Foundations of Natural Language Processing

Lecture 9: Distributional Semantics

Mirella Lapata
School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

Slides based on content from: Philipp Koehn, Alex Lascarides, Sharon Goldwater, Shay Cohen, Khalil Sima'an, Ivan Titov, Hinrich Schuetze

Key Concept: Semantic Similarity

Two words are semantically similar if they have **similar meanings**.

astronaut $\Longleftrightarrow$ cosmonaut

gobble $\Longleftrightarrow$ devour

huge $\Longleftrightarrow$ large

Key Concept: Semantic Similarity

Two words are semantically similar if they have **similar meanings**.

astronaut $\Longleftrightarrow$ cosmonaut

gobble $\Longleftrightarrow$ devour

huge $\Longleftrightarrow$ large

- How about “banana” and “apple”?

Key Concept: Semantic Similarity

Two words are semantically similar if they have **similar meanings**.

astronaut $\Longleftrightarrow$ cosmonaut

gobble $\Longleftrightarrow$ devour

huge $\Longleftrightarrow$ large

- How about “banana” and “apple”?
- Are “car” and “flower” similar?

Key Concept: Semantic Similarity

Two words are semantically similar if they have **similar meanings**.

astronaut $\Longleftrightarrow$ cosmonaut

gobble $\Longleftrightarrow$ devour

huge $\Longleftrightarrow$ large

- How about “banana” and “apple”?
- Are “car” and “flower” similar?
- And what do you think about “car” and “pope”?

Why is semantic similarity interesting?

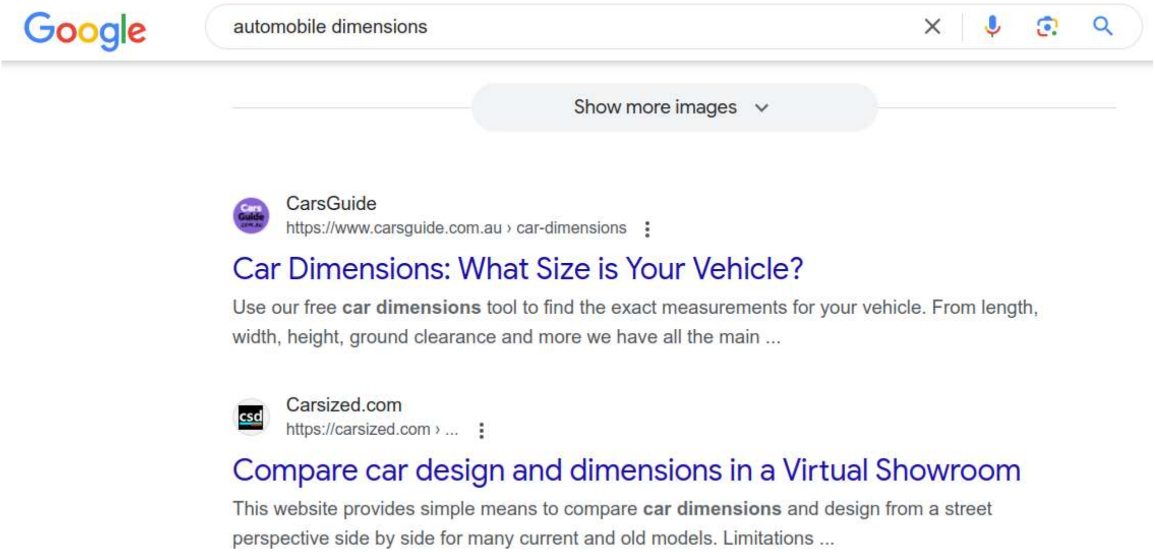
It's a **solvable problem** (see below). Many other things we want to do with language are more interesting, but much harder to solve.

- We do not need **annotated data**.
- There are **many applications** for semantic similarity.
- Two examples of applications:
  1. Direct use of measures of semantic similarity
  2. Plagiarism detection

Application 1: Direct use of semantic similarity

- **Query expansion** in information retrieval
- User types in query [automobile]
- Search engine expands with semantically similar word [car]
- The search engine then uses the query [car OR automobile]
- Better results for the user

Google: Internal model of semantic similarity



Application 2: Plagiarism Detection

ORIGINALITY REPORT

36%	11%	17%	33%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Massey University Student Paper	18%
2	Submitted to GradeGuru Publication	6%
3	Submitted to Foothill College Student Paper	4%
4	www.geography.ccsu.edu Internet Source	4%
5	Submitted to CSU, Chico Student Paper	2%
6	Submitted to South Birmingham College Student Paper	1%
7	Submitted to University of College Cork Student Paper	1%
8	Submitted to CSU, Fullerton Student Paper	1%
9	nou.edu.ng Internet Source	<1%

Body

- **Humidity**

Humidity refers to water vapour in the air. The capacity of air to hold water vapour is primarily a function of temperature. Warmer air has a greater capacity for holding water vapour than cooler air. The temperature at which a body of air becomes saturated is its dew-point temperature.

Relative humidity is a ratio of the amount of water vapour that is actually in the air, compared with the maximum water vapour the air could hold at a given temperature. If the air is saturated with all the moisture it can hold for its temperature, the relative humidity is 100%. A further increase of water vapour or a decrease in temperature results in active condensation. Relative humidity varies due to evaporation, condensation or temperature changes. All three affect both the moisture content and the capacity of the air to hold water vapour. It is highest at dawn, when air temperature is lowest and the capacity of air is less, and also lowest in late afternoon, where higher air temperatures increase the capacity of air to hold water vapour.

- **Adiabatic Processes**
  - **Adiabatic Warming and Cooling**

In order for precipitation to occur, processes need to take place. Adiabatic processes are the changes in temperature that occur due to variations in the air pressure. When water

The Distributional Hypothesis

How many of you know what **tesgüino** means?

The Distributional Hypothesis

How many of you know what **tesgüino** means?

a bottle of **tesgüino** is on the table
everybody likes **tesgüino**
tesgüino makes you drunk
we make **tesgüino** out of corn

The Distributional Hypothesis

How many of you know what **tesgüino** means?

a bottle of **tesgüino** is on the table
everybody likes **tesgüino**
tesgüino makes you drunk
we make **tesgüino** out of corn

Tesgüino is cold fermented beverage made from corn and popularly consumed in the Mexican states of Jalisco, Colima, Nayarit and Oaxaca.

The Distributional Hypothesis

How many of you know what **tesgüino** means?

a bottle of **tesgüino** is on the table
everybody likes **tesgüino**
tesgüino makes you drunk
we make **tesgüino** out of corn

Tesgüino is cold fermented beverage made from corn and popularly consumed in the Mexican states of Jalisco, Colima, Nayarit and Oaxaca.

- Perhaps we can infer meaning just by looking at the contexts a word occurs in
- Perhaps meaning IS the contexts a word occurs in (Wittgenstein!)
- Either way, similar contexts imply similar meanings
- This idea is known as the **distributional hypothesis** (Harris, 1954; Firth, 1857).

Distributional Semantics and Word Embeddings

- **Distributional semantics** is an approach to semantics that is based on the **contexts** of words in **large corpora**.
- The basic notion formalized in distributional semantics is **semantic similarity**.
- **Word embeddings** are the modern incarnation of distributional semantics– adapted to work well with deep learning.

In this lecture, **semantic similarity** also includes **semantic relatedness** (e.g., “car” and “motorway” are related but not similar).

Key concept: Cooccurrence count

Cooccurrence Count

Basis for precise definition of “semantic similarity”. The cooccurrence count of words w_1 and w_2 in a corpus is the number of times that w_1 and w_2 cooccur.

Different definitions of cooccurrence:

- in a linguistic relationship with each other (e.g., w_1 is a modifier of w_2) or
- in the same sentence or
- in the same document or
- within a distance of at most k words (where k is a parameter)

Word cooccurrence in Wikipedia: Examples

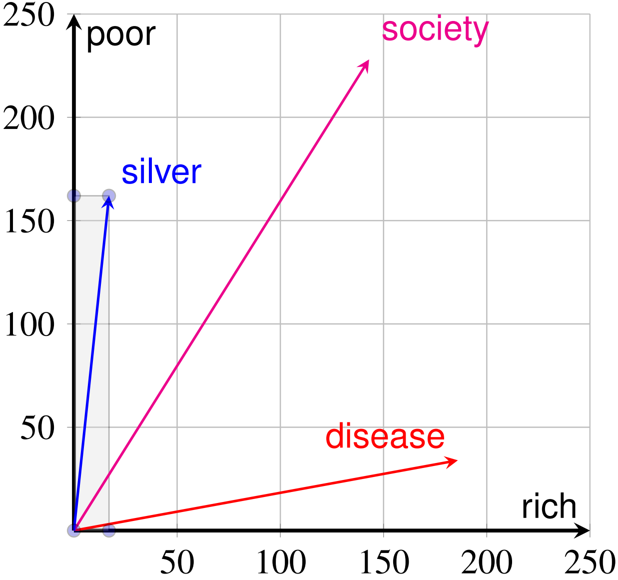
We define cooccurrence in this example as occurrence within $k = 10$ words of each other.

corpus = English Wikipedia

cooc.(rich,silver) = 186
cooc.(rich,society) = 143
cooc.(rich,disease) = 17

cooc.(poor,silver) = 34
cooc.(poor,society) = 228
cooc.(poor,disease) = 162

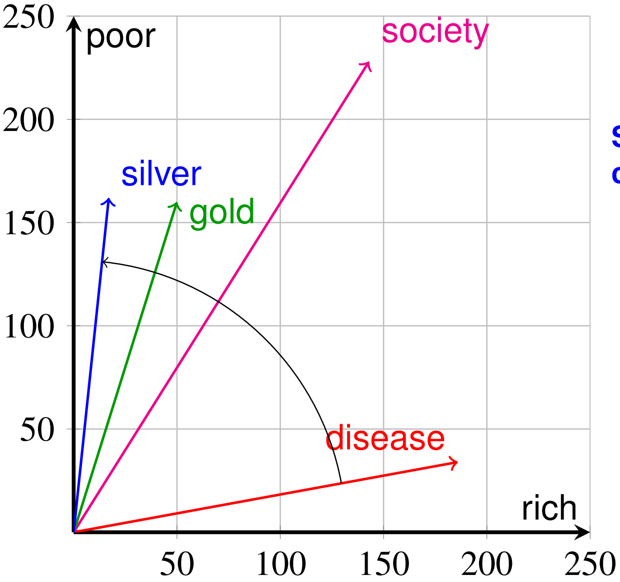
Cooccurrence counts → Count vectors



cooc.(rich,silver) = 186
cooc.(rich,society) = 143
cooc.(rich,disease) = 17

cooc.(poor,silver) = 34
cooc.(poor,society) = 228
cooc.(poor,disease) = 162

Cooccurrence counts → Vectors → Similarity



Similarity between two words is the **cosine** of the angle between them.

- Small angle: silver and gold are similar.
- Medium-size angle: silver and society are not very similar.
- Large angle: silver and disease are even less similar.

Dimensionality of vectors

- Up to now we've only used two dimension words: rich and poor
- Now do this for a very large number of dimension words: hundreds, thousands, or even millions of dimension words.
- This is now a very high-dimensional space with a large number of vectors represented in it.
- But formally, there is no difference to a two-dimensional space with four vectors.

Note: a word has **dual role** in the vector space
(1) each word is a **dimension word**, an axis of the space.
(2) but each word is also a **vector** in that space.

Measures of Similarity

The cosine of the angle between two vectors $\mathbf{x}$ and $\mathbf{y}$ is:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

The Euclidean distance of two vectors $\mathbf{x}$ and $\mathbf{y}$ is:

$$||\mathbf{x} - \mathbf{y}|| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Many more similarity measures exist.

Cooccurrence count (CC) matrix

		w_2				
		rich	poor	silver	society	disease
w_1	rich					
	poor					
	silver					
	society					
	disease					

Cooccurrence count (CC) matrix

		w_2				
		rich	poor	silver	society	disease
	rich	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)
	poor	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)
w_1	silver	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)
	society	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)
	disease	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)	CC(w_1, w_2)

Cases where distributional semantics fails

- Antonyms are judged to be similar: “disease” and “cure”.
- Ambiguity: “Cambridge”
- Non-specificity (occurs in a large variety of different contexts and has few/no specific semantic associations): “person”
- The corpus meaning is different from the meaning that comes to mind when the word is encountered without context: “umbrella”.
- Tokenization issues: “metal”

Pointwise Mutual Information

Pointwise Mutual Information (PMI): weighting of cooccurrence counts. We are replacing the raw cooccurrence count with PMI, a measure of surprise.

$$\mathrm{PMI}(w_1, w_2) = \log \frac{P(w_1,w_2)}{P(w_1)P(w_2)}$$

- If w_1, w_2 independent: PMI(w_1, w_2) = 0
- If w_1, w_2 perfectly correlated:
  $P(w_1, w_2) = P(w_1) = P(w_2)$, PMI$(w_1, w_2) = \log \frac{P(w_2)}{P(w_2)P(w_2)} = \log \frac{1}{P(w_2)}$
- If w_1, w_2 positively correlated: PMI(w_1, w_2) is large and positive.
- If w_1, w_2 negatively correlated: PMI(w_1, w_2) is large and negative.

Pointwise Mutual Information

Pointwise Mutual Information (PMI): weighting of cooccurrence counts. We are replacing the raw cooccurrence count with PMI, a measure of surprise.

$$\mathrm{PMI}(w_1, w_2) = \log \frac{P(w_1,w_2)}{P(w_1)P(w_2)}$$

- If w_1, w_2 independent: PMI(w_1, w_2) = 0
- If w_1, w_2 perfectly correlated:
  $P(w_1, w_2) = P(w_1) = P(w_2)$, PMI$(w_1, w_2) = \log \frac{P(w_2)}{P(w_2)P(w_2)} = \log \frac{1}{P(w_2)}$
- If w_1, w_2 positively correlated: PMI(w_1, w_2) is large and positive.
- If w_1, w_2 negatively correlated: PMI(w_1, w_2) is large and negative.
- What does it mean to have a negative PMI?

Pointwise Mutual Information

Pointwise Mutual Information (PMI): weighting of cooccurrence counts. We are replacing the raw cooccurrence count with PMI, a measure of surprise.

$$\mathrm{PMI}(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad \mathrm{PPMI}(w_1, w_2) = \max\left(\log \frac{P(w_1, w_2)}{P(w_1)P(w_2)}, 0\right)$$

- If w_1, w_2 independent: PMI(w_1, w_2) = 0
- If w_1, w_2 perfectly correlated:
  $P(w_1, w_2) = P(w_1) = P(w_2)$, PMI$(w_1, w_2) = \log \frac{P(w_2)}{P(w_2)P(w_2)} = \log \frac{1}{P(w_2)}$
- If w_1, w_2 positively correlated: PMI(w_1, w_2) is large and positive.
- If w_1, w_2 negatively correlated: PMI(w_1, w_2) is large and negative.
- What does it mean to have a negative PMI? Replace negative PMI values with zero.

Summary: Constructing Vector Spaces

Informal algorithm for constructing vector spaces:

- Select a corpus
- Select n target words which will be represented as vectors in the space;
- Select k dimension words (they are found around target word in the context window)
- compute $k \times n$ cooccurrence matrix
- Compute (PPMI): weighted cooccurrence matrix
- Compute similarity of any two focus words as the cosine of their vectors

Bag of words model

- We do not consider the **order** of words in a context.
- *John is quicker than Mary* and *Mary is quicker than John* give rise to same cooccurrence counts.
- This is called a **bag of words model**.
- More sophisticated models: compute dimension features based on the parse of a sentence – the feature “is object of the verb cook” would be recovered from both “John cooked the ham” and “the ham was cooked”.

Embeddings

Definition

The embedding of a word w is a dense vector $\vec{v}(w) \in \mathcal{R}^k$ that represents semantic and other properties of w. Typical values are $50 \leq k \leq 1,000$.

- It appears there is little difference to count vectors: Both embeddings and count vectors are representations of words, primarily semantic, but also capturing other properties.
- Embeddings have much lower dimensionality than count vectors.
- Count vectors are **sparse** (most entries are 0), embeddings **dense** (almost never happens that an entry is 0).
- Embeddings are **lower-dimensional** (e.g., 100–300 dimensions).

Embedding Learning Algorithms

Singular Value Decomposition (SVD)

- Also called Latent Semantic Indexing (LSI)
- Factorization of cooccurrence matrix
- Least squares objective optimized by power method

Word2Vec

- A group of related models used to generate word embeddings
- Word2Vect models are optimized by gradient descent
- Skip-gram model predicts surrounding words (context) given a target word

Embedding Learning Algorithms

Singular Value Decomposition (SVD)

- Also called Latent Semantic Indexing (LSI)
- Factorization of cooccurrence matrix
- Least squares objective optimized by power method

Linear Algebra: Recap

Dot product

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

Example

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3$$

Matrix Multiplication

$$\begin{matrix} A & \cdot & B & = & AB \\ m \times n & & n \times p & & m \times p \end{matrix}$$

Example

$$\begin{pmatrix} a_1 & b_1 \\ c_2 & d_1 \end{pmatrix} \cdot \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} = \begin{pmatrix} a_1 a_2 + b_1 c_2 & a_1 b_2 + b_1 d_2 \\ c_1 a_2 + d_1 c_2 & c_1 b_2 + d_1 d_2 \end{pmatrix}$$

Length of vector

$$|\mathbf{d}| = \sqrt{\sum_{i=1}^{n} d_i^2}$$

Orthogonal Vectors

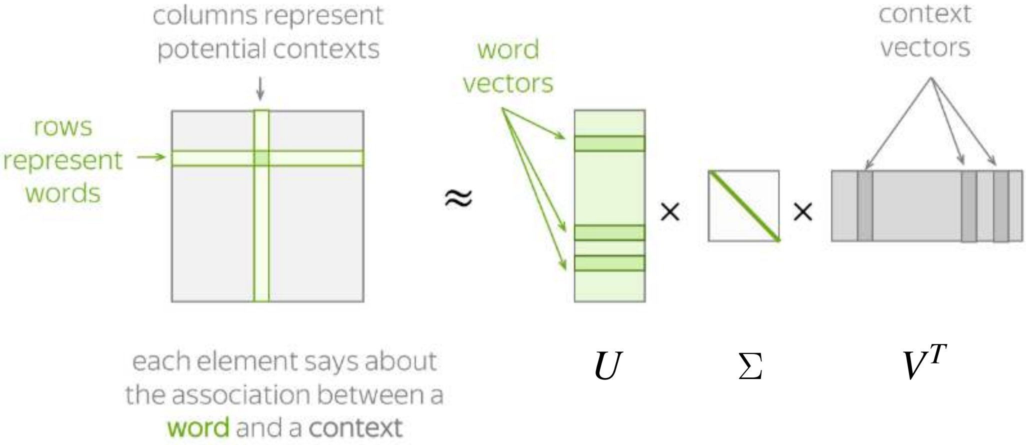
$\mathbf{c}$ and $\mathbf{d}$ are orthogonal iff

$$\sum_{i=1}^{n} c_i \cdot d_i = 0$$

Matrix Factorization: Embeddings

- We will decompose the cooccurrence matrix into a product of matrices.
- The particular decomposition we'll use: singular value decomposition (SVD).
- SVD: $C = U\Sigma V^T$ (where C = cooccurrence matrix, with PPMI weighting)
- We will then use the SVD to compute a new, improved cooccurrence matrix C'.
- We'll get better and more compact word representations out of C' (compared to C).

SVD Visualization



SVD Summary

- We decompose the cooccurrence matrix C into a product of three matrices: U^T
- The input word matrix U – consists of one (row) vector for each word
- The context word matrix V^T – consists of one (column) vector for each context word
- The singular value matrix Σ is a diagonal matrix with singular values, reflecting importance of each dimension
- We only keep first k dimensions and set the others to zero.

Property of SVD that we exploit here

- **Key property:** each singular value tells us how important its dimension is.
- By setting less important dimensions to zero, we keep the important information, but get rid of the “details”.
- These details may be noise – in that case, reduced SVD vectors are a better representation because they are less noisy or make things dissimilar that should be similar – again, reduced SVD vectors are a better representation because they represent similarity better.
- Analogy for “fewer details is better”: Image of a blue flower Image of a yellow flower, Omitting color makes is easier to see the similarity

Example of $C = U\Sigma V^T$: All four matrices

SVD is decomposition of C into a representation of the input words, a representation of the context and a representation of the importance of the "semantic" dimensions

C	w_1	w_2	w_3	w_4	w_5	w_6
rich	1	0	1	0	0	0
poor	0	1	0	0	0	0
silver	1	1	0	0	0	0
society	1	0	0	1	1	0
disease	0	0	0	1	0	1

$=$

U	1	2	3	4	5
rich	−0.44	−0.30	0.57	0.58	0.25
poor	−0.13	−0.33	−0.59	0.00	0.73
silver	−0.48	−0.51	−0.37	0.00	−0.61
society	−0.70	0.35	0.15	−0.58	0.16
disease	−0.26	0.65	−0.41	0.58	−0.09

$\times$

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

$\times$

V^T	w_1	w_2	w_3	w_4	w_5	w_6
1	0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

Embeddings = Left Singular Vectors

U	1	2	3	4	5
rich	−0.44	−0.30	0.00	0.00	0.00
poor	−0.13	−0.33	0.00	0.00	0.00
silver	−0.48	−0.51	0.00	0.00	0.00
society	−0.70	0.35	0.00	0.00	0.00
disease	−0.26	0.65	0.00	0.00	0.00

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

$\times$

V^T	w_1	w_2	w_3	w_4	w_5	w_6
1	0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Summary

- The meaning of a word is learned from its contexts in a large corpus.
- The main analysis method of contexts is co-occurrence.
- Distributional semantics is a good model of semantic similarity. There is a lot more in semantics that distributional semantics is not a good model for.
- Embeddings have lower-dimensionality than count vectors
- Singular value decomposition is one method to obtain dense vector representations.

Next time: generating embeddings with Word2Vec.