Word embeddings

CS 490A, Fall 2020

Applications of Natural Language Processing

https://people.cs.umass.edu/~brenocon/cs490a_f20/

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

including slides from Eisenstein (2019) and Jurafsky & Martin 3rd Ed.

What do words mean?

First thought: look in a dictionary

http://www.oed.com/

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).

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.

Relation: Synonymity

Synonyms have the same meaning in some or all contexts.

- couch / sofa
- big / large
- automobile / car
- vomit / throw up
- Water / H_20

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

Relation: Similarity

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

car, bicycle

cow, horse

Ask humans how similar two words are on scale of 1-10

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)

in NLP, we commonly represent word types with **vectors**!

why use vectors to encode meaning?

- computing the similarity between two words (or phrases, or documents) is *extremely* useful for many NLP tasks

- Q: how **tall** is Mount Everest?

  A: The official **height** of Mount Everest is 29029 ft

Word similarity for plagiarism detection

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

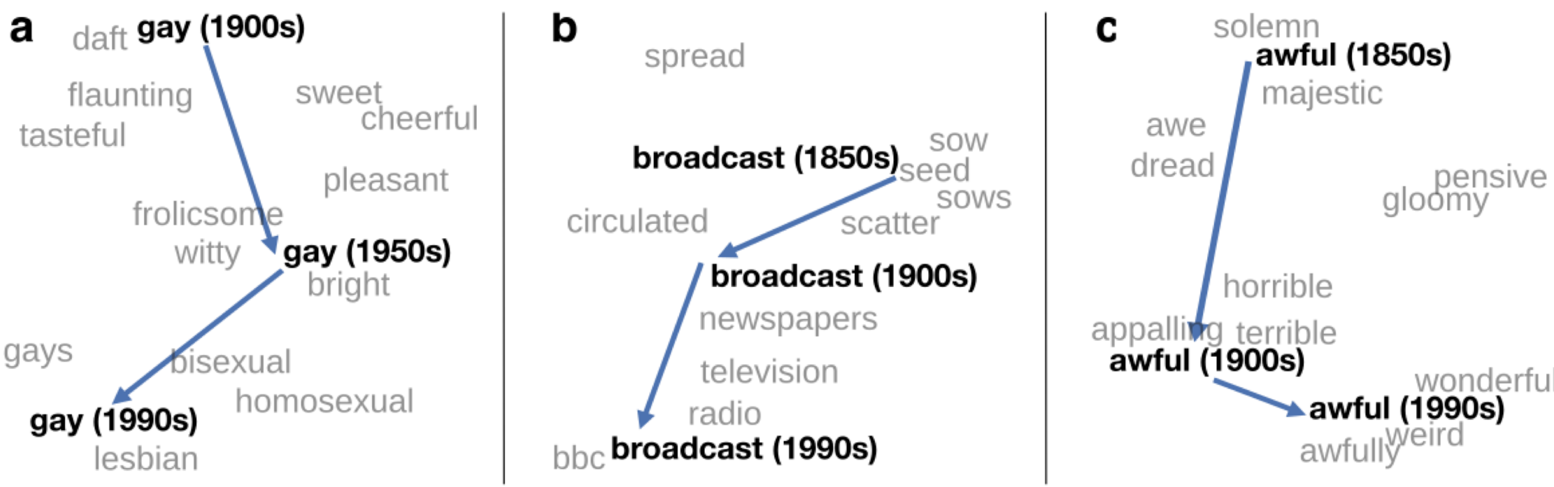
Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand

visualizing semantic word change over time



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

ML with text: one-hot vectors

- The bag-of-words representation
- represent each word as a vector of zeros with a single 1 identifying the index of the word

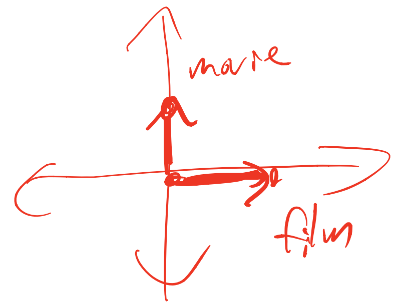
vocabulary
i
hate
love
the
movie
film

movie = <0, 0, 0, 0, 1, 0>

film = <0, 0, 0, 0, 0, 1>

what are the issues of representing a word this way?

all words are equally (dis)similar!



movie = <0, 0, 0, 0, 1, 0>

film = <0, 0, 0, 0, 0, 1>

dot product is zero!

these vectors are **orthogonal**

how can we compute a vector representation such that the dot product correlates with word similarity?

could also support *transfer learning*. labeled datasets are very small, but unlabeled data is large…

Transfer learning

- Sparsity problems for traditional bag-of-words
- Labeled datasets are small … but *unlabeled* data is much bigger!

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

Intuitions: Zellig Harris (1954):

- "oculist and eye-doctor … occur in almost the same environments"
- "If A and B have almost identical environments we say that they are synonyms."

Firth (1957):

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

Intuition of distributional word similarity

```
A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.
```

- From context words humans can guess **tesgüino** means...

Intuition of distributional word similarity

```
A bottle of tesgüino is on the table
Everybody likes tesgüino
Tesgüino makes you drunk
We make tesgüino out of corn.
```

- From context words humans can guess **tesgüino** means…
- an alcoholic beverage like **beer**
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

Word-word co-occurence 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	

cosine similarity of two vectors

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

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 $\vec{w}$

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!

Pointwise Mutual Information

$P(x,y) = P(x)P(y)$

Pointwise mutual information:

Do events x and y co-occur more than if they were independent?

$=0$ $=1$

control for overall frequency

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

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

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

what is the range of values **PMI**(w_1, w_2) can take?

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

$$(-\infty, \infty)$$

Positive **PMI**(w_1, w_2):

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

dense word vectors

- issue: context vectors are long and sparse (why an issue?) ① Comp. issues ② weight sharing
- model the meaning of a word as an **embedding** in a vector space
    - this vector space is commonly “low” dimensional (e.g., 100-500d).
    - what is the dimensionality of a one-hot word representation?

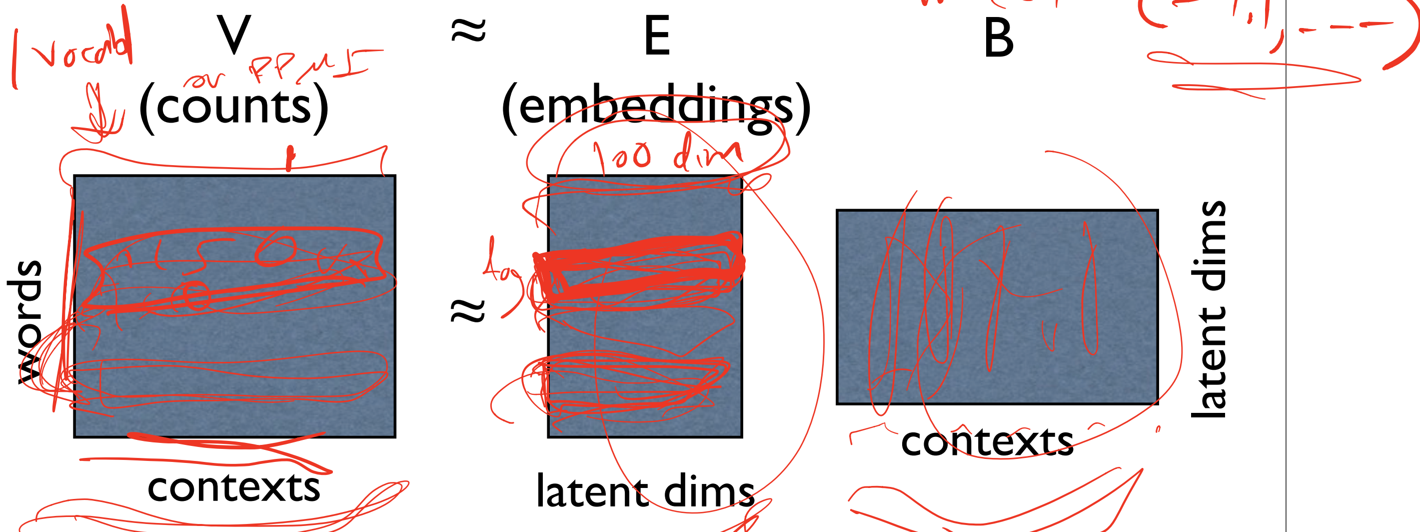
      Vocab size = |V|
- embeddings are real-valued vectors (not binary or counts)

Learning word embeddings from word-context data

- Sparse vectors
  - Context co-occurrence frequencies/PPMI (Just count and normalize, no learning)
- Latent vectors [next]
  - Matrix factorization
  - word2vec context prediction (SGNS)
- Latent hierarchy
  - Brown clusters
  - …

also
GLOVE

Matrix factorization



Reconstruct the co-occurrence matrix

$$V_{i,c} \approx \sum_k E_{i,k} B_{k,c}$$

Singular Value Decomposition learns E,B
(or other matrix factorization techniques)

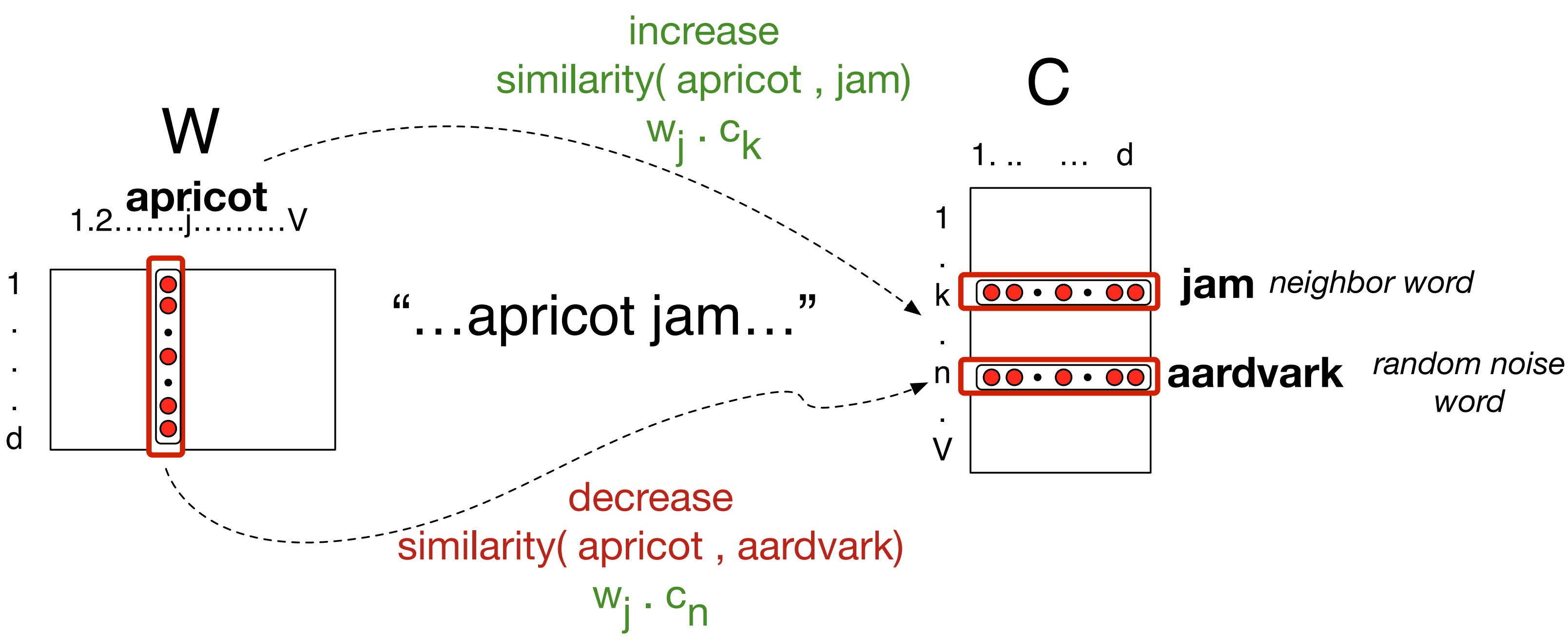
↔

Preserve pairwise distances between words i, j

$$V_i^{\mathsf{T}} V_j \approx E_i^{\mathsf{T}} E_j$$

Eigen Decomposition learns E

in practice, we learn two different sets of embeddings (**W** for *target* words, **C** for context words), but throw away **C**



Word embedding models

- GLOVE: one way to do that matrix factorization
- SVD: another
- word2vec: same thing, but depicted as *predicting* surrounding contexts

Stopped here 10/6

Defining contexts

Window size C affects the nature of the similarity something like…

syntax <—> basic meaning <—> topical meaning

C = ±2 The nearest words to *Hogwarts:*

- *Sunnydale*
- *Evernight*

C = ±5 The nearest words to *Hogwarts:*

- *Dumbledore*
- *Malfoy*
- *halfblood*

Defining contexts

*The moment one **learns** English, complications set in* (Alfau, 1999)	
Brown Clusters	$\{one\}$
WORD2VEC, $h = 2$	$\{moment, one, English, complications\}$
Structured WORD2VEC, $h = 2$	$\{(moment, -2), (one, -1), (English, +1), (complications, +2)\}$
Dependency contexts,	$\{(one, \text{NSUBJ}), (English, \text{DOBJ}), (moment, \text{ACL}^{-1})\}$

Alternate/mis- spellings

- Distributional methods are really good at this
- Brown clusters on Twitter: http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html

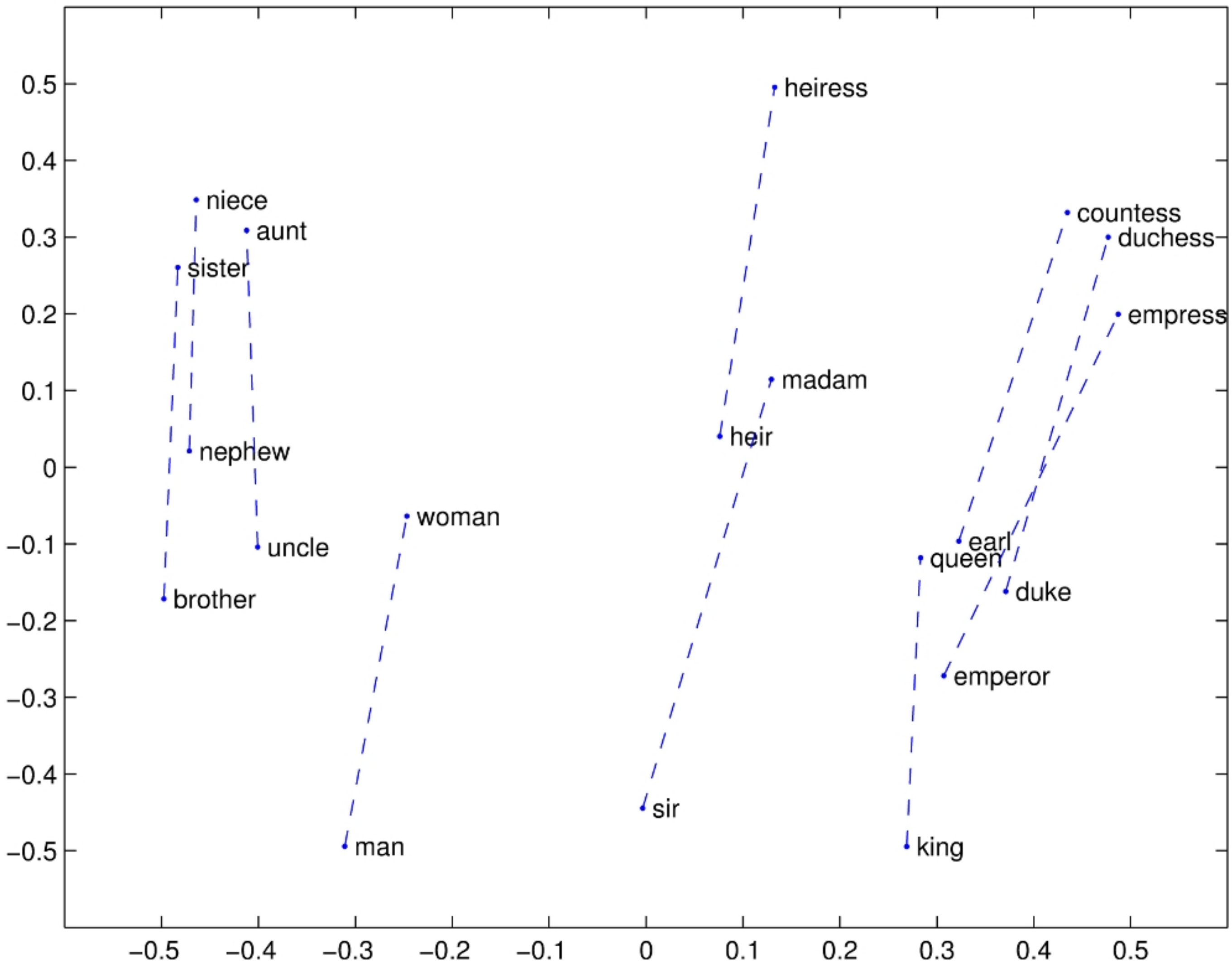
Pre-trained embeddings

- Widely useful. But make sure you know what you're getting!
  - Examples: GLOVE, fasttext, word2vec, etc.
  - Is the corpus similar to what you care about?
  - Should you care about the *data*?

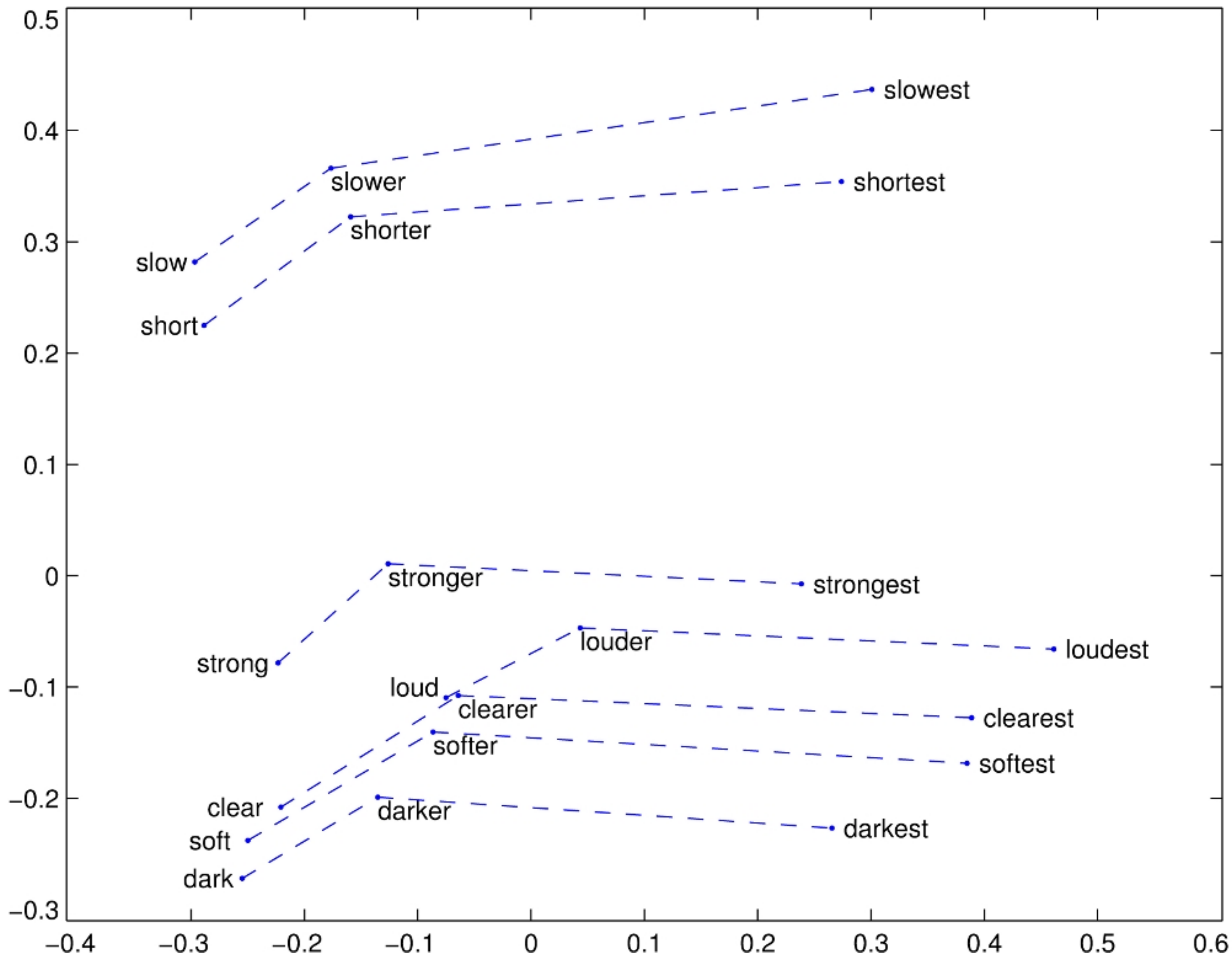
Evaluating embeddngs

- Intrinsic evaluations
  - Compare embeddings' word pair similarities to human judgments
    - TOEFL: "*Levied* is closest to *imposed*, *believed*, *requested*, *correlated*"
    - Numerical similarity judgments (e.g. Wordsim-353)
  - Word analogies and other evaluations possible too — though much controversy (see Linzen)
- Extrinsic evaluation: use embeddings in some task

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



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



Extensions

- Alternative: Task-specific embeddings (always better...)
- Multilingual embeddings
- Better contexts: direction, syntax, morphology / characters...
- Phrases and meaning composition
  - vector(red cat) = g(vector(red), vector(cat))
  - vector(black cat) = g(vector(black), vector(cat))
  - vector(hardly awesome) = g(vector(hardly), vector(awesome))
  - *(Averaging sometimes works ok…)*

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 *Advances in Neural Information Processing Systems*, 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

huge concern for NLP systems deployed in the real world that use embeddings!

Occupations		Adjectives	
Man	Woman	Man	Woman
carpenter	nurse	honorable	maternal
mechanic	midwife	ascetic	romantic
mason	librarian	amiable	submissive
blacksmith	housekeeper	dissolute	hysterical
retired	dancer	arrogant	elegant
architect	teacher	erratic	caring
engineer	cashier	heroic	delicate
mathematician	student	boyish	superficial
shoemaker	designer	fanatical	neurotic
physicist	weaver	aimless	attractive

Table 7: Top occupations and adjectives by gender in the Google News embedding.

Changes in framing: adjectives associated with Chinese

Garg, Nikhil, Schiebinger, Londa, Jurafsky, Dan, and Zou, James (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644

1910	1950	1990
Irresponsible	Disorganized	Inhibited
Envious	Outrageous	Passive
Barbaric	Pompous	Dissolute
Aggressive	Unstable	Haughty
Transparent	Effeminate	Complacent
Monstrous	Unprincipled	Forceful
Hateful	Venomous	Fixed
Cruel	Disobedient	Active
Greedy	Predatory	Sensitive
Bizarre	Boisterous	Hearty