

# Natural Language Inference, Reading Comprehension and Deep Learning



Christopher Manning

@chrmanning • @stanfordnlp

Stanford University

SIGIR 2016

# Machine Comprehension Tested by question answering (Burges)

“A machine **comprehends** a passage of **text** if, for any **question** regarding that text that can be **answered** correctly by a majority of native speakers, that machine can provide a string which those speakers would agree both answers that question, and does not contain information irrelevant to that question.”

Towards the Machine Comprehension of Text: An Essay

Christopher J.C. Burges  
Microsoft Research  
One Microsoft Way  
Redmond, WA 98052, USA

December 23, 2013

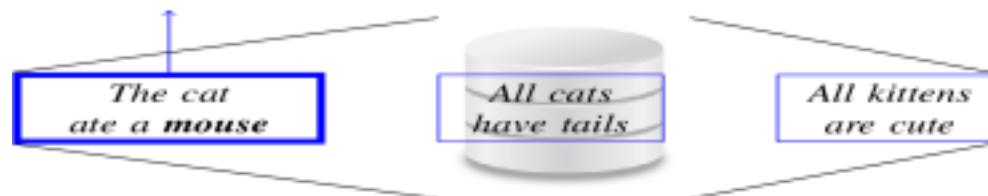


# IR needs Language understanding

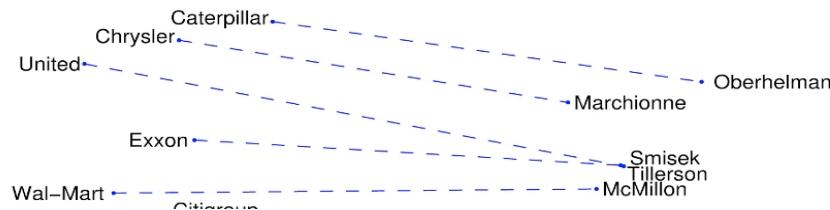
- There were some things that kept IR and NLP apart
  - IR was heavily focused on efficiency and scale
  - NLP was way too focused on form rather than meaning
- Now there are compelling reasons for them to come together
  - Taking IR precision and recall to the next level
    - [car parts for sale]
    - **Should match:** Selling automobile and pickup engines, transmissions
      - Example from Jeff Dean's WSDM 2016 talk
  - Information retrieval/question answering in mobile contexts
    - Web snippets no longer cut it on a watch!

# Menu

## 1. Natural logic: A weak logic over human languages for inference



## 2. Distributed word representations



## 3. Deep, recursive neural network language understanding



How can information retrieval  
be viewed more as theorem  
proving (than matching)?

## AI2 4<sup>th</sup> Grade Science Question

Answering [Angeli, Nayak, & Manning, ACL 2016]

Our “knowledge”:

*Ovaries are the female part of the flower, which produces eggs that are needed for making seeds.*

The question:

*Which part of a plant produces the seeds?*

The answer choices:

**the flower**

the leaves

the stem

the roots

# How can we represent and reason with broad-coverage knowledge?

1. Rigid-schema knowledge bases with well-defined logical inference
2. Open-domain knowledge bases (Open IE) – no clear ontology or inference  
[Etzioni et al. 2007ff]
3. Human language text KB – No rigid schema, but with “Natural logic” can do formal inference over human language text [MacCartney and Manning 2008]



# Natural Language Inference

[Dagan 2005, MacCartney & Manning, 2009]

Does a piece of text follows from or contradict another?

*Two senators received contributions engineered  
by lobbyist Jack Abramoff in return for political favors.*

*Jack Abramoff attempted to bribe two legislators.*

Follows

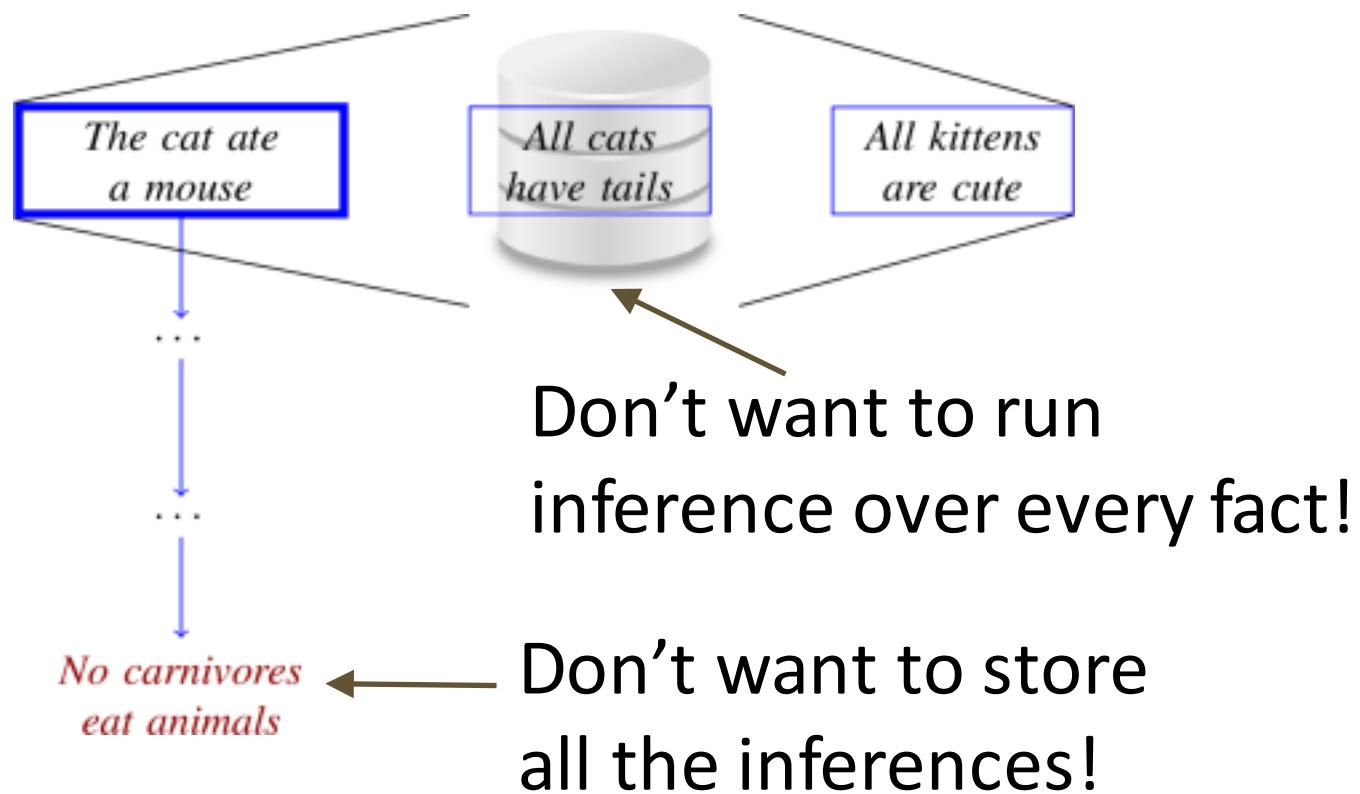
Here try to prove or refute according to a large text collection:

1. The flower of a plant produces the seeds
2. The leaves of a plant produces the seeds
3. The stem of a plant produces the seeds
4. The roots of a plant produces the seeds

# Text as Knowledge Base

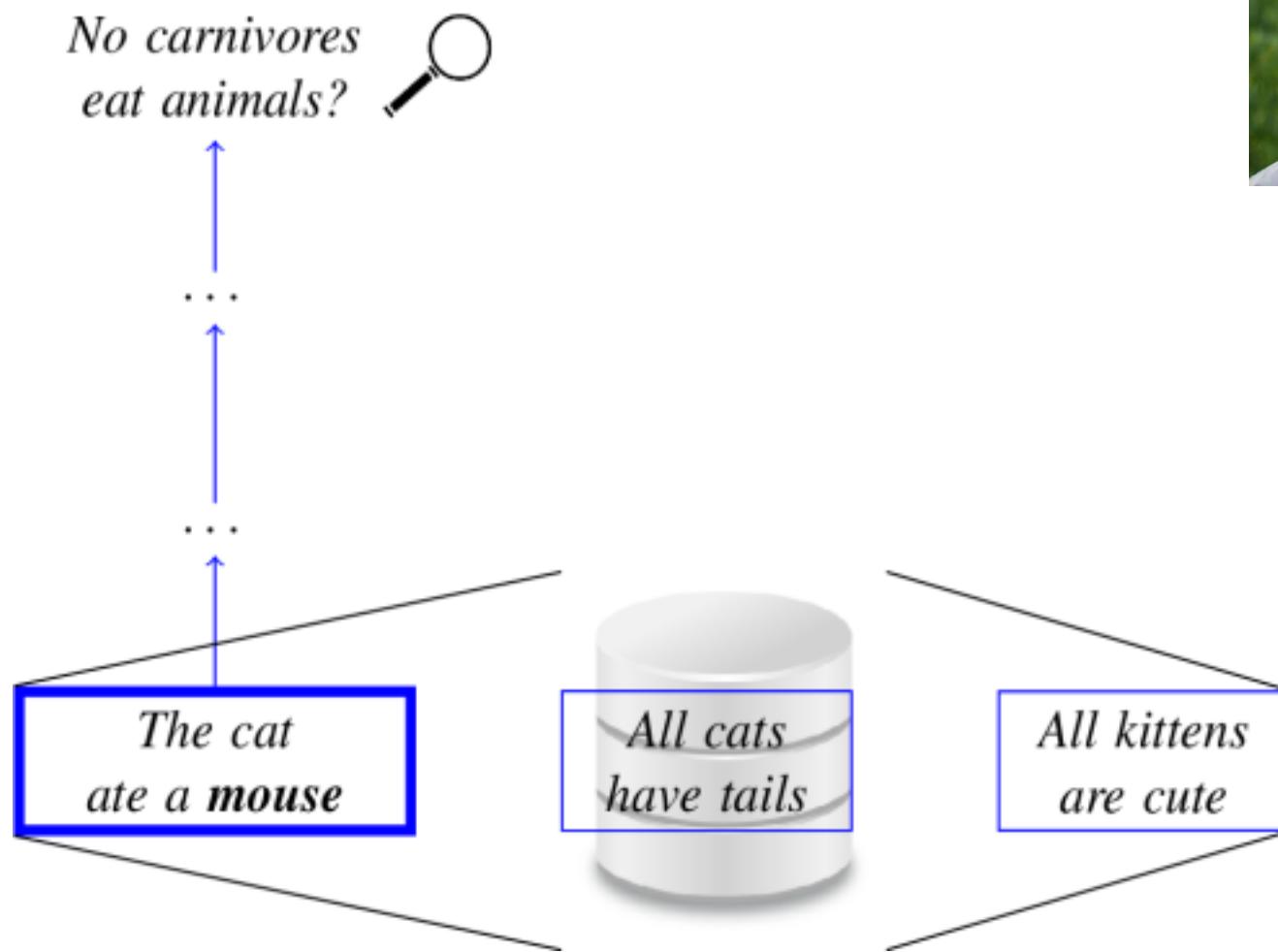
Storing knowledge as text is easy!

Doing inferences over text might be hard



# Inferences ... on demand from a query ...

[Angeli and Manning 2014]



# ... using text as the meaning representation

*No carnivores  
eat animals?*



***The carnivores  
eat animals***

***The cat  
eats animals***

***The cat  
ate an animal***

***The cat  
ate a mouse***

***All cats  
have tails***

***All kittens  
are cute***



# Natural Logic: logical inference over text

We are doing logical inference

*The cat ate a mouse*  $\models \neg \text{No carnivores eat animals}$

We do it with natural logic

If I mutate a sentence in this way, do I preserve its truth?

*Post-Deal Iran Asks if U.S. Is Still ‘Great Satan,’ or Something Less*  $\models$   
*A Country Asks if U.S. Is Still ‘Great Satan,’ or Something Less*

- A sound and complete weak logic [Icard and Moss 2014]
- Expressive for common human inferences\*
- “Semantic” parsing is just syntactic parsing
- Tractable: Polynomial time entailment checking
- Plays nicely with lexical matching back-off methods

# #1. Common sense reasoning

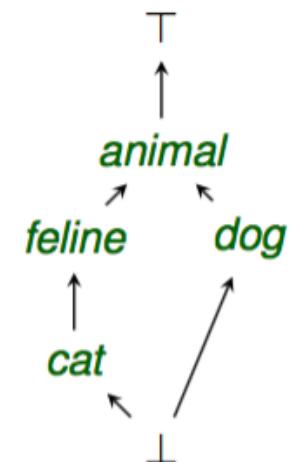
## Polarity in Natural Logic

We order phrases in *partial orders*

Simplest one: is-a-kind-of

Also: geographical containment, etc.

*Polarity:* In a certain context, is it valid  
to move up or down in this order?



animal

feline

↑ cat

house cat

# Example inferences

Quantifiers determine the *polarity* of phrases

Valid mutations consider polarity

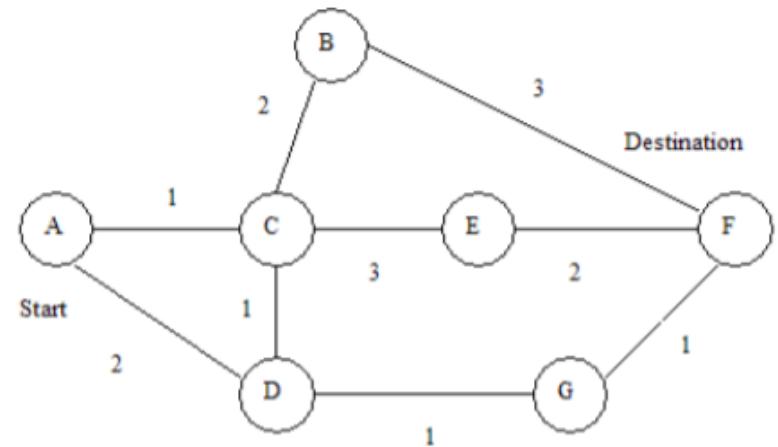


Successful toy inference:

- *All cats eat mice*  $\models$  *All house cats consume rodents*

# “Soft” Natural Logic

- We also want to make likely (but not certain) inferences
- Same motivation as Markov logic, probabilistic soft logic, etc.
- Each mutation *edge template feature* has a cost  $\theta \geq 0$
- Cost of an edge is  $\theta_i \cdot f_i$
- Cost of a path is  $\theta \cdot f$
- Can learn parameters  $\theta$
- Inference is then graph search



## #2. Dealing with real sentences

Natural logic works with facts like these in the knowledge base:

*Obama was born in Hawaii*

But real-world sentences are complex and long:

***Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the Harvard Law Review.***

Approach:

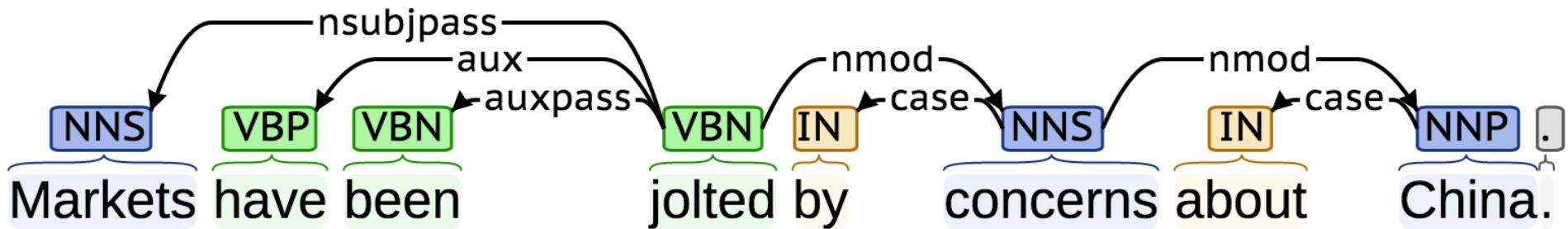
1. Classifier divides long sentences into entailed clauses
2. Natural logic inference can shorten these clauses

# Universal Dependencies (UD)

<http://universaldependencies.github.io/docs/>

A single level of typed dependency syntax that

- (i) works for all human languages
- (ii) gives a simple, human-friendly representation of sentence

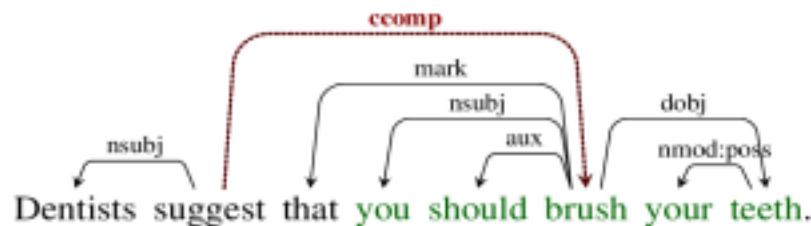


Dependency syntax is better than a phrase-structure tree for machine interpretation – it's almost a semantic network

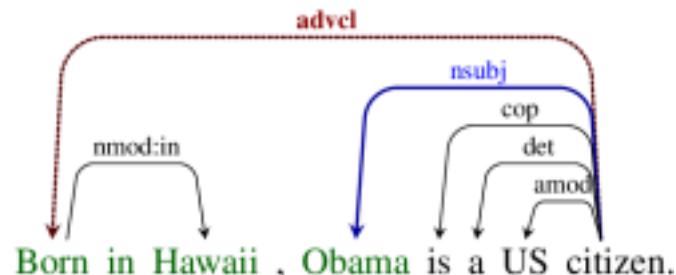
UD aims to be **linguistically better across languages** than earlier representations, such as CoNLL dependencies

# Generation of minimal clauses

1. Classification problem:  
given a dependency edge,  
does it introduce a clause?



2. Is it missing a controlled subject from subj/object?

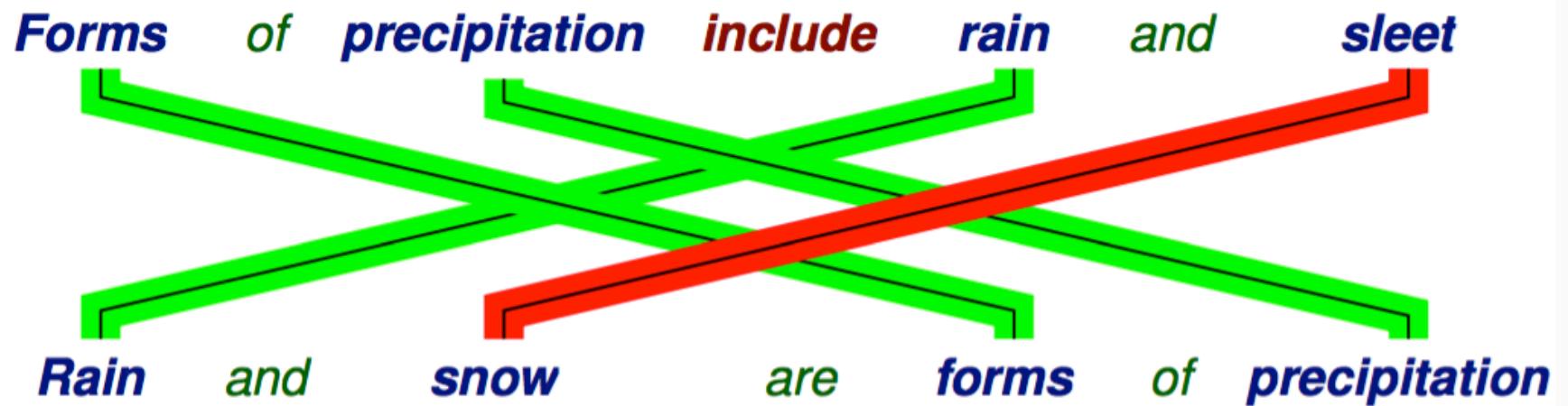


3. Shorten clauses while preserving validity, using natural logic!

- *All young rabbits drink milk*  
 $\neq$  *All rabbits drink milk*
- **OK:** *SJC, the Bay Area's third largest airport, often experiences delays due to weather.*
- **Often better:** *SJC often experiences delays.*

## #3. Add a Lexical alignment classifier

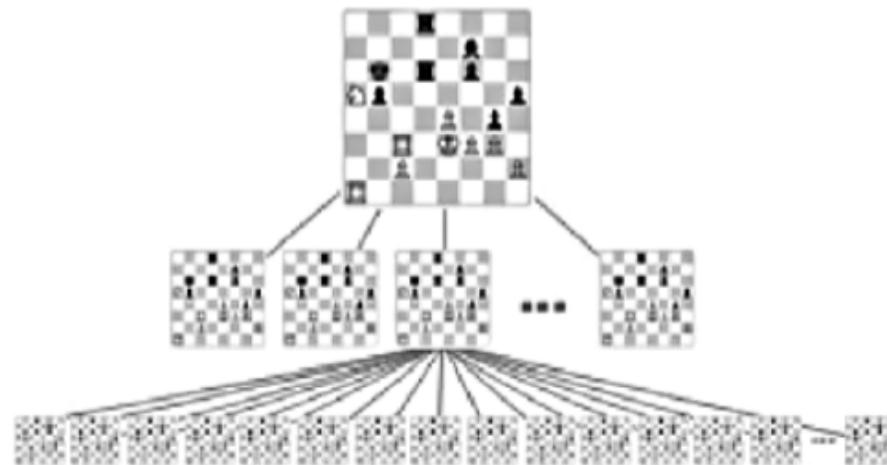
- Sometimes we can't quite make the inferences that we would like to make:



- We use a simple lexical match back-off classifier with features:
  - Matching words, mismatched words, unmatched words
  - These always work pretty well
    - This was the lesson of RTE evaluations and perhaps or IR in general

# The full system

- We run our usual search over split up, shortened clauses
  - If we find a premise, great!
  - If not, we use the lexical classifier as an *evaluation function*



- We work to do this quickly at scale
  - Visit 1M nodes/second, don't refeaturize, just delta
  - 32 byte search states (thanks Gabor!)

# Solving NY State 4<sup>th</sup> grade science (Allen AI Institute datasets)

**Multiple choice questions from real 4th grade science exams**

Which activity is an example of a good health habit?

- (A) Watching television (B) Smoking cigarettes (C) Eating candy
- (D) Exercising every day

In our ~~corpus~~ knowledge base:

- *Plasma TV's can display up to 16 million colors ... great for watching TV ... also make a good screen.*
- *Not smoking or drinking alcohol is good for health, regardless of whether clothing is worn or not.*
- *Eating candy for dinner is an example of a poor health habit.*
- *Healthy is exercising*

# Solving 4<sup>th</sup> grade science (ALLEN AI NDMC)

System	Dev	Test
KnowBot [Hixon et al. NAACL 2015]	45	–
KnowBot (augmented with <b>human</b> in loop)	57	–
IR baseline (Lucene)	49	42
NaturalLI	52	51
More data + IR baseline	62	58
More data + NaturalLI	65	61
NaturalLI +  +  (lex. classifier)	74	67
Aristo [Clark et al. 2016] 6 systems, even more data		<b>71</b>

Test set: New York Regents 4th Grade Science exam multiple-choice questions from AI2

Training: Basic is Barron's study guide; more data is SciText corpus from AI2. Score: % correct

# Natural Logic

- Can we just use text as a knowledge base?
- Natural logic provides a useful, formal (weak) logic for textual inference
- Natural logic is easily combinable with lexical matching methods, including neural net methods
- The resulting system is useful for:
  - Common-sense reasoning
  - Question Answering
  - Open Information Extraction
    - i.e., getting out relation triples from text



Can information retrieval  
benefit from distributed  
representations of words?

## From symbolic to distributed representations

The vast majority of rule-based or statistical NLP and IR 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 now call this a “one-hot” representation.

# From symbolic to distributed representations

Its problem:

- If user searches for [Dell notebook battery size], we would like to match documents with “Dell laptop battery capacity”
- If user searches for [Seattle motel], we would like to match documents containing “Seattle hotel”

But

$$\begin{aligned} \text{motel} & [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \\ \text{hotel} & [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] = 0 \end{aligned}$$

Our query and document vectors are **orthogonal**

There is no natural notion of similarity in a set of one-hot vectors

# Capturing similarity

There are many things you can do about similarity, many well known in IR

- Query expansion with synonym dictionaries

- Learning word similarities from large corpora

But a word representation that encodes similarity wins

- Less parameters to learn (per word, not per pair)

- More sharing of statistics

- More opportunities for multi-task learning

# Distributional similarity-based representations

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

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

(J. R. Firth 1957: 11)

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* ↗

# Basic idea of Learning neural network word embeddings

We define some model that aims to predict a word based on other words in its context

$$\text{Choose } \operatorname{argmax}_w w \cdot ((w_{j-1} + w_{j+1})/2)$$

which has a loss function, e.g.,

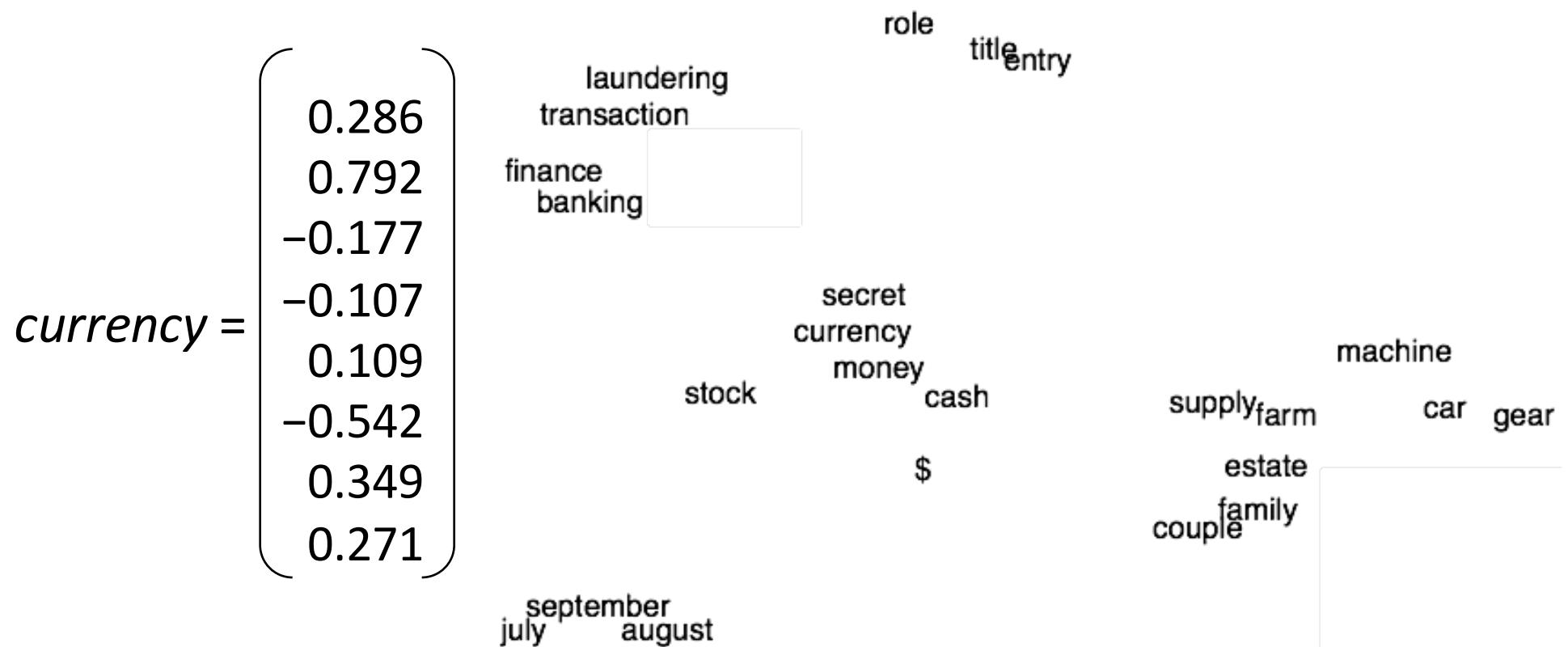
$$J = 1 - w_j \cdot ((w_{j-1} + w_{j+1})/2)$$

Unit norm  
vectors

We look at many samples from a big language corpus

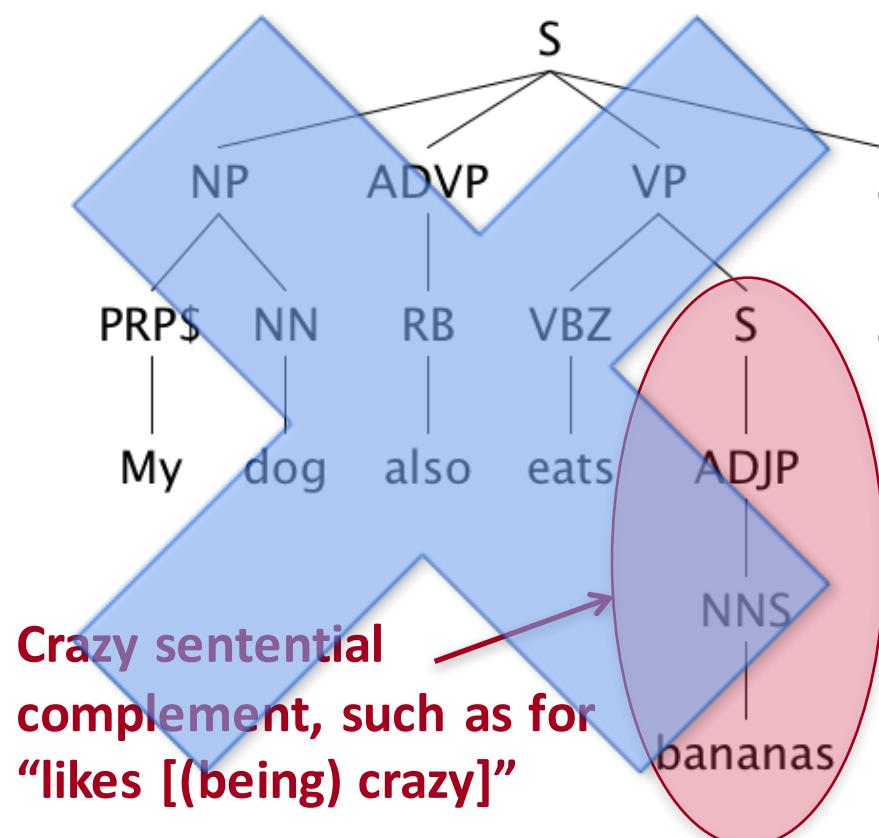
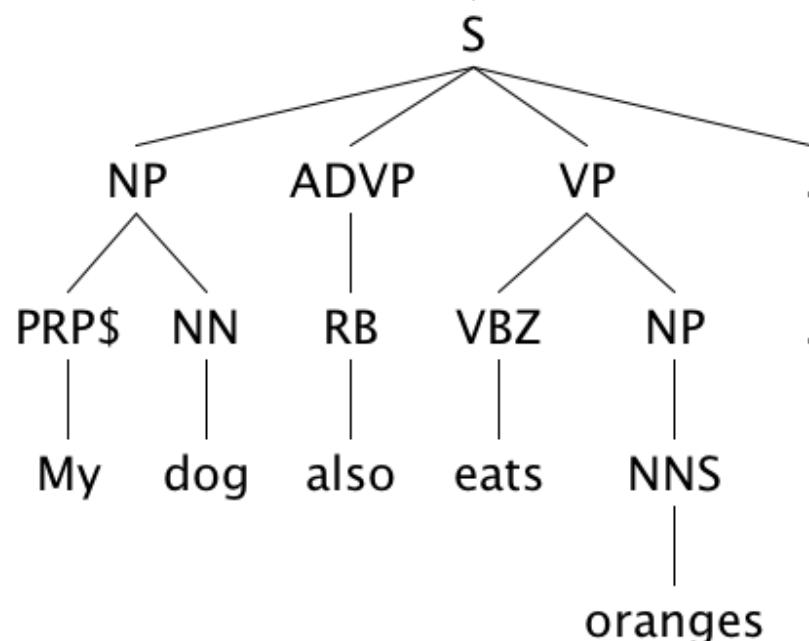
We keep adjusting the vector representations of words to minimize this loss

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



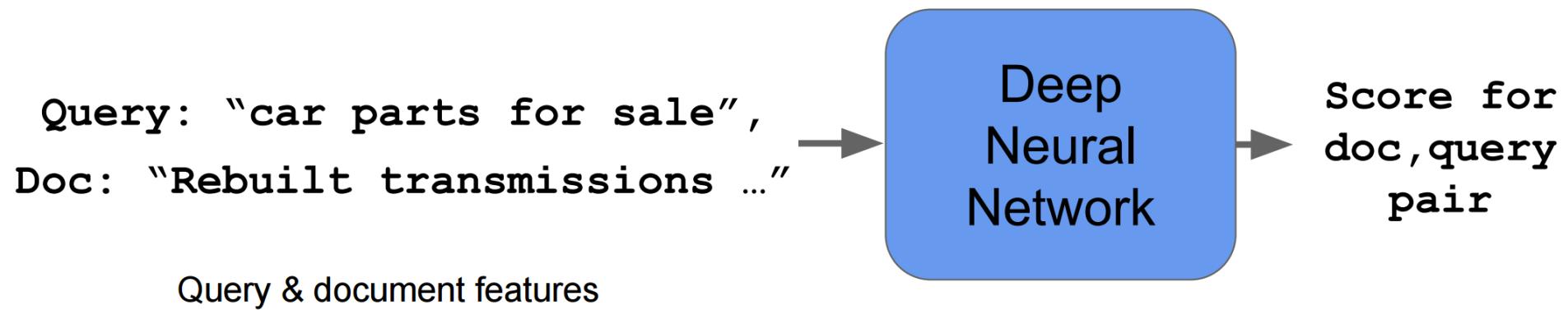
# Distributional representations can solve the fragility of NLP tools

Standard NLP systems – here, the Stanford Parser – are incredibly fragile because of symbolic representations



Distributional representations can capture the long tail of IR similarity

## Google's RankBrain



Not necessarily as good for the head of the query distribution, but great for seeing similarity in the tail

3<sup>rd</sup> most important ranking signal (we're told...)

# LSA (Latent Semantic Analysis) vs. word2vec

## LSA: Count! models

- Factorize a (maybe weighted, often log-scaled) term-document ([Deerwester et al. 1990](#)) or word-context matrix ([Schütze 1992](#)) into  $U\Sigma V^T$
- Retain only  $k$  singular values, in order to generalize

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & * \\ * & * \\ * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & \\ & \bullet & \\ & & \bullet \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{V^T}$$

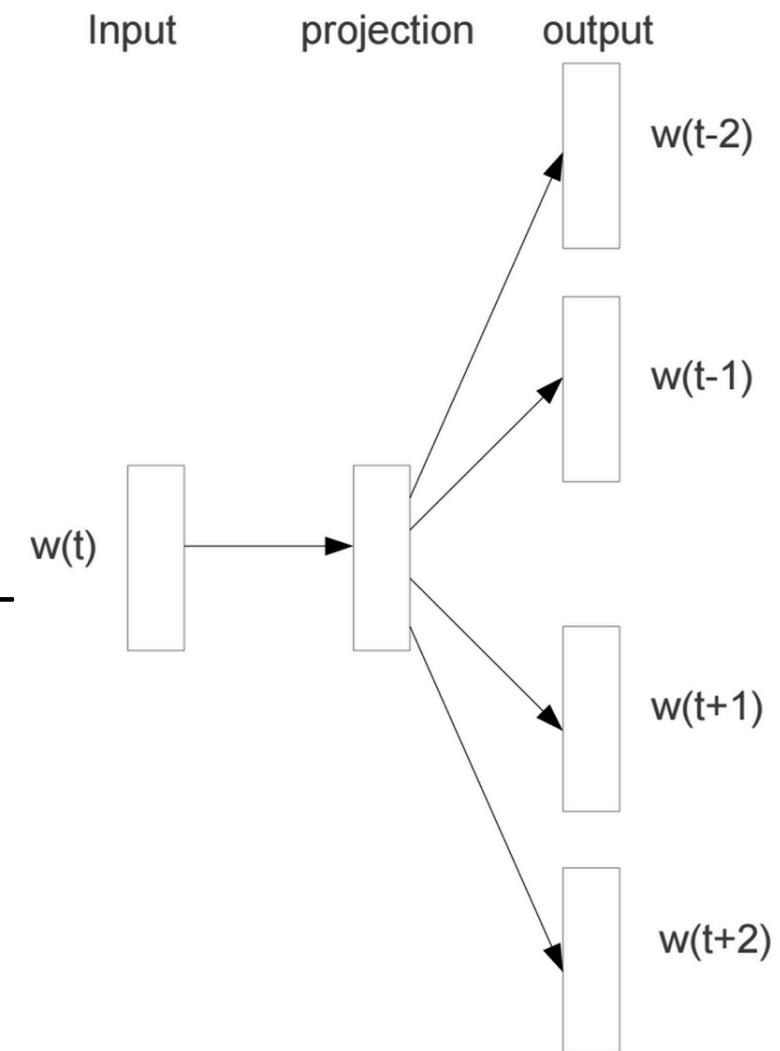
[Cf. Baroni: Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. ACL 2014]

# LSA vs. word2vec

word2vec CBOW/SkipGram: **Predict!**

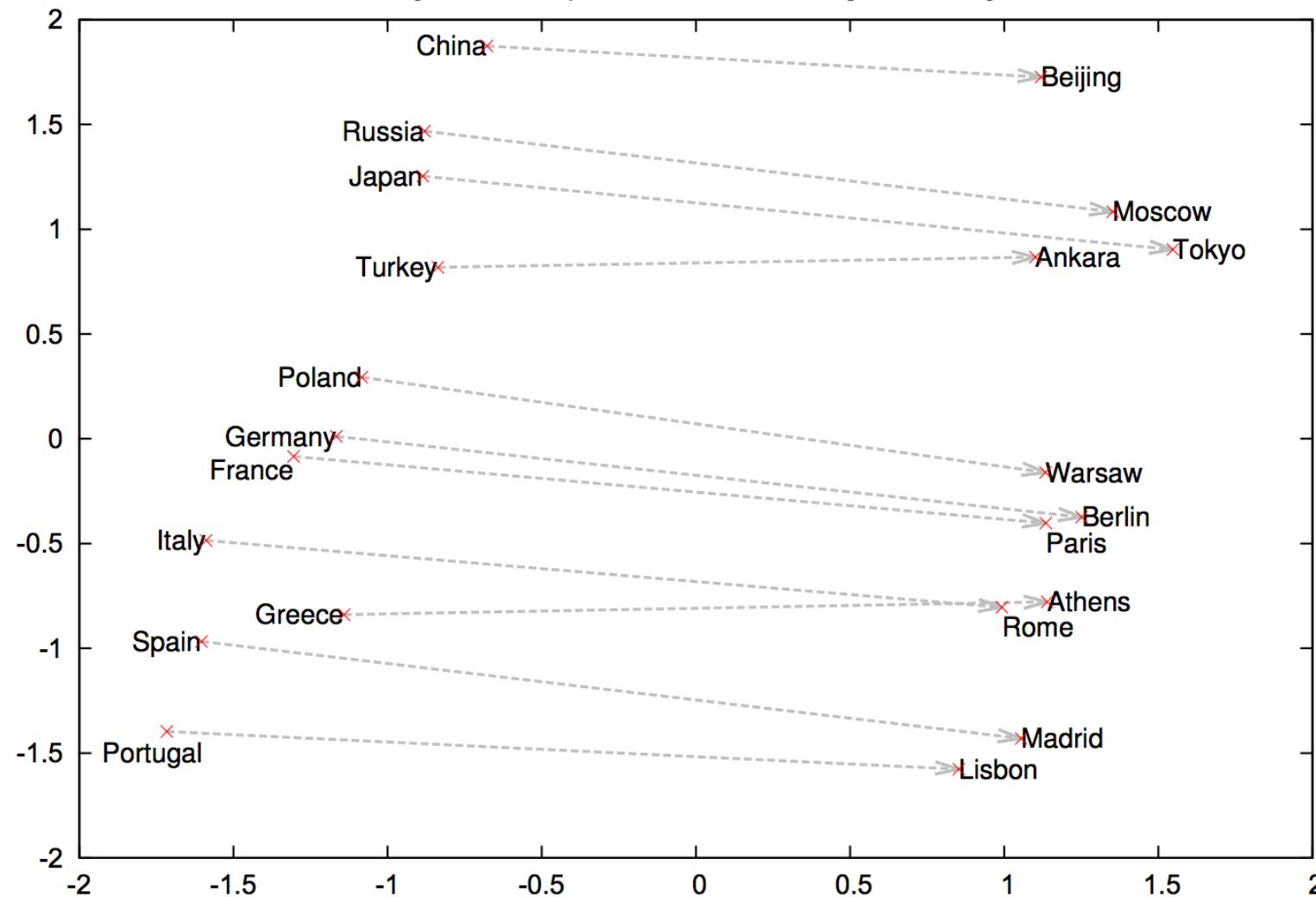
[Mikolov et al. 2013]: Simple predict models for learning word vectors

- 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

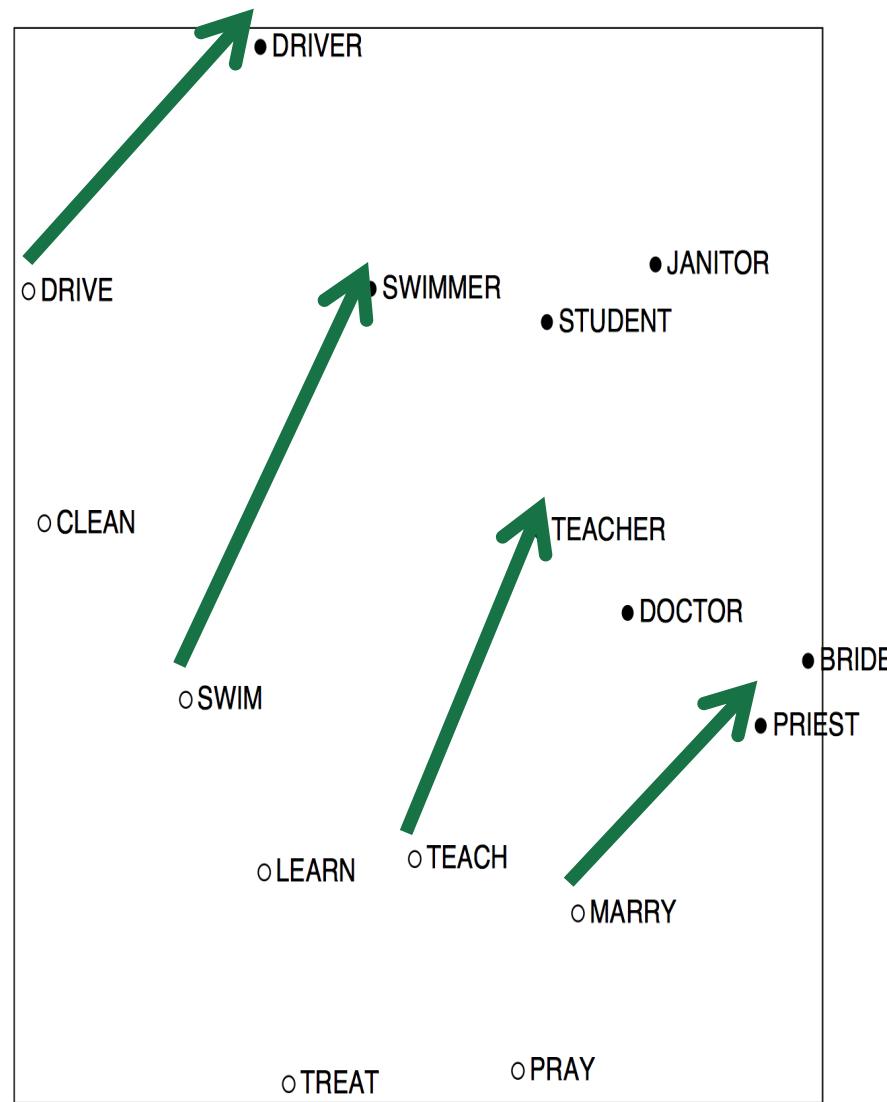


# word2vec encodes semantic components as linear relations

Country and Capital Vectors Projected by PCA



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



# 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
- May not use the best methods for scaling 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

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

# Encoding meaning in vector differences

[Pennington, Socher, and Manning, EMNLP 2014]

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

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

# Encoding meaning in vector differences

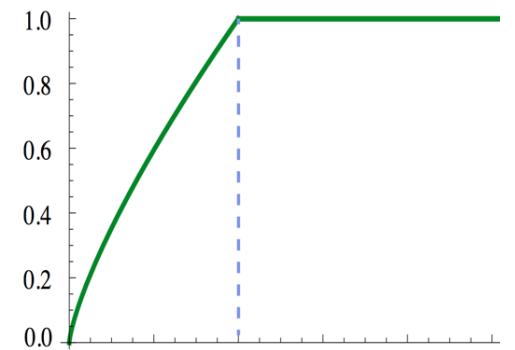
Q: How can we capture ratios of co-occurrence probabilities as meaning components in a word vector space?

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

with vector differences

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

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



# Glove Word similarities

[Pennington et al., EMNLP 2014]



Nearest words to **frog**:

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



*litoria*



*leptodactylidae*



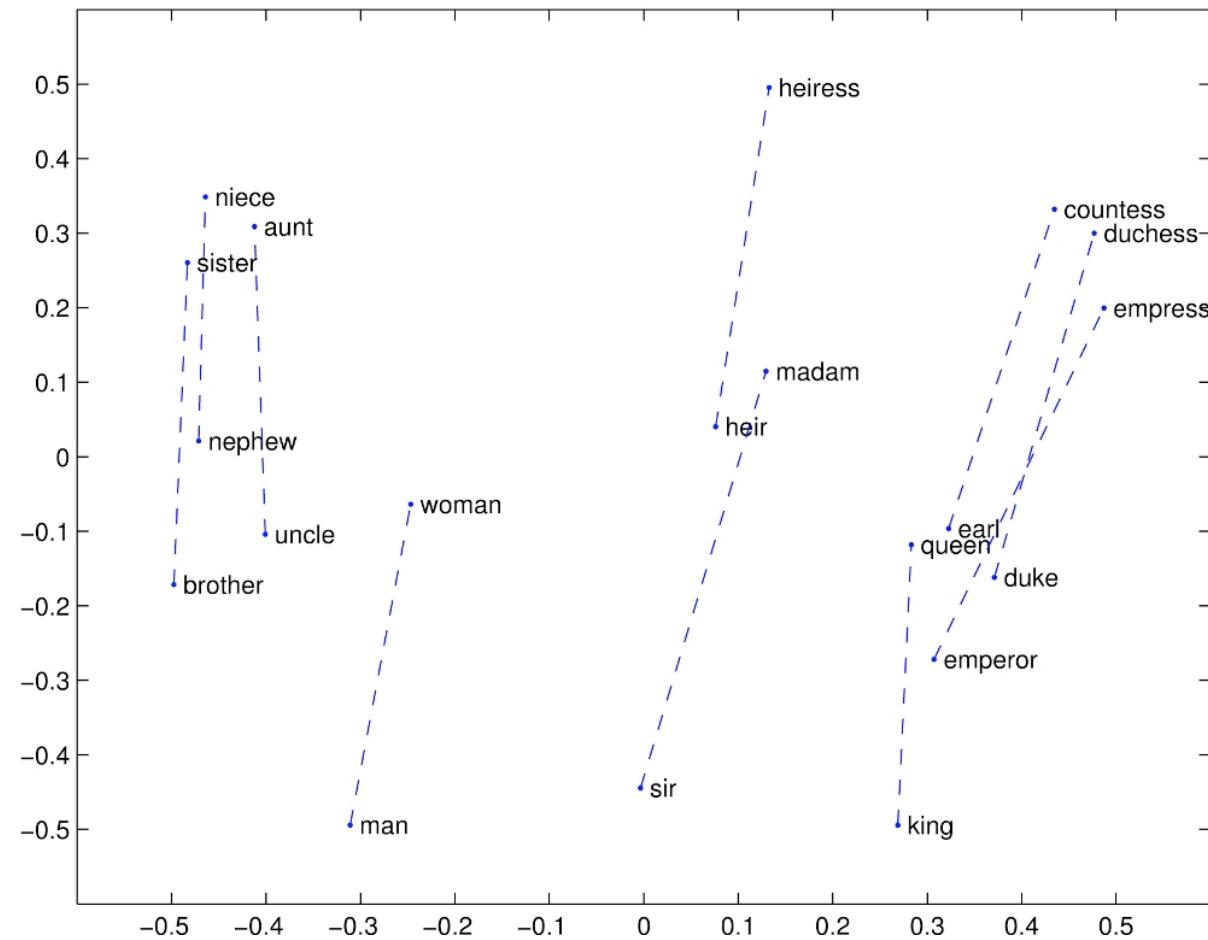
*rana*



*eleutherodactylus*

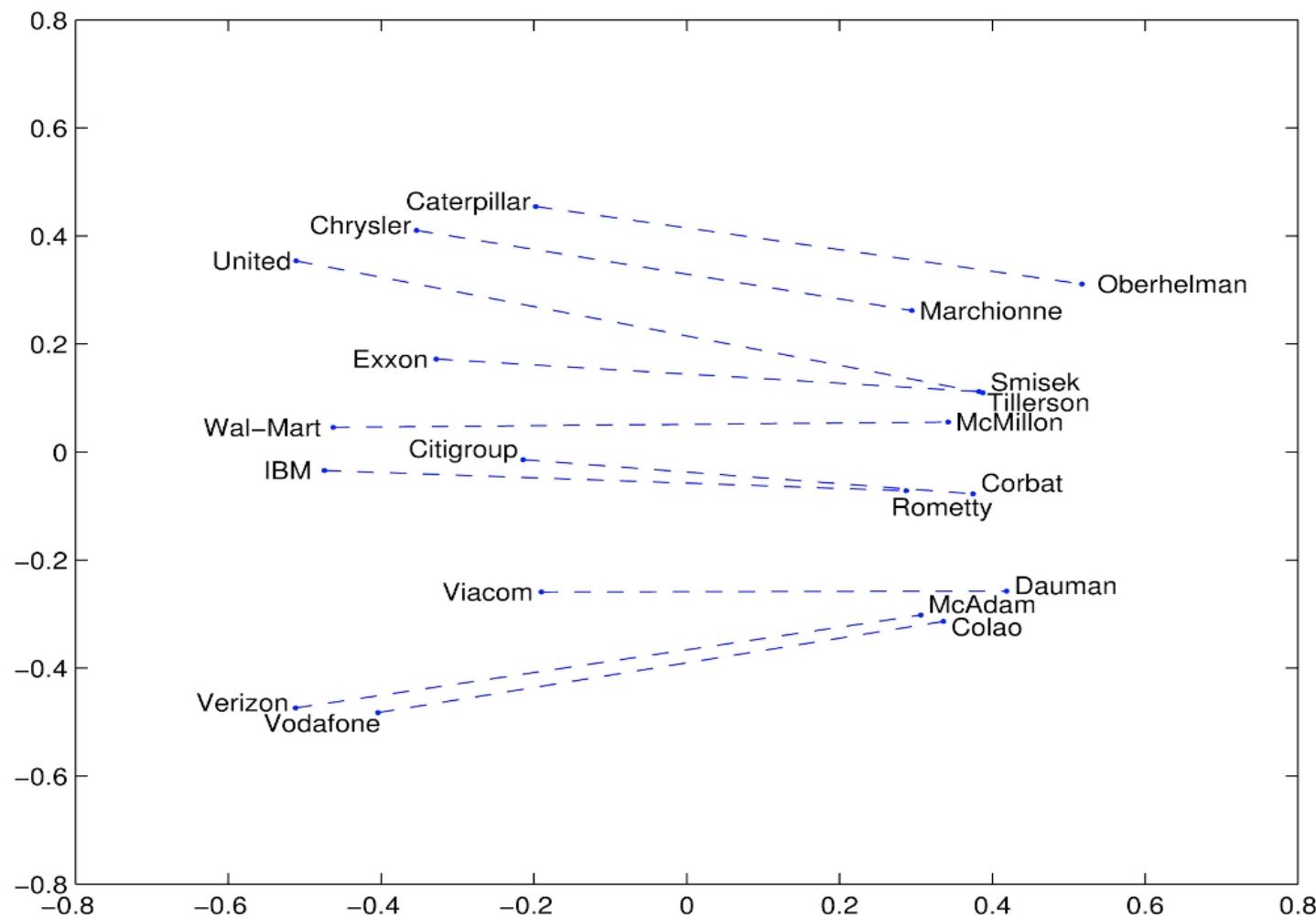
<http://nlp.stanford.edu/projects/glove/>

# Glove Visualizations



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

# Glove Visualizations: Company - CEO



# Named Entity Recognition Performance

Model on CoNLL	CoNLL '03 dev	CoNLL '03 test	ACE 2	MUC 7
Categorical CRF	91.0	85.4	77.4	73.4
SVD (log tf)	90.5	84.8	73.6	71.5
HPCA	92.6	88.7	81.7	80.7
C&W	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
<b>GloVe</b>	<b>93.2</b>	<b>88.3</b>	<b>82.9</b>	<b>82.2</b>

F1 score of CRF trained on CoNLL 2003 English with 50 dim word vectors

# Word embeddings: Conclusion

Glove translates meaningful relationships between word-word **co-occurrence counts** 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

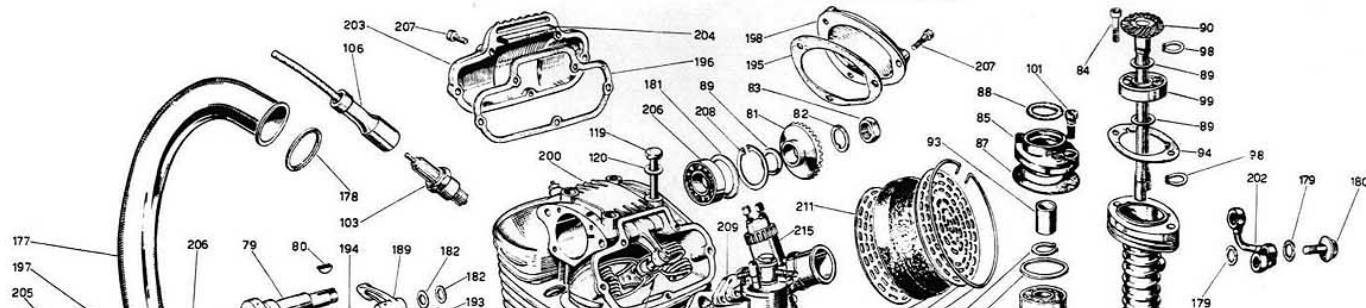
A lot of other important work in this line of research:

[Levy & Goldberg, 2014]

[Arora, Li, Liang, Ma & Risteski, 2015]

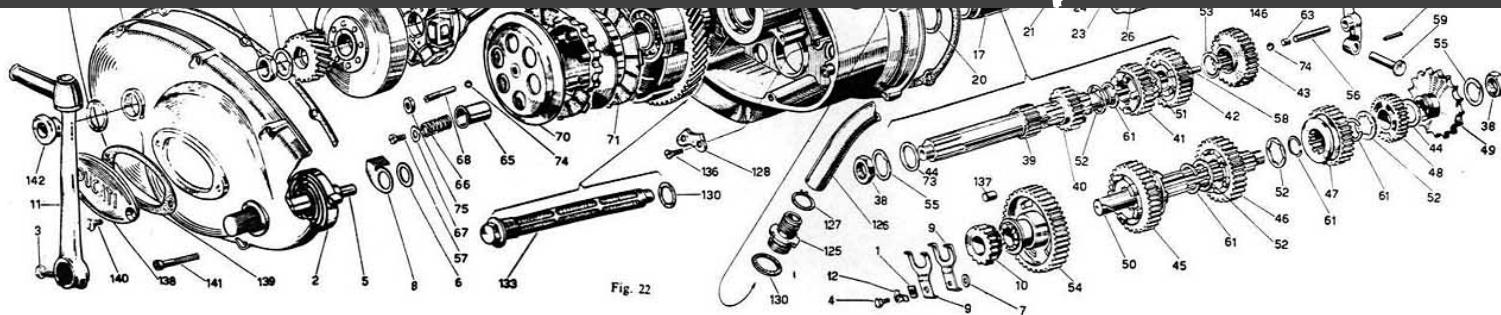
[Hashimoto, Alvarez-Melis & Jaakkola, 2016]

Can we use neural networks to understand, not just word similarities, but language meaning in general?



# Compositionality

Artificial Intelligence requires being able  
to understand bigger things from knowing  
about smaller parts



# We need more than word embeddings!

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

*The snowboarder is leaping over the mogul*

*A person on a snowboard jumps into the air*

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

# Beyond the bag of words: Sentiment detection

Is the tone of a piece of text positive, negative, or neutral?

- Sentiment is that sentiment is “easy”
- Detection accuracy for longer documents ~90%, BUT

... ... loved ... ... ... great ... ... ... ... ... impressed  
... ... ... ... ... marvelous ... ... ... ...

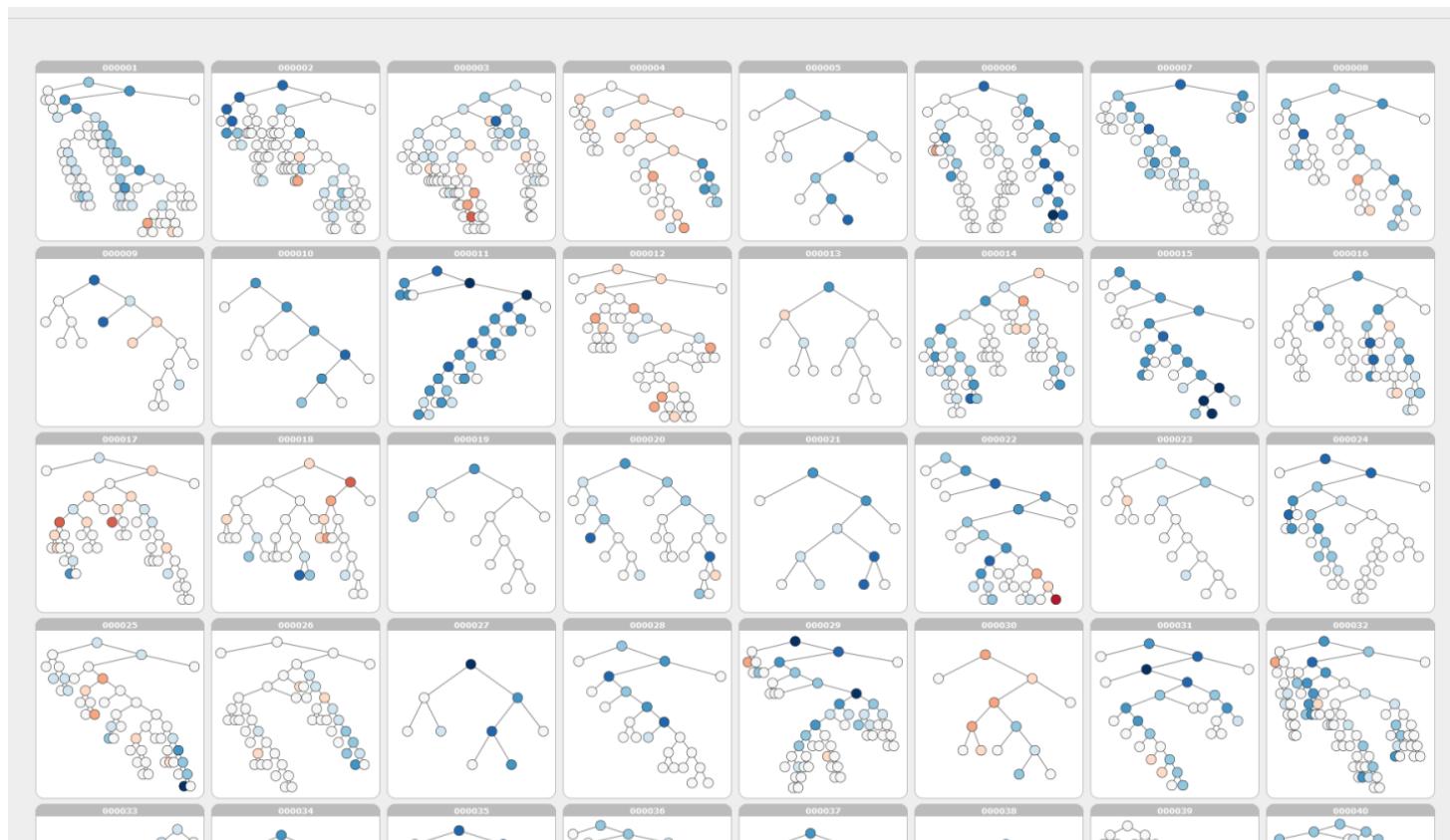


With this cast, and this subject matter, the movie should have been funnier and more entertaining.



# Stanford Sentiment Treebank

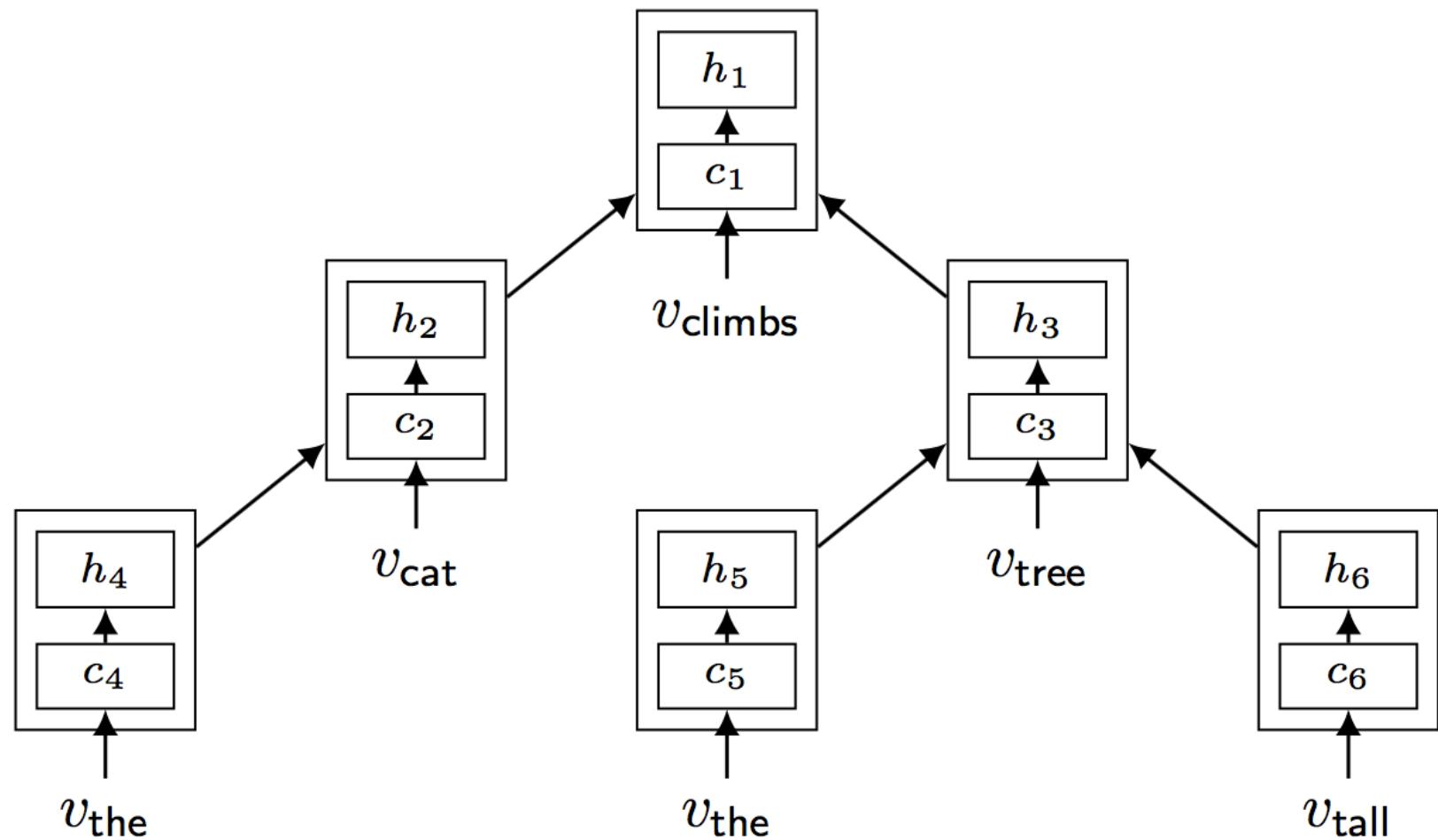
- 215,154 phrases labeled in 11,855 sentences
- Can train and test compositions



<http://nlp.stanford.edu:8080/sentiment/>

# Tree-Structured Long Short-Term Memory Networks

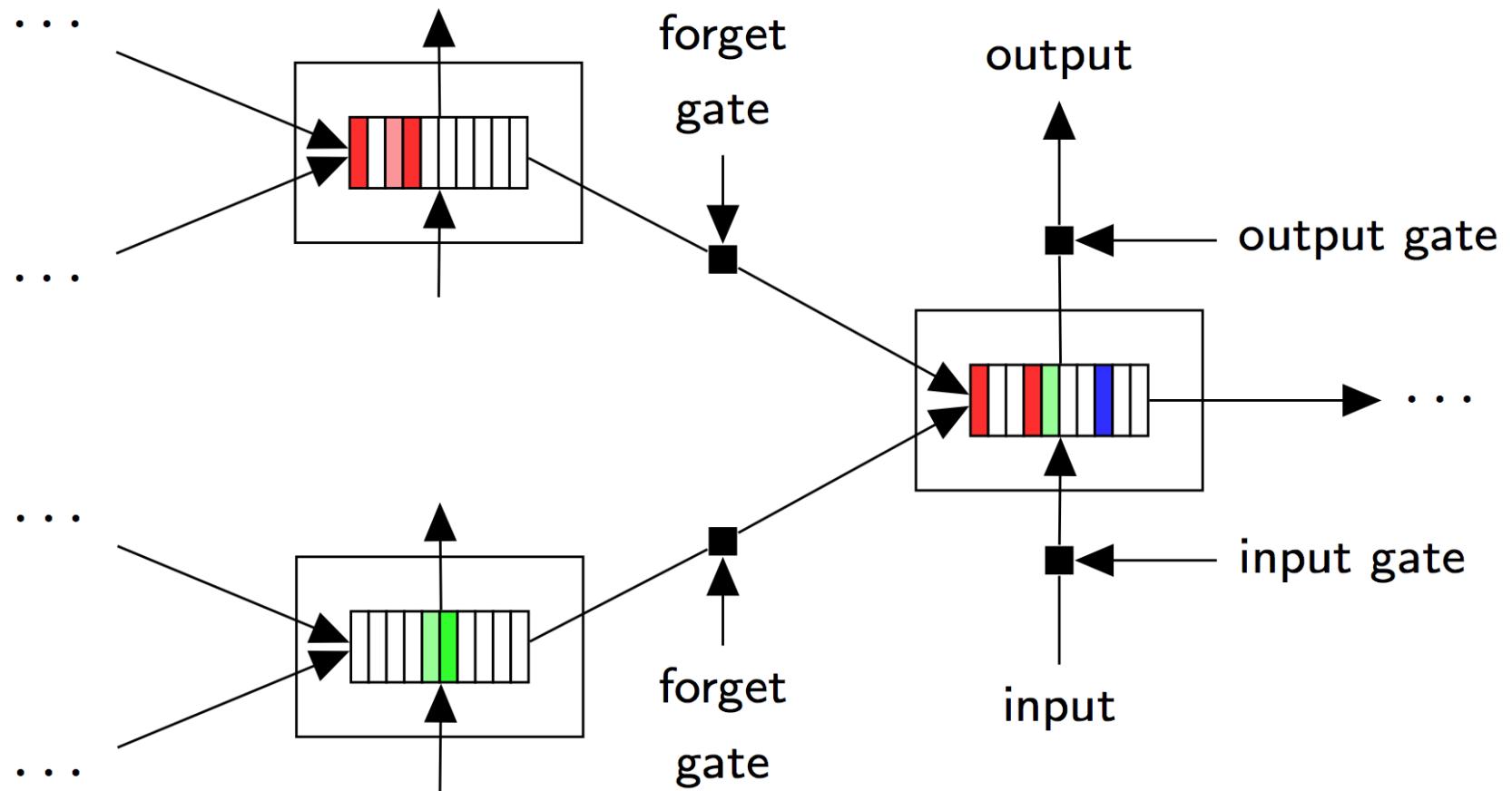
[Tai et al., ACL 2015]



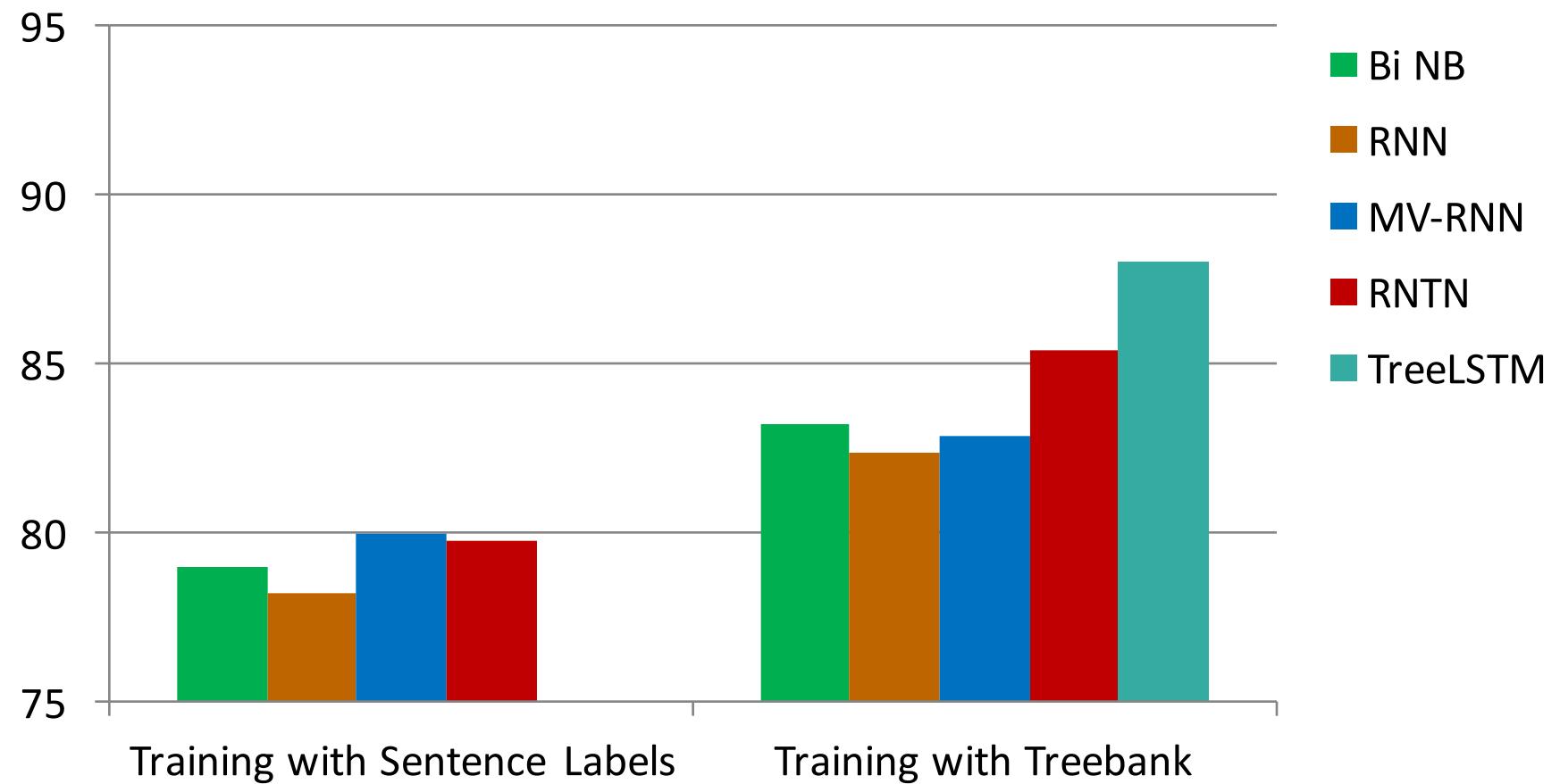
# Tree-structured LSTM



Generalizes sequential LSTM to trees with any branching factor

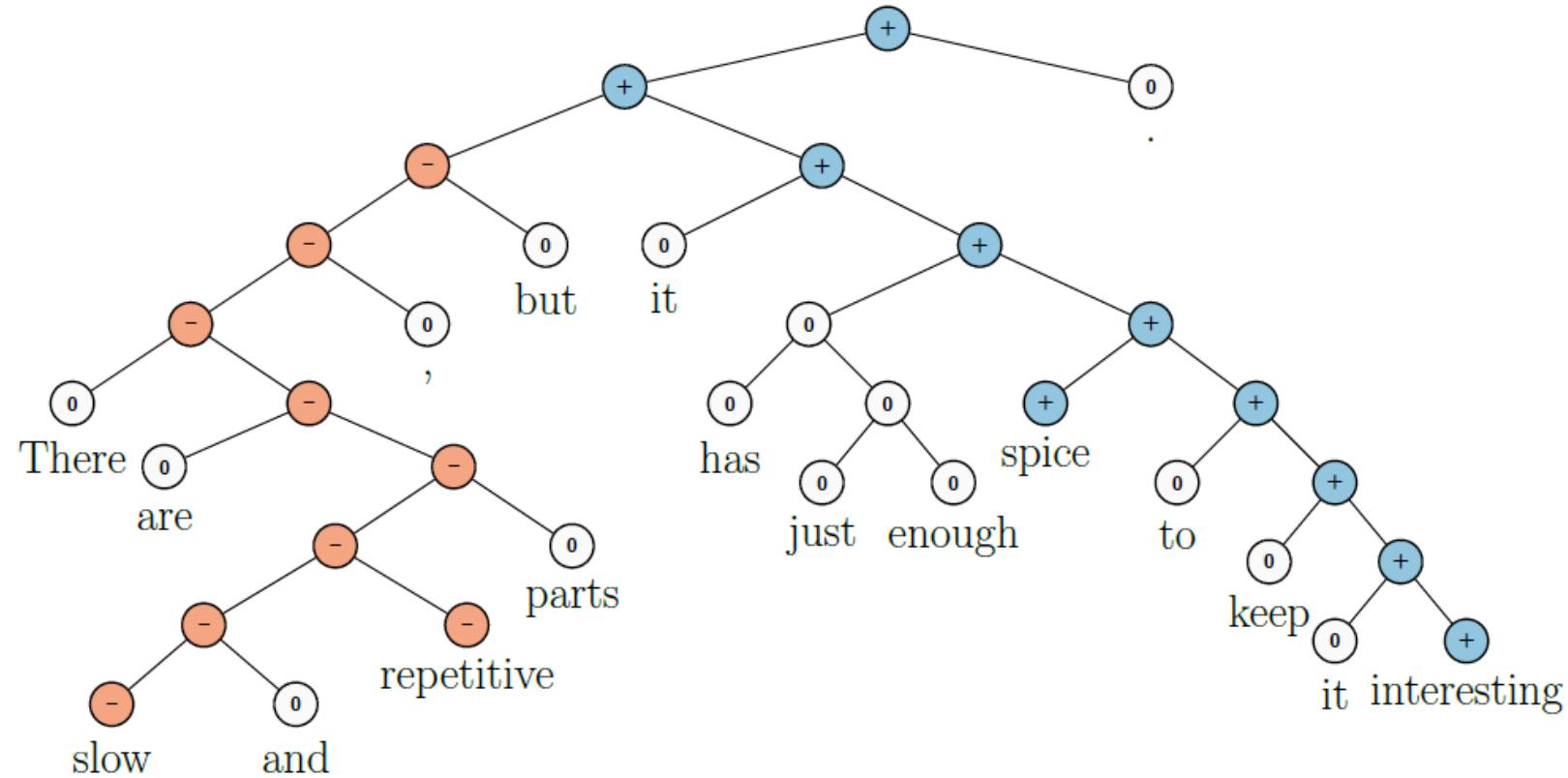


# Positive/Negative Results on Treebank



# Experimental Results on Treebank

- TreeRNN can capture constructions like  $X$  but  $Y$
- Biword Naïve Bayes is only 58% on these



# Stanford Natural Language Inference Corpus

<http://nlp.stanford.edu/projects/snli/>  
570K Turker-judged pairs, based on an assumed picture

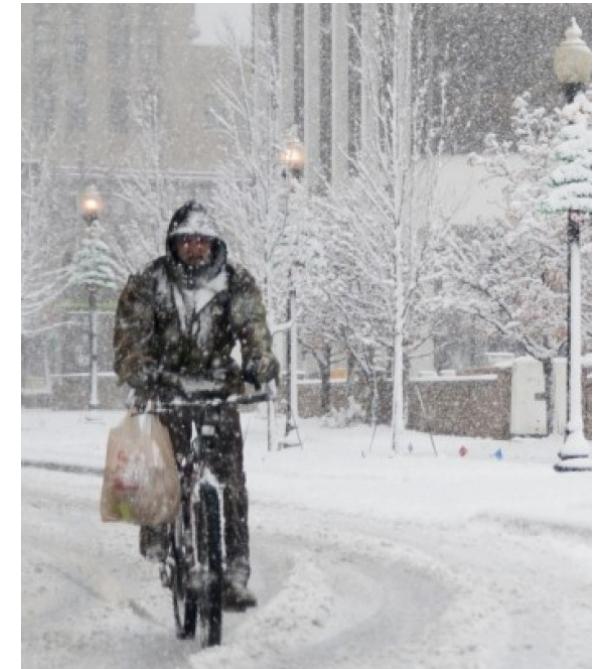
*A man rides a bike on a snow covered road.*

*A man is outside.*   ENTAILMENT

*2 female babies eating chips.*

*Two female babies are enjoying chips.*

NEUTRAL



*A man in an apron shopping at a market.*

*A man in an apron is preparing dinner.*

CONTRADICTION

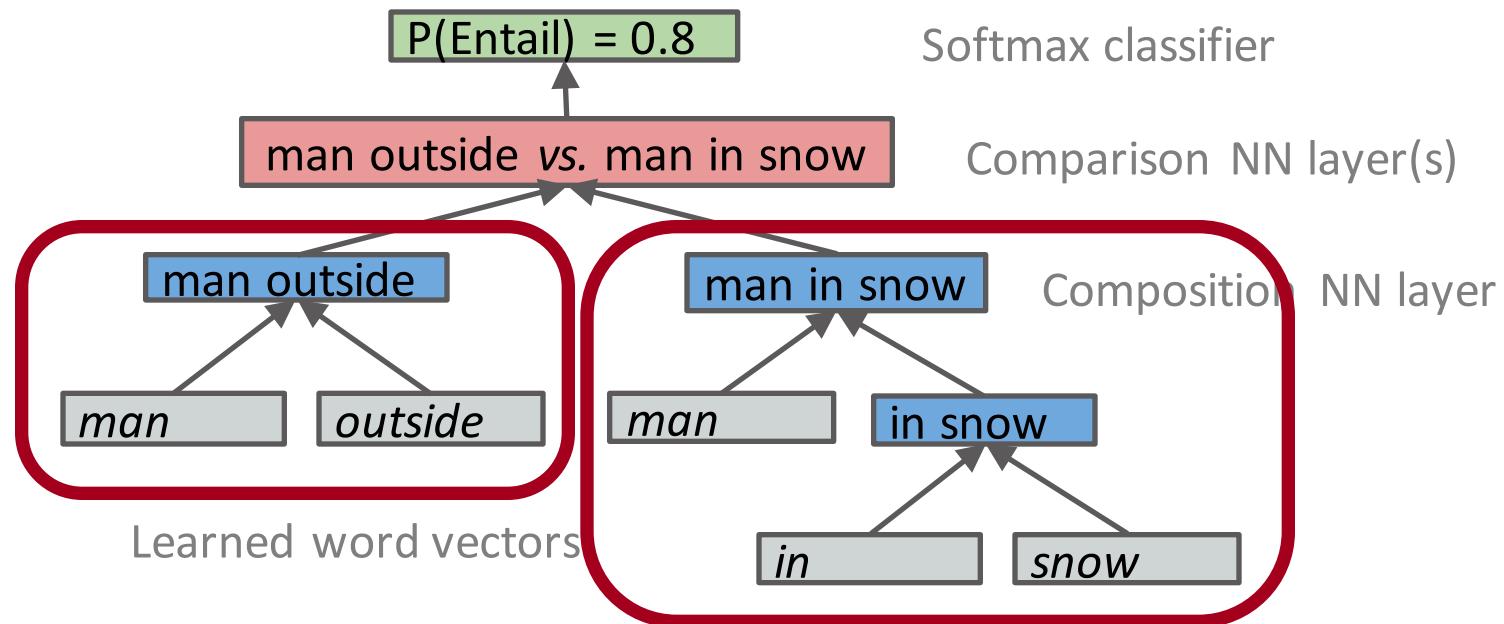


# NLI with Tree-RNNs

[Bowman, Angeli, Potts & Manning, EMNLP 2015]

Approach: We would like to work out the meaning of each sentence separately – a pure compositional model

Then we compare them with NN & classify for inference



# Tree recursive NNs (TreeRNNs)

Theoretically appealing

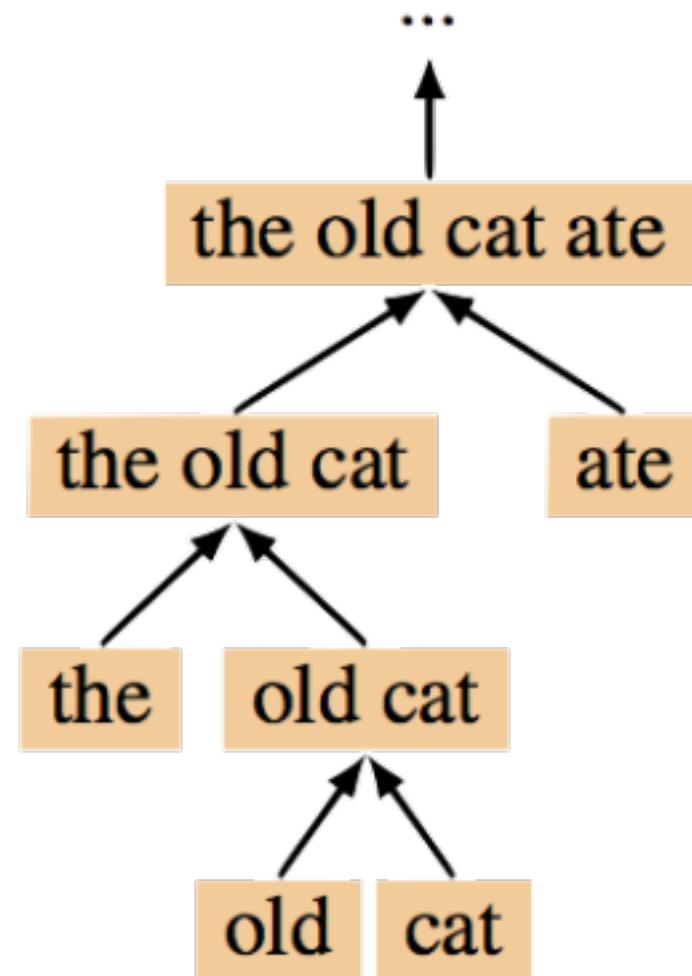
Very empirically competitive

But

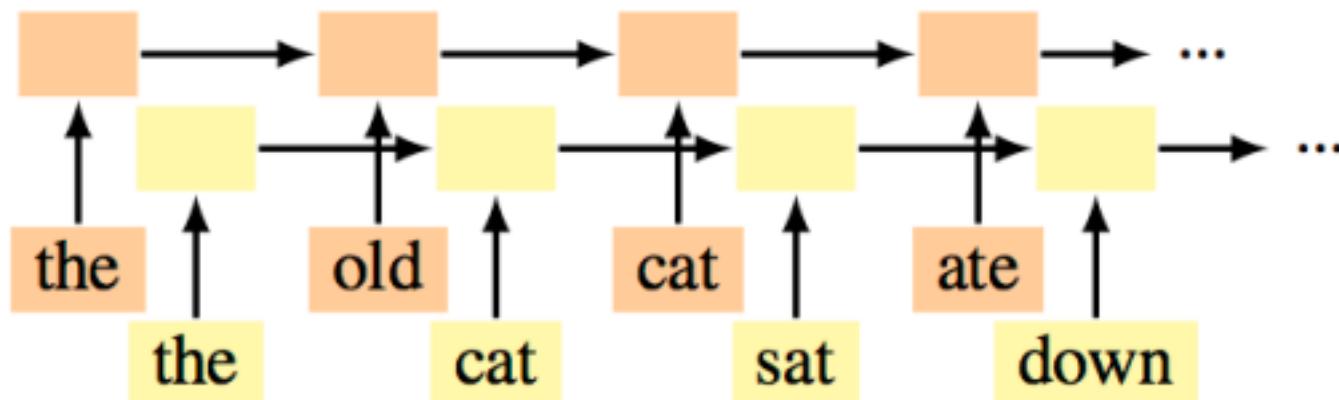
Prohibitively slow

Usually require an external  
parser

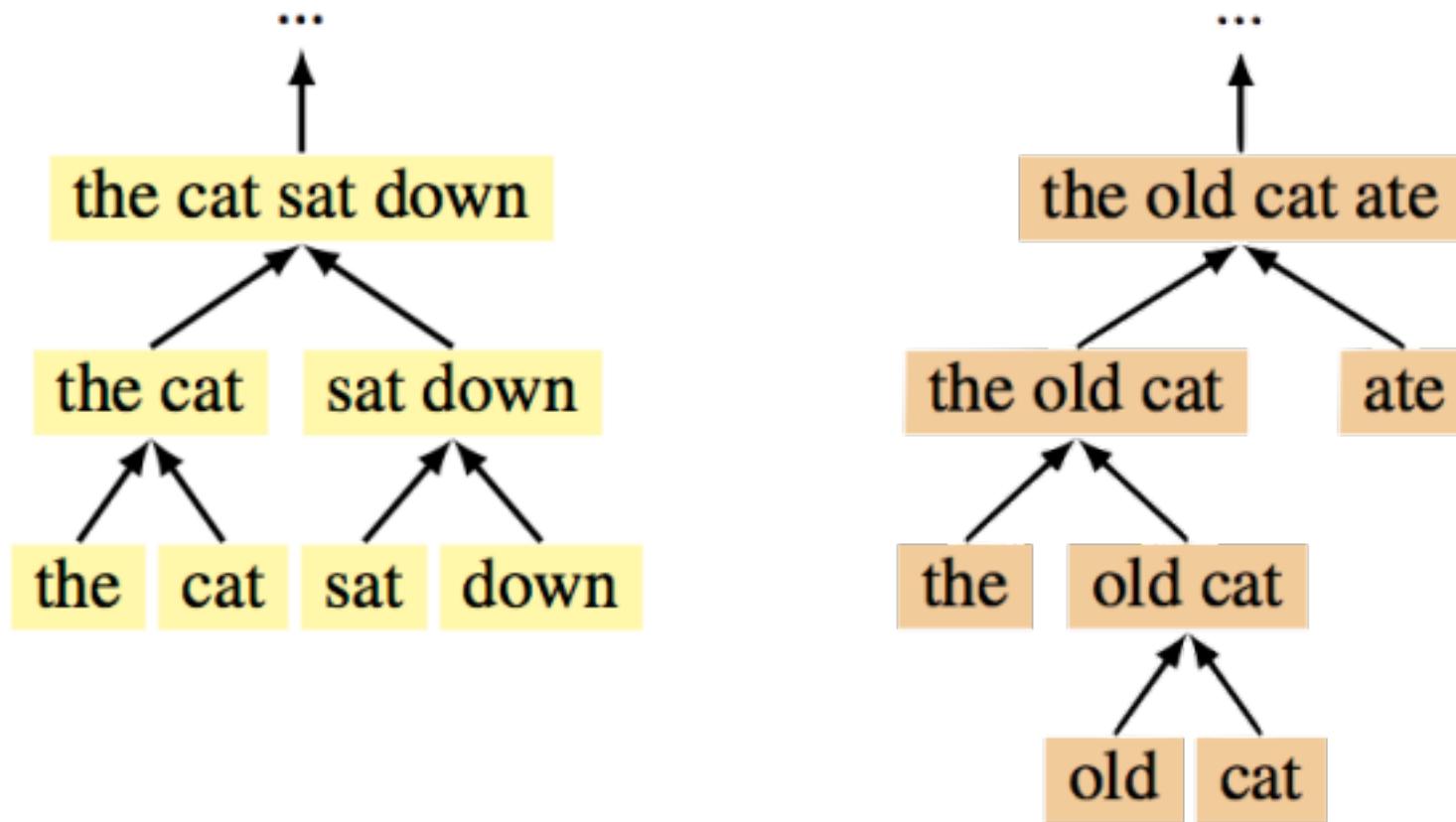
Don't exploit complementary  
linear structure of language



A recurrent NN allows efficient batched computation on GPUs



# TreERNN: Input-specific structure undermines batched computation



# The Shift-reduce Parser-Interpreter NN (SPINN) [Bowman, Gauthier et al. 2016]

Base model equivalent to a TreeRNN, but ...

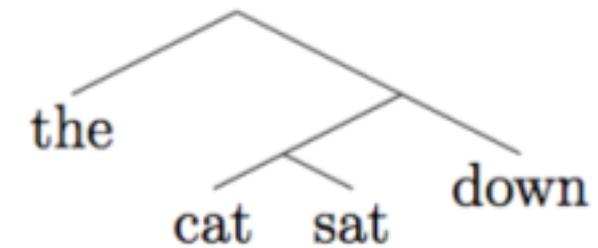
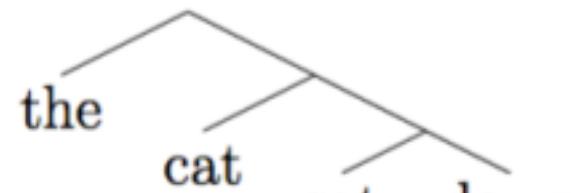
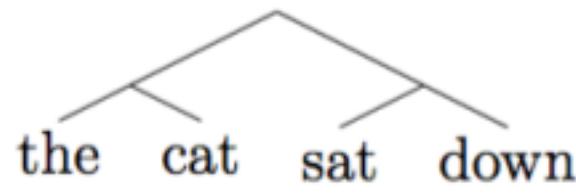
supports batched computation:  $25 \times$  speedups

Plus:

Effective new hybrid that combines linear and tree-structured context

Can stand alone without a parser

# Beginning observation: binary trees = transition sequences

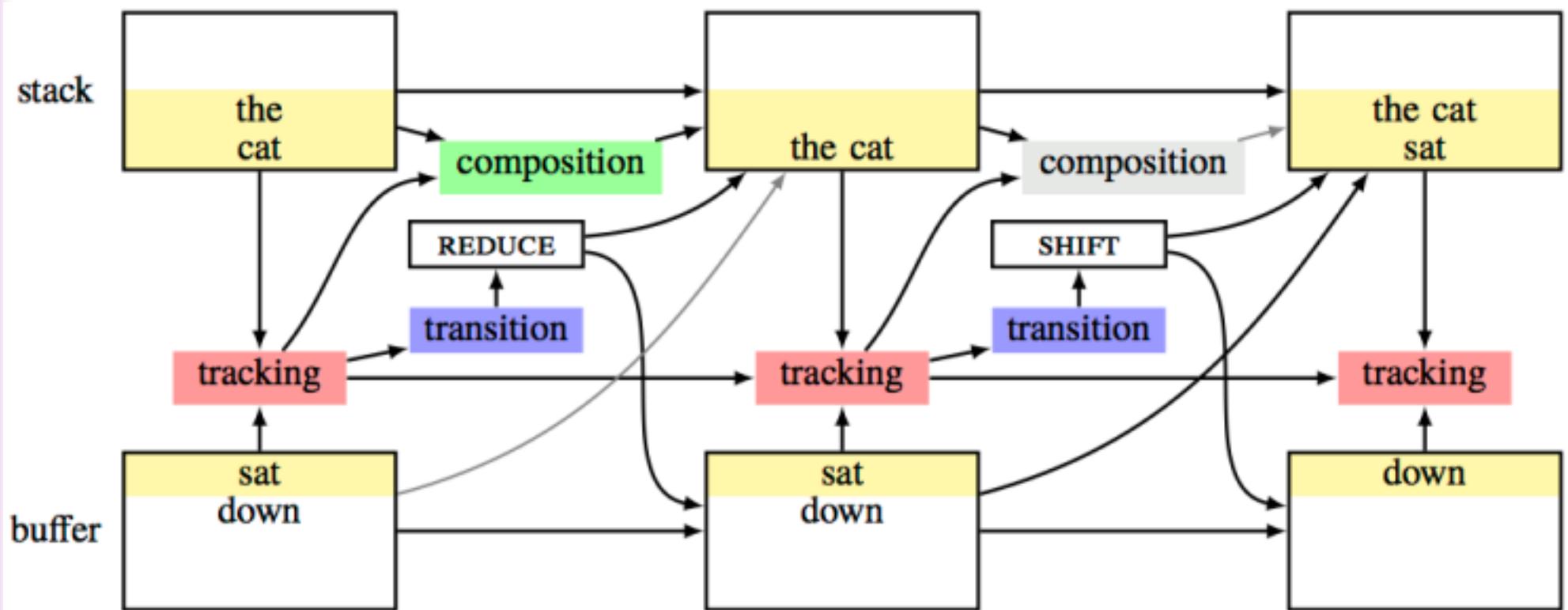


SHIFT SHIFT  
REDUCE SHIFT  
SHIFT REDUCE  
REDUCE

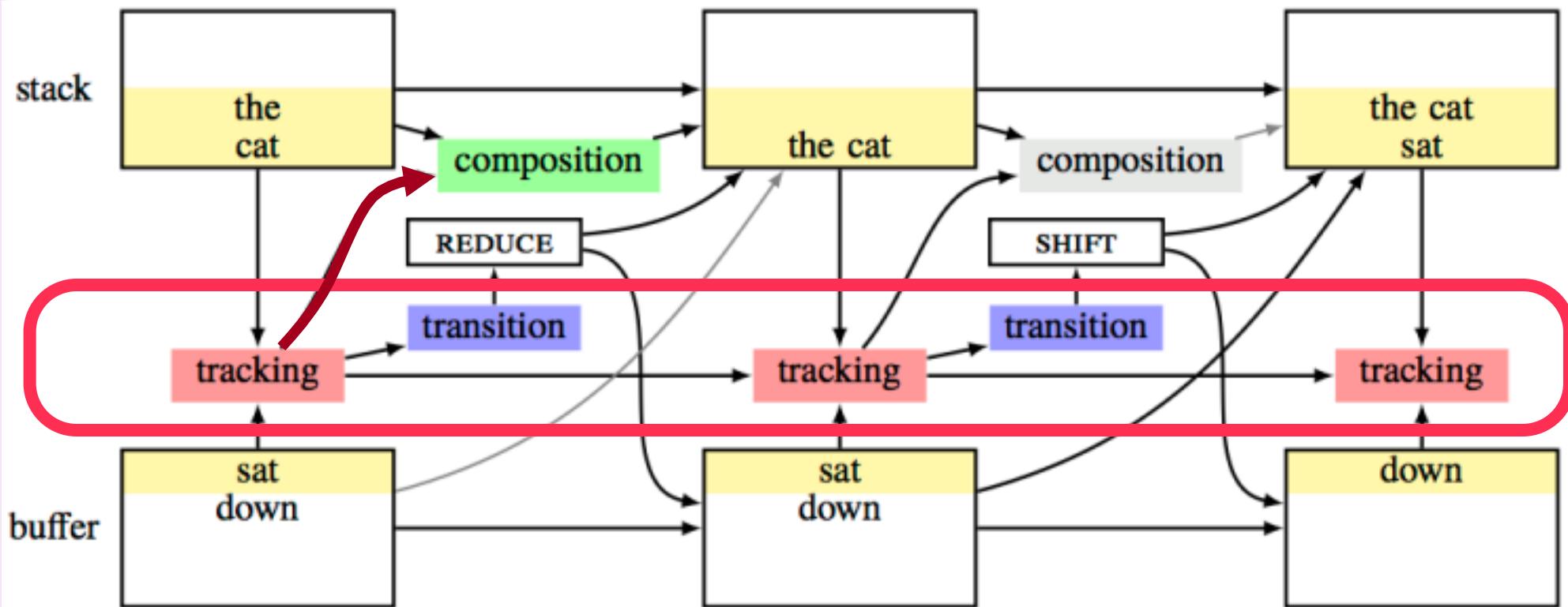
SHIFT SHIFT  
SHIFT SHIFT  
REDUCE REDUCE  
REDUCE

SHIFT SHIFT  
SHIFT REDUCE  
SHIFT REDUCE  
REDUCE

# The Shift-reduce Parser-Interpreter NN (SPINN)



# The Shift-reduce Parser-Interpreter NN (SPINN)



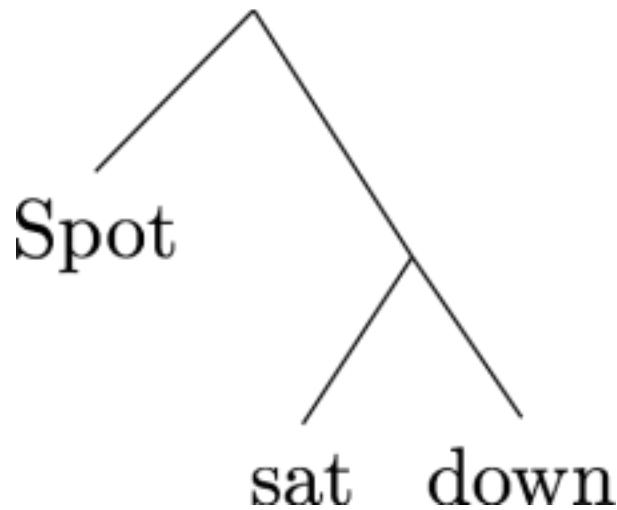
The model includes a sequence LSTM RNN

- This acts as a simple parser by predicting SHIFT or REDUCE
- It also gives left sequence context as input to composition

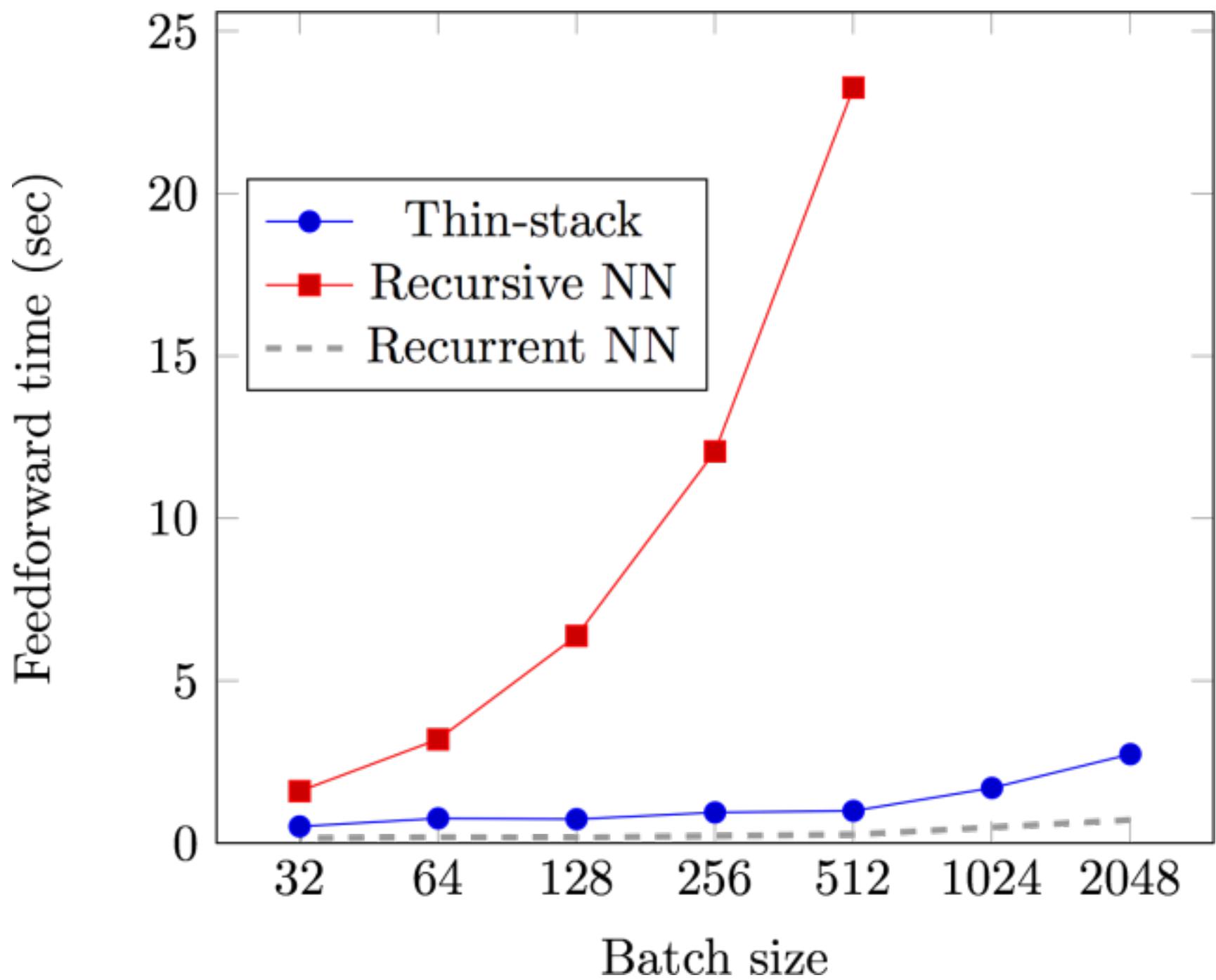
# Implementing the stack

- Naïve implementation: simulates stacks in a batch with a fixed-size multidimensional array at each timestep
  - Backpropagation requires that each intermediate stack be maintained in memory
  - ⇒ Large amount of data copying and movement required
- Efficient implementation
  - Have only one stack array for each example
  - At each timestep, augment with the current head of the stack
  - Keep list of backpointers for REDUCE operations
- Similar to zipper data structures employed elsewhere

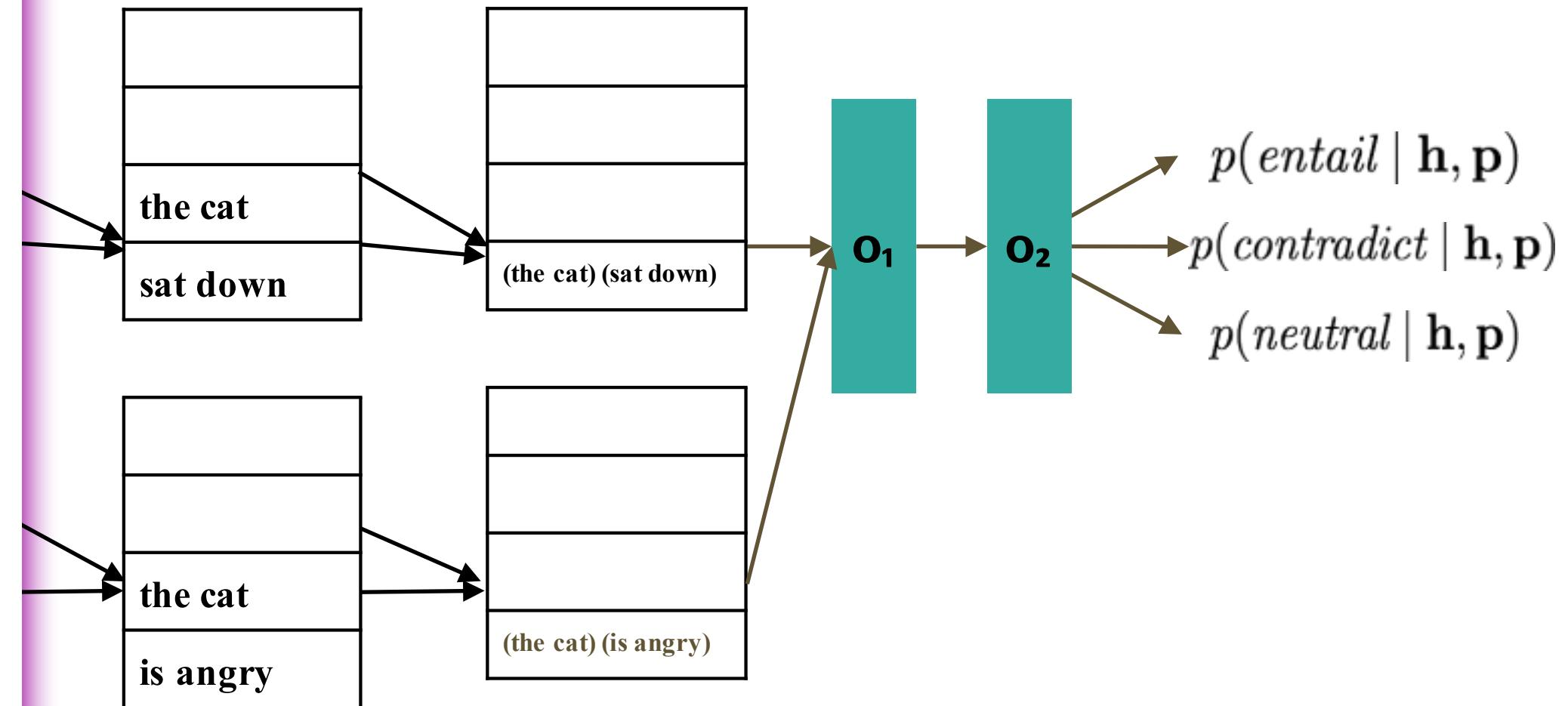
# A thinner stack



	<b>Array</b>	<b>Backpointers</b>
1	Spot	1
2	sat	1 2
3	down	1 2 3
4	(sat down)	1 4
5	(Spot (sat down))	5



# Using SPINN for natural language inference



# SNLI Results

Model	% Accuracy (Test set)
Feature-based classifier	78.2
Previous SOTA sentence encoder [Mou et al. 2016]	82.1
LSTM RNN sequence model	80.6
Tree LSTM	80.9
<b>SPINN</b>	<b>83.2</b>
SOTA (sentence pair alignment model) [Parikh et al. 2016]	86.8

# Successes for SPINN over LSTM

## Examples with negation

- P: The rhythmic gymnast completes her floor exercise at the competition.
- H: The gymnast cannot finish her exercise.

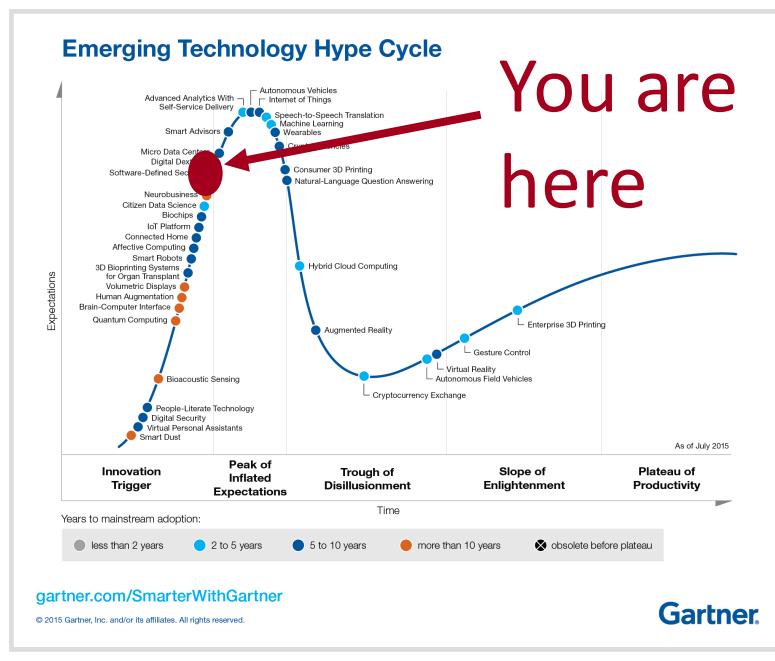
## Long examples (> 20 words)

- P: A man wearing glasses and a ragged costume is playing a Jaguar electric guitar and singing with the accompaniment of a drummer.
- H: A man with glasses and a disheveled outfit is playing a guitar and singing along with a drummer.

# Envoi

- There are very good reasons for wanting 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, but also:
  - Meanings of larger units, calculated compositionally
  - The ability to do natural language inference
- The SPINN model is fast — close to recurrent networks!
- Its hybrid sequence/tree structure is psychologically plausible and out-performs other sentence composition methods

# Final Thoughts



# Final Thoughts

I'm certain that deep learning will come to dominate SIGIR over the next couple of years ... just like speech, vision, and NLP before it. This is a good thing. Deep learning provides some powerful new techniques that are just being amazingly successful on many hard applied problems. However, we should realize that there is also currently a huge amount of hype about deep learning and artificial intelligence. We should not let a genuine enthusiasm for important and successful new techniques lead to irrational exuberance or a diminished appreciation of other approaches. Finally, despite the efforts of a number of people, in practice there has been a considerable division between the human language technology fields of IR, NLP, and speech. Partly this is due to organizational factors and partly that at one time the subfields each had a very different focus. However, recent changes in emphasis – with IR people wanting to understand the user better and NLP people much more interested in meaning and context – mean that there are a lot of common interests, and I would encourage much more collaboration between NLP and IR in the next decade.