Computational Linguistics

4A

CSC 485/2501
Fall 2025

4a. Vector-based Semantics

Gerald Penn
Department of Computer Science, University of Toronto

(slides borrowed from Chris Manning)

From symbolic to distributed representations

The vast majority of rule-based **and** statistical NLP work regarded words as atomic symbols: hotel, conference, walk

In vector space terms, this is a vector with one 1 and a lot of zeroes

$$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$$

We call this a "one-hot" representation.

Its problem:

$$\text{motel}\ [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$$
$$\text{hotel}\ [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0] = 0$$

Distributional similarity based representations

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

“Noscitur a sociis”

- 19th-century rule of interpretation in English civil courts

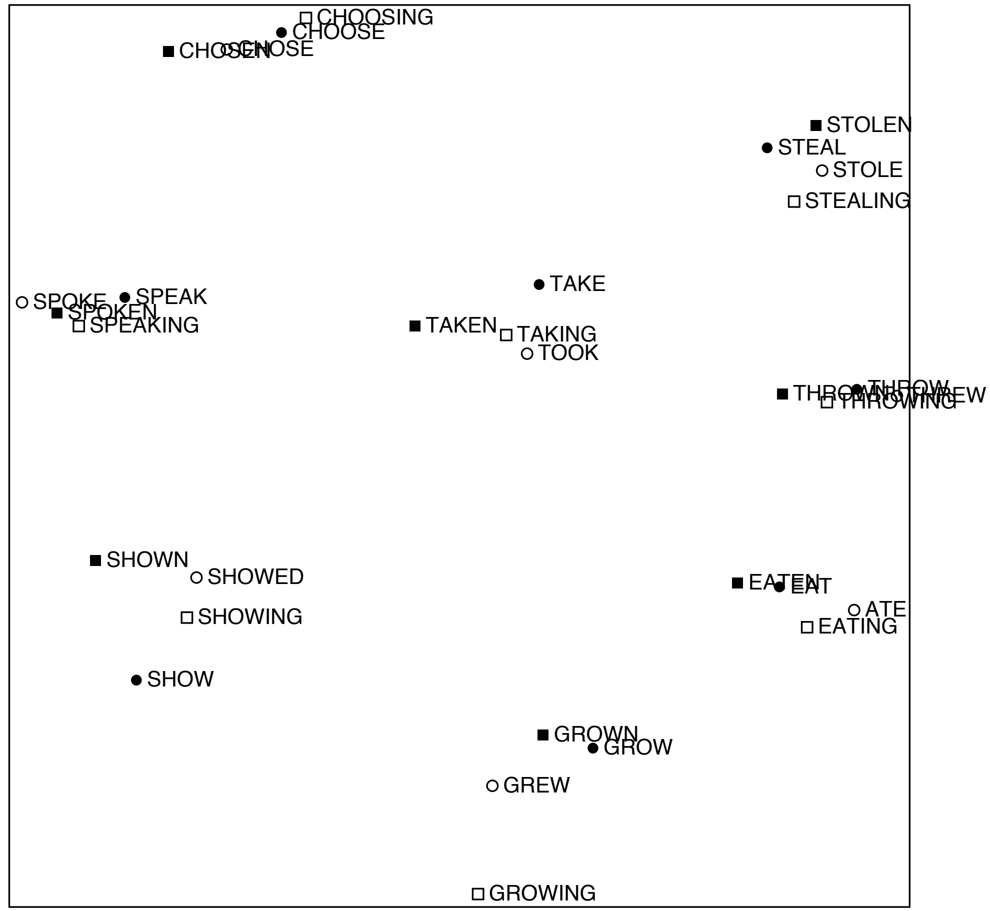
One of the most successful ideas of modern NLP

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

↖ These words will represent *banking* ↗

With distributed, distributional representations, syntactic and semantic patterning is captured

$$shown = \begin{array}{r} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{array}$$



Synonymy? Hyponymy?
Morphology?

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

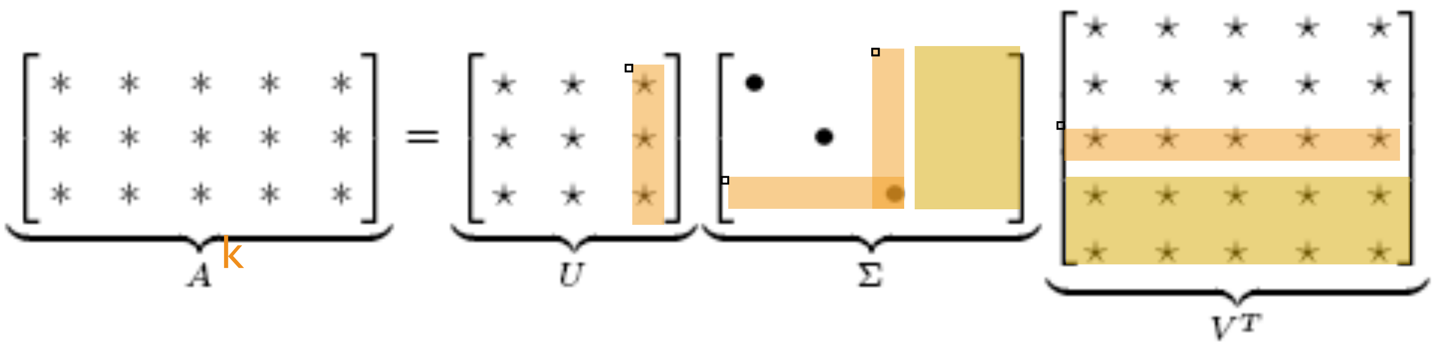
Menu

1. Vector space representations of language
2. Predict! vs. Count!: The GloVe model of word vectors
3. Wanted: meaning composition functions
4. Tree-structured Recursive Neural Networks for Semantics
5. Natural Language Inference with TreeRNNs

LSA vs. word2vec

LSA: **Count!**

- Factorize a (maybe weighted, maybe log scaled) term-document or word-context matrix (Schütze 1992) into $U\Sigma V^T$
- Retain only *k* singular values, in order to generalize



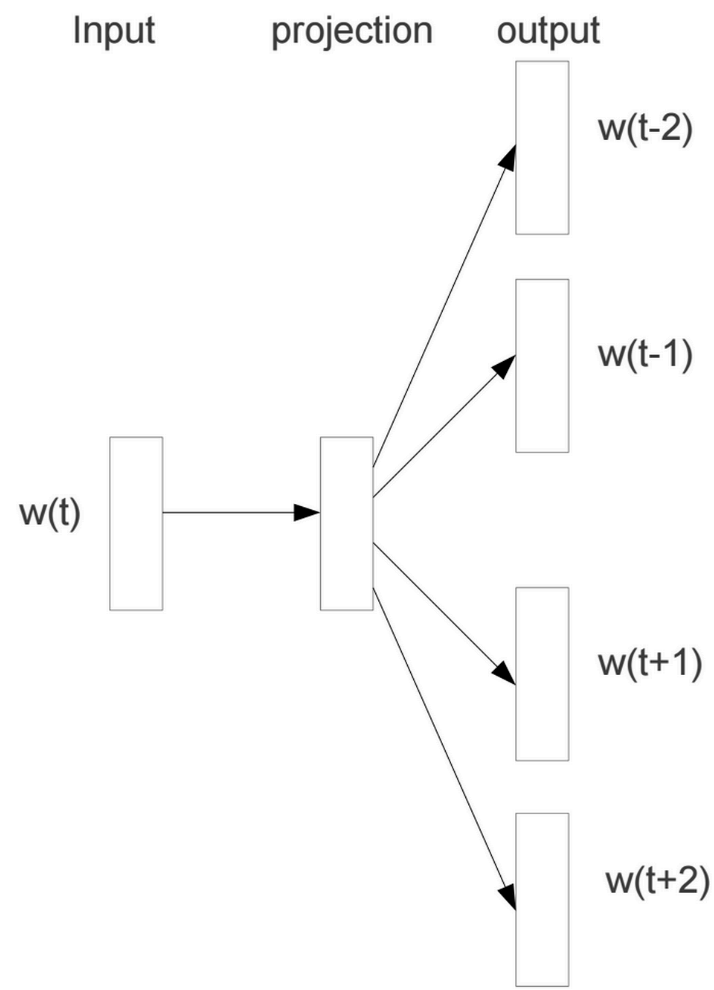
[Cf. Baroni: Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. ACL 2014]

LSA vs. word2vec

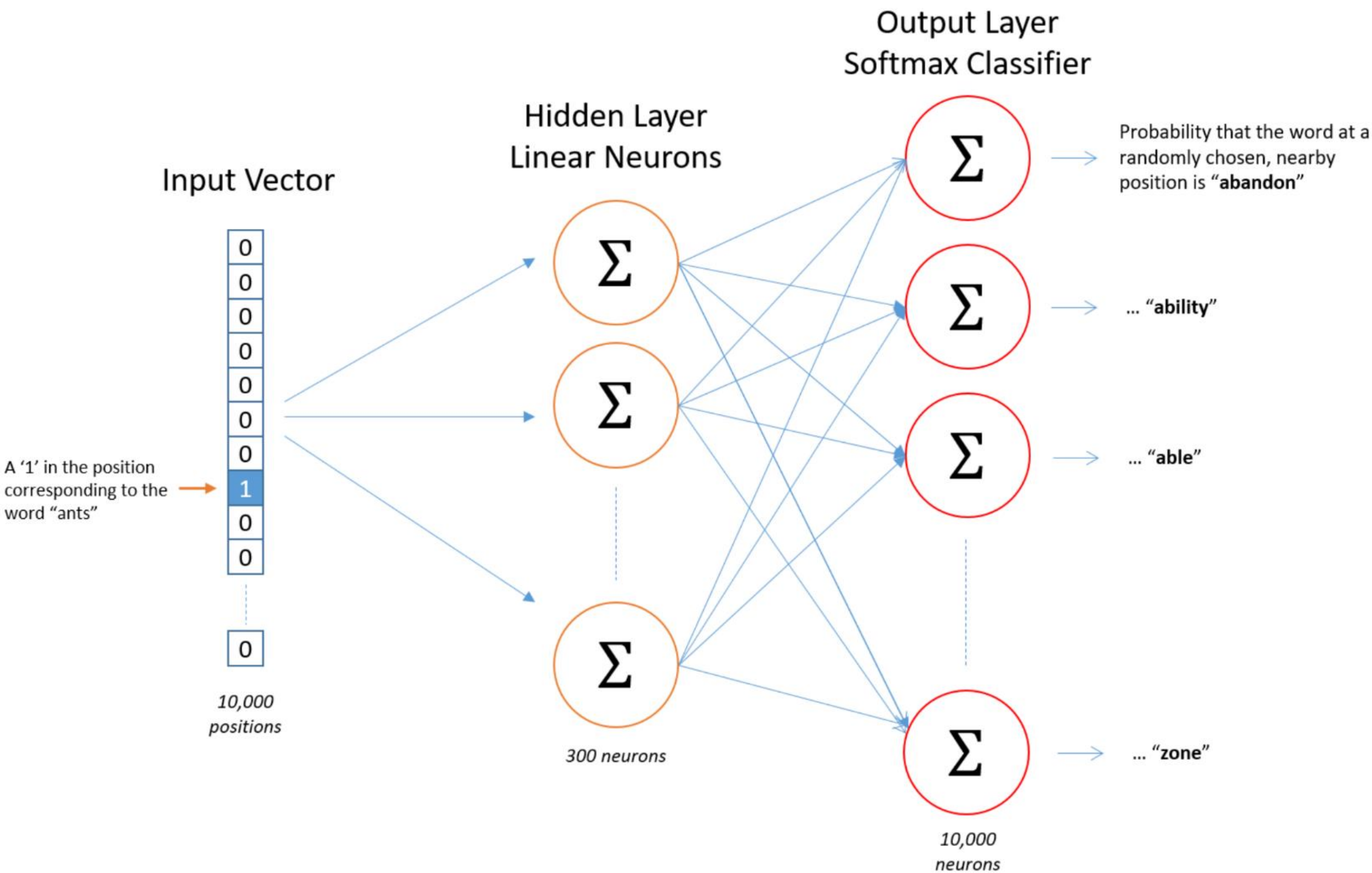
LSA: **Count!** vs.

word2vec CBOW/SkipGram: **Predict!**

- Train word vectors to try to either:
  - Predict a word given its bag-of-words context (CBOW); or
  - Predict a context word (position-independent) from the center word
- Update word vectors until they can do this prediction well



word2vec training regimen



Word Analogies: word2vec captures *dimensions of similarity* as linear relations

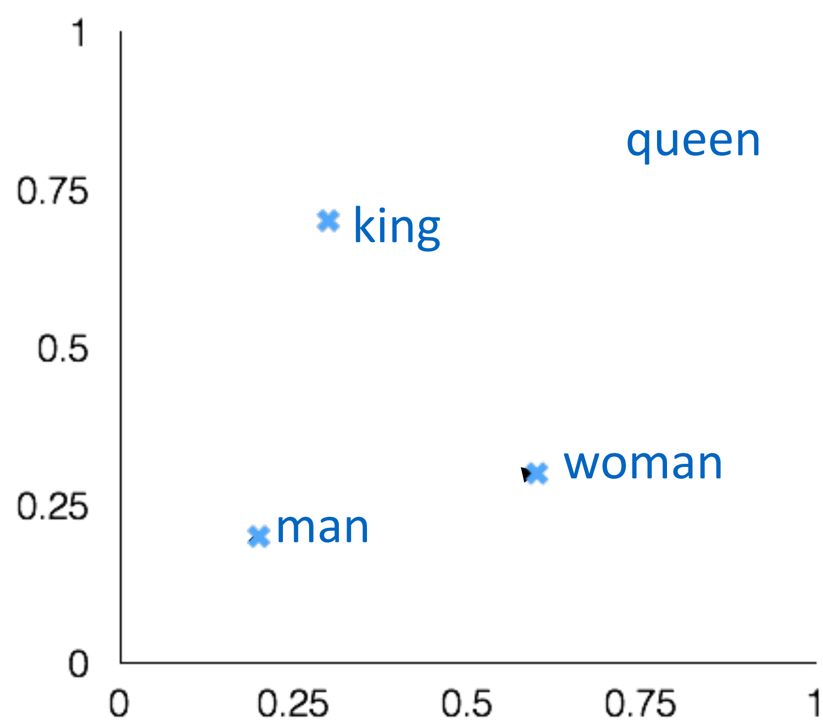
Test for linear relationships, examined by Mikolov et al. 2013

a:b :: c:? $\longrightarrow$

$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

+	king	[0.30 0.70]
–	man	[0.20 0.20]
+	woman	[0.60 0.30]
	queen	[0.70 0.80]

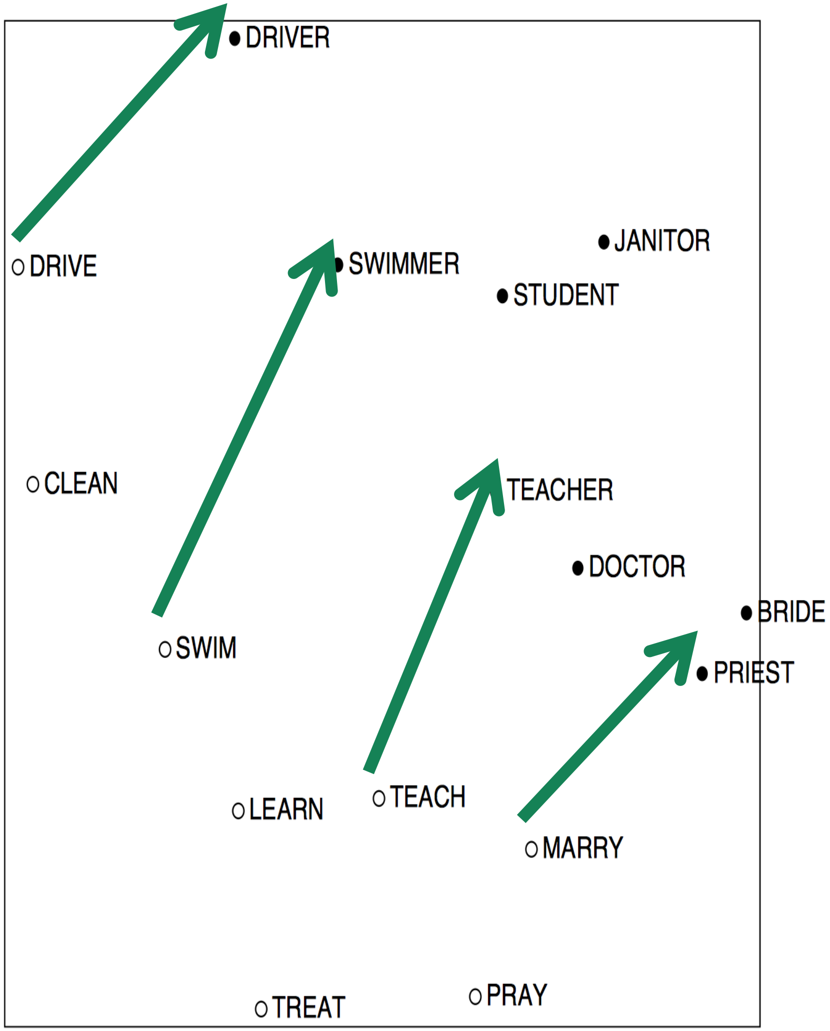


Word Analogies
[Mikolov et al., 2012, 2013]

Task: predict the last column

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

COALS model (count-modified LSA)
[Rohde, Gonnerman & Plaut, ms., 2005]



Analogy	Reported	Index	1st answer	2nd answer
Mikolov et al. (2013a)				
man king woman	queen	2	king	queen
Paris France Tokyo	Japan	1	Japan	Tokyo
brother sister grandson	granddaughter	1	granddaughter	niece
big bigger cold	colder	2	cold	colder
Einstein scientist Picasso	painter	1	painter	scientist
Bolukbasi et al. (2016)				
man computer_programmer woman	homemaker	2	computer_programmer	homemaker
he doctor she	nurse	2	doctor	nurse
she interior_designer he	architect	2	interior_designer	architect
she feminism he	conservatism	4	feminism	liberalism
she lovely he	brilliant	10	lovely	magnificent
she sewing he	carpentry	4	sewing	woodworking
Manzini et al. (2019b)				
black criminal caucasian	lawful	13	legal	statutory
caucasian lawful black	criminal	2	lawful	criminal
caucasian hillbilly asian	yuppie	3	hillbilly	hippy
asian yuppie caucasian	hillbilly	2	yuppie	hillbilly
asian engineer black	killer	39	operator	jockey
black killer asian	engineer	7	killer	impostor
christian conservative jew	liberal	4	centrist	democrat
jew liberal christian	conservative	2	liberal	conservative
muslim terrorist jew	journalist	4	hacker	protestor
jew journalist muslim	terrorist	2	purportedly	terrorist
christian conservative muslim	regressive	53	moderate	conservative
muslim regressive christian	conservative	13	regressive	progressive

From Nissim et al. (2019) arXiv:1905.09866v1 [cs.CL]

Word analogy task [Mikolov, Yih & Zweig 2013a]

Model	Dimensions	Corpus size	Performance (Syn + Sem)
CBOW (Mikolov et al. 2013b)	300	1.6 billion	36.1

Count based vs. direct prediction

LSA, HAL (Lund & Burgess),
COALS (Rohde et al),
Hellinger-PCA (Lebret & Collobert)

- Fast training
- Efficient usage of statistics
- Primarily used to capture word similarity
- Disproportionate importance given to small counts

- NNLM, HLBL, RNN, word2vec Skip-gram/CBOW, (Bengio et al; Collobert & Weston; Huang et al; Mnih & Hinton; Mikolov et al; Mnih & Kavukcuoglu)

- Scales with corpus size
- Inefficient usage of statistics
- Generate improved performance on other tasks
- Can capture complex patterns beyond word similarity

Encoding meaning in vector differences

[Pennington, Socher, and Manning, EMNLP 2014]

Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	x = solid	x = gas	x = water	x = random
$P(x \mid \text{ice})$	large	small	large	small
$P(x \mid \text{steam})$	small	large	large	small
$\frac{P(x \mid \text{ice})}{P(x \mid \text{steam})}$	large	small	~1	~1

Encoding meaning in vector differences

[Pennington, Socher, and Manning, EMNLP 2014]

Crucial insight: Ratios of co-occurrence probabilities can encode meaning components

	x = solid	x = gas	x = water	x = fashion
$P(x \mid \text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(x \mid \text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(x \mid \text{ice})}{P(x \mid \text{steam})}$	8.9	8.5×10^{-2}	1.36	0.96

Encoding meaning in vector differences

Q: How can we capture ratios of co-occurrence probabilities as meaning components in a word vector space?

A: Log-bilinear model: $$w_i \cdot w_j = \log P(i|j)$$

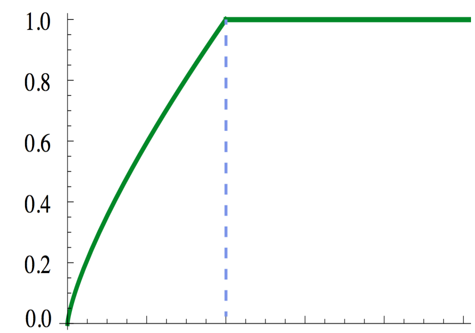
with vector differences $$w_x \cdot (w_a - w_b) = \log \frac{P(x|a)}{P(x|b)}$$

GloVe: A new model for learning word representations [Pennington et al., EMNLP 2014]

$$w_i \cdot w_j = \log P(i|j)$$

$$w_x \cdot (w_a - w_b) = \log \frac{P(x|a)}{P(x|b)}$$

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \qquad f \sim$$



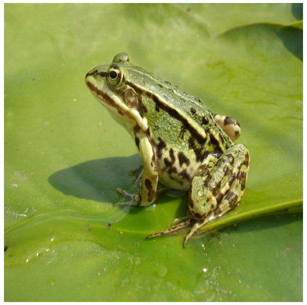
Word similarities

Nearest words to frog:

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

litoria

leptodactylidae

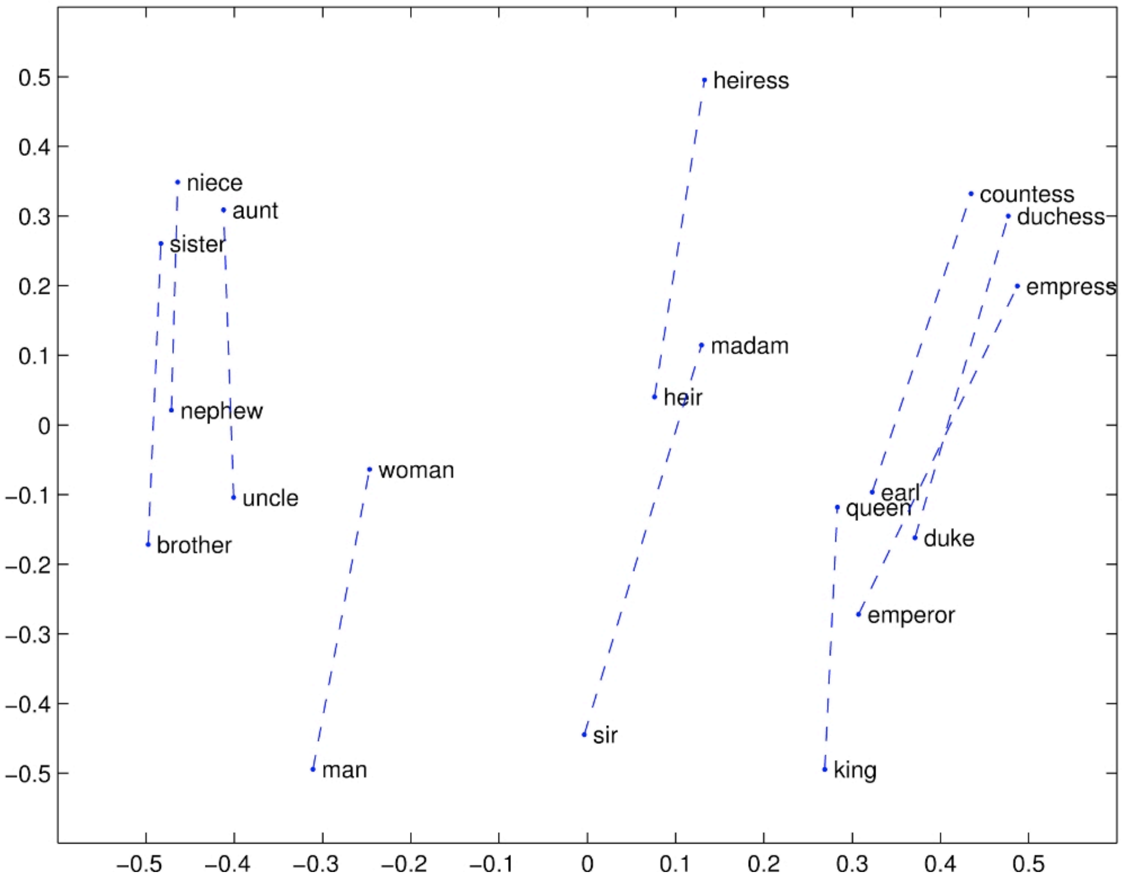


rana

eleutherodactylus

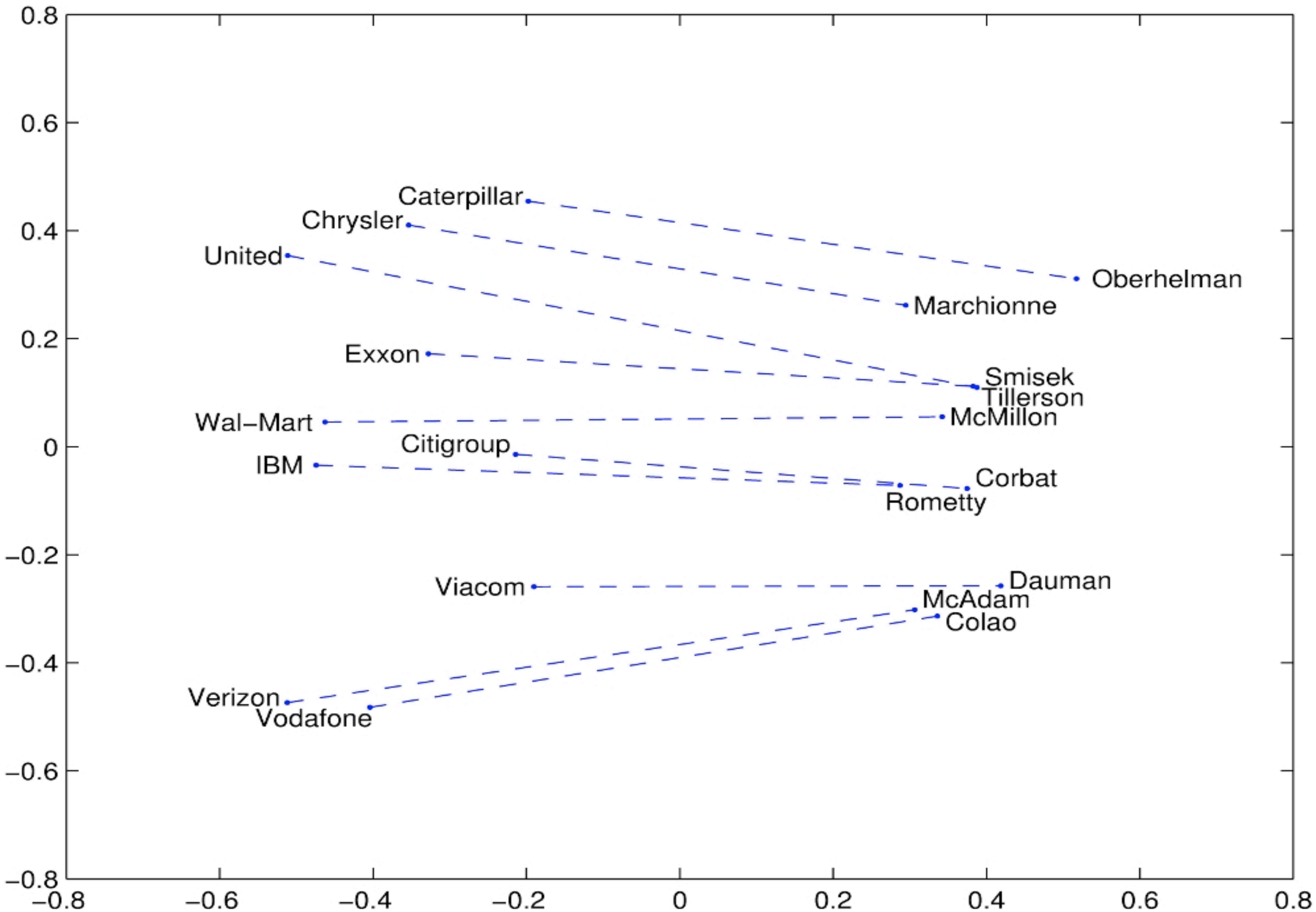
http://nlp.stanford.edu/projects/glove/

Glove Visualizations

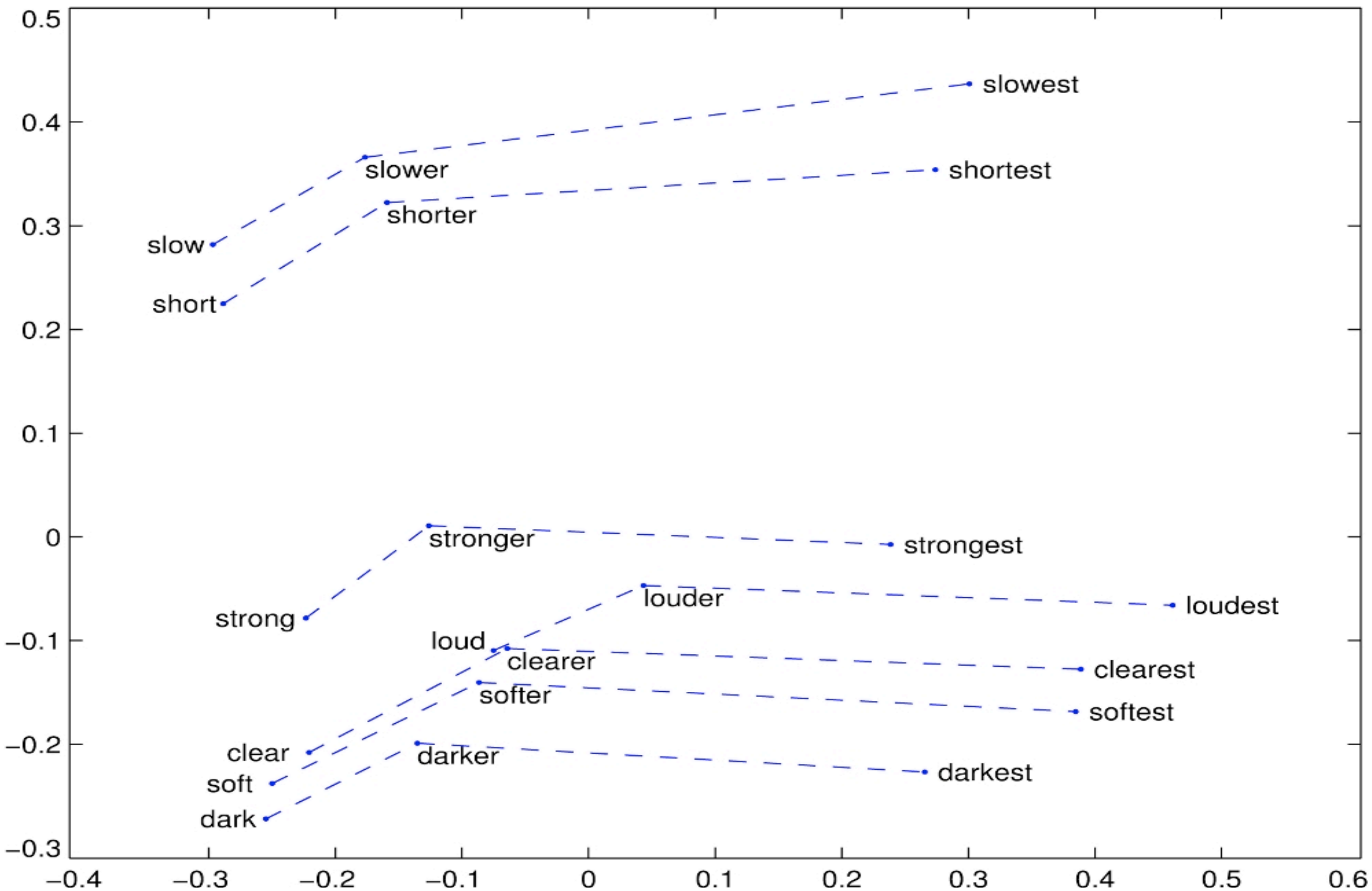


http://nlp.stanford.edu/projects/glove/

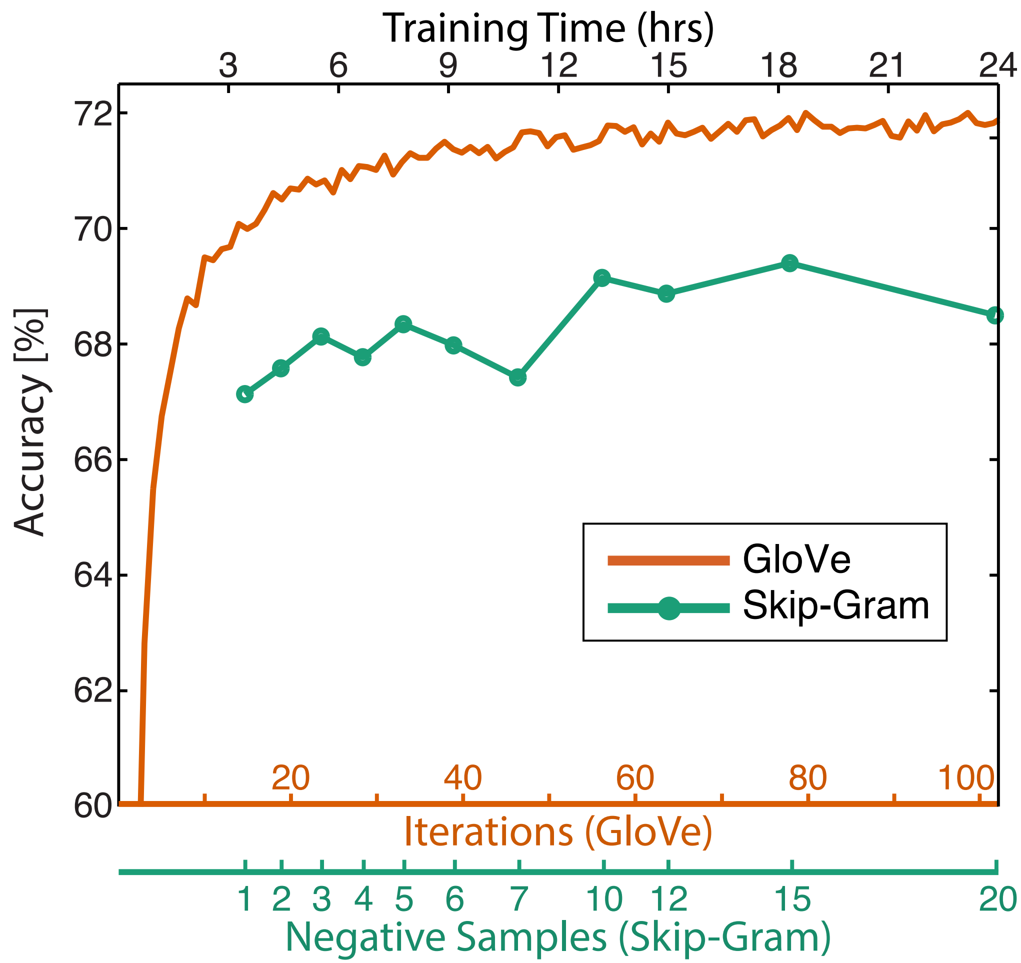
Glove Visualizations: Company - CEO



Glove Visualizations: Superlatives

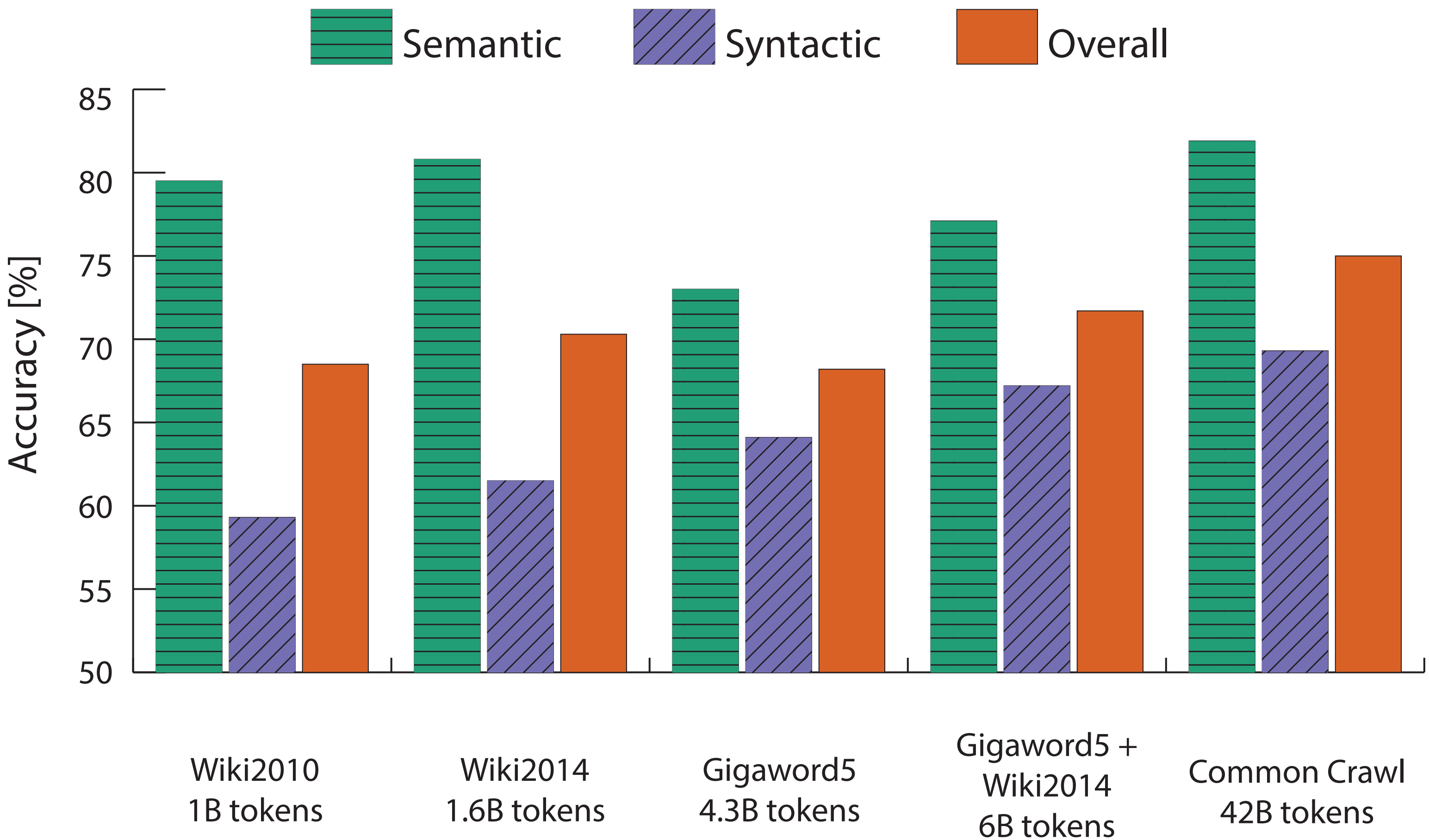


Analogy evaluation and hyperparameters



NB! ➔

Analogy evaluation and hyperparameters



Word Embeddings Conclusion

Developed a model that can translate meaningful relationships between word-word co-occurrence probabilities into linear relations in the word vector space

GloVe shows the connection between **Count!** work and **Predict!** work – appropriate scaling of counts gives the properties and performance of **Predict!** models

Can one **explain** word2vec's linear structure?
See Arora, Li, Liang, Ma, & Risteski. 2015. Random Walks on Context Spaces: Towards an Explanation of the Mysteries of Semantic Word Embeddings. [Develops a generative model.]

Compositionality

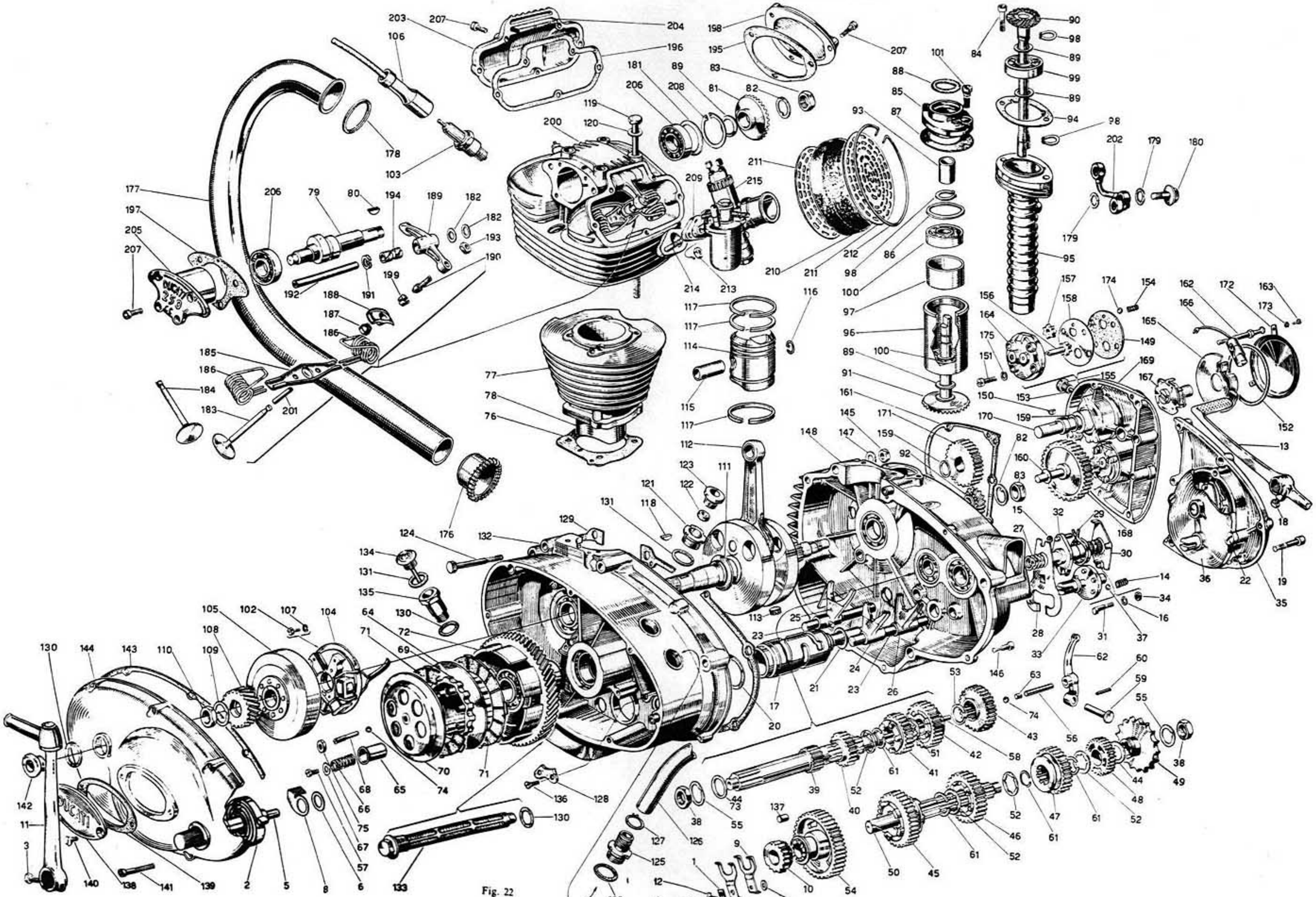


Fig. 22

Artificial Intelligence requires understanding bigger things from knowing about smaller things

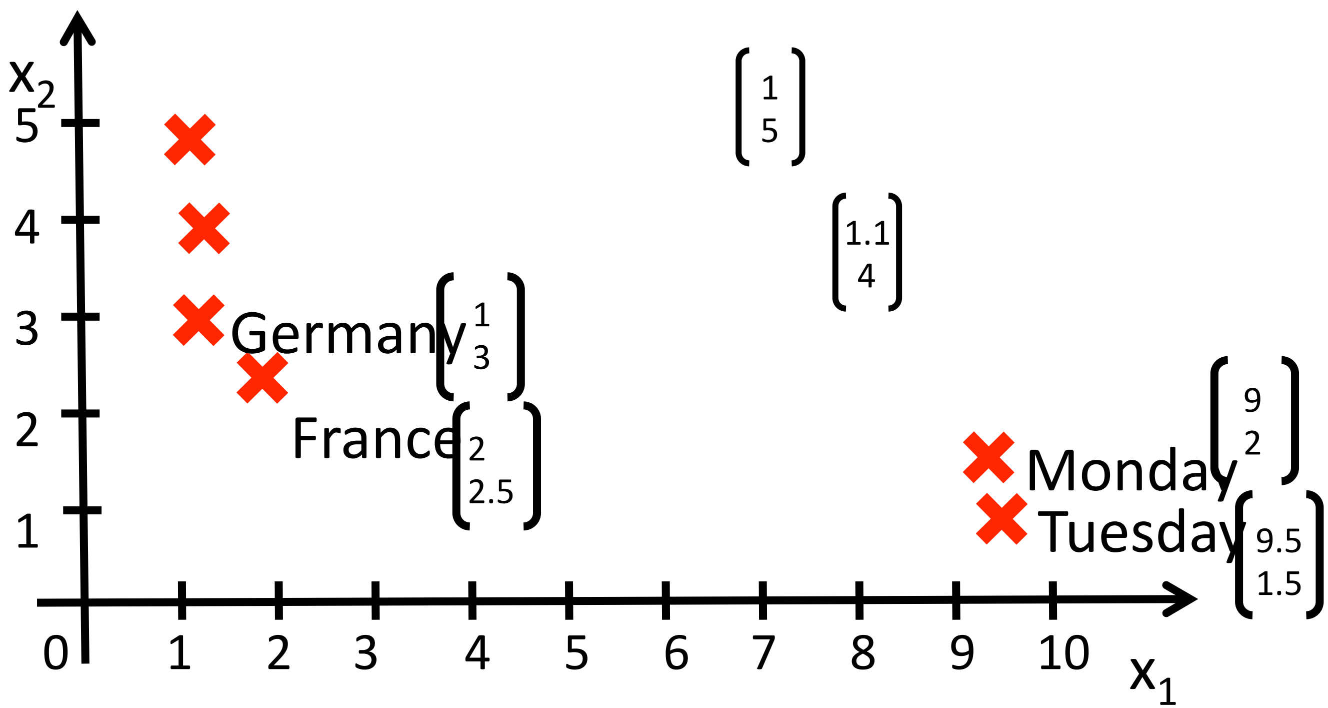
WE need more! What of larger semantic units?

How can we know when larger units are similar in meaning?

- *Two senators received contributions engineered by lobbyist Jack Abramoff in return for political favors.*
- *Jack Abramoff attempted to bribe two legislators.*

People interpret the meaning of larger text units – entities, descriptive terms, facts, arguments, stories – by **semantic composition** of smaller elements

Representing Phrases as Vectors



Vector for single words are useful as features but limited!

the country of my birth

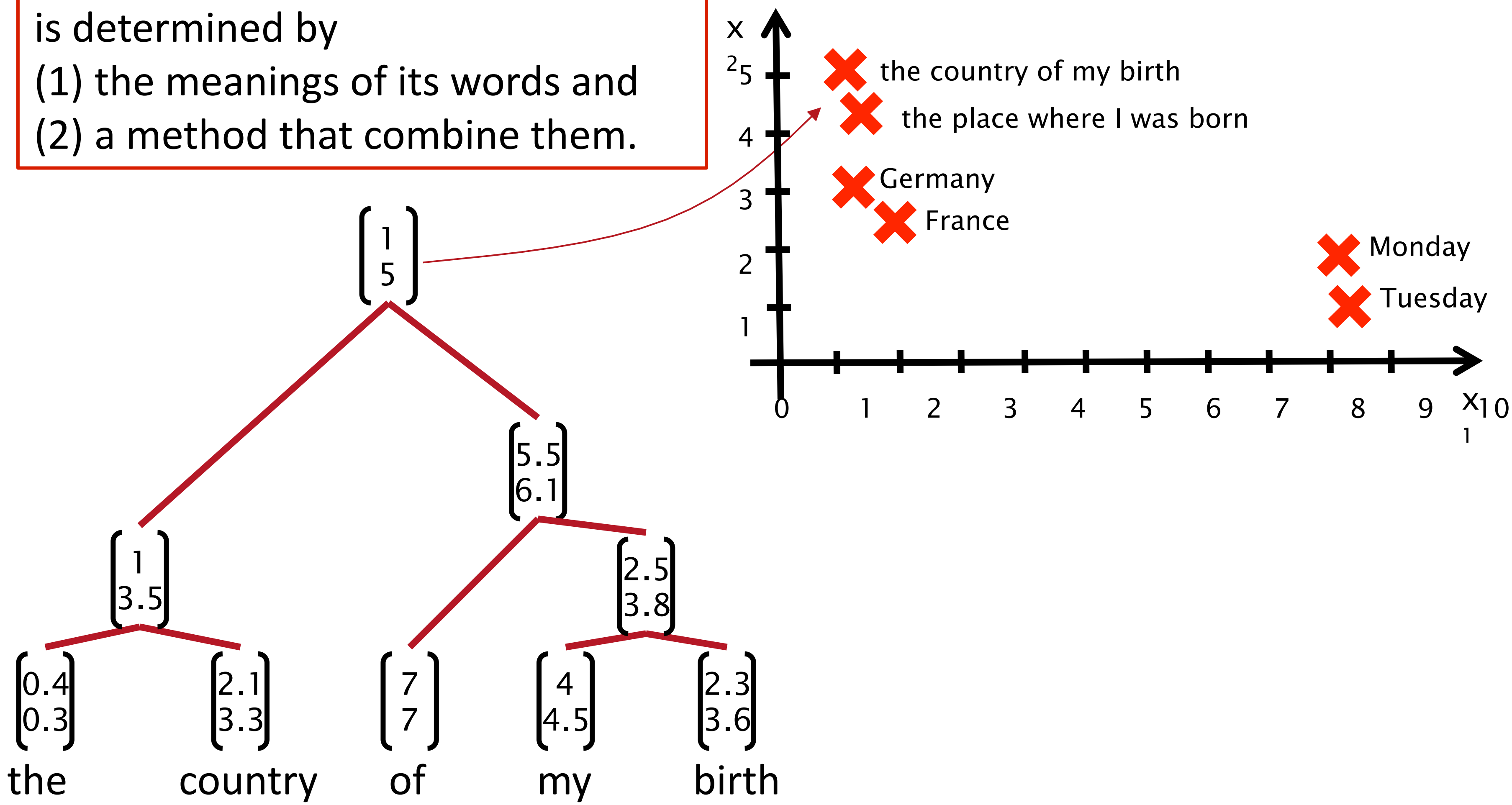
the place where I was born

Can we extend the ideas of word vector spaces to phrases?

How should we map phrases into a vector space?

Use the principle of compositionality!

The meaning (vector) of a sentence is determined by
(1) the meanings of its words and
(2) a method that combine them.

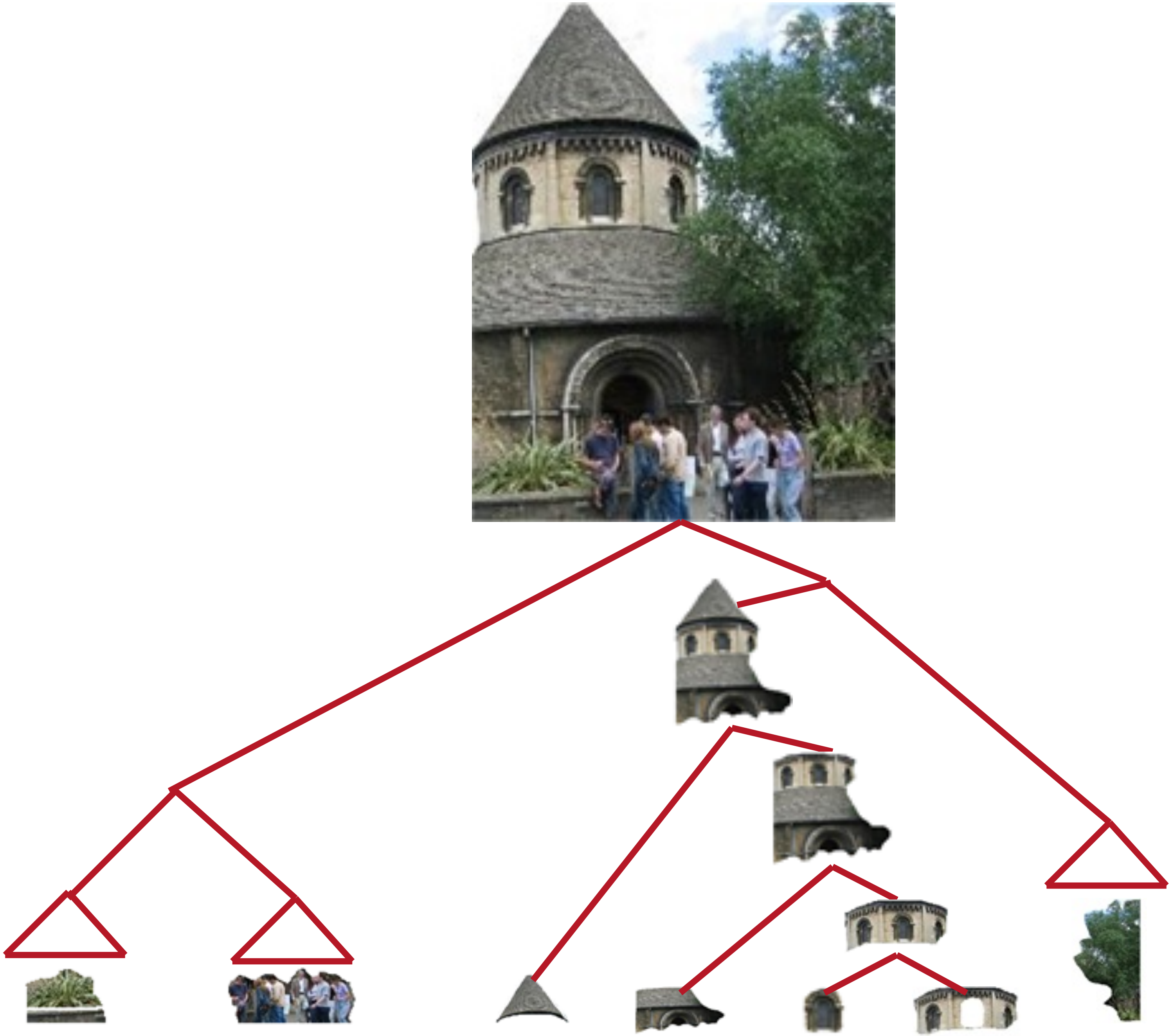


$$e^{-} \quad (x-\mu)^2 \quad 2\sigma^2$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$$\frac{1}{\sqrt{2\pi\sigma}}$$

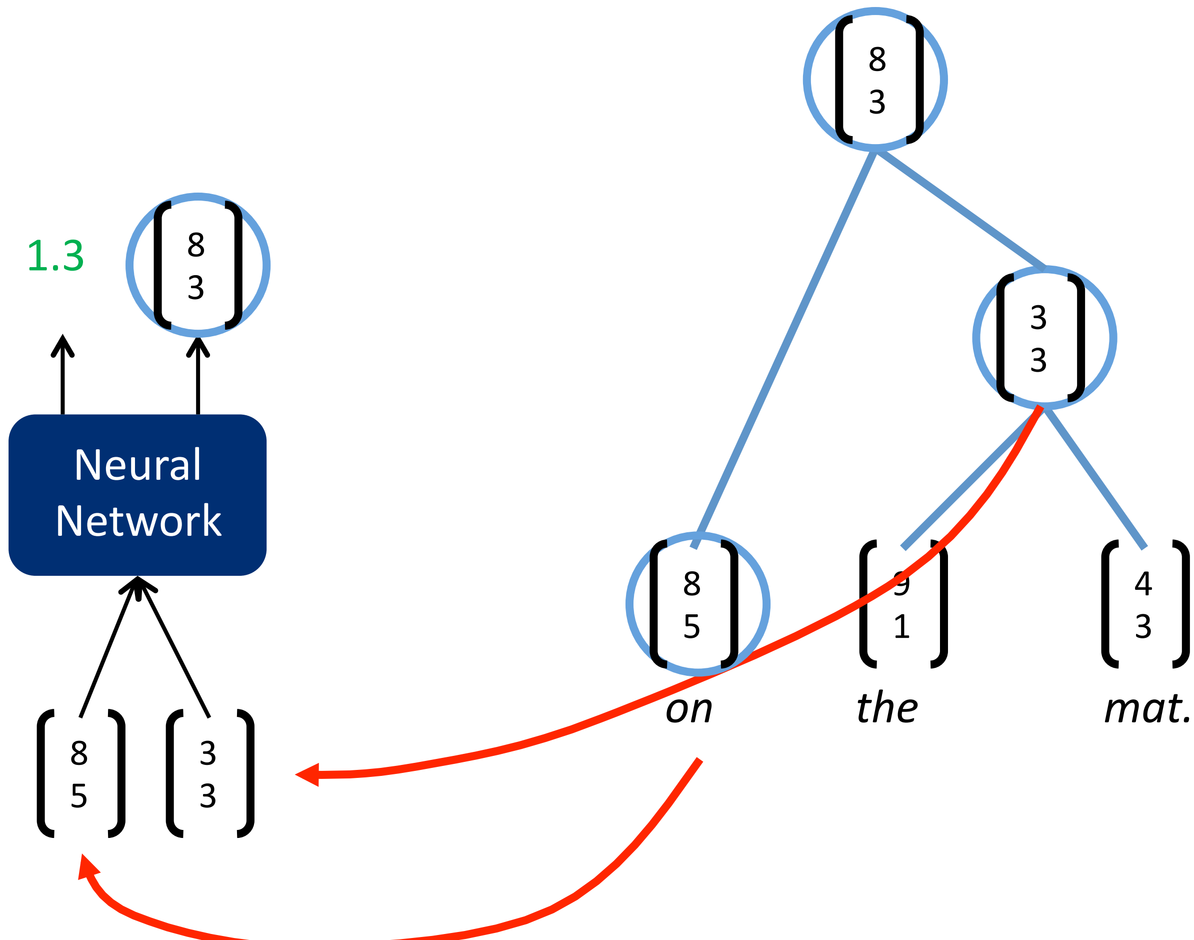
$$e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



Tree Recursive Neural Networks (Tree RNNs)

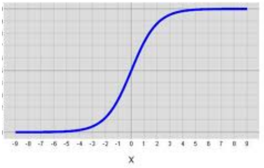
Basic computational unit:
Recursive Neural Network

(Goller & Küchler 1996, Costa et al. 2003, Socher et al. ICML, 2011)

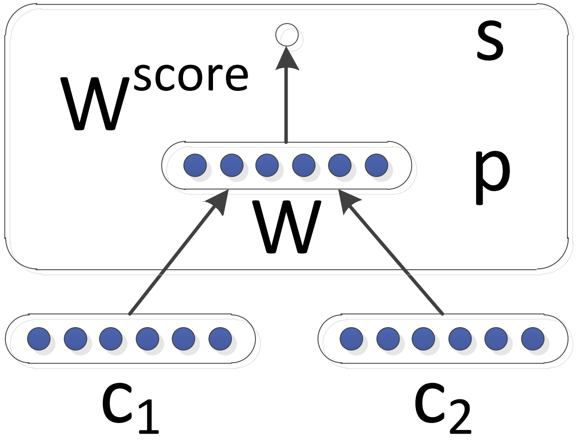


Version 1: Simple concatenation Tree RNN

$$p = \tanh\left(W \begin{matrix} c_1 \\ c_2 \end{matrix} + b\right),$$

where tanh: 

$$\text{score} = V^T p$$



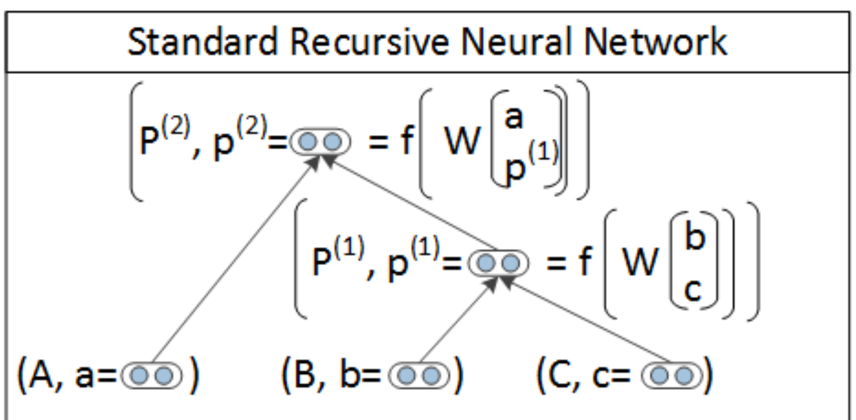
Only a single weight matrix = composition function!

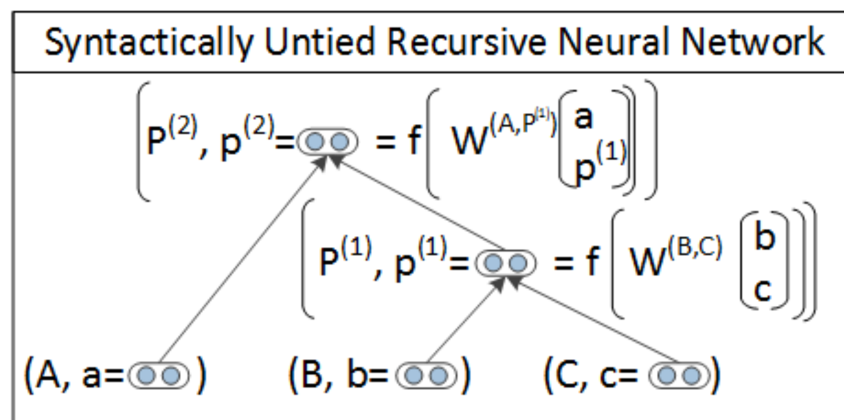
No real interaction between the input words!

Not adequate for human language composition function

Version 2: PCFG + Syntactically-Untied RNN

- A symbolic Context-Free Grammar (CFG) backbone is adequate for basic syntactic structure
- We use the discrete syntactic categories of the children to choose the composition matrix
- An RNN can do better with a different composition matrix for different syntactic environments
- The result gives us a better semantics

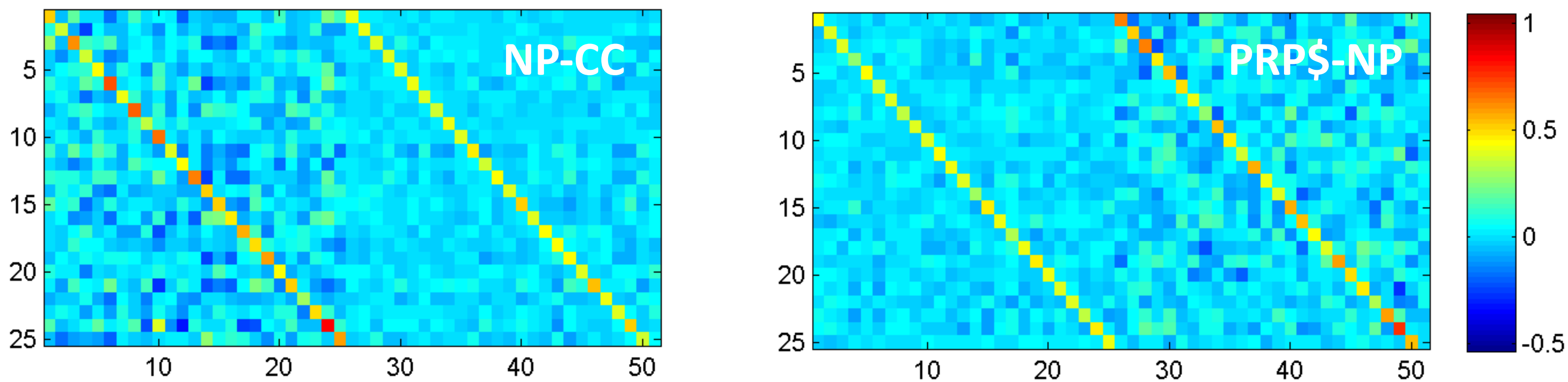




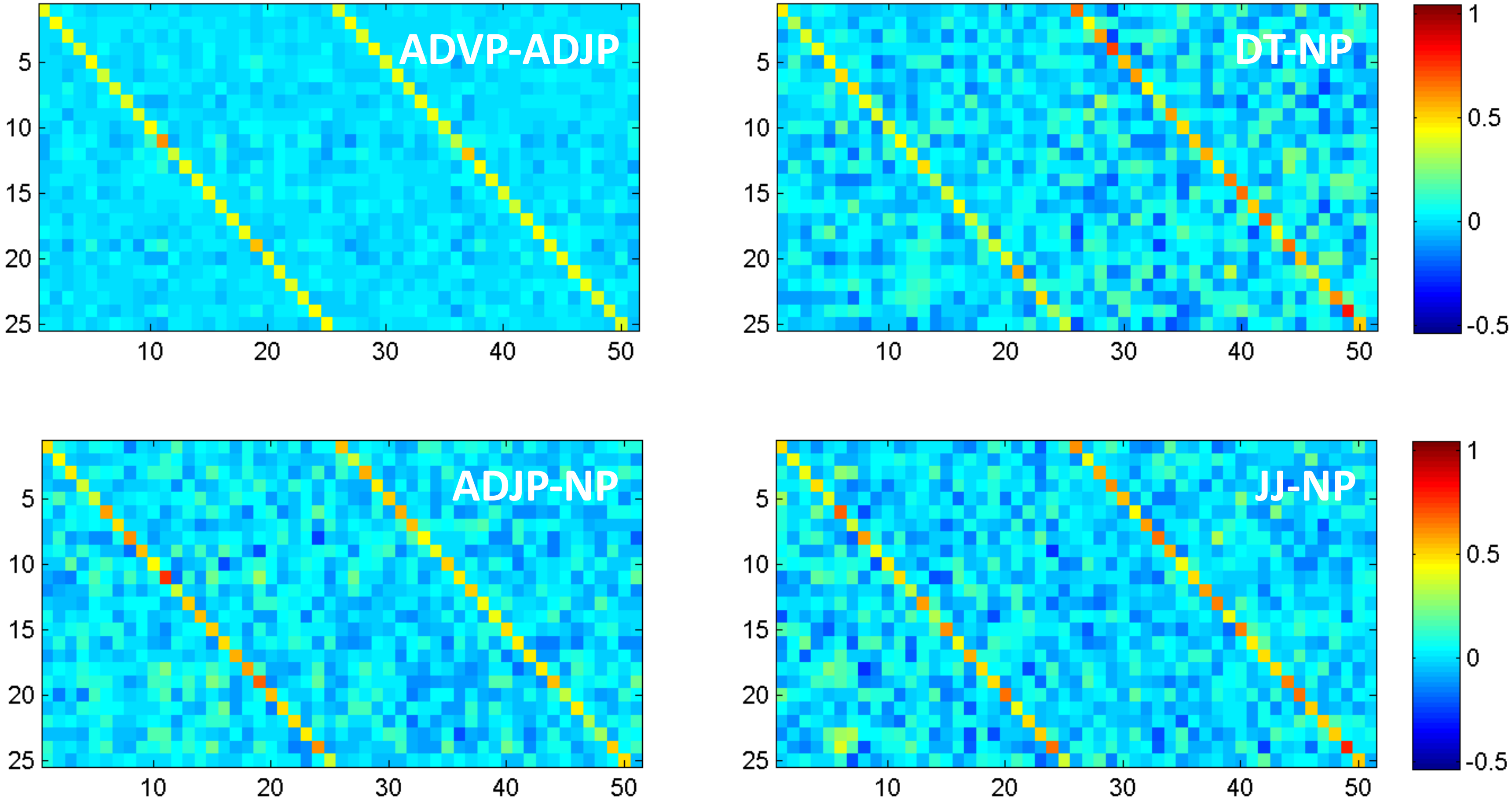
SU-RNN

Learns soft notion of head words

Initialization: $W^{(\cdot\cdot)} = 0.5[I_{n\times n} I_{n\times n} 0_{n\times 1}] + \epsilon$



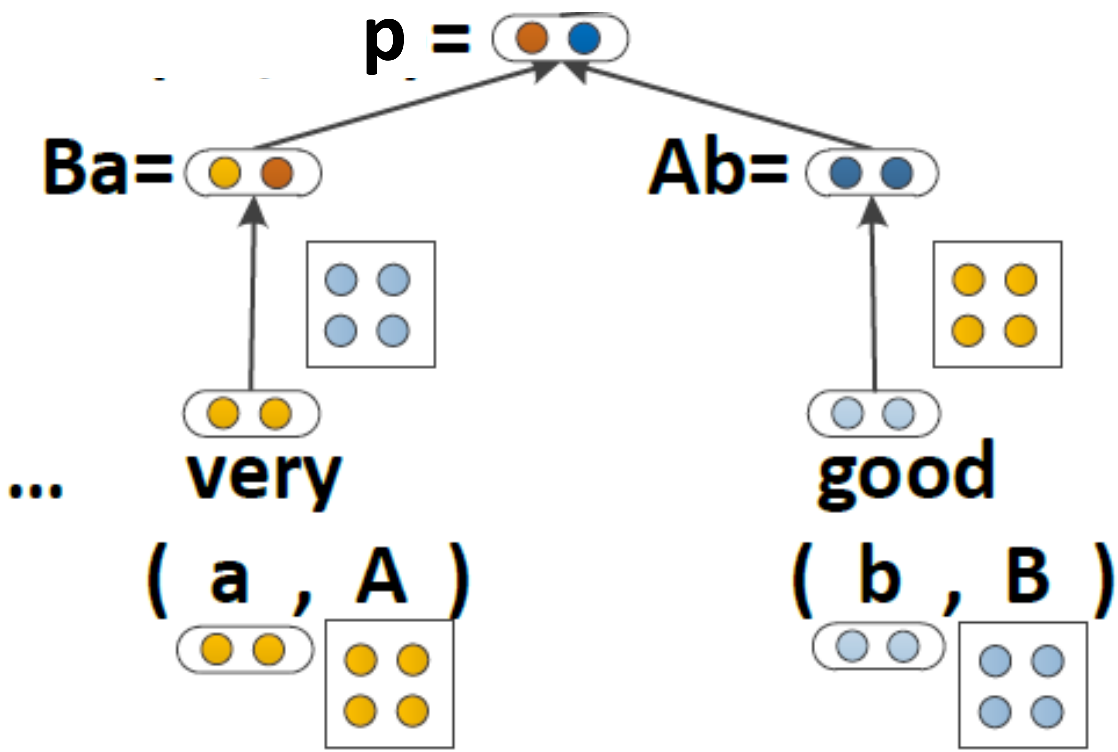
SU-RNN



Version 3: Matrix-vector RNNs

[Socher, Huval, Bhat, Manning, & Ng, 2012]

$$p = f\left(W\begin{bmatrix} a \\ b \end{bmatrix}\right) \qquad p = f\left(W\begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$

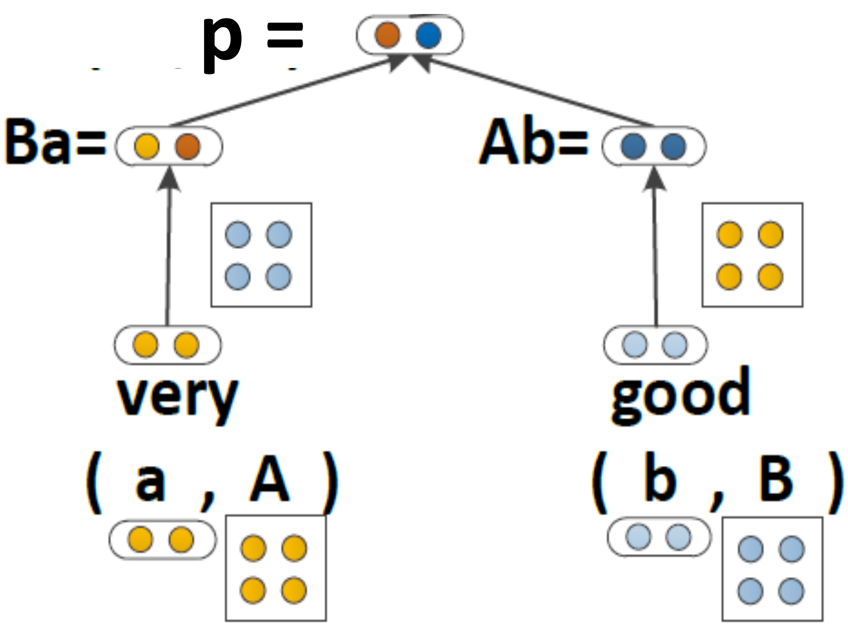


Version 3: Matrix-vector RNNs

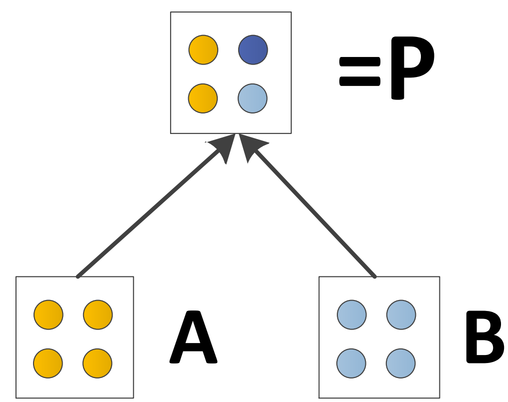
[Socher, Huval, Bhat, Manning, & Ng, 2012]

$$p = f\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$

$$P = g(A, B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$$

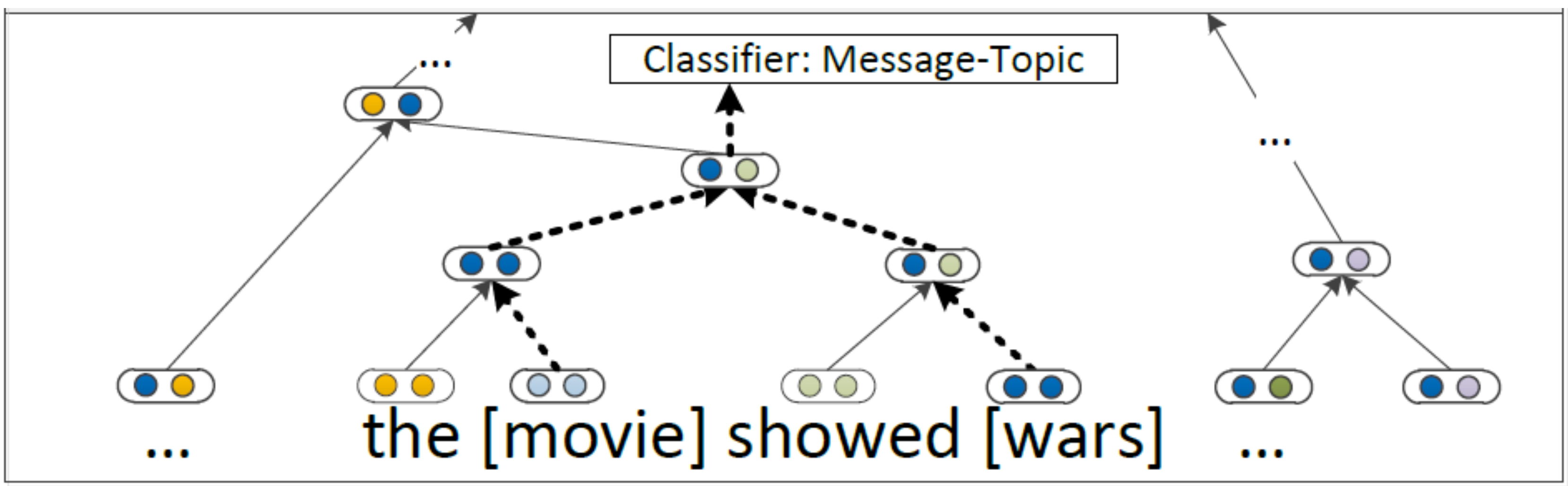


$$W_M \in \mathbb{R}^{n \times 2n}$$



Classification of Semantic Relationships

- Can an MV-RNN learn how a large syntactic context conveys a semantic relationship?
- My [apartment]$_{e1}$ has a pretty large [kitchen] $_{e2}$
  → component-whole relationship (e2,e1)
- Build a single compositional semantics for the minimal constituent including both terms

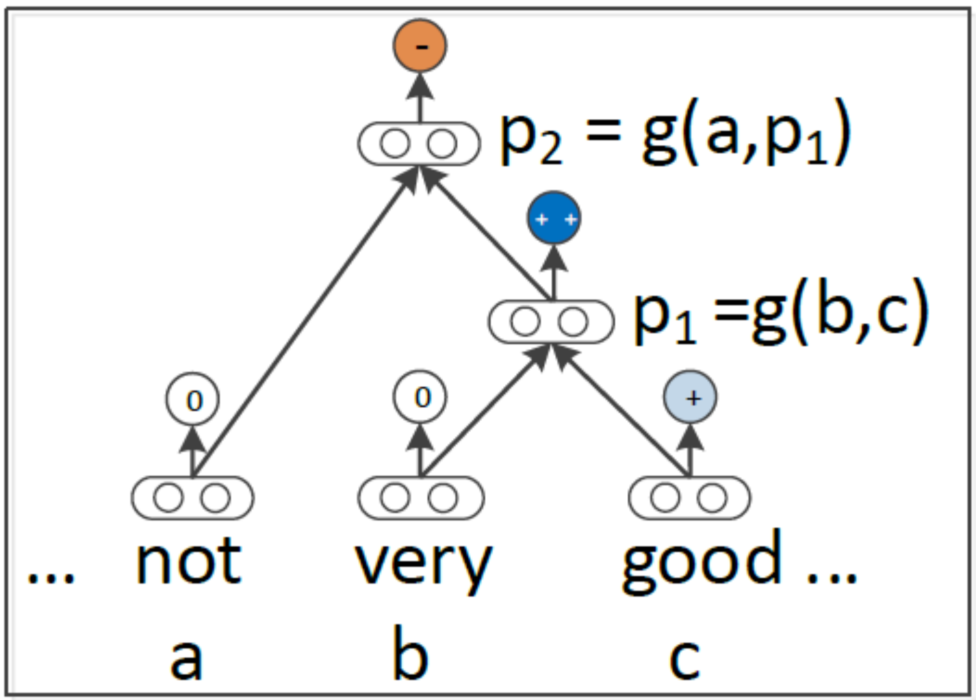


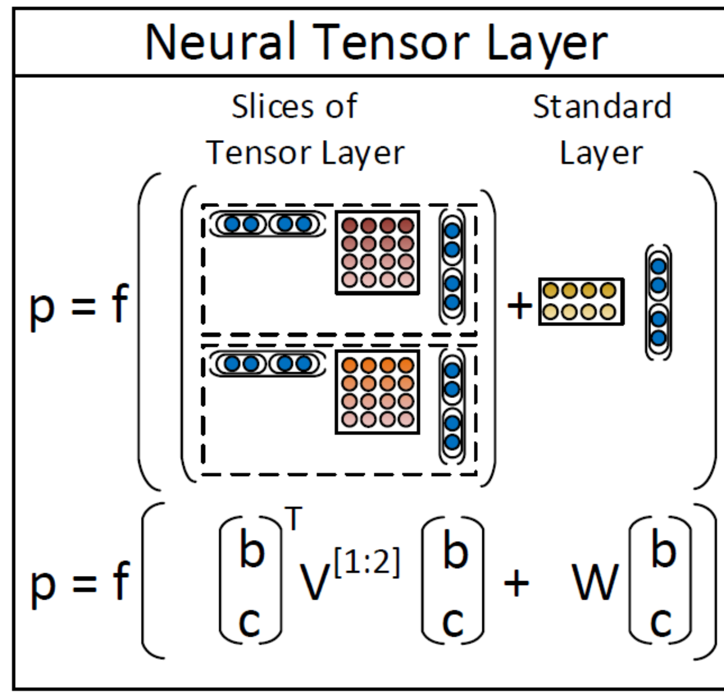
Classification of Semantic Relationships

Classifier	Features	F1
SVM	POS, stemming, syntactic patterns	60.1
MaxEnt	POS, WordNet, morphological features, noun compound system, thesauri, Google n-grams	77.6
SVM	POS, WordNet, prefixes, morphological features, dependency parse features, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner	82.2
RNN	–	74.8
MV-RNN	–	79.1
MV-RNN	**POS, WordNet, NER**	**82.4**

Version 4: Recursive Neural Tensor Network

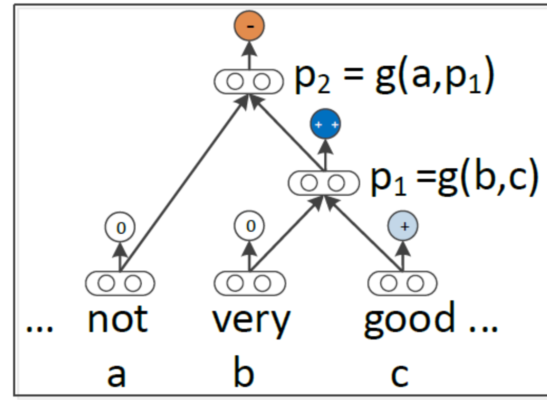
- Less parameters than MV-RNN
- Allows the two word or phrase vectors to interact multiplicatively

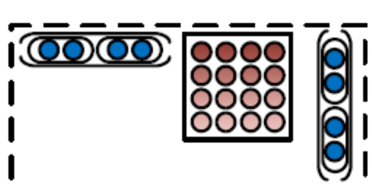




Version 4: Recursive Neural Tensor Network

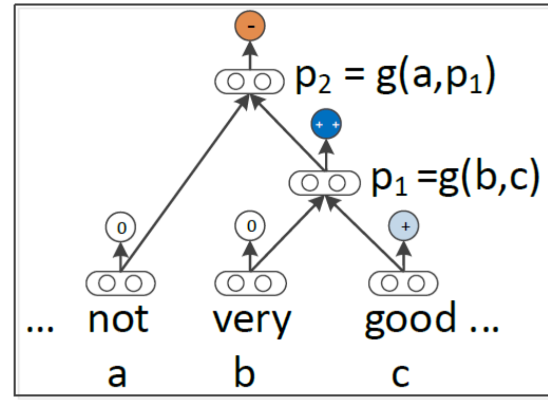
- Idea: Allow both additive and mediated multiplicative interactions of vectors [Mitchell & Lapata, 2010]

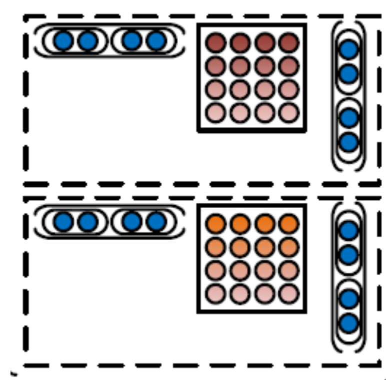




$$\begin{bmatrix} b \\ c \end{bmatrix}^T V \qquad \begin{bmatrix} b \\ c \end{bmatrix}$$

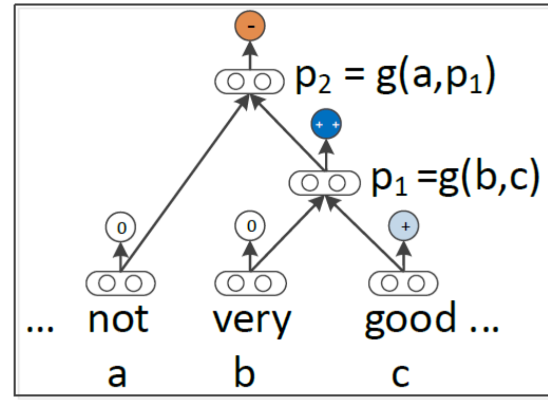
Recursive Neural Tensor Network

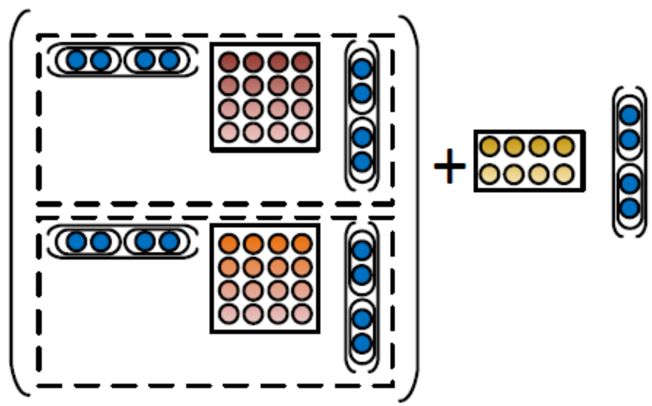




$$\begin{pmatrix} b \\ c \end{pmatrix}^T V^{[1:2]} \begin{pmatrix} b \\ c \end{pmatrix}$$

Recursive Neural Tensor Network

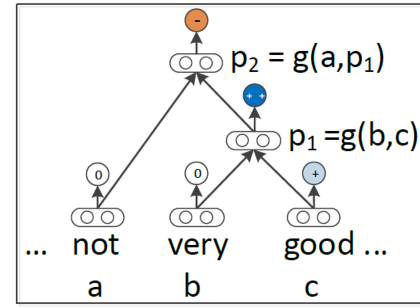




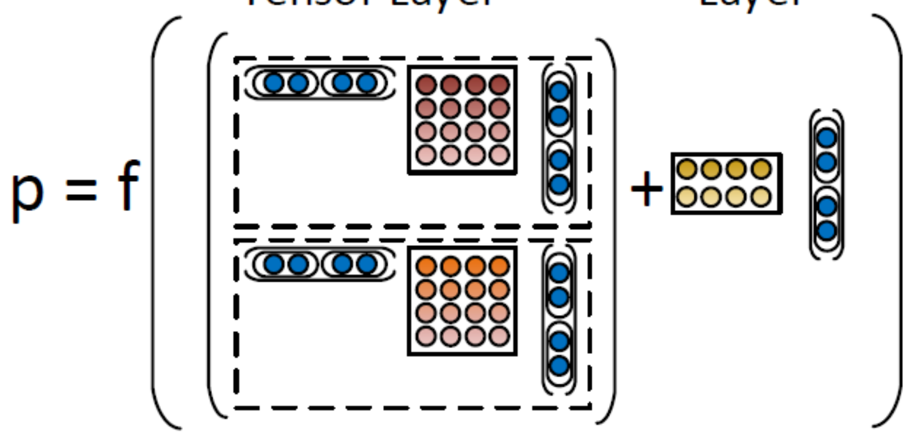
$$\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:2]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}$$

Recursive Neural Tensor Network

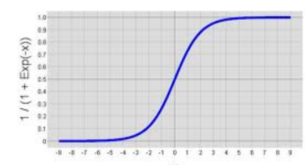
- Use resulting vectors in tree as input to a classifier like logistic regression
- Train all weights jointly with gradient descent



Neural Tensor Layer



$$p = f\left(\begin{bmatrix}b\\c\end{bmatrix}^T V^{[1:2]}\begin{bmatrix}b\\c\end{bmatrix} + W\begin{bmatrix}b\\c\end{bmatrix}\right)$$



Version 5: Improving Deep Learning Semantic Representations using a TreeLSTM

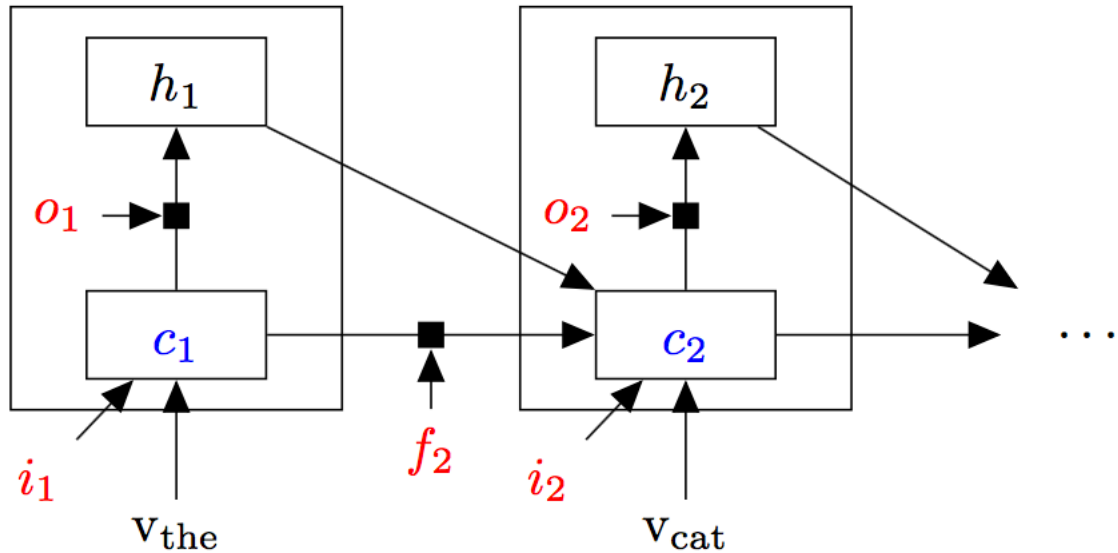
[Tai et al., ACL 2015]

Goals:

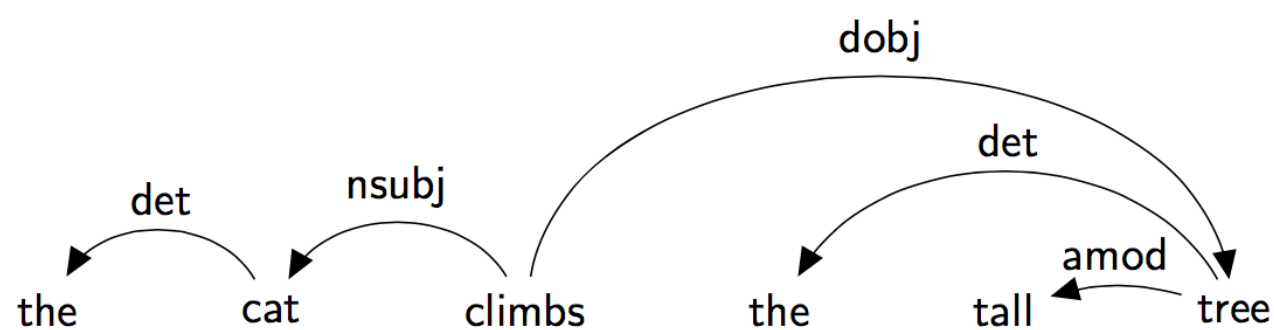
- Still trying to represent the meaning of a sentence as a location in a (high-dimensional, continuous) vector space
- In a way that accurately handles semantic composition and sentence meaning
- Beat Paragraph Vector!

Tree-Structured Long Short-Term Memory Networks

Use Long Short-Term Memories (Hochreiter and Schmidhuber 1997)



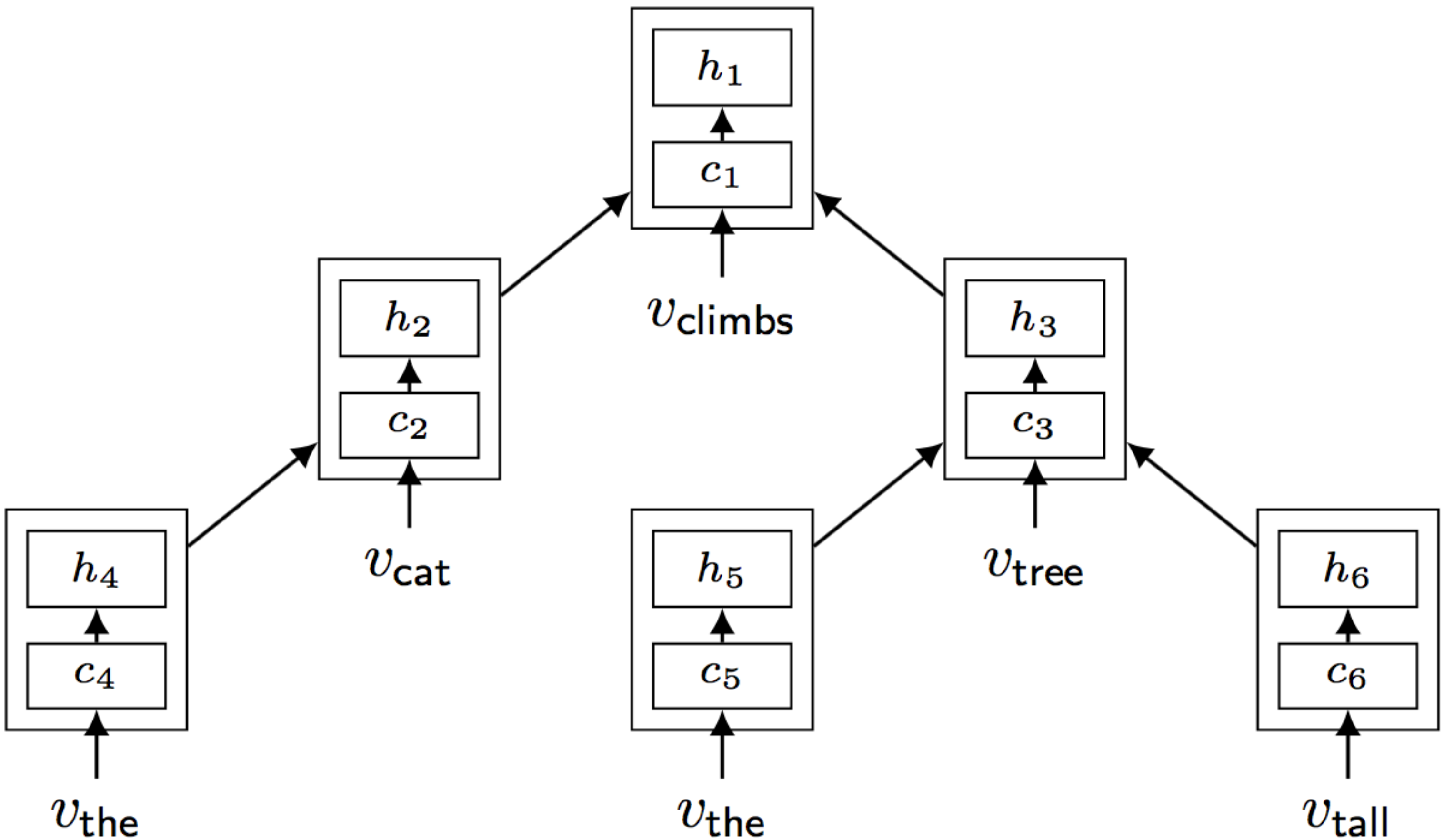
Use syntactic structure



- An LSTM creates a sentence representation via left-to-right composition
- Natural language has syntactic structure
- We can use this additional structure over inputs to guide how representations should be composed

Tree-Structured Long Short-Term Memory Networks

[Tai et al., ACL 2015]



Results: Semantic Relatedness
SICK 2014 (Sentences Involving Compositional Knowledge)

Method	Pearson correlation
Meaning Factory (Bjerva et al. 2014)	0.827
ECNU (Zhao et al. 2014)	0.841
LSTM (sequence model)	0.853
Tree LSTM	**0.868**

Natural Language Inference

Can we tell if one piece of text follows from another?

- *Two senators received contributions engineered by lobbyist Jack Abramoff in return for political favors.*
- *Jack Abramoff attempted to bribe two legislators.*

Natural Language Inference = Recognizing Textual Entailment [Dagan 2005, MacCartney & Manning, 2009]

The task: Natural language inference

James Byron Dean refused to move without blue jeans

{**entails**, contradicts, neither}

James Dean didn’t dance without pants

MacCartney's natural logic

An implementable logic for natural language inference without logical forms. (MacCartney and Manning '09)

- Sound logical interpretation (Icard and Moss '13)

P	*James Dean*	*refused to*			*move*	*without*	*blue*	*jeans*
H	*James Byron Dean*		*did*	*n't*	*dance*	*without*		*pants*
edit index	1	2	3	4	5	6	7	8
edit type	SUB	DEL	INS	INS	SUB	MAT	DEL	SUB
lex feats	strsim= 0.67	implic: –/o	cat:aux	cat:neg	hypo			hyper
lex entrel	$=$	$\vert$	$=$	$\wedge$	$\sqsupset$	$=$	$\sqsubset$	$\sqsubset$
projectivity	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$
atomic entrel	$=$	$\vert$	$=$	$\wedge$	$\sqsubset$	$=$	$\sqsubset$	$\sqsubset$

inversion

The task: Natural language inference

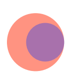
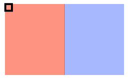
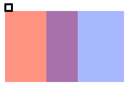
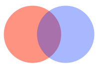
Claim: Simple task to define, but engages the full complexity of compositional semantics:

- Lexical entailment
- Quantification
- Coreference
- Lexical/scope ambiguity
- Commonsense knowledge
- Propositional attitudes
- Modality
- Factivity and implicativity

  ...

Natural logic: relations

Seven possible relations between phrases/sentences:

	$x \equiv y$	equivalence	couch $\equiv$ sofa
	$x \sqsubset y$	forward entailment (strict)	crow $\sqsubset$ bird
	$x \sqsupset y$	reverse entailment (strict)	European $\sqsupset$ French
	$x \wedge y$	negation (exhaustive exclusion)	human ^ nonhuman
	$x \mid y$	alternation (non-exhaustive exclusion)	cat | dog
	$x \smile y$	cover (exhaustive non-exclusion)	animal ‿ nonhuman
	$x \# y$	independence	hungry # hippo

Natural logic: relation joins

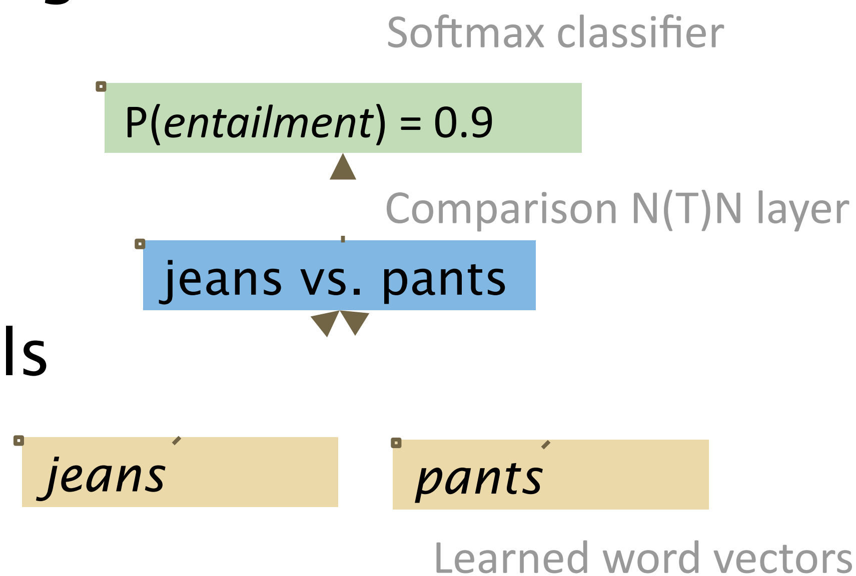
	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\equiv$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\sqsubset$	$\sqsubset$	$\sqsubset$	$\cdot$	$\mid$	$\mid$	$\cdot$	$\cdot$
$\sqsupset$	$\sqsupset$	$\cdot$	$\sqsupset$	$\smile$	$\cdot$	$\smile$	$\cdot$
$\wedge$	$\wedge$	$\smile$	$\mid$	$\equiv$	$\sqsupset$	$\sqsubset$	$\#$
$\mid$	$\mid$	$\cdot$	$\mid$	$\sqsubset$	$\cdot$	$\sqsubset$	$\cdot$
$\smile$	$\smile$	$\smile$	$\cdot$	$\sqsupset$	$\sqsupset$	$\cdot$	$\cdot$
$\#$	$\#$	$\cdot$	$\cdot$	$\#$	$\cdot$	$\cdot$	$\cdot$

Can our NNs learn to make these inferences over pairs of embedding vectors?

A minimal NN for lexical relations

[Bowman 2014]

- Words are learned embedding vectors.
- One plain TreeRNN or TreeRNTN layer
- Softmax emits relation labels
- Learn everything with SGD.



Lexical relations: results

	Train	Test
# only	53.8 (10.5)	53.8 (10.5)
15d NN	99.8 (99.0)	94.0 (87.0)
15d NTN	**100 (100)**	**99.6 (95.5)**

- Both models tuned, then trained to convergence on five randomly generated datasets
- Reported figures: % correct (macroaveraged F1)
- Both NNs used 15d embeddings, 75d comparison layer

Quantifiers

Experimental paradigm: Train on relational statements generated from some formal system, test on other such relational statements.

The model needs to:

- Learn the relations between individual words. (lexical relations)
- Learn how lexical relations impact phrasal relations. (projectivity)
- Quantifiers present some of the harder cases of both of these.

Quantifiers

- Small vocabulary
  - Three basic types:
    - Quantifiers: *some*, *all*, *no*, *most*, *two*, *three, not-all, not-most, less-than-two, less-than-three*
    - Predicates: *dog, cat, mammal, animal …*
    - Negation: *not*
- 60k examples generated using a generative implementation of the relevant portion of MacCartney and Manning's logic.
- All sentences of the form *QPP*, with optional negation on each predicate.

(most warthogs) walk	$\wedge$	(not-most warthogs) walk
(most mammals) move	$\#$	(not-most (not turtles)) move
(most (not pets)) (not swim)	$\sqsupset$	(not-most (not pets)) move

Quantifier results

	Train	Test
Most freq. class (# only)	35.4%	35.4%
25d SumNN (sum of words)	96.9%	93.9%
25d TreeRNN	99.6%	99.2%
25d TreeRNTN	**100%**	**99.7%**

Natural language inference data

[Bowman, Manning & Potts 2015]

- To do NLI on real English, we need to teach an NN model English almost from scratch.
- What data do we have to work with:
  - GloVe/word2vec (useful w/ any data source)
  - SICK: Thousands of examples created by editing and pairing hundreds of sentences.
  - RTE: Hundreds of examples created by hand.
  - DenotationGraph: Millions of extremely noisy examples (~73% correct?) constructed fully automatically.

Results on SICK (+DG, +tricks) so far

	SICK Train	DG Train	Test
Most freq. class	**56.7%**	**50.0%**	**56.7%**
30 dim TreeRNN	95.4%	67.0%	74.9%
50 dim TreeRNTN	97.8%	74.0%	**76.9%**

Is it competitive? Sort of...

Best result (UIllinois)	**84.5%**
≈ interannotator agreement!	
Median submission (out of 18):	77%
TreeRNTN:	76.9%

TreeRNTN is a purely-learned system
None of the ones in the competition were

Natural language inference data

- To do NLI on real English, we need to teach an NN model English almost from scratch.
- What data do we have to work with:
  - GloVe/word2vec (useful w/ any data source)
  - SICK: Thousands of examples created by editing and pairing hundreds of sentences.
  - RTE: Hundreds of examples created by hand.
  - DenotationGraph: Millions of extremely noisy examples (~73% correct?) constructed fully automatically.
  - **Stanford NLI corpus: ~600k examples, written by Turkers.**

The Stanford NLI corpus

Instructions

The Stanford University NLP Group is collecting data for use in research on computer understanding of English. We appreciate your help!

We will show you the caption for a photo. We will not show you the photo. Using only the caption and what you know about the world:

- Write one alternate caption that is **definitely** a **true** description of the photo.
- Write one alternate caption that **might be** a **true** description of the photo.
- Write one alternate caption that is **definitely** an **false** description of the photo.

Photo caption **A little boy in an apron helps his mother cook.**

Definitely correct Example: For the caption *"Two dogs are running through a field."* you could write *"There are animals outdoors."*

Write a sentence that follows from the given caption.

Maybe correct Example: For the caption *"Two dogs are running through a field."* you could write *"Some puppies are running to catch a stick."*

Write a sentence which may be true given the caption, and may not be.

Definitely incorrect Example: For the caption *"Two dogs are running through a field."* you could write *"The pets are sitting on a couch."*

Write a sentence which contradicts the caption.

Problems (optional) *If something is wrong with the caption that makes it difficult to understand, do your best above and let us know here.*

Envoi

There are very good reasons to want to represent meaning with distributed representations

So far, distributional learning has been most effective for this

- But cf. [Young, Lai, Hodosh & Hockenmaier 2014] on denotational representations, using visual scenes

However, we want not just word meanings! We want:

- Meanings of larger units, calculated compositionally
- The ability to do natural language inference