Word Embeddings and Neural Language Models

David Belanger

CS 585

REVIEW

Question:

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

Question:

What is the “Distributional Hypothesis” (from last lecture)?

Answer:

- “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)

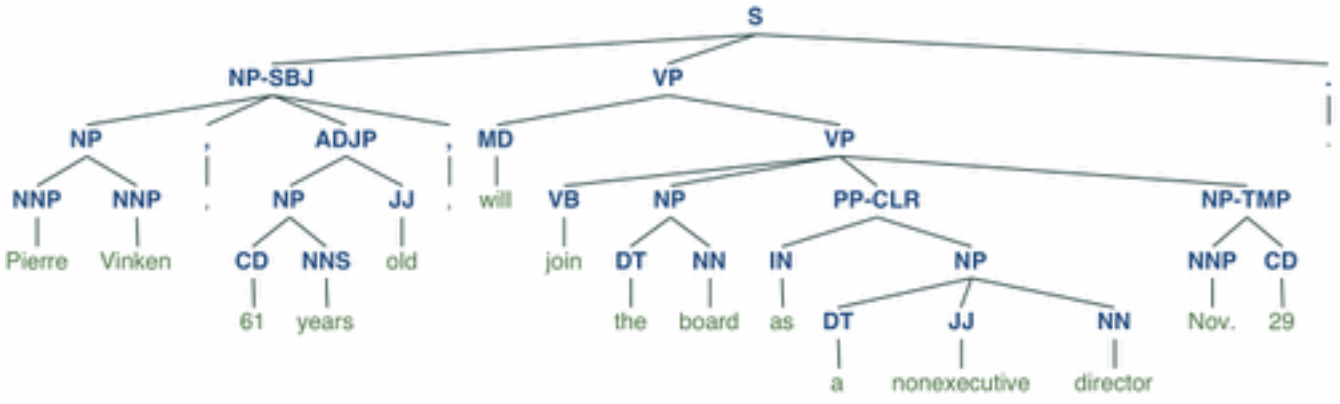
The “context” of a token

Target word: blue
Context words: red

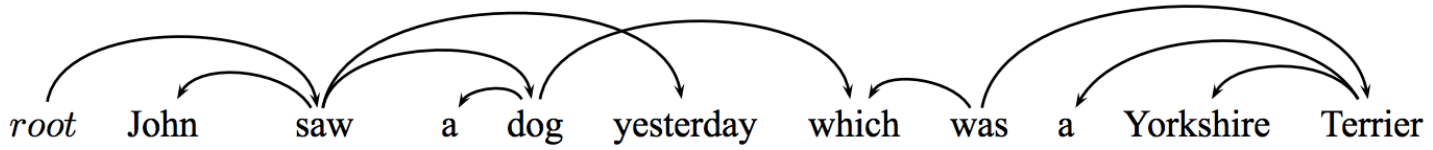
She told the story, however, with great spirit among her friends; for she had a lively, playful disposition, which delighted in anything ridiculous.

(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

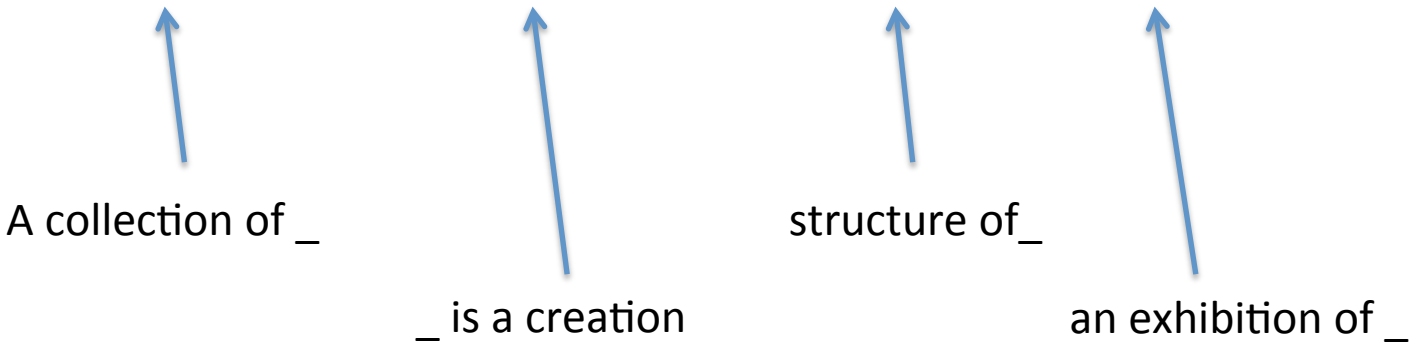
$$V_i = [0, 1, 0, 0, 0, 4, 0, 0, 0, 2, 0, 0, 1]$$

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

Example

- Find contexts containing “art”

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



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

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

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:

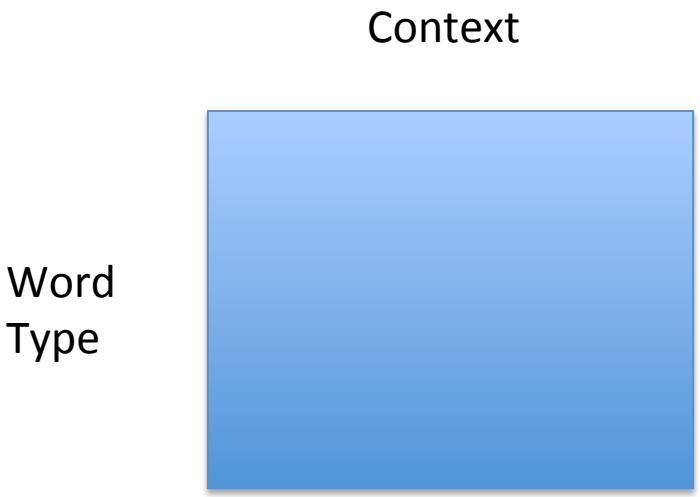
What are the pros and cons of using a wide window for a token's context?

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

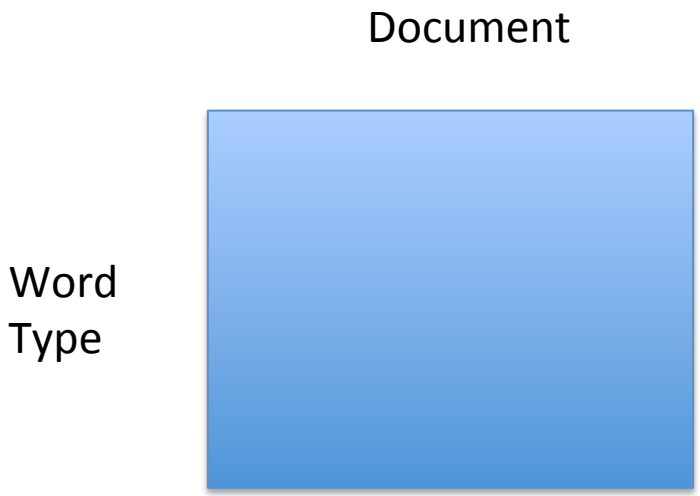


Distributional hypothesis:

– 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

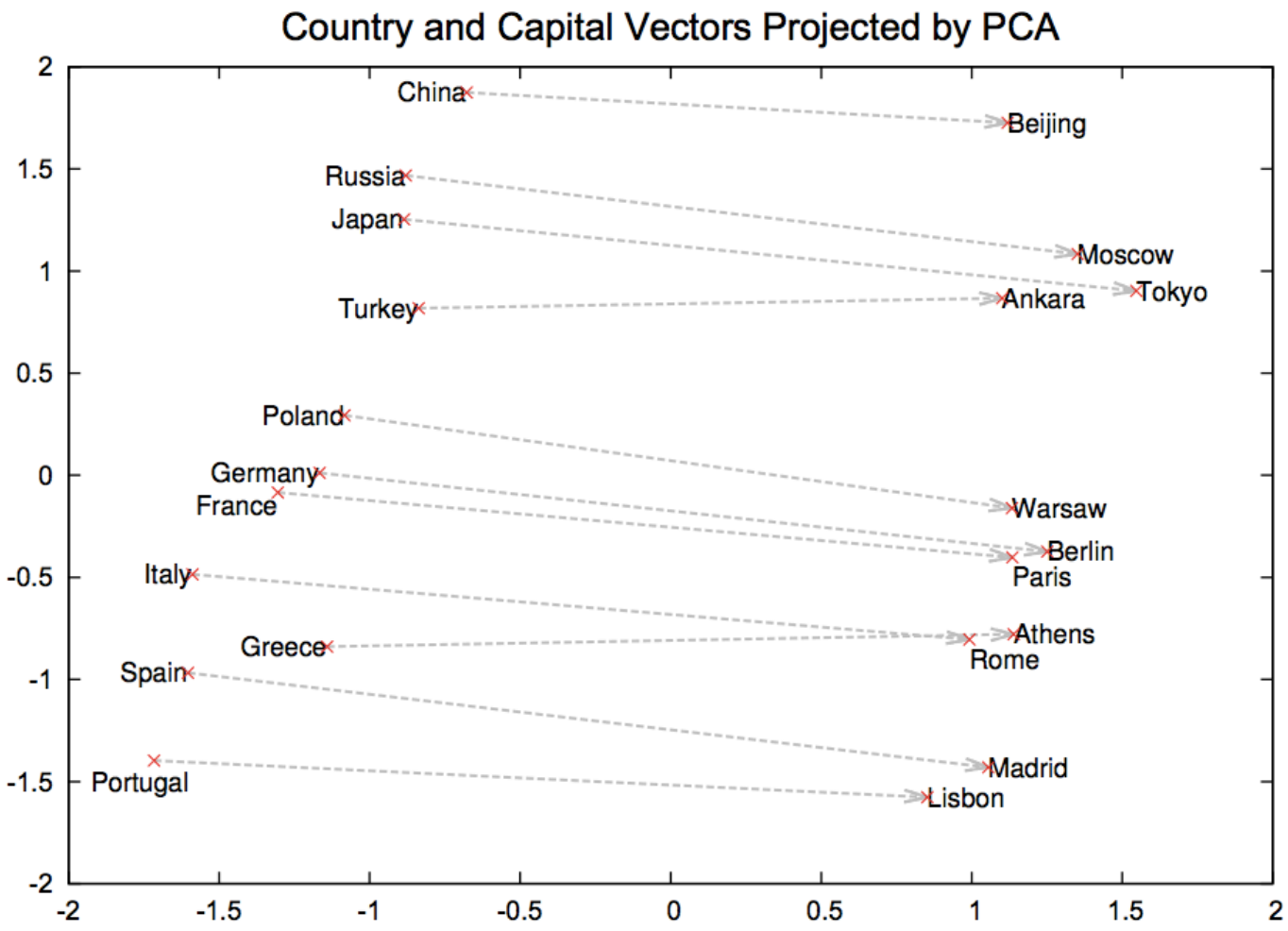
Sparse Context Vector (10 million+ dimensional):

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

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

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

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


Country and Capital Vectors Projected by PCA
China
Beijing
Russia
Moscow
Japan
Tokyo
Turkey
Ankara
Poland
Warsaw
Germany
Berlin
France
Paris
Italy
Athens
Greece
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.

Learning Word Embeddings

- Try to recover the cooccurrence matrix.
  - Easily doable using eigen decomposition.
- Treat unsupervised learning as supervised learning.
  - Next word prediction (i.e. language modeling) is a supervised task.

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).

Neural Language Model

Trigram Language Model:

$$P(w_t|w_{t-1}, w_{t-2})$$

Neural Language Model

$$P(w_t|w_{t-1}, w_{t-2}) = P(E(w_t)|E(w_{t-1}), E(w_{t-2}))$$

The log-likelihood is differentiable. We can optimize the embeddings with gradient descent.

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.

Have Questions or Want to Read More?

Post on Piazza

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