Dan Jurafsky and James Martin

Speech and Language Processing

Chapter 6:
Vector Semantics

How can we more robustly match a user's search intent?

We want to **understand** a query, not just do String equals()

- If user searches for [Dell notebook battery size], we would like to match documents discussing "Dell laptop battery capacity"
- If user searches for [Seattle motel], we would like to match documents containing "Seattle hotel"

A pure keyword-matching IR system does nothing to help….

Simple facilities that we have already discussed do a bit to help

- Spelling correction
- Stemming / case folding

But we'd like to better **understand** when query/document match

How can we more robustly match a user's search intent?

Query expansion:

Relevance feedback could allow us to capture this if we get near enough to matching documents with these words

We can also use information on **word similarities**:

- A manual **thesaurus** of synonyms for query expansion
- A **measure of word similarity**
  - Calculated from a big document collection
  - Calculated by query log mining (common on the web)

Document expansion:

Use of **anchor text** may solve this by providing human authored synonyms, but not for new or less popular web pages, or non-hyperlinked collections

Search log query expansion

Context-free query expansion ends up problematic

- [wet ground] ≈ [wet earth]
- So expand [ground] ⇒ [ground earth]
- But [ground coffee] ≠ [earth coffee]

You can learn query context-specific rewritings from search logs by attempting to identify the same user making a second attempt at the same user need

- [Hinton word vector]
- [Hinton word embedding]

In this context, [vector] ≈ [embedding]

- But not when talking about a *disease vector* or C++!

Automatic Thesaurus Generation

Attempt to generate a thesaurus automatically by analyzing a collection of documents

Fundamental notion: similarity between two words

Definition 1: Two words are similar if they co-occur with similar words.

Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.

You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.

Co-occurrence based is more robust, grammatical relations are more accurate.

Words, Lemmas, Senses, Definitions

lemma **sense** **definition**

pepper, *n.*

Pronunciation: Brit. /ˈpɛpə/ , U.S. /ˈpɛpər/

Forms: OE **peopor** (*rare*), OE **pipcer** (transmission error), OE **pipor**, OE **pipur** (*rare*

Frequency (in current use):

Etymology: A borrowing from Latin. **Etymon:** Latin *piper*.

< classical Latin *piper*, a loanword < Indo-Aryan (as is ancient Greek *πέπερι*); compare Sa

I. The spice or the plant.

1.

a. A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

The ground spice from *Piper nigrum* comes in two forms, the more pungent *black pepper*, produced from black peppercorns, and the milder *white pepper*, produced from white peppercorns: see BLACK *adj.* and *n.* Special uses 5a, PEPPERCORN *n.* 1a, and WHITE *adj.* and *n.*[1] Special uses 7b(a).

2.

a. The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate stalked entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

b. Usu. with distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper (1a) in taste and in some cases are used as a substitute for it.

c. *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.*

3. Any of various forms of capsicum, esp. *Capsicum annuum* var. *annuum*. Originally (chiefly with distinguishing word): any variety of the *C. annuum* Longum group, with elongated fruits having a hot, pungent taste, the source of cayenne, chilli powder, paprika, etc., or of the perennial *C. frutescens*, the source of Tabasco sauce. Now frequently (more fully ***sweet pepper***): any variety of the *C. annuum* Grossum group, with large, bell-shaped or apple-shaped, mild-flavoured fruits, usually ripening to red, orange, or yellow and eaten raw in salads or cooked as a vegetable. Also: the fruit of any of these capsicums.

Sweet peppers are often used in their green immature state (more fully ***green pepper***), but some new varieties remain green when ripe.

Lemma pepper

Sense 1: spice from pepper plant

Sense 2: the pepper plant itself

Sense 3: another similar plant (Jamaican pepper)

Sense 4: another plant with peppercorns (California pepper)

Sense 5: *capsicum* (i.e. chili, paprika, bell pepper, etc)

A sense or “concept” is the meaning component of a word

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"*

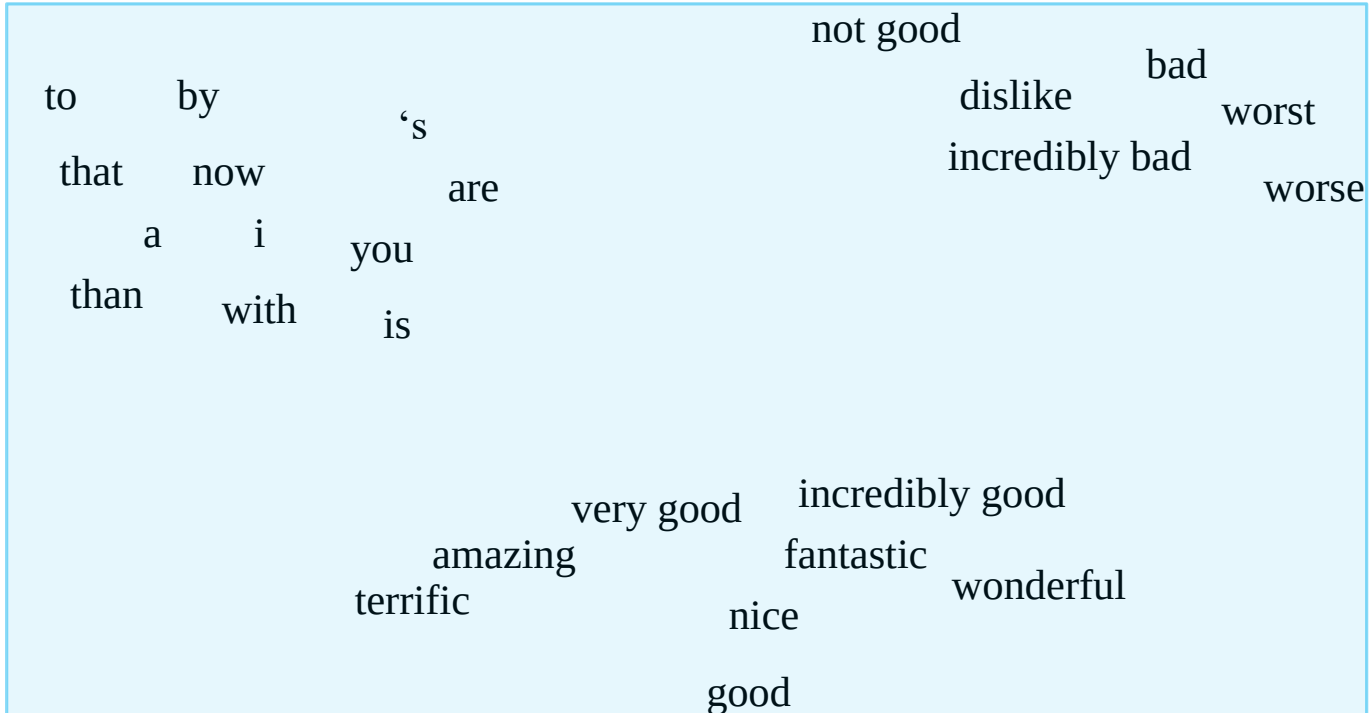
Yamaguchi, Wikimedia Commons, public domain

Build a new model of meaning focusing on similarity

Each word = a vector

- Not just "word" or word45.

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

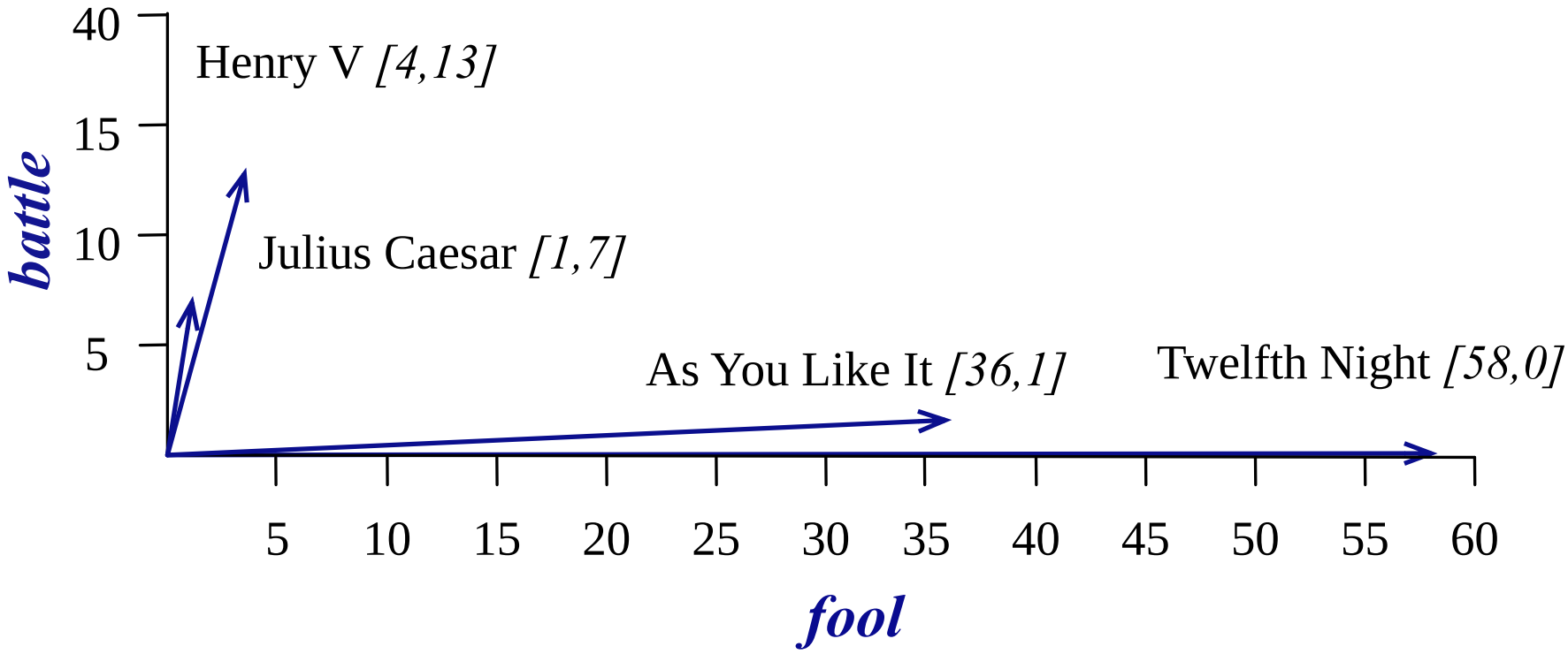
Review: words, vectors, and co-occurrence matrices

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

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"

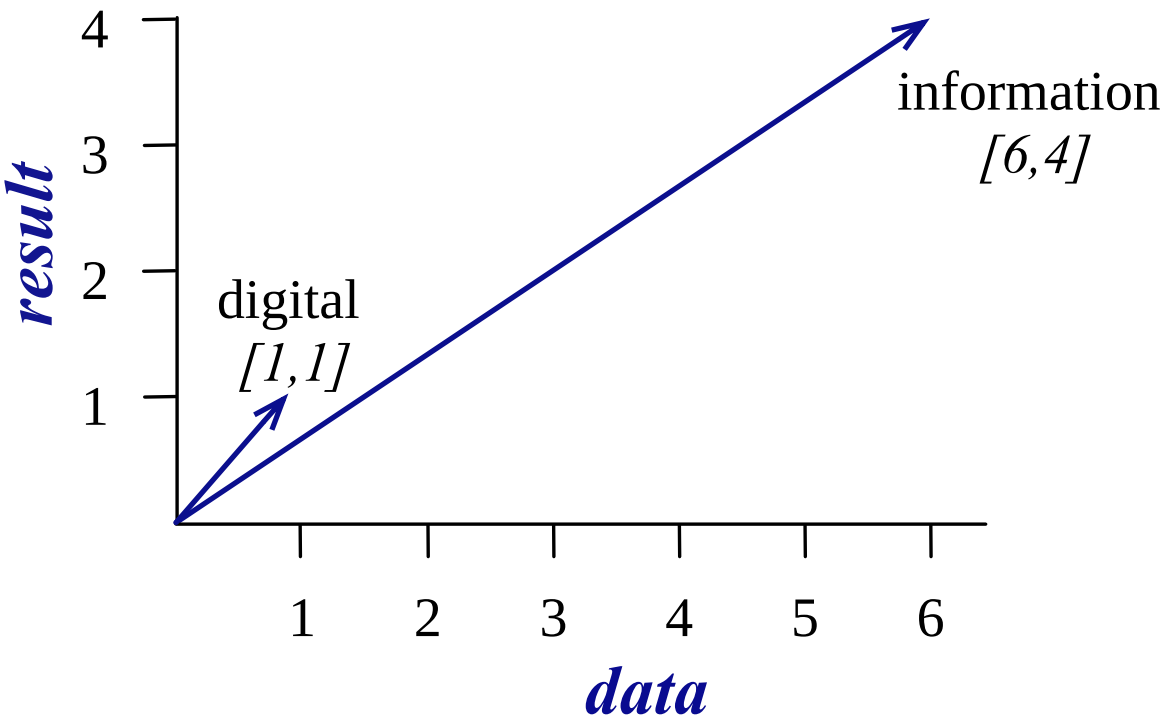
fool is "the kind of word that occurs in comedies, especially Twelfth Night"

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

Cosine for computing similarity

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

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

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

v_i is the count for word v in context i

w_i is the count for word w in context i.

$\vec{v}$ $\vec{w}$ $\vec{v}$ $\vec{w}$

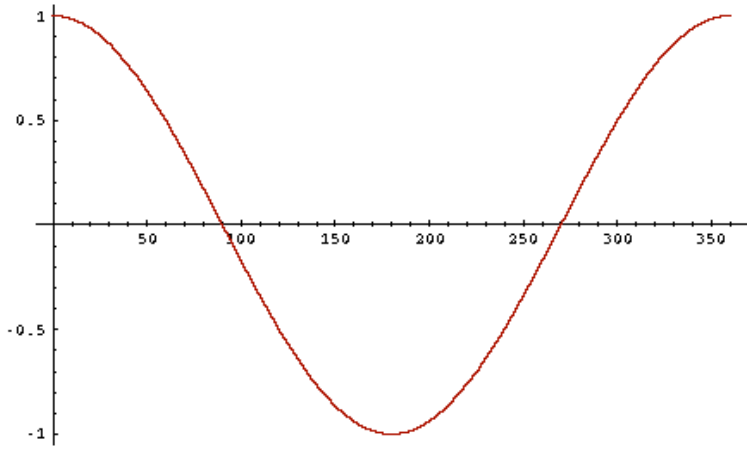
Cos(v,w) is the cosine similarity of v and w

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

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

	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) =

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

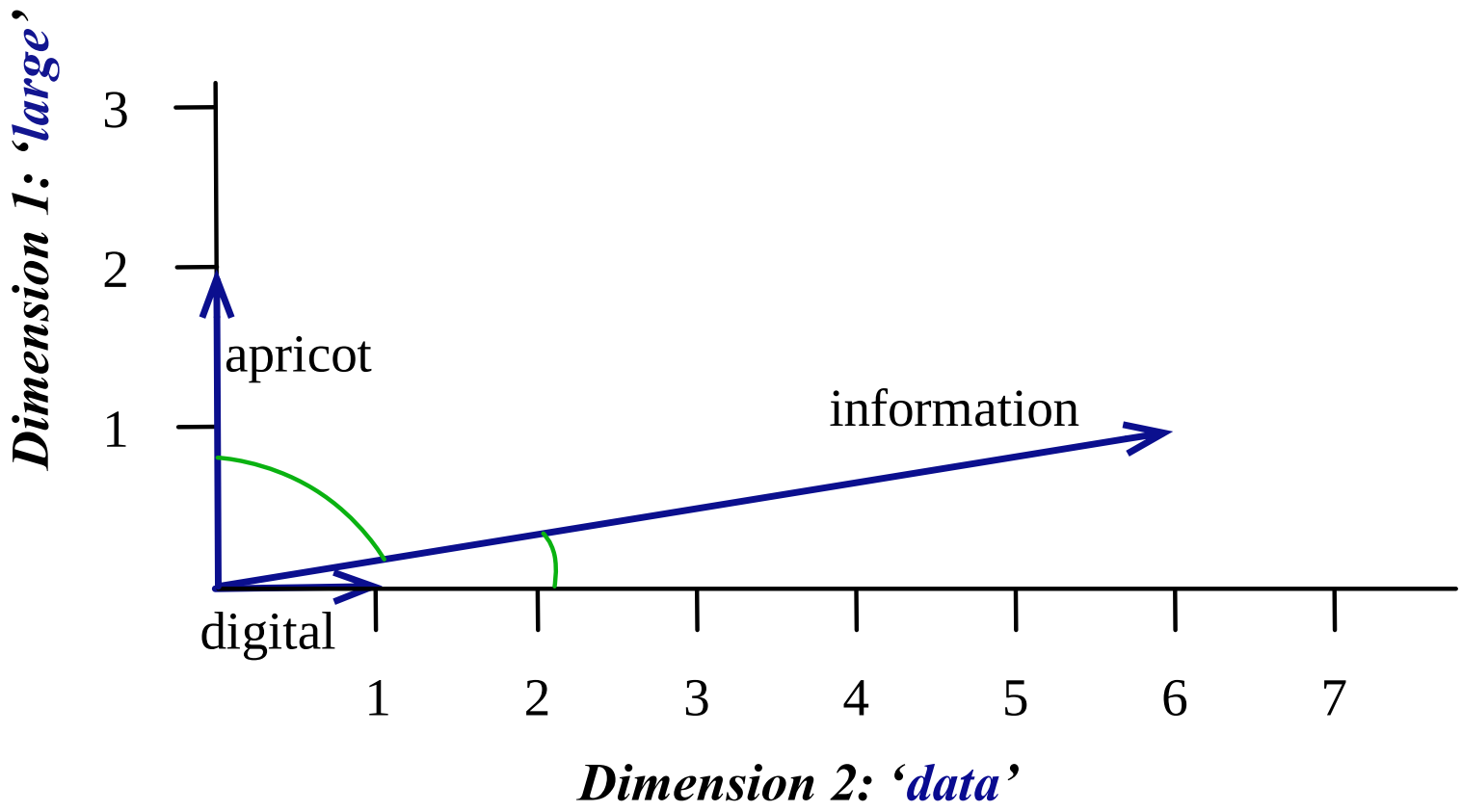
cosine(digital,information) =

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

cosine(apricot,digital) =

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

Visualizing cosines (well, angles)



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-

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

Total # of docs in collection

# of docs that have word i

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

tf-idf value for word t in document d:

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

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

	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

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

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**

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:

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