WordSpace A Basic Embedding Model

Hinrich Schütze

Center for Information and Language Processing, LMU Munich

2017-07-17

Overview



2 WordSpace



Outline



2 WordSpace



Semantic similarity

- Two words are semantically similar if they have similar meanings.
- Examples of similar words:
 - "furze" \leftrightarrow "gorse"
 - "astronaut" \leftrightarrow "cosmonaut"
 - "car" \leftrightarrow "automobile"
 - "banana" \leftrightarrow "apple" (these two are less similar)
- Examples of not similar words:
 - "car" \leftrightarrow "flower"
 - "car" \leftrightarrow "pope"
- Examples of similar words that are not nouns:
 - "huge" \leftrightarrow "large"
 - "eat" \leftrightarrow "devour"

Furze = gorse = whin



Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" ↔ "autobahn"
- A car is not similar to an autobahn, but there is an obvious relationship between them.
- Linguistically / ontologically well defined relations: synonymy, antonymy, hypernymy, meronymy, troponymy, ...
- Note: "car" ↔ "autobahn" isn't an instance of any of these!
- More generally: Two words are semantically related if their meanings are related in the real world. For example, if one word describes a given situation ("I'm on the autobahn"), then it is very likely that the other word also describes this situation ("I'm in a car").
- There is a spectrum here: synonymous, very similar, less similar, related, unrelated

Distributional semantics WordSpace Norms & scores

Here: Similarity includes relatedness

In what follows,

I will use semantic similarity as a general term that includes semantic similarity and semantic relatedness.

Distributional semantics

- 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.

Why is distributional semantics interesting?

- It's a solvable problem (see below).
 - Many other things we want to do with language are more interesting, but nobody has been able to solve them so far.
- We do not need annotated data.
- There are many applications for distributional semantic similarity.
- Two examples of applications
 - 1. Direct use of measures of semantic similarity
 - 2. OOVs, representations for unknown words

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

auto	comobile prices						Q	
All	News	Shopping	Images	Maps	More		Settings	Tool
Abou	t 69,500,00	0 results (0.41	seconds)					
Auto	omobile	aus Deutso	hland - 2,	,4 Mio. G	Bebrauch	t- & Neu	wagen	
Auto Ad V 4.3 Jetzt Europ Mode	schnell, ein paweite Ang	aus Deutso cout24.de/auto ating for autoso fach & unkomp gebote · Alle Fa ran, Kia Sportag	chland - 2, p/mobile cout24.de liziert Autos a hrzeugdetails ge, BMW X1,	4 Mio. G aller Marken s · Kostenlo Audi A3	Sebrauch in Ihrer Nä s verkaufen	t- & Neu he finden. · Ausgezeid	wagen	

Kelley Blue Book - New and Used Car Price Values, Expert Car Reviews https://www.kbb.com/ *

Check KBB car price values when buying and selling new or used vehicles. Recognized by consumers and the automotive industry since 1926. Resale Value - Used Car Prices - New Cars - Motorcycles

NADAguides: New Car Prices and Used Car Book Values

https://www.nadaguides.com/ -

Research the latest new car prices, deals, used car values, specs and more. NADA Guides is the leader in accurate vehicle pricing and vehicle information. New Car Prices & Used Car... · Motorcycles - RV Prices and Values · Trucks

Application 2: OOVs, representations for unknown words

• Entity typing

- We often need to infer properties of a new (OOV) entity.
- For example, if the system encounters "Fonsorbes" for the first time, it is useful to be able to infer that it is a town.
- Embeddings contain valuable information about OOVs.

Entity embeddings (learned with word2vec)



Schütze: WordSpace

Embedding-based entity typing: Given embedding, predict correct types of entity



Distributional Semantics: History

- Harris
- Firth
- Leibniz
- Miller

WordSpac

Zellig Harris



...difference in meaning correlates with difference of distribution. (1954)

John Rupert Firth



You shall know a word by the company it keeps. (1957)

Gottfried Wilhelm Leibniz



Eadem sunt quorum unum potest substitui alteri salva veritate. (17th century) – Those things are identical of which one can be substituted for the other without loss of truth. This is a definition of synonymy.

George A. Miller



Those things are similar of which one can be substituted for the other without loss of plausibility. (1991)

Miller & Charles

- Starting point: Leibniz
- It is doubtful there are any true synonyms if this is our definition.
- Replace "loss of truth" with "loss of plausibility": Those things are similar of which one can be substituted for the other without loss of plausibility.
- Hence: The semantic similarity [between words] is a function of the contexts in which they are used. (Miller and Charles 1991)

Exercise

- Given: a large text corpus (e.g., of English)
- Come up with an algorithm that computes a rough measure of semantic similarity between two words
 - For example, the algorithm should tell us that "car" and "automobile" are similar, but "car" and "flower" are not.

Outline







Semantic similarity based on cooccurrence

- Assume the equivalence of:
 - Two words are semantically similar.
 - Two words occur in similar contexts (Miller & Charles, roughly).
 - Two words have similar word neighbors in the corpus.
- Elements of this are from Harris, Firth, Leibniz and Miller.
- Strictly speaking, similarity of neighbors is neither necessary nor sufficient for semantic similarity.
- But perhaps this is good enough.

Key concept: Cooccurrence count

Cooccurrence count:

basis for precise definition of "similar neighbor"

- The cooccurrence count of words w_1 and w_2 in corpus G 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., w₁ is a modifier of w₂) 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

- Here: cooccurrence defined as occurrence within k = 10 words of each other
- corpus = English Wikipedia
 - cooc.(rich,silver) = 186
 - cooc.(poor,silver) = 34
 - cooc.(rich,disease) = 17
 - cooc.(poor,disease) = 162
 - cooc.(rich,society) = 143
 - cooc.(poor,society) = 228

Cooccurrence counts \rightarrow Vector space



cooc.(poor,silver)=34, cooc.(rich,silver)=186, cooc.(poor,disease)=162, cooc.(rich,disease)=17, cooc.(poor,society)=228, cooc.(rich,society)=143



Add "society" to the graph.



The similarity between two words is the cosine of the angle between them.



The similarity between two words is the cosine of the angle between them.



The similarity between two words is the cosine of the angle between them.

Small angle: silver and gold are similar.



The similarity between two words is the cosine of the angle between them.

Medium-size angle: silver and society are not very similar.

Schütze: WordSpace



The similarity between two words is the cosine of the angle between them.

Large angle: silver and disease are even less similar.

Dimensionality of WordSpace

- 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 three vectors.
- Note: a word has dual role in WordSpace.
 - Each word is a dimension word, an axis of the space.
 - But each word is also a vector in that space.

Same formalism, but more dimensions & more vectors



Nearest neighbors of "silver" in WordSpace

1.000 silver / 0.865 bronze / 0.842 gold / 0.836 medal / 0.826 medals / 0.761 relay / 0.740 medalist / 0.737 coins / 0.724 freestyle / 0.720 metre / 0.716 coin / 0.714 copper / 0.712 golden / 0.706 event / 0.701 won / 0.700 foil / 0.698 Winter / 0.684 Pan / 0.680 vault / 0.675 jump

Nearest neighbors of "disease" in WordSpace

1.000 disease / 0.858 Alzheimer / 0.852 chronic / 0.846 infectious / 0.843 diseases / 0.823 diabetes / 0.814 cardiovascular / 0.810 infection / 0.807 symptoms / 0.805 syndrome / 0.801 kidney / 0.796 liver / 0.788 Parkinson / 0.787 disorders / 0.787 coronary / 0.779 complications / 0.778 cure / 0.778 disorder / 0.778 Crohn / 0.773 bowel

TensorBoard

Wikipedia WordSpace demonstration

Exercise

- Find an example word w where WordSpace fails
- That is: the list of words you get from a person when asking them to give you "similar words to *w*" ...
- ... is very different from what the WordSpace gives you.
- Two subtasks
 - find the word
 - explain why it fails

Cases where WordSpace 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 Wikipedia meaning is different from the meaning that comes to mind when the word is encountered without context: "umbrella".
- Tokenization issues: "metal"

Outline



2 WordSpace



How to make WordSpace work well: Two important details

Norms:

When comparing vectors, we often want to normalize them first.

Scores:

Raw cooccurrence counts don't work well. We need to weight / transform them.

Norms

- How do we formalize semantic similarity in WordSpace?
- Earlier we used cosine.
- Would distance between two points not be simpler?
- ... i.e., Euclidean distance between the end points of the two vectors?
- Euclidean distance is a bad idea
- ... because Euclidean distance is large for vectors of different lengths.

Why distance is a bad idea



The Euclidean distance of "sick" and "disease" is large although the types of neighbors they occur with are very similar. "sick" is just a lot more frequent than "disease". Distance is bad as a similarity measure: How do we fix this?

- There are two equivalent ways of fixing it.
- Use angle/cosine of vectors as similarity measure
- Use distance of length-normalized vectors as similarity measure

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- The angle between "sick" and "disease" is close to 0, corresponding to maximal similarity ...
- ... even though the Euclidean distance between the two vectors is large.

Cosine similarity illustrated



Cosine similarity illustrated



From angles to cosines

- The following two notions are equivalent.
 - Rank words *w_i* according to the angle between *w_i* and a target word *v* in decreasing order.
 - Rank words w_i according to $cosine(w_i, v)$ in increasing order
- Cosine is a monotonically decreasing function of the angle for the interval [0°, 180°]

WordSpac

Cosine



Cosine similarity between two words $\cos(\vec{c}, \vec{d}) = \sin(\vec{c}, \vec{d})$

$$\begin{aligned} \cos(\vec{c}, \vec{d}) &= \frac{\vec{c}}{|\vec{c}|} \cdot \frac{\vec{d}}{|\vec{d}|} \\ &= \frac{\vec{c} \cdot \vec{d}}{|\vec{c}| |\vec{d}|} \\ &= \frac{\sum_{i=1}^{|V|} c_i d_i}{\sqrt{\sum_{i=1}^{|V|} c_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}} \end{aligned}$$

 $|\vec{c}|$ and $|\vec{d}|$ are the lengths of \vec{c} and \vec{d} .

Length normalization

- A vector is (length-) normalized by dividing each of its components by its length here we use the L_2 norm: $||x||_2 = \sqrt{\sum_i x_i^2}$
- This maps vectors onto the unit sphere ...
- ... since after normalization: $||x||_2 = \sqrt{\sum_i x_i^2} = 1.0$
- As a result, less frequent words and more frequent words have weights of the same order of magnitude.
- Effect on the vectors of "sick" and "disease": they have almost identical vectors after length-normalization.

Norms & scores

Cosine similarity between two words for normalized vectors

$$\begin{aligned} \cos(\vec{c}, \vec{d}) &= \frac{\vec{c}}{|\vec{c}|} \cdot \frac{\vec{d}}{|\vec{d}|} \\ &= \frac{\vec{c}}{1} \cdot \frac{\vec{d}}{1} \\ &= \frac{\sum_{i=1}^{|V|} c_i d_i}{1} \\ &= \sum_{i=1}^{|V|} c_i d_i \end{aligned}$$

For normalized vectors, cosine and dot product are the same.

Raw cooccurrence counts: Limitations

- Recall our raw data are cooccurrence counts like these: cooc.(rich,silver) = 186 cooc.(poor,silver) = 34
- False hope: Cooccurrence measures how strongly two words are associated.
- Why this is a false hope: cooc.(rich,silver) = 186 cooc.(rich,rhodium) = 2
- Coccurrence counts are influenced by base frequency.
- $\bullet~$ "rhodium" is infrequent \rightarrow low cooccurence counts
- What we really need is a measure of: how much higher/lower than expected is the count?

Rhodium: Most expensive metal



PMI: Normalization of cooccurrence counts

- PMI: pointwise mutual information
- $PMI(w_1, w_2) = \log \frac{P(w_1 w_2)}{P(w_1)P(w_2)}$
- P(x): probability of event x
- We are replacing the raw cooccurrence count with PMI, a measure of surprise.

PMI: Normalization of cooccurrence counts

- $PMI(w_1, w_2) = \log \frac{P(w_1w_2)}{P(w_1)P(w_2)}$, a measure of surprise
- If w_1 , w_2 independent: PMI $(w_1, w_2) = 0$
- If w_1 , w_2 perfectly correlated: PMI(w_1 , w_2) = log[1/P(w_2)]
- If w₁, w₂ positively correlated: PMI(w₁, w₂) is large and positive.
- If w₁, w₂ negatively correlated:
 PMI(w₁, w₂) is large and negative.

PPMI

• PPMI =

positive pointwise mutual information

- $PPMI(w_1, w_2) = max(0, PMI(w_1, w_2))$
- More generally (with offset k): $PPMI(w_1, w_2) = max(0, PMI(w_1, w_2) - k)$

Motivation for using PPMI instead of PMI

- $PPMI(w_1, w_2) = max(0, PMI(w_1, w_2) k)$
- Most interesting correlations of the sort we're interested in are positive.
- For example, it is very hard to find negative correlations among words that are meaningful.
- (give example)
- Motivation for offset: Small correlations may be due to noise, so discard them as well.

Cooccurrence count matrix

		vectors			
		rhodium	gold	disease	
9					
ous	take	100	10000	10000	
nsi	rich	4	400	100	
me	poor	1	100	400	
- - -					

Cooccurrence count matrix: Cosine, no PPMI

			vectors	
		rhodium	gold	disease
ons	take	100	10000	10000
nsi	rich	4	400	100
ше	poor	1	100	400
<u>.</u>		•		
		cosines		
		rhodium	gold	disease
rhodium		1.0	1.0	0.9991
gold		1.0	1.0	0.9991
disease		0.9991	0.9991	1.0

Cooccurrence count matrix: Cosine, PPMI weighting

			vectors	
		rhodium	gold	disease
ons	take	100	10000	10000
nsi	rich	4	400	100
ше	poor	1	100	400
.				
			cosines	
		rhodium	gold	disease
rhodium		1.0	1.0	0.3497
gold		1.0	1.0	0.3497
disease		0.3497	0.3497	1.0

Exercise

$$\begin{pmatrix} 0.5\\0\\1 \end{pmatrix} \cdot \begin{pmatrix} 2\\4\\2 \end{pmatrix} = ?$$

$$C(w) \quad C(c) \quad C(wc) \quad \mathsf{PMI} \text{ (use } \log_{10})$$

$$100 \quad 100 \quad 1 \quad ?$$

$$100 \quad 100 \quad 100 \quad ?$$

$$5000 \quad 5000 \quad 250 \quad ?$$

$$(\mathsf{total} = 10000)$$

Summary: How to build a WordSpace model

- Select a corpus
- Select *k* dimension words
- Select *n* focus words these will be represented as points or vectors in the space
- Compute $k \times n$ cooccurrence matrix
- Compute number of distinct neighbor statistics
- 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 for k = 10.
- 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".

Limits of distributional semantics?

Taxonomies

- fruit reproductive structure plant organ plant part natural object whole/unit
- seafood food nutrient substance matter
- Distributional semantics has a hard time with traditional semantic notions like negation, scope and quantification although there is currently a lot of research on these topics.
- Ambiguity?

Takeaway Distributional semantics

- 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.

Takeaway WordSpace

- The representation/embedding of a word is a vector of cooccurrence counts.
- Semantic similarity is measured as cosine of cooccurrence vectors.
- The representations are specific to the training corpus. ("umbrella", "gold")

Takeaway Norms & Scores

- Euclidean distance is not a good measure of semantic similarity in WordSpace.
- Cosine is appropriate because it implicitly normalizes for length and (global) frequency.
- PPMI is a good weighting to use for cooccurrence counts because it removes noise and measures "increase compared to expected count" instead of raw cooccurrence.

Resources

- Magnus Sahlgren's 2006 PhD thesis (detailed review of non-embedding WordSpace models)
- P. D. Turney and P. Pantel (2010) "From Frequency to Meaning: Vector Space Models of Semantics", Journal of Artificial Intelligence Research, Volume 37, pages 141–188