Embeddings: The Basics

Hinrich Schütze

Center for Information and Language Processing, LMU Munich

2017-07-17

Overview

1. Distributional semantics
2. WordSpace
3. Norms & scores

Outline

1 Distributional semantics

2 WordSpace

3 Norms & scores

Semantic similarity

Semantic similarity

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

Semantic similarity

- Two words are semantically similar
  if they have similar meanings.
- Examples of similar words:

Semantic similarity

- Two words are semantically similar if they have similar meanings.
- Examples of similar words:
  - “furze” $\leftrightarrow$ “gorse”

Semantic similarity

- Two words are semantically similar if they have similar meanings.
- Examples of similar words:
  - "furze" $\leftrightarrow$ "gorse"
  - "astronaut" $\leftrightarrow$ "cosmonaut"

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"

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)

Semantic similarity

- Two words are semantically similar
  if they have similar meanings.
- Examples of similar words:
  - “furze” ↔ “gorse”
  - “astronaut” ↔ “cosmonaut”
  - “car” ↔ “automobile”
  - “banana” ↔ “apple” (these two are less similar)
- Examples of not similar words:

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"

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”

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:

Semantic similarity

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

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”

Semantic relatedness

Semantic relatedness

- Two words are semantically related if their meanings are related.

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "autobahn"

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "autobahn"
- A car is not similar to an autobahn, but there is an obvious relationship between them.

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "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, ...

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "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 that car-autobahn is not an instance of any of these!

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "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 that car-autobahn is not 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").

Semantic relatedness

- Two words are semantically related if their meanings are related.
- Example: "car" $\leftrightarrow$ "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 that car-autobahn is not 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

Distributional semantics

- Distributional semantics is an approach to semantics that is based on the contexts of words and linguistic expressions in large corpora.

Distributional semantics

- Distributional semantics is an approach to semantics that is based on the contexts of words and linguistic expressions in large corpora.
- The basic notions formalized in distributional semantics are semantic similarity and semantic relatedness.

Why is distributional semantics interesting?

Why is distributional semantics interesting?

- It's a solvable problem (see below).

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.

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.
- There are many applications for distributional semantic similarity/relatedness.

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.
- There are many applications for distributional semantic similarity/relatedness.
  - E.g., query expansion in information retrieval

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.
- There are many applications for distributional semantic similarity/relatedness.
  - E.g., query expansion in information retrieval
  - User types in query [automobile]

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.
- There are many applications for distributional semantic similarity/relatedness.
  - E.g., query expansion in information retrieval
  - User types in query [automobile]
  - Search engine expands with semantically similar word [car]

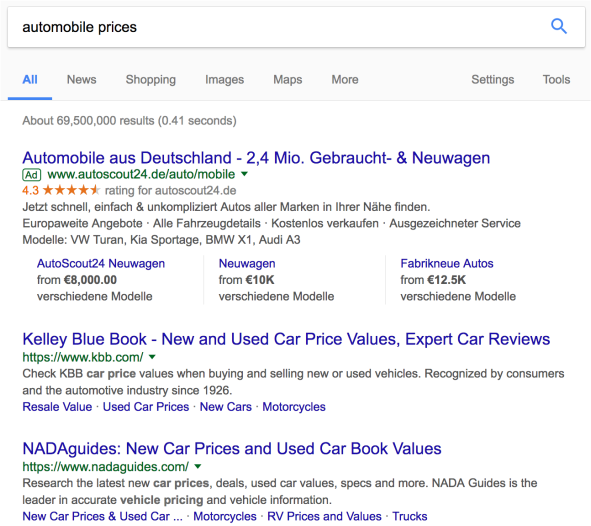
Why is distributional semantics interesting?

- 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.
- There are many applications for distributional semantic similarity/relatedness.
  - E.g., 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]

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.
- There are many applications for distributional semantic similarity/relatedness.
  - E.g., 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



automobile prices

All News Shopping Images Maps More Settings Tools

About 69,500,000 results (0.41 seconds)

Automobile aus Deutschland - 2,4 Mio. Gebraucht- & Neuwagen
Ad www.autoscout24.de/auto/mobile
4.3 ★★★★★ rating for autoscout24.de
Jetzt schnell, einfach & unkompliziert Autos aller Marken in Ihrer Nähe finden.
Europaweite Angebote · Alle Fahrzeugdetails · Kostenlos verkaufen · Ausgezeichneter Service
Modelle: VW Turan, Kia Sportage, BMW X1, Audi A3

AutoScout24 Neuwagen
from **€8,000.00**
verschiedene Modelle

Neuwagen
from **€10K**
verschiedene Modelle

Fabrikneue Autos
from **€12.5K**
verschiedene Modelle

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

Distributional Semantics: History

Distributional Semantics: History

- Leibniz
- Harris
- Firth
- Miller

Gottfried Wilhelm Leibniz

Gottfried Wilhelm Leibniz

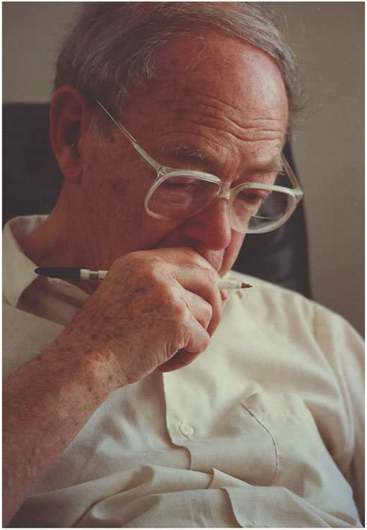
Eadem sunt quorum unum potest substitui alteri salva veritate. (17th century) – Those things are identical of which one can be substituted for the other without loss of truth.

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.

Zellig Harris

Zellig Harris



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

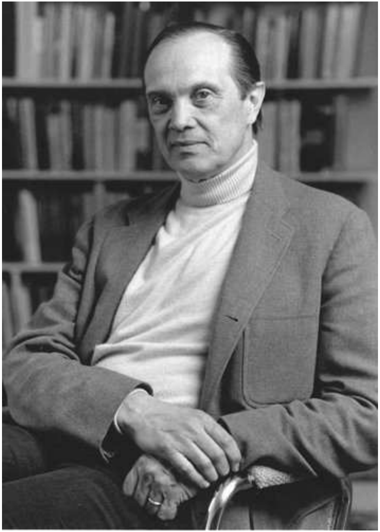
John Rupert Firth

John Rupert Firth

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

George A. Miller

George A. Miller



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

Miller & Charles

Miller & Charles

- Starting point: Leibniz

Miller & Charles

- Starting point: Leibniz
- It is doubtful there are any true synonyms if this is our definition.

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.

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

Exercise

- Given: a large text corpus (e.g., of English)

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

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

1 Distributional semantics

2 WordSpace

3 Norms & scores

Semantic similarity based on cooccurrence

Semantic similarity based on cooccurrence

- Assume the equivalence of:

Semantic similarity based on cooccurrence

- Assume the equivalence of:
  - Two words are semantically similar.

Semantic similarity based on cooccurrence

- Assume the equivalence of:
  - Two words are semantically similar.
  - Two words occur in similar contexts (Miller & Charles, roughly).

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.

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 Leibniz, Harris, Firth, and Miller.

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 Leibniz, Harris, Firth, and Miller.
- Strictly speaking, similarity of neighbors is neither necessary nor sufficient for semantic similarity.

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 Leibniz, Harris, Firth, and Miller.
- Strictly speaking, similarity of neighbors is neither necessary nor sufficient for semantic similarity.
- But perhaps this is good enough.

Variants of neighbors / cooccurrence

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.
- 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

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.
- 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
  - in a linguistic relationship with each other (e.g., w_1 is a modifier of w_2) or

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.
- 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
    - in a linguistic relationship with each other (e.g., w_1 is a modifier of w_2) or
    - in the same sentence or

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.
- 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
  - 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

Variants of neighbors / cooccurrence

- Two words are neighbors if they cooccur.
- 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
  - 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

Word cooccurrence in Wikipedia: Examples

- corpus = English Wikipedia

Word cooccurrence in Wikipedia: Examples

- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other

Word cooccurrence in Wikipedia: Examples

- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other
  - cooc.(rich,silver) = 186

Word cooccurrence in Wikipedia: Examples

- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other
  - cooc.(rich,silver) = 186
  - cooc.(poor,silver) = 34

Word cooccurrence in Wikipedia: Examples

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

Word cooccurrence in Wikipedia: Examples

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

Word cooccurrence in Wikipedia: Examples

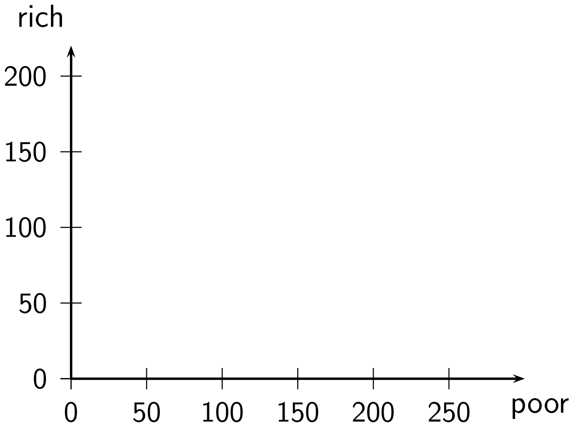
- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other
  - cooc.(rich,silver) = 186
  - cooc.(poor,silver) = 34
  - cooc.(rich,disease) = 17
  - cooc.(poor,disease) = 162
  - cooc.(rich,society) = 143

Word cooccurrence in Wikipedia: Examples

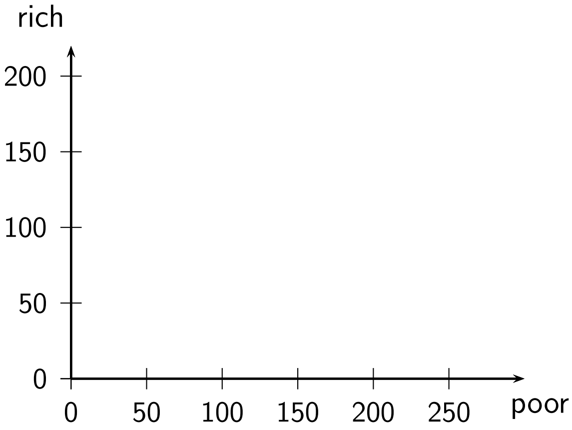
- corpus = English Wikipedia
- cooccurrence defined as occurrence within $k = 10$ words of each other
  - cooc.(rich,silver) = 186
  - cooc.(poor,silver) = 34
  - cooc.(rich,disease) = 17
  - cooc.(poor,disease) = 162
  - cooc.(rich,society) = 143
  - cooc.(poor,society) = 228

Basis for WordSpace: Cooccurrence

Basis for WordSpace: Cooccurrence

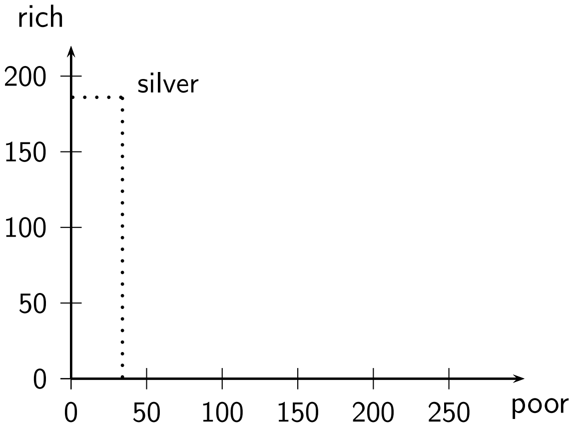


Basis for WordSpace: Cooccurrence



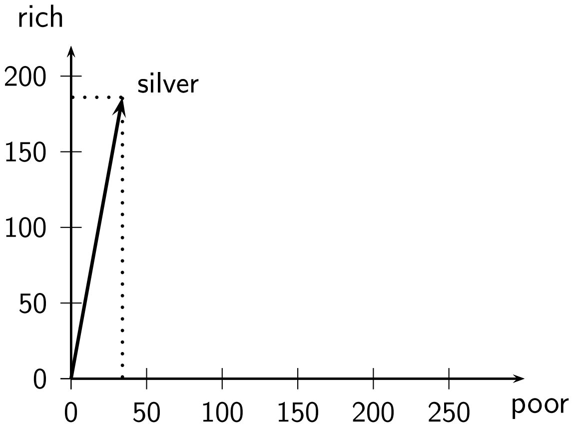
cooc.(poor,silver)=34, cooc.(rich,silver)=186,

Basis for WordSpace: Cooccurrence



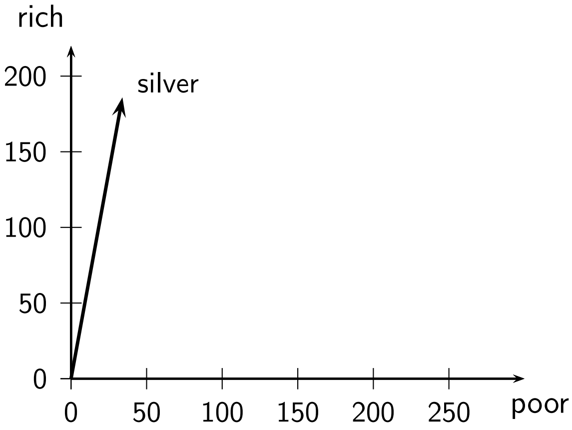
cooc.(poor,silver)=34, cooc.(rich,silver)=186,

Basis for WordSpace: Cooccurrence



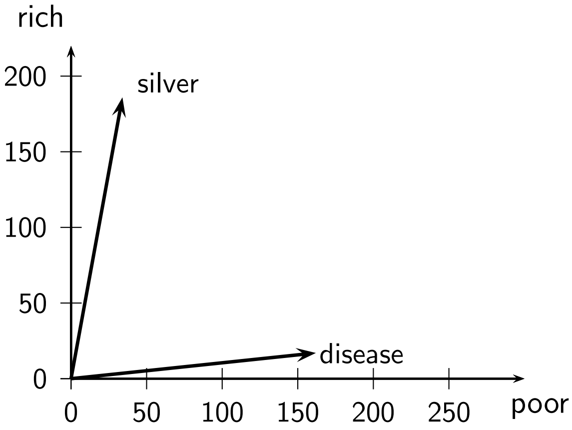
cooc.(poor,silver)=34, cooc.(rich,silver)=186,

Basis for WordSpace: Cooccurrence



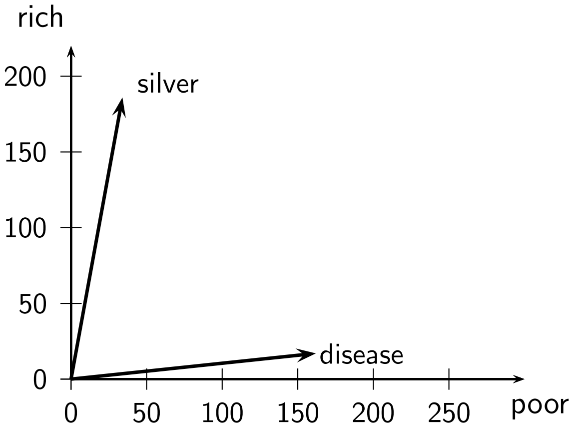
cooc.(poor,silver)=34, cooc.(rich,silver)=186,

Basis for WordSpace: Cooccurrence



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

Basis for WordSpace: Cooccurrence

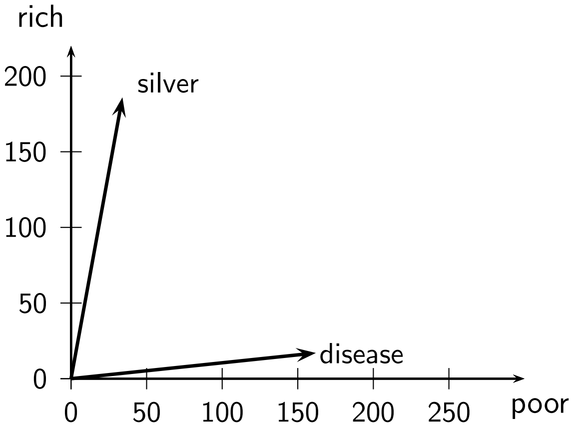


cooc.(poor,silver)=34, cooc.(rich,silver)=186,
cooc.(poor,disease)=162, cooc.(rich,disease)=17,
cooc.(poor,society)=228, cooc.(rich,society)=143

Exercise

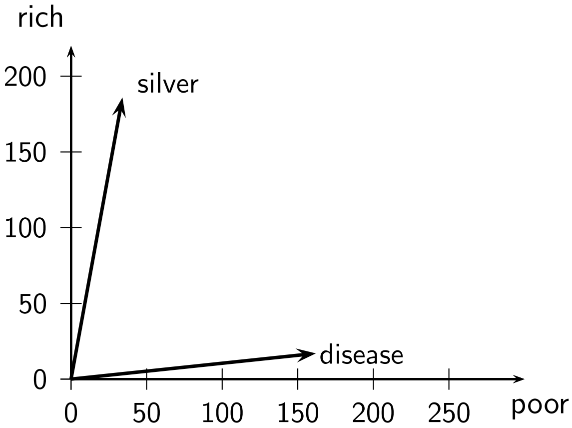
Add “society” to the graph.

Basis for WordSpace: Cooccurrence



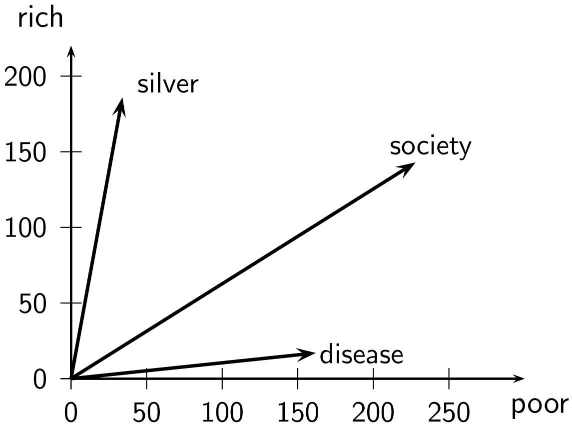
cooc.(poor,silver)=34, cooc.(rich,silver)=186,
cooc.(poor,disease)=162, cooc.(rich,disease)=17,
cooc.(poor,society)=228, cooc.(rich,society)=143

Basis for WordSpace: Cooccurrence



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

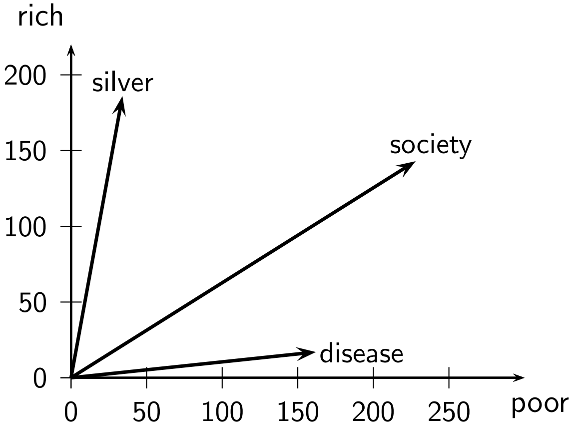
Basis for WordSpace: Cooccurrence



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

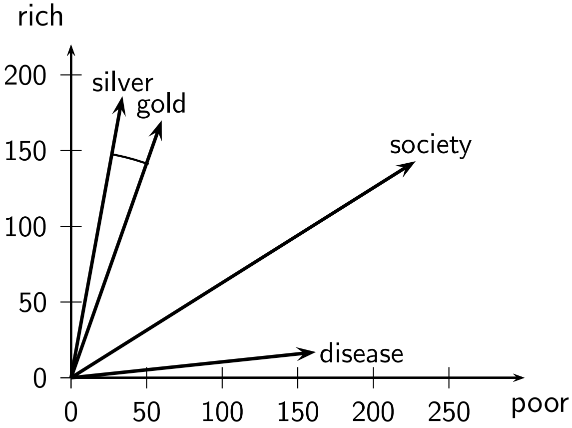
Basis for WordSpace: Cooccurrence$\rightarrow$ Similarity

Basis for WordSpace: Cooccurrence → Similarity



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

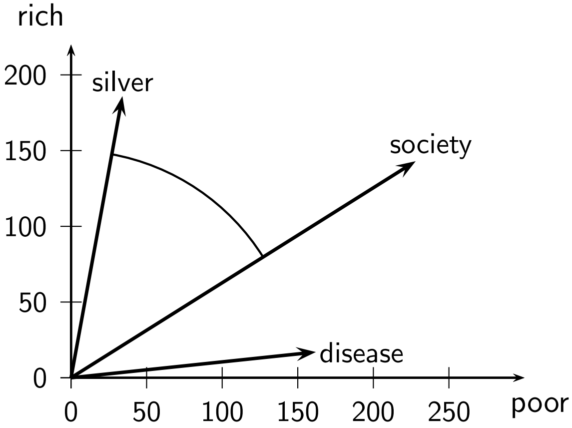
Basis for WordSpace: Cooccurrence→ Similarity



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

Small angle: silver and gold are similar.

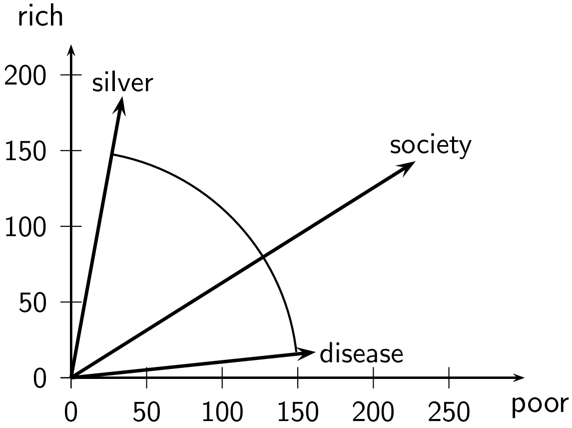
Basis for WordSpace: Cooccurrence→ Similarity



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

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

Basis for WordSpace: Cooccurrence→ Similarity



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

Dimensionality of WordSpace

- Up to now we've only used two dimension words: rich and poor

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.

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.

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.

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.

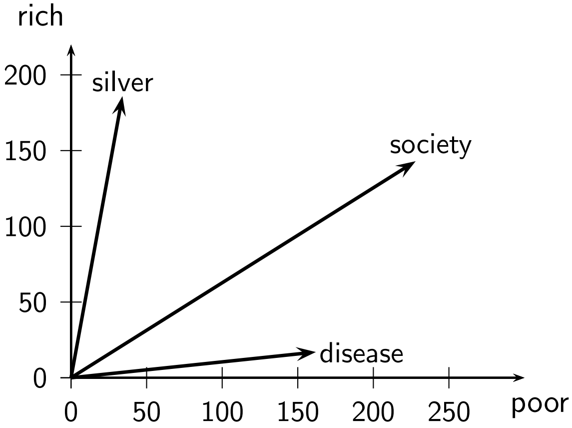
Dimensionality of WordSpace

- 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.

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.

Basis for WordSpace: Cooccurrence



Nearest neighbors of "silver" in WordSpace

Nearest neighbors of “silver” in WordSpace

1.000 silver

Nearest neighbors of "silver" in WordSpace

1.000 silver / 0.865 bronze

Nearest neighbors of “silver” in WordSpace

1.000 silver / 0.865 bronze / 0.842 gold

Nearest neighbors of “silver” in WordSpace

1.000 silver / 0.865 bronze / 0.842 gold / 0.836 medal

Nearest neighbors of “silver” in WordSpace

1.000 silver / 0.865 bronze / 0.842 gold / 0.836 medal / 0.826 medals

Nearest neighbors of "silver" in WordSpace

1.000 silver / 0.865 bronze / 0.842 gold / 0.836 medal / 0.826 medals / 0.761 relay

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Nearest neighbors of "disease" in WordSpace

1.000 disease

Nearest neighbors of “disease” in WordSpace

1.000 disease / 0.858 Alzheimer

Nearest neighbors of “disease” in WordSpace

1.000 disease / 0.858 Alzheimer / 0.852 chronic

Nearest neighbors of “disease” in WordSpace

1.000 disease / 0.858 Alzheimer / 0.852 chronic / 0.846 infectious

Nearest neighbors of “disease” in WordSpace

1.000 disease / 0.858 Alzheimer / 0.852 chronic / 0.846 infectious / 0.843 diseases

Nearest neighbors of “disease” in WordSpace

1.000 disease / 0.858 Alzheimer / 0.852 chronic / 0.846 infectious / 0.843 diseases / 0.823 diabetes

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Wikipedia WordSpace demonstration

Exercise

Exercise

- Find an example word w where WordSpace fails

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*" . . .

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.

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 (i) find the word (ii) explain why it fails

Cases where WordSpace fails

Cases where WordSpace fails

- Antonyms are judged to be similar: “disease” and “cure”

Cases where WordSpace fails

- Antonyms are judged to be similar: "disease" and "cure"
- Ambiguity: "Cambridge"

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”

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”

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"

General framework for embedding learning

General framework for embedding learning

Up to this point: Formalization of WordSpace

Words have two roles: (i) objects located in the space (each gets a vector/embedding) (ii) dimensions of the space

General framework for embedding learning

Up to this point: Formalization of WordSpace

Words have two roles: (i) objects located in the space (each gets a vector/embedding) (ii) dimensions of the space

More general: Dual formalization

We have two different types of objects: (i) primary objects located in the space (usually words, each gets a vector/embedding) (ii) secondary objects (often: contexts, documents, but can also be words, see above)

General framework for embedding learning

Up to this point: Formalization of WordSpace

Words have two roles: (i) objects located in the space (each gets a vector/embedding) (ii) dimensions of the space

More general: Dual formalization

We have two different types of objects: (i) primary objects located in the space (usually words, each gets a vector/embedding) (ii) secondary objects (often: contexts, documents, but can also be words, see above)

Dual formalization:
The roles of primary and secondary objects can be flipped.

General framework for embedding learning

Up to this point: Formalization of WordSpace

Words have two roles: (i) objects located in the space (each gets a vector/embedding) (ii) dimensions of the space

More general: Dual formalization

We have two different types of objects: (i) primary objects located in the space (usually words, each gets a vector/embedding) (ii) secondary objects (often: contexts, documents, but can also be words, see above)

Dual formalization:
The roles of primary and secondary objects can be flipped.
(This is just a first overview.)

"Match score" between primary/secondary object

"Match score" between primary/secondary object

- In these examples: primary object = word

"Match score" between primary/secondary object

- In these examples: primary object = word
- For WordSpace: secondary object = word
  - Match score: cosine, correlation of neighbors

"Match score" between primary/secondary object

- In these examples: primary object = word
- For WordSpace: secondary object = word
    - Match score: cosine, correlation of neighbors
- For LSI (information retrieval): secondary object = document
    - Match score: weighted occurrence count

"Match score" between primary/secondary object

- In these examples: primary object = word
- For WordSpace: secondary object = word
  - Match score: cosine, correlation of neighbors
- For LSI (information retrieval): secondary object = document
  - Match score: weighted occurrence count
- For word2vec skipgram: secondary object = context word
  - Match score: PPMI (see below)

"Match score" between primary/secondary object

- In these examples: primary object = word
- For WordSpace: secondary object = word
  - Match score: cosine, correlation of neighbors
- For LSI (information retrieval): secondary object = document
  - Match score: weighted occurrence count
- For word2vec skipgram: secondary object = context word
  - Match score: PPMI (see below)
- For word2vec cbow: secondary object = sum of context word
  - Match score: cosine

"Match score" between primary/secondary object

- In these examples: primary object = word
- For WordSpace: secondary object = word
  - Match score: cosine, correlation of neighbors
- For LSI (information retrieval): secondary object = document
  - Match score: weighted occurrence count
- For word2vec skipgram: secondary object = context word
  - Match score: PPMI (see below)
- For word2vec cbow: secondary object = sum of context word
  - Match score: cosine

(This is just a first overview.)

General framework for embedding learning

Training objective

Minimize $\sum_{(w,c)} |\vec{w}\vec{c} - \text{match-score}(w, c)|$

General framework for embedding learning

Training objective

Minimize $\sum_{(w,c)} |\vec{w}\vec{c} - \text{match-score}(w, c)|$

(This is just a first overview.)

Dot product / Skalarprodukt

$$\vec{w}\vec{c} = \sum_i w_i c_i$$

Example:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = w_1 c_1 + w_2 c_2 + w_3 c_3$$

Outline

1. Distributional semantics
2. WordSpace
3. Norms & scores

How to make WordSpace work well: Two important details

How to make WordSpace work well: Two important details

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

How to make WordSpace work well: Two important details

- Norms:
  When comparing vectors,
  we often want to normalize them first.
- Scores:
  Designing the right matching score can be critical.

Norms: Let's look at WordSpace again

Norms: Let's look at WordSpace again

- How do we formalize semantic similarity in the vector space?

Norms: Let's look at WordSpace again

- How do we formalize semantic similarity in the vector space?
- First cut: (negative) distance between two points

Norms: Let's look at WordSpace again

- How do we formalize semantic similarity in the vector space?
- First cut: (negative) distance between two points
- ($=$ distance between the end points of the two vectors)

Norms: Let's look at WordSpace again

- How do we formalize semantic similarity in the vector space?
- First cut: (negative) distance between two points
- (= distance between the end points of the two vectors)
- Euclidean distance?

Norms: Let's look at WordSpace again

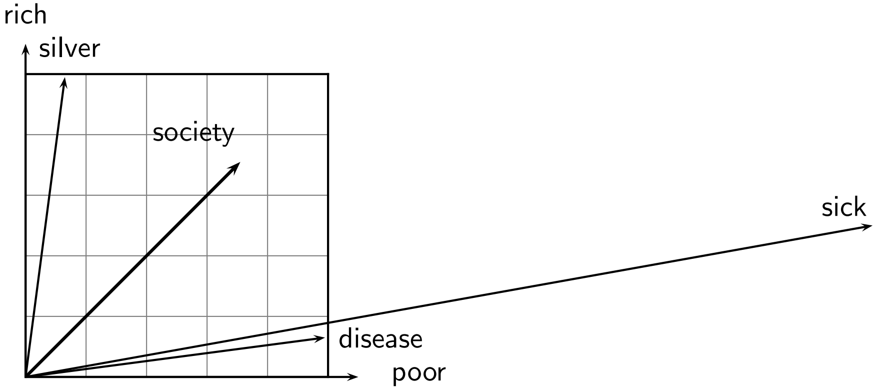
- How do we formalize semantic similarity in the vector space?
- First cut: (negative) distance between two points
- ($=$ distance between the end points of the two vectors)
- Euclidean distance?
- Euclidean distance is a bad idea ...

Norms: Let's look at WordSpace again

- How do we formalize semantic similarity in the vector space?
- First cut: (negative) distance between two points
- ($=$ distance between the end points of the two vectors)
- Euclidean distance?
- Euclidean distance is a bad idea . . .
- . . . because Euclidean distance is
  large for vectors of different lengths.

Why distance is a bad idea

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 that “disease”.

Distance is bad as a similarity measure: How do we fix this?

- There are two equivalent ways of doing this.
- Use distance of length-normalized vectors as similarity measure
- Use angle/cosine of (unnormalized) vectors as similarity measure

Use angle instead of distance

Use angle instead of distance

- Measure similarity as the angle between word vectors.

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- Thought experiment: Suppose that for a particular corpus we have vector $\vec{w}$ for word w.

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- Thought experiment: Suppose that for a particular corpus we have vector $\vec{w}$ for word w.
- Double the size of the corpus by appending it to itself. Compute vector $\vec{w}'$ for word w on new corpus.

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- Thought experiment: Suppose that for a particular corpus we have vector $\vec{w}$ for word w.
- Double the size of the corpus by appending it to itself. Compute vector $\vec{w}'$ for word w on new corpus.
- $\vec{w}$ and $\vec{w}'$ are semantically identical.

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- Thought experiment: Suppose that for a particular corpus we have vector $\vec{w}$ for word w.
- Double the size of the corpus by appending it to itself. Compute vector $\vec{w}'$ for word w on new corpus.
- $\vec{w}$ and $\vec{w}'$ are semantically identical.
- The angle between the two vectors is close to 0, corresponding to maximal similarity ...

Use angle instead of distance

- Measure similarity as the angle between word vectors.
- Thought experiment: Suppose that for a particular corpus we have vector $\vec{w}$ for word w.
- Double the size of the corpus by appending it to itself. Compute vector $\vec{w}'$ for word w on new corpus.
- $\vec{w}$ and $\vec{w}'$ are semantically identical.
- The angle between the two vectors is close to 0, corresponding to maximal similarity ...
- ... even though the Euclidean distance between the two vectors is large.

From angles to cosines

From angles to cosines

- The following two notions are equivalent.

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.

From angles to cosines

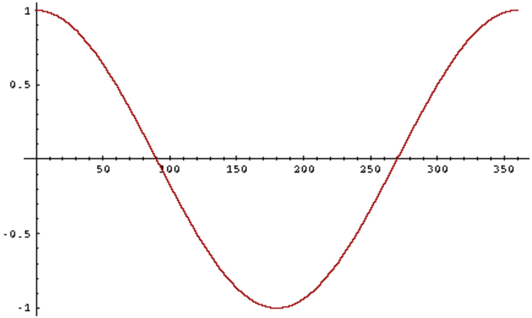
- The following two notions are equivalent.
  - Rank words w_i according to the angle between w_i and a target word v in decreasing order.
  - Rank words w_i according to cosine(w_i,v) in increasing order

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 $[0^\circ, 180^\circ]$

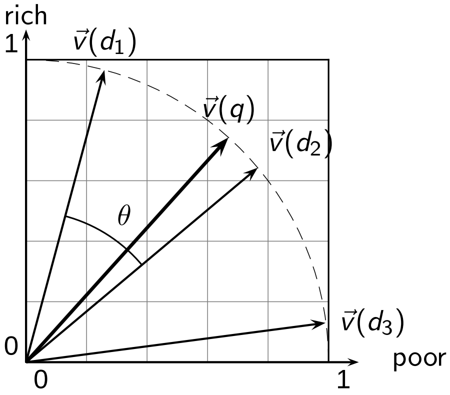
Cosine

Cosine



Cosine similarity illustrated

Cosine similarity illustrated



Cosine similarity between two words

Cosine similarity between two words

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

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

Cosine similarity between two words

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

- $|\vec{c}|$ and $|\vec{d}|$ are the lengths of $\vec{c}$ and $\vec{d}$.
- This is the cosine similarity of $\vec{c}$ and $\vec{d}$ . . . . . . or, equivalently, the cosine of the angle between $\vec{c}$ and $\vec{d}$.

Distance is bad as a similarity measure: How do we fix this?

- There are two equivalent ways of doing this.
- Use distance of length-normalized vectors as similarity measure
- Use angle/cosine of (unnormalized) vectors as similarity measure

Length normalization

Length normalization

- How do we compute the cosine?

Length normalization

- How do we compute the cosine?
- A vector can be (length-) normalized by dividing each of its components by its length – here we use the L_2 norm: $||x||_2 = \sqrt{\sum_i x_i^2}$

Length normalization

- How do we compute the cosine?
- A vector can be (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 . . .

Length normalization

- How do we compute the cosine?
- A vector can be (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: $||x||_2 = \sqrt{\sum_i x_i^2} = 1.0$

Length normalization

- How do we compute the cosine?
- A vector can be (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.

Length normalization

- How do we compute the cosine?
- A vector can be (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 two word vectors $\vec{w}$ and $\vec{w'}$ (based on a simple corpus and a second twice the size of the original) from earlier slide: they have almost identical vectors after length-normalization.

Cosine for normalized vectors

- For normalized vectors, the cosine is equivalent to the dot product or scalar product.
- $\cos(\vec{q}, \vec{d}) = \vec{q} \cdot \vec{d} = \sum_i q_i \cdot d_i$
  - (if $\vec{q}$ and $\vec{d}$ are length-normalized).

Cosine similarity: Summary and example

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

	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

Distance is bad as a similarity measure: How do we fix this?

- There are two equivalent ways of doing this.
- Use distance of length-normalized vectors as similarity measure
- Use angle/cosine of (unnormalized) vectors as similarity measure

How to make WordSpace work well: Two important details

- Norms:
  When comparing vectors,
  we often want to normalize them first.
- Scores:
  Designing the right matching score can be critical.

PMI: Main matching score we will use here

- PMI: pointwise mutual information
- $\mathrm{PMI}(w, c) = \log \frac{P(wc)}{P(w)P(c)}$
- If w, c independent: $\mathrm{PMI}(w, c) = 0$
- If w, c perfectly correlated: $\mathrm{PMI}(w, c) = \log 1/P(c)$
- If w, c positively correlated: $\mathrm{PMI}(w, c)$ is large and positive.
- If w, c negatively correlated: $\mathrm{PMI}(w, c)$ is large and negative.

PMI: Main matching score we will use here

- PMI: pointwise mutual information
- $\text{PMI}(w, c) = \log \frac{P(wc)}{P(w)P(c)}$
- If w, c independent: $\text{PMI}(w, c) = 0$
- If w, c perfectly correlated: $\text{PMI}(w, c) = \log 1/P(c)$
- If w, c positively correlated: $\text{PMI}(w, c)$ is large and positive.
- If w, c negatively correlated: $\text{PMI}(w, c)$ is large and negative.
- We are replacing cooccurrence (raw counts) with a measure of surprise (PMI).

PPMI

- PPMI = positive pointwise mutual information
- $\text{PPMI}(w, c) = \max(0, \text{PMI}(w, c))$
- More generally (with offset k): $\text{PPMI}(w, c) = \max(0, \text{PMI}(w, c) - \text{k})$

PPMI: Motivation

- 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
dimensions	take	100	10000	10000
	rich	4	400	100
	poor	1	100	400

Cooccurrence count matrix: Cosine, no PPMI

		vectors		
		rhodium	gold	disease
dimensions	take	100	10000	10000
	rich	4	400	100
	poor	1	100	400

	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
dimensions	take	100	10000	10000
	rich	4	400	100
	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

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

$C(w)$	$C(c)$	$C(wc)$	PMI (use $\log_{10}$)
100	100	1	?
100	100	100	?
5000	5000	250	?

(total = 10000)

Bag of words model

Bag of words model

- We do not consider the order of words in a context.

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 $k = 10$.

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.

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?

Limits of distributional semantics?

- Taxonomies
  - fruit - reproductive structure - plant organ - plant part - natural object - whole/unit
  - seafood - food - nutrient - substance - matter

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.

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

Takeaway
Distributional semantics

- The meaning of a word is learned from its contexts in a large corpus.

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.

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/relatedness.

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/relatedness.
- There is a lot more in semantics that distributional semantics is not a good model for.

Takeaway
WordSpace

Takeaway
WordSpace

- The representation/embedding of a word is a vector of cooccurrence counts.

Takeaway
WordSpace

- The representation/embedding of a word is
  a vector of cooccurrence counts.
- Semantic similarity/relatedness
  is measured as cosine of cooccurrence vectors.

Takeaway
WordSpace

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

Takeaway
Norms & Scores

Takeaway
Norms & Scores

- Euclidean distance is not a good measure of semantic relatedness in WordSpace.

Takeaway
Norms & Scores

- Euclidean distance is not a good measure of semantic relatedness in WordSpace.
- Cosine is appropriate because it implicitly normalizes for length and (global) frequency.

Takeaway
Norms & Scores

- Euclidean distance is not a good measure of semantic relatedness 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.