COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Words & their Meaning: Distributional Semantics

CMSC 470

Marine Carpuat

Slides credit: Dan Jurafsky

Reminders

- Read the syllabus
- Respond to office hour survey on piazza TODAY
- Get started on homework 1 – due Tue Sep 3 by 1:00pm
    - Only available to students who are officially registered
- If you have conflicts with exam dates, send me private message on piazza by tomorrow Aug 29

Words & their Meaning

2 core issues from an NLP perspective

- **Semantic similarity**: given two words, how similar are they in meaning?
- **Word sense disambiguation**: given a word that has more than one meaning, which one is used in a specific context?

Word similarity for question answering

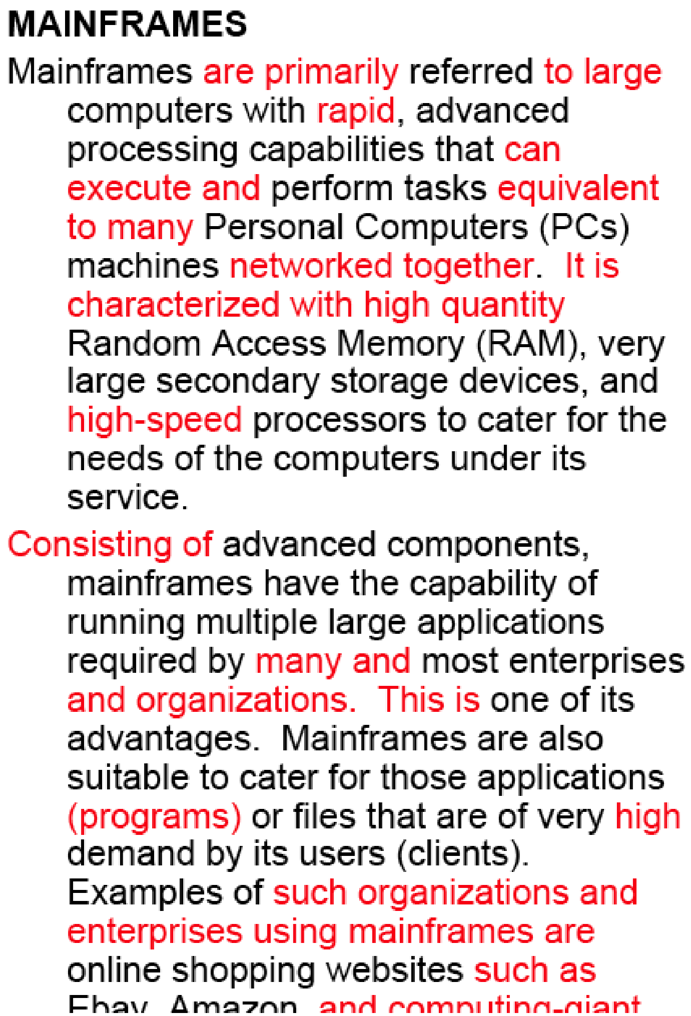
“**fast**” is similar to “**rapid**”

“**tall**” is similar to “**height**”

Question answering:

*Q: “How **tall** is Mt. Everest?”*
*Candidate A: “The official **height** of Mount Everest is 29029 feet”*

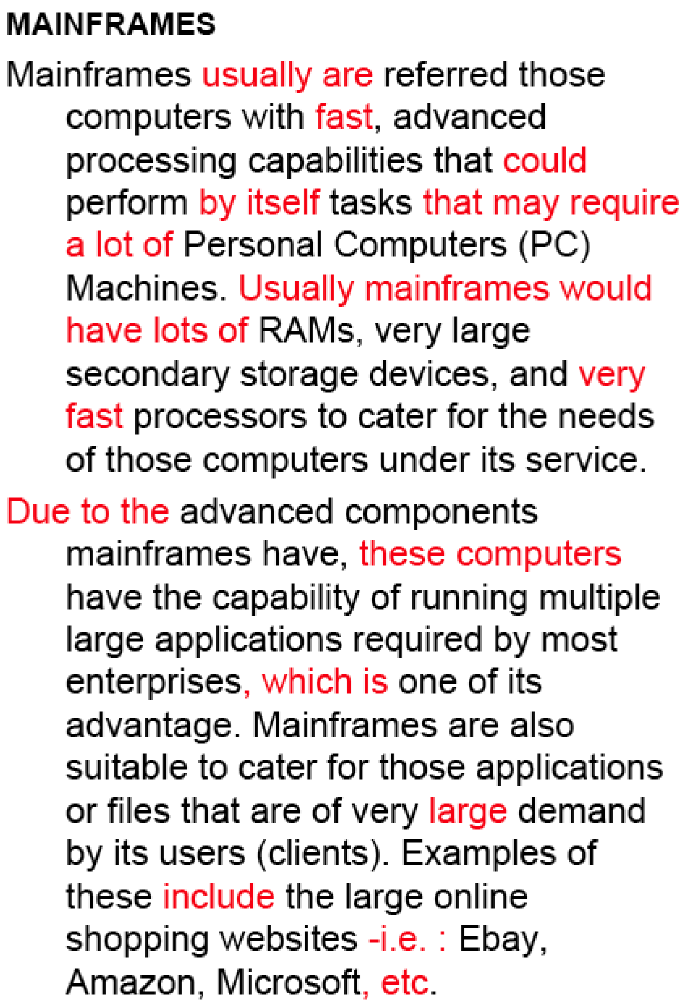
Word similarity
for plagiarism detection



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

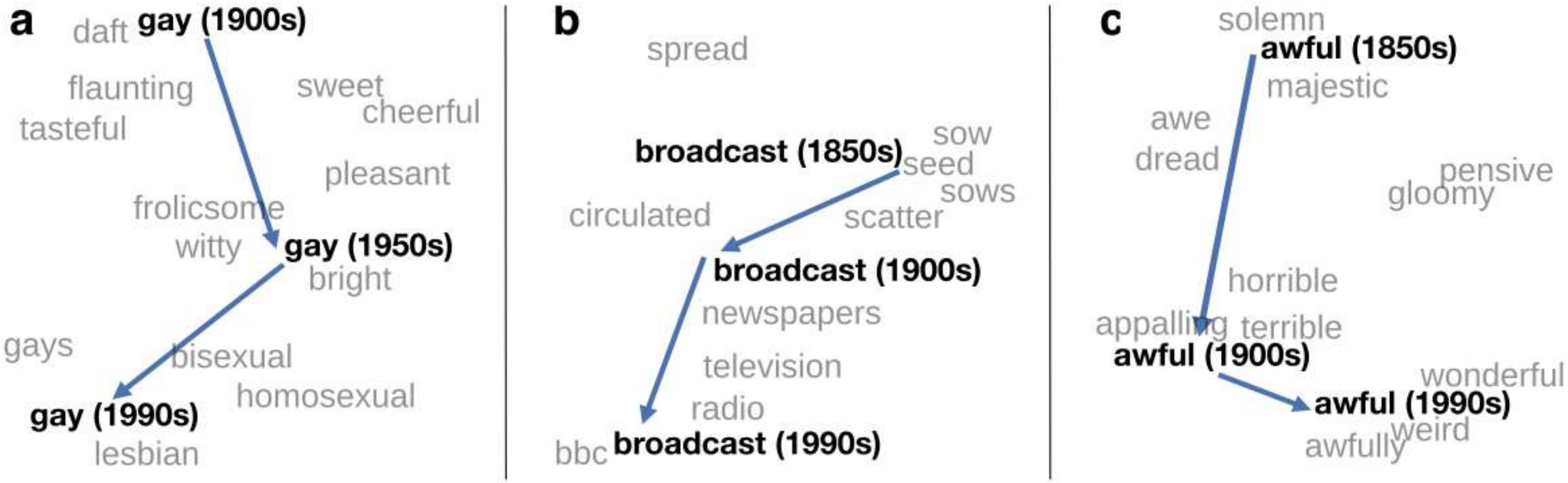


MAINFRAMES

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.

Word similarity for historical linguistics: semantic change over time



~30 million books, 1850-1990, Google Books data

Distributional models of meaning
aka vector-space models of meaning
aka vector semantics

Vector Semantics

Intuition

Zellig Harris (1954):

- “If A and B have almost identical environments we say that they are synonyms.”

J.R. Firth (1957):

- “You shall know a word by the company it keeps!”

tesgüino

A bottle of ***tesgüino*** is on the table

Everybody likes ***tesgüino***

Tesgüino makes you drunk

We make ***tesgüino*** out of corn.

Intuition: two words are similar if they have similar word contexts.

Vector Semantics

- Model the meaning of a word by “embedding” in a vector space.

- The meaning of a word is a vector of numbers
  - Vector models are also called “**embeddings**”.

- Contrast: word represented by a vocabulary index (“word number 545”)

Many varieties of vector models

Sparse vector representations

1. **Mutual-information weighted word co-occurrence matrices**

Dense vector representations:

2. Singular value decomposition (and Latent Semantic Analysis)
3. Neural-network-inspired models (word2vec, skip-grams, CBOW)

Term-document matrix

- 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

The words
in a term-document matrix

- Each word is a 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

The words
in 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

The word-word or word-context matrix

- Instead of entire documents, use smaller contexts
  - Window of ± N words
- A word is now defined by a vector over counts of context words
  - Instead of each vector being of length D
- Each vector is now of length |V|
- The word-word matrix is |V|x|V|

Word-word matrix
Sample contexts ± 7 words

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or 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	
…	…						

Word-word matrix

- The |V|x|V| matrix is very **sparse** (most values are 0)

- The size of windows depends on representation goals
  - The shorter the windows , the more **syntactic** the representation

    ± 1-3 very “syntactic-y”

  - The longer the windows, the more **semantic** the representation

    ± 4-10 more “semantic-y”

Positive Pointwise Mutual Information (PPMI)

Vector Semantics

Problem with raw counts

- Raw word frequency is not a great measure of association between words

- We'd rather have a measure that asks whether a context word is **particularly informative** about the target word.

  - **Positive Pointwise Mutual Information (PPMI)**

Pointwise Mutual Information

Pointwise mutual information (PMI):

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
  - Unreliable without enormous corpora

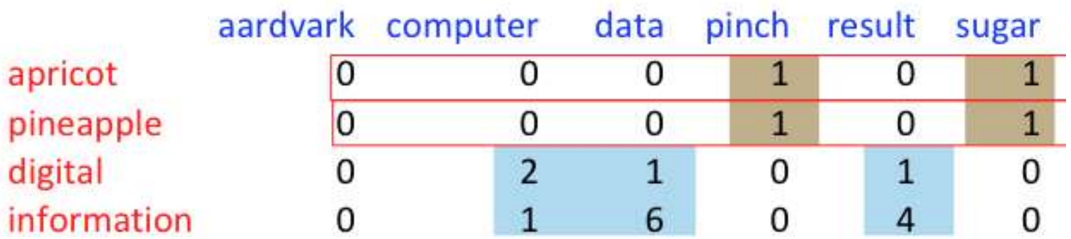
- So we just replace negative PMI values by 0

- Positive PMI (PPMI) between word1 and word2:

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

Computing PPMI on a term-context matrix

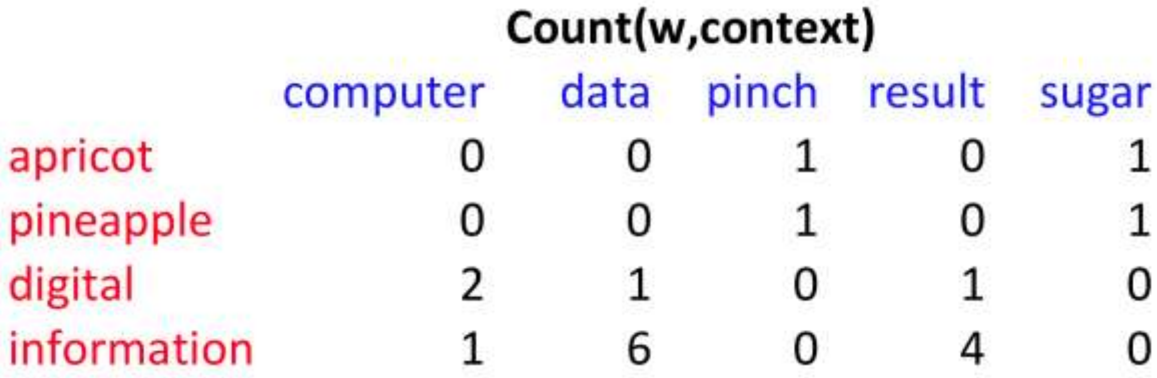
- Matrix F with W rows (words) and C columns (contexts)
- f_{ij} is # of times w_i occurs in context c_j



	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

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

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



Count(w,context)

	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

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

	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	

	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	

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

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	-

Weighting PMI

- PMI is biased toward infrequent events
  - Very rare words have very high PMI values

- Two solutions:
  - Give rare words slightly higher probabilities
  - Use add-*k* smoothing (which has a similar effect)

Weighting PMI: Giving rare context words slightly higher probability

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

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

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

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

$$P_{\alpha}(a) = \frac{.99^{.75}}{.99^{.75} + .01^{.75}} = .97 \quad P_{\alpha}(b) = \frac{.01^{.75}}{.01^{.75} + .01^{.75}} = .03$$

Add-2 smoothing

	Add-2 Smoothed Count(w,context				
	computer	data	pinch	result	sugar
apricot	2	2	3	2	3
pineapple	2	2	3	2	3
digital	4	3	2	3	2
information	3	8	2	6	2

PPMI vs add-2 smoothed PPMI

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	-

PPMI(w,context)[add-2]

	computer	data	pinch	result	sugar
apricot	0.00	0.00	0.56	0.00	0.56
pineapple	0.00	0.00	0.56	0.00	0.56
digital	0.62	0.00	0.00	0.00	0.00
information	0.00	0.58	0.00	0.37	0.00

tf.idf: an alternative to PPMI for measuring association

- The combination of two factors
  - **TF: Term frequency** (Luhn 1957): frequency of the word
  - **IDF: Inverse document frequency** (Sparck Jones 1972)
    - N is the total number of documents
    - df$_i$ = “document frequency of word *i*”
      = # of documents with word *i*

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

  - w_{ij} *= word* i *in document j*

$$w_{ij}=tf_{ij}\,idf_i$$

Measuring similarity: the cosine

Vector Semantics

Cosine for computing similarity

Sec. 6.3

Dot product

Unit vectors

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

v_i is the PPMI value for word *v* in context *i*
w_i is the PPMI value for word *w* in context *i.*

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

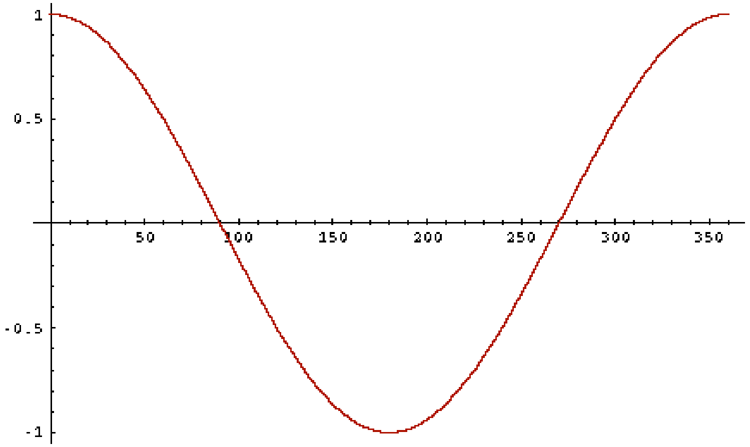
Reminders from linear algebra

$$\text{dot-product}(\vec{v},\vec{w}) = \vec{v}\cdot\vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + ... + v_N w_N$$

$$\text{vector length} \quad |\vec{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

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

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

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

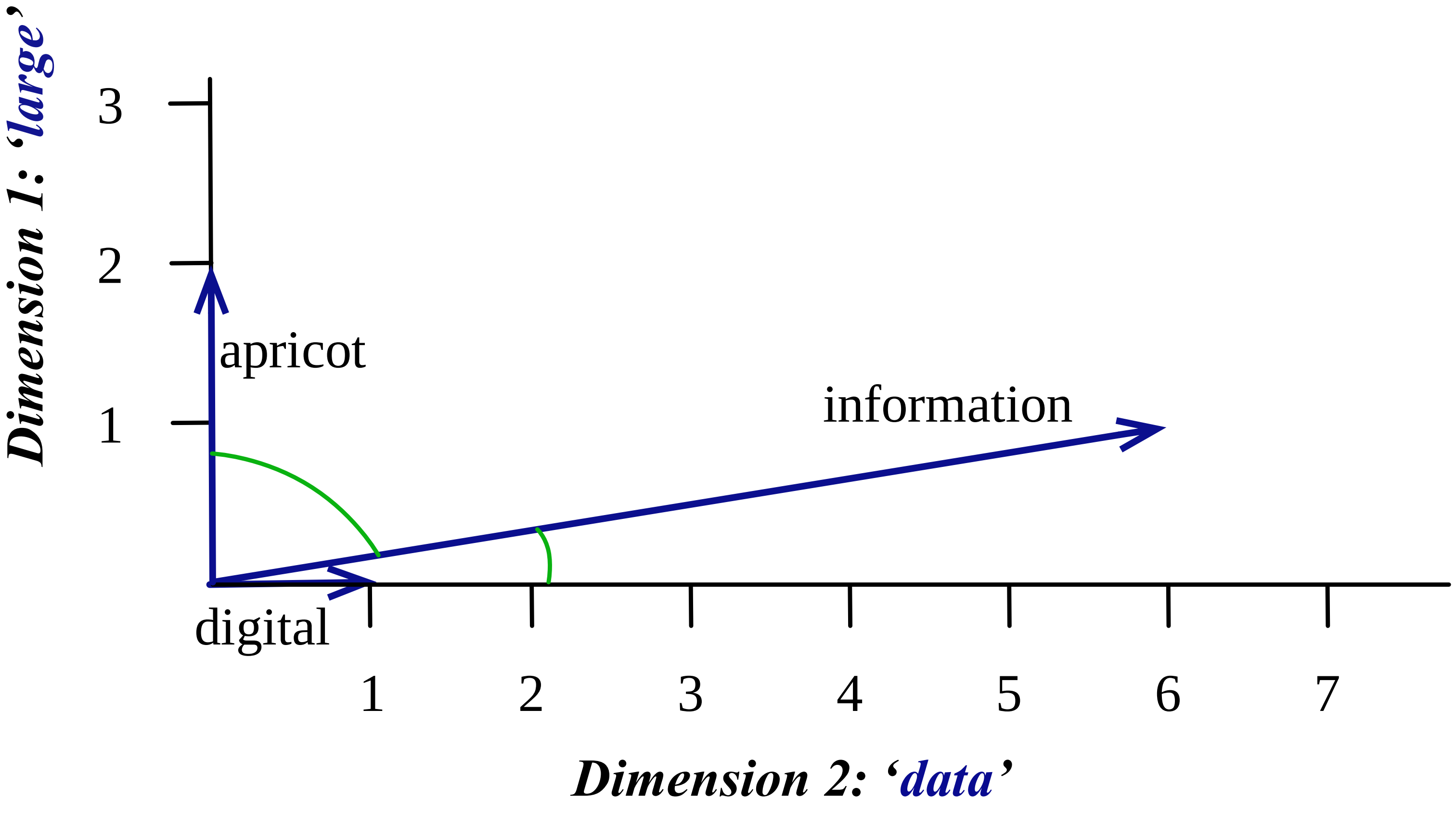
Which pair of words is more similar?

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

Visualizing cosines
(well, angles)



Other possible similarity measures

$$
\begin{aligned}
\text{sim}_{\text{cosine}}(\vec{v},\vec{w}) &= \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \\
\text{sim}_{\text{Jaccard}}(\vec{v},\vec{w}) &= \frac{\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} \max(v_i,w_i)} \\
\text{sim}_{\text{Dice}}(\vec{v},\vec{w}) &= \frac{2\times\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N}(v_i+w_i)} \\
\text{sim}_{\text{JS}}(\vec{v}||\vec{w}) &= D(\vec{v}|\frac{\vec{v}+\vec{w}}{2}) + D(\vec{w}|\frac{\vec{v}+\vec{w}}{2})
\end{aligned}
$$

Evaluating similarity

Vector Semantics

Evaluating similarity

- Extrinsic (task-based, end-to-end) Evaluation:
  - Question Answering
  - Spell Checking
  - Essay grading

- Intrinsic Evaluation:
  - Correlation between algorithm and human word similarity ratings
    - Wordsim353: 353 noun pairs rated 0-10. *sim(plane,car)=5.77*
  - Taking TOEFL multiple-choice vocabulary tests
    - Levied is closest in meaning to:

    imposed, believed, requested, correlated

Words & their Meaning: what you should know

- **Semantic similarity**: quantify how similar in meaning two words are

- **Distributional semantics**
  - Define word meaning based on context
  - Implemented as vector space model: each word is represented by a vector
  - Vector space models can be induced from raw text
    - By defining context (e.g., window, document)
    - By computing association between word & context using metrics such as PPMI or tfidf
    - By handling sparsity (e.g., with add-n smoothing)
  - Given vectors, similarity is computed using cosine or other metrics

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Words & their Meaning: Distributional Semantics

CMSC 470

Marine Carpuat

Slides credit: Dan Jurafsky