Word Vectors

Similarity

We have seen how it is possible to compute the similarity of documents on the basis of the words that they contain.

Similarity

We have seen how it is possible to compute the similarity of documents on the basis of the words that they contain.

These books are similar since they contain similar words.

word	ship	sail	ocean	boat	vessel	horse	dog
title							
Moby Dick; Or, The Whale by Herman Melville	471.0	84.0	71.0	288.0	53.0	12.0	16.0
The Odyssey by Homer	239.0	38.0	8.0	3.0	15.0	11.0	6.0
The Count of Monte Cristo, Illustrated by Alexandre Dumas	35.0	26.0	24.0	62.0	93.0	73.0	13.0
The Call of the Wild by Jack London	1.0	0.0	0.0	7.0	0.0	1.0	57.0

Similarity

But we could also say that *words* are similar if they often both appear in the same books.

These words co-occur frequency, so they are similar.

word title	ship	sail	ocean	boat	vessel	horse	dog
Moby Dick; Or, The Whale by Herman Melville	471.0	84.0	71.0	288.0	53.0	12.0	16.0
The Odyssey by Homer	239.0	38.0	8.0	3.0	15.0	11.0	6.0
The Count of Monte Cristo, Illustrated by Alexandre Dumas	35.0	26.0	24.0	62.0	93.0	73.0	13.0
The Call of the Wild by Jack London	1.0	0.0	0.0	7.0	0.0	1.0	57.0

Word Vectors

We can actually take the counts as a vector representation of each word.

ship = (471, 239, 35, 1)
sail = (84, 38, 26, 0)
etc.

word title	ship	sail	ocean	boat	vessel	horse	dog
Moby Dick; Or, The Whale by Herman Melville	471.0	84.0	71.0	288.0	53.0	12.0	16.0
The Odyssey by Homer	239.0	38.0	8.0	3.0	15.0	11.0	6.0
The Count of Monte Cristo, Illustrated by Alexandre Dumas	35.0	26.0	24.0	62.0	93.0	73.0	13.0
The Call of the Wild by Jack London	1.0	0.0	0.0	7.0	0.0	1.0	57.0

Word Vectors

The Distributional Hypothesis: Words that occur in similar contexts tend to have similar meanings.

We can gauge if words are similar by looking at their contexts.

Word Vectors

What is ongchoi?

Word Vectors

What is ongchoi?

(6.1) Ongchoi is delicious sauteed with garlic.

(6.2) Ongchoi is superb over rice.

(6.3) ...ongchoi leaves with salty sauces...

Word Vectors

What is ongchoi?

(6.1) Ongchoi is delicious sauteed with garlic.
(6.2) Ongchoi is superb over rice.
(6.3) ...ongchoi leaves with salty sauces...

We also know these sentences:

(6.4) ...spinach sauteed with garlic over rice...
(6.5) ...chard stems and leaves are delicious...
(6.6) ...collard greens and other salty leafy greens

Word Vectors

What is ongchoi?

Ongchoi

Ong Choi (Ipomoea aquatica), is **a member of the Morning Glory family (Convolvulaceae)**. The most common varieties are bright green and grow up to 14 inches tall. Ong Choi is of East Indian origin and is extremely popular in Southern China, Vietnam, Malaysia and Thailand.

Word Vectors

Looking across whole documents might be losing resolution.

We might be able to do better by just looking at smaller context windows.

Given a corpus of text, we can create a **co-occurrence matrix** by counting the number of times that two words appear in the corpus within a certain window size (say, within +/- 2 words).

Word Vectors

Ongchoi is delicious sauteed with garlic.

Word Vectors

Ongchoi is delicious sauteed with garlic.

{(ongchoi, is): 1, (ongchoi, delicious): 1}

Word Vectors

Ongchoi **is** delicious sauteed with garlic.

{(ongchoi, is): 1, (ongchoi, delicious): 1, (is, ongchoi): 1,
(is, delicious): 1, (is, sauteed): 1}

Word Vectors

Ongchoi is **delicious** sauteed with garlic.

{(ongchoi, is): 1, (ongchoi, delicious): 1, (is, ongchoi): 1,
(is, delicious): 1, (is, sauteed): 1, (delicious, ongchoi): 1,
(delicious, is): 1, (delicious, sauteed): 1, (delicious, with): 1}

Word Vectors

Ongchoi is delicious **sauteed** with garlic.

{(ongchoi, is): 1, (ongchoi, delicious): 1, (is, ongchoi): 1, (is, delicious): 1, (is, sauteed): 1, (delicious, ongchoi): 1, (delicious, is): 1, (delicious, sauteed): 1, (delicious, with): 1, (sauteed, is): 1, (sauteed, delicious): 1, (sauteed, with): 1, (sauteed, garlic): 1}

Word Vectors

Ongchoi is delicious sauteed **with** garlic.

{(ongchoi, is): 1, (ongchoi, delicious): 1, (is, ongchoi): 1, (is, delicious): 1, (is, sauteed): 1, (delicious, ongchoi): 1, (delicious, is): 1, (delicious, sauteed): 1, (delicious, with): 1, (sauteed, is): 1, (sauteed, delicious): 1, (sauteed, with): 1, (sauteed, garlic): 1, (with, delicious): 1, (with, sauteed): 1, (with, garlic): 1}

Word Vectors

Ongchoi is delicious sauteed with **garlic**.

{(ongchoi, is): 1, (ongchoi, delicious): 1, (is, ongchoi): 1,
(is, delicious): 1, (is, sauteed): 1, (delicious, ongchoi): 1,
(delicious, is): 1, (delicious, sauteed): 1, (delicious, with): 1,
(sauteed, is): 1, (sauteed, delicious): 1, (sauteed, with): 1,
(sauteed, garlic): 1, (with, delicious): 1, (with, sauteed): 1,
(with, garlic): 1, (garlic, sauteed): 1, (garlic, with): 1}

Word Vectors

We get a vector representation of each word using the rows of this matrix.

whale = (111, 1219, 37, 2, …)

Here's a (subset) of the co-occurrence matrix for Moby Dick.

	white	whale	ship	sea
white	282.0	111.0	0.0	3.0
whale	111.0	1219.0	37.0	2.0
ship	0.0	37.0	520.0	2.0
sea	3.0	2.0	2.0	457.0

Word Vectors

The problem is that this matrix is big (lots of rows and columns) and *sparse* - it contains mostly zeros.

Perhaps we could find a lower-dimensional representation that captures most of the relevant information.

Word Vectors

We can take the **singular value decomposition** of the co-occurrence matrix.

This decomposes it into

$$M = UDV$$

Where U and V are orthogonal and D is diagonal.

By keeping only the first *k* columns and *k* rows and columns of U and D and multiplying, we get a lower-dimensional representation.

Word Vectors

We can take the **singular value decomposition** of the co-occurrence matrix.

This decomposes it into

$$M = UDV$$

Where U and V are orthogonal and D is diagonal.

By keeping only the first *k* columns and *k* rows and columns of U and D and multiplying, we get a lower-dimensional representation.

Think: PCA

Word Vectors

Here's a snippet from the reduced matrix for Moby Dick.

	0	1	2	3
white	-7.847041	1.657775	-7.370778	-5.157919
whale	-9.049004	4.974995	-8.788873	-5.465692
ship	67.439162	-23.195442	18.534424	25.583082
sea	12.490784	0.568647	-3.033531	-3.671485

Word Vectors

Using these reduced vectors, you can find that “ship” and “boat” have similar vectors.

```
word_similarity('ship', 'boat')
```

```
0.946667114763397
```

```
word_similarity('sea', 'ocean')
```

```
0.9749745118845384
```

```
word_similarity('sea', 'ahab')
```

```
0.3415594888044475
```

Word Vectors

Alternative: Instead of counts, use the **pointwise mutual information (PMI)**

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

This is a measure of dependence of two words.

Recall: events A and B are independent if
P(A, B) = P(A)*P(B)

Word Vectors

Alternative: Instead of counts, use the **pointwise mutual information (PMI)**

$$\mathrm{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$$

If independent, PMI = 0.

If joint probability > product of individual probabilities, PMI > 0.

If joint probability < product of individual probabilities, PMI < 0.

Word Vectors

Alternative: Instead of counts, use the **pointwise mutual information (PMI)**

$$\mathrm{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

If independent, PMI = 0.

If joint probability > product of individual probabilities, PMI > 0.

If joint probability < product of individual probabilities, PMI < 0.

If joint probability = 0,
PMI = -infinity.

Word Vectors

To avoid problems with 0, it is usually advised to use the **Positive PMI**.

$$\mathrm{PPMI}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$