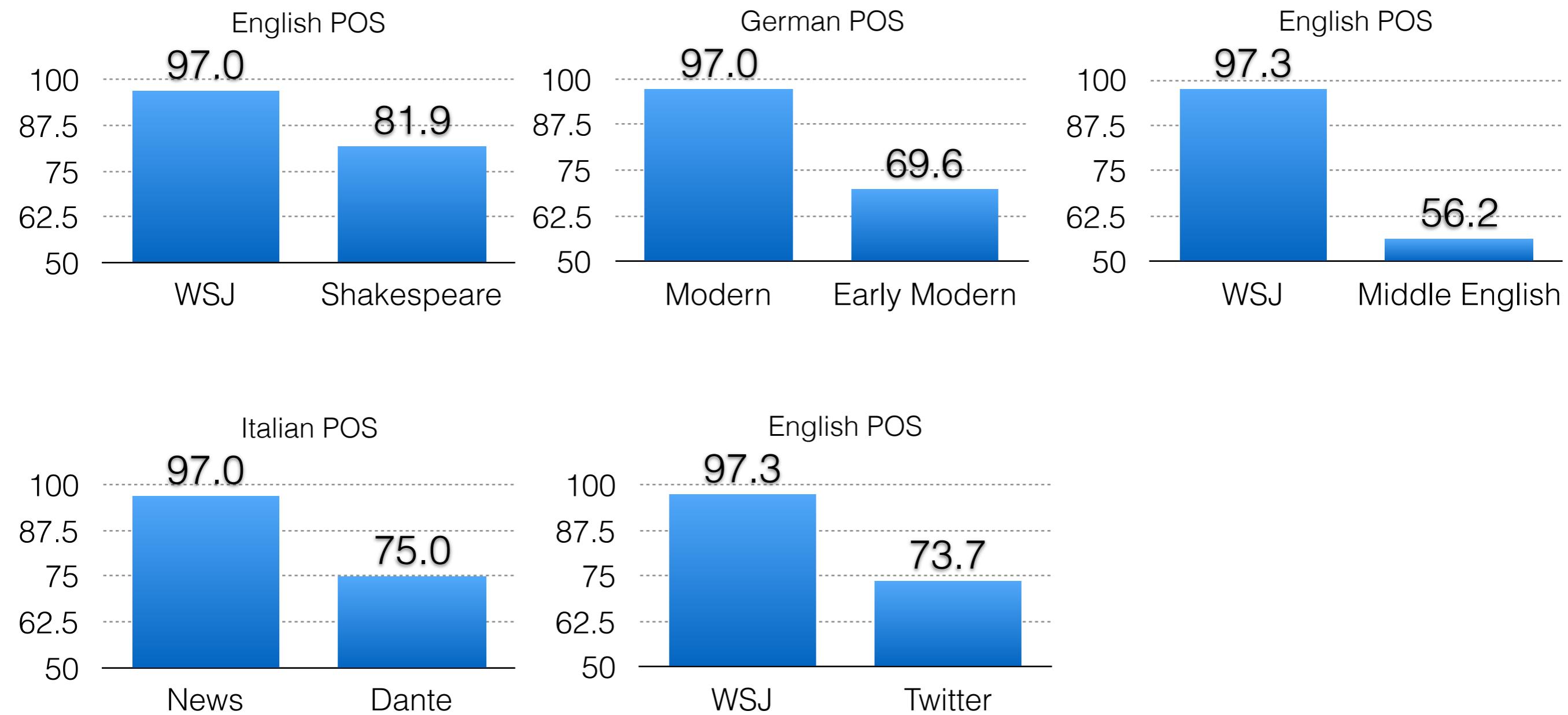
Domain difference



Domain difference



Domain difference



Sources of error

Lexicon gap

4.5%

a 60% slash/NN the common stock dividend

Sources of error

Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction

Sources of error

Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction
Could plausibly get right	16.0%	market players overnight/RB in Tokyo began bidding up oil prices

Sources of error

Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction
Could plausibly get right	16.0%	market players overnight/RB in Tokyo began bidding up oil prices
Difficult linguistics	19.5%	They set/VBP up absurd situations, detached from reality

Sources of error

Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction
Could plausibly get right	16.0%	market players overnight/RB in Tokyo began bidding up oil prices
Difficult linguistics	19.5%	They set/VBP up absurd situations, detached from reality
Underspecified/unclear	12.0%	it will take a \$ 10 million fourth-quarter charge against/IN discontinued/JJ

Sources of error

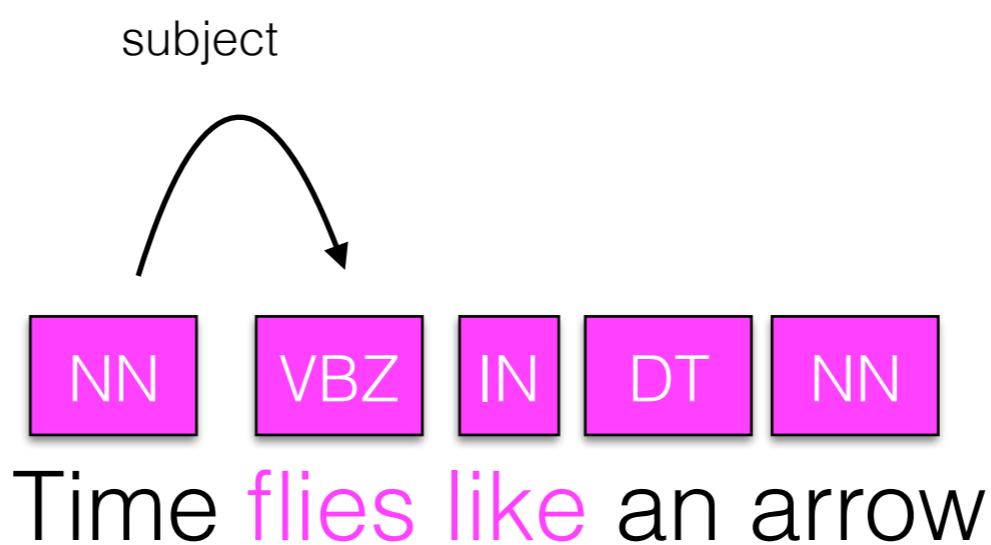
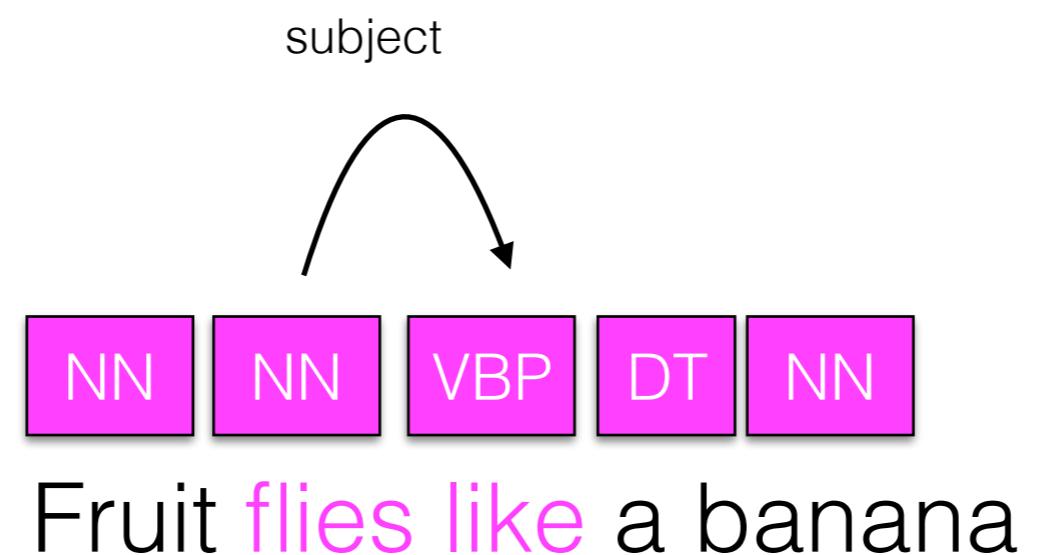
Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction
Could plausibly get right	16.0%	market players overnight/RB in Tokyo began bidding up oil prices
Difficult linguistics	19.5%	They set/VBP up absurd situations, detached from reality
Underspecified/unclear	12.0%	it will take a \$ 10 million fourth-quarter charge against/IN discontinued/JJ
Inconsistent/no standard	28.0%	Orson Welles 's Mercury Theater in the '30s/NNS

Sources of error

Lexicon gap	4.5%	a 60% slash/NN the common stock dividend
Unknown word	4.5%	blaming the disaster on substandard/JJ construction
Could plausibly get right	16.0%	market players overnight/RB in Tokyo began bidding up oil prices
Difficult linguistics	19.5%	They set/VBP up absurd situations, detached from reality
Underspecified/unclear	12.0%	it will take a \$ 10 million fourth-quarter charge against/IN discontinued/JJ
Inconsistent/no standard	28.0%	Orson Welles 's Mercury Theater in the '30s/NNS
Gold standard wrong	15.5%	Our market got hit/VB a lot harder on Monday than the listed market

Why is part of speech tagging useful?

POS indicative of syntax



POS indicative of MWE

at least one adjective/noun or noun phrase

and definitely
one noun

$$((A \mid N)^+ \mid ((A \mid N)^*(NP))(A \mid N)^*)N$$

AN: linear function; lexical ambiguity; mobile phase

NN: regression coefficients; word sense; surface area

AAN: Gaussian random variable; lexical conceptual paradigm; aqueous mobile phase

ANN: cumulative distribution function; lexical ambiguity resolution; accessible surface area

NAN: mean squared error; domain independent set; silica based packing

NNN: class probability function; text analysis system; gradient elution chromatography

NPN: degrees of freedom; [*no example*]; energy of adsorption

POS is indicative of
pronunciation

POS is indicative of pronunciation

Noun

Verb

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me
That is an insult	Don't insult me

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me
That is an insult	Don't insult me
Rebel without a cause	He likes to rebel

POS is indicative of pronunciation

Noun	Verb
My conduct is great	I conduct myself well
She won the contest	I contest the ticket
He is my escort	He escorted me
That is an insult	Don't insult me
Rebel without a cause	He likes to rebel
He is a suspect	I suspect him

Tagsets

- Penn Treebank
- Universal Dependencies
- Twitter POS



Meet the (part of
speech) family

What kind of POS tags are there?

What kind of POS tags are there?

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	<i>and, but, or</i>	SYM	symbol	+%, &
CD	cardinal number	<i>one, two</i>	TO	“to”	<i>to</i>
DT	determiner	<i>a, the</i>	UH	interjection	<i>ah, oops</i>
EX	existential ‘there’	<i>there</i>	VB	verb base form	<i>eat</i>
FW	foreign word	<i>mea culpa</i>	VBD	verb past tense	<i>ate</i>
IN	preposition/sub-conj	<i>of, in, by</i>	VBG	verb gerund	<i>eating</i>
JJ	adjective	<i>yellow</i>	VBN	verb past participle	<i>eaten</i>
JJR	adj., comparative	<i>bigger</i>	VBP	verb non-3sg pres	<i>eat</i>
JJS	adj., superlative	<i>wildest</i>	VBZ	verb 3sg pres	<i>eats</i>
LS	list item marker	<i>1, 2, One</i>	WDT	wh-determiner	<i>which, that</i>
MD	modal	<i>can, should</i>	WP	wh-pronoun	<i>what, who</i>
NN	noun, sing. or mass	<i>llama</i>	WP\$	possessive wh-	<i>whose</i>
NNS	noun, plural	<i>llamas</i>	WRB	wh-adverb	<i>how, where</i>
NNP	proper noun, sing.	<i>IBM</i>	\$	dollar sign	\$
NNPS	proper noun, plural	<i>Carolinas</i>	#	pound sign	#
PDT	predeterminer	<i>all, both</i>	“	left quote	‘ or “
POS	possessive ending	<i>'s</i>	”	right quote	’ or ”
PRP	personal pronoun	<i>I, you, he</i>	(left parenthesis	[, (, {, <
PRP\$	possessive pronoun	<i>your, one's</i>)	right parenthesis],), }, >
RB	adverb	<i>quickly, never</i>	,	comma	,
RBR	adverb, comparative	<i>faster</i>	.	sentence-final punc	. ! ?
RBS	adverb, superlative	<i>fastest</i>	:	mid-sentence punc	: ; ... --
RP	particle	<i>up, off</i>			

Verbs

tag	description	example
VB	base form	I want to like
VBD	past tense	I/we/he/she/you liked
VBG	present participle	He was liking it
VBN	past participle	I had liked it
VBP	present (non 3rd-sing)	I like it
VBZ	present (3rd-sing)	He likes it
MD	modal verbs	He can go

Nouns

non-proper

tag	description	example
NN	non-proper, singular or mass	the company
NNS	non-proper, plural	the companies
NNP	proper, singular	Carolina
NNPS	proper, plural	Carolinas

proper

JJ

(Adjectives)

- General adjectives
 - *happy person*
 - *new mail*
- Ordinal numbers
 - *fourth person*

2002	other/jj
1925	new/jj
1563	last/jj
1174	many/jj
1142	such/jj
1058	first/jj
824	major/jj
715	federal/jj
698	next/jj
644	financial/jj

DT (Article)

- Articles (*a*, *the*, *every*, *no*)
- Indefinite determiners
(*another*, *any*, *some*, *each*)
- *That*, *these*, *this*, *those* when preceding noun
- *All*, *both* when not preceding another determiner or possessive pronoun

65548	<i>the/dt</i>
26970	<i>a/dt</i>
4405	<i>an/dt</i>
3115	<i>this/dt</i>
2117	<i>some/dt</i>
2102	<i>that/dt</i>
1274	<i>all/dt</i>
1085	<i>any/dt</i>
953	<i>no/dt</i>
778	<i>those/dt</i>

RB (Adverb)

- Most words that end in -ly
- Degree words (quite, too, very)
- Negative markers: not, n't, never

4410	n't/rb
2071	also/rb
1858	not/rb
1109	now/rb
1070	only/rb
1027	as/rb
961	even/rb
839	so/rb
810	about/rb
804	still/rb

IN (preposition, subordinating conjunction)

- All prepositions
(except *to*) and
subordinating
conjunctions
 - He jumped **on** the
table **because** he
was excited
- | | |
|-------|---------|
| 31111 | of/in |
| 22967 | in/in |
| 11425 | for/in |
| 7181 | on/in |
| 6684 | that/in |
| 6399 | at/in |
| 6229 | by/in |
| 5940 | from/in |
| 5874 | with/in |
| 5239 | as/in |

CC (Coordinating conjunction)

- And, but, not, or 22362 and/cc
- Math operators (plus, 4604 but/cc
minor, less, times) 3436 or/cc
1410 &/cc
94 nor/cc
68 either/cc
53 yet/cc
53 plus/cc
37 both/cc
32 neither/cc
- For (meaning “because”)
[he asked to be transferred, for he was unhappy]

Quiz Time

IN or RB

The credit car you won't want to do **without**

We'll just have to do **without**

Santorini 1990

IN or RB

- Prepositions usually precede noun phrases (to form a prepositional phrase) but don't have to

IN

The credit car you won't want to do **without**

RB

We'll just have to do **without**

IN or RB

Blaze out into space

Come out of the woodwork

Santorini 1990

IN or RB

- A preposition may precede another preposition

Blaze **out into** space

Come **out of** the woodwork

NN or JJ

- Nouns used as modifiers = NN
 - *wool sweater*
 - *life insurance company*
- Substantive adjectives = JJ if they can be modified by an adverb
 - The (very) *rich* pay far too few taxes

JJ or NP/NPS

French cuisine is delicious

The French tend to be inspired cooks

JJ or NP/NPS

- Proper names can be adjectives or nouns

JJ

French cuisine is delicious

NNPS

The French tend to be inspired cooks

JJ or RB

rapid growth

rapid growing plants

Santorini 1990

JJ or RB

- If a word modifies a noun, it's usually an adjective (JJ); if it modifies a non-noun it's typically an adverb (RB)

JJ

rapid growth

RB

rapid growing plants

JJ or VBG

- JJ if it precedes a noun and the corresponding verb is intransitive or does not have the same meaning

an appealing/JJ face;
an appetizing/JJ dish;
a revolving/JJ fund;
a winning/JJ smile;

*a face that appeals
*a dish that appetizes
*a fund that revolves
*a smile that wins

But:

the existing/VBG safeguards;
a holding/VBG company;
a managing/VBG director;
the ruling/VBG class;

safeguards that exist
a company that holds another one
a director who manages
the class that rules

JJ or VBN

- If it's gradable (can insert *very*) = JJ

JJ

- He was very surprised

- If can be followed by a *by* phrase = VBN. If that conflicts with #1 above, then = JJ

VBN

- He was invited by some friends of her

JJ

- He was very surprised by her remarks

NN or VBG

Good **cooking** is something to enjoy

Cooking well is a useful skill

NN or VBG

- Only nouns can be modified by adjectives; only gerunds can be modified by adverbs

NN

Good **cooking** is something to enjoy

VBG

Cooking well is a useful skill

Sometimes you need the POS
tags to match the domain

ikr smh he asked fir yo last name so he can add u on fb lololol

The Penn Treebank POS tags aren't the only option

The Penn Treebank POS tags aren't the only option

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0

The Penn Treebank POS tags aren't the only option

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0

Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

The Penn Treebank POS tags aren't the only option

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0

Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:-(:b (: <3 o_O	1.0

The Penn Treebank POS tags aren't the only option

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0

Other closed-class words			
D	determiner (WDT, DT, WP\$, PRP\$)	the teh its it's	6.5
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., to) 4 (i.e., <i>for</i>)	8.7
&	coordinating conjunction (CC)	and n & + BUT	1.7
T	verb particle (RP)	out off Up UP	0.6
X	existential <i>there</i> , predeterminers (EX, PDT)	both	0.1
Y	X + verbal	there's all's	0.0

Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:‐) :b (: <3 o_O	1.0

The Penn Treebank POS tags aren't the only option

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= <i>I don't know</i>)	1.6
M	proper noun + verbal	Mark'll	0.0

Other closed-class words			
D	determiner (WDT, DT, WP\$, PRP\$)	the teh its it's	6.5
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., to) 4 (i.e., <i>for</i>)	8.7
&	coordinating conjunction (CC)	and n & + BUT	1.7
T	verb particle (RP)	out off Up UP	0.6
X	existential <i>there</i> , predeterminers (EX, PDT)	both	0.1
Y	X + verbal	there's all's	0.0

Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:‐) :b (: <3 o_O	1.0

Miscellaneous			
\$	numeral (CD)	2010 four 9:30	1.5
,	punctuation (#, \$, ' ', (,), , ., :, ` `)	!!! ?!?	11.6
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)	ily (<i>I love you</i>) wby (<i>what about you</i>) 's --> awesome...I'm	1.1

Sometimes you need the POS tags to match the domain

ikr smh he asked fir yo last name so he can add u on fb lololol

word	tag
ikr	!
smh	G
he	O
asked	V
fir	P
yo	D
last	A
name	N
so	P
he	O
can	V
add	V
u	O
on	P
fb	^
lololol	!

Sometimes you need the POS tags to match the domain

ikr smh he asked fir yo last name so he can add u on fb lololol

word	tag
ikr	!
smh	G
he	O
asked	V
fir	P
yo	D
last	A
name	N
so	P
he	O
can	V
add	V
u	O
on	P
fb	^
lololol	!

Other open-class words

V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't is eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

Sometimes you need the POS tags to match the domain

ikr smh he asked fir yo last name so he can add u on fb lololol

word	tag
ikr	!
smh	G
he	O
asked	V
fir	P
yo	D
last	A
name	N
so	P
he	O
can	V
add	V
u	O
on	P
fb	^
lololol	!

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal (= <i>I don't know</i>)	he's book'll iono	1.6
M	proper noun + verbal	Mark'll	0.0

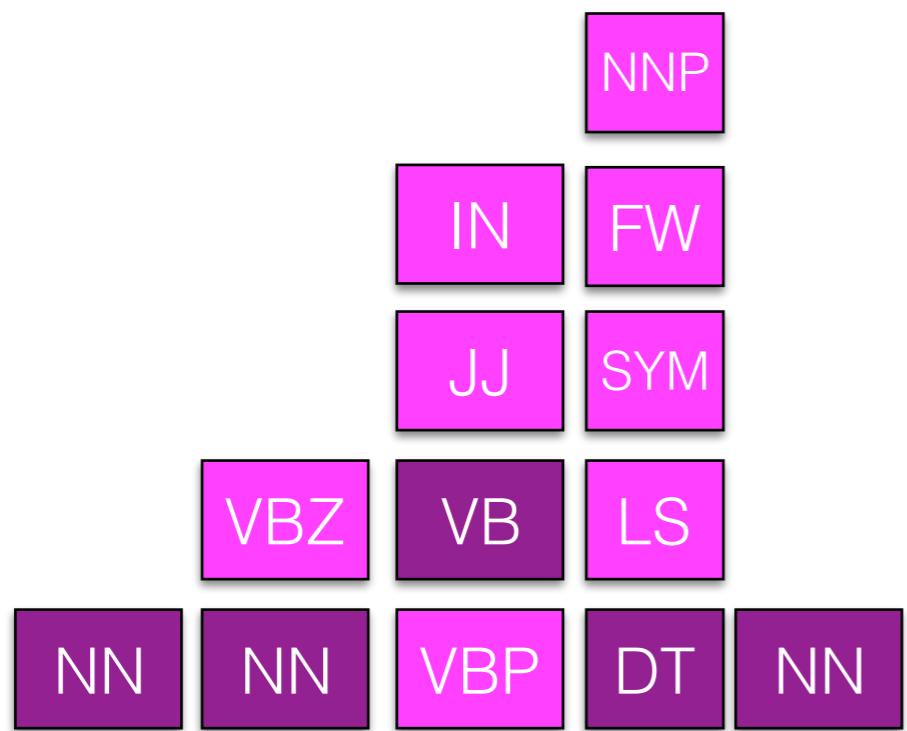
Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., <i>too</i>)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6



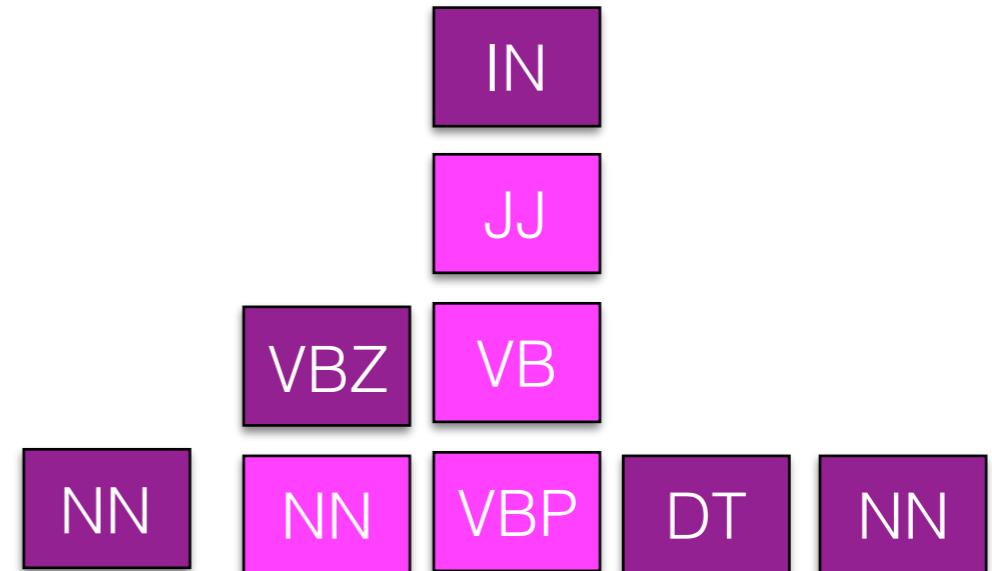
Sequence labeling

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!)

Named entity recognition

PERS PERS

ORG

tim cook is the ceo of apple

3 or 4-class:

- person
- location
- organization
- (misc)

7-class:

- person
- location
- organization
- time
- money
- percent
- date

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.

1	person	7	cognition	13	attribute	19	quantity	25	plant
2	communication	8	possession	14	object	20	motive	26	relation
3	artifact	9	location	15	process	21	animal		
4	act	10	substance	16	Tops	22	body		
5	group	11	state	17	phenomenon	23	feeling		
6	food	12	time	18	event	24	shape		

Book segmentation



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

Majority class

- Pick the label each word is seen most often with in the training data

fruit	flies	like	a	banana
NN 12	VBZ 7	VB 74	FW 8	NN 3
	NNS 1	VBP 31	SYM 13	
		JJ 28	LS 2	
		IN 533	JJ 2	
			IN 1	
			DT 25820	
			NNP 2	

Naive Bayes

- Treat each prediction as independent of the others

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{\sum_{y' \in \mathcal{Y}} P(y')P(x \mid y')}$$

$$P(\text{VBZ} \mid \text{flies}) = \frac{P(\text{VBZ})P(\text{flies} \mid \text{VBZ})}{\sum_{y' \in \mathcal{Y}} P(y')P(\text{flies} \mid y')}$$

Naive Bayes

- Treat each prediction as independent of the others

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{\sum_{y' \in \mathcal{Y}} P(y')P(x \mid y')}$$

$$P(\text{VBZ} \mid \text{flies}) = \frac{P(\text{VBZ})P(\text{flies} \mid \text{VBZ})}{\sum_{y' \in \mathcal{Y}} P(y')P(\text{flies} \mid y')}$$

Reminder: how do we learn $P(y)$ and $P(x|y)$ from training data?

Logistic regression

- Treat each prediction as independent of the others but condition on much more expressive set of features

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

$$P(\text{VBZ} \mid \text{flies}) = \frac{\exp(x^\top \beta_{\text{VBZ}})}{\sum_{y' \in \mathcal{Y}} \exp(x^\top \beta_{y'})}$$

Discriminative Features

Features are scoped over
entire observed input

Fruit **flies** like a banana

Discriminative Features

feature	example
Features are scoped over entire observed input	Fruit flies like a banana

Discriminative Features

Features are scoped over entire observed input

feature	example
$x_i = \text{flies}$	1

Fruit **flies** like a banana

Discriminative Features

Features are scoped over entire observed input

feature	example
$x_i = \text{flies}$	1
$x_i = \text{car}$	0

Fruit **flies** like a banana

Discriminative Features

Features are scoped over entire observed input

Fruit **flies** like a banana

feature	example
$x_i = \text{flies}$	1
$x_i = \text{car}$	0
$x_{i-1} = \text{fruit}$	1

Discriminative Features

Features are scoped over entire observed input

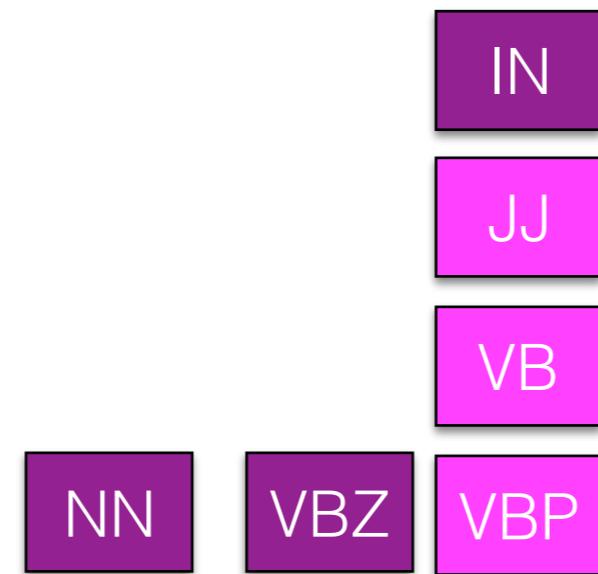
Fruit **flies** like a banana

feature	example
$x_i = \text{flies}$	1
$x_i = \text{car}$	0
$x_{i-1} = \text{fruit}$	1
$x_{i+1} = \text{like}$	1

Sequences

- Models that make independent predictions for elements in a sequence can reason over expressive representations of the **input** x (including correlations among inputs at different time steps x_i and x_j).
- But they don't capture another important source of information: correlations in the **labels** y .

Sequences



Time flies like an arrow

Sequences

Most common tag bigrams in
Penn Treebank training

DT	NN	41909
NNP	NNP	37696
NN	IN	35458
IN	DT	35006
JJ	NN	29699
DT	JJ	19166
NN	NN	17484
NN	,	16352
IN	NNP	15940
NN	.	15548
JJ	NNS	15297
NNS	IN	15146
TO	VB	13797
NNP	,	13683
IN	NN	11565

Sequences

x	time	flies	like	an	arrow
y	NN	VBZ	IN	DT	NN

$$P(\textcolor{magenta}{y} = \text{NN VBZ IN DT NN} \mid \textcolor{magenta}{x} = \text{time flies like an arrow})$$

Generative vs. Discriminative models

- Generative models specify a joint distribution over the labels and the data. With this you could **generate** new data

$$P(x, y) = P(y) P(x | y)$$

- Discriminative models specify the conditional distribution of the label y given the data x . These models focus on how to **discriminate** between the classes

$$P(y | x)$$

Generative

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{\sum_{y' \in \mathcal{Y}} P(x \mid y')P(y')}$$

$$P(y \mid x) \propto P(x \mid y)P(y)$$

$$\max_y P(x \mid y)P(y)$$

Generative

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{\sum_{y' \in \mathcal{Y}} P(x \mid y')P(y')}$$

$$P(y \mid x) \propto P(x \mid y)P(y)$$

$$\max_y P(x \mid y)P(y)$$

How do we parameterize these probabilities when x and y are sequences?

Hidden Markov Model (HMM)

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 (HMM)

$$\begin{aligned} P(y_i, \dots, y_n) &= P(y_1) \\ &\quad \times P(y_2 \mid y_1) \\ &\quad \times P(y_3 \mid y_1, y_2) \\ &\quad \dots \\ &\quad \times P(y_n \mid y_1, \dots, y_{n-1}) \end{aligned}$$

- Remember: a Markov assumption is an **approximation** to this **exact** decomposition (the chain rule of probability)

Hidden Markov Model (HMM)

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

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

- 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

NNP VBZ

is	1121
has	854
says	420
does	77
plans	50
expects	47
's	40
wants	31
owns	30
makes	29
hopes	24
remains	24
claims	19
seems	19
estimates	17

NN VBZ

is	2893
has	1004
does	128
says	109
remains	56
's	51
includes	44
continues	43
makes	40
seems	34
comes	33
reflects	31
calls	30
expects	29
goes	27

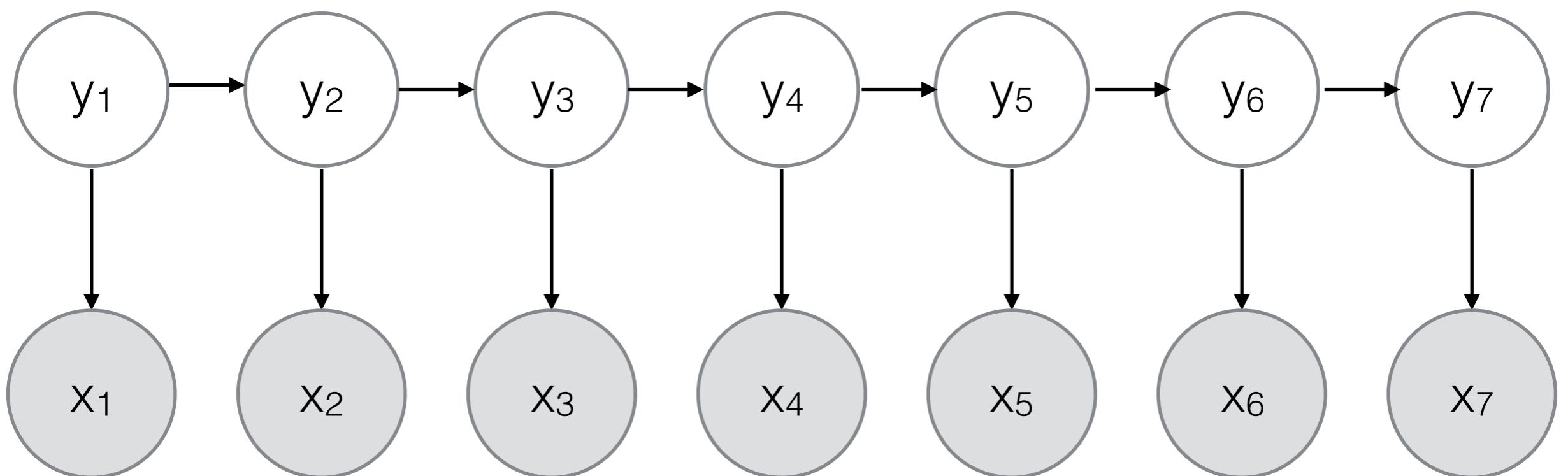
$$P(x_i \mid y_i, y_{i-1})$$

HMM

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

HMM

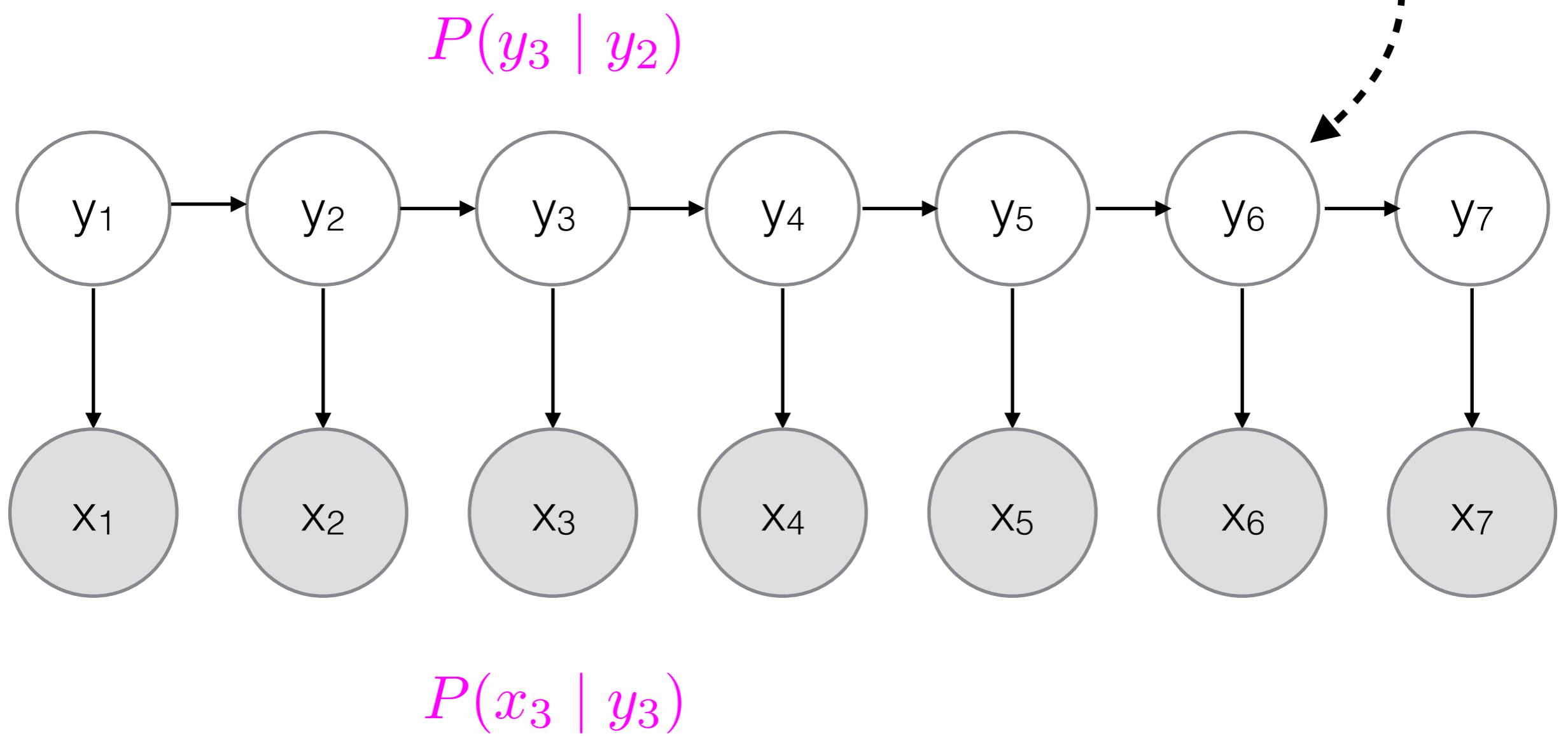
$$P(y_3 \mid y_2)$$



$$P(x_3 \mid y_3)$$

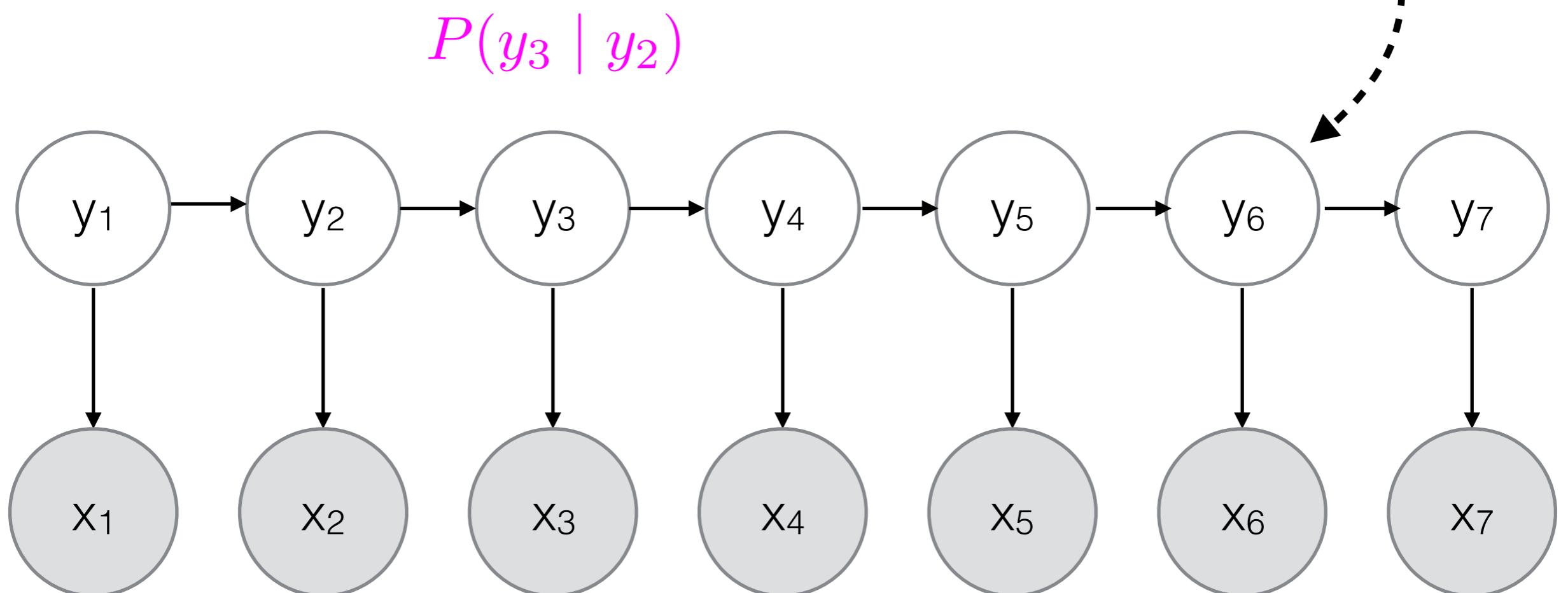
HMM

Uncolored circles are things we don't observe, i.e., things we have to *infer*



HMM

Uncolored circles are things we don't observe, i.e., things we have to *infer*



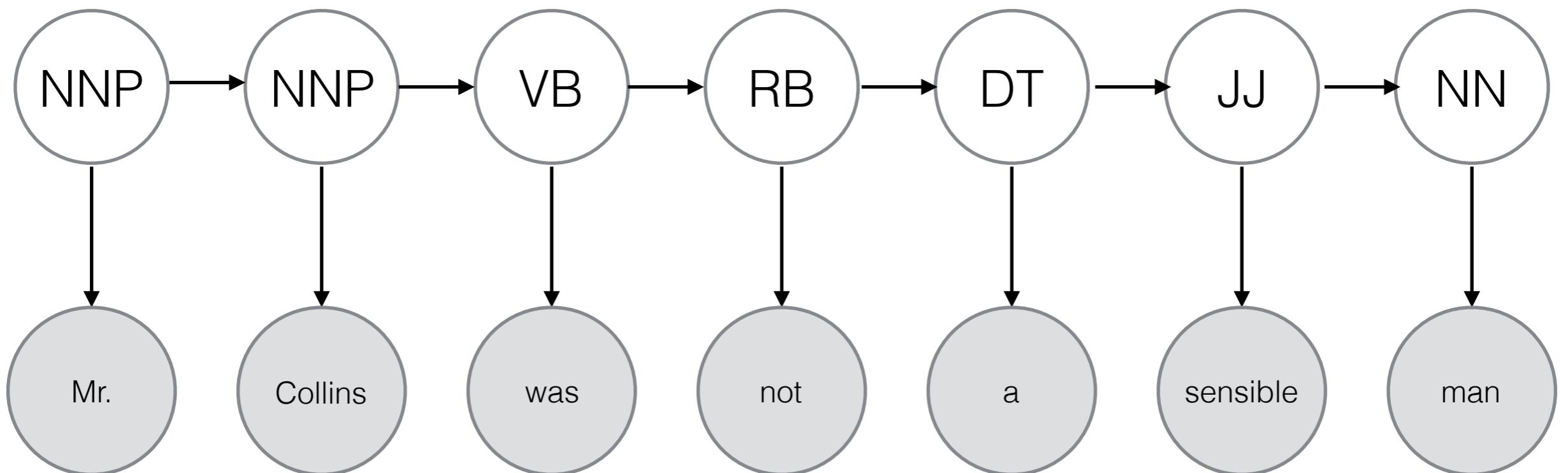
$$P(y_3 \mid y_2)$$

$$P(x_3 \mid y_3)$$

Shaded circles are things that we observe

HMM

$$P(VB \mid NNP)$$



$$P(was \mid VB)$$

Parameter estimation

$$P(y_t \mid y_{t-1})$$

$$\frac{c(y_1, y_2)}{c(y_1)}$$

MLE for both is just counting
(as in Naive Bayes)

$$P(x_t \mid y_t)$$

$$\frac{c(x, y)}{c(y)}$$

Transition probabilities

	NNP	MD	VB	JJ	NN	RB	DT
<i>< s ></i>	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Figure 10.5 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

Emission probabilities

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0.000097	0
NN	0	0.000200	0.000223	0.000006	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Figure 10.6 Observation likelihoods B computed from the WSJ corpus without smoothing.

Smoothing

- One solution: add a little probability mass to every element.

maximum likelihood
estimate

$$P(x_i | y) = \frac{n_{i,y}}{n_y}$$

$n_{i,y}$ = count of word i in class y
 n_y = number of words in y
 V = size of vocabulary

smoothed estimates

$$P(x_i | y) = \frac{n_{i,y} + a}{n_y + Va}$$

same a for all x_i

$$P(x_i | y) = \frac{n_{i,y} + a_i}{n_y + \sum_{j=1}^V a_j}$$

possibly different a for each x_i

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

NN

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

NN

VB

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

NN VB IN

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

NN VB IN DT

Decoding

- Greedy: proceed left to right, committing to the best tag for each time step (given the sequence seen so far)

Fruit flies like a banana

NN VB IN DT NN

Decoding

Decoding

The

Decoding

The horse

Decoding

The horse raced

Decoding

The horse raced past

Decoding

The horse raced past the

Decoding

The horse raced past the barn

Decoding

The horse raced past the barn fell

Decoding

DT

The horse raced past the barn fell

Decoding

DT

NN

The horse raced past the barn fell

Decoding

DT

NN

VBD

The horse raced past the barn fell

Decoding

DT

NN

VBD

IN

The horse raced past the barn fell

Decoding

DT

NN

VBD

IN

DT

The horse raced past the barn fell

Decoding

DT

NN

VBD

IN

DT

NN

The horse raced past the barn fell

Decoding

DT NN VBD IN DT NN ???

The horse raced past the barn fell

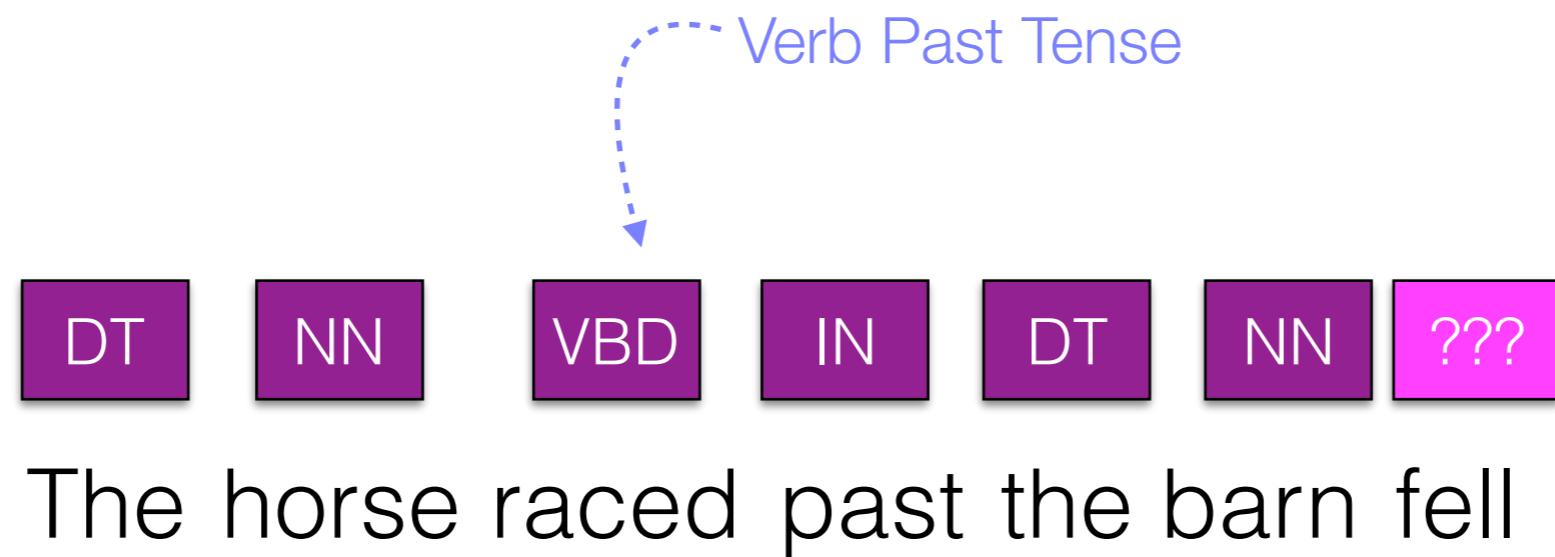
Decoding

DT NN VBD IN DT NN ???

The horse raced past the barn fell

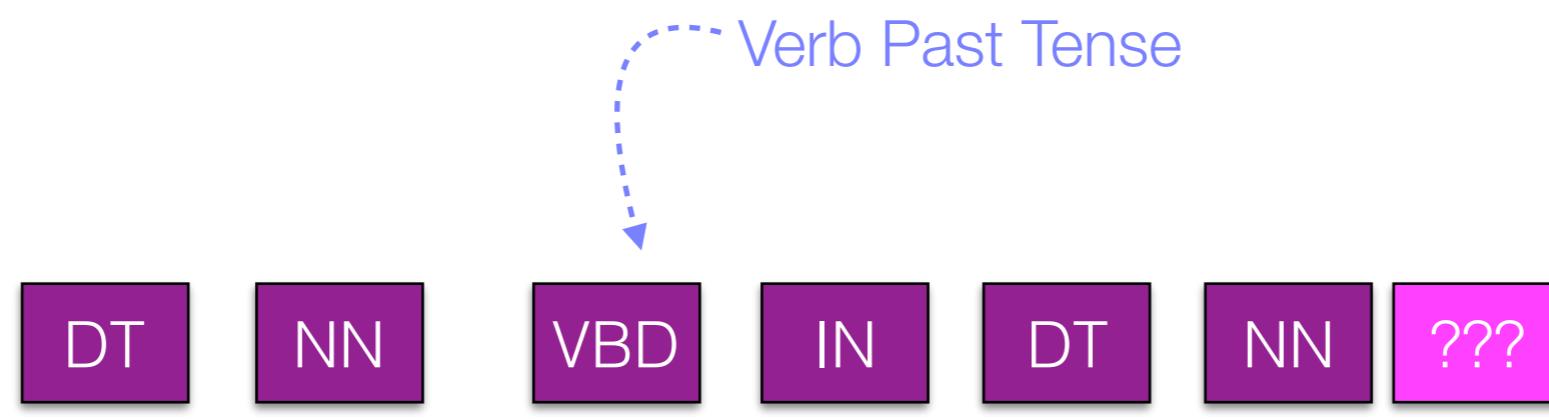
Information later on in the sentence can influence the best tags earlier on.

Decoding



Information later on in the sentence can influence the best tags earlier on.

Decoding



The horse raced past the barn fell



Information later on in the sentence can influence the best tags earlier on.

All paths

END							
DT							
NNP							
VB							
NN							
MD							
START							

^ Janet will back the bill \$

Ideally, what we want is to calculate the joint probability of **each path** and pick the one with the highest probability. But for N time steps and K labels, number of possible paths = K^N

END							
DT							
VBZ							
...							
NN							
MD							
START							

^ Janet will back the bill \$

END							
DT							
VBZ							
...							
NN							
MD							
START							

^ Janet will back the bill \$

5 word sentence with 45 Penn Treebank tags

END							
DT							
VBZ							
...							
NN							
MD							
START							

^ Janet will back the bill \$

5 word sentence with 45 Penn Treebank tags

$$45^5 = 184,528,125 \text{ different paths}$$

END							
DT							
VBZ							
...							
NN							
MD							
START							

^ Janet will back the bill \$

5 word sentence with 45 Penn Treebank tags

$$45^5 = 184,528,125 \text{ different paths}$$

$$45^{20} = 1.16\text{e}33 \text{ different paths}$$



Image credit: Claudio Caravano https://commons.wikimedia.org/wiki/File:Palazzo_dei_Papi_Viterbo.jpg

Viterbi algorithm

- Basic idea: if an optimal path through a sequence uses **label L at time T**, then it must have used an optimal path to get to label L at time T
- We can discard all non-optimal paths up to label L at time T

END							
DT							
NNP							
VB							
NN							
MD							
START							

^ Janet will back the bill \$

- At each time step t ending in label K, we find the max probability of any path that led to that state

END		
DT		$v_1(DT)$
NNP		$v_1(NNP)$
VB		$v_1(VB)$
NN		$v_1(NN)$
MD		$v_1(MD)$
START		

Janet

What's the HMM probability of ending in Janet = NNP?

$$P(y_t \mid y_{t-1})P(x_t \mid y_t)$$

$$P(\text{NNP} \mid \text{START})P(\text{Janet} \mid \text{NNP})$$

END		
DT		$v_1(DT)$
NNP		$v_1(NNP)$
VB		$v_1(VB)$
NN		$v_1(NN)$
MD		$v_1(MD)$
START		

Best path through time step 1
ending in tag y (trivially, best
path for all is just START)

Janet

$$v_1(y) = \max_{u \in \mathcal{Y}} [P(y_t = y \mid y_{t-1} = u) P(x_t \mid y_t = y)]$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Janet will

What's the **max** HMM probability of ending in will = MD?

First, what's the HMM probability of a single path
ending in will = MD?

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Janet will

$$P(y_1 \mid START)P(x_1 \mid y_1) \times P(y_2 = \text{MD} \mid y_1)P(x_2 \mid y_2 = \text{MD})$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Best path through time step 2
ending in tag MD

Janet will

$$P(DT \mid START) \times P(Janet \mid DT) \times P(y_t = MD \mid P(y_{t-1} = DT) \times P(will \mid y_t = MD))$$

$$P(NNP \mid START) \times P(Janet \mid NNP) \times P(y_t = MD \mid P(y_{t-1} = NNP) \times P(will \mid y_t = MD))$$

$$P(VB \mid START) \times P(Janet \mid VB) \times P(y_t = MD \mid P(y_{t-1} = VB) \times P(will \mid y_t = MD))$$

$$P(NN \mid START) \times P(Janet \mid NN) \times P(y_t = MD \mid P(y_{t-1} = NN) \times P(will \mid y_t = MD))$$

$$P(MD \mid START) \times P(Janet \mid MD) \times P(y_t = MD \mid P(y_{t-1} = MD) \times P(will \mid y_t = MD))$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Best path through time step 2
ending in tag MD

Janet will

Let's say the best path ending $\text{will} = \text{MD}$ includes $\text{Janet} = \text{NNP}$. By definition, every other path has lower probability.

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Best path through time step 2
ending in tag MD

Janet will

$$P(DT \mid START) \times P(Janet \mid DT) \times P(y_t = MD \mid P(y_{t-1} = DT) \times P(will \mid y_t = MD))$$

$$P(NNP \mid START) \times P(Janet \mid NNP) \times P(y_t = MD \mid P(y_{t-1} = NNP) \times P(will \mid y_t = MD))$$

$$P(VB \mid START) \times P(Janet \mid VB) \times P(y_t = MD \mid P(y_{t-1} = VB) \times P(will \mid y_t = MD))$$

$$P(NN \mid START) \times P(Janet \mid NN) \times P(y_t = MD \mid P(y_{t-1} = NN) \times P(will \mid y_t = MD))$$

$$P(MD \mid START) \times P(Janet \mid MD) \times P(y_t = MD \mid P(y_{t-1} = MD) \times P(will \mid y_t = MD))$$

$$v_1(y) = \max_{\textcolor{magenta}{u} \in \mathcal{Y}} [P(y_t = y \mid \textcolor{magenta}{y_{t-1}} = u) P(x_t \mid y_t = y)]$$

$$P(\text{DT} \mid \text{START}) \times P(\textit{Janet} \mid \text{DT}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{DT}) \times P(\textit{will} \mid y_t = \text{MD})$$

$$P(\text{NNP} \mid \text{START}) \times P(\textit{Janet} \mid \text{NNP}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{NNP}) \times P(\textit{will} \mid y_t = \text{MD})$$

$$P(\text{VB} \mid \text{START}) \times P(\textit{Janet} \mid \text{VB}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{VB}) \times P(\textit{will} \mid y_t = \text{MD})$$

$$P(\text{NN} \mid \text{START}) \times P(\textit{Janet} \mid \text{NN}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{NN}) \times P(\textit{will} \mid y_t = \text{MD})$$

$$P(\text{MD} \mid \text{START}) \times P(\textit{Janet} \mid \text{MD}) \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{MD}) \times P(\textit{will} \mid y_t = \text{MD})$$

$$\textcolor{magenta}{v_1(DT)} \times P(y_t = \text{MD} \mid P(y_{t-1} = \text{DT}) \times P(\textit{will} \mid y_t = \text{MD})$$

END			
DT		$v_1(DT)$	$v_2(DT)$
NNP		$v_1(NNP)$	$v_2(NNP)$
VB		$v_1(VB)$	$v_2(VB)$
NN		$v_1(NN)$	$v_2(NN)$
MD		$v_1(MD)$	$v_2(MD)$
START			

Janet will

$$v_t(y) = \max_{\textcolor{magenta}{u} \in \mathcal{Y}} [v_{t-1}(\textcolor{magenta}{u}) \times P(y_t = y \mid \textcolor{magenta}{y_{t-1}} = \textcolor{magenta}{u}) P(x_t \mid y_t = y)]$$

END				
DT		v ₁ (DT)	v ₂ (DT)	v ₃ (DT)
NNP		v ₁ (NNP)	v ₂ (NNP)	v ₃ (NNP)
VB		v ₁ (VB)	v ₂ (VB)	v ₃ (VB)
NN		v ₁ (NN)	v ₂ (NN)	v ₃ (NN)
MD		v ₁ (MD)	v ₂ (MD)	v ₃ (MD)
START				

Janet will back

25 paths ending in back = VB

END				
DT		v ₁ (DT)	v ₂ (DT)	v ₃ (DT)
NNP		v ₁ (NNP)	v ₂ (NNP)	v ₃ (NNP)
VB		v ₁ (VB)	v ₂ (VB)	v ₃ (VB)
NN		v ₁ (NN)	v ₂ (NN)	v ₃ (NN)
MD		v ₁ (MD)	v ₂ (MD)	v ₃ (MD)
START				

Janet will back

Let's say the best path ending in **back = VB** includes
will = MD.

END				
DT		v ₁ (DT)	v ₂ (DT)	v ₃ (DT)
NNP		v ₁ (NNP)	v ₂ (NNP)	v ₃ (NNP)
VB		v ₁ (VB)	v ₂ (VB)	v ₃ (VB)
NN		v ₁ (NN)	v ₂ (NN)	v ₃ (NN)
MD		v ₁ (MD)	v ₂ (MD)	v ₃ (MD)
START				

Janet will back

If the best path ending in **will = MD** includes Janet=NNP, we can forget all paths with Janet != NNP for any path including **will = MD** because we know they are less likely.

END					
DT		v ₁ (DT)	v ₂ (DT)	v ₃ (DT)	v ₄ (DT)
NNP		v ₁ (NNP)	v ₂ (NNP)	v ₃ (NNP)	v ₄ (NNP)
VB		v ₁ (VB)	v ₂ (VB)	v ₃ (VB)	v ₄ (MD)
NN		v ₁ (NN)	v ₂ (NN)	v ₃ (NN)	v ₄ (NN)
MD		v ₁ (MD)	v ₂ (MD)	v ₃ (MD)	v ₄ (MD)
START					

Janet will back the

125 possible paths ending in the = DT, but we only need to consider 5 (best path ending in back = DT, back = NNP, back = VB, back = NN, back = MD)

END						
DT		v ₁ (DT)	v ₂ (DT)	v ₃ (DT)	v ₄ (DT)	v ₅ (DT)
NNP		v ₁ (NNP)	v ₂ (NNP)	v ₃ (NNP)	v ₄ (NNP)	v ₅ (NNP)
VB		v ₁ (VB)	v ₂ (VB)	v ₃ (VB)	v ₄ (MD)	v ₅ (MD)
NN		v ₁ (NN)	v ₂ (NN)	v ₃ (NN)	v ₄ (NN)	v ₅ (NN)
MD		v ₁ (MD)	v ₂ (MD)	v ₃ (MD)	v ₄ (MD)	v ₅ (MD)
START						

Janet will back the bill

END							$v_T(\text{END})$
DT		$v_1(\text{DT})$	$v_2(\text{DT})$	$v_3(\text{DT})$	$v_4(\text{DT})$	$v_5(\text{DT})$	
NNP		$v_1(\text{NNP})$	$v_2(\text{NNP})$	$v_3(\text{NNP})$	$v_4(\text{NNP})$	$v_5(\text{NNP})$	
VB		$v_1(\text{VB})$	$v_2(\text{VB})$	$v_3(\text{VB})$	$v_4(\text{MD})$	$v_5(\text{MD})$	
NN		$v_1(\text{NN})$	$v_2(\text{NN})$	$v_3(\text{NN})$	$v_4(\text{NN})$	$v_5(\text{NN})$	
MD		$v_1(\text{MD})$	$v_2(\text{MD})$	$v_3(\text{MD})$	$v_4(\text{MD})$	$v_5(\text{MD})$	
START							

Janet will back the bill

$v_T(\text{END})$ encodes the best path through the entire sequence

END								$v_T(\text{END})$
DT								
NNP								
VB								
NN								
MD								
START								

Janet will back the bill

For each timestep t + label, keep track of the max element from $t-1$ to reconstruct best path

function VITERBI(*observations* of len T ,*state-graph* of len N) **returns** *best-path*

create a path probability matrix $viterbi[N+2, T]$

for each state s **from** 1 **to** N **do** ; initialization step

$viterbi[s, 1] \leftarrow a_{0,s} * b_s(o_1)$

$backpointer[s, 1] \leftarrow 0$

for each time step t **from** 2 **to** T **do** ; recursion step

for each state s **from** 1 **to** N **do**

$viterbi[s, t] \leftarrow \max_{s'=1}^N viterbi[s', t-1] * a_{s', s} * b_s(o_t)$

$backpointer[s, t] \leftarrow \operatorname{argmax}_{s'=1}^N viterbi[s', t-1] * a_{s', s}$

$viterbi[q_F, T] \leftarrow \max_{s=1}^N viterbi[s, T] * a_{s, q_F}$; termination step

$backpointer[q_F, T] \leftarrow \operatorname{argmax}_{s=1}^N viterbi[s, T] * a_{s, q_F}$; termination step

return the backtrace path by following backpointers to states back in time from $backpointer[q_F, T]$

Figure 10.8 Viterbi algorithm for finding optimal sequence of tags. Given an observation sequence and an HMM $\lambda = (A, B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence. Note that states 0 and q_F are non-emitting.

END		
DT		$v_1(DT)$
NNP		$v_1(NNP)$
VB		$v_1(VB)$
NN		$v_1(NN)$
MD		$v_1(MD)$
START		

Can Viterbi decoding help with independent predictions? (e.g., Naive Bayes or logreg)

Janet

$$v_1(y) = \max_{\textcolor{magenta}{u} \in \mathcal{Y}} [P(y_t = y \mid \textcolor{magenta}{y}_{t-1} = \textcolor{magenta}{u}) P(x_t \mid y_t = y)]$$

END		
DT		$v_1(DT)$
NNP		$v_1(NNP)$
VB		$v_1(VB)$
NN		$v_1(NN)$
MD		$v_1(MD)$
START		

Can Viterbi decoding help with independent predictions? (e.g., Naive Bayes or logreg)

Janet

$$v_1(y) = \max_{u \in \mathcal{Y}} [P(y_t = y \mid y_{t-1} = u) P(x_t \mid y_t = y)]$$

When making independent predictions:

$$P(y_t = y \mid y_{t-1} = u) = P(y_t = y)$$

Generative vs. Discriminative models

- Generative models specify a joint distribution over the labels and the data. With this you could **generate** new data

$$P(x, y) = P(y) P(x | y)$$

- Discriminative models specify the conditional distribution of the label y given the data x . These models focus on how to **discriminate** between the classes

$$P(y | x)$$

Maximum Entropy Markov Model (MEMM)

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

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

Maximum Entropy Markov Model (MEMM)

General maxent form

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

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

Maximum Entropy Markov Model (MEMM)

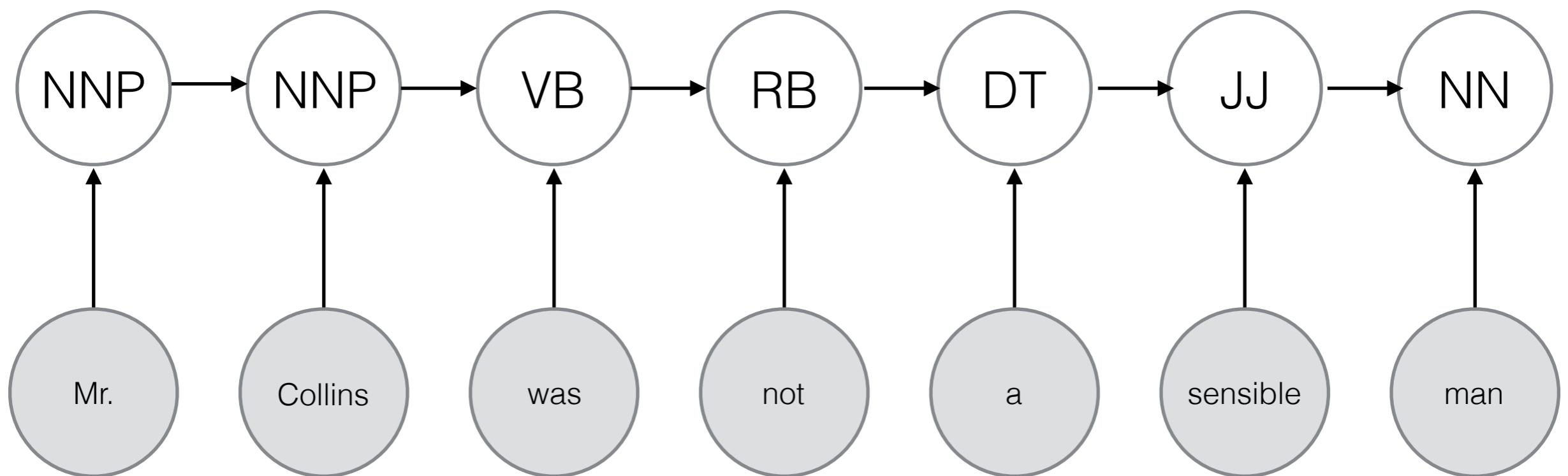
General maxent form

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

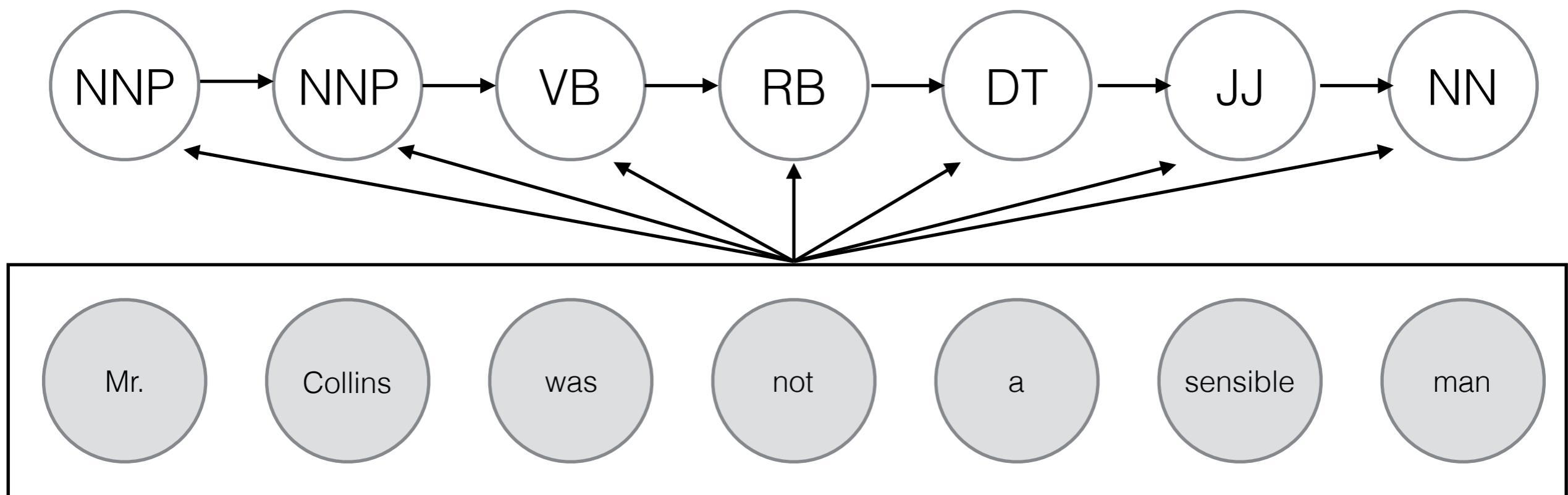
Maxent with first-order Markov assumption:
Maximum Entropy Markov Model

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

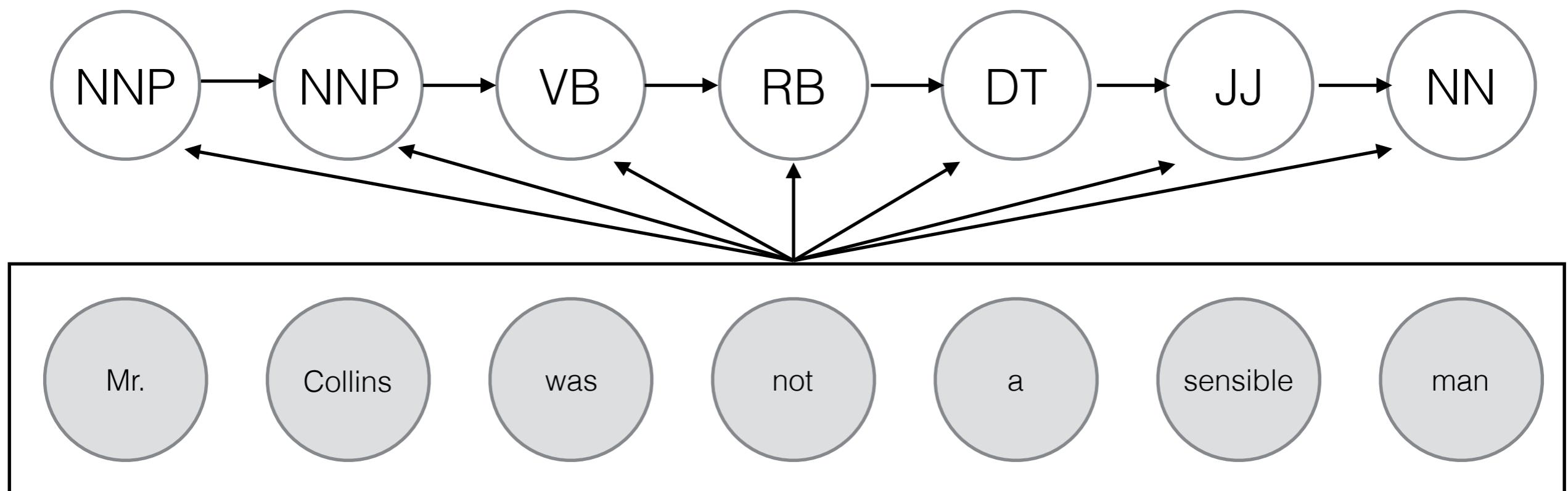
MEMM



MEMM

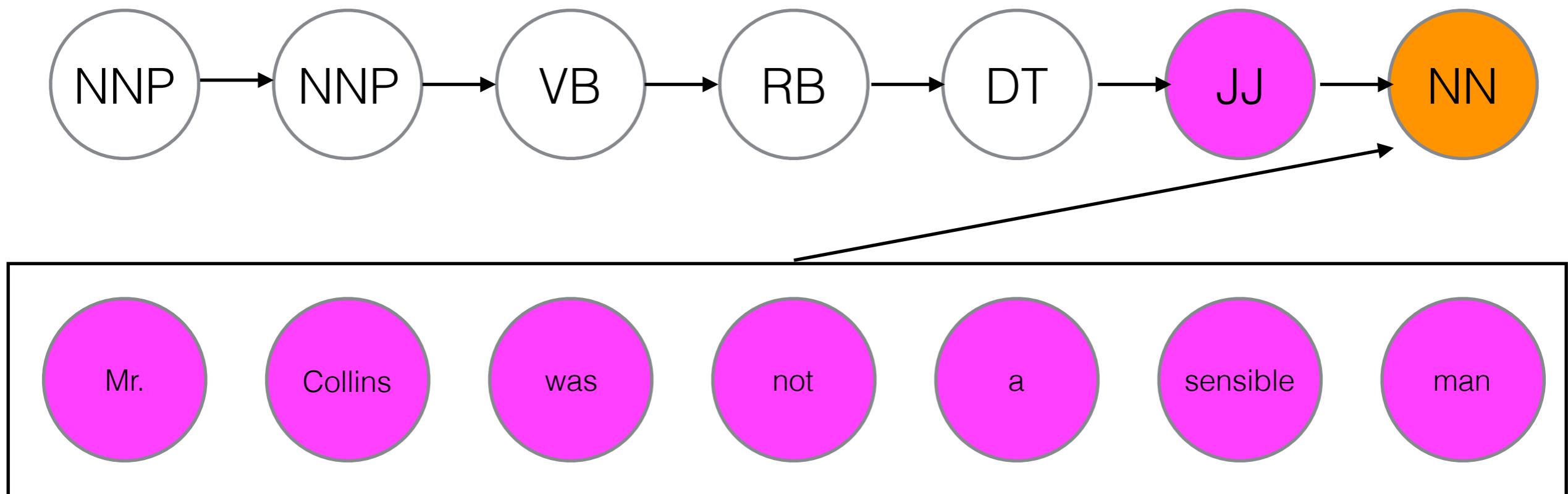


MEMM



MEMMs condition on the *entire* input

MEMM



Features

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

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

Features

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

feature	example
---------	---------

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

Features

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

feature	example
$x_i = \text{man}$	1

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

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

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

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
$x_i \text{ ends in } -ly$	0

Training

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

For all training data, we want probability of the true label y_i conditioned on the previous true label y_{i-1} to be high.

This is simply multiclass logistic regression

Decoding

- With logistic regression, our prediction is simply the argmax y :

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

- With an MEMM, we know the true y_{i-1} during training but we never of course know it at test time

$$P(y_i \mid \textcolor{magenta}{y_{i-1}}, x, \beta)$$

Greedy decoding

- At $i=1$, predict the argmax given START:

$$P(y_1 \mid \textcolor{magenta}{START}, x, \beta)$$

- For each subsequent time step, condition on the y just predicted during the step before

$$P(y_i \mid \textcolor{magenta}{y_{i-1}}, x, \beta)$$

Viterbi decoding

$$P(y)P(x \mid y) = P(x, y)$$

$$v_t(y) = \max_{\textcolor{magenta}{u} \in \mathcal{Y}} [v_{t-1}(\textcolor{magenta}{u}) \times P(y_t = y \mid \textcolor{magenta}{y}_{t-1} = \textcolor{magenta}{u})P(x_t \mid y_t = y)]$$

$$P(y \mid x)$$

$$v_t(y) = \max_{u \in \mathcal{Y}} [v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u, x, \beta)]$$

Viterbi decoding

Viterbi for HMM: max joint probability

$$P(y)P(x \mid y) = P(x, y)$$

$$v_t(y) = \max_{u \in \mathcal{Y}} [v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u)P(x_t \mid y_t = y)]$$

$$P(y \mid x)$$

$$v_t(y) = \max_{u \in \mathcal{Y}} [v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u, x, \beta)]$$

Viterbi decoding

Viterbi for HMM: max joint probability

$$P(y)P(x \mid y) = P(x, y)$$

$$v_t(y) = \max_{u \in \mathcal{Y}} [v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u)P(x_t \mid y_t = y)]$$

Viterbi for MEMM: max conditional probability

$$P(y \mid x)$$

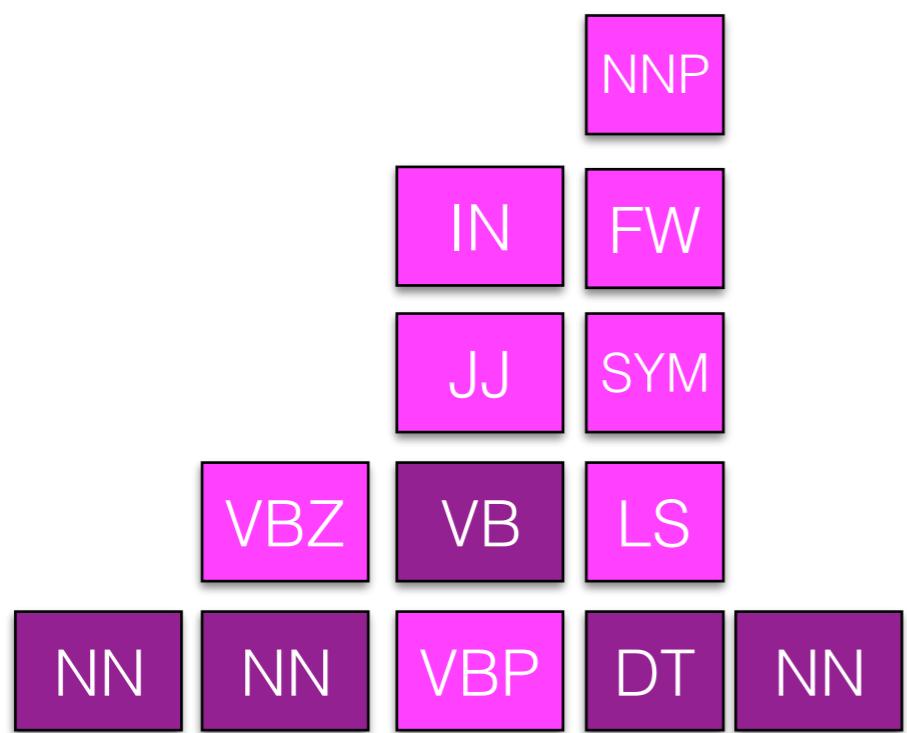
$$v_t(y) = \max_{u \in \mathcal{Y}} [v_{t-1}(u) \times P(y_t = y \mid y_{t-1} = u, x, \beta)]$$

Let's make a POS-tagger
with the Viterbi algorithm!

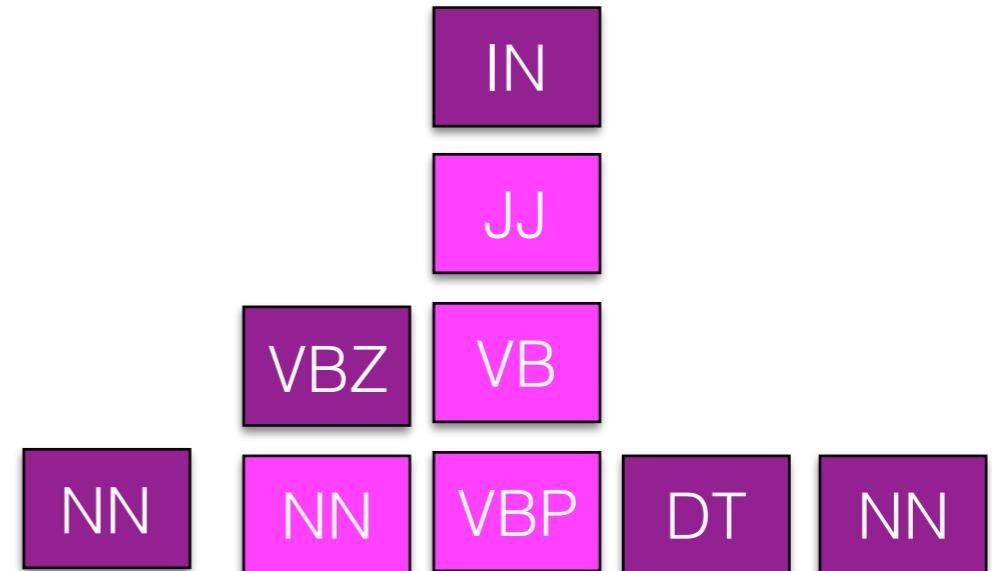


POS tagging

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!)

MEMM Training

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

For all training data, we want probability of the true label y_i conditioned on the previous true label y_{i-1} to be high.

This is simply multiclass logistic regression

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

Label bias

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

- For a given conditioning context, the probability of the tag (e.g., VBZ) only competes against other tags with that same context (e.g., NN)

Label bias

NN TO VB

will to fight

	NN	MD
$x_i = \text{will}$	10	40
$y_{i-1} = \text{START}$	-1	7
BIAS	7	-2

Modals show up much more frequently at the start of the sentence than nouns do (e.g., questions)

Label bias

NN TO VB

will to fight

But we know that MD + TO is very rare

- *can to eat
- *would to eat
- *could to eat
- *may to eat

Label bias

NN TO VB

will to fight

	TO
$x_i=to$	10000000
$y_{i-1}=NN$	0
$y_{i-1}=MD$	0

to is relatively deterministic
(almost always TO) so it doesn't
matter what tag precedes it.

Label bias

NN TO VB

will to fight

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

Because of this local normalization, $P(\text{TO} \mid \text{context})$ will always be 1 if $x = \text{"to"}$

Label bias

NN TO VB

will to fight

That means our prediction for *to* can't help us disambiguate will. We lose the information that MB + TO sequences rarely happen.

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

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

Conditional random fields

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

Feature vector scoped over the entire input and label sequence

$$\Phi(x, y) = \sum_{i=1}^n \phi(x, i, y_i, y_{i-1})$$

ϕ is the same feature vector we used for local predictions using MEMMs

Features

$$\phi(x,i,y_i,y_{i-1})$$

Features

$$\phi(x, i, y_i, y_{i-1})$$

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

Features

$\phi(x, i, y_i, y_{i-1})$

feature

example

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

Features

$$\phi(x, i, y_i, y_{i-1})$$

feature	example
$x_i = \text{man}$	1

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

Features

$$\phi(x, i, y_i, y_{i-1})$$

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

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

Features

$$\phi(x, i, y_i, y_{i-1})$$

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

feature	example
$x_i = \text{man}$	1
$y_{i-1} = \text{JJ}$	1
$i=n$ (last word of sentence)	1

Features

$$\phi(x, i, y_i, y_{i-1})$$

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

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

Conditional random fields

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

- In MEMMs, we normalize over the set of 45 POS tags
- CRFs are **globally normalized**, but the normalization complexity is huge — every possible sequence of labels of length n.

Forward algorithm (CRF)

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

- Calculating the denominator naively would involve a summation over K^N terms
- But we can do this efficiently in NK^2 time using the forward algorithm

For details, see: Collins, “The Forward-Backward Algorithm”

In practice

- CRF training is slow! NK^2 complexity for each sequence at each gradient step.
- Accuracy is typically better than MEMMs
- Most people now use Recurrent Neural Networks for tasks with lots of data



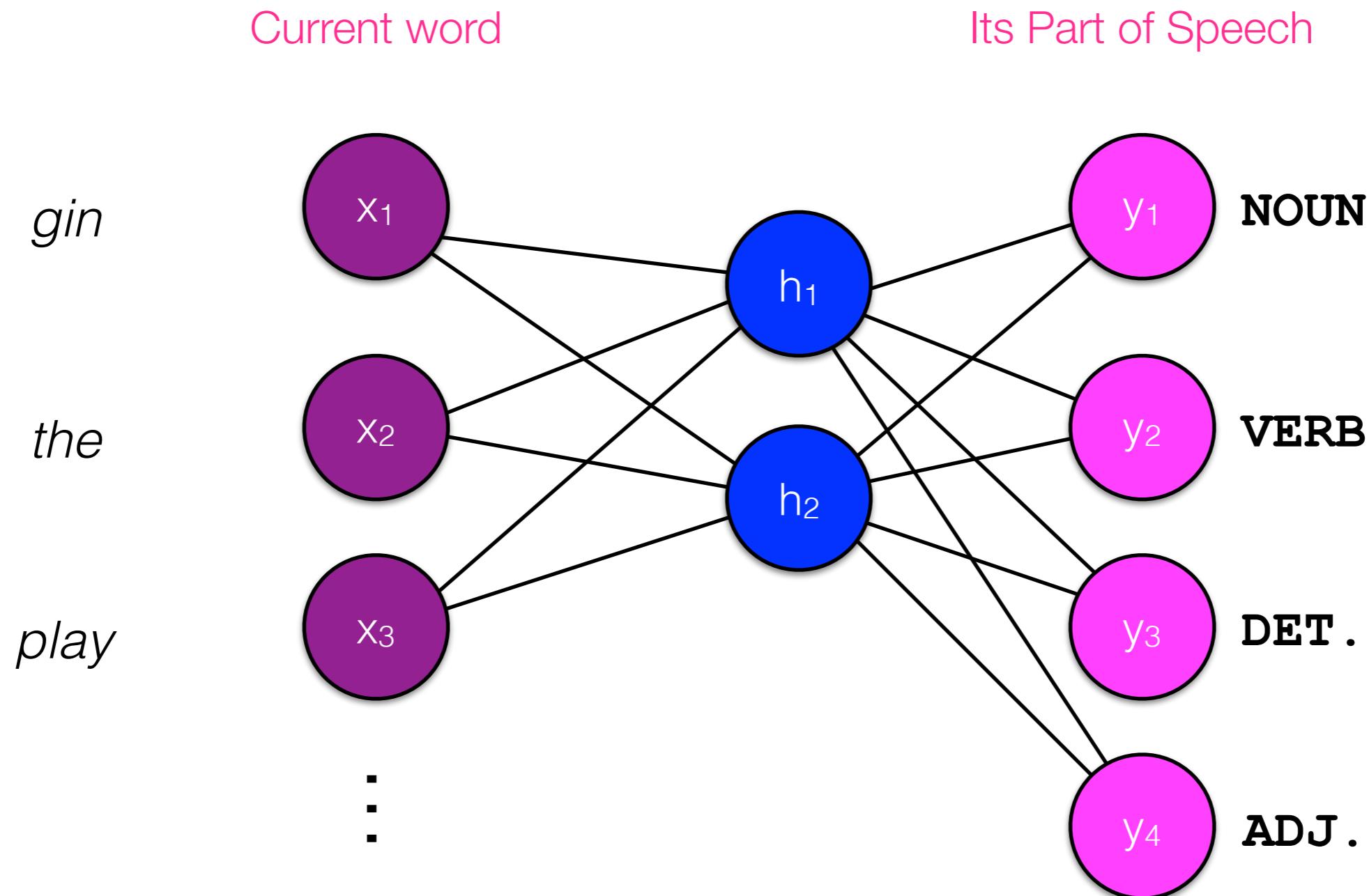
Recurrent Neural Nets

Recurrent neural networks

- RNN allow arbitrarily-sized conditioning contexts; condition on the **entire sequence history**.

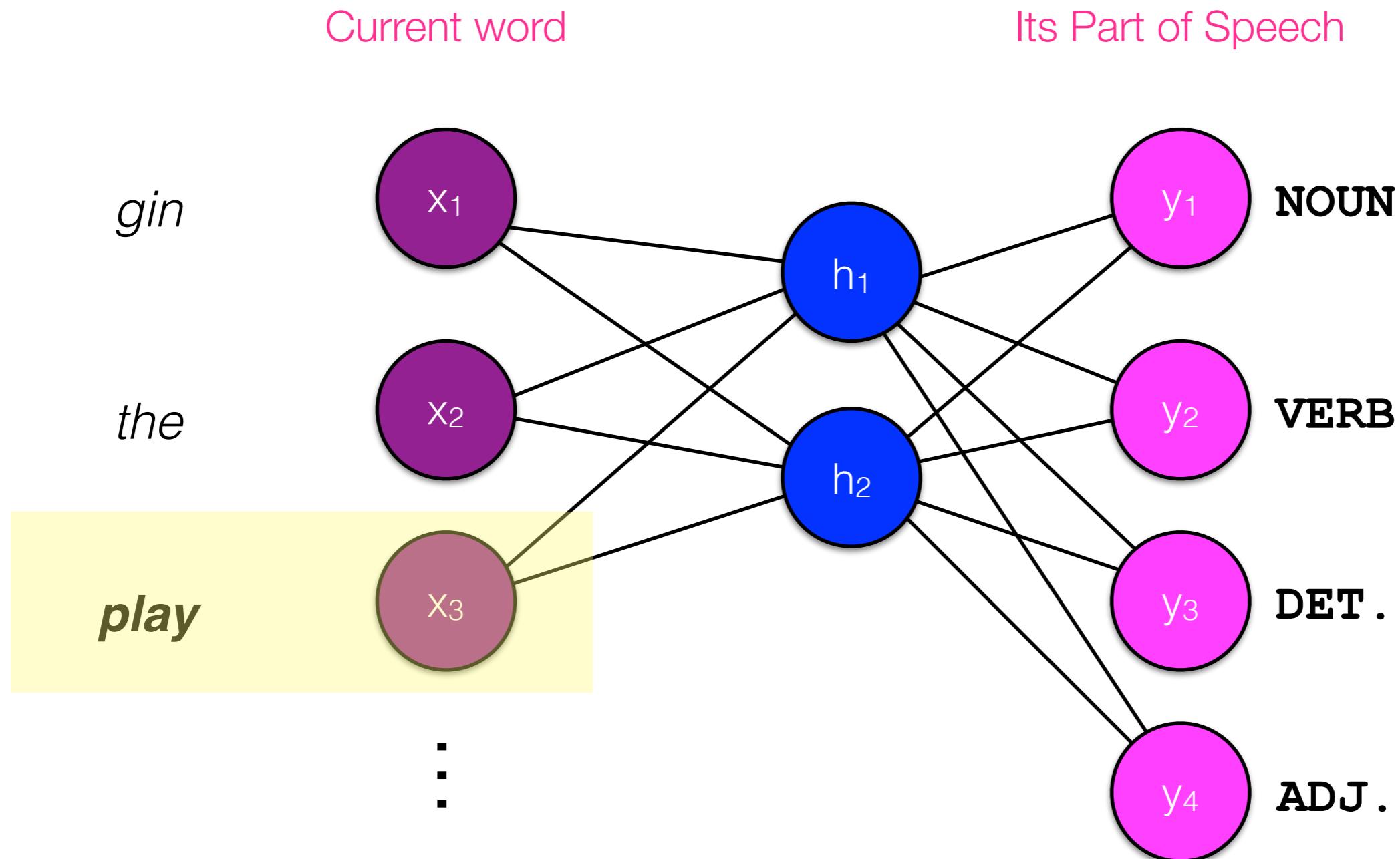
Motivating example: Feed-forward POS tagging

“The play took two hours”



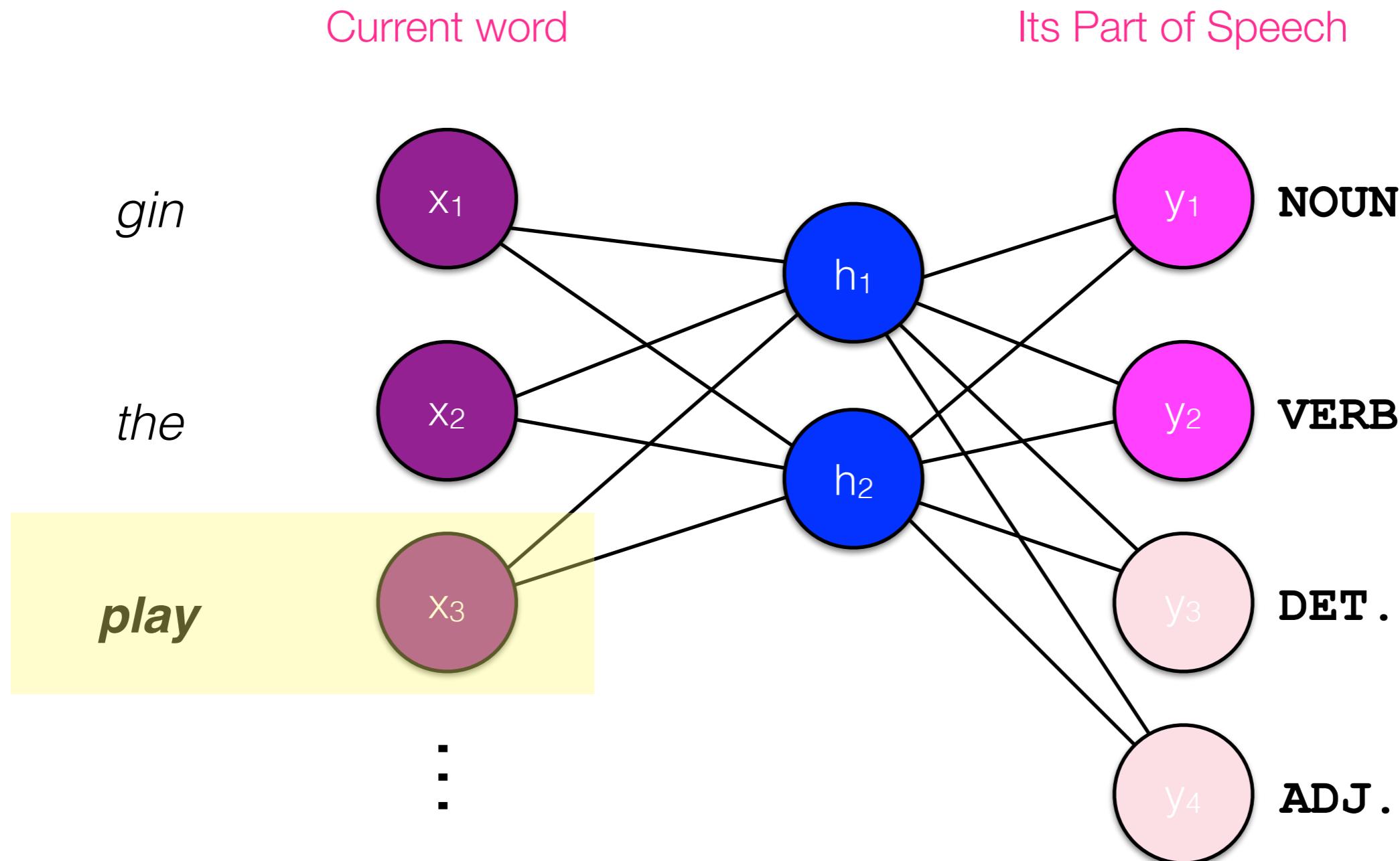
Motivating example: Feed-forward POS tagging

“The **play** took two hours”



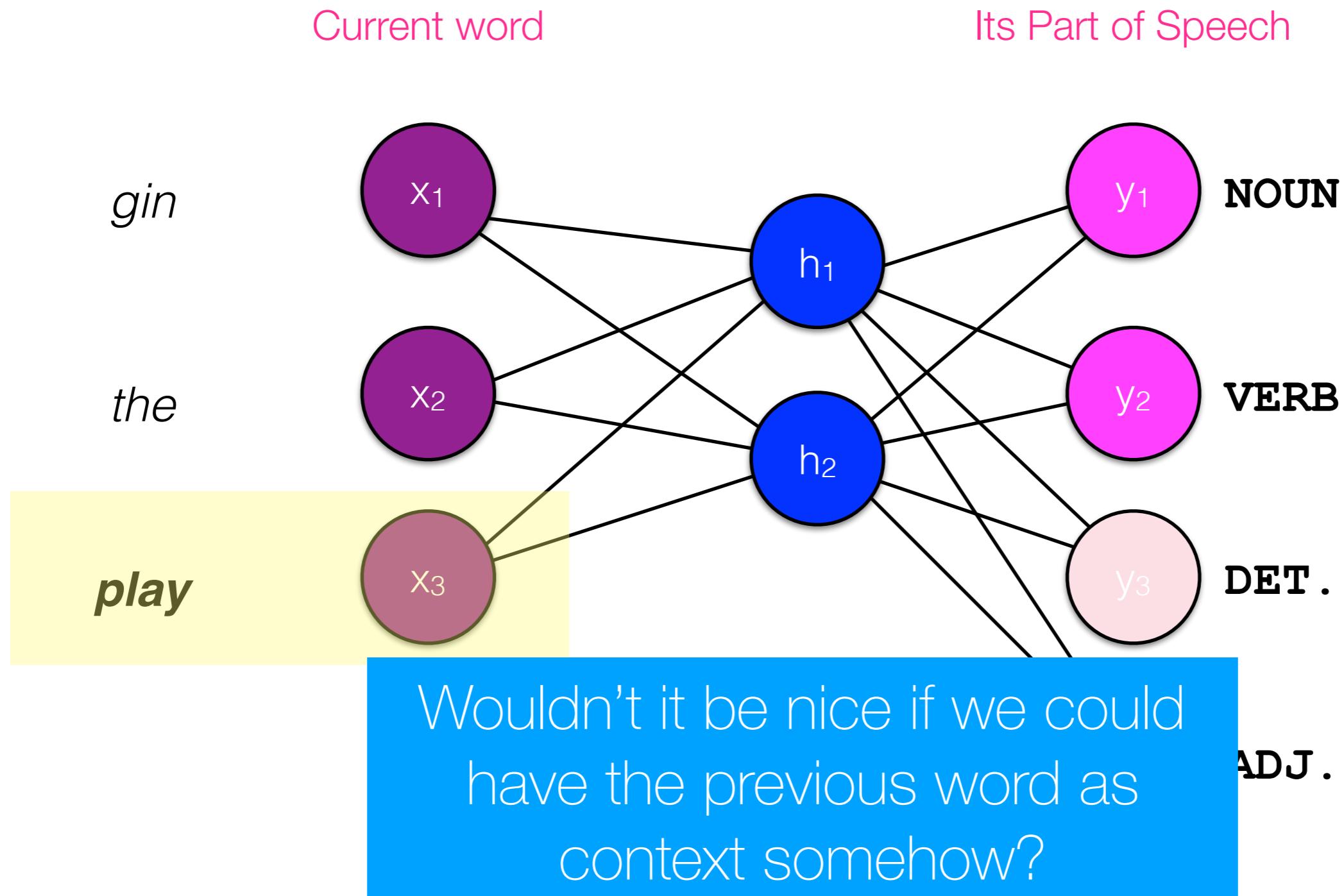
Motivating example: Feed-forward POS tagging

“The play took two hours”



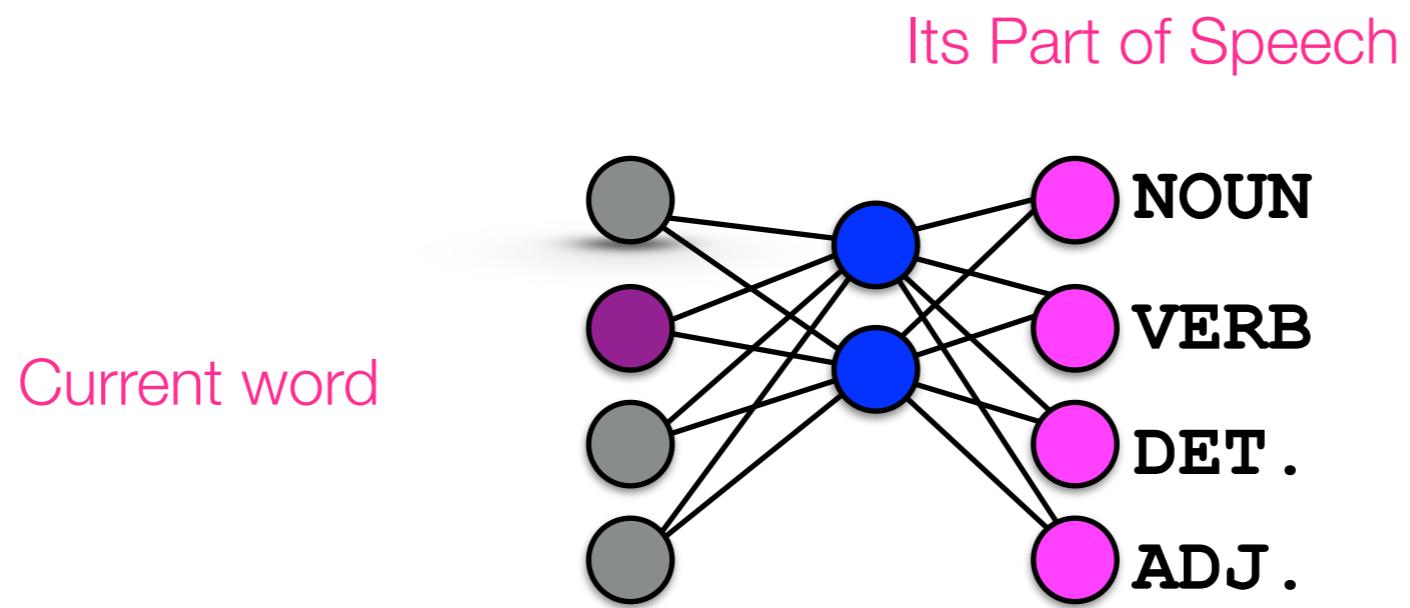
Motivating example: Feed-forward POS tagging

“The **play** took two hours”



Motivating example: Feed-forward POS tagging

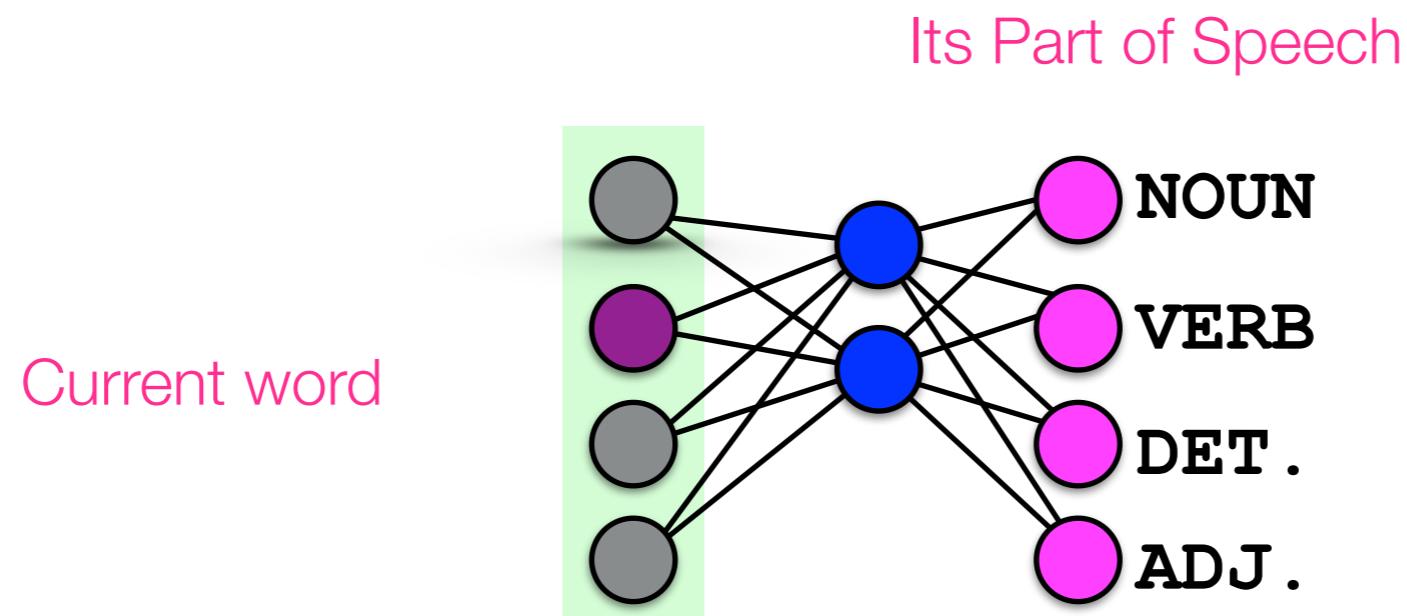
“The **play** took two hours”



Wouldn't it be nice if we could have the previous word as context somehow?

Motivating example: Feed-forward POS tagging

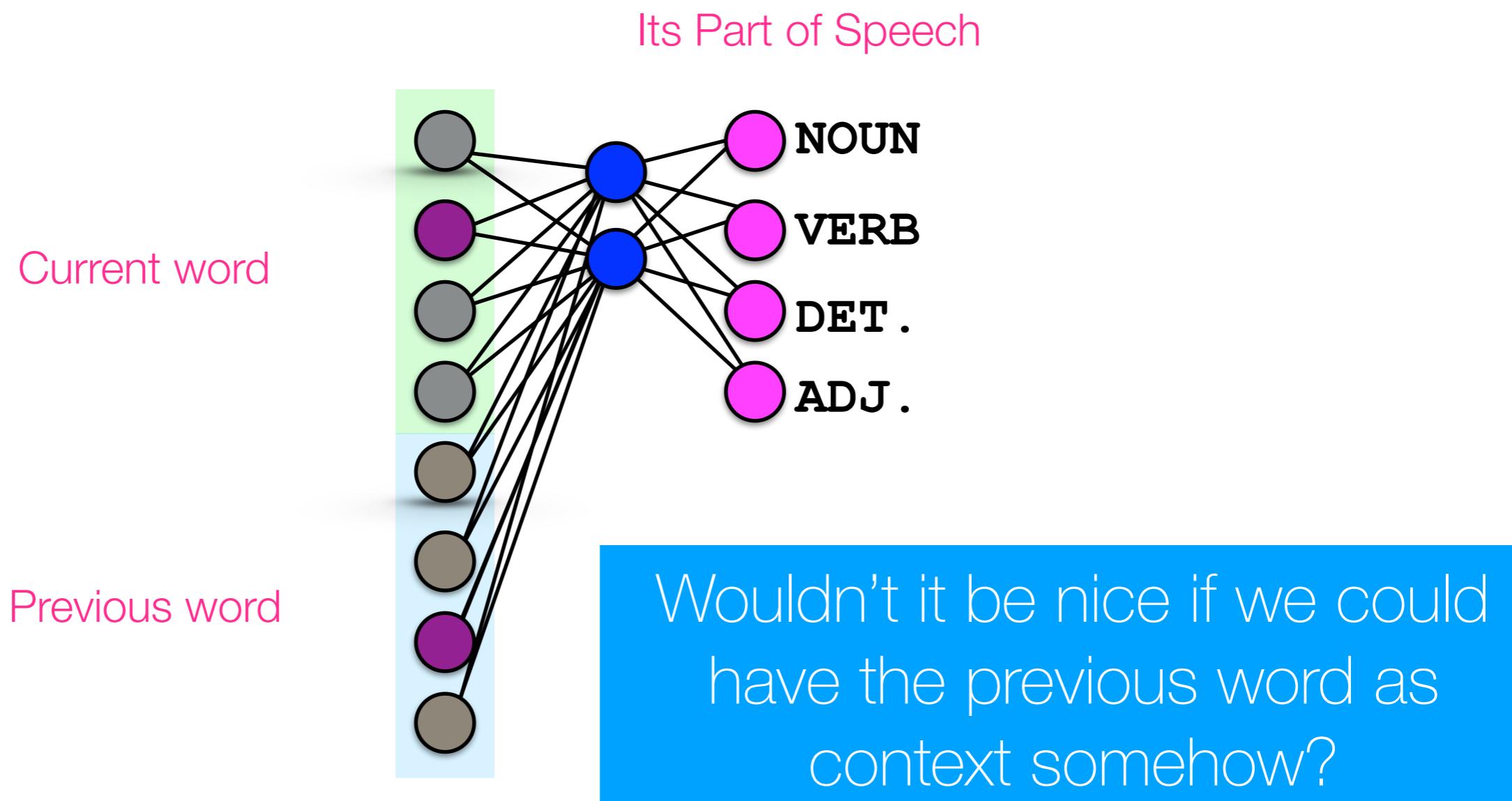
“The **play** took two hours”



Wouldn't it be nice if we could have the previous word as context somehow?

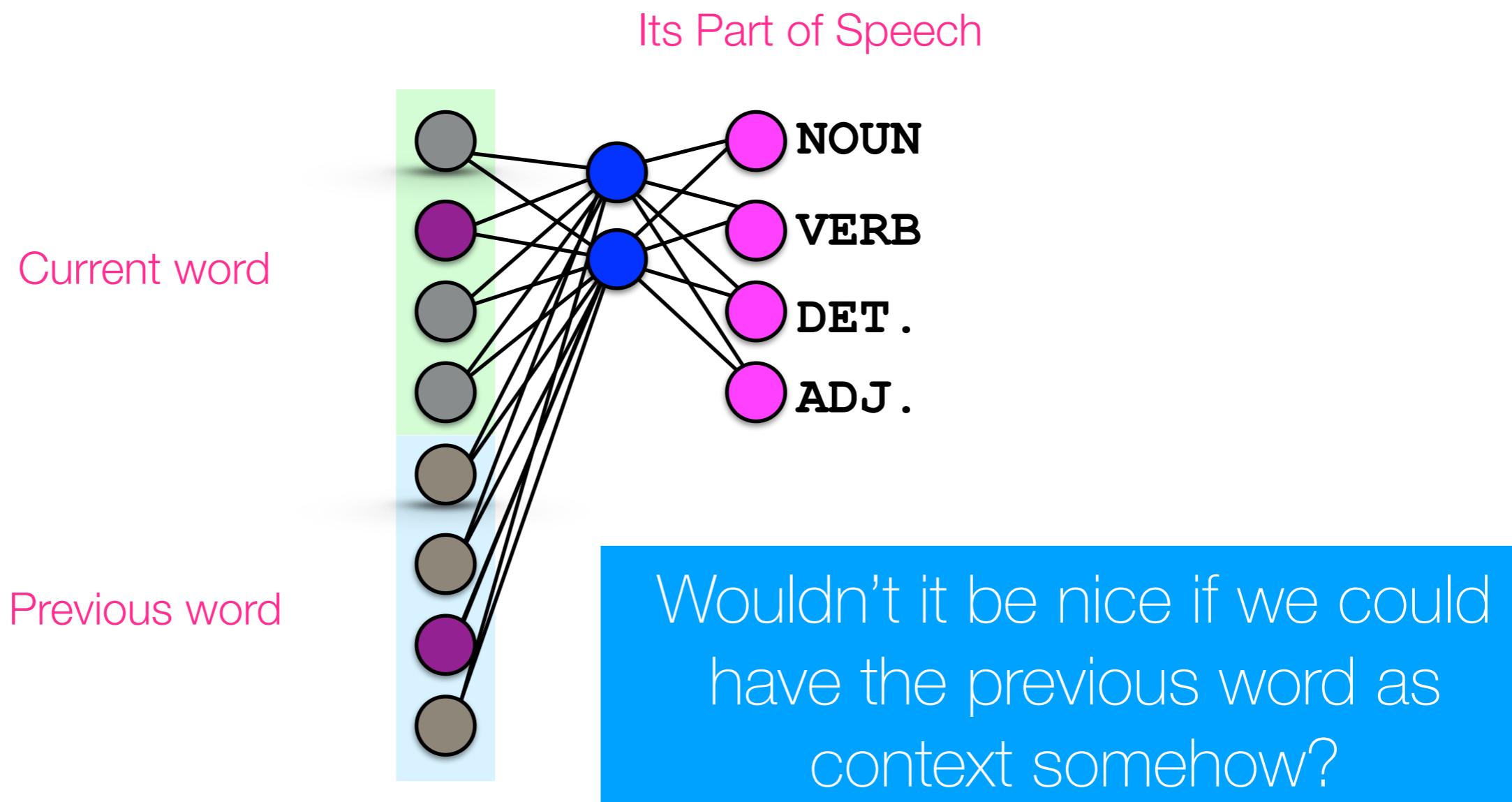
Motivating example: Feed-forward POS tagging

“The **play** took two hours”



Motivating example: Feed-forward POS tagging

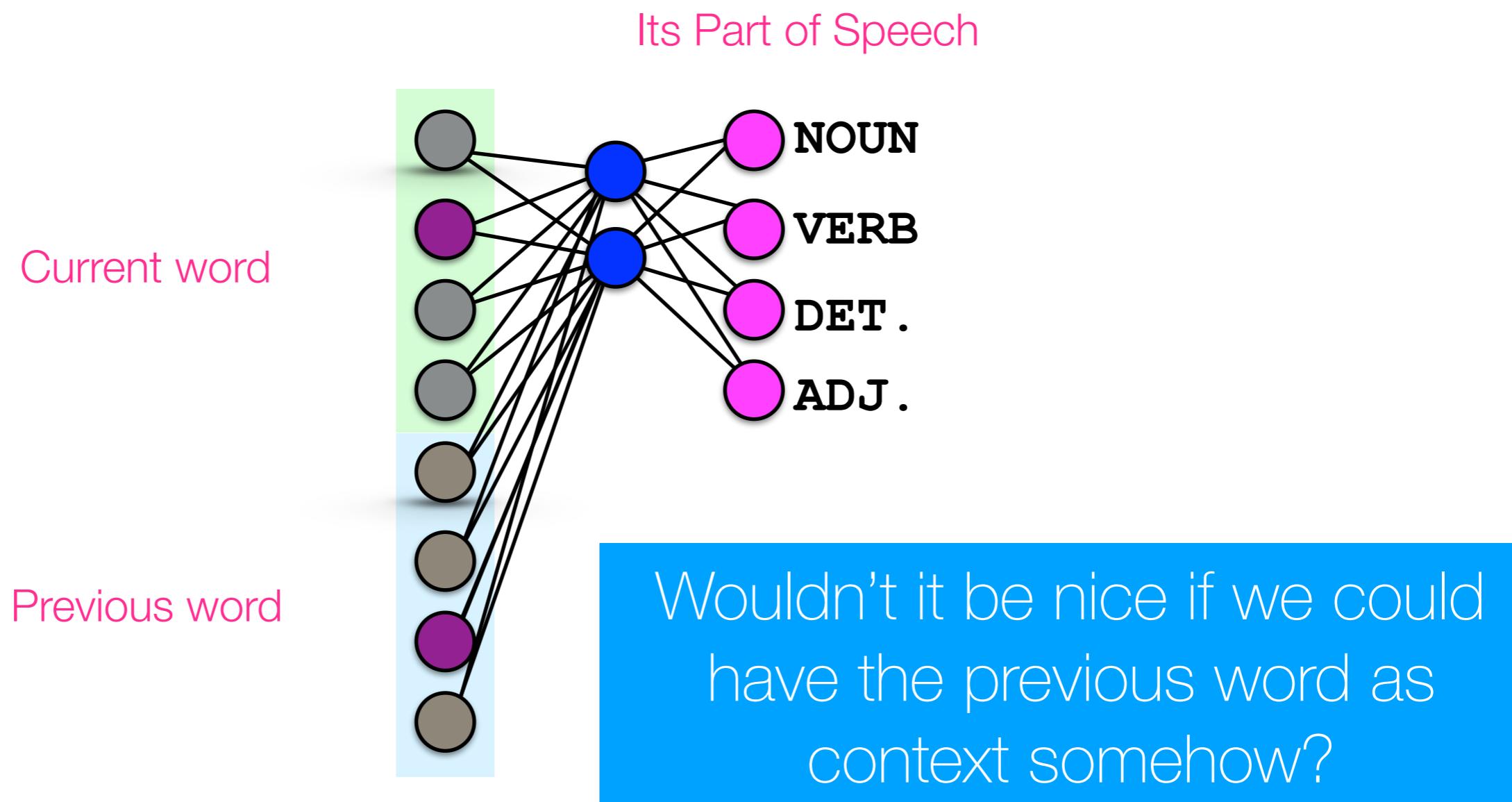
“The **play** took two hours”



Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

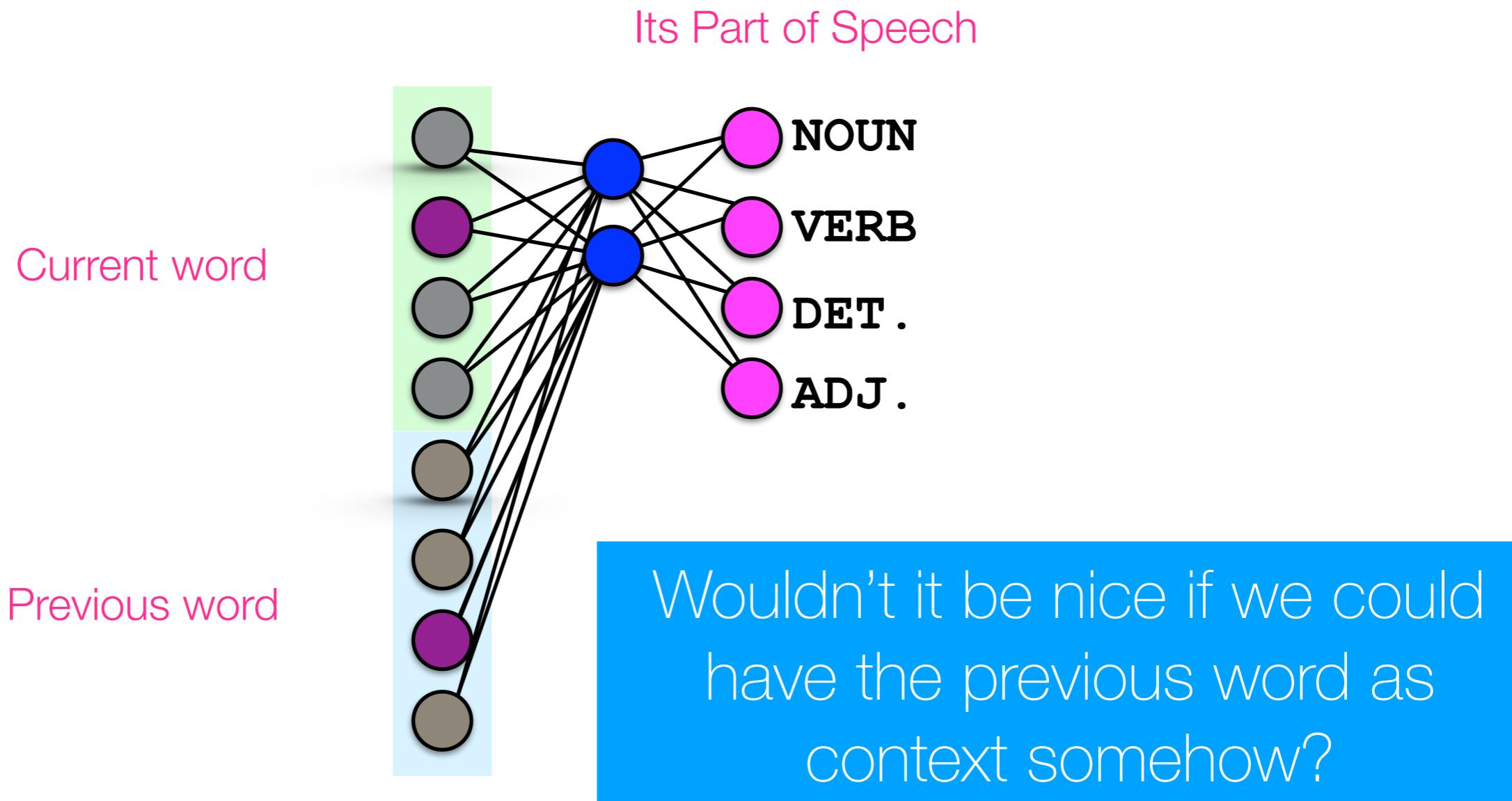


Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

“The very long **play** took two hours”



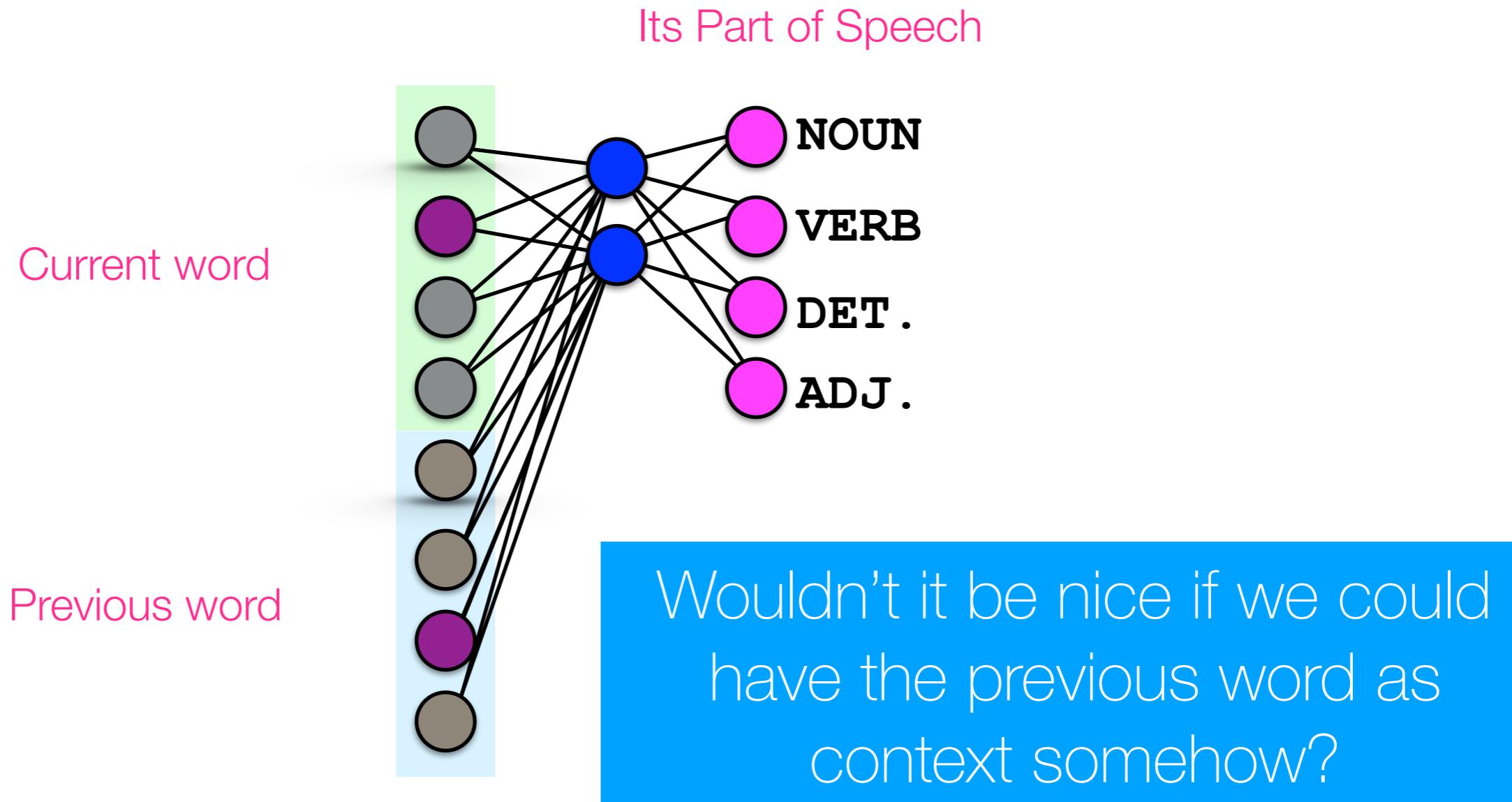
Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

“The very long **play** took two hours”

“The very long but inspiring **play** took two hours”



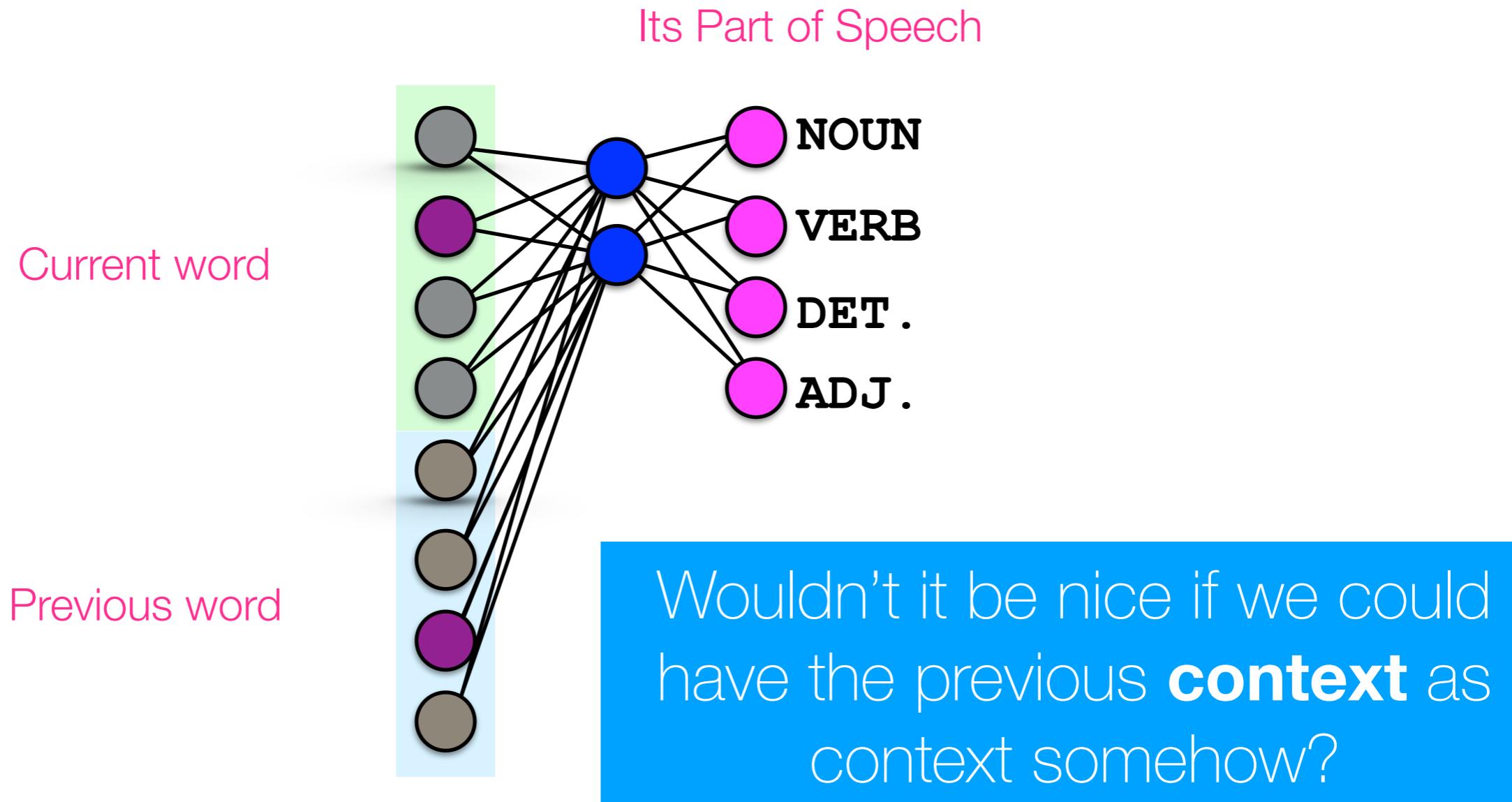
Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

“The very long **play** took two hours”

“The very long but inspiring **play** took two hours”



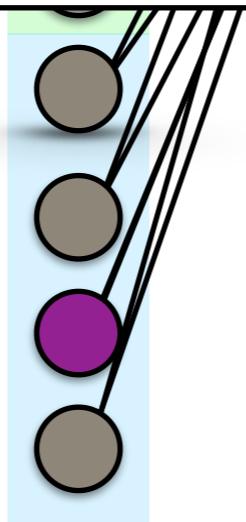
Motivating example: Feed-forward POS tagging

“The **play** took two hours”

Its Part of Speech

Key Insight: If we’re processing a sequence, the **hidden state** captures the previous context

Previous word



Wouldn’t it be nice if we could have the previous word as context somehow?

Motivating example: Feed-forward POS tagging

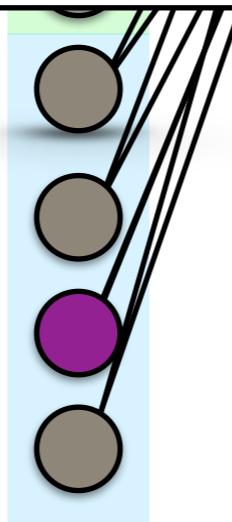
“The **play** took two hours”

“The long **play** took two hours”

Its Part of Speech

Key Insight: If we’re processing a sequence, the **hidden state** captures the previous context

Previous word



Wouldn’t it be nice if we could have the previous word as context somehow?

Motivating example: Feed-forward POS tagging

“The **play** took two hours”

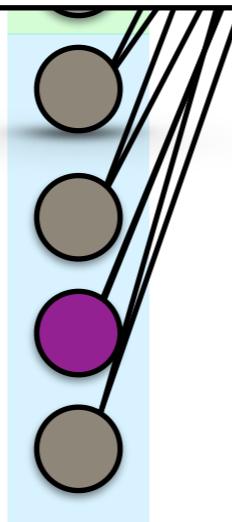
“The long **play** took two hours”

“The very long **play** took two hours”

Its Part of Speech

Key Insight: If we’re processing a sequence, the **hidden state** captures the previous context

Previous word



Wouldn’t it be nice if we could have the previous word as context somehow?

Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

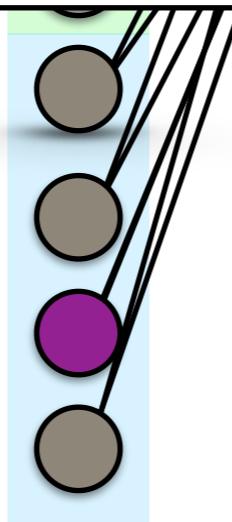
“The very long **play** took two hours”

“The very long but inspiring **play** took two hours”

Its Part of Speech

Key Insight: If we’re processing a sequence, the **hidden state** captures the previous context

Previous word



Wouldn’t it be nice if we could have the previous word as context somehow?

Motivating example: Feed-forward POS tagging

“The **play** took two hours”

“The long **play** took two hours”

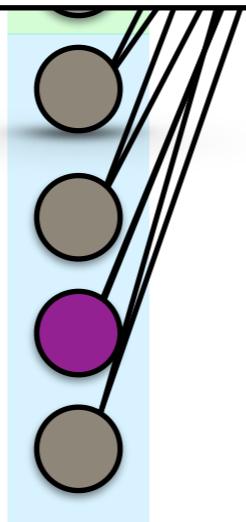
“The very long **play** took two hours”

“The very long but inspiring **play** took two hours”

Its Part of Speech

Key Insight: If we’re processing a sequence, the **hidden state** captures the previous context

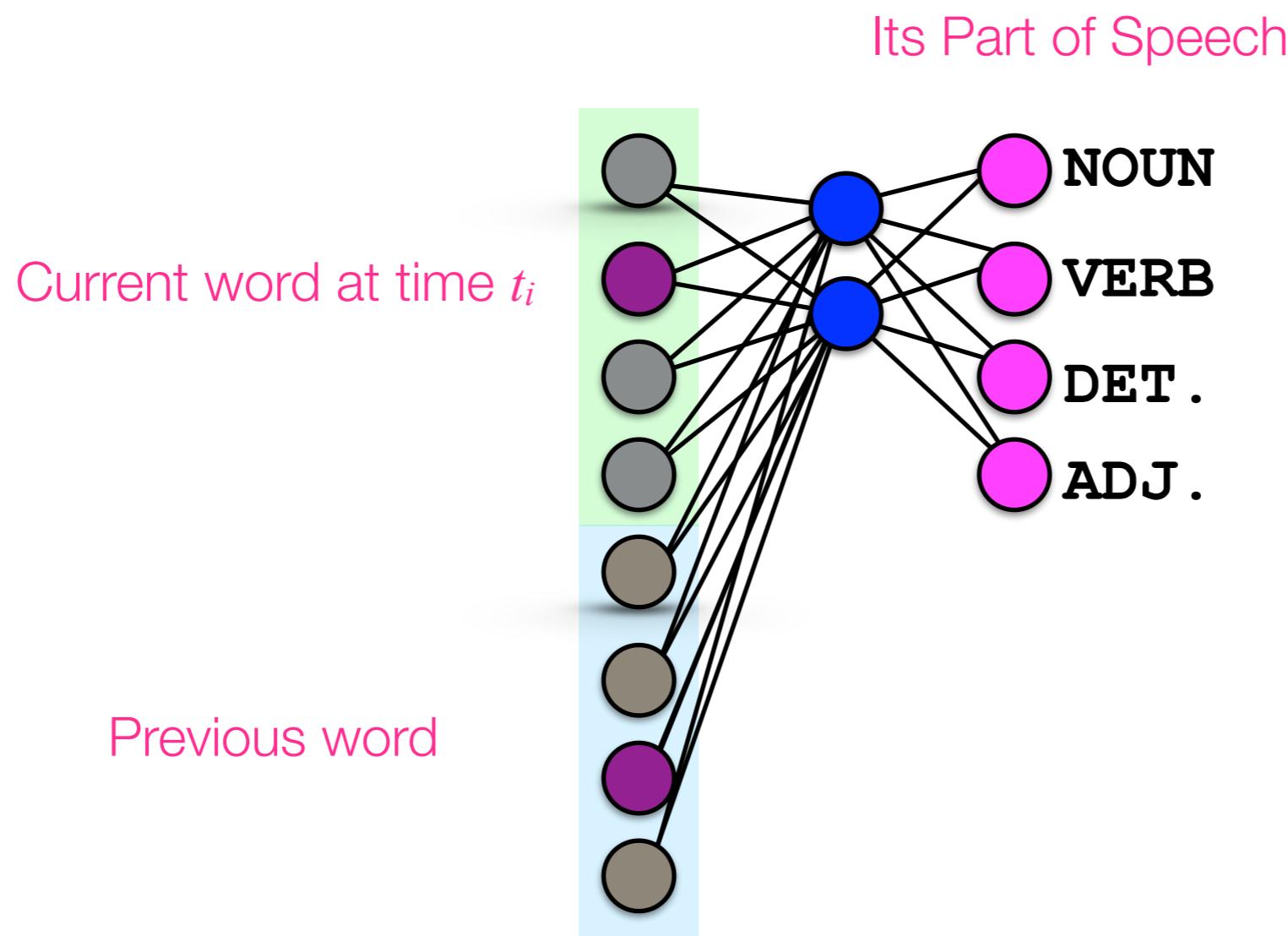
Previous word



Wouldn’t it be nice if we could have the previous **context** as context somehow?

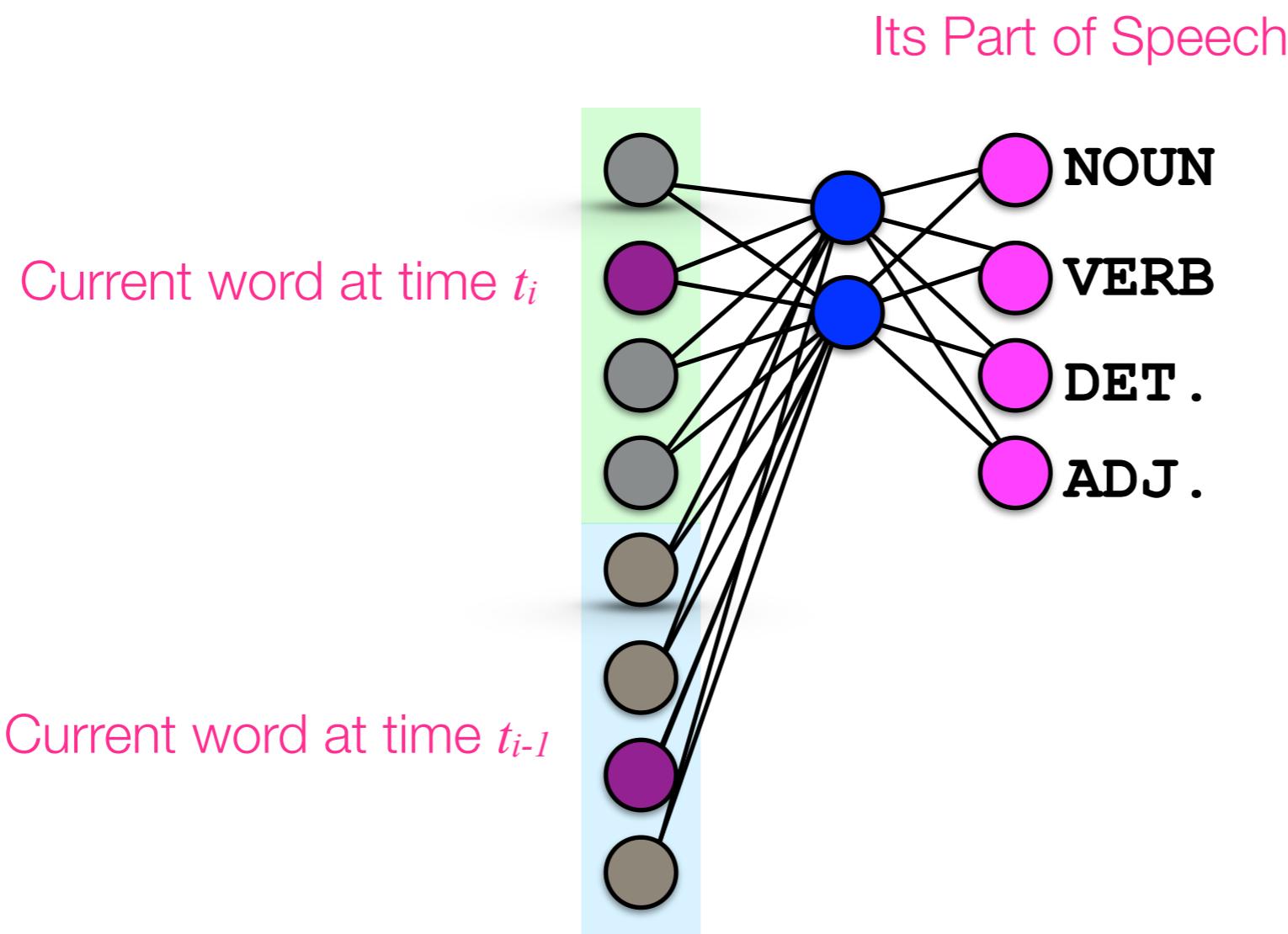
Building a recurrent neural network using the hidden state to capture the previous context

“The **play** took two hours”



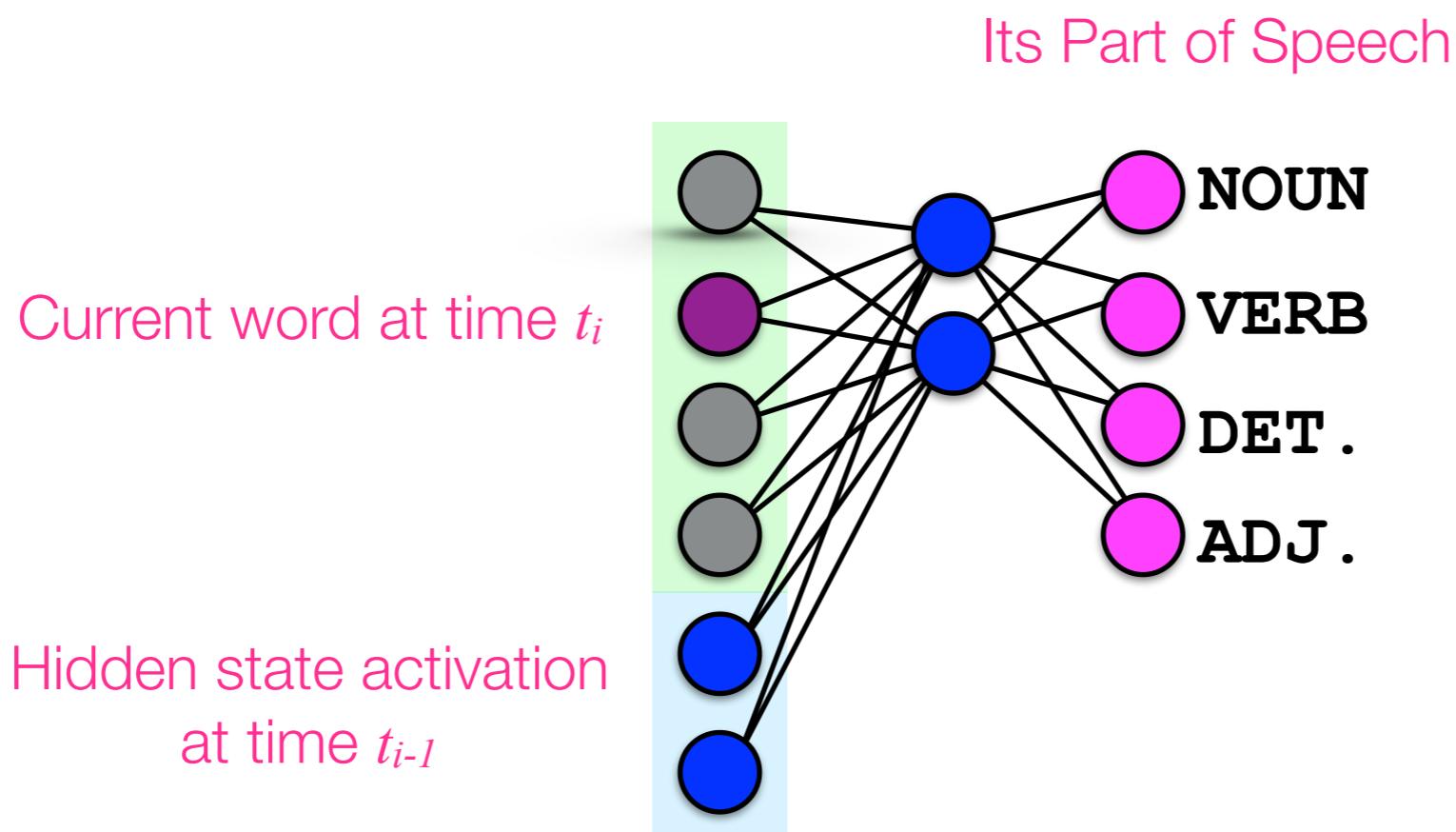
Building a recurrent neural network using the hidden state to capture the previous context

“The **play** took two hours”



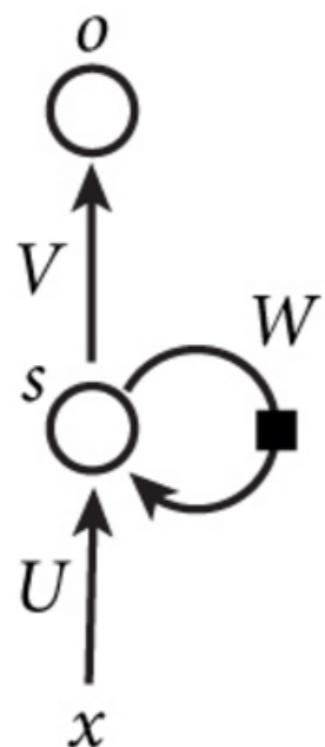
Building a recurrent neural network using the hidden state to capture the previous context

“The play took two hours”



The input consists of the **current word** in the sequence and the **previous word's hidden state activation** which gives the network the **ability to remember** what came before

RNN Diagram



x_t : input at step t

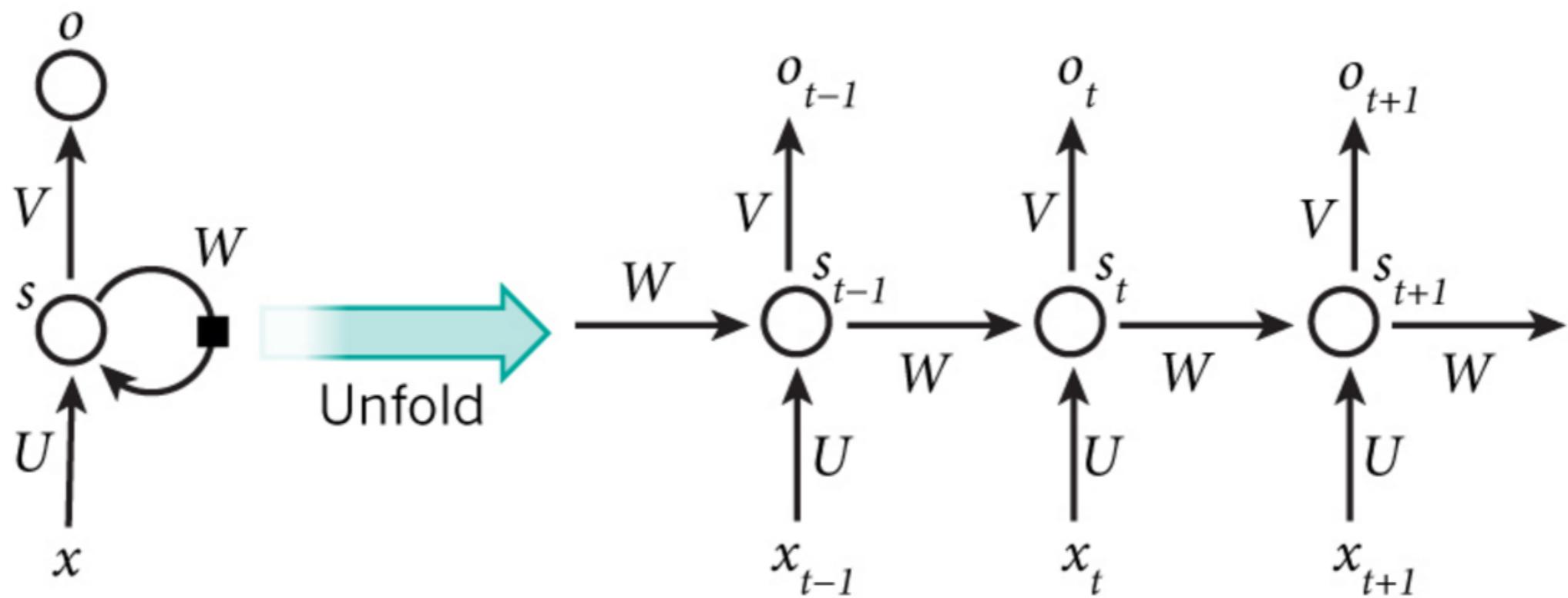
o_t : output at step t

s_t : hidden state at step t

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

RNN Diagram



x_t : input at step t

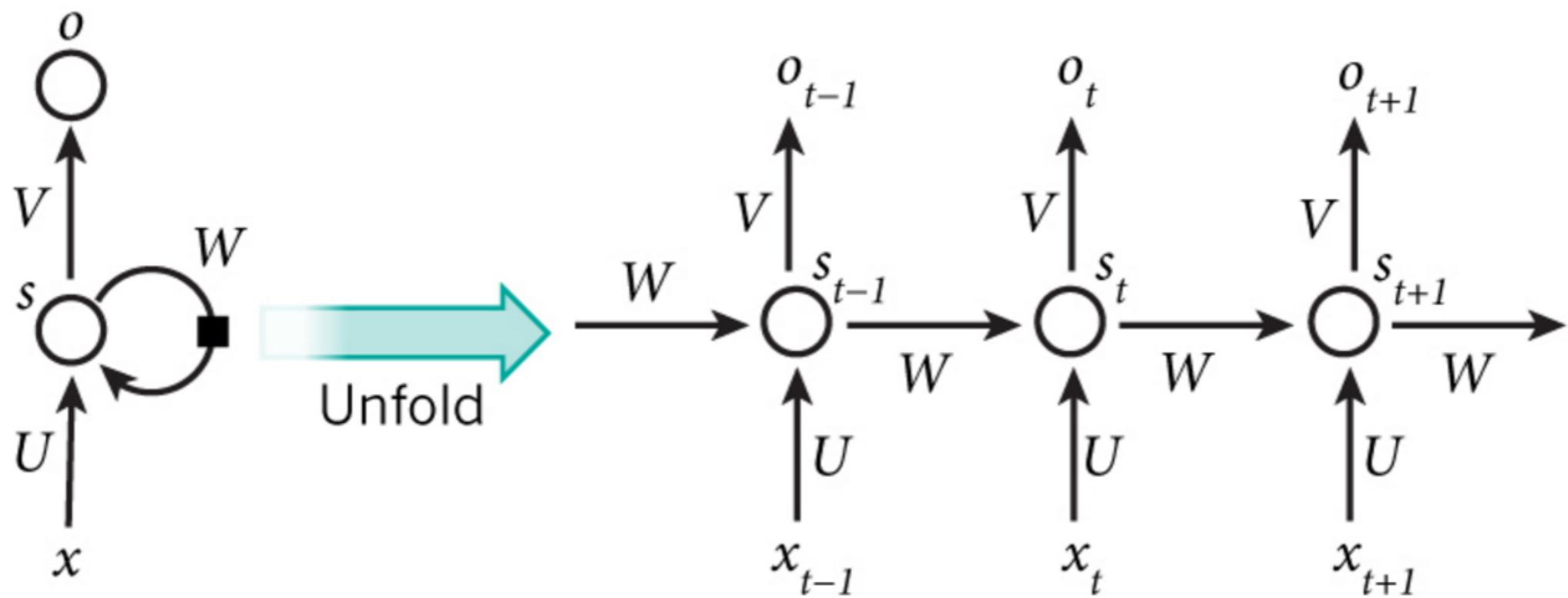
o_t : output at step t

s_t : hidden state at step t

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

RNN Diagram



x_t : input at step t

o_t : output at step t

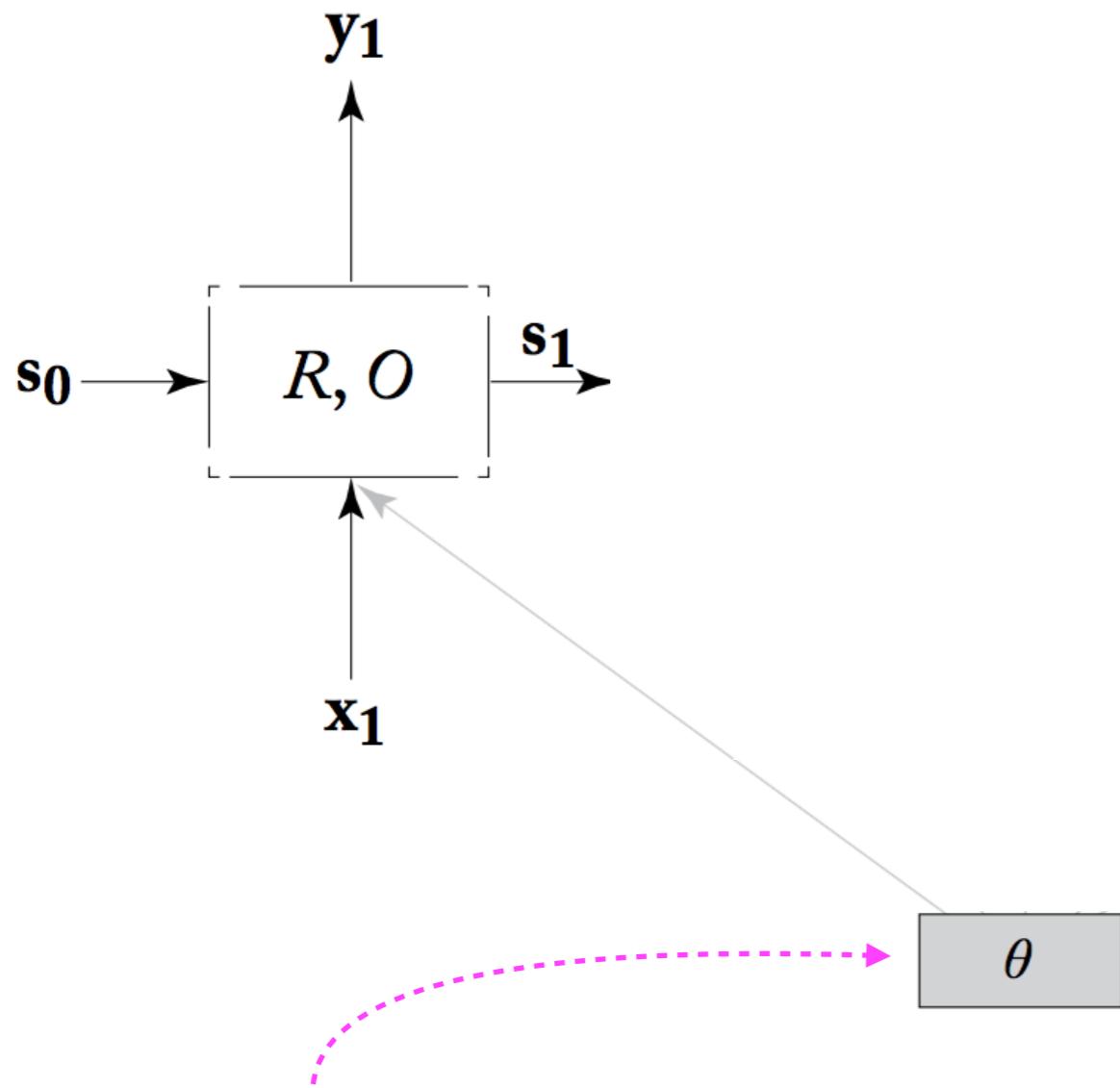
s_t : hidden state at step t ← RNN's Memory

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

Recurrent neural network

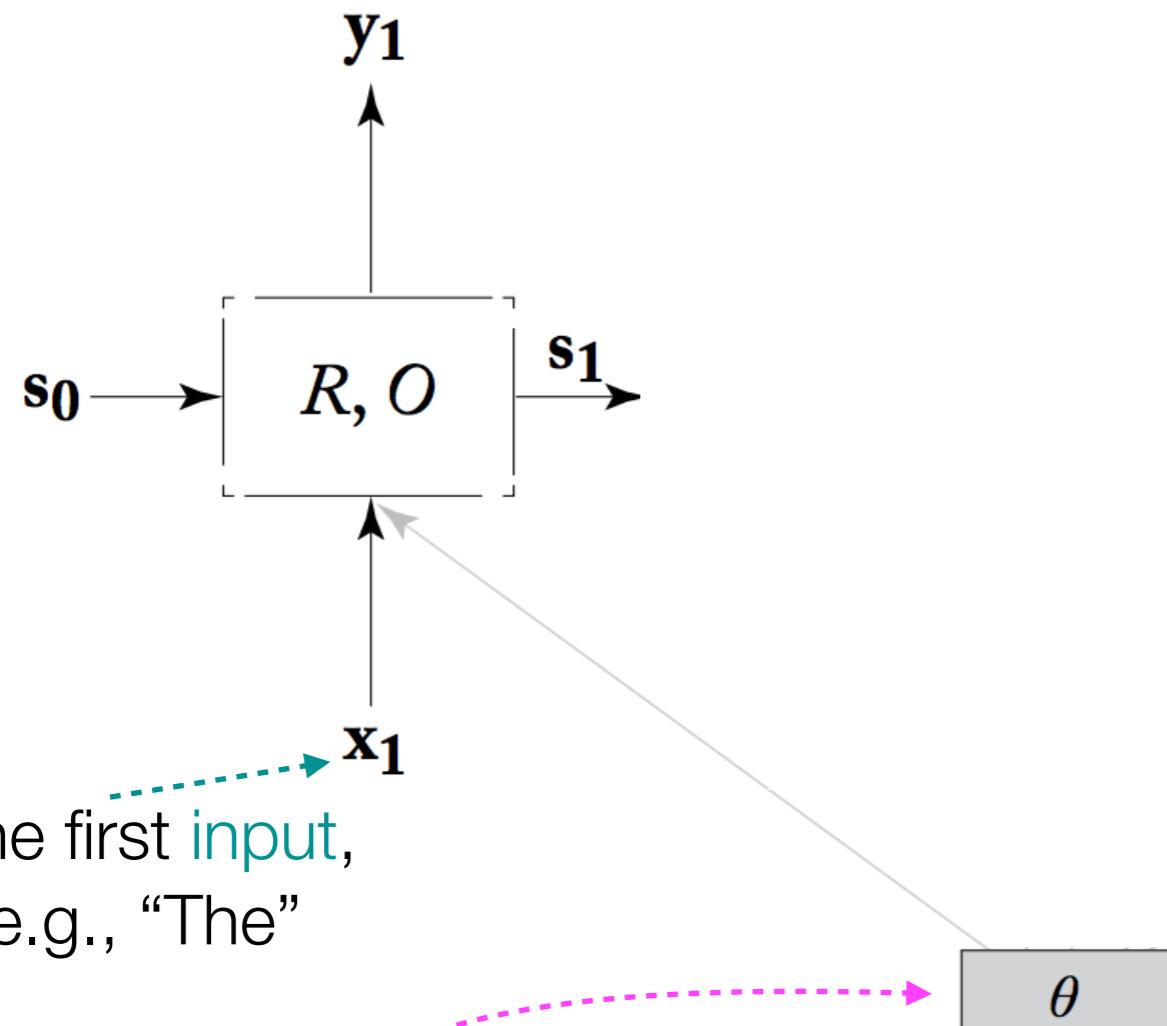
Recurrent neural network



The **parameters** of the model

Goldberg 2017

Recurrent neural network



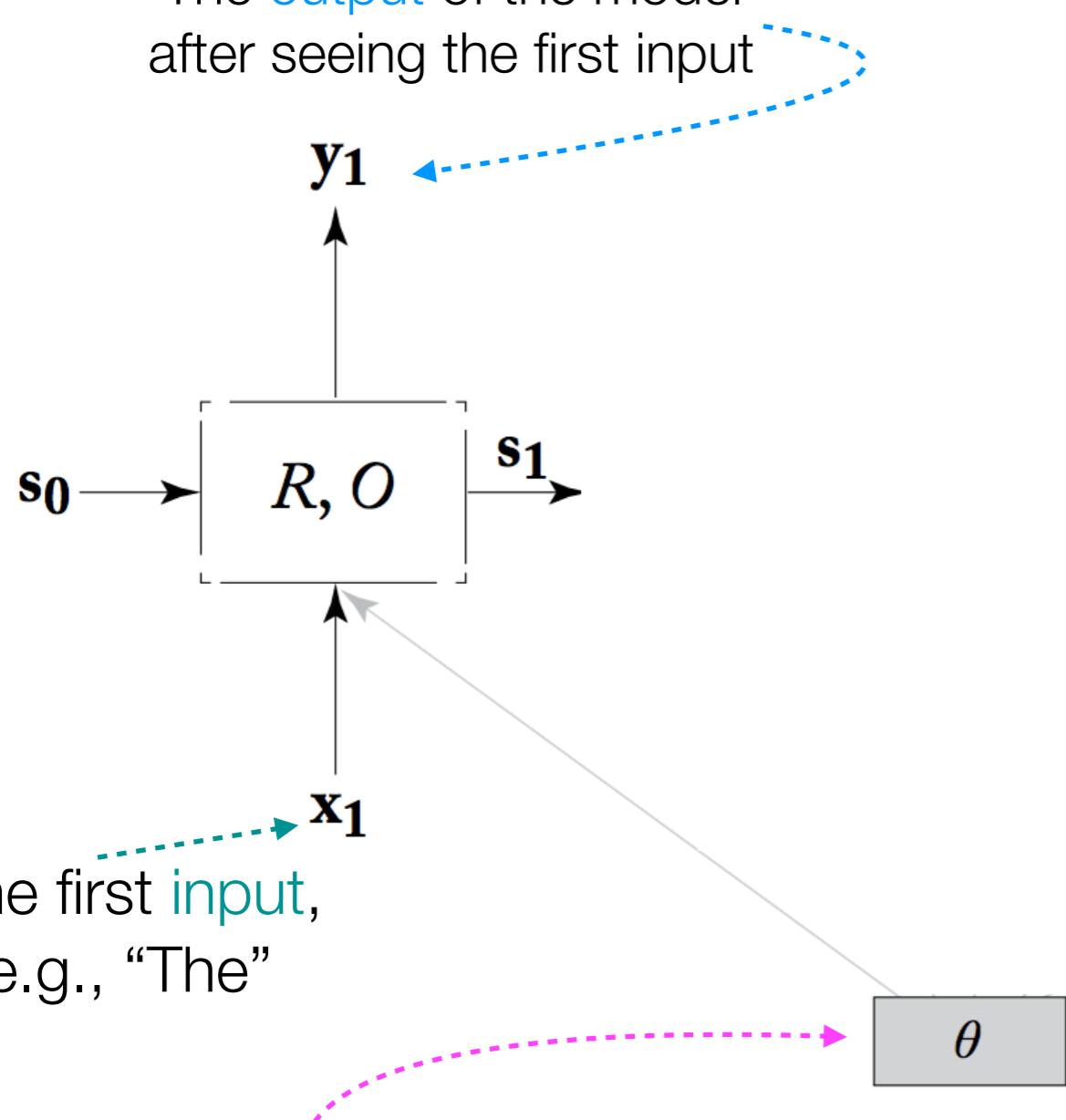
The first input,
e.g., "The"

The parameters of the model

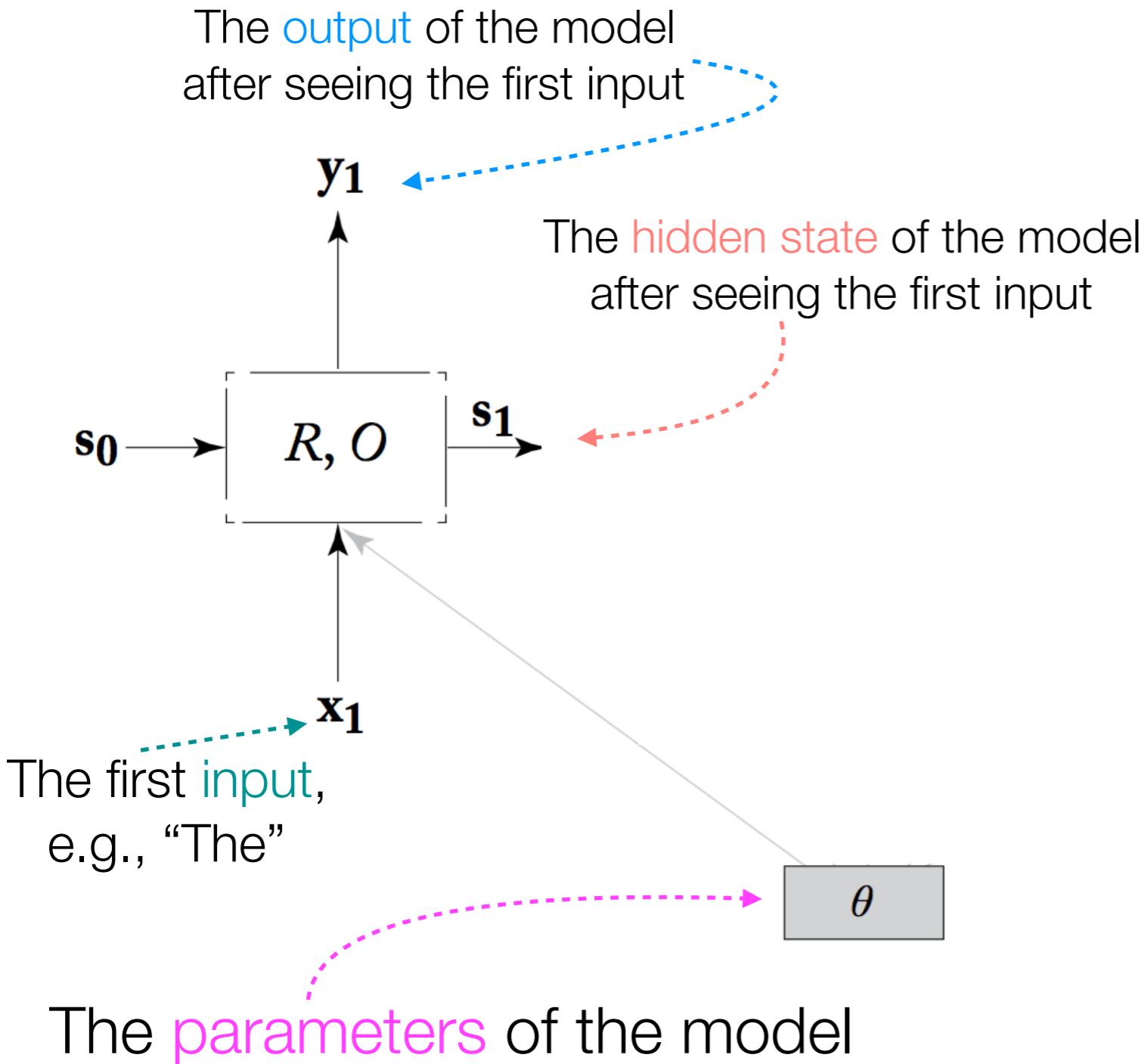
Goldberg 2017

Recurrent neural network

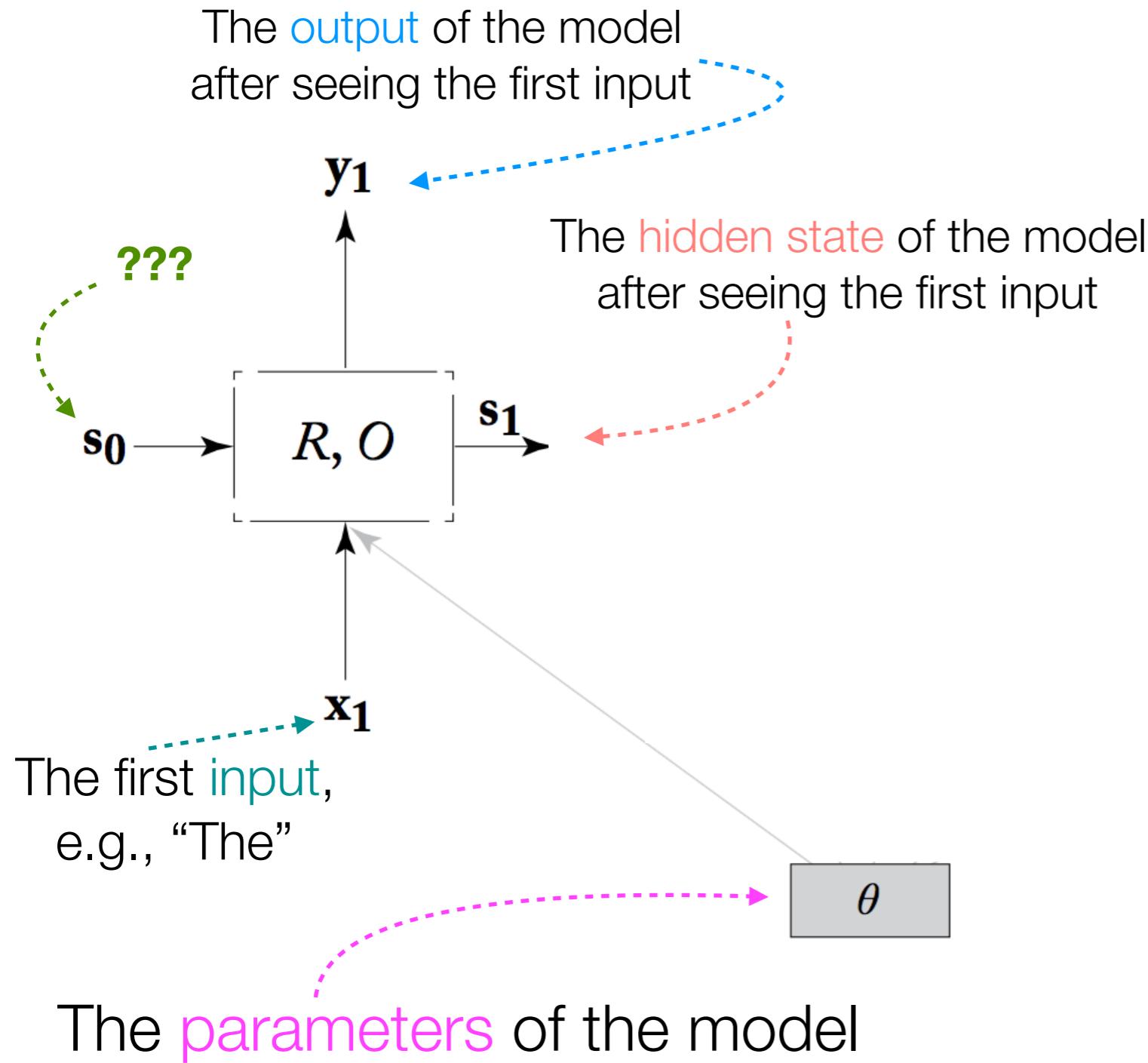
The **output** of the model
after seeing the first input



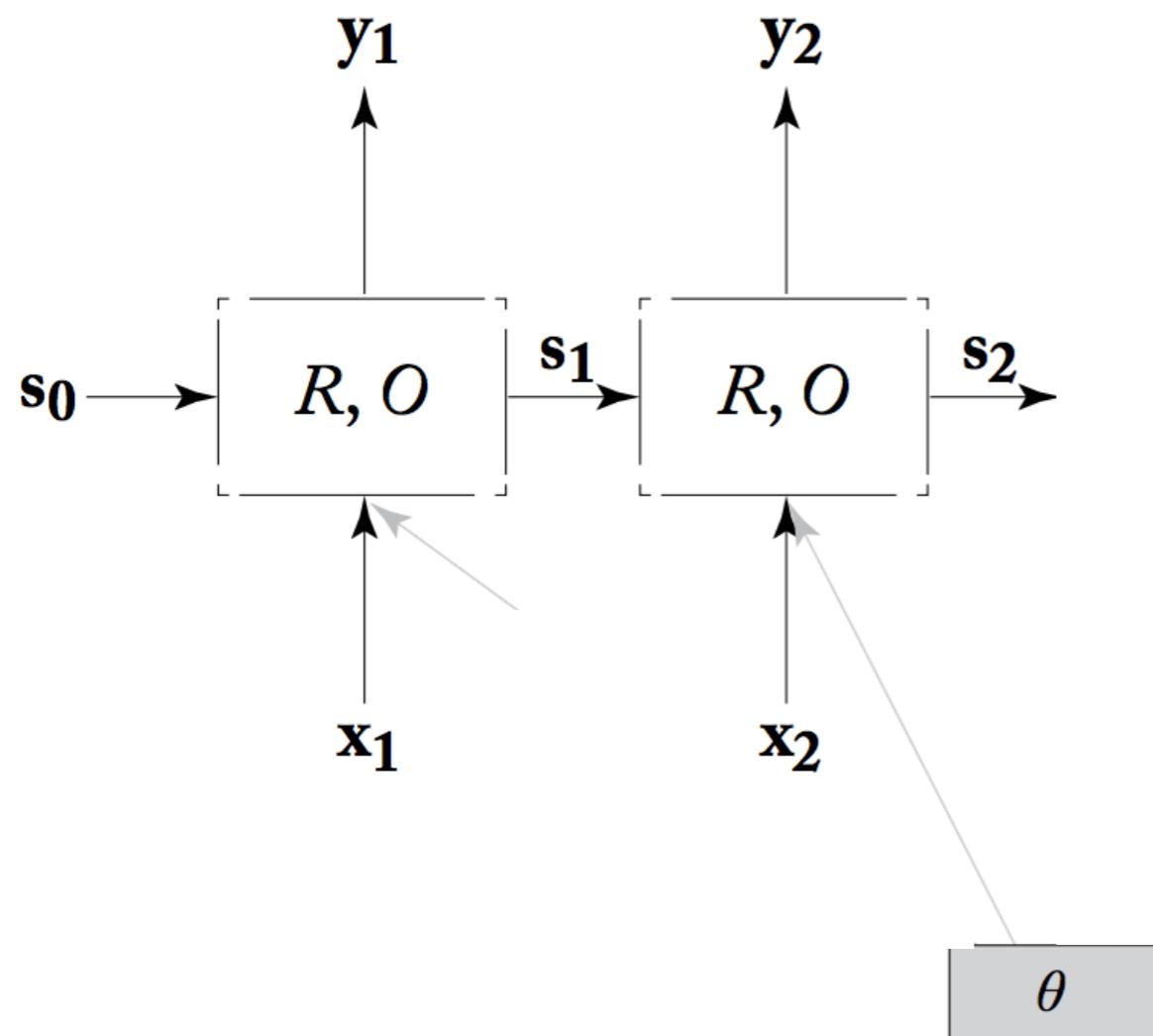
Recurrent neural network



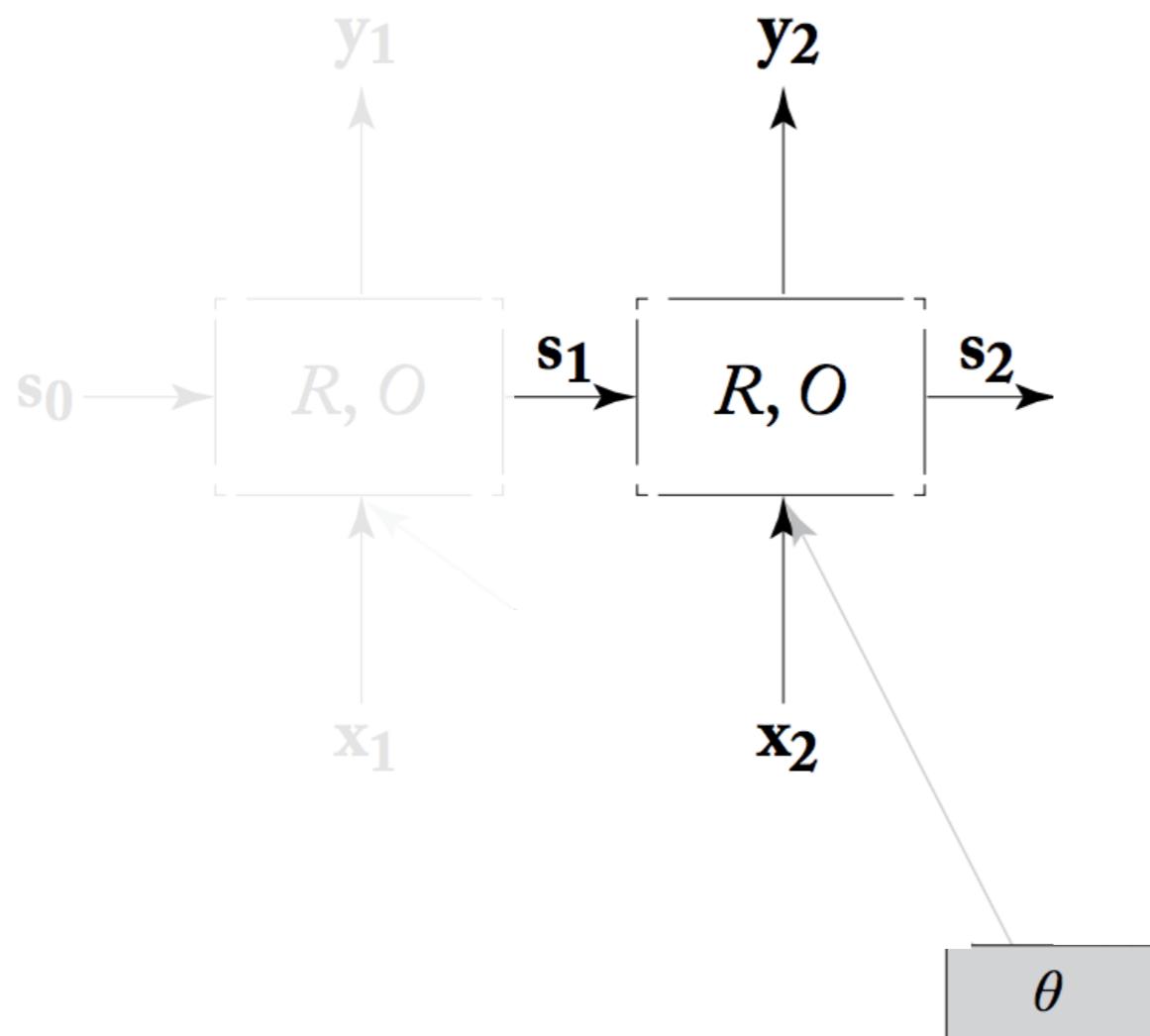
Recurrent neural network



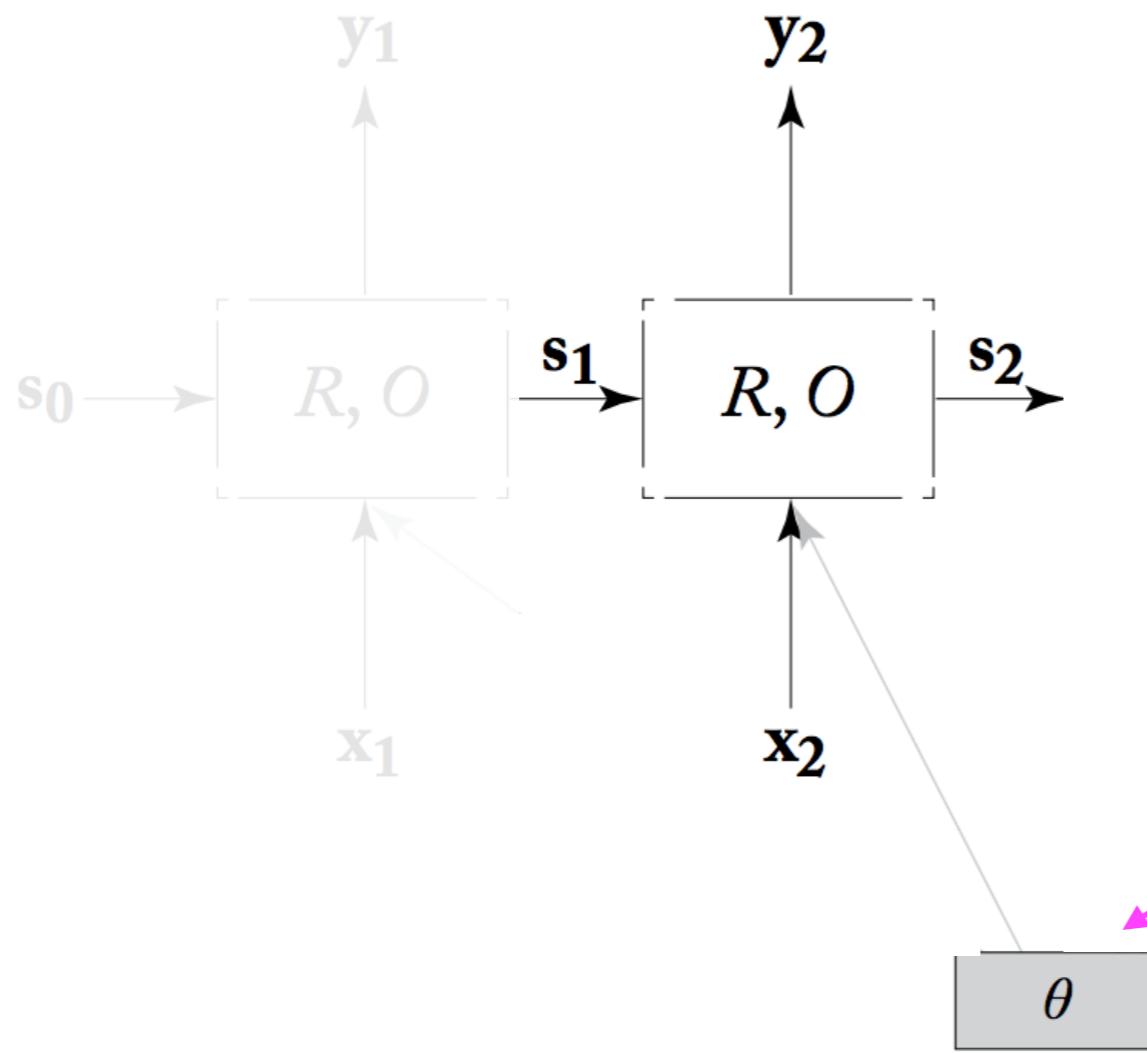
Recurrent neural network



Recurrent neural network

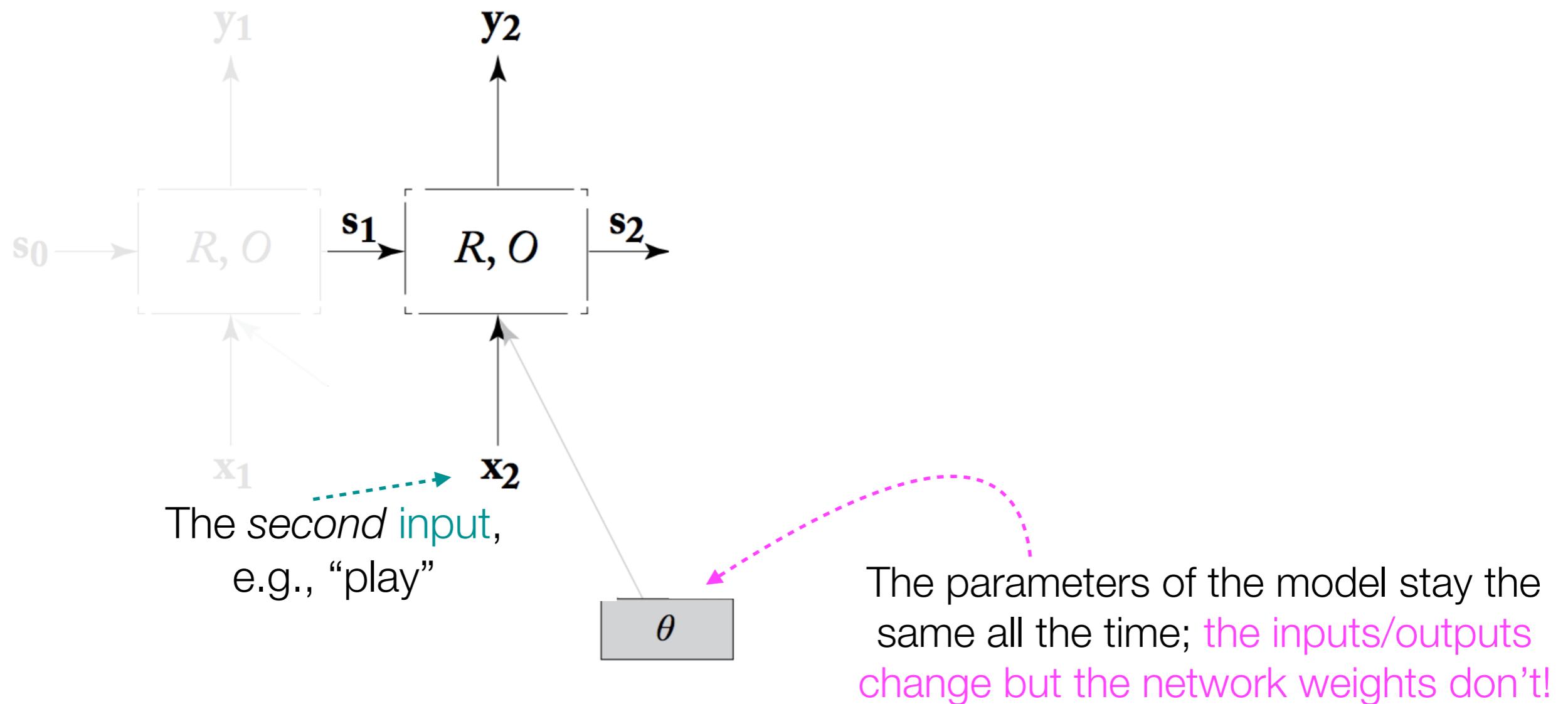


Recurrent neural network

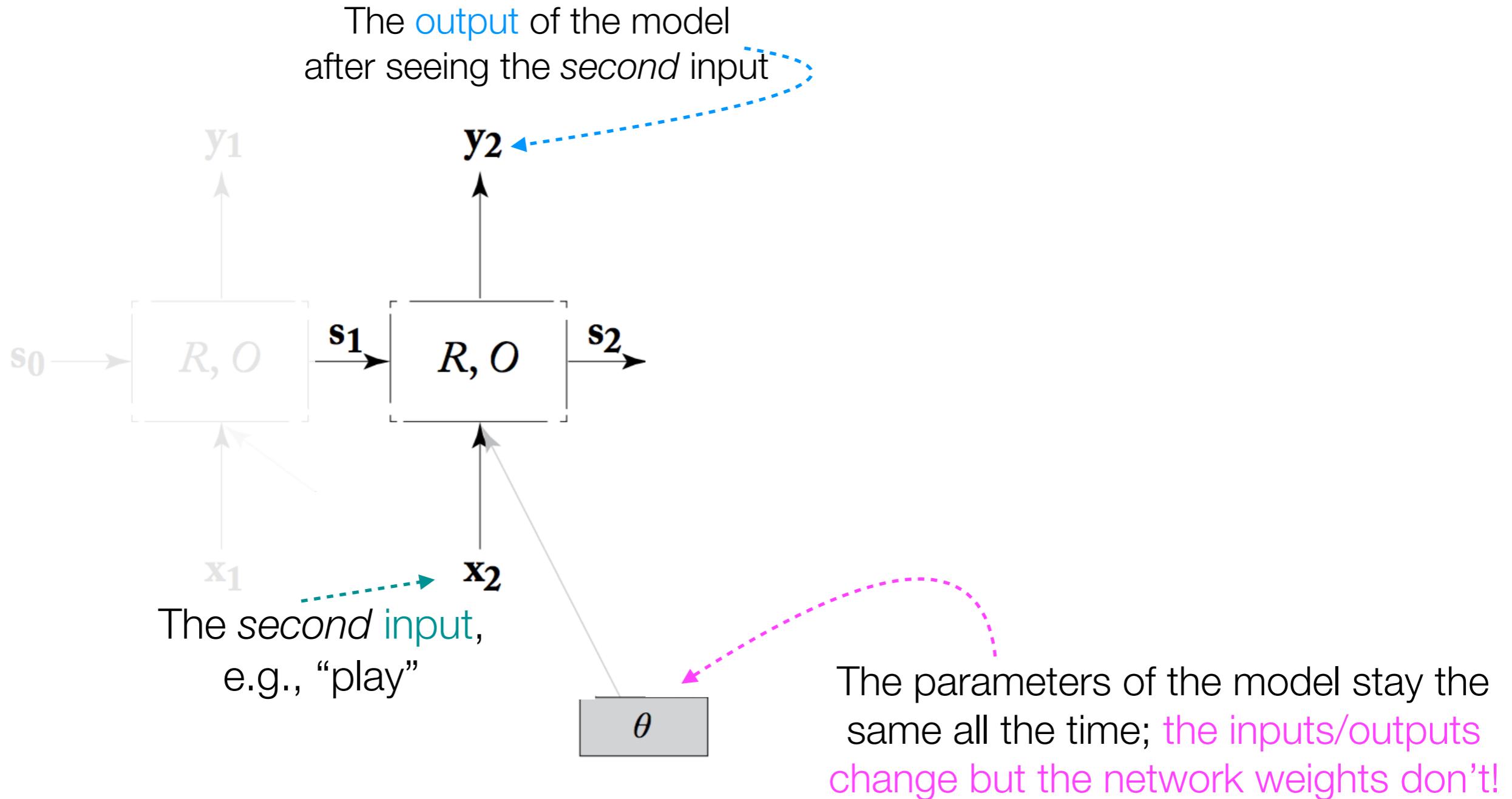


The parameters of the model stay the same all the time; the inputs/outputs change but the network weights don't!

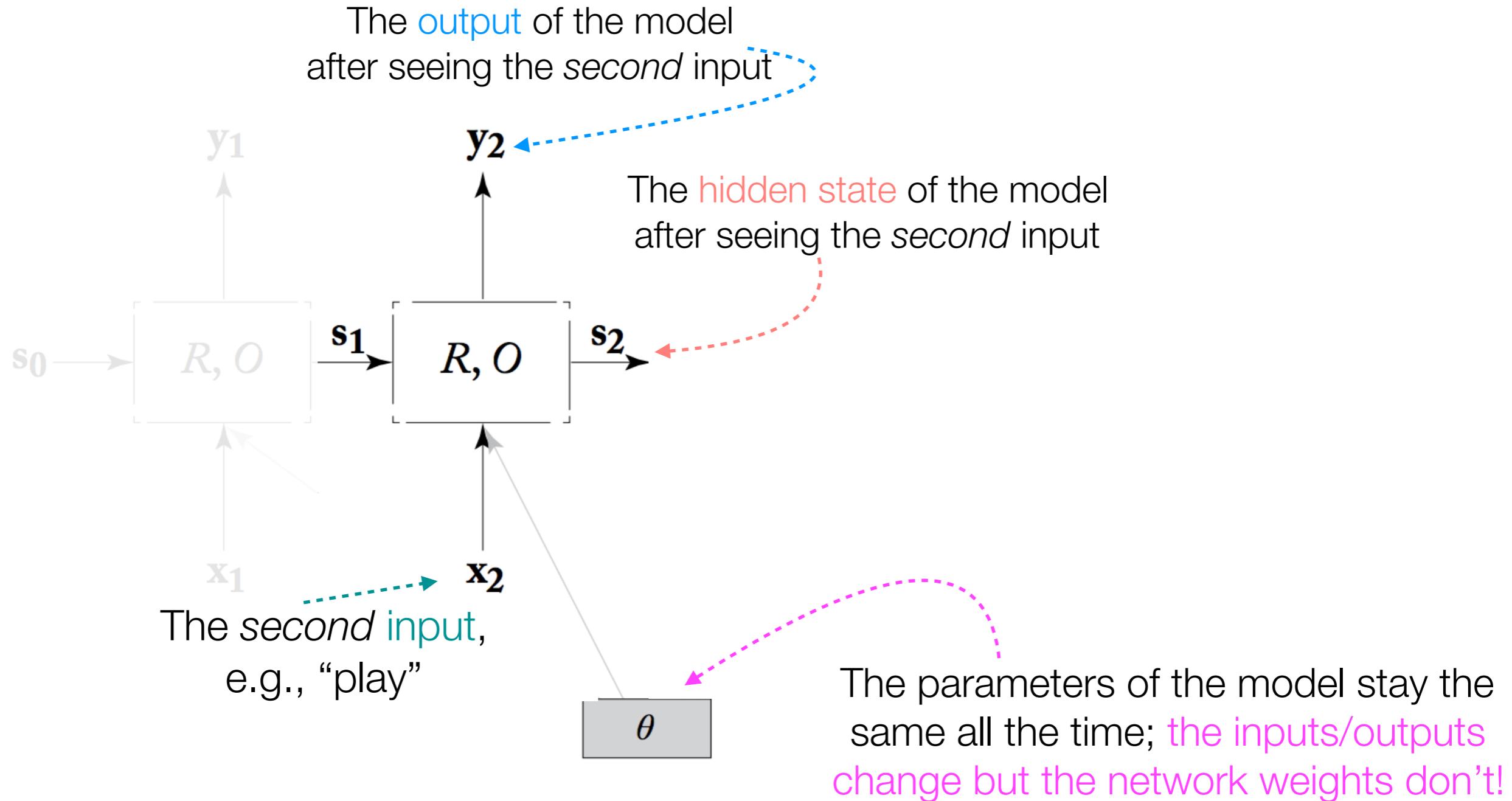
Recurrent neural network



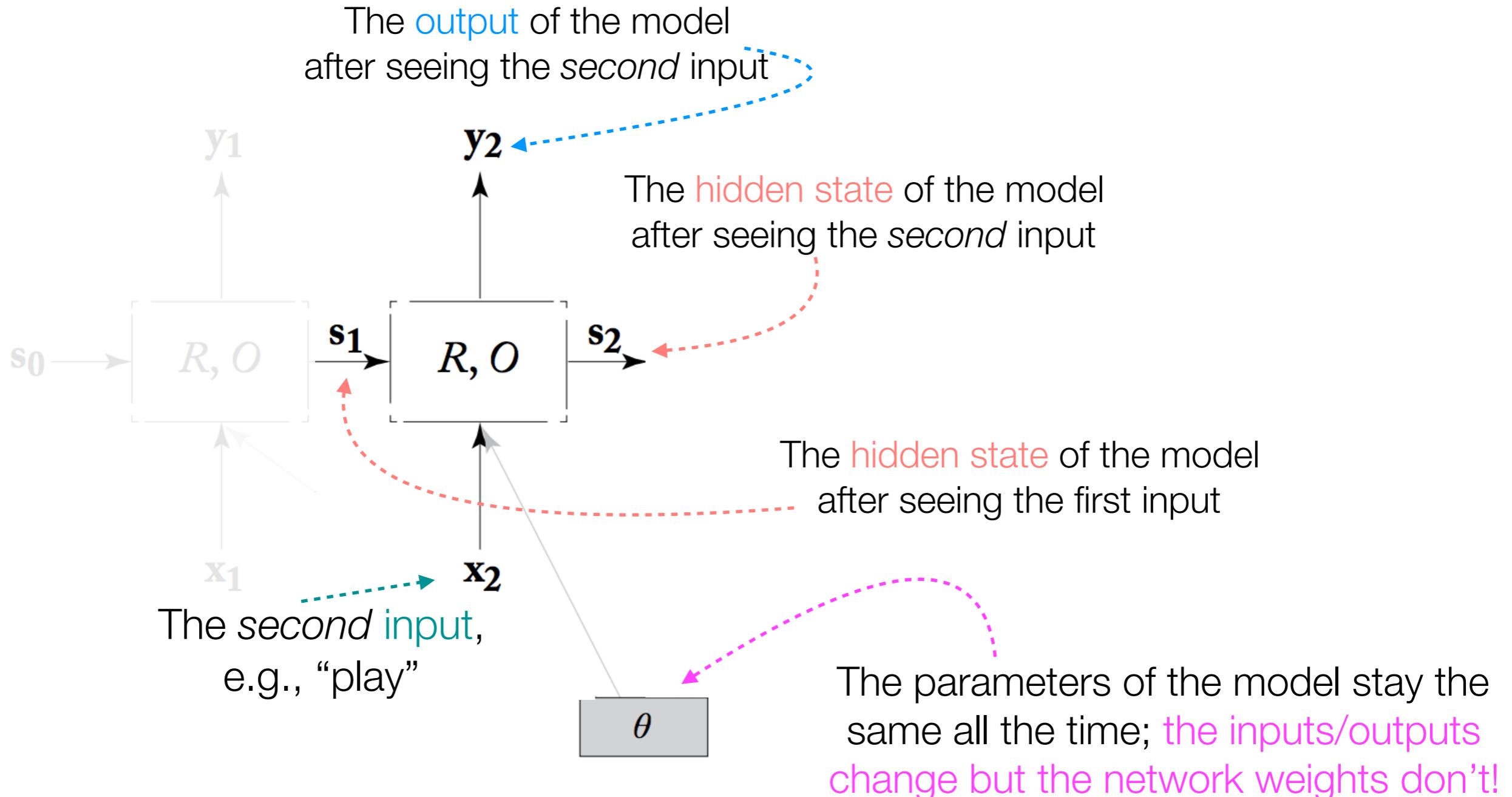
Recurrent neural network



Recurrent neural network

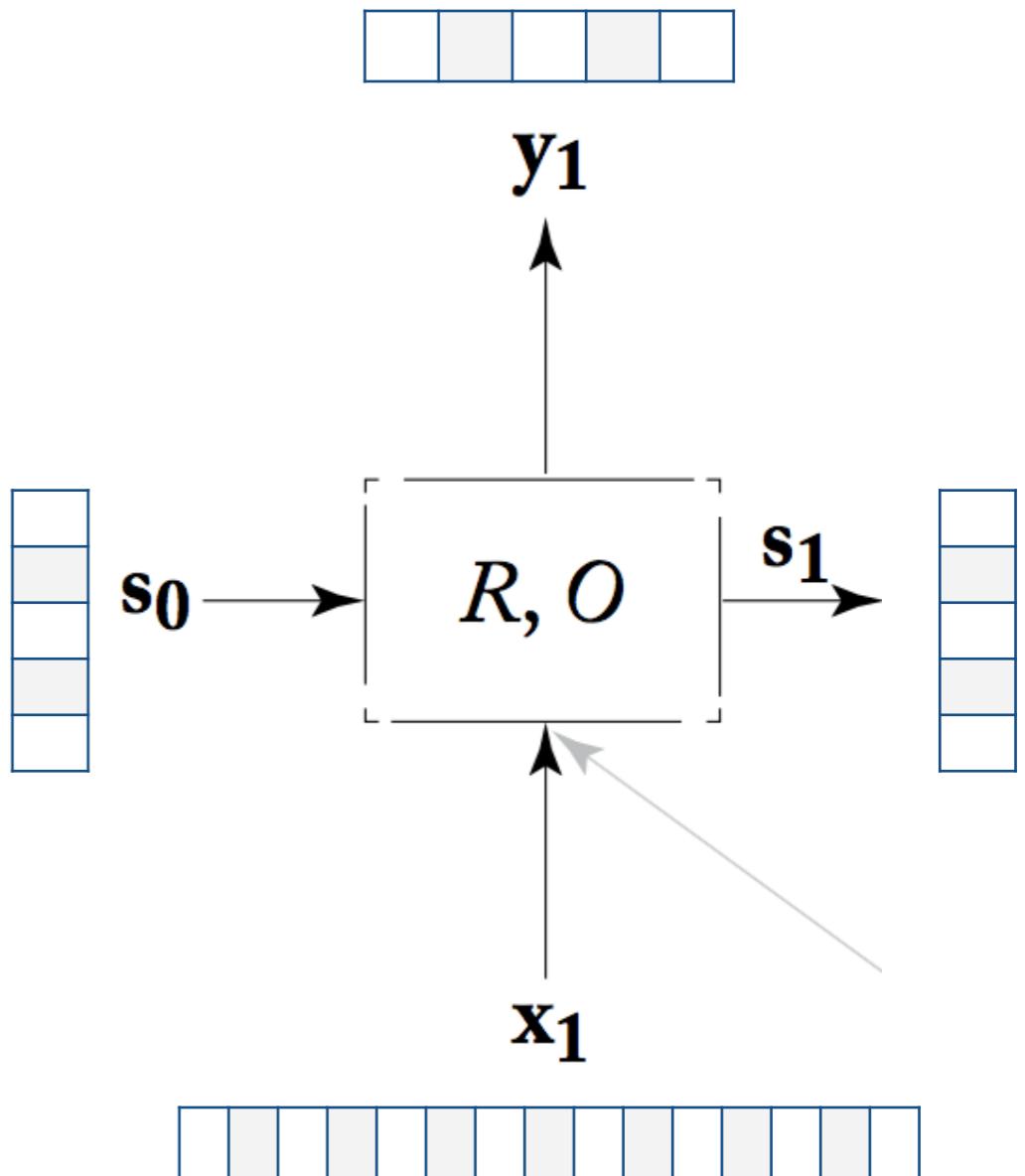


Recurrent neural network



Recurrent neural network

- Each time step has two inputs:
 - x_i (the observation at time step i); one-hot vector, feature vector or **distributed representation**.
 - s_{i-1} (the output of the previous state); **base case:** $s_0 = 0$ vector



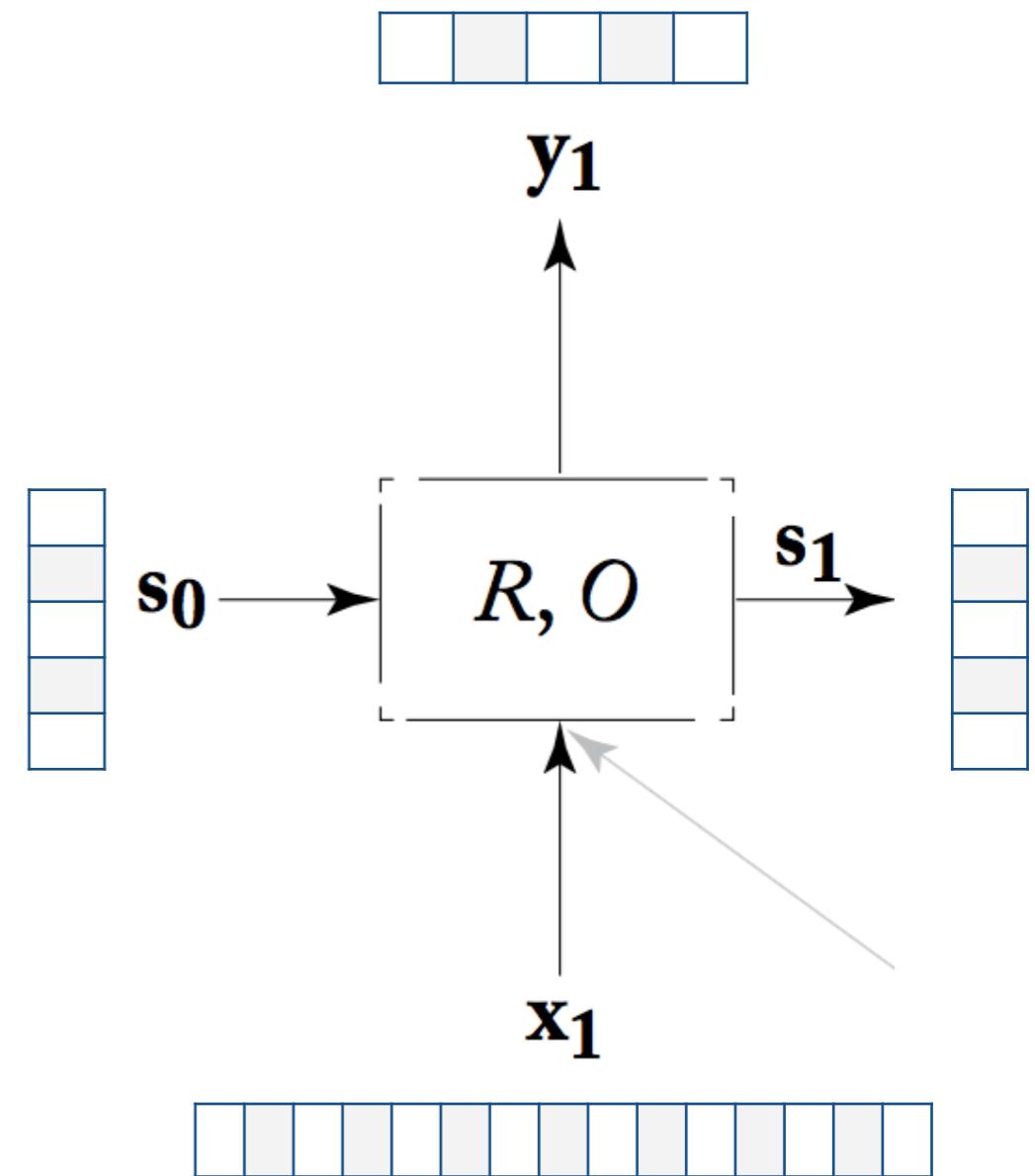
Recurrent neural network

$$s_i = R(x_i, s_{i-1})$$

R computes the output state as a function of the current input and previous state

$$y_i = O(s_i)$$

O computes the output as a function of the current output state



Simple RNN

$g = \tanh$ or relu

$$s_i = R(x_i, s_{i-1}) = g(s_{i-1}W^s + x_iW^x + b)$$

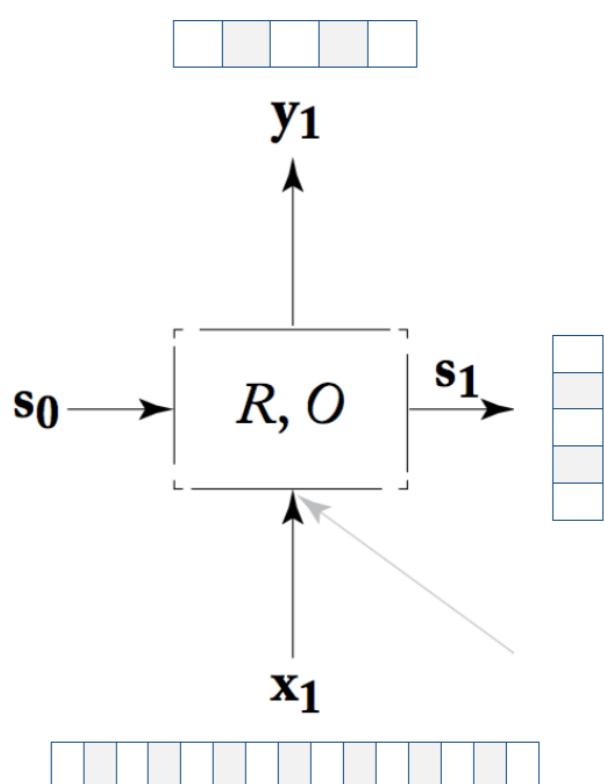
Different weight vectors W transform the previous state and current input before combining

$$W^s \in \mathbb{R}^{H \times H}$$

$$W^x \in \mathbb{R}^{D \times H}$$

$$b \in \mathbb{R}^H$$

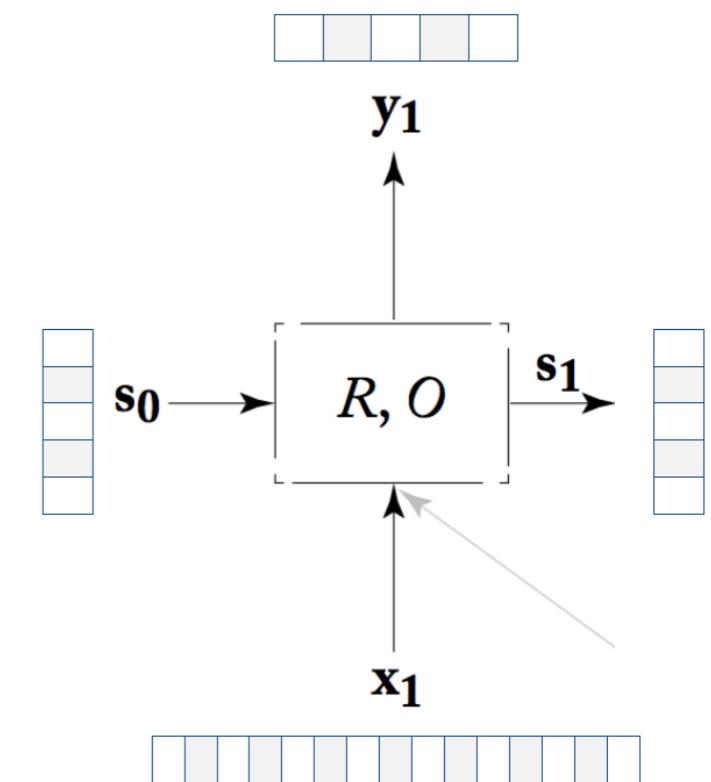
$$y_i = O(s_i) = s_i$$



RNN Language Model

- The output state s_i is an H -dimensional real vector; we can transform that vector into a probability distribution by passing it through an additional linear transformation followed by a softmax

$$y_i = O(s_i) = \text{softmax}(s_i W^o + b^o)$$



RNNs are very effective language models in practice

- Comparison on Penn Treebank, which is a small, standard dataset, ~1M words
- RNN outperforms Feed-forward NN by about 10%

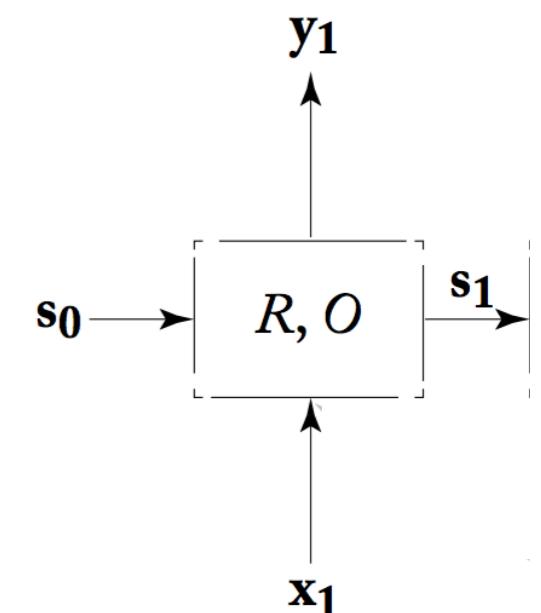
Model	Perplexity
Kneser-Ney 5-gram	141
Maxent 5-gram	142
Random forest	132
Feedforward NNLM	140
Recurrent NNLM	125

Training RNNs

- Given this definition of an RNN:

$$s_i = R(x_i, s_{i-1}) = g(s_{i-1}W^s + x_iW^x + b)$$

$$y_i = O(s_i) = \text{softmax}(s_iW^o + b^o)$$



- We have five sets of parameters to learn:

$$W^s, W^x, W^o, b, b^o$$

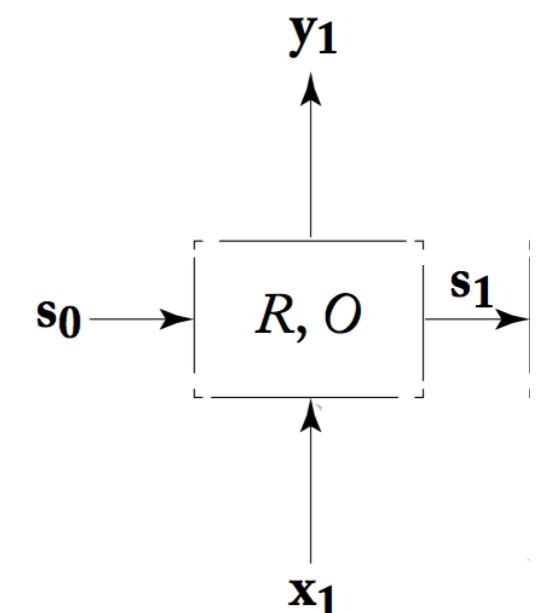
Training RNNs

- Given this definition of an RNN:

$$g = \text{tanh or relu}$$

$$s_i = R(x_i, s_{i-1}) = g(s_{i-1}W^s + x_iW^x + b)$$

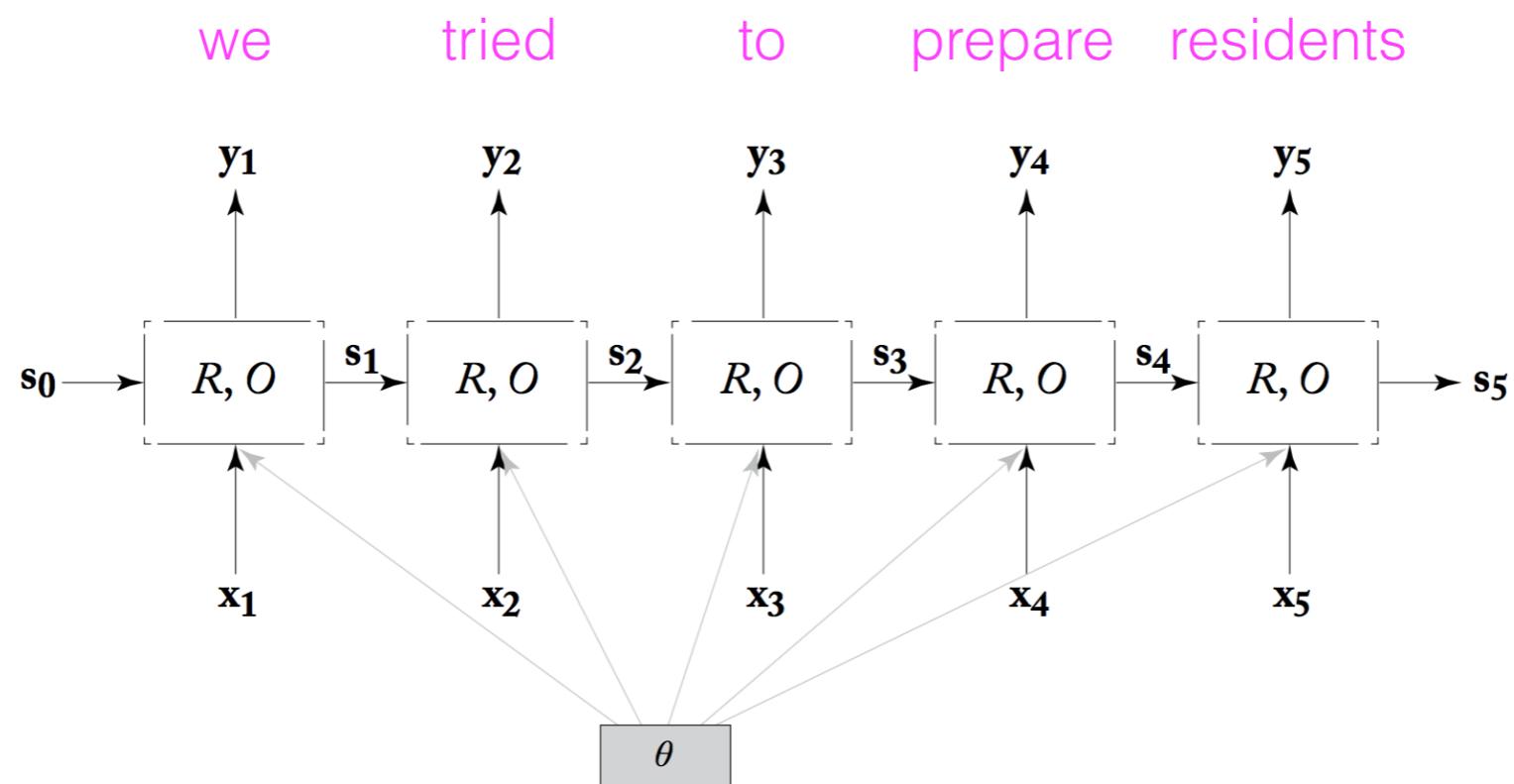
$$y_i = O(s_i) = \text{softmax}(s_iW^o + b^o)$$



- We have five sets of parameters to learn:

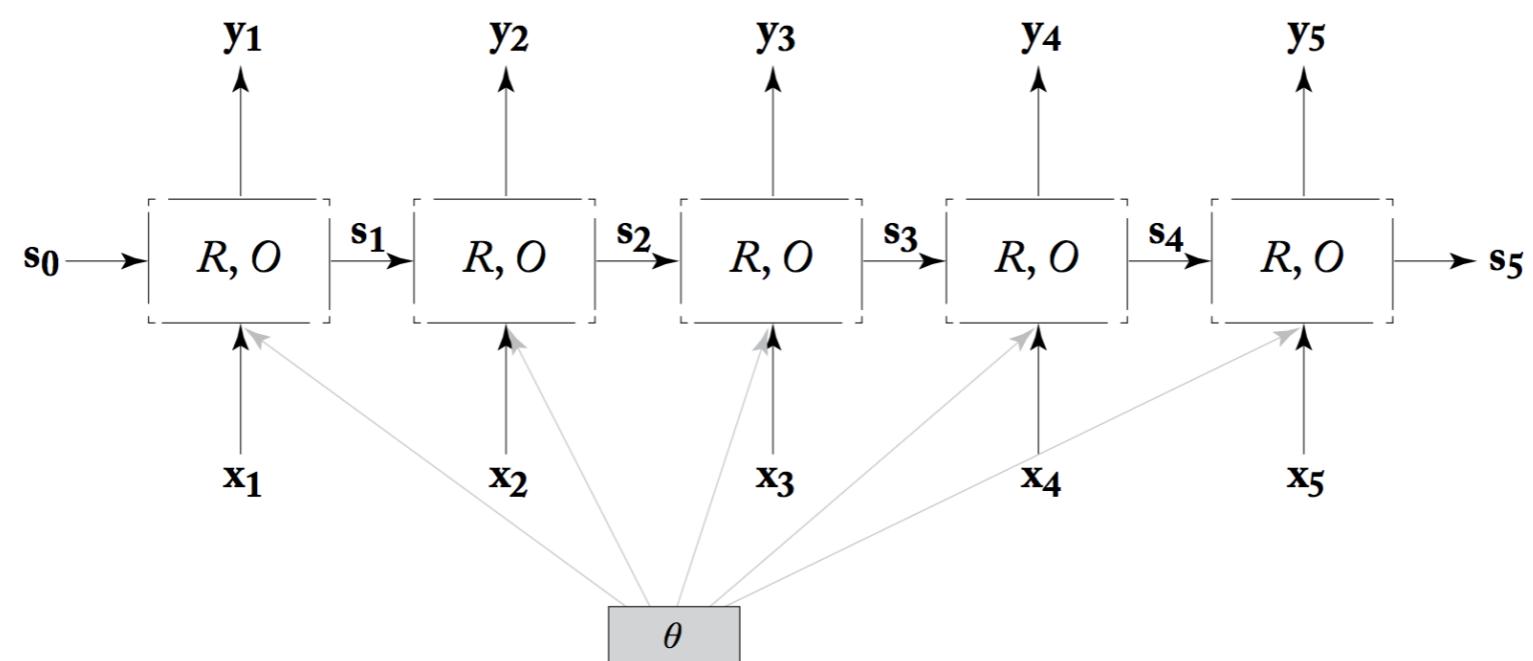
$$W^s, W^x, W^o, b, b^o$$

Training RNNs

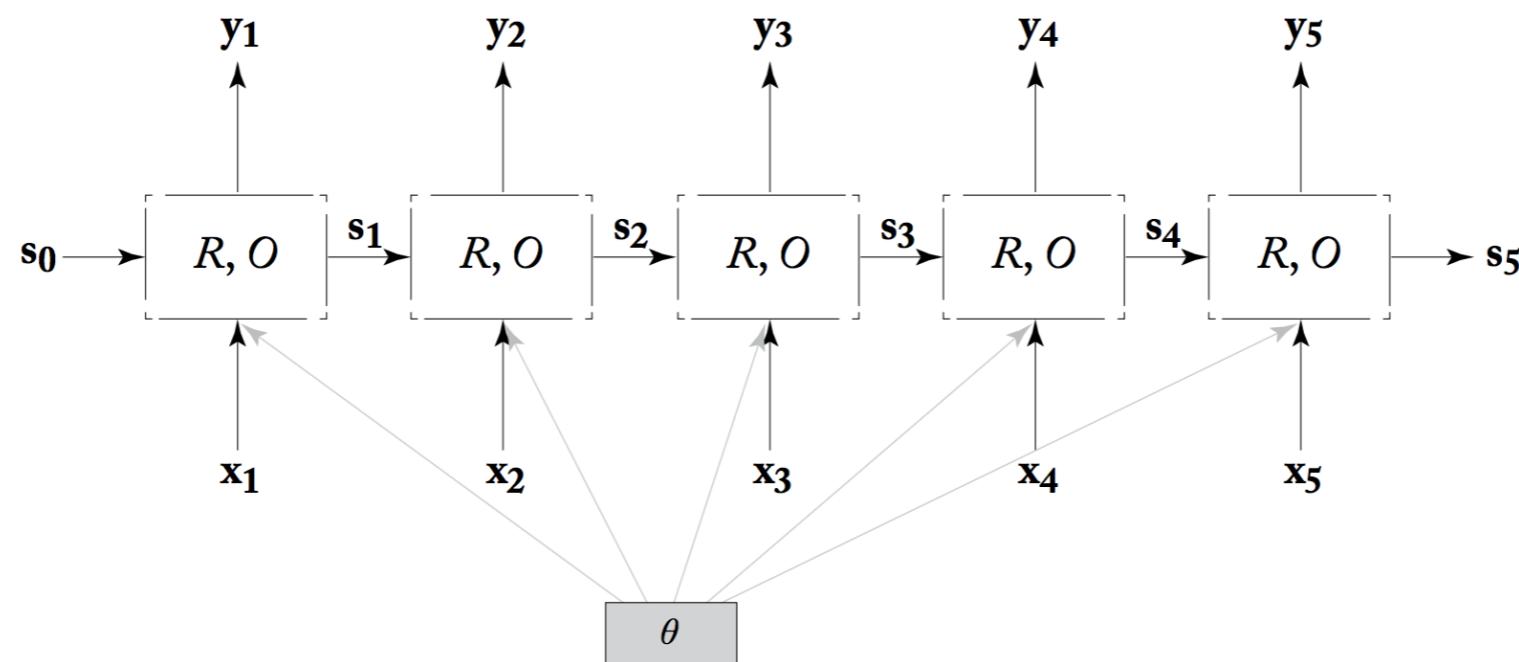


- At each time step, we make a prediction and incur a loss; we know the true y (the word we see in that position)

we tried to prepare residents



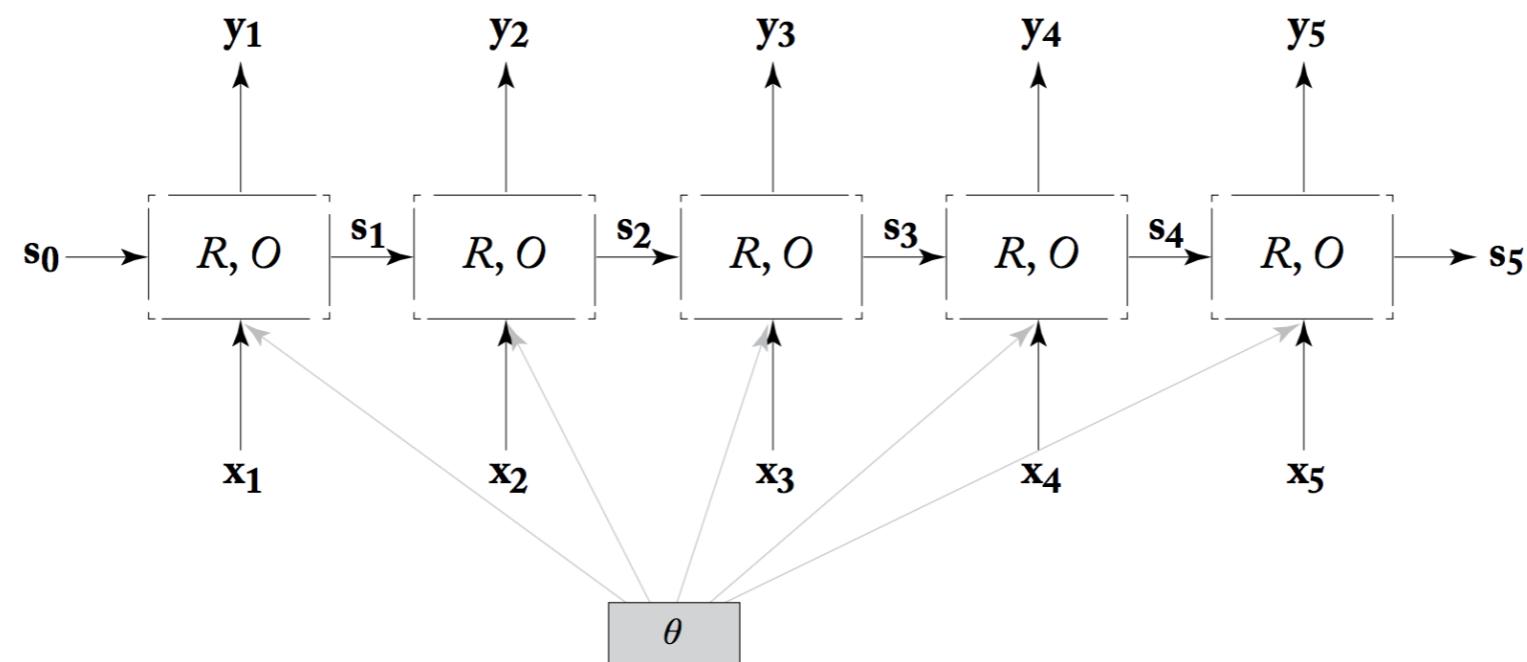
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s}$$

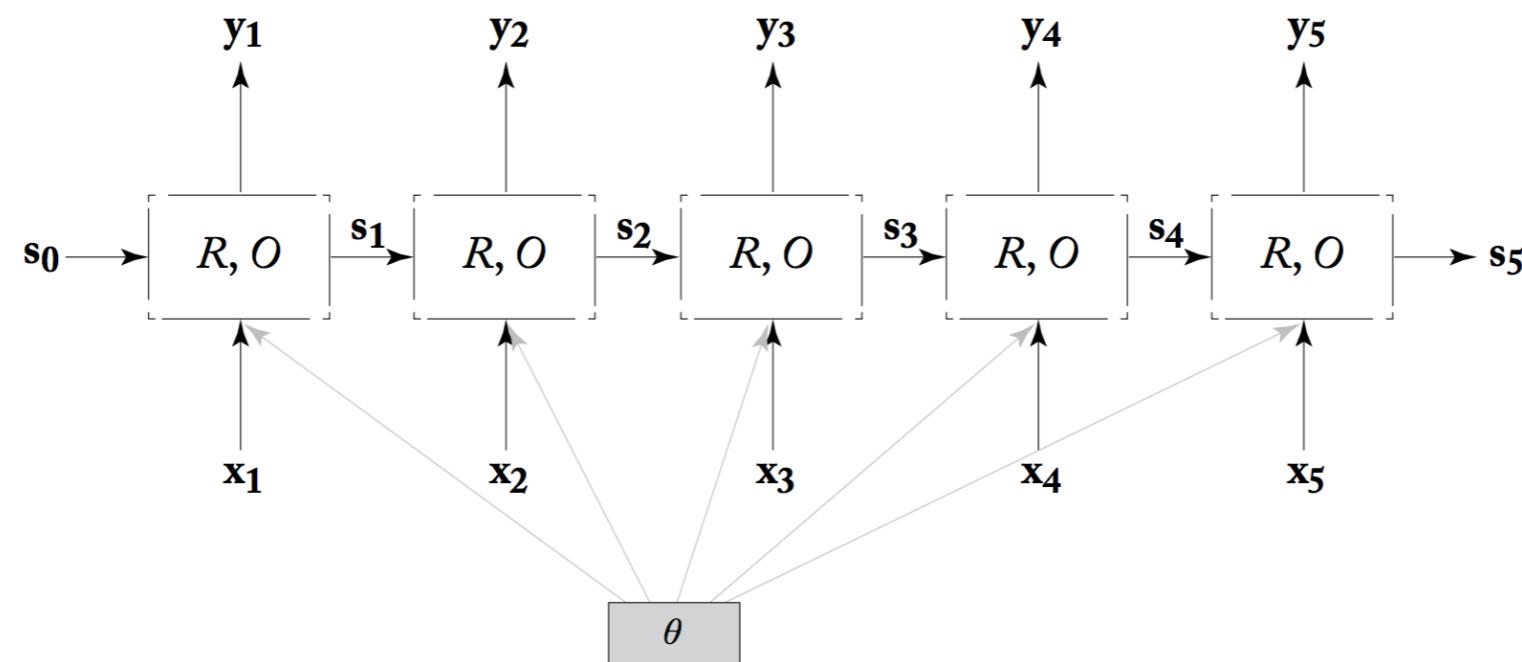
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_2}}{\partial W^s}$$

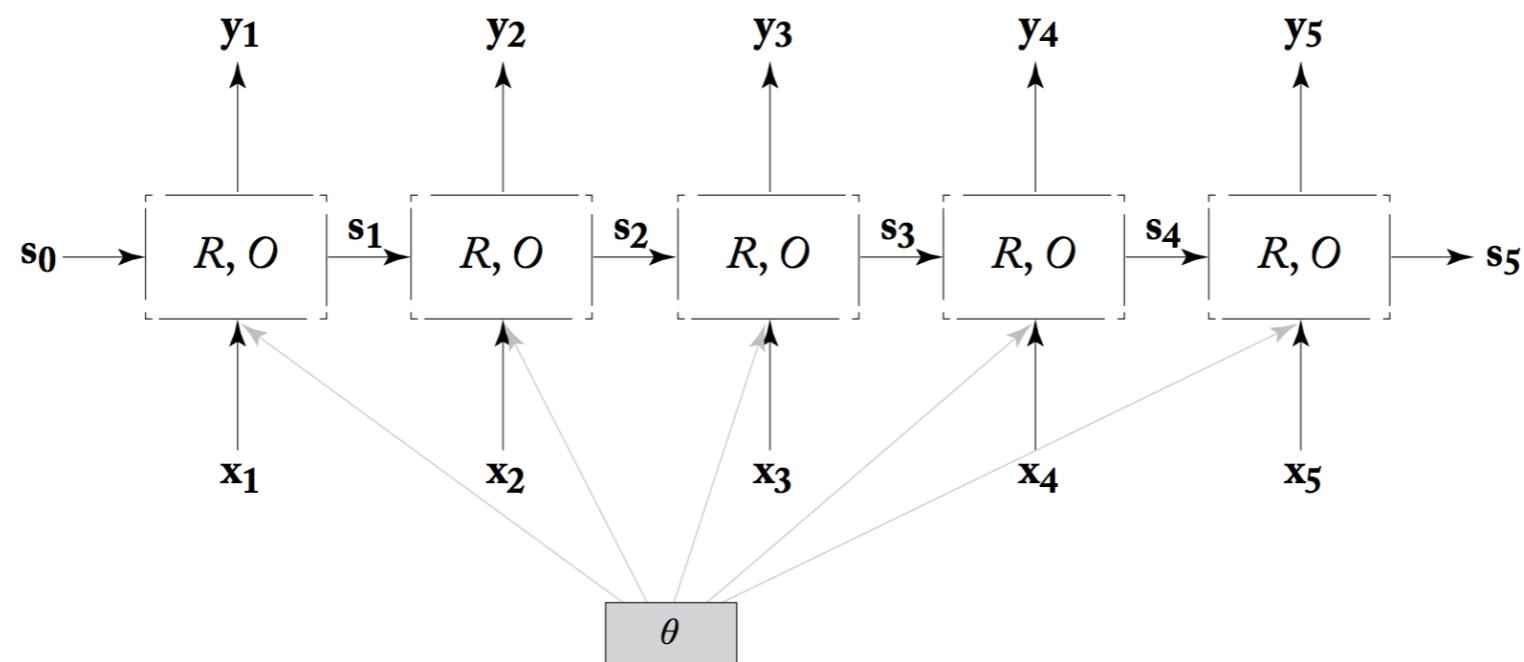
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_2}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_3}}{\partial W^s}$$

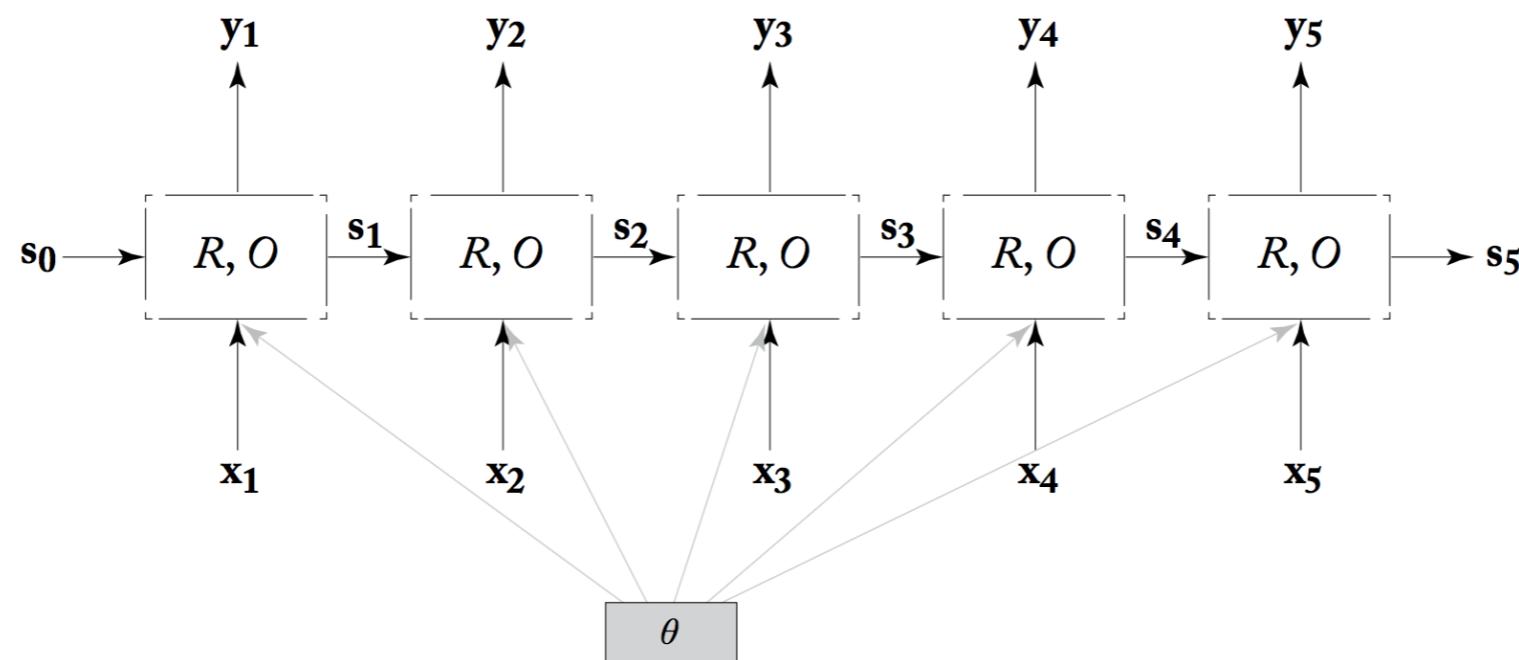
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_2}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_3}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_4}}{\partial W^s}$$

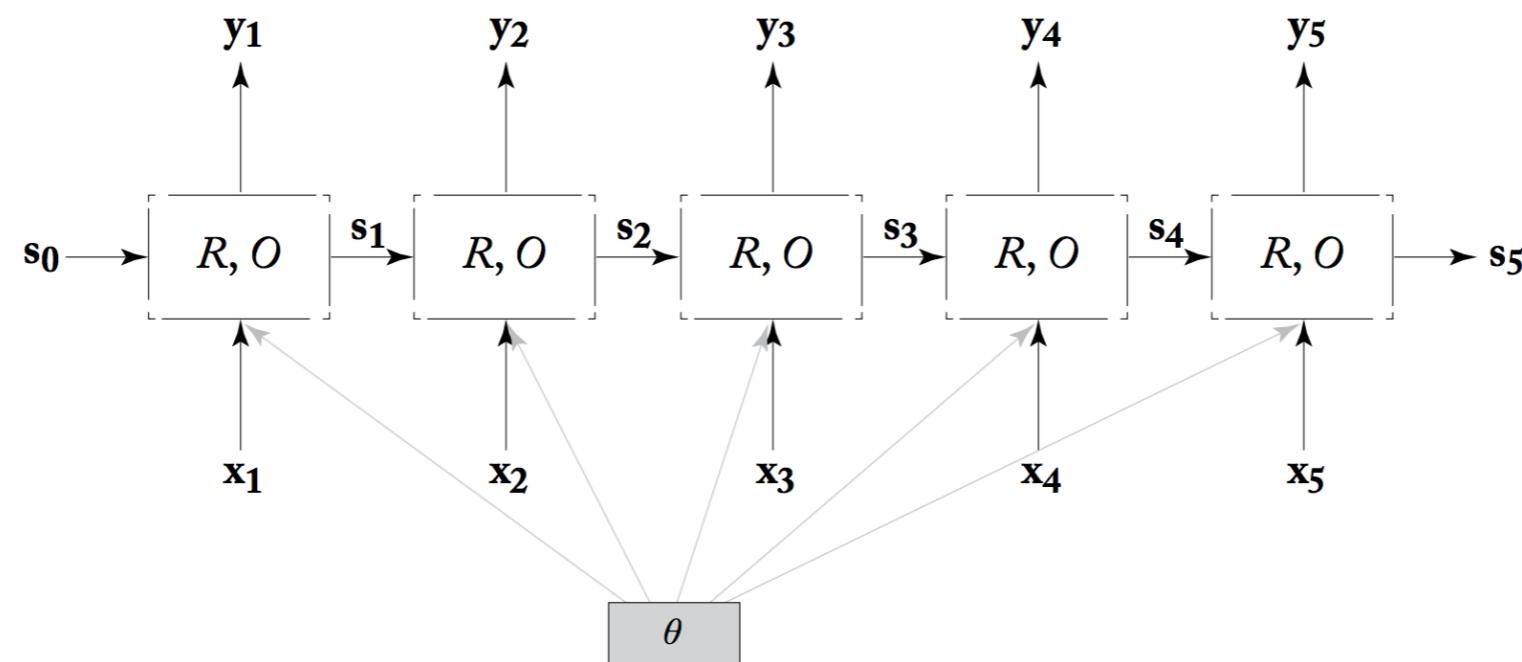
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_2}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_3}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_4}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_5}}{\partial W^s}$$

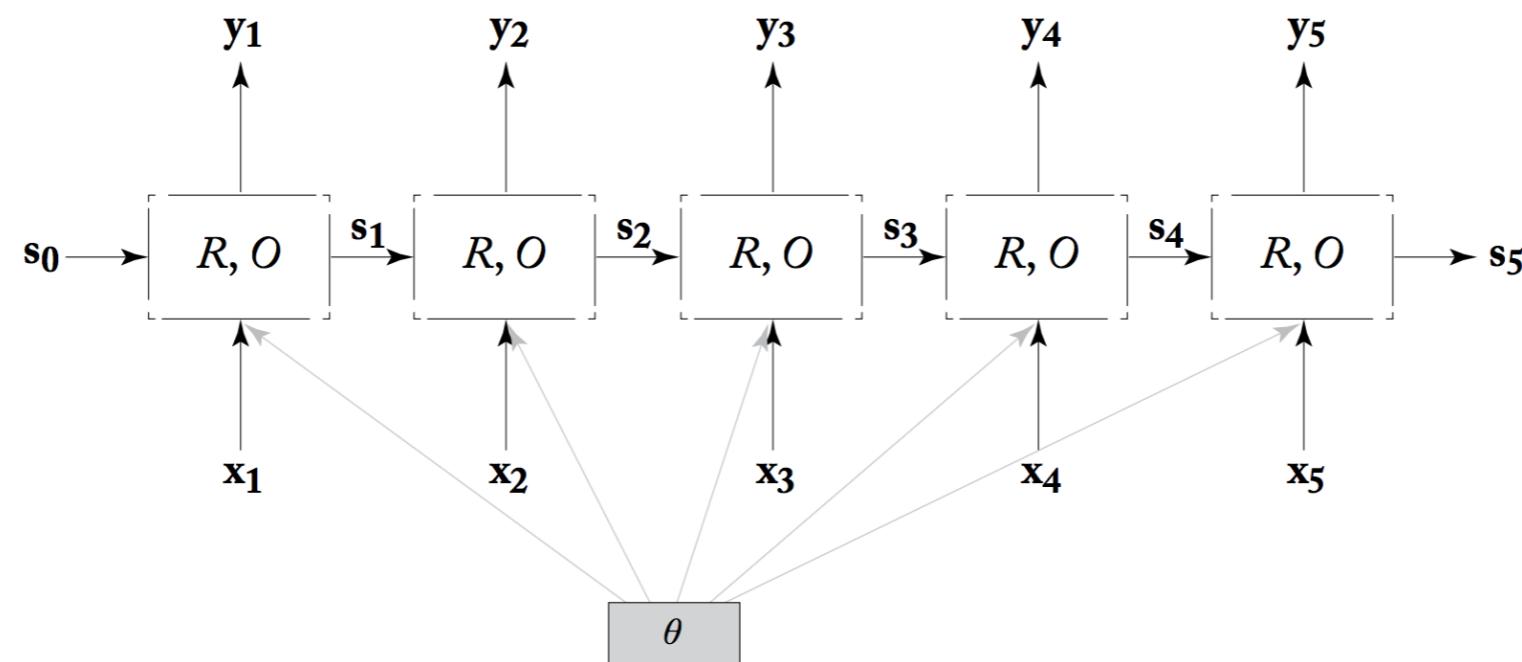
we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.

$$\frac{\partial L(\theta)_{y_1}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_2}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_3}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_4}}{\partial W^s} \quad \frac{\partial L(\theta)_{y_5}}{\partial W^s}$$

we tried to prepare residents



- Training here is standard **backpropagation**, taking the derivative of the loss we incur at step t with respect to the parameters we want to update.
 - You might see this called **backpropagation through time** (BPTT)

Generation

- As we sample, the words we generate form the new context we condition on

Generation



- As we sample, the words we generate form the new context we condition on

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The
START	The	

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The
START	The	dog

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The
START	The	dog
The	dog	

Generation

- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The
START	The	dog
The	dog	walked

Generation

- As we sample, the words we generate form the new context we condition on

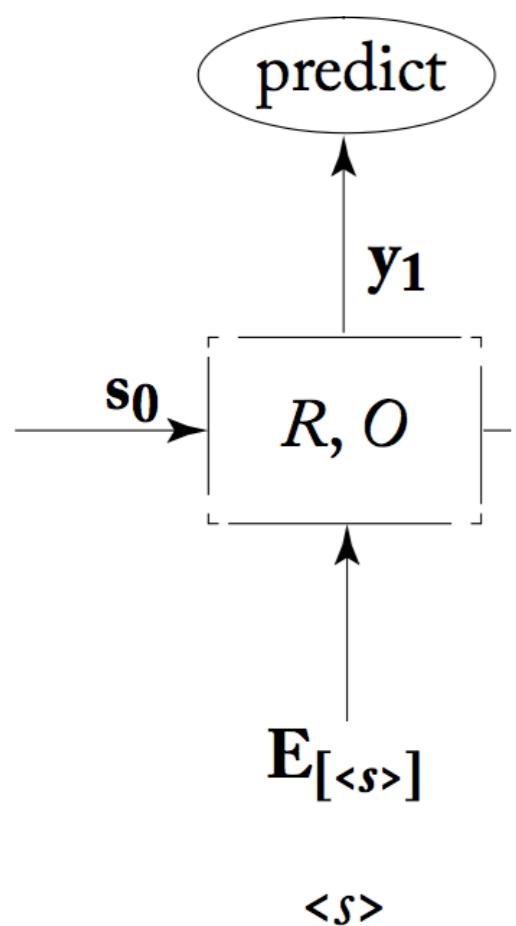
context1	context2	generated word
START	START	The
START	The	dog
The	dog	walked
dog	walked	

Generation

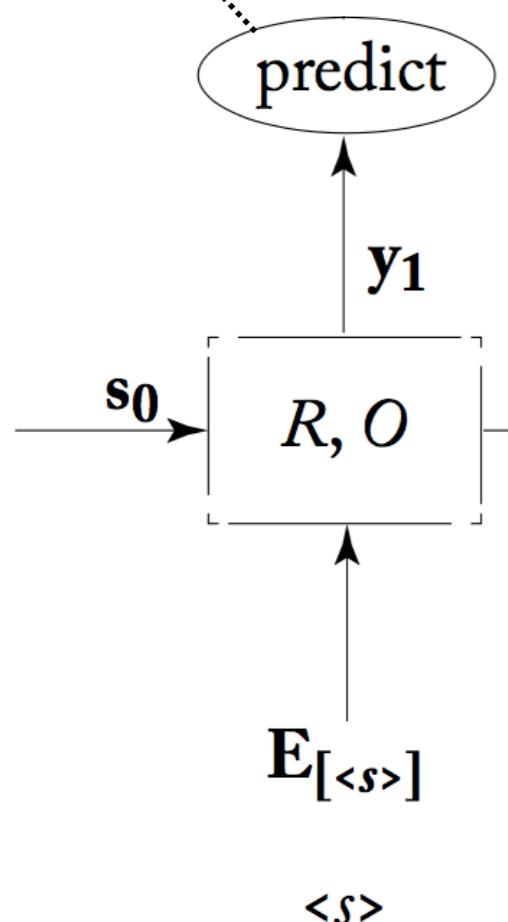
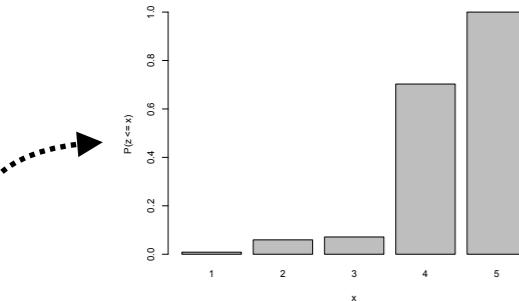
- As we sample, the words we generate form the new context we condition on

context1	context2	generated word
START	START	The
START	The	dog
The	dog	walked
dog	walked	in

Generation

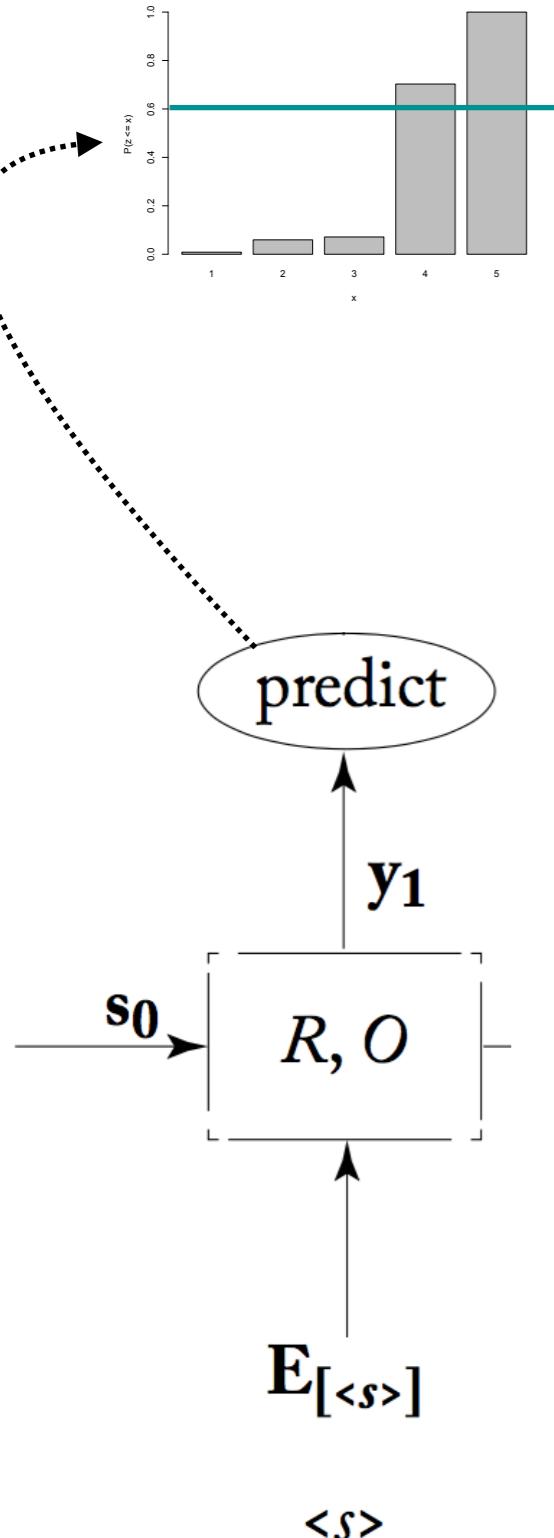


CDF of the softmax predictions, which we can sample from for generation



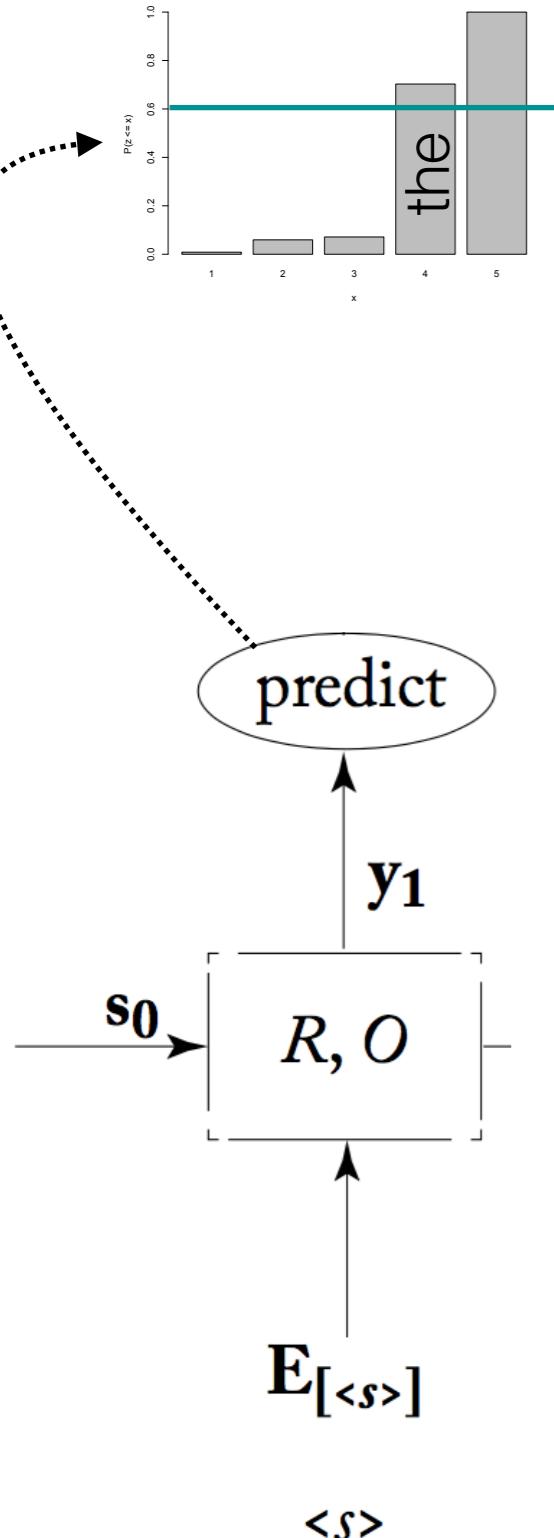
Generation

CDF of the softmax predictions, which we can sample from for generation



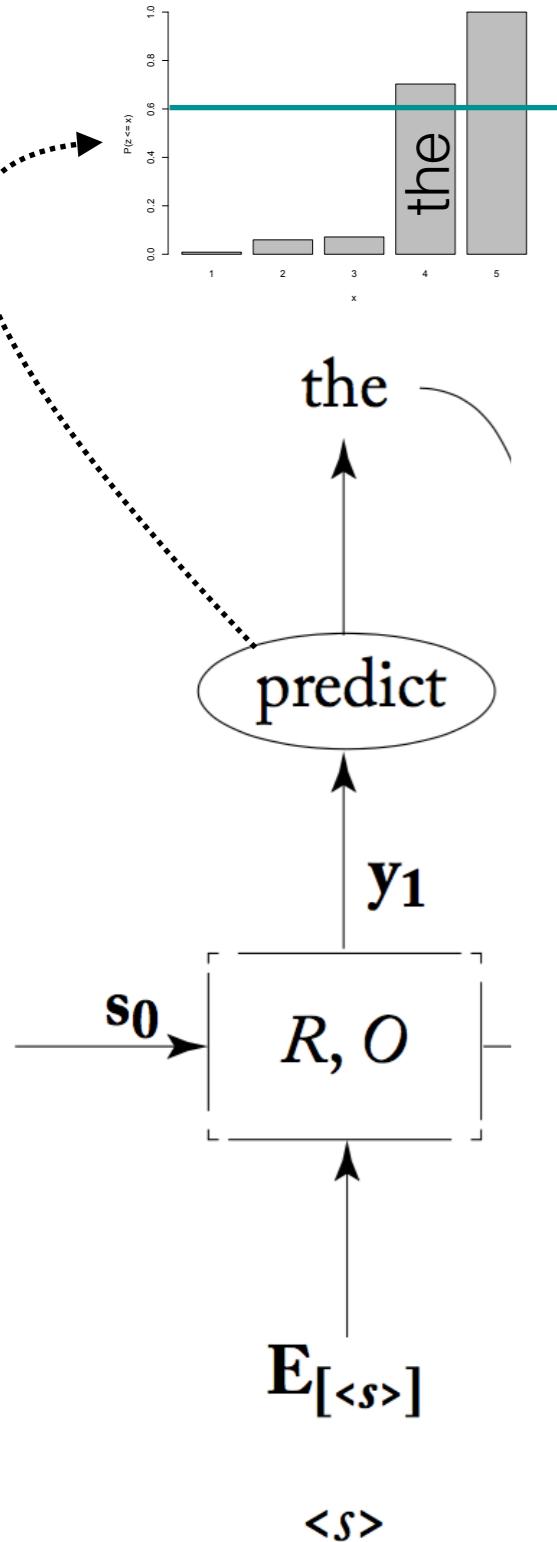
Generation

CDF of the softmax predictions, which we can sample from for generation



Generation

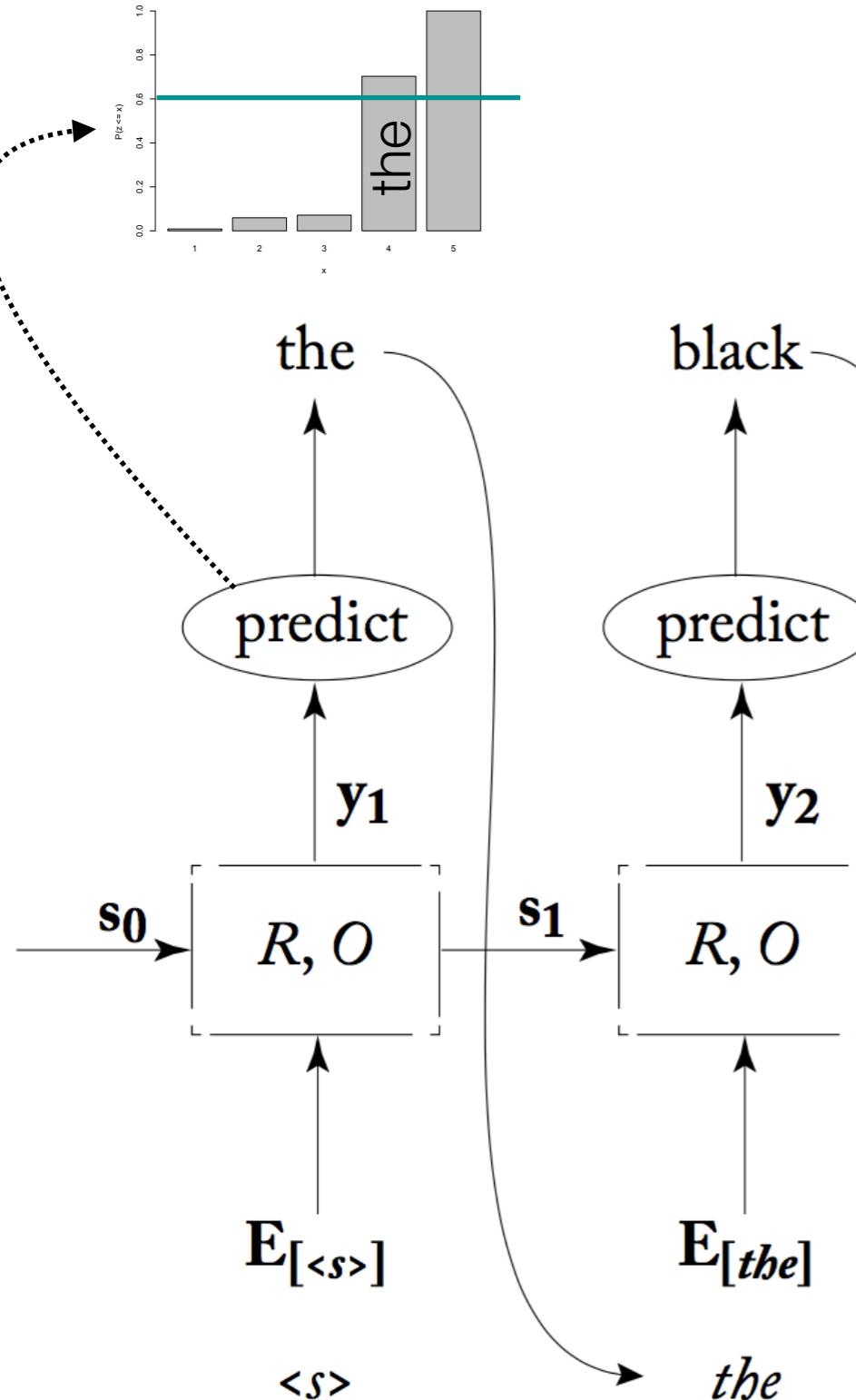
CDF of the softmax predictions, which we can sample from for generation



Generation

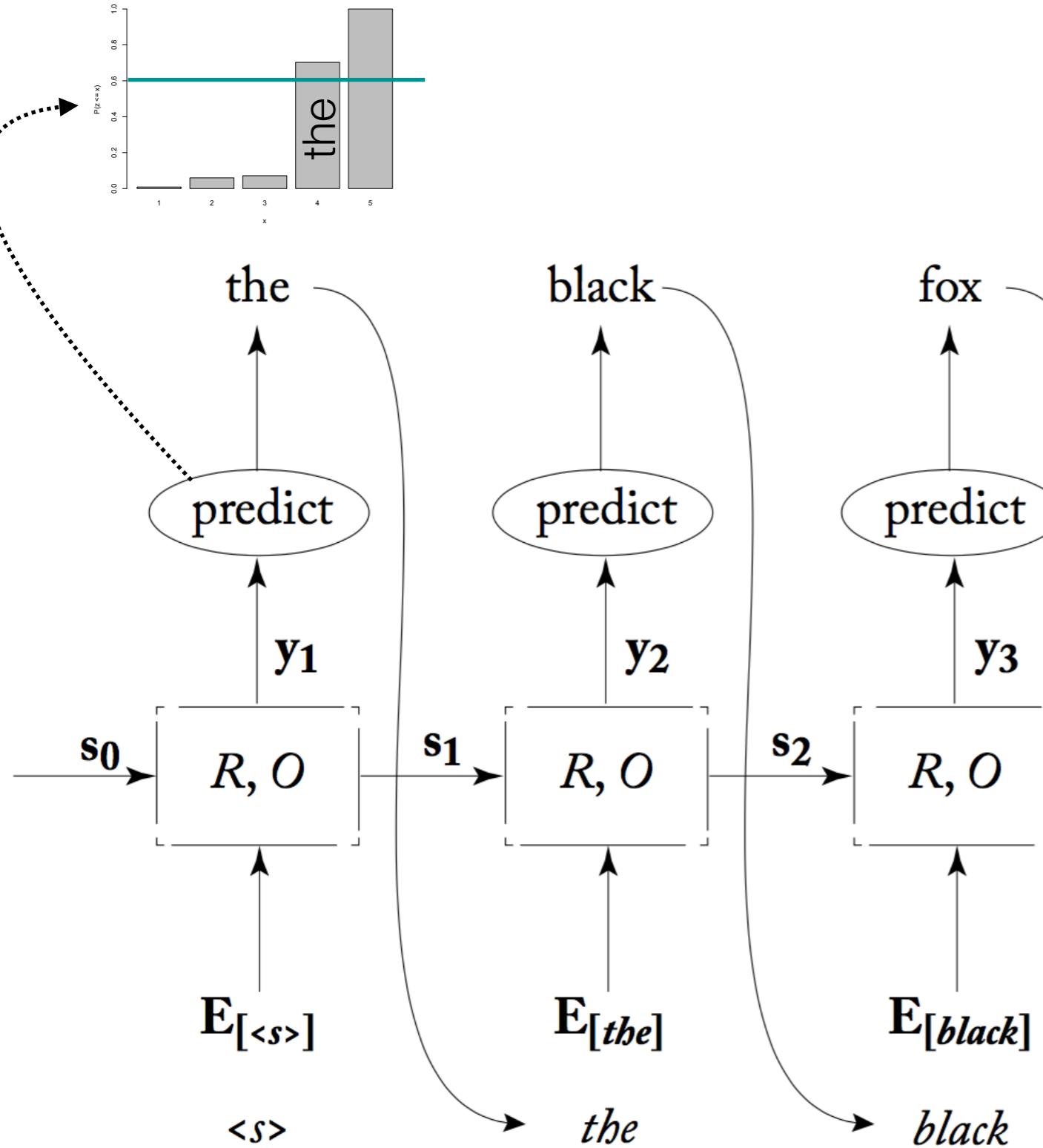
CDF of the softmax predictions, which we can sample from for generation

Generation



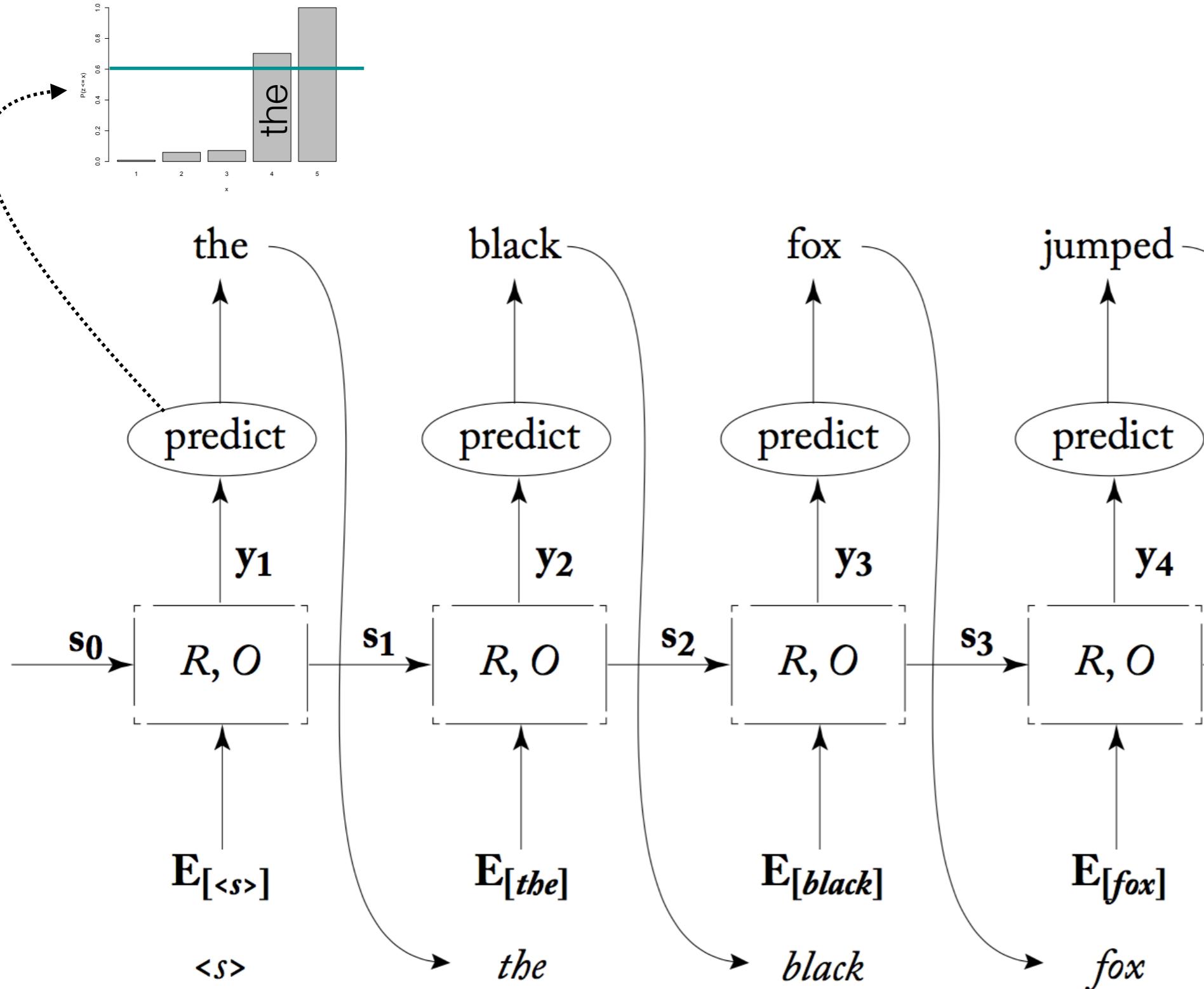
CDF of the softmax predictions, which we can sample from for generation

Generation



CDF of the softmax predictions, which we can sample from for generation

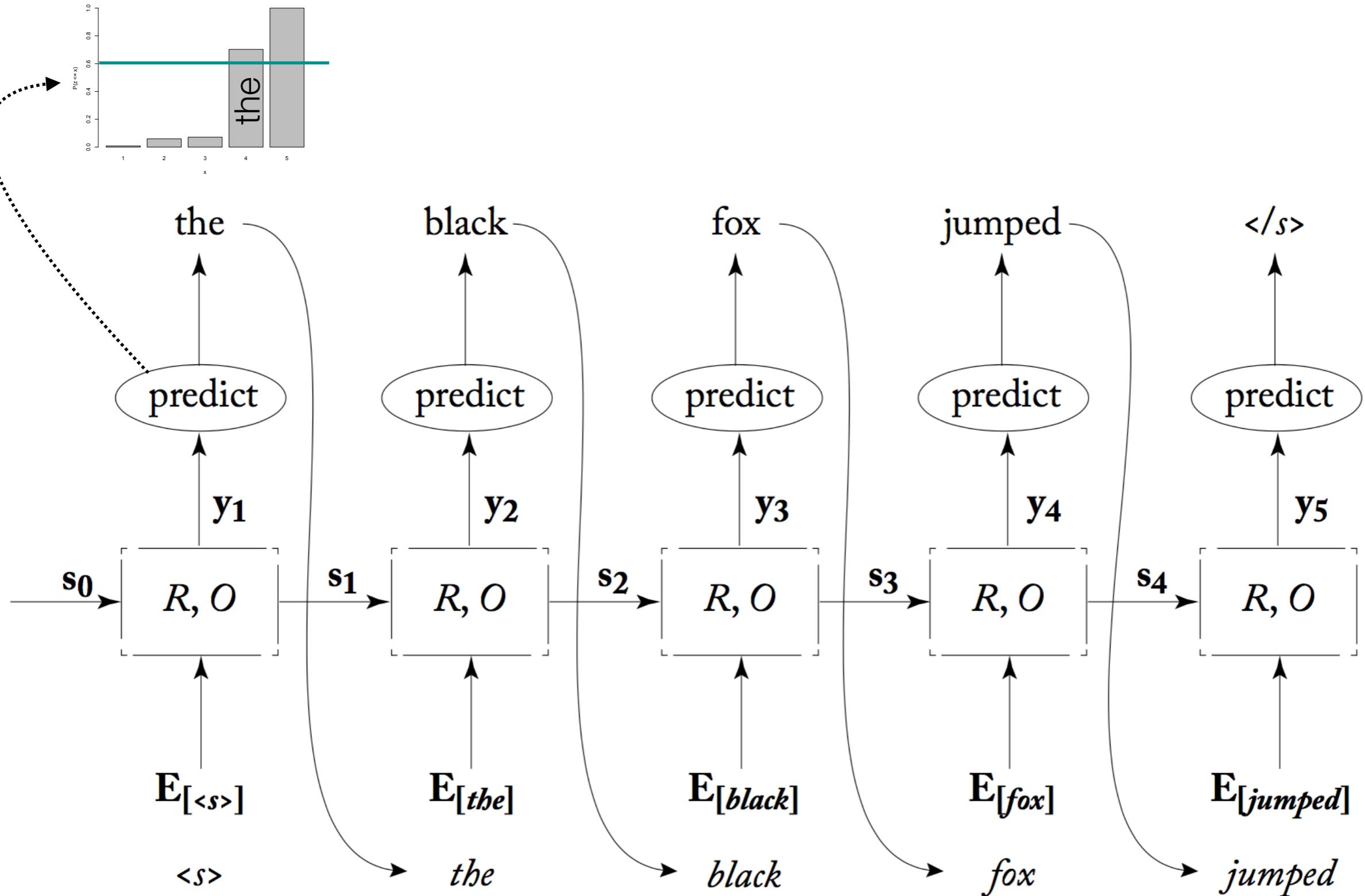
Generation



CDF of the softmax

predictions, which we can sample from for **generation**

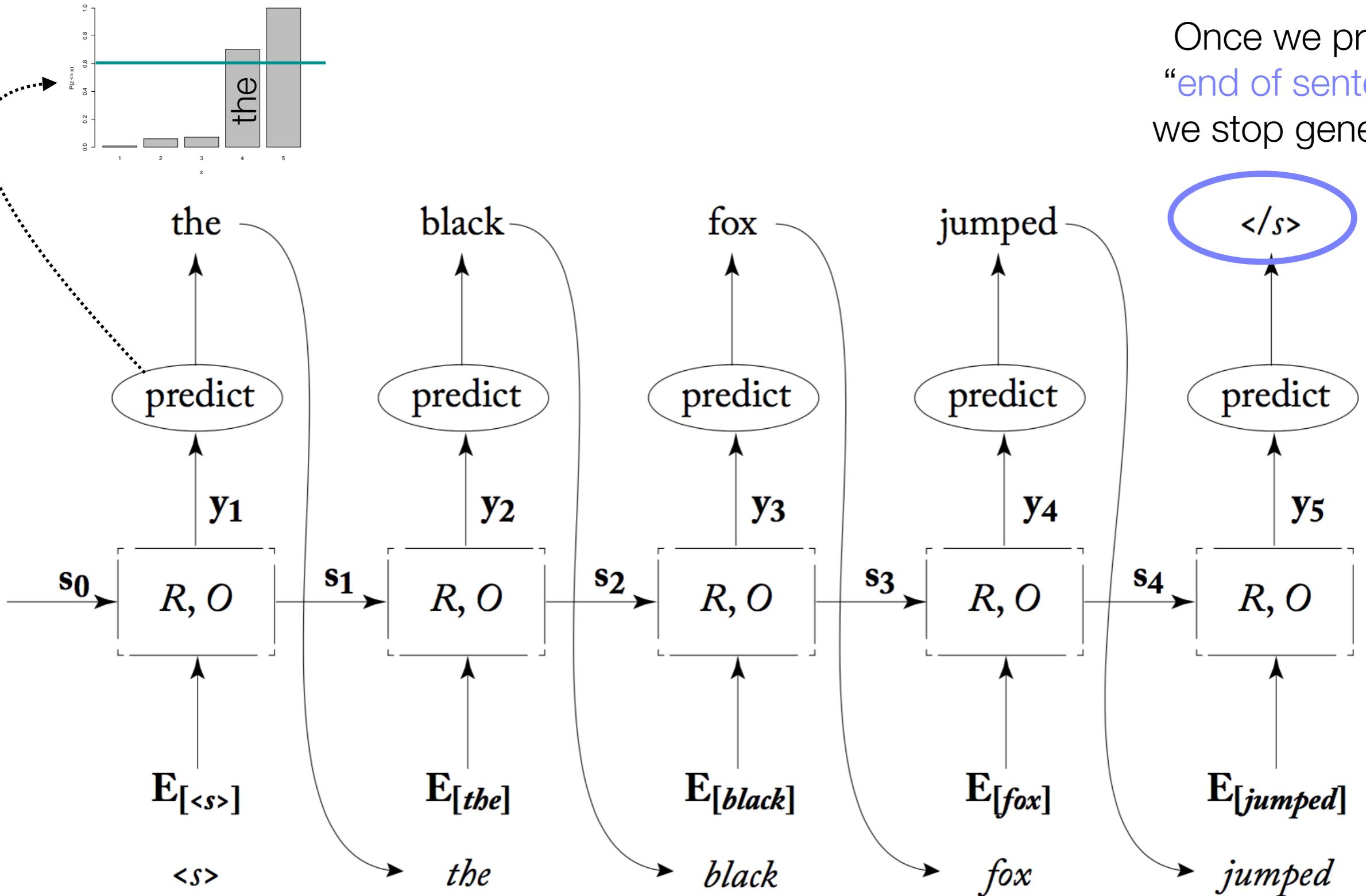
Generation



CDF of the softmax predictions, which we can sample from for generation

Generation

Once we predict “end of sentence” we stop generating



Conditioned generation

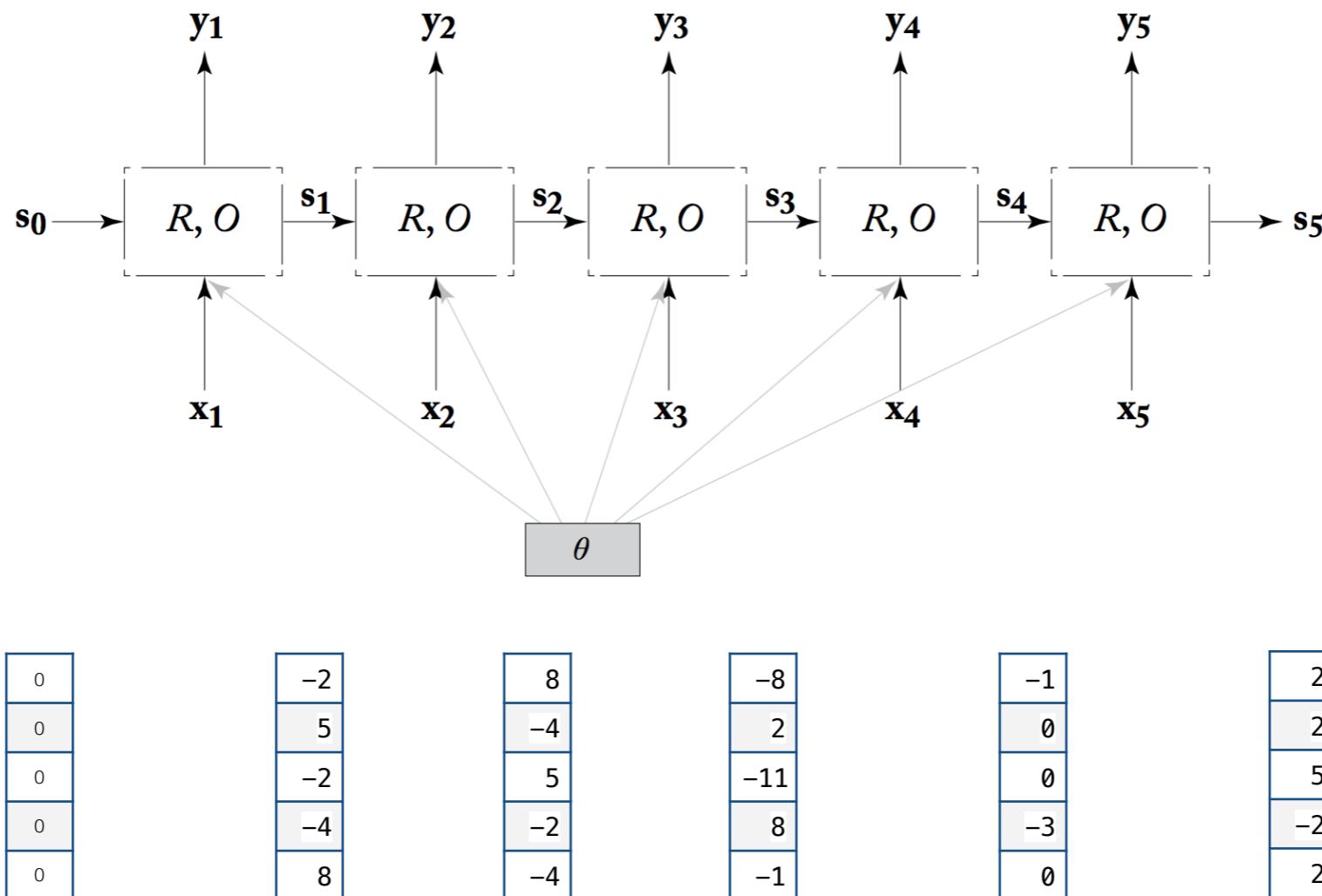
- In a basic RNN, the input at each timestep is a representation of the word at that position

$$s_i = R(x_i, s_{i-1}) = g(s_{i-1}W^s + \textcolor{magenta}{x}_i W^x + b)$$

- But we can also condition on any arbitrary context (topic, author, date, metadata, dialect, etc.)

$$s_i = R(x_i, s_{i-1}) = g(s_{i-1}W^s + [\textcolor{magenta}{x}_i; \textcolor{magenta}{c}] W^x + b)$$

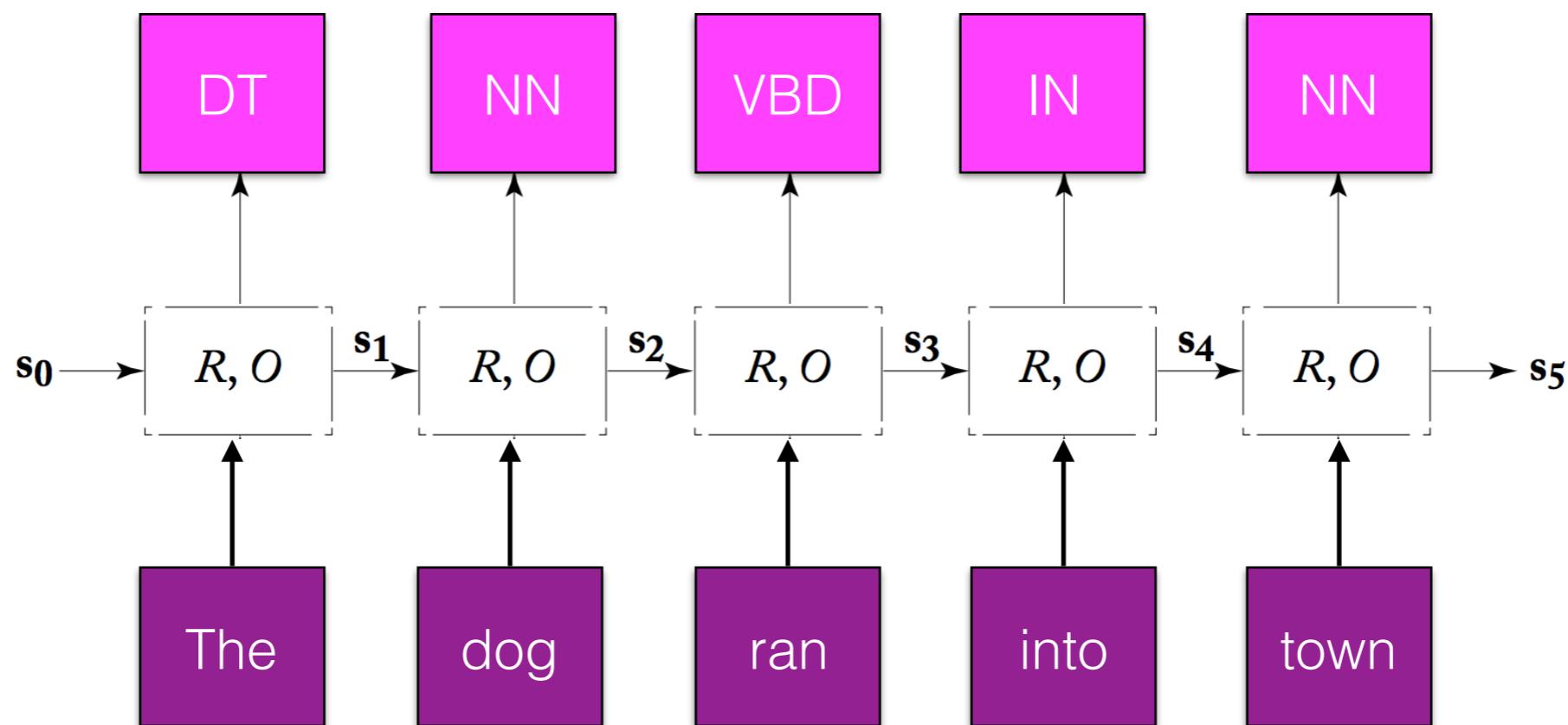
we tried to prepare residents



Each state i encodes information seen until time i
and its structure is optimized to predict the next word

Recurrent neural network

- RNNs for POS tagging, predict the tag from \mathcal{Y} conditioned on the context



RNNs for POS

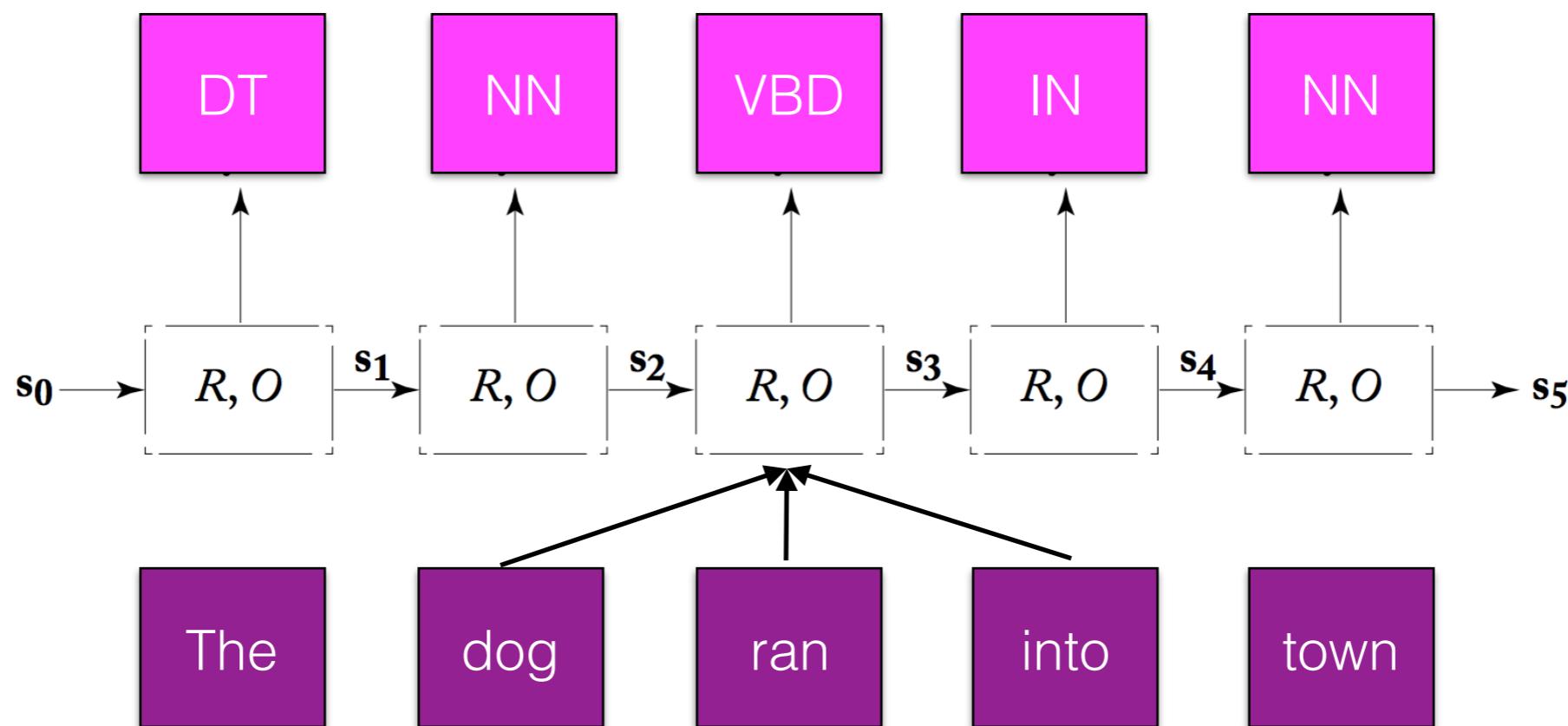
NN TO VB

will to fight

- To make a prediction for y_t , RNNs condition on all input seen through time t (x_1, \dots, x_t)
- But knowing something about the **future** can help (x_{t+1}, \dots, x_n)

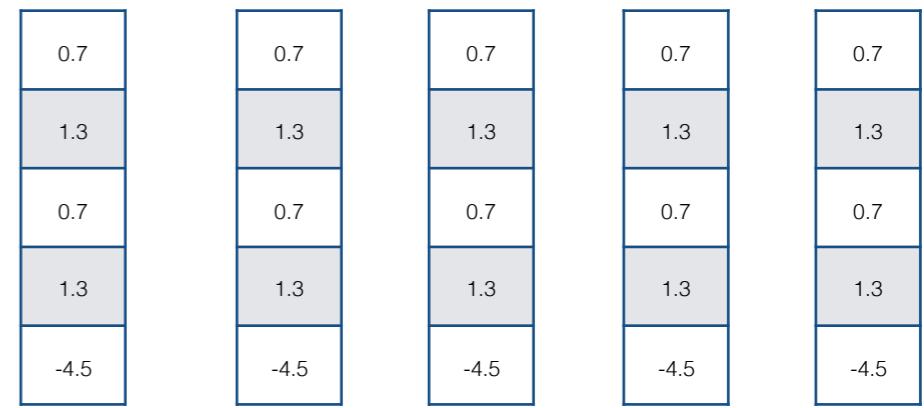
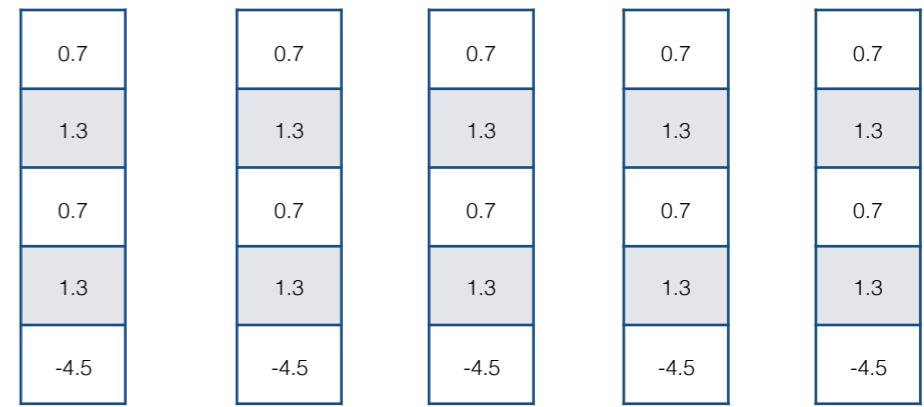
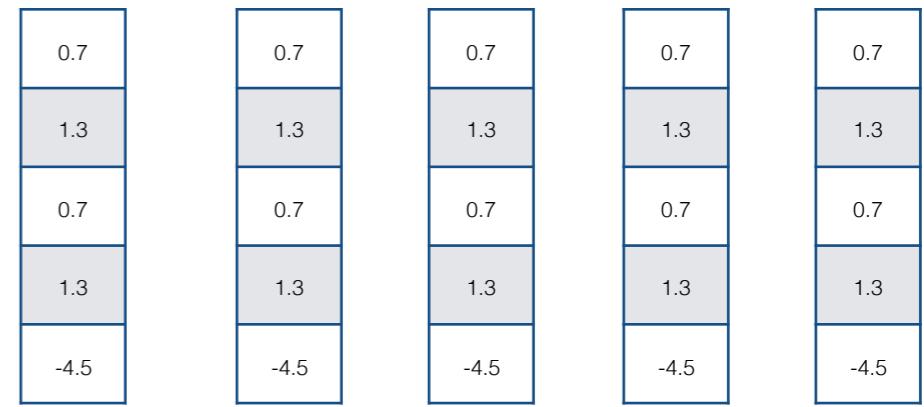
Recurrent neural network

- The simplest thing we could do is condition on a window of inputs



Recurrent neural network

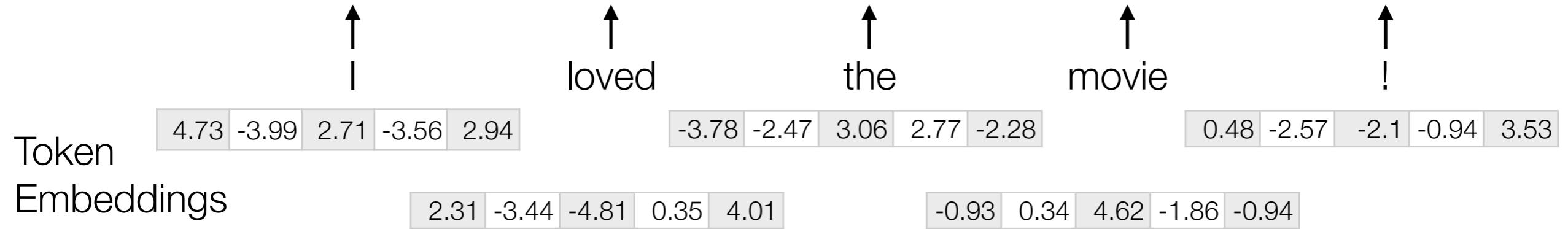
- e.g. 5 time steps, input =
 $[x_{i-1}, x_i, x_{i+1}]$
- Quite similar to what we do
with logistic regression



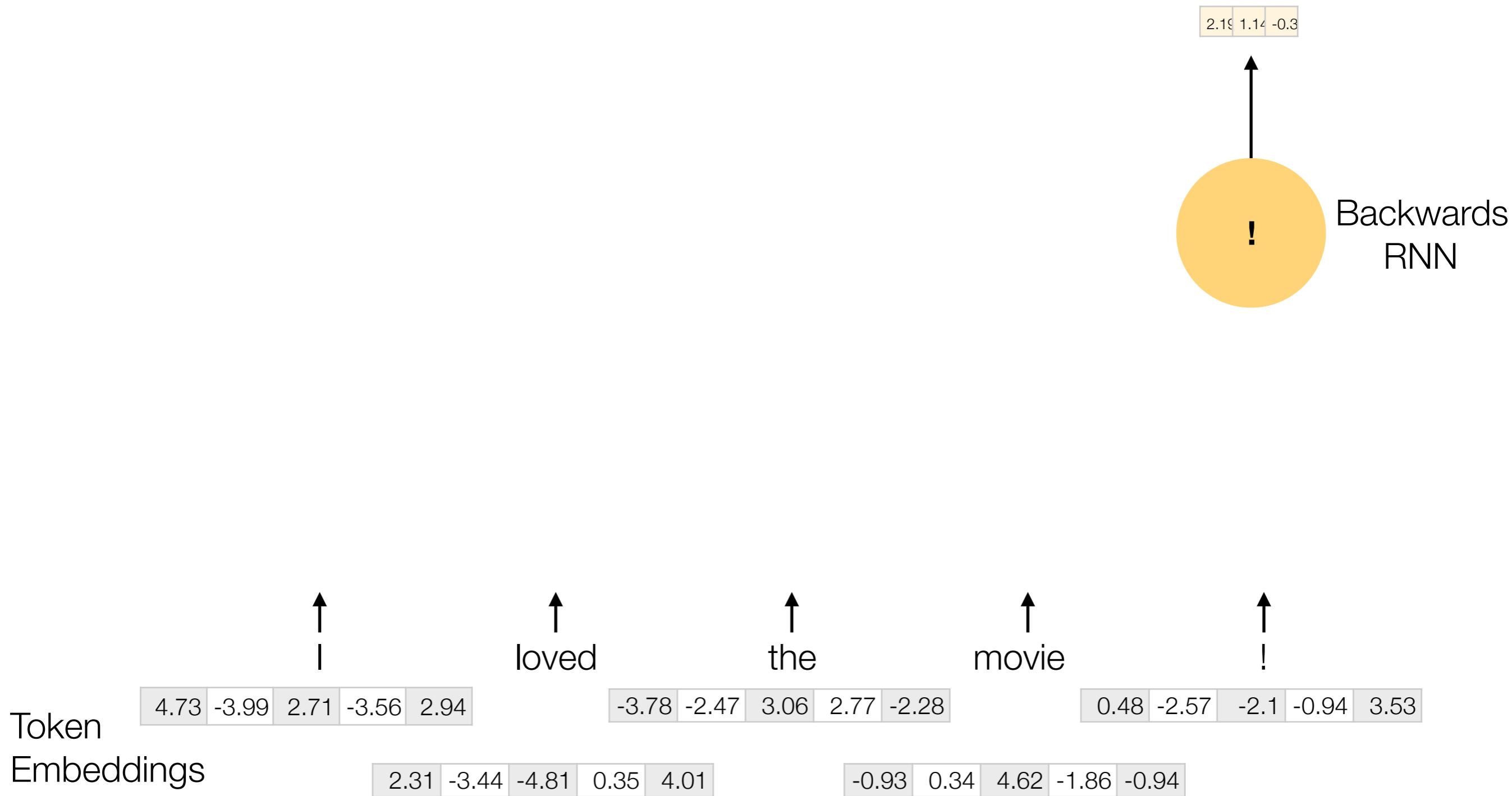
Bidirectional RNN

- 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

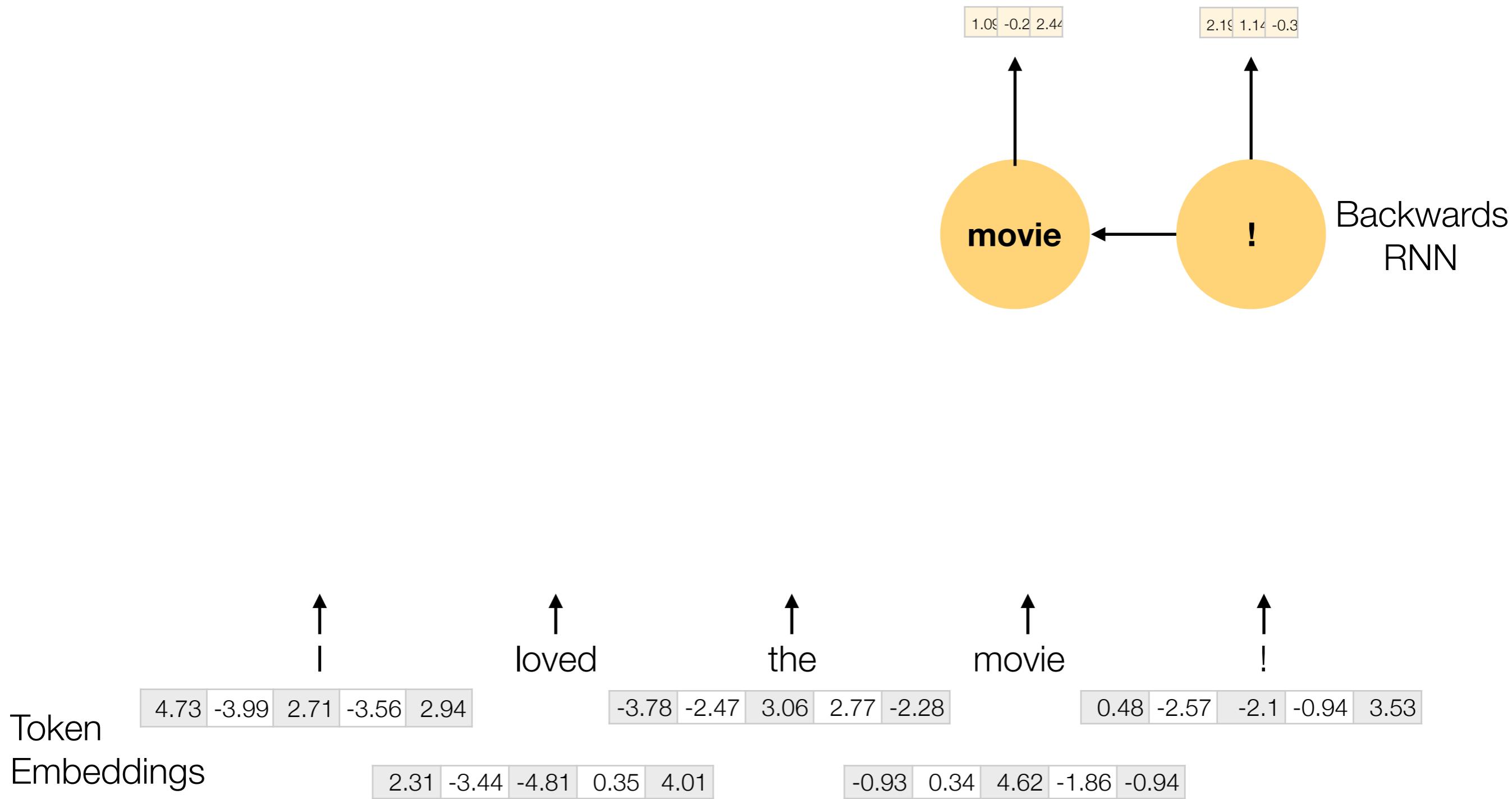
Bidirectional RNN



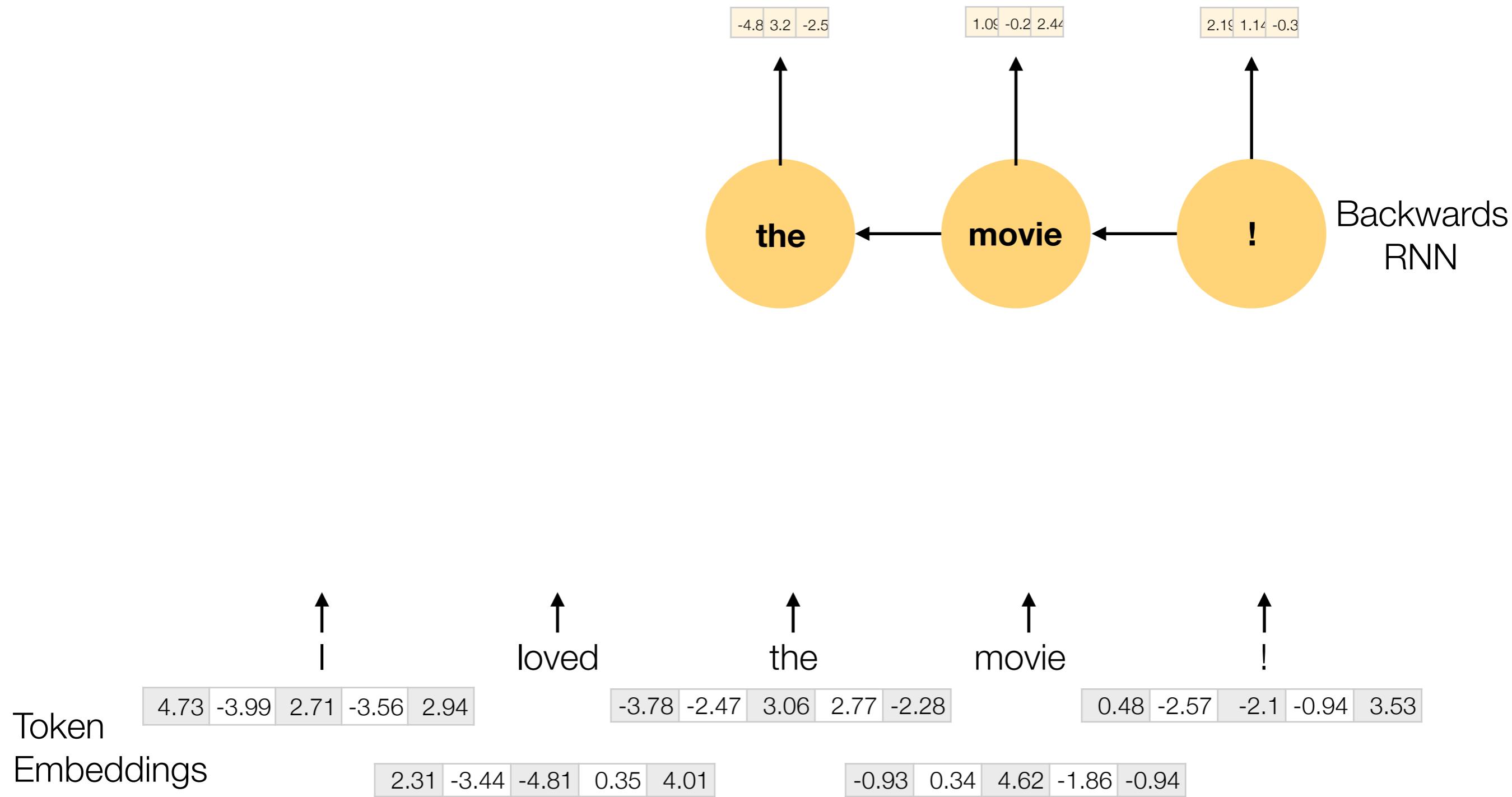
Bidirectional RNN



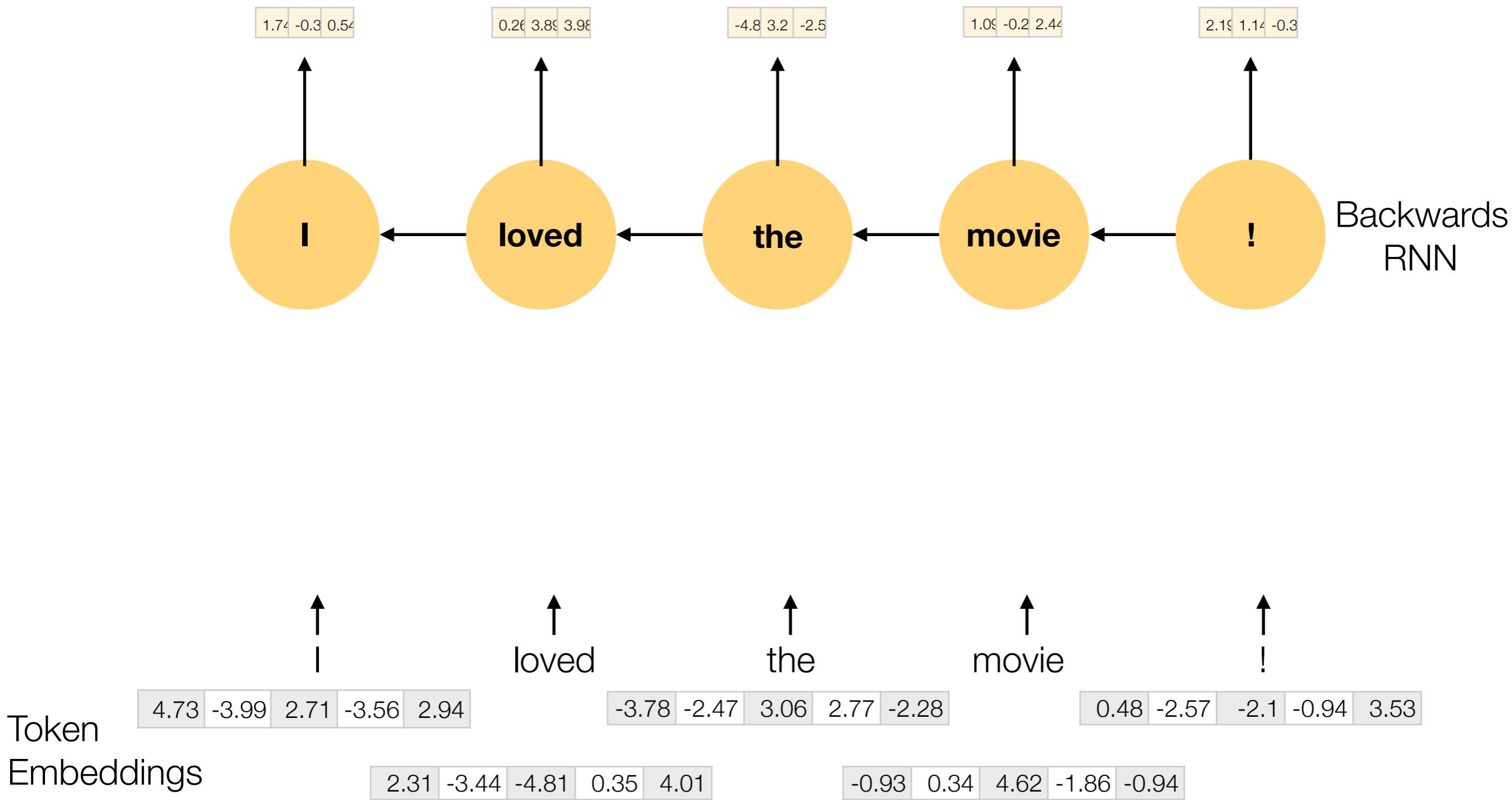
Bidirectional RNN



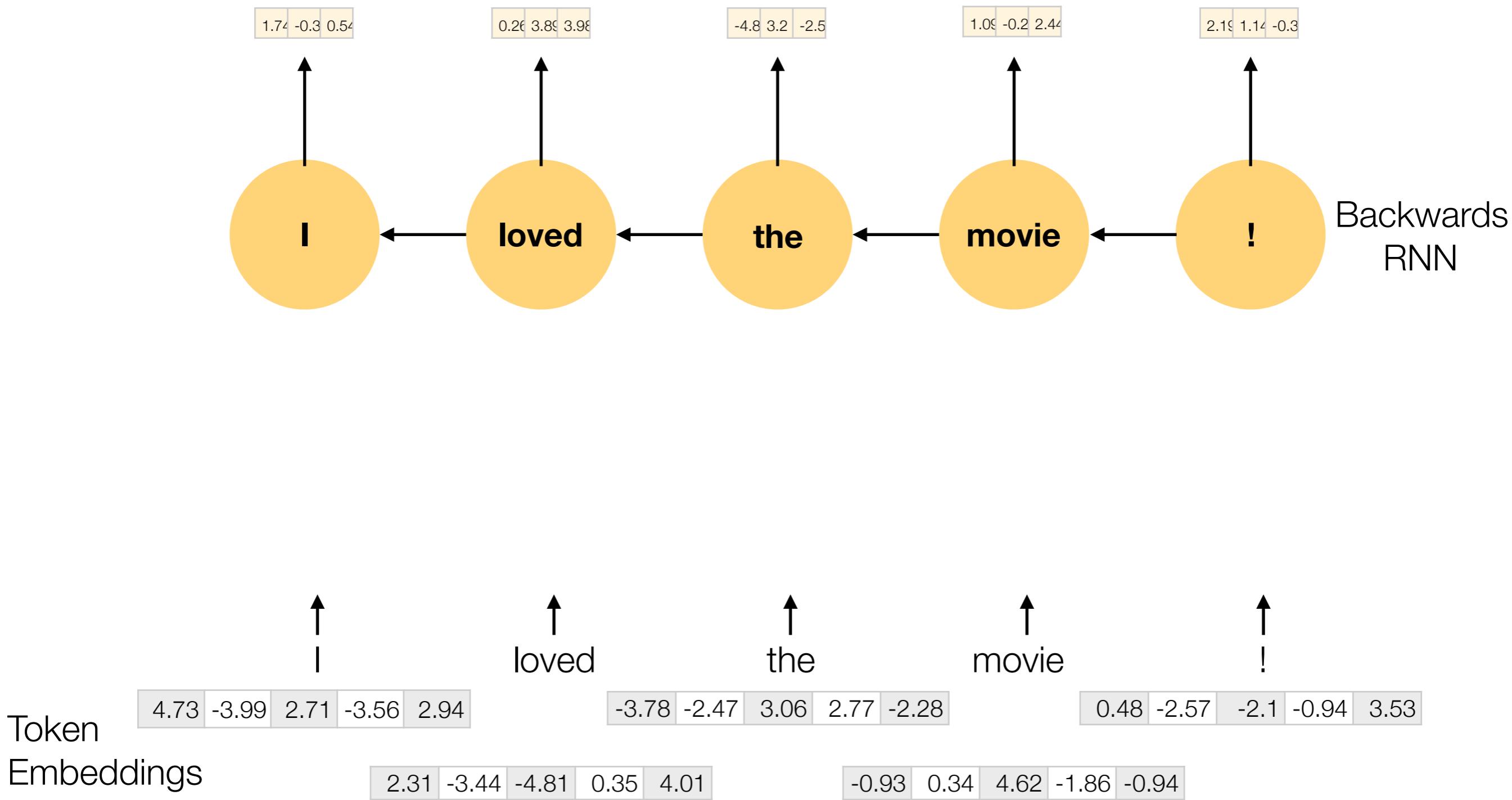
Bidirectional RNN



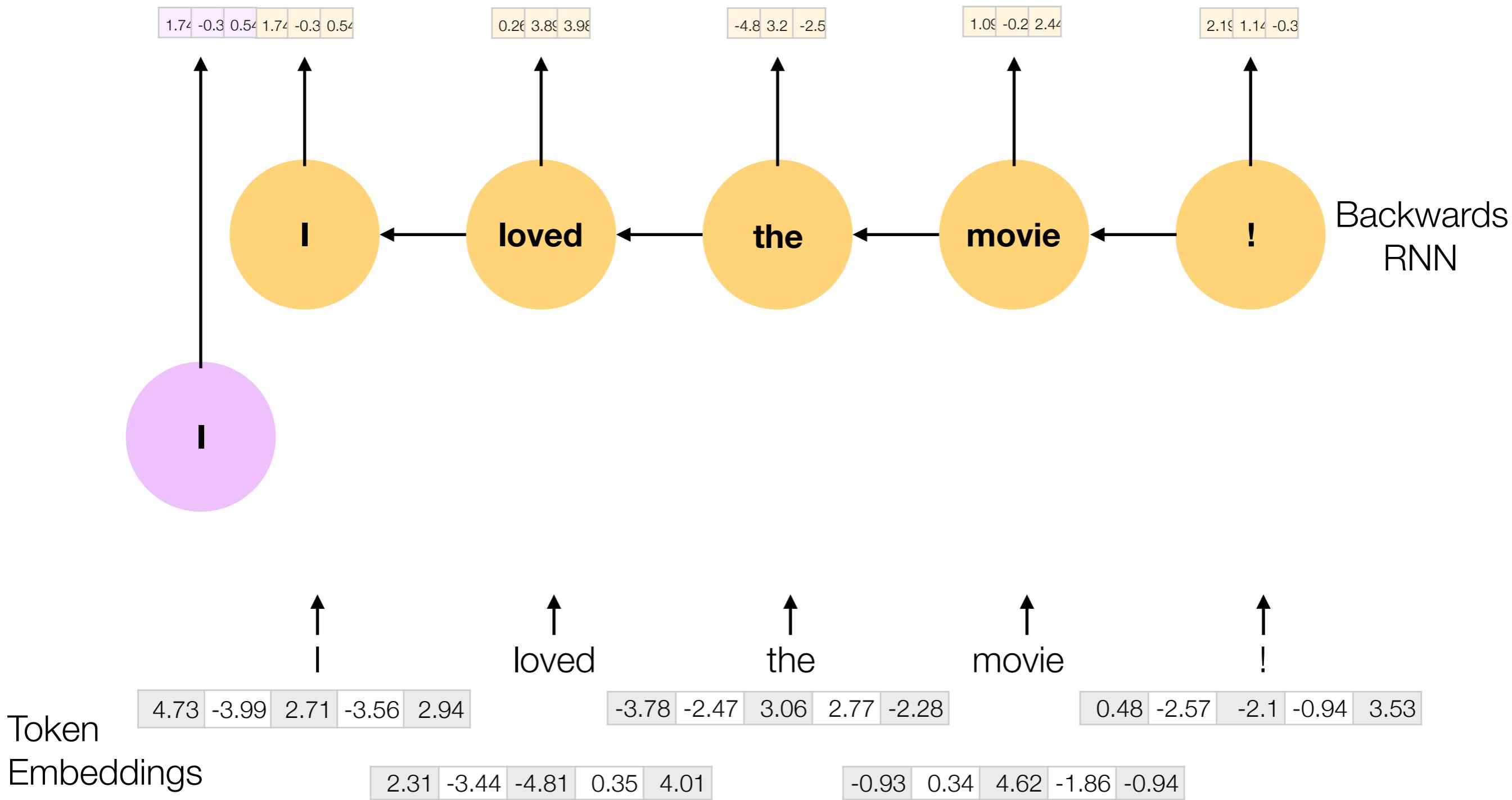
Bidirectional RNN



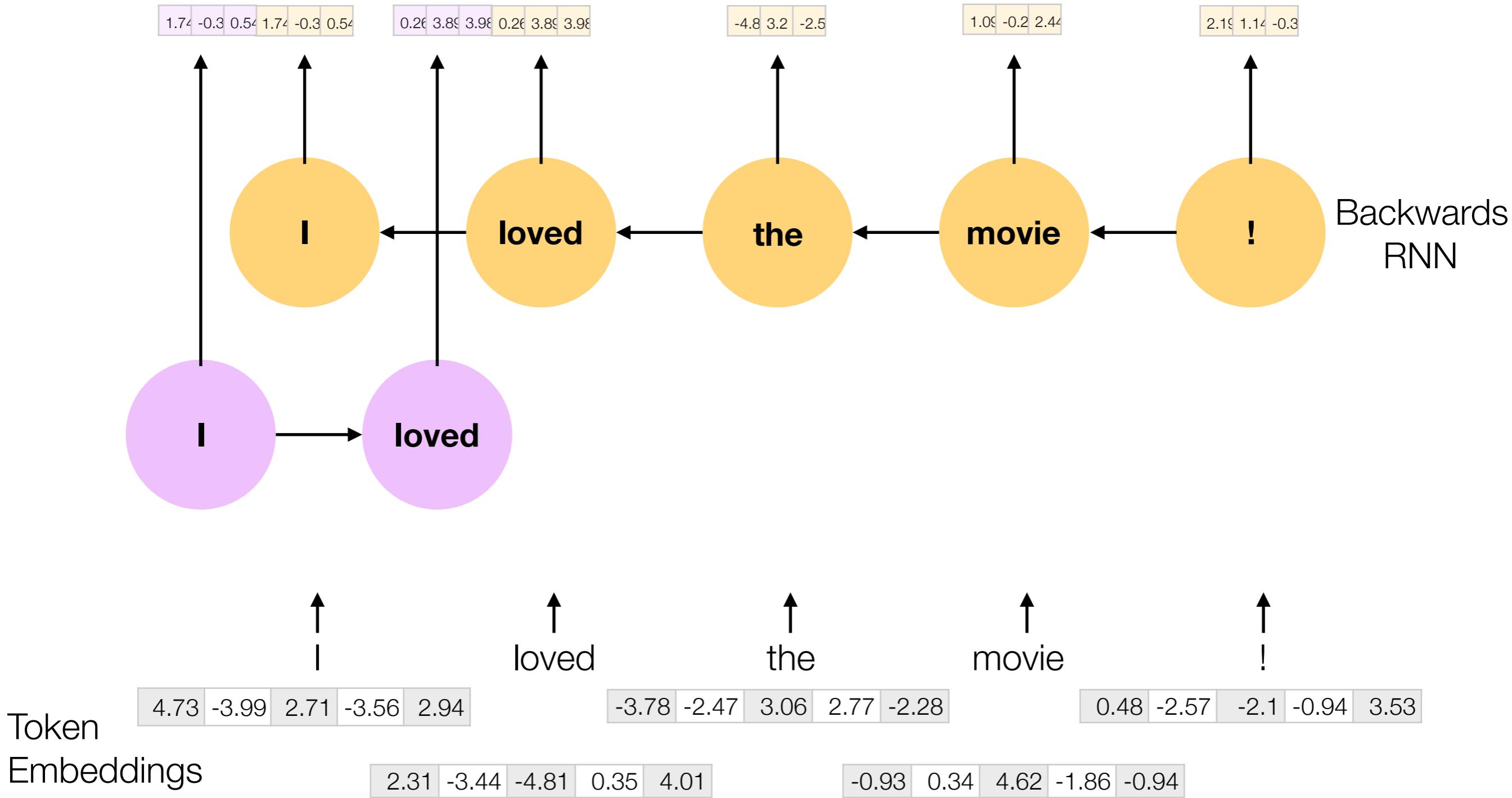
Bidirectional RNN



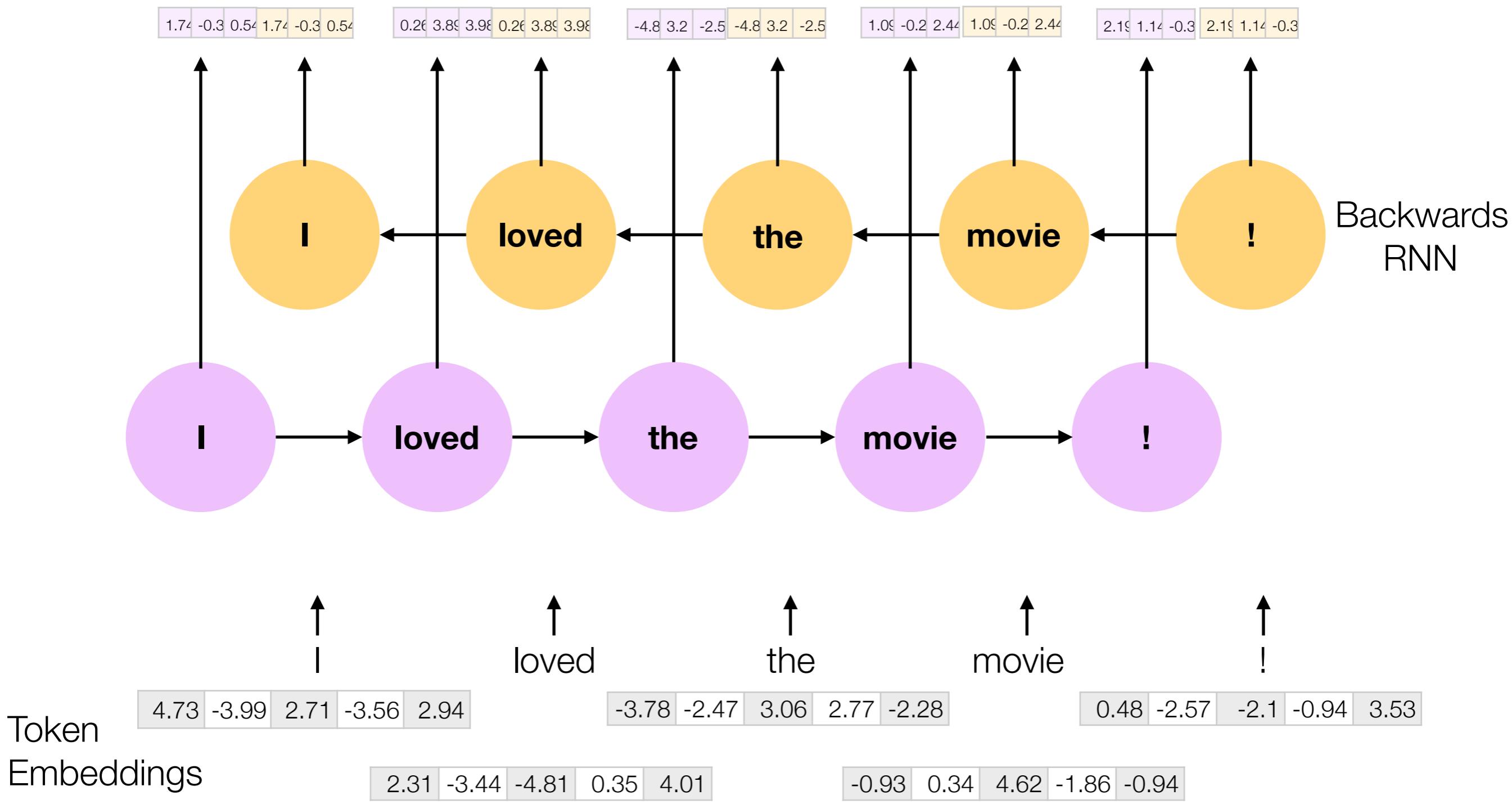
Bidirectional RNN



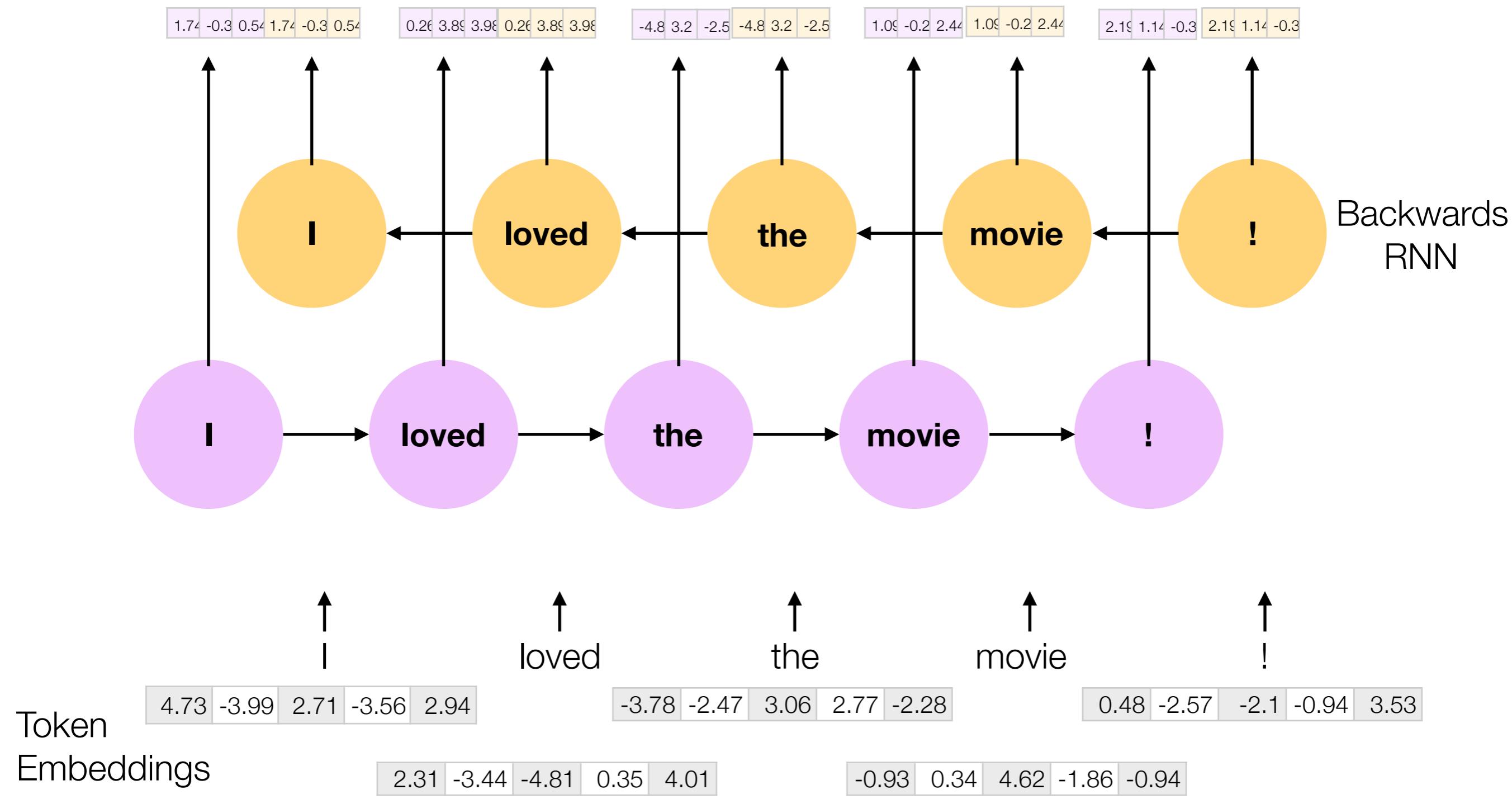
Bidirectional RNN



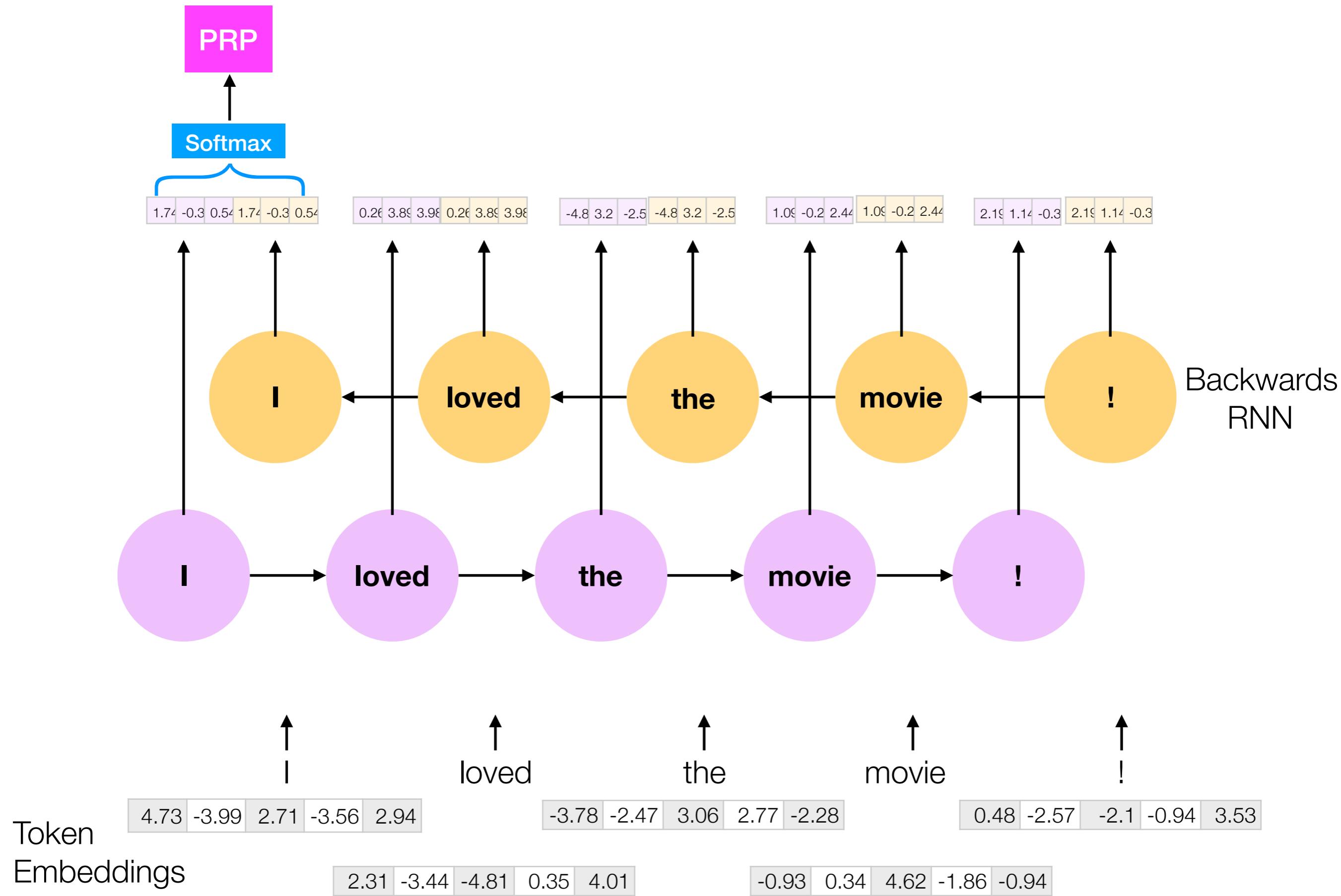
Bidirectional RNN



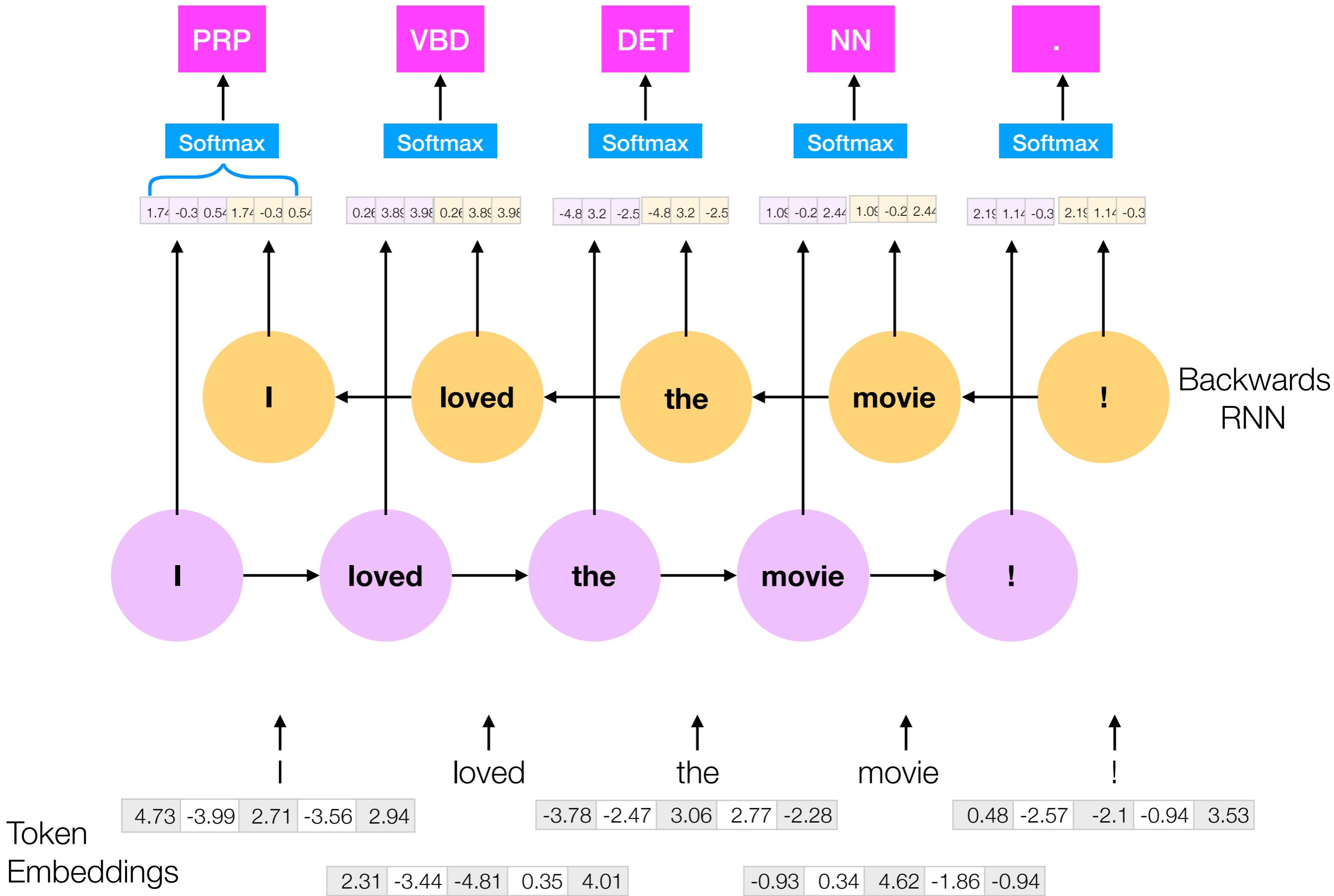
Bidirectional RNN

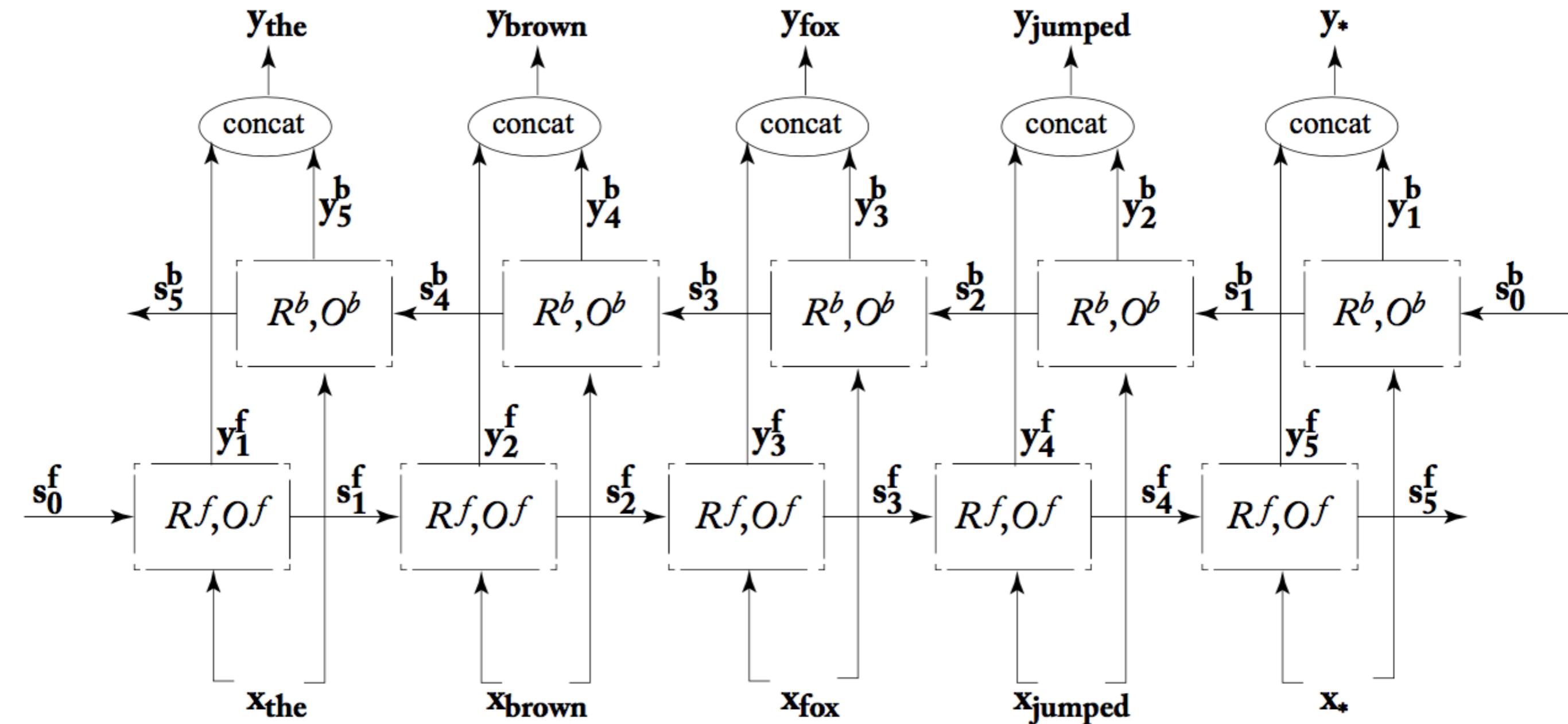


Bidirectional RNN



Bidirectional RNN





Training BiRNNs

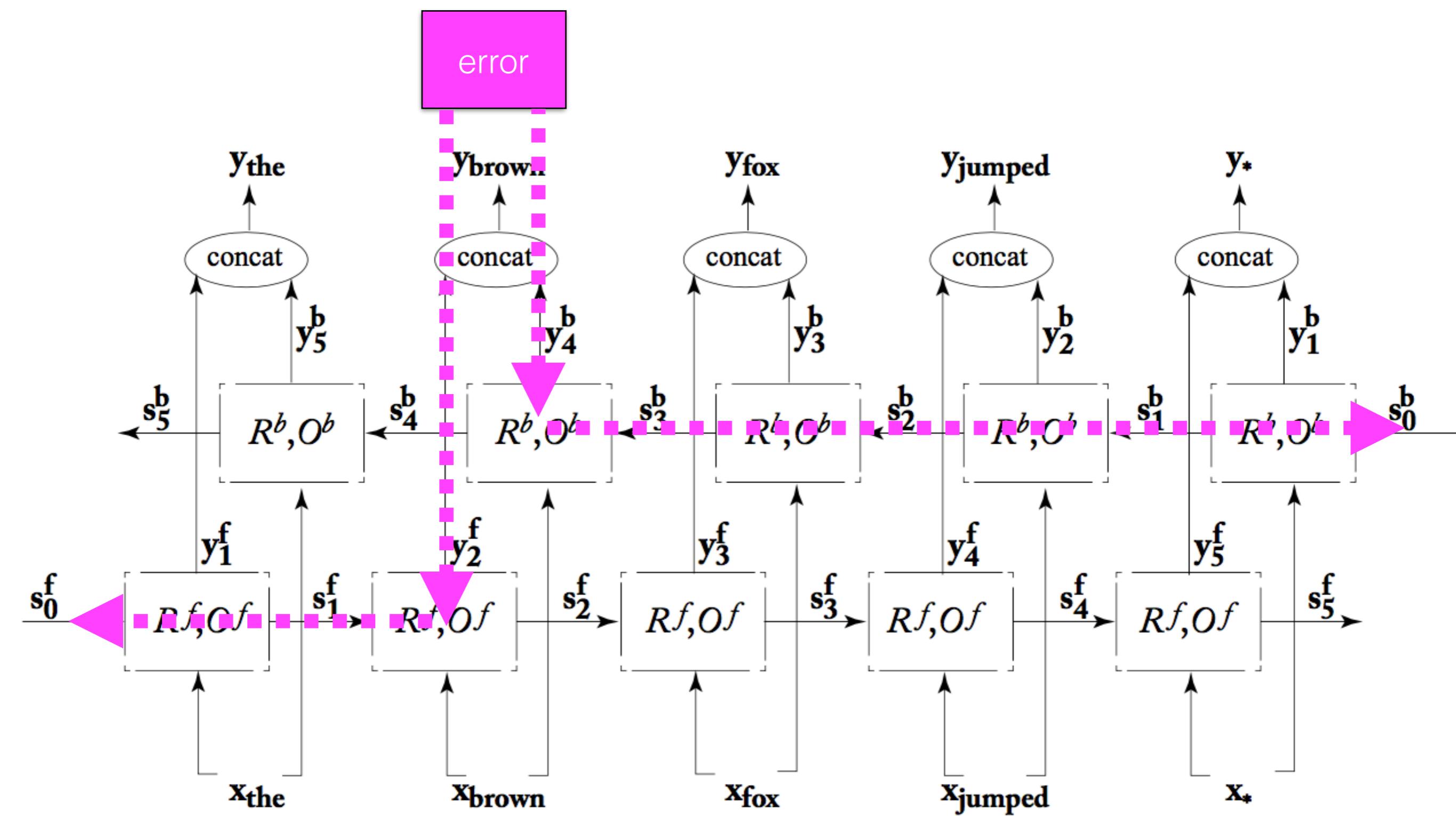
- Given this definition of an BiRNN:

$$s_b^i = R_b(x^i, s_b^{i+1}) = g(s_b^{i+1} \mathbf{W}_b^s + x^i \mathbf{W}_b^x + \mathbf{b}_b)$$

$$s_f^i = R_f(x^i, s_f^{i-1}) = g(s_f^{i-1} \mathbf{W}_f^s + x^i \mathbf{W}_f^x + \mathbf{b}_f)$$

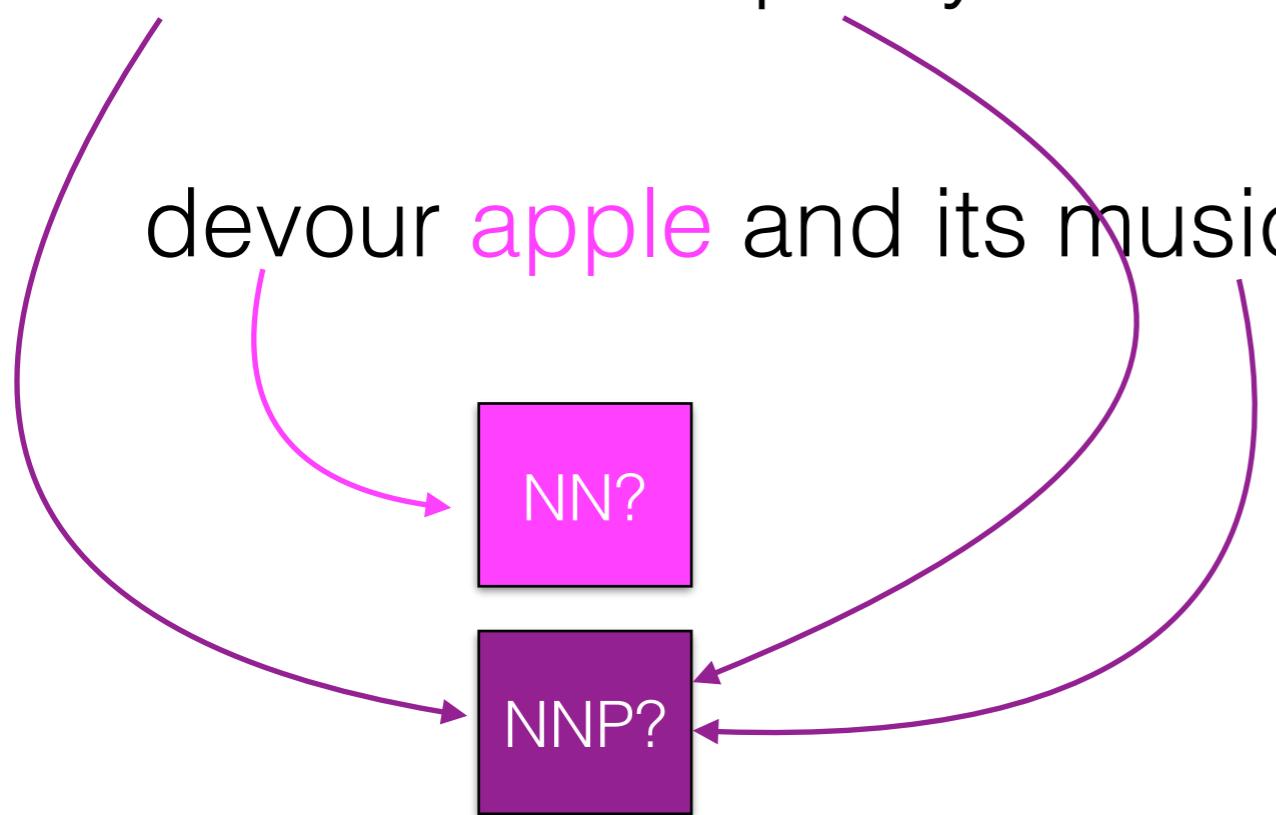
$$y_i = \text{softmax} \left([s_f^i; s_b^i] \mathbf{W}^o + \mathbf{b}^o \right)$$

- We have 8 sets of parameters to learn (3 for each RNN + 2 for the final layer)



RNNs for POS

amazon and spotify's streaming services are going to
devour **apple** and its music purchasing model



RNNs for POS

amazon and spotify's streaming services are going to
devour apple and its music purchasing model

Prediction:

Can the information from far away get to the time step that needs it?

Training:

Can error reach that far back during backpropagation?

Necessary information can be far away in the sentence

I **was** on vacation

when I read the book

Necessary information can be far away in the sentence

I **was** on vacation

when I **read** the book

Necessary information can
be far away in the sentence

I was on vacation

VBP?

when I read the book

VBZ?

Necessary information can be far away in the sentence

- | | |
|----------------------|------|
| I was on vacation | VBP? |
| when I read the book | VBZ? |
| will be on vacation | |

Necessary information can be far away in the sentence

VBP?

VBZ?

I was on vacation when I the book

will be on vacation

Necessary information can be far away in the sentence

I was on vacation [] when I read the book

VBP?

VBZ?

will be on vacation

in this crazy place where...

The diagram illustrates the concept of long-distance dependencies. It shows two sentences. The first sentence contains the verb 'was' (highlighted in pink) and a blank red box. The second sentence contains the verb 'read' (highlighted in pink) and another red box. A curved arrow originates from the word 'will' in the third-person singular present form ('will be') in the first sentence and points to the red box containing 'read' in the second sentence. This visualizes how necessary information can be positioned far apart in a sentence structure.

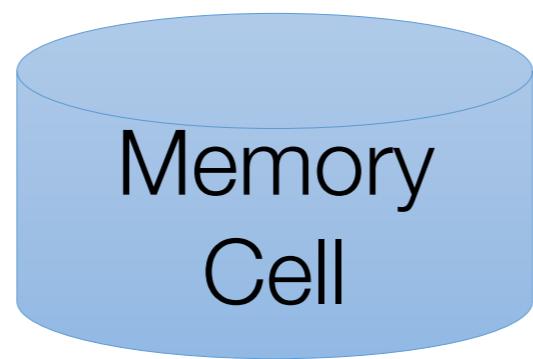
RNNs

- Recurrent networks are deep in that they involve one “layer” for each time step (e.g., words in a sentence)
- **Vanishing gradient problem:** as error is back propagated through the layers of a deep network, they tend toward 0.

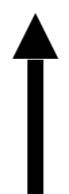
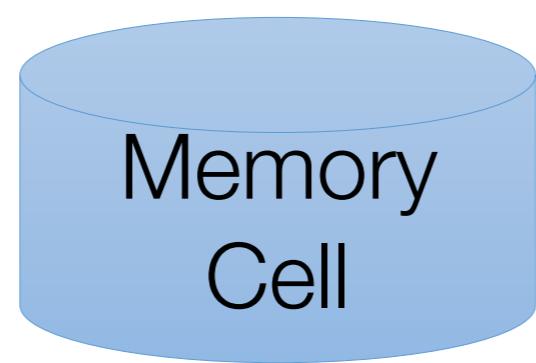
Long short-term memory network (LSTM)

- Designed to account for the vanishing gradient problem
- Basic idea: split the s vector propagated between time steps into a **memory** component and a **hidden state** component

Long Short-term Memory (LSTM)

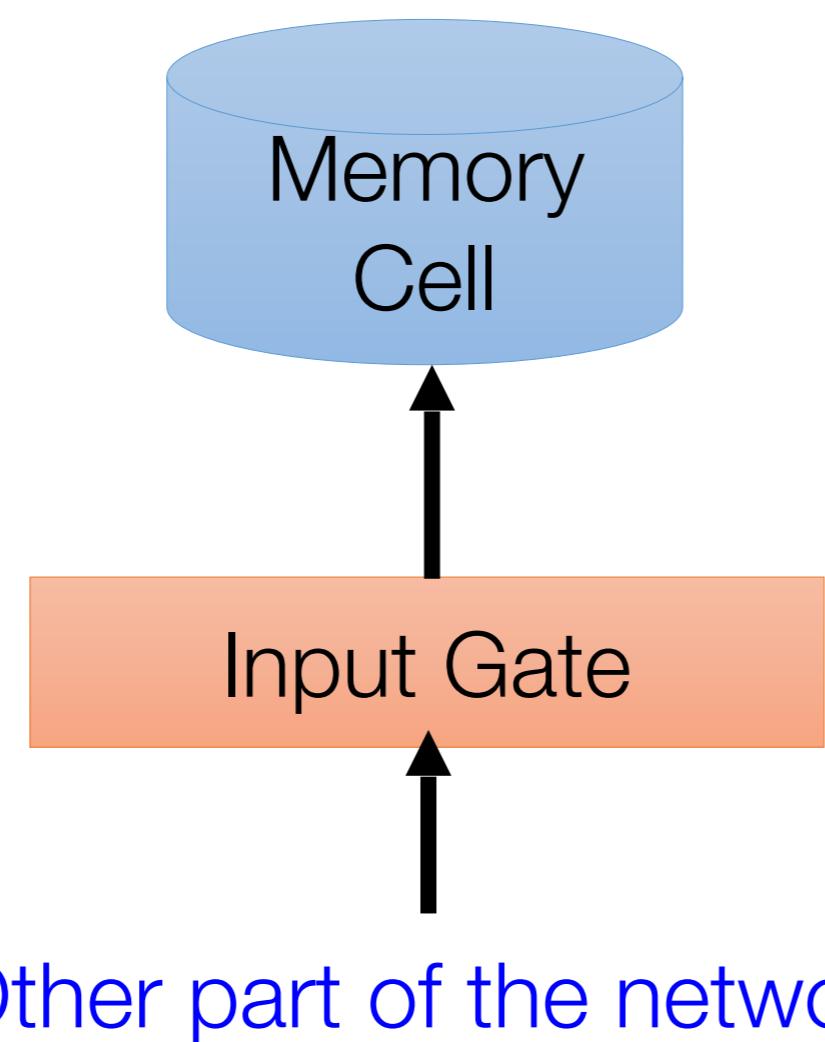


Long Short-term Memory (LSTM)

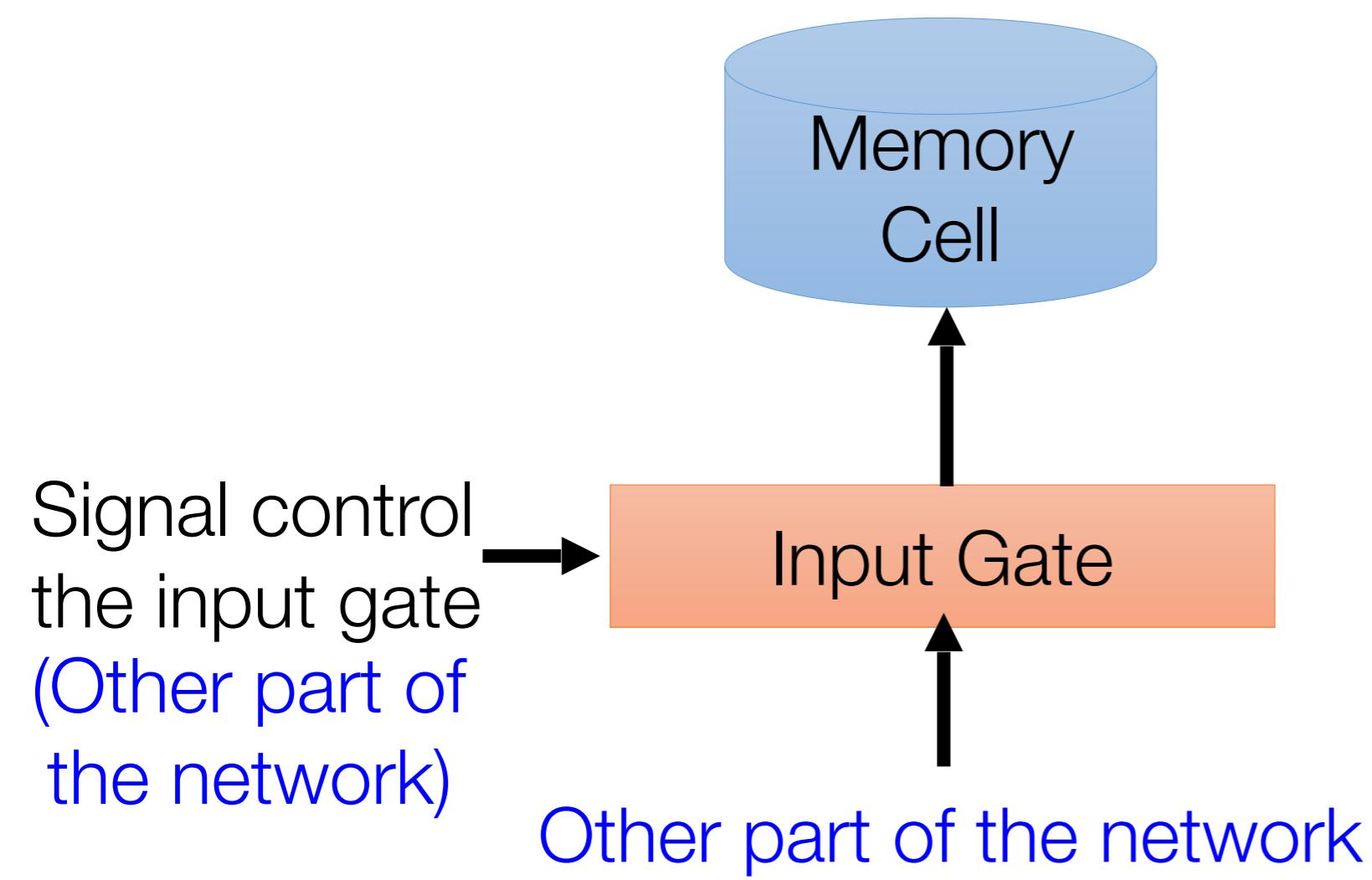


Other part of the network

Long Short-term Memory (LSTM)

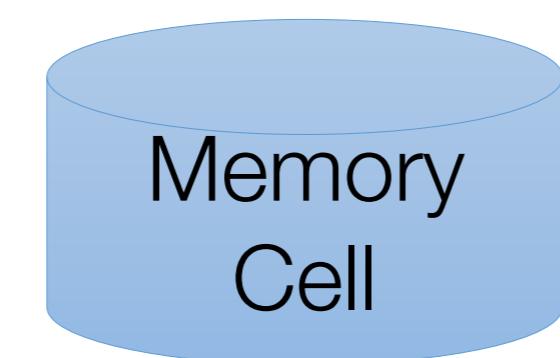


Long Short-term Memory (LSTM)



Long Short-term Memory (LSTM)

Other part of the network



Memory
Cell



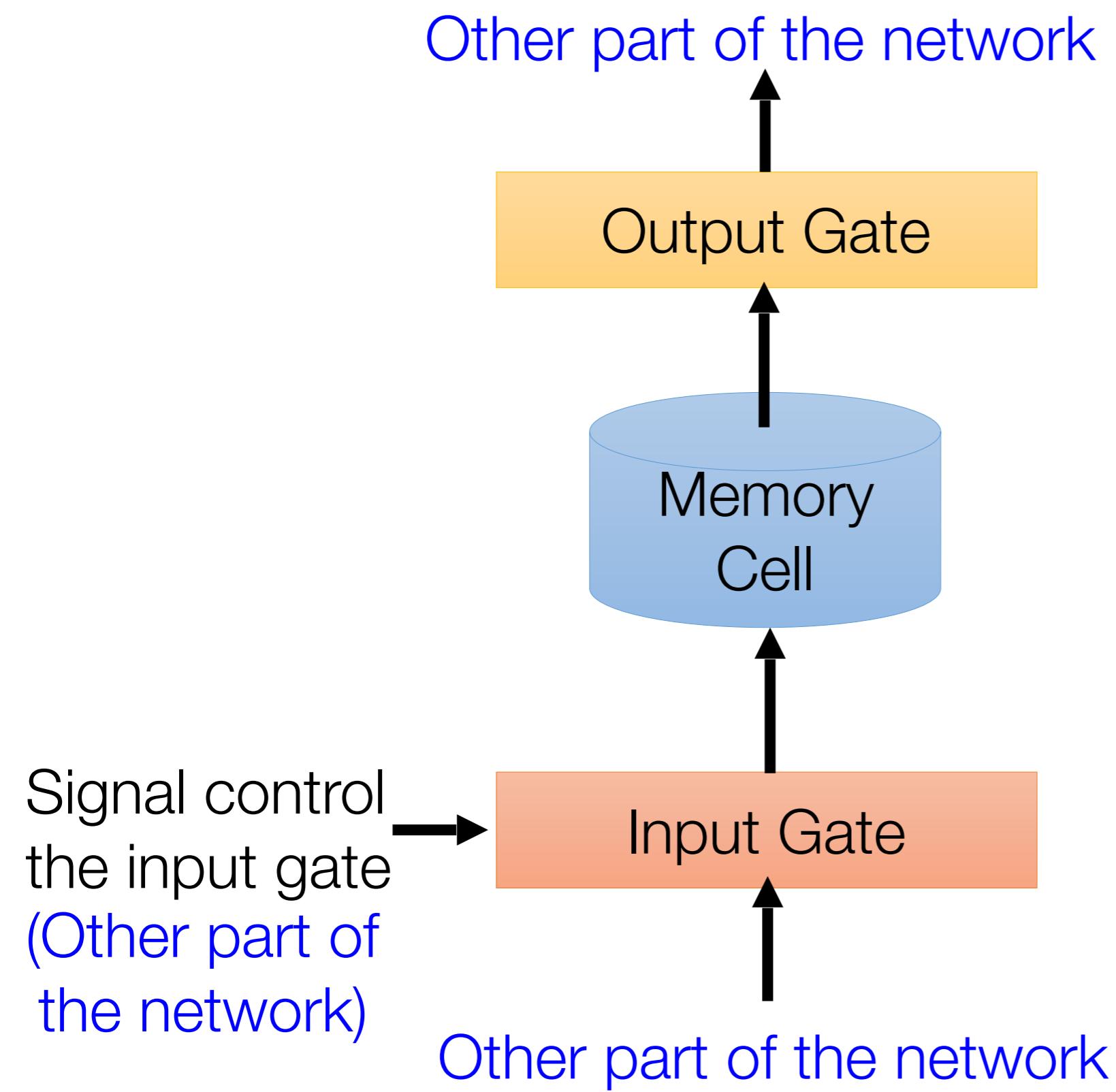
Signal control
the input gate
(Other part of
the network)

Input Gate

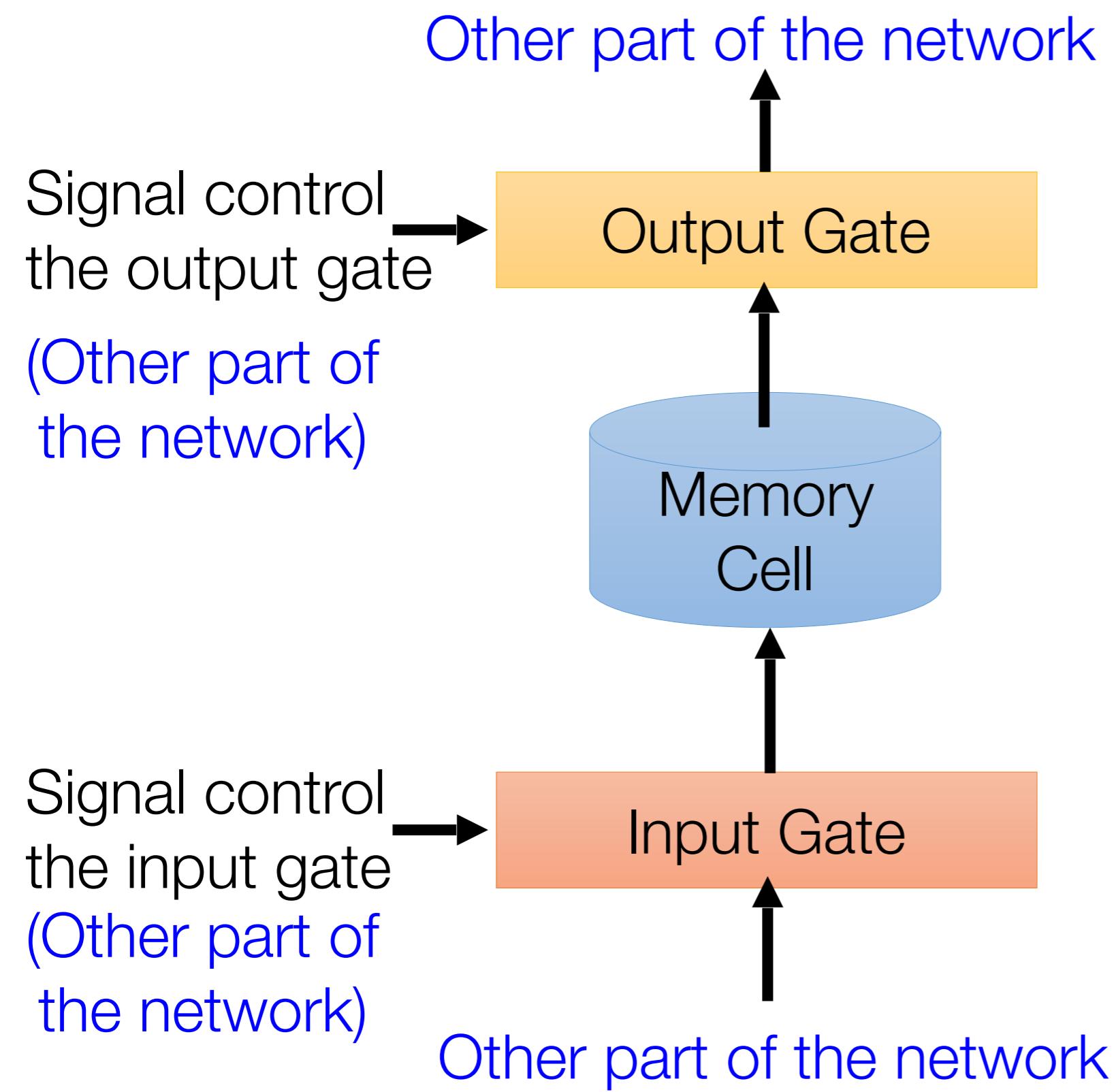


Other part of the network

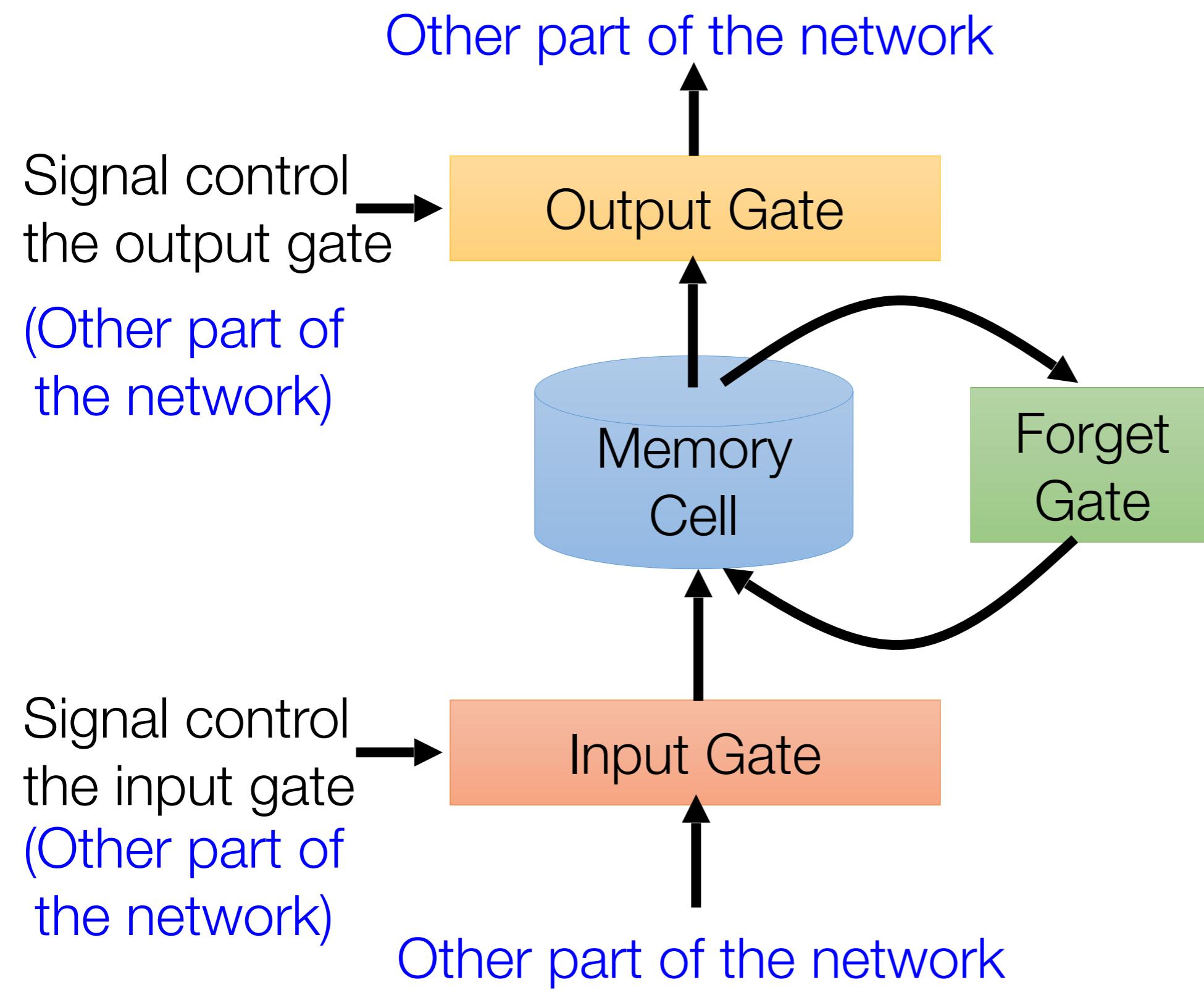
Long Short-term Memory (LSTM)



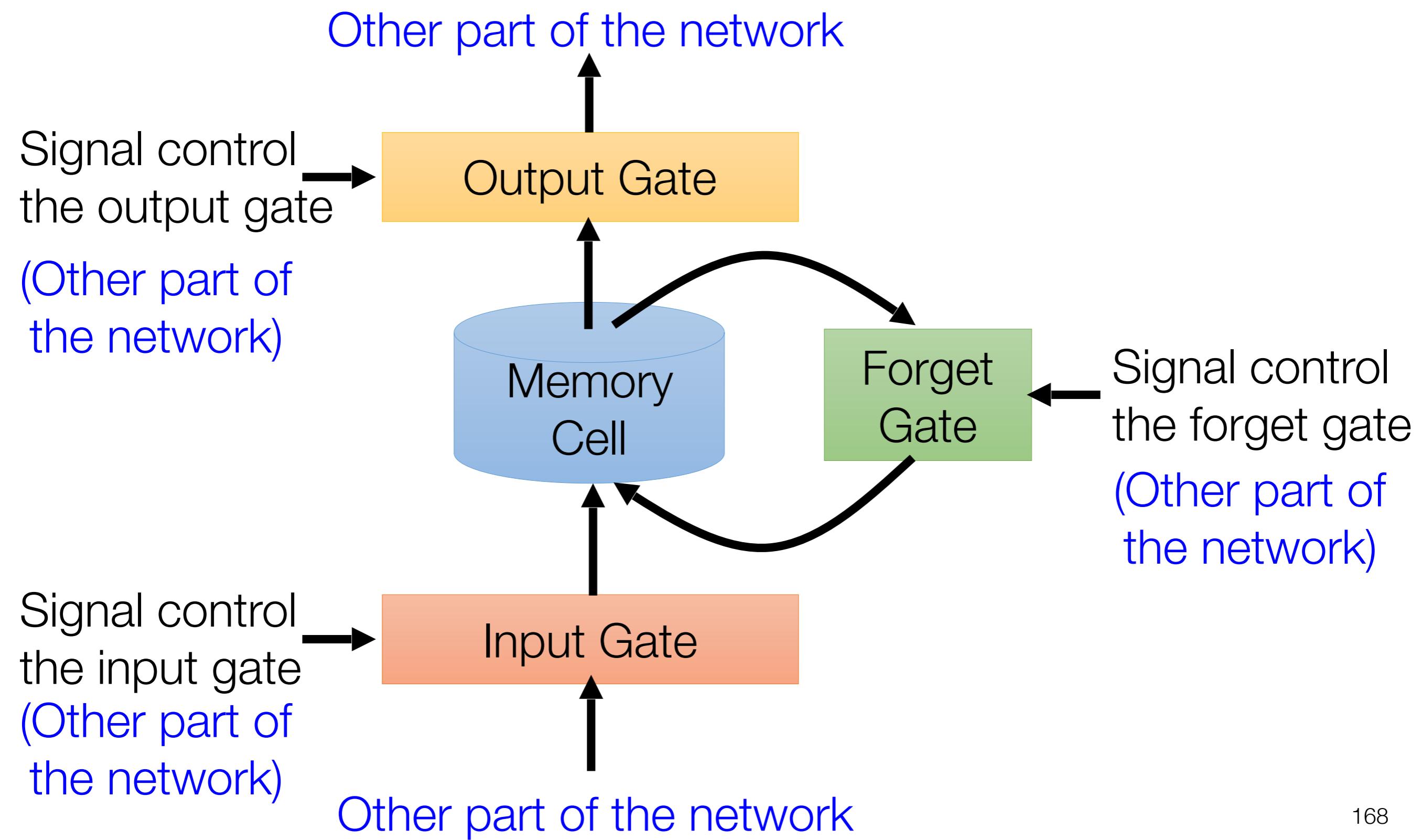
Long Short-term Memory (LSTM)



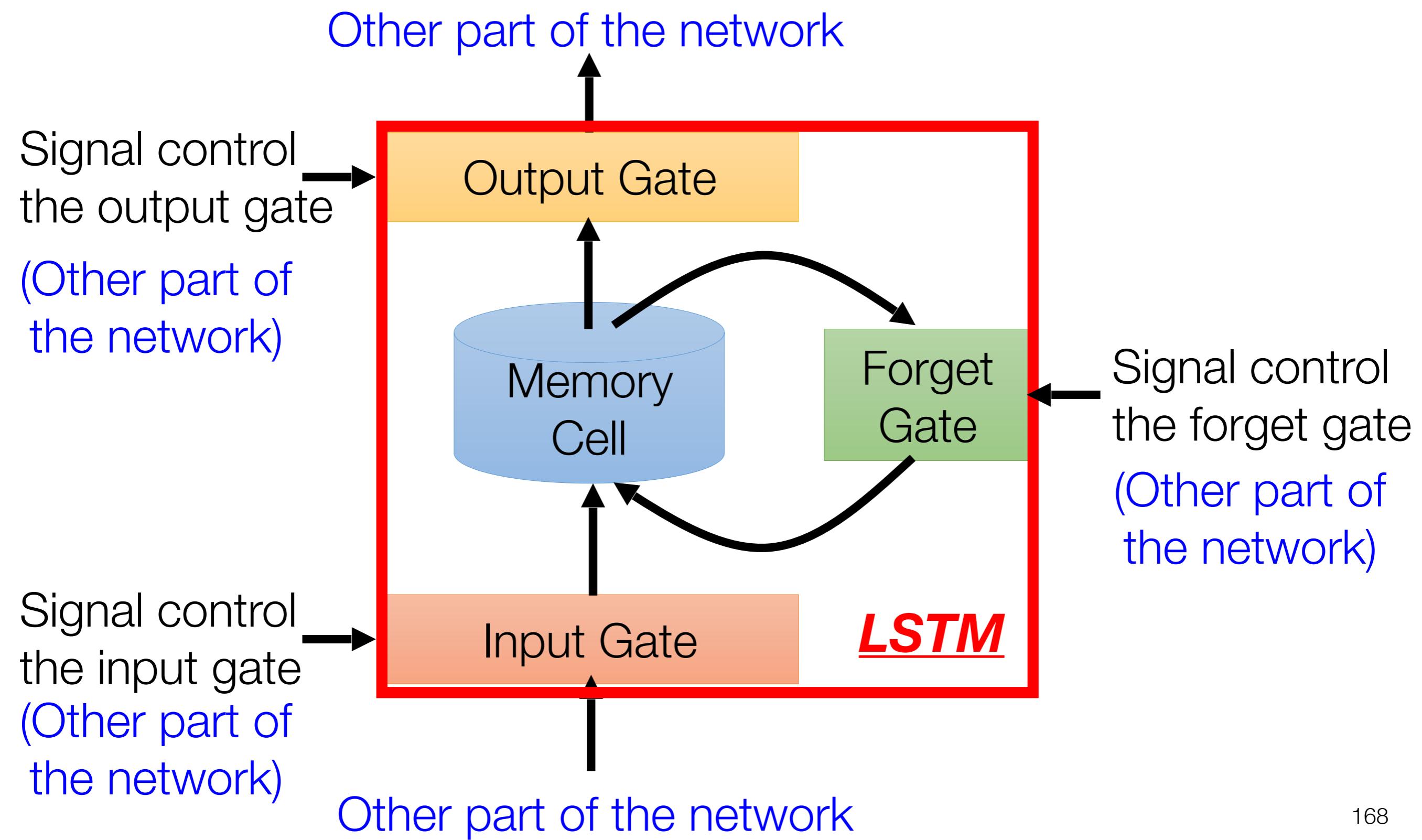
Long Short-term Memory (LSTM)



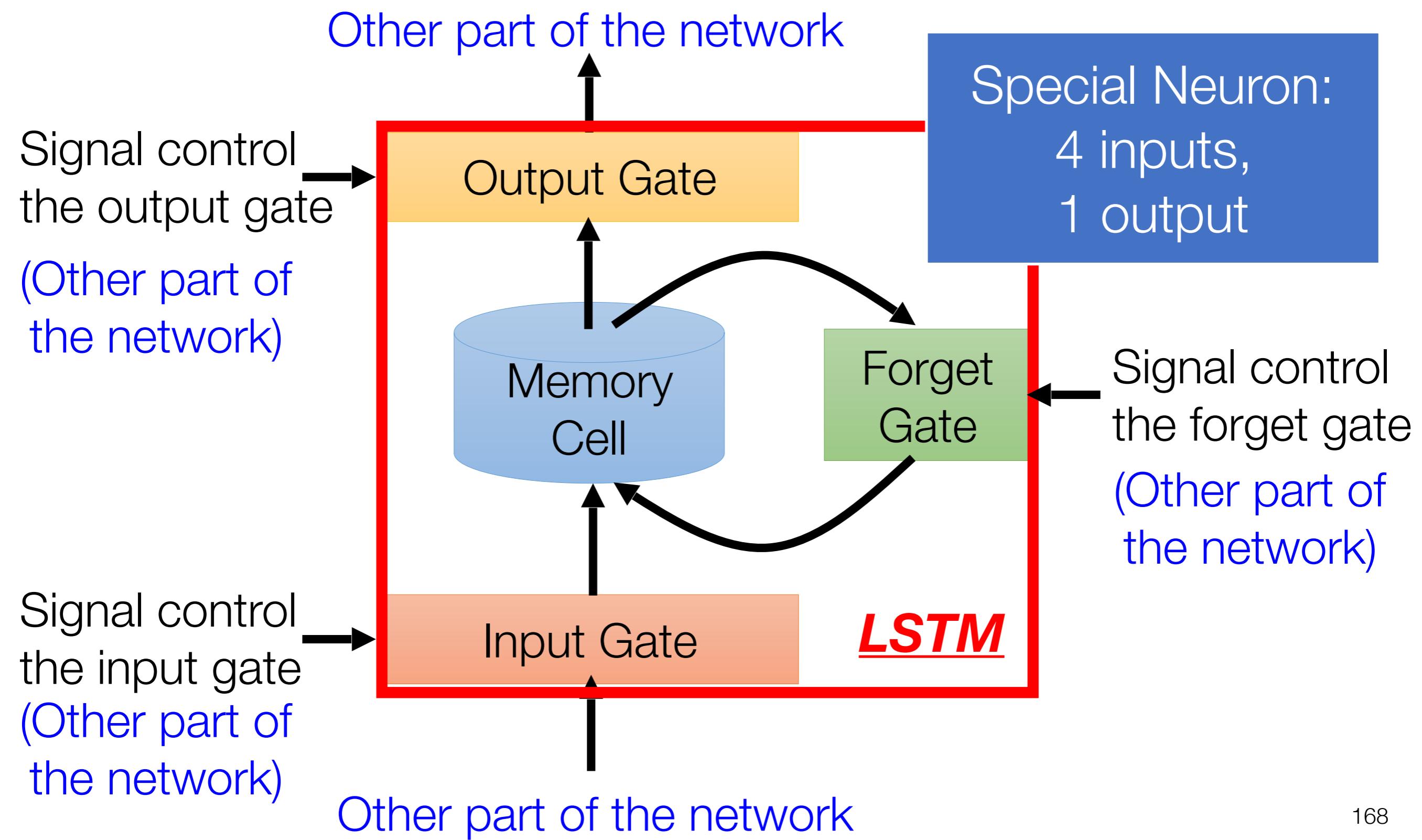
Long Short-term Memory (LSTM)



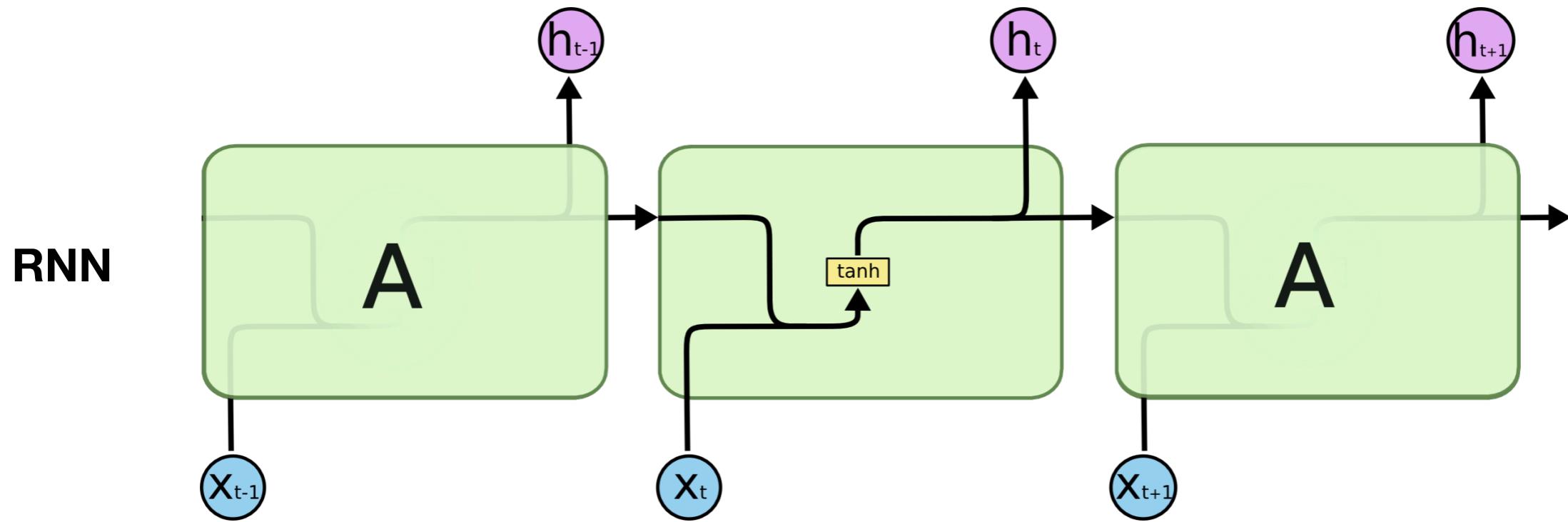
Long Short-term Memory (LSTM)



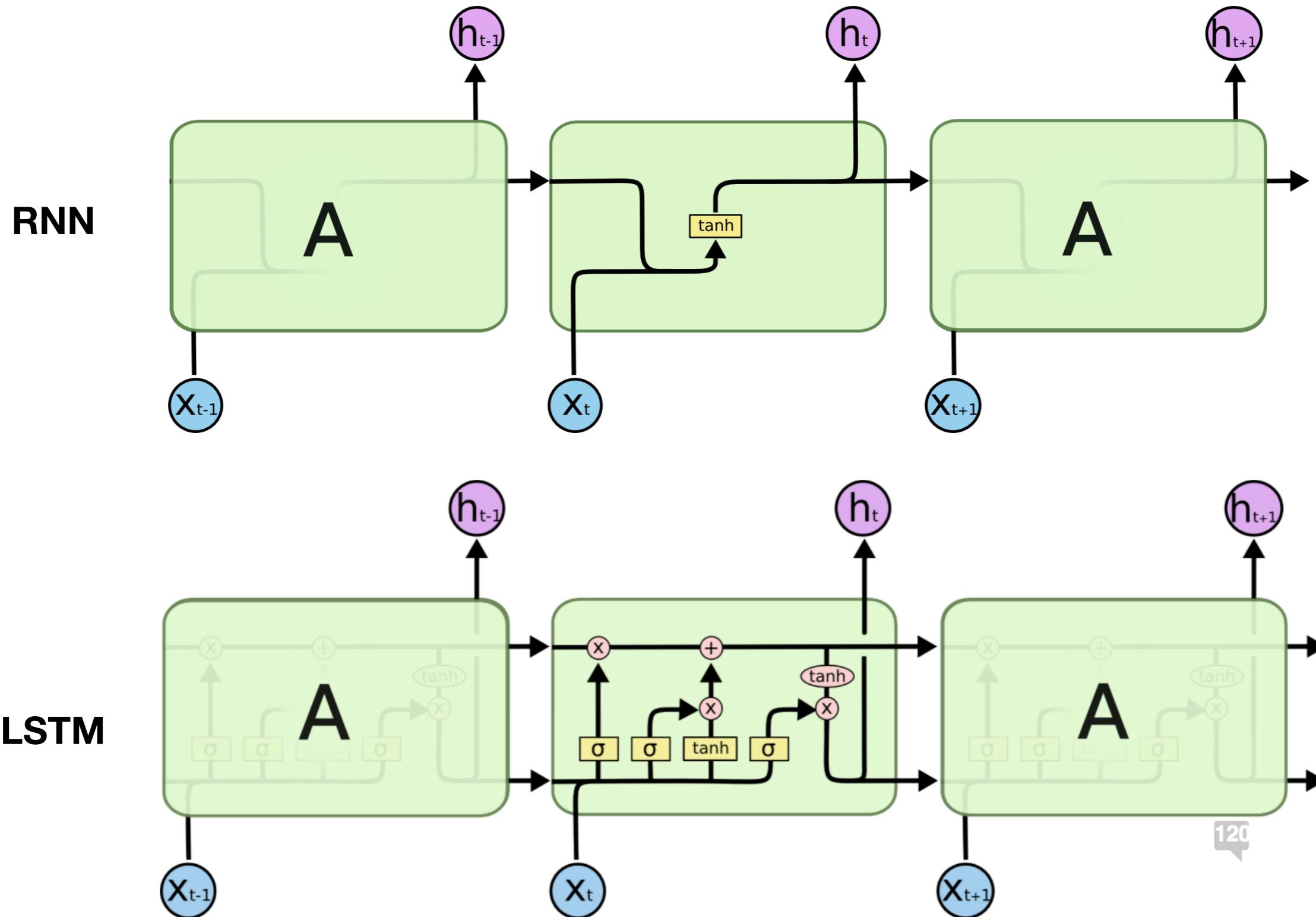
Long Short-term Memory (LSTM)



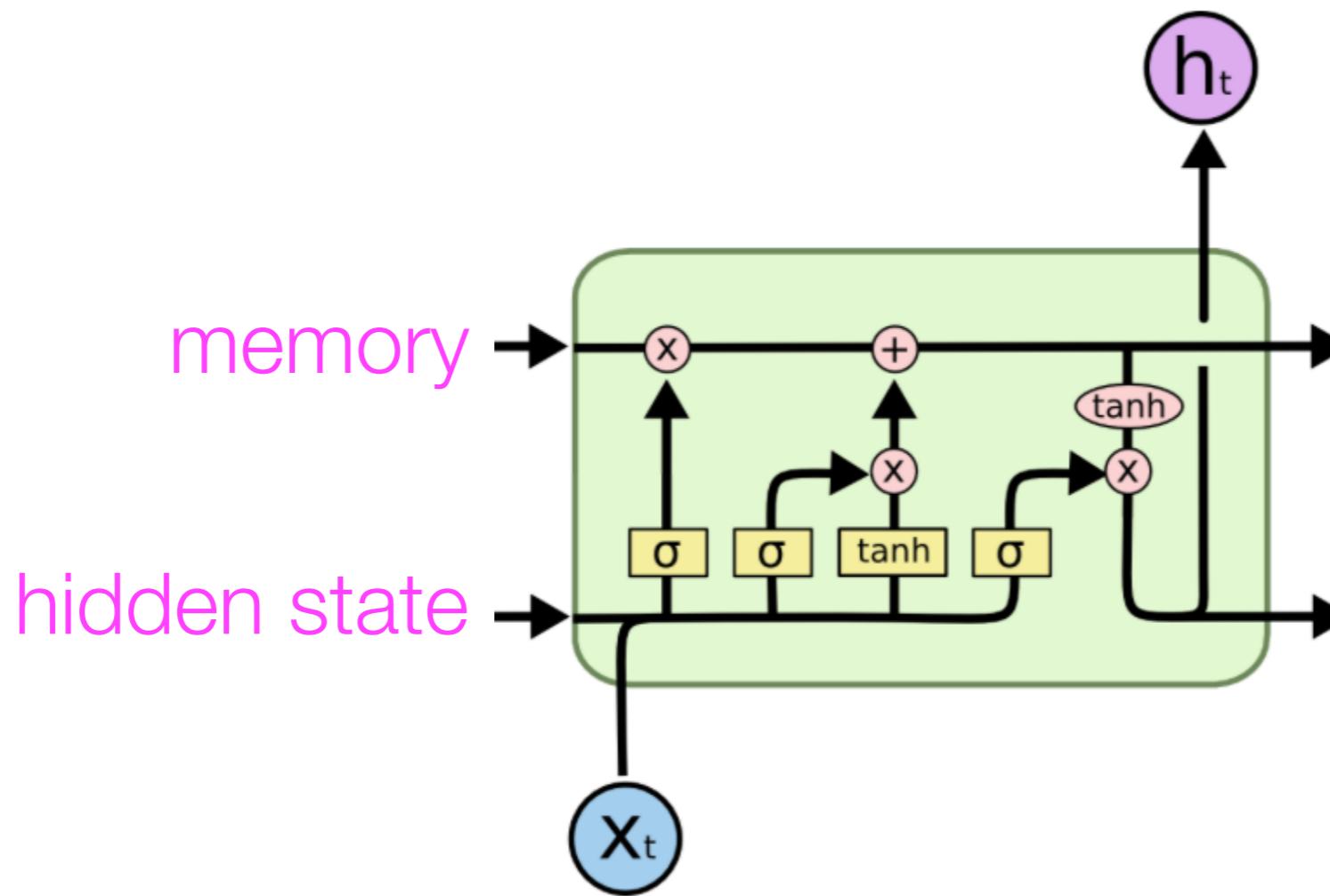
Comparing RNNs with LSTMs



Comparing RNNs with LSTMs



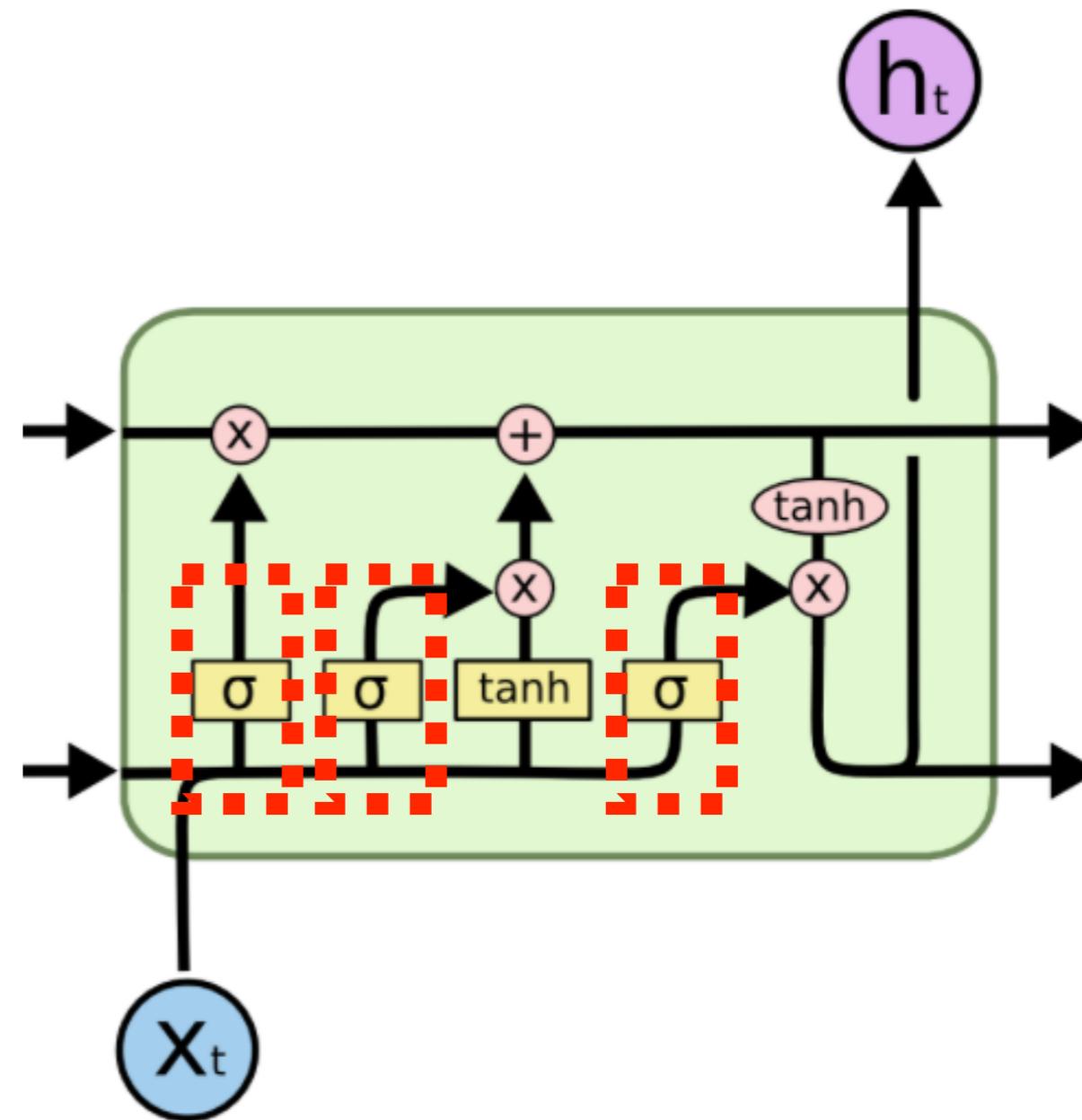
LSTM Gates



- LSTMs gates control the flow of information

LSTM Gates

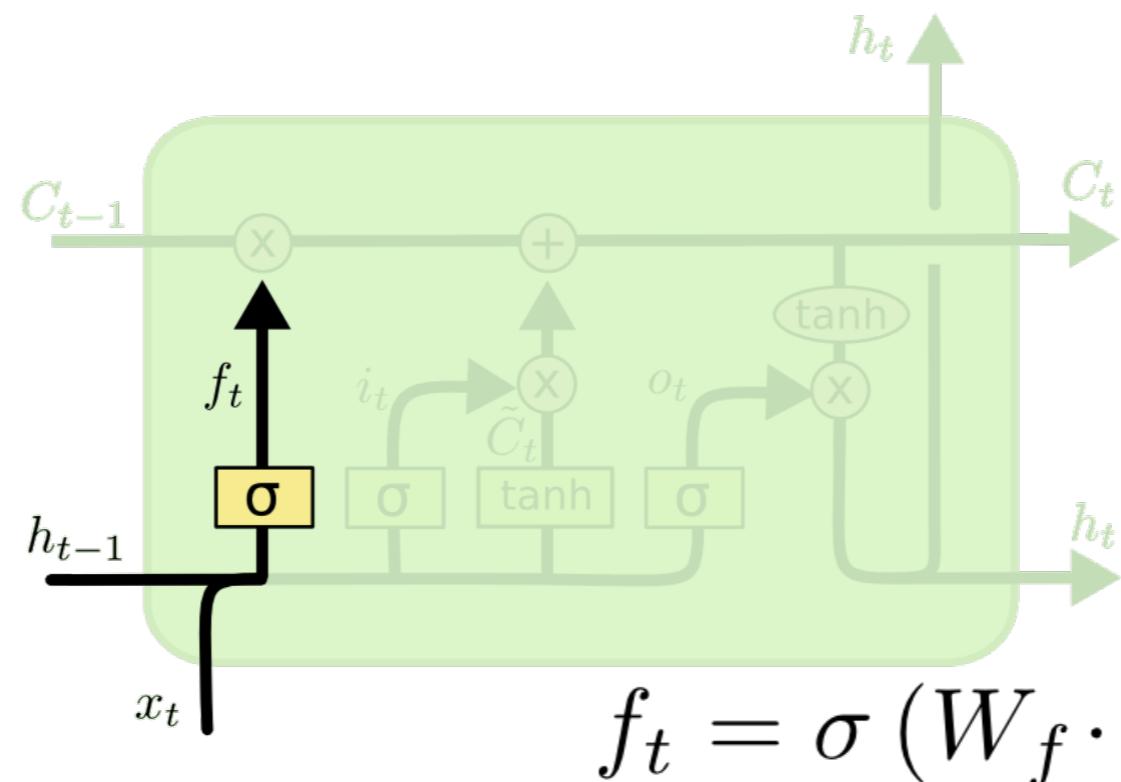
memory
hidden state



- A sigmoid squashes its input to between 0 and 1
- By multiplying the output of a sigmoid element-wise with another vector, we **forget** information in the vector (if multiplied by 0) or **allow it to pass** (if multiplied by 1)

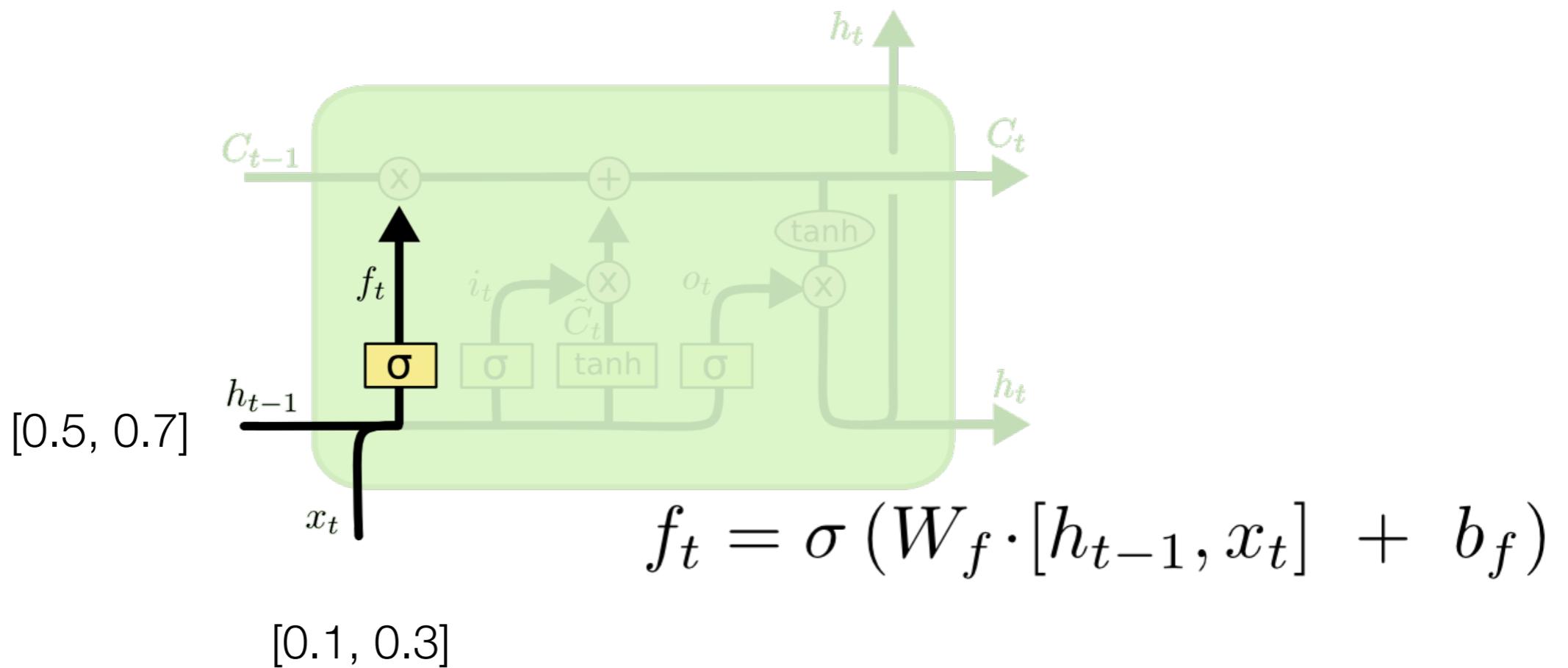
The forget gate

The sigmoid layer tells us which (or what proportion of) values to update



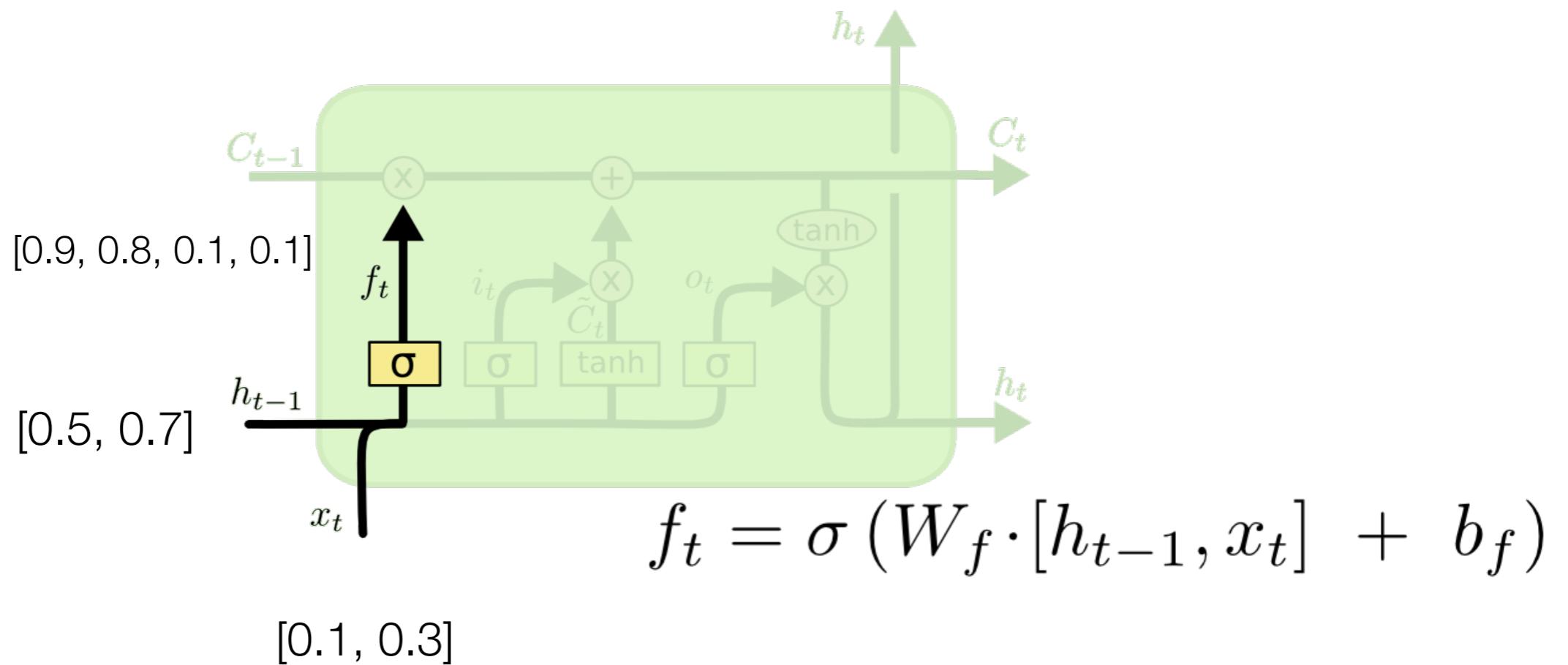
The forget gate

The sigmoid layer tells us which (or what proportion of) values to update

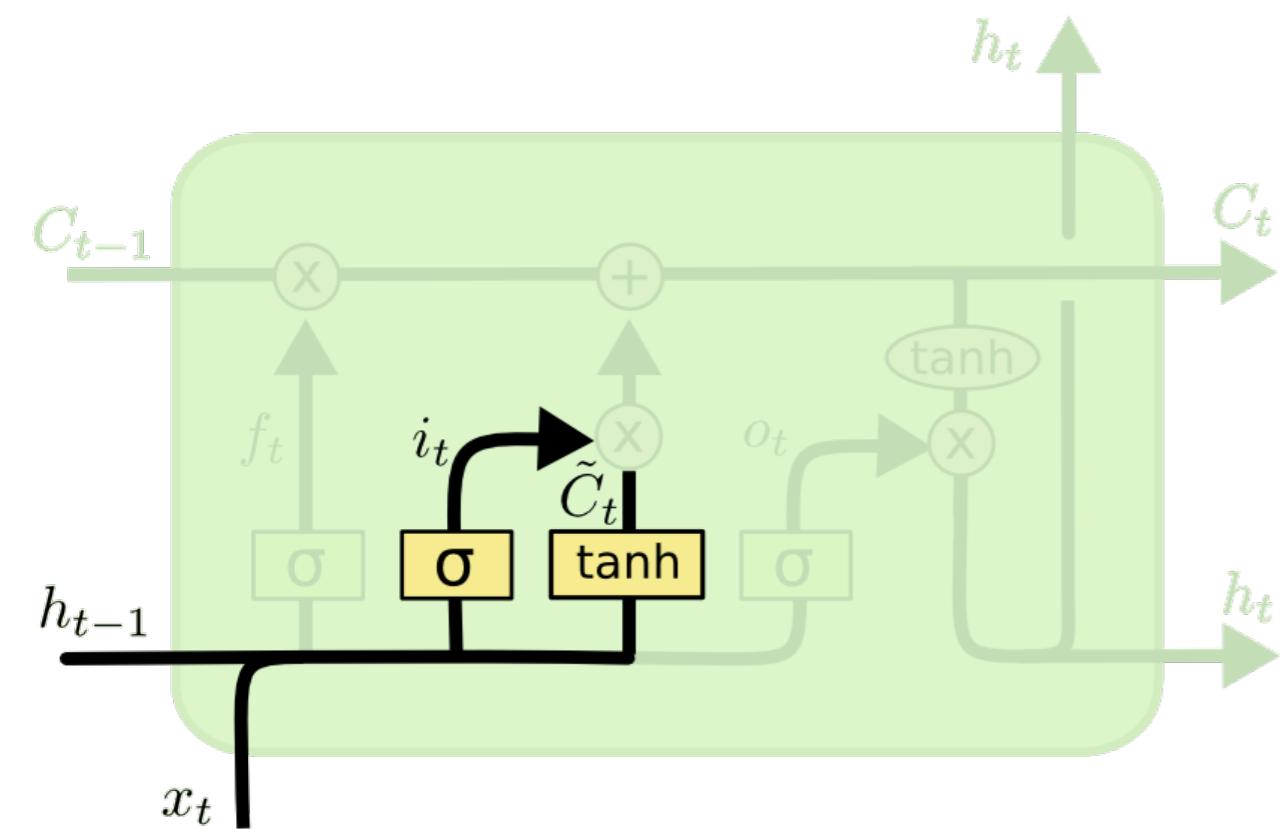


The forget gate

The sigmoid layer tells us which (or what proportion of) values to update



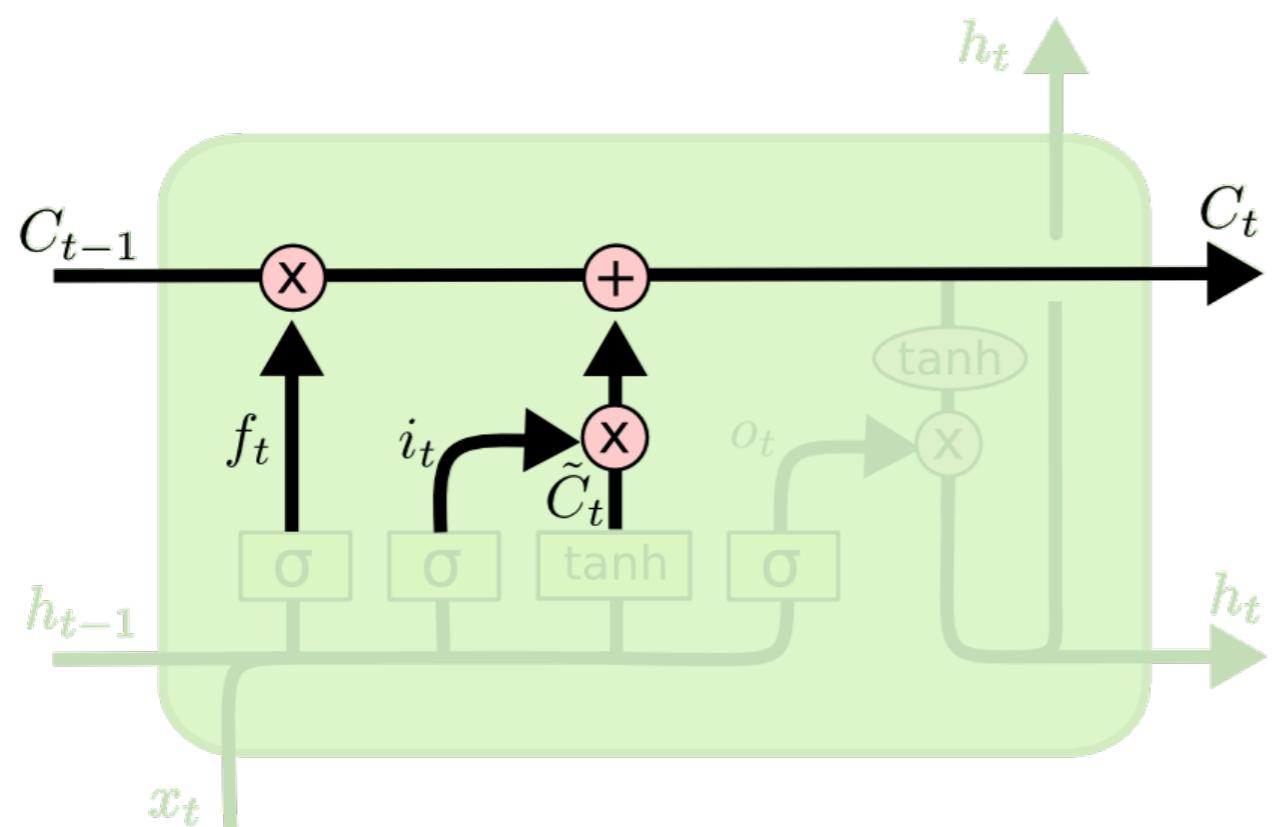
The input gate



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

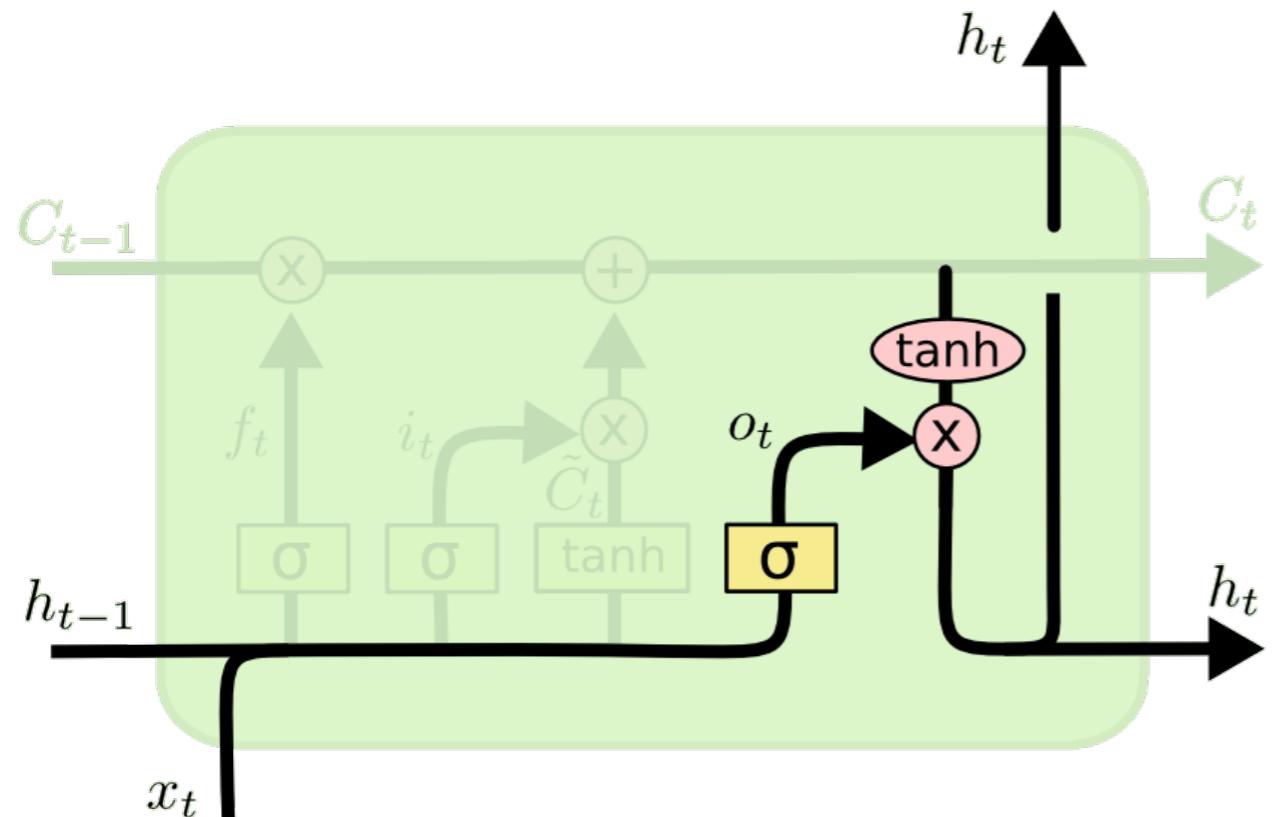
The tanh layer tells us how to update the state

Updating the cell state



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Deciding what to output from the cell



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

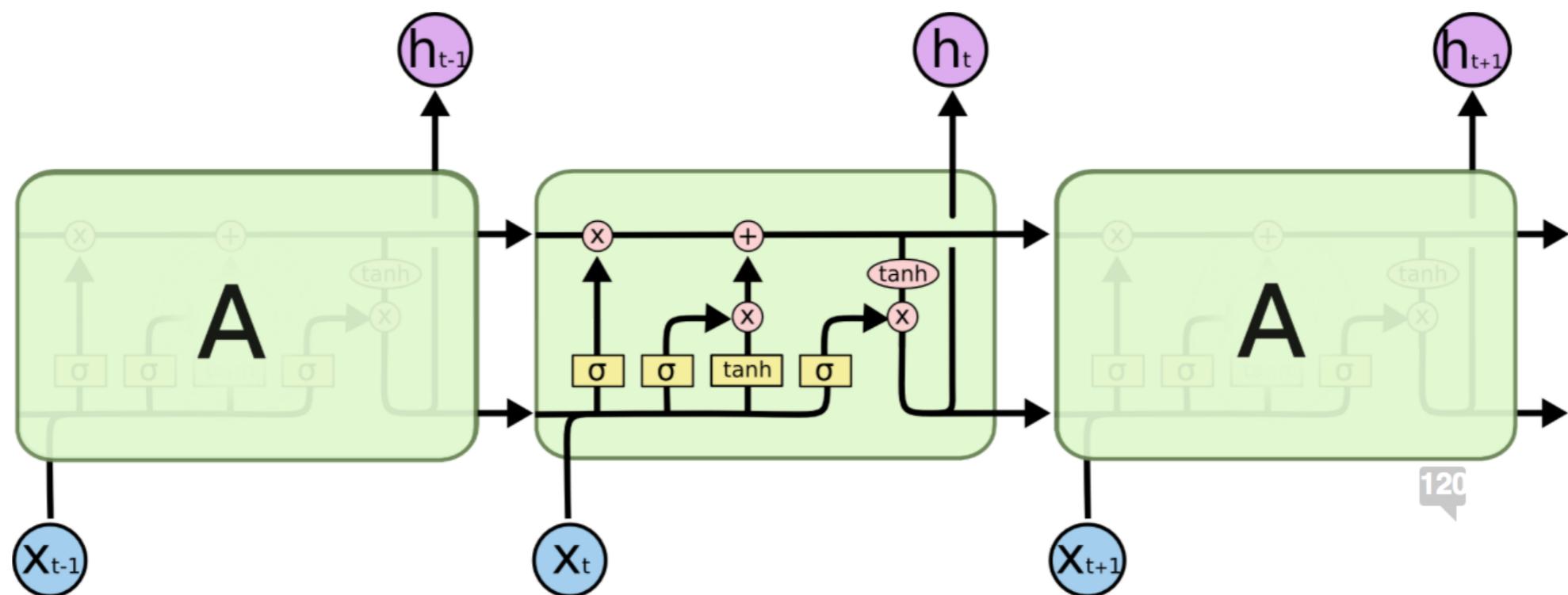
$$h_t = o_t * \tanh (C_t)$$

- RNN allow arbitrarily-sized conditioning contexts; condition on the entire sequence history.

How much context is the LSTM capable of capturing?

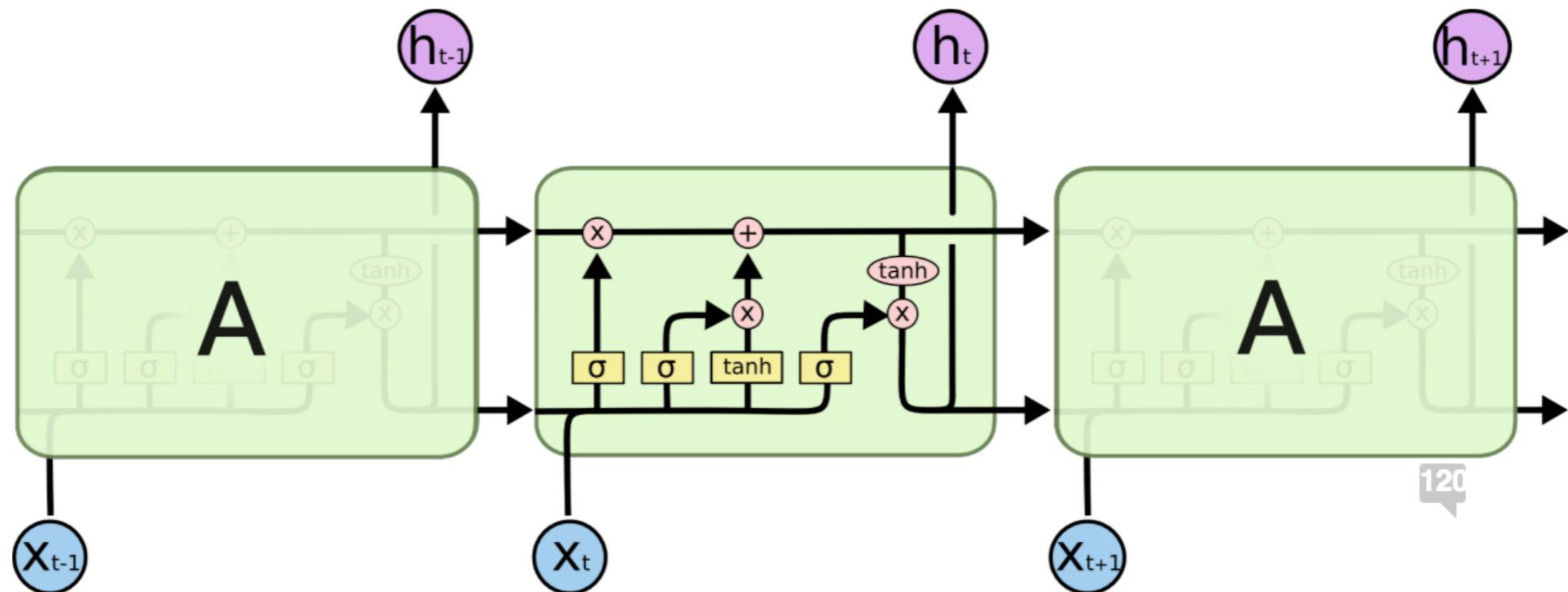
$$P(x_i | x_{i-1}, x_{i-2}, x_{i-\text{????}})$$

How much context is the LSTM capable of capturing?



$$P(x_i | x_{i-1}, x_{i-2}, x_{i-?????})$$

How much context is the LSTM capable of capturing?

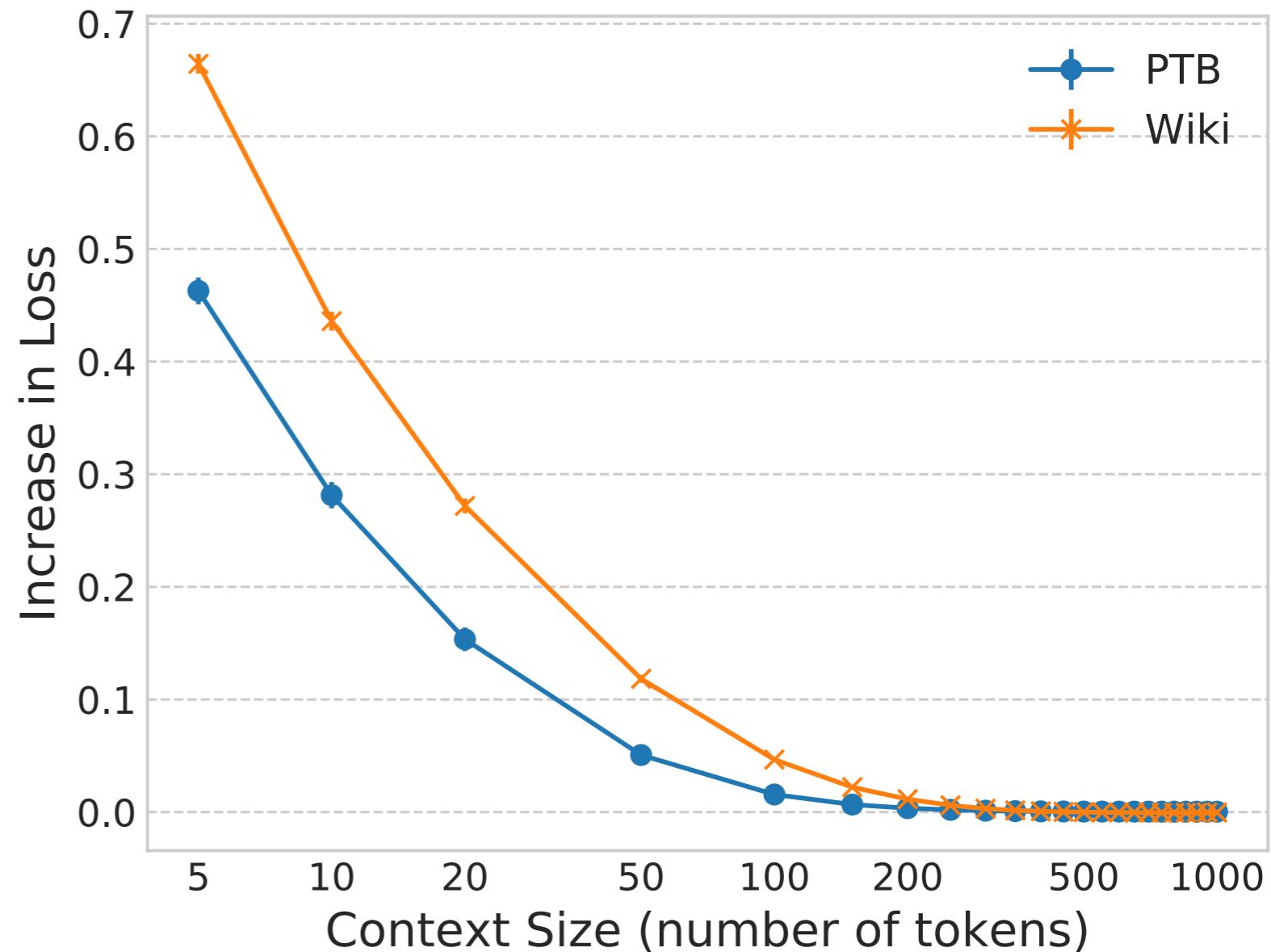


$$P(x_i \mid x_{i-1}, x_{i-2}, x_{i-????})$$

How would we design an experiment to measure how context an LSTM is using?

How much context is the LSTM capable of capturing?

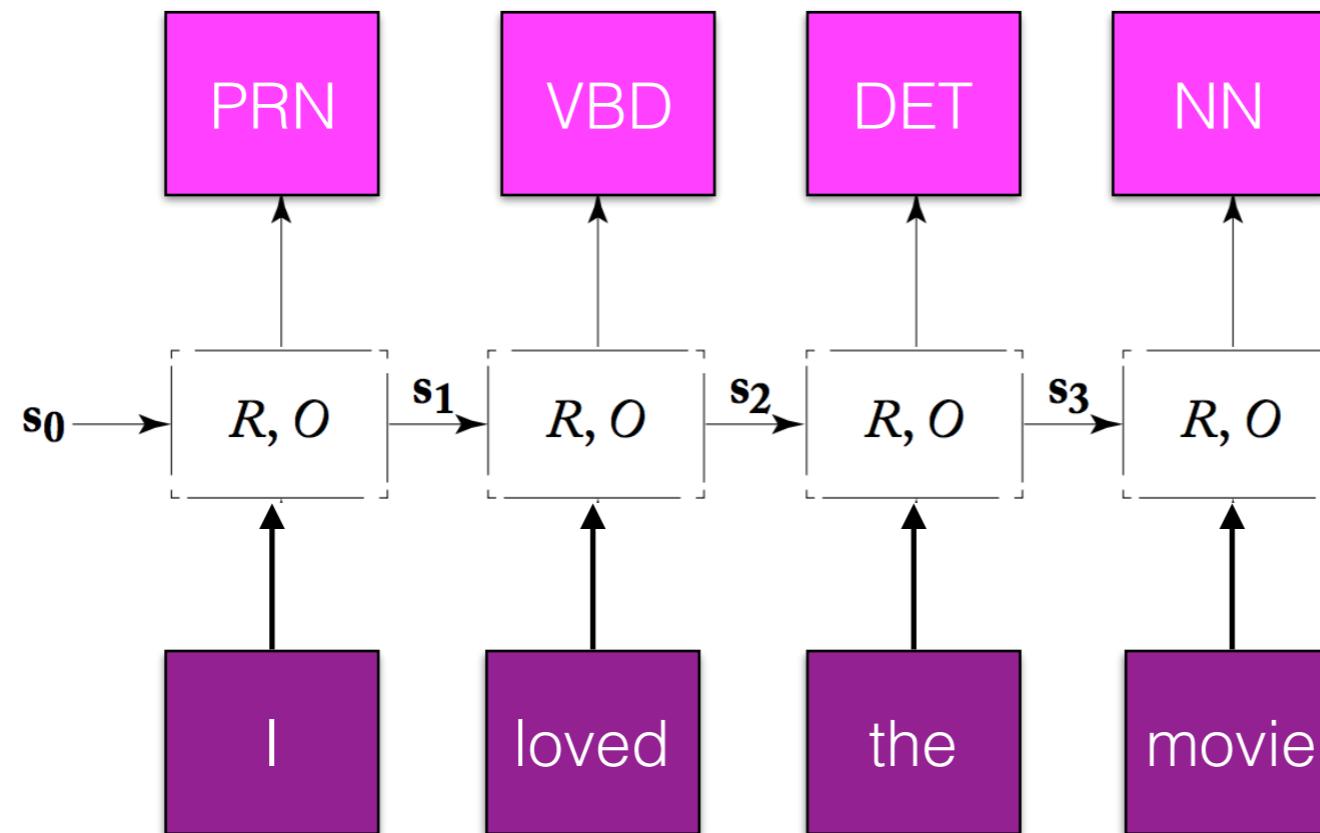
- For language modeling, LSTMs are aware of about 200 words of context
- Ignores word order beyond 50 words



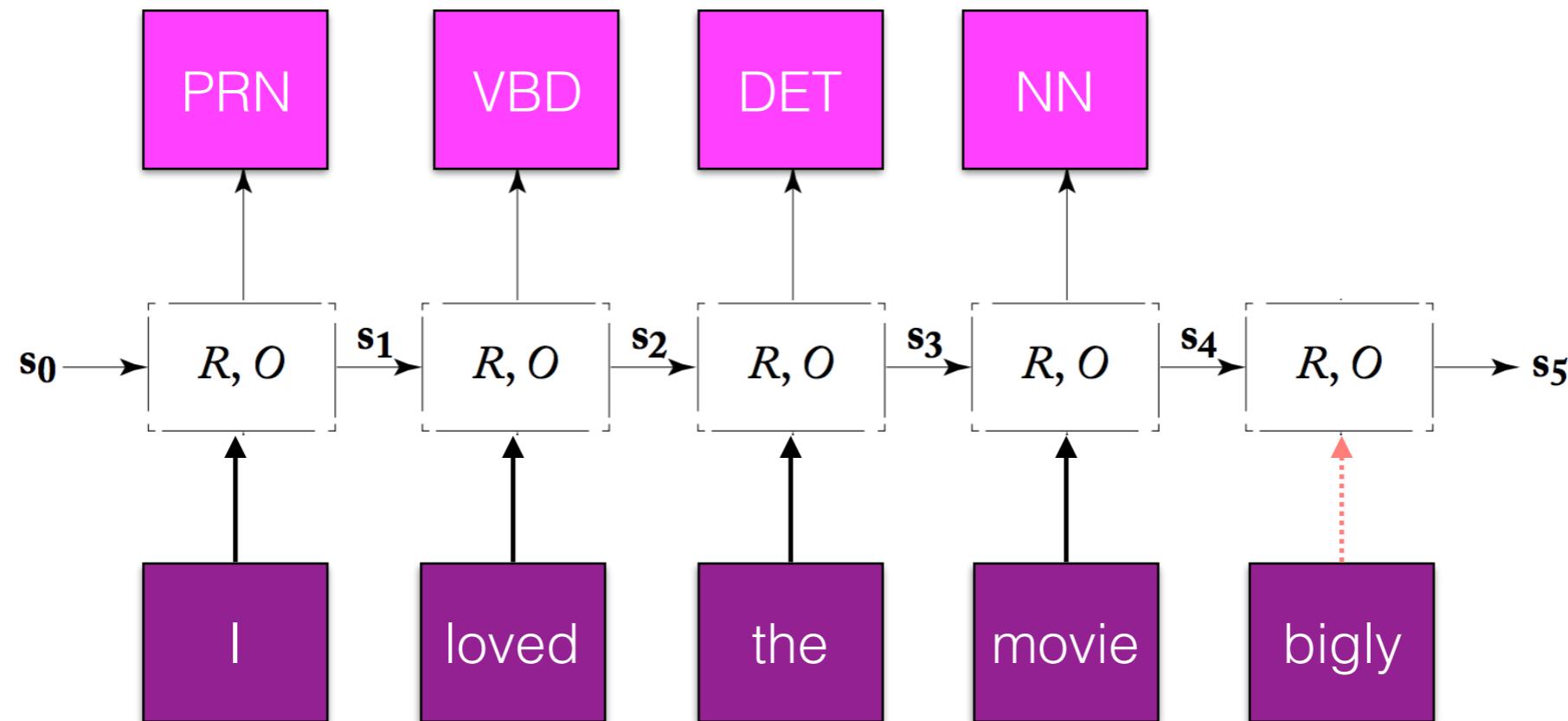


Networks below the
word level

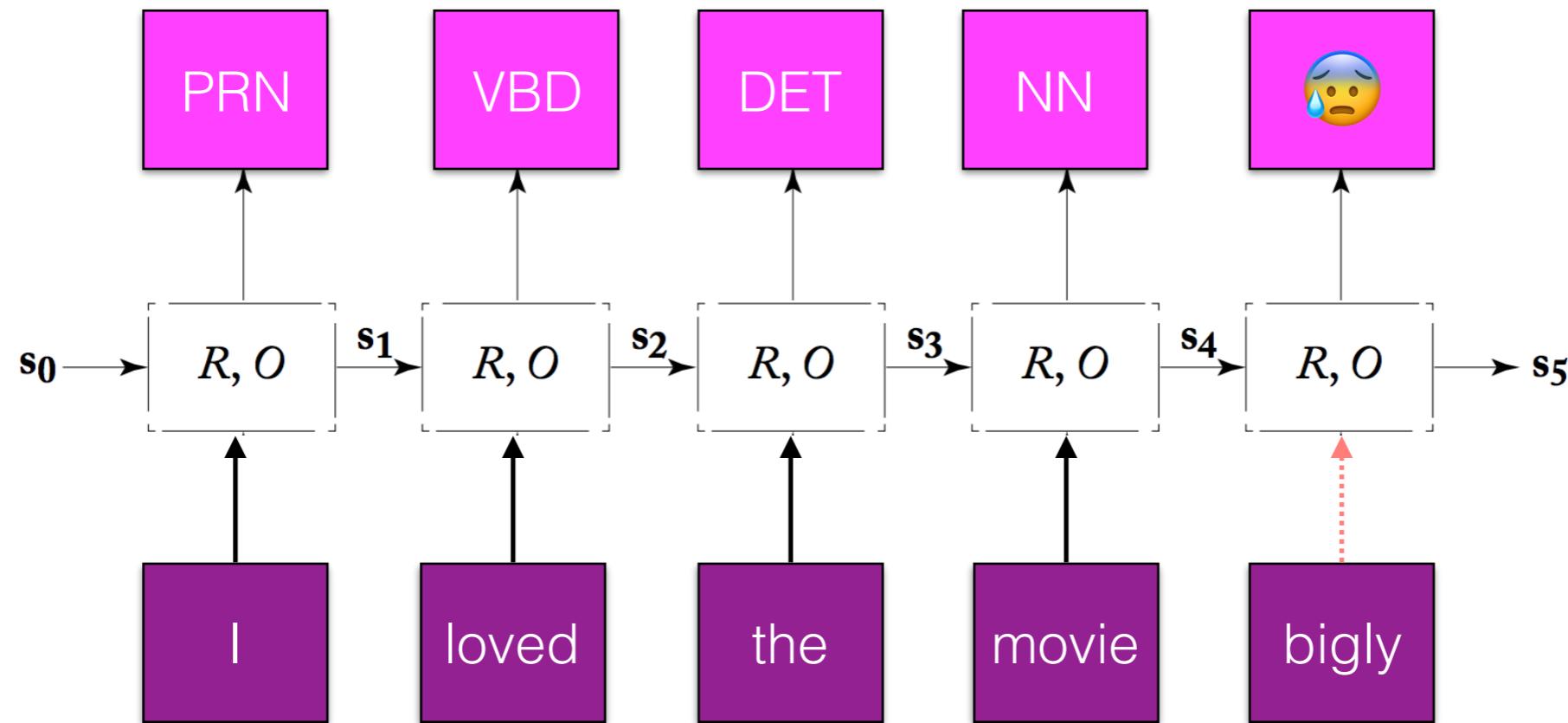
LSTMs don't solve all the problems



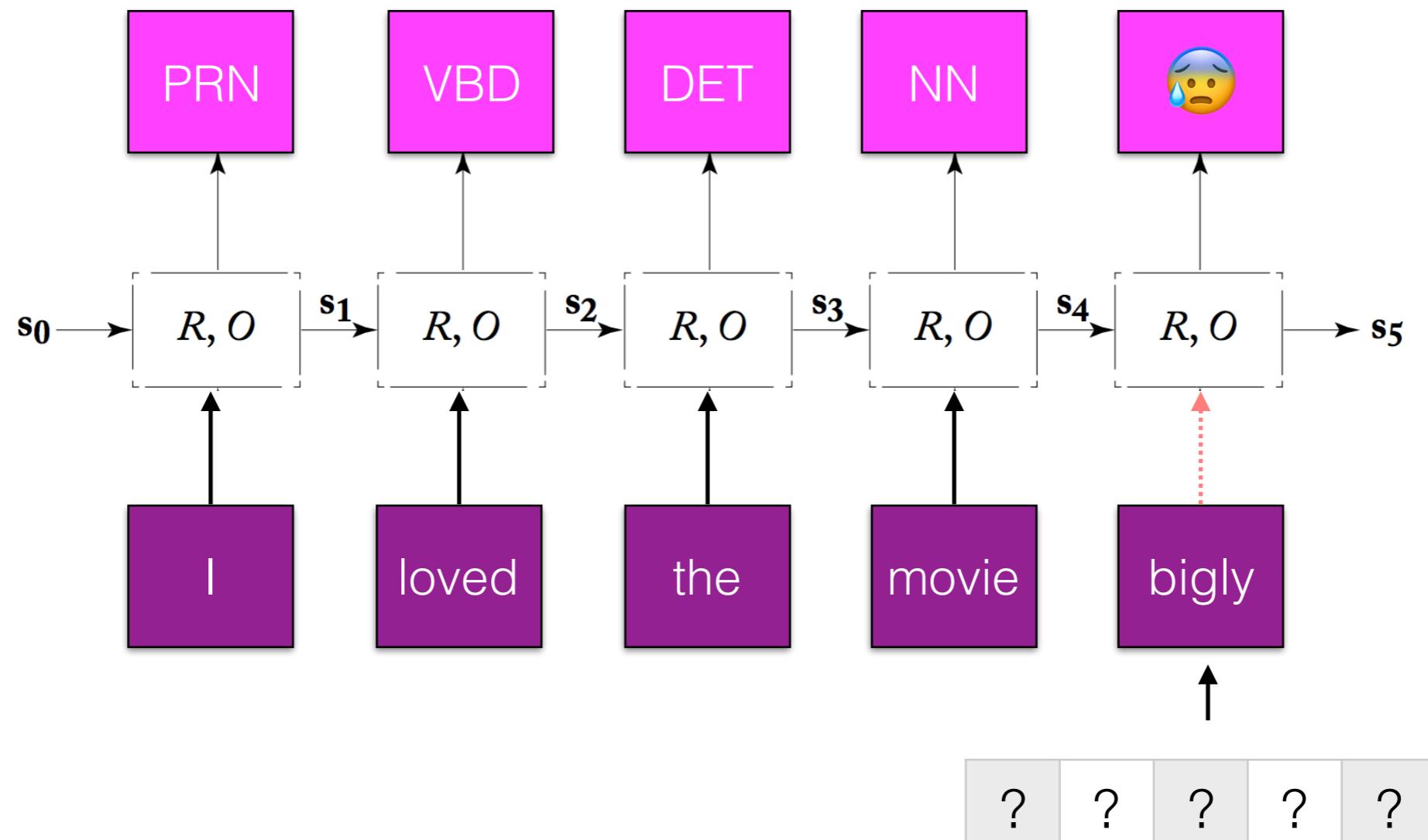
LSTMs don't solve all the problems



LSTMs don't solve all the problems



How do we handle out of vocabulary words?



Muvaffak

Successful

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverbil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekler	Those who will not be able to make one easily/quickly a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekler	Those who will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde n	Among/From those whom we will not be able to easily/quickly make a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekler	Those who will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde n	Among/From those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmış	(He/she) happens to be have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekler	Those who will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde n	Among/From those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmış	(He/she) happens to be have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmışsiniz	You happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones

Muvaffak	Successful
Muvaffakiyet	Success ('successfulness')
Muvaffakiyetsiz	Unsuccessful ('without success')
Muvaffakiyetsizleş(-mek)	(To) become unsuccessful
Muvaffakiyetsizleştirici	Maker of unsuccessful ones
Muvaffakiyetsizleştiricileş(-mek)	(To) become a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştir(-mek)	(To) make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriver(-mek)	(To) easily/quickly make one a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriverebil(-mek)	(To) be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecek	(He/she who) will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebilecekler	Those who will not be able to make one easily/quickly a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde n	Among/From those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmiş	(He/she) happens to be have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmişsiniz	You happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones
Muvaffakiyetsizleştiricileştiriveremeyebileceklerimizde nmişsinizcesine	As though you happen to have been from among those whom we will not be able to easily/quickly make a maker of unsuccessful ones

Many words have shared structure

- Even in languages like English that are not agglutinative and aren't highly inflected, words share important structure.

friend

friended

friendless

friendly

friendship

unfriend

unfriendly

Many words have shared structure

- Even in languages like English that are not agglutinative and aren't highly inflected, words share important structure.

friend

friended

friendless

friendly

friendship

unfriend

unfriendly

un + friend + ly

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

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

FastText

3-grams

$e(\text{where}) =$

$e(*)$ = embedding for $*$ word

FastText

e(<wh)
+ e(whe)
+ e(her)
+ e(ere)
+ e(re>)

3-grams

e(where) =

4-grams

e(*) = embedding for * word

FastText

$e(\text{where}) =$

$e(<\text{wh}>)$
+ $e(\text{whe})$
+ $e(\text{her})$
+ $e(\text{ere})$
+ $e(\text{re}>)$

+ $e(<\text{whe}>)$
+ $e(\text{wher})$
+ $e(\text{here})$
+ $e(\text{ere}>)$

3-grams

4-grams

$e(^*)$ = embedding for * word

FastText

$e(\text{where}) =$

$e(<\text{wh}>)$
+ $e(\text{whe})$
+ $e(\text{her})$
+ $e(\text{ere})$
+ $e(\text{re}>)$

3-grams

+ $e(<\text{whe}>)$
+ $e(\text{wher})$
+ $e(\text{here})$
+ $e(\text{ere}>)$

4-grams

+ $e(<\text{wher}>)$
+ $e(\text{where})$
+ $e(\text{here}>)$

5-grams

$e(^*)$ = embedding for * word

FastText

$e(\text{where}) =$

$e(*)$ = embedding for $*$ word

$e(<\text{wh}>)$
+ $e(\text{whe})$
+ $e(\text{her})$
+ $e(\text{ere})$
+ $e(\text{re}>)$

3-grams

+ $e(<\text{whe}>)$
+ $e(\text{wher})$
+ $e(\text{here})$
+ $e(\text{ere}>)$

4-grams

+ $e(<\text{wher}>)$
+ $e(\text{where})$
+ $e(\text{here}>)$

5-grams

+ $e(<\text{where}>)$
+ $e(\text{where}>)$

6-grams

FastText

$e(\text{where}) =$

$e(*)$ = embedding for * word

$e(<\text{wh}>)$
+ $e(\text{whe})$
+ $e(\text{her})$
+ $e(\text{ere})$
+ $e(\text{re}>)$

+ $e(<\text{whe}>)$
+ $e(\text{wher})$
+ $e(\text{here})$
+ $e(\text{ere}>)$

+ $e(<\text{wher}>)$
+ $e(\text{where})$
+ $e(\text{here}>)$

+ $e(<\text{where}>)$
+ $e(\text{where}>)$

+ $e(<\text{where}>)$

3-grams

4-grams

5-grams

6-grams

word

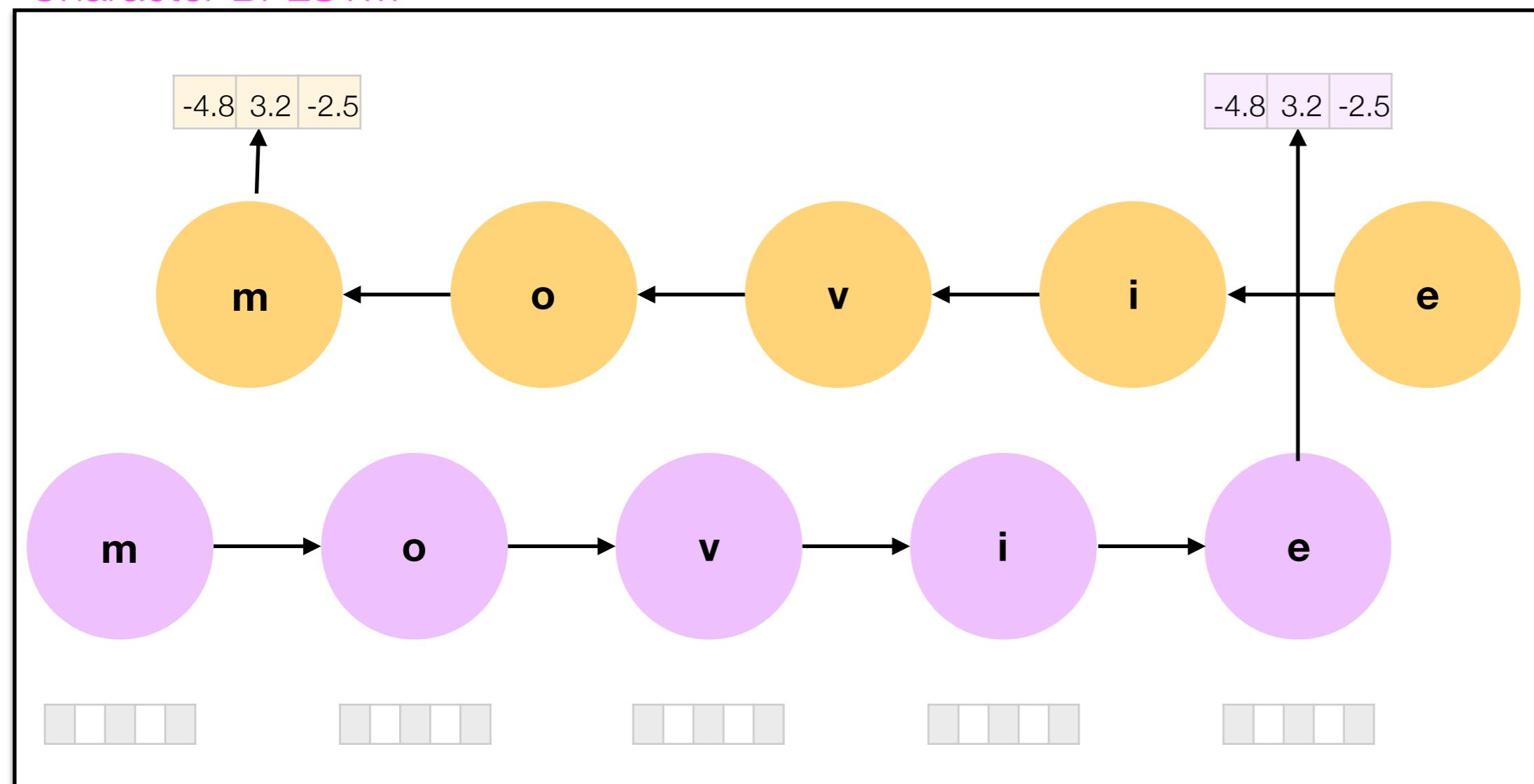
Subword information

- We saw subword information used for creating embeddings (FastText)
- Another alternative is to use standard word embeddings and reason about subword information within a model.

BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

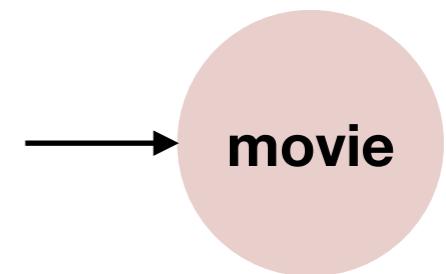
Lample et al. (2016), “Neural Architectures for Named Entity Recognition”

Character Bi-LSTM

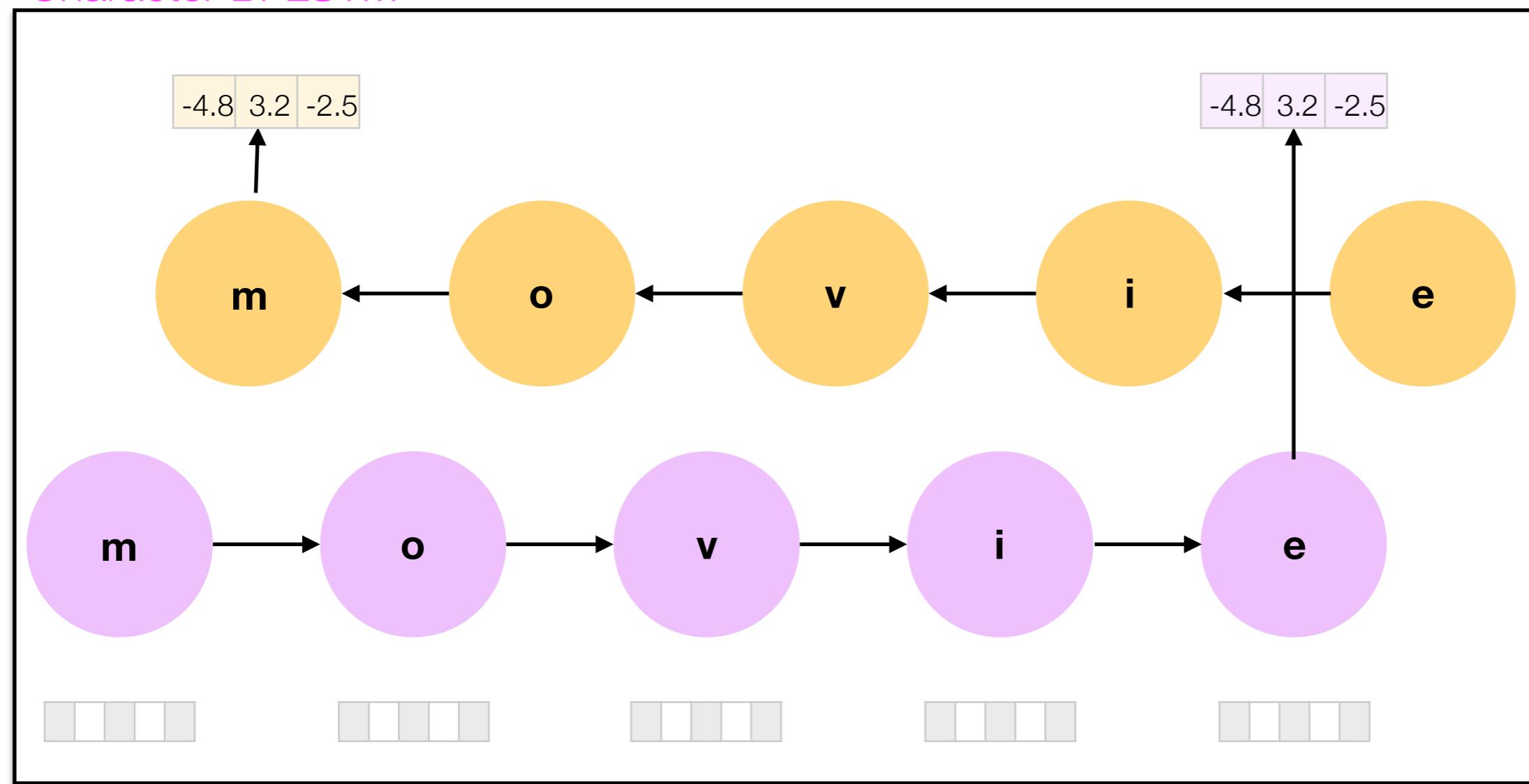


BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

Lample et al. (2016), “Neural Architectures for Named Entity Recognition”

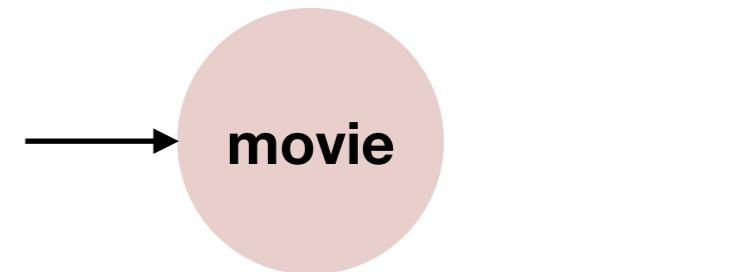


Character Bi-LSTM

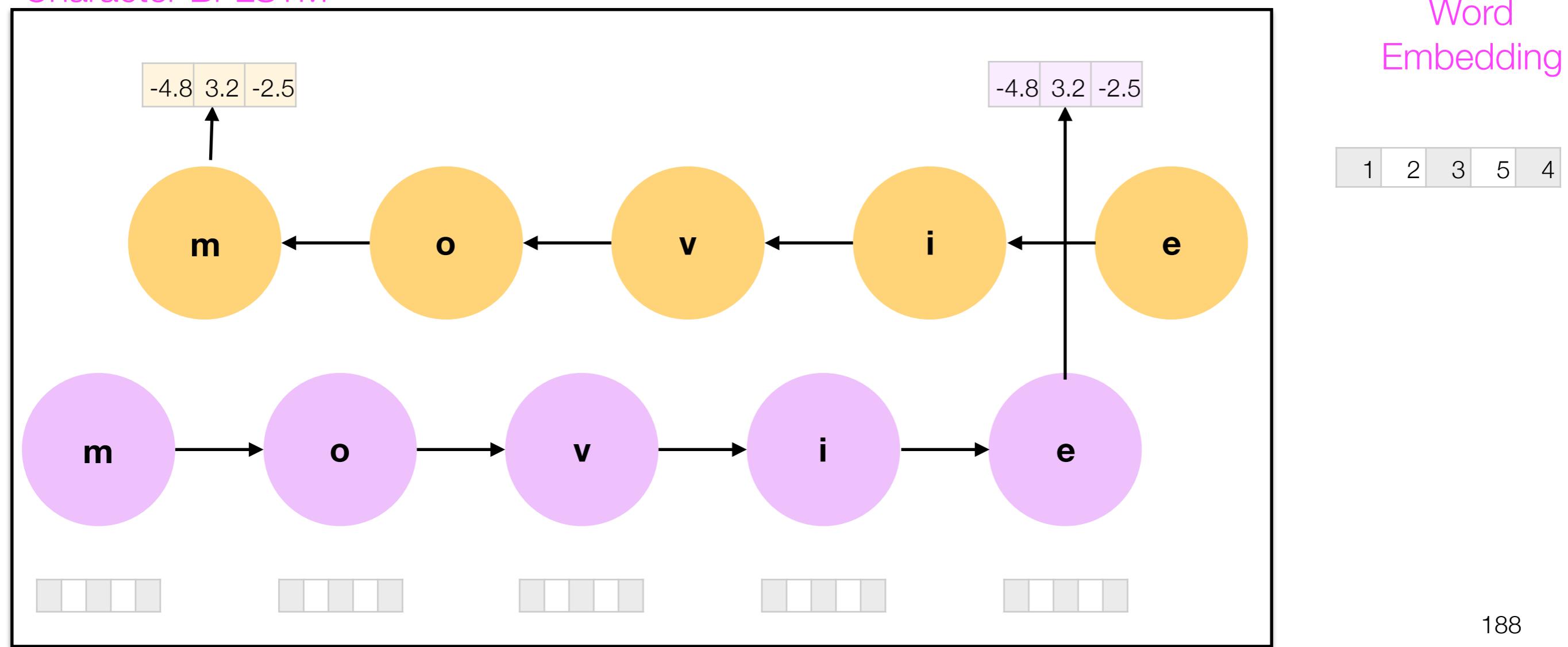


BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

Lample et al. (2016), “Neural Architectures for Named Entity Recognition”

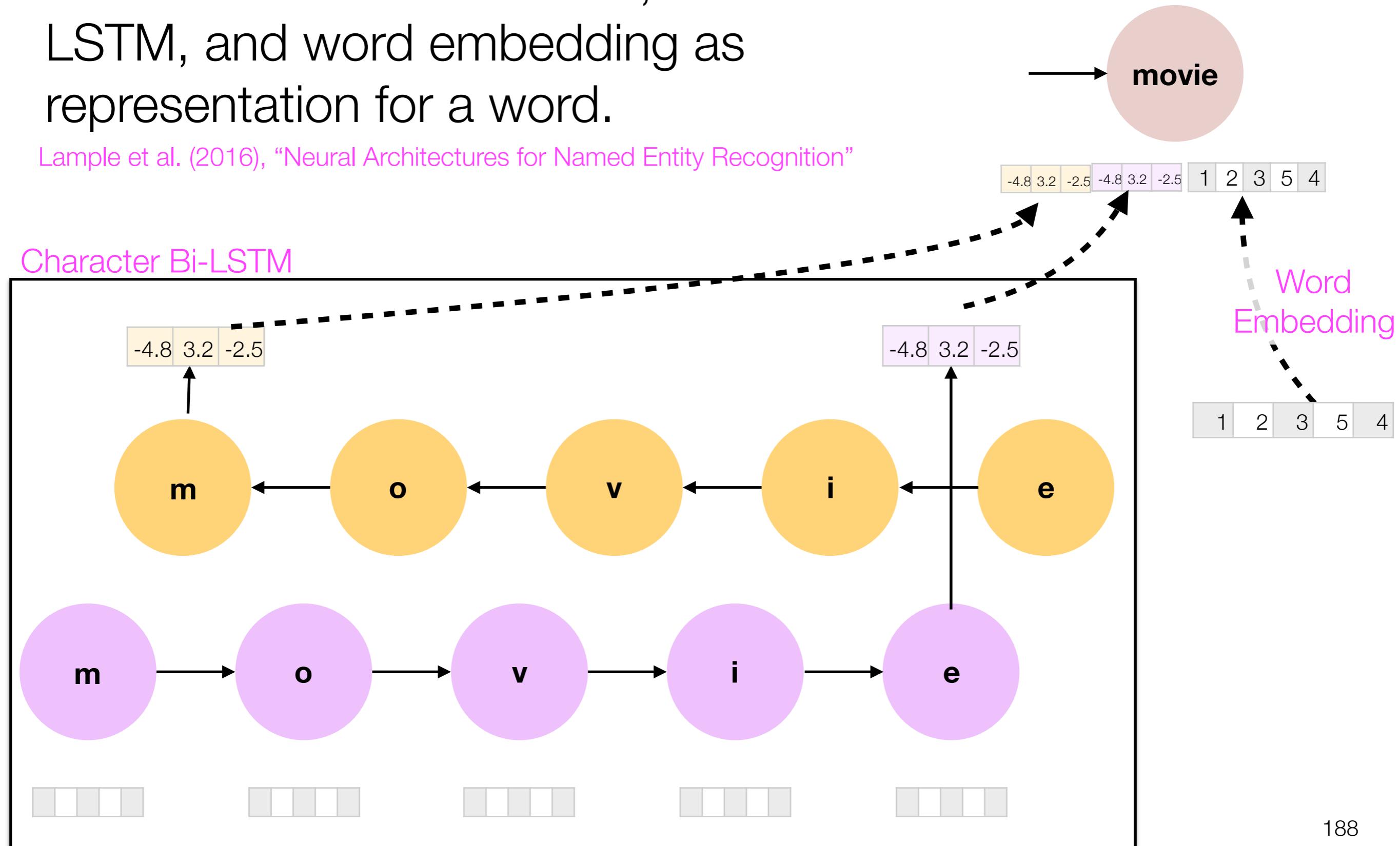


Character Bi-LSTM



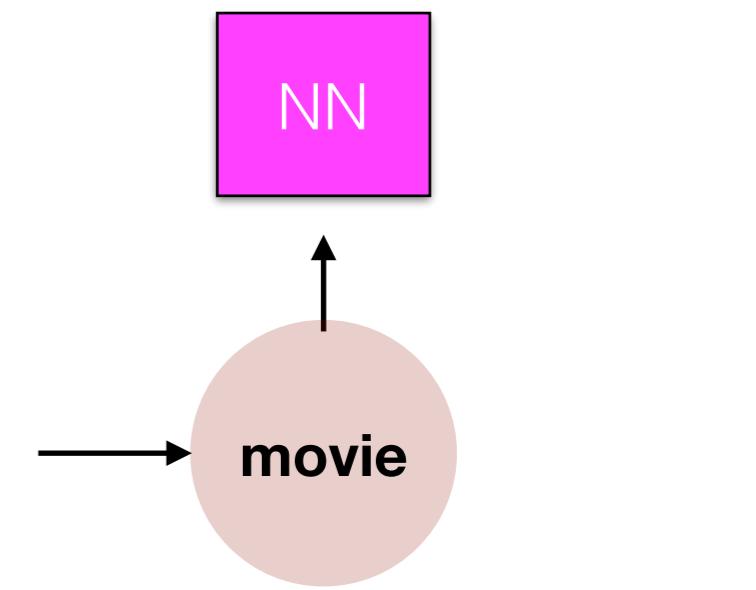
BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

Lample et al. (2016), “Neural Architectures for Named Entity Recognition”

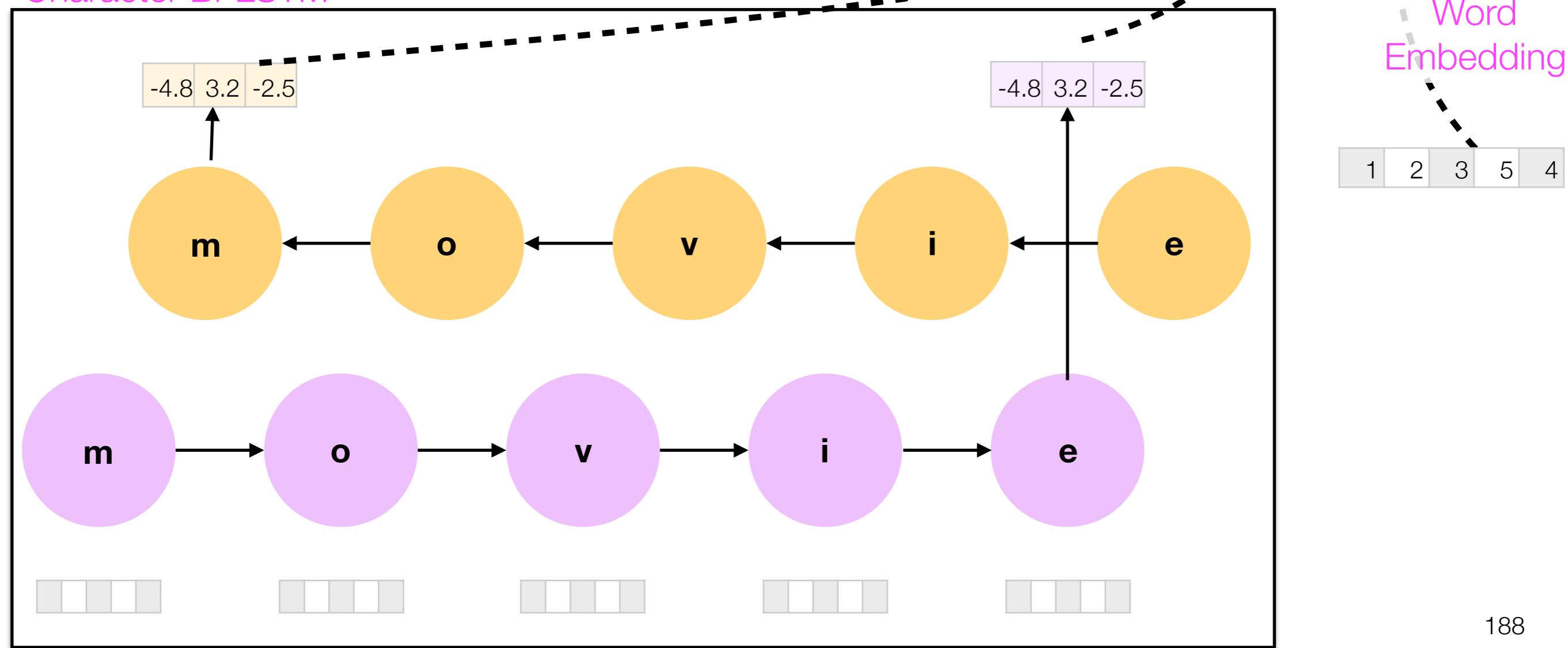


BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

Lample et al. (2016), “Neural Architectures for Named Entity Recognition”

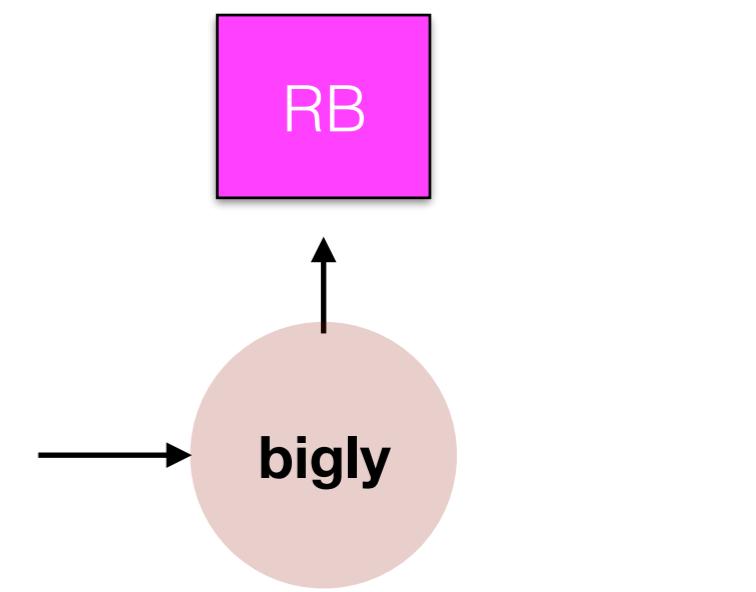


Character Bi-LSTM

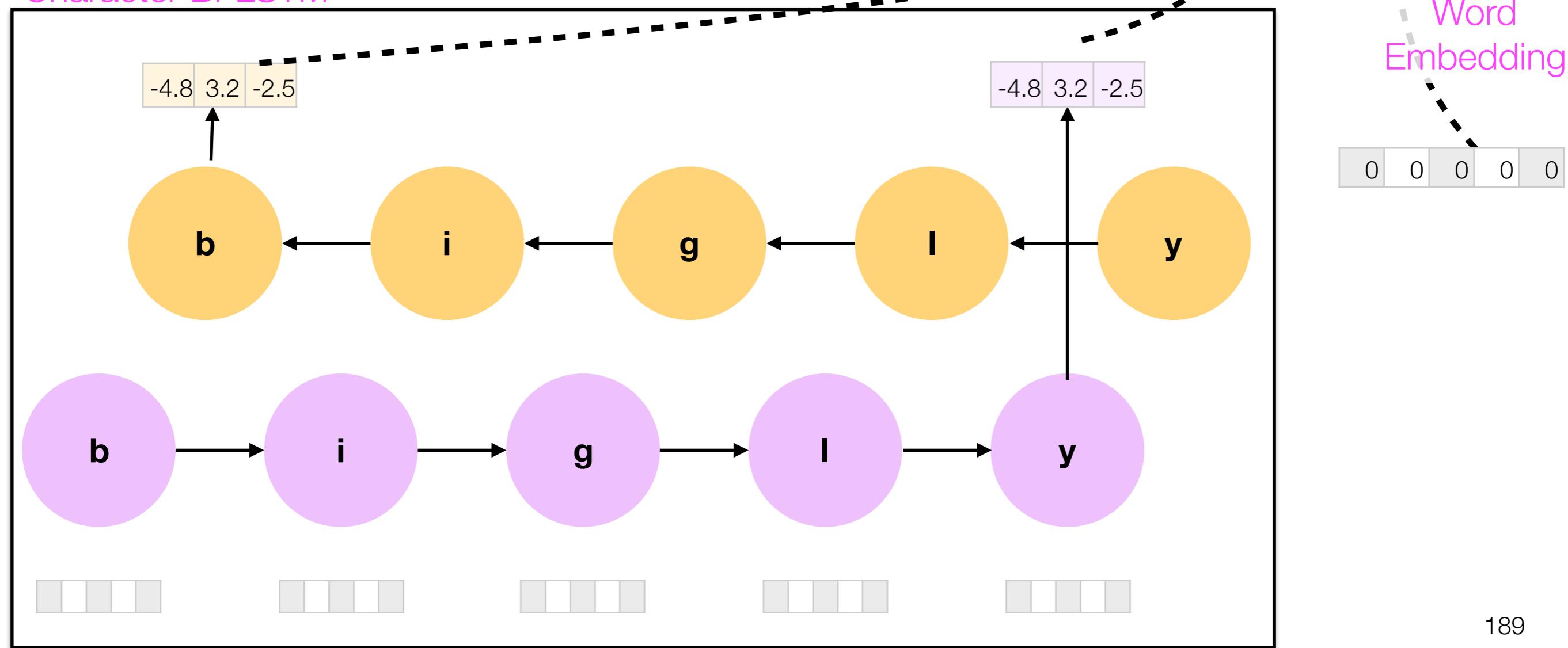


BiLSTM for each word; concatenate final state of forward LSTM, backward LSTM, and word embedding as representation for a word.

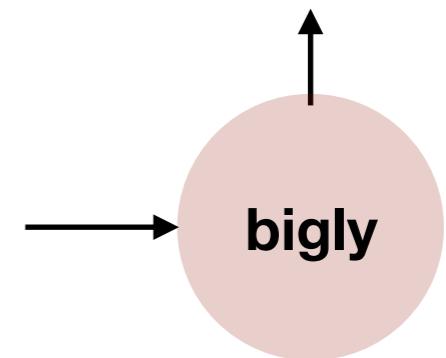
Lample et al. (2016), “Neural Architectures for Named Entity Recognition”



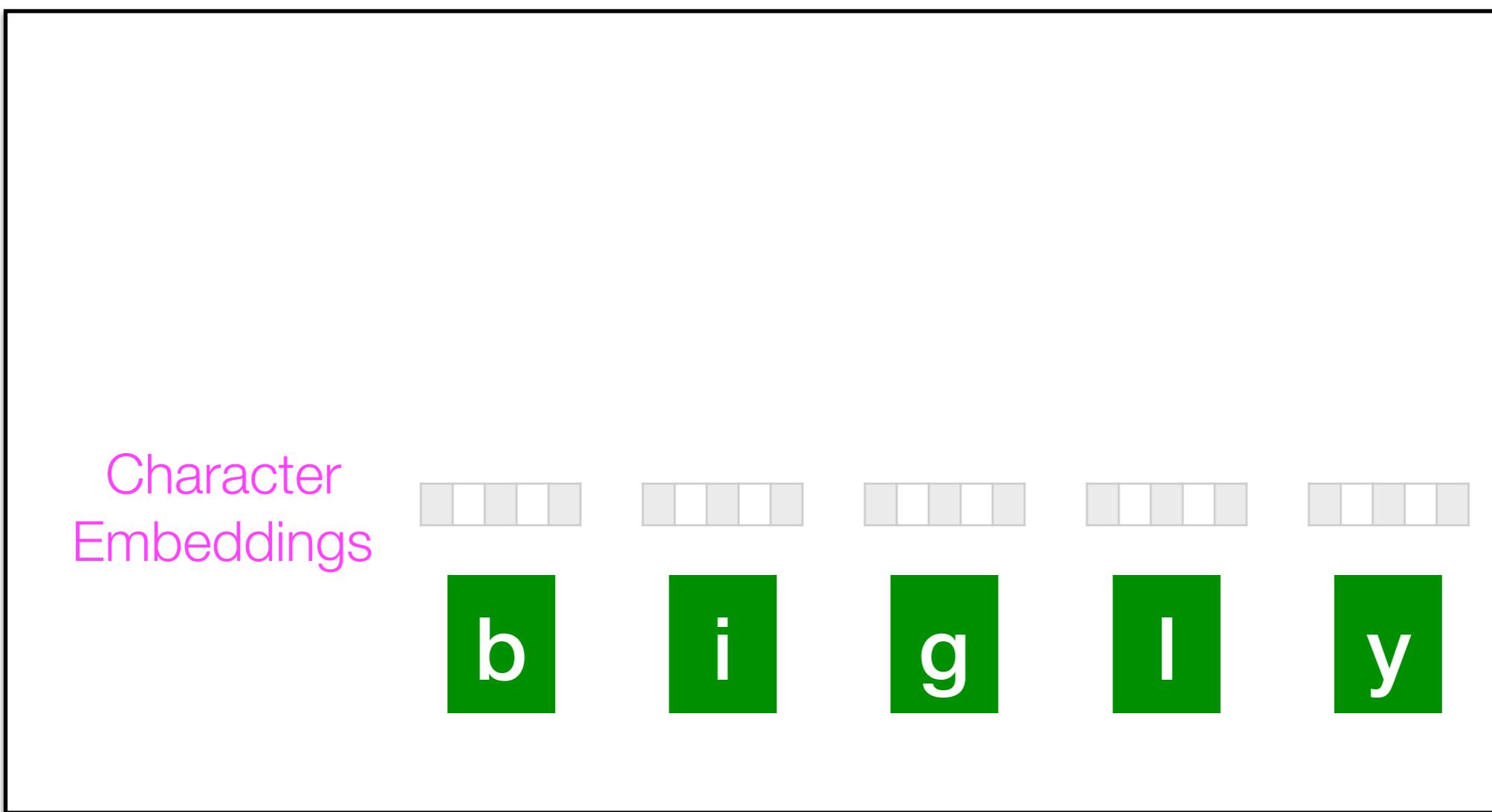
Character Bi-LSTM



Character CNN for each word;
concatenate character CNN output
and word embedding as
representation for a word.



Chu et al. (2016), “Named Entity Recognition with Bidirectional LSTM-CNNs”

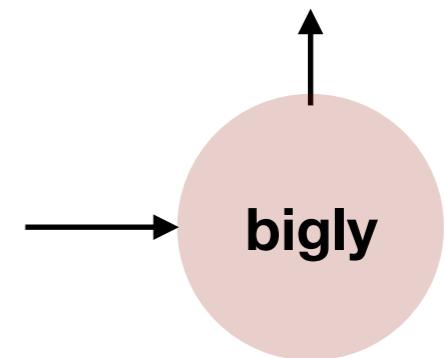


0 0 0 0 0

Word
Embedding

0 0 0 0 0

Character CNN for each word;
concatenate character CNN output
and word embedding as
representation for a word.

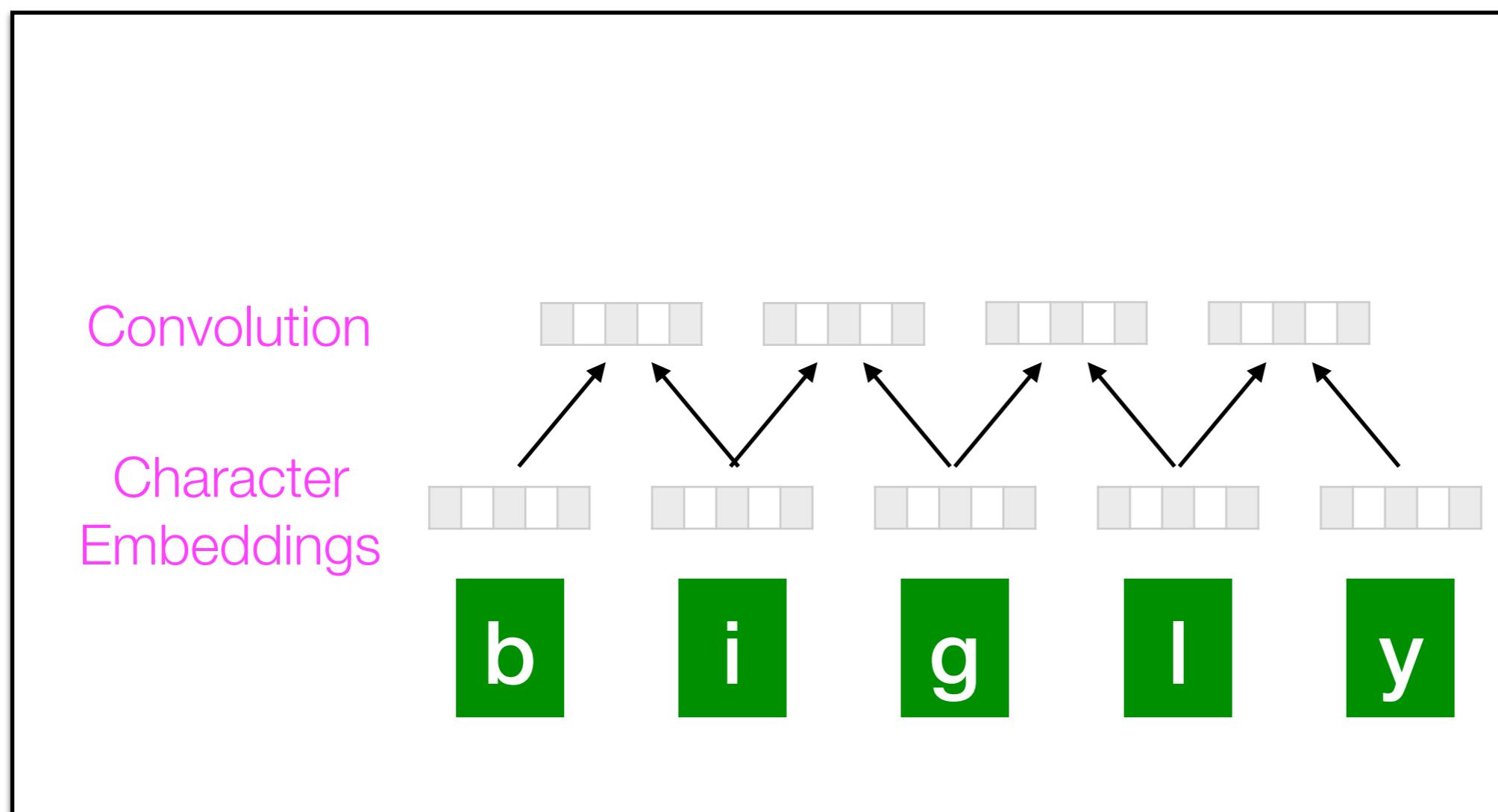


0 0 0 0 0

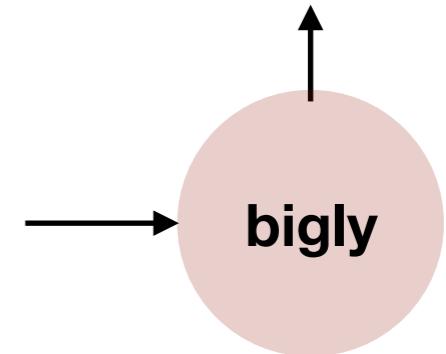


Word
Embedding

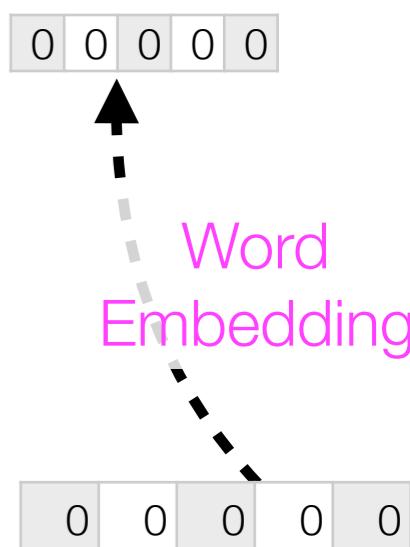
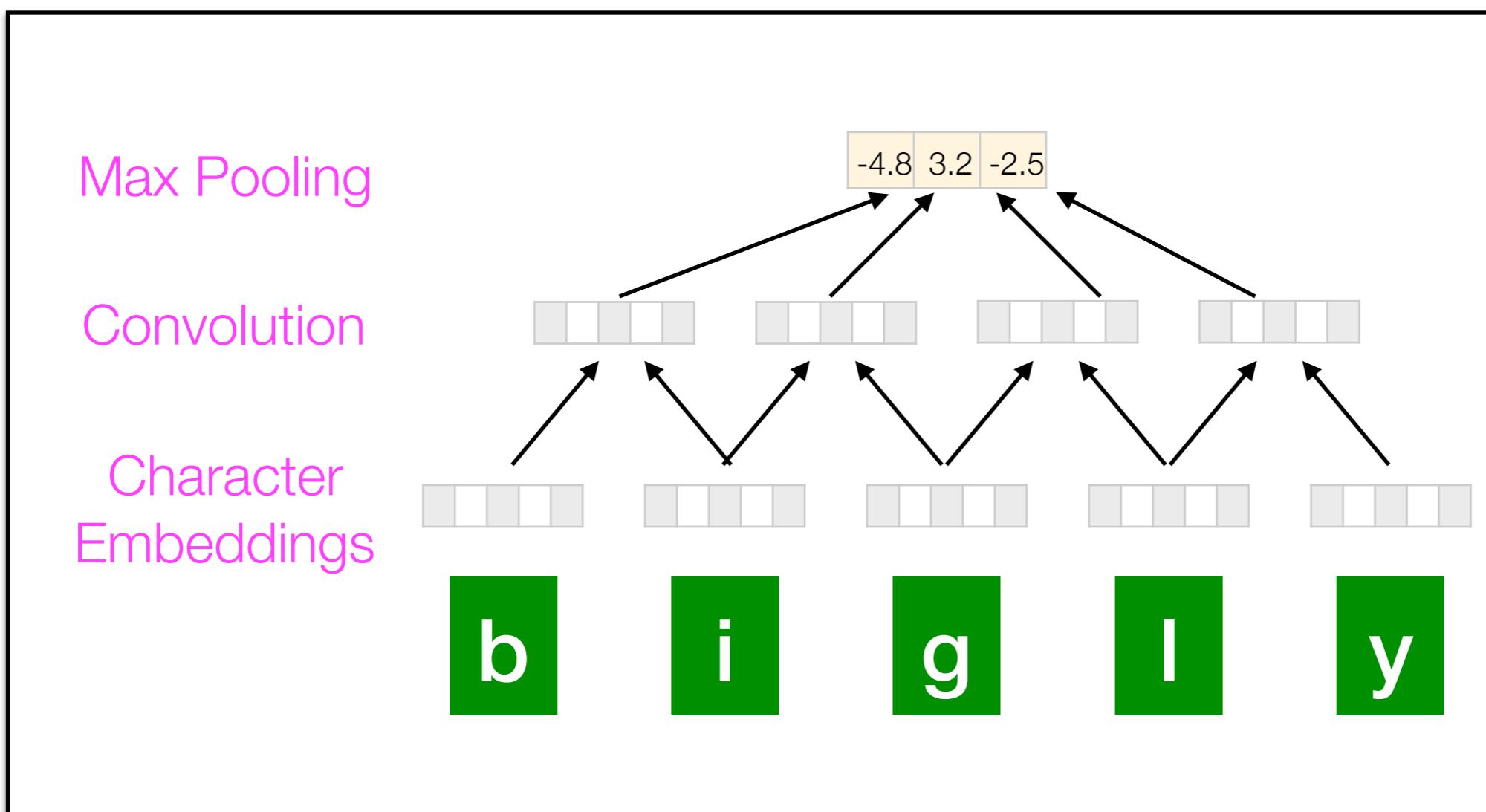
0 0 0 0 0



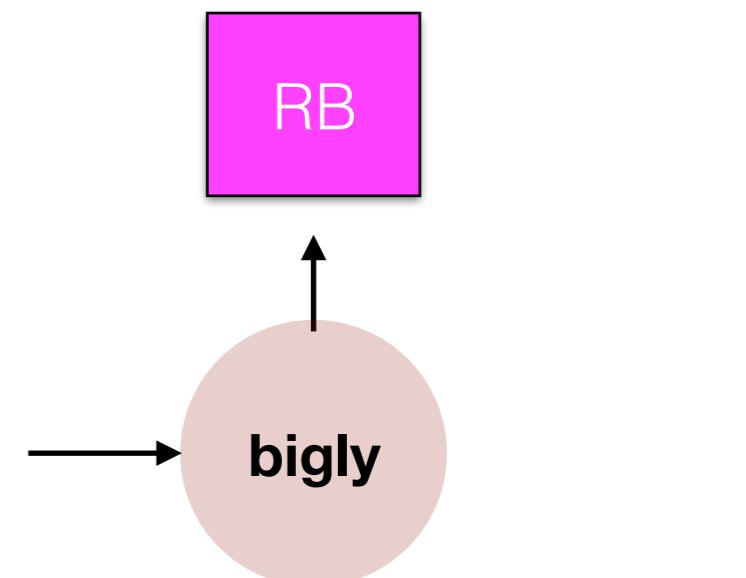
Character CNN for each word;
concatenate character CNN output
and word embedding as
representation for a word.



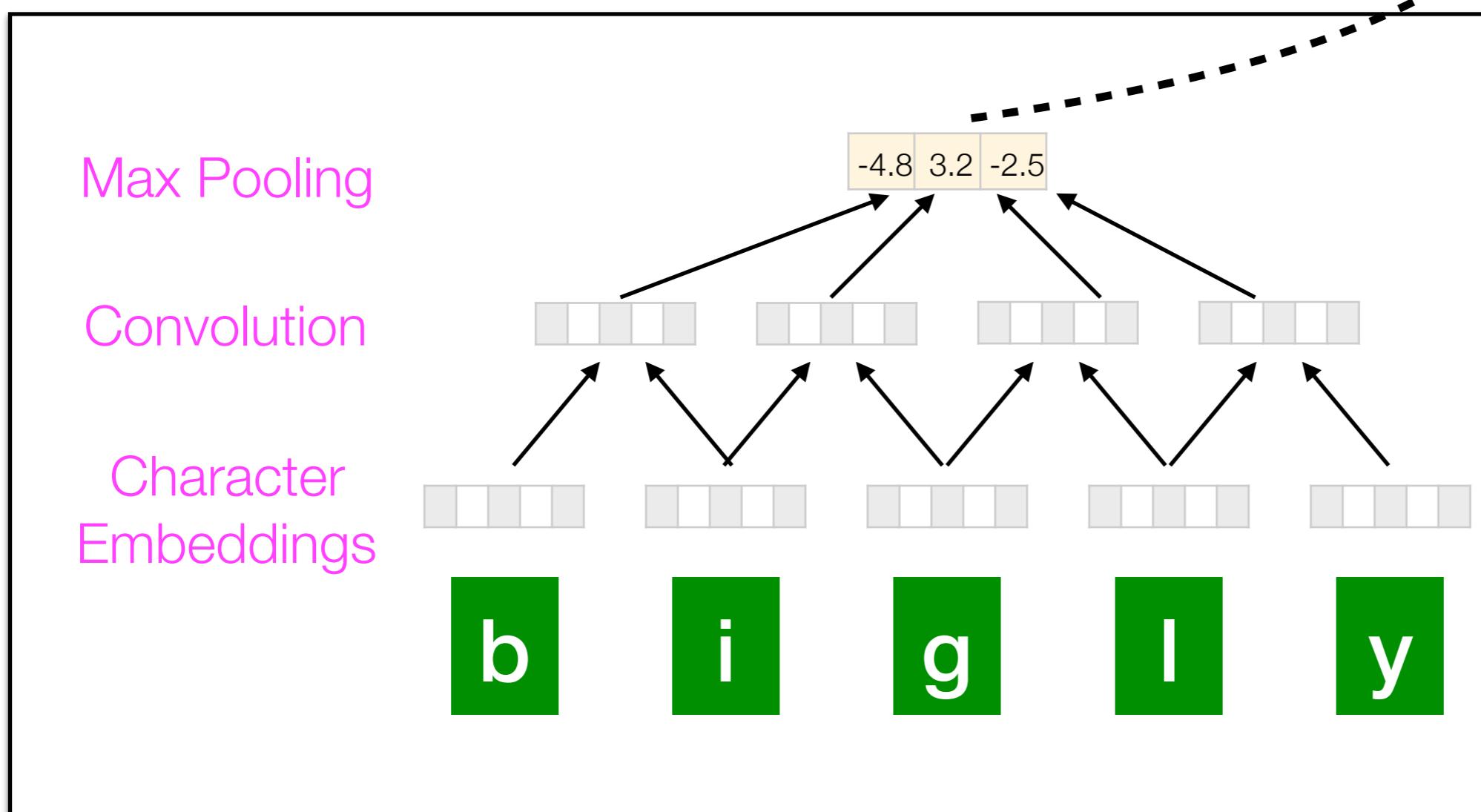
Chu et al. (2016), “Named Entity Recognition with Bidirectional LSTM-CNNs”



Character CNN for each word;
concatenate character CNN output
and word embedding as
representation for a word.



Chu et al. (2016), “Named Entity Recognition with Bidirectional LSTM-CNNs”





Building deep networks in practice is easier than you think

```
print('Loading data...')
(X_train, y_train), (X_test, y_test) = imdb.load_data(nb_words=max_features,
                                                       test_split=0.2)
print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128, input_length=maxlen, dropout=0.2))
model.add(LSTM(128, dropout_W=0.2, dropout_U=0.2)) # try using a GRU instead, for fun
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

Building deep networks in practice is easier than you think

```
print('Loading data...')
(X_train, y_train), (X_test, y_test) = imdb.load_data(nb_words=max_features,
                                                       test_split=0.2)
print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128, input_length=maxlen, dropout=0.2))
model.add(LSTM(128, dropout_W=0.2, dropout_U=0.2)) # try using a GRU instead, for fun
model.add(Dense(1))
model.add(Activation('sigmoid'))

# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
               optimizer='adam',
               metrics=['accuracy'])

print('Train...')
print(X_train.shape)
print(y_train.shape)
model.fit(X_train, y_train, batch_size=batch_size, nb_epoch=15,
           validation_data=(X_test, y_test))
score, acc = model.evaluate(X_test, y_test,
                            batch_size=batch_size)
```



You should feel comfortable:

- Manually labeling a sentence with parts of speech
- Understanding a Hidden Markov Model (HMM) and the Viterbi algorithm (Note: you won't have to implement this!)
- What is a Maximum Entropy Markov Model (MEMM)
- The concepts of a Long Short-Term Memory (LSTM) and why this is preferable to a Recurrent Neural Network (RNN)

Homework 2 is due next week!

- Due next Wednesday at 5:30