Natural Language Processing

CSCI 5832—Lecture 10

Jim Martin

Today

- Next assignment
- Word meanings and how we might capture them in representations that are useful computationally
  - Chapter 6: Vector semantics
    - Dimensions of word meaning
    - Word embeddings

Question

- What role did the meaning of words play in our discussion of text classification with either naïve Bayes or logistic regression?

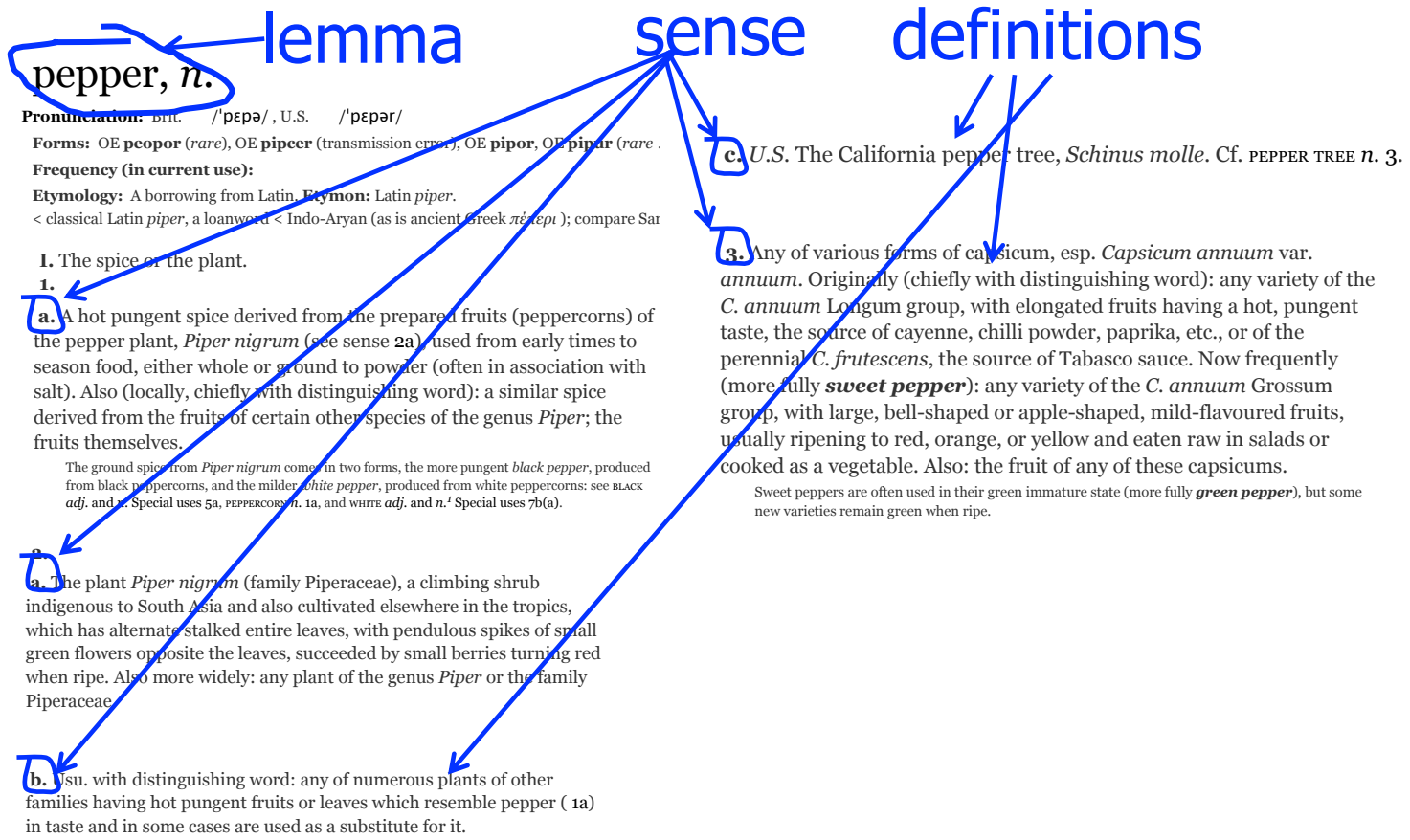
Question

- What role did the meaning of the words play in our discussion of text classification with either naïve Bayes or logistic regression?
  - For NB, counting the co-occurrence of words in a particular class gets at some part of meaning
  - For LR, whatever meaning there is is captured via the features we use (like lists of positive and negative words)
- Seems like we should be able to do more than this.

Question

- Look in a dictionary?

Words, Lemmas, Senses, Definitions



Senses

- *Sense* refers to a single coherent meaning conventionally associated with a word
  - As opposed to its spelling, pronunciation, lexical category, origins, etc.
  - Words can have many different senses
    - Some related to each other, some not

Bank (From Wordnet)

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)
- S: (n) savings bank, coin bank, money box, **bank** (a container (usually with a slot in the top) for keeping money at home) *"the coin bank was empty"*
- S: (n) **bank**, bank building (a building in which the business of banking transacted) *"the bank is on the corner of Nassau and Witherspoon"*
- S: (n) **bank** (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)) *"the plane went into a steep bank"*

Word Relations

- It is useful to consider how senses of different words are related to one another

Relation: Synonymy

- Word (senses) that have the same meaning in some or all contexts.
  - `filbert / hazelnut`
  - `couch / sofa`
  - `big / large`
  - `automobile / car`
  - `vomit / throw up`
  - `Water / H2O`

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.
- The Linguistic Principle of Contrast:
  - Difference in form -> Difference in meaning

Relation: Antonymy

- Senses that are opposites with respect to 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 either:
  - Define a binary opposition
    - `long/short, fast/slow`
  - Be *reversives*:
    - `rise/fall, up/down, enter/exit`

Relation: Word Relatedness

- Also called "word association"
- Word senses may be related in many ways
  - Same semantic field
    - Buy, sell, product, price, cost, merchant, goods, etc.
  - Part-whole relations
    - Wheel, fender, roof, axle, tire, engine
  - Functional relationship
    - Car/gasoline
  - Taxonomically
    - Coyote, dog, wolf, etc.

Semantic Frames

- Words that
  - Cover a particular semantic domain
  - Have structured relations with each other

hospital

surgeon, scalpel, nurse, anesthetic, hospital

restaurant

waiter, menu, plate, food, menu, chef

house

door, roof, kitchen, family, bed

Taxonomic Relations: 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 subordinate of *mango*

Connotation

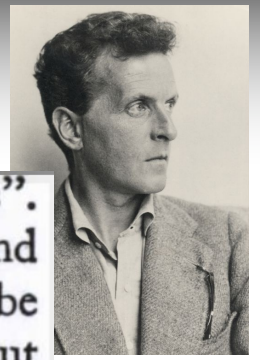
- Words can have *affective* meanings: aspects of meaning related to emotional state or attitude.
  - Positive/negative emotional state: `happy / sad`
  - Positive/negative evaluation: `great / terrible`
  - Positive/negative frame: `thrifty / stingy`
- Like word association, the meanings are very close even if the connotations are different

Summary

- Words can have multiple senses. And those senses can be distinct and arbitrary or related
- Word senses can be related to the senses of other words along many different dimensions
  - Synonymy
  - Antonymy
  - Similarity
  - Relatedness
  - Taxonomic
  - Connotation

But What is a Sense?

Ludwig Wittgenstein (1889-1951)



What is a g
definition
- In the g

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

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?

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?

Distributional Hypothesis

- Words are defined by their environments (the words that occur around them)
- Zellig Harris (1954): If A and B have almost identical environments, we say that they are synonyms.
- Firth (1957): "a word is characterized by the company it keeps"

What does *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

Meaning Based on Distributional Similarity

- Distributional similarity
  - Characterize a words environment, or the company it keeps, by its neighborhood in a high dimensional space.
  - Words nearby are related, words far away are not.
  - The nature of the relationship is captured by the direction

Words as Vectors

- Called "embeddings" because they are embedded in a space
- The dominant way to represent word meanings 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
- Allows for simple methods of meaning composition
  - “Big dog” == Vector(big) + Vector(dog)
  - “Tokyo” == Vector(Japan) + Vector(capital)

2 Kinds of Embeddings

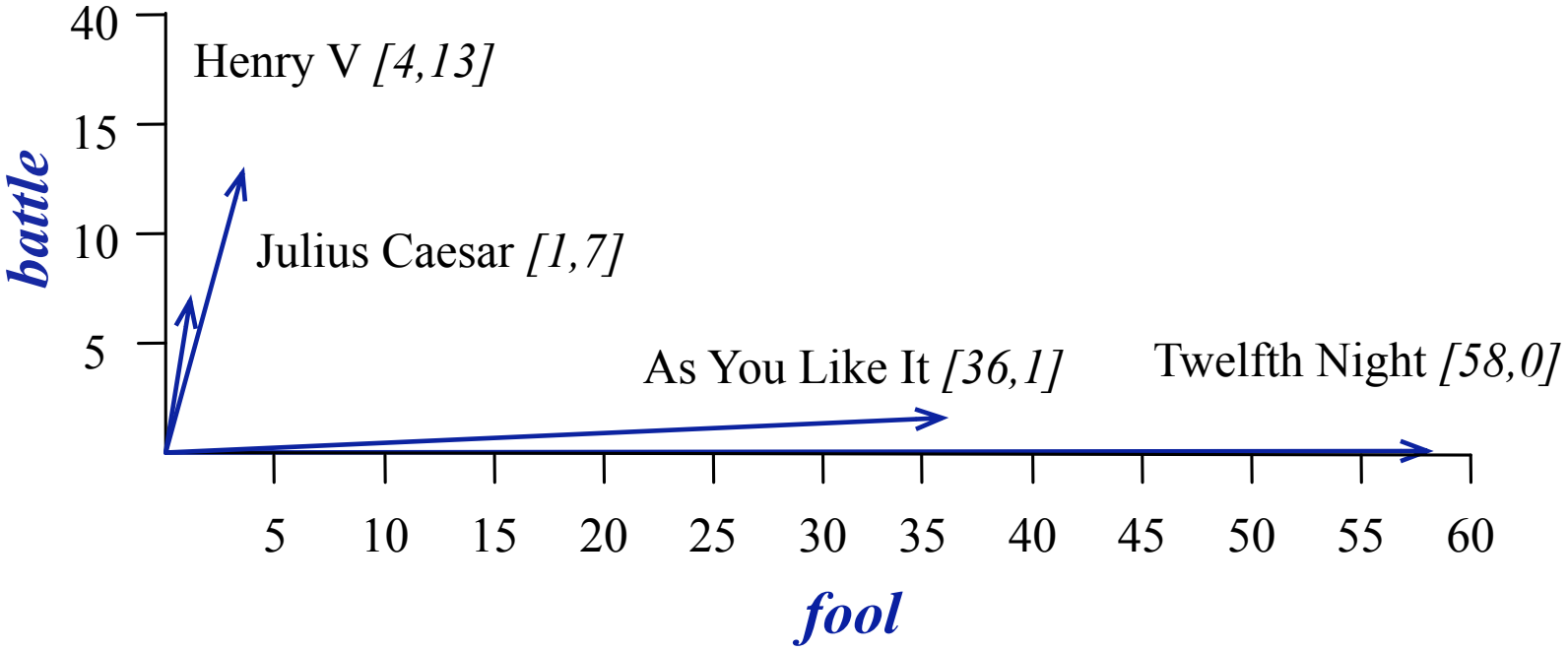
- Sparse
  - A common model from the information retrieval/search world
  - Words are represented as a function of the counts of words they co-occur with
  - Long, often the length of the vocabulary
- Dense
  - Representations trained from larger corpora using semi-supervised learning based on co-occurrence statistics
  - Short (or shorter than long) on the order of 500

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



Basis for Modern 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 for the two comedies are similar to each other and different than the histories

Comedies have more fools and wit and fewer battles.

Words Are 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"

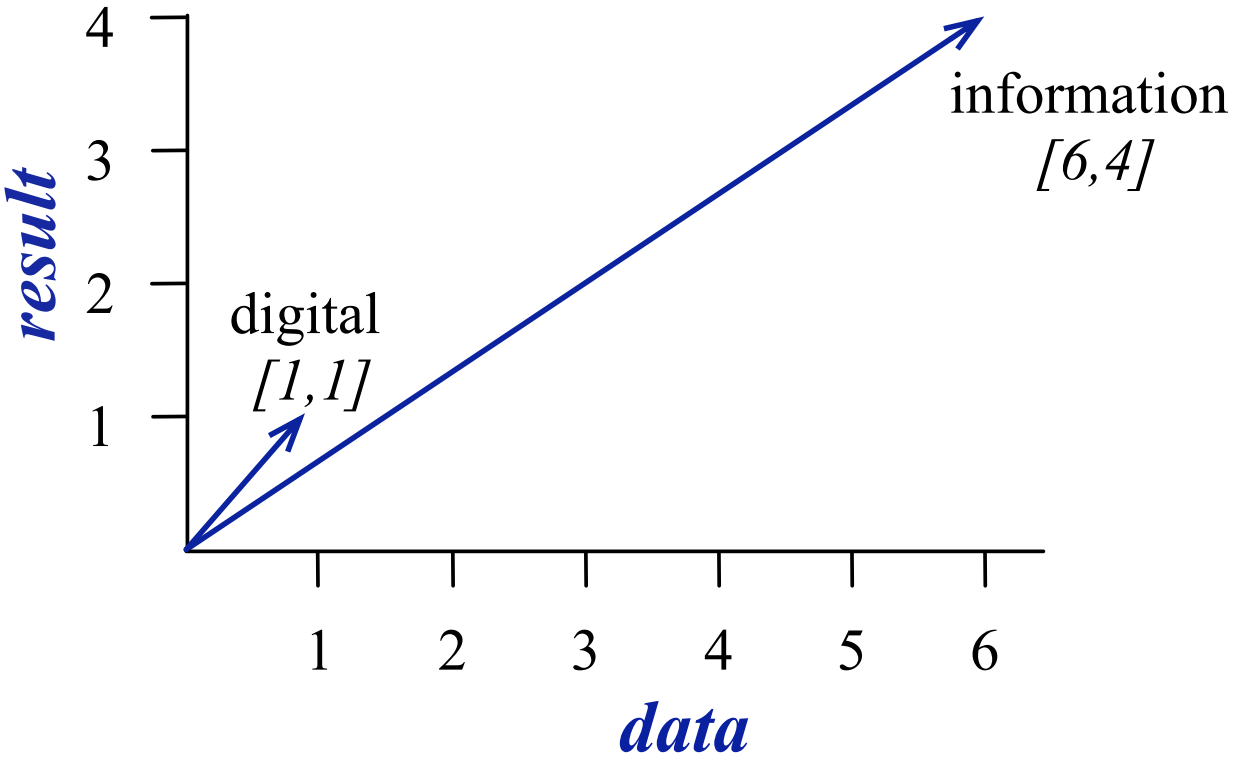
Word-Word Matrix

- Words are similar in meaning to the extent that 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	

Visualizing Word Vectors



Reminders from linear algebra

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

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

Cosine for Similarity

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

v_i is the count for word v in context i

w_i is the count for word w in context i.

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

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

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

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

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

Which pairs of words are 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) = $\frac{0+0+0}{\sqrt{1+0+0}\sqrt{0+1+4}} = 0$

Frequency?

- Frequency is clearly useful; data and information appear a lot together, that's clearly useful information.
- But frequent words like *the*, *it*, or *they* are not very informative about the context
  - The fact that *the* frequently appears with *data* is not helpful with respect to its meaning
- Need a function that resolves this. So instead of counts we'll use a function of the counts.

TF-IDF

- TF: term frequency

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

- IDF: inverse document frequency

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

Total # of docs in collection

# of docs that have word i

Spark-Jones, 1972

- TF-IDF value for word t in document d:

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

Summary: TF-IDF

- Represent word meanings as vectors of length |V| with values from TF-IDF scores derived from a document collection
- Can compare two words using cosine to see if they are similar/related
- Can compare documents using document vectors
- Basis for most text search libraries (lucene, solr, elastic search, etc.) and search engines

Sparse Vector Problem

- Words can have similar meanings and appear in "complementary" distributions. Meaning their vectors will be dissimilar
  - The words *boat* and *ship* don't appear together or even in the same kinds of contexts.
- But many of the words they do co-occur with, do co-occur together in similar contexts
  - Wait what?
- What we need is a way to reduce the dimensionality of the vectors in a way that reveals these deeper relationships.

Dense Vectors

- Why dense vectors?
  - Short vectors are easier to use as features in machine learning (fewer weights to learn)
  - Dense vectors generalize better than storing explicit counts
  - They do better at capturing synonymy among words that appear in different contexts
  - In practice, they work better

Variants

- Matrix methods
  - Latent Semantic Analysis
  - Non-negative matrix factorization
- Probabistic models
  - Latent Dirichlet Allocation
- Neural models
  - Word2vec, GLOVE

Word2Vec

- One popular embedding method
  - Very fast to train
  - Code available on the web
    - As are the embeddings
  - Simple neural model

Word2Vec

- Instead of counting how often each word *w* occurs near "*apricot*"
- Let's train a classifier on a binary prediction task:
  - Is *w* likely to show up in the same kind of context as "*apricot*" ?
- We don't care about this classifier
  - But we'll use the classifier weights learned by the classifier as the word embeddings

Semi-Supervised Training

- We can use raw unannotated text as supervised training data!
  - A word *w* occurring near *apricot* in real text serves as gold 'correct answer' to the question
  - "Is word *w* likely to show up near *apricot* ?"
- No need for hand-labeled annotation

Word2Vec: Skip-Gram Task

- Word2Vec provides a variety of options. Today we'll cover:
  - "skip-gram with negative sampling" (SGNS)

Skip-Gram Algorithm

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

Skip-Gram Objective

- Given a tuple (t,c) = target word, context word
  - (*apricot*, *jam*)
  - (*apricot*, *aardvark*)

- Return the probability that *c* is a real context word:

- P(+ | t, c)
- $P(- \mid t, c) = 1 - P(+ \mid t, c)$

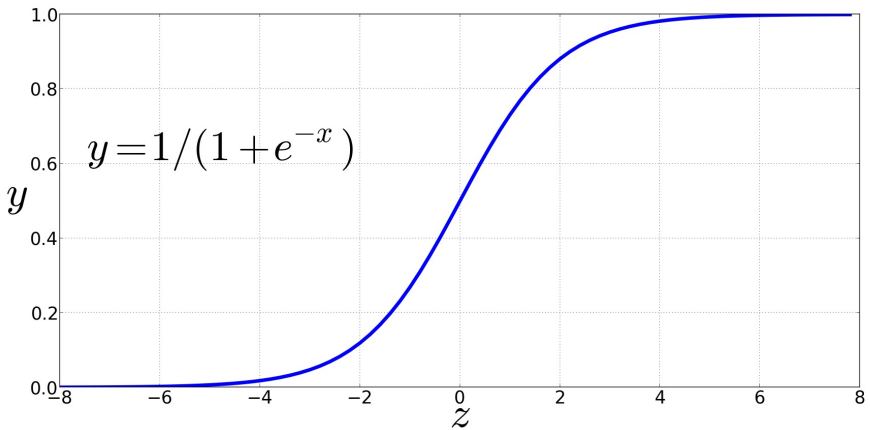
How to Compute p(+|t,c)?

- Intuition
  - Words that are similar in meaning should be near to each other.
  - Model similarity with dot-product!
  - Similarity(t,c) $\propto t \cdot c$
- Problem:
  - But dot product is not a probability, so…

Dot Product into a Probability

- Back to the sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Skip-Gram Training Data

- Training sentence:
- ... lemon, a tablespoon of **apricot** jam a pinch ...
- c1 c2 t c3 c4

- Training data: input/output pairs centering on *apricot*
- Asssume a +/- 2 word window

Skip-Gram Training

- Training sentence:
- ... lemon, a tablespoon of **apricot** jam a pinch ...
- c1 c2 t c3 c4

positive examples +	
t	c
apricot	tablespoon
apricot	of
apricot	preserves
apricot	or

- For each positive example, we'll create *k* pseudo-negative examples.
- Using *noise* words
- Any random word that isn't *t*

Skip-Gram Training

- Training sentence:
- ... lemon, a tablespoon of **apricot** jam a pinch ...
- c1 c2 t c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	preserves
apricot	or

k=2

negative examples -

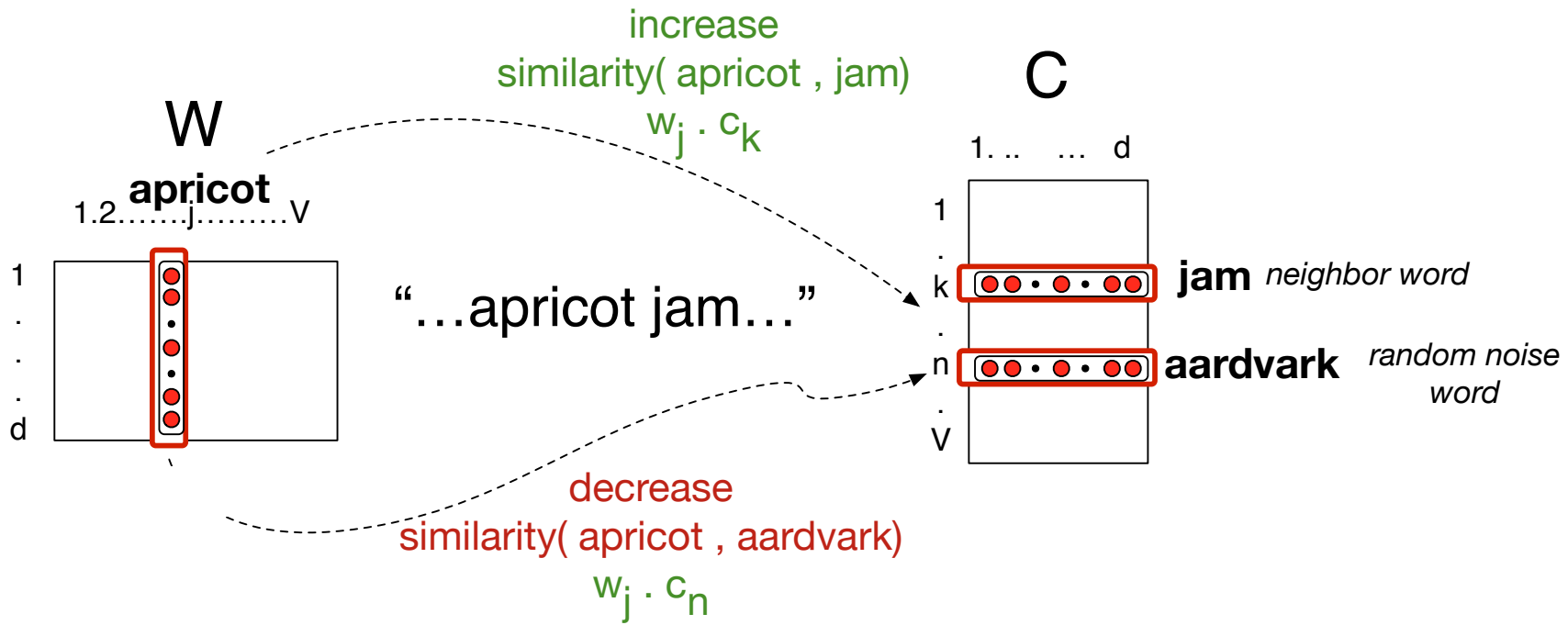
t	c	t	c
apricot	aardvark	apricot	twelve
apricot	puddle	apricot	hello
apricot	where	apricot	dear
apricot	coaxial	apricot	forever

Setup

- Represent words as vectors of some length (say 300), randomly initialized.
  - So, we start with 300 * |V| random parameters
    - Really 2 * that. We need context C and target W vectors
- Goal to is to find values that:
  - Maximize the similarity of the target word, context word pairs (t,c) drawn from the positive data
  - Minimize the similarity of the (t,c) pairs drawn from the negative data.

Learning

- Iterative process.
- We'll start with random weights
- Then adjust the word weights to
  - make the positive pairs more likely
    - More similar
  - and the negative pairs less likely
    - Less similar
- Over the entire training set


increase
similarity(apricot , jam)
$w_j \cdot c_k$
W
apricot
1.2……j……V
1
.
.
.
d
"…apricot jam…"
C
1. .. … d
1
.
k
.
n
.
V
jam
neighbor word
aardvark
random noise word
decrease
similarity(apricot , aardvark)
$w_j \cdot c_n$

Objective Criteria

- We want to maximize…

$$\sum_{(t,c)\in+} logP(+|t,c) + \sum_{(t,c)\in-} logP(-|t,c)$$

- Maximize the + label for the pairs from the positive training data, and the – label for the pairs sample from the negative data.

Train using gradient descent

- Two separate embedding matrices W and C are learned
- Can use W and throw away C, or merge them somehow

Skip-Gram Summary

- Start with V random 300-dimensional vectors as initial embeddings
- Use logistic regression, the second most basic classifier used in machine learning after naïve Bayes
  - 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 and keep the embeddings.

Dense Embeddings Without the Work

- Word2vec (Mikolov et al.) https://code.google.com/archive/p/word2vec/
- Fasttext http://www.fasttext.cc/
- GLOVE (Pennington, Socher, Manning) http://nlp.stanford.edu/projects/glove/

Properties of embeddings

Similarity depends on window size C

- C = ±2 The nearest words to *Hogwarts:*
  - *Sunnydale*
  - *Evernight*
- C = ±5 The nearest words to *Hogwarts:*
  - *Dumbledore*
  - *Malfoy*
  - *halfblood*

Analogy

- Remember your analogy questions from standardized tests

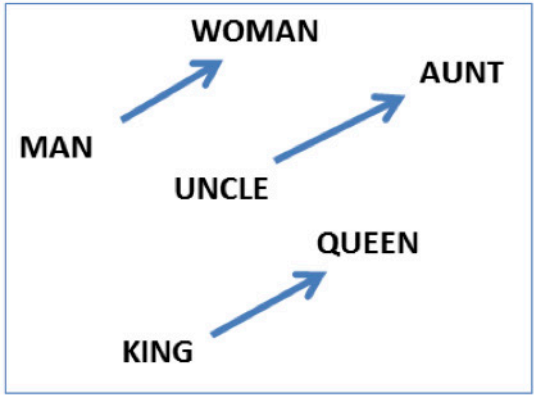
man : king :: woman: ?

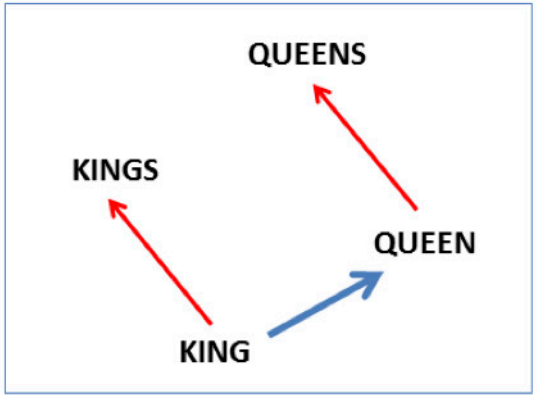
France : Paris :: Italy: ?

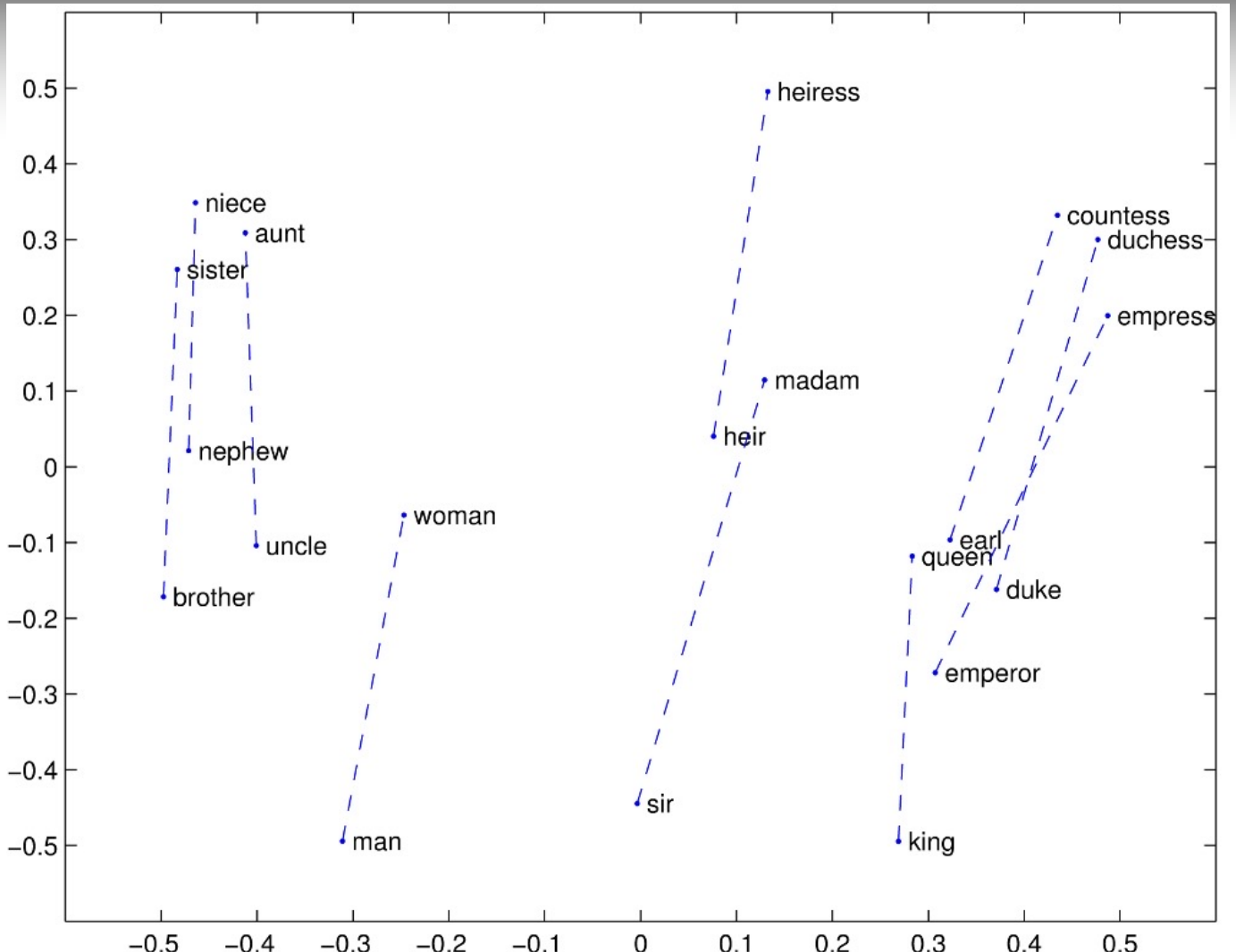
Analogy

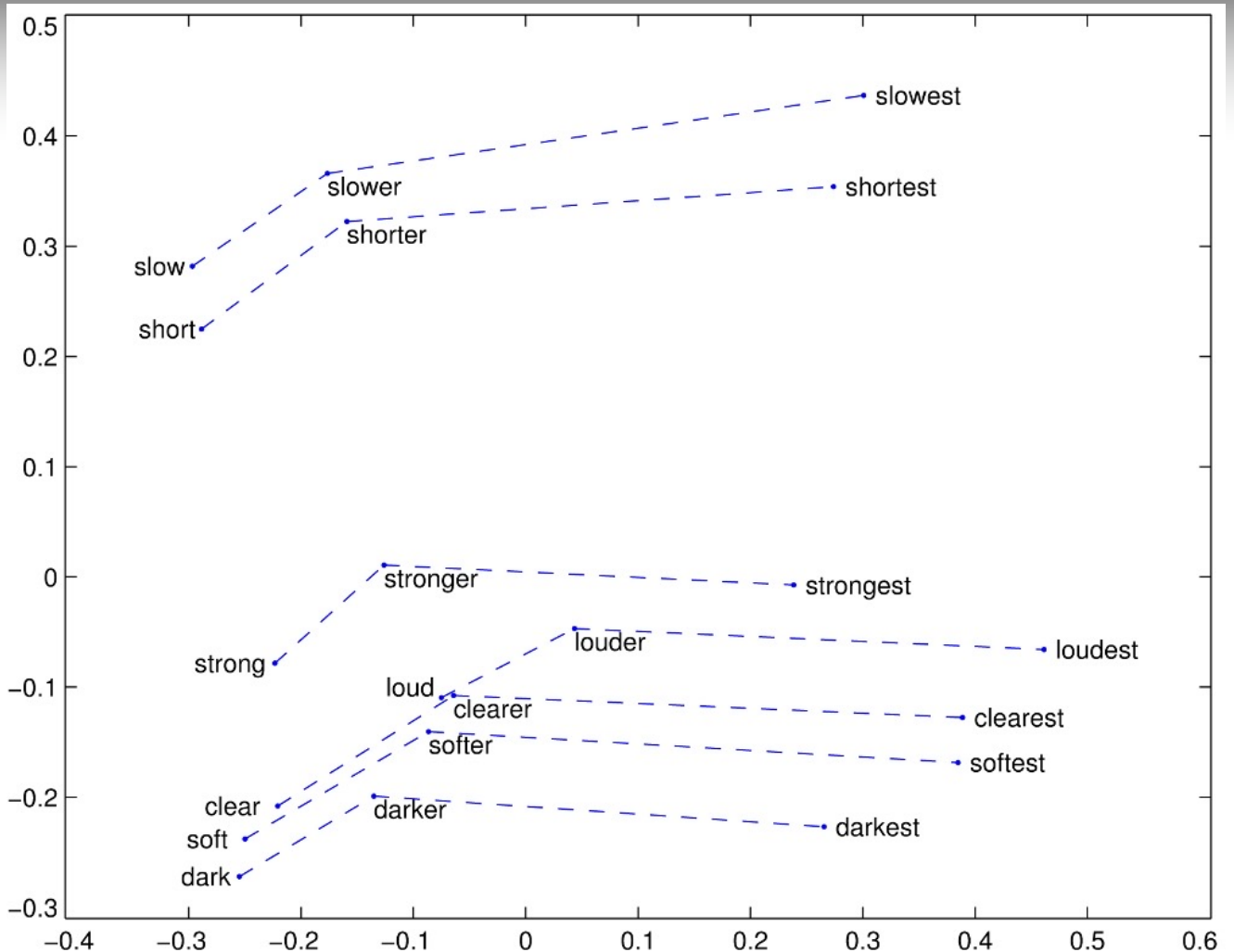
vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')






0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
niece
aunt
sister
nephew
uncle
brother
woman
man
heiress
madam
heir
sir
countess
duchess
empress
earl
queen
duke
emperor
king


slowest
slower
shortest
shorter
slow
short
stronger
strongest
louder
loudest
strong
loud
clearer
clearest
softer
softest
clear
darker
darkest
soft
dark
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4
−0.3
−0.2
−0.1
0
0.1
0.2
0.3
0.4
0.5
0.6

Embeddings Can Reflect Cultural Biases

- Ask “Paris : France :: Tokyo : x”
  - x = Japan
- Ask “father : doctor :: mother : x”
  - x = nurse
- Ask “man : computer programmer :: woman : x”
  - x = homemaker

Summary

- **Concepts (Word senses)**
  - Have a complex many-to-many association with **words** (homonymy, multiple senses)
  - Have many relations with each other
    - Synonymy, Antonymy, Superordinate
  - But are hard to define formally (necessary & sufficient conditions)
- **Embeddings** = vector models of meaning
  - More fine-grained than just a string or index
  - Especially good at modeling similarity/analogy
    - Just download them and use cosines!!
  - Can use sparse models (tf-idf) or dense models (word2vec, GLoVE)
  - Useful in practice but know they encode bias and stereotypes