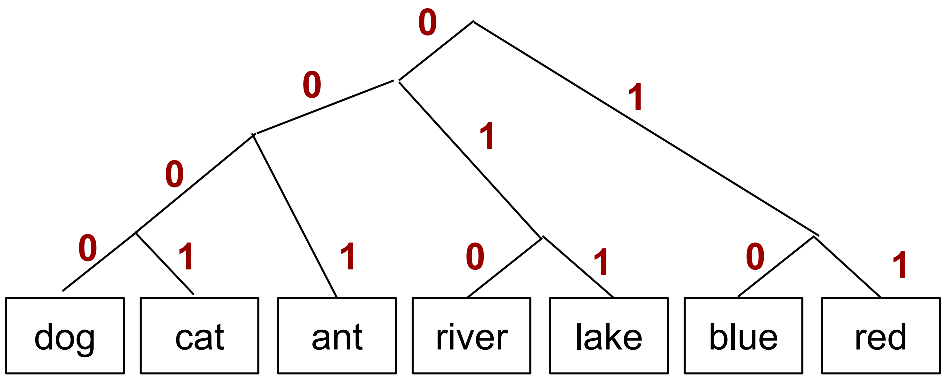
Algorithms for NLP

Word Embeddings

Yulia Tsvetkov – CMU

Slides: Dan Jurafsky – Stanford,
Mike Peters – AI2, Edouard Grave – FAIR

Brown Clustering



dog [0000]

cat [0001]

ant [001]

river [010]

lake [011]

blue [10]

red [11]

Brown Clustering

Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays

June March July April January December October November September August

people guys folks fellows CEOs chaps doubters commies unfortunates blokes

down backwards ashore sideways southward northward overboard aloft downwards adrift

water gas coal liquid acid sand carbon steam shale iron

great big vast sudden mere sheer gigantic lifelong scant colossal

man woman boy girl lawyer doctor guy farmer teacher citizen

American Indian European Japanese German African Catholic Israeli Italian Arab

pressure temperature permeability density porosity stress velocity viscosity gravity tension

mother wife father son husband brother daughter sister boss uncle

machine device controller processor CPU printer spindle subsystem compiler plotter

John George James Bob Robert Paul William Jim David Mike

anyone someone anybody somebody

feet miles pounds degrees inches barrels tons acres meters bytes

director chief professor commissioner commander treasurer founder superintendent dean custodian

liberal conservative parliamentary royal progressive Tory provisional separatist federalist PQ

had hadn't hath would've could've should've must've might've

asking telling wondering instructing informing kidding reminding bothering thanking deposing

that tha theat

head body hands eyes voice arm seat eye hair mouth

[Brown et al, 1992]

Brown Clustering

lawyer	1000001101000
newspaperman	100000110100100
stewardess	100000110100101
toxicologist	10000011010011
slang	1000001101010
babysitter	100000110101100
conspirator	1000001101011010
womanizer	1000001101011011
mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110
...	
Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000
....	
John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	10111001000000001100
Phillip	101110010000000011010
WILLIAM	101110010000000011011
Timothy	10111001000000001110
Terrence	101110010000000011110
Jerald	101110010000000011111
Harold	101110010000000100
Frederic	101110010000000101
Wendell	10111001000000011

Table 1: Sample bit strings

[Miller et al., 2004]

Brown Clustering

- $\mathcal{V}$ is a vocabulary
- $C : \mathcal{V} \rightarrow \{1, 2, \ldots k\}$ is a partition of the vocabulary into *k* clusters
- $p(C(w_i)|C(w_{i-1}))$ is a probability of cluster of w_i to follow the cluster of w_{i-1}
- $p(w_i|C(w_i)) = \dfrac{count(w_i)}{\Sigma_{x \in C(w_i)} count(x)}$

The model:

$$\text{Quality}(C) = \prod_{i=1}^{n} p(w_i|C(w_i))p(C(w_i)|C(w_{i-1}))$$

Quality(C)

- Define

$$P(c, c') = \frac{n(c, c')}{n} \quad P(w) = \frac{n(w)}{n} \quad P(c) = \frac{n(c)}{n}$$

- Then (again from Percy Liang, 2005):

$$\begin{aligned} \text{Quality}(C) &= \sum_{c,c'} P(c, c') \log \frac{P(c, c')}{P(c)P(c')} + \sum_{w} P(w) \log P(w) \\ &= I(C) - H \end{aligned}$$

The first term $I(C)$ is the mutual information between adjacent clusters and the second term H is the entropy of the word distribution. Note that the quality of C can be computed as a sum of mutual information weights between clusters minus the constant H, which does not depend on C. This decomposition allows us to make optimizations.

Slide by Michael Collins

A Naive Algorithm

- We start with $|\mathcal{V}|$ clusters: each word gets its own cluster
- Our aim is to find k final clusters
- We run $|\mathcal{V}| - k$ merge steps:
  - At each merge step we pick two clusters c_i and c_j , and merge them into a single cluster
  - We greedily pick merges such that Quality(C) for the clustering C after the merge step is maximized at each stage
- Cost? Naive = $O(|\mathcal{V}|^5)$. Improved algorithm gives $O(|\mathcal{V}|^3)$: still too slow for realistic values of $|\mathcal{V}|$

Slide by Michael Collins

Brown Clustering Algorithm

- Parameter of the approach is m (e.g., $m = 1000$)
- Take the top m most frequent words, put each into its own cluster, $c_1, c_2, \ldots c_m$
- For $i = (m + 1) \ldots |\mathcal{V}|$
  - Create a new cluster, c_{m+1}, for the i'th most frequent word. We now have $m + 1$ clusters
  - Choose two clusters from $c_1 \ldots c_{m+1}$ to be merged: pick the merge that gives a maximum value for Quality(C). We're now back to m clusters

- Carry out $(m - 1)$ final merges, to create a full hierarchy

- Running time: $O(|\mathcal{V}|m^2 + n)$ where n is corpus length

Slide by Michael Collins

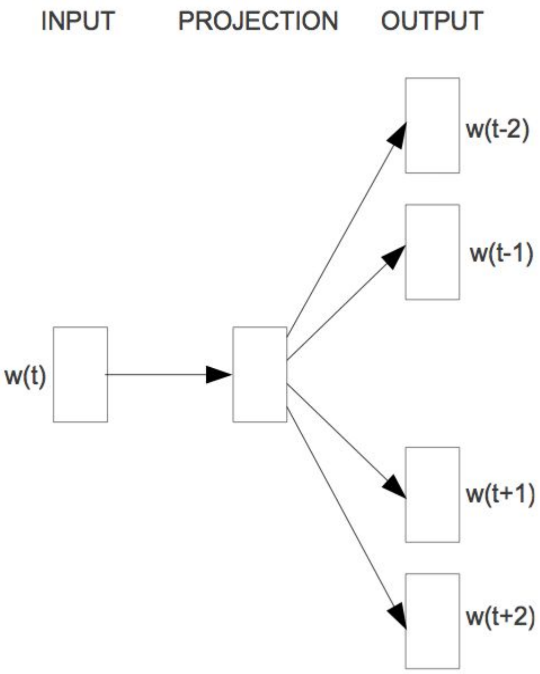
Plan for Today

- Word2Vec
  - Representation is created by training a classifier to distinguish nearby and far-away words
- FastText
  - Extension of word2vec to include subword information
- ELMo
  - Contextual token embeddings
- Multilingual embeddings
- Using embeddings to study history and culture

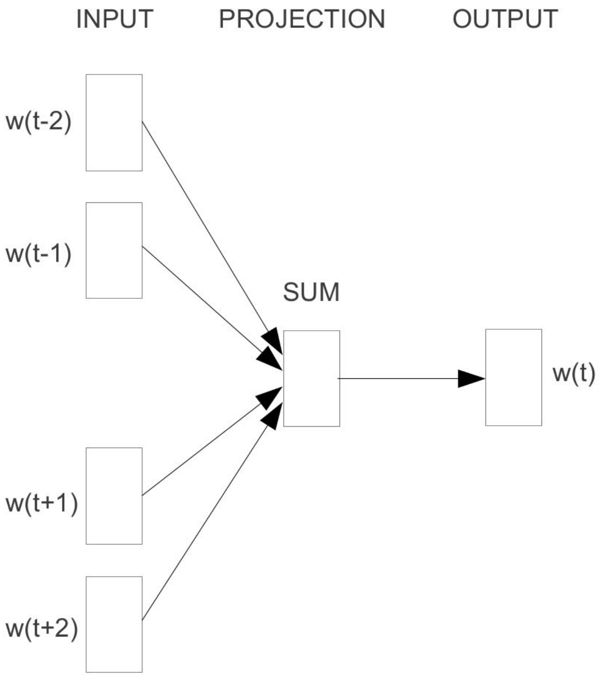
Word2Vec

- Popular embedding method
- Very fast to train
- Code available on the web
- Idea: predict rather than count

Word2Vec



Skip-gram



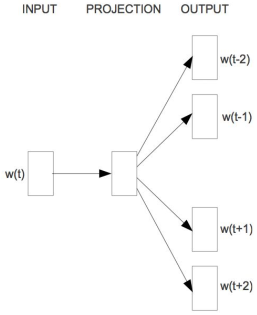
CBOW

[Mikolov et al.’ 13]

Skip-gram Prediction

- Predict vs Count

the cat sat on the mat



Skip-gram

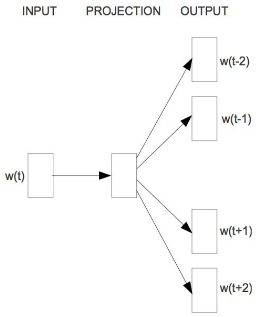
Skip-gram Prediction

- Predict vs Count

<u>the</u> cat sat on the mat

w_t = the → **CLASSIFIER** →

w_{t-2} = <start$_{-2}$>
w_{t-1} = <start$_{-1}$>
w_{t+1} = cat
w_{t+2} = sat

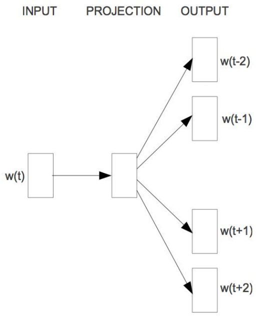


Skip-gram

context size = 2

Skip-gram Prediction

- Predict vs Count



Skip-gram

the cat sat on the mat

w_t = cat → CLASSIFIER →

w_{t-2} = <start$_{-1}$>
w_{t-1} = the
w_{t+1} = sat
w_{t+2} = on

context size = 2

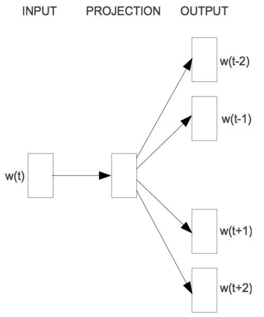
Skip-gram Prediction

- Predict vs Count

the cat $\underline{\text{sat}}$ on the mat

w_t = sat ⟶ **CLASSIFIER** ⟶

w_{t-2} = the
w_{t-1} = cat
w_{t+1} = on
w_{t+2} = the



Skip-gram

context size = 2

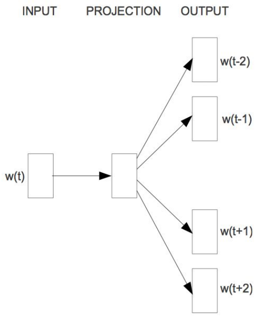
Skip-gram Prediction

- Predict vs Count

the cat sat on the mat

w_t = on → CLASSIFIER →

w_{t-2} = cat
w_{t-1} = sat
w_{t+1} = the
w_{t+2} = mat



Skip-gram

context size = 2

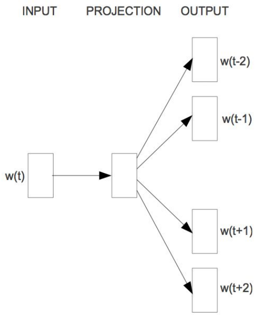
Skip-gram Prediction

- Predict vs Count

the cat sat on the mat

w_t = the → **CLASSIFIER** →

w_{t-2} = sat
w_{t-1} = on
w_{t+1} = mat
w_{t+2} = <end$_{+1}$>



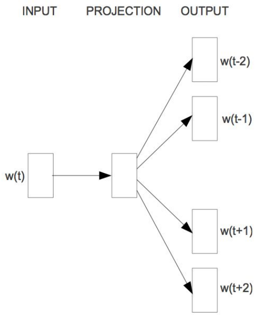
Skip-gram

context size = 2

Skip-gram Prediction

- Predict vs Count

the cat sat on the <u>mat</u>



Skip-gram

w_t = mat → **CLASSIFIER** →

w_{t-2} = on
w_{t-1} = the
w_{t+1} = <end$_{+1}$>
w_{t+2} = <end$_{+2}$>

context size = 2

Skip-gram Prediction

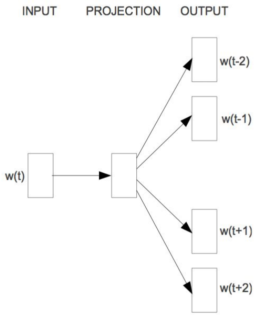
- Predict vs Count

w_t = the → **CLASSIFIER** →

w_{t-2} = sat
w_{t-1} = on
w_{t+1} = mat
w_{t+2} = <end$_{+1}$>

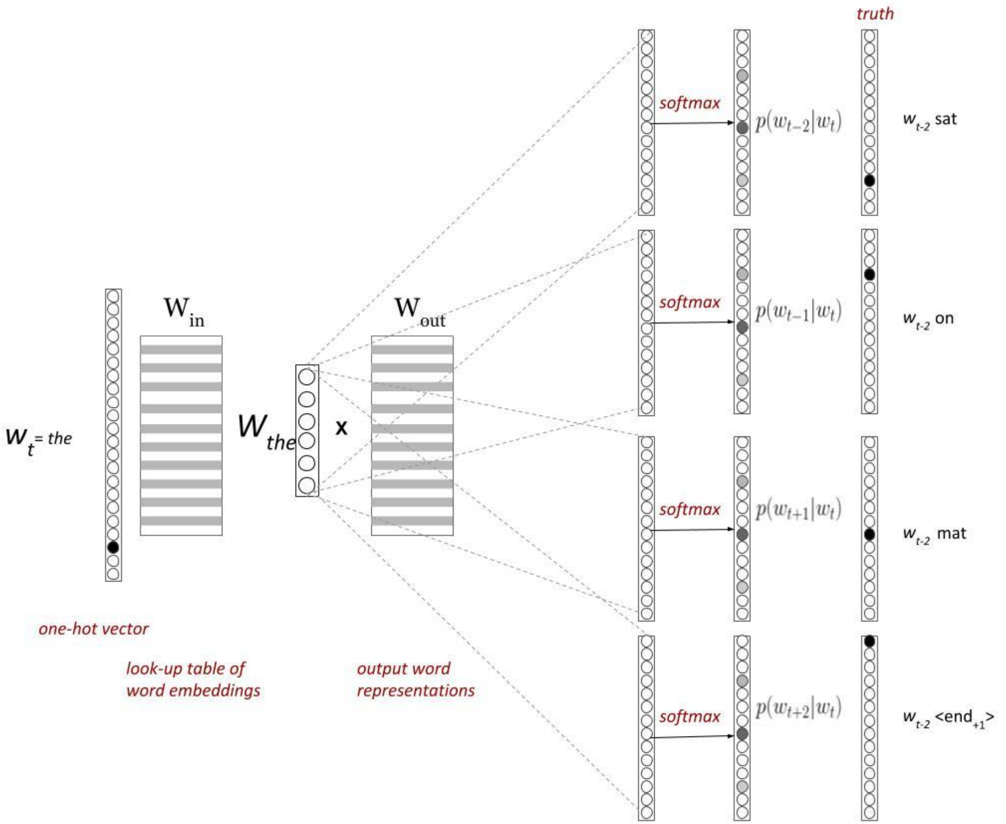
w_t = the → **CLASSIFIER** →

w_{t-2} = <start$_{-2}$>
w_{t-1} = <start$_{-1}$>
w_{t+1} = cat
w_{t+2} = sat



Skip-gram

Skip-gram Prediction

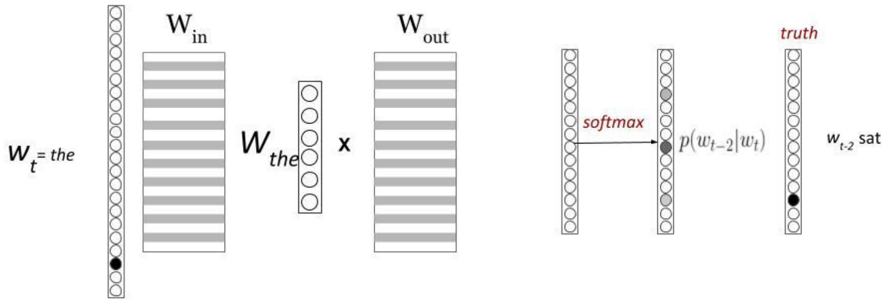


Skip-gram Prediction

- Training data

w_t , w_{t-2}
w_t , w_{t-1}
w_t , w_{t+1}
w_t , w_{t+2}
...

Skip-gram Prediction



- For each word in the corpus *t= 1 … T*

$$J(\Theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} p(w_{t+j} | w_t; \Theta)$$

$$J(\Theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t; \Theta)$$

Maximize the probability of any context window given the current center word

Skip-gram Prediction

- Softmax

$$softmax(x_i) = \frac{e^{x_i}}{\Sigma_j e^{x_j}}$$

SGNS

- Negative Sampling
  - Treat the target word and a neighboring context word as positive examples.
    - subsample very frequent words
  - Randomly sample other words in the lexicon to get negative samples
    - x2 negative samples

Given a tuple (t,c) = target, context

- (cat, sat)
- (cat, aardvark)

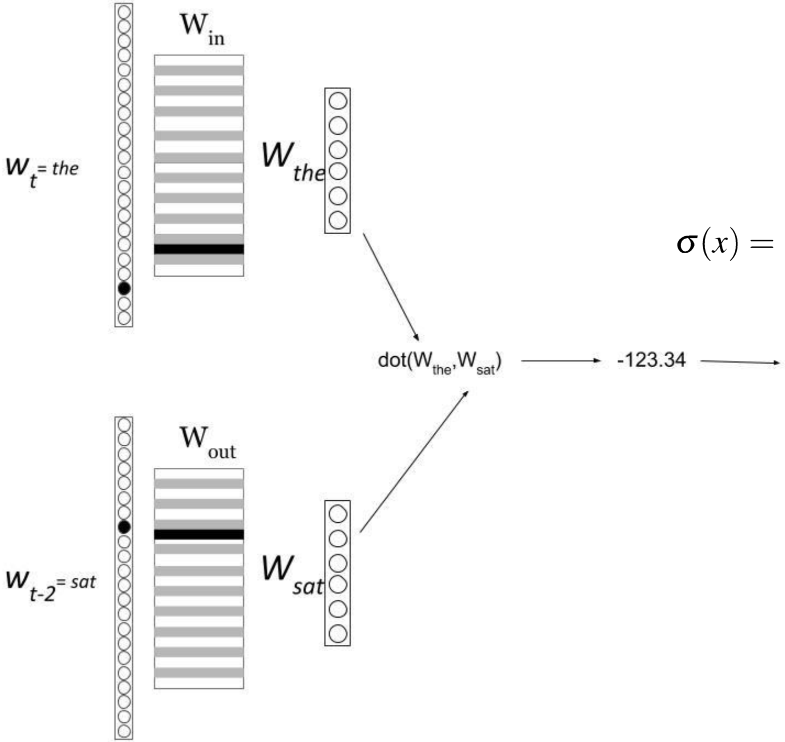
Learning the classifier

- Iterative process
  - We'll start with 0 or random weights
  - Then adjust the word weights to
    - make the positive pairs more likely
    - and the negative pairs less likely
  - over the entire training set:

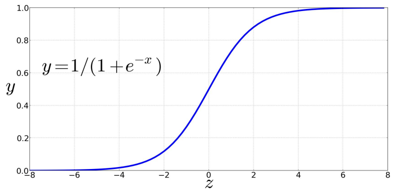
$$\sum_{(t,c)\in +} logP(+|t,c) + \sum_{(t,c)\in -} logP(-|t,c)$$

- Train using gradient descent

How to compute p(+|t,c)?



$$\sigma(x) = \frac{1}{1+e^{-x}}$$



SGNS

Given a tuple (t,c) = target, context

- (cat, sat)
- (cat, aardvark)

Return probability that c is a real context word:

$$P(+|t,c) = \frac{1}{1+e^{-t\cdot c}}$$

$$\begin{aligned} P(-|t,c) &= 1 - P(+|t,c) \\ &= \frac{e^{-t\cdot c}}{1+e^{-t\cdot c}} \end{aligned}$$

Choosing noise words

Could pick w according to their unigram frequency P(w)

More common to chosen then according to $p_\alpha(w)$

$$P_\alpha(w) = \frac{count(w)^\alpha}{\sum_w count(w)^\alpha}$$

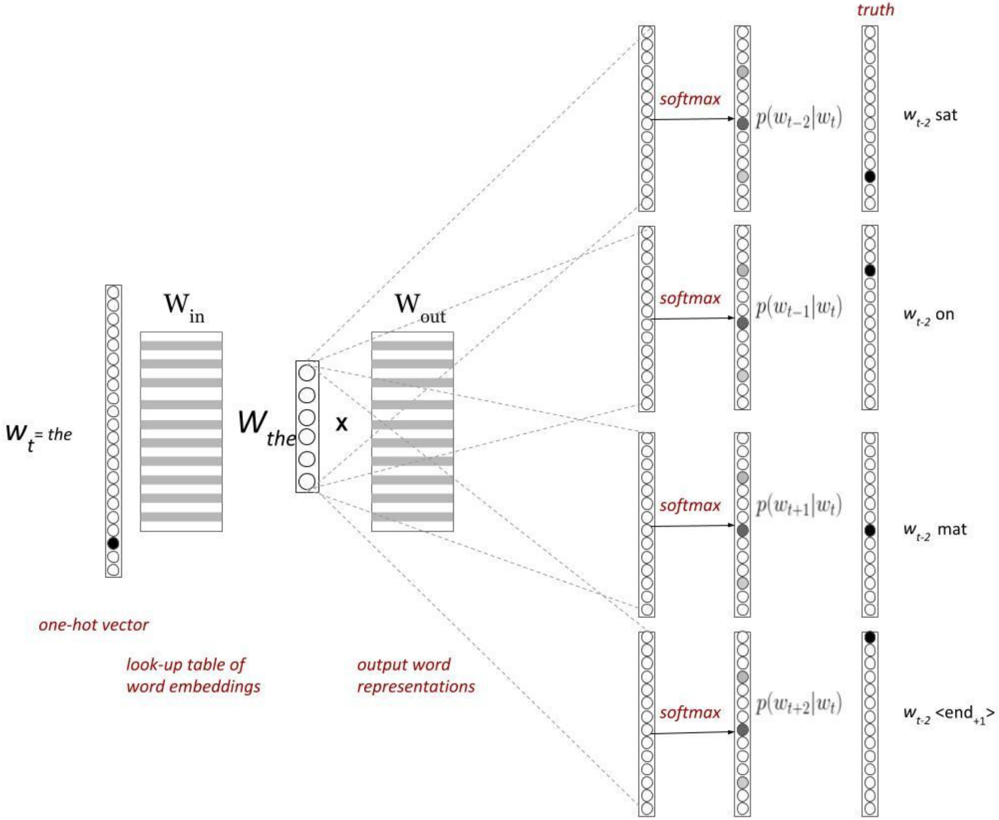
α= ¾ works well because it gives rare noise words slightly higher probability

To show this, imagine two events p(a)=.99 and p(b) = .01:

$$P_\alpha(a) = \frac{.99^{.75}}{.99^{.75} + .01^{.75}} = .97$$

$$P_\alpha(b) = \frac{.01^{.75}}{.99^{.75} + .01^{.75}} = .03$$

Skip-gram Prediction



FastText

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and **Edouard Grave*** and **Armand Joulin** and **Tomas Mikolov**
Facebook AI Research
{bojanowski,egrave,ajoulin,tmikolov}@fb.com

https://fasttext.cc/

FastText: Motivation

Much’ananayakapushasqakupuniñataqsunamá

Much’a -na -naya -ka -pu -sha -sqa -ku -puni -ña -taq -suna -má

“So they really always have been kissing each other then”

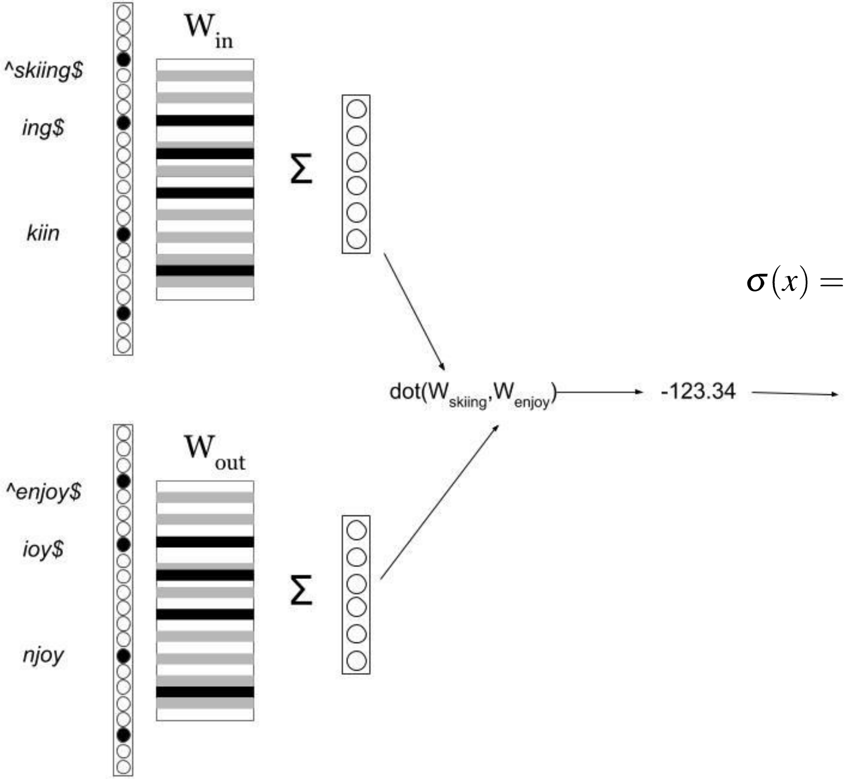
Much’a	to kiss
-na	expresses obligation, lost in translation
-naya	expresses desire
-ka	diminutive
-pu	reflexive (kiss *eachother*)
-sha	progressive (kiss*ing*)
-sqa	declaring something the speaker has not personally witnessed
-ku	3rd person plural (they kiss)
-puni	definitive (really*)
-ña	always
-taq	statement of contrast (...then)
-suna	expressing uncertainty (So...)
-má	expressing that the speaker is surprised

	Singular+neut	Plural+neut	
Nominative	предложение	предложения	sentence (s)
Genitive	предложения	предложений	(of) sentence (s)
Dative	предложению	предложениям	(to) sentence (s)
Accusative	предложение	предложения	sentence (s)
Instrumental	предложением	предложениями	(by) sentence (s)
Prepositional	предложении	предложениях	(in/at) sentence (s)

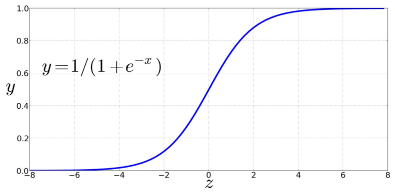
Subword Representation

skiing = `{^skiing$, ^ski, skii, kiin, iing, ing$}`

FastText



$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Details

- *n*-grams between 3 and 6 characters
- how many possible ngrams?
  - $|\text{character set}|^n$
  - Hashing to map n-grams to integers in 1 to K=2M
- get word vectors for out-of-vocabulary words using subwords.
- less than 2× slower than word2vec skipgram

- short n-grams (n = 4) are good to capture syntactic information
- longer n-grams (n = 6) are good to capture semantic information

FastText Evaluation

- Intrinsic evaluation

word1	word2	similarity (humans)
vanish	disappear	9.8
behave	obey	7.3
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

similarity (embeddings)
1.1
0.5
0.3
1.7
0.98
0.3

Spearman's rho (human ranks, model ranks)

- Arabic, German, Spanish, French, Romanian, Russian

FastText Evaluation

- All models trained on Wikipedia:

		sg	cbow	ours*	ours
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

Table: Correlation between human judgement and similarity scores. OoV words are represented as null vectors (ours*) or sum of *n*-grams (ours).

[Grave et al, 2017]

FastText Evaluation

	DE		EN		ES	FR
	GUR350	ZG222	WS	RW	WS	RG
Luong et al. (2013)	-	-	64	34	-	-
Qiu et al. (2014)	-	-	65	33	-	-
Soricut and Och (2015)	64	22	**71**	42	47	**67**
Ours	**73**	**43**	**73**	**48**	**54**	**69**
Botha and Blunsom (2014)	56	25	39	30	28	45
Ours	**66**	**34**	**54**	**41**	**49**	**52**

Table: Spearman's rank correlation coefficient between human judgement and model scores for different methods using morphology to learn word representations.

FastText Evaluation

DE	autofahrer	fahr	fahrer	auto
	freundeskreis	kreis	kreis>	<freun
	grundwort	wort	wort>	grund
	sprachschule	schul	hschul	sprach
	tageslicht	licht	gesl	tages
EN	anarchy	chy	<anar	narchy
	monarchy	monarc	chy	<monar
	kindness	ness>	ness	kind
	politeness	polite	ness>	eness>
	unlucky	<un	cky>	nlucky
	lifetime	life	<life	time
	starfish	fish	fish>	star
	submarine	marine	sub	marin
	transform	trans	<trans	form
FR	finirais	ais>	nir	fini
	finissent	ent>	finiss	<finis
	finissions	ions>	finiss	sions>

Table 6: Illustration of most important character n-grams for selected words in three languages. For each word, we show the n-grams that, when removed, result in the most different representation.

ELMo

Deep contextualized word representations

Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark[*], **Kenton Lee**[*], **Luke Zettlemoyer**[†*]
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
[*]Paul G. Allen School of Computer Science & Engineering, University of Washington

https://allennlp.org/elmo

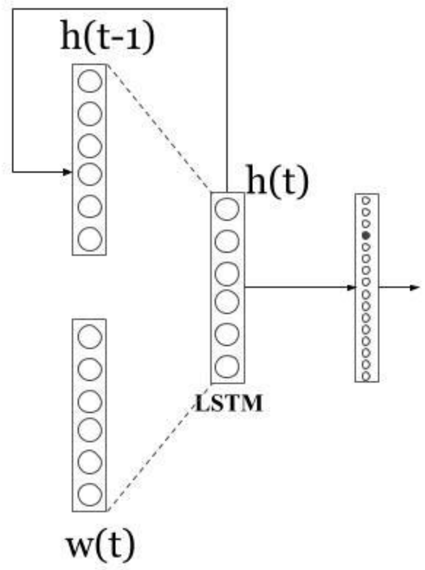
Motivation

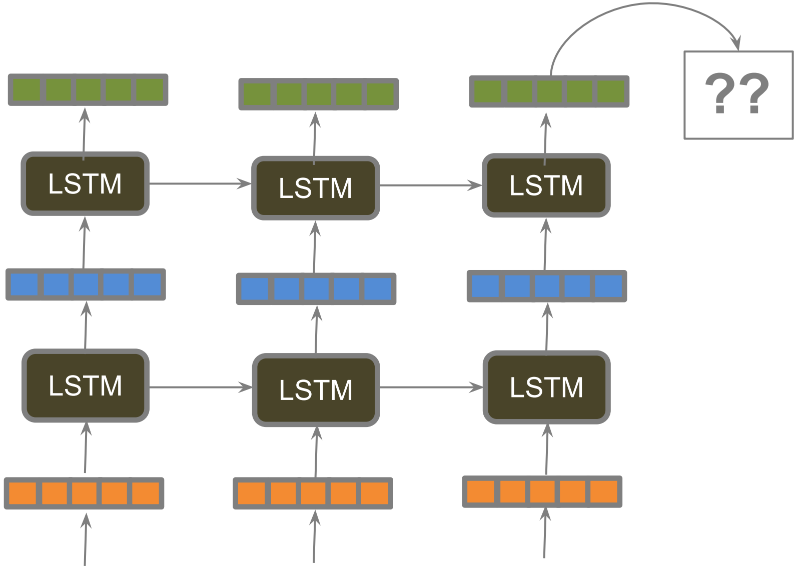
p(play | Elmo and Cookie Monster play a game .)

$\neq$

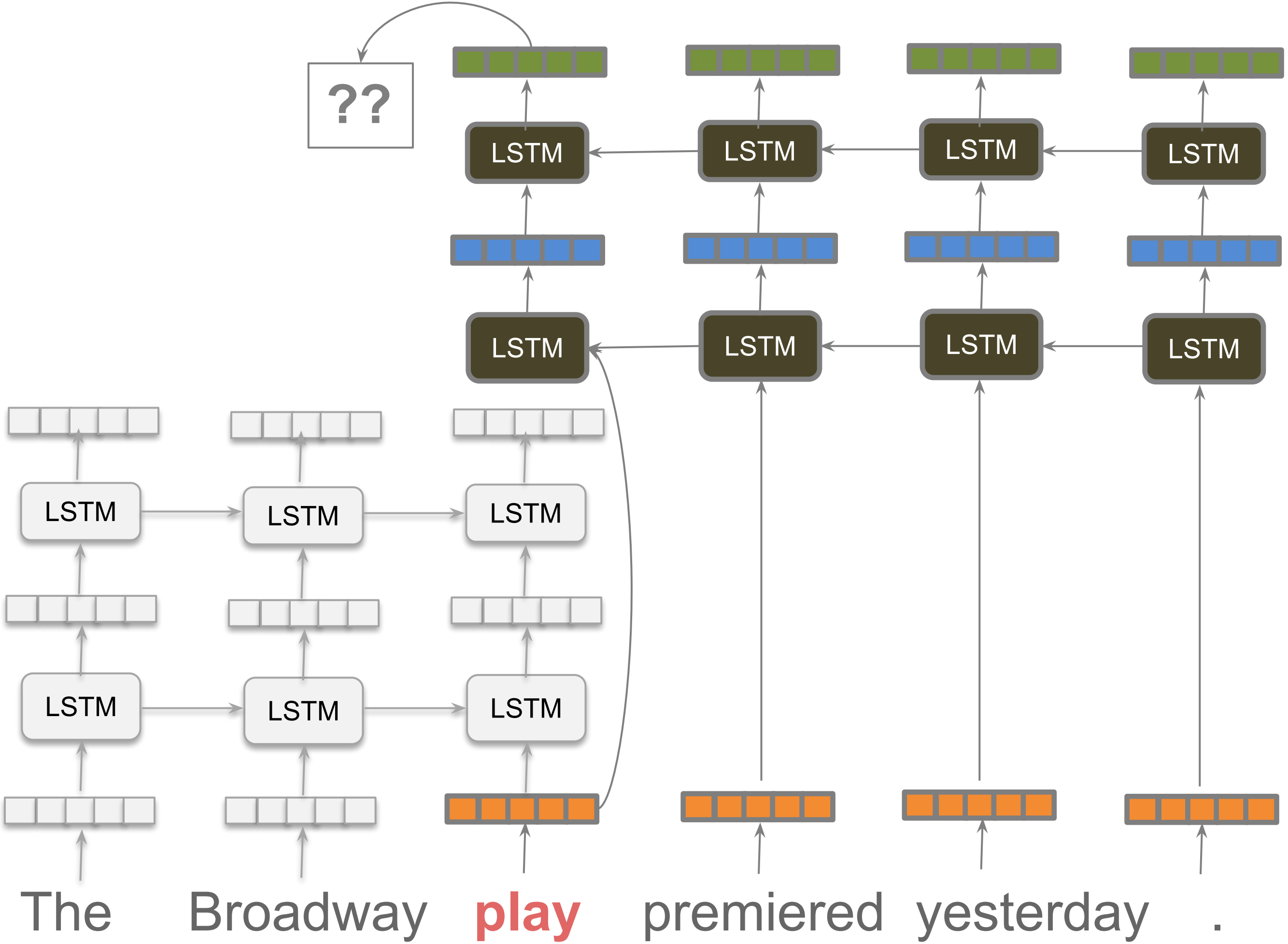
p(play | The Broadway play premiered yesterday .)

Background

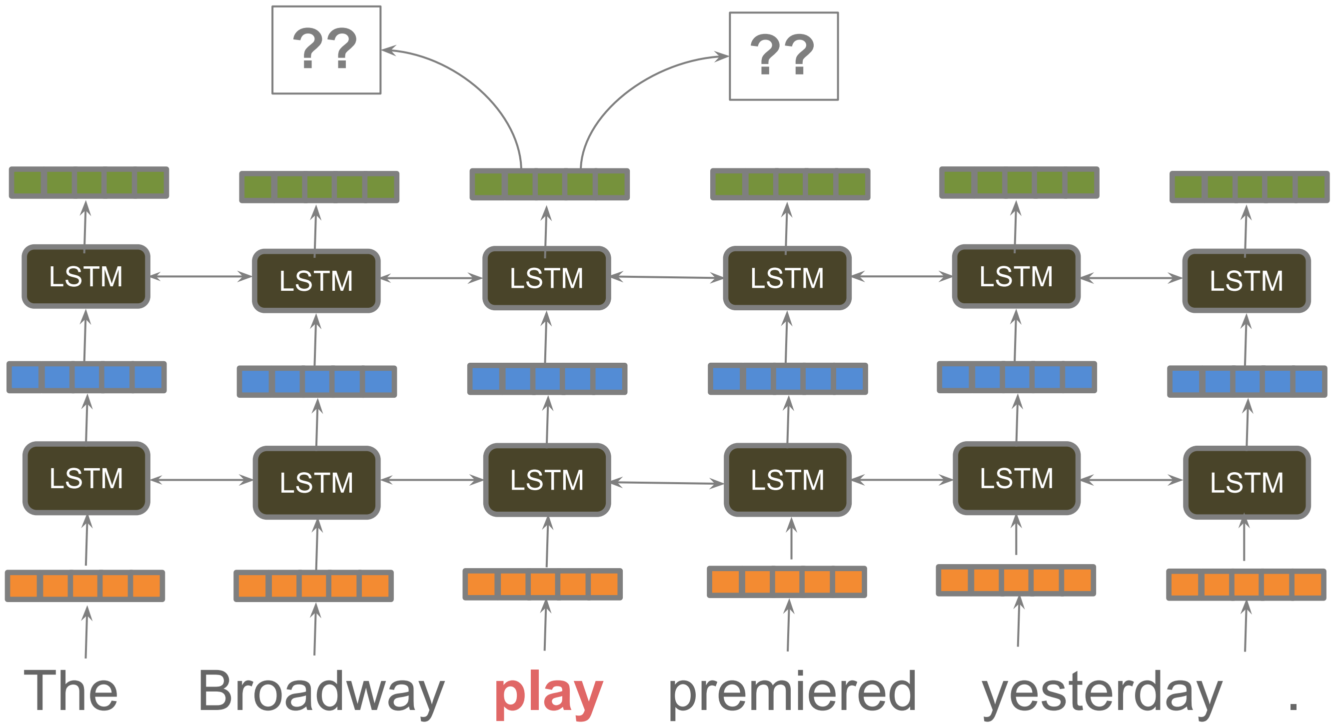




The Broadway **play** premiered yesterday .



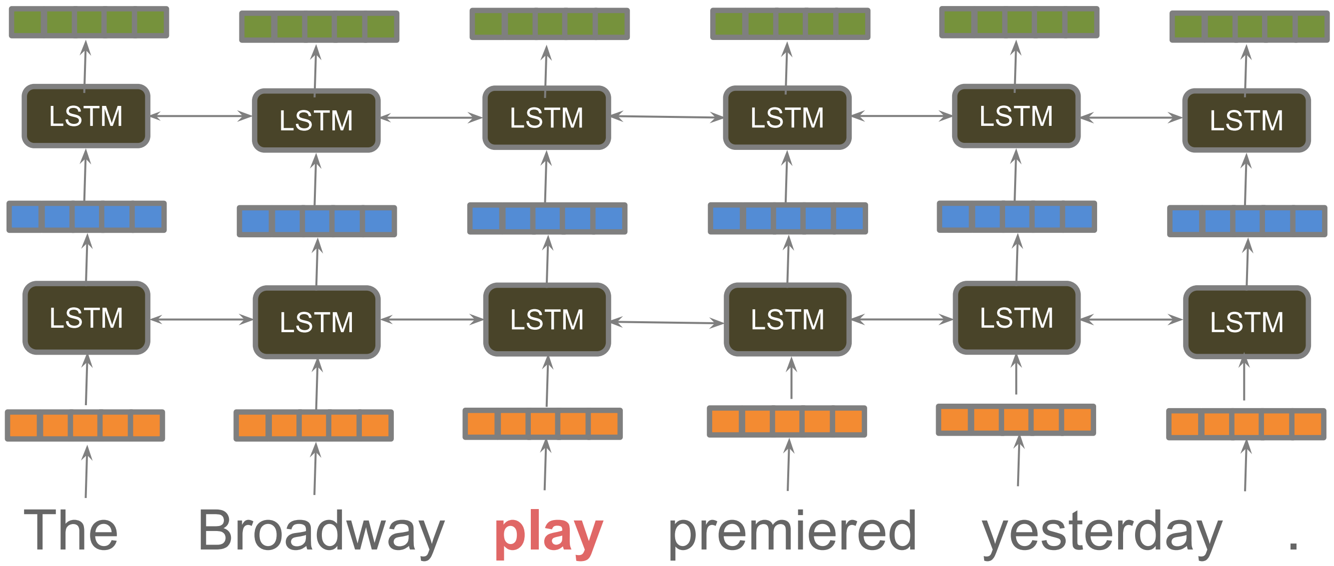
ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE



ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

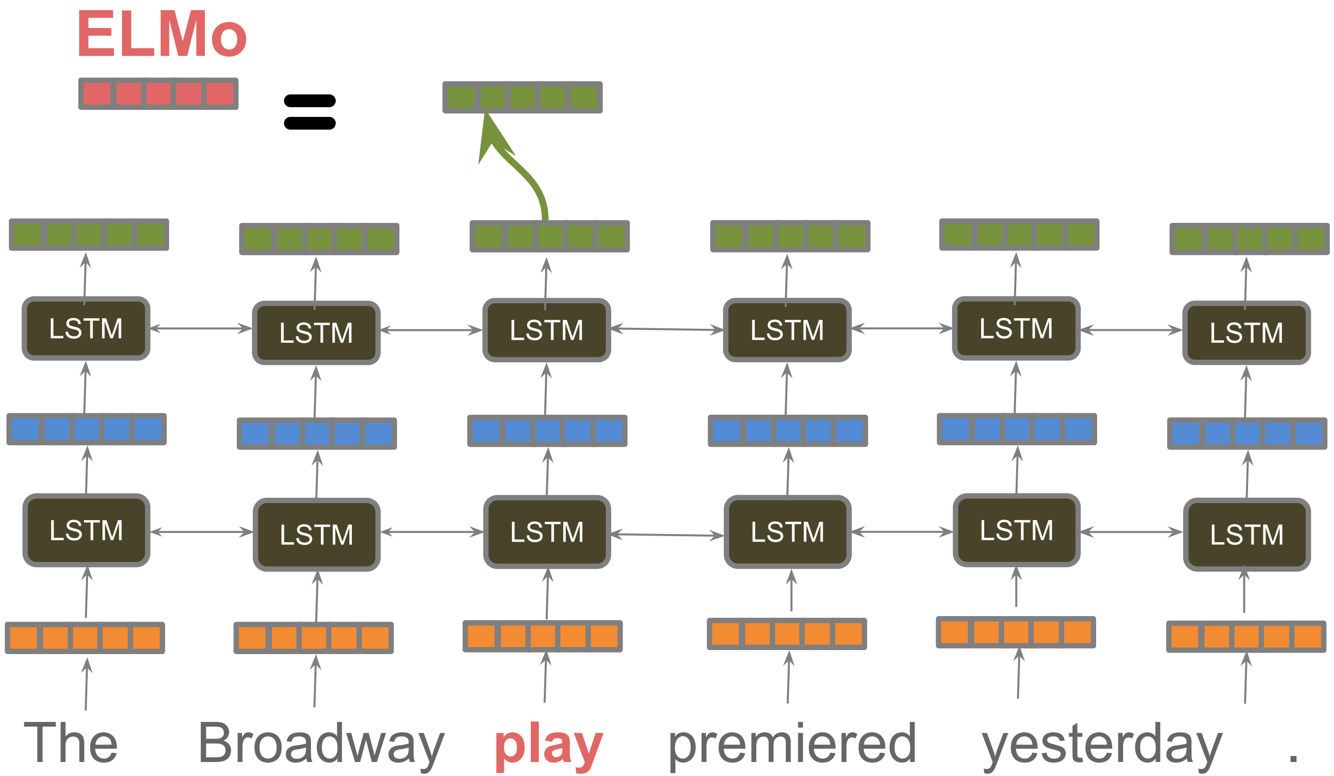
Embeddings from Language Models

ELMo = ??

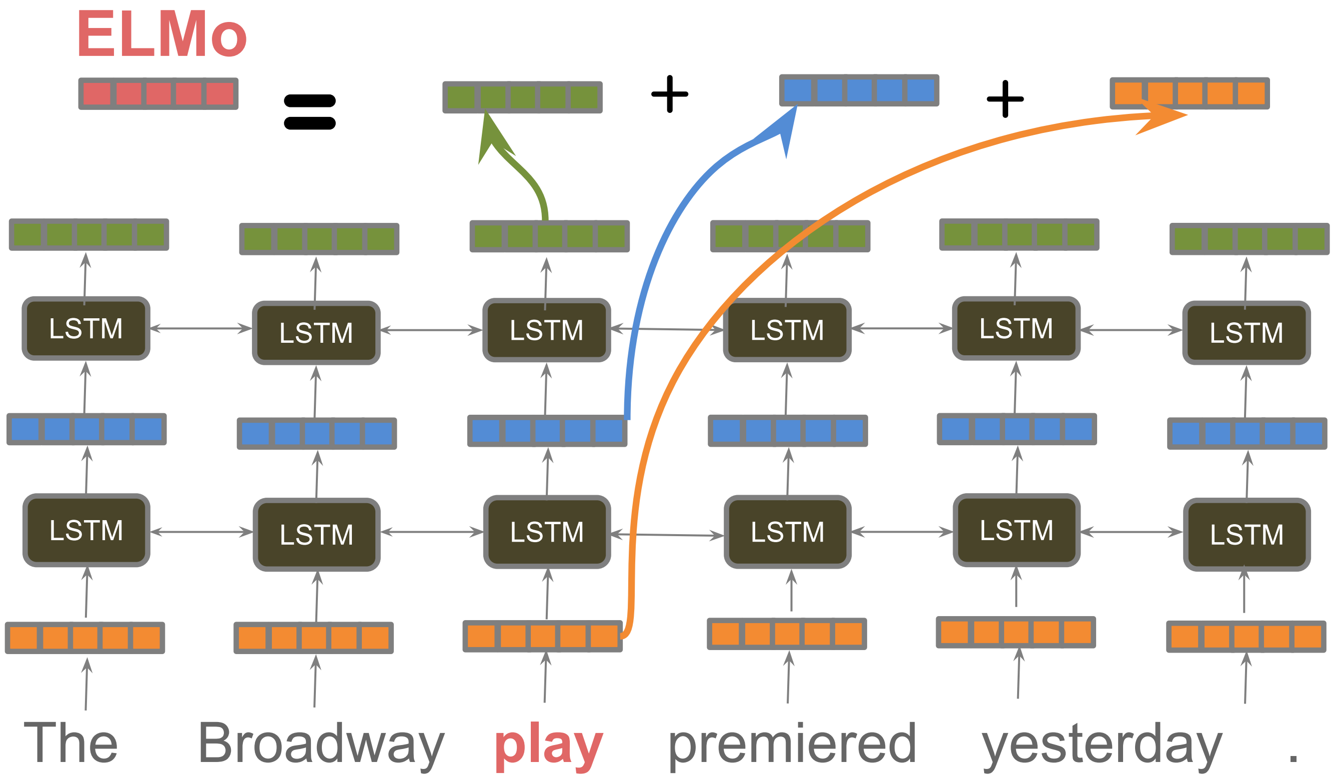


ALLEN INSTITUTE *for* ARTIFICIAL INTELLIGENCE

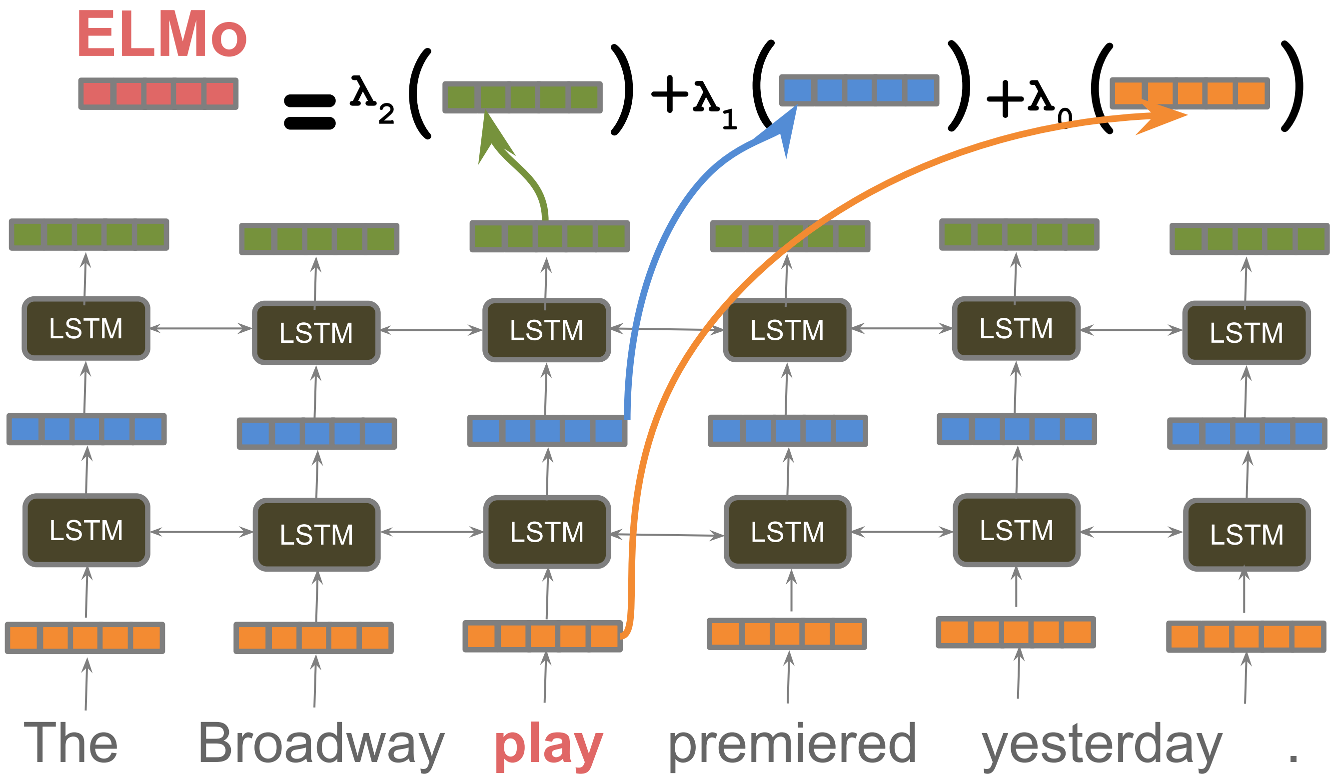
Embeddings from Language Models



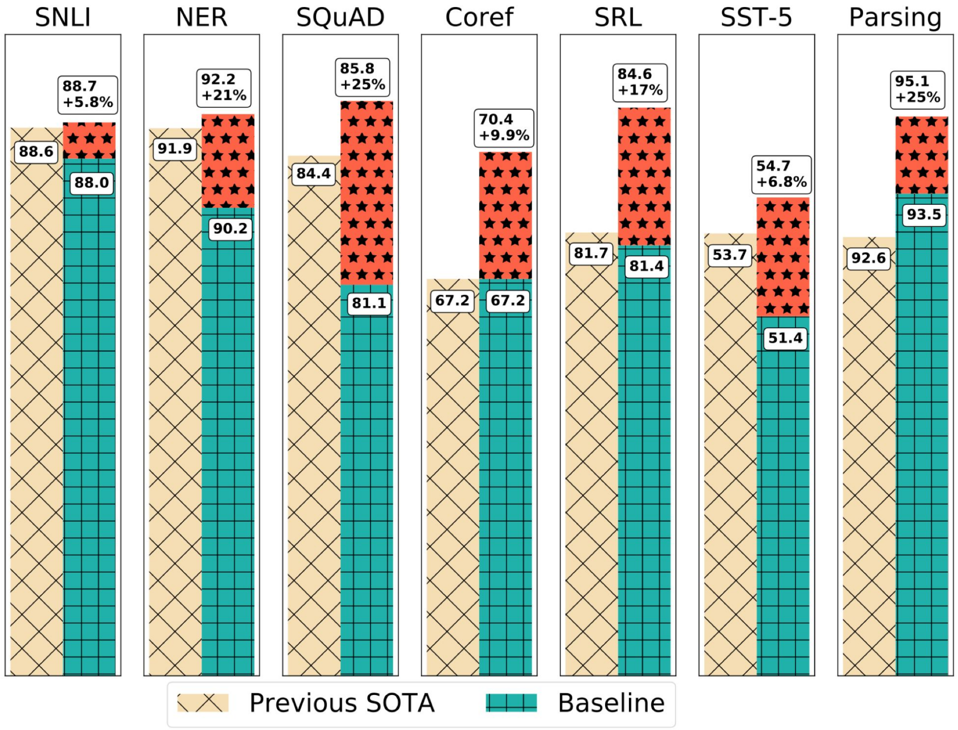
Embeddings from Language Models



Embeddings from Language Models



Evaluation: Extrinsic Tasks



Stanford Question Answering Dataset (SQuAD)

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Figure 1: Question-answer pairs for a sample passage in the SQuAD dataset. Each of the answers is a segment of text from the passage.

[Rajpurkar et al, '16, '18]

SNLI

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

Table 1: Randomly chosen examples from the development section of our new corpus, shown with both the selected gold labels and the full set of labels (abbreviated) from the individual annotators, including (in the first position) the label used by the initial author of the pair.

[Bowman et al, ‘15]

Multilingual Embeddings

Improving Vector Space Word Representations Using Multilingual Correlation

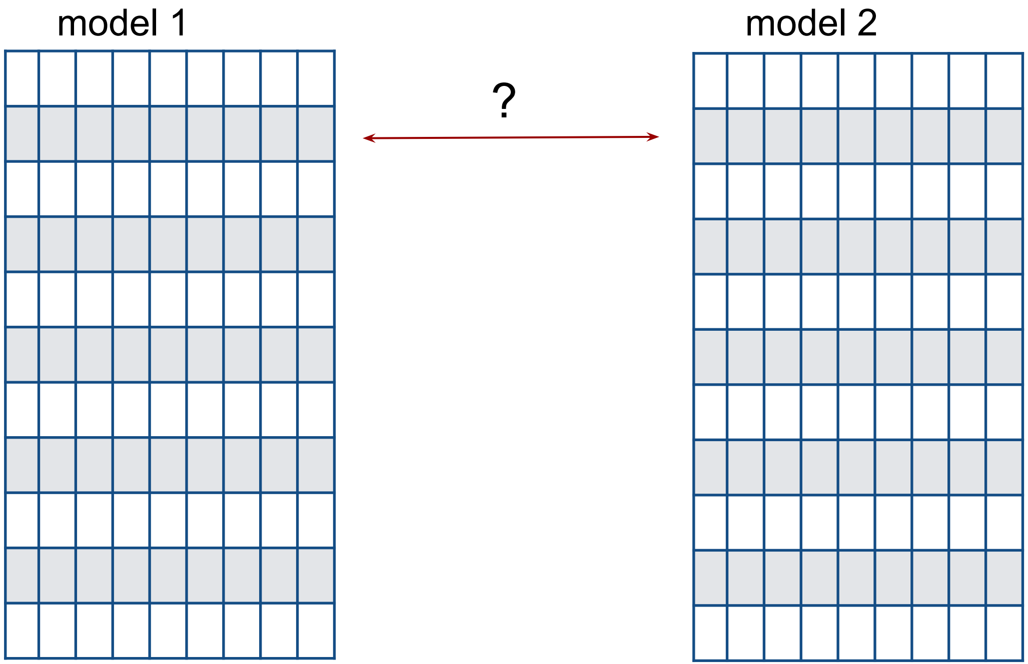
Manaal Faruqui and **Chris Dyer**
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
{mfaruqui, cdyer}@cs.cmu.

Massively Multilingual Word Embeddings

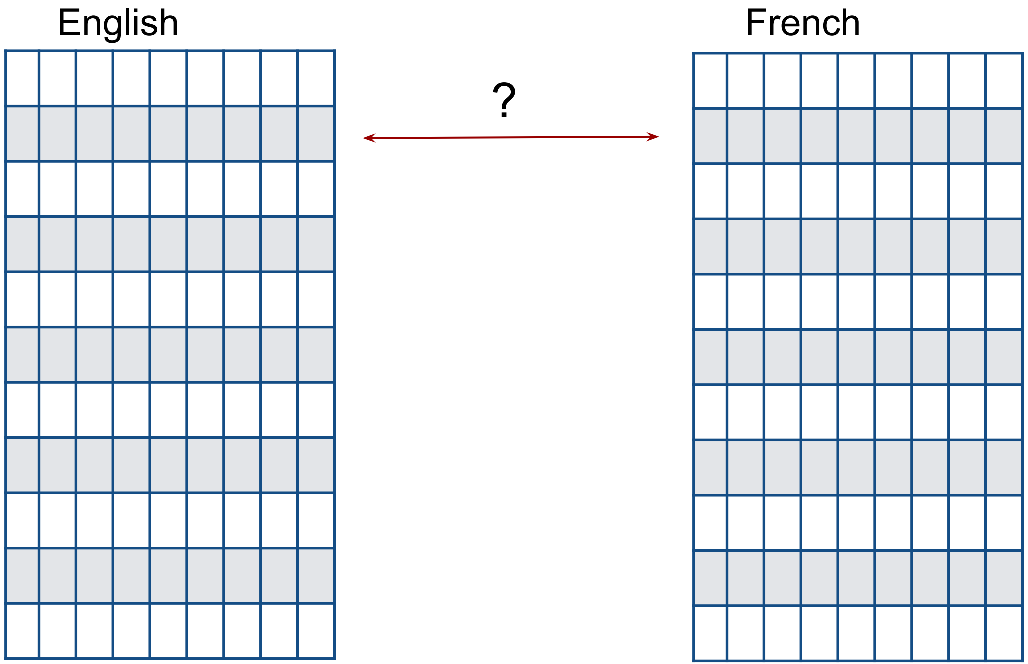
Waleed Ammar[◇] **George Mulcaire**[♡] **Yulia Tsvetkov**[◇]
Guillaume Lample[◇] **Chris Dyer**[◇] **Noah A. Smith**[♡]
[◇]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
[♡]Computer Science & Engineering, University of Washington, Seattle, WA, USA
wammar@cs.cmu.edu, gmulc@uw.edu, ytsvetko@cs.cmu.edu
{glample,cdyer}@cs.cmu.edu, nasmith@cs.washington.edu

https://github.com/mfaruqui/crosslingual-cca
http://128.2.220.95/multilingual/

Motivation



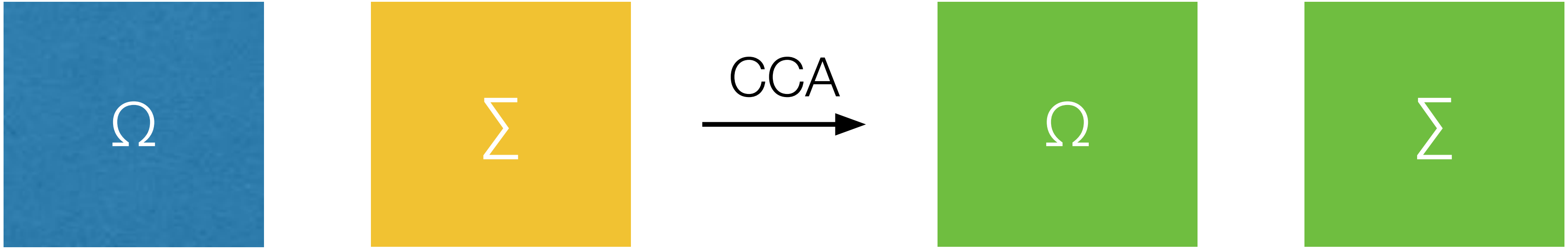
Motivation



Canonical Correlation Analysis (CCA)

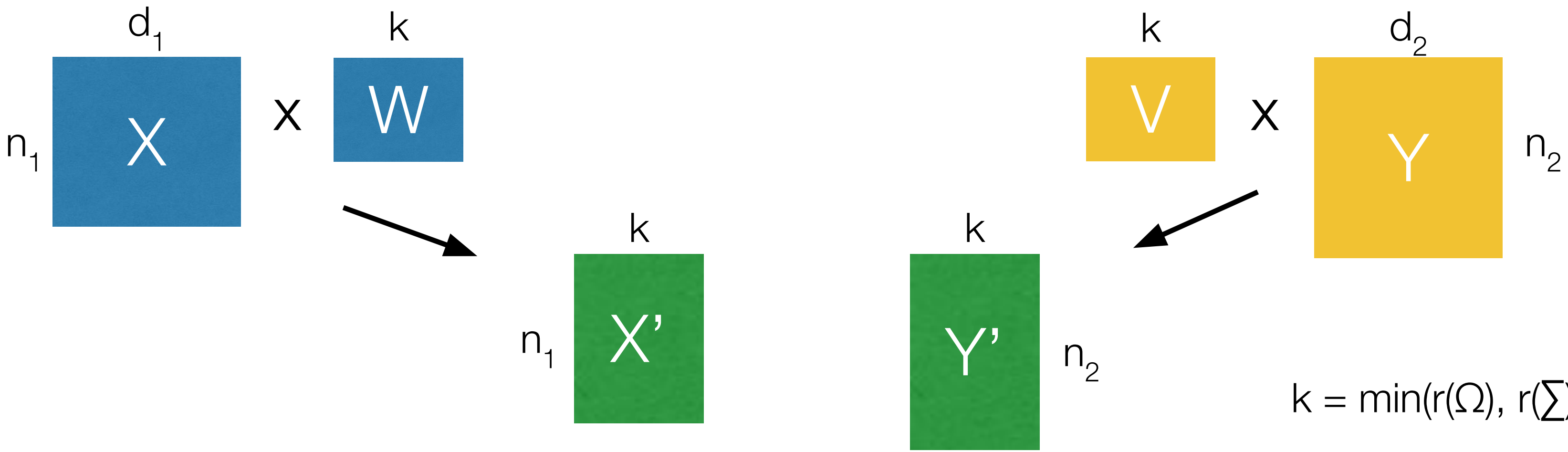
Canonical Correlation Analysis (Hotelling, 1936)

Projects two sets of vectors (of equal cardinality) in a space where they are maximally correlated.



Canonical Correlation Analysis (CCA)

$\Omega \subseteq X, \sum \subseteq Y$ $\qquad$ $W, V = CCA(\Omega, \sum)$

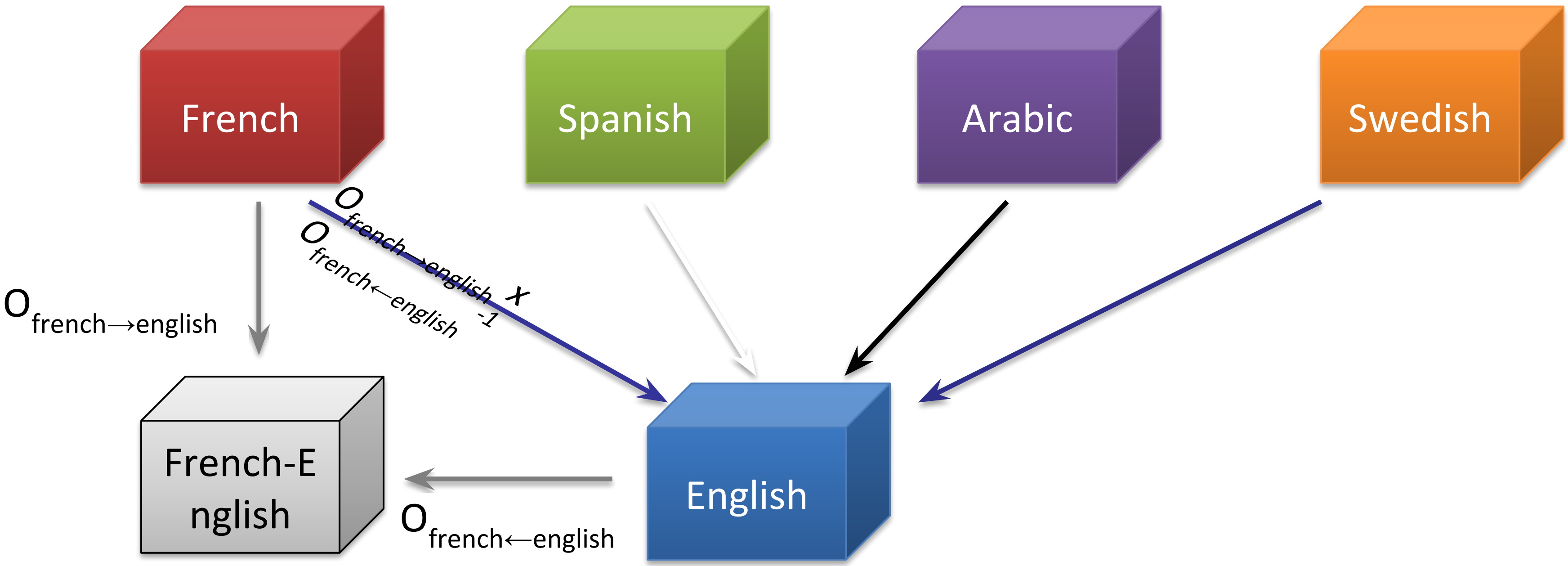


$k = \min(r(\Omega), r(\sum))$

X' and Y' are now maximally correlated.

[Faruqui & Dyer, '14]

Extension: Multilingual Embeddings



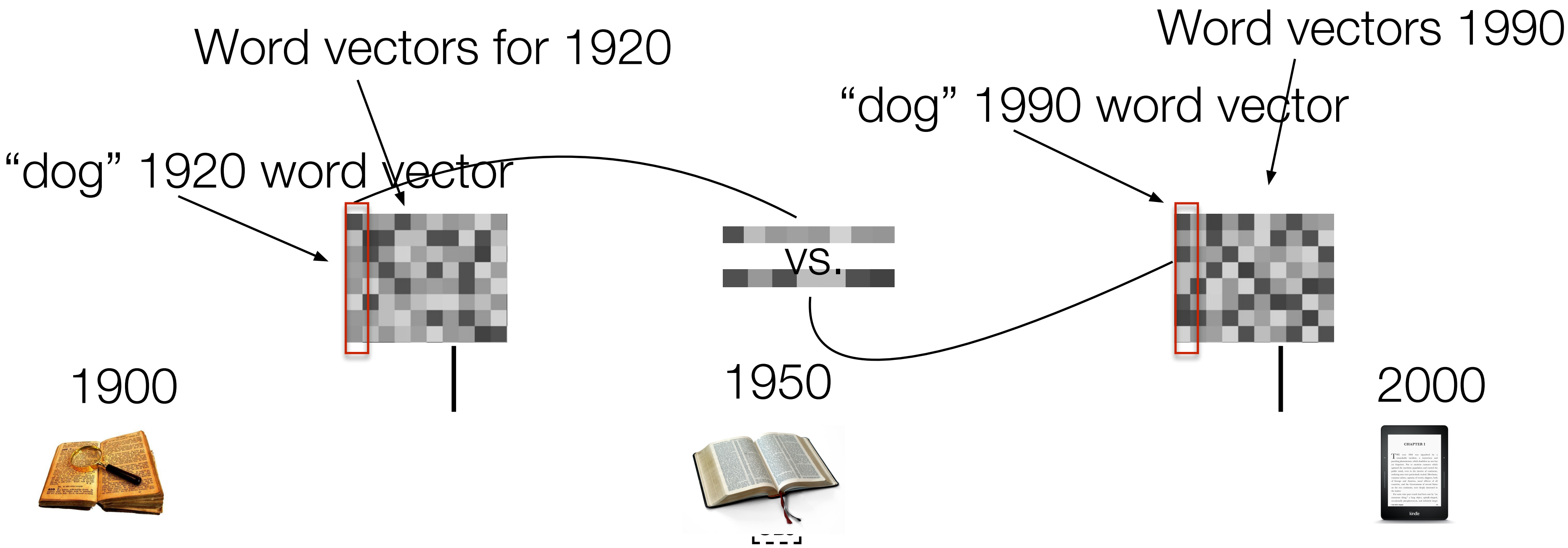
[Ammar et al., '16]

Embeddings can help study word history!

Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change

William L. Hamilton, Jure Leskovec, Dan Jurafsky
Department of Computer Science, Stanford University, Stanford CA, 94305
wleif,jure,jurafsky@stanford.edu

Diachronic Embeddings

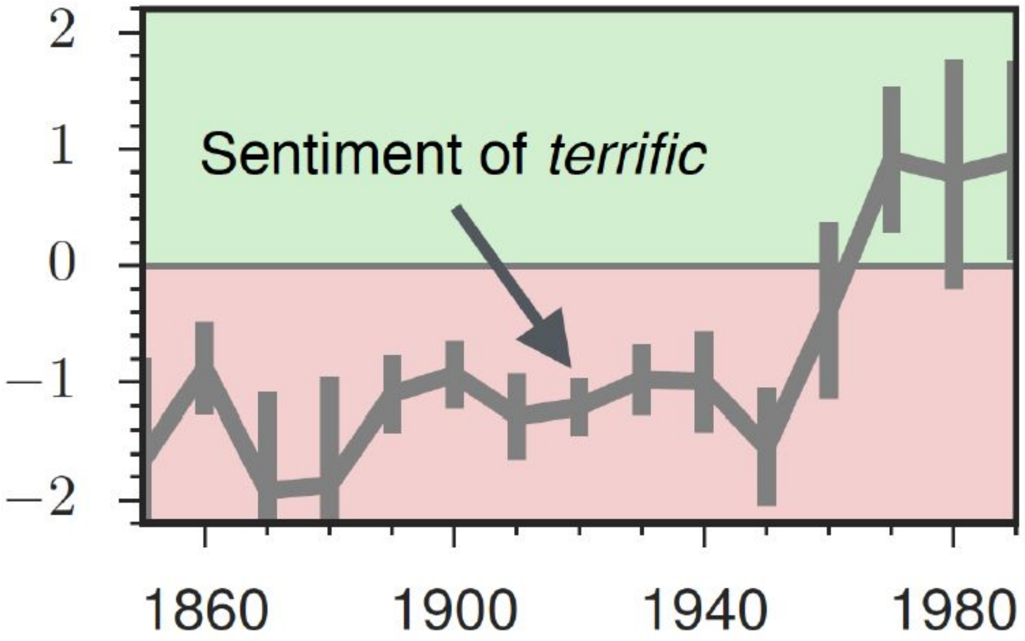


- count-based embeddings w/ PPMI
- projected to a common space

Project 300 dimensions down into 2

~30 million books, 1850-1990, Google Books data

Negative words change faster than positive words



Embeddings reflect ethnic stereotypes over time

PNAS Proceedings of the National Academy of Sciences of the United States of America

Home | Articles | Front Matter | News | Podcasts

NEW RESEARCH IN Physical Sciences | Social Sc

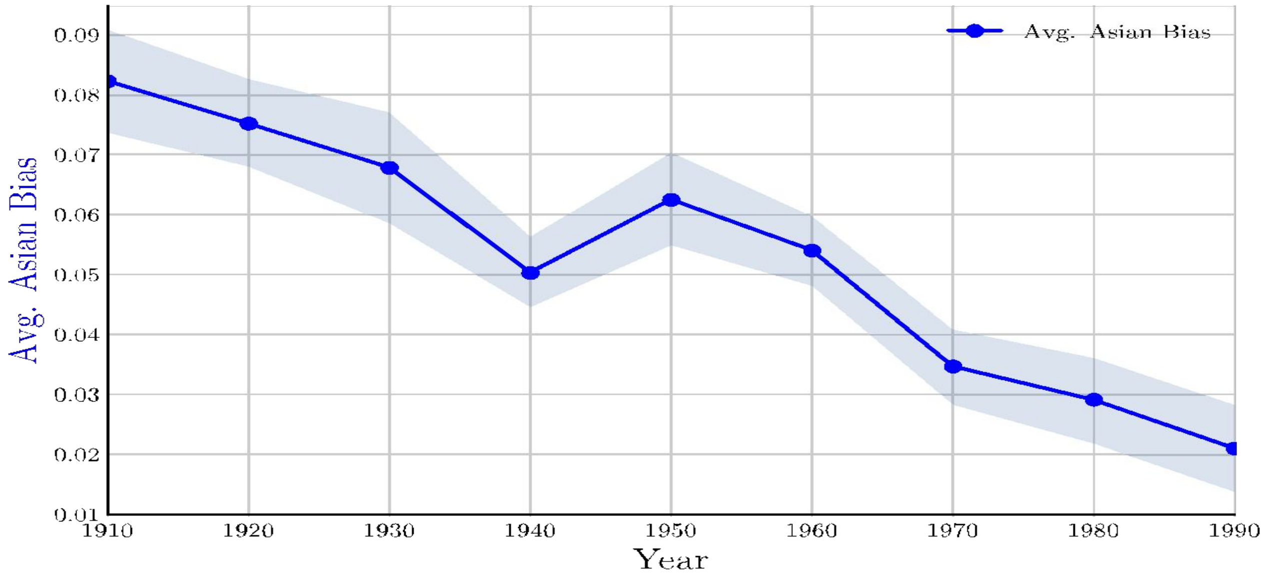
Word embeddings quantify 100 years of gender and ethnic stereotypes

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou

PNAS April 17, 2018 115 (16) E3635-E3644; published ahead of print April 3, 2018

Change in linguistic framing 1910-1990

Change in association of Chinese names with adjectives framed as "othering" (barbaric, monstrous, bizarre)



Conclusion

- Concepts or word senses
  - Have a complex many-to-many association with words (homonymy, multiple senses)
  - Have relations with each other
  - Synonymy, Antonymy, Superordinate
  - But are hard to define formally (necessary & sufficient conditions)

- Embeddings = vector models of meaning
  - More fine-grained than just a string or index
  - Especially good at modeling similarity/analogy
  - Just download them and use cosines!!
  - Useful in many NLP tasks
  - But know they encode cultural stereotypes