Distributional Word Similarity

External → [NN, ESA]

Introspective → C

"Birds of the same feather flock together"

looking within the corpus.

Example

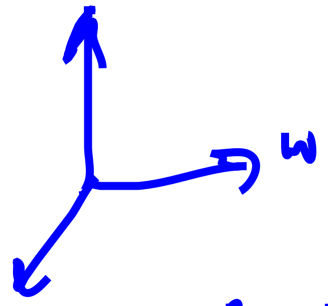
- A bottle of ***tesgüino*** is on the table
  Everybody likes ***tesgüino***
  Tesgüino makes you drunk
  We make ***tesgüino*** out of corn.
- From context words humans can guess ***tesgüino*** means
  - an alcoholic beverage like **beer**
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

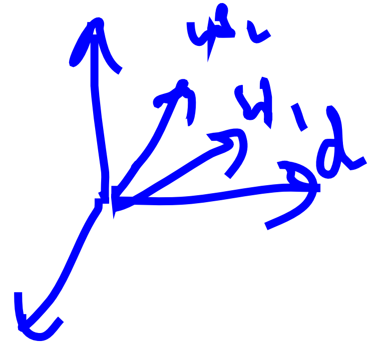
Ack: Slides by Jurafsky (Online Lectures)

Term doc matrices

- Each cell: count of term *t* in a document *d*: $\mathrm{tf}_{t,d}$:
  - Each document is a count vector in $\mathbb{N}^{V}$: a column below

	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





Ack: Slides by Jurafsky (Online Lectures)

Term doc matrices

- 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

Ack: Slides by Jurafsky (Online Lectures)

Words across docs

- Each word is a $\boxed{\text{count vector}}$ in $\mathbb{N}^D$: a row below

	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

Ack: Slides by Jurafsky (Online Lectures)

Term context matrix

- Instead of using entire documents, use smaller contexts
  - Paragraph
  - Window of 10 words
- A word is now defined by a vector over counts of context words

Ack: Slides by Jurafsky (Online Lectures)

Example

Sample contexts: 20 words (Brown corpus)

- equal amount of sugar, a sliced lemon, a tablespoonful of **apricot** preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened to that of
- of a recursive type well suited to programming on the **digital** computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and **information** necessary for the study authorized in the first section of this

Ack: Slides by Jurafsky (Online Lectures)

Term context matrix

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

Context words

	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	

Ack: Slides by Jurafsky (Online Lectures)

PPMI

- For the term-document matrix
  - We used tf-idf instead of raw term counts
- For the term-context matrix
  - Positive Pointwise Mutual Information (PPMI) is common

Ack: Slides by Jurafsky (Online Lectures)

Definitions

Hong Kong

bread, computer

- **Pointwise mutual information:**
  - Do events x and y co-occur more than if they were independent?

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

bread butter

$$\log \frac{1}{P(\text{butter})}$$

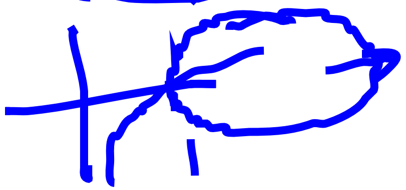
- **PMI between two words:** (Church & Hanks 1989)
  - Do words x and y co-occur more than if they were independent?

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

P(bread)

- **Positive PMI between two words** (Niwa & Nitta 1994)
  - Replace all PMI values less than 0 with zero

$$[P(x,y) < P(x)P(y)]$$



Ack: Slides by Jurafsky (Online Lectures)

- 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

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

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

Ack: Slides by Jurafsky (Online Lectures)

Worked out example

Count(w,context)

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

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

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

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

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

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

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

	computer	data	pinch	result	sugar	p(w)
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	

Ack: Slides by Jurafsky (Online Lectures)

Worked out example

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

p(w,context)	computer	data	pinch	result	sugar	p(w)
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)

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	-

Ack: Slides by Jurafsky (Online Lectures)

Another example

phrase extraction

word 1	word 2	count word 1	count word 2	count of co-occurrences	PMI
puerto	rico	1938	1311	1159	10.0349081703
hong	kong	2438	2694	2205	9.72831972408
los	angeles	3501	2808	2791	9.56067615065
carbon	dioxide	4265	1353	1032	9.09852946116
prize	laureate	5131	1676	1210	8.85870710982
san	francisco	5237	2477	1779	8.83305176711
nobel	prize	4098	5131	2498	8.68948811416
ice	hockey	5607	3002	1933	8.6555759741
star	trek	8264	1594	1489	8.63974676575
car	driver	5578	2749	1384	8.41470768304
it	the	283891	3293296	3347	-1.72037278119
are	of	234458	1761436	1019	-2.09254205335
this	the	199882	3293296	1211	-2.38612756961
is	of	565679	1761436	1562	-2.54614706831
and	of	1375396	1761436	2949	-2.79911817902

Apriori

Ack: Wikipedia page on PMI

Mutual Information

m1 m2 ... mn

Background

Consider a memoriless source m emitting messages $m_1, m_2, \ldots, m_n$ with probabilities P_1, $P_2, \ldots, P_n$, respectively ($P_1 + P_2 + \cdots + P_n = 1$). A **memoriless source** implies that each message emitted is independent of the previous message(s).

The information content of message m_i is I_i, given by

$$I_i = \log \frac{1}{P_i} \quad \text{bits}$$

The probability of occurrence of m_i is P_i. Hence, the mean, or average, information per message emitted by the source is given by $\sum_{i=1}^{n} P_i I_i$ bits. The average information per message of a source m is called its **entropy**, denoted by $H(\text{m})$. Hence,

Randomness

Uncertainty

avg. information per message

$$H(\text{m}) = \sum_{i=1}^{n} P_i I_i \quad \text{bits}$$

$$= \sum_{i=1}^{n} P_i \left[\log \frac{1}{P_i}\right] \quad \text{bits}$$

$$= -\sum_{i=1}^{n} P_i \log P_i \quad \text{bits}$$

Ack: Modern Digital and Analog Communication Systems By B.P. Lathi

Background

Source

Attendance

Channel

$H(D) - H(D|S)$

$H(G) - H(G|A)$

uncertainty about x when I receive y

$$\boxed{H(x|y) = \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)} \quad \text{bits per symbol}}$$

Channel Matrix:

$$\begin{array}{cc} & \text{Outputs} \\ & \begin{array}{cccc} y_1 & y_2 & \cdots & y_s \end{array} \\ \text{Inputs} \begin{array}{c} x_1 \\ x_2 \\ \cdots \\ x_r \end{array} & \begin{pmatrix} P(y_1|x_1) & P(y_2|x_1) & \cdots & P(y_s|x_1) \\ P(y_1|x_2) & P(y_2|x_2) & \cdots & P(y_s|x_2) \\ \cdots & \cdots & \cdots & \cdots \\ P(y_1|x_r) & P(y_2|x_r) & \cdots & P(y_s|x_r) \end{pmatrix} \end{array}$$

$MI = H(x)$ $H(x|y) = 0$

Mutual Information

$$\boxed{I(x; y) = H(x) - H(x|y) \quad \text{bits per symbol}}$$

$MI = 0 \rightarrow H(x|y) = H(x$

$$I(x; y) = \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i)} - \sum_i \sum_j P(x_i, y_j) \log \frac{1}{P(x_i|y_j)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i|y_j)}{P(x_i)}$$

$$= \sum_i \sum_j P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}$$

PMI

Ack: Modern Digital and Analog Communication Systems
By B.P. Lathi

$H(C)$

$H(C) - H(C|E)$

Using syntax to define a word's context

- Zelig Harris (1968)
  - "The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities"
- Two words are similar if they have similar parse contexts
- **Duty** and **responsibility** (Chris Callison-Burch's example)

verbs

objects

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

Ack: Slides by Jurafsky (Online Lectures)

Co-occurrence vectors based on syntactic dependencies

Dekang Lin, 1998 "Automatic Retrieval and Clustering of Similar Words"

- The contexts C are different dependency relations
  - Subject-of- "absorb"
  - Prepositional-object of "inside"
- Counts for the word cell:

	subj-of, absorb	subj-of, adapt	subj-of, behave	...	pobj-of, inside	pobj-of, into	...	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	...	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	...	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

Ack: Slides by Jurafsky (Online Lectures)

PMI applied to dependency relations

Hindle, Don. 1990. Noun Classification from Predicate-Argument Structure. ACL

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"

drink wine

drink it

$$\frac{P(x,y)}{P(x)\cdot P(y)}$$

it

Ack: Slides by Jurafsky (Online Lectures)

A measure of distance in the probabilistic world

Given two probability distributions P and Q, the Kullback-Leibler divergence between P and Q is:

$$\mathrm{KLD}(P, Q) = \sum_{x} P(x) \cdot \log\left(\frac{P(x)}{Q(x)}\right).$$

KL divergence is also referred to as *relative entropy*.

Ack: Slides by Stefan Bűttcher

$$KLD(P, Q) = \sum_x P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

$$= \underbrace{- \sum_x P(x) \log Q(x)}_{\substack{\text{Suboptimal code} \\ [\,Q(x) \text{ used as} \\ \text{Surrogate for } P(x)\,]}} - \left(\underbrace{- \sum_x P(x) \log P(x)}_{\substack{\text{Optimal code} \\ [\,P(x) \text{ used to} \\ \text{encode}\,]}} \right)$$

Intuition: in compression (like Huffman coding), if probability distribution of letters is known, we can get an ideal coding scheme.

Properties

KL divergence has two essential properties:

- $\text{KLD}(P, Q) \geq 0$ for all distributions P, Q.
- $\text{KLD}(P, Q) = 0$ if and only if $P = Q$.

This indicates that it can be used to determine "how far away" a probability distribution P is from another distribution Q. Maybe we can use it as a distance measure between two documents? It would be great if we could use it as a metric!

Ack: Slides by Stefan Bűttcher