Semantic Analysis: Word Similarity

Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

Semantic Similarity

Let's consider the ambiguity of similarity:

Think of yourself as being labeled "physically similar" to someone else.

You could be perceived as similar in:

- Height
- Weight
- BMI
- Hair color
- Eye color
- Skin color
- Nose shape
- Ear size
- Etc.

Semantic Similarity

Likewise there are various “dimensions” of semantic similarity:

- Word similarity
- Sense similarity
- Text similarity
- Taxonomy similarity
- Frame similarity
- Context similarity

Word Similarity

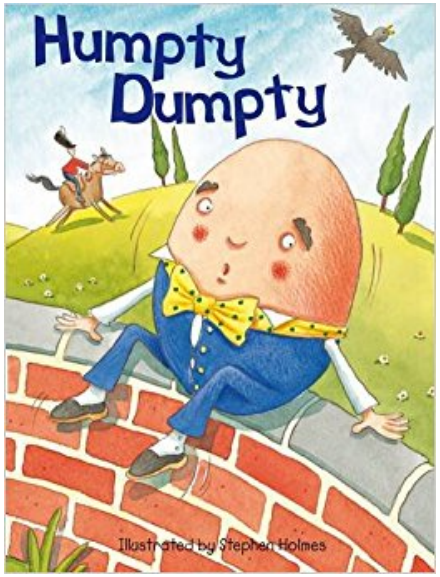
Approaches to measuring word similarity

- Statistical approaches—how closely associated are two words in a corpus?
  - PPMI ("positive pointwise mutual information")
  - Vector semantics and LSA ("latent semantic analysis")
  - Cosine similarity
- Structural approaches—how close are two words within a semantic graph?
    - Ontological distance
    - Overlap of parse contexts

Dependent Word Probabilities

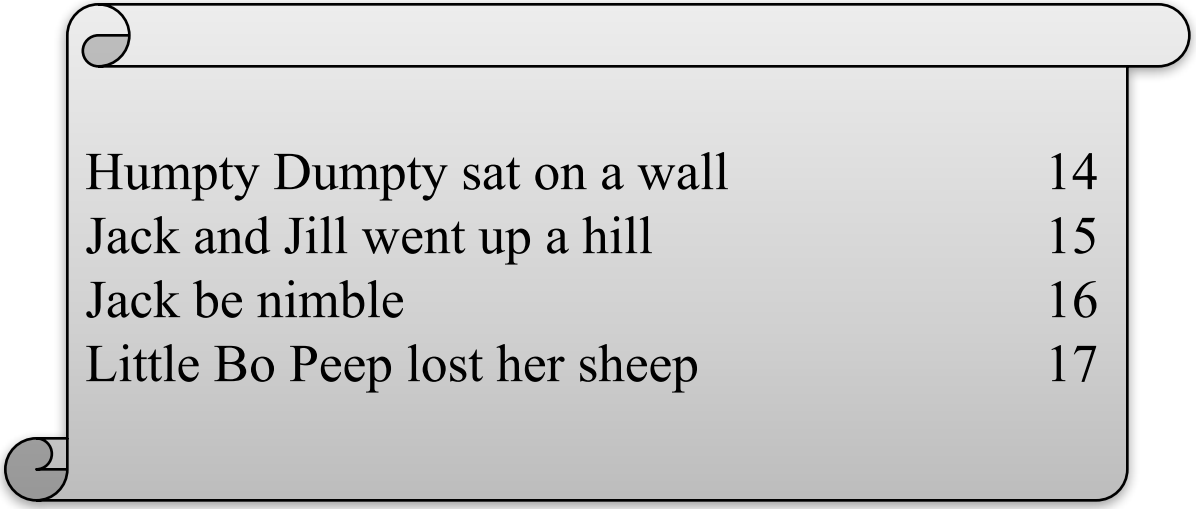
Let's consider the word probabilities of a collection of nursery rhymes that includes:

- Humpty Dumpy
- Jack and Jill
- Jack Be Nimble
- Little Bo Peep



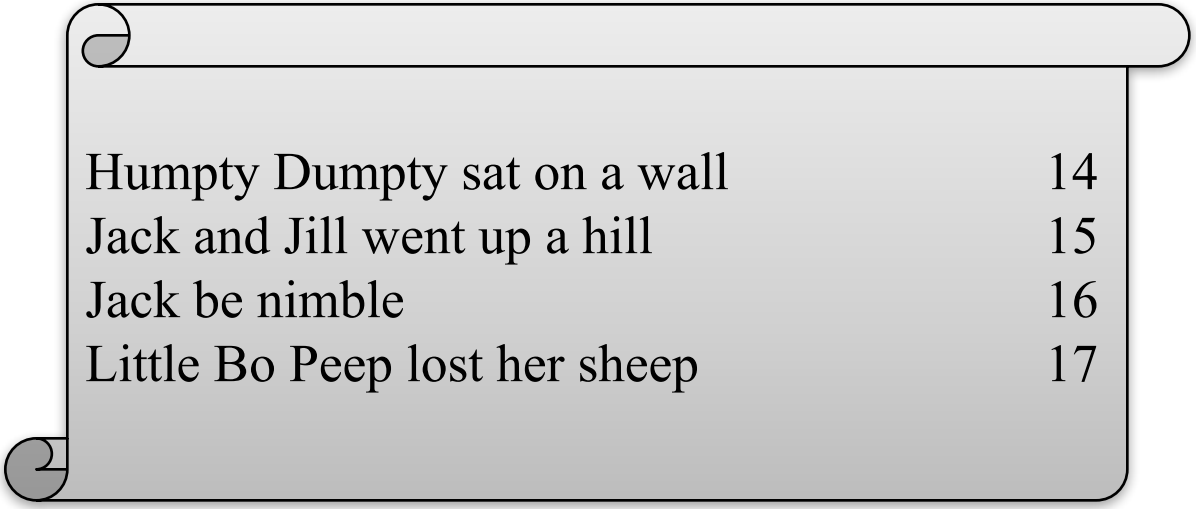
Dependent Word Probabilities

- Suppose that “Jack” has a probability of 0.04 in our corpus of children’s nursery rhymes, and “Jill” 0.02.
- If the two words are *independent* (not associated), then the probability of them co-occurring would be:
- 0.04 x 0.02 = 0.0008 ← *very low!*



Dependent Word Probabilities

- But let's suppose that *when* "Jack" occurs, "Jill" occurs 50% of the time.
  So the actual probability of co-occurrence is:
  0.04 x 0.5 = 0.02 ← *much higher than the previous calculation!*
- We say that the *dependent* probability is higher than the *independent* probability.



PMI

- Finally, let's take a log function of the ratio of dependent to independent probabilities of this word pair. Where x = "Jack" and y = "Jill":

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

In our example, this would give us a result of about 4.6—a nicely positive PMI. PMI is the "pointwise mutual information" measure, and a positive PMI means the words are related (associated).

PPMI

PMI can range from -∞ to ∞

- The negative values here make people uncomfortable.
  - Harder to normalize
  - Implies increasing (to infinity) “unrelatedness”
  - Counterintuitive. People take it like this: Imagine if zero represented “not at all pregnant,” then what would -1 be? “Even more not pregnant?”
- So in practice, we replace all negative outcomes with a zero—this is called the “positive pointwise mutual information,” or PPMI.

DataScience@SMU

Semantic Analysis: Vector Semantics

Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

Vector Semantics

- Vector semantics (a.k.a. distributional semantics) can be used to judge word similarity as well as text similarity.
- A vector represents a distribution of other features (usually other words) found in the same context as each target word.
- This approach is inspired by a famous quote from J. R. Firth from the 1950s: “a word is characterized by the company it keeps.”

Vector Semantics

A term-document matrix:

- Two documents are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Vector Semantics

A term-document matrix:

- Two **words** are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

In a real application, instead of the raw counts above, we would likely use tf-idf.

Vector Semantics

Instead of having the columns represent documents, we could have them represent *context words* (e.g., words occurring within a ±10 word window of each target word throughout the corpus).

Vector Semantics

A term-context matrix:

- Two **words** are similar in meaning if their context vectors are similar

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

In a real application, we would use PPMI instead of these raw counts.

A Matter of Practicality

- That example was a 4x6 grid.
- In a real-world application, it would be 50,000 x 50,000.
- And the vectors would be *sparse* rather than *dense*, meaning most of the values would be 0.

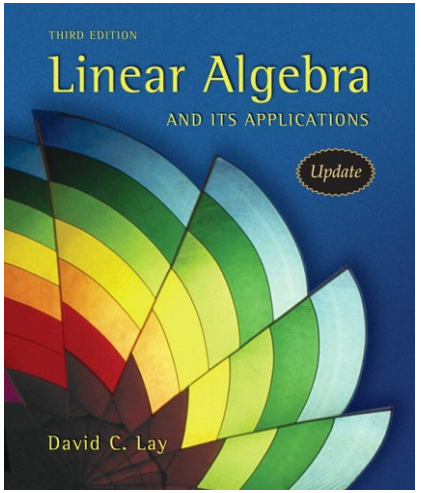
A Matter of Practicality

For this reason we have algorithms to "reduce the dimensions" of the vector space to a more manageable number (e.g., about 300) where the variance between the values is the greatest (hence the vectors are the *most informative*).

The most famous such algorithm is LSA ("latent semantic analysis").

LSA

- If you want to know the ins and outs of LSA, dust off your linear algebra textbook and refresh yourself on matrix multiplication and SVD (singular value decomposition).
- From an application builder's perspective, it's a tool that does this:



Value of LSA

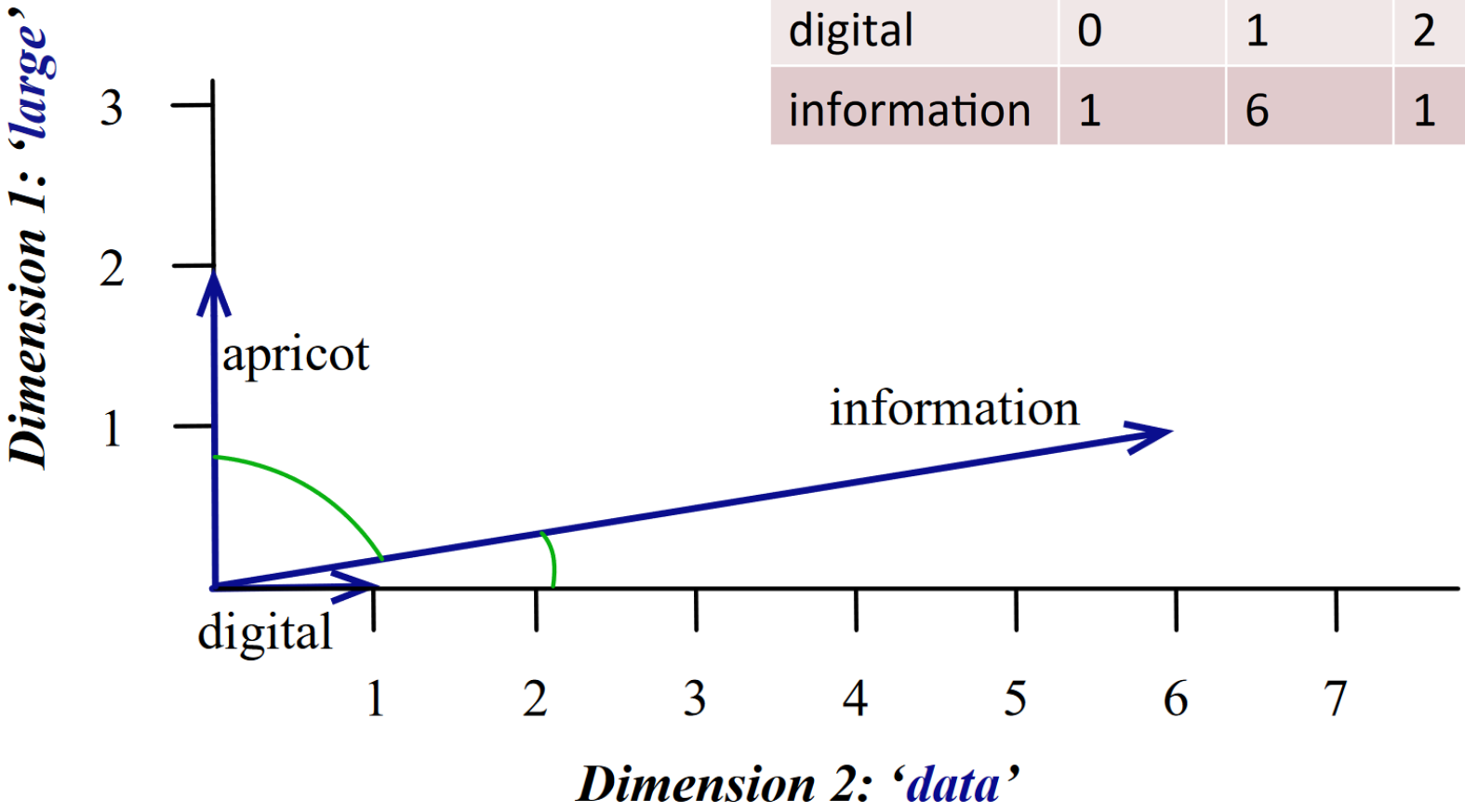
- The smaller number of dense vectors is valuable—it tells which words are most associated by vector semantics without needing huge vectors.

Limits of LSA

- LSA is still subject to “garbage-in, garbage-out.”
- An early implementation of LSA by its inventor yielded a stronger similarity score between “nurse” and “doctor” than between “physician” and “doctor.”
- This was of course due to the distribution of these words in the corpus.
  Be forewarned!

Visualizing Vector Semantics

	large	data	computer
apricot	1	0	0
digital	0	1	2
information	1	6	1



Cosine Similarity

$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}\bullet\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|}\bullet\frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

	large	data	computer
apricot	1	0	0
digital	0	1	2
information	1	6	1

Which pair of words is more similar?

cosine(apricot,information) = $\frac{1+0+0}{\sqrt{1+0+0}\ \sqrt{1+36+1}} = \frac{1}{\sqrt{38}} = .16$

cosine(digital,information) = $\frac{0+6+2}{\sqrt{0+1+4}\ \sqrt{1+36+1}} = \frac{8}{\sqrt{38}\sqrt{5}} = .58$

cosine(apricot,digital) = $\frac{0+0+0}{\sqrt{1+0+0}\ \sqrt{0+1+4}} = 0$

DataScience@SMU

Semantic Analysis:
Structural Approaches to Word Similarity

Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

Similarity of Parse Contexts

Let's reuse our syntax parse trees to determine contexts for comparing words!

- Two words are similar if they have similar parse contexts
- **Duty** and **responsibility** (Chris Callison-Burch's example)

Modified by adjectives	additional, administrative, assumed, collective, congressional, constitutional …
Objects of verbs	assert, assign, assume, attend to, avoid, become, breach …

Example Result

- This gives us better (more useful) results than straight PPMI.

Object of "drink"	Count	PMI
tea	2	11.8
liquid	2	10.5
wine	2	9.3
anything	3	5.2
it	3	1.3

- "Drink it" more common than "drink wine"
- But "wine" is a better "drinkable" thing than "it"

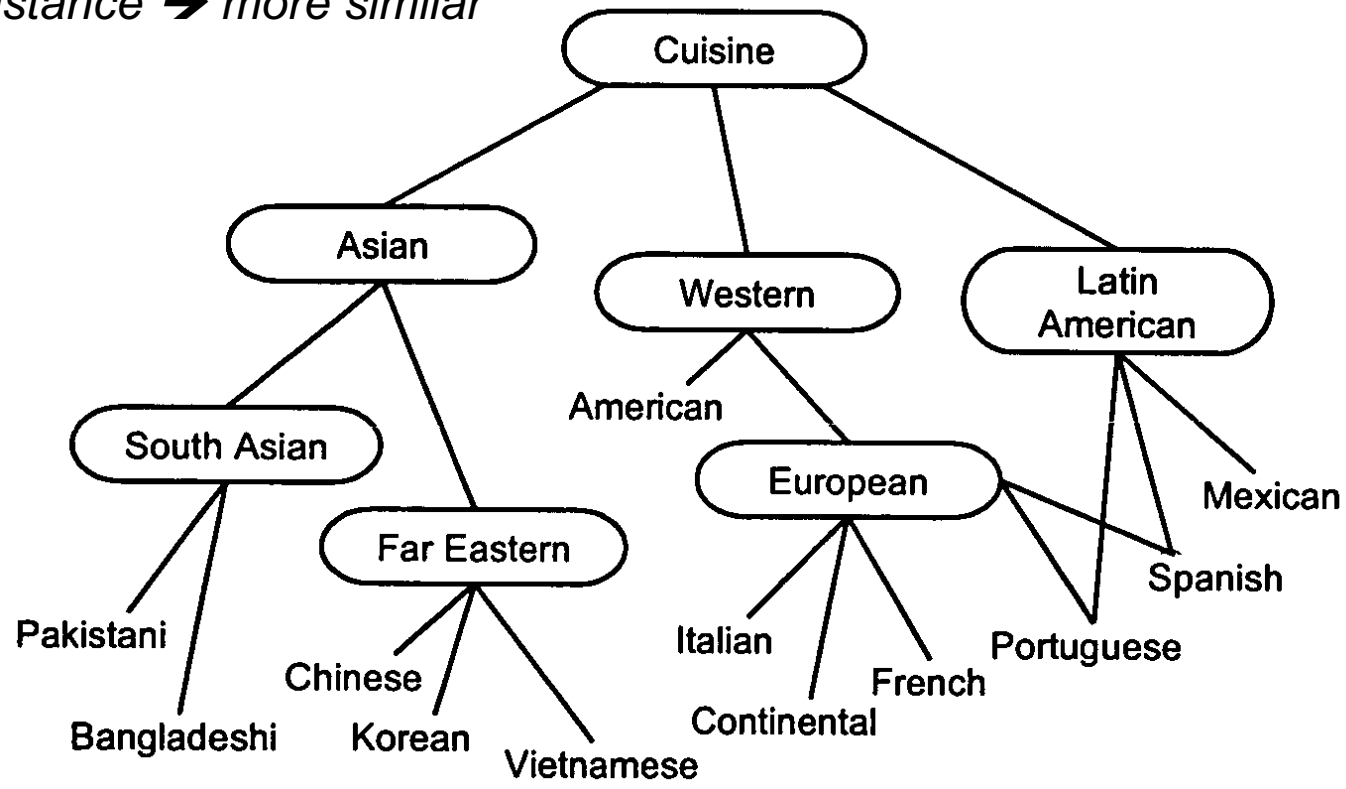
Don Hindle, 1990

Word Similarity as Ontological Distance

Distance French–Chinese = 6

Distance French–Italian = 2

Shorter distance ➔ more similar



DataScience@SMU

Semantic Analysis: Document Similarity

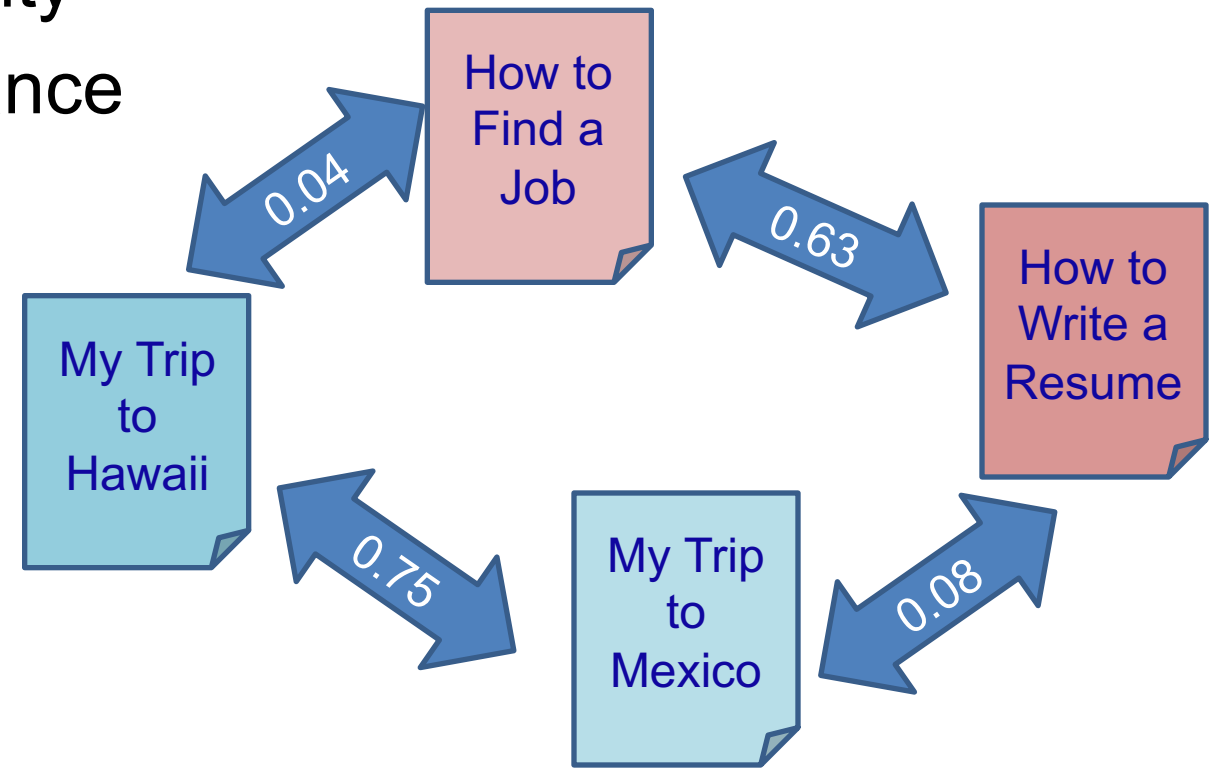
Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

Document Similarity

Methods of measuring document similarity

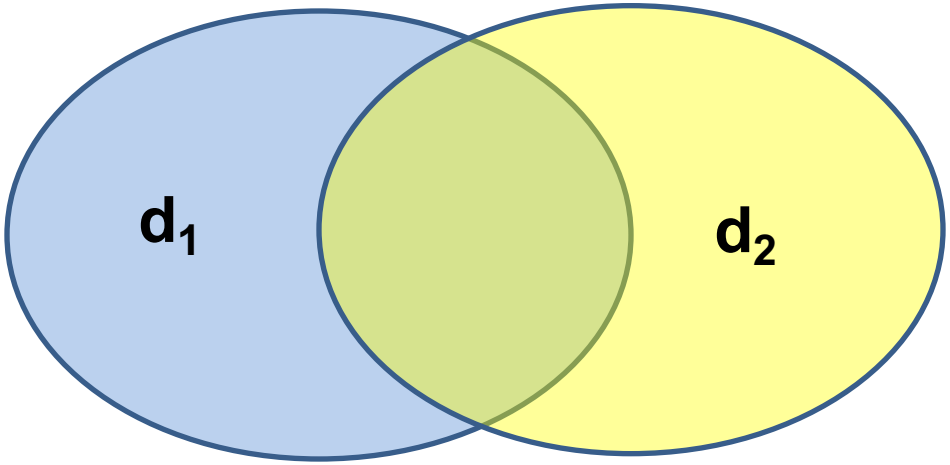
- Jaccard distance
- Cosine similarity
- Hellinger distance
- Many more…



Jaccard Similarity

Measures how many terms the two documents share, compared to the total vocabulary of both documents (i.e., the intersection of their terms compared to their union).

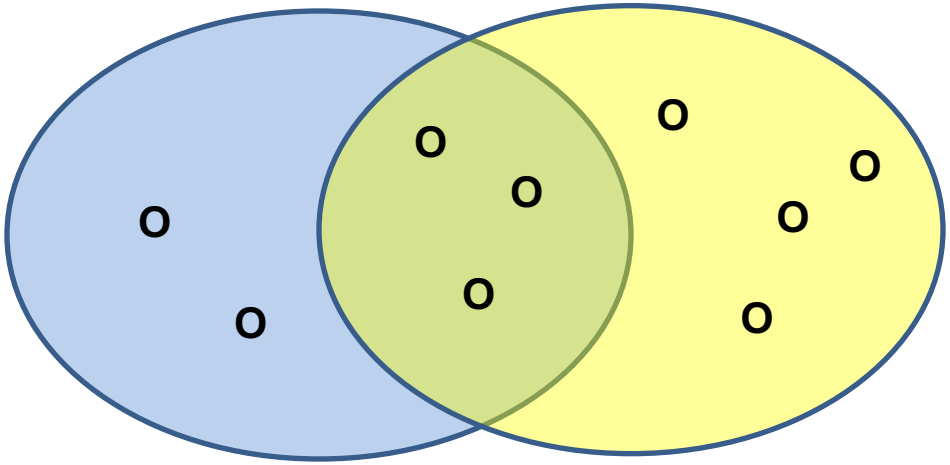
$$\mathtt{jac\_sim}(d_1, d_2) = (d_1 \cap d_2) \,/\, (d_1 \cup d_2)$$



Jaccard Similarity

Measures how many terms the two documents share, compared to the total vocabulary of both documents (i.e., the intersection of their terms compared to their union).

$$\texttt{jac\_sim}(d_1, d_2) = (d_1 \cap d_2) / (d_1 \cup d_2)$$

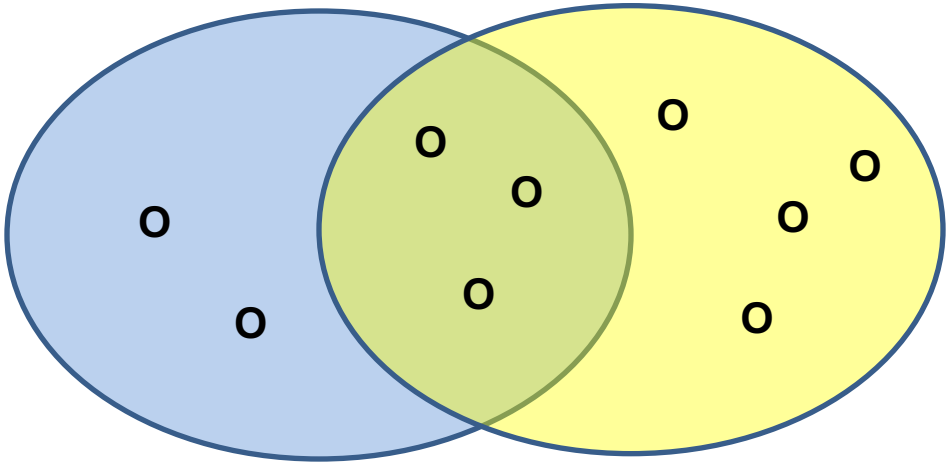


What's the Jaccard similarity here?

Jaccard Similarity

Measures how many terms the two documents share, compared to the total vocabulary of both documents (i.e., the intersection of their terms compared to their union).

$$\mathbf{jac\_sim}(d_1, d_2) = (d_1 \cap d_2) \,/\, (d_1 \cup d_2)$$



3 in intersection
9 in union
jac_sim = 3/9

Jaccard Similarity

In Python (assuming you have term vectors with Boolean values indicating the presence of each term in the term vector):

```
>>> def jac_sim(x,y):
...     x = np.asarray(x, np.bool)
...     y = np.asarray(y, np.bool)
...     return np.double(np.bitwise_and(x,y).sum()) / np.double(np.bitwise_or(x,y).sum())
...
>>> jac_sim([1,0,1,0,1], [1,1,1,0,0])
0.5
>>>
```

D_1 = “See Spot run.”
D_2 = “See Rover run.”
Term vector labels = “see”, “rover”, “run”, “to”, “spot”

We Do Better than That

Jaccard similarity pays no mind to the frequency of each term, so let's look further…

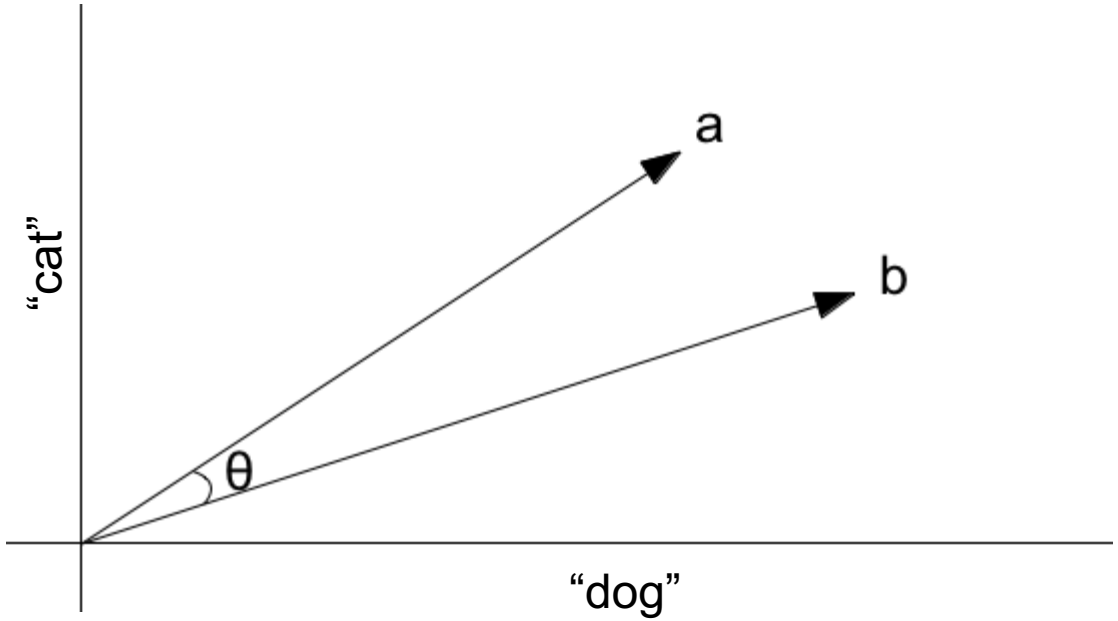
Using Cosine Similarity

Remember cosine similarity?

- We used this previously for word similarity, and now we can use it at the document level.
- It's relatively efficient to evaluate on sparse vectors (as only nonzero dimensions are considered).
- Long sparse vectors are exactly what we have when we generate tf-idf vectors for documents across a broad vocabulary.
- And it autonormalizes to document length.

What It Does

Visualizing the cosine similarity of documents *a* and *b*, if we only had two terms (thus two dimensions):



Cosine Similarity

Consider the raw term frequencies of some documents:

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

Cosine similarity of two documents, whose vectors are d_1 and d_2, is defined as:

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / \| d_1 \| \| d_2 \|$$

where $\bullet$ is the dot product of the vectors, and $\| d_1 \|$ is the length of the vector d_1 (Euclidean distance in n-dimensional vector-space where n is the number of terms).

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

So to compare the first two documents, we do the following:

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

So to compare the first two documents, we do the following:

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

So to compare the first two documents, we do the following:

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$

$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$

$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$

$\cos(d_1, d_2) = 0.94$

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

So to compare the first two documents, we do the following:

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

So to compare the first two documents, we do the following:

$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$
$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$
$\cos(d_1, d_2) = 0.94$

Cosine Similarity

Text	question	socrates	triangle	drawing	slave	answer	boy	virtue	games	pleasure
Doc1	5	0	3	0	2	0	0	2	0	0
Doc2	3	0	2	0	1	1	0	1	0	1
Doc3	0	7	0	2	1	0	0	3	0	0
Doc4	0	1	0	0	1	2	2	0	3	0

$\cos(d_1, d_2) = 0.94$

Wow, the first two documents are very similar!

Remember, cosine similarity ranges 0 to 1, with 1 meaning the documents have identical term frequencies, and 0 meaning they have *no* words in common.

For fun, compute cosine similarity manually on a different pair of documents from the table!

For simplicity we used simple term frequencies here, but in practice we would use tf-idf.

Cosine Similarity

In Python:

```
from math import*

def vec_len(x):
    return round(sqrt(sum([a*a for a in x])),3)

def dot_prod(x,y)
    return sum(a*b for a,b in zip(x,y))


def cos_sim(x,y):
    return round(dot_prod(x,y)/float(vec_len(x)*vec_len(y)),3)
```

```
cos_sim([1,3,3,6,0,1], [12,2,1,19,1,1])
```

returns

```
0.803
```

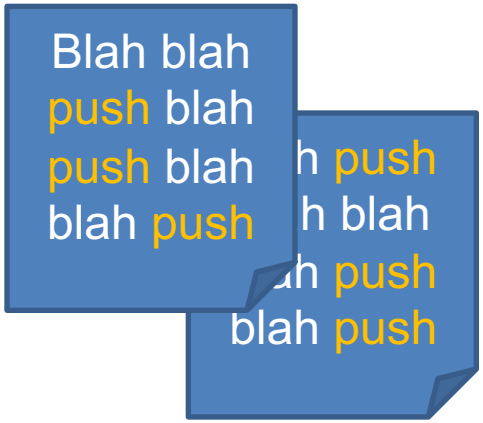
About These Values

Why would we want to use tf-idf values instead of just tf values?

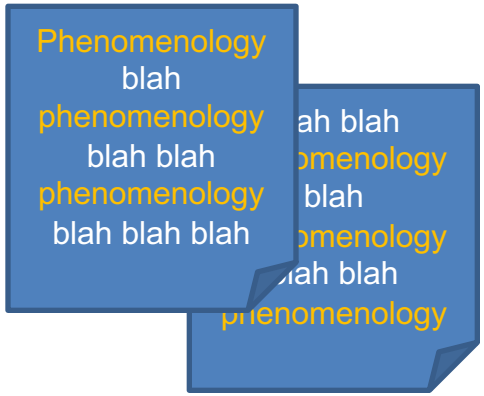
About These Values

Why would we want to use tf-idf values instead of just tf values?

- Consider a pair of documents that both use the word “push” three times.
- Another pair of documents uses the word “phenomenology” three times.
- All other things being equal, each pair has the same cosine similarity.



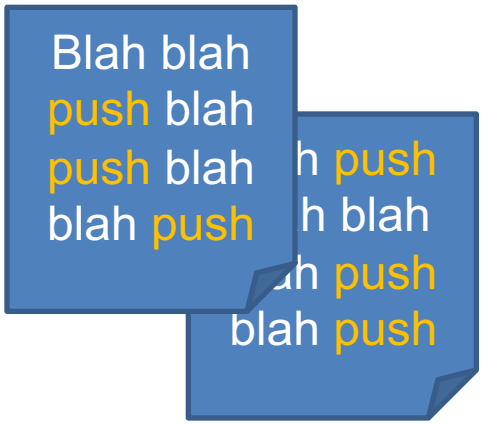
sim: 0.802



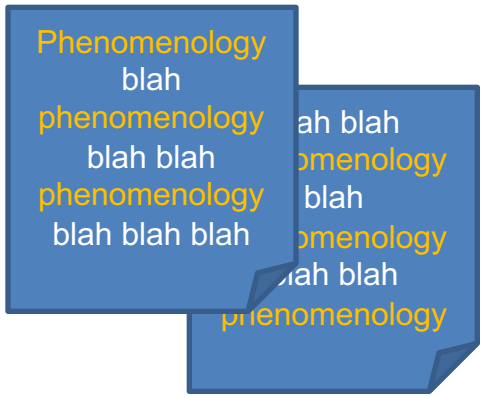
sim: 0.802

About These Values

- But that's not right, is it?
- If we substitute tf-idf for tf in the vectors, the second pair of documents will have a much higher cosine similarity—exactly what we want.



sim: 0.640



sim: 0.921

DataScience@SMU

Semantic Analysis: Document Similarity—Documents as Probability Distributions

Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

Documents as Probability Distributions

A ***discrete probability distribution*** denotes possible states of affairs, such as the results of flipping a coin:

	heads	tails
coin	0.5	0.5

Documents as Probability Distributions

It is possible to construe *documents* as discrete probability distributions.
Take:

```
text1 = "Jack jumped over the candlestick."
text2 = "The cow jumped over the moon."
text3 = "Jack and Jill went up the hill."
```

Now instead of flipping a coin, imagine each text is a bag of words, from which you randomly pull out a word.

Documents as Probability Distributions

This means there's a straightforward probability distribution:

	and	candlestick	cow	hill	jack	jill	jumped	moon	over	the	up	went
text1	0.00	0.20	0.00	0.00	0.20	0.00	0.20	0.00	0.20	0.20	0.00	0.00
text2	0.00	0.00	0.17	0.00	0.00	0.00	0.17	0.17	0.17	0.33	0.00	0.00
text3	0.14	0.00	0.00	0.14	0.14	0.14	0.00	0.00	0.00	0.14	0.14	0.14

In practice, we'll use tf-idf values in place of these raw term probabilities. (Remember tf-idf?)

Document Similarity

Now we can employ any method of measuring distance between probability distributions as a measure of document similarity.
For example, Hellinger distance:

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2},$$

$\boldsymbol{P}$ and $\boldsymbol{Q}$ represent the pair of documents we wish to measure, such that $\boldsymbol{p}$ and $\boldsymbol{q}$ are the tf-idf vectors of those documents.

Which One to Use?

- Besides cosine similarity and Hellinger distance, there is BM25, and there exist many other document similarity measures, and people are still inventing more.
- It's always best to try out a couple of different ones for each application and find out which works best for you.

A Weakness of These Methods

These two sentences have a low similarity using any of the preceding measures:

"Doctors commonly receive additional instruction to keep up with new investigations in medicine."

"Physicians usually participate in supplementary education programs covering the latest research."

How could we tackle that problem?

A Weakness of These Methods

These two sentences have a low similarity:

```
"Doctors commonly receive additional instruction to keep
up with new investigations in medicine."

"Physicians usually participate in supplementary education
programs covering the latest research."
```

Consider normalizing the text to synsets and/or hypernym trees, and use vectors of these for computing cosine similarity (or another measure).

DataScience@SMU

Applications of Semantic Similarity

Natural Language Processing

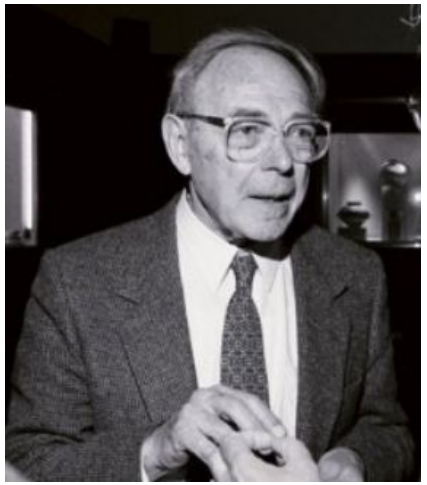
Some slides after Carpuat 2016, Jurafsky 2014

First, a Warning

The ideology of vector semantics dictates that synonymy *is* vector match.

Some famous quotes to this effect:

- Zellig Harris (1954)
  "oculist and eye doctor...occur in almost the same environments…. If A and B have almost identical environments, we say that they are synonyms."
- And remember the quote from J. R. Firth: "You shall know a word by the company it keeps!"



Similarity ≠ Synonymy

However, in practice, we must be careful invoking vector semantics as a technique for detecting synonymy. It doesn't automatically ring true.

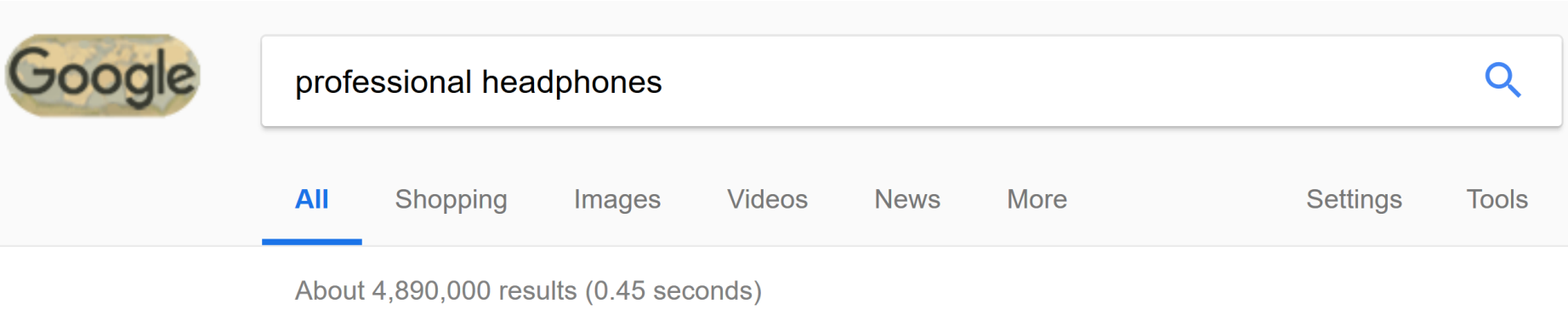
Similarity ≠ Synonymy

One thing to watch out for is collocation in key terms.

Boot + Camp ≠ Boot Camp

Using Semantic Similarity in Search Engines

Let’s take an example.



What do you suppose the synonymous hits should be?

Semantic Similarity in Search Engines

Google hits “earphones” as a synonym of the single word “headphones”…

The Best Studio Headphones of 2018 | PCMag.com
https://www.pcmag.com › Reviews › Consumer Electronics › Audio › Headphones ▾
Mar 7, 2018 - Bottom Line: The stunning Etymotic ER4 XR **earphones** deliver the sonic accuracy sound **professionals** need, and add some subtle depth in the ...

… but doesn’t see “studio headphones” as a synonym of the phrase “professional headphones”—instead it keeps hitting “headphones” and “professional” as separate words:

Studio Headphones VS Consumer Headphones VS Gaming Headsets ...
https://medium.com/.../studio-headphones-vs-consumer-headphones-vs-gaming-heads... ▾
Apr 3, 2017 - There are a billion million **headphones** to choose from right now, ... Studio **headphones** are built primarily for **professional** work, and have ...

Semantic Similarity in Search Engines

Google meanwhile hits pages where “professional” and “headphones” function independently, in ways that are irrelevant to the meaning of our query:

Headphones & Earbuds | JBL - JBL.com
https://www.jbl.com/headphones/ ▾
JBL **headphones**, including earbuds, in-ear **headphones**, and on-ear ... Sport **headphones** for all athletes, from **professional** runners to weekend warriors.

So understanding that “professional headphones” needs to be treated *as a single headword* is the real takeaway here.

Similarity ≠ Synonymy

A useful synset for “professional headphones” would be:

- reference headphones
- studio headphones
- monitor headphones
- monitoring headphones
- DJ headphones
- ~~earphones~~

Note that “reference” and “studio” are in no way synonyms of each other—nor of “DJ” or “monitor.” Most importantly, they are not synonyms of “professional.”

We could arrive at this sysnet by an effort in lexicography or in terminology extraction.

DataScience@SMU

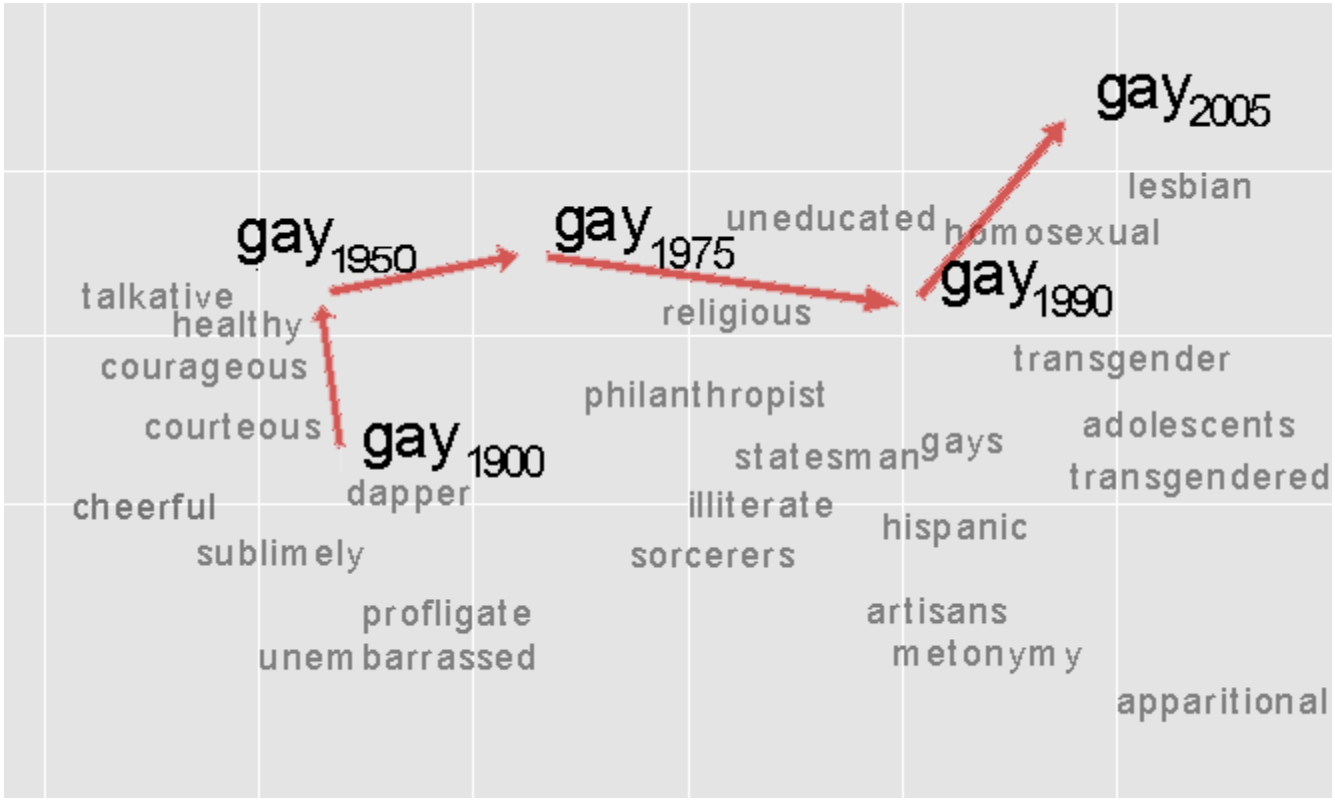
More Applications of Semantic Similarity

Natural Language Processing

Some slides after Carpuat 2016, Jurafsky 2014

More Uses of Word Similarity Measurement

Statistically Significant Detection of Linguistic Change
Kulkarni, Al-Rfou, Perozzi, Skiena 2015



Disambiguating Acronyms: Which Should Be Sense 1?

Strongest vector-based interpretation of the acronym “HP,” derived from a sample of randomly crawled, *recently published* web pages at the times indicated:

There are many less frequent construals, such as “high pressure,” “Hilary Putnam.”

Plagiarism Detection

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

It’s Not Plagiarism (Apparently) If…

…you’re a journalist or blogger

…and you don’t have time to research your own story

…and you’re good at paraphrasing

Girl Scout sells 312 boxes of cookies in six hours outside pot dispensary

FOX NEWS

A brilliant young **Girl Scout from San Diego sold** more than 300 boxes of Tagalongs, Thin Mints and other munchie-friendly **snacks** after setting up her wares **near a marijuana dispensary** over the weekend.

The 9-year-old girl's father, who was not identified either, **confirmed** to San Diego's KGTV that she ended up **selling a total of 312 boxes over the course of about six hours**, presumably to customers of the **Urbn Leaf dispensary.**

The dispensary, too, advertised that the girl would be appearing outside the facility in a **post shared to the shop's Instagram page**.

"**Get some Girl Scout Cookies with your GSC today until 4pm**," wrote Urbn Leaf in the caption, making reference to its own GSC strains of marijuana (short for "Girl Scout Cookies") which also includes a phenotype named after Thin Mints. "**Have a friend that wants to #tagalong? Bring them with — shopping is more fun with friends anyways**," the shop added. …

Despite some critical comments, Alison Bushan, a spokesperson for **Girl Scouts San Diego**, has confirmed to KGTV that the girl did not technically violate any official Girl Scout codes of conduct, as she wasn't **selling from** a booth directly outside the shop, but rather a **wagon** on the sidewalk **alongside her father**.

"**If that's what they say they were doing... then they were right within the rules**," Bushan **said** to KGTV.

Girl Scout Sells 300 Boxes of Cookies Outside Marijuana Dispensary

sheknows

If there's any group that understands their target market like the backs of their hands, it's the Girl Scouts.

Case in point: one particular **Girl Scout from California who sold cookies outside a marijuana dispensary** in San Diego on Feb. 2. **The girl's father told** Dallas, Texas, news station Fox 4 she **sold 300 boxes in just six hours**.

The Girl Scout received some help from **the dispensary itself, Urbn Leaf**, who **posted a photo of her on its Instagram** account with the caption, "**Get some Girl Scout Cookies with your GSC today until 4pm**! **Have a friend that wants to #tagalong? Bring them with – shopping is more fun with friends anyways."**

According to **Girl Scouts San Diego**, booth sales do not start for another week, but the scouts are allowed to **sell from wagons** as long as a **parent or guardian is present**.

"So**, if that's what they say they were doing ... then they were right within the rules**," the Girl Scout's father **said**.

Girl Scout sells 312 boxes of cookies in six hours outside pot dispensary

FOX NEWS

A brilliant young **Girl Scout from San Diego sold** more than 300 boxes of Tagalongs, Thin Mints and other munchie-friendly **snacks** after setting up her wares **near a** **mariju**

The 9-
confir
a total
presur
The di
appea
shop's
"**Get s**
until 4
referer
"Girl S
named
#tagal
with fr
Despit
spokes
KGTV
Scout
booth
sidewalk **alongside her father**.

"**If that's what they say they were doing... then they were right within the rules**," Bushan **said** to KGTV.

Girl Scout Sells 300 Boxes of Cookies Outside Marijuana Dispensary

sheknows

If there's any group that understands their target market like the backs of their hands, it's the Girl Scouts.

ana
ather
300

oto
n,

ore

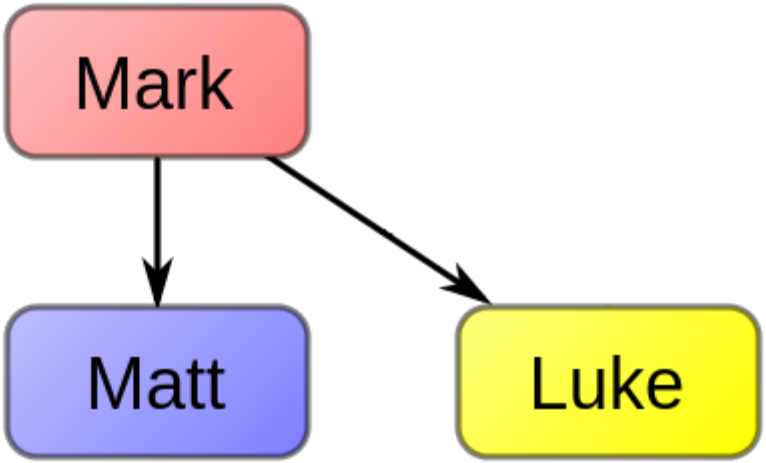
es do
owed
ardian

"So, **if that's what they say they were doing ... then they were right within the rules**," the Girl Scout's father **said**.

So if we are building a news aggregator or a news alert service or a news search engine, and we want to know the *information gain* that an article represents vis-à-vis previous articles, then the trace patterns of these word similarities would be very useful.

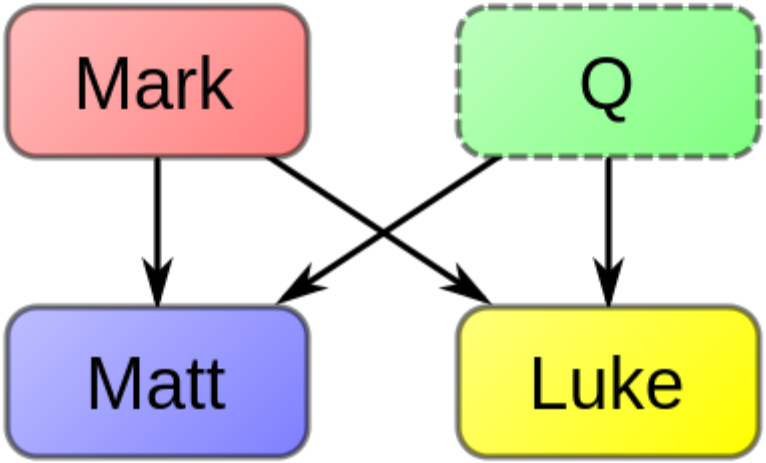
An Independent Validation of This Approach

1. A great number of sentences are very similar in the gospel texts of Mark, Matthew, and Luke (from the Christian scriptures).
2. There is a large set of these sentences having similar term frequency as the rest of Mark but dissimilar term frequency from the rest of Matthew and the rest of Luke.
3. Thus, scholars presume that Matthew and Luke both utilized Mark as a source (meaning Mark was written earlier than either Matthew and Luke).



An Independent Validation of This Approach

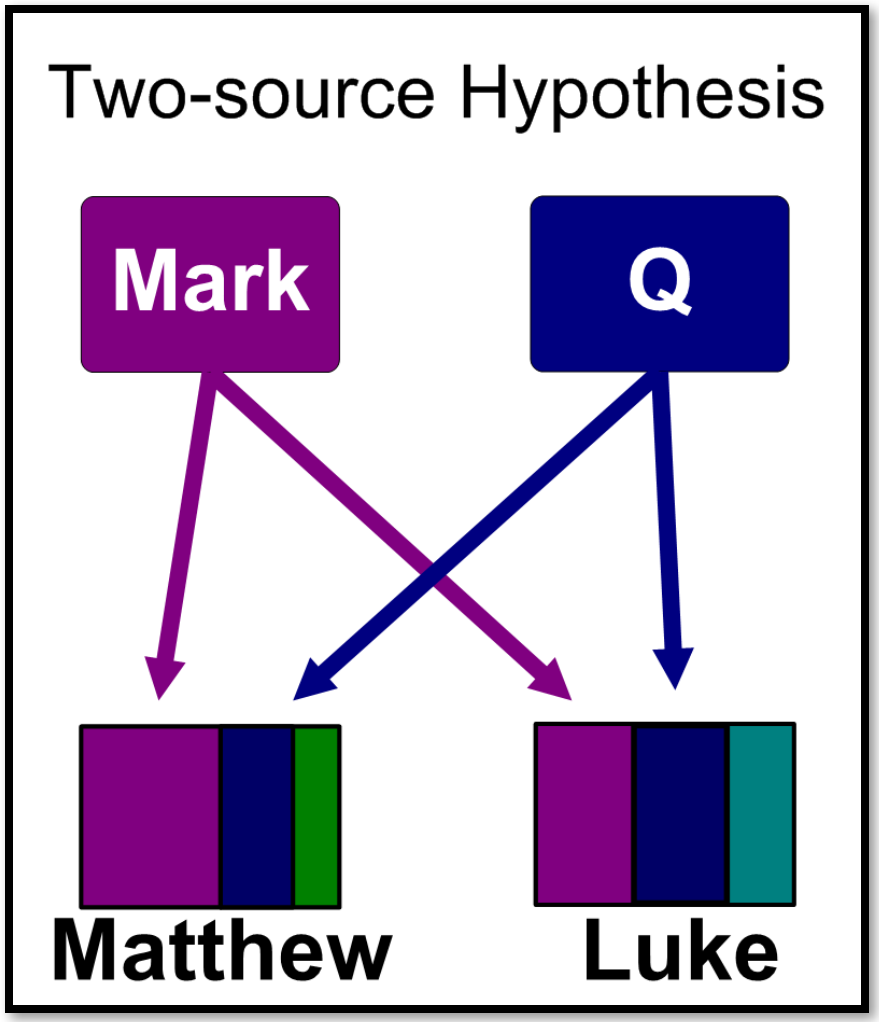
1. Interestingly, there is a sizable set of additional sentences that are similar in Matthew and Luke but that do not appear in Mark at all.
2. These sentences, taken as a set, have very different term frequencies from the rest of Matthew and Luke (and from Mark).
3. Thus, scholars presume that Matthew and Luke both utilized some other outside source, besides just Mark—call it “Q”*.



**“Quelle,” German for “source”*

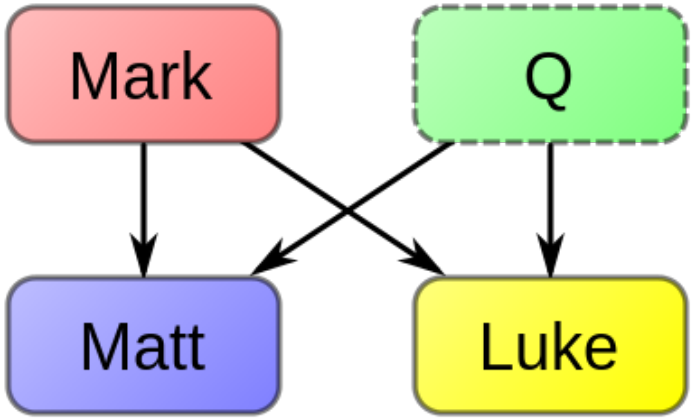
This is called the “two-source hypothesis.”

It is illustrated at right, where colors represent how much material Matthew and Luke appear to have borrowed from Mark and Q.



An Independent Validation of This Approach

1. For several decades, Q was a much-debated hypothesis.
2. That's because it had been "reconstructed" purely by linguistic analysis, but no actual manuscript looking anything like it had been found.
3. Then in the 1960s, the *Gospel of Thomas* surfaced—looking very similar to the hypothesized Q!



*"Quelle," German for "source"

DataScience@SMU