Distributional Semantics

CS 585, Fall 2015

Introduction to Natural Language Processing

http://people.cs.umass.edu/~brenocon/inlp2015/

Brendan O’Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

[Many slides borrowed from David Belanger]

Question:

How can we use unsupervised data improve accuracy on a supervised task?

The Distributional Hypothesis

- "You shall know a word by the company it keeps." (Firth, 57)
- Words with similar roles in text have similar meanings.
- This is why unsupervised learning works in nlp.

COOCCURRENCE COUNT DATA

The “context” of a token

Target word: blue
Context words: red

She told the story, however, with great spirit among her friends; for she had a lively, playful disposition, which delighted in anything ridiculous.

(source: Pride and Prejudice)

The "context" of a token

Target word: blue
Context words: red

She told the story, however, with great spirit among her friends; for she had a lively, playful disposition, which delighted in anything ridiculous.

(source: Pride and Prejudice)

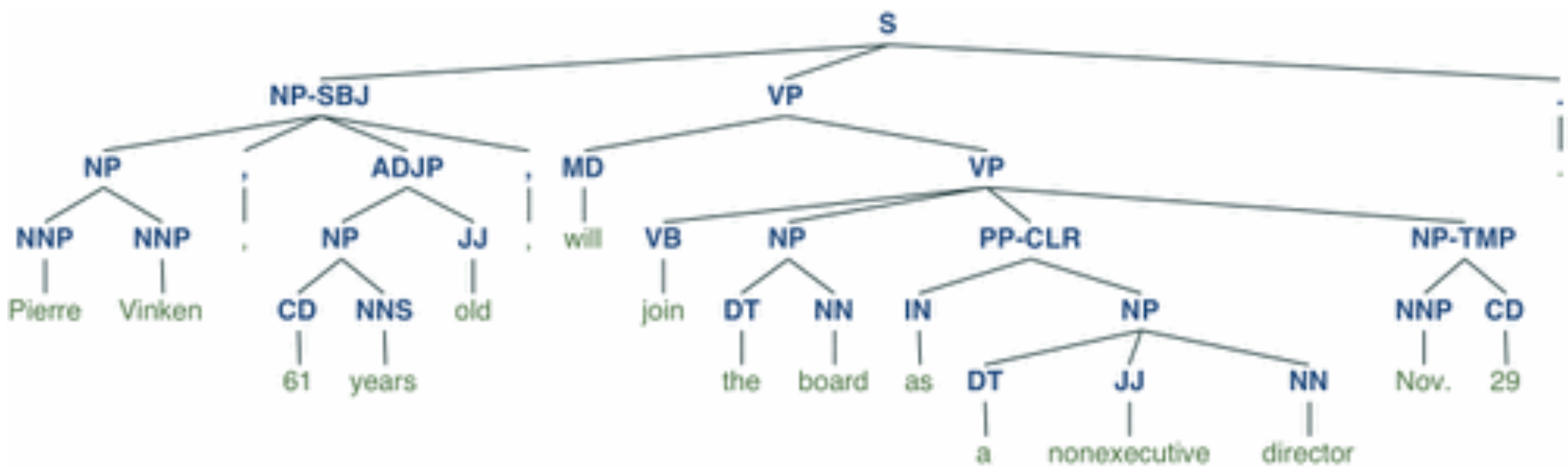
The “context” of a token

Target word: blue
Context words: red

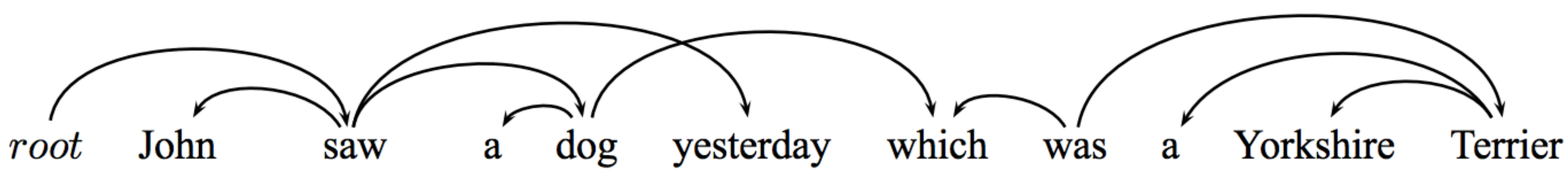
She told the story, however, with great spirit among her friends; for she had a lively, playful disposition, which delighted in anything ridiculous.

(source: Pride and Prejudice)

Contexts In Terms of Parses



(nltk.org)



(Ryan McDonald Thesis)

Context Types

Each possible context is a tuple.

- Trigram context: (the,dog)
- Unigram context: (the) or (dog)
- Parse context: (red_amod,ran_nsubj)

Context Count Vector

- Represent word type i, as a vector Vi

$$V_i = [0, 1, 0, 0, 0, 4, 0, 0, 0, 2, 0, 0, 1]$$

- Value in index k = #times context type k occurred.

Example

- Find contexts containing “art”

$$V_i = [0, 1, 0, 0, 0, 4, 0, 0, 0, 2, 0, 0, 1]$$

A collection of _

_ is a creation

structure of_

an exhibition of _

Vi is very long, but very sparse.

Question:

Example sentence:
The dog caught the frisbee.

What are 3 reasonable ways to define context, and what are the vectors for “caught” in each?

Question:

What do ‘art’ and ‘pharmaceuticals’ have in common?

What are contexts that they would both have?

What are contexts that they wouldn’t share?

Comparing Context Vectors

common contexts for "art" but not "pharmaceuticals" [7394 total]	common contexts for both "art" and "pharmaceuticals" [165 total]	common contexts for "pharmaceuticals" but not "art" [206 total]
'm into _	areas such as _	a greater amount of _
's interested in _	prices of _	standards for _
A collection of _	storage of _	marketer of _
_ has been described by	producers of _	market for _
structure of _	_ designed for	prescriptions for _
study in _	the provision of _	the supply of _
_ have been shown in	_ sold in	the availability of _
The knowledge of _	the same way as _	advertising for _
_ is a commodity	_ are among	the appropriate use of _
_ is a creation	The production of _	shipment of _
_ is a world	the analysis of _	a cocktail of _
an exhibition of _	advances in _	classes of _
the commercialization of _	specialising in _	a complete inventory of _
the confinement of _	a career in _	_ related downloads
_ is cast in	_ stolen from	new generations of _

Comparing Vectors

$$D_{\text{Euclidean}}(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

$$D_{\text{Manhattan}}(x, y) = \sum_i |x_i - y_i|$$

Dot Product: $x^\top y = \sum_i x_i y_i$

$$Cos(x, y) = \frac{x^\top y}{\sqrt{x^\top x}\sqrt{y^\top y}}$$

Cosine similarity:
very commonly used

Vector-Space Interpretation of Distributional Hypothesis

Two words are similar if their context vectors are similar.

Question:

What does it mean for two words to be similar?

Are “dog” and “tiger” similar?
How about “dog” and “fetch?”

Question:

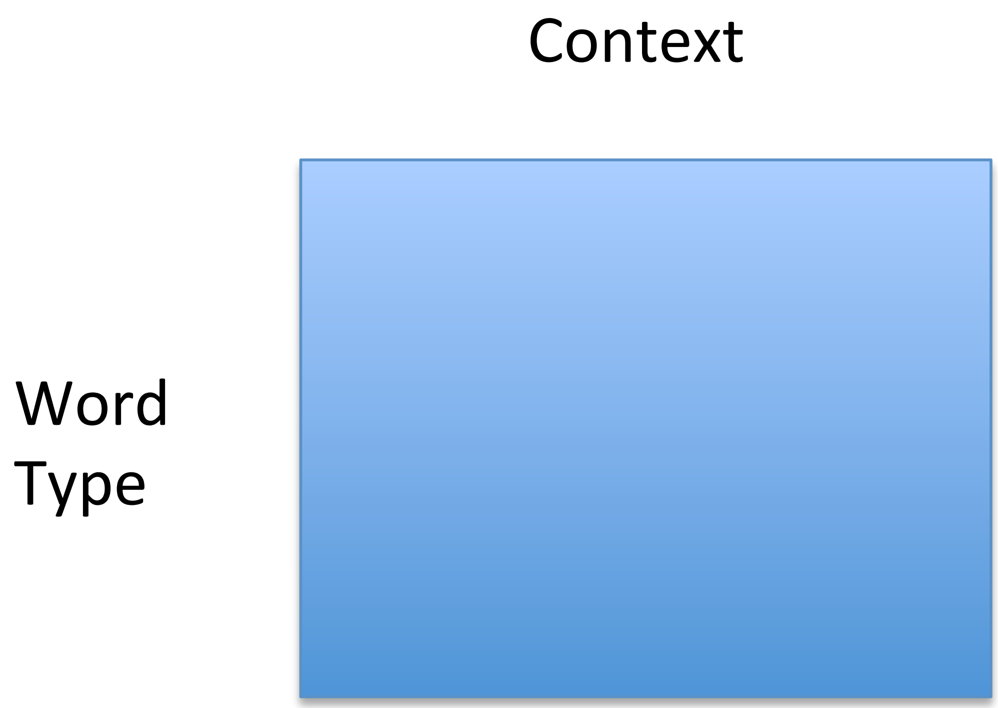
What are the pros and cons of using a wide window for a token's context?

Hint: Syntax v.s. Topics.

Question:

We now have a function sim(word1,word2). How could we use this to improve accuracy in the tasks we've discussed in class?

Word-Context Matrix

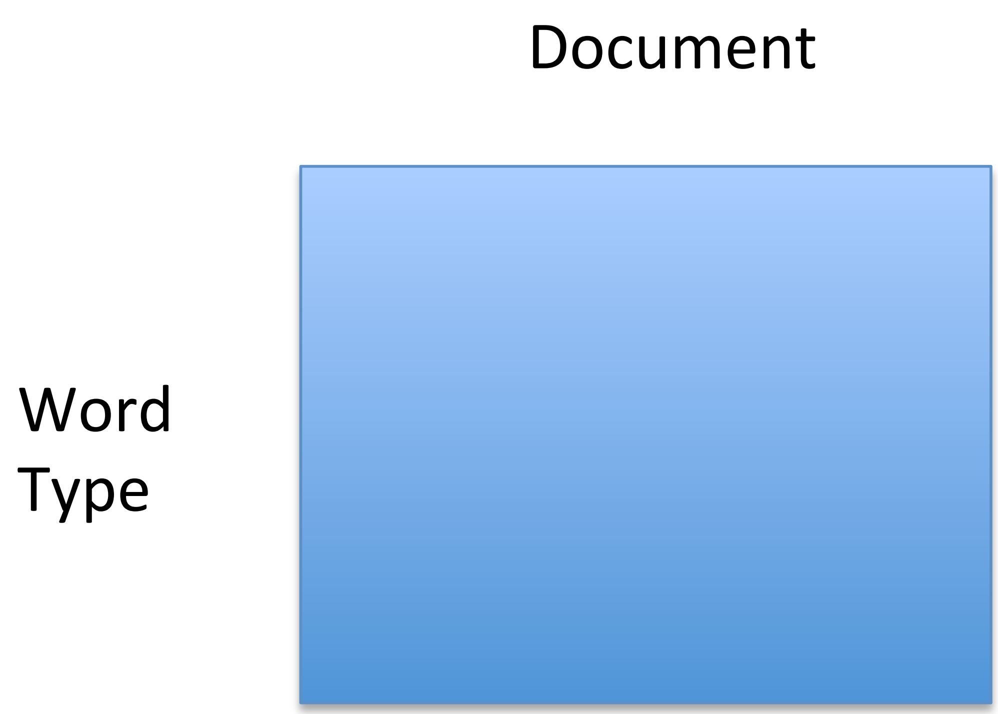


Distributional hypothesis:

– A word is characterized by its row in this matrix.

– Similar words have similar rows

Topic Model



A document is characterized by the distribution of words in it.

Documents are similar if their columns are similar.

LDA Topic Model: this distribution is a mixture of ‘topics’

WORD EMBEDDINGS

Word Embeddings

Sparse Context Vector (10 million+ dimensional):

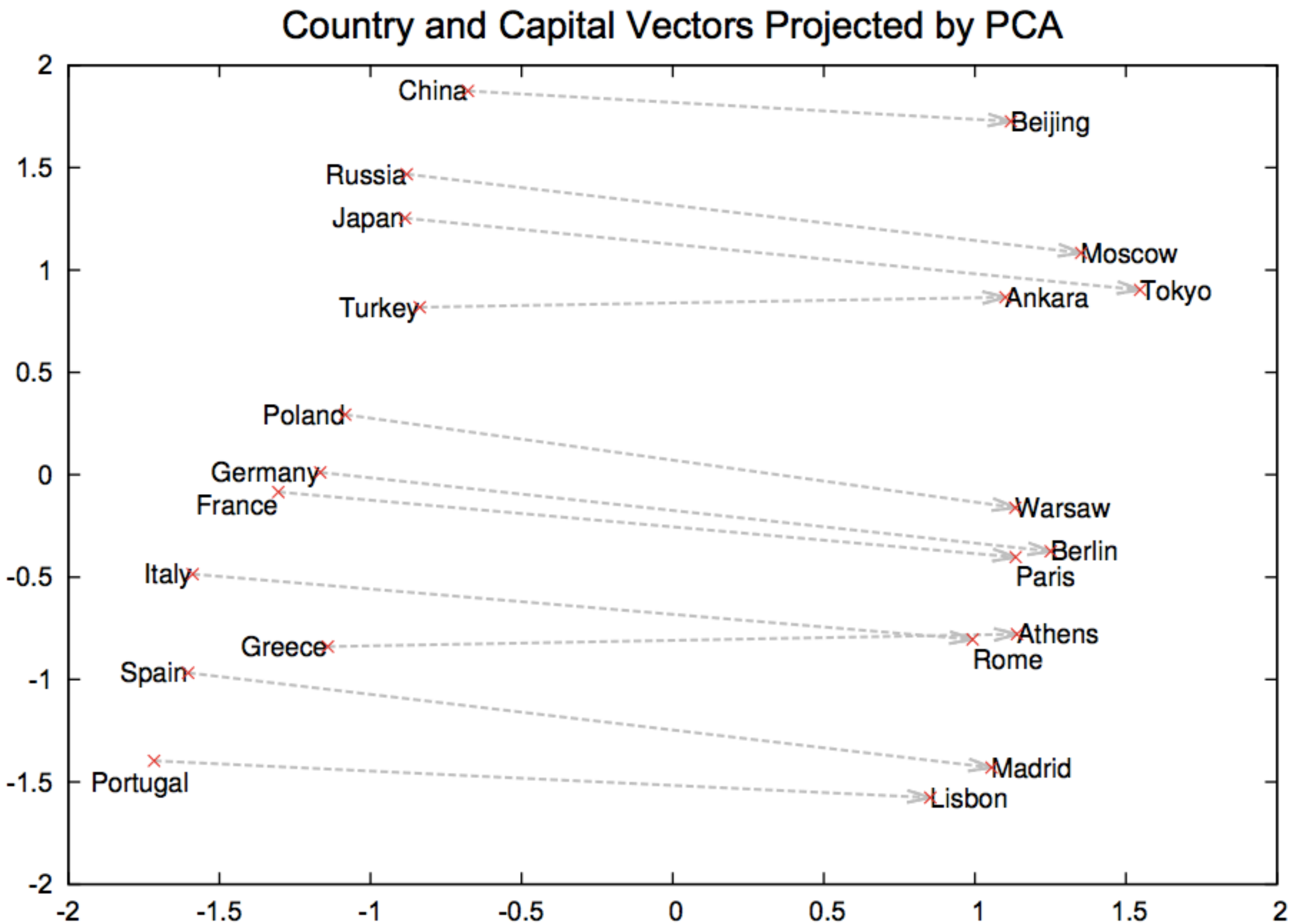
$$V_i = \boxed{[0, 1, 0, 0, 0, 4, 0, 0, 0, 2, 0, 0, 1, \ldots]}$$

[This can be directly used, but maybe too slow, sparse]

Instead represent every word type as a low-dimensional dense vector (about 100 dimensional).

$$E_i = \boxed{[.253, 458, 4.56, 78.5, 120, \ldots]}$$

These don't come directly from the data. They need to be learned.



Country and Capital Vectors Projected by PCA
China
Beijing
Russia
Japan
Moscow
Tokyo
Turkey
Ankara
Poland
Germany
France
Warsaw
Berlin
Paris
Italy
Greece
Athens
Rome
Spain
Madrid
Portugal
Lisbon
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Nearest Neighbors

- deals --> checks approvals vents stickers cuts
- warned --> suggested speculated predicted stressed argued
- ability --> willingness inability eagerness disinclination desire
- dark --> comfy wild austere cold tinny
- possibility --> possiblity possibilty dangers notion likelihood

Nearest Neighbors

- deals --> checks approvals vents stickers cuts
- warned --> suggested speculated predicted stressed argued
- ability --> willingness inability eagerness disinclination desire
- dark --> comfy wild austere cold tinny
- possibility --> possiblity possibilty dangers notion likelihood

Question:

What are the pros and cons of representing word types with such small vectors?

Answer:

Pro:

It requires less annotated data to train an ML model on low dimensional features.

Con:

You can't capture all of the subtlety of language in 100 dimensions. (...can you?)

Learning Embeddings by Preserving Similarity

- Given long, sparse context cooccurrence vectors V_i and V_j

- Goal: Choose Embeddings E_i and E_j such that similarity is approximately preserved

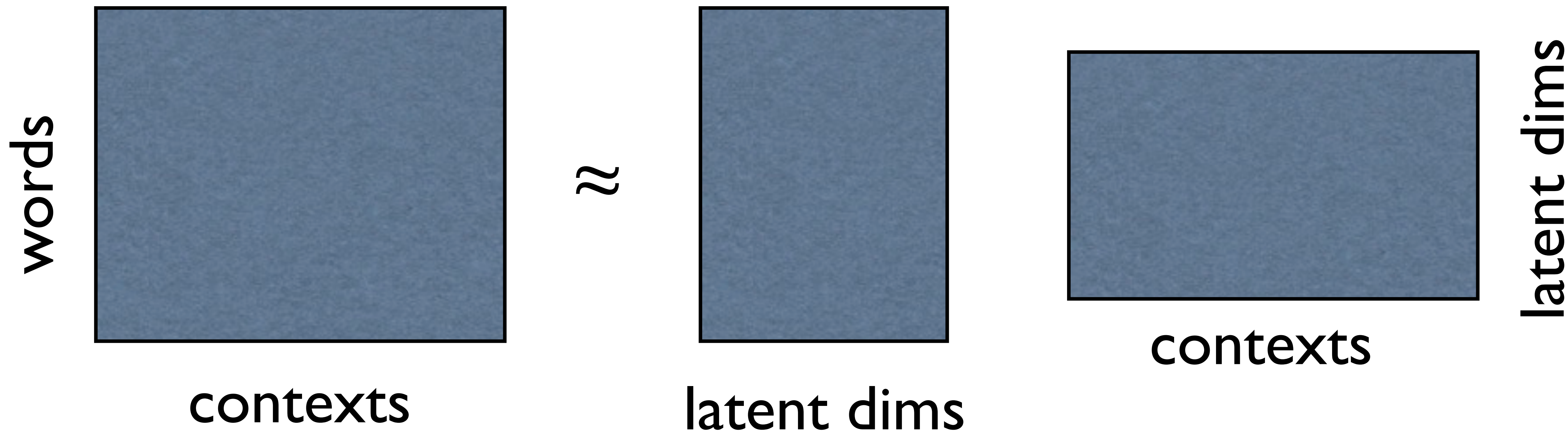
$$V_i^\top V_j \approx E_i^\top E_j$$

- Difficulty: need to do this for all words jointly.
- Solution: Use an eigen-decomposition (implemented in every language).

Matrix factorization

V ≈ E B

(counts) (embeddings)



Reconstruct the co-occurrence matrix

$$V_{i,c} \approx \sum_k E_{i,k} B_{k,c}$$

Singular Value Decomposition learns E,B
(or other matrix factorization techniques)

↔

Preserve pairwise distances between words i, j

$$V_i^\mathsf{T} V_j \approx E_i^\mathsf{T} E_j$$

Eigen Decomposition learns E

Word clustering

- Alternative word representation: associate words with (hierarchical) clusters, as opposed to embeddings (real-valued vectors)
- "Brown clustering":
  Unsupervised HMM with hierarchical clustering
    - Word belongs to only one class
      (bad assumption, but better than alternative; *Blunsom et al. 2011*)
    - Iteratively merge words with high contextual similarity

Word clusters as features

- Labeled data is small and sparse. Lexical generalization via induced word classes.
  - Both embeddings and clusters can be used as features!
- Examples from Twitter, for POS tagging
- Big Data vs. I Make My Own Data
  - Unlabeled: 56 M tweets, 847 M tokens
  - Labeled: 2374 tweets, 34k tokens
- 1000 clusters over 217k word types
  - Preprocessing: discard words that occur < 40 times

http://www.ark.cs.cmu.edu/TweetNLP/cluster_viewer.html

What does it learn?

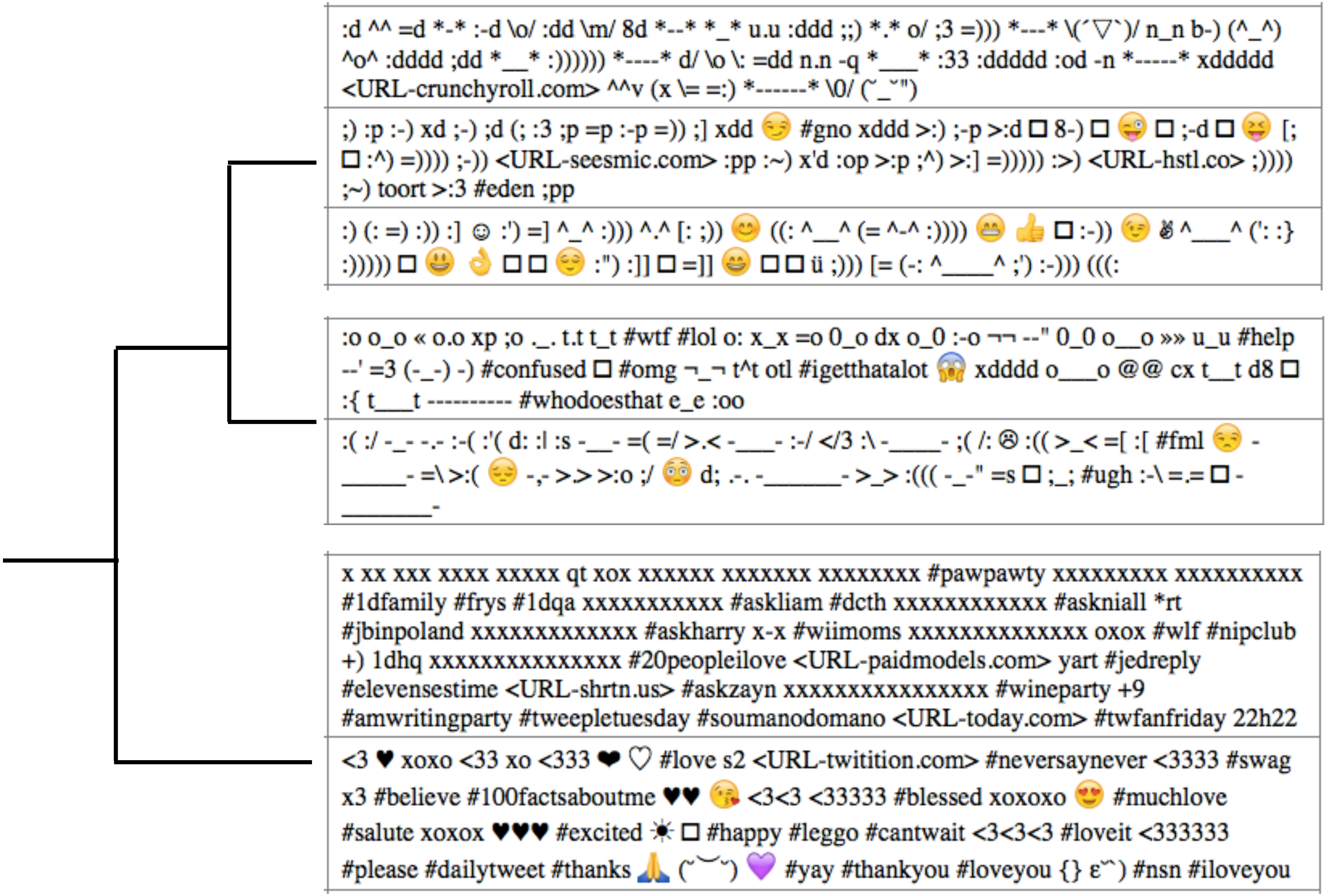
- Orthographic normalizations

so s0 -so so- $o /so //so

soo sooo soooo sooooo soooooo sooooooo soooooooo sooooooooo soooooooooo sooooooooooo soooooooooooo sooooooooooooo soso soooooooooooooo sooooooooooooooo soooooooooooooooo sososo superrr sooooooooooooooooo ssooo so0o superrrr so0 soooooooooooooooooo sosososo sooooooooooooooooooo ssoo sssooo soooooooooooooooooooo #too s0o ssoooo s00 sooooooooooooooooooooo so0o0o sososososo soooooooooooooooooooooo sssoooo ssooooo superrrrr very2 s000 sooooooooooooooooooooooo soooooooooooooooooooooooo sooooooooooooooooooooooooo _so_ soooooooooooooooooooooooooo /so/ sssooooo sosososososo

- suggests joint model for morphology/spelling

- Emoticons etc.
(Clusters/tagger useful for sentiment analysis: NRC-Canada SemEval 2013, 2014)



(Immediate?) future auxiliaries

gonna gunna gona gna guna gnna ganna qonna gonnna gana qunna gonne goona gonnaa g0nna goina gonnah goingto gunnah gonaa gonan gunnna going2 gonnnna gunnaa gonny gunaa quna goonna qona gonns goinna gonnae qnna gonnaaa gnaa

tryna gon finna bouta trynna boutta gne fina gonn tryina fenna qone trynaa qon boutaa funna finnah bouda boutah abouta fena bouttah boudda trinna qne finnaa fitna aboutta goin2 bout2 finnna trynah finaa ginna bouttaa fna try'na g0n trynn tyrna trna bouto finsta fnna tranna finta tryinna finnuh tryingto boutto

- finna ~ "fixing to"
- tryna ~ "trying to"
- bouta ~ "about to"

Subject-AuxVerb constructs

i'd you'd we'd he'd they'd she'd who'd i’d u'd youd you’d iwould theyd icould we’d i`d #whydopeople he’d i´d #iusedto they’d i'ld she’d #iwantsomeonewhowill i'de imust a:i'd you`d yu'd icud l'd

[Contraction splitting?]

ill ima imma i'ma i'mma ican iwanna umma imaa #imthetypeto iwill amma #menshouldnever igotta #whywouldyou #iwishicould #sometimesyouhaveto #thoushallnot #ihatewhenpeople illl #thingspeopleshouldnotdo #howdareyou #thingsgirlswantboystodo im'a #womenshouldnever #thingsblackgirlsdo immma iima #ireallyhatewhenpeople ishould #thingspeopleshouldntdo #irefuseto itl #howtospoilahoodrat iwont imight #thingsweusedtodoaskids ineeda #thingswhitepeopledo we'l #whycantyoujust #whydogirls #everymanshouldknowhowto #ushouldnt #howtopissyourgirloff #amanshouldnot #uwannaimpressme #realfriendsdont immaa #ilovewhenyou

[Mixed]

you'll we'll it'll he'll they'll she'll it'd that'll u'll that'd youll ull you’ll itll there'll we’ll itd there'd theyll this'll thatd thatll they’ll didja he’ll it’ll yu'll she’ll youl you`ll you'l you´ll yull u'l it'l we´ll we`ll didya that’ll it’d he'l shit'll they'l theyl she'l everything'll he`ll things'll u’ll this'd

i'll i’ll i'l i`ll i´ll i'lll l'll i\'ll i"ll -i'll /must @pretweeting she`ll

Word clusters as features

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

w fo fa fr fro ov fer **fir** whit abou aft serie fore fah fuh w/her w/that fron isn agains

"non-standard prepositions"

yeah yea nah naw yeahh nooo yeh noo noooo yeaa **ikr** nvm yeahhh nahh nooooo

"interjections"

facebook **fb** itunes myspace skype ebay tumblr bbm flickr aim msn netflix pandora

"online service names"

smh jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying

"hashtag-y interjections"??

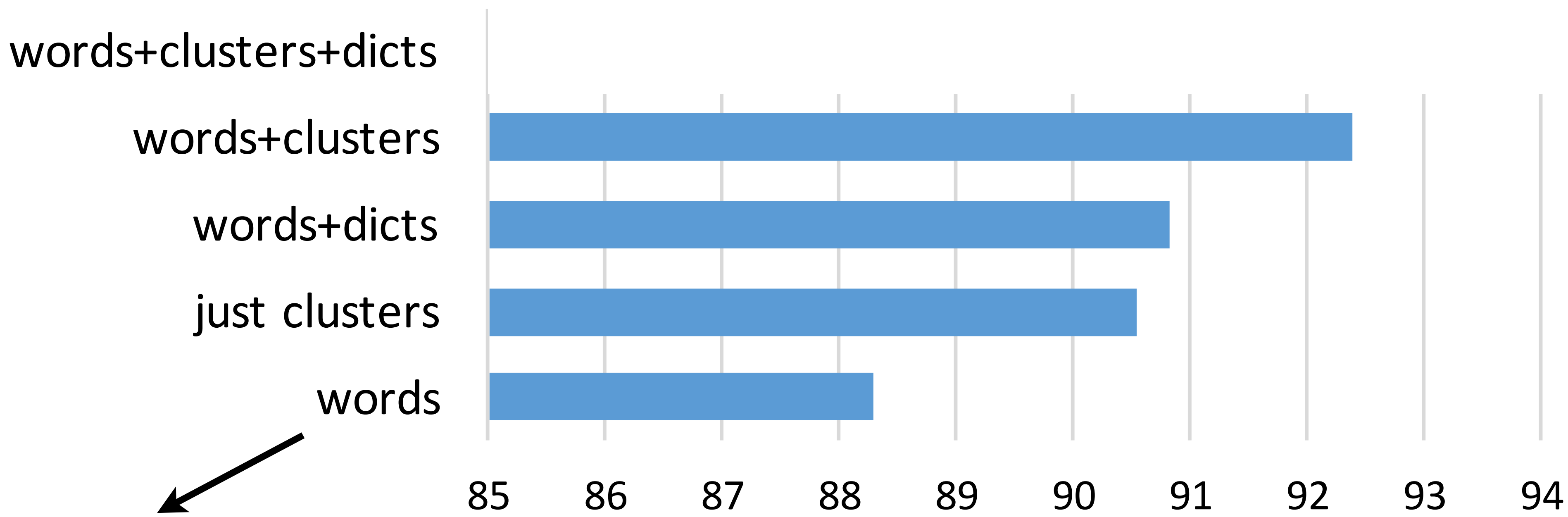
Highest-weighted POS–treenode features

hierarchical structure generalizes nicely.

Cluster prefix	Tag	Types	Most common word in each cluster with prefix
11101010*	!	8160	lol lmao haha yes yea oh omg aww ah btw wow thanks sorry congrats welcome yay ha hey goodnight hi dear please huh wtf exactly idk bless whatever well ok
11000*	L	428	i'm im you're we're he's there's its it's
1110101100*	E	2798	x <3 :d :p :) :o :/
111110*	A	6510	young sexy hot slow dark low interesting easy important safe perfect special different random short quick bad crazy serious stupid weird lucky sad
1101*	D	378	the da my your ur our their his
01*	V	29267	do did kno know care mean hurts hurt say realize believe worry understand forget agree remember love miss hate think thought knew hope wish guess bet have
11101*	O	899	you yall u it mine everything nothing something anyone someone everyone nobody
100110*	&	103	or n & and

Clusters help POS tagging

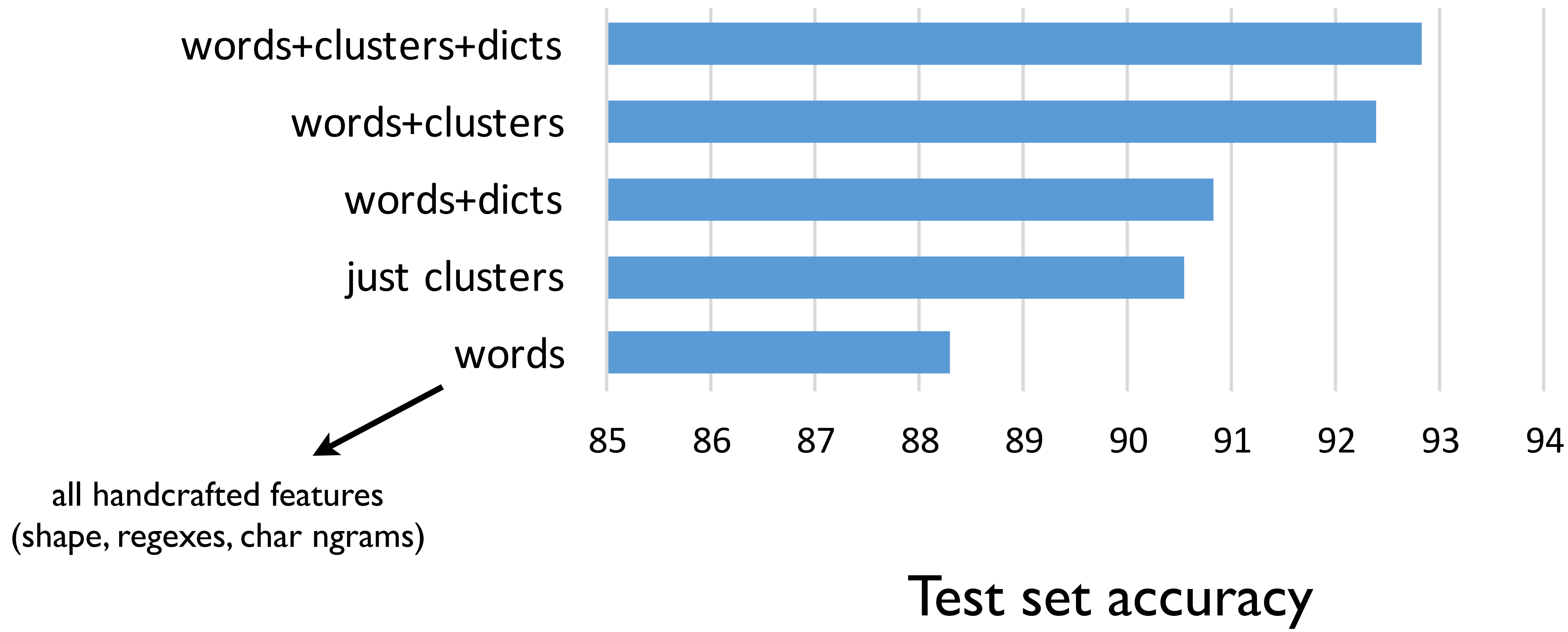
Clusters help POS tagging



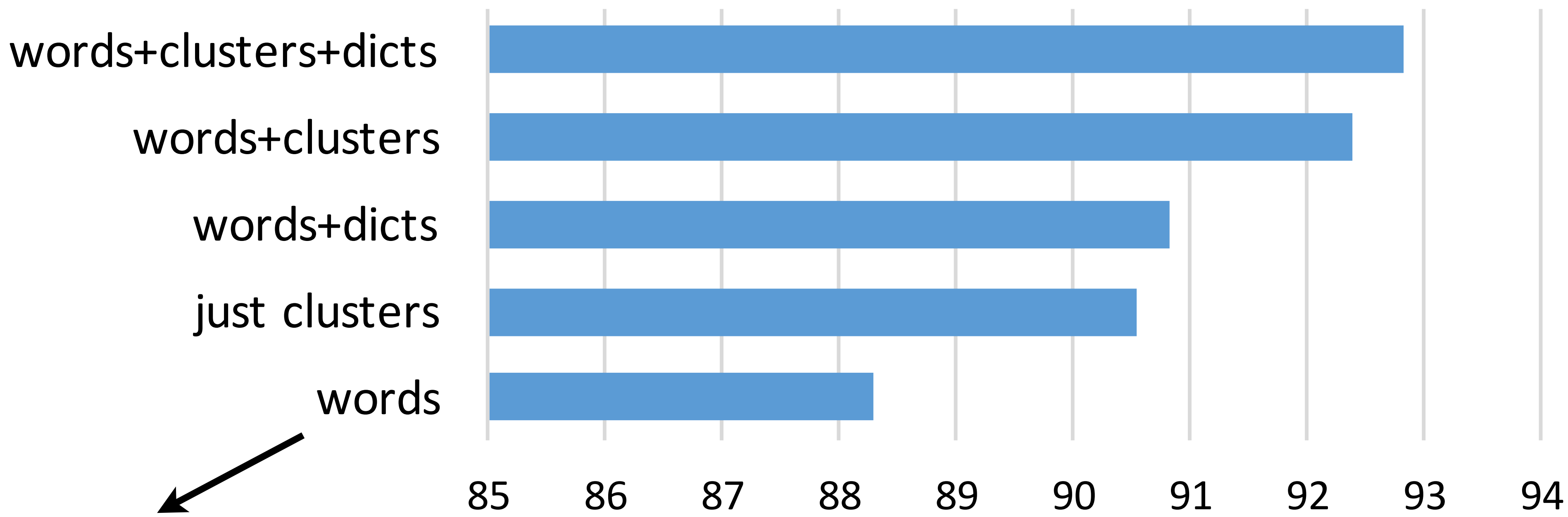
all handcrafted features
(shape, regexes, char ngrams)

Test set accuracy

Clusters help POS tagging

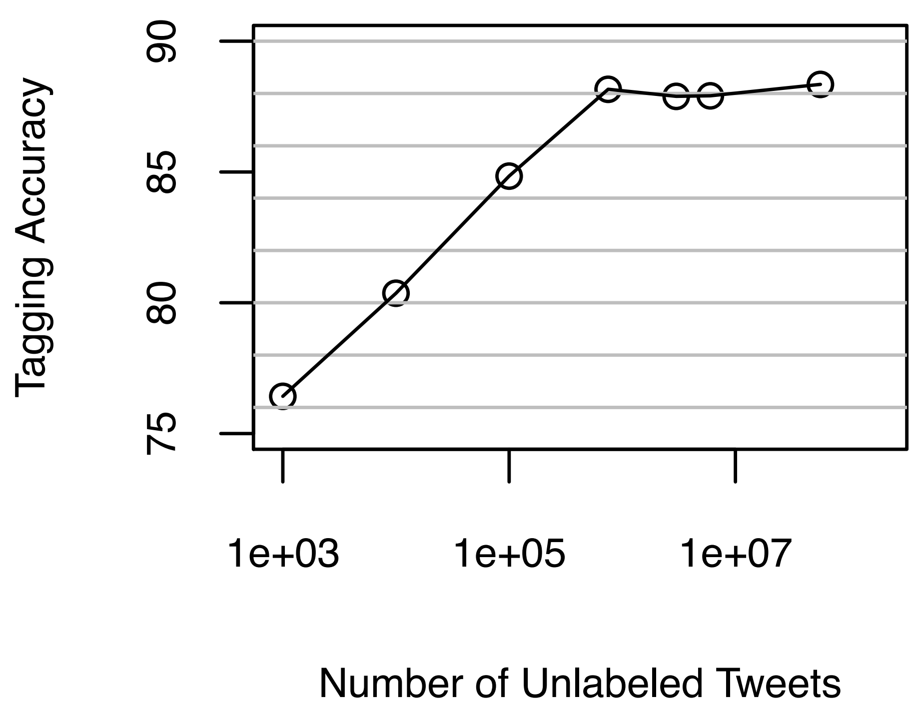


Clusters help POS tagging



all handcrafted features
(shape, regexes, char ngrams)

Test set accuracy



Dev set accuracy
using only clusters as features

Using word representations

- Pre-trained word representations you can download
  - Vectors: GloVe algorithm, trained on news, web, twitter
    http://nlp.stanford.edu/projects/glove/
  - Vectors: word2vec algorithm, trained on web news
    https://code.google.com/p/word2vec/
  - Clusters: Brown hierarchical clustering, trained on twitter
    http://www.ark.cs.cmu.edu/TweetNLP/
  - Or, train your own embeddings; open-source software for all the above
- Incorporate as features for supervised learning
  - (Hierarchical) cluster ID
  - Embedding dimension value (or cluster it...)
  - Similarity to seed term(s)
- Manual dictionary building: go through similarity list, manually select ones you want
- Compare: manually curated lexical resources like WordNet and Freebase

Ongoing research in word representations

- Morphology
- Phrases and multiwords: compositionality
- Vector-valued summaries of sentences, paragraphs, documents?
- Neural networks / deep learning

Question:

What do the words ‘spinning’ and ‘repeating’ have in common?

How could we use this to learn better word embeddings?

Morphological Neural Language Model

- Represent every word type as a feature vector.
- Learn an embedding for every feature.
- The embedding for a word is the sum of the embeddings of its features.

Word Pair - Path

I *ate the* cake
He *ate the* burger
Michelle *ate the* pizza

I *ate the* cake
He *ate the* burger
Michelle *ate the* pizza

Word pairs that appear with similar patterns have similar semantic relationships (Turney et al., 2003)

I, He, and Michelle are similar

Cake, Burger, and Pizza are similar

Word Pair - Path

I *ate the* cake, He *ate the* burger, Michelle *ate the* pizza

Path

(Word Type, Word Type)

Word pairs that appear with similar patterns have similar semantic relationships (Turney et al., 2003)

I, He, and Michelle are similar

Cake, Burger, and Pizza are similar

Word Pair - Path

Path

(Word Type, Word Type)

Patterns are similar if they have similar arguments.

Zuckerberg, CEO of Facebook, Zuckerberg, head of Facebook, Zuckerberg, head honcho at Facebook