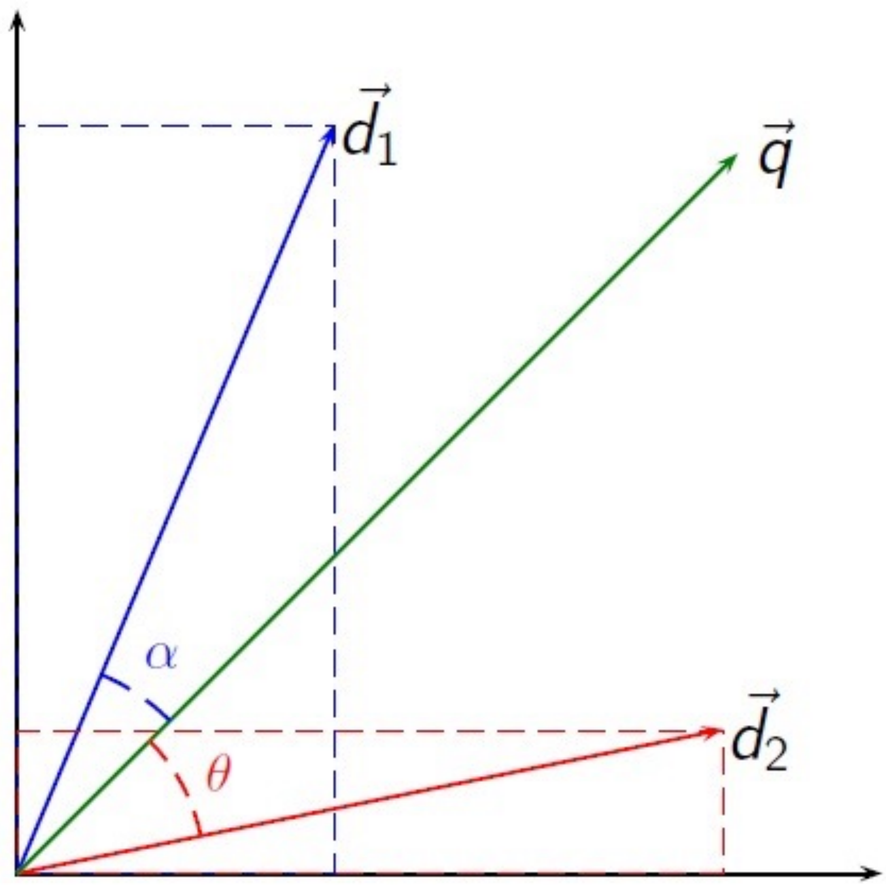
Lecture 28: Vector Semantics

Mark Hasegawa-Johnson

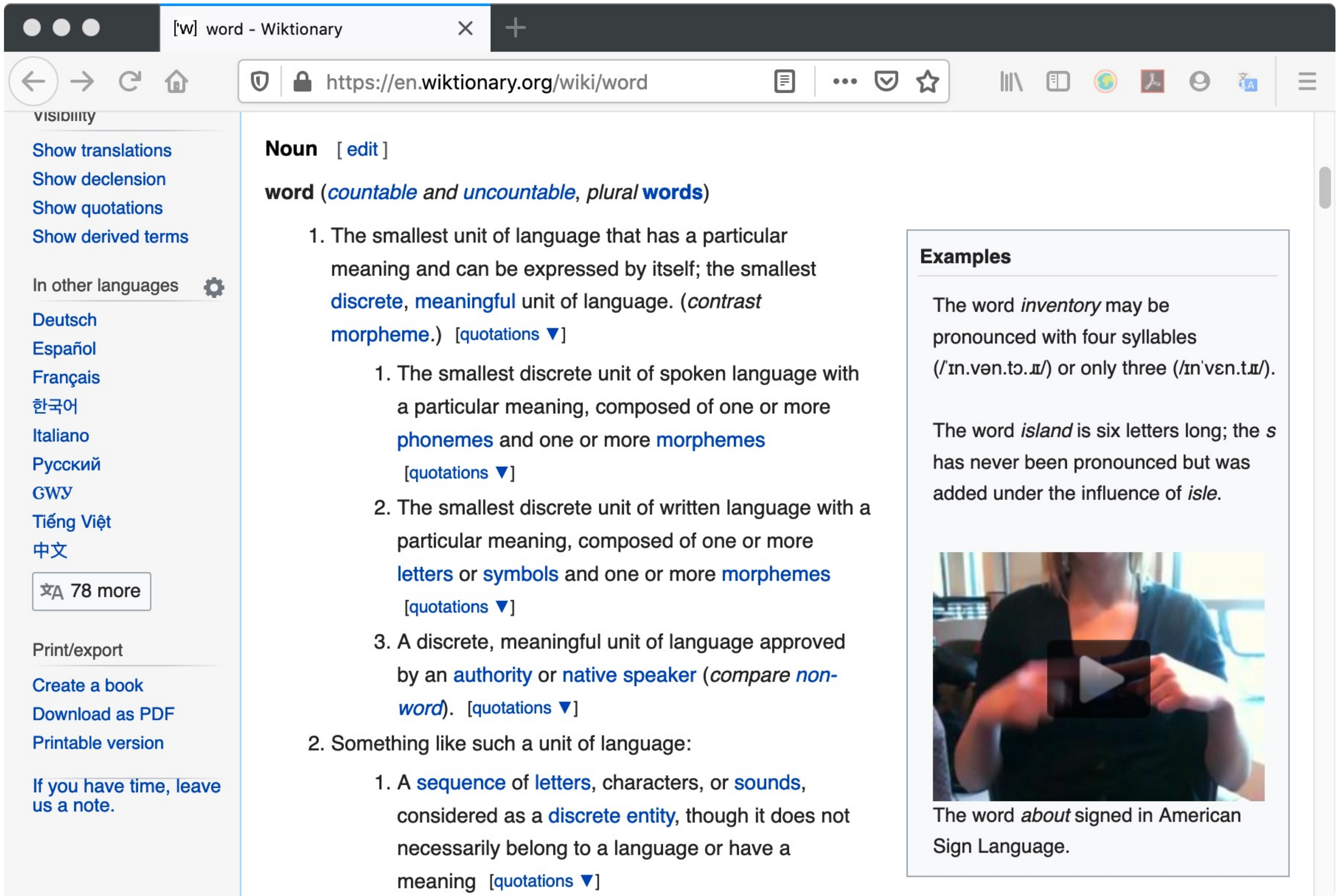
4/2024



Outline

- What is a word? wordforms vs. lemmas vs. word senses
- What is meaning? synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

What is a word?


[w] word - Wiktionary

https://en.wiktionary.org/wiki/word

Visibility

Show translations
Show declension
Show quotations
Show derived terms

In other languages

Deutsch
Español
Français
한국어
Italiano
Русский
ᏣᎳᎩ
Tiếng Việt
中文

78 more

Print/export

Create a book
Download as PDF
Printable version

If you have time, leave us a note.

Noun [edit]

word (*countable and uncountable*, *plural* **words**)

1. The smallest unit of language that has a particular meaning and can be expressed by itself; the smallest discrete, meaningful unit of language. (*contrast* morpheme.) [quotations ▼]
    1. The smallest discrete unit of spoken language with a particular meaning, composed of one or more phonemes and one or more morphemes [quotations ▼]
    2. The smallest discrete unit of written language with a particular meaning, composed of one or more letters or symbols and one or more morphemes [quotations ▼]
    3. A discrete, meaningful unit of language approved by an authority or native speaker (*compare non-word*). [quotations ▼]
2. Something like such a unit of language:
    1. A sequence of letters, characters, or sounds, considered as a discrete entity, though it does not necessarily belong to a language or have a meaning [quotations ▼]

Examples

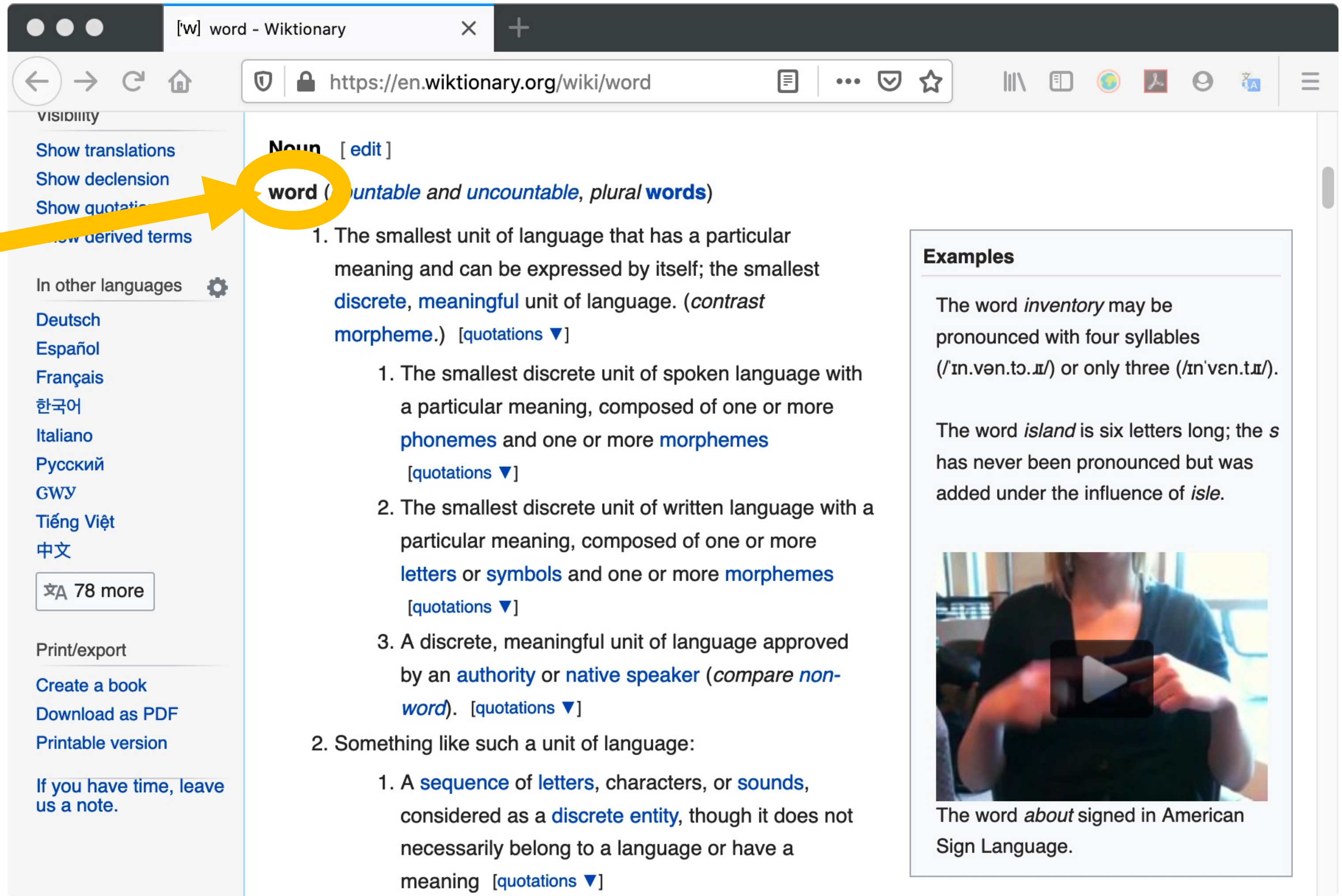
The word *inventory* may be pronounced with four syllables (/ˈɪn.vən.tɔ.ɹi/) or only three (/ɪnˈvɛn.tɹi/).

The word *island* is six letters long; the *s* has never been pronounced but was added under the influence of *isle*.

The word *about* signed in American Sign Language.

What is a word?

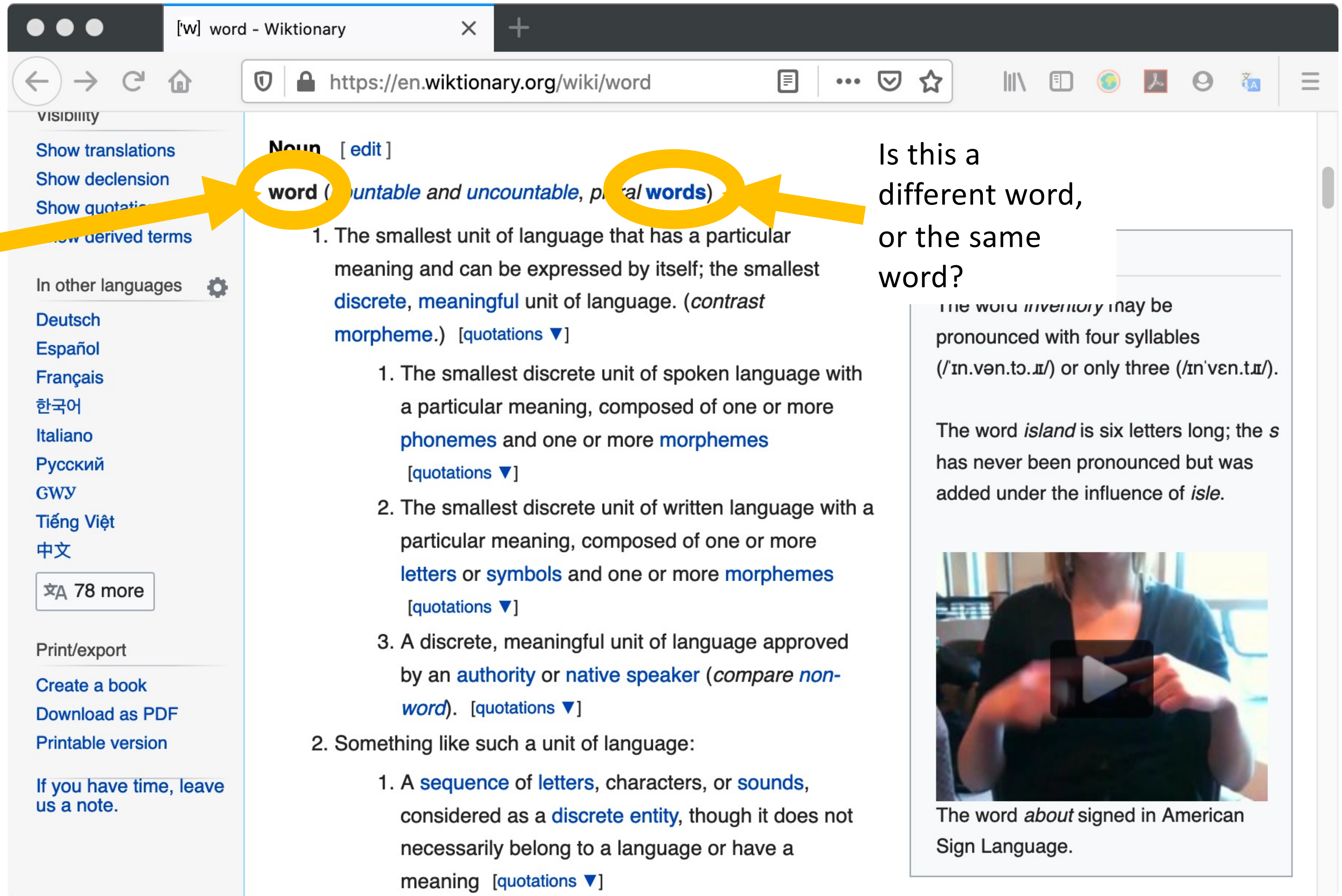
Is this a word?



What is a word?

Is this a word?

Is this a different word, or the same word?



Wordform

A wordform is a unique sequence of characters.

- Wordforms are much easier for computers to find than lemmas, therefore most automatic processing deals with wordforms.
- …however, we lose something. "dog" and "dogs" become completely unrelated – as unrelated as "dog" and "exaggerate."

word (*countable and uncountable*, *plural* **words**)

1. The smallest unit of language that has a particular meaning and can be expressed by itself; the smallest discrete, meaningful unit of language. (*contrast* morpheme.) [quotations ▼]
    1. The smallest discrete unit of spoken language with a particular meaning, composed of one or more phonemes and one or more morphemes [quotations ▼]
    2. The smallest discrete unit of written language with a particular meaning, composed of one or more letters or symbols and one or more morphemes [quotations ▼]
    3. A discrete, meaningful unit of language approved by an authority or native speaker (*compare non-word*). [quotations ▼]
2. Something like such a unit of language:
    1. A sequence of letters, characters, or sounds, considered as a discrete entity, though it does not necessarily belong to a language or have a meaning [quotations ▼]

Lemma

A lemma is what humans usually think of as a “word.” It is defined to be the form of the word that appears in a dictionary.

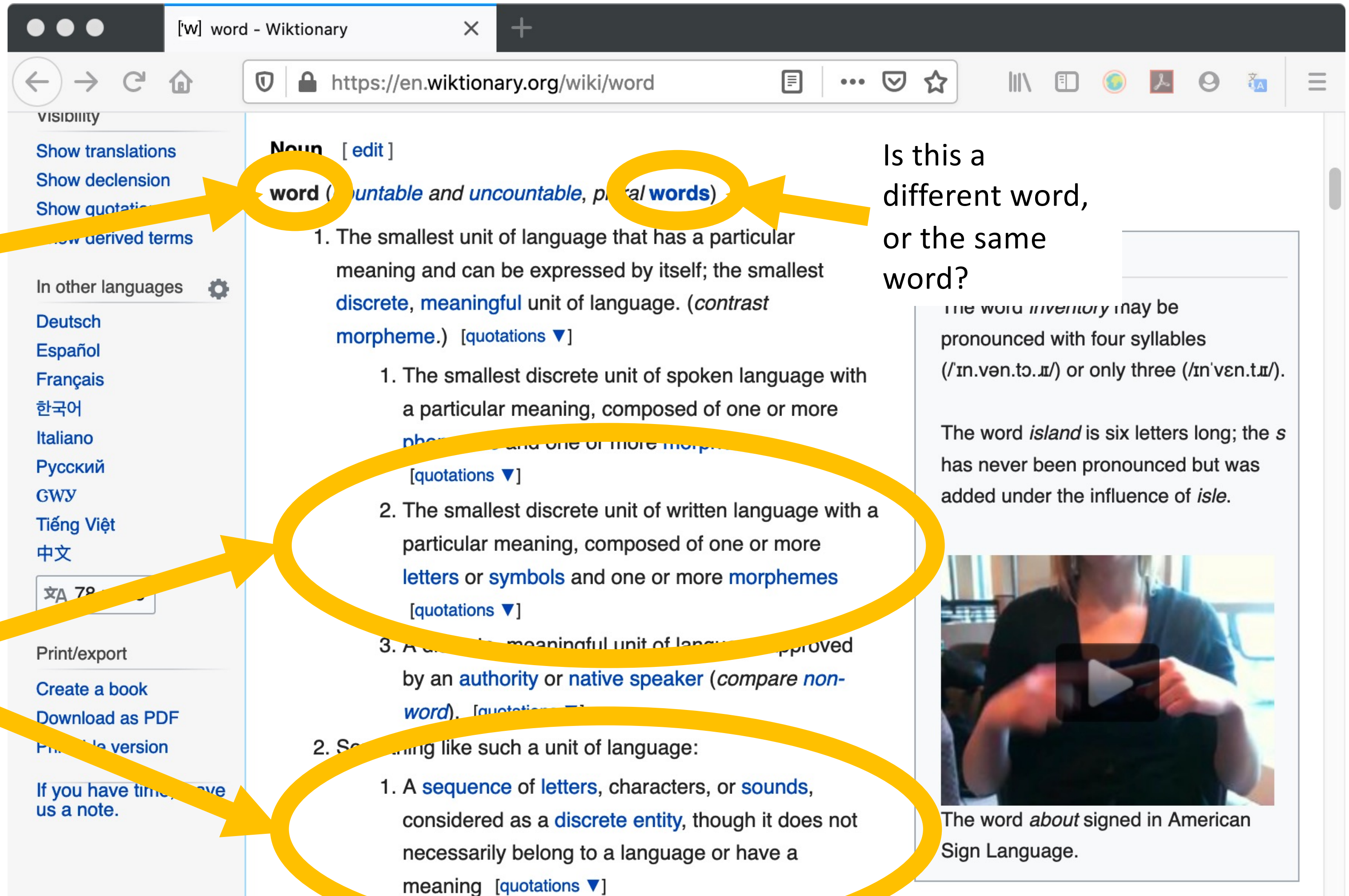
- In dictionaries designed for human beings,…
- other wordforms that can be easily predicted from the lemma are not listed.

word (*countable* and *uncountable*, *plural* **words**)

1. The smallest unit of language that has a particular meaning and can be expressed by itself; the smallest discrete, meaningful unit of language. (*contrast* morpheme.) [quotations ▼]
    1. The smallest discrete unit of spoken language with a particular meaning, composed of one or more phonemes and one or more morphemes [quotations ▼]
    2. The smallest discrete unit of written language with a particular meaning, composed of one or more letters or symbols and one or more morphemes [quotations ▼]
    3. A discrete, meaningful unit of language approved by an authority or native speaker (*compare non-word*). [quotations ▼]
2. Something like such a unit of language:
    1. A sequence of letters, characters, or sounds, considered as a discrete entity, though it does not necessarily belong to a language or have a meaning [quotations ▼]

What is a word?

Is this a word?



Is this a different word, or the same word?

Are these the same word, or different words?

Word sense

Often, a word has different meanings that are completely unrelated. We think of them as different words, that just happen to be spelled and pronounced the same way.

We say that these are different “senses” of the same word.

The Bank of England. By Diliff - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=40912212

The Bank of the Thames. By Diliff - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=3639626

Wordform, lemma, and word sense

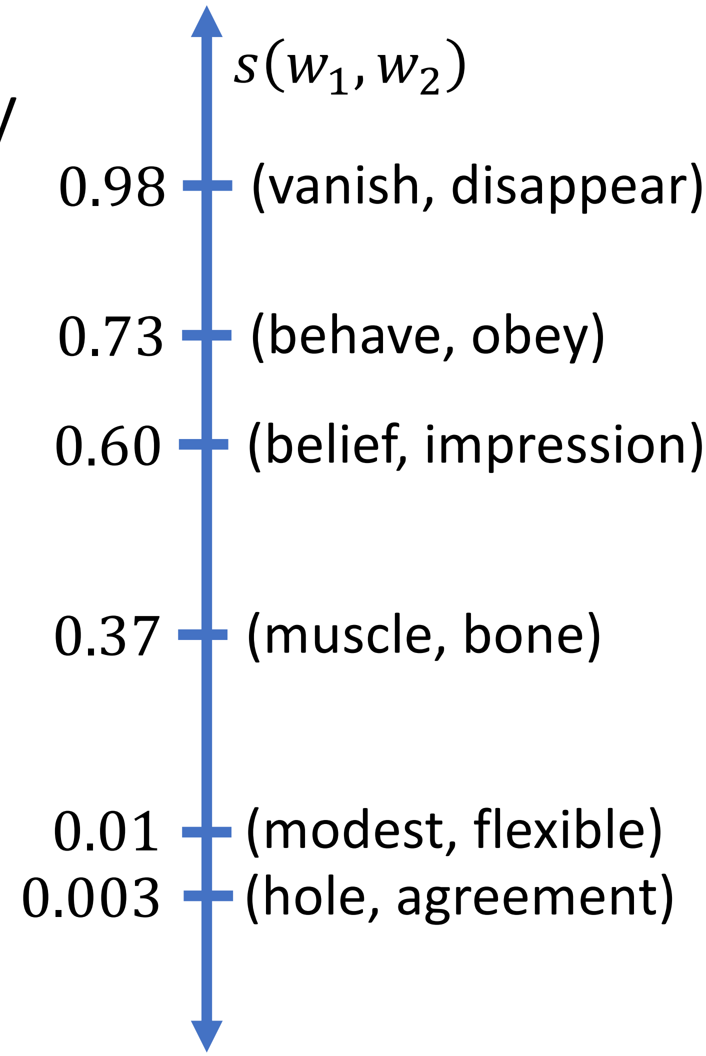
- wordform
  - easy for a computer to work with: just look for space-bounded sequences of characters
- lemma
  - This is what humans think of as a word. A set of wordforms whose spellings, pronunciations, and meanings can all be derived from one another by applying simple rules.
- word sense
  - A meaning so distinct from the other meanings of the word that it's hard to consider them the same word.

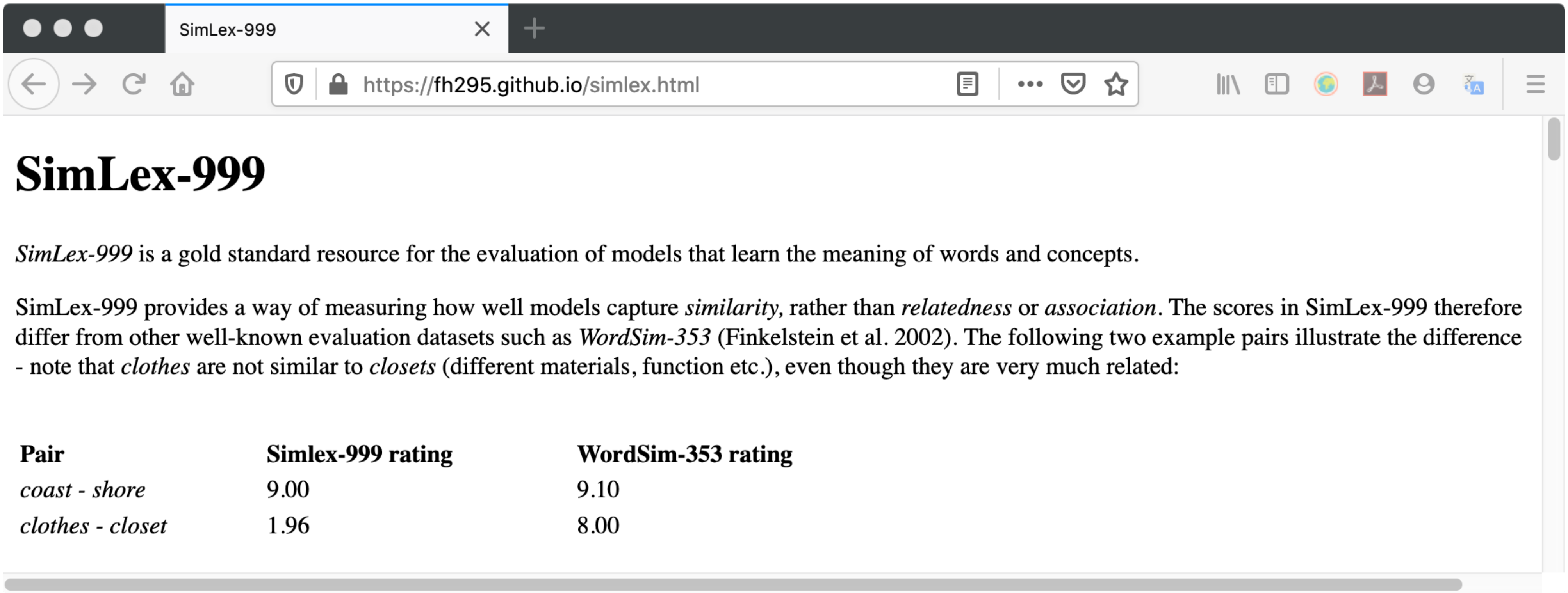
Outline

- What is a word? wordforms vs. lemmas vs. word senses
- What is meaning? synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

Synonymy and similarity

- Words are "synonyms" if they have exactly the same meaning.
- No words ever have ***<u>exactly</u>*** the same meaning, so no two words are ever exactly synonyms.
- We prefer to talk about word similarity, $0 \le s(w_1, w_2) \le 1$
  - $s(w_1, w_2) = 1$: w_1 and w_2 are perfect synonyms. Never happens in practice, but sometimes close.
  - $s(w_1, w_2) = 0$: w_1 and w_2 are completely different.





SimLex-999

https://fh295.github.io/simlex.html

SimLex-999

SimLex-999 is a gold standard resource for the evaluation of models that learn the meaning of words and concepts.

SimLex-999 provides a way of measuring how well models capture *similarity,* rather than *relatedness* or *association*. The scores in SimLex-999 therefore differ from other well-known evaluation datasets such as *WordSim-353* (Finkelstein et al. 2002). The following two example pairs illustrate the difference - note that *clothes* are not similar to *closets* (different materials, function etc.), even though they are very much related:

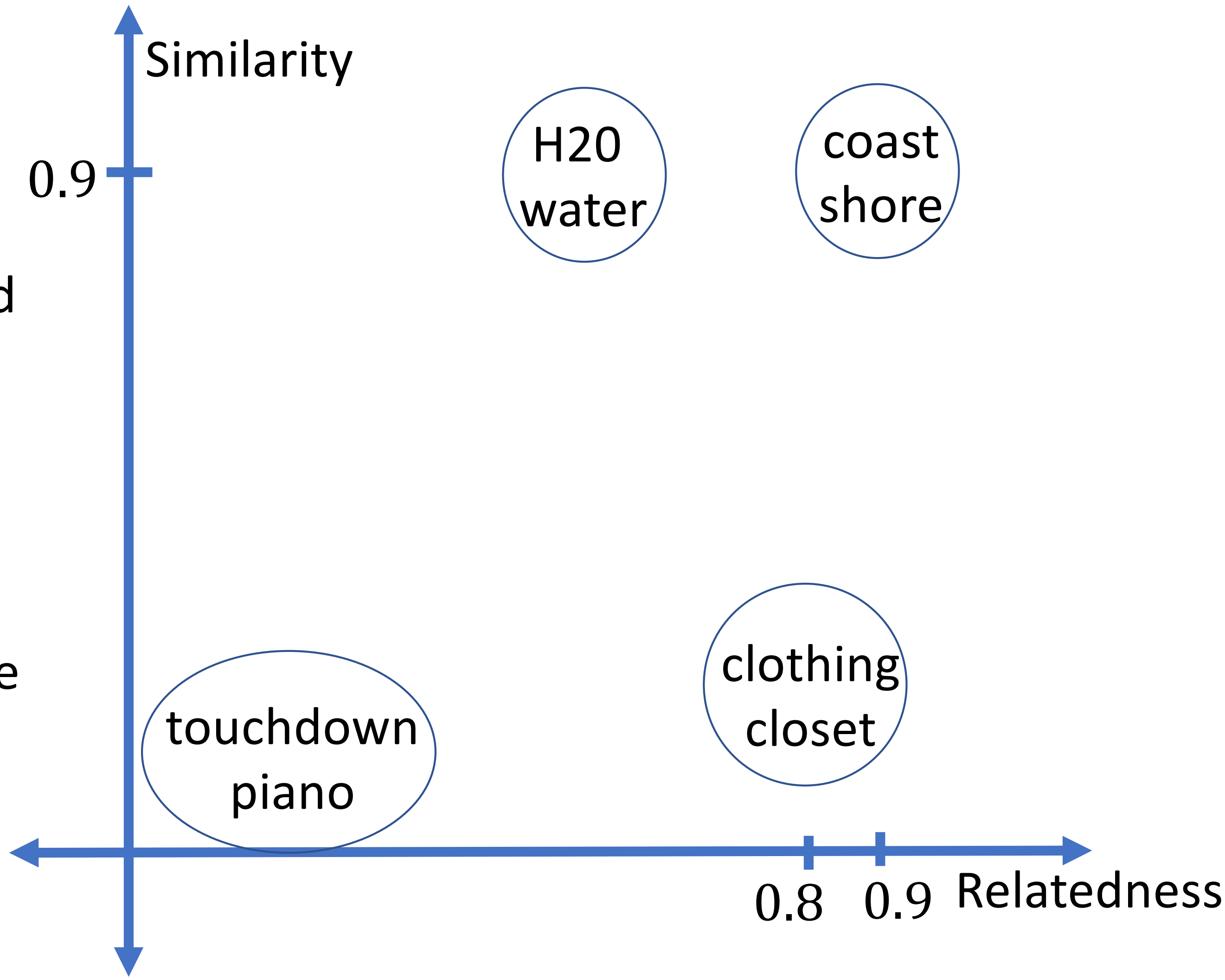
Pair	Simlex-999 rating	WordSim-353 rating
coast - shore	9.00	9.10
clothes - closet	1.96	8.00

- Algorithms that try to estimate the similarity of two wordforms can be tested on databases such as SimLex-999.
- Humans rated the similarity of each word pair on a 10-point scale.

Similarity vs. Relatedness

Similar: words can be used interchangeably in most contexts

Related: there is some connection between the two words, such that they tend to appear in the same documents.



Outline

- What is a word? wordforms vs. lemmas vs. word senses
- What is meaning? synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

Review: Naïve Bayes: the "Bag-of-words" model

We can estimate the likelihood of an e-mail by pretending that the e-mail is just a bag of words (order doesn't matter).

With only a few thousand spam e-mails, we can get a pretty good estimate of these things:

- $P(W = \text{"hi"} | Y = \text{spam})$, $P(W = \text{"hi"} | Y = \text{ham})$
- $P(W = \text{"vitality"} | Y = \text{spam})$, $P(W = \text{"vitality"} | Y = \text{ham})$
- $P(W = \text{"production"} | Y = \text{spam})$, $P(W = \text{"production"} | Y = \text{ham})$

Then we can approximate $P(X|Y)$ by assuming that the words, W, are ***conditionally independent of one another given the category label***:

$$P(X = x | Y = y) \approx \prod_{i=1}^{n} P(W = w_i | Y = y)$$

Similarity: The Internet is the database

Similarity = words can be used interchangeably in most contexts

How do we measure that in practice?

Answer: extract examples of word w_1, +/- C words (C=2 or 3):

…hot, although iced **coffee** is a popular…

…indicate that moderate **coffee** consumption is benign…

…and of w_2:

…consumed as iced **tea**. Sweet tea is…

…national average of **tea** consumption in Ireland…

The words "iced" and "consumption" appear in both contexts, so we can conclude that $s(\text{coffea}, \text{tea}) > 0$. No other words are shared, so we can conclude $s(\text{coffee}, \text{tea}) < 1$.

skip-gram context probability

Consider the "…hot although iced **coffee** is a popular…".

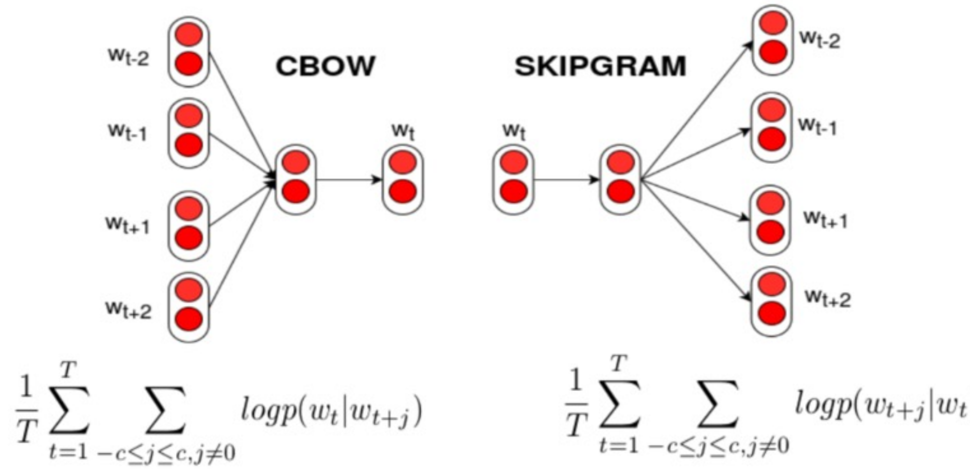
Define the target word to be w_t =coffee.

Define the context words w_{t-3} =hot, w_{t-2} =although, …, w_{t+3} =popular.

The skip-gram probability is a naïve Bayes model of the context:

$$p(w_{t-3}, \dots, w_{t+3} | w_t) = \prod_{\substack{i \neq 0 \\ i=-3}}^{3} p(w_{t+j} | w_t)$$

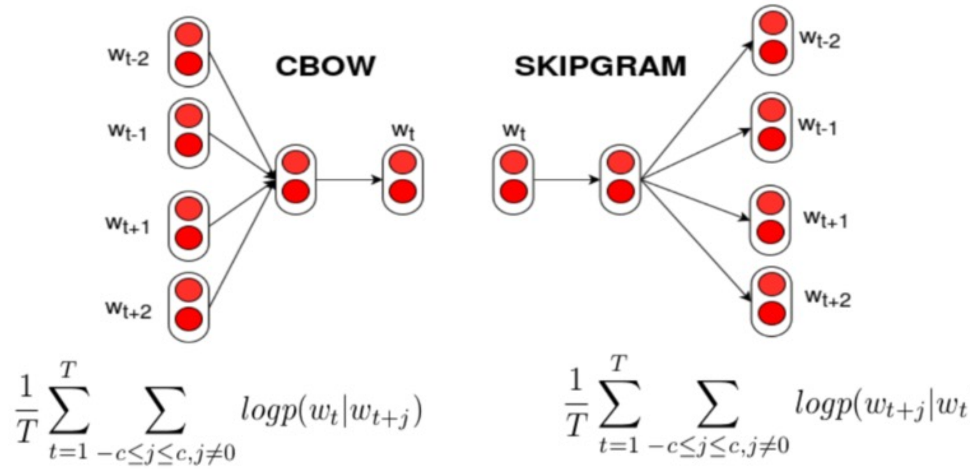
The skip-gram model



- Skip-gram is a model of word meaning:
- The meaning of a word is defined to be the distribution of context words that it can predict.
- We find out which words w_t can predict by learning neural nets that predict its context words w_{t+j}:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=-c,j\neq 0}^{c} \ln P\left(w_{t+j}|w_t\right)$$

The "continuous bag of words" model (CBOW)



- CBOW is a similar model of word meaning:
- The meaning of a word is defined to be the distribution of context words that predict it the best.
- We find out which words predict w_t by learning neural nets that predict w_t given its context words, w_{t+j}, for $-c \le j \le c$:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=-c, j\neq 0}^{c} \ln P\left(w_t|w_{t+j}\right)$$

"Probability," for a NN, means softmax

- What does it mean that we train a neural net to compute $P(w_t|w_{t+j})$?
- It's a probability, so it must mean a softmax:

$$P(W_t = w_t | W_{t+j} = w_{t+j}) = \frac{\exp(e_{t,t+j})}{\sum_{t'} \exp(e_{t',t+j})}$$

- But what are the inputs to the neural net? What is $e_{t,t+j}$?

Vector Semantics

- The simplest useful assumption is this: a word is a vector.

$$P\left(W_t = w_t | W_{t+j} = w_{t+j}\right) = \frac{\exp\left(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)}$$

- …where $\boldsymbol{v}_t$ is a d-dimensional vector, $\boldsymbol{v}_t = \left[v_{t,1}, \ldots, v_{t,d}\right]^T$, and $\mathcal{V}$ is the set of all such vectors (the "vocabulary")
- The only trainable parameters in this model are the word vectors!
- The dictionary, $\mathcal{V}$, is a matrix, with as many columns as there are words in the vocabulary:

$$\mathcal{V} = [\text{vec}(''a''), \ldots, \text{vec}(''zzz'')] = \begin{bmatrix} \text{vec}(''a'')_1 & \cdots & \text{vec}(''zzz'')_1 \\ \vdots & \ddots & \vdots \\ \text{vec}(''a'')_d & \cdots & \text{vec}(''zzz'')_d \end{bmatrix}$$

cosine similarity

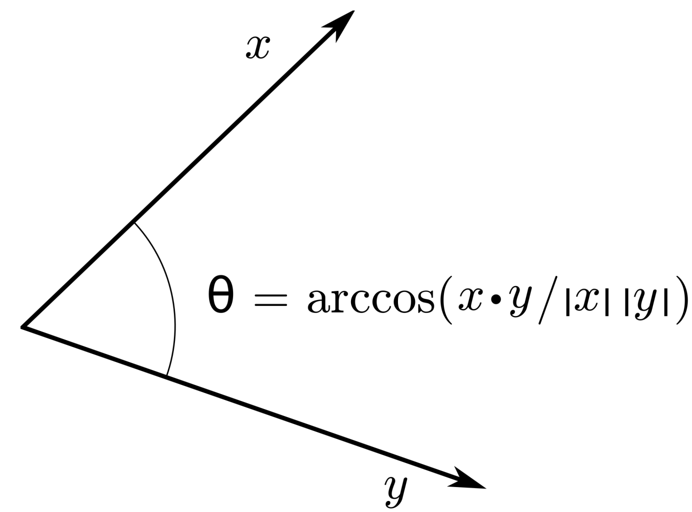
If words w_1 and w_2 are similar, w_1 is represented by vector $\boldsymbol{v}_1$, and w_2 by vector $\boldsymbol{v}_2$, then the angle between the two vectors should be small.

Angle between two vectors can be measured by their dot product:

$$\cos\theta = \frac{\boldsymbol{v}_1^T \boldsymbol{v}_2}{|\boldsymbol{v}_1||\boldsymbol{v}_2|}$$

where

$$\boldsymbol{v}_1^T \boldsymbol{v}_2 = \sum_{i=1}^{d} v_{1,i} v_{2,i}\,, \qquad |\boldsymbol{v}_1| = \sqrt{\sum_{i=1}^{d} v_{1,i}^2}$$



Vector Semantics: Variations

There are many ways to make this model more flexible. For example:

- Every word could have two different vectors: one ($\boldsymbol{v}$) for when it's being predicted, one ($\boldsymbol{v}'$) for when it is predicting, thus $e_{t,t+j} = \boldsymbol{v}_t^T \boldsymbol{v}'_{t+j}$.
- We could weight the dot-product depending on the delay between the word vectors, thus $e_{t,j,t+j} = \boldsymbol{v}_t^T \boldsymbol{W}(j) \boldsymbol{v}'_{t+j}$.
- We could use a two-layer network to calculate the similarity, for example, $e_{t,t+j} = \boldsymbol{w}_2^T \max\left(\boldsymbol{0}, \boldsymbol{W}_1(j) \begin{bmatrix} \boldsymbol{v}_t \\ \boldsymbol{v}'_{t+j} \end{bmatrix}\right)$.

Vector Semantics

The basic CBOW probability is:

$$P\left(W_t = w_t | W_{t+j} = w_{t+j}\right) = \frac{\exp\left(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)}$$

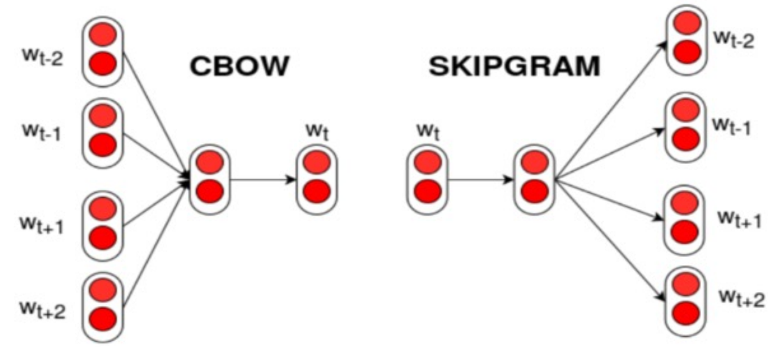
Its logarithm is:

$$\ln P\left(W_t = w_t | W_{t+j} = w_{t+j}\right) = \boldsymbol{v}_t^T \boldsymbol{v}_{t+j} - \ln \sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)$$

The gradient of the log softmax is:

$$\nabla_{\boldsymbol{v}_t} \ln P\left(W_t = w_t | W_{t+j} = w_{t+j}\right) = \boldsymbol{v}_{t+j} \left(1 - \frac{\exp\left(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)}\right)$$

Training a CBOW model



$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} logp(w_t|w_{t+j})$$

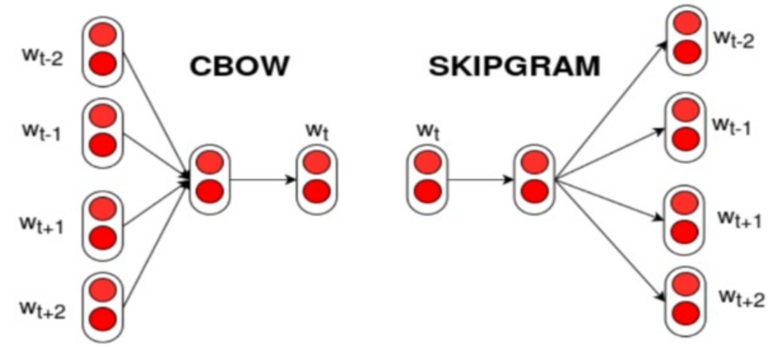
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} logp(w_{t+j}|w_t)$$

To find the parameters, we use gradient descent:

$$\nabla_{\boldsymbol{v}_t}\mathcal{L} = -\frac{1}{T}\sum_{t:W_t=w_t}\sum_{j=-c,j\neq 0}^{c}\nabla_{\boldsymbol{v}_t}\ln P\big(W_t = w_t | W_{t+j} = w_{t+j}\big)$$

$$= -\frac{1}{T}\sum_{t:W_t=w_t}\sum_{j=-c,j\neq 0}^{c}\boldsymbol{v}_{t+j}\left(1 - \frac{\exp\big(\boldsymbol{v}_t^T\boldsymbol{v}_{t+j}\big)}{\sum_{\boldsymbol{v}\in\mathcal{V}}\exp\big(\boldsymbol{v}^T\boldsymbol{v}_{t+j}\big)}\right)$$

Training a CBOW model



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

The CBOW model is trained by setting every vector equal to a weighted average of the words that occurred near it!

$$\boldsymbol{v}_t \leftarrow \boldsymbol{v}_t - \eta\nabla_{\boldsymbol{v}_t}\mathcal{L} = \boldsymbol{v}_t + \frac{\eta}{T}\sum_{t:W_t=w_t}\sum_{j=-c,j\neq 0}^{c} \boldsymbol{v}_{t+j}\left(1 - P\big(W_t = w_t|W_{t+j} = w_{t+j}\big)\right)$$

- There is more weight on words that don't yet predict well ($P\left(w_t|w_{t+j}\right)$ small).
- The weight is accumulated over the corpus, so if a word occurs near w_t often, then it gets more total weight.

Try the quiz!

Try the quiz:
https://us.prairielearn.com/pl/course_instance/147925/assessment/2411691

Outline

- What is a word? wordforms vs. lemmas vs. word senses
- Synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

Contrastive loss vs. Generative loss

- A generative loss is one like this:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-c,j\neq 0}^{c} \ln\frac{\exp\left(\boldsymbol{v}_t^T\boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}}\exp\left(\boldsymbol{v}^T\boldsymbol{v}_{t+j}\right)}$$

- Notice that this loss term compares each word, w_t, to every other word in the dictionary.
- Sometimes, generative training can take a very long time to converge.
- Sometimes, we get faster training using contrastive loss.

Contrastive loss

We train the neural network by listing, as positive examples, the words that occur in the context of "w_t =coffee," e.g.,

$$\mathcal{D}_+(w_t) = \{w_{t-3}, w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}, w_{t+3}\}$$

Create a contrastive database by choosing the same number of words, at random, from outside that context window

$$\mathcal{D}_-(w_t) = \{\text{aardvark}, \text{dog}, \text{gazebo}, \text{actor}, \text{precipitates}, \text{iceberg}\}$$

Contrastive loss

The generative loss is based on the probability of generating w_t, which, for CBOW, is:

$$P\big(W_t = w_t | W_{t+j} = w_{t+j}\big) = \frac{\exp\big(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\big)}{\sum_{v \in \mathcal{V}} \exp\big(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\big)}$$

The contrastive loss is based on the probability of the sets $\mathcal{D}_+(w_t)$ and $\mathcal{D}_-(w_t)$. For skip-gram, we could write:

$$\Pr(\mathcal{D}_+(w_t), \mathcal{D}_-(w_t) | w_t) = \prod_{w' \in \mathcal{D}_+(w_t)} \Pr(w' \in \mathcal{D}_+(w_t) | w_t) \prod_{w' \in \mathcal{D}_-(w_t)} \Pr(w' \in \mathcal{D}_-(w_t) | w_t)$$

$$= \prod_{v' \in \mathcal{D}_+(w_t)} \frac{1}{1 + e^{-\boldsymbol{v'}^T \boldsymbol{v}_t}} \prod_{v' \in \mathcal{D}_-(w_t)} \left(1 - \frac{1}{1 + e^{-\boldsymbol{v'}^T \boldsymbol{v}_t}}\right) = \prod_{v' \in \mathcal{D}_+(w_t)} \frac{1}{1 + e^{-\boldsymbol{v'}^T \boldsymbol{v}_t}} \prod_{v' \in \mathcal{D}_-(w_t)} \frac{1}{1 + e^{\boldsymbol{v'}^T \boldsymbol{v}_t}}$$

Training with contrastive loss

The coefficients $\boldsymbol{v}_t = \left[v_{t,1}, \dots, v_{t,d}\right]^T$ for each vector are chosen to maximize the log probability of the dataset:

$$\mathcal{L} = -\ln p(\text{Data}) = -\frac{1}{T}\sum_{t=1}^{T}\left(\ln p(\mathcal{D}_+(w_t)|w_t) + \ln p(\mathcal{D}_-(w_t)|w_t)\right)$$

$$= -\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{\boldsymbol{v}\prime\in\mathcal{D}_+(w_t)} \ln\frac{1}{1+e^{-\boldsymbol{v}\prime^T\boldsymbol{v}_t}} + \sum_{\boldsymbol{v}\prime\in\mathcal{D}_-(w_t)} \ln\frac{1}{1+e^{\boldsymbol{v}\prime^T\boldsymbol{v}_t}}\right)$$

Outline

- What is a word? wordforms vs. lemmas vs. word senses
- Synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

Visualizations: Similarity

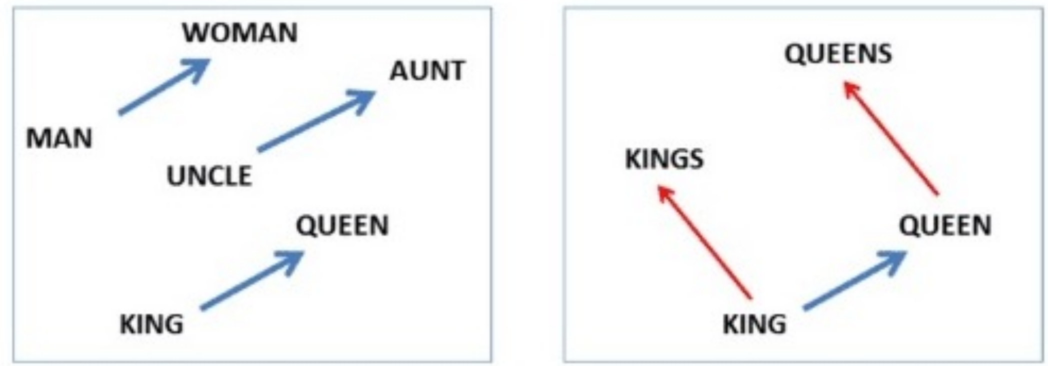
Mikolov et al. (2013) tested word2vec on SimLex-999, and had better results than previously published baselines. Here are some examples from their paper.

Model (training time)	Redmond	Havel	ninjutsu	graffiti	capitulate
Collobert (50d) (2 months)	conyers lubbock keene	plauen dzerzhinsky osterreich	reiki kohona karate	cheesecake gossip dioramas	abdicate accede rearm
Turian (200d) (few weeks)	McCarthy Alston Cousins	Jewell Arzu Ovitz	- - -	gunfire emotion impunity	- - -
Mnih (100d) (7 days)	Podhurst Harlang Agarwal	Pontiff Pinochet Rodionov	- - -	anaesthetics monkeys Jews	Mavericks planning hesitated
Skip-Phrase (1000d, 1 day)	Redmond Wash. Redmond Washington Microsoft	Vaclav Havel president Vaclav Havel Velvet Revolution	ninja martial arts swordsmanship	spray paint grafitti taggers	capitulation capitulated capitulating

Table 6: Examples of the closest tokens given various well known models and the Skip-gram model trained on phrases using over 30 billion training words. An empty cell means that the word was not in the vocabulary.

Visualizations: Relatedness

vec("woman") - vec("man") + vec("king") = vec("queen")



Christian S. Perone, "Voynich Manuscript: word vectors and t-SNE visualization of some patterns," in *Terra Incognita*, 16/01/2016, http://blog.christianperone.com/2016/01/voynich-manuscript-word-vectors-and-t-sne-visualization-of-some-patterns/.

Mikolov (2013) showed that word2vec captures similarity relationships among words. For example, the difference between the vectors for "woman" and "man" is roughly the same as the difference between the vectors for "queen" and "king." Perone (2016) showed that this effect works differently depending on the training corpus: in his blog post, he looks at word relatedness in the 15th century Voynich manuscript.

Outline

- What is a word? wordforms vs. lemmas vs. word senses
- Synonymy, similarity, and relatedness
- Vector semantics: CBOW and skip-gram
- Generative training vs. Contrastive training
- Visualizations
- Bias

Learning biased analogies from data

- It's useful that algorithms like word2vec learn appropriate analogies, like "Paris → France as Tokyo → Japan" and "kings → king as queens → queen."
- Unfortunately, it also learns other analogies that were implied in the training corpus, but that are invalid analogies.
- The paper that first demonstrated that problem was named after one of the worst such discovered analogies:

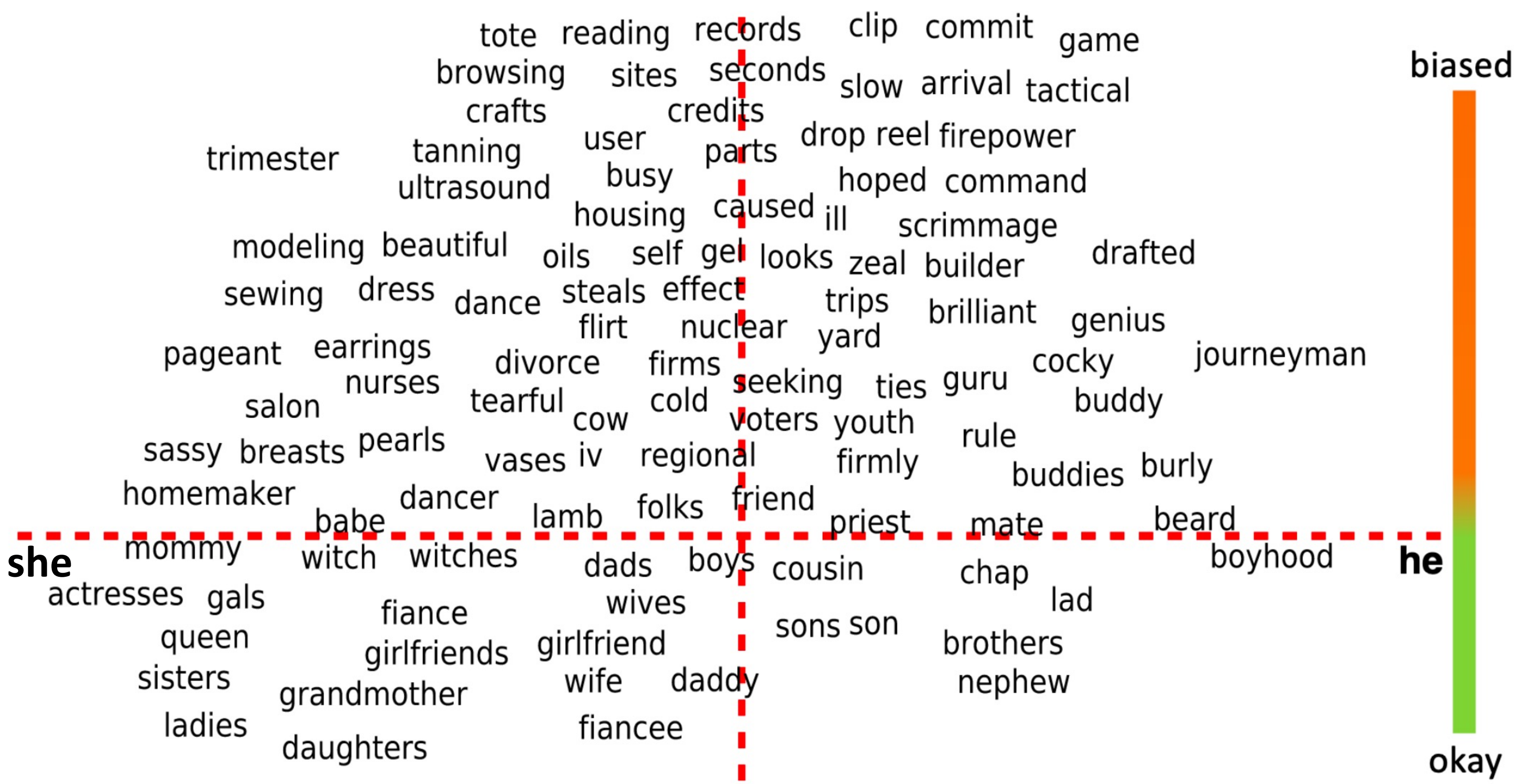
"Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings," Bolukbasi et al., 2016

Biased analogies

- Bolukbasi et al. defined a “male-female” continuum by subtracting vec(“female”)-vec(“male”), vec(“woman”)-vec(“man”), and so on, then averaging these difference vectors.
- They then created a “neutral-specific” continuum by averaging gender-specific words, averaging gender-neutral words, and subtracting.
  - Gender-specific: dictionary definition includes gender-specific language
  - Gender-neutral: all other words

The Male-Female vs. Neutral-Specific Space

Here's the resulting 2D space, from Bolukbasi et al., 2016:



Summary

- What is a word? Lemmas, wordforms, and word sense
- Synonymy, similarity, and relatedness
- Context bag-of-words (CBOW), generative loss:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-c,j\neq 0}^{c} \ln \frac{\exp\left(\boldsymbol{v}_t^T \boldsymbol{v}_{t+j}\right)}{\sum_{\boldsymbol{v}\in\mathcal{V}} \exp\left(\boldsymbol{v}^T \boldsymbol{v}_{t+j}\right)}$$

- Skip-gram, contrastive loss:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{\boldsymbol{v}\prime\in\mathcal{D}_+(w_t)} \ln \frac{1}{1+e^{-\boldsymbol{v}\prime^T \boldsymbol{v}_t}} + \sum_{\boldsymbol{v}\prime\in\mathcal{D}_-(w_t)} \ln \frac{1}{1+e^{\boldsymbol{v}\prime^T \boldsymbol{v}_t}}\right)$$

- Visualizations: vec(“aunt”)-vec(“uncle”)+vec(“king”) = vec(“queen”)
- vec(“male”)-vec(“female”): differences OK for words whose dictionary definition includes gender, but not for other words