Vector Semantics

Embeddings inspired by neural language models: word2vec

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

Word2Vec: Skip-Gram Task

Word2vec provides a variety of options. Let's do

"skip-gram with negative sampling" (SGNS)

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

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

Skip-Gram Training Data

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

- *I would like to go someplace nearby for lunch*

  `[target]`

The training data:

- input/output pairs (centering on *go)*

Skip-Gram Training data

I would like to go someplace nearby for lunch

Positive

{target, context}

- *{go, like}*
- *{go, to}*
- *{go, someplace}*
- *{go, nearby}*

Skip-Gram Training data

I would like to go someplace nearby for lunch

Positive

{target, context}

- *{go, like}*
- *{go, to}*
- *{go, someplace}*
- *{go, nearby}*

That's fine for positive data. But for training a binary classifier we need negative examples.

Let's sample other words from the lexicon (that don't occur with the target word in this context).

Skip-Gram Training data

I would like to go someplace nearby for lunch

Positive

{+ target, context}

{go, like}
{go, to}
{go, someplace}
{go, nearby}

Negative

{- target, context}

{go, aardvark}
{go, incubate}
{go, twelve}
{go, therefore}

Setup

Let's represent words as vectors of some length (say 300), randomly initialized.

So we start with 300 * V random parameters

Over the entire training set, we'd like to adjust those word vectors such that we

- 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.

The classifier's goal

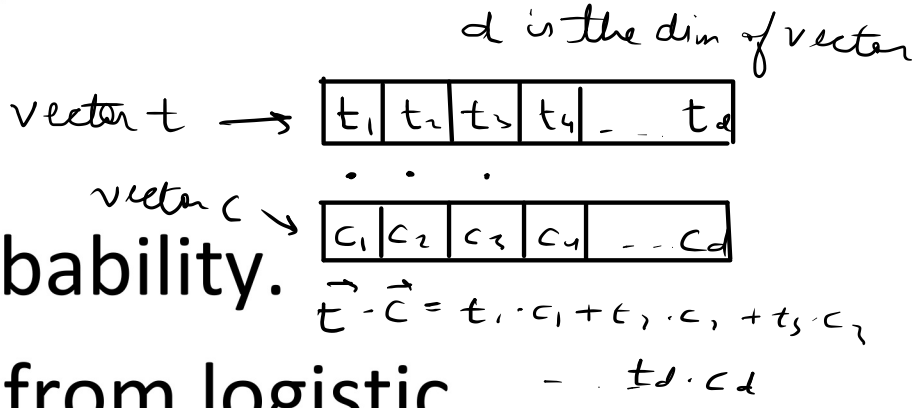
- Compute the probability that c is a real context word and not a fake noise word
- P(real|t,c)
- P(fake|t,c)

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(t,c) $\propto t \cdot c$
- We'll need to normalize to get a probability
  - (cosine isn't a probability either)

Turning dot products into probabilities

- $\text{Sim}(t,c) = t \cdot c$
- To turn this into a probability.
- We'll use the sigmoid from logistic regression:

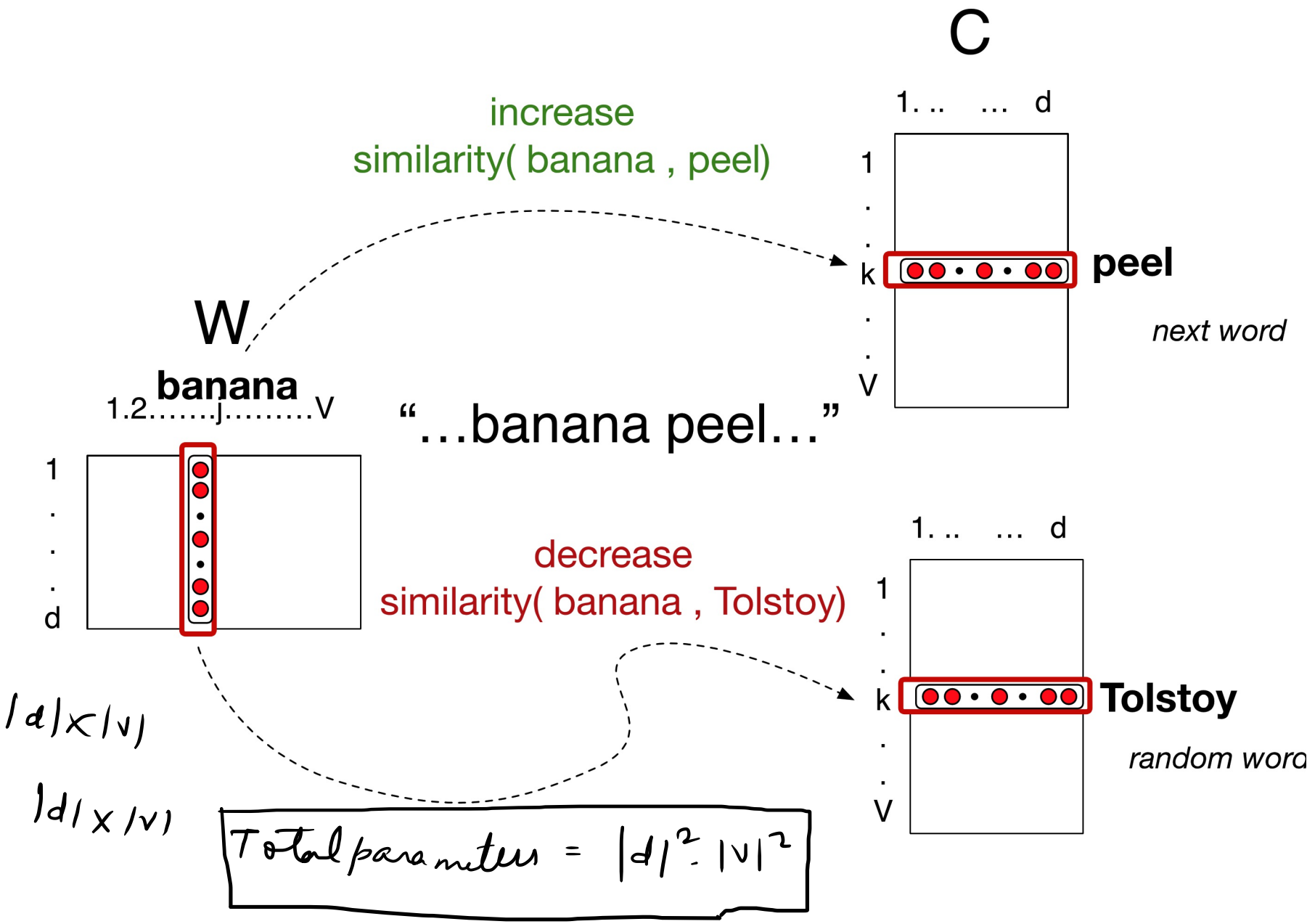


$$P(+|t,c) = \frac{1}{1+e^{-sim(t,c)}} = \frac{1}{1+e^{-(t_1 \cdot c_1 + t_2 \cdot c_2 + t_3 \cdot c_3 \cdots t_d v_d)}}$$

$$P(-|t,c) = 1 - P(+|t,c) = \frac{e^{-\text{Sim}(t,c)}}{1+e^{-\text{Sim}(t,c)}}$$

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.


C
increase
similarity(banana , peel)
1. .. ... d
1
k
V
peel
next word
W
banana
1.2.....j.....V
"…banana peel…"
1
d
decrease
similarity(banana , Tolstoy)
Tolstoy
random word
|d|x|V|
|d|x|V|
Total parameters = |d|² . |V|²

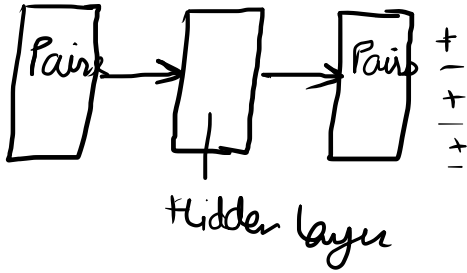
t_1^i — ith word of corpus
t_1 → dim

Objective Criteria

$L(t_1^1, t_1^2, t_1^3 \ldots t_d^1 \ldots t_d^V, c_1^1, c_1^2, c_1^3 \ldots c_d^1 \ldots c_d^V)$

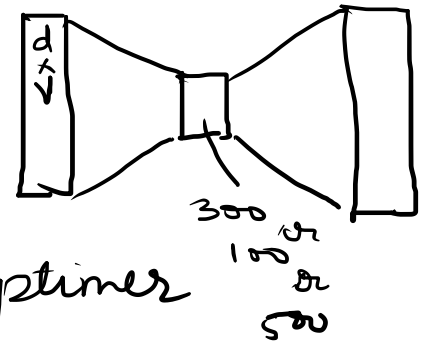
We want to maximize…

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



Maximize

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



Train using stochastic gradient descent

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

- Start with V random 300-dimensional vectors as initial embeddings
- Use logistic regression:
  - 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.

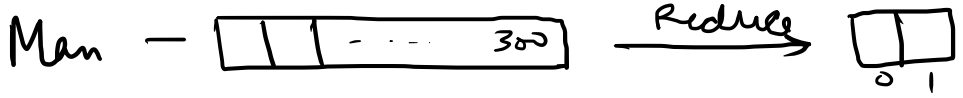
Or in other words

- Start with some initial embeddings (e.g., random)
- iteratively make the embeddings for a word
  - more like the embeddings of its neighbors
  - less like the embeddings of other words.

Properties of embeddings: Word similarity/relatedness!

Nearest words to some embeddings (Mikolov et al. 2013)

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating



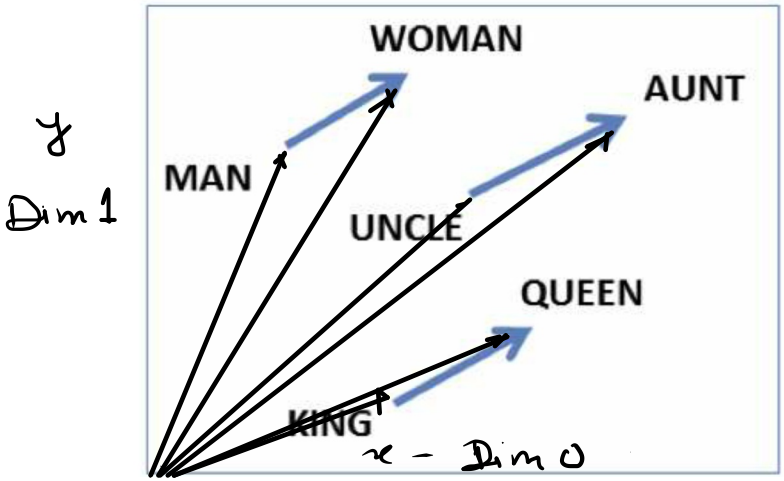
Analogies!

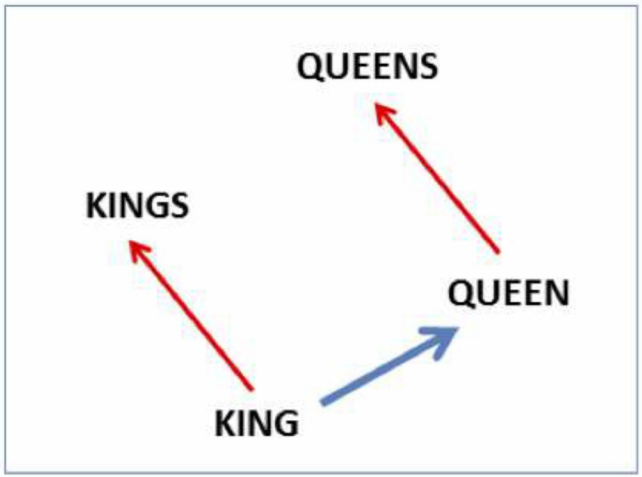
Embeddings capture relational meaning!

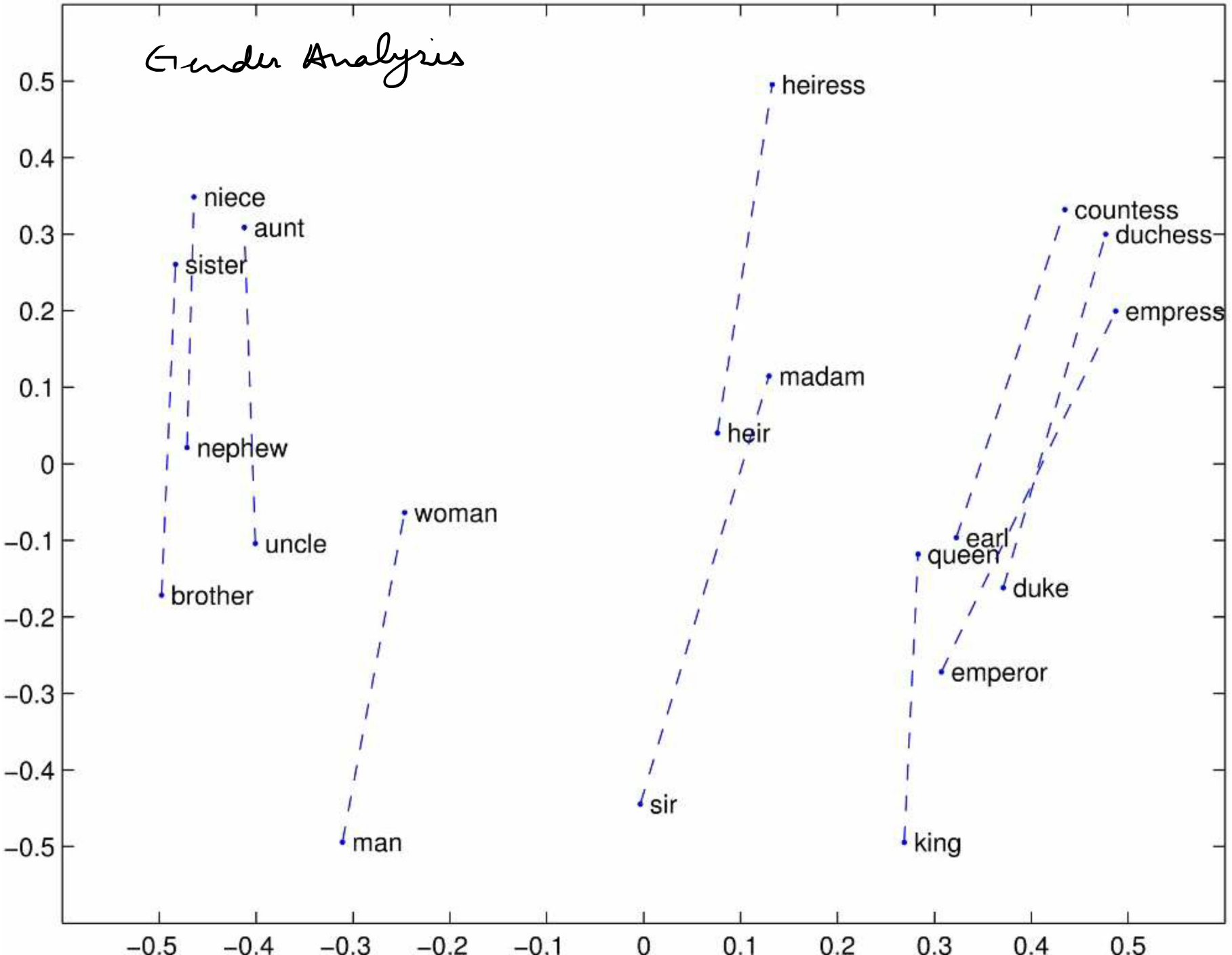
vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

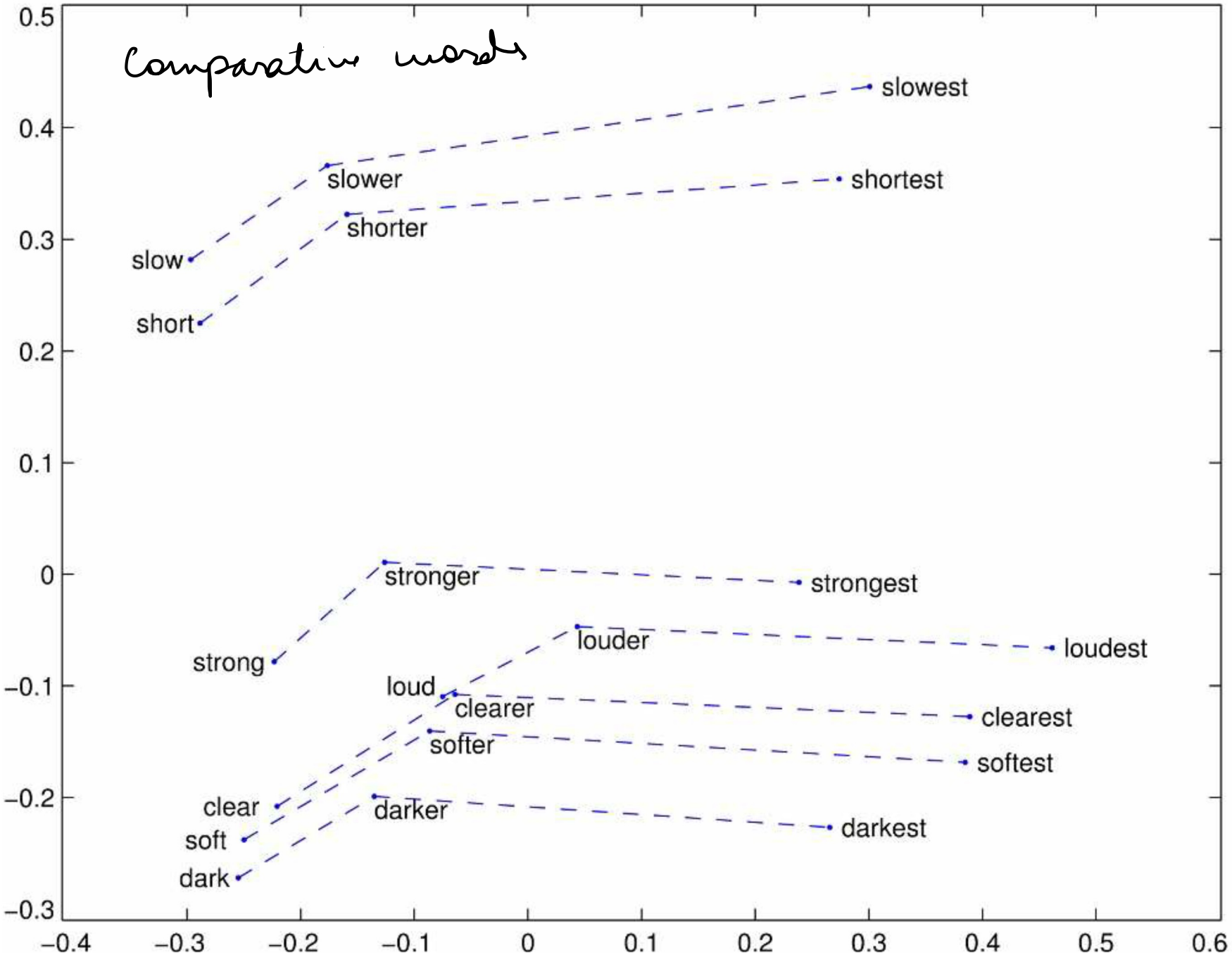
vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')

- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- t-SNE






Gender Analysis
heiress
niece
aunt
sister
countess
duchess
empress
madam
heir
nephew
woman
uncle
earl
queen
duke
brother
emperor
sir
king
man
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


Comparative words
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 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

Two things we can do about this cultural bias problem

1. Find ways to debias word embeddings
2. Use the embeddings to **study** cultural bias!

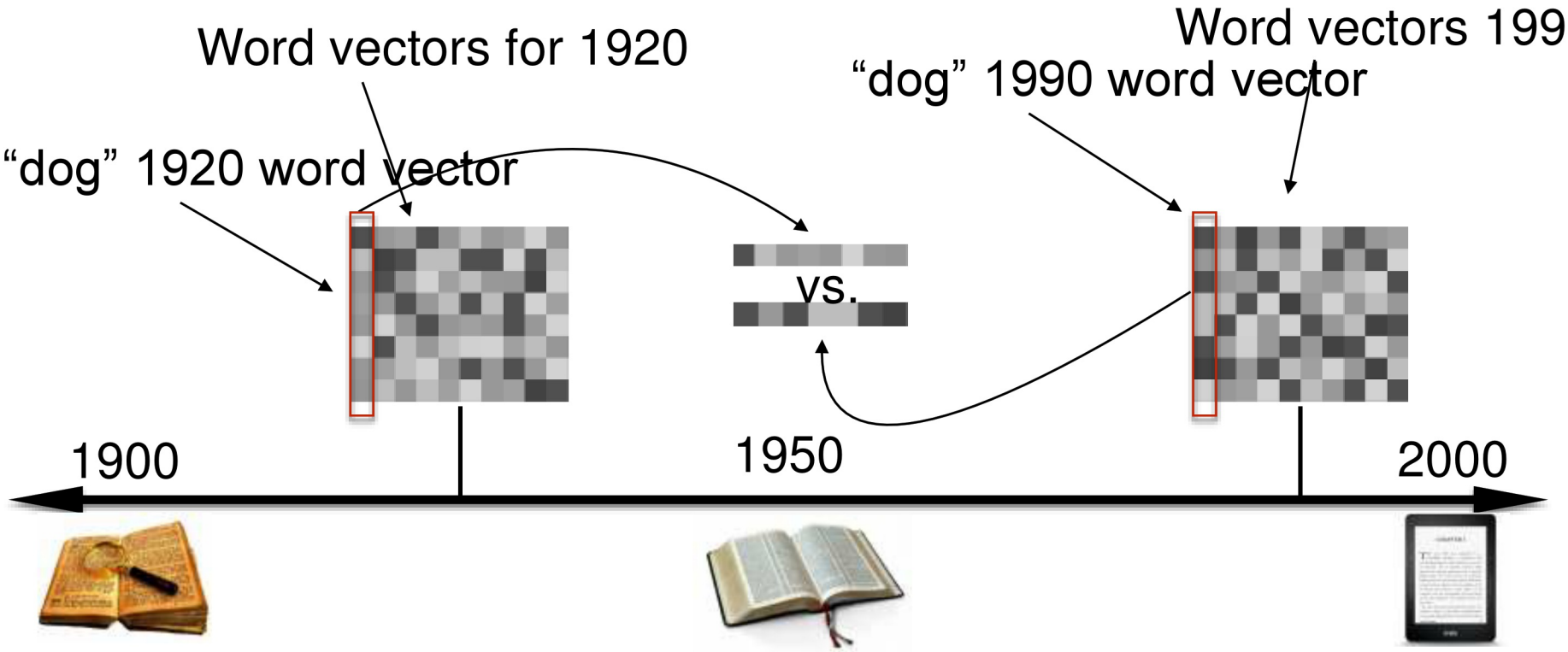
First: embeddings as a window on history

- William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. Proceedings of ACL 2016.
- William L. Hamilton, Jure Leskovec, Dan Jurafsky. 2016. Cultural Shift or Linguistic Drift? Comparing Two Computational Models of Semantic Change. Proceedings of EMNLP 2016.
- William L. Hamilton, Kevin Clark, Jure Leskovec, Dan Jurafsky. 2016. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. Proceedings of EMNLP 2016.

Will Hamilton

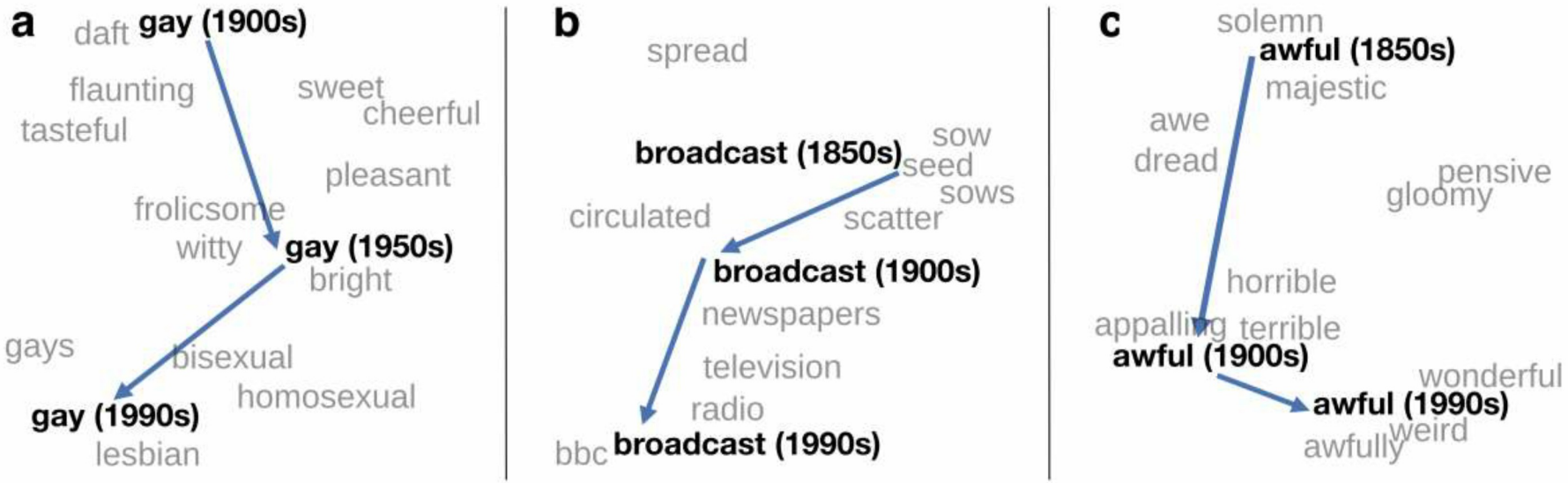
Jure Leskovec

Train embeddings on old books to study changes in word meaning



Visualizing changes

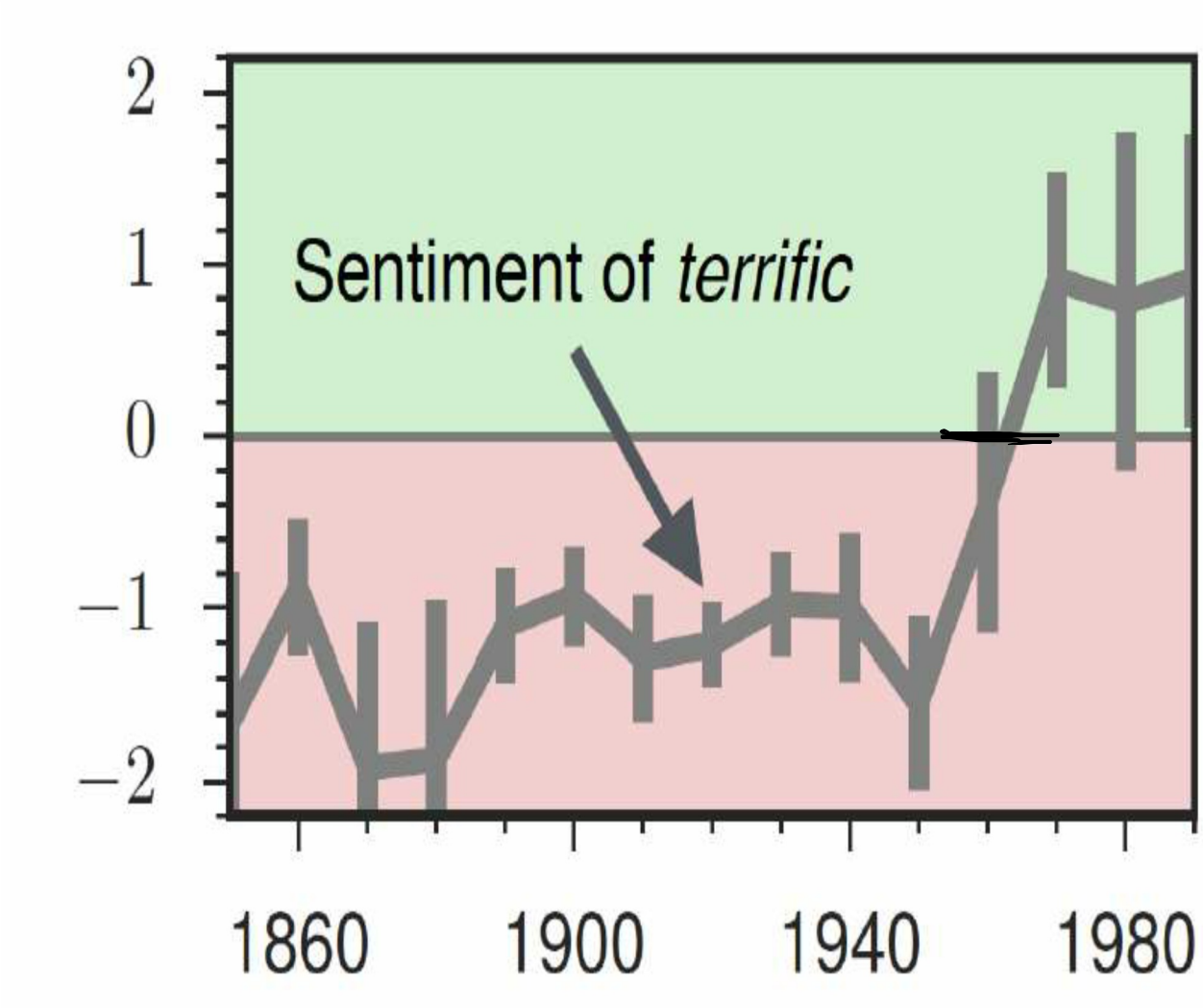
Project 300 dimensions down into 2



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

The evolution of sentiment words

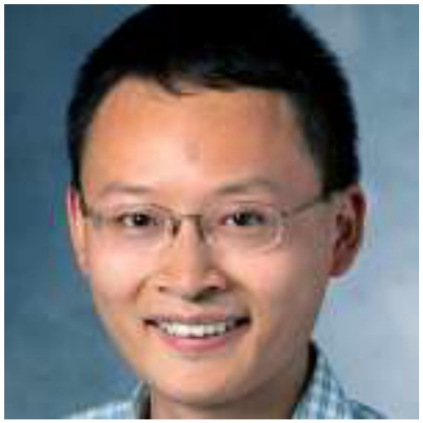
Negative words change faster than positive words



Historical embedding: a tool to investigate history of cultural biases

Nikhil Garg

Londa Schiebinger



James Zou

- Nikhil Garg, Londa Schiebinger, Dan Jurafsky, James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. Proceedings of the National Academy of Sciences 2018.

Now use our historical embeddings as a tool to investigate cultural biases

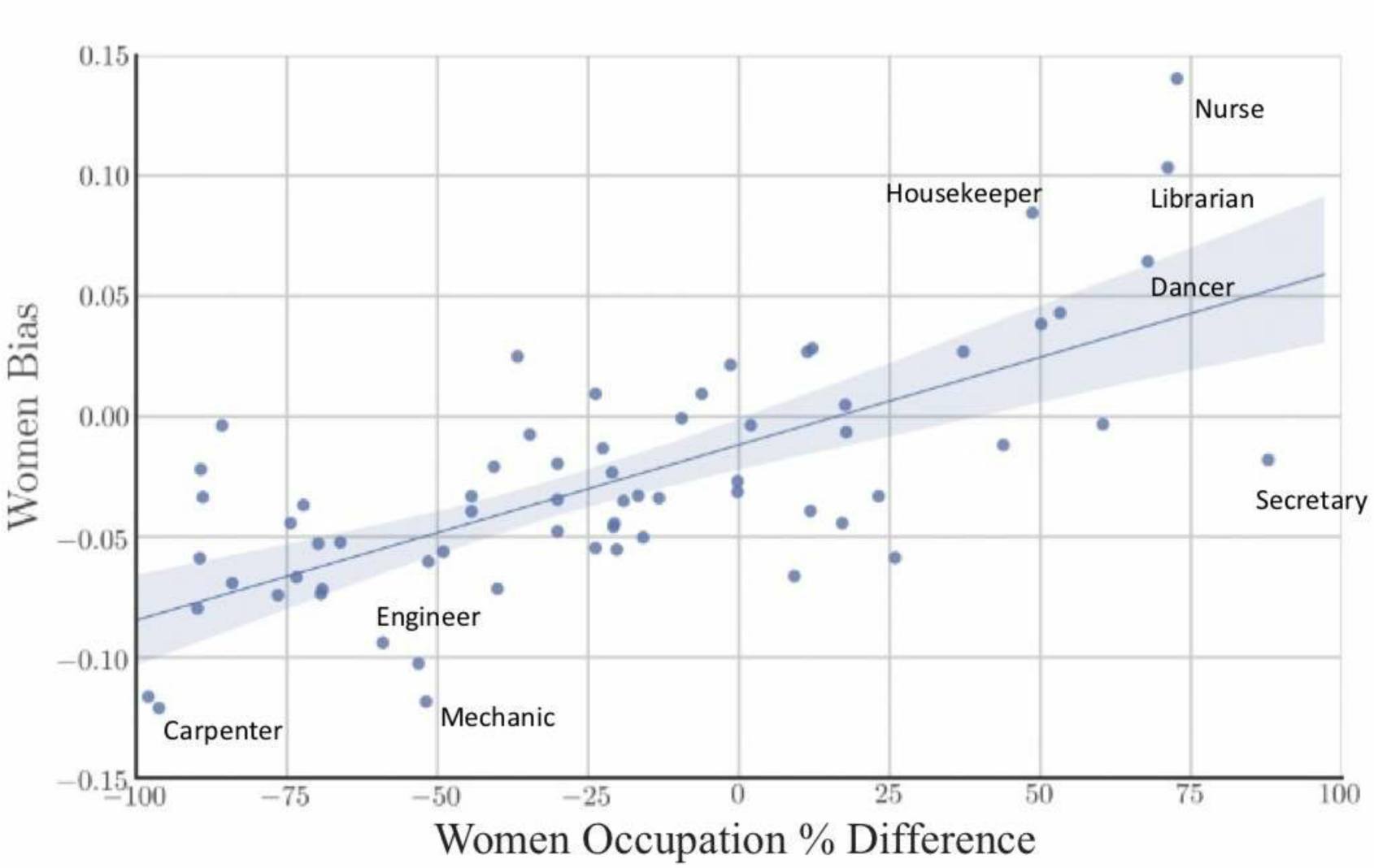
- From the historical embeddings
- Compute historical biases of words:

  Gender bias: how much closer a word is to "woman" synonyms than "man" synonyms.

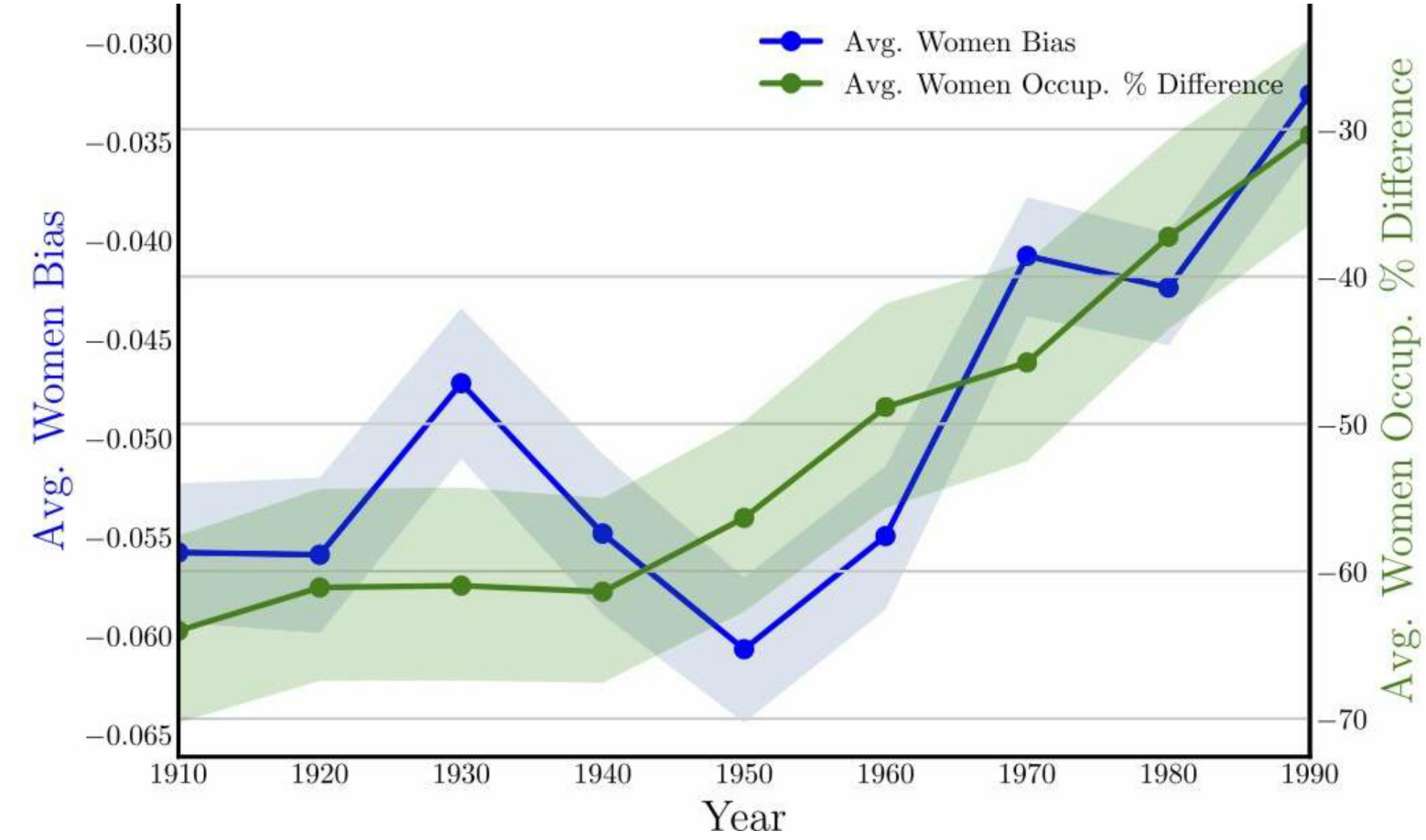
  Ethnic bias: how much closer a word is to last names of a given ethnicity than to names of Anglo ethnicity
- Correlate with occupational data from historical census
- Look at how all these change over time

Embedding bias correlates with actual occupation data

Is "nurse" closer to "man" than "woman"?



Embeddings reflects gender bias in occupations across time (1910-1990)



Embeddings also reflect **framings** of women over time

Embeddings for **competence** adjectives are biased toward men

- *Smart, wise, brilliant, intelligent, resourceful, thoughtful, logical, etc.*

This bias is slowly decreasing 1960-1990

Embeddings reflect ethnic stereotypes over time

- "Princeton trilogy" experiments
- Attitudes toward ethnic groups (1933, 1951, 1969) scores for adjectives
  - *industrious, superstitious, nationalistic*, etc
- Embedding association with Chinese ethnicity correlates with adjective scores and with the change 1933-1979
- In other words**: we can run social psychology experiments in the past!**

Changes in Framing: The most biased Asian (vs. White) adjectives over time

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

Conclusion

- **Concepts** or word senses
  - Have a complex many-to-many association with **words** (homonymy, multiple senses)
  - Have 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 count models (tf-idf/PPMI) or dense predict models (word2vec, GLoVE)
  - Useful in practice but also encode cultural stereotypes
    - Can **debias** embeddings, and use them to **study bias**