CS388: Natural Language Processing

Lecture 5: Word Embeddings

Elias Stengel-Eskin

TEXAS
The University of Texas at Austin

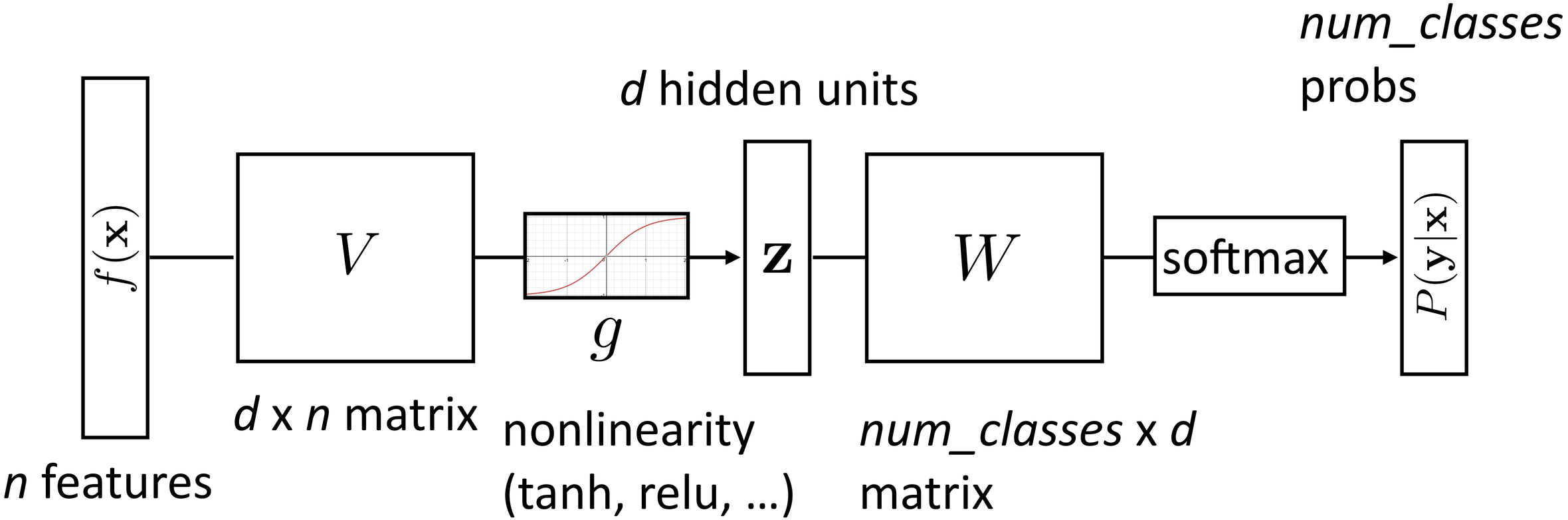
Slide Credits: Greg Durrett

Administration

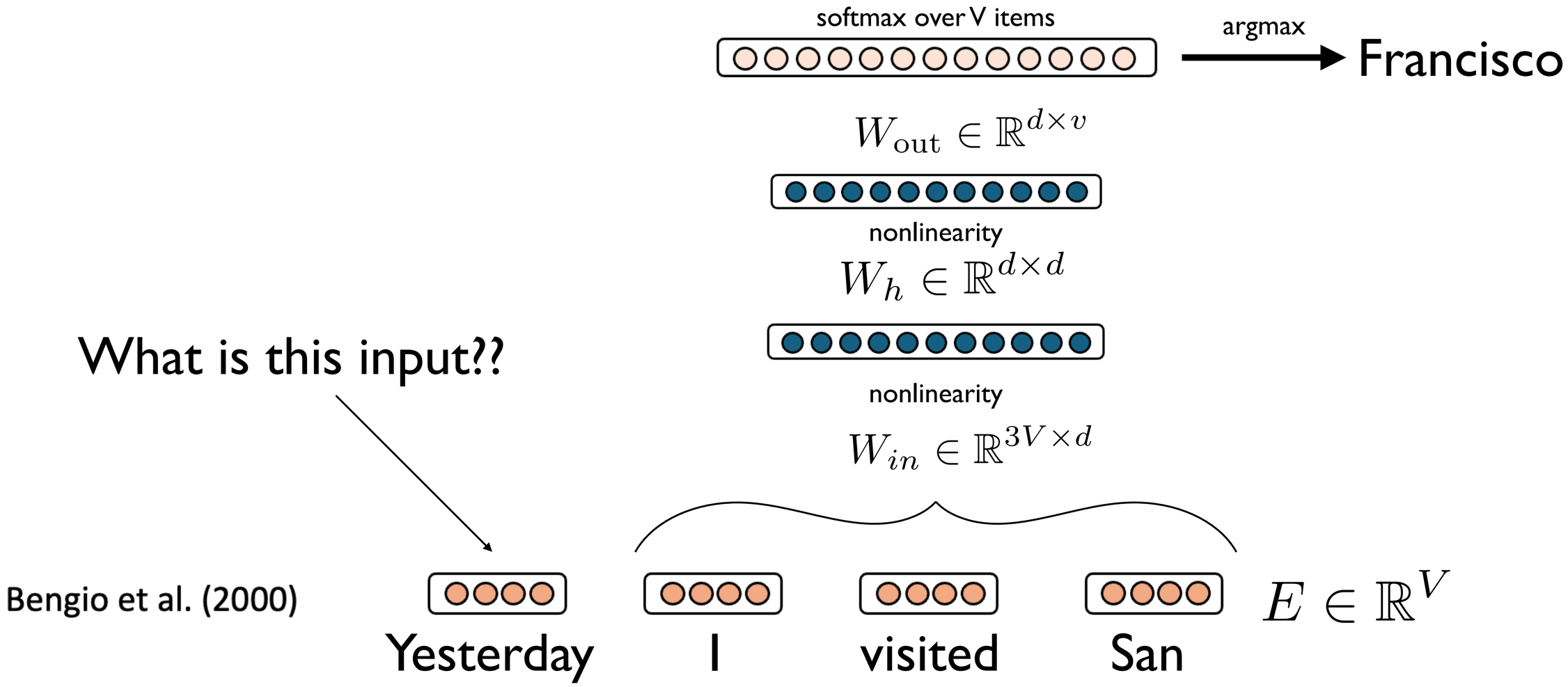
- Homework 1: done
- Quiz 1: next class

Recall: Feedforward NNs

$$P(\mathbf{y}|\mathbf{x}) = \text{softmax}(W\,g(V\,f(\mathbf{x})))$$



Recall: Neural LM



This Lecture

- Word representations
- Skip-gram/word2vec
- GloVe
- Other word embedding methods
- Evaluating word embeddings

Static Word Representations

What did we do so far?

- One-hot representation
- What are some problems with this?
- 1. grows with V. More words = bigger dimensionality
- 2. What happens if we compute similarity?

V-dimensional vectors:

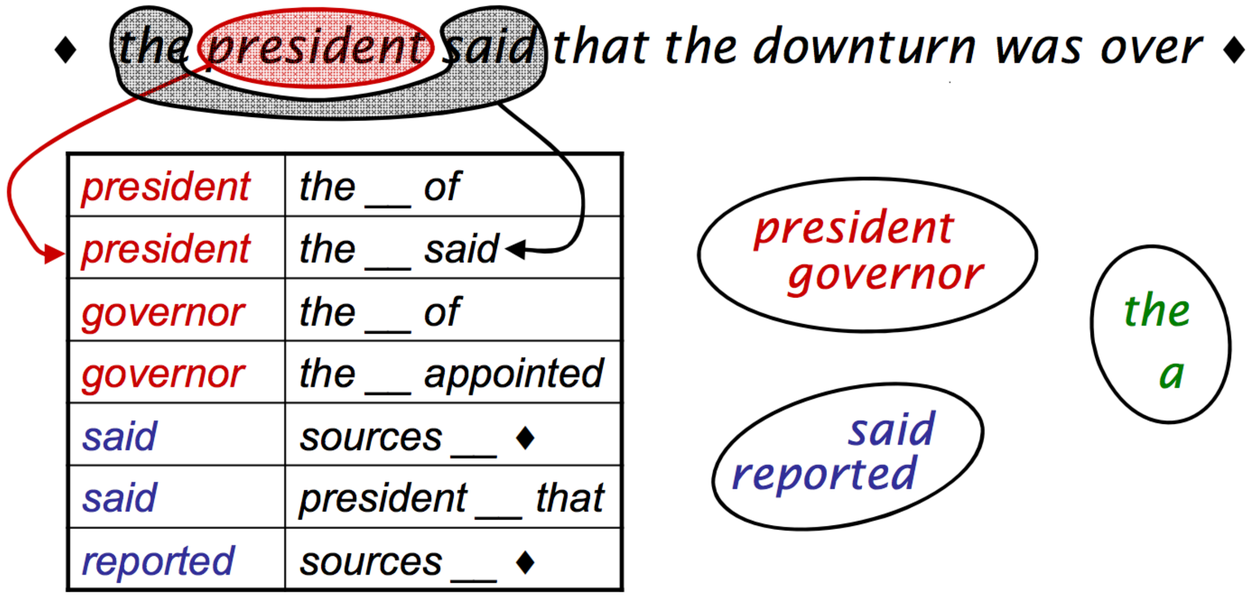
word 1: [1,0,0,…]
word 2: [0,1,0,…]
word 3: [0,0,1,…]

cosine(word i, word j) = 0

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A|| \times ||B||}$$

Word Representations

- Neural networks work very well at continuous data, but words are discrete
- Continuous model <-> expects continuous semantics from input
- “You shall know a word by the company it keeps” Firth (1957)



[Finch and Chater 92, Shuetze 93, many others]

slide credit: Dan Klein

Word Embeddings

- Want a vector space where similar words have similar embeddings

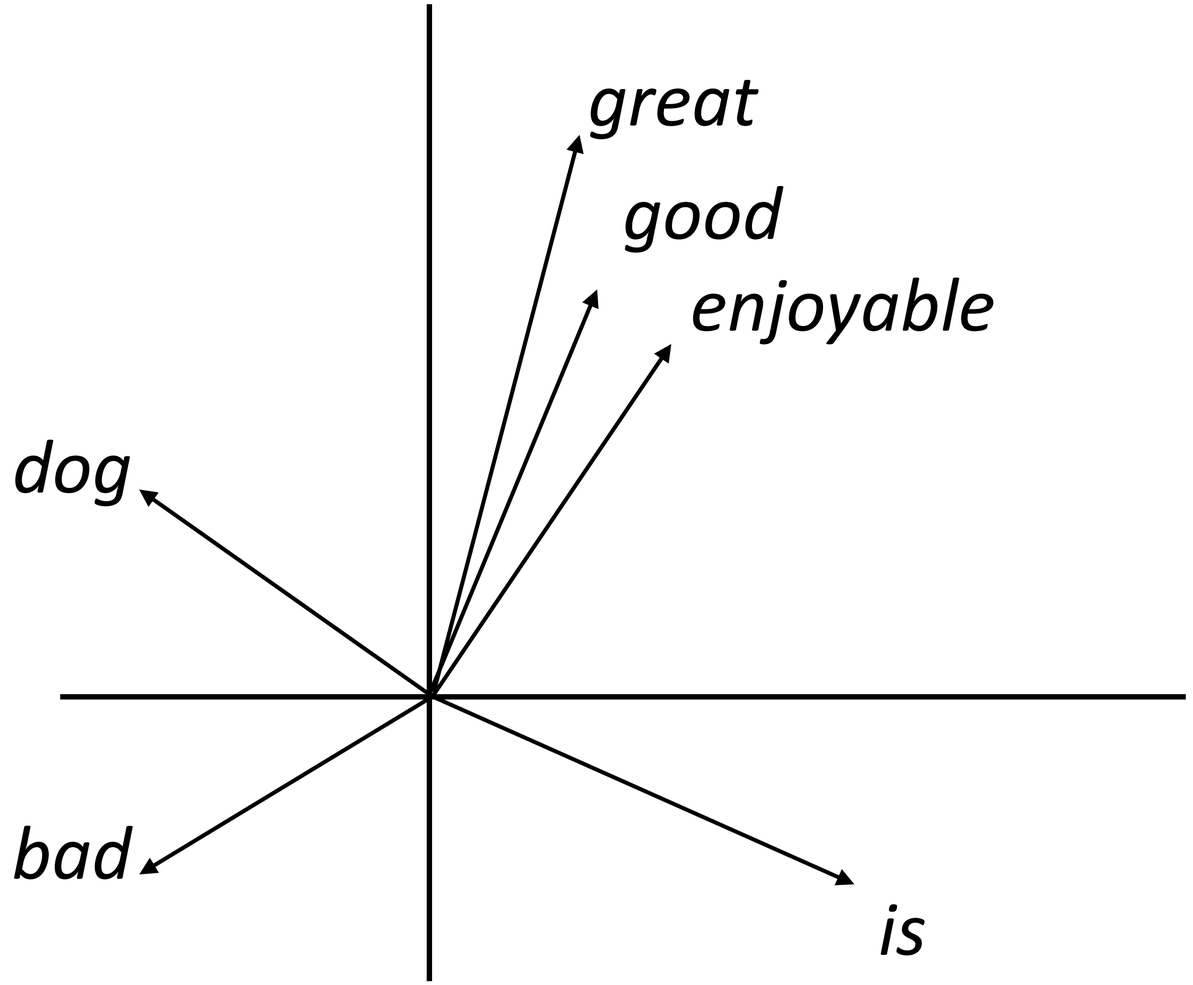
the movie was great

≈

the movie was good

- Goal: come up with a way to produce these embeddings

- For each word, want "medium" dimensional vector (50-300 dims) representing it



Skip-gram

Skip-Gram

- Input: a corpus of raw text. (Same as the input to “real” language modeling)
- Output: a set of *embeddings*: a real-valued vector for each word in the vocabulary
- We are going to learn these by setting up a fake prediction problem: predict a word’s context from that word

the dog bit the man

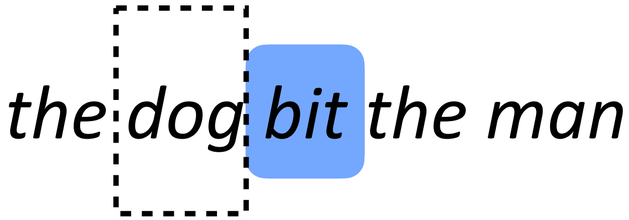
*(*word = *bit,* context = *dog)*

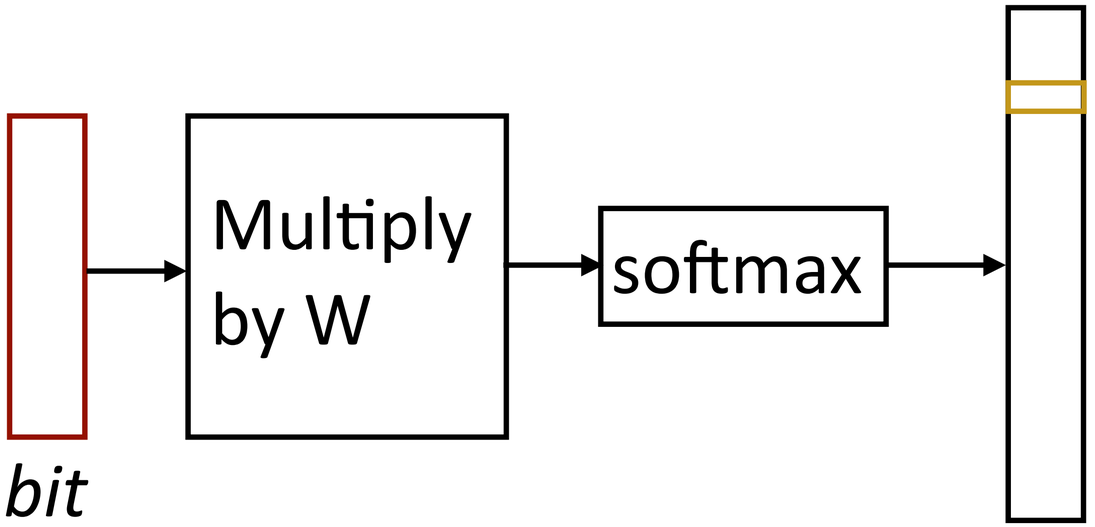
*(*word = *bit,* context = *the)*

Mikolov et al. (2013)

Skip-Gram

- Predict one word of context from word





gold = *dog*

$$P(w'|w) = \text{softmax}(We(w))$$

- Another training example: *bit* -> *the*
- Parameters: *d* x |V| vectors, |V| x *d* output parameters (W) (also usable as vectors!). *d* is a hyperparameter

Mikolov et al. (2013)

Using Skip-Gram

- Context window size: how many words around the "center" word do we look?

the dog bit the man

k=1: two words of context

k=2: four words of context

  - Advantages/disadvantages of different sizes of *k*?

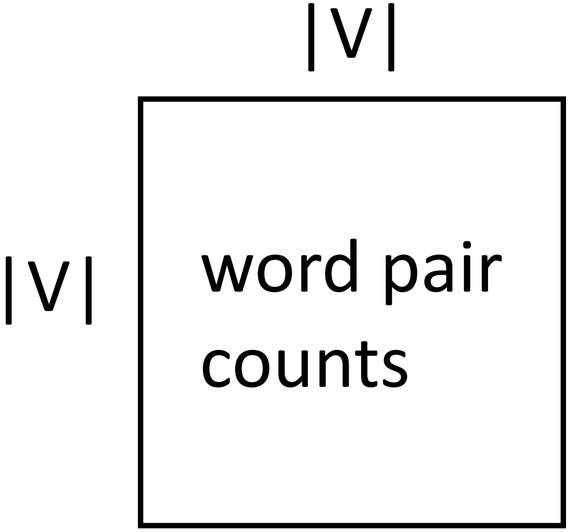
- Training: maximize log likelihood of the examples derived given *k*, summed over a corpus (but we'll never use the model as is, only its embeddings)
- Initialization: need to randomly initialize in a reasonable way
- Vector size: controls capacity of model

Mikolov et al. (2013)

GloVe

GloVe (**Glo**bal **Ve**ctors)

- Counts matrix: how often does word occur in context of another word (e.g. count(apple, iphone))
- GloVe also operates on counts matrix, weighted regression on the log co-occurrence matrix



- Objective = $\sum_{i,j} f(\text{count}(w_i, c_j)) \left(w_i^\top c_j + a_i + b_j - \log \text{count}(w_i, c_j)\right)^2$
- Constant in the dataset size (just need counts), quadratic in voc size
- One of the most common word vectors used today (50000+ citations)

Pennington et al. (2014)

GloVe (**Glo**bal **Ve**ctors): Example

- Objective = $\sum_{i,j} f(\text{count}(w_i, c_j)) \left(w_i^\top c_j + a_i + b_j - \log \text{count}(w_i, c_j)\right)^2$

	the	*dog*	*cat*	*ran*
the	0	200	200	0
dog	200	0	0	15
cat	200	0	0	15
ran	0	15	15	0

Linear regression with 12 pairs: each element is plugged into the above equation

 + constant = log count of pair

(made up values — matrix will generally be symmetric, though)

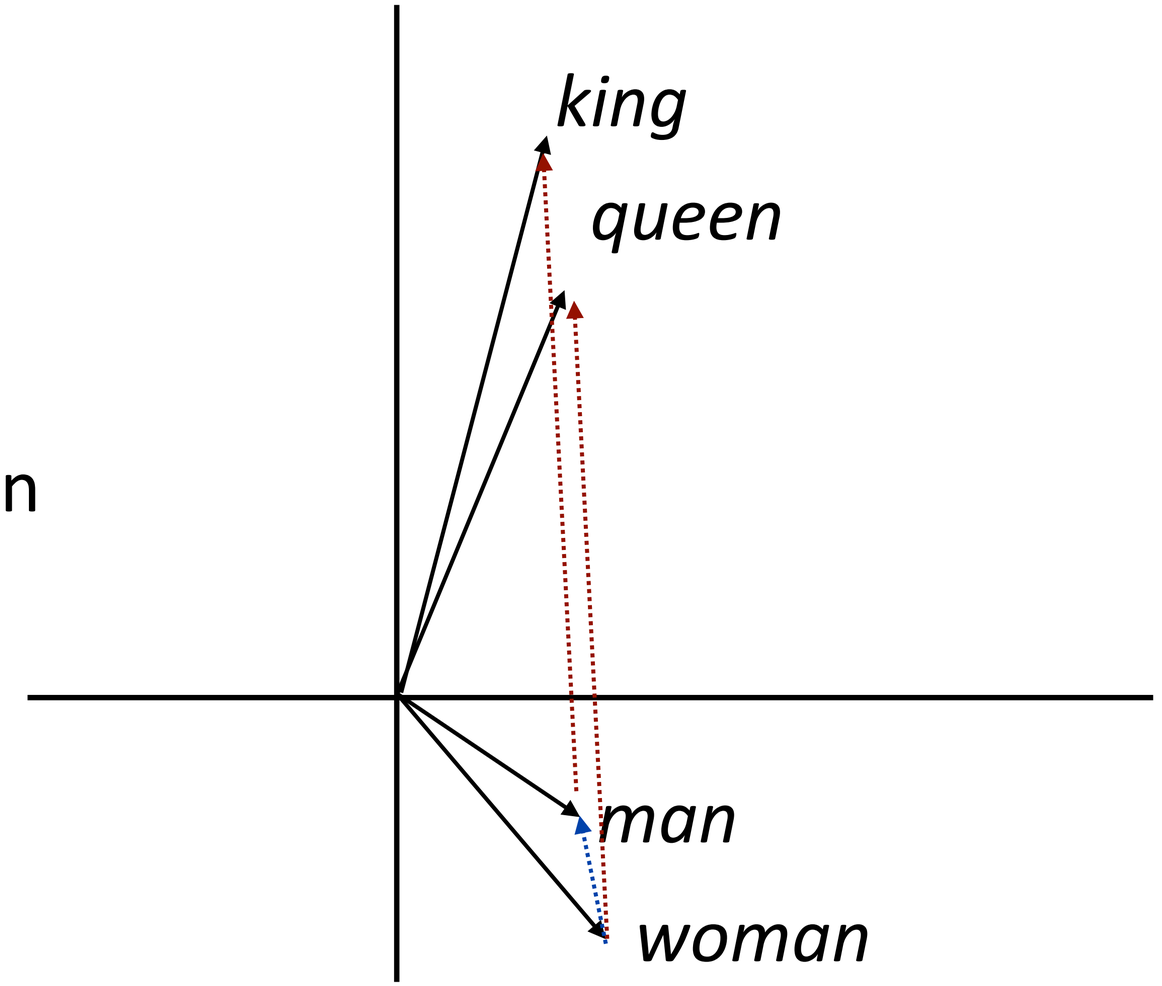
Pennington et al. (2014)

Analogies

(king - man) + woman = queen

king + (woman - man) = queen

- Why would this be?
- woman - man captures the difference in the contexts that these occur in



GloVe Motivation

Table 1: Co-occurrence probabilities for target words *ice* and *steam* with selected context words from a 6 billion token corpus. Only in the ratio does noise from non-discriminative words like *water* and *fashion* cancel out, so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific of steam.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- GloVe objective is derived to preserve regularities in cooccurrence of words with other words

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

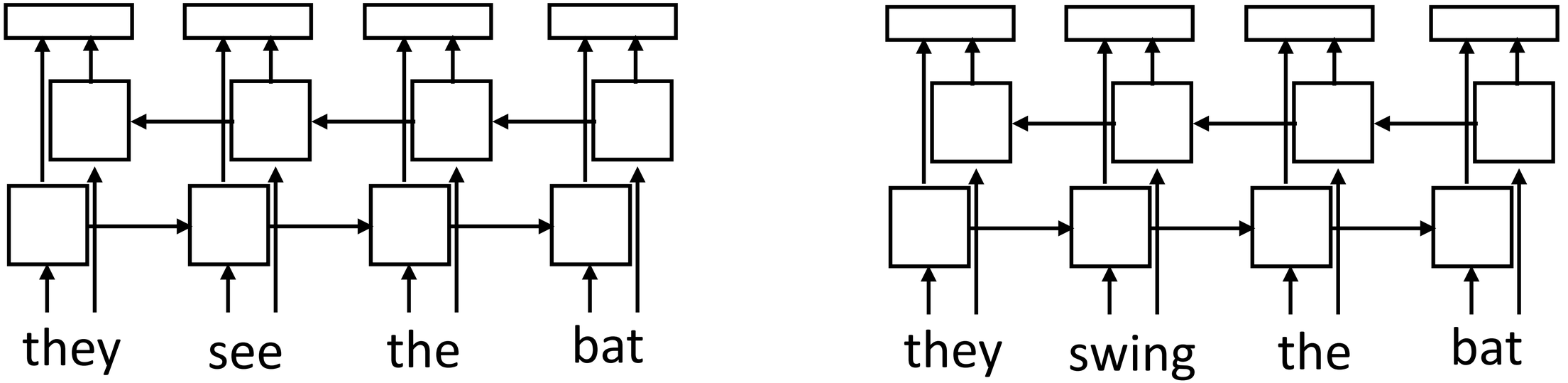
Pennington et al. (2014)

Problem?

- (at least) two problems with this approach
- 1. Word senses
  - *bat: (noun)* baseball bat, *(noun)* bat (mammal), *(verb)* to hit, *(verb)* bat eyelashes, …
  - All of this is squeezed into a single vector
- What about composition? How do we combine words?

Preview: Context-dependent Embeddings

- How to handle different word senses? One vector for *bat*



- Train a neural language model to predict the next word given previous words in the sentence, use its internal representations as word vectors
- *Context-sensitive* word embeddings: depend on rest of the sentence
- *Huge* improvements across nearly all NLP tasks over GloVe

Peters et al. (2018)

Evaluating Word Embeddings

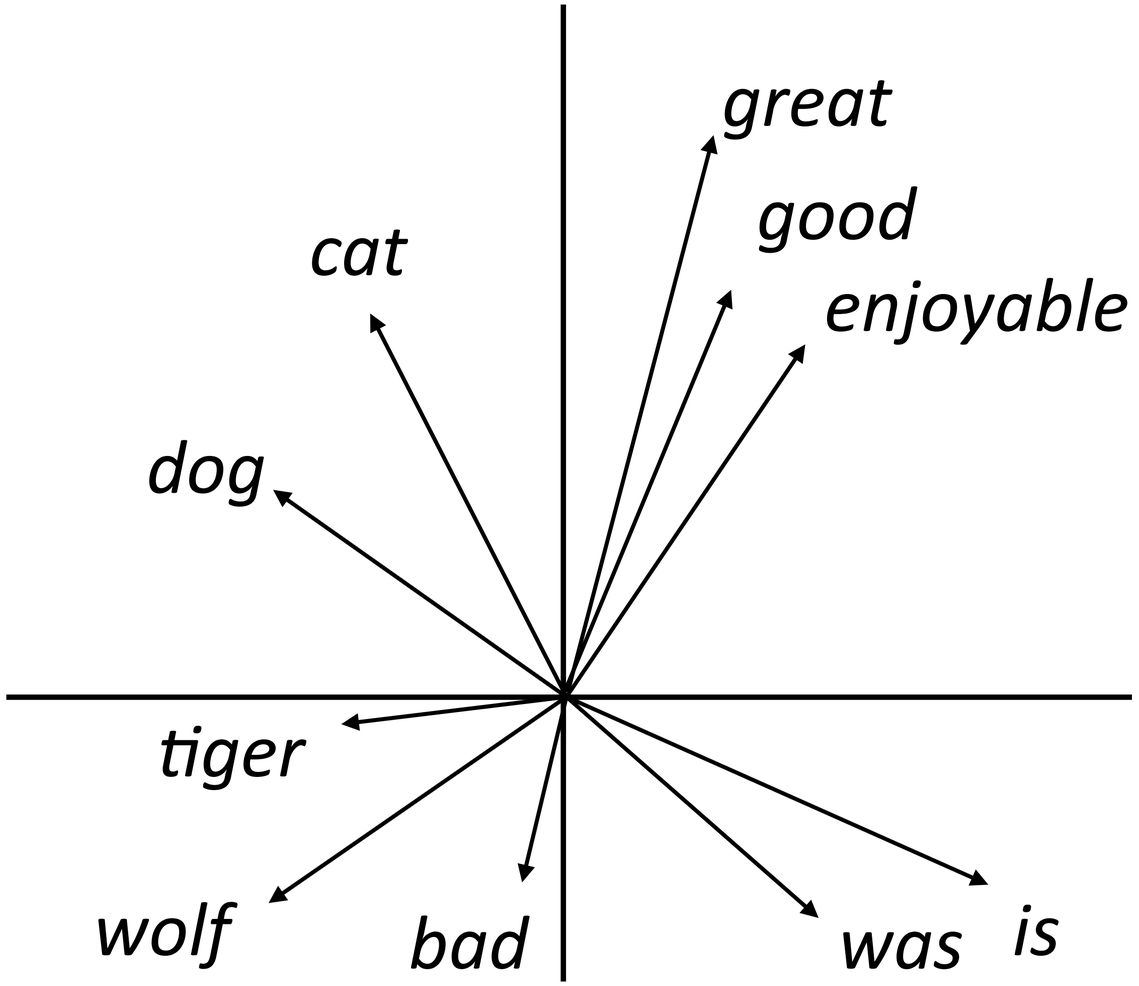
Suggestions?

Evaluating Word Embeddings

- What properties of language should word embeddings capture?
- Similarity: similar words are close to each other
- Analogy:

  good is to best as smart is to ???

  Paris is to France as Tokyo is to ???



Similarity

Method	WordSim Similarity	WordSim Relatedness	Bruni et al. MEN	Radinsky et al. M. Turk	Luong et al. Rare Words	Hill et al. SimLex
PPMI	.755	**.697**	.745	.686	.462	.393
SVD	**.793**	.691	**.778**	.666	**.514**	.432
SGNS	**.793**	.685	.774	**.693**	.470	**.438**
GloVe	.725	.604	.729	.632	.403	.398

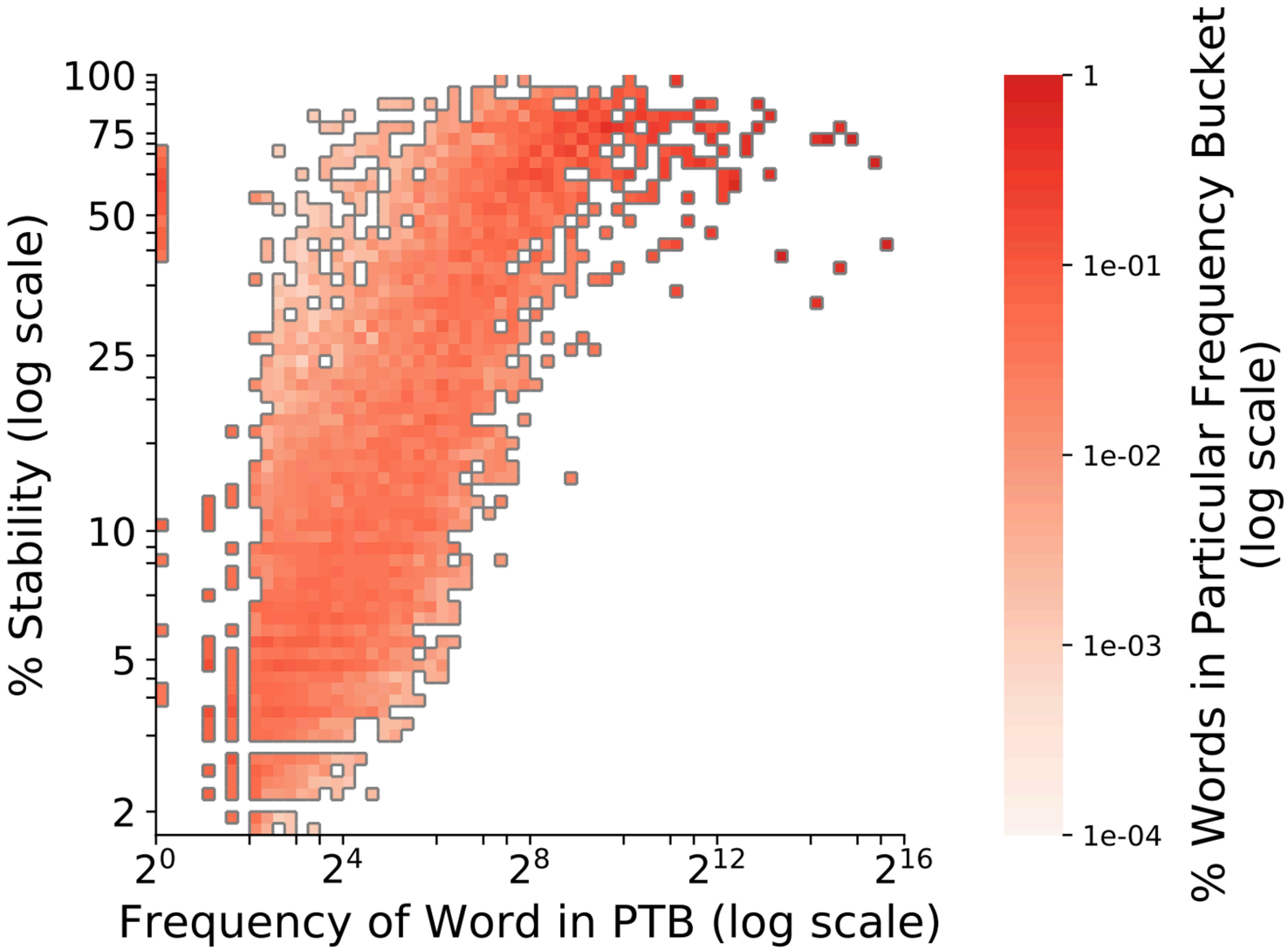
- SVD = singular value decomposition on PMI matrix
- GloVe does not appear to be the best when experiments are carefully controlled, but it depends on hyperparameters + these distinctions don't matter in practice

Levy et al. (2015)

Stability

- To what extent are the relationships captured by word embeddings consistent?

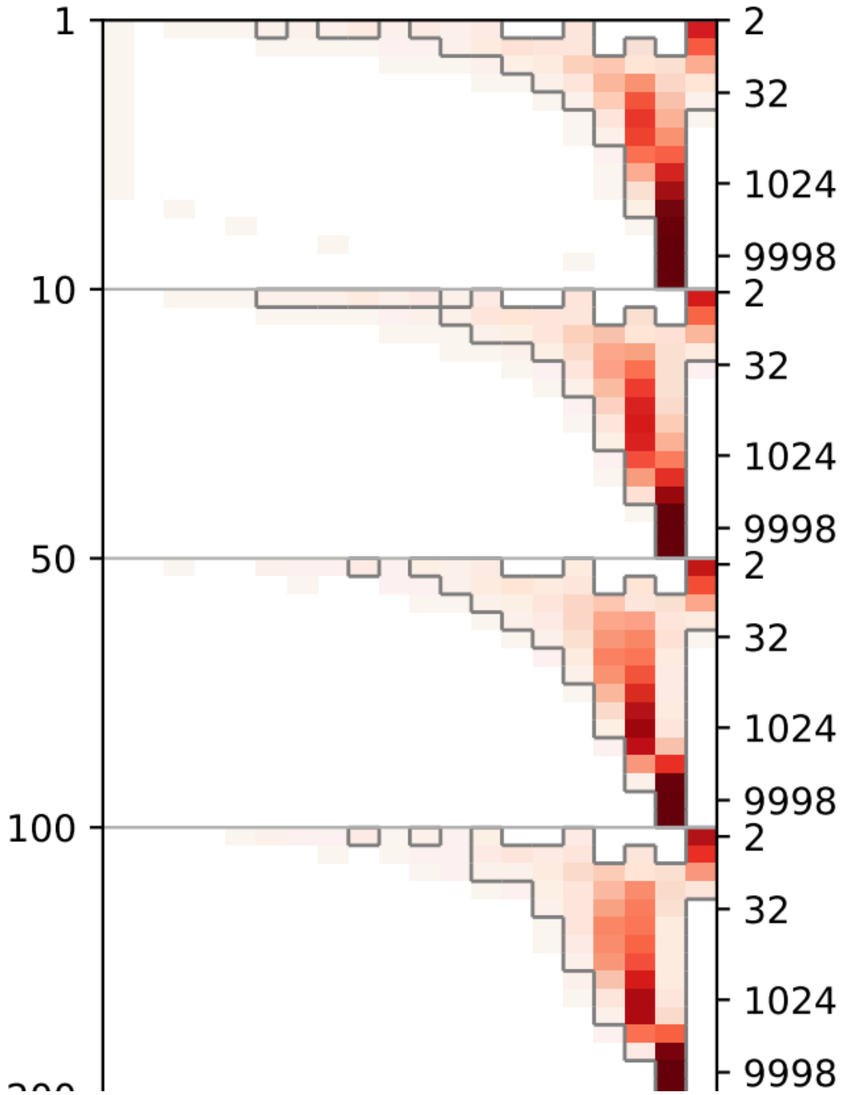
- Stability: percent overlap between nearest neighbors in embedding space if you retrain embeddings from different initialization



Burdick et al. (2018)

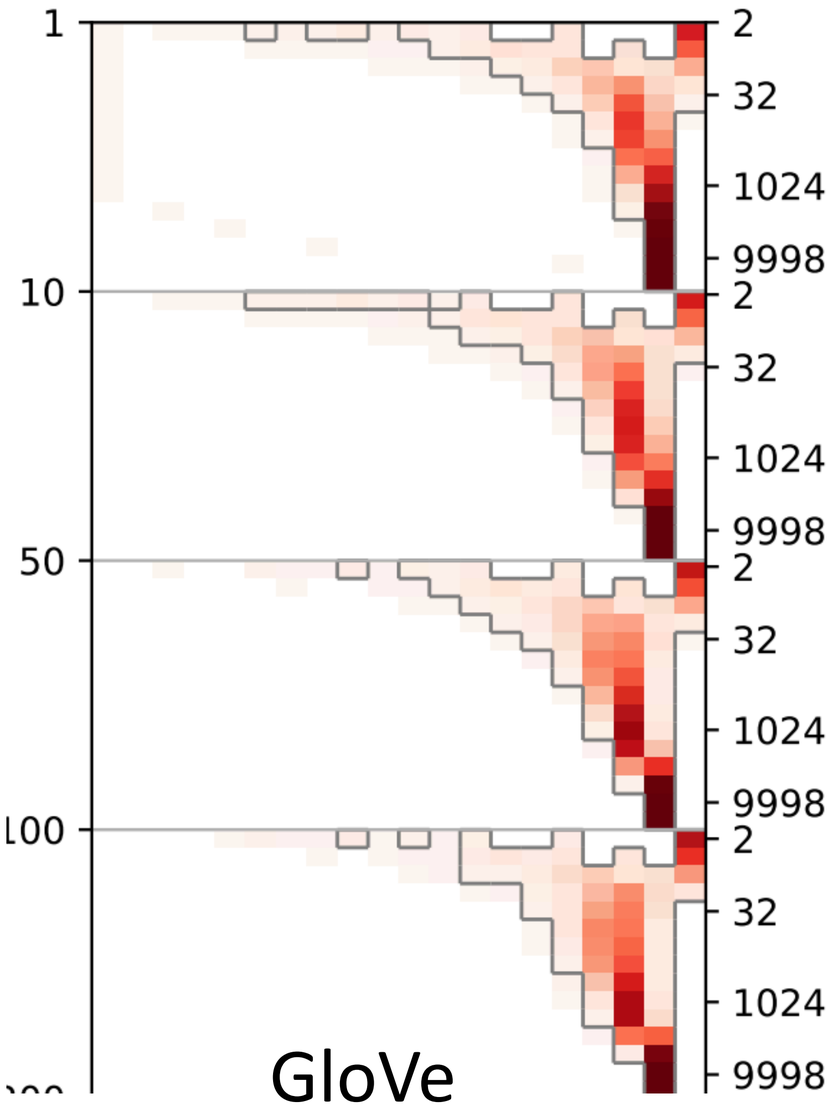
Stability: GloVe

- Left y-axis: bucketed corpus frequency
- Right y-axis: number of neighbors
- x-axis: percent of neighbors stable across samples
- Being all the way to the right is better (most neighbors are stable)

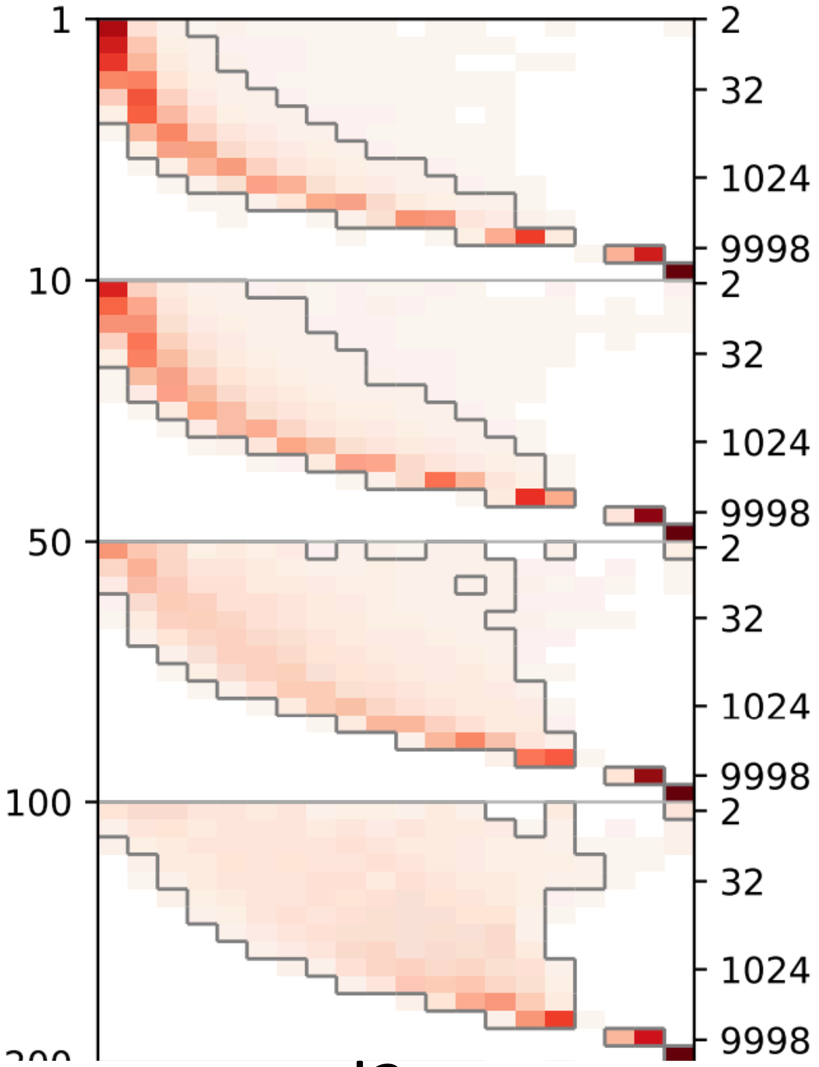


Burdick et al. (2018)

GloVe vs. word2vec: w2v much less stable!



GloVe



word2vec

Burdick et al. (2018)

What can go wrong with word embeddings?

- What's wrong with learning a word's "meaning" from its usage?

Ukraine's deputy defense minister resigns amid corruption allegations

Some claim Covid was hoax

What do we mean by bias?

- Identify *she* - *he* axis in word vector space, project words onto this axis

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

Bolukbasi et al. (2016)

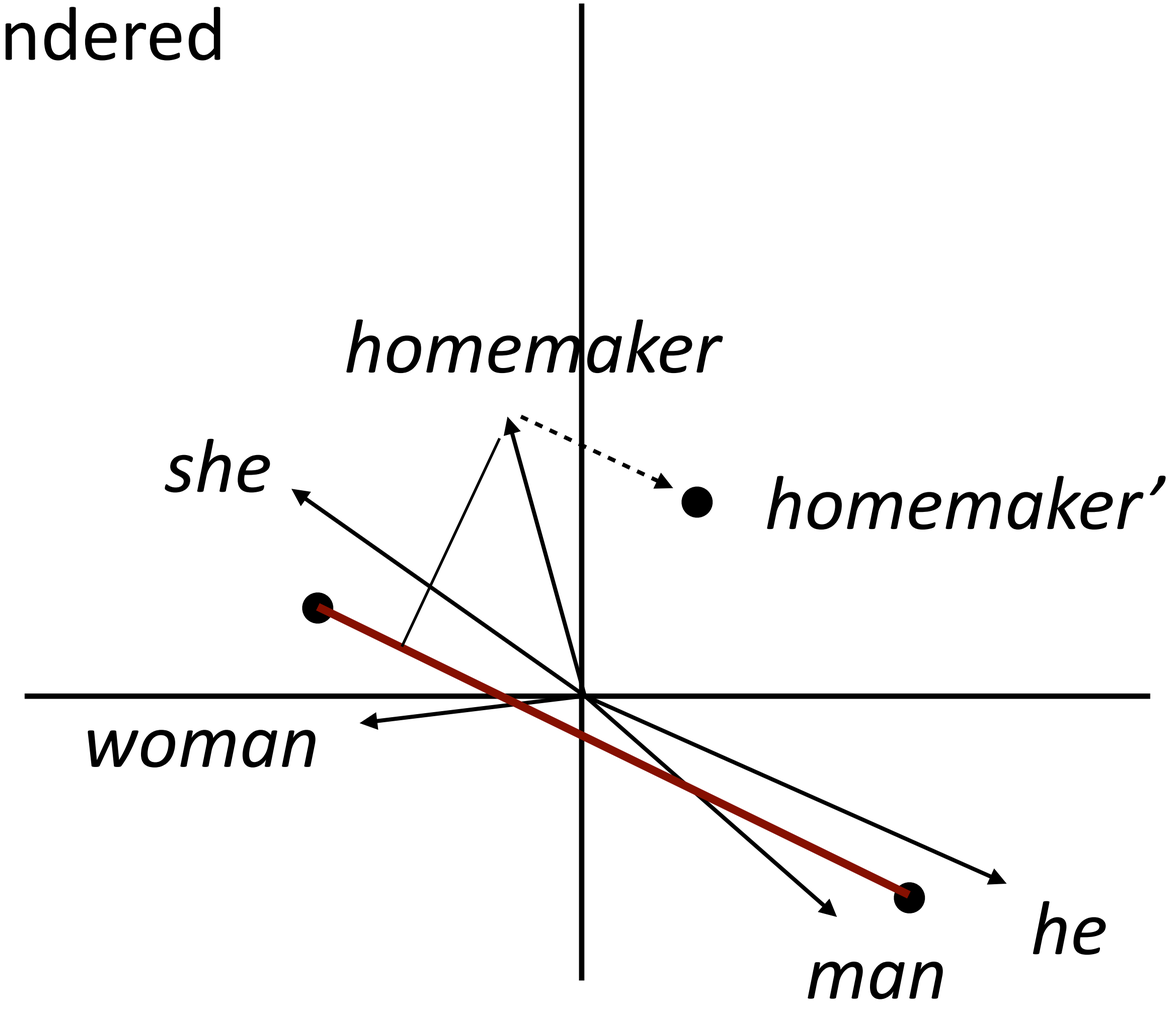
- Nearest neighbor of (b - a + c)

Racial Analogies	
black → homeless	caucasian → servicemen
caucasian → hillbilly	asian → suburban
asian → laborer	black → landowner
Religious Analogies	
jew → greedy	muslim → powerless
christian → familial	muslim → warzone
muslim → uneducated	christian → intellectually

Manzini et al. (2019)

Debiasing

- Identify gender subspace with gendered words
- Project words onto this subspace
- Subtract those projections from the original word

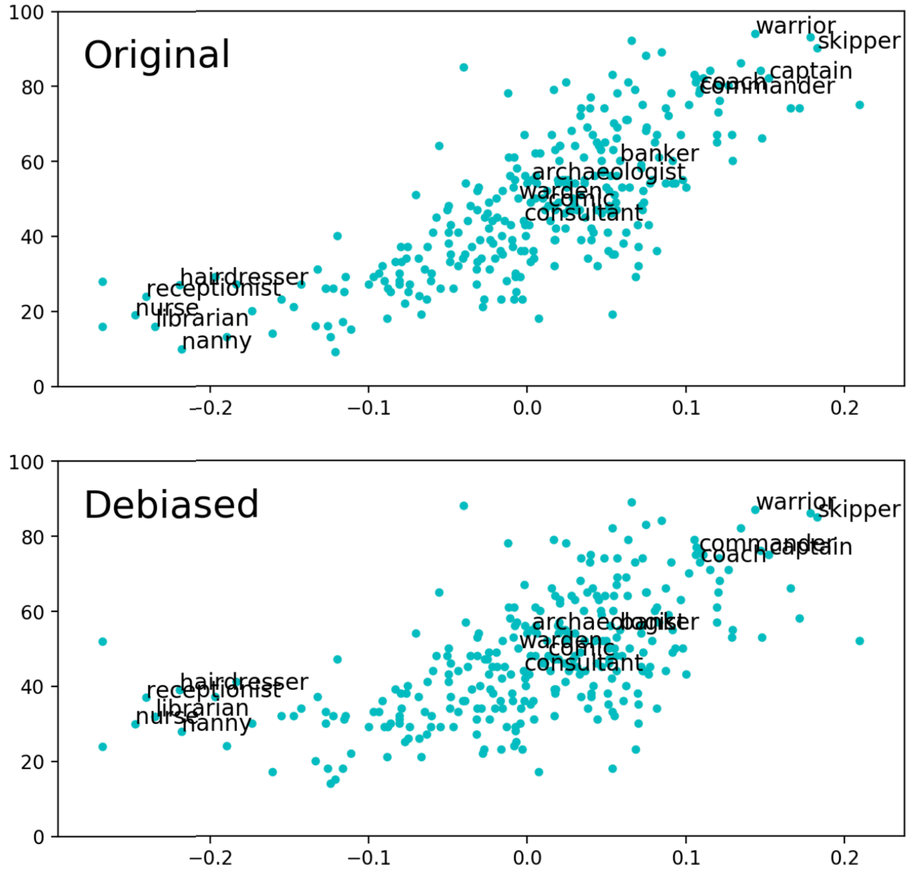


Bolukbasi et al. (2016)

Hardness of Debiasing

- Not that effective…and the male and female words are still clustered together

- Bias pervades the word embedding space and isn't just a local property of a few words



(a) The plots for HARD-DEBIASED embedding, before (top) and after (bottom) debiasing.

Gonen and Goldberg (2019)

Takeaways

- Word vectors: learning word -> context mappings has given way to matrix factorization approaches (constant in dataset size)
- Lots of pretrained embeddings work well in practice, they capture some desirable properties
- Even better: context-sensitive word embeddings (ELMo)
- Next time: language modeling and Transformers