One-Hot Vector Representations

Sparse vector representation:

- V is the vocabulary
- Each word *w* is mapped to a unique id(*w*) between 1 and |V|.
  - i.e. the position of the word in the vocabulary.
- Represent a word w using a "one-hot" vector **w** of length |V|:
  - **w**[i] = 1, if i = id(w).
  - **w**[i] = 0, otherwise

Example:

- Suppose id(ocean) = 2 and id(water) = 4 and id(laptop) = 5. Then:
  - **w**(ocean) = [0, 1, 0, 0, 0, ..., 0]
  - **w**(water) = [0, 0, 0, 1, 0, ..., 0]
  - **w**(laptop) = [0, 0, 0, 0, 1, ..., 0]

Sparse Representations of Words are Problematic for Machine Learning in NLP

1. Document classification:
   - Bag-of-words representation:
     - each document is the sum of the vectors of all the words in the document, normalized to unit length.
   - Suppose we use *softmax regression* to classify into classes in C.
     - A parameter is needed for each (word, class) pair:
       - => |V| × |C| parameters => 100K × 10 => **1M parameters**.
       - The number of labeled documents needed to train these many parameters may be unfeasible to obtain.
     - If voleyball does not appear in the training documents, but is mentioned in the test document, it will be completely ignored:
       - Even though voleyball is semantically close to basketball, which appeared many times in training documents from the *Sports* category.

Vector Semantics & Embeddings

Word Meaning

What do words mean?

Introductory logic classes:

- The meaning of "dog" is DOG; cat is CAT

  $\forall x\ \mathrm{DOG}(x) \longrightarrow \mathrm{MAMMAL}(x)$

Old joke by Barbara Partee:

- Q: What's the meaning of life?
- A: LIFE

That seems unsatisfactory!

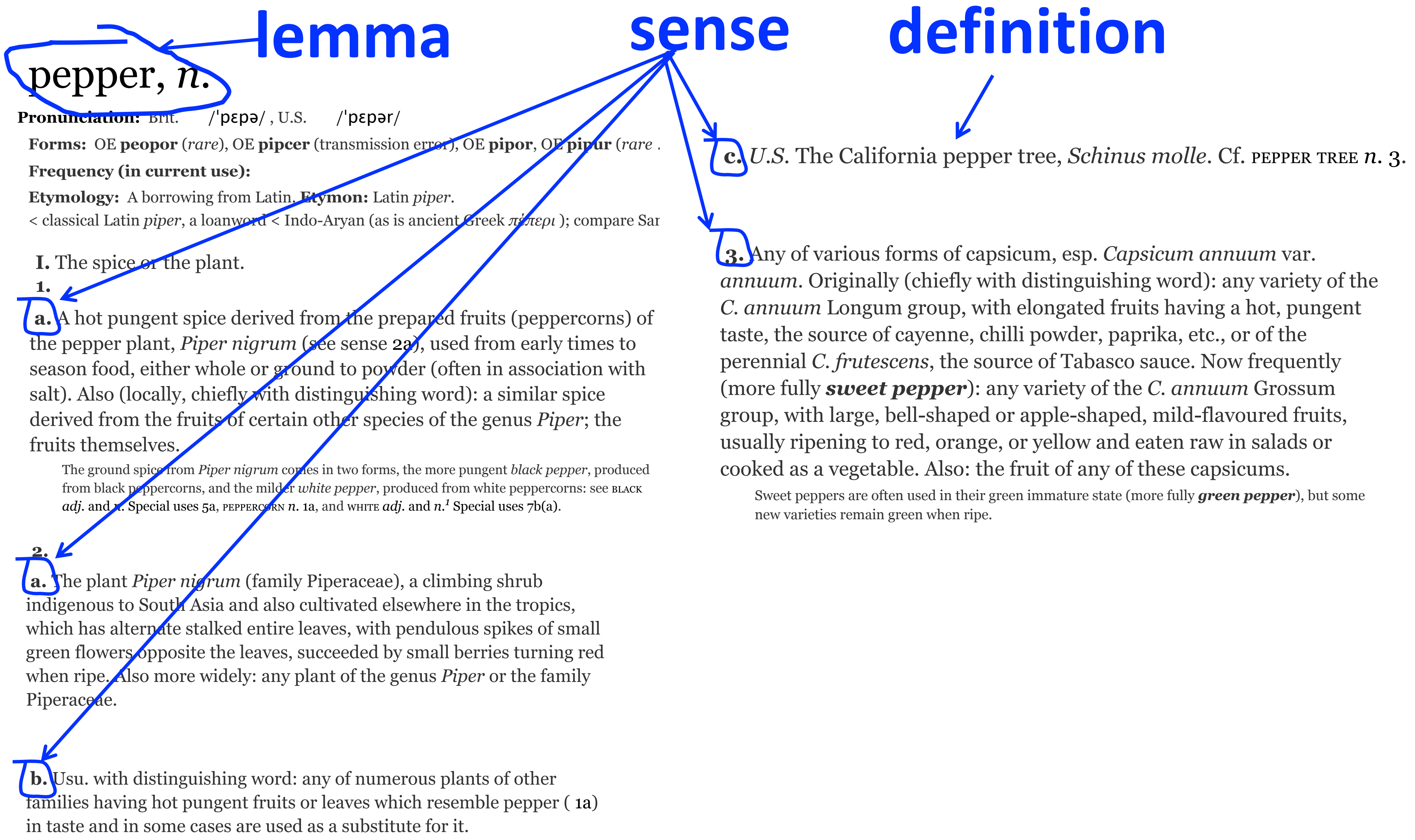
What do words mean?

Next thought: look in a dictionary

http://www.oed.com/

http://wordnetweb.princeton.edu/perl/webwn

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 Sar

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).

c. *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.* 3.

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.

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

Word Sense Disambiguation (WSD)

WSD = mapping a word in context to its correct sense in a dictionary.

I like to season my fries with ground **pepper**.

1. *Sense 1: spice from pepper plant*
2. *Sense 2: the pepper plant itself*
3. *Sense 3: another similar plant (Jamaican pepper)*
4. *Sense 4: another plant with peppercorns (California pepper)*
5. *Sense 5: capsicum (i.e. chili, paprika, bell pepper, etc)*

Word Sense Disambiguation (WSD)

WSD = mapping a word in context to its correct sense in a dictionary.

I like to season my fries with ground **pepper**.

1. *Sense 1: spice from pepper plant*
2. *Sense 2: the pepper plant itself*
3. *Sense 3: another similar plant (Jamaican pepper)*
4. *Sense 4: another plant with peppercorns (California pepper)*
5. *Sense 5: capsicum (i.e. chili, paprika, bell pepper, etc)*

Many papers have been written on WSD (before the advent of word embeddings).

Word Sense Disambiguation (WSD)

WSD = mapping a word in context to its correct sense in a dictionary.

I like to **season** my fries with **ground** **pepper**.

1. *Sense 1: spice from pepper plant*:
   - 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).

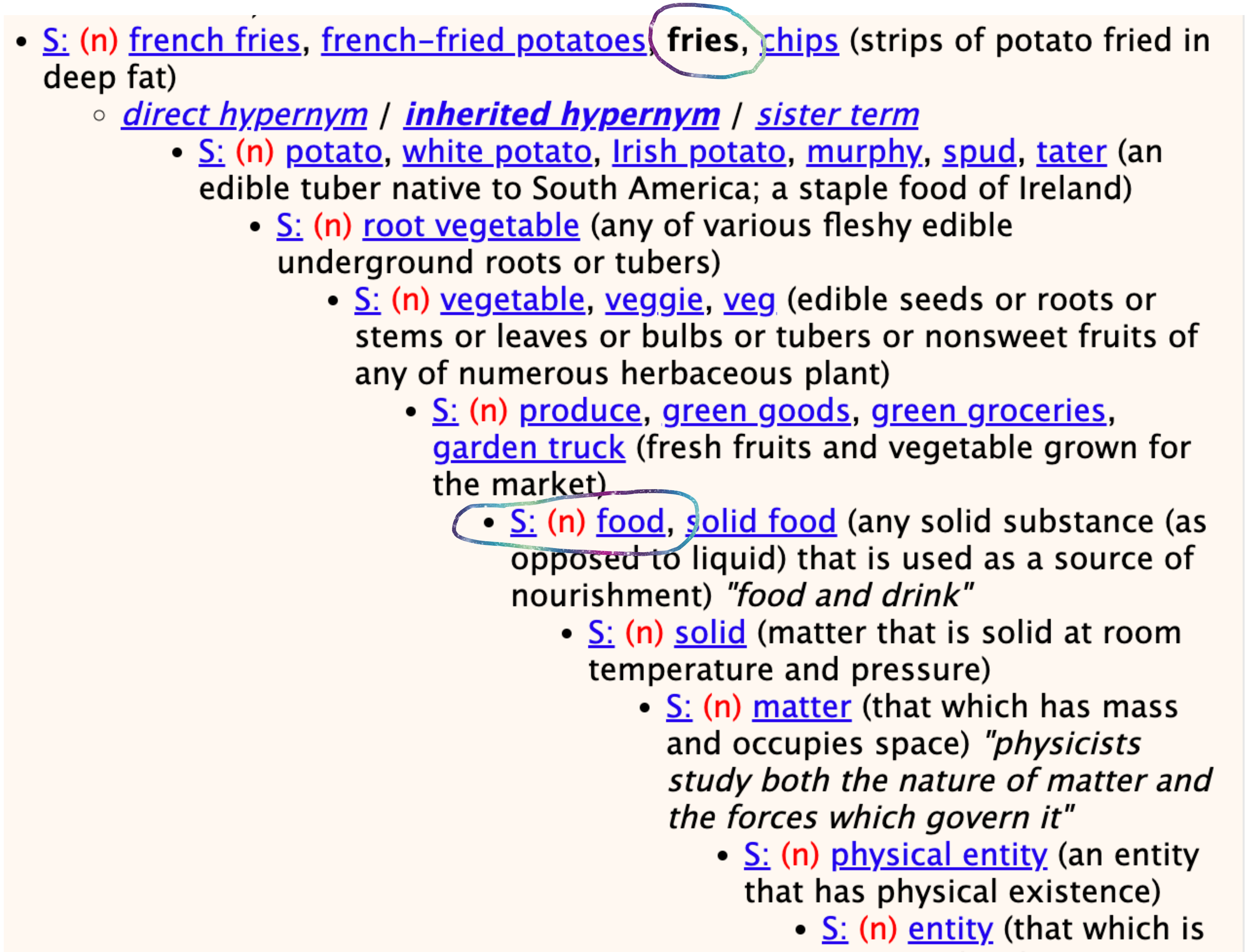
Word Sense Disambiguation (WSD)

WSD = mapping a word in context to its correct sense in a dictionary.

I like to **season** my **fries** with **ground** **pepper**.

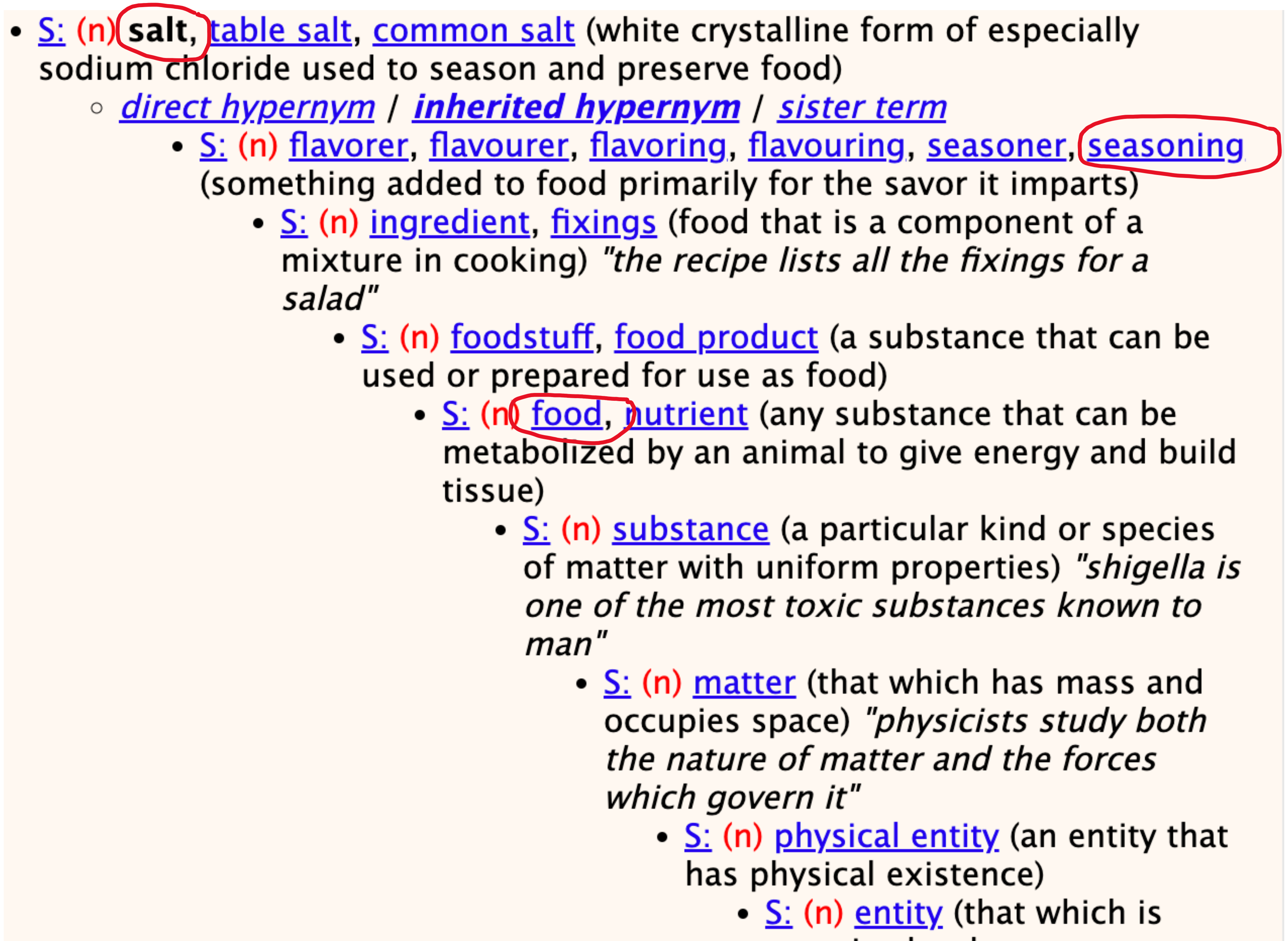
1. *Sense 1: spice from pepper plant*:
   - A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to **season** **food**, either whole or **ground** to powder (often in association with salt).

Word Sense Disambiguation (WSD)



- S: (n) french fries, french-fried potatoes, fries, chips (strips of potato fried in deep fat)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) potato, white potato, Irish potato, murphy, spud, tater (an edible tuber native to South America; a staple food of Ireland)
      - S: (n) root vegetable (any of various fleshy edible underground roots or tubers)
        - S: (n) vegetable, veggie, veg (edible seeds or roots or stems or leaves or bulbs or tubers or nonsweet fruits of any of numerous herbaceous plant)
          - S: (n) produce, green goods, green groceries, garden truck (fresh fruits and vegetable grown for the market)
            - S: (n) food, solid food (any solid substance (as opposed to liquid) that is used as a source of nourishment) *"food and drink"*
              - S: (n) solid (matter that is solid at room temperature and pressure)
                - S: (n) matter (that which has mass and occupies space) *"physicists study both the nature of matter and the forces which govern it"*
                  - S: (n) physical entity (an entity that has physical existence)
                    - S: (n) entity (that which is

Word Sense Disambiguation (WSD)



- S: (n) **salt**, table salt, common salt (white crystalline form of especially sodium chloride used to season and preserve food)
    - *direct hypernym* / ***inherited hypernym*** / *sister term*
        - S: (n) flavorer, flavourer, flavoring, flavouring, seasoner, seasoning (something added to food primarily for the savor it imparts)
            - S: (n) ingredient, fixings (food that is a component of a mixture in cooking) *"the recipe lists all the fixings for a salad"*
                - S: (n) foodstuff, food product (a substance that can be used or prepared for use as food)
                    - S: (n) food, nutrient (any substance that can be metabolized by an animal to give energy and build tissue)
                        - S: (n) substance (a particular kind or species of matter with uniform properties) *"shigella is one of the most toxic substances known to man"*
                            - S: (n) matter (that which has mass and occupies space) *"physicists study both the nature of matter and the forces which govern it"*
                                - S: (n) physical entity (an entity that has physical existence)
                                    - S: (n) entity (that which is

Relations between senses: Synonymy

Synonyms have the same meaning in some or all contexts.

- filbert / hazelnut
- couch / sofa
- big / large
- automobile / car
- vomit / throw up
- water / H_20

Relation: Synonymy

Note that there are probably no examples of perfect synonymy.

- Even if many aspects of meaning are identical
- Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.

Relation: Synonymy?

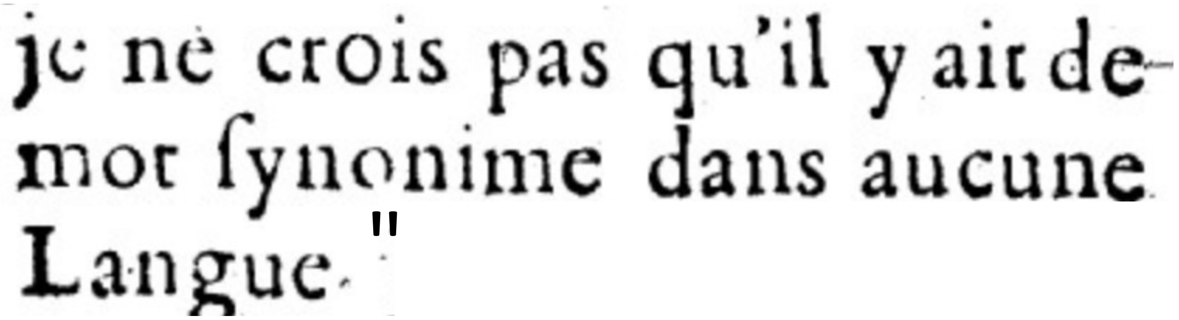
water / H_20
big / large
brave / courageous
strong / powerful

The Linguistic Principle of Contrast

Difference in form => difference in meaning

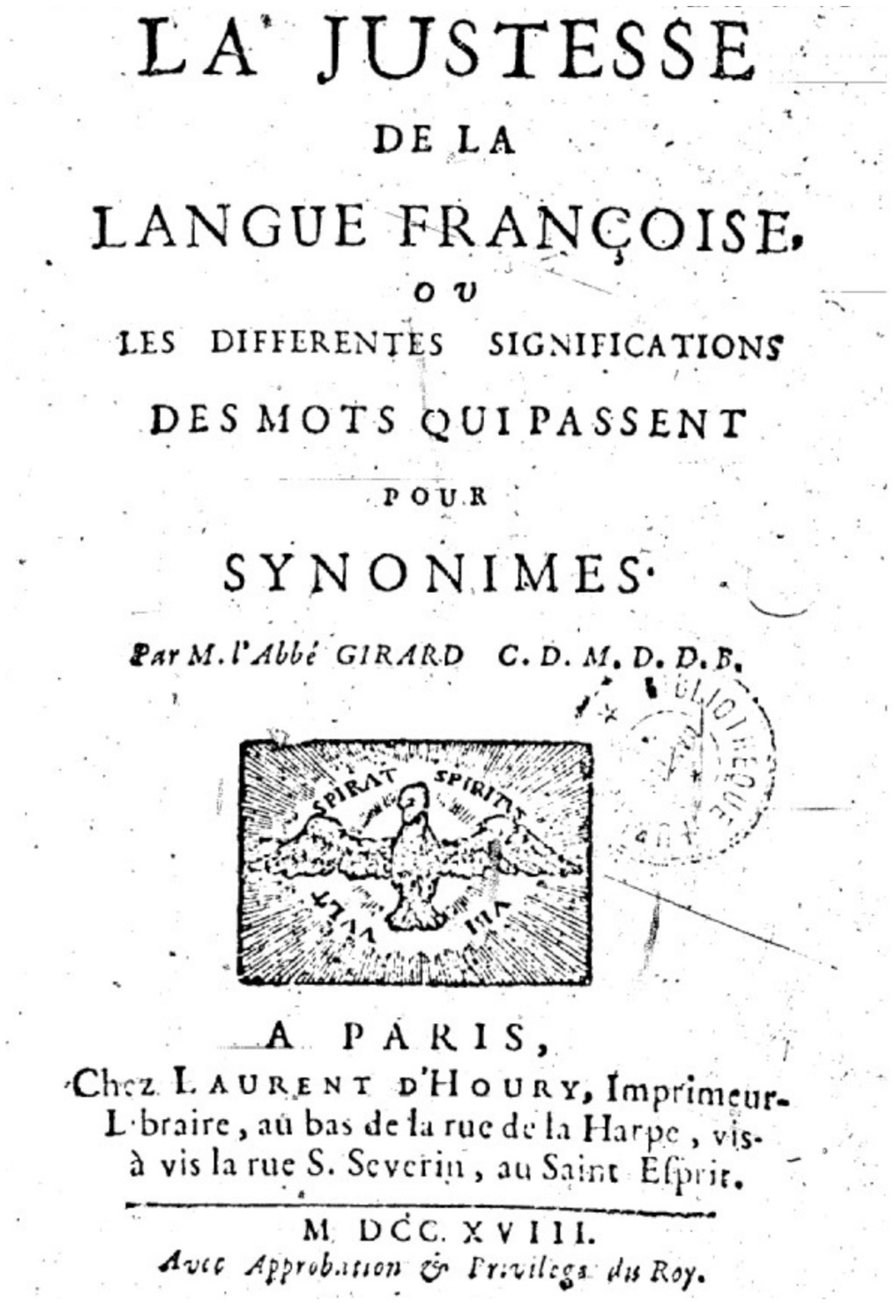
Abbé Gabriel Girard 1718

Re: "exact" synonyms



"je ne crois pas qu'il y ait de-mot ſynonime dans aucune Langue."

[I do not believe that there is a synonymous word in any language]



LA JUSTESSE
DE LA
LANGUE FRANÇOISE,
OU
LES DIFFERENTES SIGNIFICATIONS
DES MOTS QUI PASSENT
POUR
SYNONIMES.

Par M. l'Abbé GIRARD C. D. M. D. D. B.

A PARIS,
Chez LAURENT D'HOURY, Imprimeur-Libraire, au bas de la rue de la Harpe, vis-à vis la rue S. Severin, au Saint Eſprit.

M. DCC. XVIII.
Avec Approbation & Privilege du Roy.

Relation: Similarity

Words with similar meanings. Not synonyms, but sharing some element of meaning

```
car, bicycle
cow, horse
```

Ask humans how similar 2 words are

word1	word2	similarity
vanish	disappear	9.8
behave	obey	7.3
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

SimLex-999 dataset (Hill et al., 2015)

Relation: Word relatedness

Also called "word association”.

Words can be related in any way, perhaps via a *semantic frame* or *field*:

- `car, bicycle:` **similar**
- `car, gasoline:` **related**, not similar

- See also FrameNet:
  - https://framenet.icsi.berkeley.edu/
  - https://en.wikipedia.org/wiki/FrameNet

Semantic field

Words that:

- cover a particular semantic domain.
- bear structured relations with each other.

hospitals

surgeon, scalpel, nurse, anaesthetic, hospital

restaurants

waiter, menu, plate, food, menu, chef

houses

door, roof, kitchen, family, bed

Relation: Antonymy

Senses that are opposites with respect to only one feature of meaning.

Otherwise, they are very similar!

`dark/light` `short/long` `fast/slow` `rise/fall`
`hot/cold` `up/down` `in/out`

More formally: antonyms can

- define a binary opposition or be at opposite ends of a scale
  - `long/short, fast/slow`
- Be *reversives*:
  - `rise/fall, up/down`

Relation: Superordinate/ subordinate

One sense is a **subordinate** of another if the first sense is more specific, denoting a subclass of the other

- *car* is a subordinate of *vehicle*
- *mango* is a subordinate of *fruit*

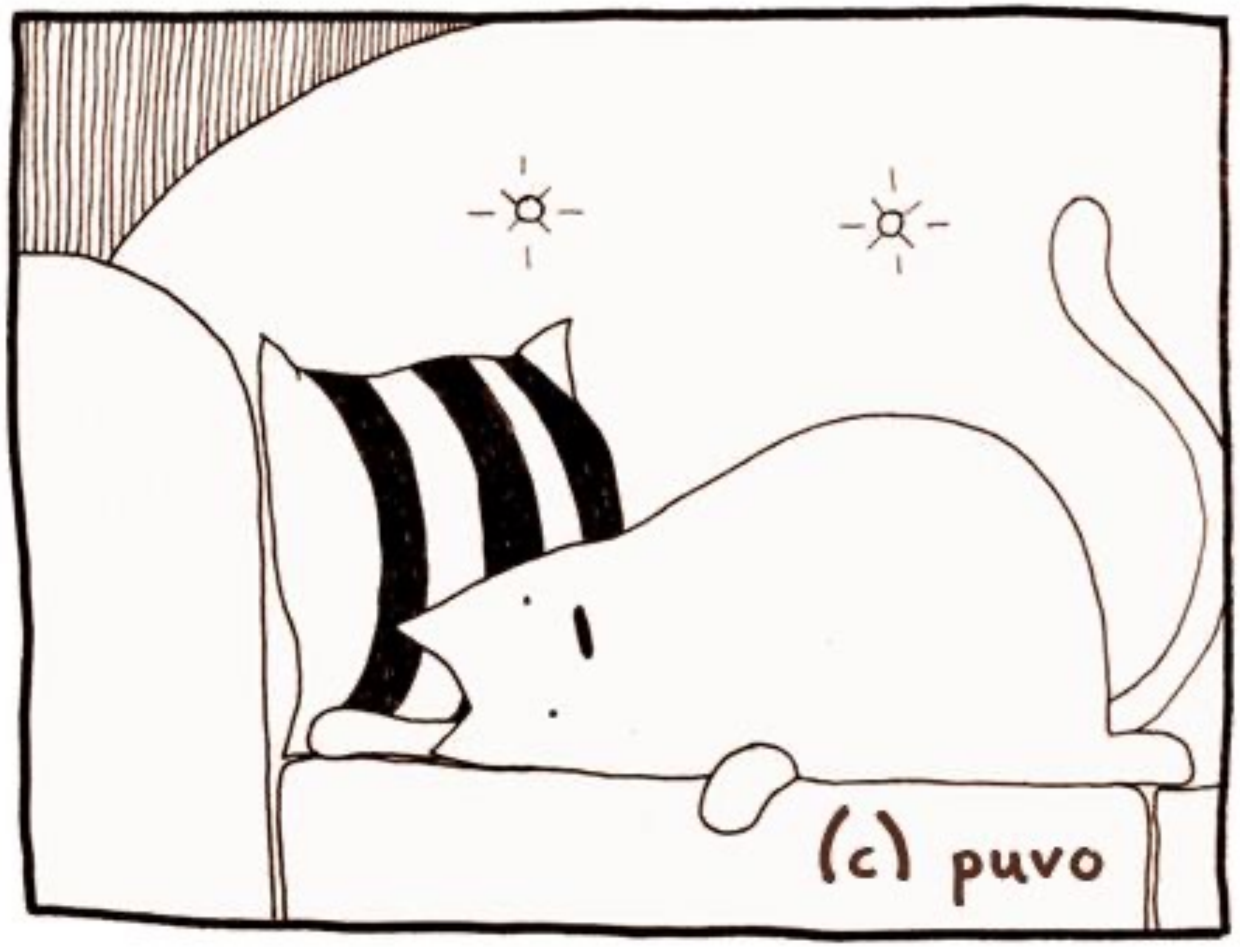
Conversely **superordinate**

- *vehicle* is a superordinate of *car*
- *fruit* is a subodinate of *mango*

Superordinate	vehicle	fruit	furniture
Subordinate	car	mango	chair

These levels are not symmetric

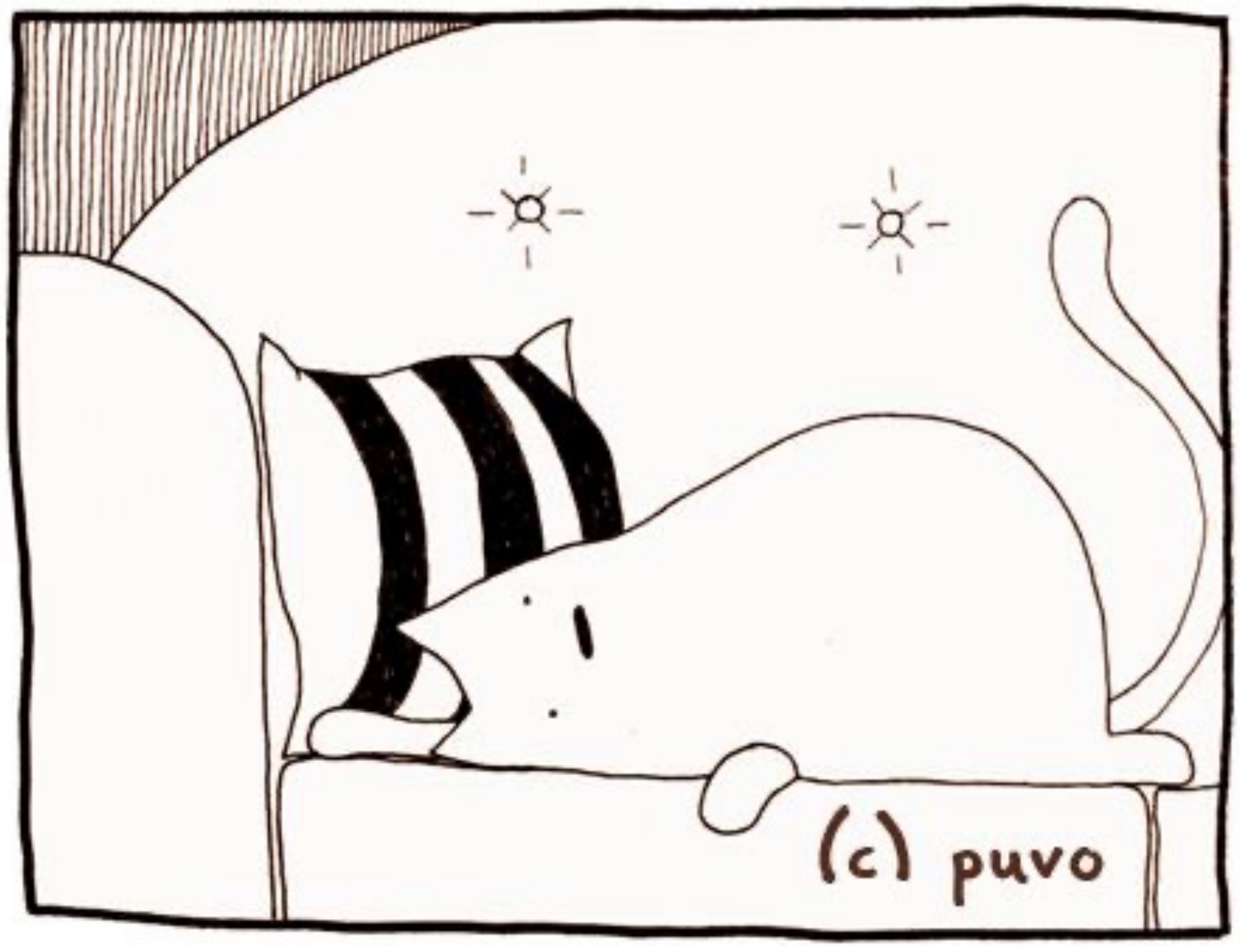
- Subordinate is the **antonym** (reverse) of superordinate.
- They lead to a tree of senses.
- One categorization level is distinguished from the others:
    - The **basic level**.



An *animal* sleeps on the couch.

A **cat** sleeps on the couch.

A *white Persian cat* sleeps on the couch.

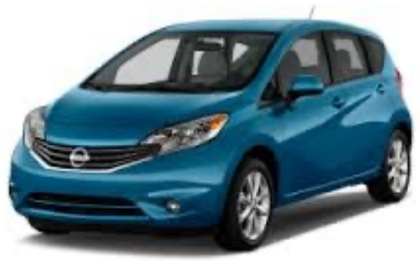
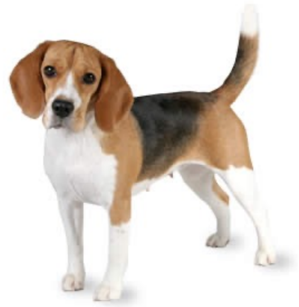
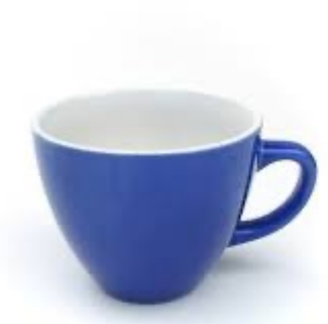
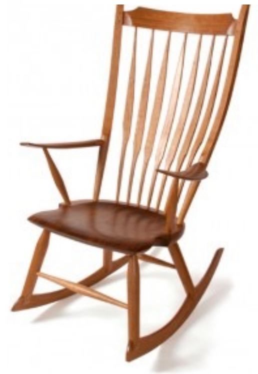
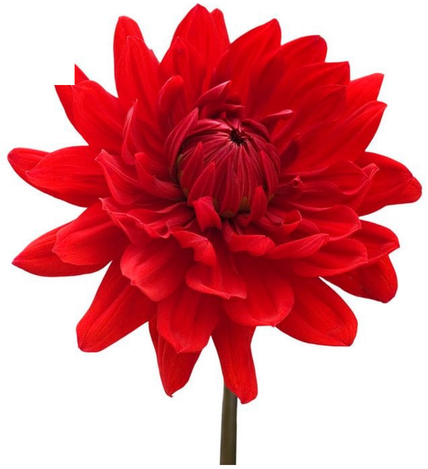


A cat sleeps on the *piece of furniture*.

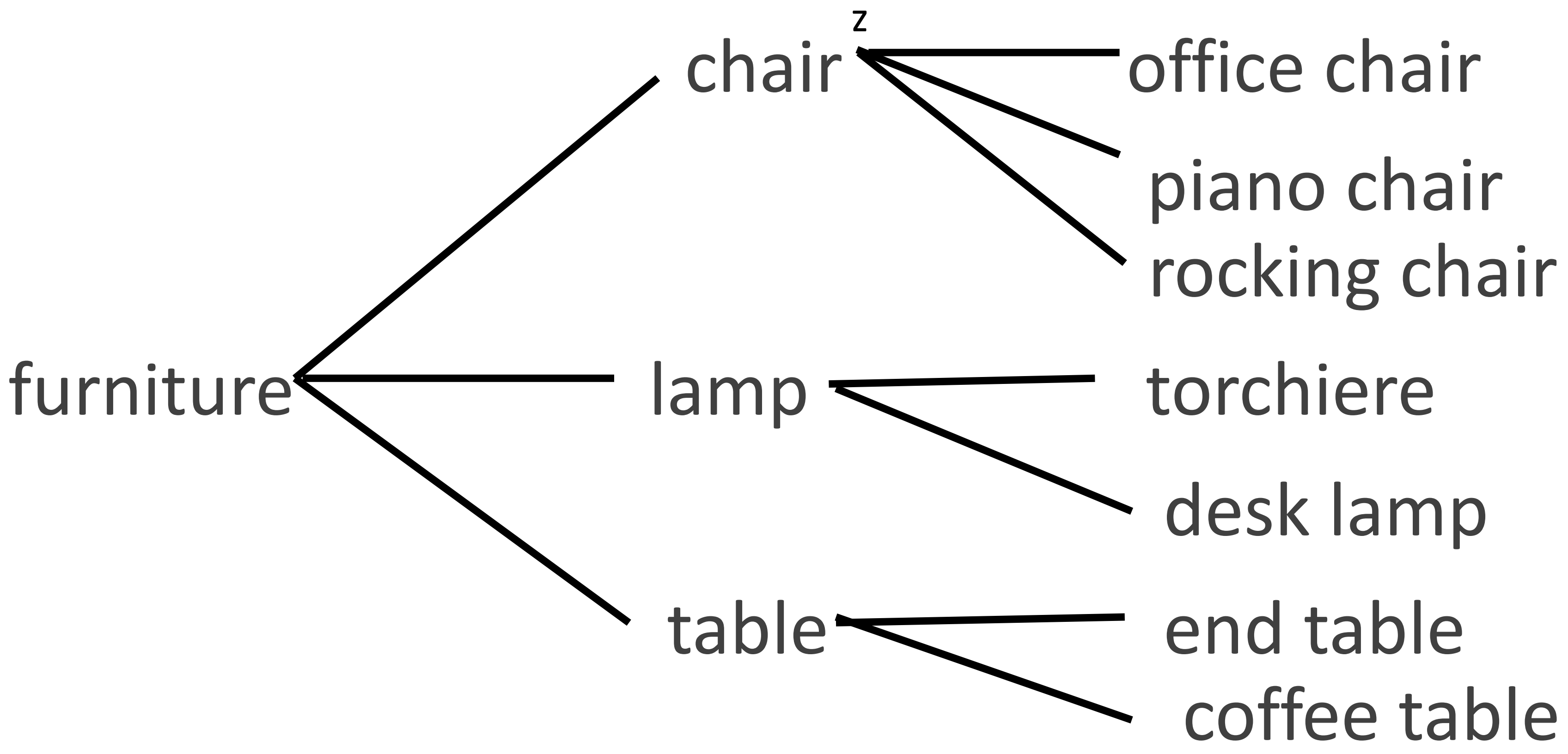
A cat sleeps on the **couch**.

A cat sleeps on the *white leather couch*.

Name these items

Superordinate **Basic** **Subordinate**



Cluster of Interactional Properties

Basic level things are “human-sized”

Consider chairs

- We know how to interact with a chair (sit)
- Not so clear for superordinate categories like furniture
  - “Imagine a furniture without thinking of a bed/table/chair/specific basic-level category”

The basic level

Distinctive actions

Learned earliest in childhood

Names are shortest

Names are most frequent

Connotation (sentiment)

Words have **affective** meanings:

- positive *connotations* (*happy*)
- negative *connotations* (*sad*)

- positive *evaluation* (*great, love*)
- negative *evaluation* (*terrible, hate*).

Connotation

[Osgood et al. (1957)]

Words seem to vary along 3 affective dimensions:

- **valence**: the pleasantness of the stimulus (e.g.: *unhappy* vs. *happy*)
- **arousal**: the intensity of emotion provoked by the stimulus (*excited* vs. *calm*)
- **dominance**: the degree of control exerted by the stimulus (*controlling* vs. *awed*)

	Valence	Arousal	Dominance
courageous	8.05	5.5	7.38
music	7.67	5.57	6.5
heartbreak	2.45	5.65	3.58
cub	6.71	3.95	4.24

So far

Concepts or word senses

- Have a complex many-to-many association with **words** (homonymy, multiple senses)

Have relations with each other

- Synonymy
- Antonymy
- Similarity
- Relatedness
- Superordinate/subordinate, basic level
- Connotation

What do words mean?

Look in a dictionary:

http://www.oed.com/

http://wordnetweb.princeton.edu/perl/webwn

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)

- Words are defined using words which are defined using words which are defined using words …
- Recursive definition => use recursive algorithms for finding representations of word meaning?

Vector Semantics & Embeddings

Word Meaning

Vector Semantics & Embeddings

It's hard to define a concept

But how to define a concept?

Classical ("Aristotelian") Theory of Concepts

The meaning of a word:

a concept defined by **necessary** and **sufficient** conditions

A **necessary** condition for being an X is a condition C that X must satisfy in order for it to be an X.

- If not C, then not X
- "Having four sides" is necessary to be a square.

A **sufficient** condition for being an X is condition such that if something satisfies condition C, then it must be an X.

- If and only if C, then X
- The following necessary conditions, jointly, are sufficient to be a square
  - x has (exactly) four sides
  - each of x's sides is straight
  - x is a closed figure
  - x lies in a plane
  - each of x's sides is equal in length to each of the others
  - each of x's interior angles is equal to the others (right angles)
  - the sides of x are joined at their ends

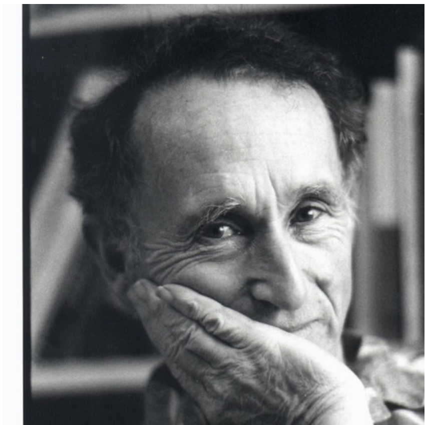
Example from Norman Swartz, SFU

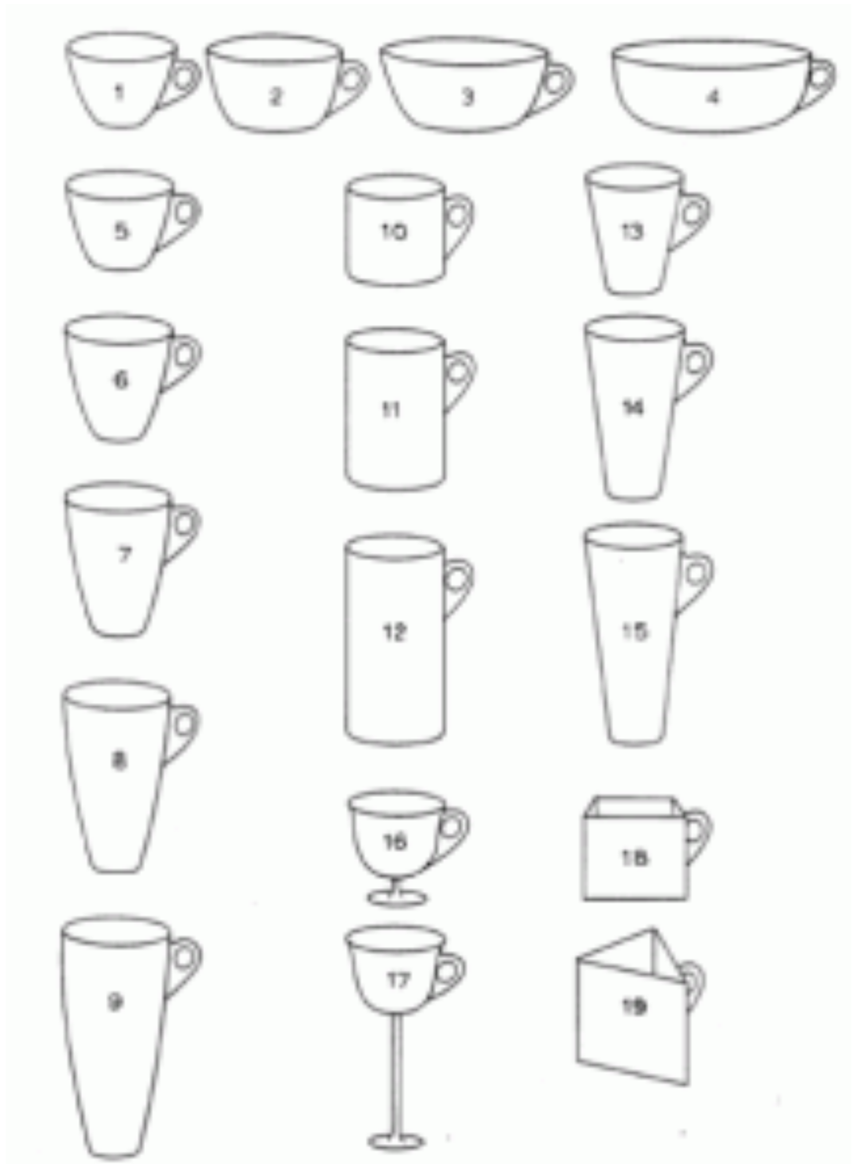
Problem 1: The features are complex & may be context-dependent

William Labov. 1975

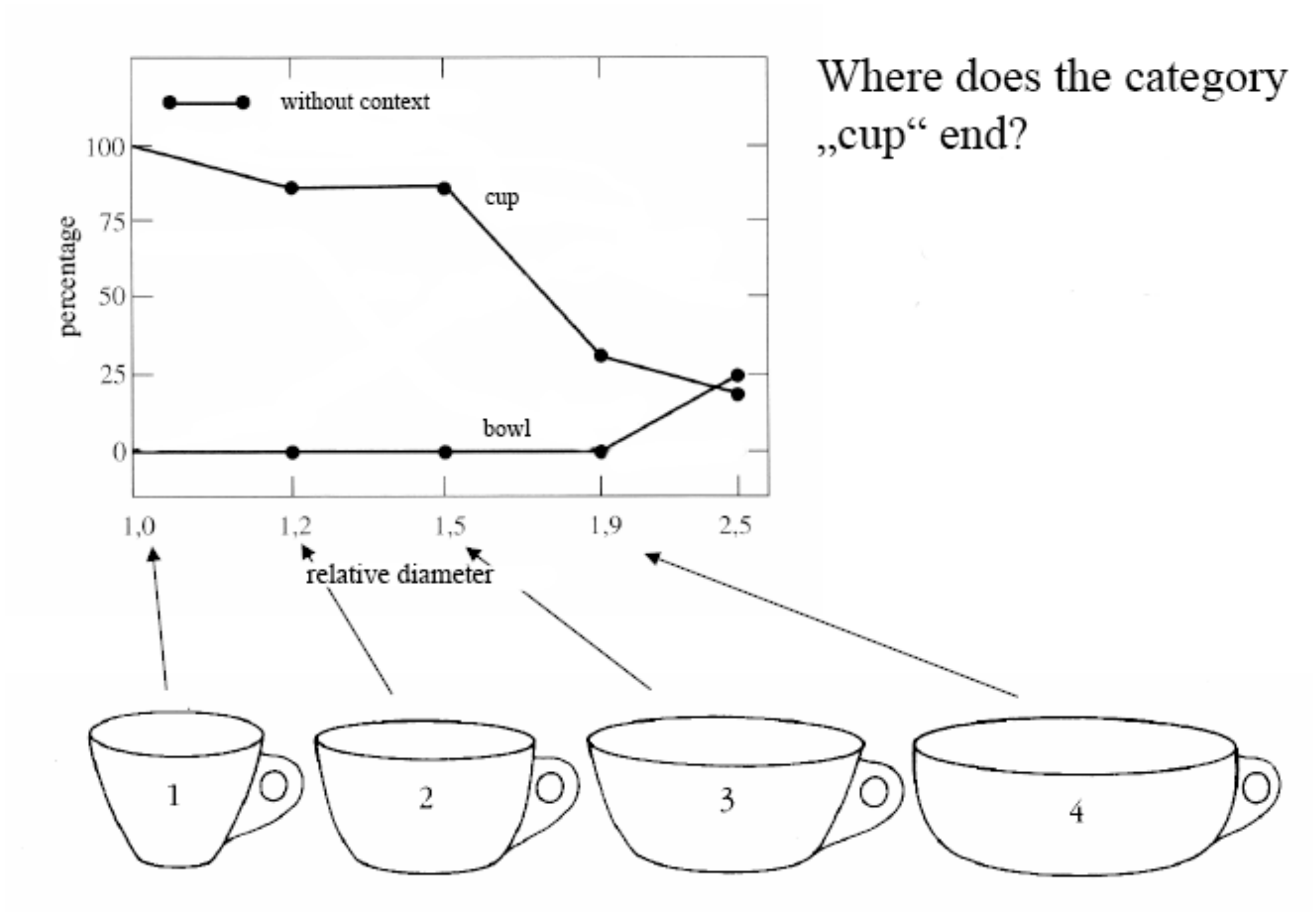
What are these?

Cup or bowl?



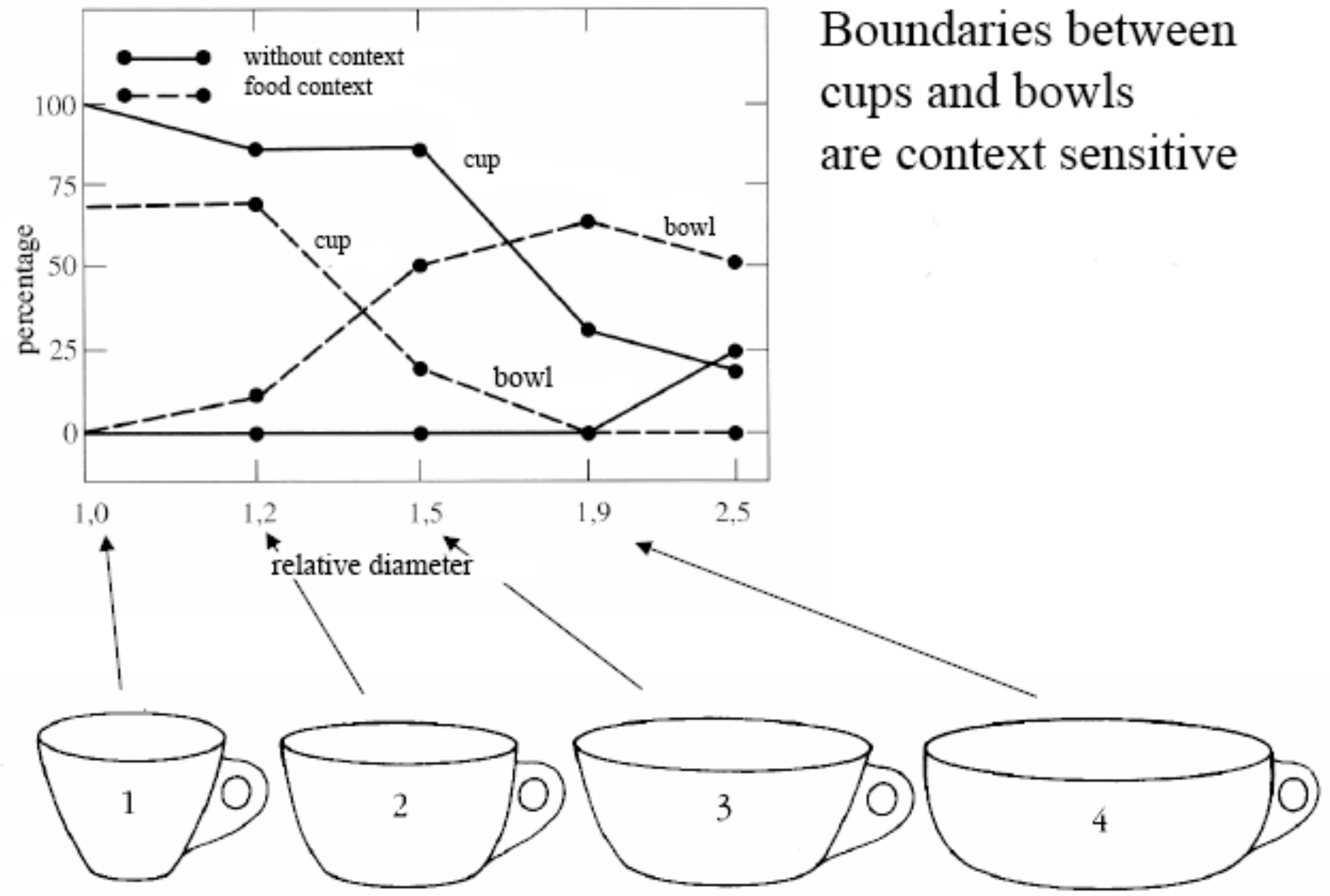


The category depends on complex features of the object (diameter, etc)



Where does the category „cup“ end?

The category depends on the context!
(If there is food in it, it's a bowl)



Boundaries between cups and bowls are context sensitive

Labov’s definition of cup

The term *cup* is used to denote round containers with a ratio of depth to width of $1 \pm r$ where $r \leq r_b$, and $r_b = \alpha_1 + \alpha_2 + \ldots\alpha_\upsilon$ and α_1 is a positive quality when the feature i is present and 0 otherwise.

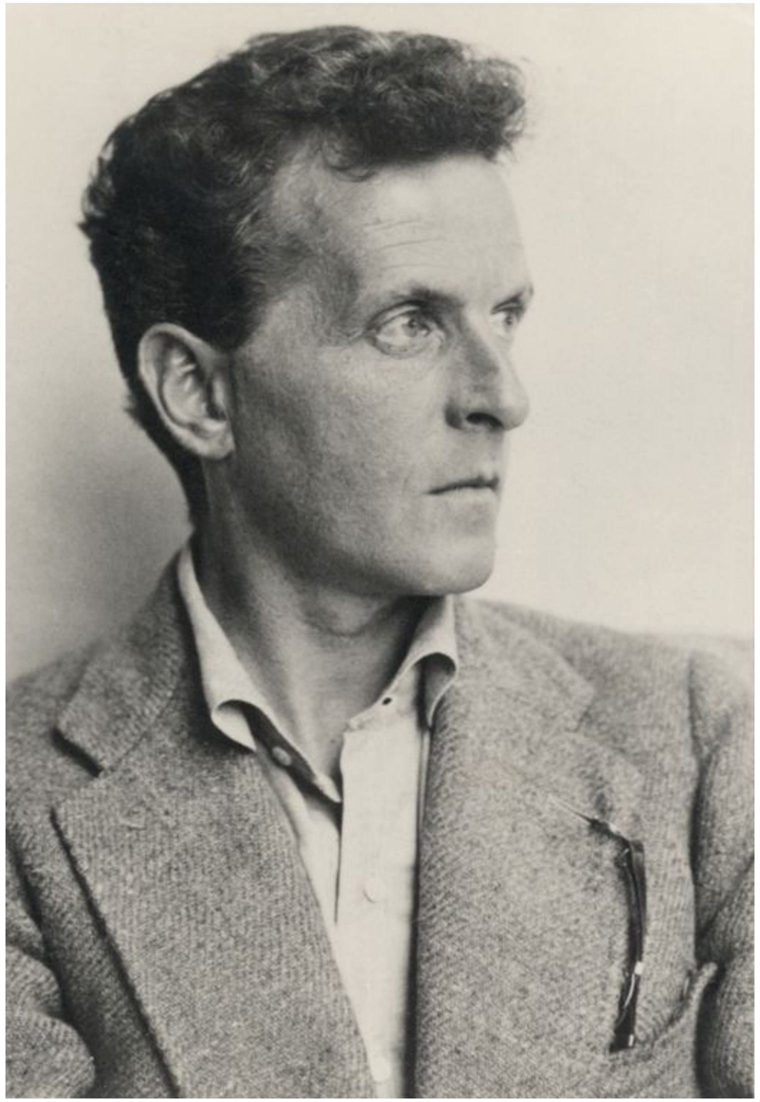
feature 1 = with one handle
2 = made of opaque vitreous material
3 = used for consumption of food
4 = used for the consumption of liquid food
5 = used for consumption of hot liquid food
6 = with a saucer
7 = tapering
8 = circular in cross-section

Cup is used variably to denote such containers with ratios width to depth $1 \pm r$ where $r_b \leq r \leq r_1$ with a probability of r_1 - $r/r_t - r_b$. The quantity $1 \pm r_b$ expresses the distance from the modal value of width to height.

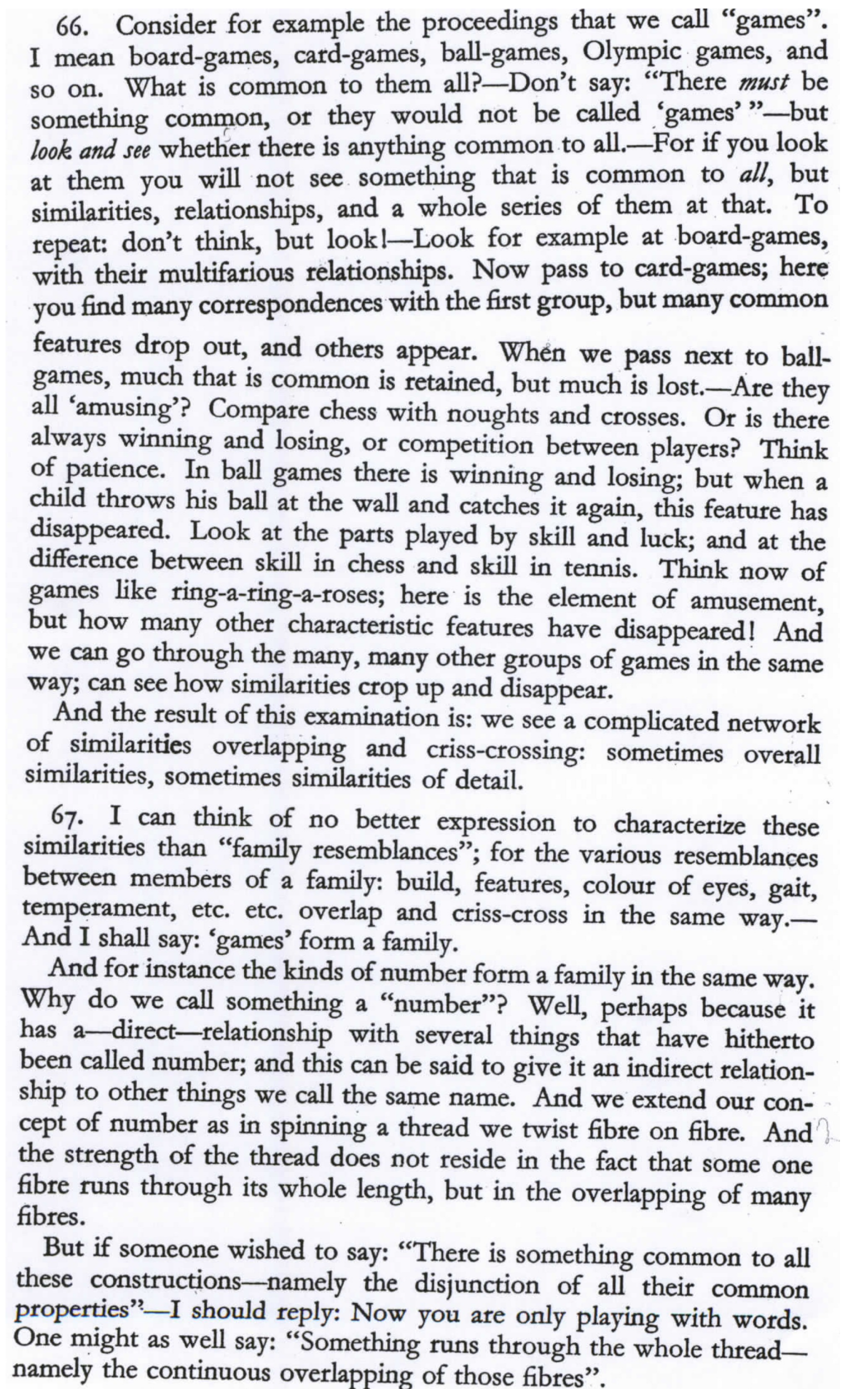
Ludwig Wittgenstein (1889-1951)

Philosopher of language

In his late years, a proponent of studying "ordinary language"



Wittgenstein (1945) *Philosophical Investigations.* Paragraphs 66,67



66. Consider for example the proceedings that we call "games". I mean board-games, card-games, ball-games, Olympic games, and so on. What is common to them all?—Don't say: "There *must* be something common, or they would not be called 'games' "—but *look and see* whether there is anything common to all.—For if you look at them you will not see something that is common to *all*, but similarities, relationships, and a whole series of them at that. To repeat: don't think, but look!—Look for example at board-games, with their multifarious relationships. Now pass to card-games; here you find many correspondences with the first group, but many common features drop out, and others appear. When we pass next to ball-games, much that is common is retained, but much is lost.—Are they all 'amusing'? Compare chess with noughts and crosses. Or is there always winning and losing, or competition between players? Think of patience. In ball games there is winning and losing; but when a child throws his ball at the wall and catches it again, this feature has disappeared. Look at the parts played by skill and luck; and at the difference between skill in chess and skill in tennis. Think now of games like ring-a-ring-a-roses; here is the element of amusement, but how many other characteristic features have disappeared! And we can go through the many, many other groups of games in the same way; can see how similarities crop up and disappear.

And the result of this examination is: we see a complicated network of similarities overlapping and criss-crossing: sometimes overall similarities, sometimes similarities of detail.

67. I can think of no better expression to characterize these similarities than "family resemblances"; for the various resemblances between members of a family: build, features, colour of eyes, gait, temperament, etc. etc. overlap and criss-cross in the same way.—And I shall say: 'games' form a family.

And for instance the kinds of number form a family in the same way. Why do we call something a "number"? Well, perhaps because it has a—direct—relationship with several things that have hitherto been called number; and this can be said to give it an indirect relationship to other things we call the same name. And we extend our concept of number as in spinning a thread we twist fibre on fibre. And the strength of the thread does not reside in the fact that some one fibre runs through its whole length, but in the overlapping of many fibres.

But if someone wished to say: "There is something common to all these constructions—namely the disjunction of all their common properties"—I should reply: Now you are only playing with words. One might as well say: "Something runs through the whole thread—namely the continuous overlapping of those fibres".

What is a game?

Wittgenstein’s thought experiment "What is a game”:

PI #66:

"Don’t say “there must be something common, or they would not be called `games’”—but *look and see* whether there is anything common to all"

Is it amusing?
Is there competition?
Is there long-term strategy?
Is skill required?
Must luck play a role?
Are there cards?
Is there a ball?

Family Resemblance

Game 1	Game 2	Game 3	Game 4
ABC	BCD	ACD	ABD

“each item has at least one, and probably several, elements in common with one or more items, but no, or few, elements are common to all items” [Rosch and Mervis]

Vector Semantics & Embeddings

It's hard to define a concept

Vector Semantics & Embeddings

Vector Semantics

How about a radically different approach?

Ludwig Wittgenstein

PI #43:

"The meaning of a word is its use in the language"

Let's define words by their usages

One way to define "usage":

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 recent English borrowing *ongchoi* mean?

Suppose you see these sentences:

- Ong choi is delicious **sautéed with garlic**.
- Ong choi is superb **over rice**
- Ong choi **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

Ongchoi: *Ipomoea aquatica "Water Spinach"*

空心菜

kangkong

rau muống

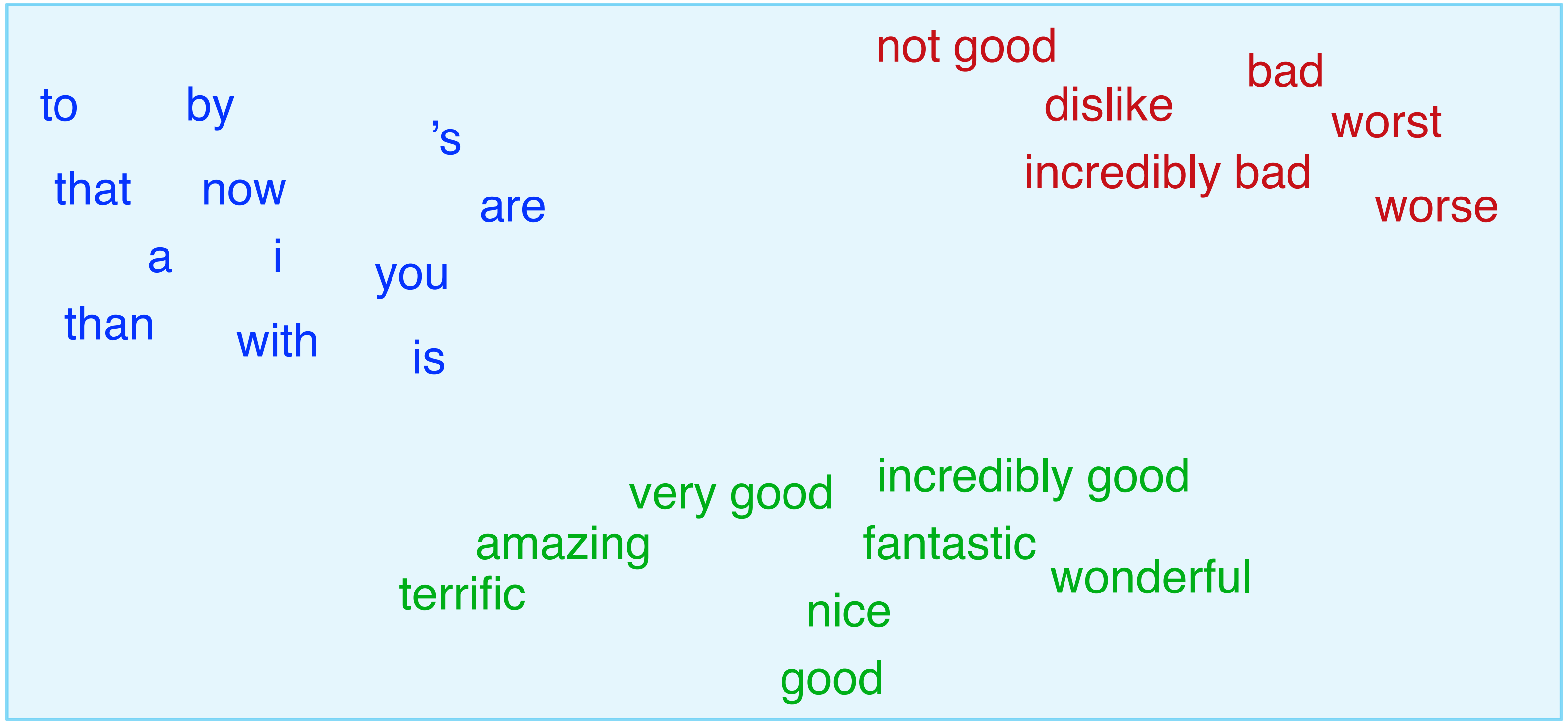
…

Yamaguchi, Wikimedia Commons, public domain

A new model of meaning focusing on distributional 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 vector space

The standard way to represent meaning in NLP

Every modern NLP algorithm uses embeddings as the representation of word meaning

Fine-grained model of meaning for similarity

Intuition: why vectors?

Consider sentiment analysis:

- With **words**, a feature is a word identity
  - Feature 5: 'The previous word was "terrible"'
  - requires **exact same word** to be in training and test
- With **embeddings**:
  - Feature is a word vector
  - 'The previous word was vector [35,22,17…]
  - Now in the test set we might see a similar vector [34,21,14]
  - We can generalize to **similar but unseen** words!!!

We'll discuss 2 kinds of embeddings

tf-idf

- Information Retrieval workhorse!
- 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 **predict** whether a word is likely to appear nearby
- In later chapters we'll discuss extensions called **contextual embeddings**

Vector Semantics & Embeddings

Vector Semantics

Vector Semantics & Embeddings

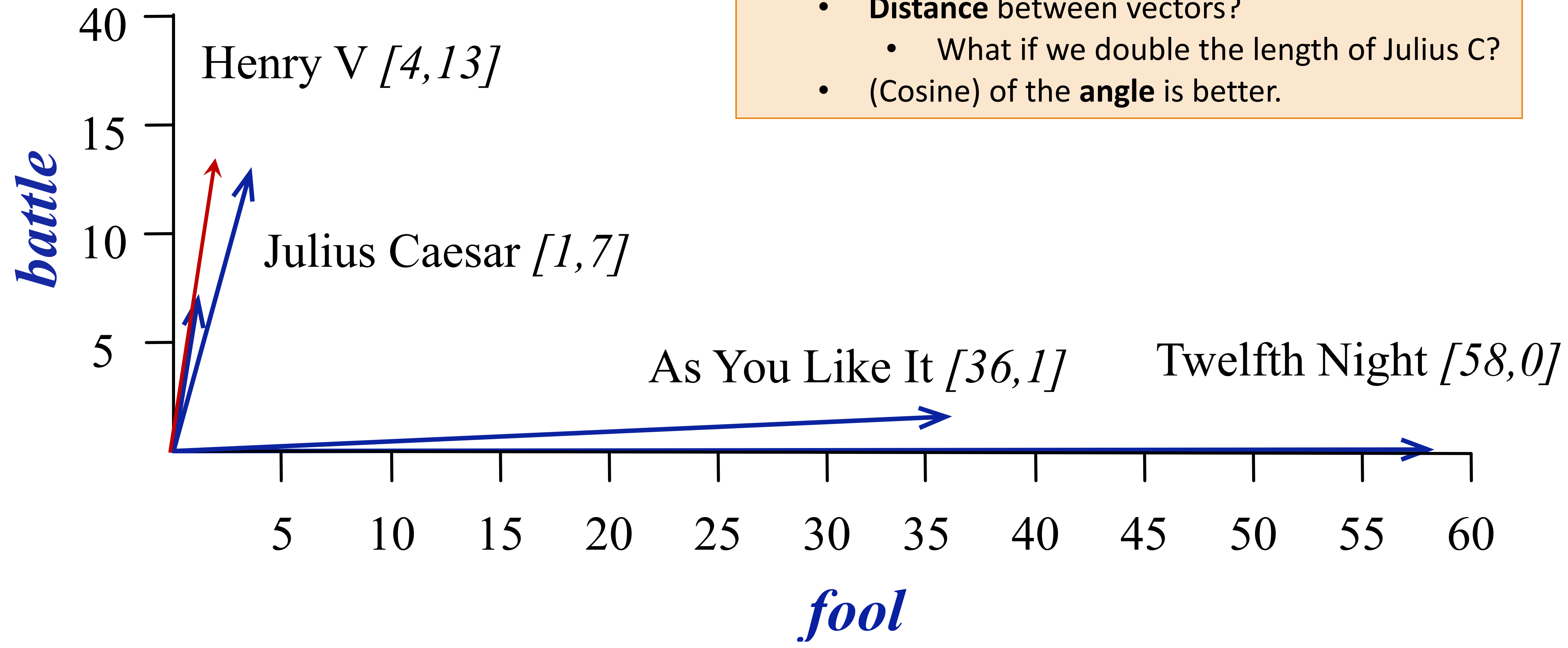
Words and Vectors

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



1. How to measure distance or similarity?
    - **Distance** between vectors?
        - What if we double the length of Julius C?
    - (Cosine) of the **angle** is better.

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*.

Idea for word meaning: 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"

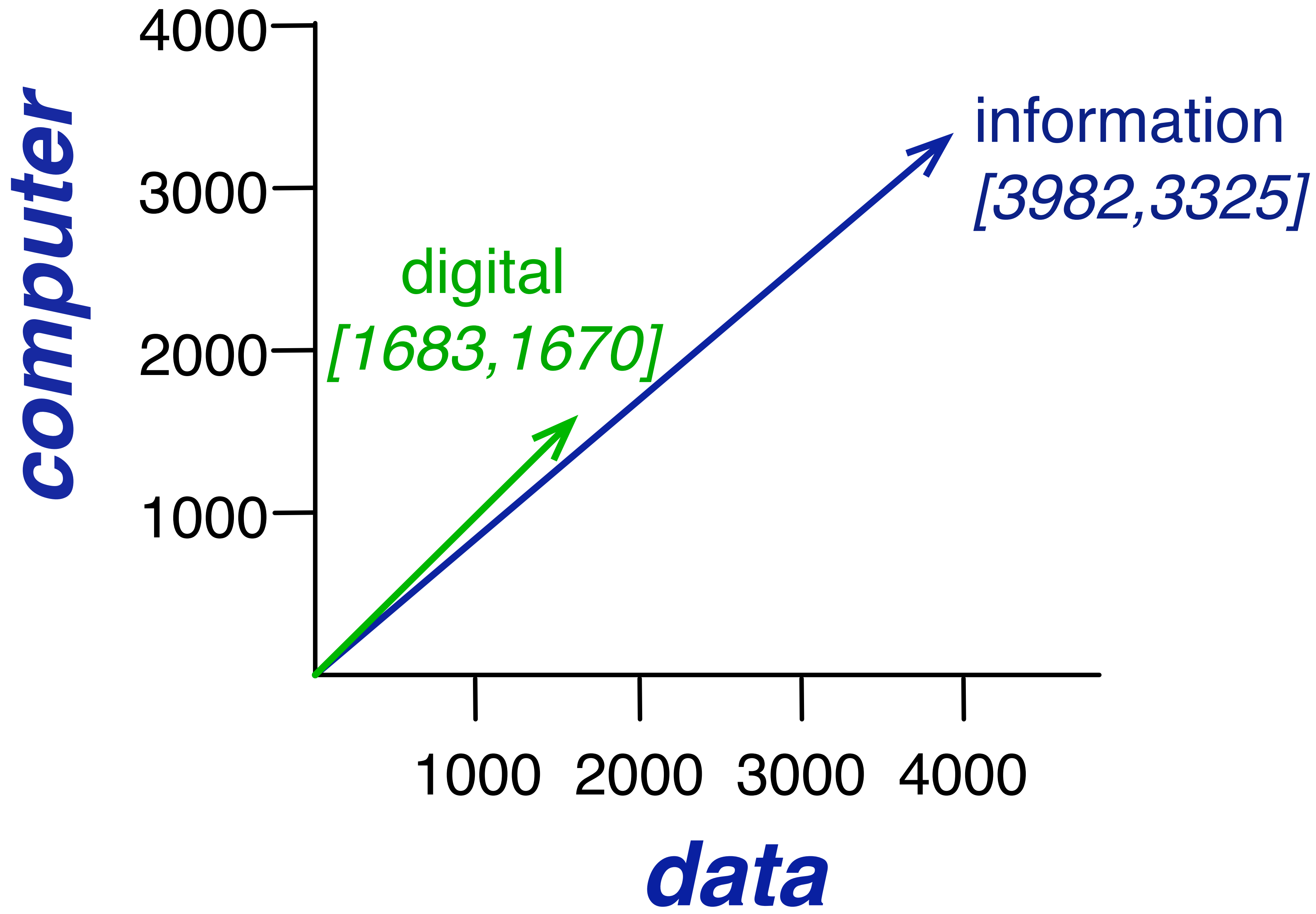
fool is "the kind of word that occurs in comedies, especially Twelfth Night"

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

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

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	**computer**	**data**	**result**	**pie**	**sugar**	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...


computer
4000
3000
2000
1000
information
[3982,3325]
digital
[1683,1670]
1000
2000
3000
4000
data

Vector Semantics & Embeddings

Words and Vectors

Vector Semantics & Embeddings

Cosine for computing word similarity

Dot product and cosine

The dot product between two vectors is a scalar:

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

The dot product tends to be high when the two vectors have large values in the same dimensions

Dot product can be a similarity metric between vectors

Problem with raw dot-product

Dot product favors long vectors

Dot product is higher if a vector is longer (has higher values in many dimension)

Vector length:

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

Frequent words (of, the, you) have long vectors (since they occur many times with other words).

So dot product overly favors frequent words

Alternative: cosine for computing word similarity

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

Cosine examples

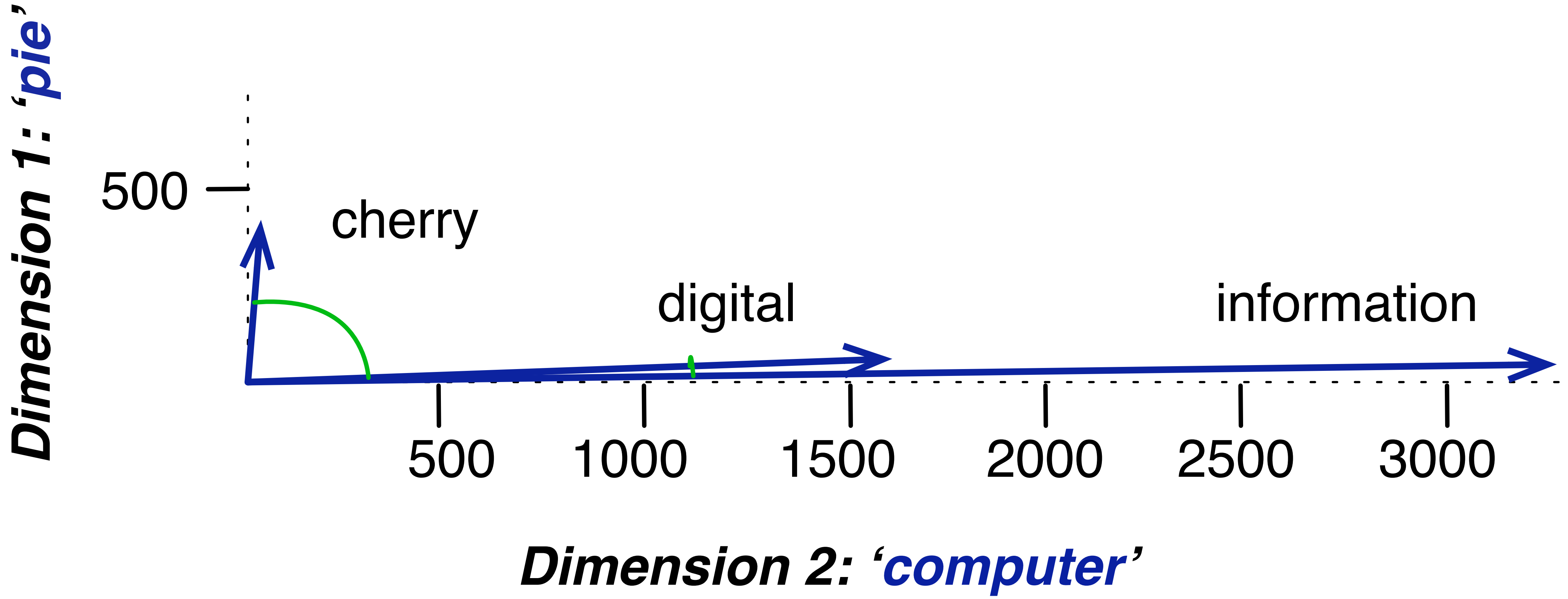
$$\cos(\vec{v},\vec{w}) = \frac{\vec{v} \bullet \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|} \bullet \frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325

$$\cos(\text{cherry}, \text{information}) = \frac{442 * 5 + 8 * 3982 + 2 * 3325}{\sqrt{442^2 + 8^2 + 2^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .017$$

$$\cos(\text{digital}, \text{information}) = \frac{5 * 5 + 1683 * 3982 + 1670 * 3325}{\sqrt{5^2 + 1683^2 + 1670^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .996$$

Visualizing cosines (well, angles)



Vector Semantics & Embeddings

Cosine for computing word similarity

Vector Semantics & Embeddings

TF-IDF

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!

Two common solutions for word weighting

tf-idf: tf-idf value for word t in document d:

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

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

PMI: (Pointwise mutual information)

- $\text{PMI}(w_1, w_2) = log \frac{p(w_1,w_2)}{p(w_1)p(w_2)}$

See if words like "good" appear more often with "great" than we would expect by chance

Term frequency (tf)

$\mathrm{tf}_{t,d} = \mathrm{count}(t,d)$

Instead of using raw count, we can squash a bit:

$\mathrm{tf}_{t,d} = \log_{10}(\mathrm{count}(t,d)+1)$

Document frequency (df)

df$_t$ *is* the number of documents *t* occurs in.

- Note this is **not** collection frequency, i.e. total count across all documents.

"*Romeo*" is very distinctive for one Shakespeare play:

	Collection Frequency	Document Frequency
Romeo	113	1
action	113	31

Important: documents can be **anything**; we can call each paragraph a document

Inverse document frequency (idf)

$$\text{idf}_t = \log_{10}\left(\frac{N}{\text{df}_t}\right)$$

N is the total number of documents in the collection (37 docs)

Word	**df**	**idf**
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

Final tf-idf weighted value for a word

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

Raw counts:

	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

Tf-idf:

	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

Vector Semantics & Embeddings

TF-IDF

Vector Semantics & Embeddings

PPMI

Pointwise Mutual Information

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

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

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

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

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
    - Imagine w_1 and w_2 whose probability is each 10^{-6}
    - Hard to be sure $p(w_1,w_2)$ is significantly different than 10^{-12}
  - Plus it's not clear people are good at "unrelatedness"
- 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)

$\mathrm{f_{ij}}$ is # of times $\mathrm{w_i}$ occurs in context (of) $\mathrm{c_j}$

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

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

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

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

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	$\boxed{3982}$	378	5	13	$\boxed{7703}$
count(context)	4997	$\boxed{5673}$	473	512	61	$\boxed{11716}$

p(*w* = information, *c* = data) = 3982/11716 = .3399

p(*w* = information) = 7703/11716 = .6575

p(*c* = data) = 5673/11716 = .4842

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

	p(w,context)					p(w)
	computer	**data**	**result**	**pie**	**sugar**	**p(w)**
cherry	0.0002	0.0007	0.0008	0.0377	0.0021	0.0415
strawberry	0.0000	0.0000	0.0001	0.0051	0.0016	0.0068
digital	0.1425	0.1436	0.0073	0.0004	0.0003	0.2942
information	0.2838	$\boxed{0.3399}$	0.0323	0.0004	0.0011	$\boxed{0.6575}$
p(context)	0.4265	$\boxed{0.4842}$	0.0404	0.0437	0.0052	

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

	p(w,context)					p(w)
	computer	data	result	pie	sugar	p(w)
cherry	0.0002	0.0007	0.0008	0.0377	0.0021	0.0415
strawberry	0.0000	0.0000	0.0001	0.0051	0.0016	0.0068
digital	0.1425	0.1436	0.0073	0.0004	0.0003	0.2942
information	0.2838	0.3399	0.0323	0.0004	0.0011	0.6575
p(context)	0.4265	0.4842	0.0404	0.0437	0.0052	

pmi(information, data) = $\log_2$ (.3399 / (.6575*.4842)) = .0944

Resulting PPMI matrix (negatives replaced by 0)

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Weighting PMI

PMI is biased toward infrequent events

- Very rare words have very high PMI values

Two solutions:

- Give rare context words slightly higher probabilities
- Use add-one smoothing (which has a similar effect)

Weighting PMI: Giving rare context words slightly higher probability

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

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

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

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

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

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

Vector Semantics & Embeddings

Dense vectors

Sparse versus dense vectors

tf-idf vectors are

- **long** (length |V|= 20,000 to 50,000)
- **sparse** (most elements are zero)

Alternative: learn vectors which are

- **short** (length 50-1000)
- **dense** (most elements are non-zero)

Sparse versus dense vectors

Why dense vectors?

- Short vectors may be easier to use as **features** in machine learning (fewer weights to tune)
- Dense vectors may **generalize** better than explicit counts
- They may do better at capturing synonymy:
  - *car* and *automobile* are synonyms; but are distinct dimensions
    - a word with *car* as a neighbor and a word with *automobile* as a neighbor should be similar, but aren't
- **In practice, they work better**

Common methods for getting short dense vectors

"Neural Language Model"-inspired models

- Word2vec (skipgram, CBOW), Glove

Singular Value Decomposition (SVD)

- A special case of this is called LSA – Latent Semantic Analysis

Alternative to these "static embeddings":

- Contextual Embeddings (ELMo, BERT)
- Compute distinct embeddings for a word in its context
- Separate embeddings for each token of a word
- We'll return to this in a later chapter

Vector Semantics & Embeddings

Dense vectors

Vector Semantics & Embeddings

Word2vec: The classifier

Embeddings you can download!

Word2vec (Mikolov et al)

https://code.google.com/archive/p/word2vec/

Glove (Pennington, Socher, Manning)

http://nlp.stanford.edu/projects/glove/

Word2vec

Popular embedding method

Very fast to train

Code available on the web

Idea: **predict** rather than **count**

Word2vec

Instead of **counting** how often each word *w* occurs near "*apricot*"

- Train a classifier on a binary **prediction** task:
  - Is *w* likely to show up near "*apricot*"?

We don’t actually care about this task

- But we'll take the learned **classifier weights** as the **word embeddings**

Big idea is **self-supervision**:

- A word c that occurs near apricot in the corpus asks as the gold "correct answer" for supervised learning
- No need for human labels
- Bengio et al. (2003); Collobert et al. (2011)

Word2Vec: Skip-Gram Task

Word2vec provides a variety of options. We'll do:

Skip-gram with negative sampling (SGNS)

Approach: predict if candidate word *c* is a "neighbor"

1. Treat the target word *t* and a neighboring context word *c* as **positive examples**.
2. Randomly sample other words in the lexicon to get **negative examples**
3. Use **logistic regression** to train a classifier to distinguish those two cases
4. Use the learned weights as the embeddings

Skip-Gram Training Data

Assume a +/- 2 word window, given training sentence:

…lemon, a [tablespoon of apricot jam, a] pinch…

c1 c2 [target] c3 c4

Skip-Gram Classifier

(assuming a +/- 2 word window)

…lemon, a [tablespoon of apricot jam,   a] pinch…

c1 c2 [target] c3 c4

Goal: train a classifier that is given a candidate (**w**ord, **c**ontext) pair

(apricot, tablespoon) --> *positive*

(apricot, aardvark) --> *negative*

…

And assigns each pair a probability:

$P(+|w, c)$

Similarity is computed from dot product

Remember: two vectors are similar if they have a high dot product

- Cosine is just a normalized dot product

So:

- Similarity(w,c) $\propto w \cdot c$

We’ll need to normalize to get a probability

- (cosine isn't a probability either)

Turning dot products into probabilities

$\mathrm{Sim}(\mathrm{w,c}) \approx \mathrm{w} \cdot c$

To turn this into a probability

We'll use the sigmoid from logistic regression:

$$P(+|w,c) = \sigma(c \cdot w) = \frac{1}{1+\exp(-c \cdot w)}$$

$$\begin{aligned} P(-|w,c) &= 1 - P(+|w,c) \\ &= \sigma(-c \cdot w) = \frac{1}{1+\exp(c \cdot w)} \end{aligned}$$

How Skip-Gram Classifier computes $P(+|w, c)$

$$P(+|w,c) \;=\; \sigma(c \cdot w) = \frac{1}{1+\exp(-c \cdot w)}$$

This is for one context word, but we have lots of context words.
We'll assume independence and just multiply them:

$$P(+|w,c_{1:L}) \;=\; \prod_{i=1}^{L} \sigma(c_i \cdot w)$$

$$\log P(+|w,c_{1:L}) \;=\; \sum_{i=1}^{L} \log \sigma(c_i \cdot w)$$

Skip-gram classifier: summary

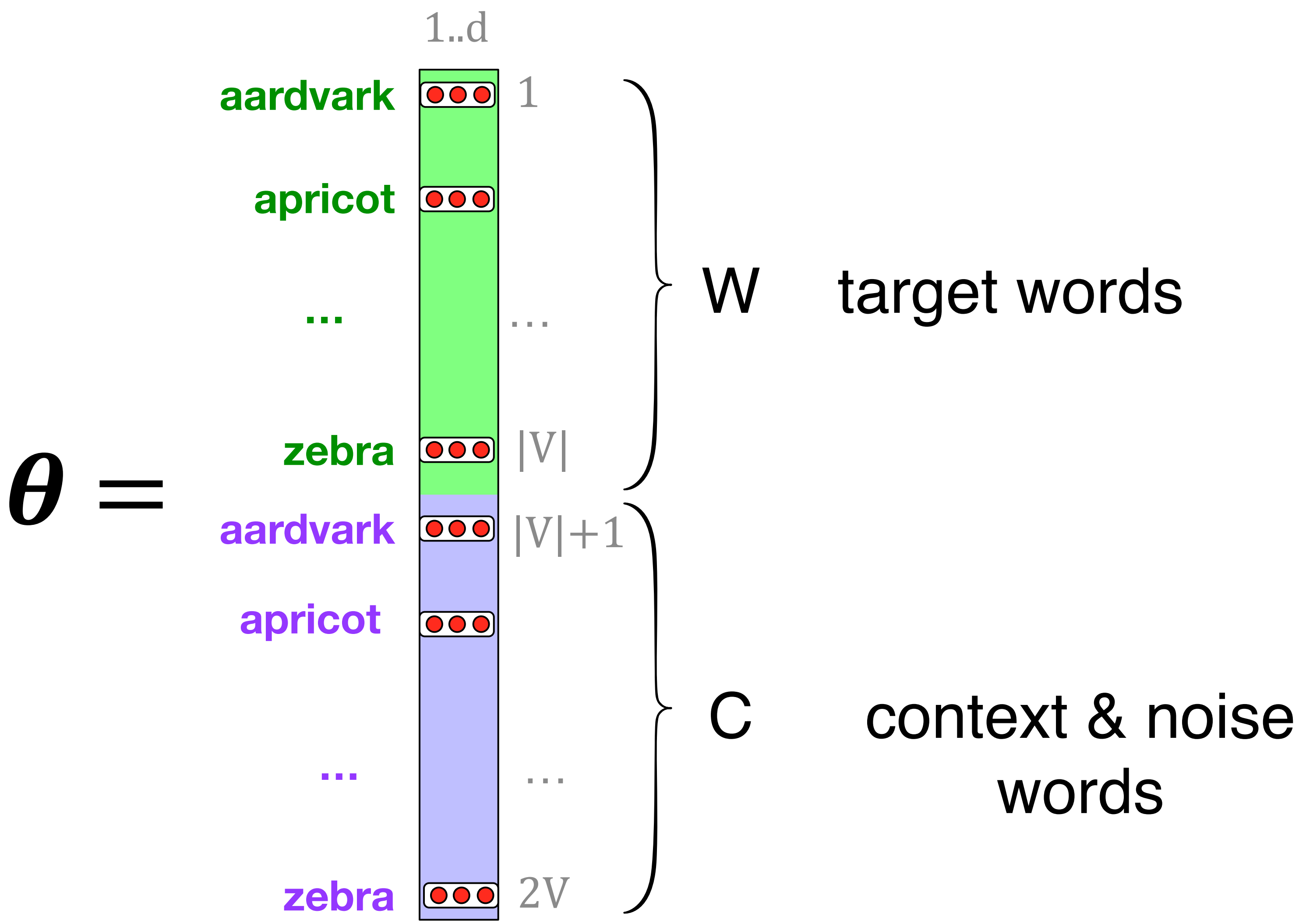
A probabilistic classifier that,

- given a test target word *w*
- its context window of *L* words $c_{1:L}$,

assigns a probability that *w* occurs in this window.

To compute this probability, we just need **classifier weights** = **embeddings** for all the words.

These embeddings we'll need: a set for w, a set for c



Vector Semantics & Embeddings

Word2vec: The classifier

Vector Semantics & Embeddings

Word2vec: Learning the embeddings

Skip-Gram Training data

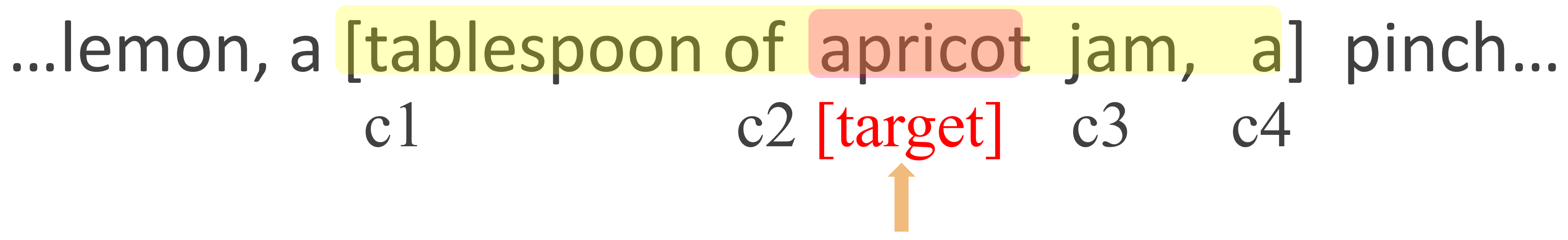
…lemon, a [tablespoon of apricot jam, a] pinch…

c1 c2 [target] c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

Skip-Gram Training data



positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

For each positive example we'll grab k negative examples, sampling by frequency

Skip-Gram Training data

…lemon, a [tablespoon of apricot jam, a] pinch…

c1 c2 [target] c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

Word2vec: how to learn vectors

Given the set of positive and negative training instances, and an initial set of embedding vectors

The goal of learning is to adjust those word vectors such that we:

- **Maximize** the similarity of the target word, context word pairs (w , c_{pos}) drawn from the positive data
- **Minimize** the similarity of the (w , c_{neg}) pairs drawn from the negative data.

Loss function for one *w with* c_{pos}, $c_{neg1} \ldots c_{negk}$

Maximize the dot product of the word with the actual context words, and minimize the dot products of the word with the *k* negative sampled non-neighbor words.

$$
\begin{aligned}
L_{\text{CE}} &= -\log\left[P(+|w,c_{pos})\prod_{i=1}^{k}P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log\left(1-P(+|w,c_{neg_i})\right)\right] \\
&= -\left[\log\sigma(c_{pos}\cdot w)+\sum_{i=1}^{k}\log\sigma(-c_{neg_i}\cdot w)\right]
\end{aligned}
$$

Learning the classifier

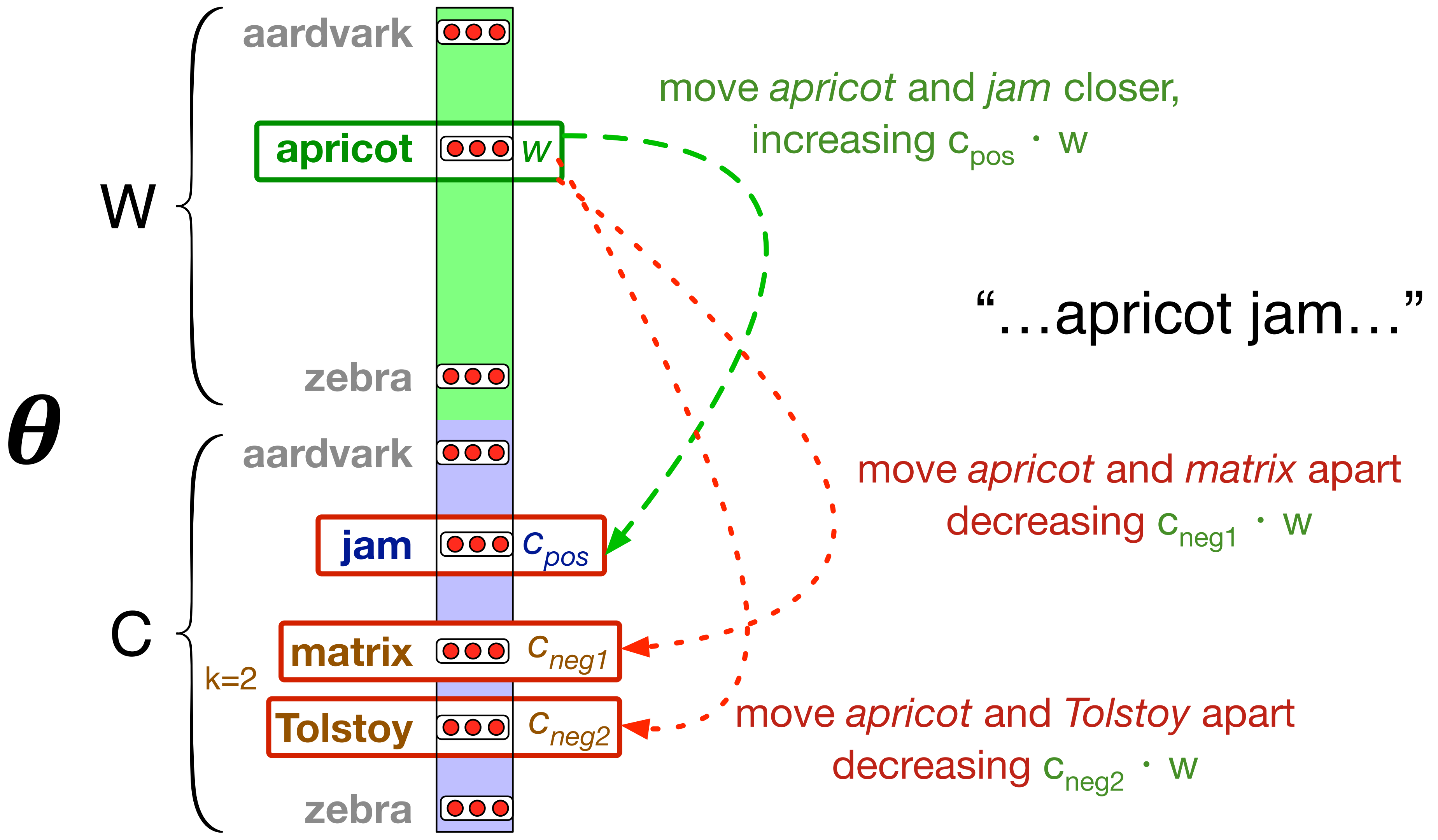
How to learn?

- Stochastic gradient descent!

We'll adjust the word weights to

- make the positive pairs more likely
- and the negative pairs less likely,
- over the entire training set.

Intuition of one step of gradient descent



aardvark
apricot
w
W
zebra
θ
aardvark
jam
c_{pos}
C
matrix
c_{neg1}
k=2
Tolstoy
c_{neg2}
zebra
move *apricot* and *jam* closer, increasing $c_{pos} \cdot w$
"…apricot jam…"
move *apricot* and *matrix* apart decreasing $c_{neg1} \cdot w$
move *apricot* and *Tolstoy* apart decreasing $c_{neg2} \cdot w$

The derivatives of the loss function

$$L_{CE} = -\left[\log \sigma(c_{pos} \cdot w) + \sum_{i=1}^{k} \log \sigma(-c_{neg_i} \cdot w)\right]$$

$$\frac{\partial L_{CE}}{\partial c_{pos}} = [\sigma(c_{pos} \cdot w) - 1]w$$

$$\frac{\partial L_{CE}}{\partial c_{neg}} = [\sigma(c_{neg} \cdot w)]w$$

$$\frac{\partial L_{CE}}{\partial w} = [\sigma(c_{pos} \cdot w) - 1]c_{pos} + \sum_{i=1}^{k} [\sigma(c_{\mathrm{neg}_i} \cdot w)]c_{neg_i}$$

Update equation in SGD

Start with randomly initiatlzed C and W matrices, then incrementally do updates

$$\begin{aligned} c_{pos}^{t+1} &= c_{pos}^{t} - \eta[\sigma(c_{pos}^{t} \cdot w) - 1]w \\ c_{neg}^{t+1} &= c_{neg}^{t} - \eta[\sigma(c_{neg}^{t} \cdot w)]w \end{aligned}$$

$$w^{t+1} = w^{t} - \eta[\sigma(c_{pos} \cdot w^{t}) - 1]c_{pos} + \sum_{i=1}^{k}[\sigma(c_{\mathrm{neg}_i} \cdot w^{t})]c_{neg_i}$$

Two sets of embeddings

SGNS learns two sets of embeddings

Target embeddings matrix W

Context embedding matrix C

It's common to just add them together, representing word *i* as the vector $w_i + c_i$

Summary: How to learn word2vec (skip-gram) embeddings

Start with V random *d*-dimentional vectors as initial embeddings

Train a classifier based on embedding similarity

- Take a corpus and take pairs of words that co-occur as positive examples
- Take pairs of words that don't co-occur as negative examples
- Train the classifier to distinguish these by slowly adjusting all the embeddings to improve the classifier performance
- Throw away the classifier code and keep the embeddings.

Vector Semantics & Embeddings

Word2vec: Learning the embeddings

Vector Semantics & Embeddings

Properties of Embeddings

The kinds of neighbors depend on window size

Large windows (C= +/- 5) : nearest words are related words in same **semantic field**

- *Hogwarts* nearest neighbors are Harry Potter world:
  - *Dumbledore, Half-blood, Malfoy*

Small windows (C= +/- 2) : nearest words are similar nouns, words in same **taxonomy**

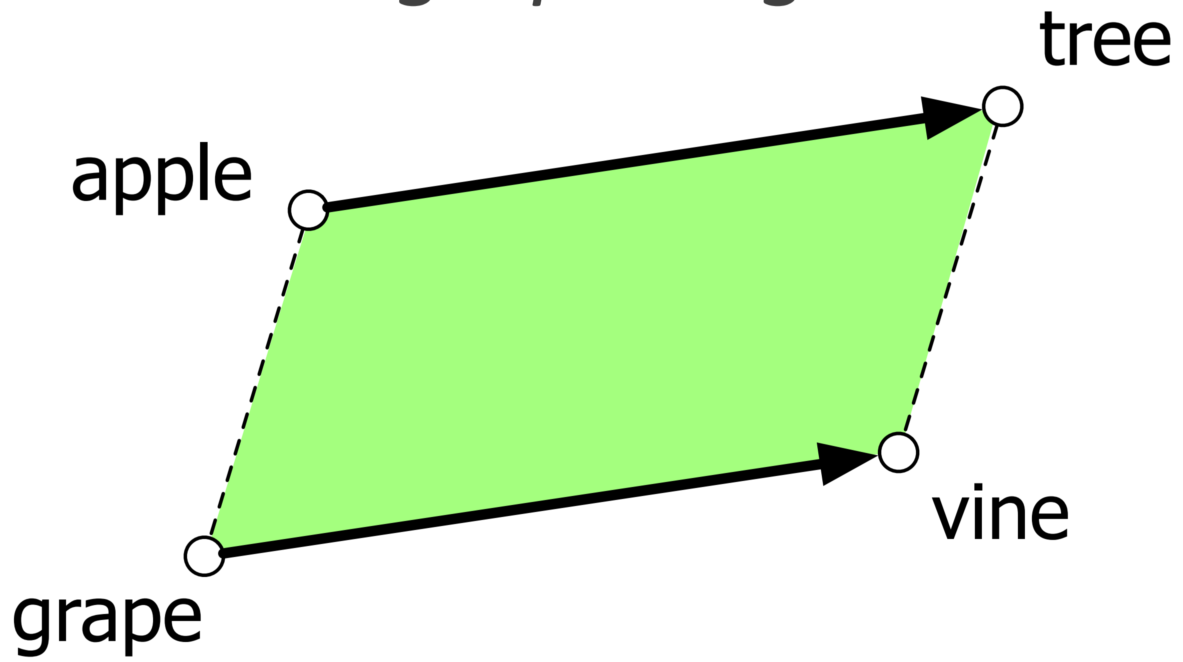
- *Hogwarts* nearest neighbors are other fictional schools
  - *Sunnydale, Evernight, Blandings*

Analogical relations

The classic parallelogram model of analogical reasoning (Rumelhart and Abrahamson 1973)

To solve: *"apple is to tree as grape is to ______"*

Add $\overrightarrow{apple} - \overrightarrow{tree}$ *to* $\overrightarrow{grape}$ *to get* $\overrightarrow{vine}$



Analogical relations via parallelogram

The parallelogram method can solve analogies with both sparse and dense embeddings (Turney and Littman 2005, Mikolov et al. 2013b)

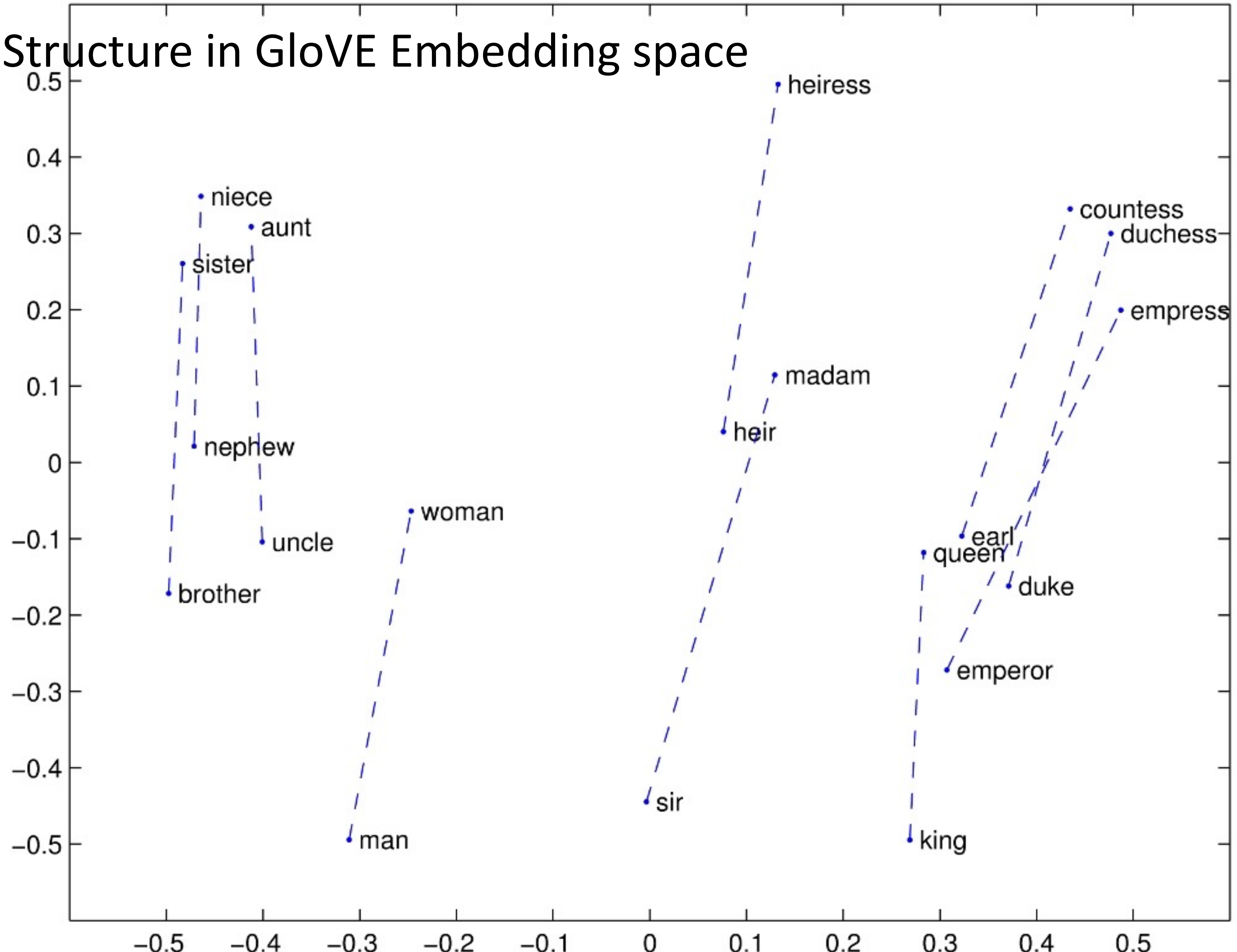
$\overrightarrow{\textit{king}} - \overrightarrow{\textit{man}} + \overrightarrow{\textit{woman}}$ is close to $\overrightarrow{\textit{queen}}$

$\overrightarrow{\textit{Paris}} - \overrightarrow{\textit{France}} + \overrightarrow{\textit{Italy}}$ is close to $\overrightarrow{\textit{Rome}}$

For a problem a:a*::b:b*, the parallelogram method is:

$$\hat{b}^* = \underset{x}{\operatorname{argmin}}\ \operatorname{distance}(x, a^* - a + b)$$

Structure in GloVE Embedding space



Caveats with the parallelogram method

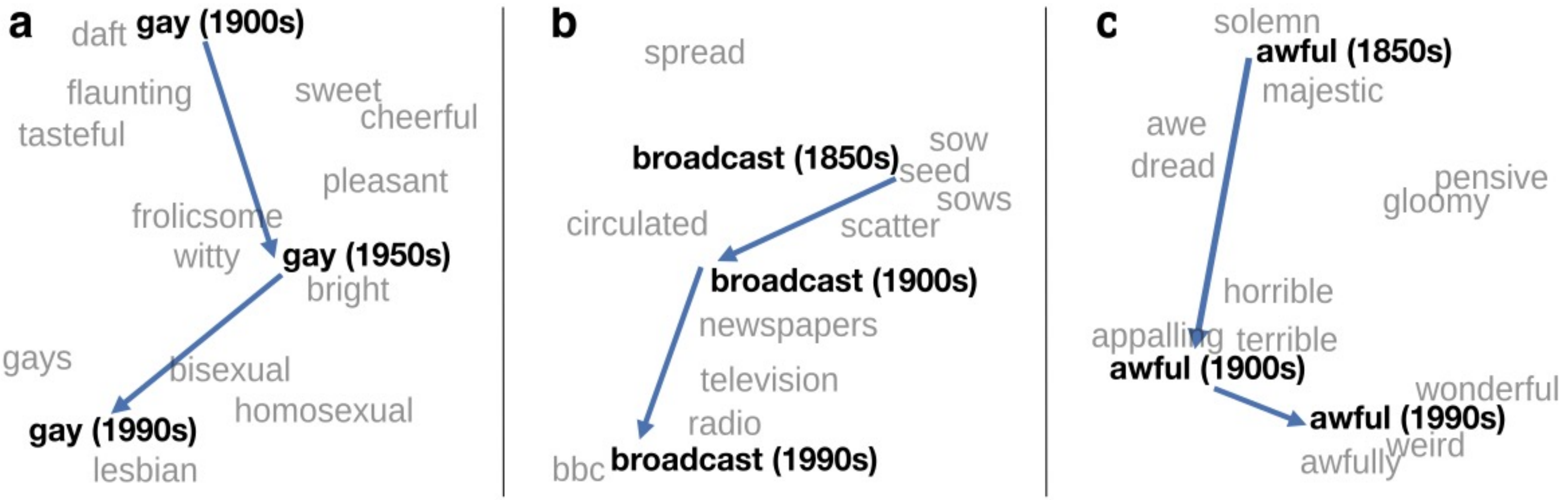
It only seems to work for frequent words, small distances and certain relations (relating countries to capitals, or parts of speech), but not others. (Linzen 2016, Gladkova et al. 2016, Ethayarajh et al. 2019a)

Understanding analogy is an open area of research (Peterson et al. 2020)

Embeddings as a window onto historical semantics

Train embeddings on different decades of historical text to see meanings shift

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



William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. Proceedings of ACL.

Embeddings reflect cultural bias!

Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." In *NeurIPS*, pp. 4349-4357. 2016.

Ask "Paris : France :: Tokyo : x"

- x = Japan

Ask "father : doctor :: mother : x"

- x = nurse

Ask "man : computer programmer :: woman : x"

- x = homemaker

Algorithms that use embeddings as part of e.g., hiring searches for programmers, might lead to bias in hiring

Historical embedding as a tool to study cultural biases

Garg, N., Schiebinger, L., Jurafsky, D., and Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. Proceedings of the National Academy of Sciences 115(16), E3635–E3644.

- Compute a **gender or ethnic bias** for each adjective: e.g., how much closer the adjective is to "woman" synonyms than "man" synonyms, or names of particular ethnicities
  - Embeddings for **competence** adjective (*smart, wise, brilliant, resourceful, thoughtful, logical)* are biased toward men, a bias slowly decreasing 1960-1990
  - Embeddings for **dehumanizing** adjectives (barbaric, monstrous, bizarre) were biased toward Asians in the 1930s, bias decreasing over the 20th century.
- These match the results of old surveys done in the 1930s.

Vector Semantics & Embeddings

Supplementary Readings

1. Chapter 6 in J&M.
2. Chapter 14 in Eisenstein.