

CS11-747 Neural Networks for NLP

# Models of Words

Graham Neubig



**Carnegie Mellon University**

Language Technologies Institute

Site

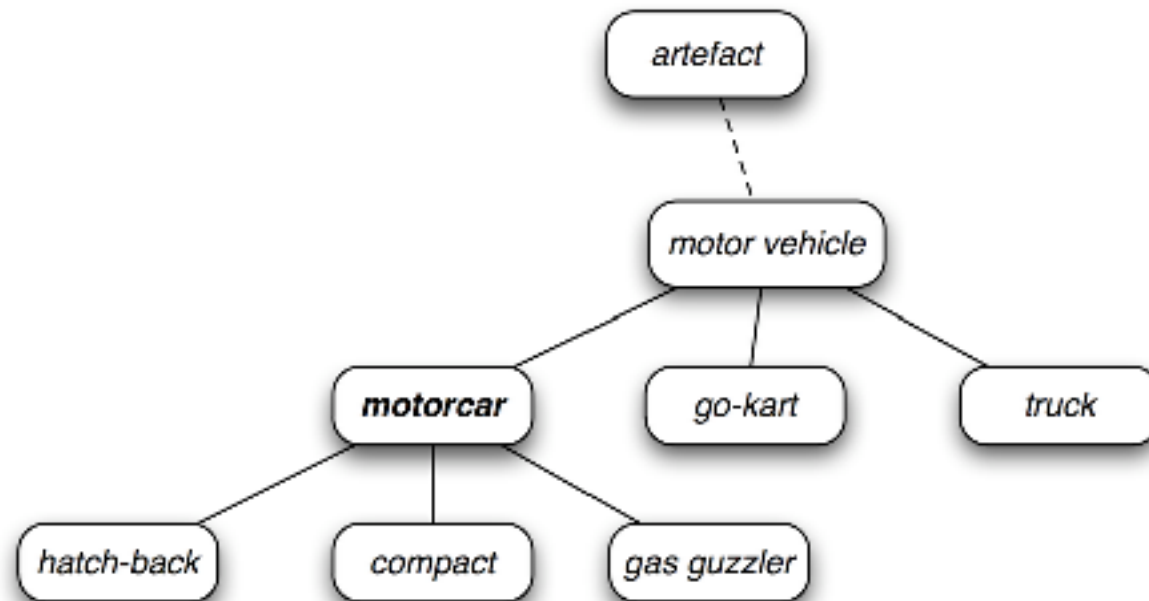
<https://phontron.com/class/nn4nlp2018/>

# What do we want to know about words?

- Are they the same part of speech?
- Do they have the same conjugation?
- Do these two words mean the same thing?
- Do they have some semantic relation (is-a, part-of, went-to-school-at)?

# A Manual Attempt: WordNet

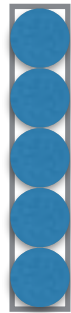
- WordNet is a large database of words including parts of speech, semantic relations



- Major effort to develop, projects in many languages.
- But can we do something similar, more complete, and without the effort?

# An Answer (?): Word Embeddings!

- A continuous vector representation of words

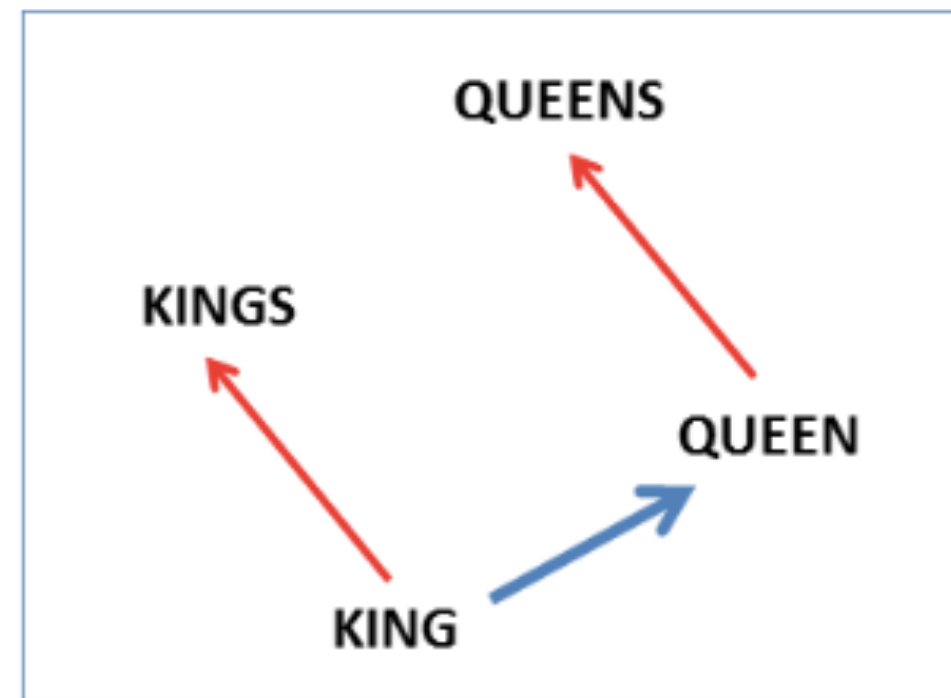
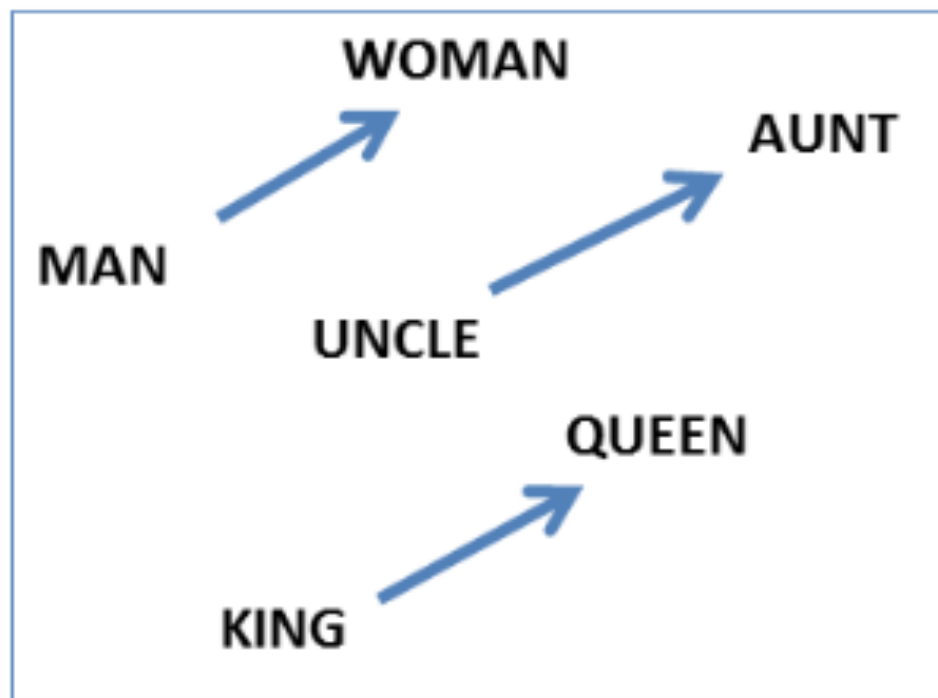


- Within the word embedding, these features of syntax and semantics may be included
  - Element 1 might be **more positive for nouns**
  - Element 2 might be **positive for animate objects**
  - Element 3 might have **no intuitive meaning whatsoever**

# Word Embeddings are Cool!

(An Obligatory Slide)

- e.g. king-man+woman = queen (Mikolov et al. 2013)



- “What is the female equivalent of king?” is not easily accessible in many traditional resources

# How to Train Word Embeddings?

- **Initialize randomly**, train jointly with the task
- Pre-train on a **supervised** task (e.g. POS tagging) and test on another, (e.g. parsing)
- Pre-train on an **unsupervised** task (e.g. `word2vec`)

# Unsupervised Pre-training of Word Embeddings

(Summary of Goldberg 10.4)

# *Distributional vs. Distributed* Representations

- **Distributional representations**
  - Words are similar if they appear in similar contexts (Harris 1954); distribution of words indicative of usage
  - In contrast: *non-distributional* representations created from lexical resources such as WordNet, etc.
- **Distributed representations**
  - Basically, something is represented by a vector of values, each representing activations
  - In contrast: *local* representations, where represented by a discrete symbol (one-hot vector)



# Distributional Representations

(see Goldberg 10.4.1)

- **Words** appear in a **context**

<s>	<s>	<unk>	communications	pittsburgh	acquired	<unk>	&	co.
investment	managemen	inc.	a	pittsburgh	firm	that	runs	a
<s>	mr.	allen	's	pittsburgh	firm	advanced	investment	management
look	stupid	<unk>	former	pittsburgh	<unk>	second	<unk>	<unk>
through	the	university	of	pittsburgh	law	school	<s>	<s>
with	the	university	of	pittsburgh	<s>	<s>	<s>	<s>
<unk>	he	heads	the	pittsburgh	branch	of	the	committee
at	the	university	of	pittsburgh	earn	up	to	\$
for	society	corp.	a	cleveland	bank	said	demand	for
as	washington	<unk>	r.i.	cleveland	<unk>	n.c.	minneapolis	and
<s>	<s>	<unk>	a	cleveland	merchant	bank	owns	about
new	stadiums	ranging	from	cleveland	to	san	antonio	and
<s>	the	philadelphia	and	cleveland	districts	for	example	reported
mcdonald	&	co.	in	cleveland	said	<unk>	's	unanticipated
<unk>	tumor	at	the	cleveland	clinic	in	N	<s>
at	mcdonald	&	co.	cleveland	<s>	<s>	<s>	<s>

(try it yourself w/ `kwic.py`)

# Count-based Methods

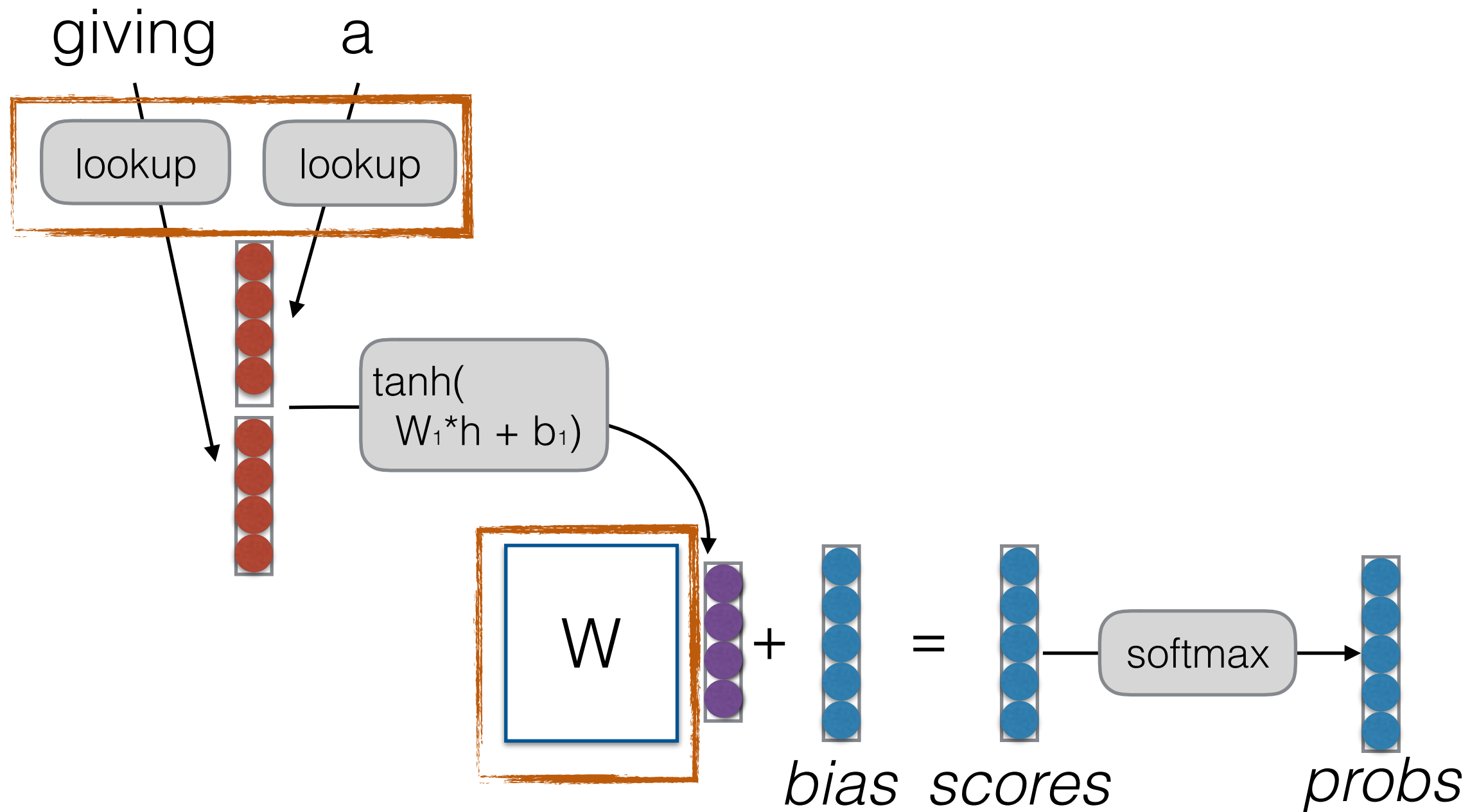
- Create a word-context count matrix
  - **Count** the number of co-occurrences of word/context, with rows as word, columns as contexts
  - Maybe **weight** with pointwise mutual information
  - Maybe **reduce dimensions** using SVD
- **Measure their closeness** using cosine similarity (or generalized Jaccard similarity, others)

# Prediction-based Methods

(See Goldberg 10.4.2)

- Instead, try to **predict** the words within a neural network
- Word embeddings are the byproduct

# Word Embeddings from Language Models



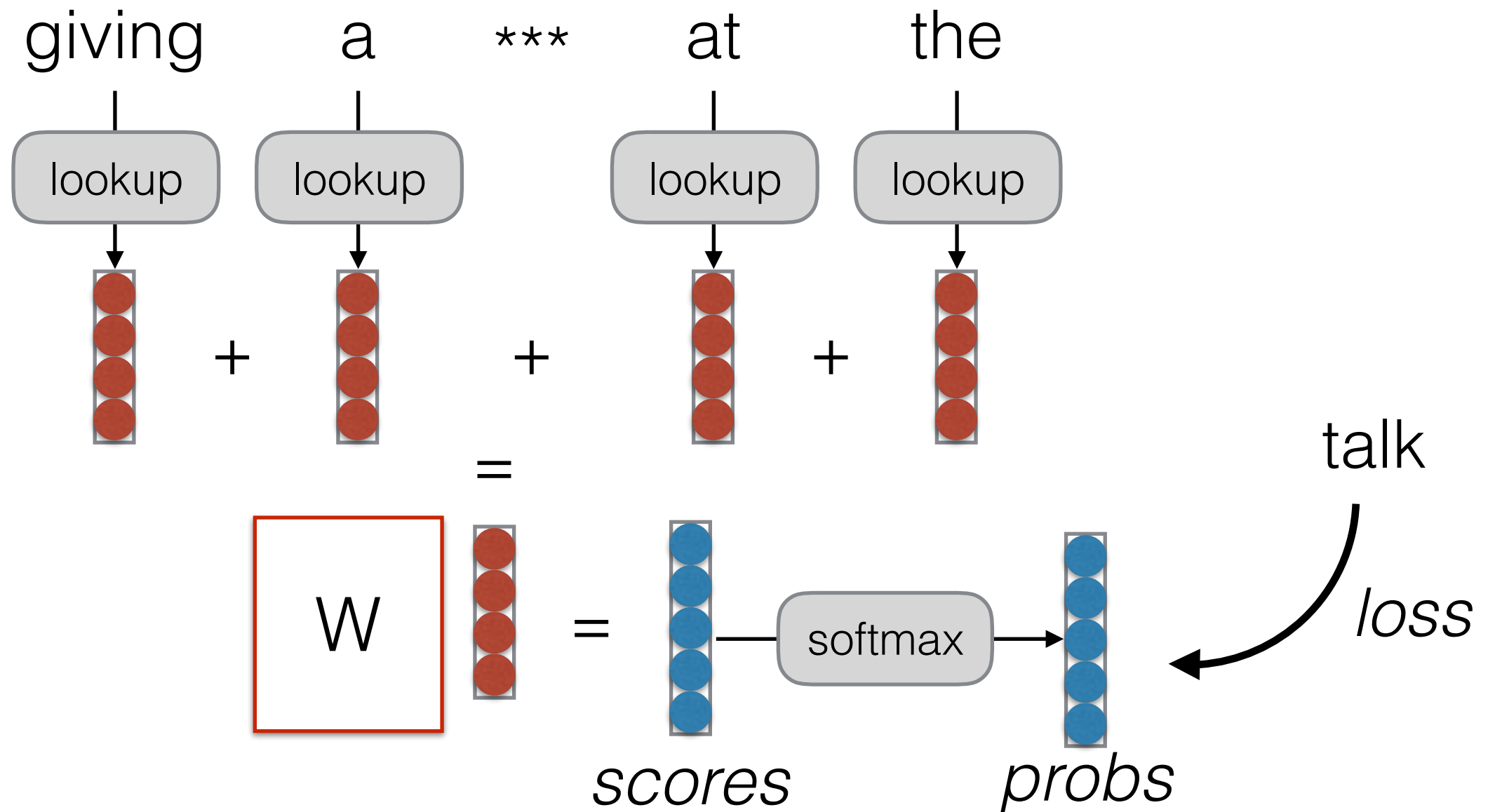
# Context Window Methods

- If we don't need to calculate the probability of the sentence, other methods possible!
- These can move **closer to the contexts used in count-based methods**
- These drive word2vec, etc.

# CBOW

(Mikolov et al. 2013)

- Predict word based on sum of surrounding embeddings



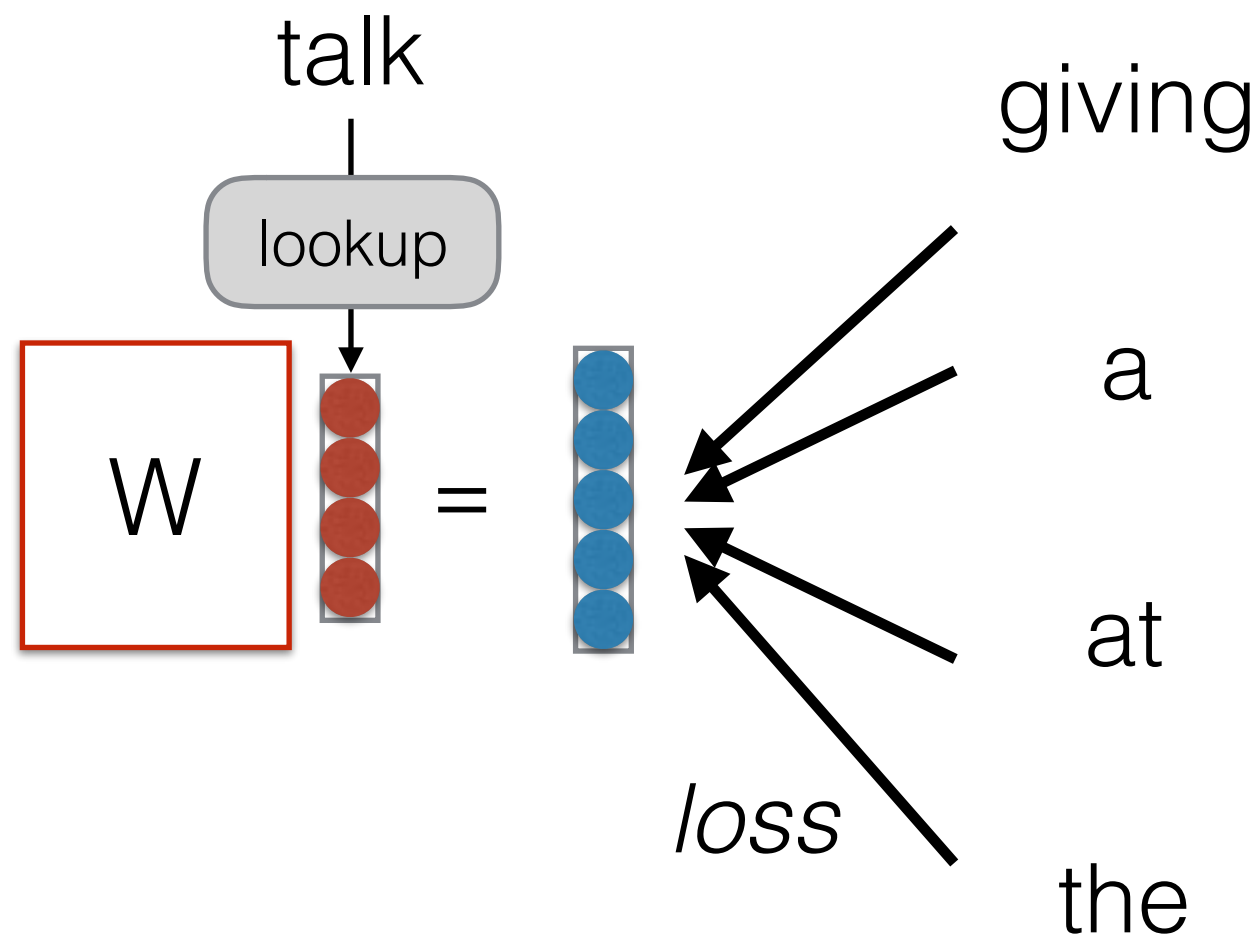
Let's Try it Out!

`wordemb-cbow.py`

# Skip-gram

(Mikolov et al. 2013)

- Predict each word in the context given the word





Let's Try it Out!

`wordemb-skipgram.py`

# Count-based and Prediction-based Methods

- Strong connection between count-based methods and prediction-based methods (Levy and Goldberg 2014)
- Skip-gram objective is equivalent to matrix factorization with PMI and discount for number of samples  $k$  (sampling covered next time)

$$M_{w,c} = \text{PMI}(w, c) - \log(k)$$

# GloVe (Pennington et al. 2014)

- A matrix factorization approach motivated by ratios of  $P(\text{word} | \text{context})$  probabilities

**Why?**

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

- Nice derivation from start to final loss function that satisfies desiderata

**Start:**

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

Meaningful in linear space  
(differences, dot products)  
Word/context invariance  
Robust to low-freq. ctxts.

**End:**

$$J = \sum_{i,j=1}^v f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

# What Contexts?

- Context has a large effect!
- **Small context window:** more syntax-based embeddings
- **Large context window:** more semantics-based, topical embeddings
- **Context based on syntax:** more functional, w/ words with same inflection grouped

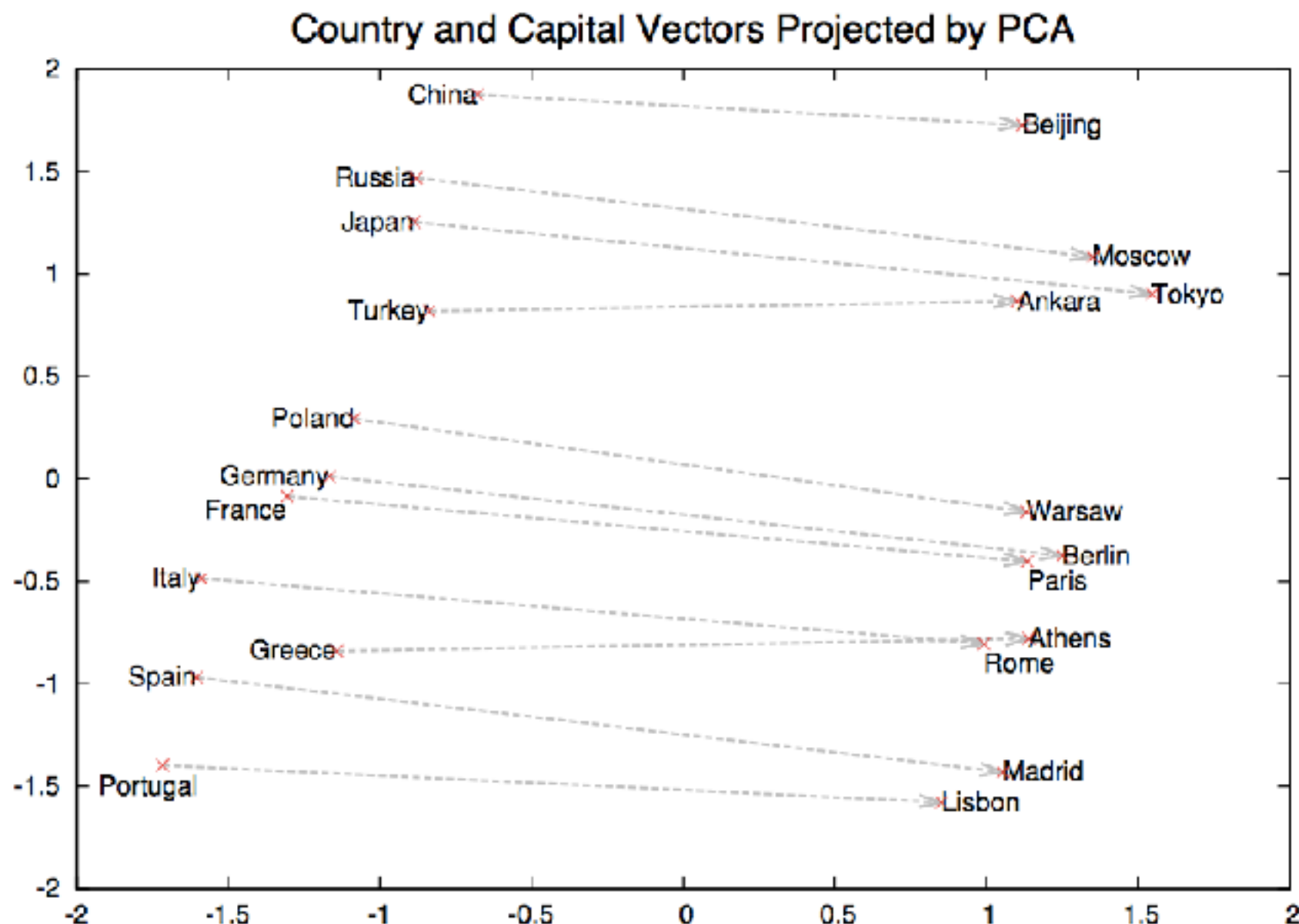
# Evaluating Embeddings

# Types of Evaluation

- Intrinsic vs. Extrinsic
  - **Intrinsic:** How good is it based on its features?
  - **Extrinsic:** How useful is it downstream?
- Qualitative vs. Quantitative
  - **Qualitative:** Examine the characteristics of examples.
  - **Quantitative:** Calculate statistics

# Visualization of Embeddings

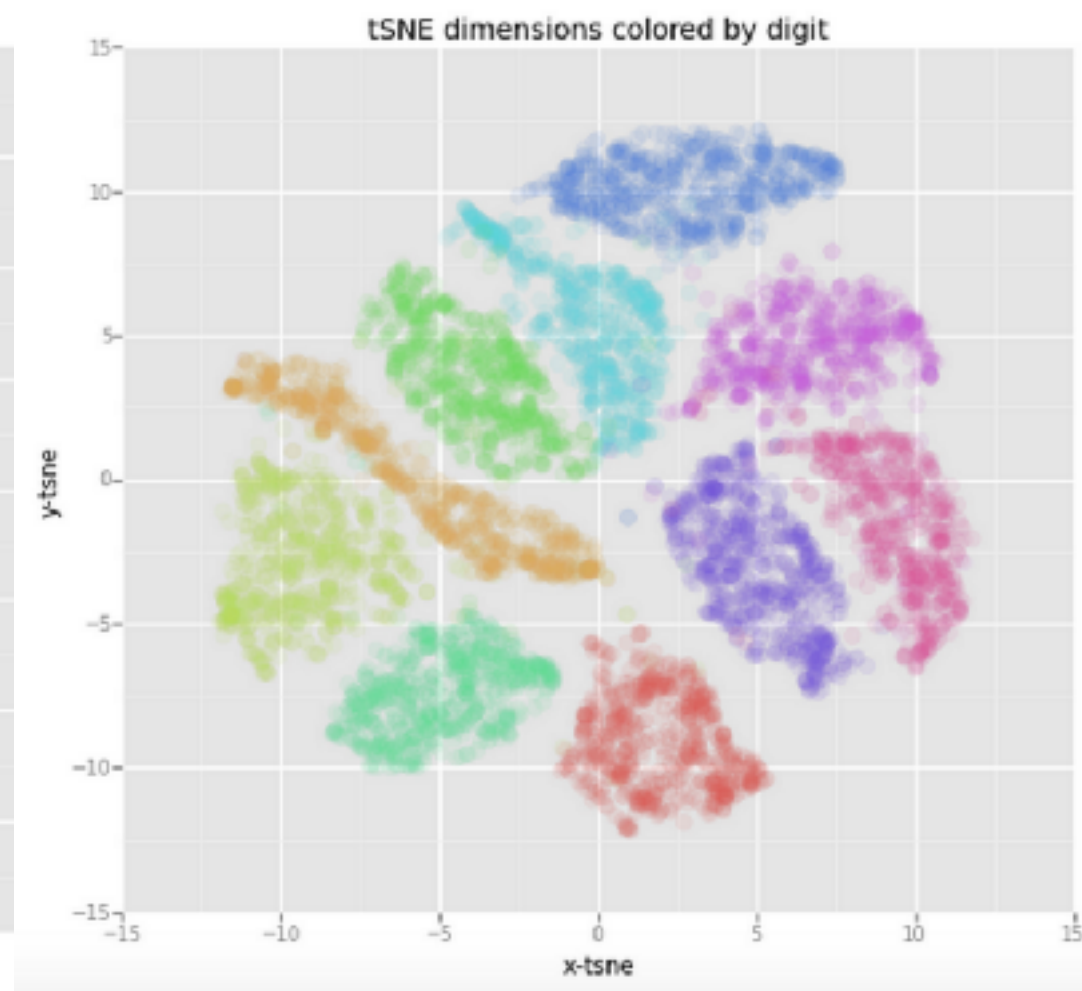
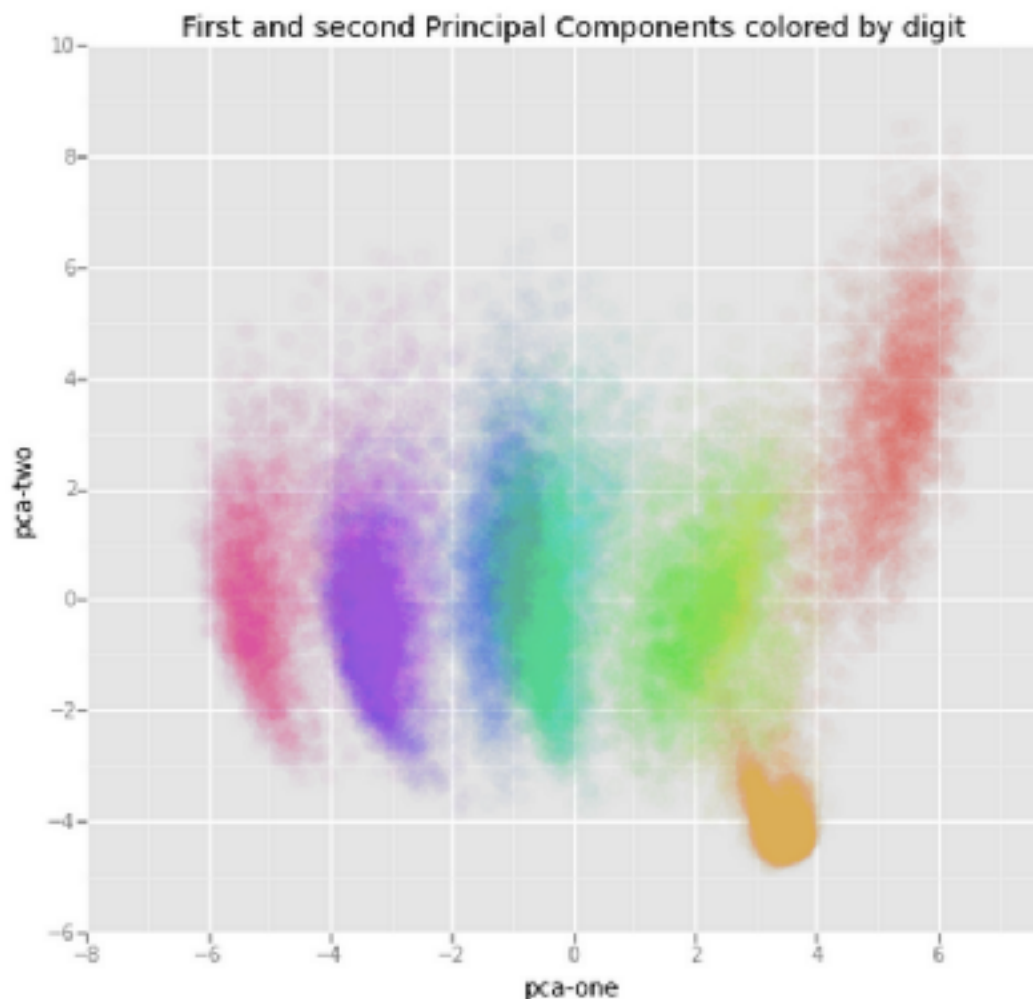
- Reduce high-dimensional embeddings into 2/3D for visualization (e.g. Mikolov et al. 2013)



# Non-linear Projection

- Non-linear projections group things that are close in high-dimensional space
- e.g. SNE/t-SNE (van der Maaten and Hinton 2008) group things that give each other a high probability according to a Gaussian

PCA



t-SNE

(Image credit: Derksen 2016)

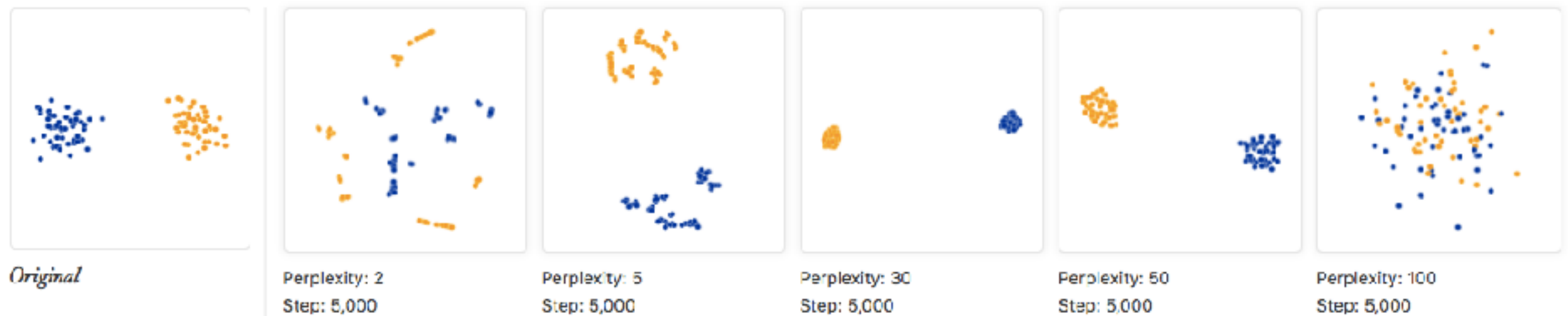


Let's Try it Out!

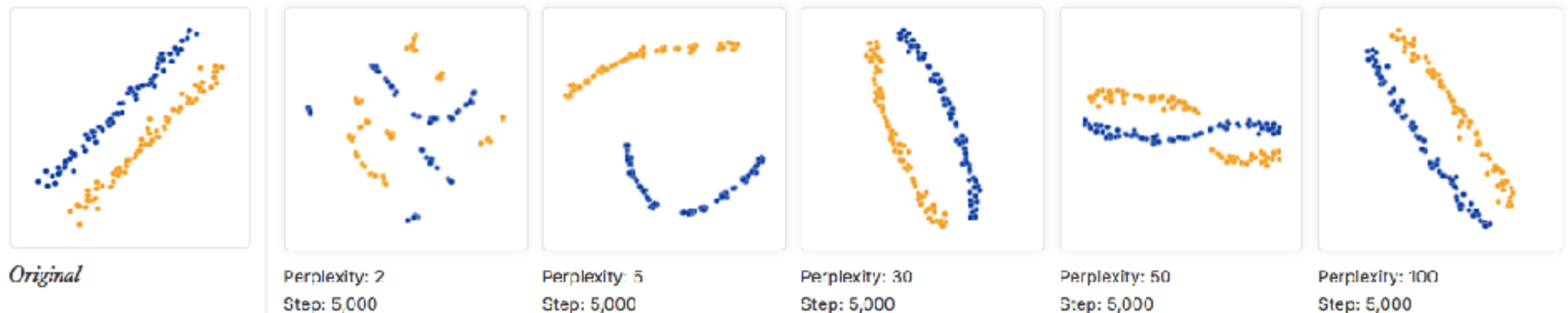
`wordemb-vis-tsne.py`

# t-SNE Visualization can be Misleading! (Wattenberg et al. 2016)

- Settings matter



- Linear correlations cannot be interpreted



# Intrinsic Evaluation of Embeddings

(categorization from Schnabel et al 2015)

- **Relatedness:** The correlation btw. embedding cosine similarity and human eval of similarity?
- **Analogy:** Find  $x$  for “*a is to b, as x is to y*”.
- **Categorization:** Create clusters based on the embeddings, and measure purity of clusters.
- **Selectional Preference:** Determine whether a noun is a typical argument of a verb.

# Extrinsic Evaluation: Using Word Embeddings in Systems

- **Initialize** w/ the embeddings
- **Concatenate** pre-trained embeddings with learned embeddings
- Latter is more expressive, but leads to increase in model parameters

# How Do I Choose Embeddings?

- No one-size-fits-all embedding (Schnabel et al 2015)

	relatedness						categorization			sel. prefs		analogy			
	rg	ws	wss	wsr	men	toefl	ap	essli	batt.	up	mcrae	an	ansyn	ansem	average
CBOW	<b>74.0</b>	<b>64.0</b>	<b>71.5</b>	<b>56.5</b>	<b>70.7</b>	66.7	<b>65.9</b>	<b>70.5</b>	<b>85.2</b>	24.1	13.9	<b>52.2</b>	<b>47.8</b>	<b>57.6</b>	<b>58.6</b>
GloVe	63.7	54.8	65.8	49.6	64.6	<b>69.4</b>	64.1	65.9	77.8	27.0	<b>18.4</b>	42.2	44.2	39.7	53.4
TSCCA	57.8	54.4	64.7	43.3	56.7	58.3	57.5	<b>70.5</b>	64.2	<b>31.0</b>	14.0				
C&W	48.1	49.8	60.7	40.1	57.5	66.7	60.6	61.4	80.2	28.3	16.0				
H-PCA	19.8	32.9	43.6	15.1	21.3	54.2	34.1	50.0	42.0	-2.5	3.0				
Rand. Proj.	17.1	19.5	24.9	16.1	11.3	51.4	21.9	38.6	29.6	-8.5	1.0				

	dev	test	<i>p</i> -value
Baseline	94.18	93.78	0.000
Rand. Proj.	94.33	93.90	0.006
GloVe	94.28	93.93	0.015
H-PCA	94.48	93.96	0.029
C&W	<b>94.53</b>	<b>94.12</b>	
CBOW	94.32	93.93	0.012
TSCCA	<b>94.53</b>	94.09	0.357

Table 1: Results on absolute intrinsic evaluation. The best result for each task is in bold. The second row contains the names of the corresponding datasets.

Table 4: F1 chunking results using different word embeddings as features. The *p*-values are with respect to the best performing method.

- Be aware, and use the best one for the task

# When are Pre-trained Embeddings Useful?

- Basically, when training data is insufficient
- **Very useful:** tagging, parsing, text classification
- **Less useful:** machine translation
- **Basically not useful:** language modeling

# Improving Embeddings

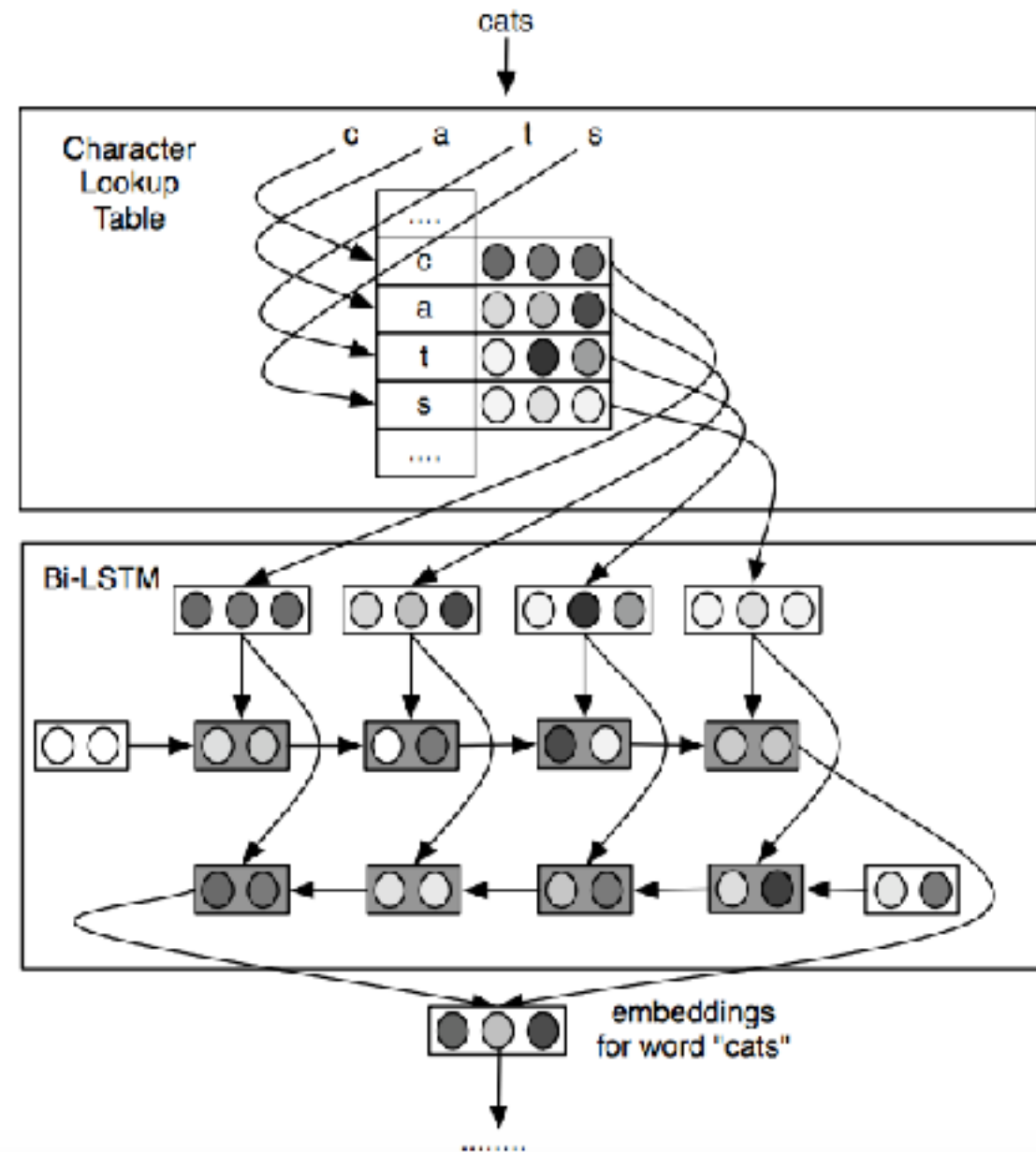
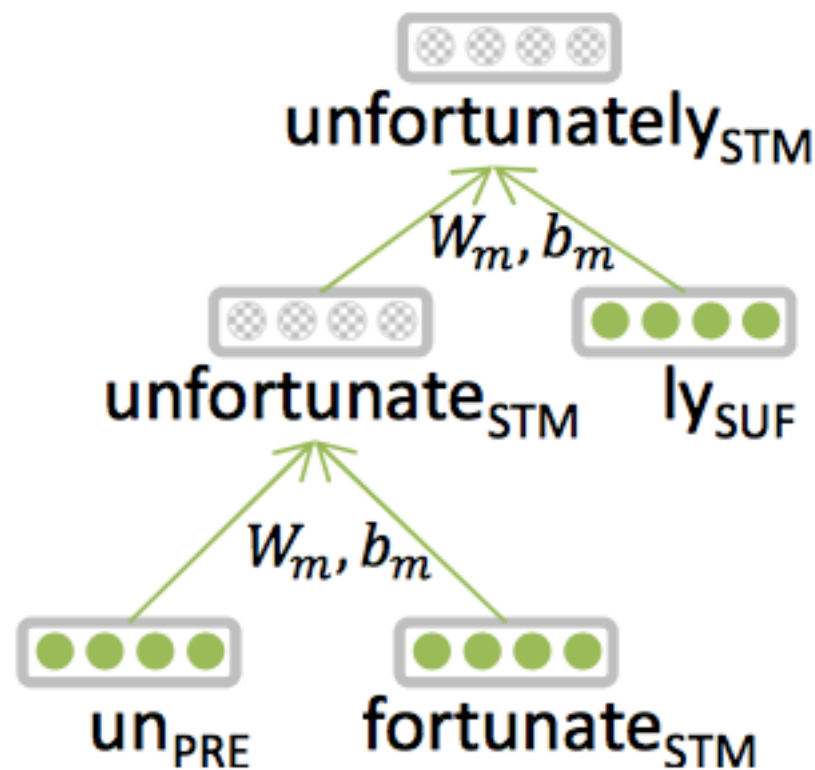
# Limitations of Embeddings

- Sensitive to **superficial differences** (dog/dogs)
- **Insensitive to context** (financial bank, bank of a river)
- **Not necessarily coordinated** with knowledge or across languages
- **Not interpretable**
- Can **encode bias** (encode stereotypical gender roles, racial biases)



# Sub-word Embeddings (1)

- Can capture sub-word regularities
- Morpheme-based (Luong et al. 2013)
- Character-based (Ling et al. 2015)



# Sub-word Embeddings (2)

- **Bag of character n-grams** used to represent word (Wieting et al. 2016)

where

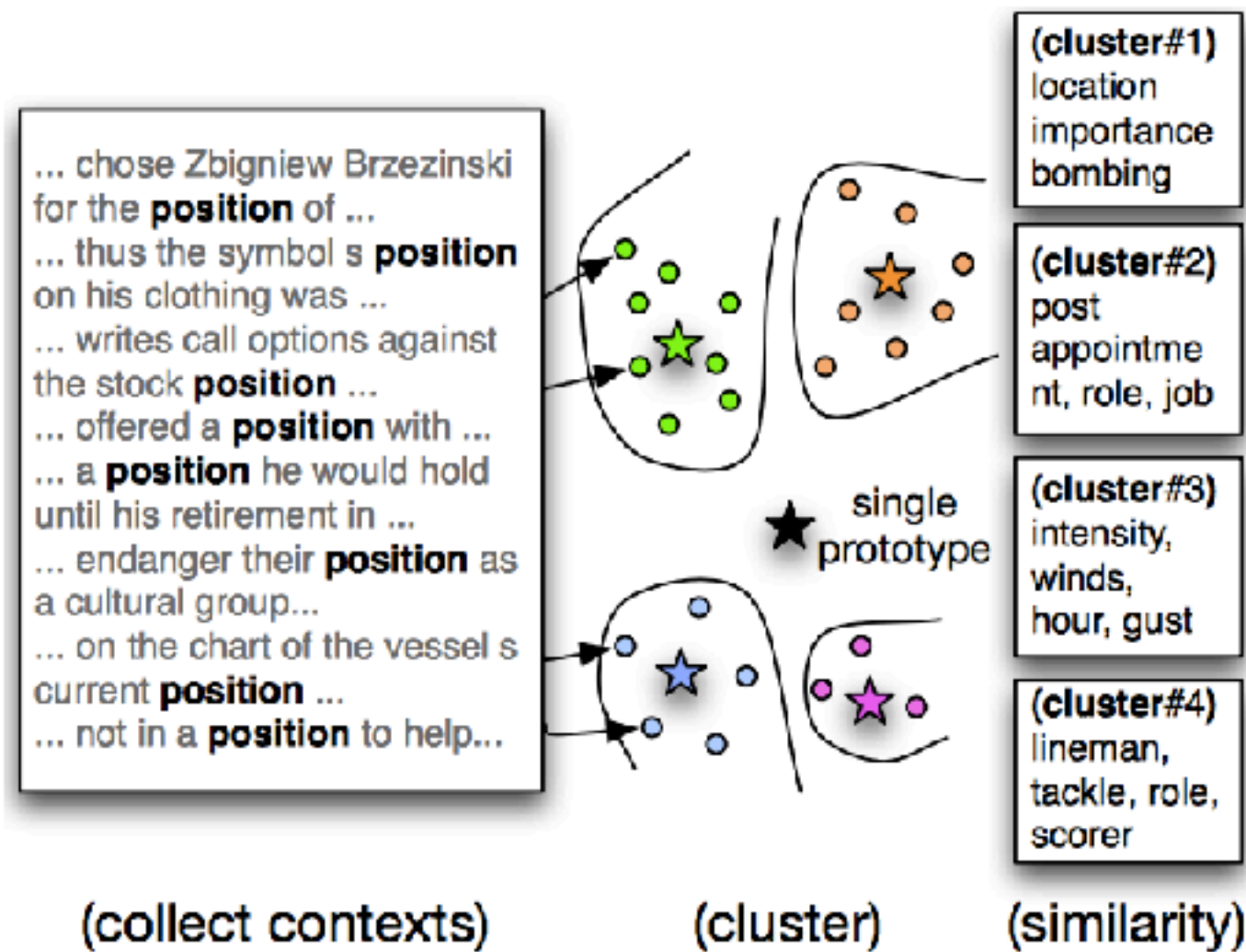


<wh, whe, her, ere, re>

- Use n-grams from 3-6 plus word itself
- Used in the “fasttext” toolkit (Bojanowski et al. 2017)

# Multi-prototype Embeddings

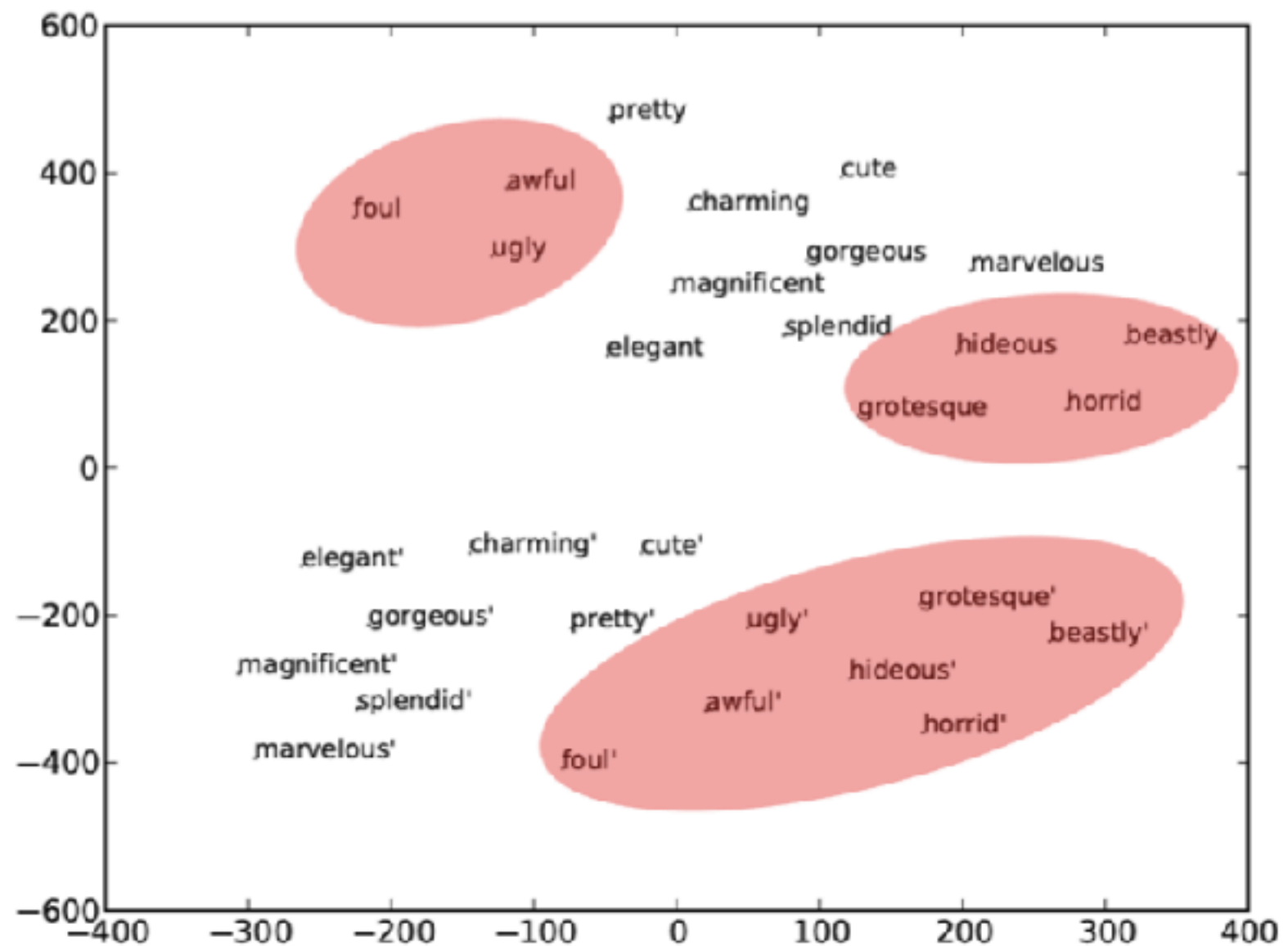
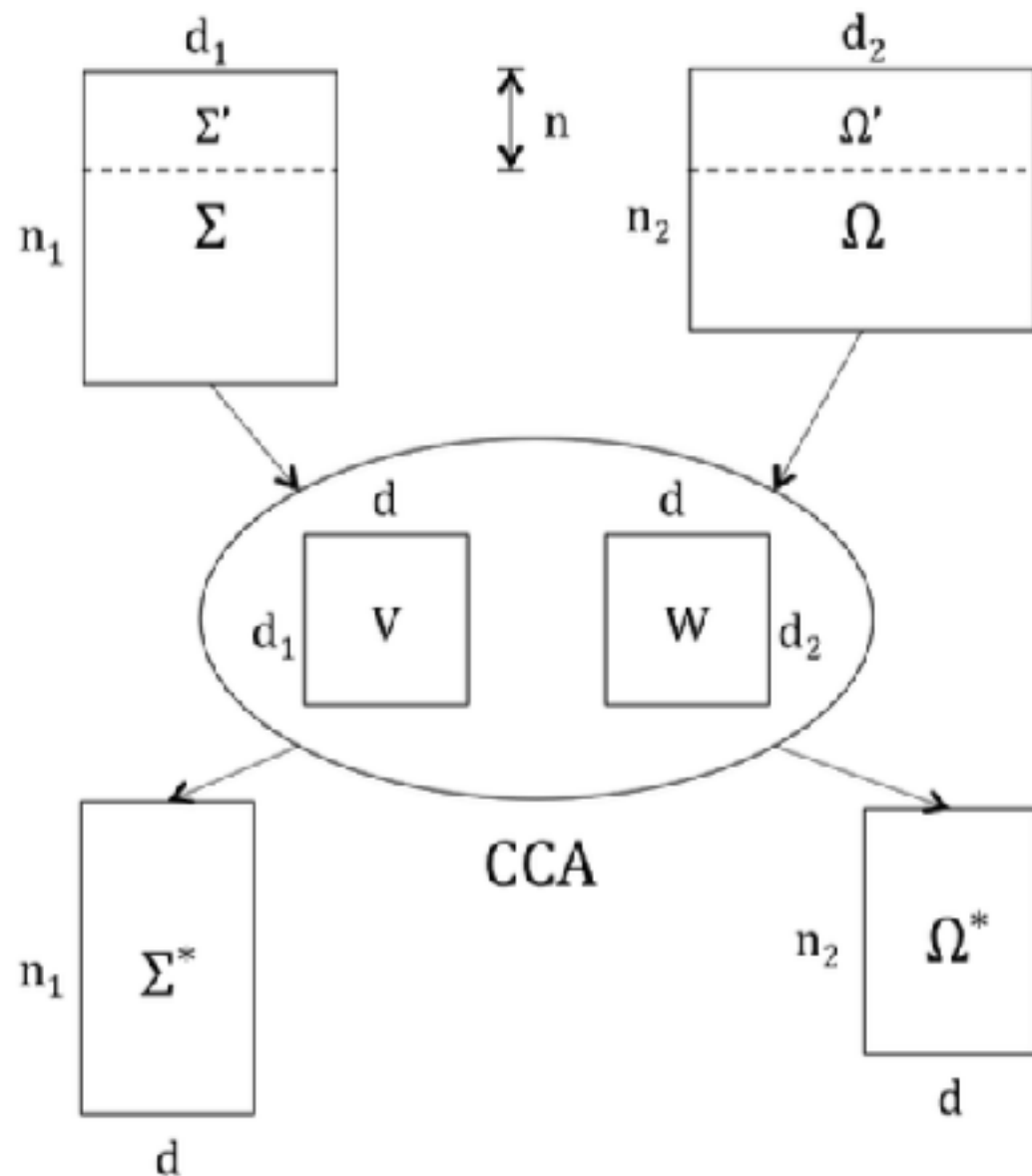
- Simple idea, words with multiple meanings should have different embeddings (Reisinger and Mooney 2010)



- Non-parametric estimation (Neelakantan et al. 2014) also possible

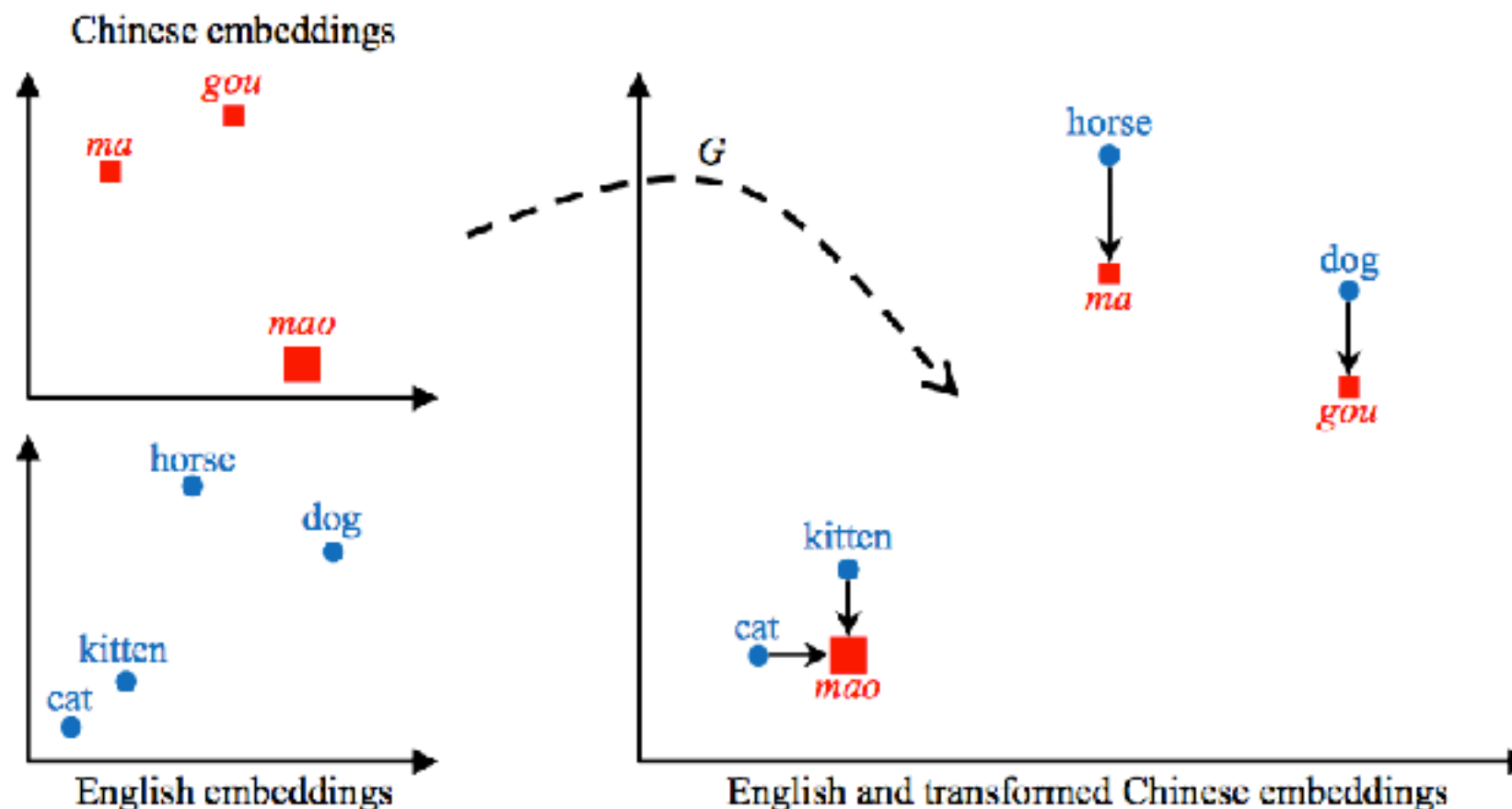
# Multilingual Coordination of Embeddings (Faruqui et al. 2014)

- We have word embeddings in two languages, and want them to match



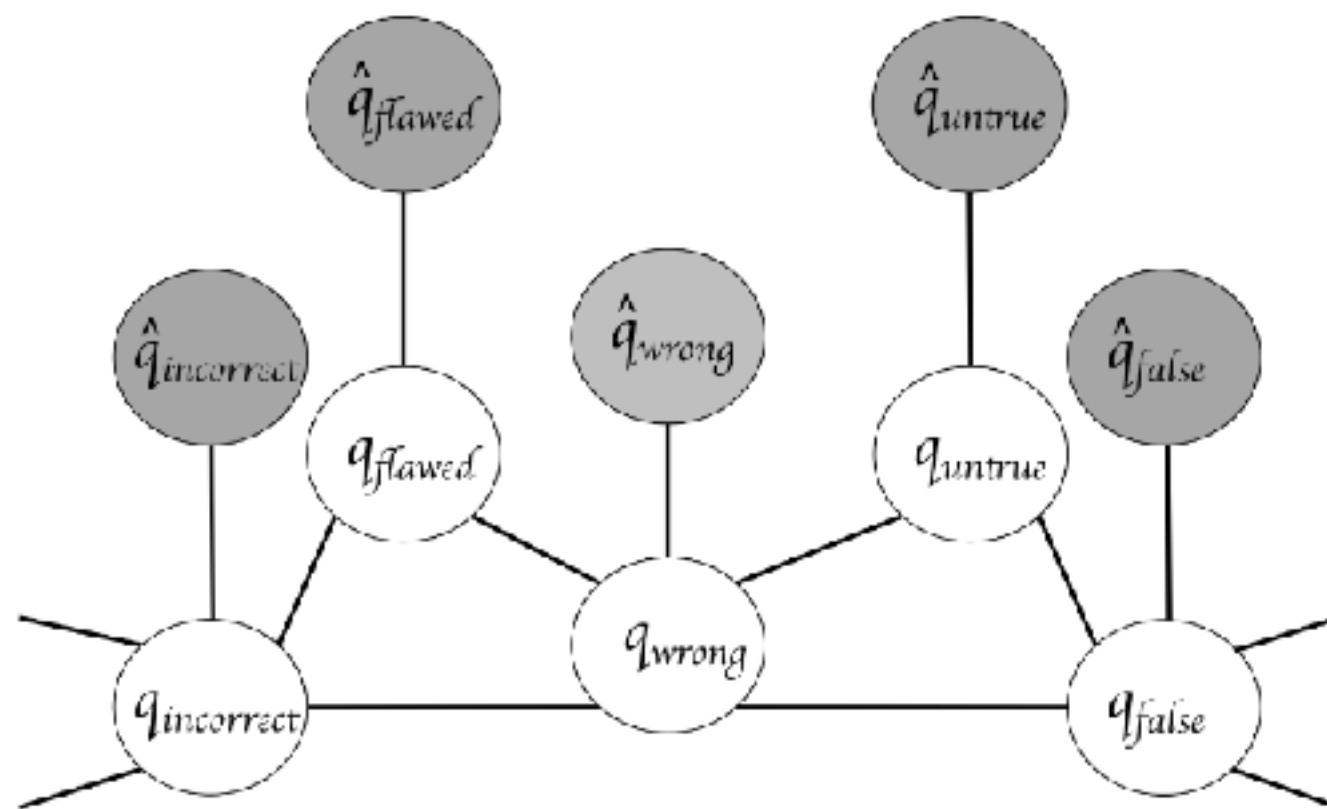
# Unsupervised Coordination of Embeddings

- In fact we can do it with no dictionary at all!
  - Just use identical words, e.g. the digits (Artexte et al. 2017)
  - Or just match distributions (Zhang et al. 2017)



# Retrofitting of Embeddings to Existing Lexicons

- We have an existing lexicon like WordNet, and would like our vectors to match (Faruqui et al. 2015)



$$\Psi(Q) = \sum_{i=1}^n \left[ \alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

# Sparse Embeddings

- Each dimension of a word embedding is not interpretable
- Solution: add a sparsity constraint to increase the information content of non-zero dimensions for each word (e.g. Murphy et al. 2012)

Model	Top 5 Words (per dimension)
$SVD_{300}$	well, long, if, year, watch plan, engine, e, rock, very get, no, features, music, via features, by, links, free, down works, sound, video, building, section
$NNSE_{1000}$	inhibitor, inhibitors, antagonists, receptors, inhibition bristol, thames, southampton, brighton, poole delhi, india, bombay, chennai, madras pundits, forecasters, proponents, commentators, observers nosy, averse, leery, unsympathetic, snotty



# De-biasing Word Embeddings (Bolukbasi et al. 2016)

- Word embeddings reflect bias in statistics

<b>Extreme <i>she</i></b>	<b>Extreme <i>he</i></b>		<b>Gender stereotype <i>she-he</i> analogies</b>	
1. homemaker	1. maestro	sewing-carpentry	registered nurse-physician	housewife-shopkeeper
2. nurse	2. skipper	nurse-surgeon	interior designer-architect	softball-baseball
3. receptionist	3. protege	blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
4. librarian	4. philosopher	giggle-chuckle	vocalist-guitarist	petite-lanky
5. socialite	5. captain	sassy-snappy	diva-superstar	charming-affable
6. hairdresser	6. architect	volleyball-football	cupcakes-pizzas	lovely-brilliant
7. nanny	7. financier			
8. bookkeeper	8. warrior	queen-king	<b>Gender appropriate <i>she-he</i> analogies</b>	
9. stylist	9. broadcaster	waitress-waiter	sister-brother	mother-father
10. housekeeper	10. magician		ovarian cancer-prostate cancer	convent-monastery

- Identify pairs to “neutralize”, find the direction of the trait to neutralize, and ensure that they are neutral in that direction



Questions?