Economics 2355: Models of Words

Melissa Dell

Harvard University

March 2021

Outline

NLP Portion of Course

The course website has an updated syllabus for the NLP portion of the course:

Natural Language Processing	
Mar 10	Models of Words (word2vec, GLoVe)
Mar 15	Dependency Parsing; Language Models (N-Grams, RNN/LSTM review, GRU)
Mar 17	Sequence to Sequence Learning
Mar 22	The Transformer
Mar 24	Transformer-Based Models
Mar 29	What's in an Embedding; Sentiment Analysis
Mar 31	NO CLASS (Wellness Day)
Apr 5	Retrieval and Question Answering
Apr 7	Zero-Shot and Few-Shot Learning in NLP
Apr 12	Natural Language Generation; Summarization (time permitting)
Apr 14	NLP on Noisy Data

Course Projects

- The final two weeks of the course will be devoted to presenting the class projects
- Please start getting your hands dirty with coding and implementation now. This will provide time to adjust for any deficiencies in your coding background and adjust the scope of the project if it ends up (as it typically does) being more ambitious than initially anticipated

Outline

How Do We Create a Computable Measure of Word Meaning

- Traditional approach: use a dataset like WordNet, which is effectively a thesaurus containing lists of synonyms and hypernyms ("is a" relationships)
- These human engineered relationships require a lot of human labor to create and maintain
- May lack nuance or comprehensiveness

Localist Representation of Words

- In traditional NLP, words are treated as discrete objects
- These can be encoded as one hot vectors, where vector dimension is the number of words in the vocabulary
- No notion of similarity with one hot vectors, by definition they are orthogonal
- Suppose there are 500,000 words in the vocab and you want to encode a thesaurus using the one-hot vectors. This would be a 500,000 x 500,000 matrix
- How else could we represent words?

Representing Words by their Context

In just about every treatment of NLP, there is the following obligatory 1957 quote by linguist J.R. Firth:

> *You shall know a word by the company it keeps.*

If you can explain the correct context in which to use the word, then you understand the meaning of the word.

Word Embeddings

- A word embedding is a dense vector representing a word
- Words are embedded such that words that appear in similar contexts have similar word vectors
- This is an example of a *distributed* representation

Count Based Methods

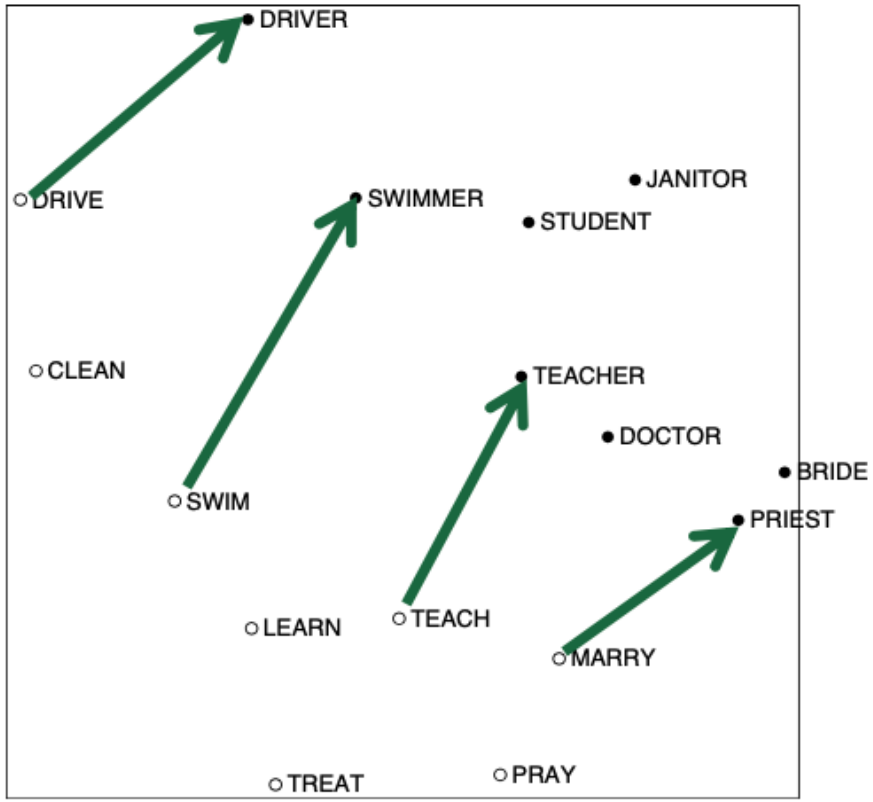
- These pre-dominated in the pre-deep learning era
- Consider words and their contexts - the words that appear in a nearby window
- Count based methods - also known as distributional models - create a word context matrix that count the number of co-occurrences between words and the surrounding words in its contexts
- Massive matrix: do dimensionality reduction with singular value decomposition

Count-Based Methods

- In practice, need a lot of hacks to make this work (Rohde et al., 2005)
  - log the frequencies
  - ignore common words
  - weight the windows
  - use Pearson correlations rather than counts and set negative values to zero, etc, ,etc, etc

Semantic Properties

When we implement these tricks, vectors appear to have some meaningfully properties:

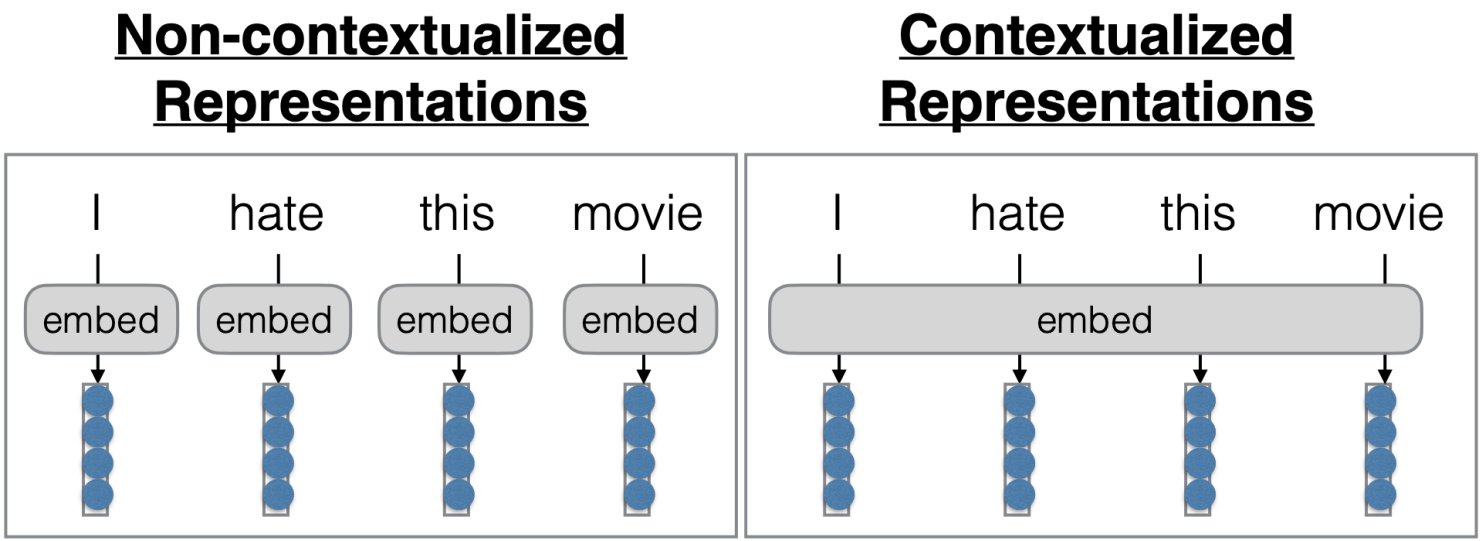


Rohde, 2005

The model was primarily used to capture word similarity.

Outline

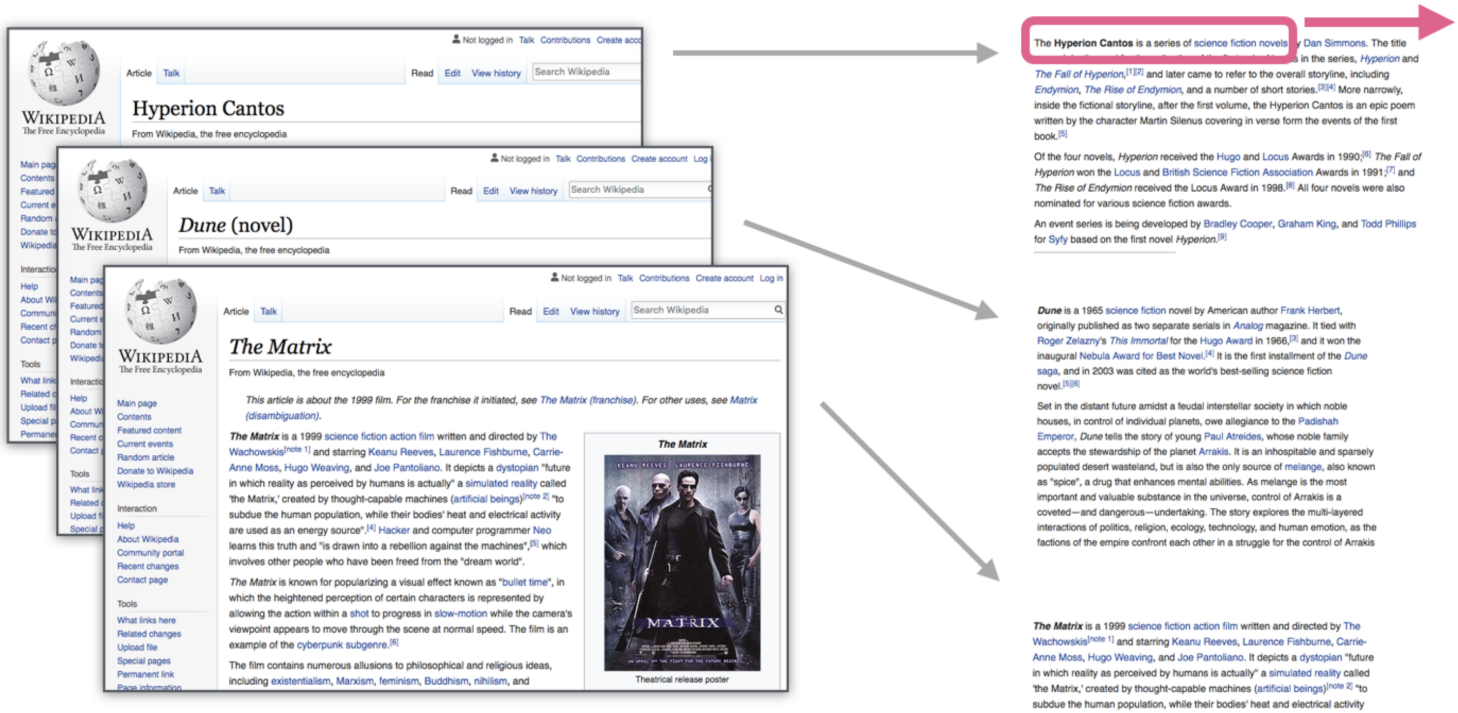
Contextualization of Word Representations



Graham Neugib CS11-747

Word embeddings are non-contextualized.

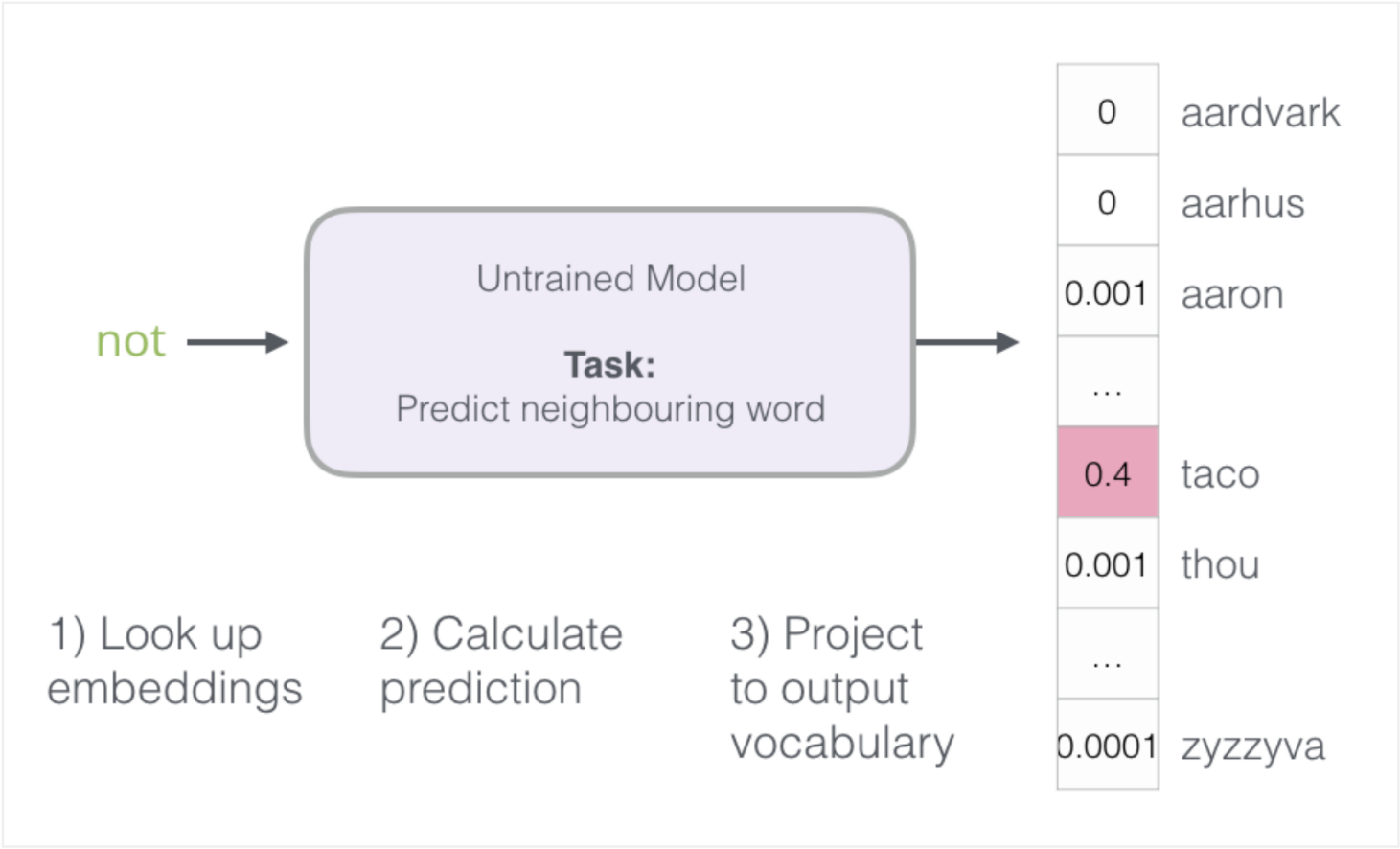
Data for Training Word Embeddings



https://jalammar.github.io/illustrated-word2vec/

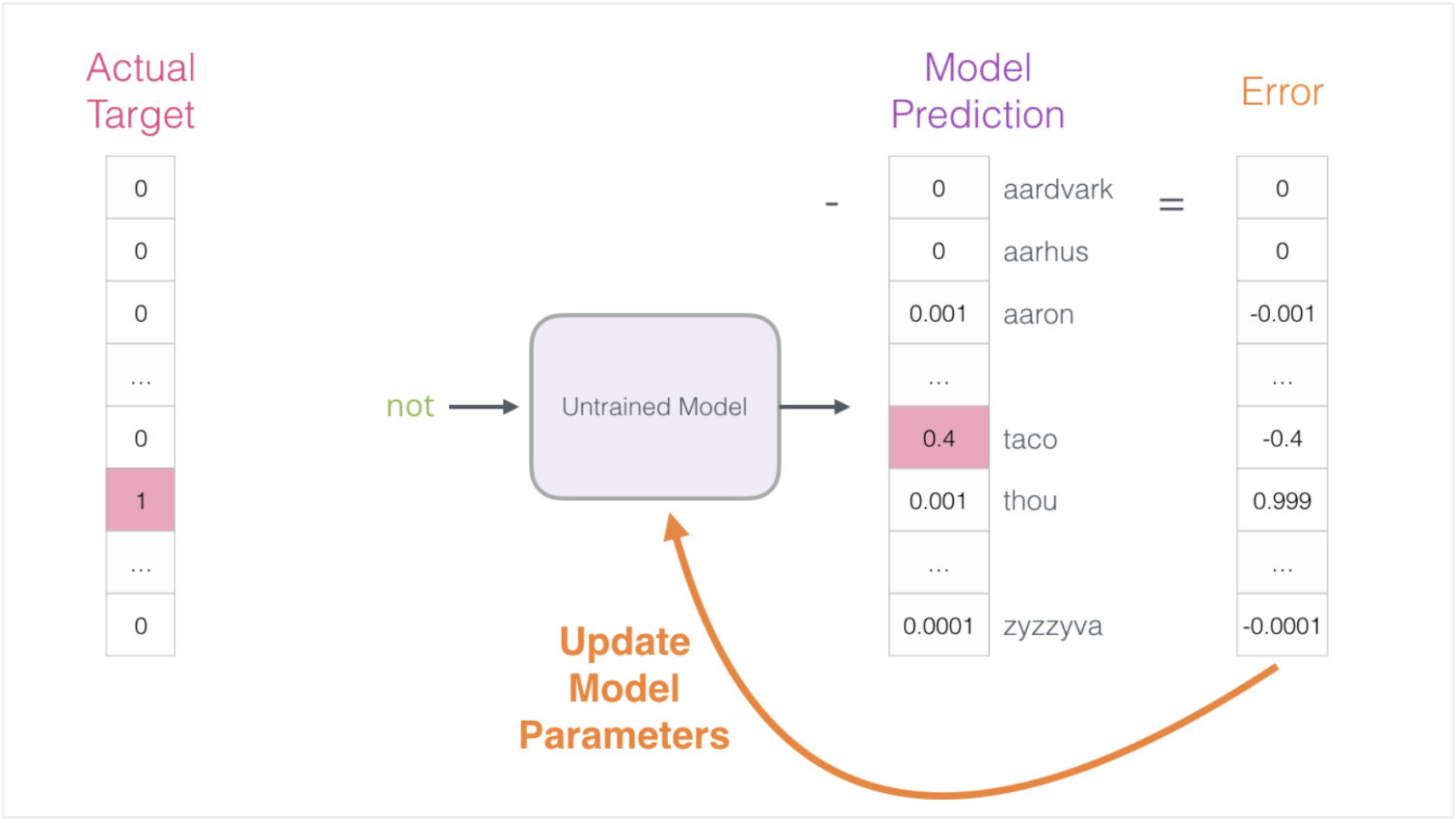
Word2Vec Intuition (Mikolov et al. 2013)

Predict words in context (output) using center (input) word



https://jalammar.github.io/illustrated-word2vec/

Word2Vec Intuition (Mikolov et al. 2013)



https://jalammar.github.io/illustrated-word2vec/

Word2Vec Objective Function (Mikolov et al. 2013)

For each position $t = 1 \ldots T$ predict context words within a window of fixed size m, given center word w_j.

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P(w_{t+j} | w_t; \theta) \quad (1)$$

Want to minimize:

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} logP(w_{t+j} | w_t; \theta) \quad (2)$$

Word2Vec Objective Function

How do we calculate $P(w_{t+j}|w_t;\theta)$?

Use two vectors per word w: v_w when w is the center word and u_w when w is a context word

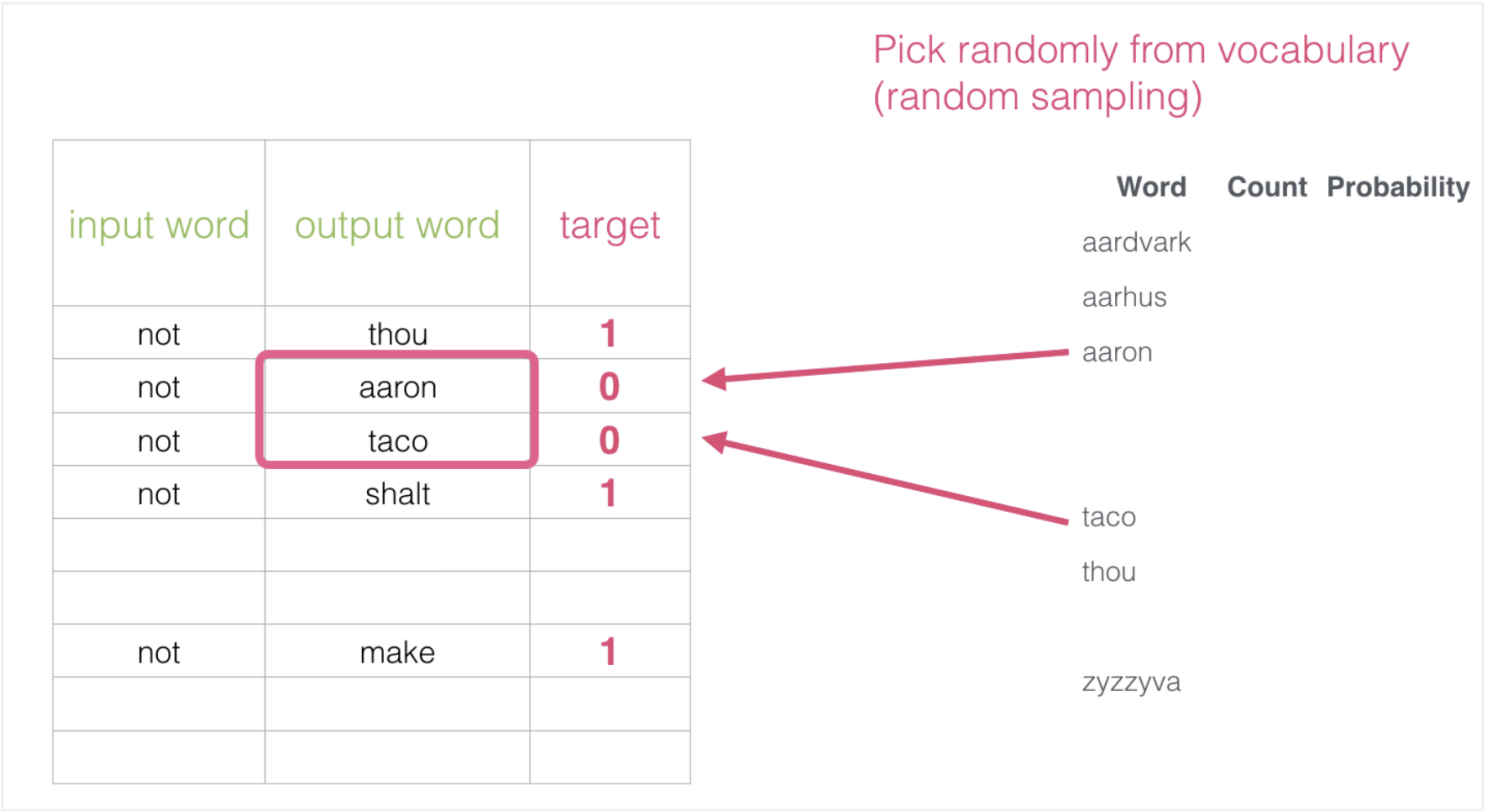
For a center word c and context (output) word o:

$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w \in V} exp(u_w^T v_c)}$$

Dot product compares similarity of o and c. Larger dot product implies more similarity: a higher probability that used together. We normalize over the entire vocabulary to get a probability distribution. Recall from earlier in the class that this is just an example of a softmax function.

Negative Sampling

Calculating the denominator of the softmax requires computing similarity with every word in the (potentially very large) vocabulary. Infeasible to do for every training example. Do something called negative sampling instead:



https://jalammar.github.io/illustrated-word2vec/

Noise-Contrastive Estimation

- Create positive samples (words that appear in the context) and negative samples (words sampled at random from the vocabulary).
- This effectively creates a logistic regression model (replaces softmax loss with sigmoid) that predicts whether the words are neighbors or not.

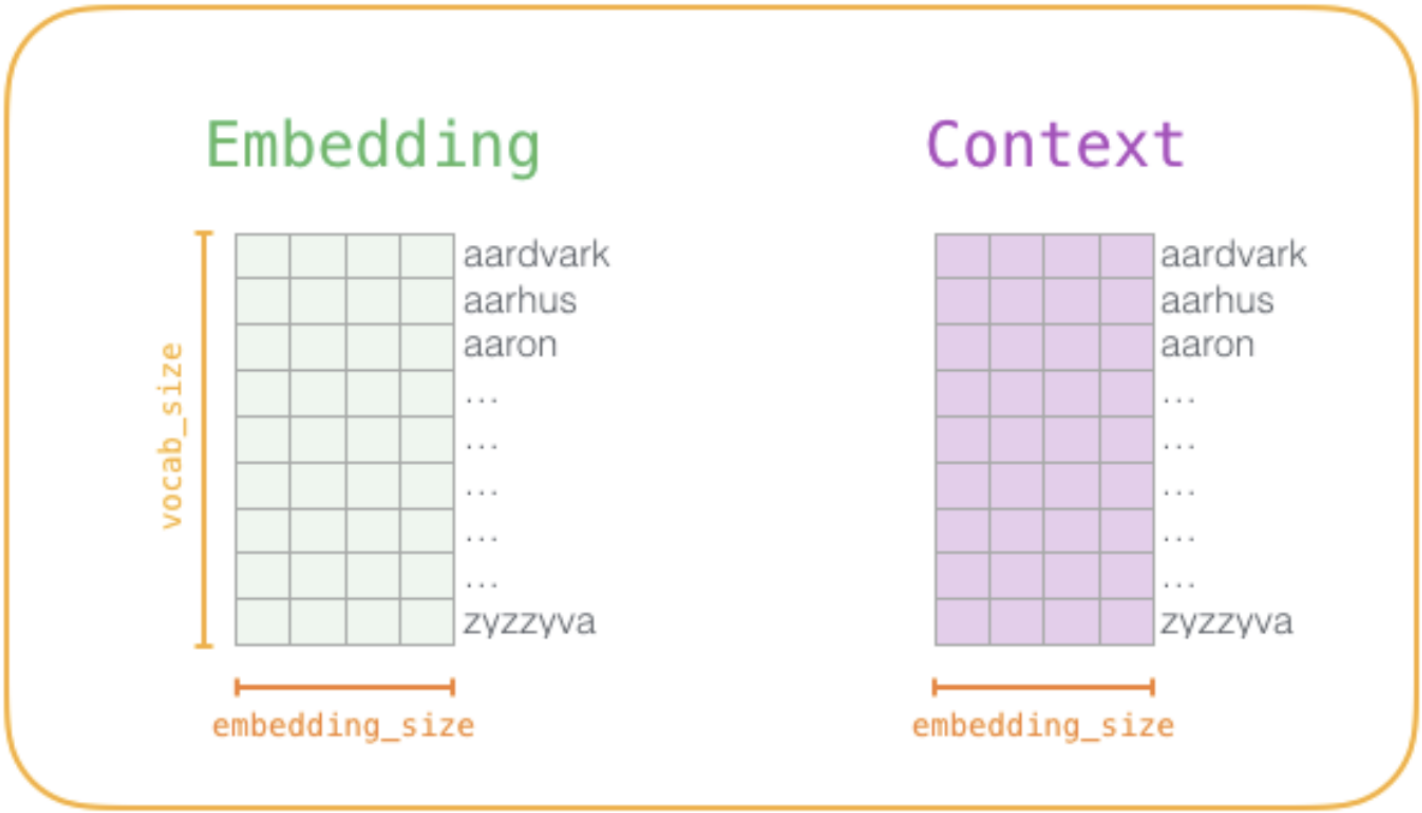
Training Word2Vec

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} logP(w_{t+j}|w_t;\theta)$$

- Our goal is to move word vectors around so good at predicting what words occur in the context of other words
- In other words, we maximize the objective by assigning similar embeddings to words that tend to be used in the same context

Training Word2Vec

Create a matrix for embeddings for the target word and a matrix for embeddings for contexts



https://jalammar.github.io/illustrated-word2vec/

At the start of training, it is randomly initialized

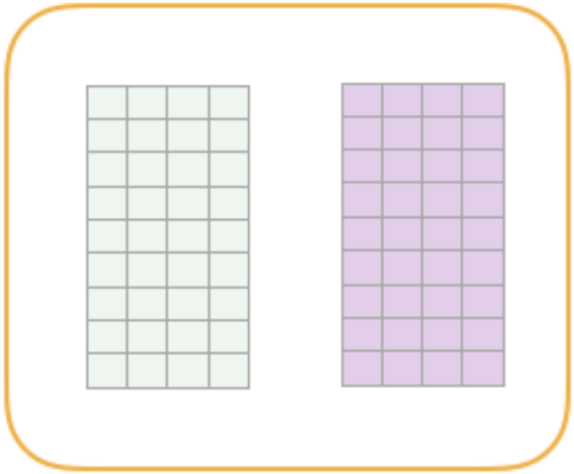
Training Word2Vec

At each training step, take a positive example and sample some negative examples.

dataset

input word	output word	target
not	thou	1
not	aaron	0
not	taco	0
not	shalt	1
not	mango	0
not	finglonger	0
not	make	1
not	plumbus	0
...	...	...

model

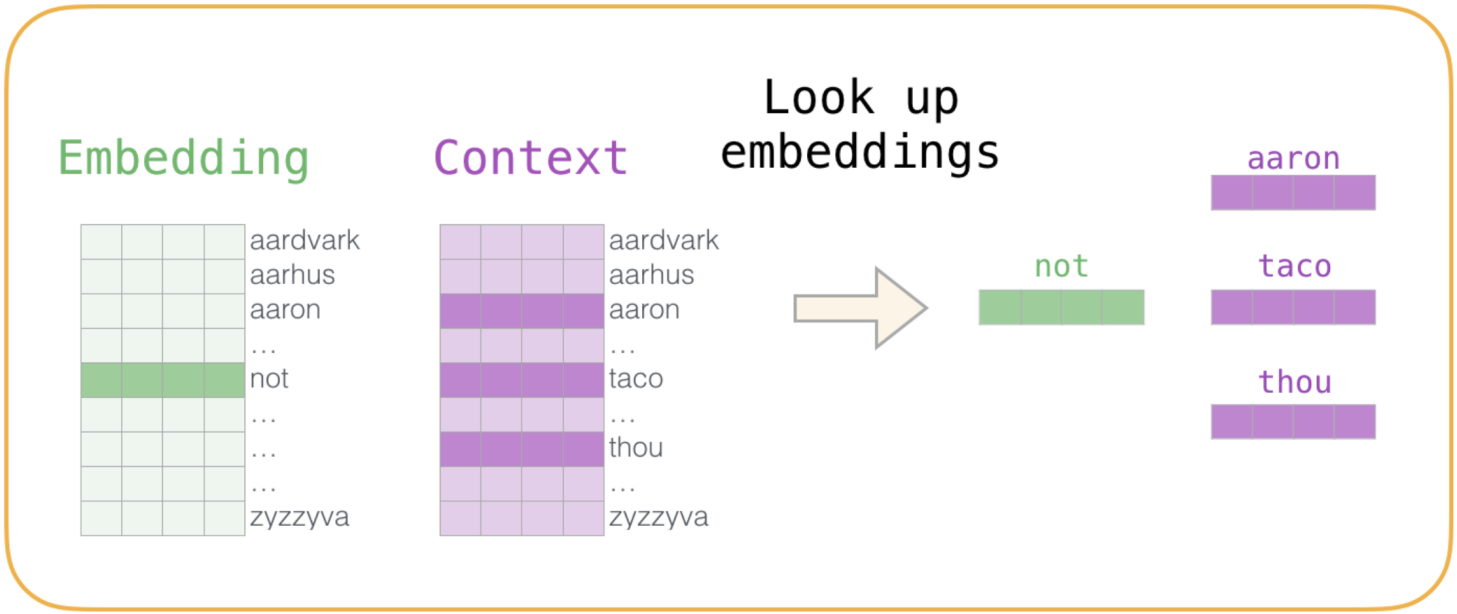


https://jalammar.github.io/illustrated-word2vec/

In this example, we now have four words: the input word *not* and context words: *thou* (the actual neighbor), *aaron*, and *taco* (the negative examples).

Training Word2Vec

We look up the input word in the embedding matrix and the context words in the context matrix.



https://jalammar.github.io/illustrated-word2vec/

Training Word2Vec

Take the dot product of the embedding vector with each of the context vectors. The result number measures the similarity of the two vectors.

input word	output word	target	input • output
not	thou	1	0.2
not	aaron	0	-1.11
not	taco	0	0.74

https://jalammar.github.io/illustrated-word2vec/

Training Word2Vec

Pass it through the sigmoid to get probabilities

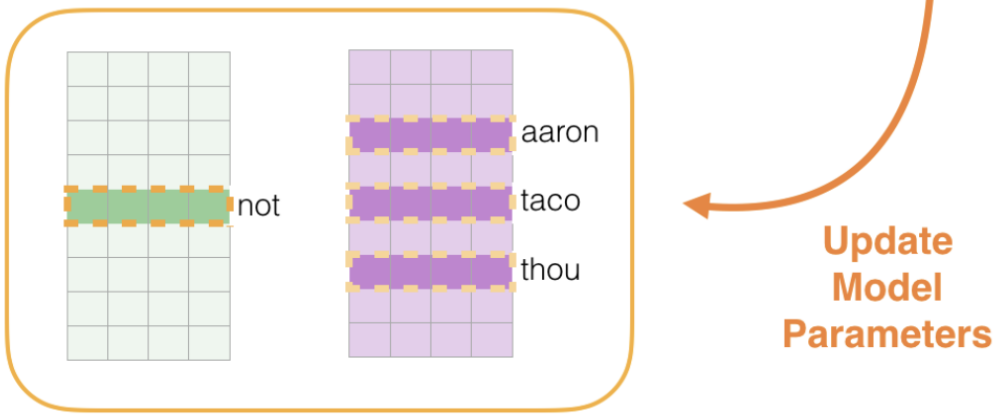
input word	output word	target	input • output	sigmoid()
not	thou	1	0.2	0.55
not	aaron	0	-1.11	0.25
not	taco	0	0.74	0.68

https://jalammar.github.io/illustrated-word2vec/

Training Word2Vec

Compute the gradient of the loss and update

input word	output word	target	input • output	sigmoid()	Error
not	thou	1	0.2	0.55	0.45
not	aaron	0	-1.11	0.25	-0.25
not	taco	0	0.74	0.68	-0.68



https://jalammar.github.io/illustrated-word2vec/

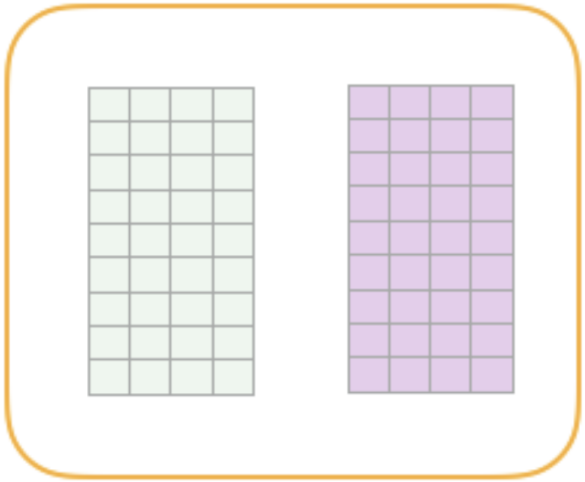
Training Word2Vec

Now move on to the next positive sample and its negative samples

dataset

input word	output word	target
not	thou	1
not	aaron	0
not	taco	0
not	shalt	1
not	mango	0
not	finglonger	0
not	make	1
not	plumbus	0
...	...	...

model



https://jalammar.github.io/illustrated-word2vec/

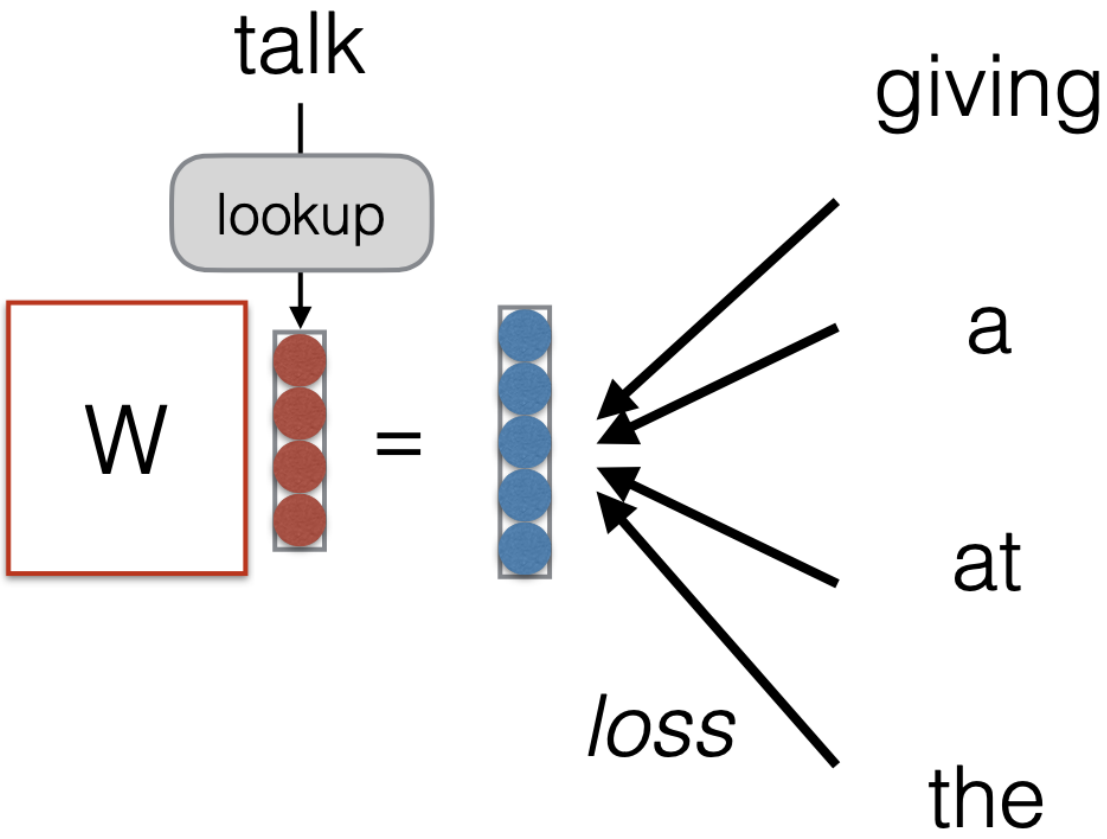
Variants

The Word2Vec paper introduces two variants of the model:

- Skip-Gram
- Continuous Bag of Words

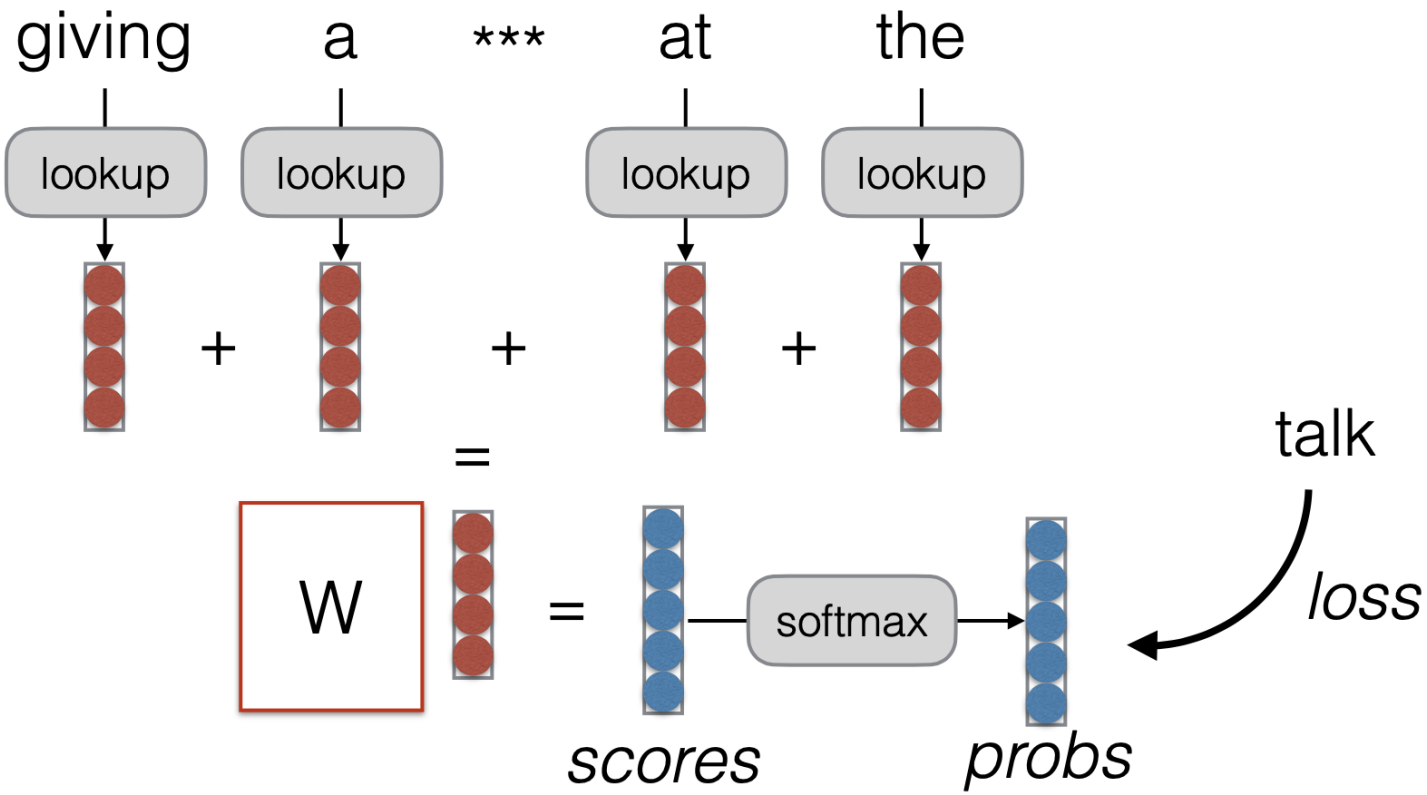
Skip-Gram

Predict each word in the surrounding context of the center work (Mikolov et al., 2013)



Continuous Bag of Words

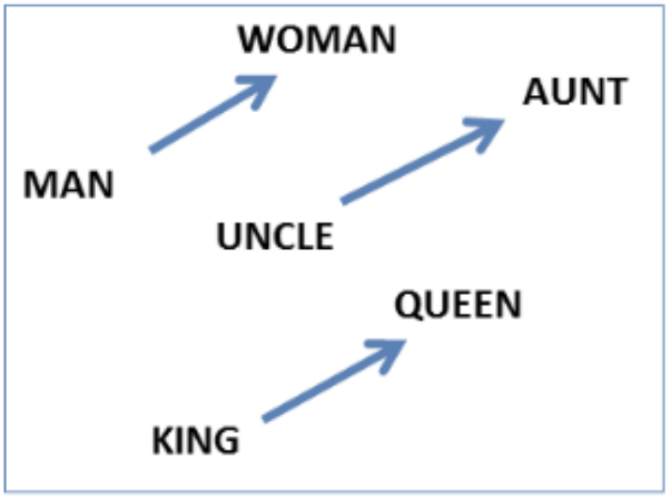
Predict center word based on sum of surrounding embeddings (Mikolov et al., 2013)

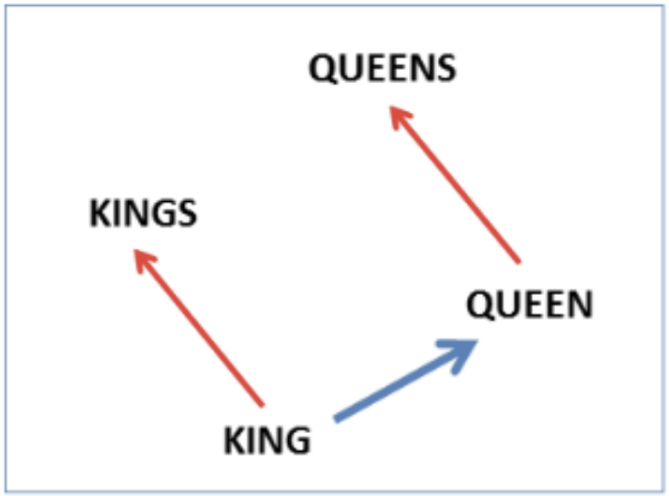


Graham Neugib CS11-747

The Power of Word2Vec

Word embeddings have some pretty remarkable properties, which served as the impetus for machine learning methods to revolutionize NLP. This is the most famous example.





Mikolov et al., 2013

Analogies

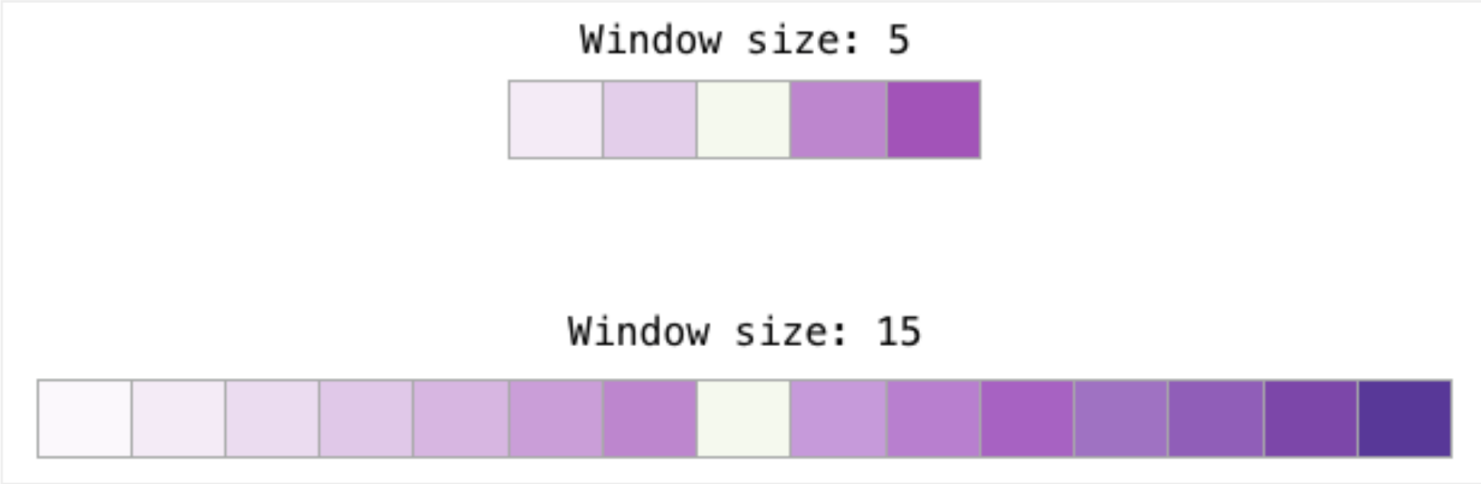
People use these to do lots of analogy tests. i.e. compute vector difference between woman and man, and it to king, what word is closest in vector space? Examples from a package called Gensim (Stanford CS 224n).

- Australia is to beer as France is to champagne
- Good is to fantastic as bad is to terrible
- Obama is to Clinton as Reagan is to Nixon

In high dimensional vector space, a word can be close to lots of other words in many different directions; can capture many dimensions of similarity

Selecting the Window Size

The window size you want depends on what relationships you hope to capture



https://jalammar.github.io/illustrated-word2vec/

Selecting the Window Size

- **Small Windows**: good for indicating words are interchangeable (note that antonyms are often interchangeable if look at only the surrounding words; i.e. “This class is awesome/horrible”). Smaller windows will tend to extract more syntax based information (i.e. generic words indicate the input word is a noun).
- **Larger Windows**: context has a larger effect, and embeddings are more likely to measure similarity in terms of relatedness of words.

FastText

- Word embedding toolkit: `https://github.com/facebookresearch/fastText/`
- Fast implementation for training
- Pre-trained embeddings on Wikipedia for many different languages

Outline

GloVe (Pennington et al., 2014)

Recall that we've discussed two different methods:

- Count methods using the co-occurrence matrix
- ML-based Word2Vec

GloVe aims to fuse these two methods by using ML on the co-occurrence matrix, to produce a co-occurrence based model that can maintain linear relationships in vector space (i.e. king - man + woman $->$ queen or better-good+bad $->$ worse)

GloVE

- The training objective of GloVe is to learn word vectors such that their dot product equals the log of the words' probability of co-occurrence
- Because the log of a ratio equals the difference of logs, this objective associates the log of ratios of co-occurrence probabilities with vector differences in the word vector space
- Since co-occurrence ratios encode meaning, so do vector differences
- Hence, the model can do well on analogy tasks (king-man+woman $-$ >queen)

Crucial Insight of GloVe

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

Pennington, Socher, and Manning, EMNLP 2014

Ratios of co-occurence probabilities can encode meaning

GloVe Results

Nearest words to frog:

1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

litoria

leptodactylidae

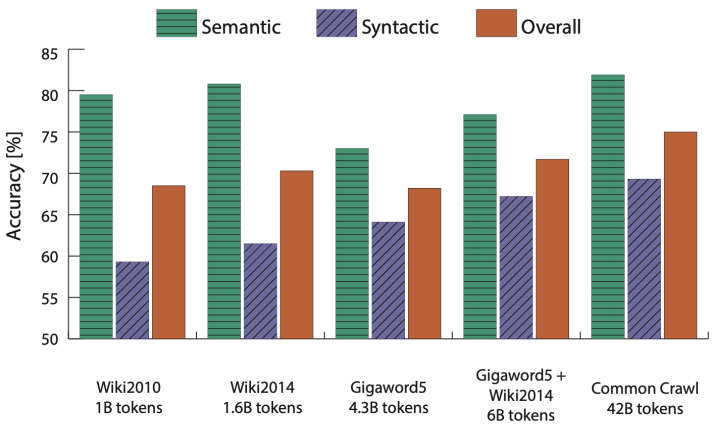
rana

eleutherodactylus

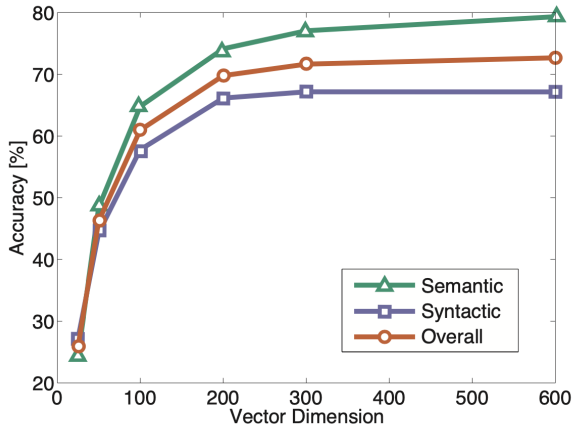
Pennington, Socher, and Manning, EMNLP 2014

GloVe Training Data

- More data helps
- Wikipedia is better than news text!

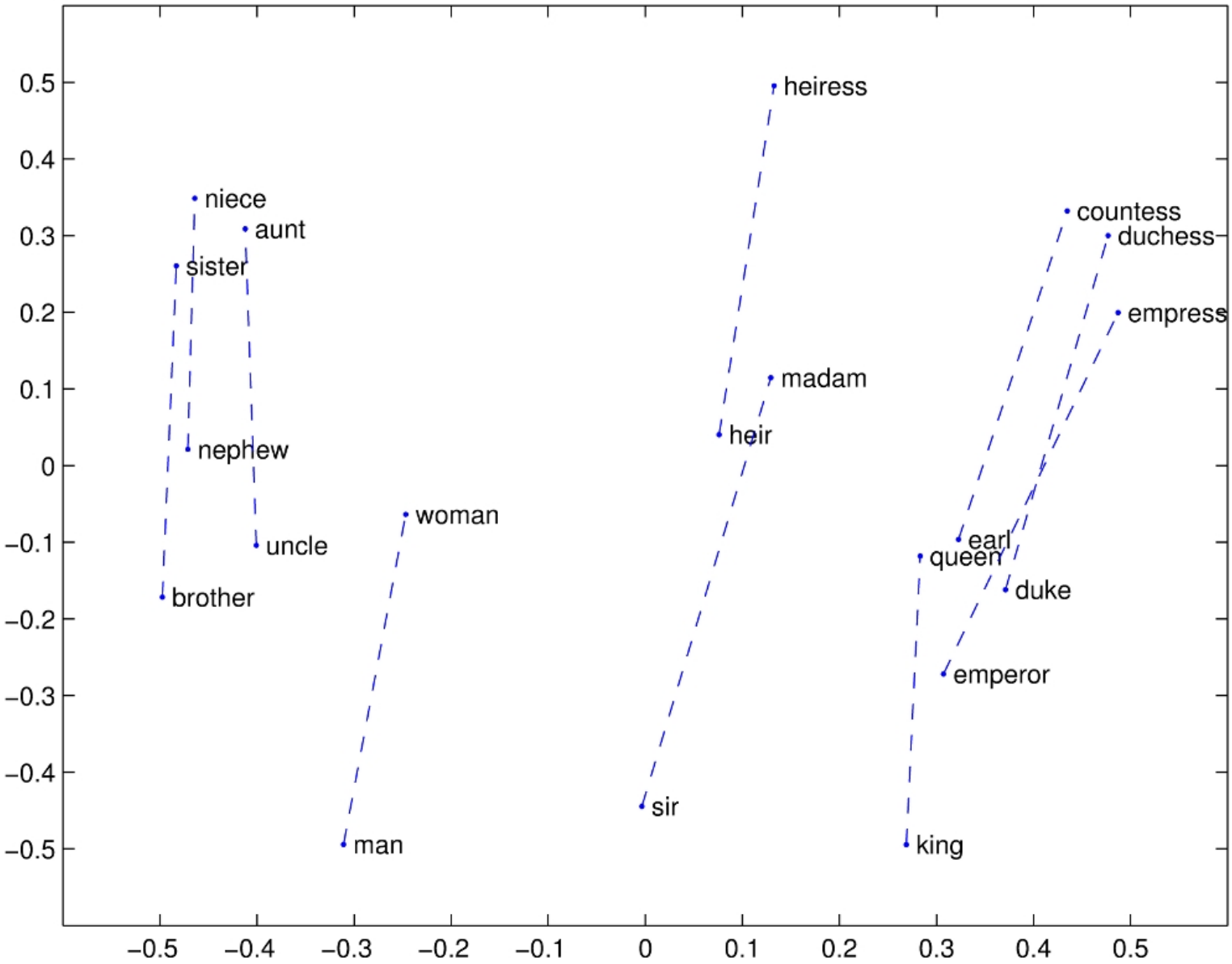


- Dimensionality
- Good dimension is ~300



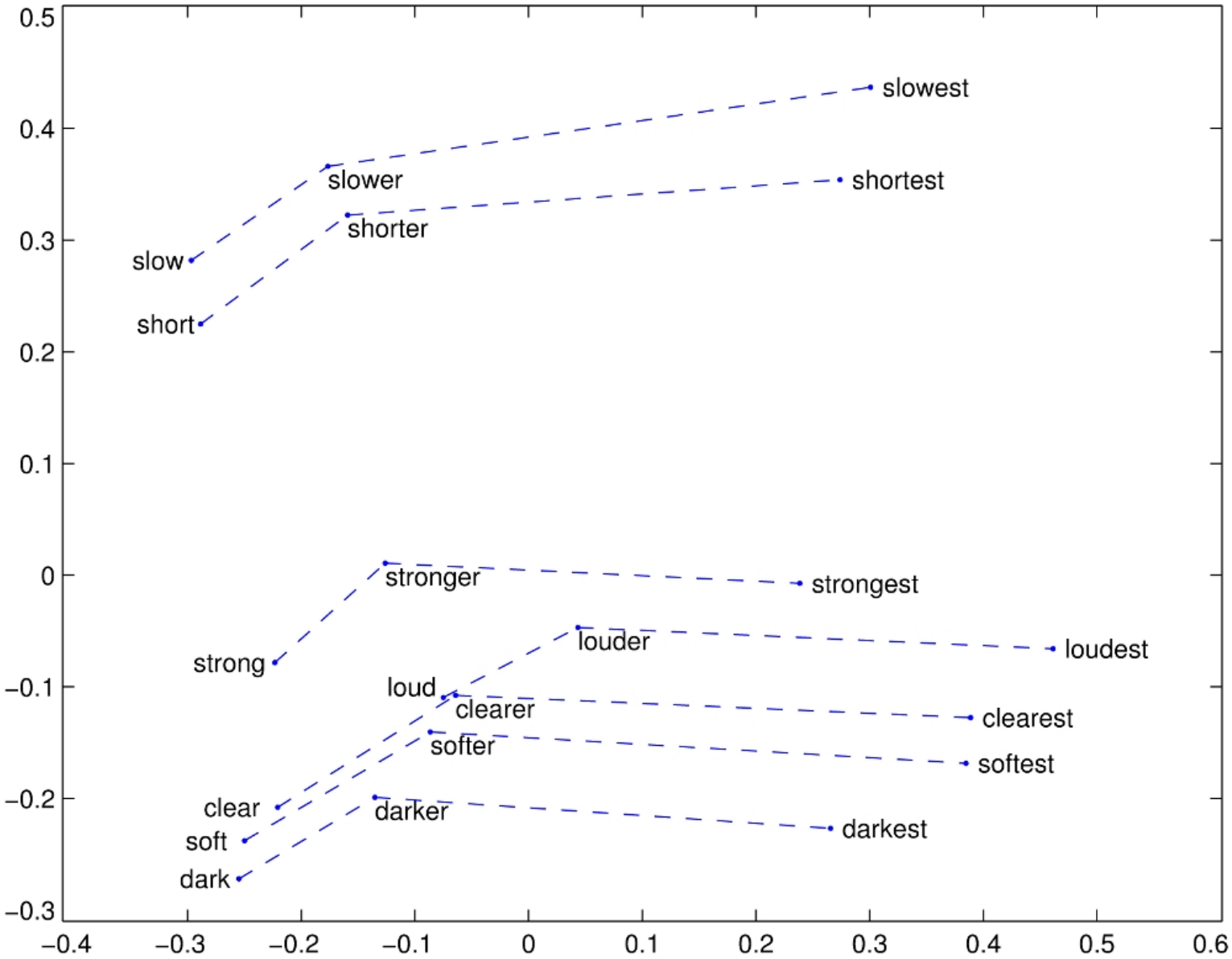
Pennington, Socher, and Manning, EMNLP 2014

GloVe Results (Gender)



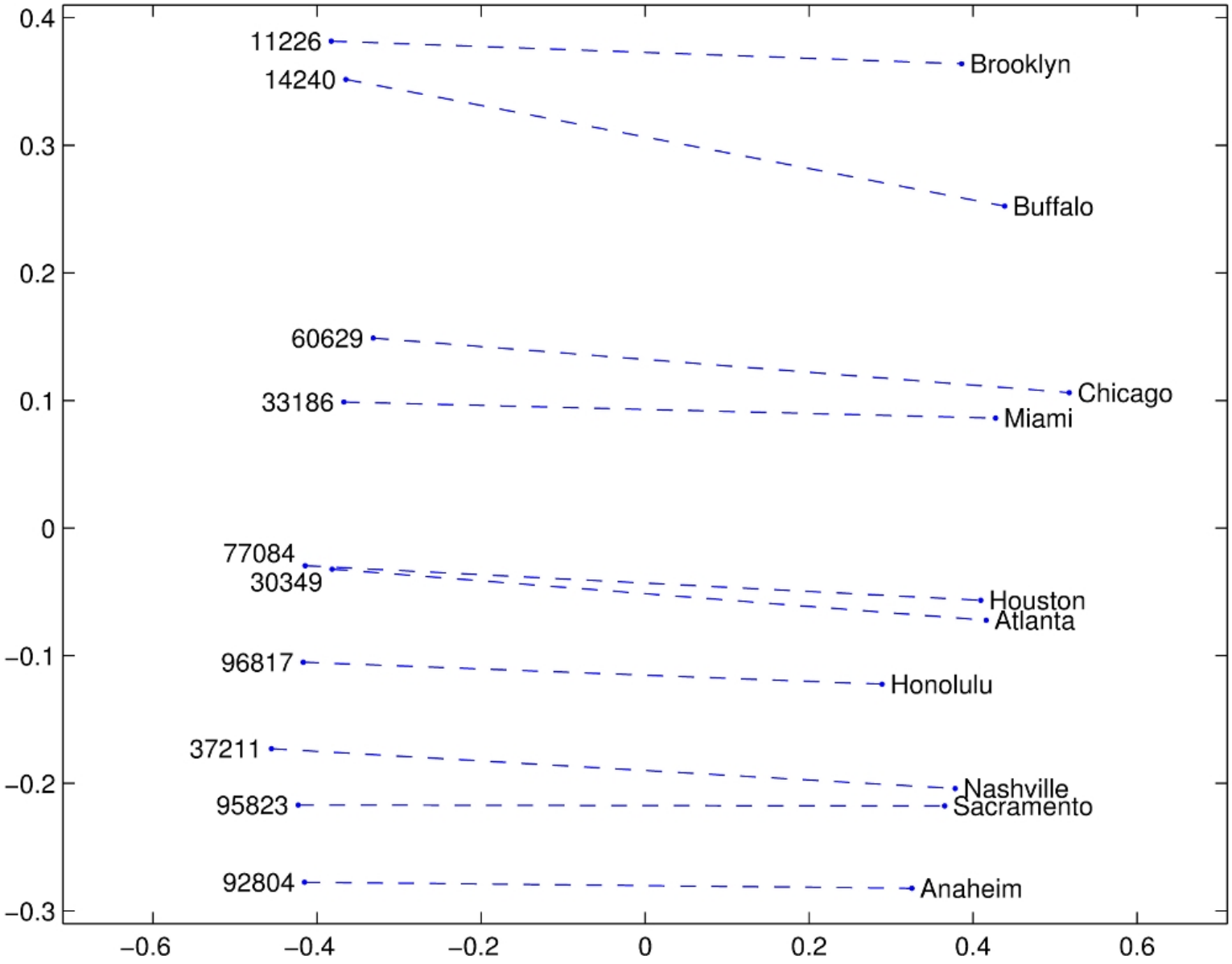
Pennington, Socher, and Manning, EMNLP 2014

GloVe Results (Superlative)



Pennington, Socher, and Manning, EMNLP 2014

GloVe Results (Zip Codes)



Pennington, Socher, and Manning, EMNLP 2014

Outline

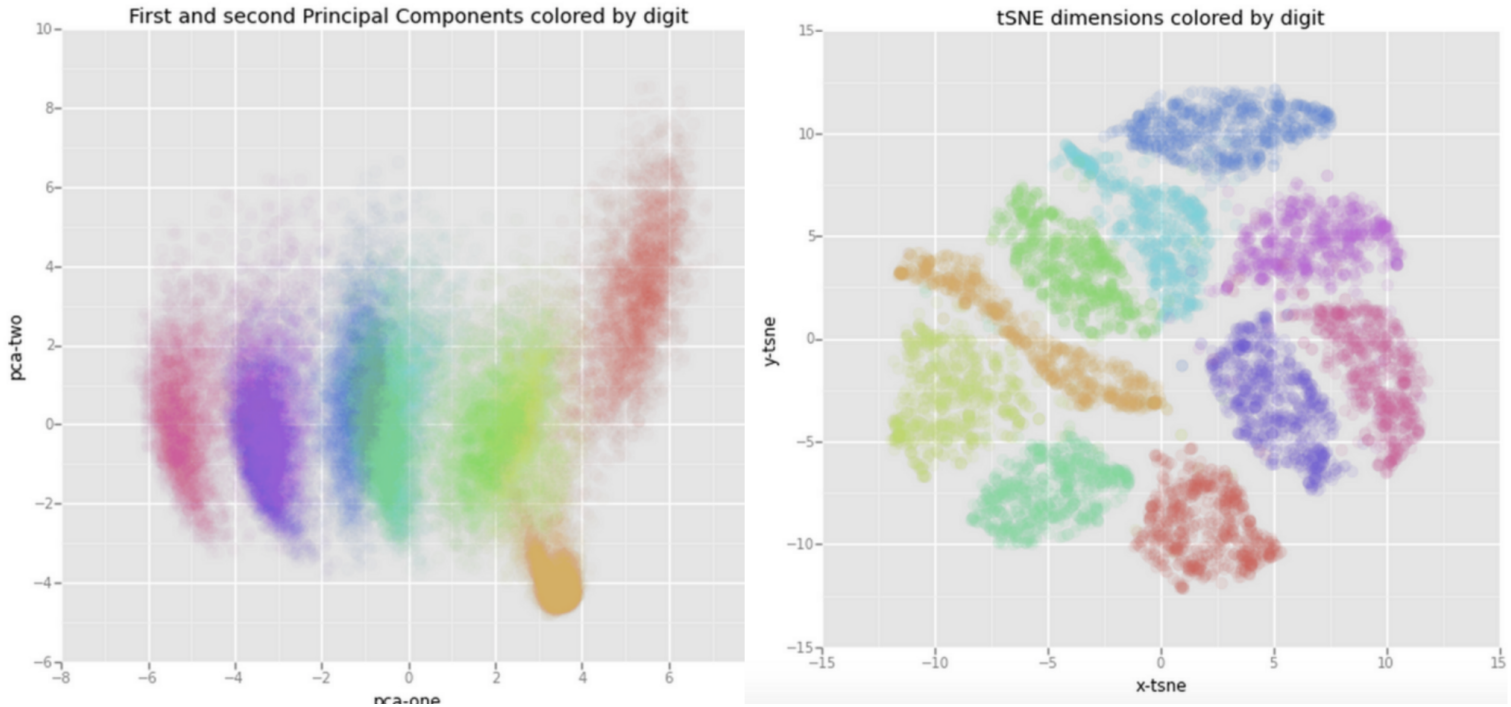
How do We Evaluate How We're Doing?

Can be hard to think about with unsupervised tasks, as they don't have clear evaluation metrics.

- Intrinsic evaluation (how good based on the features produced)
- Extrinsic evaluation (how useful for downstream tasks)

Visualization Via Dimensionality Reduction

Dimensionality reduction for digits. Use non-linear projections to group things that are close in high dimensional space.



Derksen, 2016

Correlation with Human Evaluations

- Can correlate word vector distances with human judgments (datasets collected by psychologists)
- Example dataset: WordSim353
  `http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/`

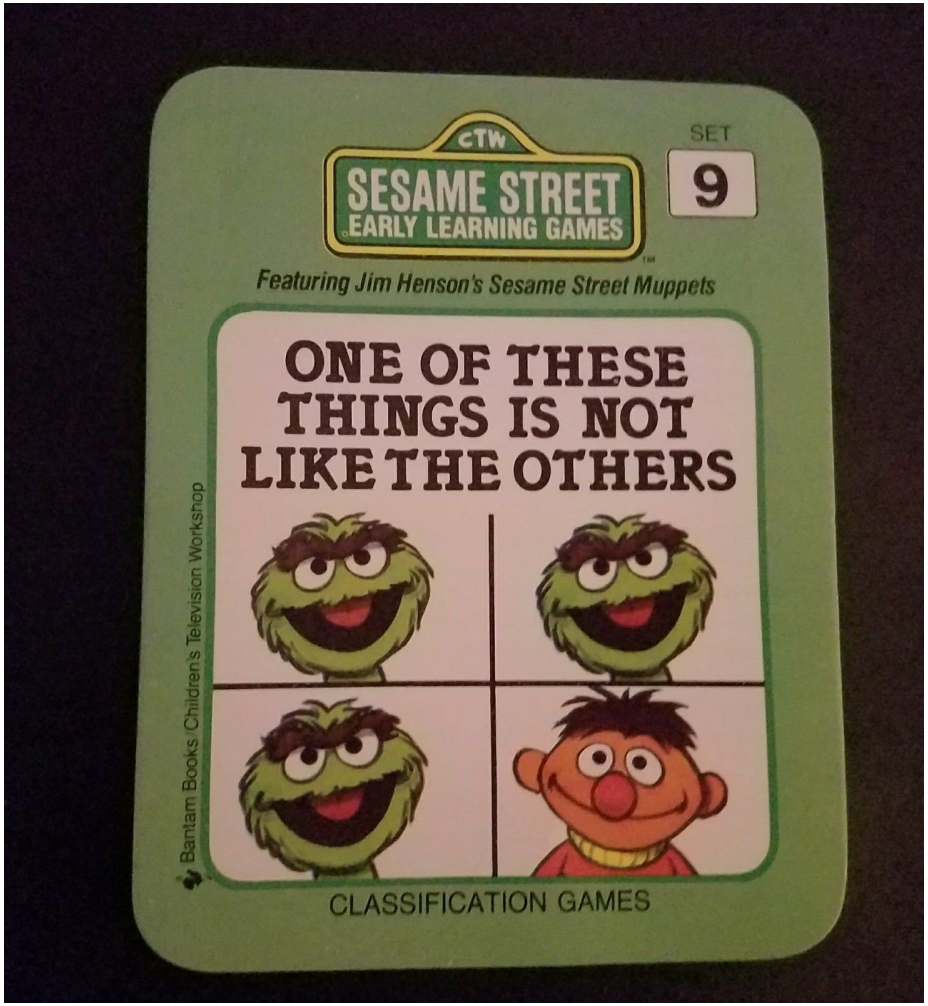
Example Evaluation

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Pennington, Socher, and Manning, EMNLP 2014

Evaluating performance with analogy datasets is also common.

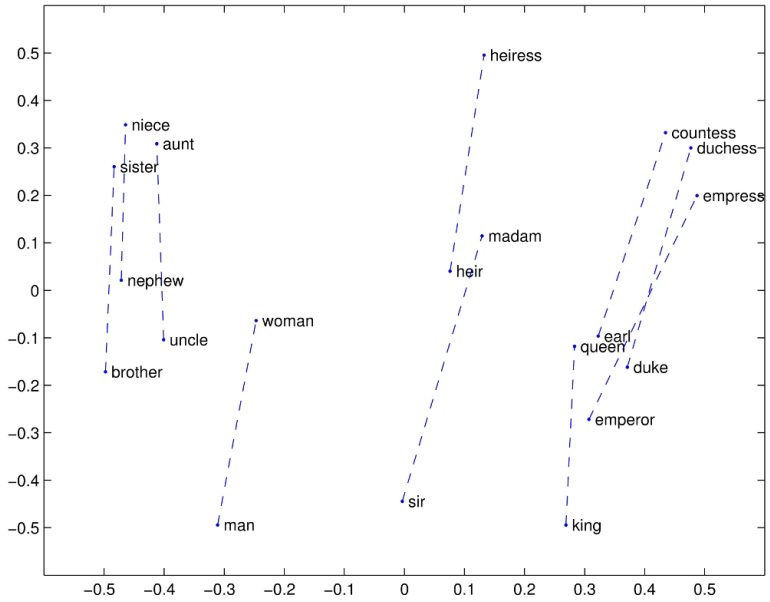
Evaluating with Categorization Tasks



Another common evaluation is categorization tasks. Human created datasets with four words, which one is not like the other?

Outline

Interpreting Word Vectors



Pennington, Socher, and Manning, EMNLP 2014

- Importantly, we didn't optimize the model to do well on analogy or other language tasks
- We tried to do a simple task - predict words in context using unsupervised methods - and the word vectors with these powerful properties are a side effect
- This will be a theme in NLP

Interpreting Word Vectors

- If you only learn **one thing from the course**, it should be that neural networks can learn better ways to represent data automatically than what we humans can engineer by hand.
- For example, we saw how features vectors from ConvNets can be used for many downstream tasks, in a way that is more robust than using human engineered features vectors or the raw image data. Word embeddings are a striking example of neural networks learning representations.
- Representing data well is essential to the downstream tasks we want to perform.

The Power of Transfer Learning

- This general approach of learning a good representation for task 1 and then using it to perform a downstream task 2 is a powerful theme in deep learning. We saw, for example, that an ImageNet pre-trained backbone - estimated on a dataset of natural images (over 10% of which are dog breeds) - with limited fine tuning can be a powerful features extractor for very different tasks.
- This idea is even more prominent in NLP and variously known as transfer learning, pre-training, or multi-task learning.
- Word embeddings have powered named entity recognition, part-of-speech tagging, parsing, and things like online recommendation engines during the mid-to-late 2010s.

Transfer Learning

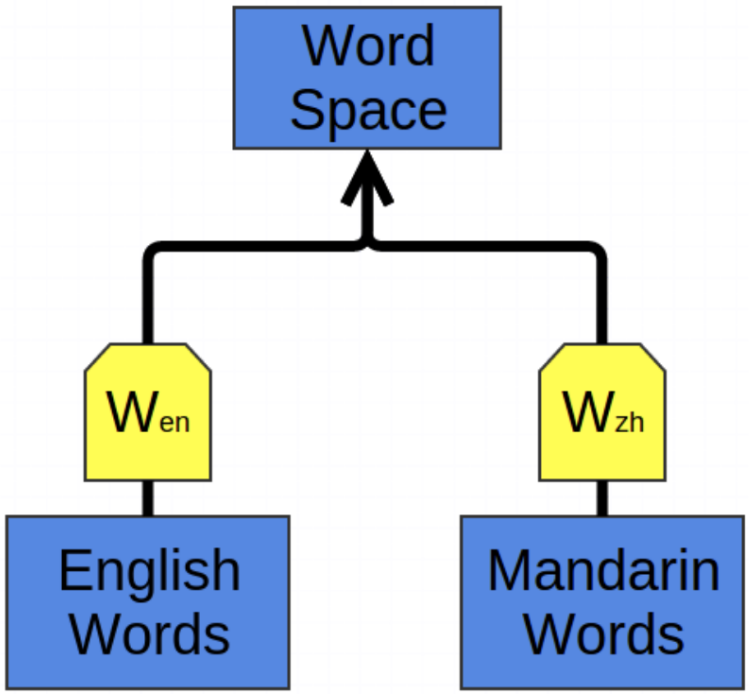
- We often initialize a downstream NLP task with embeddings and then fine-tune to extract the relevant dimensions from the embeddings
- This is the norm whenever you have smaller training datasets (i.e. any annotated dataset you have to create). Less common i.e. in machine translation, where there are massive training datasets

A Counterpart to Transfer Learning

- Pre-training learns a way to represent one type of information (i.e. which words are used in similar contexts) and utilizes that for downstream tasks
- Instead, we can learn a way to map multiple types of data into a single representation
- This can be very powerful

Bilingual Word Embeddings

Learn to embed words from two different languages into a single shared space.

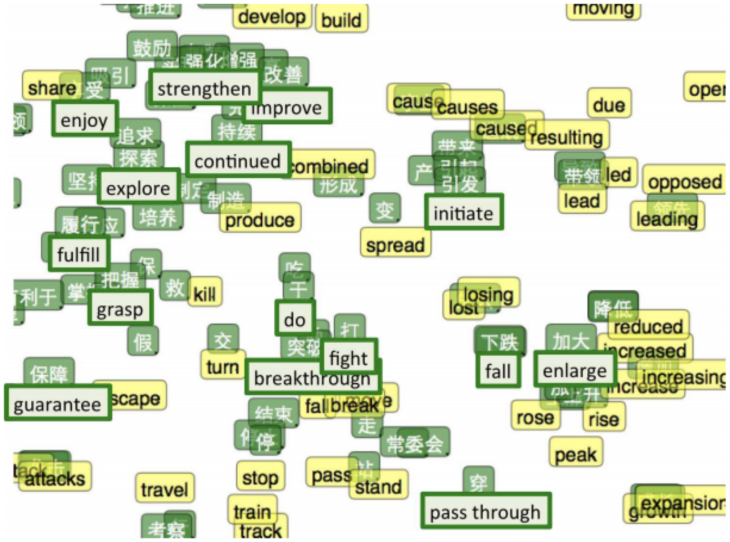


Socher et al. (2013)

Train two word embeddings, as above, but optimize such that words we know are good translations are close

Bilingual Word Embeddings

Does well on words not exposed to during training



t-SNE visualization of the bilingual word embedding. Green is Chinese, Yellow is English. (Socher *et al.* (2013a))

Socher et al. (2013)

This would seem to suggest that languages have similar topologies, such that when some words are forced to align, others do as well

Embedding Images and Words into the Same Representation

- Nothing about this that says we can only embed data of the same type into a single representation.
- One could, for example, classify images by producing a word embedding vector. When you show an image of a dog, the model will output an embedding close to the "dog" word embedding. Similarly for cats, etc.
- What is remarkable is what happens when you test the model on a new class of images. i.e. suppose the model wasn't exposed to horses in training. With a fully supervised image classifier (as we saw), this would be a problem.
- Offers promising for zero shot image classification. Promise for OCR?

Outline

Word Embeddings Reflect Biases in the Corpus

Extreme *she*	**Extreme *he***
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

Gender stereotype *she-he* analogies

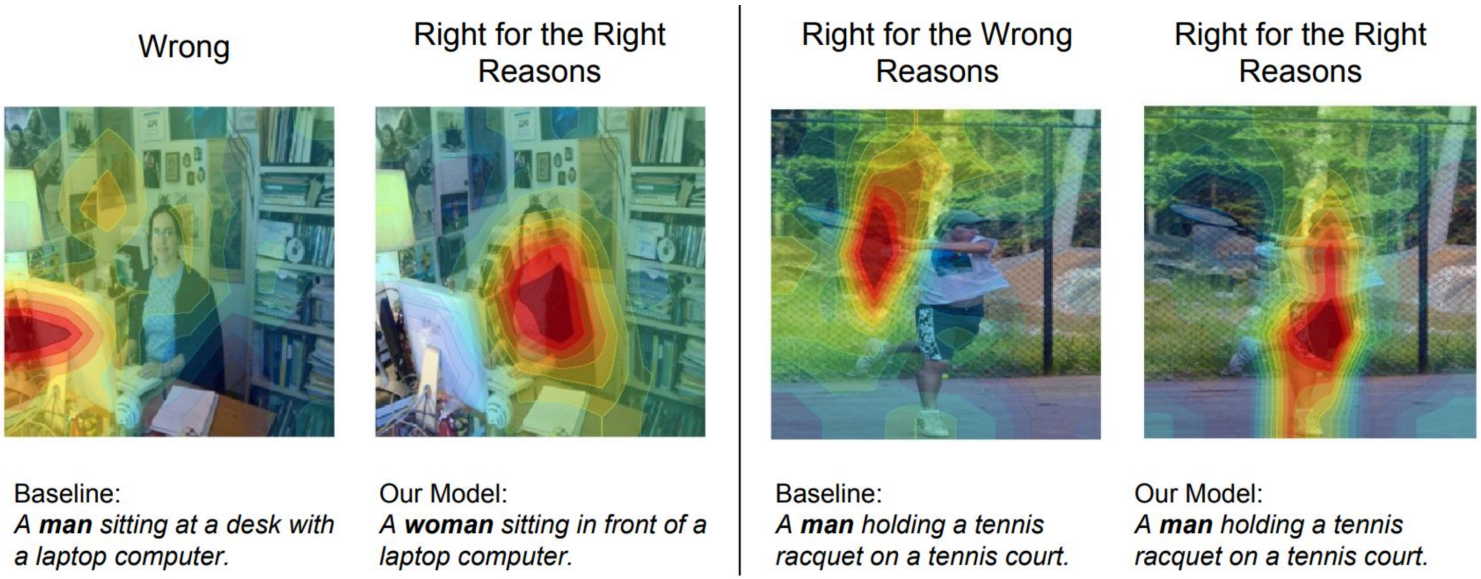
sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant

Gender appropriate *she-he* analogies

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Bolukbasi et al. 2016

Another Example: Image Captioning



Burns et al., 2018 ("Women Also Snowboard")

Word embeddings quantify 100 years of gender and ethnic stereotypes (PNAS)

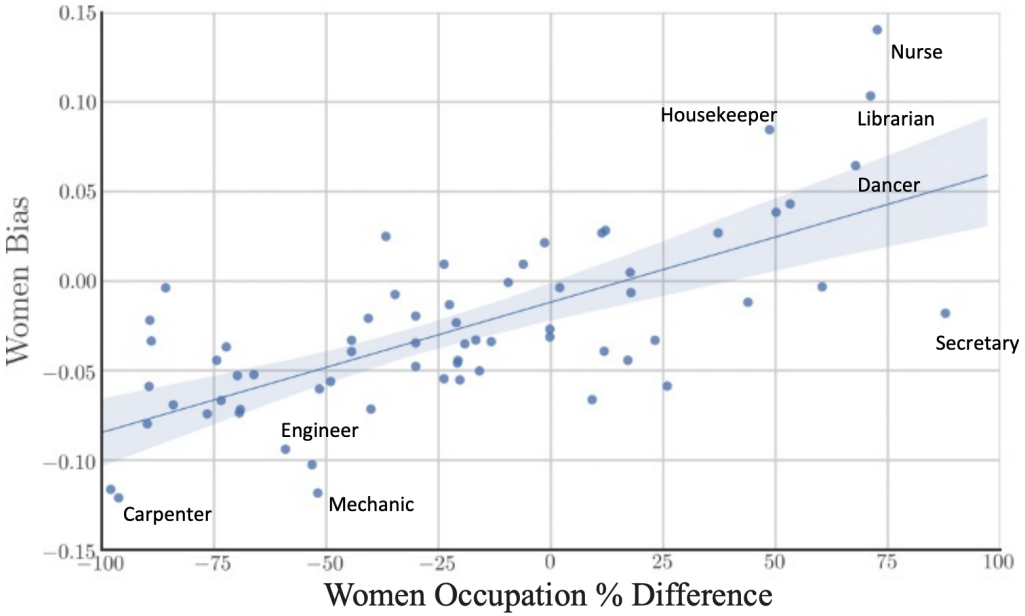


Fig. 1. Women's occupation relative percentage vs. embedding bias in Google News vectors. More positive indicates more associated with women on both axes. $P < 10^{-10}$, $r^2 = 0.499$. The shaded region is the 95% bootstrapped confidence interval of the regression line. In this single embedding, then, the association in the embedding effectively captures the percentage of women in an occupation.

Garg et al., 2018

Word embeddings quantify 100 years of gender and ethnic stereotypes (PNAS)

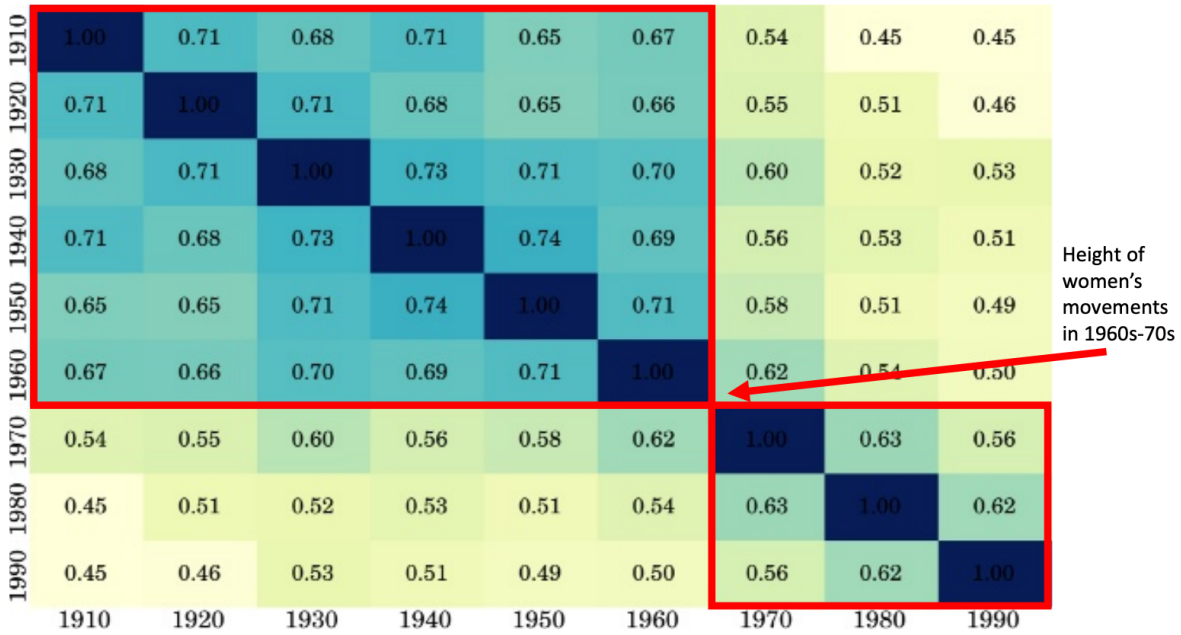


Fig. 4. Pearson correlation in embedding bias scores for adjectives over time between embeddings for each decade. The phase shift in the 1960s–1970s corresponds to the US women's movement.

COHA embeddings: trained on the Corpus of Historical American English/Google Books, by decade. Also correlate with census occupation shifts.

Word embeddings quantify 100 years of gender and ethnic stereotypes (PNAS)

Table 2. Top adjectives associated with women in 1910, 1950, and 1990 by relative norm difference in the COHA embedding

1910	1950	1990
Charming	Delicate	Maternal
Placid	Sweet	Morbid
Delicate	Charming	Artificial
Passionate	Transparent	Physical
Sweet	Placid	Caring
Dreamy	Childish	Emotional
Indulgent	Soft	Protective
Playful	Colorless	Attractive
Mellow	Tasteless	Soft
Sentimental	Agreeable	Tidy

COHA embeddings: trained on the Corpus of Historical American English/Google Books, by decade

Word embeddings quantify 100 years of gender and ethnic stereotypes (PNAS)

Table 3. Top Asian (vs. White) adjectives in 1910, 1950, and 1990 by relative norm difference in the COHA embedding

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

Word embeddings quantify 100 years of gender and ethnic stereotypes (PNAS)

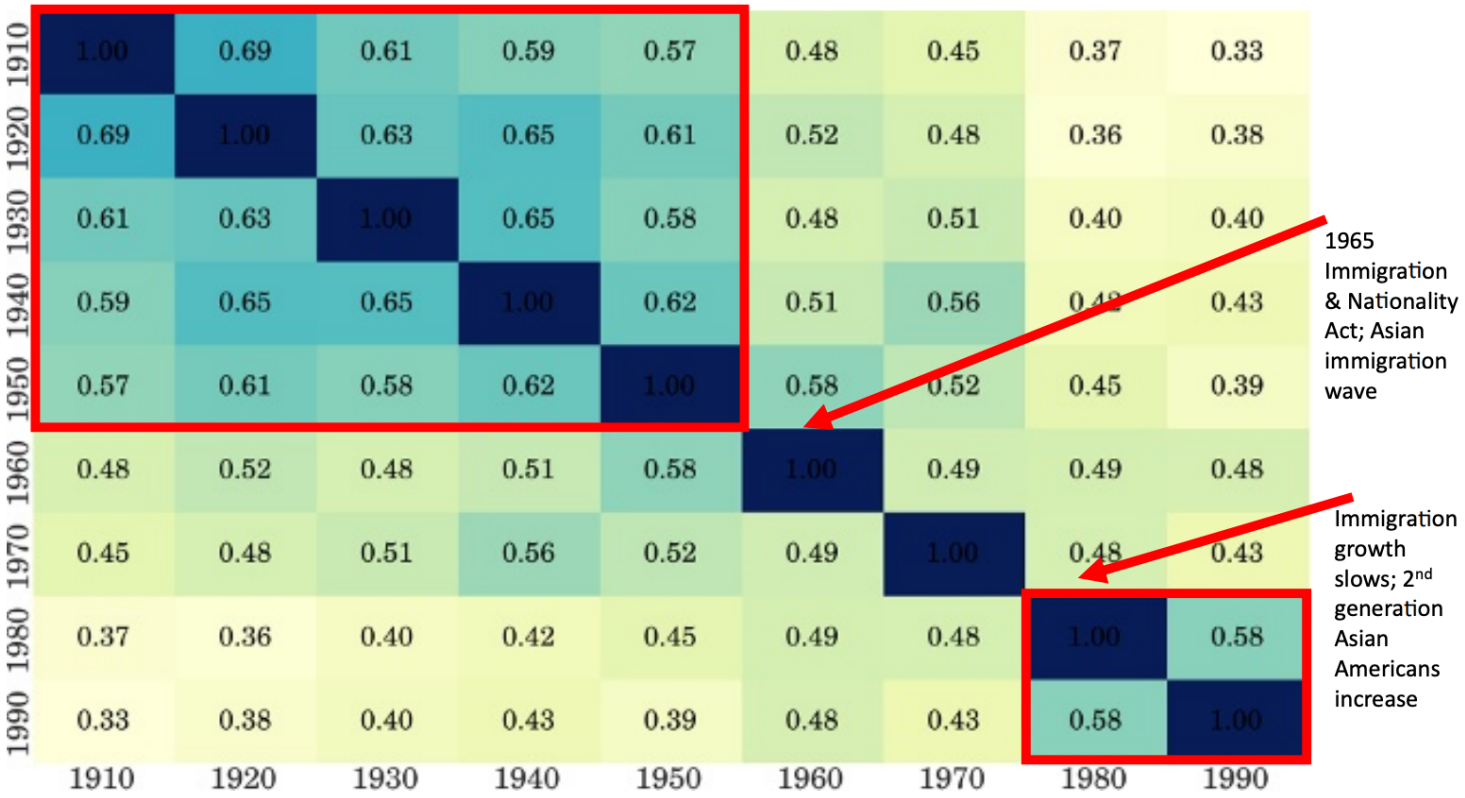


Fig. 5. Pearson correlation in embedding Asian bias scores for adjectives over time between embeddings for each decade.

Word Sense Ambiguity

- Most words have multiple meanings
- Very prevalent with common words
- Very prevalent with old words