Lecture 38 – tf/idf and information retrieval

Mark Hasegawa-Johnson

5/1/2020

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

Similarity: The Internet is the database

Similarity = words can be used interchangeably in most contexts

How do we measure that in practice?

Answer: extract examples of word w_1, +/- k words ($2 \leq k \leq 5$, for example):

…hot, although iced **coffee** is a popular…

…indicate that moderate **coffee** consumption is benign…

…and of w_2:

…consumed as iced **tea**. Sweet tea is…

…national average of **tea** consumption in Ireland…

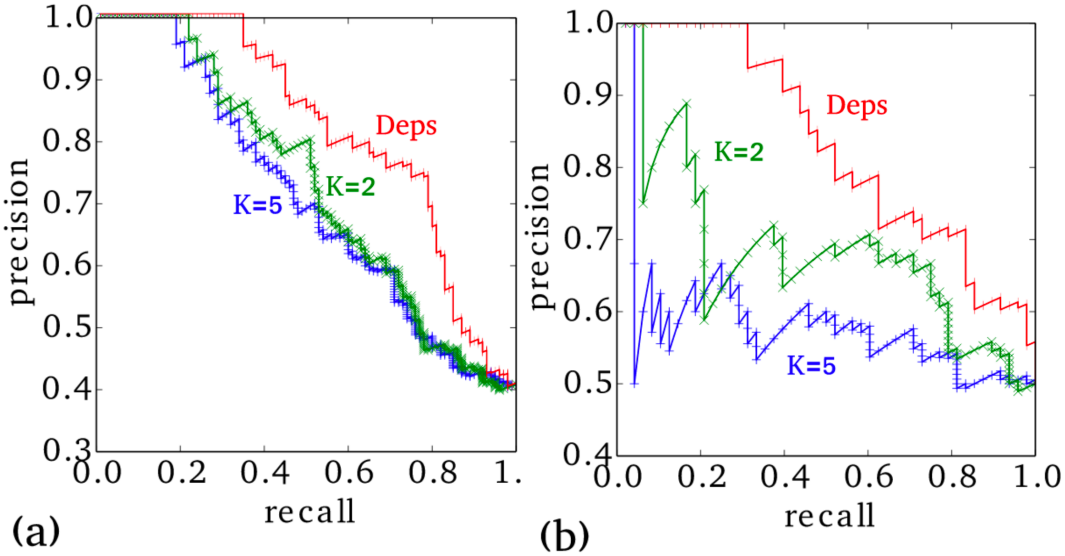
The words "iced" and "consumption" appear in both contexts, so we can conclude that $s(\text{coffea}, \text{tea}) > 0$. No other words are shared, so we can conclude $s(\text{coffee}, \text{tea}) < 1$.

Similarity vs. Relatedness

Levy & Goldberg (2014) trained word2vec in three different ways:

- k=2
- k=5
- Context determined by first parsing the sentence to get syntactic dependency structure (Deps)

They tested all three method for the similarity vs. relatedness of the nearest-neighbor of each word.

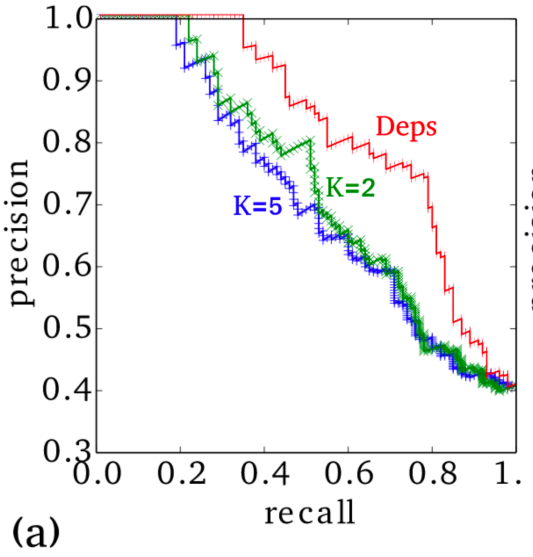


Precision vs. Recall on the WordSim-353 database, in which word pairs may be either ***related*** or ***similar*** (Fig. 2(a), Levy & Goldberg 2014)

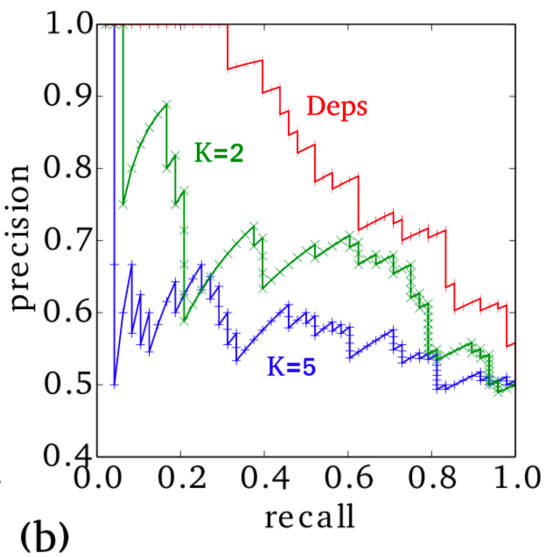
Precision vs. Recall on the Chiarello et al. database, in which word pairs are only ***similar*** (Fig. 2(b), Levy & Goldberg 2014)

Similarity vs. Relatedness

- Apparently, the smaller context window (k=2) produces vectors whose nearest neighbors are more ***similar*** (they can be used identically in a sentence).
- The larger context (k=5) produces vectors whose nearest neighbors are ***related***, not just ***similar***.
- More specifically, the latter words pairs are said to inhabit the same ***semantic field***.
- A ***semantic field*** is a group of words that refers to the same subject.



Precision vs. Recall on the WordSim-353 database, in which word pairs may be either ***related*** or ***similar*** (Fig. 2(a), Levy & Goldberg 2014)



Precision vs. Recall on the Chiarello et al. database, in which word pairs are only ***similar*** (Fig. 2(b), Levy & Goldberg 2014)

Similarity vs. Relatedness

…studied at hogwarts, a castle…

w=hogwarts

… harry potter studied at hogwarts…

	vector nearest neighbors, context k=2	vector nearest neighbors, context k=5	
…studied at evernight, a castle…	evernight	dumbledore	…harry potter learned from dumbledore…
…studied at sunnydale…	sunnydale	hallows	…harry potter and the deathly hallows..
…a castle garderobe…	garderobe	half-blood	…harry potter and the half-blood…
…lives at blandings, a castle…	blandings	malfoy	…harry potter said to malfoy…
…lives at collinwood, a castle…	collinwood	snape	…harry potter said to snape…

Examples of k=2 and k=5 nearest-neighbors, from (Levy & Goldberg, 2014)

What if you wanted ***semantic field***, not similarity?

w=hogwarts

vector nearest neighbors, context k=2	vector nearest neighbors, context k=5
evernight	dumbledore
sunnydale	hallows
garderobe	half-blood
blandings	malfoy
collinwood	snape

- What if you wanted your vector embedding to capture semantic field, as in the second column (not similar usage, like the first column)?
- If you want that, it seems that larger contexts are better.
- Why not just set context window = the whole document?

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

the term-document matrix

Hogwarts School of Witchcraft and Wizardry, commonly shortened to Hogwarts, is a fictional British school of magic for students aged eleven to eighteen, and is the primary setting for the first six books in J. K. Rowling's Harry Potter series…
Albus Percival Wulfric Brian Dumbledore is a fictional character in J. K. Rowling's Harry Potter series. For most of the series, he is the headmaster of the wizarding school Hogwarts. As part of his backstory, it is revealed that he is the founder and leader of …
Collinwood Mansion is a fictional house featured in the Gothic horror soap opera Dark Shadows (1966–1971). Built in 1795 by Joshua Collins, Collinwood has been home to the Collins family—and other sometimes unwelcome supernatural visitors…

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling’s	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

the term-document matrix

From the term-document matrix, we can define each term vector to be just the vector of term frequencies:

$$\vec{v}(i) = [tf(i,1), \ldots, tf(i,D)]$$

…where we now define the ***term frequency*** (of term i in document j) to be the number of times the term occurs in the document:

$$tf(i,j) = \text{Count}(\text{word } i \text{ in document } j)$$

For example,

$$\vec{v}(\text{a}) = [1,1,1]$$
$$\vec{v}(\text{of}) = [1,2,1]$$
$$\vec{v}(\text{potter}) = [1,1,0]$$

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

cosine similarity

The relatedness of two words can now be measured using their cosine similarity. For example,

$$s(\text{rowling's}, \text{harry}) = \cos \measuredangle(\text{rowling's}, \text{harry})$$

$$= \frac{\vec{v}(\text{rowling's}) \cdot \vec{v}(\text{harry})}{|\vec{v}(\text{rowling's})||\vec{v}(\text{harry})|}$$

$$= \frac{1\times1 + 1\times1 + 0\times0}{\sqrt{2}\times\sqrt{2}} = 1$$

$$s(\text{harry}, \text{gothic}) = \cos \measuredangle(\text{harry}, \text{gothic})$$

$$= \frac{\vec{v}(\text{harry}) \cdot \vec{v}(\text{gothic})}{|\vec{v}(\text{harry})||\vec{v}(\text{gothic})|}$$

$$= \frac{1\times0 + 1\times0 + 0\times1}{\sqrt{2}\times1} = 0$$

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

document vectors

Now let's try something different. Let's define a vector for each document, rather than for each term:

$$\vec{d}(j) = [tf(1,j), \ldots, tf(V,j)]$$

Thus,

$$\vec{d}(\mathrm{H}) = [1,1,1,2,1,1,1,1,1,0,0,0]$$

$$\vec{d}(\mathrm{D}) = [1,2,1,4,1,0,1,1,1,1,0,0,0]$$

$$\vec{d}(\mathrm{C}) = [1,0,2,1,1,0,0,0,0,0,1,1,1]$$

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

information retrieval

Document vectors are useful because they allow us to retrieve a document, based on the degree to which it matches a query. For example, the query:

"What school did Harry Potter attend?"

…can be written as a query vector:

$$\vec{q} = [0,0,0,0,0,1,0,1,1,0,0,0,0]$$

We can sometimes find the most relevant document using cosine distance:

$$\frac{\vec{q} \cdot \vec{d}(\mathrm{H})}{|\vec{q}||\vec{d}(\mathrm{H})|} = \frac{3}{\sqrt{3}\sqrt{13}} = 0.48$$

$$\frac{\vec{q} \cdot \vec{d}(\mathrm{D})}{|\vec{q}||\vec{d}(\mathrm{D})|} = \frac{2}{\sqrt{3}\sqrt{27}} = 0.22$$

$$\frac{\vec{q} \cdot \vec{d}(\mathrm{C})}{|\vec{q}||\vec{d}(\mathrm{C})|} = \frac{0}{\sqrt{3}\sqrt{10}} = 0.00$$

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

document classification

Suppose that we find a new document on the web:

Dark Shadows is an American Gothic soap opera that originally aired weekdays on the ABC television network, from June 27, 1966, to April 2, 1971. The show depicted the lives, loves, trials, and tribulations of …

Now we want to determine whether this document is about the Dark Shadows soap opera, or about the Harry Potter series.

How?

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

document classification

To start with, let's create a single merged document class vector, for each class, by just adding together all of the document vectors in the class:

$$\vec{x}(\text{Harry Potter}) = \vec{d}(\text{H}) + \vec{d}(\text{D})$$

$$\vec{x}(\text{Dark Shadows}) = \vec{d}(\text{C})$$

	document class	
term	Harry Potter	Dark Shadows
a	2	1
of	3	
in	2	2
is	6	1
fictional	2	1
school	1	
rowling's	2	
harry	2	
potter	2	
series	2	
house		1
featured		1
gothic		1

document classification

Now we turn the new document into a vector with the same dimensions:

Dark Shadows is an American Gothic soap opera that originally aired weekdays on the ABC television network, from June 27, 1966, to April 2, 1971. The show depicted the lives, loves, trials, and tribulations of …

$$\vec{q} = [0,1,0,1,0,0,0,0,0,0,0,0,1]$$

	document class	
term	Harry Potter	Dark Shadows
a	2	1
of	3	
in	2	2
is	6	1
fictional	2	1
school	1	
rowling's	2	
harry	2	
potter	2	
series	2	
house		1
featured		1
gothic		1

document classification

Now let's just compute the cosine similarity with each document class:

Dark Shadows is an American Gothic soap opera that originally aired weekdays on the ABC television network, from June 27, 1966, to April 2, 1971. The show depicted the lives, loves, trials, and tribulations of …

$$\vec{q} = [0,1,0,1,0,0,0,0,0,0,0,0,1]$$

$$\frac{\vec{q} \cdot \vec{x}(\mathrm{HP})}{|\vec{q}|\left|\vec{d}(\mathrm{HP})\right|} = \frac{1\times3 + 1\times6 + 1\times0}{\sqrt{3}\sqrt{74}} = 0.60$$

$$\frac{\vec{q} \cdot \vec{x}(\mathrm{DS})}{|\vec{q}|\left|\vec{d}(\mathrm{DS})\right|} = \frac{1\times0 + 1\times1 + 1\times1}{\sqrt{3}\sqrt{10}} = 0.37$$

…oops…

	document class	
term	Harry Potter	Dark Shadows
a	2	1
of	3	
in	2	2
is	6	1
fictional	2	1
school	1	
rowling's	2	
harry	2	
potter	2	
series	2	
house		1
featured		1
gothic		1

document classification: tf on a log scale

- We need some way to point out that the difference between $tf(\text{HP}, \text{gothic}) = 0$ and $tf(\text{DS}, \text{gothic}) = 1$ is much more important than the difference between $tf(\text{HP}, \text{is}) = 6$ and $tf(\text{DS}, \text{is}) = 1$.
- One way to think about it: it's not the difference between term frequencies that matters, it's their ratio that matters.

$$6 - 1 \gg 1 - 0$$
$$\frac{6}{1} \ll \frac{1}{0}$$

	document class	
term	Harry Potter	Dark Shadows
a	2	1
of	3	
in	2	2
is	6	1
fictional	2	1
school	1	
rowling's	2	
harry	2	
potter	2	
series	2	
house		1
featured		1
gothic		1

document classification: tf on a log scale

We can emphasize ratios, rather than differences, by measuring the log of tf, rather than the raw frequencies:

$$\log 6 - \log 1 \ll \log 1 - \log 0$$

So let's redefine term frequency to be

$$tf(i,j) = \log_{10} \text{Count}(\text{word } i \text{ in document } j)$$

The use of a base-10 logarithm is a sort of anachronism; it's because this definition was first published in 1972. Really, though, the base of the logarithm doesn't matter much.

	document class	
term	Harry Potter	Dark Shadows
a	0.3	0
of	0.5	$-\infty$
in	0.3	0.3
is	0.8	0
fictional	0.3	0
school	0	$-\infty$
rowling's	0.3	$-\infty$
harry	0.3	$-\infty$
potter	0.3	$-\infty$
series	0.3	$-\infty$
house	$-\infty$	0
featured	$-\infty$	0
gothic	$-\infty$	0

document classification: tf on a log scale

All those $-\infty$ terms are annoying and numerically awful. There are two standard ways to deal with them:

- If you're in the big data regime, where the difference between 0 and 1 is unimportant, and the difference between 1 and 10 is about the same as the difference between 10 and 100:

$$tf(i,j) = 1 + \max(0, \log_{10} \text{Count})$$

- If you're in the small-data regime (as in our example), where the difference between 0 and 1 is about as important as the difference between 1 and 3:

$$tf(i,j) = \log_{10}(1 + \text{Count})$$

	document class	
term	Harry Potter	Dark Shadows
a	0.5	0.3
of	0.6	0
in	0.5	0.5
is	0.8	0.3
fictional	0.5	0.3
school	0.3	0
rowling's	0.5	0
harry	0.5	0
potter	0.5	0
series	0.5	0
house	0	0.3
featured	0	0.3
gothic	0	0.3

document classification: tf on a log scale

Using this new notation, our query vector is:

$$\vec{q} = [0,0.3,0,0.3,0,0,0,0,0,0,0,0,0.3]$$

$$\frac{\vec{q} \cdot \vec{x}(\text{HP})}{|\vec{q}||\vec{d}(\text{HP})|} = \frac{0.18 + 0.24 + 0}{\sqrt{0.27}\sqrt{2.84}} = 0.48$$

$$\frac{\vec{q} \cdot \vec{x}(\text{DS})}{|\vec{q}||\vec{d}(\text{DS})|} = \frac{0 + 0.09 + 0.09}{\sqrt{0.27}\sqrt{0.79}} = 0.39$$

So, now the "Dark Shadows" class is closer to correctly claiming this query. But we're not quite there yet…

	document class	
term	Harry Potter	Dark Shadows
a	0.5	0.3
of	0.6	0
in	0.5	0.5
is	0.8	0.3
fictional	0.5	0.3
school	0.3	0
rowling's	0.5	0
harry	0.5	0
potter	0.5	0
series	0.5	0
house	0	0.3
featured	0	0.3
gothic	0	0.3

Digression: relationship between tf and naïve Bayes

Did you notice that most words occur in a query either once, or zero times? So every element of the query vector is either $\log_{10}(1+0) = 0$ or $\log_{10}(1+1) = 0.3$. So, for q but not for x, let's return it to binary, $\vec{q} = [0,1,0,\ldots]$. Then:

$$\vec{q} \cdot \vec{x}(j) = \sum_{i=1}^{V} \text{Count}(i,q) \log_{10}(1 + \text{Count}(i,j))$$

$$= \log_{10} \prod_{i=1}^{V} (1 + \text{Count}(i,j))^{\text{Count}(i,q)}$$

Just for the heck of it, let's divide by $(V + \text{N}(j))^{\text{N}(q)}$, where V is vocabulary size, $\text{N}(j)$ is the number of words in class j, and $\text{N}(q)$ is the number of words in the query. That gives us:

$$\vec{q} \cdot \vec{x}(j) = \log_{10} \prod_{i=1}^{V} \left(\frac{1 + \text{Count}(i,j)}{V + \text{N}(j)} \right)^{\text{Count}(i,q)} = \log_{10} \prod_{\substack{i:\text{Word } i \\ \text{is in the Query}}} p(\text{word } i | \text{class } j)$$

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

document classification: idf

We saw that putting tf on a log scale is not quite enough for us to correctly classify the test document as being part of class “Dark Shadows,” so let’s look for more problems to fix.

Here’s a problem: why do the words “a,” “of,” “in,” “is” count more than “potter” and “gothic”?

Those function words are used by all classes, so we shouldn’t really pay so much attention to them.

	document class	
term	Harry Potter	Dark Shadows
a	0.5	0.3
of	0.6	0
in	0.5	0.5
is	0.8	0.3
fictional	0.5	0.3
school	0.3	0
rowling’s	0.5	0
harry	0.5	0
potter	0.5	0
series	0.5	0
house	0	0.3
featured	0	0.3
gothic	0	0.3

document classification: idf

Inverse document frequency (idf) is a discount weight, meant to reduce the importance of any word that's used equally across all classes. A typical definition is:

$$idf(i) = \log_{10}\left(\frac{D}{df(i)}\right)$$

...where D is the number of document classes (2, in our example), and $df(i)$ is the number of documents in which the ith word appears.

	document class	
term (idf)	Harry Potter	Dark Shadows
a(0)	0.5	0.3
of(0.3)	0.6	0
in(0)	0.5	0.5
is(0)	0.8	0.3
fictional(0)	0.5	0.3
school(0.3)	0.3	0
rowling's(0.3)	0.5	0
harry(0.3)	0.5	0
potter(0.3)	0.5	0
series(0.3)	0.5	0
house(0.3)	0	0.3
featured(0.3)	0	0.3
gothic(0.3)	0	0.3

document classification: tf-idf

With that definition, we get

$$tf(i,j)idf(i)$$
$$= \log_{10}(1 + \text{Count}(i,j)) \log_{10}\left(\frac{D}{df(i)}\right)$$

…and the document class vectors are now

$$\vec{x}(j) = [tf(1,j)idf(1), \ldots, tf(V,j)idf(V)]$$

	document class	
term (idf)	Harry Potter	Dark Shadows
a(0)	0	0
of(0.3)	0.18	0
in(0)	0	0
is(0)	0	0
fictional(0)	0	0
school(0.3)	0.09	0
rowling's(0.3)	0.15	0
harry(0.3)	0.15	0
potter(0.3)	0.15	0
series(0.3)	0.15	0
house(0.3)	0	0.09
featured(0.3)	0	0.09
gothic(0.3)	0	0.09

document classification: tf-idf

Remember, the original word counts in our query were:

$$\vec{q} = [0,1,0,1,0,0,0,0,0,0,0,0,1]$$

If we convert those into tf-idf, we get

$$\vec{q} = [0,0.09,0,0,0,0,0,0,0,0,0,0,0.09]$$

Then

$$\frac{\vec{q} \cdot \vec{x}(\mathrm{HP})}{|\vec{q}|\left|\vec{d}(\mathrm{HP})\right|} = \frac{0.0162 + 0 + 0}{\sqrt{0.0162}\sqrt{0.1305}} = 0.35$$

$$\frac{\vec{q} \cdot \vec{x}(\mathrm{DS})}{|\vec{q}|\left|\vec{d}(\mathrm{DS})\right|} = \frac{0 + 0 + 0.0081}{\sqrt{0.0162}\sqrt{0.0243}} = 0.41$$

It worked! We got the right answer!

	document class	
term (idf)	Harry Potter	Dark Shadows
a(0)	0	0
of(0.3)	0.18	0
in(0)	0	0
is(0)	0	0
fictional(0)	0	0
school(0.3)	0.09	0
rowling's(0.3)	0.15	0
harry(0.3)	0.15	0
potter(0.3)	0.15	0
series(0.3)	0.15	0
house(0.3)	0	0.09
featured(0.3)	0	0.09
gothic(0.3)	0	0.09

tf-idf for information retrieval: key concepts

1. It's not the difference between counts that matters, it's the ratio. So instead of raw counts, use log counts:
$$tf(i,j) = \log_{10}(1 + \text{Count})$$

2. Words that occur in many documents are unimportant. Discount them by the factor
$$idf(i) = \log_{10}\left(\frac{D}{df(i)}\right)$$

	document class	
term (idf)	Harry Potter	Dark Shadows
a(0)	0	0
of(0.3)	0.18	0
in(0)	0	0
is(0)	0	0
fictional(0)	0	0
school(0.3)	0.09	0
rowling's(0.3)	0.15	0
harry(0.3)	0.15	0
potter(0.3)	0.15	0
series(0.3)	0.15	0
house(0.3)	0	0.09
featured(0.3)	0	0.09
gothic(0.3)	0	0.09

Outline

- similarity vs. semantic field: word2vec at different scales
- term frequency (tf): the term-document matrix
- cosine similarity
- document classification: tf on a log scale
- document classification: inverse document frequency (idf)
- relatedness again: the word co-occurrence matrix

The Word Co-Occurrence Matrix

Now that we understand information retrieval, let's go back to our original question:

How can we determine whether or not two words are related?

The Word Co-Occurrence Matrix

Instead of creating a term-document matrix, let's create a matrix that shows how often each pair of words occurs in the same document.

This will be

$$W(i,k) = \sum_{j=1}^{D} \text{Count}(i,j)\text{Count}(k,j)$$

For example, for the words i =a and k =of,

$$W(\text{a},\text{of}) = 1\times1 + 1\times2 + 0 = 3$$

	document		
term	Hogwarts	Dumbledore	Collinwood
a	1	1	1
of	1	2	
in	1	1	2
is	2	4	1
fictional	1	1	1
school	1		
rowling's	1	1	
harry	1	1	
potter	1	1	
series	1	1	
house			1
featured			1
gothic			1

The Word Co-Occurrence Matrix

Here's a subset of the word co-occurrence matrix.

Notice that this seems, again, to give too much credit to the function words. Let's reduce their importance using tf-idf.

	term 2							
term 1	a	of	in	school	harry	potter	house	gothic
a	3	3	4	1	2	2	1	1
of	3	5	3	1	3	3		
in	4	3	6	1	2	2	2	2
school	1	1	1	1	1	1		
harry	2	3	2	1	2	2		
potter	2	3	2	1	2	2		
house	1		2				1	1
gothic	1		2				1	1

The Word Co-Occurrence Matrix

In this example, we have D=3 documents, so the possible values of idf are

$$\log_{10}(3/3) = 0$$
$$\log_{10}(3/2) \approx 0.2$$
$$\log_{10}(3/1) \approx 0.3$$

	term 2							
term 1	a	of	in	school	harry	potter	house	gothic
a								
of		0.032		0.018	0.024	0.024		
in								
school		0.018		0.027	0.018	0.018		
harry		0.024		0.018	0.020	0.020		
potter		0.024		0.018	0.020	0.020		
house							0.027	0.027
gothic							0.027	0.027

$$W(i,k) = \log_{10}\left(1 + \sum_{j=1}^{D} \text{Count}(i,j)\text{Count}(k,j)\right) \log_{10}\left(\frac{D}{df(i)}\right) \log_{10}\left(\frac{D}{df(k)}\right)$$

Conclusions

- semantic field = a group of words that refers to the same subject
- term frequency (tf): Count(term i appears in document j)
- cosine similarity

$$s(\text{rowling's}, \text{harry}) = \cos \measuredangle(\text{rowling's}, \text{harry}) = \frac{\vec{v}(\text{rowling's}) \cdot \vec{v}(\text{harry})}{|\vec{v}(\text{rowling's})||\vec{v}(\text{harry})|}$$

- document classification: tf on a log scale

$$tf(i,j) = \log_{10}(1 + \text{Count})$$

- document classification: inverse document frequency (idf)

$$idf(i) = \log_{10}\left(\frac{D}{df(i)}\right)$$

- word co-occurrence matrix

$$W(i,k) = \sum_{j=1}^{D} \text{Count}(i,j)\text{Count}(k,j)$$