Dan Jurafsky and James Martin

Speech and Language Processing

# Chapter 6:
# Vector Semantics

# Let's define words by their usages

In particular, words are defined by their environments (the words around them)

Zellig Harris (1954): **If A and B have almost identical environments we say that they are synonyms**.

# What does ong choi mean?

Suppose you see these sentences:

- Ongchoi is delicious **sautéed with garlic**.
- Ongchoi is superb **over rice**
- Ongchoi **leaves** with salty sauces

And you've also seen these:

- …spinach **sautéed with garlic over rice**
- Chard stems and **leaves** are **delicious**
- Collard greens and other **salty** leafy greens

Conclusion:

- Ongchoi is a leafy green like spinach, chard, or collard greens

# Ong choi: *Ipomoea aquatica "Water Spinach"*

Yamaguchi, Wikimedia Commons, public domain

# Build a new model of meaning focusing on similarity

Each word = a vector

Similar words are "nearby in space"

# Define a word as a vector

Called an "embedding" because it's embedded into a space

The standard way to represent meaning in NLP

Fine-grained model of meaning for similarity

- NLP tasks like sentiment analysis
  - With words,  requires **same** word to be in training and test
  - With embeddings: ok if **similar** words occurred!!!
- Question answering, conversational agents, etc

# 2 kinds of embeddings

## Tf-idf

- A common baseline model
- Sparse vectors
- Words are represented by a simple function of the counts of nearby words

## Word2vec

- Dense vectors
- Representation is created by training a classifier to distinguish nearby and far-away words

# Review: words, vectors, and co-occurrence matrices

# Term-document matrix

Each document is represented by a vector of words

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 1 | 0 | 7 | 13 |
| **good** | 114 | 80 | 62 | 89 |
| **fool** | 36 | 58 | 1 | 4 |
| **wit** | 20 | 15 | 2 | 3 |

<3, 5>

x, y

<6, 2>

# Visualizing document vectors

# Vectors are the basis of information retrieval

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

Vectors are similar for the two comedies
Different than the history

Comedies have more fools and wit and fewer battles.

# Words can be vectors too

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

*battle* is "the kind of word that occurs in Julius Caesar and Henry V"

*fool* is "the kind of word that occurs in comedies, especially Twelfth Night"

# More common: word-word matrix (or "term-context matrix")

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

| | | |
|---|---|---|
| sugar, a sliced lemon, a tablespoonful of | **apricot** | jam, a pinch each of, |
| their enjoyment. Cautiously she sampled her first | **pineapple** | and another fruit whose taste she likened |
| well suited to programming on the digital | **computer**. | In finding the optimal R-stage policy from |
| for the purpose of gathering data and | **information** | necessary for the study authorized in the |

| | 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 | |

4
3
2
1
result
information
[6,4]
digital
[1,1]
1
2
3
4
5
6
data

# Visualizing cosines (well, angles)

# Cosine for computing similarity

Sec. 6.3

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

$$\vec{a}\cdot\vec{b} = |\vec{a}||\vec{b}|\cos\theta$$

$$\frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} = \cos\theta$$

$v_i$ is the count for word $v$ in context $i$
$w_i$ is the count for word $w$ in context $i$.

Cos($\vec{v}$,$\vec{w}$) is the cosine similarity of $\vec{v}$ and $\vec{w}$

0 — 90

cos(0) = 1  cos(90) = 0

# Cosine as a similarity metric

-1: vectors point in opposite directions

+1: vectors point in same directions

0: vectors are orthogonal

Frequency is non-negative, so cosine range 0-1

Normalized vector $= \frac{\vec{v}}{|\vec{v}|}$

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

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

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) =

$$\frac{0+0+0}{\sqrt{1+0+0}\sqrt{0+1+4}} = 0$$

Normalized vector of apricot $= \langle \frac{1}{\sqrt{1+0+0}}, \frac{0}{\sqrt{1+0+0}}, \frac{0}{\sqrt{1+0+0}} \rangle$ $\langle 1, 0, 0 \rangle$

# But raw frequency is a bad representation

Frequency is clearly useful; if *sugar* appears a lot near *apricot*, that's useful information.

But overly frequent words like *the*, *it,* or *they* are not very informative about the context

Need a function that resolves this frequency paradox!

| raw count | log count + 1 |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 1.3 |
| 3 | 1.47 |
| 4 | 1.6 |

| raw count | (log count) + 1 |
|---|---|
| 1 | 0 + 1 = 1 |
| 10 | 1 + 1 = 2 |
| 100 | 2 + 1 = 3 |
| 1000 | 3 + 1 = 4 |

# tf-idf: combine two factors

**tf: term frequency**. frequency count (usually log-transformed):

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

good

(a) the (50)

(b) the gos (450)

**Idf: inverse document frequency: tf-**

Total # of docs in collection

$$\mathrm{idf}_i = \log\left(\frac{N}{\mathrm{df}_i}\right)$$

$\frac{N}{N} = 1$

a) similar

b) Not similar

c) Don't know

Words like "the" or "good" have very low idf

# of docs that have word i

tf-idf value for word t in document d:

$$\boxed{w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t}$$

cure

a disease (50) (diabetes) 300

medicine

b disease (45) (diabetes 290)

100

corpus wide importance

(the ✗)

Informative
descriminative
✓

diabetes

Document frequency

| Word | df | idf |
|---|---|---|
| Romeo | 1 | 1.57 |
| salad | 2 | 1.27 |
| Falstaff | 4 | 0.967 |
| forest | 12 | 0.489 |
| battle | 21 | 0.074 |
| fool | 36 | 0.012 |
| good | 37 | 0 |
| sweet | 37 | 0 |

Total docs = 1000

idf of apple

$= \log\left(\frac{1000}{50}\right)$

$=$

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 0.074 | 0 | 0.22 | 0.28 |
| good | 0 | 0 | 0 | 0 |
| fool | 0.019 | 0.021 | 0.0036 | 0.0083 |
| wit | 0.049 | 0.044 | 0.018 | 0.022 |

A tf-idf weighted term-document matrix for four words in four Shakespeare plays

# Summary: tf-idf

Compare two words using tf-idf cosine to see if they are similar

Compare two documents

- Take the centroid of vectors of all the words in the document
- Centroid document vector is:

$$d = \frac{w_1 + w_2 + \ldots + w_k}{k}$$

# Example of Tf*Idf Vector

Represent the word "apple" as vector using following corpus. Use TF.IDF weights. Assume the window size for word context is 2

Document 1: I like to ride cycle often.

Document 2: Ali and Hassan ate apple and oranges.

Document 3: Ali ate apple not oranges

# Example of Tf*Idf Vector

Represent the word "apples" as vector using following corpus. Use TF.IDF weights. Assume the window size for word context is 2

Document 1: I like to ride cycle often.

Document 2: Ali and **[** Hassan ate **apple** and oranges **]**.

Document 3: **[** Ali ate **apple** not oranges **].**

Context words of "apple"= Hassan, ate, and, oranges, Ali, not

$N = 3$

Idf of "I" $= \log \frac{3}{1}$

$Idf = \frac{\log N}{DF} = \left(\log \frac{3}{2}\right)$

$\log_{10} \frac{3}{2} + 1$

| Dimension | I | Like | to | ride | cycle | often | Ali | and | Hassan | ate | apple | oranges | n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Raw Count | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 0 | 2 | |
| TF | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1.3 | 0 | 1.3 | |
| IDF | 0.48 | 0.48 | 0.48 | 0.48 | 0.48 | 0.48 | 0.18 | 0.48 | 0.48 | 0.18 | 0.18 | 0.18 | 0 |
| **Tf.IDF Weight** | **0** | **0** | **0** | **0** | **0** | **0** | **0.18** | **0.48** | **0.48** | **0.23** | **0** | **0.23** | **0** |

# An alternative to tf-idf

Ask whether a context word is **particularly informative** about the target word.

- Positive Pointwise Mutual Information (PPMI)

# Pointwise Mutual Information

**Pointwise mutual information**:

Do events x and y co-occur more than if they were independent?

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

**PMI between two words**: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

# Positive Pointwise Mutual Information

- PMI ranges from $-\infty$ to $+\infty$
- But the negative values are problematic
  - Things are co-occurring **less than** we expect by chance
- So we just replace negative PMI values by 0
- Positive PMI (PPMI) between word1 and word2:

$$\mathrm{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

# Computing PPMI on a term-context matrix

Matrix F with W rows (words) and C columns (contexts)

$f_{ij}$ is # of times $w_i$ occurs in context $c_j$

| | 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 |

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \qquad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

**Count(w,context)**

| | computer | data | pinch | result | sugar | |
|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 1 | 0 | 1 | 2/19 |
| pineapple | 0 | 0 | 1 | 0 | 1 | |
| digital | 2 | 1 | 0 | 1 | 0 | |
| information | 1 | 6 | 0 | 4 | 0 | |

P(apricot) =

a b c d e f g h  Total count = 19

Co-occurrence count [ d = b ]

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

**Count(w,context)**

| | computer | data | pinch | result | sugar | |
|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 1 | 0 | 1 | = 2 |
| pineapple | 0 | 0 | 1 | 0 | 1 | = 2 |
| digital | 2 | 1 | 0 | 1 | 0 | = 4 |
| information | 1 | 6 | 0 | 4 | 0 | = 11 |
| | | 7/19 | | | | 19 |

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

$$\sum_{j=1}^{C} f_{ij} \qquad p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad \sum_{i=1}^{W} f_{ij} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

**p(w,context)** **p(w)**

| | computer | data | pinch | result | sugar | | |
|---|---|---|---|---|---|---|---|
| apricot | 0.00 | 0.00 | 0.05 | 0.00 | 0.05 | 0.11 | 2/19 |
| pineapple | 0.00 | 0.00 | 0.05 | 0.00 | 0.05 | 0.11 | 2/19 |
| digital | 0.11 | 0.05 | 0.00 | 0.05 | 0.00 | 0.21 | 4/19 |
| information | 0.05 | 0.32 | 0.00 | 0.21 | 0.00 | 0.58 | 11/19 |
| **p(context)** | 0.16 | 0.37 | 0.11 | 0.26 | 0.11 | | |
| | 3/19 | 7/19 | 2/19 | 5/19 | 2/19 | | |

4 A 00
B 10 16
C 01
D 11

$\log_2 4 = 2$ $\log_2 16 = 4$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

| | p(w,context) | | | | | p(w) |
|---|---|---|---|---|---|---|
| | computer | data | pinch | result | sugar | |
| apricot | 0.00 | 0.00 | 0.05 | 0.00 | 0.05 | 0.11 |
| pineapple | 0.00 | 0.00 | 0.05 | 0.00 | 0.05 | 0.11 |
| digital | 0.11 | 0.05 | 0.00 | 0.05 | 0.00 | 0.21 |
| information | 0.05 | 0.32 | 0.00 | 0.21 | 0.00 | 0.58 |
| p(context) | 0.16 | 0.37 | 0.11 | 0.26 | 0.11 | |

pmi(information,data) = $\log_2$ ( .32 / (.37*.58) ) = .58

*(.57 using full precision)*

Positive

**PPMI(w,context)**

| | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|
| apricot | - | - | 2.25 | - | 2.25 |
| pineapple | - | - | 2.25 | - | 2.25 |
| digital | 1.66 | 0.00 | - | 0.00 | - |
| information | 0.00 | 0.57 | - | 0.47 | - |

| | x | w | y | z | m |
|---|---|---|---|---|---|
| a | 3 | 0 | 4 | 0 | 0 |
| b | 1 | 0 | 5 | 0 | 1 |
| c | 0 | 2 | 1 | 3 | 0 |
| d | 5 | 2 | 0 | 0 | 1 |
| e | 2 | 0 | 1 | 1 | 0 |

Represent the word "c" as vector using PPMI weights.

Total = 32

| | x | w | y | z | m | | |
|---|---|---|---|---|---|---|---|
| a | 3 | 0 | 4 | 0 | 0 | 7 | $7/32 = 0.22$ |
| b | 1 | 0 | 5 | 0 | 1 | 7 | $7/32 = 0.22$ |
| c | 0 | 2 ($2/32 = 0.06$) | $1/32 = 0.03$ | 3 ($0.09$) | 0 | 6 | $6/32 = 0.19$ |
| d | 5 ($\frac{5}{32} = 0.156$) | 2 | 0 | 0 | 1 | 8 | $8/32 = 0.25$ |
| e | 2 | 0 | 1 | 1 | 0 | 4 | $4/32 = 0.125$ |
| | 11 ($0.34 = 11/32$) | 4 ($4/32$ $(0.125)$) | 11 ($11/32$ $(0.34)$) | 4 ($4/32$ $(0.125)$) | 2 ($2/32 = 0.06$) | | |

Represent the word "c" as vector using PPMI weights.

vector of "c"

| x | w | y | z | m |
|---|---|---|---|---|
| 0 | $\log_2 \frac{0.06}{(0.19)(0.125)}$ | $\log_2 \frac{0.03}{(0.19)(0.34)}$ | $\log_2 \frac{0.09}{(0.19)(0.125)}$ | 0 |
| 0 | $= \log_2 (2.6)$ | $= \log_2 (0.5)$ | $= \log_2 3.9$ | 0 |
| | $= 1.4$ | 0 | 1.9 | |

# Weighting PMI

PMI is biased toward infrequent events

- Very rare words have very high PMI values

Two solutions:

- Give rare words slightly higher probabilities

# Weighting PMI: Giving rare context words slightly higher probability

Raise the context probabilities to $\alpha = 0.75$:

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

This helps because $P_\alpha(c) > P(c)$ for rare *c*

Consider two events, P(a) = .99 and P(b)=.01

$$P_\alpha(a) = \frac{99^{.75}}{99^{.75}+1^{.75}} = .97 \quad P_\alpha(b) = \frac{1^{.75}}{99^{.75}+1^{.75}} = .03$$

# Summary for Part I

- Idea of Embeddings: Represent a word as a function of its distribution with other words
- Tf-idf
- Cosines
- PPMI