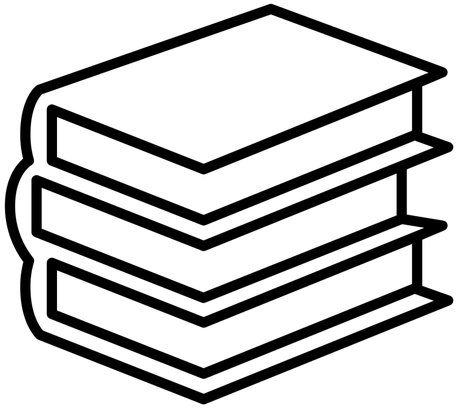
CS224C: NLP For CSS

Word Embeddings

Slides adapted from: Diyi Yang and CS 224N

How Can We Represent the Meaning of a Word?



Using dictionary definitions won’t help us ….

Numerous Ways to Represent Words

- We're going to use **embeddings** which are vector representation of words
- Let's start with using **one-hot encodings**:

  E.g., let's represent "cat" using: [0 0 0 … 0 0 1]

  E.g., let's represent "dog" using: [1 0 0 … 0 0 0]

- One-hot encoding could work… but there are no relationships between the vectors!

We're looking for vector representation that allows us to compare the vectors in semantically meaningful ways

Distributional Similarity

Idea: words that are semantically similar often occur in similar context

- "Can you pour me a **cup** of **coffee**?"
- "The math problem is really difficult to solve and required **addition** and **multiplication**"

Embeddings that are "good at predicting neighboring words" are also good at representing similarity

word2vec at a high level

word2vec at a high level

Instead of counting how often each word occurs near “coffee”

word2vec at a high level

Instead of counting how often each word occurs near “coffee”

Train a classifier on a prediction task:

p (context | *w*) = p (context | *coffee*)

word2vec at a high level

Instead of counting how often each word occurs near “coffee”

Train a classifier on a prediction task:

p (context | *w*) = p (context | *coffee*)

Loss function (how well do we do this prediction?):

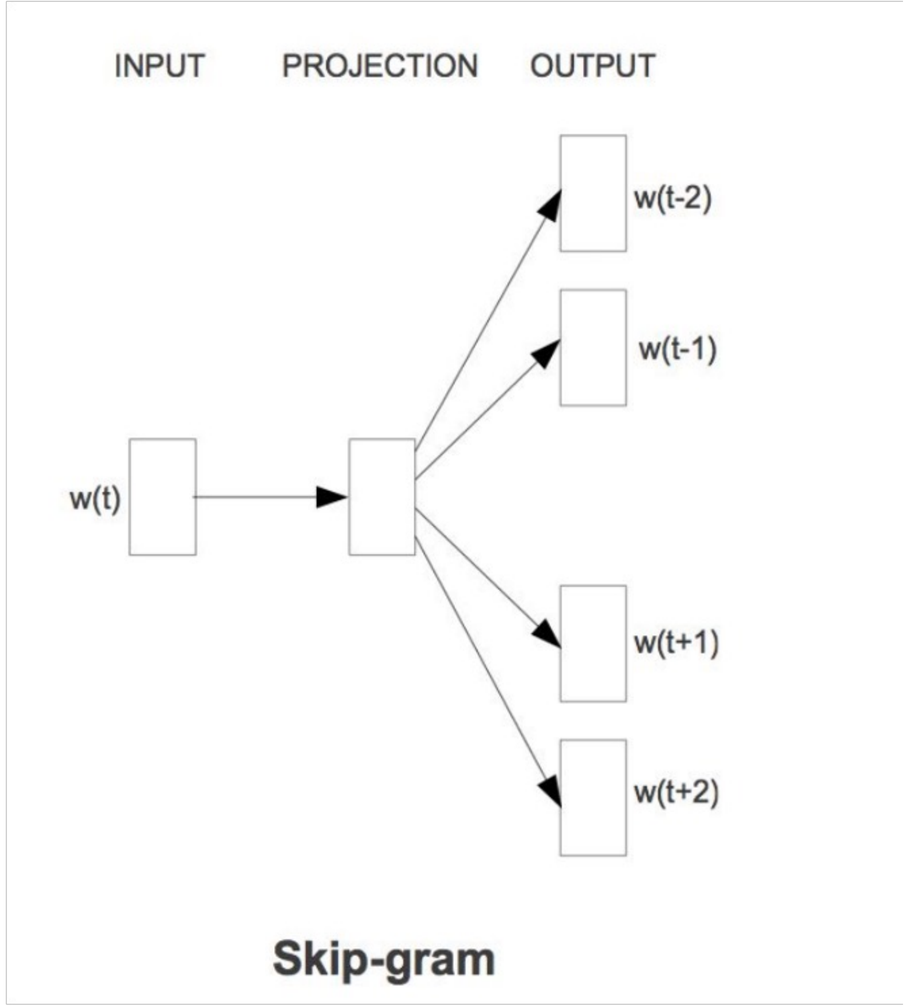
= 1 – p (context | *w)*

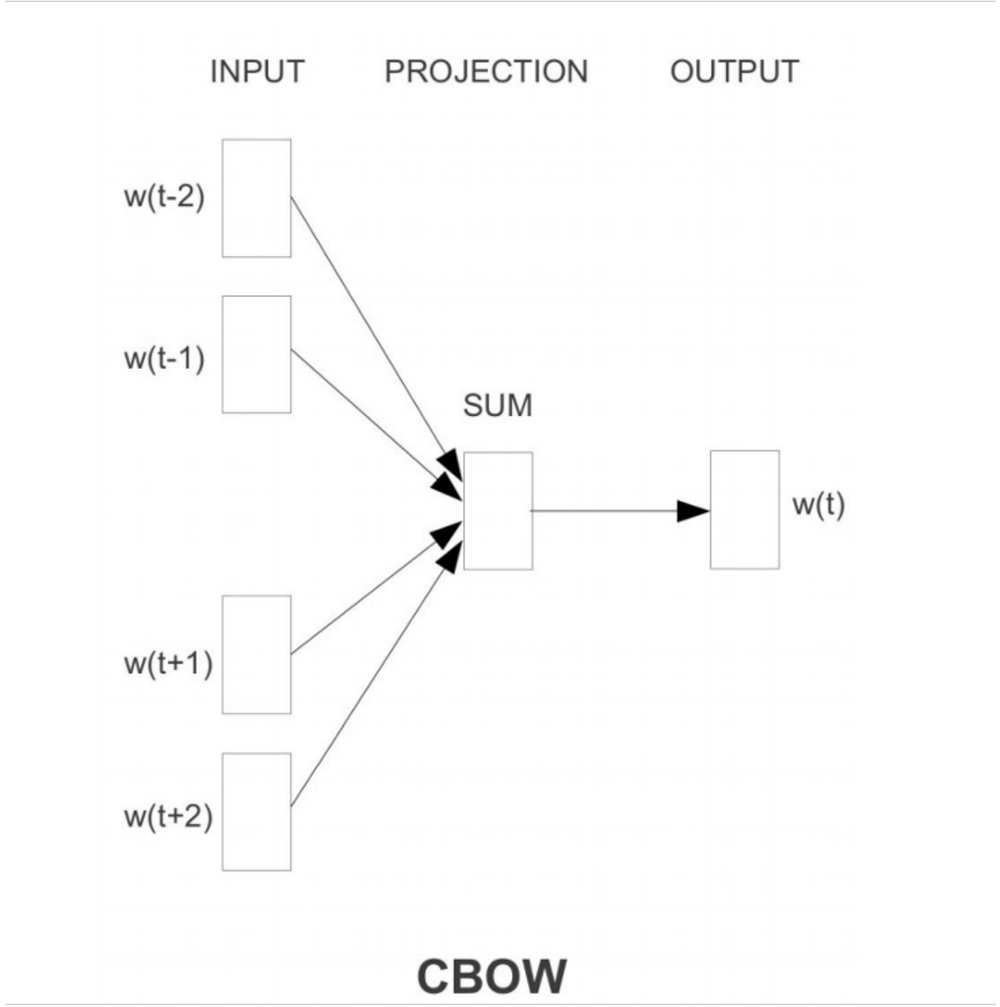
If we can perfectly predict the context words around *coffee*, then we’ll have a loss of 0.

This task itself doesn’t create embeddings…

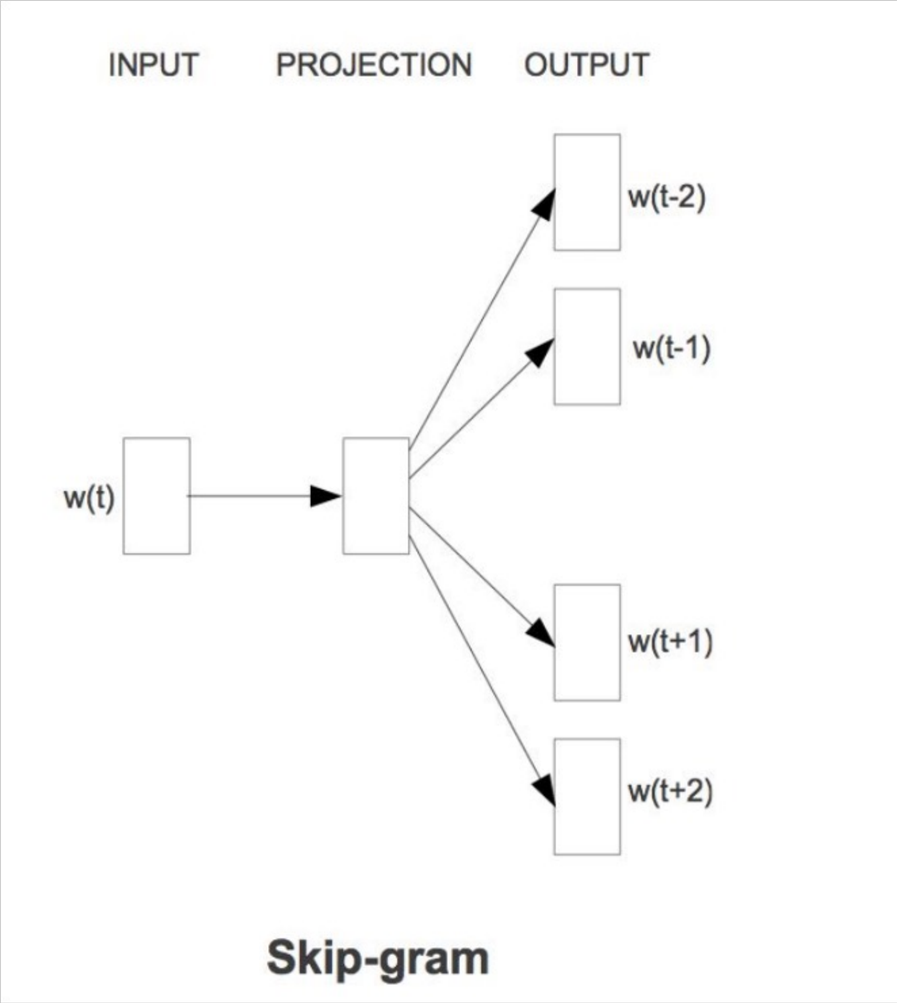
But we'll take the learned classifier weights as the word embeddings

Skip-gram vs Continuous Bag of Words



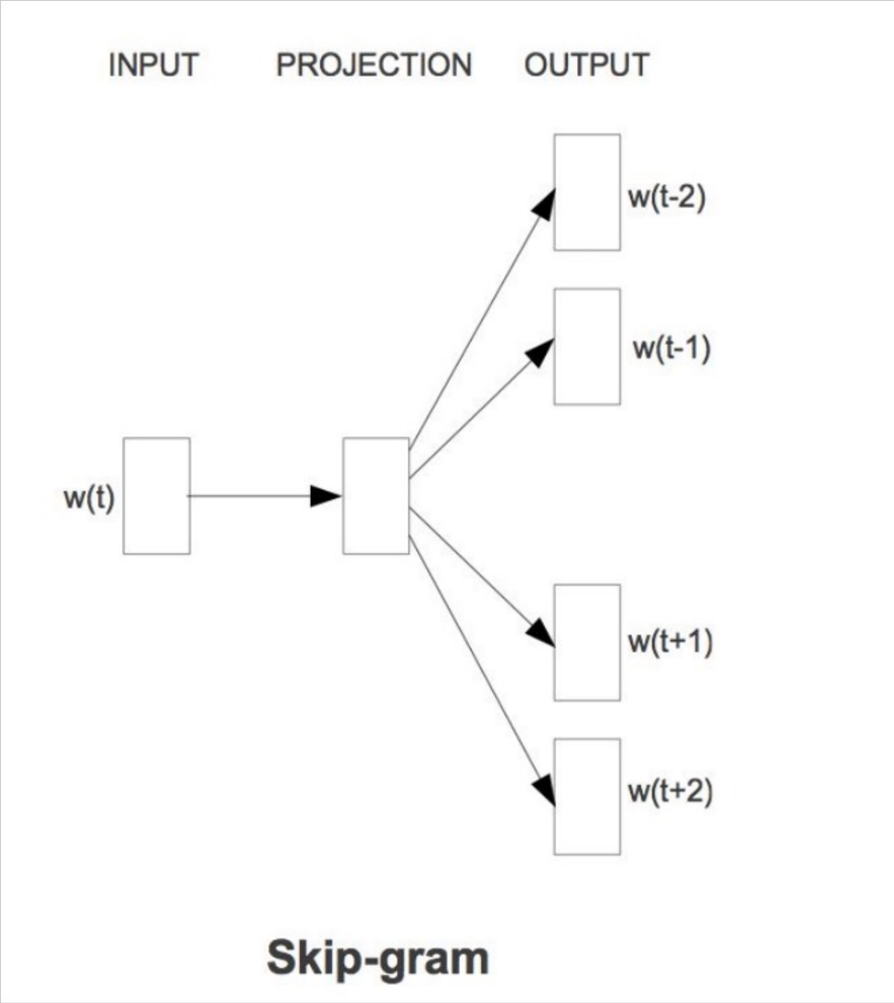


Skip-gram



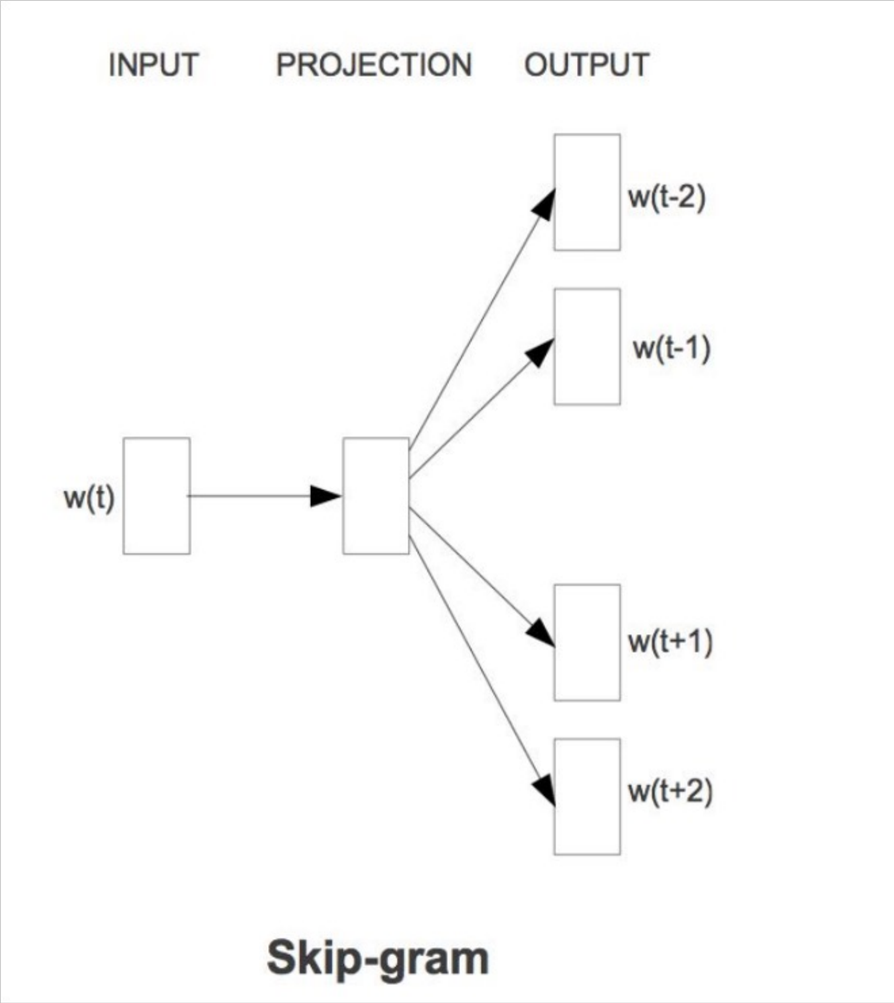
- We start with a representation of the word *coffee* and train a classifier to predict its neighboring words *(cup, milk)* using that vector

Skip-gram



- We start with a representation of the word *coffee* and train a classifier to predict its neighboring words *(cup, milk)* using that vector
- If we are successful, we will have a loss of 0

Skip-gram



- We start with a representation of the word *coffee* and train a classifier to predict its neighboring words *(cup, milk)* using that vector
- If we are successful, we will have a loss of 0
- If we are unsuccessful, we use gradients to update our representation of the word *coffee*

Measuring the Semantic Similarity of Vectors

The most common similarity metric is **cosine**, which is the angle between the vectors

- For vectors **u** and **v,** the cosine similarity is the dot product of the two vectors, divided by the product of the length of the two vectors

- Cosine (u, v) = $\frac{u \cdot v}{|u||v|}$

Other distance (Euclidean, norms) might be appropriate and meaningful for a number of other tasks

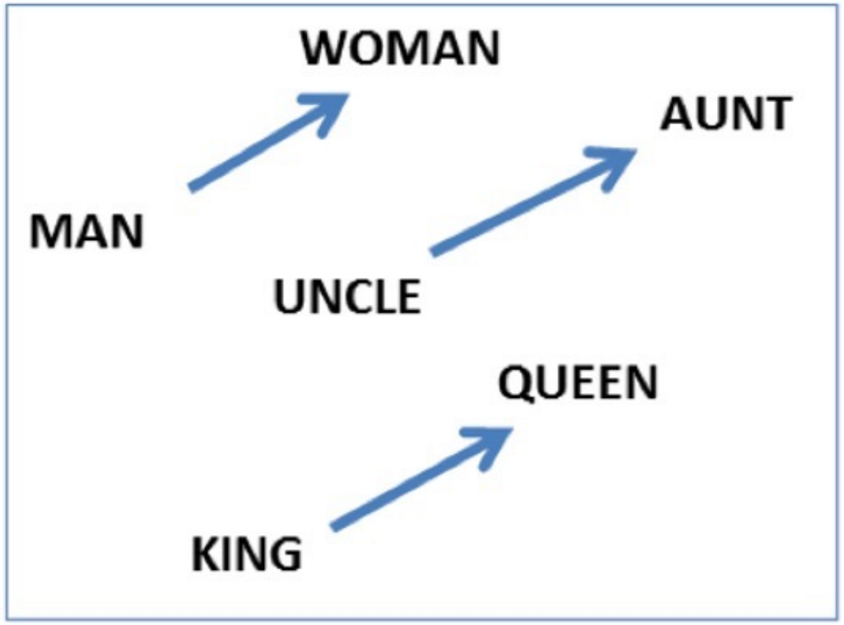
Tasks Semantic Similarity

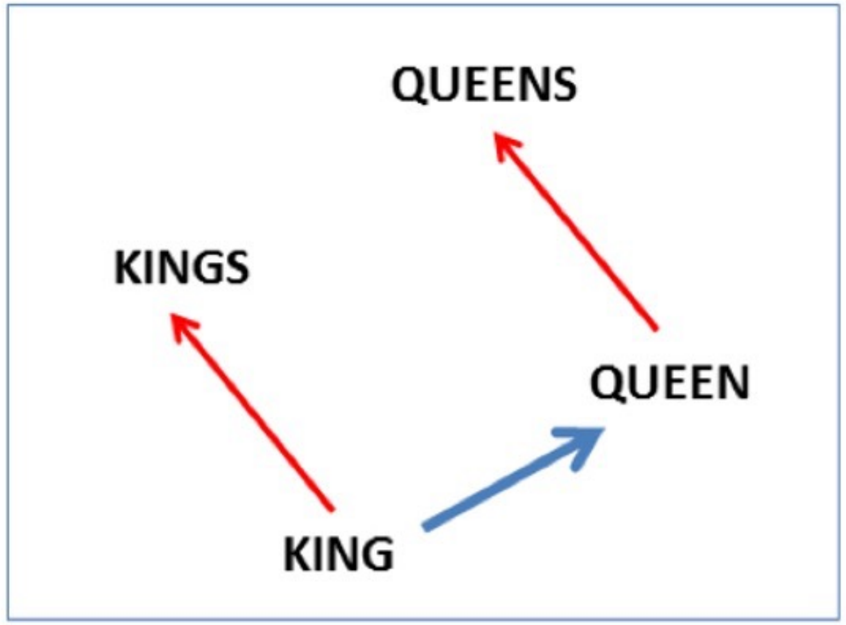
- Example 1: Automatically identifying components of parts
- Example 2: Identifying related concepts in a historical corpus
- Evaluation Datasets WordSim-353 (Finkelstein et al., 2002) and SimLex-999 (Hill et al., 2015)

Analogy: Embeddings Capture Relational Meaning

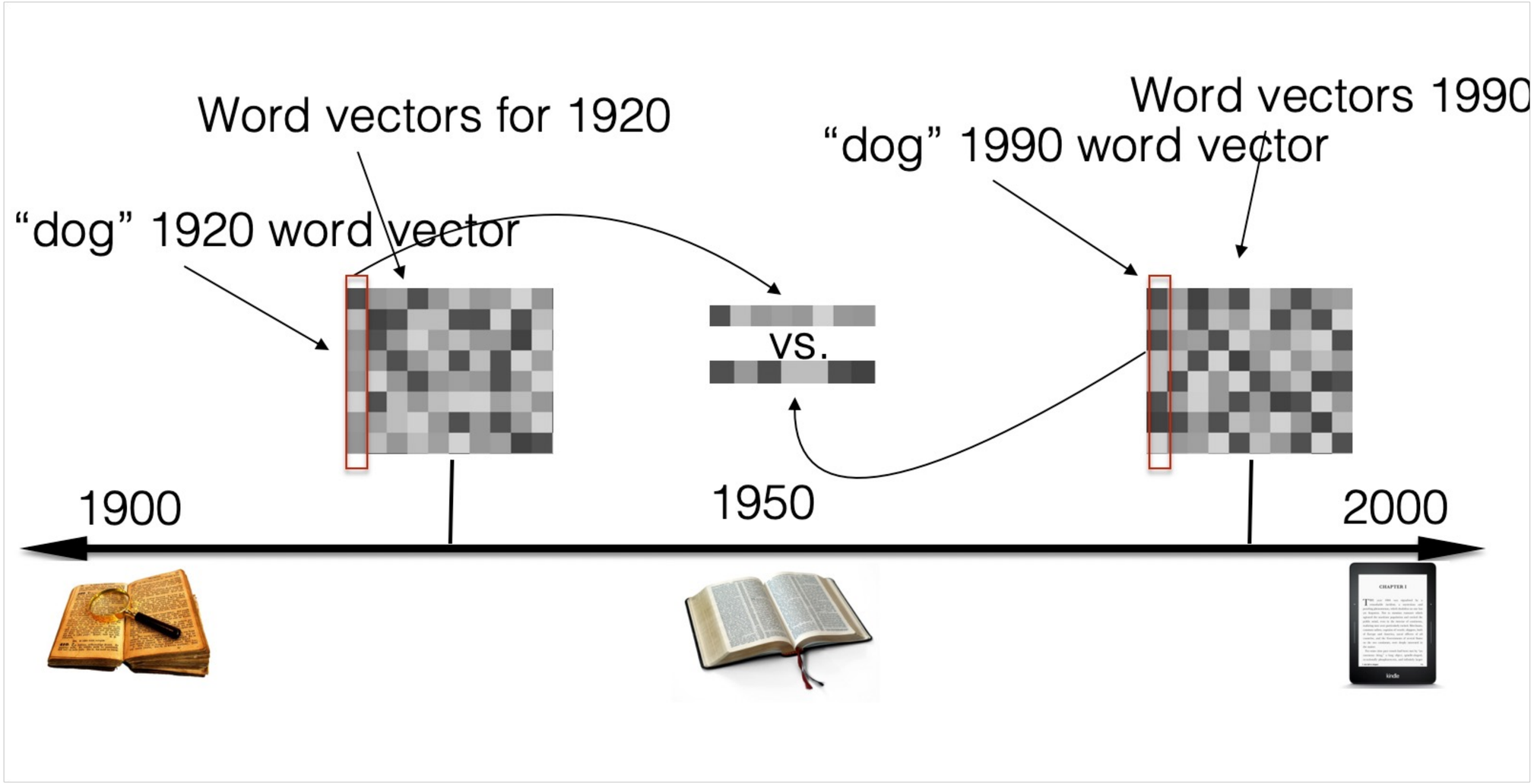
vector(‘king’) - vector(‘man’) + vector(‘woman’) ≈ vector(‘queen’)

vector(‘Paris’) - vector(‘France’) + vector(‘Italy’) ≈ vector(‘Rome’)





Diachronic word embeddings for studying language change!



Embeddings Reflect Cultural Bias

Ask “father : doctor :: mother : x”

x = nurse

Ask “man : computer programmer :: woman : x”

x = homemaker

Caliskan et al. find negative implicit associations:

- African-American names (Leroy) had a higher GloVe cosine with unpleasant words (abuse, stink, ugly)
- European American names (Brad, Greg) had a higher cosine with pleasant words (love, peace, miracle)

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.

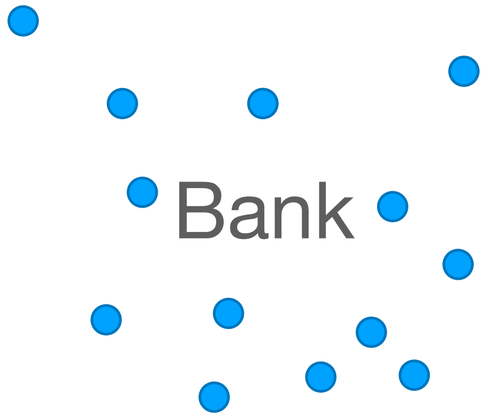
Caliskan, Aylin, Joanna J. Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases." Science 356.6334 (2017): 183-186.

Shortcoming of Static Embeddings

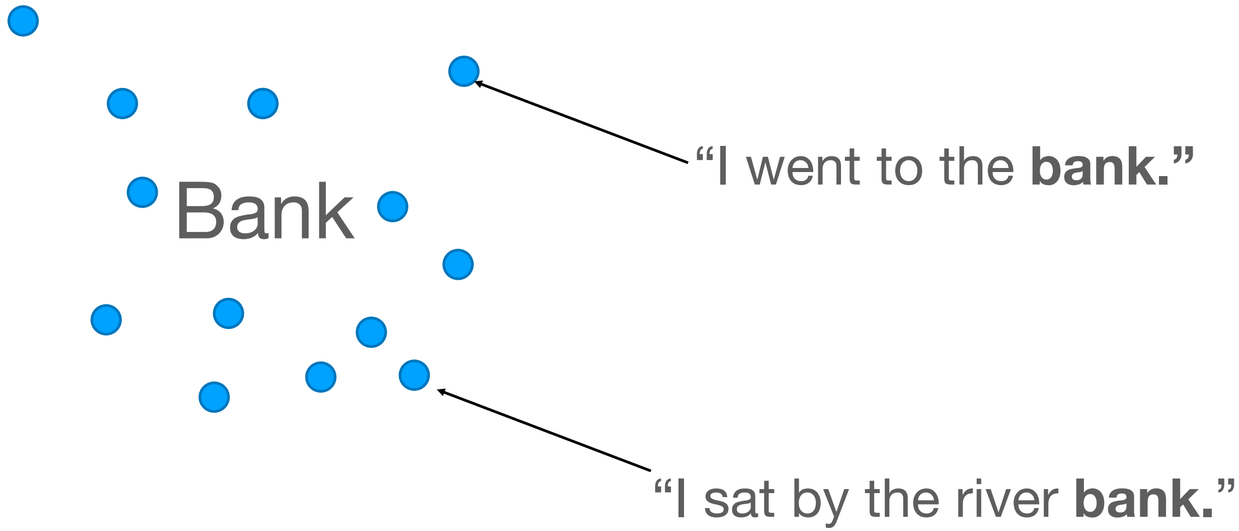
- Static embeddings vectors (word2vec, GloVe)
- However, the identity of a word can actually represent multiple different meanings (senses) depending on the context it appears in
- “I went to the **bank.”** vs “I sat by the river **bank.”**
- In static embeddings, **bank** is represented by a single vector, when in fact it represents completely different meanings in the two contexts
- In other words, there is a single representation for a **word type**, despite differences in context of the **word token**

Static to Contextualized Embeddings

Static to Contextualized Embeddings



Static to Contextualized Embeddings



Contextualized Word Embeddings

- Whereas static embeddings used context to **train** a model to build a single representation of a word
- Contextualized embeddings are pre-trained using context **and** additionally, embed words with their contexts to get a contextualized representation of word tokens

- Deep contextualized word representations (Peters et al.) (ELMo)
- Attention is All You Need (Aswani et al.) (Transformers)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al.) (BERT)

Contextualized Word Embeddings

- For BERT, to create word embeddings,
- Feed the model a sentence with the target word, “I went to the **bank.**”
- Extract the last few hidden layers from the model corresponding to the target word
- Take the average (or concatenation) of the hidden layers

What Can We Learn from Contextualized Embeddings for CSS?

- We can perform the analysis discussed above but at a more granular level!
- In the diachronic sense change example, we needed to train two separate models to extract pre-trained embeddings from two different time intervals
- With contextualized word embeddings, we simply have to pass in two different contexts of the word
- This is done, without needing to retrain the model

Applications of Contextualized Word Embeddings

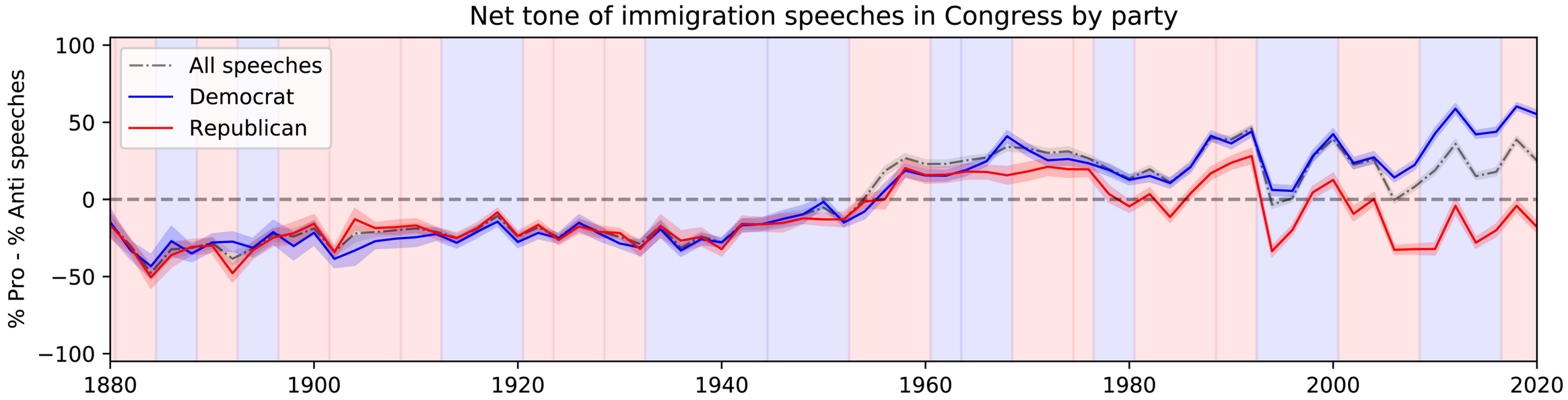
- We can also examine how contemporary speakers use the same word differently
- Card et al. (2022) examines how use of the word *immigrant* has changed over time and how the word is used differently across political parties
- Lucy et al. (2022) examines how the representation of *people* varies across online communities

Lucy, Li, Divya Tadimeti, and David Bamman. "Discovering Differences in the Representation of People using Contextualized Semantic Axes." EMNLP 2022

Card, Dallas et al. “Computational analysis of 140 years of US political speeches reveals more positive but increasingly polarized framing of immigration.” Proceedings of the National Academy of Sciences of the United States of America 119 (2022)

Increasingly polarized framing of immigration

- Quantitative analysis of 140 years of US congressional and presidential speech about immigration
- Find a rise in pro-immigration attitudes beginning in the 1940s, followed by a steady decline among Republicans (relative to Democrats)



A Method for Measuring Implicit Dehumanizing Metaphors

- For each sentence that mentions “immigrant” , they remove the mention (e.g., “foreigners”) from the sentence, replacing it with a special <MASK> (e.g., “the tendency of [MASK] to flock together”)
- Feed the sentence through the model and examine the words the model is predicting for the <MASK> token
- Over the predictions, they sum together the probability that was placed on dehumanizing terms like “animal” or “cargo”
- The lists dehumanizing terms were selected ahead of time and are sorted into categories

Contextualized Word Embeddings Aren't Free From Biases

- Static embeddings are heavily biased by frequency based on their training (words that occur more frequently are going to be represented more closely together)
- Wolfe and Caliskan (2021) illustrate how BERT embeddings also associate minority names more likely with unpleasantness
- Zhou et al. (2022) shows how the names of low frequency (typically poorer) countries are seen as less distinct than those from high frequency (typically richer countries)

Wolfe, R., & Caliskan, A. (2021). Low frequency names exhibit bias and overfitting in contextualizing language models. arXiv preprint arXiv:2110.00672.

Zhou, Kaitlyn, Kawin Ethayarajh, and Dan Jurafsky. "Richer countries and richer representations." ACL Findings 2022

Naming these Harms

Allocation Harms: where systems unfairly allocate resources

- Imagine a recommendation system that more closely associates doctors with masculine names --- resulting in fewer opportunities for those with feminine names

Representation harms: where systems represent a group of people in an unpleasant, harmful, or demeaning manner

- Certain groups of people being represented in stereotypical or limiting ways

Some of these harms are a result of the training data, but these harms are at times further exacerbated by the algorithms and systems we build

Crawford, K. 2017. The trouble with bias. Keynote at NeurIPS.
Blodgett, S. L., S. Barocas, H. Daume III, and H. Wallach. 2020. ´ Language (technology) is power: A critical survey of “bias” in NLP. ACL.

Looking forward

- The improvement in our ability to represent words has been the foundational to the transformative progress in NLP
- As methods and techniques improve on how words are represented, as computational social scientists, we are better able to conduct accurate and fine-grained analysis of language use
- This analysis reveals to us how words use changes over time, how concepts are connected, and where there are systematic biases and stereotypes to overcome