Dan Jurafsky and James Martin

Speech and Language Processing

Chapter 6:
Vector Semantics

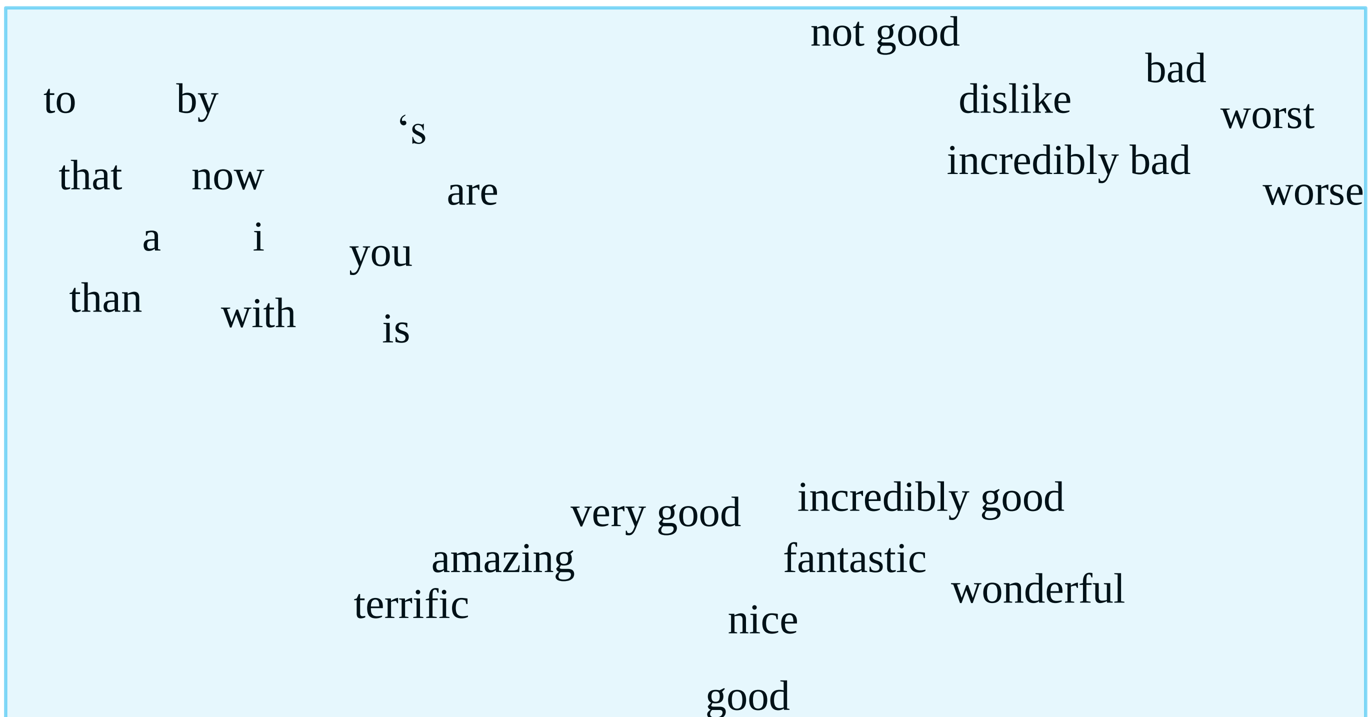
** For AIT590 -*
Only a portion of the contents selected from the original Authors' textbook slides - Part 1

We'll build a new model of meaning focusing on similarity

Each word = a vector

- Not just "word" or word45.

Similar words are "nearby in space"



We 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.

We'll introduce 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

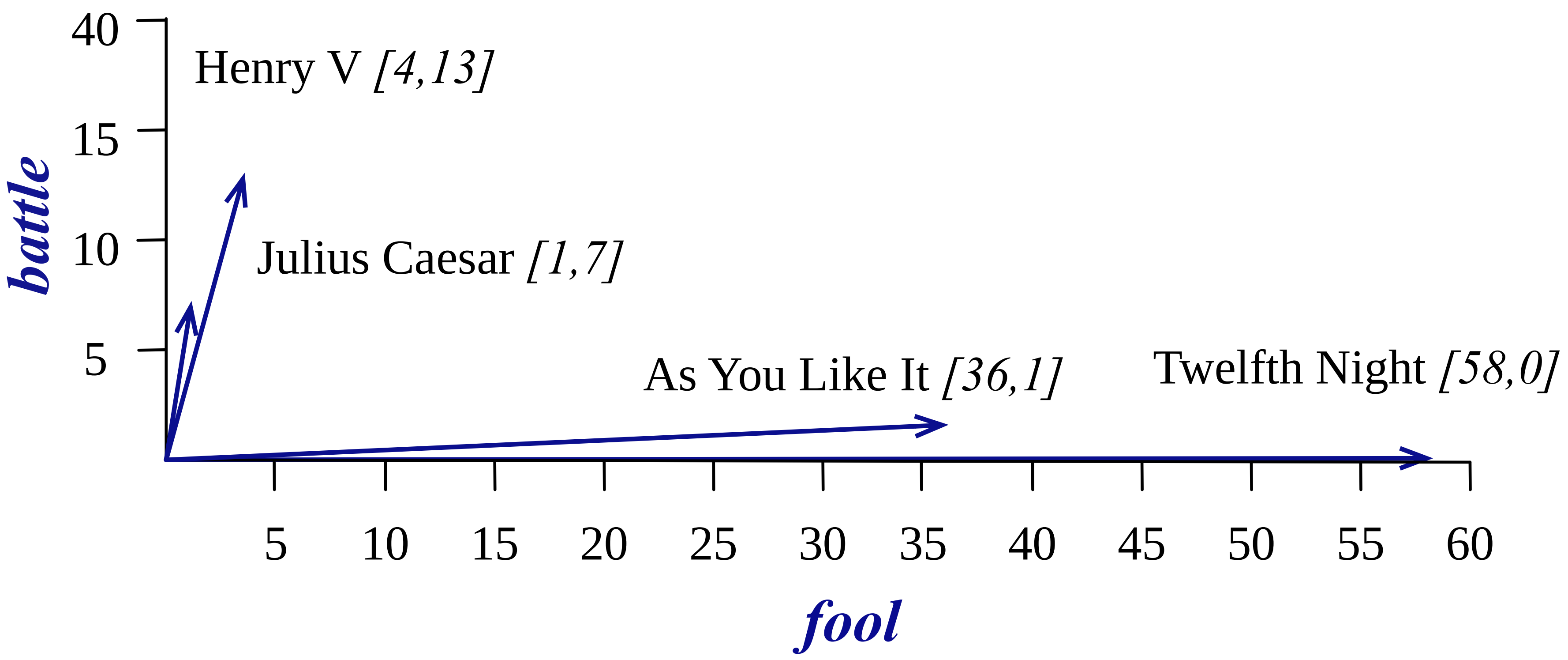
Review: words, vectors, and co-occurrence matrices

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

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"

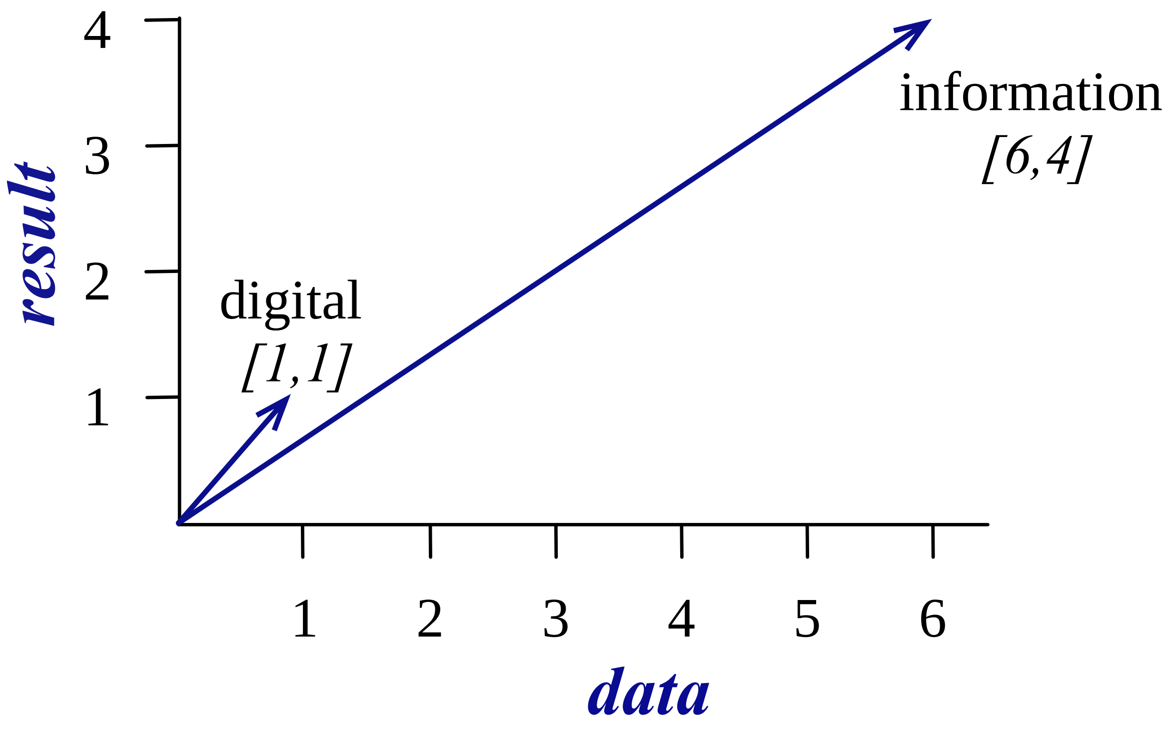
fool is "the kind of word that occurs in comedies, especially Twelfth Night"

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

Reminders from linear algebra

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

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

Cosine for computing similarity

Sec. 6.3

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

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

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

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

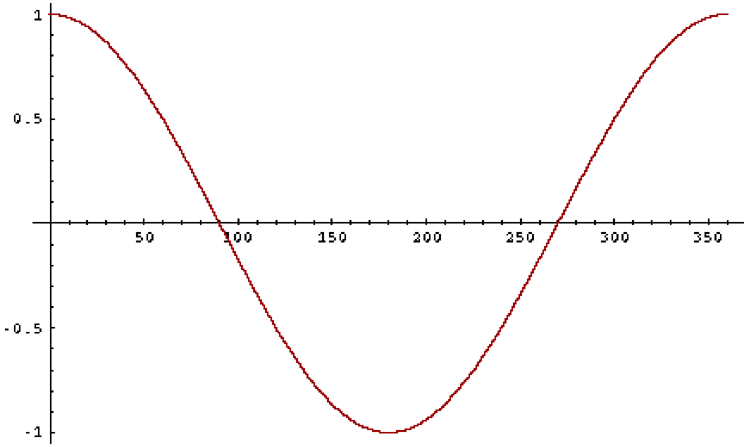
Cos($\vec{v}$,$\vec{w}$) is the cosine similarity of $\vec{v}$ and $\vec{w}$

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

	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) =

$$\frac{1+0+0}{\sqrt{1+0+0}\,\sqrt{1+36+1}} = \frac{1}{\sqrt{38}} = .16$$

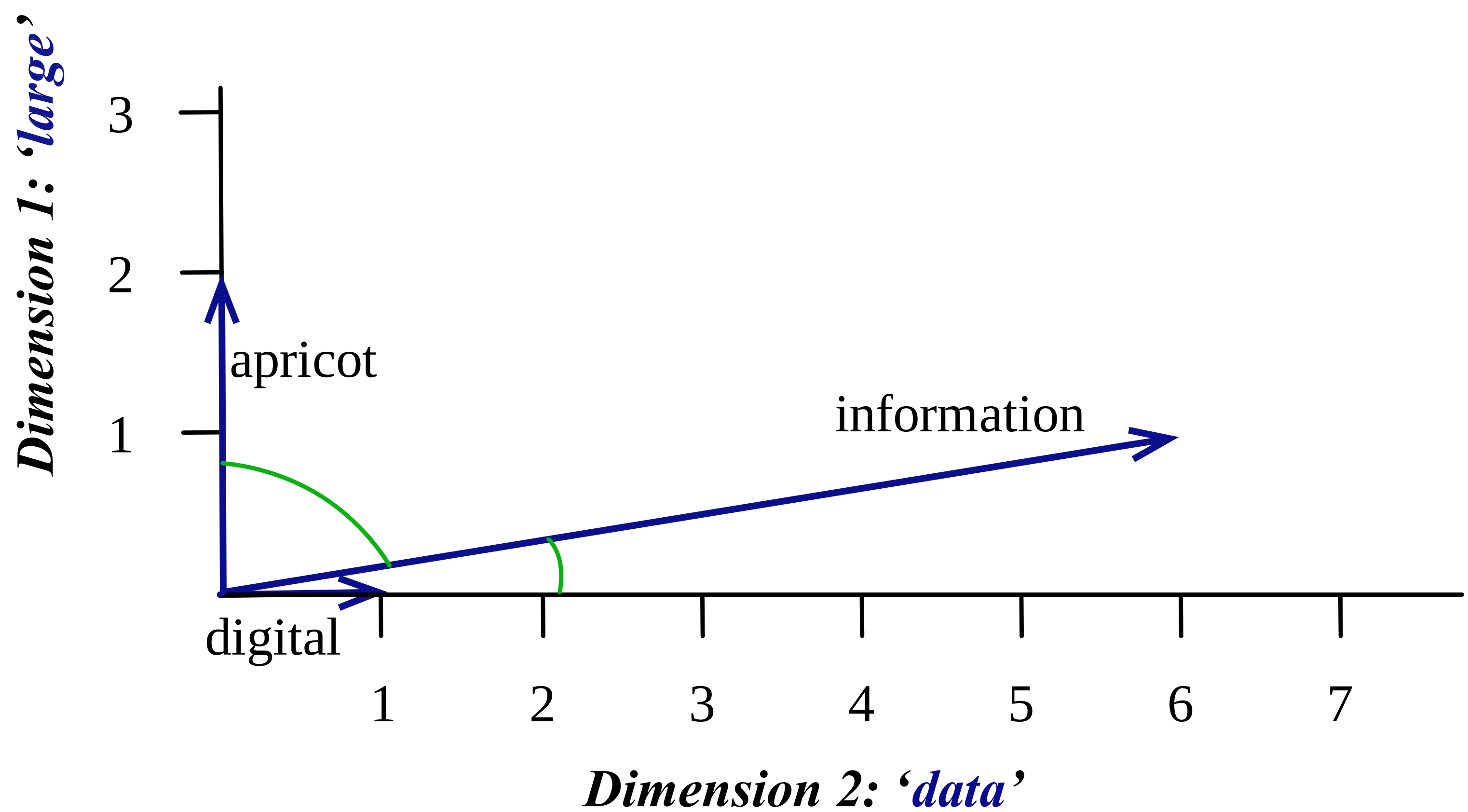
cosine(digital,information) =

$$\frac{0+6+2}{\sqrt{0+1+4}\,\sqrt{1+36+1}} = \frac{8}{\sqrt{38}\sqrt{5}} = .58$$

cosine(apricot,digital) =

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

Visualizing cosines (well, angles)



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!

tf-idf: combine two factors

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

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

Idf: inverse document frequency: tf-

Total # of docs in collection

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

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

# of docs that have word i

tf-idf value for word t in document d:

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

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:

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