

# Word Embeddings

Word2vec, skip-grams, Diachronic Embeddings, Applications.

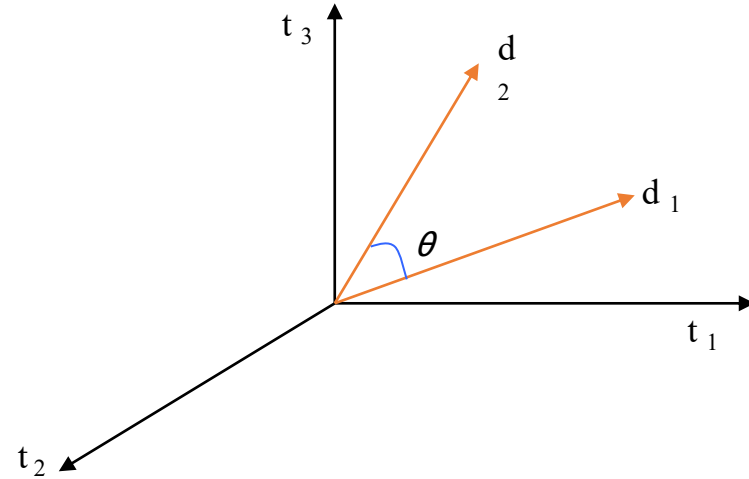
## Information Retrieval

# How to represent a word?

dog	1	[1 0 0 0 0 0 0 0 0 0]
cat	2	[0 1 0 0 0 0 0 0 0 0]
person	3	[0 0 1 0 0 0 0 0 0 0]

# Vector space model

- In the vector space model, each dimension corresponds to a term.
- The dimensionality  $V$  of the space corresponds to the size of the vocabulary.
- Each word is represented by a  $V$  dimensional vector, where only the dimension corresponding to that word is non-zero.
- Hence, each document is represented by the frequency of its terms.



# How to represent a word?

dog	1	[1 0 0 0 0 0 0 0 0 0]
cat	2	[0 1 0 0 0 0 0 0 0 0]
person	3	[0 0 1 0 0 0 0 0 0 0]

- **Problem:** distance between words using one-hot encodings always the same
- **Idea:** Instead of one-hot-encoding use a histogram of commonly co-occurring words.

# Distributional Semantics

dog	[5	5	0	5	0	0	5	5	0	2	...]
cat	[5	4	1	4	2	0	3	4	0	3	...]
person	[5	5	1	5	0	2	5	5	0	0	...]
	food	walks	window	runs	mouse	invented	legs	sleeps	mirror	tail	...

→  
This vocabulary can be extremely large

# Distributional Semantics

- How similar is **pizza** to **pasta**?
- How related is **pizza** to **Italy**?
- **Representing words as vectors** allows easy computation of similarity and relatedness.

# Approaches for Representing Words

## **Distributional Semantics (*Count*)**

- Used since the 90's
- Sparse word-context PMI/PPMI matrix
- Decomposed with SVD

## **Word Embeddings (*Predict*)**

- Inspired by deep learning
- `word2vec` (*Mikolov et al., 2013*)
- GloVe (*Pennington et al., 2014*)

Underlying Theory: **The Distributional Hypothesis** (*Harris, '54; Firth, '57*)

“Similar words occur in similar contexts”

# Approaches for Representing Words

Both approaches:

- Rely on the **same linguistic theory**
- Use the **same data**
- Are **mathematically related**
  - “Neural Word Embedding as Implicit Matrix Factorization” (NIPS 2014)
- How come word embeddings are so much better?
  - “Don’t Count, Predict!” (Baroni et al., ACL 2014)
- **More than meets the eye...**



# Word Embeddings with Word2Vec

## Algorithms

*(objective + training method)*

- Skip Grams + Negative Sampling
- CBOW + Hierarchical Softmax
- Noise Contrastive Estimation
- GloVe
- ...

## Hyperparameters

*(preprocessing, smoothing, etc.)*

- Subsampling
- Dynamic Context Windows
- Context Distribution Smoothing
- Adding Context Vectors
- ...

# What is word2vec?

- word2vec is **not** a single algorithm
- It is a **software package** for representing words as vectors, containing:
  - Two distinct models
    - CBoW
    - Skip-Gram
  - Various training methods
    - Negative Sampling
    - Hierarchical Softmax
  - A rich preprocessing pipeline
    - Dynamic Context Windows
    - Subsampling
    - Deleting Rare Words

# What is word2vec?

- word2vec is **not** a single algorithm
- It is a **software package** for representing words as vectors, containing:
  - Two distinct models
    - CBoW
    - Skip-Gram (SG)
  - Various training methods
    - Negative Sampling (NS)
    - Hierarchical Softmax
  - A rich preprocessing pipeline
    - Dynamic Context Windows (DCW)
    - Subsampling
    - Deleting Rare Words

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little **cat** hiding in the tree.

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a furry little cat hiding in the tree.

words

cat

cat

cat

cat

...

contexts

furry

little

hiding

in

...

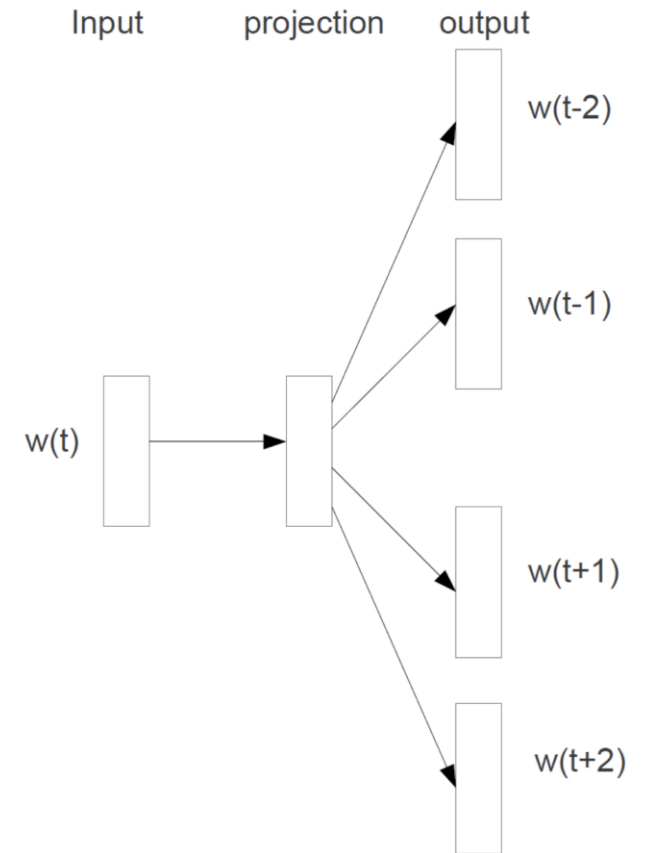
$D$  (data)

# Skip-Grams with Negative Sampling (SGNS)

Marco saw a **furry little** **cat** **hiding in** the tree.

- Word2vec models the distribution of words and context words.
- The model will maximize the log-likelihood:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$



# Softmax: words vs context words

- The  $p(w_t|w_{t-1})$  is formalized as the softmax:

$$p(w_O|w_I) = \frac{\exp(v'_{w_O}{}^\top v_{w_I})}{\sum_{w=1}^W \exp(v'_w{}^\top v_{w_I})}$$

where each word is represented by a vector  $v_{w_*} = [v_{w_*} \quad \dots \quad v_{w_*}]$ , thus rendering the argument of the softmax function:

Marco saw a furry little cat hiding in the tree.

$$v_{w_O}^T \cdot v_{w_I} = [v_{w_O,1} \quad \dots \quad v_{w_O,n}] \cdot \begin{bmatrix} v_{w_I,1} \\ \dots \\ v_{w_I,n} \end{bmatrix}$$

$$p(\text{furry}|\text{wampimuk}) = \frac{\exp(v_{\text{furry}_O}^T \cdot v_{\text{cat}_I})}{\sum_{w_I} \exp(v_{\text{furry}_O}^T \cdot v_{*I})}$$



# Stochastic Gradient Descent

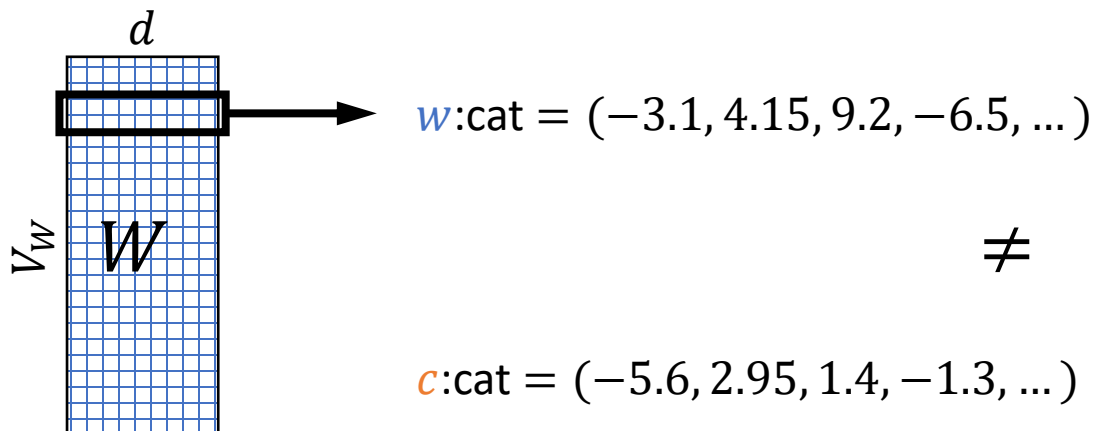
- But Corpus may have 40B tokens and windows
- You would wait a very long time before making a single update!
- **Very** bad idea for pretty much all neural nets!
- Instead: We will update parameters after each window  $t$   
→ Stochastic gradient descent (SGD)

$$v_{w_o}^{new} = v_{w_o}^{old} - \alpha \nabla_{v_{w_o}} p(w_o | w_I, D)$$

$$v_{w_I}^{new} = v_{w_I}^{old} - \alpha \nabla_{v_{w_I}} p(w_o | w_I, D)$$

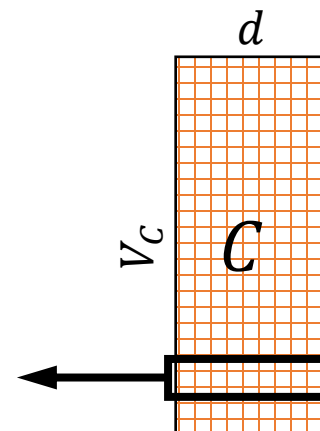
# Word vectors

- SGNS finds a vector  $\vec{w}$  for each word  $w$  in our vocabulary  $V_W$
- Each such vector has  $d$  latent dimensions (e.g.  $d = 100$ )
- Effectively, it learns a matrix  $W$  whose rows represent  $V_W$
- **Key point:** it also learns a similar auxiliary matrix  $C$  of context vectors
- In fact, each word has two embeddings



$\neq$

$c:\text{cat} = (-5.6, 2.95, 1.4, -1.3, \dots)$



# Positive Samples + Negative Sampling

- **Maximize:**  $\sigma(\vec{w} \cdot \vec{c})$

- $c$  was **observed** with  $w$

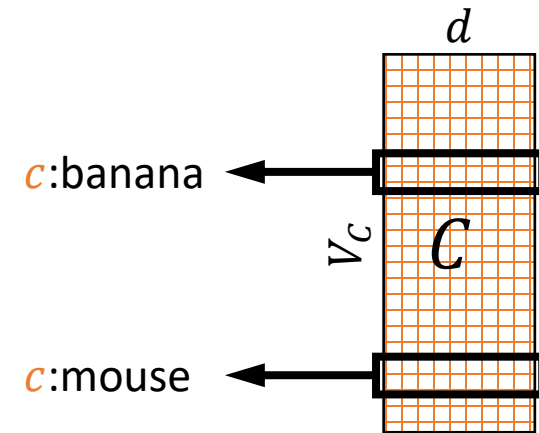
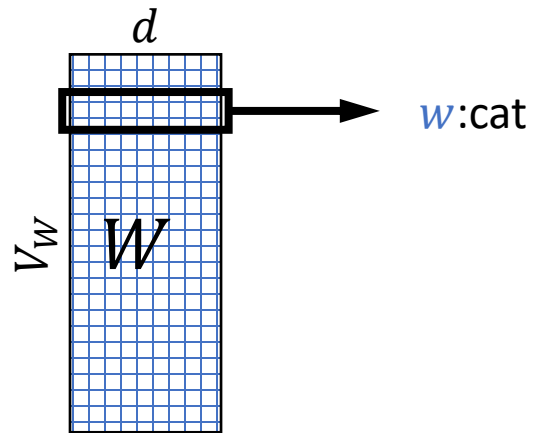
<u>words</u>	<u>contexts</u>
cat	furry
cat	little
cat	hiding
cat	in

- **Minimize:**  $\sigma(\vec{w} \cdot \vec{c}')$

- $c'$  was **hallucinated** with  $w$

<u>words</u>	<u>contexts</u>
cat	Australia
cat	cyber
cat	the
cat	1985

Exercise: How the  $\vec{w} \cdot \vec{c}$  be for these words?



# Hyperparameters

- **Preprocessing**

- Dynamic Context Windows
- Subsampling
- Deleting Rare Words

(word2vec)

- **Association Metric**

- Shifted PMI
- Context Distribution Smoothing

(SGNS)

# Dynamic Context Windows

Marco saw a furry little **cat** hiding in the tree.

# Dynamic Context Windows

saw a furry little cat hiding in the tree

# Dynamic Context Windows

saw a furry little cat hiding in the tree

word2vec:	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{4}{4}$		$\frac{4}{4}$	$\frac{3}{4}$	$\frac{2}{4}$	$\frac{1}{4}$
GloVe:	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$
Aggressive:	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{1}$		$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$

**The Word-Space Model (*Sahlgren, 2006*)**



# Context Distribution Smoothing

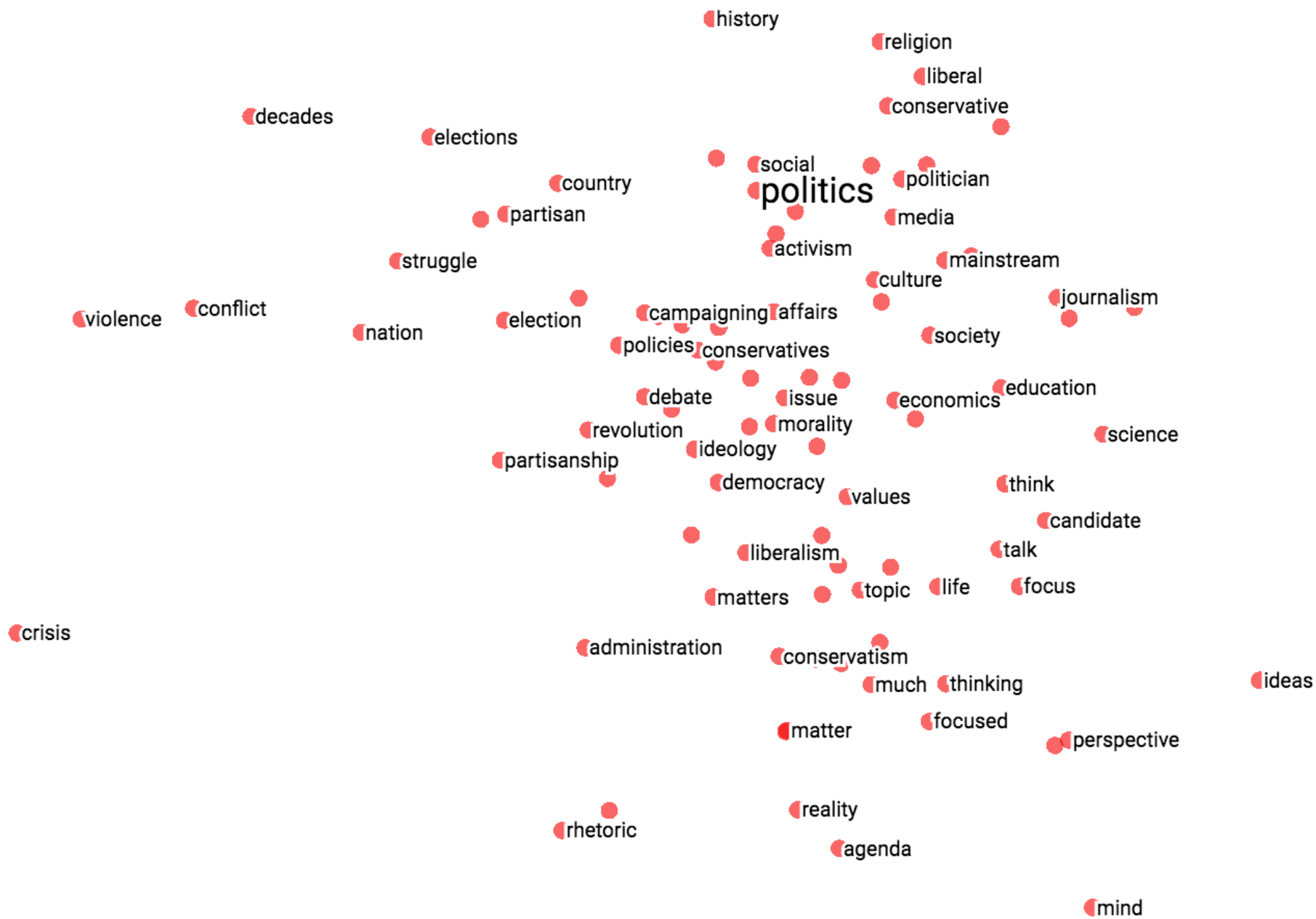
- SGNS samples  $c' \sim P$  to form **negative**  $(w, c')$  examples

- Our analysis assumes  $P$  is the unigram distribution  $P(c) = \frac{\#c}{\sum_{c' \in V_C} \#c'}$

- In practice, it's a **smoothed** unigram distribution

$$P^{0.75}(c) = \frac{(\#c)^{0.75}}{\sum_{c' \in V_C} (\#c')^{0.75}}$$

- This little change makes a big difference.



# Linear Relationships in word2vec

These representations are *very good* at encoding **similarity** and **dimensions of similarity**!

- Analogies testing dimensions of similarity can be solved quite well just by doing vector subtraction in the embedding space

Syntactically

- $X_{apple} - X_{apples} \approx X_{car} - X_{cars} \approx X_{family} - X_{families}$
- Similarly for verb and adjective morphological forms

Semantically (Semeval 2012 task 2)

- $X_{shirt} - X_{clothing} \approx X_{chair} - X_{furniture}$
- $X_{king} - X_{man} \approx X_{queen} - X_{woman}$

# Word Analogies

Test for linear relationships, examined by Mikolov et al.

a:b :: c:?



$$d = \arg \max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

man:woman :: king:?

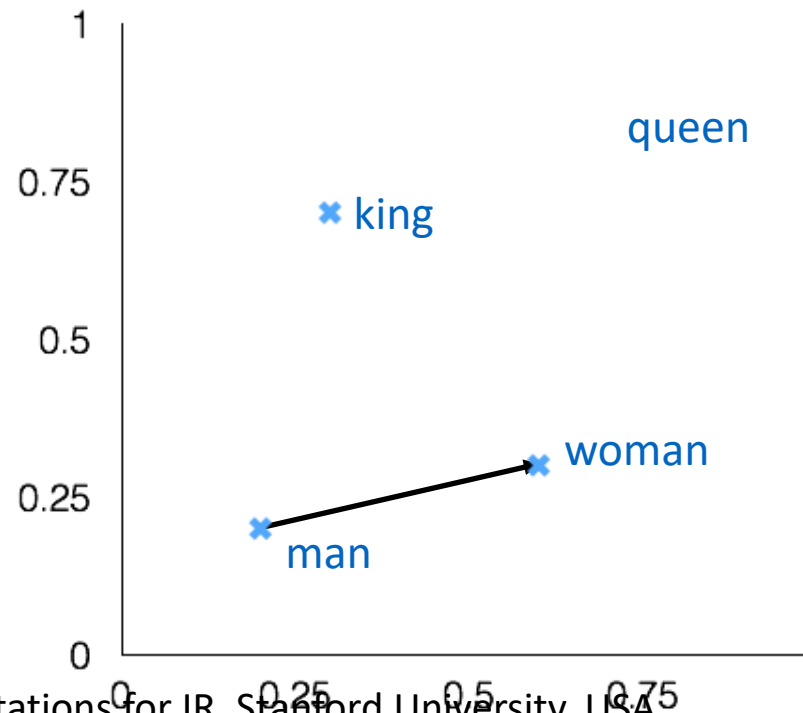
+ king [ 0.30 0.70 ]

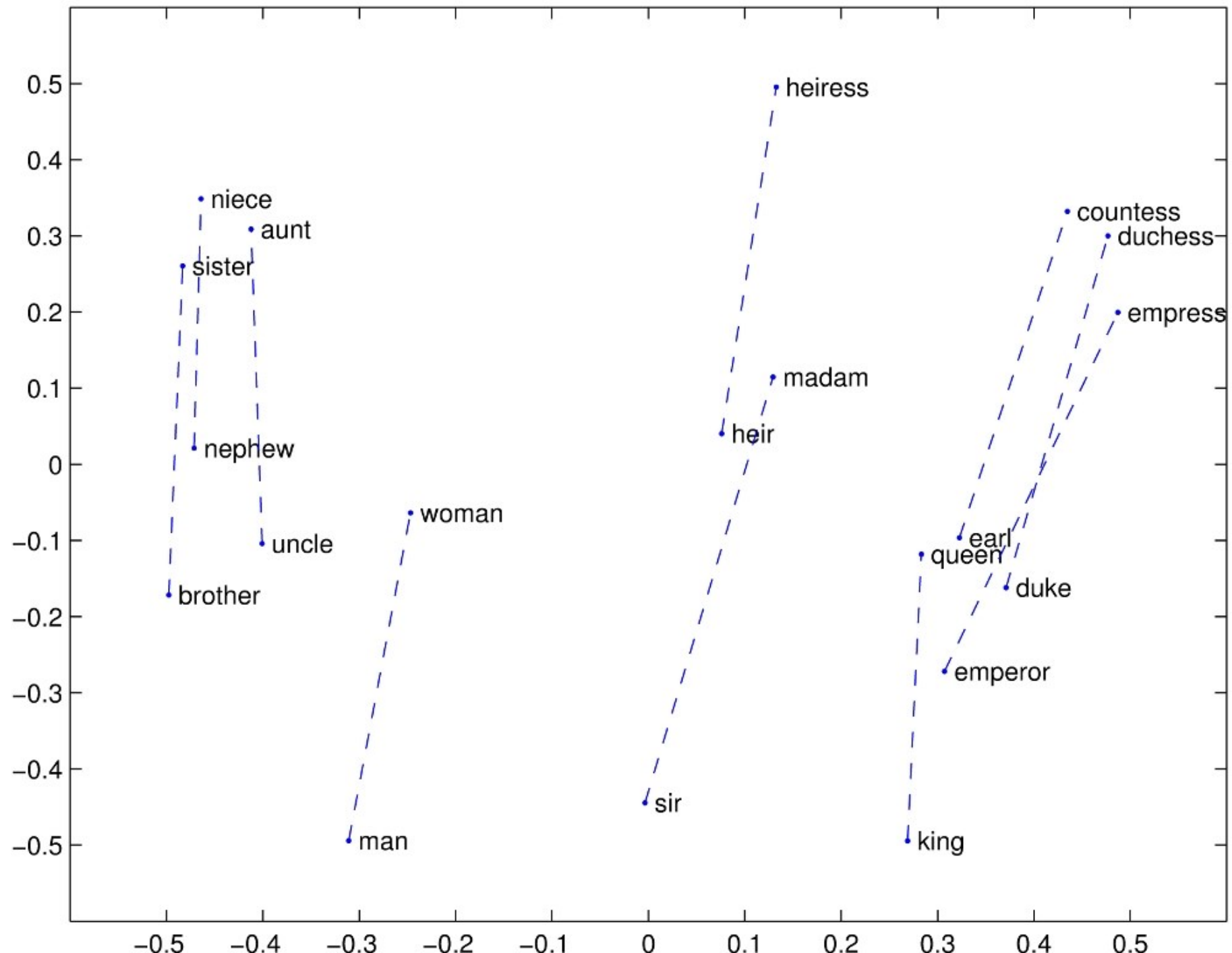
- man [ 0.20 0.20 ]

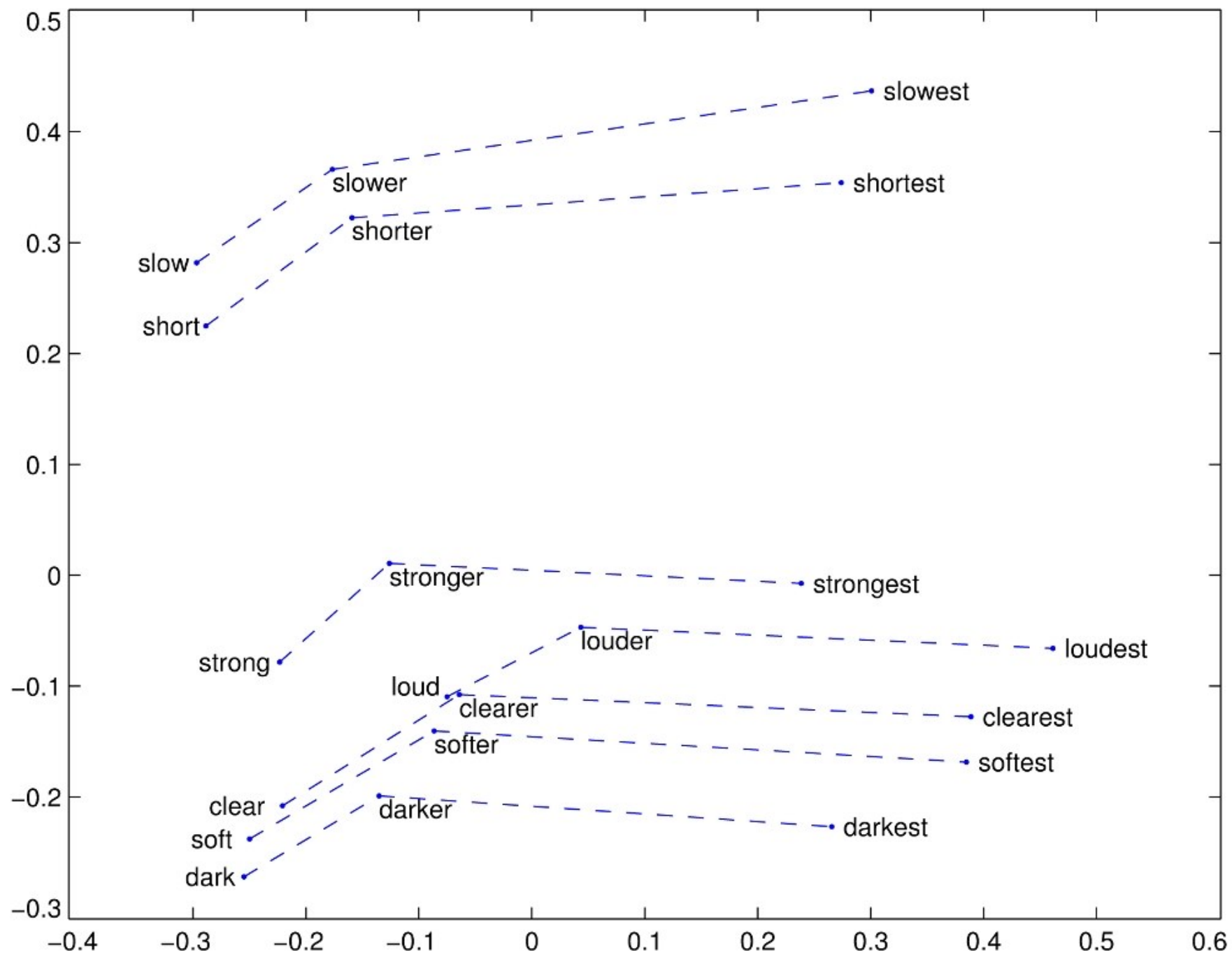
+ woman [ 0.60 0.30 ]

---

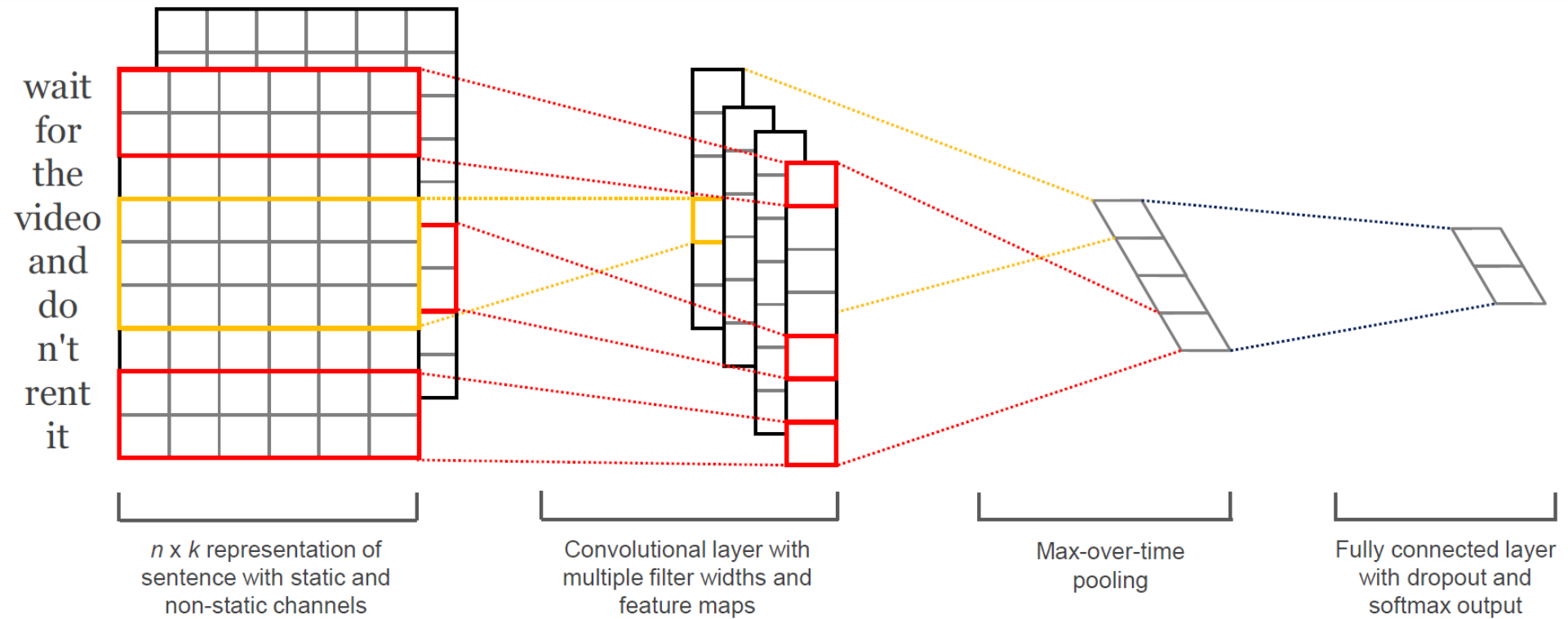
queen [ 0.70 0.80 ]







# Word embeddings and CNNs for sentence classification



# Results

- Movie reviews (MR)
- Sentiment analysis (SST-1, SST-2, Sub)
- TREC Question Answering (TREC)
- Customer reviews (CR)
- Opinion polarity (MPQA)

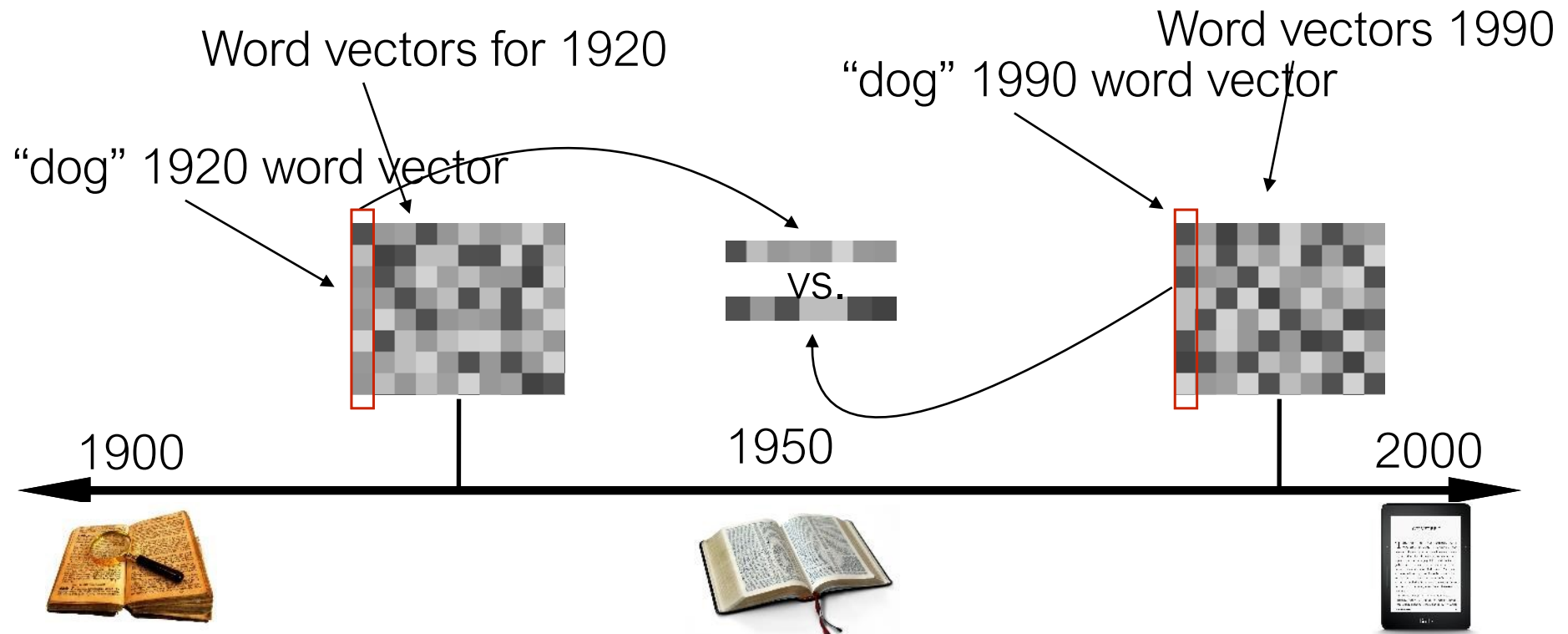
<b>Data</b>	$c$	$l$	$N$	$ V $	$ V_{pre} $	<i>Test</i>
MR	2	20	10662	18765	16448	CV
SST-1	5	18	11855	17836	16262	2210
SST-2	2	19	9613	16185	14838	1821
Subj	2	23	10000	21323	17913	CV
TREC	6	10	5952	9592	9125	500
CR	2	19	3775	5340	5046	CV
MPQA	2	3	10606	6246	6083	CV



# Results

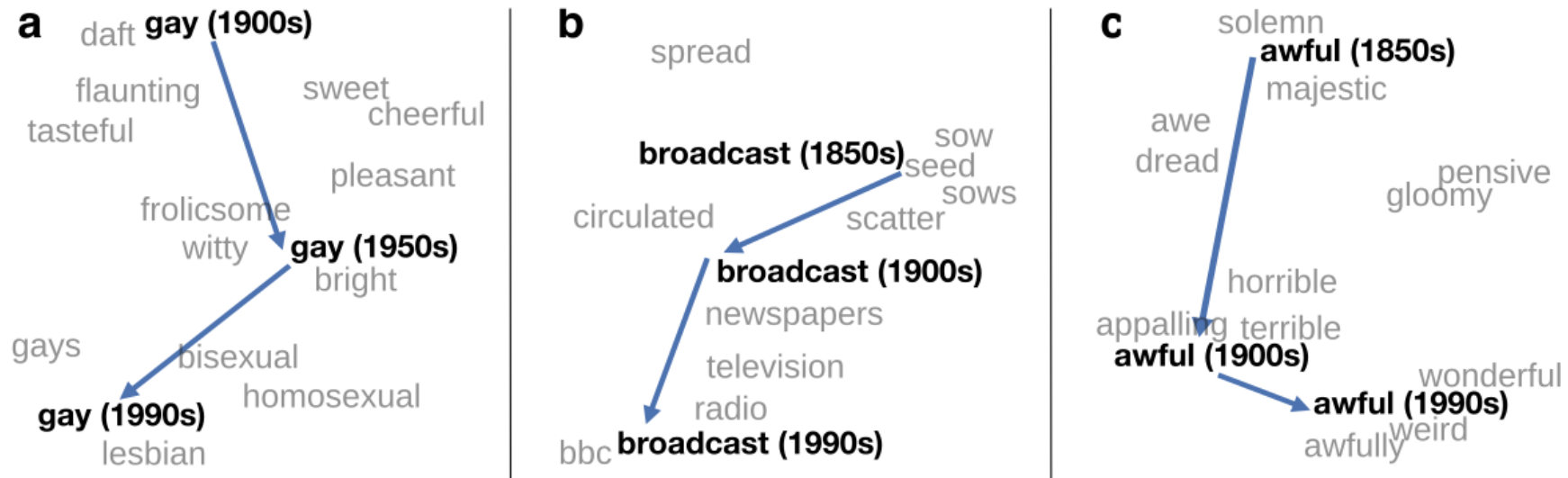
Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	<b>89.6</b>
CNN-non-static	<b>81.5</b>	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	<b>88.1</b>	93.2	92.2	<b>85.0</b>	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	—	—	—	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	—	—	—	—
RNTN (Socher et al., 2013)	—	45.7	85.4	—	—	—	—
DCNN (Kalchbrenner et al., 2014)	—	48.5	86.8	—	93.0	—	—
Paragraph-Vec (Le and Mikolov, 2014)	—	<b>48.7</b>	87.8	—	—	—	—
CCAE (Hermann and Blunsom, 2013)	77.8	—	—	—	—	—	87.2
Sent-Parser (Dong et al., 2014)	79.5	—	—	—	—	—	86.3
NBSVM (Wang and Manning, 2012)	79.4	—	—	93.2	—	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	—	—	<b>93.6</b>	—	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	—	—	93.4	—	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	—	—	<b>93.6</b>	—	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	—	—	—	—	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	—	—	—	—	—	82.7	—
SVM <sub>S</sub> (Silva et al., 2011)	—	—	—	—	<b>95.0</b>	—	—

# Diachronic word embeddings for studying language change!



# Visualizing changes

Project 300 dimensions down into 2



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

# Embeddings reflect cultural bias

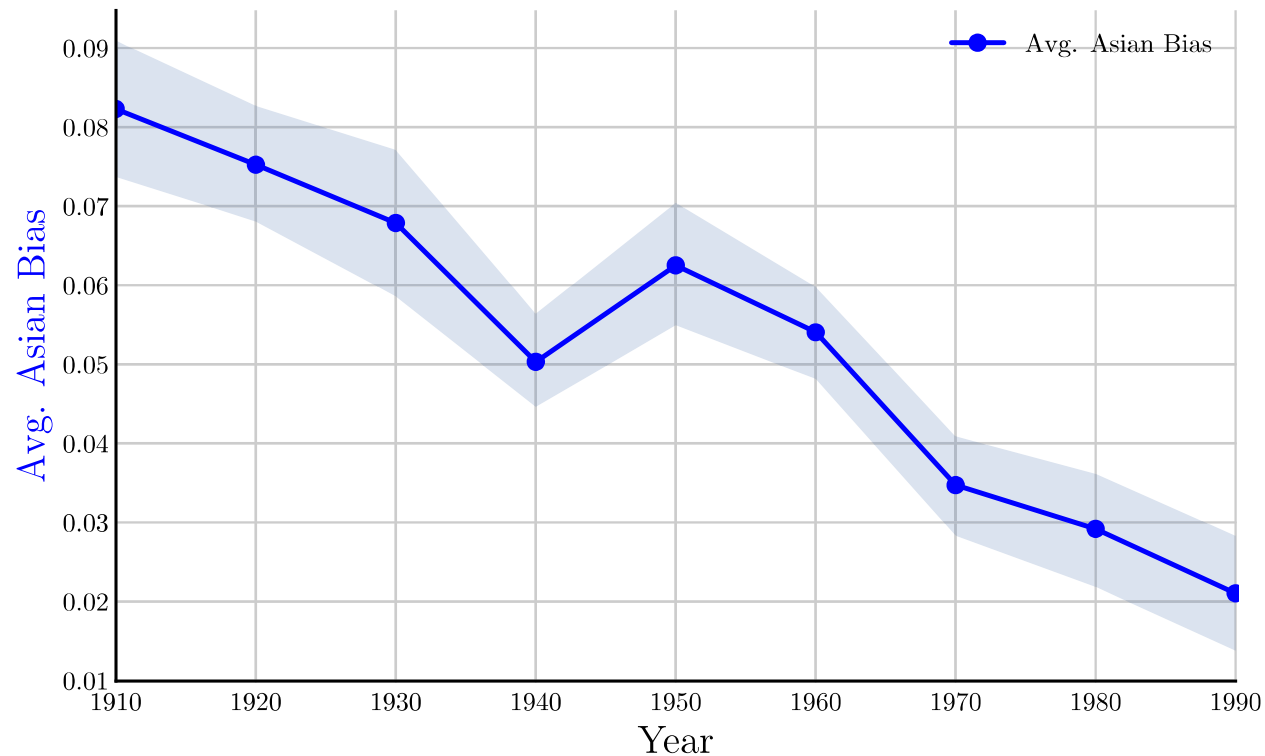
- Ask “Paris : France :: Tokyo : x”
  - x = Japan
- Ask “father : doctor :: mother : x”
  - x = nurse
- Ask “man : computer programmer :: woman : x”
  - x = homemaker

# Embeddings as a window onto history

- The cosine similarity of embeddings for decade X for occupations (like teacher) to male vs female names
  - Is correlated with the actual percentage of women teachers in decade X
- Embeddings for competence adjectives are biased toward men
  - Smart, wise, brilliant, intelligent, resourceful, thoughtful, logical, etc.
- This bias is slowly decreasing

# Change in linguistic framing 1910-1990

Change in association of Chinese names with adjectives framed as "othering" (*barbaric, monstrous, bizarre*)



# Changes in framing: adjectives associated with Chinese

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

# Summary: Embed all the things!

- Lots of applications wherever knowing word context or similarity helps prediction:
  - Synonym handling in search, Document aboutness, Ad serving, ...
- Fundamental to all other NLP tasks:
  - Language models: from spelling correction to email response
  - Machine translation
  - Sentiment analysis
  - ...
- Readings:
  - Dan Jurafsky and James H. Martin, *Speech and Language Processing (3rd ed. draft)*, Chapter 6  
<https://web.stanford.edu/~jurafsky/slp3/6.pdf>



# Paper references

- **Word2Vec:** Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
  - Omer Levy, Yoav Goldberg, Ido Dagan, Improving Distributional Similarity with Lessons Learned from Word Embeddings, Transactions of ACL, 2015
- **FastText:** Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching word vectors with subword information." *Transactions of the Association for Computational Linguistics* 5 (2017): 135-146.
- **CNN:** Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.
- **Diachronic:** Hamilton, William L., Jure Leskovec, and Dan Jurafsky. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1489-1501. 2016.
- **Biases:** Garg, Nikhil, Londa Schiebinger, Dan Jurafsky, and James Zou. "Word embeddings quantify 100 years of gender and ethnic stereotypes." Proceedings of the National Academy of Sciences 115, no. 16 (2018).