CS 2731 Introduction to Natural Language Processing

Session 8: Vector semantics, static word embeddings

Michael Miller Yoder

September 25, 2023

University of Pittsburgh | School of Computing and Information

Course logistics

- Projects
  - Soon Pantho (TA) and I will give feedback on project ideas
  - Proposal and literature review is **due Thu 10-12, 11:59pm**
    - Instructions are on the project webpage
  - It's good to start the literature review early
  - Look for NLP papers in ACL Anthology, Semantic Scholar, and Google Scholar
- Homework 2 is out today (or maybe tomorrow)
  - Text classification
  - Written and programming components
  - 5 bonus points for best feature-based system
  - 5 bonus points for best neural network system
  - We will run your code on a held-out test set

Lecture overview: vector semantics, static word embeddings

- Vector semantics
- Distributional semantics
- Types of word vectors
- Word2vec
- Bias in word vectors

A brief history of NLP

Sentences in Russian are punched into standard cards for feeding into the electronic data processing machine for translation into English

The Georgetown-IBM Experiment.
Credit: John Hutchins

- 1950s: foundations
  - Turing Test ("Computing Machinery and Intelligence" paper)
  - Georgetown-IBM Experiment translating Russian to English
- 1960s-1980s: symbolic reasoning
  - ELIZA, rule-based parsing, hand-built conceptual ontologies
- 1990s-2010s: statistical NLP
  - Learn patterns from large corpora (feature-based machine learning)
- 2000s-today: neural NLP
  - SOTA on many tasks from "deep" layers of neural networks

Vector semantics

Semantics: the study of meaning

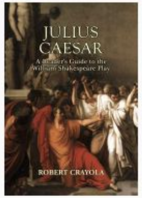
- Word representations in NLP draw on 2 areas of semantics
  a. Vector semantics
  b. Distributional semantics

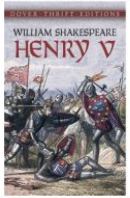
Vector semantics

- Modeling semantics as points in vector space
  - Multiple dimensions
  - Nearer = more similar words

Term-document matrix: word vectors

Two words are similar if their vectors are similar.





	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Slide adapted from David Mortensen, Jurafsky & Martin

Pairs of similar words?

Similarity and relatedness

- Synonyms: big/large, couch/sofa, automobile/car
- Similar: sharing some element of meaning
  - coffee/tea, car/bicycle, cow/horse
- Related: by a semantic field
  - coffee/cup, scalpel/surgeon

Slide adapted from Jurafsky & Martin

Distributional semantics

Distributional semantics: roots in anthropological linguistics

- In the early 20th century, many native languages of the Americas were dying due to the destruction of European colonization
- A group of American anthropologists (Boas, Sapir, Bloomfield, etc.) decided that they needed to describe all of these languages (produce grammars, dictionaries, and texts for them) before they were gone
- Earlier scholars who studied American languages tried to shoehorn them into the grammatical structure of European languages, but this group of researchers saw that they were very different from one another and from, e.g., Latin
- **They wanted to describe languages on their own terms**
- They developed techniques (in some cases, algorithms) for discovering meaning and grammatical structure without making reference to other languages
    - “To pass from one language to another is psychologically parallel from one geometrical system of reference to another.” (Sapir 1924)

Edward Sapir

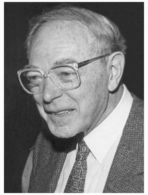
Slide adapted from David Mortensen

Distributional semantics

"The meaning of a word is its use in the language" [Wittgenstein 1953]

"You shall know a word by the company it keeps" [Firth 1957]

"If A and B have almost identical environments we say that they are synonyms" [Harris 1954]



Slide adapted from Jurafsky & Martin

Distributional semantics

Define the meaning of a word by its **distribution in language use**: its neighboring words or grammatical environments.

You Can Tell a Lot about *Beef* from Its Contexts

1	fertility. Organ meats such as	beef	and chicken liver, tongue and hear
2	controlling scours. _HOW TO FEED:	BEEF	AND DAIRY CALVES_ - 0.2 gram Dy
3	ing process discolors the treated	beef	and liquid accumulates in prepackag
4	say. He did say she could get her	beef	and vegetables in cans this summer
5	and feed efficiency of fattening	beef	animals. _HOW TO FEED:_ At the
6	steaks, chops, chicken and prime	beef	as well as Tom's favorite dish, stu
7	ross from him was surmounted by a	beef	barrel with ends knocked out. In t
8	counter of boards laid across two	beef	barrels. There was, of course, no
9	Because Holstein cattle weren't a	beef	breed, they were rarely seen on a
10	2-5 grams of phenothiazine daily;	beef	calves- .5 to 1.5 grams daily depe
11	ties of this drug. _HOW TO FEED:	BEEF	CATTLE (FINISHING RATION)_ - To
12	dairy cows and lesser amounts to	beef	cattle and poultry. About 90 percen
13	raises enough poultry, pigs, and	beef	cattle for most of their needs. Lo
14	on of liver abscesses in feed-lot	beef	cattle. Prevention of bacterial pne
15	pal feed bunk types for dairy and	beef	cattle: (1) Fence-line bunks- catt
16	es feed efficiency. _HOW TO FEED:	BEEF	CATTLE_ - 10 milligrams of diet
17	the rations you are feeding your	beef	, dairy cattle, and sheep are adequa
18	itive business more profitable for	beef	, dairy, and sheep men. The tar
19	o bear. She was ready to kill the	beef	, dress it out, and with vegetables
20	. She had raised a calf, grown it	beef	-fat. She had, with her own work-wea
21	with feeding low-moisture corn in	beef	-feeding programs. Several firms ar
22	he shelf life (at 35 F) of fresh	beef	from 5 days to 5 or 6 weeks. Howeve
23	canned pork products. Tests with	beef	have been largely unsuccessful beca
24	for eggs, pigs to eat garbage, a	beef	herd and wastes of all kinds. Separ
25	their money's worth. A good many	beef	-hungry settlers were accepting the

Slide credit: David Mortensen

Contexts for *Chicken* Are also Informative

1	y the irradiated and refrigerated	chicken	. Acceptance of radiopasteurization
2	torehouse". Glendora dropped a	chicken	and a flurry of feathers, and went
3	will specialize in steaks, chops,	chicken	and prime beef as well as Tom's fa
4	ard as the one concerned with the	chicken	and the egg. Which came first? Is
5	he millions of buffalo and prairie	chicken	and the endless seas of grass that
6	"! "Come on, there's some cold	chicken	and we'll see what else". They wen
7	ves to extend the storage life of	chicken	at a low cost of about 0.5 cent per
8	CHICKEN CADILLAC# Use one 6-ounce	chicken	breast for each guest. Salt and pe
9	ion juice, to about half cover the	chicken	breasts. Bake slowly at least one-
10	d, in butter. Sprinkle over top of	chicken	breasts. Serve each breast on a th
11	around, they had a hard time". #	CHICKEN	CADILLAC# Use one 6-ounce chicken
12	successful, and the shelf life of	chicken	can be extended to a month or more
13	ay from making a cake, building a	chicken	coop, or producing a book, to found
14	, they decided, but a deck full of	chicken	coops and pigpens was hardly suita
15	im. "Johnny insisted on cooking a	chicken	dinner in my honor- he's always bee
16	nutes. Kid Ory, the trombonist	chicken	farmer, is also one of the solid a
17	y Johnson reaching around the wire	chicken	fencing, which half covered the tr
18	yes glittering behind dull silver	chicken	fencing. "That was Tee-wah I was t
19	wine in the pot roast or that the	chicken	had been marinated in brandy, and
20	yed this same game and called it "	Chicken	". He could not go through the f
21	f the Mexicans hiding in a little	chicken	house had passed through his head,
22	I'll never forget him cleaning the	chicken	in the tub". A story, no doubt
23	. Organ meats such as beef and	chicken	liver, tongue and heart are planne
24	p. "Miss Sarah, I can't cut up no	chicken	. Miss Maude say she won't". Aga

Slide credit: David Mortensen

You Learn Words by Using Distributional Similarity

Consider

- A bottle of pocarisweat is on the table.
- Everybody likes pocarisweat.
- Pocarisweat makes you feel refreshed.
- They make pocarisweat out of ginger.

What does *pocarisweat* mean?

Slide credit: David Mortensen

You Know Pocarisweat by the Company It Keeps

From context words humans can guess *pocarisweat* means a beverage like **coke**.

How do you know?

- Other words can occur in the same context
- Those other words are often for beverages (that you drink cold)
- You assume that *pocarisweat* is probably similar

So the intuition is that **two words are similar if they have similar word contexts**.

Slide credit: David Mortensen

Sample Contexts of ±7 Words

sugar, a sliced lemon, a tablespoonful of	**apricot**	preserve or jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

	aardvark	computer	data	pinch	result	sugar …
⋮						
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0
⋮						

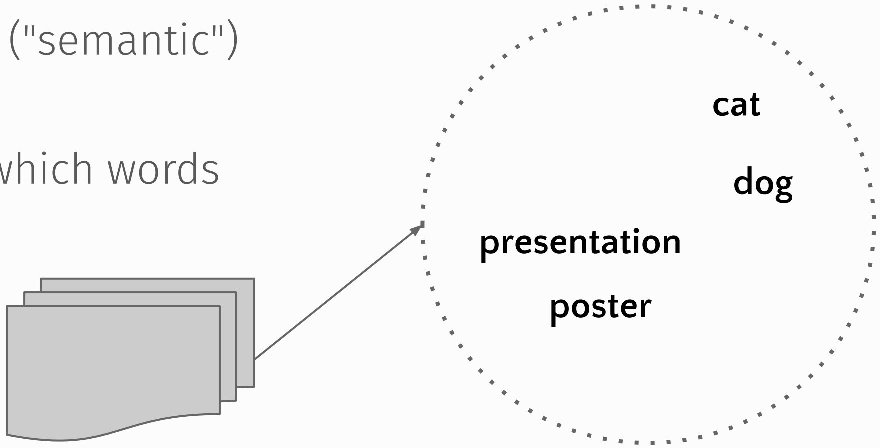
Slide credit: David Mortensen

Types of word vectors

Shared Intuition: Words are Vectors of Numbers Representing Meaning

- Model the meaning of a word by "embedding" it in a vector space.
- The meaning of a word is a vector of numbers:
  - Vector models are also called **embeddings**
  - Often, the word *embedding* is reserved for *dense* vector representations
- In contrast, word meaning is represented in many (early) NLP applications by a vocabulary index ("word number 545"; compare to one-hot representations)

- Similar words are nearby in vector ("semantic") space
- Build "semantic space" by seeing which words are nearby in text



Slide adapted from David Mortensen

Why word embeddings?

- Can generalize to similar but unseen words

cat [0.31, 0.24, 0.07, 0.65 …] **dog** [0.37, 0.29, 0.06, 0.63 …]

- Compute with meaning representations instead of string representations for words

荃者所以在鱼，得鱼而忘荃	Nets are for fish; Once you get the fish, you can forget the net.
言者所以在意，得意而忘言	Words are for meaning; Once you get the meaning, you can forget the words

庄子(Zhuangzi), Chapter 26

Slide adapted from Jurafksy & Martin

All modern NLP systems have embeddings as representations of word meaning

There are Two Kinds of Vector Models

- Sparse embeddings (vectors from term-document matrix)
  - long (length of 20,000 to 50,000)
  - sparse: most elements are 0
- Dense embeddings (Word2vec)
  - short (length of 50-1000)
  - dense (most elements are non-zero)


Slide adapted from David Mortensen, Jurafksy & Martin

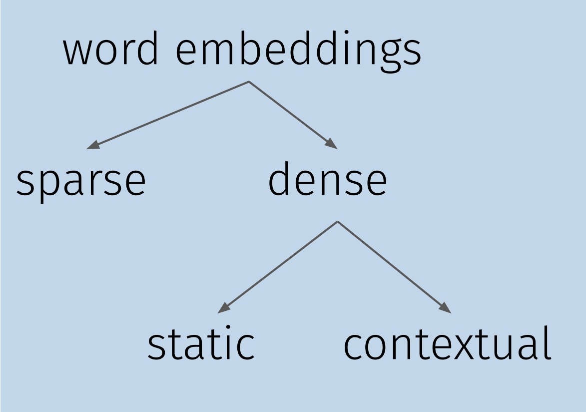
Dense Vectors Have Three Advantages over Sparse Vectors

1. Short vectors may be **easier to use as features** in machine learning (less weights to tune).
2. Dense vectors may **generalize better** than storing explicit counts.
3. They may do **better at capturing synonymy**:
    - *car* and *automobile* are synonyms
    - But, in sparse vectors, they are represented as distinct dimensions
    - This fails to capture similarity between a word with *car* as a neighbor and a word with *automobile* as a neighbor

Slide credit: David Mortensen

Methods for learning short, dense word embeddings

- Static, neural embeddings
  - Fixed embeddings for word types
  - Word2Vec, GloVe
- Contextual embeddings
  - Embeddings for words vary by context
  - ELMo, BERT, LLMs



Word2vec

Word2vec [Mikolov et al. 2013]

- Instead of counting words, train a classifier on a binary prediction task
  - Is w_1 likely to show up near w_2?

Slide adapted from Jurafsky & Martin

Word2vec [Mikolov et al. 2013]

- Instead of counting words, train a classifier on a binary prediction task
  - Is w_1 likely to show up near *apricot*?

Slide adapted from Jurafsky & Martin

Word2vec [Mikolov et al. 2013]

- Instead of counting words, train a classifier on a binary prediction task
    - Is w_1 likely to show up near *apricot*?

- Take the learned classifier weights as the word embeddings

Slide adapted from Jurafsky & Martin

Word2vec [Mikolov et al. 2013]

- Instead of counting words, train a classifier on a binary prediction task
    - Is w_1 likely to show up near *apricot*?

- Take the learned classifier weights as the word embeddings
- Training techniques: skip-gram and CBOW

Slide adapted from Jurafsky & Martin

Word2vec: training supervision

- **Self-supervision** [Bengio et al. 2003, Collobert et al. 2011]
- Use naturally occurring text as labels
- A word *c* that occurs near *apricot* in the corpus counts as the gold "correct answer" for supervised learning

Slide adapted from Jurafsky & Martin

Word2vec training overview

1. Positive examples: the target word w and a neighboring context word c_{pos}
2. Negative examples: Randomly sample other words c_{neg} in the lexicon to pair with w
3. Use logistic regression to train a classifier to distinguish those two cases
4. Use the learned weights (W, C) as the word embeddings

Slide adapted from Jurafsky & Martin

Training for Embeddings

- We do not know what *W* and *C* are. So we learn them through an iterative process.
- We use a large corpus as a training data
- We also randomly sample the corpus to find words that are NOT in the context—negative sampling.

A	soothsayer	bids	you	beware	the	Ides	of	March	.
				c_1	c_2	t	c_3	c_4	

Positive Examples		Negative Examples			
t	c	t	c	t	c
ides	beware	ides	aardvark	ides	twelve
ides	of	ides	puddle	ides	hello
ides	March	ides	where	ides	dear
ides	the	ides	coaxial	ides	forever

Slide credit: David Mortensen

Word2vec: learning embeddings

- Start with randomly initialized context *C* and target word *W* matrices
- Go through the positive and negative training pairs, adjusting word vectors such that we:
  - Maximize the similarity of the target word, context word pairs (w, c_{pos}) drawn from the positive data
  - Minimize the similarity of the (w, c_{neg}) pairs drawn from the negative data.

Slide adapted from Jurafsky & Martin

Skip-gram classifier

Classifier input pairs:

(target word *w*, context word *c*)

Classifier output: probabilities that *w* occurs with c

$P(+|w, c)$

$P(-|w, c) = 1 - P(+|w, c)$

Skip-gram classifier: calculating probabilities

- From input vectors, need to compare for similarity
  - *How to compare vectors for similarity?*
- Start with dot product: sim(**w**,**c**) ≈ **w** · **c**
- To turn this into a probability, use the sigmoid function from logistic regression:

$$P(+|w,c) \;=\; \sigma(c \cdot w) = \frac{1}{1 + \exp(-c \cdot w)}$$

Slide adapted from Jurafsky & Martin

Skip-gram classifier: calculating probabilities

$$P(+|w,c) = \sigma(c \cdot w) = \frac{1}{1+\exp(-c \cdot w)}$$

This is for one context word, but we have lots of context words.
We'll assume independence and just multiply them:

$$P(+|w,c_{1:L}) = \prod_{i=1}^{L} \sigma(c_i \cdot w)$$

$$\log P(+|w,c_{1:L}) = \sum_{i=1}^{L} \log \sigma(c_i \cdot w)$$

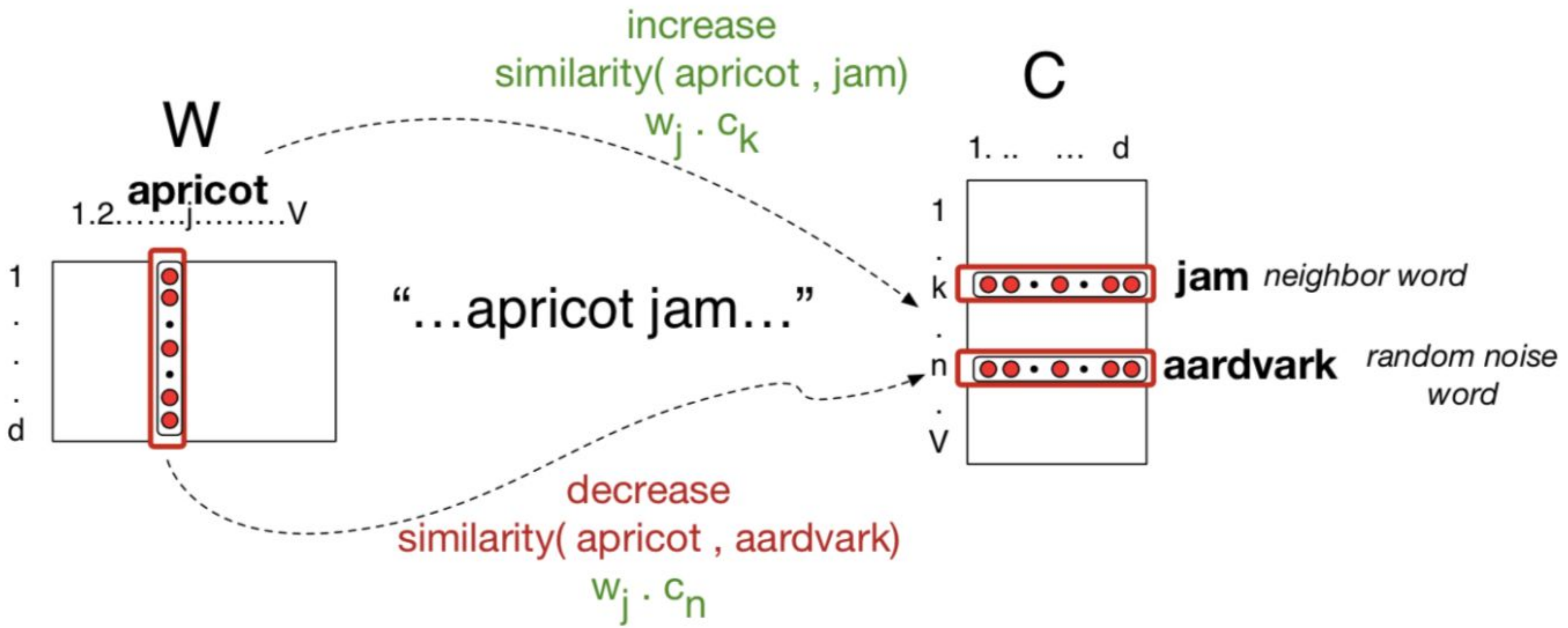
Loss function for one w with c_{pos} , c_{neg1} ...c_{negk}

Maximize the similarity of the target with the actual context words, and minimize the similarity of the target with the *k* negative sampled non-neighbor words.

$$
\begin{aligned}
L_{\text{CE}} &= -\log\left[P(+|w,c_{pos})\prod_{i=1}^{k}P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log P(-|w,c_{neg_i})\right] \\
&= -\left[\log P(+|w,c_{pos})+\sum_{i=1}^{k}\log\left(1-P(+|w,c_{neg_i})\right)\right] \\
&= -\left[\log \sigma(c_{pos}\cdot w)+\sum_{i=1}^{k}\log \sigma(-c_{neg_i}\cdot w)\right]
\end{aligned}
$$

Slide adapted from Jurafsky & Martin

Training for Embeddings



Slide adapted from David Mortensen, Jurafsky & Martin

Reminder: one step of gradient descent

- Direction: We move in the reverse direction from the gradient of the loss function
- Magnitude: we move the value of this gradient $d/dw\ L(P(+|w,c) + P(-|w,c))$ weighted by a learning rate η
- Higher learning rate means move *w* faster

Slide adapted from Jurafsky & Martin

Word2vec training process

Updates on C and W

$$c_{pos}^{t+1} = c_{pos}^{t} - \eta [\sigma(c_{pos}^{t} \cdot w^{t}) - 1]$$

new context weights

old context weights

learning rate

derivative of loss wrt c_{pos}

$$w^{t+1} = w^{t} - \eta \left[[\sigma(c_{pos} \cdot w^{t}) - 1] c_{pos} + [\sigma(c_{\mathrm{neg}_i} \cdot w^{t})] c_{neg_i} \right]$$

new target word weights

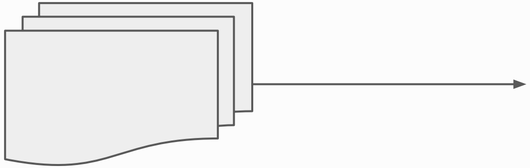
old context weights

learning rate

derivative of loss wrt w

Slide adapted from Jurafsky & Martin

Summary: How to learn word2vec embeddings



cat	[0.31, 0.24, 0.07, 0.65 …]
dog	[0.37, 0.29, 0.06, 0.63 …]
presentation	[0.65, 0.93, 0.16, 0.78 …]
poster	[0.57, 0.82, 0.21, 0.73 …]

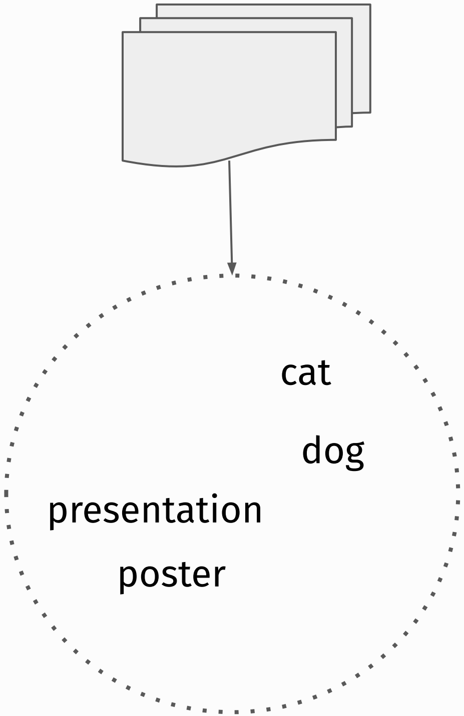
Summary: How to learn word2vec embeddings

1. Start with randomly initialized word embeddings
2. Take a corpus and extract pairs of words that co-occur (positive)
3. Take pairs of words that don't co-occur (negative)
4. Train a classifier to distinguish between positive and negative examples by slowly adjusting all the embeddings to improve the classifier performance
5. Keep the weights as our word embeddings

Slide adapted from Jurafsky & Martin

Final embeddings

- Can add representations for a word in W and in C together for final word vector for w_i
- Can just keep W and throw away C
- Can find "nearest neighbors" of certain words with cosine similarity in embedding space



Slide adapted from Jurafsky & Martin

There are Tools and Resources Available for Training and Using Embeddings

- **Pretrained embeddings**
  - Skip-gram
  - CBOW
  - fastText
  - GloVe
- **Training your own embeddings**
  - You can easily train skip-gram, CBOW, and fastText embeddings with `gensim`
  - Straightforward Python interface

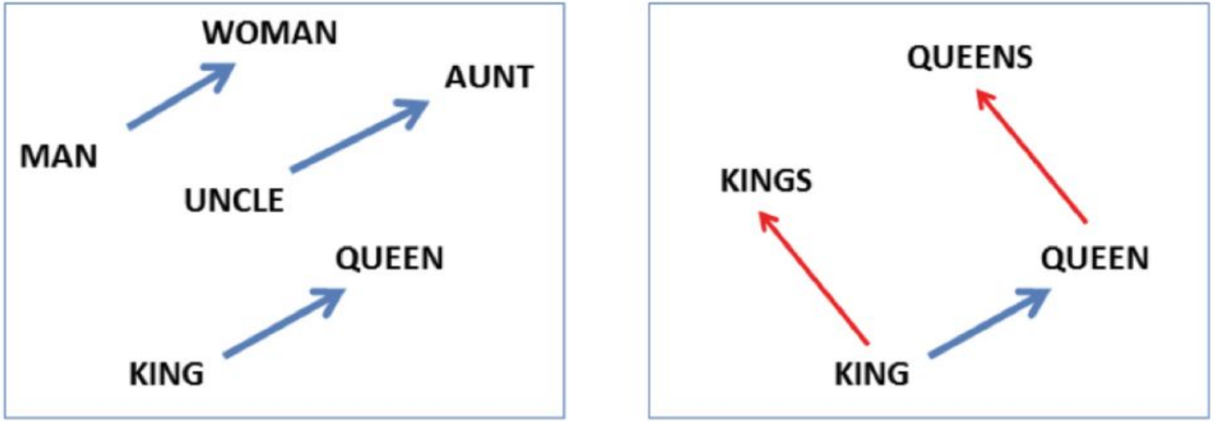
Slide credit: David Mortensen

Observations on Embeddings

- Nearest words to some embeddings in the $d-$ dimensional space.

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

- Relation meanings
    - $vector(king) - vector(man) + vector(woman) \approx vector(queen)$
    - $vector(Paris) - vector(France) + vector(Italy) \approx vector(Rome)$



Slide credit: David Mortensen

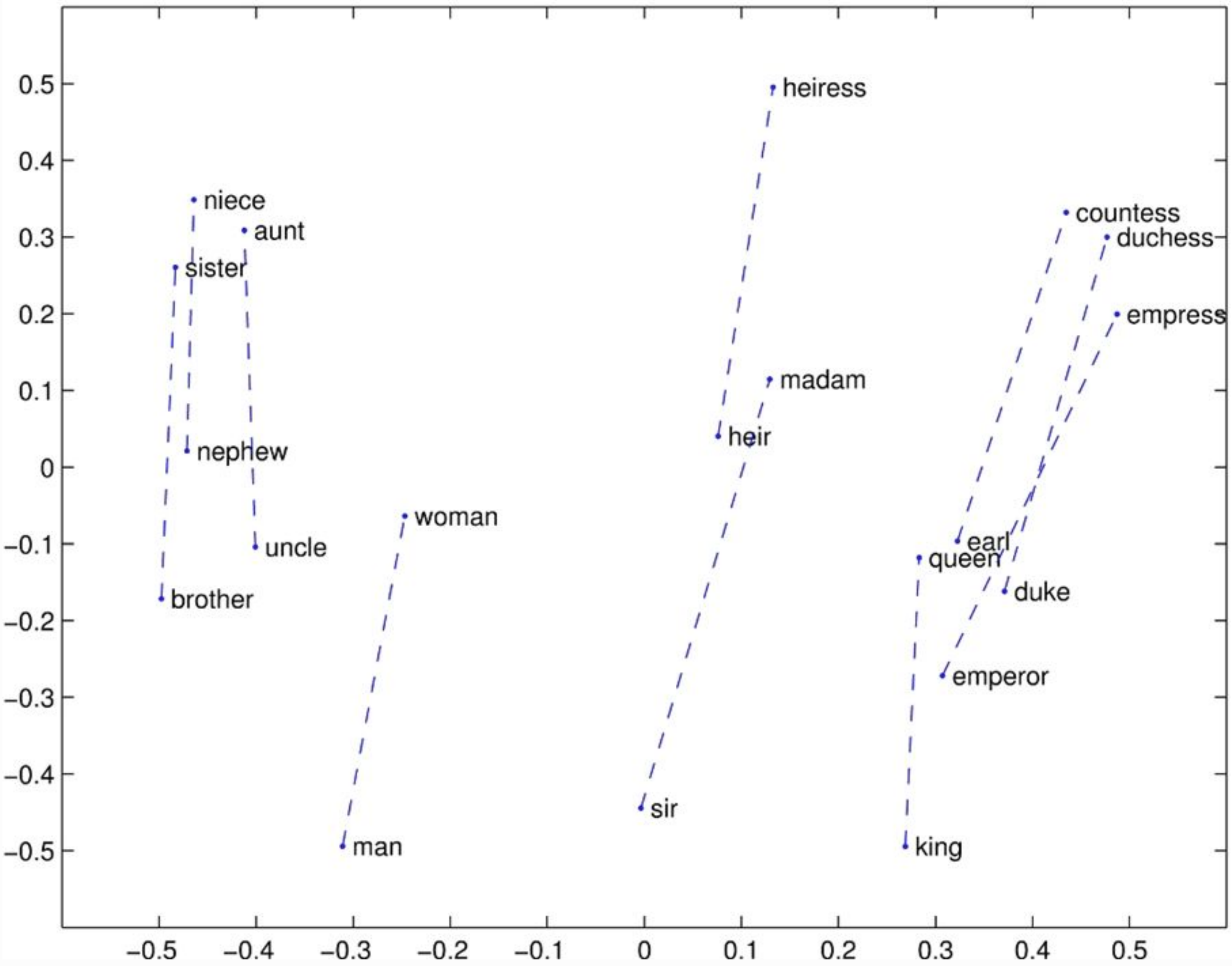
Analogies

$\overrightarrow{\text{king}} - \overrightarrow{\text{man}} + \overrightarrow{\text{woman}}$ is close to $\overrightarrow{\text{queen}}$

$\overrightarrow{\text{Paris}} - \overrightarrow{\text{France}} + \overrightarrow{\text{Italy}}$ is close to $\overrightarrow{\text{Rome}}$

Caveats: only seems to work for frequent words, small distances and certain relations, like relating countries to capitals, or parts of speech [Linzen 2016, Gladkova et al. 2016, Ethayarajh et al. 2019a]

Slide adapted from Jurafsky & Martin


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

Embeddings reflect cultural biases [Bolukbasi et al. 2016]

- Paris : France :: Tokyo : *Japan*”
- Sexist occupational stereotypes
  - father : doctor :: mother : *nurse*
  - man : computer programmer :: woman : *homemaker*
- Would be problematic to use embeddings in hiring searches for programmers

Slide adapted from Jurafsky & Martin

Discussion forum: Blodgett et al. 2020

- Recommendations from Blodgett et al. for better work on bias
    1. Ground work analyzing bias in relevant literature outside of NLP that explores relationships between language and social hierarchies. Treat representational harms as harmful in their own right
    2. Explicitly state why “bias” in systems is harmful, in what ways, and to whom. Be explicit about normative reasoning behind these judgements.
    3. Engage with the lived experiences of members of communities affected by NLP systems. Reimagine power relations between technologists and such communities.

Discussion forum: Blodgett et al. 2020

- Feasibility of Blodgett et al. 2020's recommendations
  a. R1: Specialists outside of CS get less research money—let them lead (Marcelo)
  b. R2: Explicitly stating bias helps people work from the same starting point (standardization). But this is impossible (Max)
     - Values are at play though, especially in "prescriptive" models (Ben)
     - Why is this hard? (Gina) "Objectivity", sensitive, awkward to talk about power and injustice
  c. R3 is important but hard!
     - What about CS research's problem of excluding historically marginalized people from research (Gina)

Discussion forum: Blodgett et al. 2020

- Allocational harms
    a. COMPAS, predictive policing (Lingwei)
    b. Education technologies in allocation of tutoring, etc (Haoyu)
- “Using” bias in embeddings to investigate society, etc
    a. “Descriptive models” (Ben)
    b. Avoid promoting bad material through recommendation systems (Tom)
    c. Search their own datasets for bias (Norah)
    d. Find cultural differences (Jiyuan)

Discussion forum: Blodgett et al. 2020

- Language ideologies
  a. Standard Mandarin = “good” vs dialects = “bad”/rude (Qichang, Yuxuan)
     - Lack of employment opportunities
     - Cultural loss if kids can’t speak Chinese dialects
  b. Underrepresentation in NLP systems = worse performance
     - Dialect of Malayalam (Dhanush)
  c. Code-mixing in India (with English, the “language of the educated”) is a challenge to NLP systems (Bhiman)
  d. Controversy over a movie in India using common dialect in a religious topic (Lokesh)
  e. Regional languages preferred for some jobs (government in India, Bhiman) or in Saudi Arabia vs Modern Standard Arabic (Aziz), which means they are overqualified
  f. “Good” and “bad” judgments serve the interest of those in power (Nietzsche, from Birju)

Takeaways

- NLP typically represents words as vectors in spaces where distance ≈ semantic similarity
- Word2vec learns static embeddings (vectors) for words by predicting which words occur together in training data
- These embeddings are effective in downstream NLP tasks, but also reflect social biases of training data text

Questions?