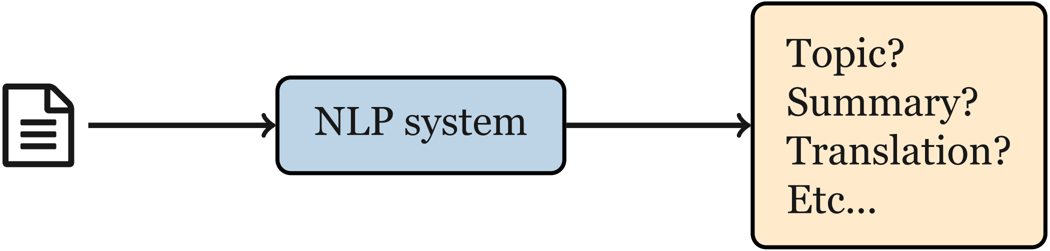
Word Embeddings

Dong Nguyen

2021

Utrecht University

Natural Language Processing (NLP)



Word representations

How can we represent the *meaning* of words?

Word representations

How can we represent the *meaning* of words?

So we can ask:

- How similar is *cat* to *dog*, or *Paris* to *London*?
- How similar is *document A* to *document B*?

Word representations

How can we represent the *meaning* of words?

So we can ask:

- How similar is *cat* to *dog*, or *Paris* to *London*?
- How similar is *document A* to *document B*?

And use such representations for:

- various NLP tasks: translation, classification, etc.
- studying linguistic questions

Dictionaries

Merriam-Webster SINCE 1828

GAMES | BROWSE THESAURUS | WORD OF THE DAY | WORDS AT PLAY

bank

DICTIONARY THESAURUS

bank noun (2)

Definition of *bank* (Entry 3 of 5)

1 **a** : an establishment for the custody, loan, exchange, or issue of money, for the extension of credit, and for facilitating the transmission of funds

// paychecks automatically deposited into the *bank*

// went to the *bank* to make a withdrawal

// open a *bank* account

b *obsolete* : the table, counter, or place of business of a money changer

2 : a person conducting a gambling house or game

specifically : DEALER

3 : a supply of something held in reserve: such as

a *in games* : the fund of supplies (such as money, chips, or pieces) held by the banker (see BANKER entry 1 sense 2) or dealer

b *in games* : a fund of pieces (such as dominoes) from which the players draw

// select another domino from the *bank*

4 : a place where something is held available

// memory *banks*

especially : a depot for the collection and storage of a biological product

// a blood *bank*

WordNet

bank Noun

- **bank** (sloping land (especially the slope beside a body of water)) "*they pulled the canoe up on the bank*"; "*he sat on the bank of the river and watched the currents*"
- depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) "*he cashed a check at the bank*"; "*that bank holds the mortgage on my home*"
- ...

Verb

- **bank** (tip laterally) "*the pilot had to bank the aircraft*"
- **bank** (do business with a bank or keep an account at a bank) "*Where do you bank in this town?*"
- ...

https://wordnet.princeton.edu

WordNet

bank Noun

- **bank** (sloping land (especi … *pulled the canoe up on the bank*"; "*he* … *he currents*"
- depository financial institu… …pany (a financial institution that ac… … lending activities) "*he cashed a che*… …*age on my home*"
- …

Verb

- **bank** (tip laterally) "*the pilot had to bank the aircraft*
- **bank** (do business with a bank or keep an account at a bank) "*Where do you bank in this town?*"
- …

Unfortunately, dictionaries and knowledge bases are hard to maintain and have limited coverage

😔

https://wordnet.princeton.edu

recap!

Vector representations

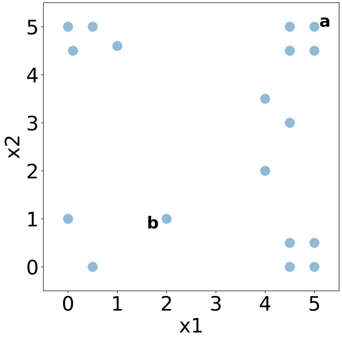


Figure: Points in a two dimensional vector space

```
a = [5, 5]
b = [2, 1]
```

a is a *two-dimensional* vector

recap!

Vector representations

```
a = [5, 5, 2]
b = [2, 1, 0]
```

a is a *three-dimensional* vector

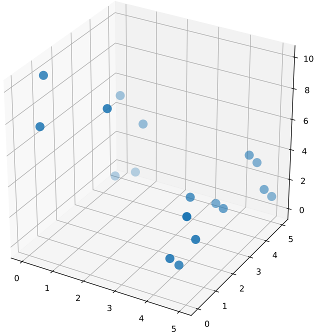


Figure: Points in a three dimensional vector space

recap!

Vector representations

```
a = [5, 5, 2]
b = [2, 1, 0]
```

a is a *three-dimensional* vector

Key idea in NLP:

Can we **represent words as vectors** (i.e. points in a vector space?)

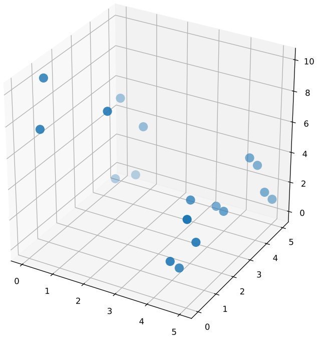


Figure: Points in a three dimensional vector space

Word as vectors

Key idea: Can we represent words as vectors?

The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

Word as vectors

Key idea: Can we represent words as vectors?

The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

How similar are *smart* and *intelligent?* (not similar 0–10 very similar):
How similar are *easy* and *big* (not similar 0–10 very similar):

Word as vectors

Key idea: Can we represent words as vectors?

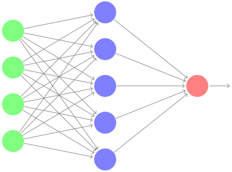
The vector representations should:

- capture semantics
  - similar words should be close to each other in the vector space
  - relation between two vectors should reflect the relationship between the two words
- be efficient (vectors with fewer dimensions are easier to work with)
- be interpretable

How similar are *smart* and *intelligent?* (not similar 0–10 very similar): 9.2
How similar are *easy* and *big* (not similar 0–10 very similar): 1.12
(*SimLex-999 dataset*)

How are they used?

How are they used?



In neural networks (text classification, sequence tagging, etc..)

cat	0.52	0.48	-0.01	···	0.28
dog	0.32	0.42	-0.09	···	0.78

As research objects

Properties

We can use cosine similarity to find similar words in the vector space.

- **dog**: *dogs, cat, man, cow, horse*
- **car**: *driver, cars, automobile, vehicle, race*
- **amsterdam**: *netherlands, rotterdam, dutch, centraal, paris*
- **chocolate**: *candy, beans, caramel, butter, liquor*

Exercise (5 min)

- Go to `https://projector.tensorflow.org/`. The site should load 'Word2Vec 10K' vectors by default (see left panel).
- What are the 5 nearest words to '*cat*'?
- What are the 5 nearest words to '*computer*'?

Words as vectors

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
$\rightarrow$ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
→ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

What are limitations of one hot encodings?

One hot encoding

Map each word to a unique identifier
e.g. *cat (3)* and *dog (5)*.
$\rightarrow$ Vector representation: all zeros, except 1 at the ID

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

Even related words have distinct vectors!

High number of dimensions

😔

Distributional hypothesis

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

Distributional hypothesis

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

What is a **wampos**?

Distributional hypothesis

Figure: Photo by Piekfrosch; CC-BY-SA-3.0

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

wampos = *pangolin*

You shall know a word by the company it keeps (Firth, J. R. 1957:11)

Distributional hypothesis

Figure: Photo by Piekfrosch; CC-BY-SA-3.0

some believe that	**wampos**	scales have medicinal qualities
approach to fighting	**wampos**	(and general wildlife) trafficking
Even though	**wampos**	scales are made of exactly the

wampos = *pangolin*

You shall know a word by the company it keeps (Firth, J. R. 1957:11)

The distributional hypothesis: Words that occur in similar contexts tend to have similar meanings

Word vectors based on co-occurrences

documents as context
word-document matrix

	doc_1	doc_2	doc_3	doc_4	doc_5	doc_6	doc_7
cat	5	2	0	1	4	0	0
dog	7	3	1	0	2	0	0
car	0	0	1	3	2	1	1

Word vectors based on co-occurrences

documents as context
word-document matrix

	doc_1	doc_2	doc_3	doc_4	doc_5	doc_6	doc_7
cat	5	2	0	1	4	0	0
dog	7	3	1	0	2	0	0
car	0	0	1	3	2	1	1

neighboring words as context
word-word matrix

	cat	dog	car	bike	book	house	tree
cat	0	3	1	1	1	2	3
dog	3	0	2	1	1	3	1
car	0	0	1	3	2	1	1

Word vectors based on co-occurrences

There are many variants:

- Context (words, documents, which window size, etc.)
- Weighting (raw frequency, etc.)

Vectors are sparse: Many zero entries.
Therefore: Dimensionality reduction is often used (e.g., SVD)

These methods are sometimes called **count-based** methods as they work directly on **co-occurrence** counts.

Word embeddings

Word embeddings

Word embeddings:

- Vectors are short; typically 50-1024 dimensions ☺
- Vectors are dense (mostly non-zero values)
- Very effective for many NLP tasks ☺
- Individual dimensions are less interpretable ☹

cat	0.52	0.48	-0.01	...	0.28
dog	0.32	0.42	-0.09	...	0.78

Agenda

- ~~What are word embeddings?~~
- How do we learn word embeddings?
- Properties of word embeddings
- Evaluation
- Biases in word embeddings
- Application: analyzing semantic change

Agenda

- ~~What are word embeddings?~~
- How do we learn word embeddings?
- Properties of word embeddings
- Evaluation
- Biases in word embeddings
- Application: analyzing semantic change

Aside: terminology

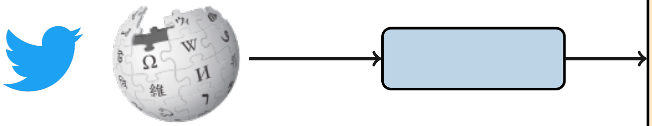
Distributed representations
Dense, real-valued vectors

Distributional representations
Derived from how words are used in a corpus

Word2vec (and others) are distributed representations based on the distributional hypothesis.

How do we learn word embeddings?

Learning word embeddings



cat =	**0.12**	. . .	**-0.2**
dog =	**0.92**	. . .	**-0.1**
tree =	**-0.12**	. . .	**0.1**
. . .	. . .	. . .	. . .

Learning word embeddings

Word2vec, GloVe, fastText

cat =	**0.12**	...	**-0.2**
dog =	**0.92**	...	**-0.1**
tree =	**-0.12**	...	**0.1**
...	...	...	...

Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Key idea:

Use text itself as training data for the model!

A form of *self-supervision*.

Training data

How can we train a model to learn the meaning of words?
Which data can we use for supervised learning?

Key idea:

Use text itself as training data for the model!

A form of *self-supervision*.

Example: Train a neural network to predict the next word given previous words.

A neural probabilistic language model. Bengio et al. (2003), JMLR [url]

Natural language processing (almost) from scratch, Collobert et al. (2011), JMLR, [url]

Exercise: Word prediction task

`yesterday I went to the` **?**

`A new study has highlighted the positive` **?**

Which word comes next?

Word2Vec

The domestic **cat** is a small, typically furry carnivorous mammal

w_{-2} w_{-1} w_0 w_1 w_2 w_3 w_4 w_5

We have **target** words (*cat*) and **context** words (here: window=5).

Remember: distributional hypothesis

Word2Vec

Two different tasks (context):

- Continuous Bag-Of-Words (CBOW)
- Skipgram

Two training regimes

- Hierachical softmax
- Negative sampling

https://code.google.com/archive/p/word2vec/

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013 [url]

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013 [url]

Word2Vec

Two different tasks (context):

- Continuous Bag-Of-Words (CBOW)
- Skipgram

Two training regimes

- Hierachical softmax
- Negative sampling

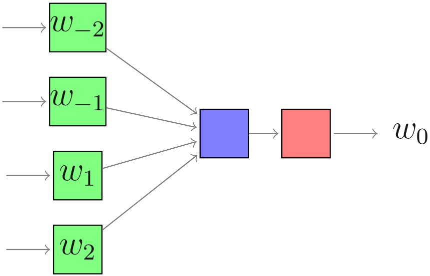
https://code.google.com/archive/p/word2vec/

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013 [url]

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013 [url]

Word2Vec

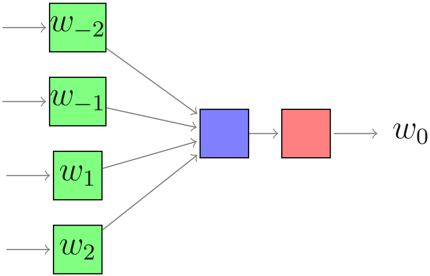
Continuous Bag-Of-Words (CBOW)



`one snowy ? she went`

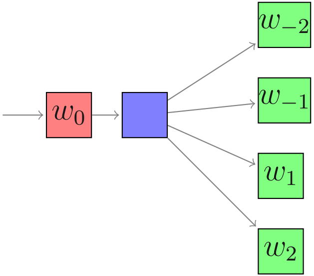
Word2Vec

Continuous Bag-Of-Words (CBOW)



`one snowy ? she went`

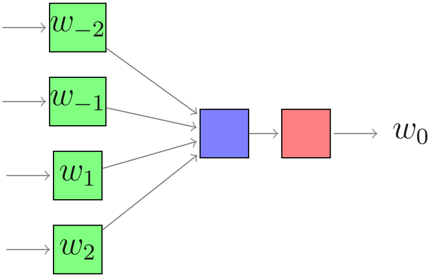
skipgram



`? ? day ? ?`

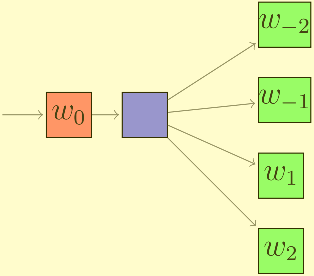
Word2Vec

Continuous Bag-Of-Words (CBOW)



`one snowy ? she went`

skipgram



`? ? day ? ?`

Word2Vec

Two different tasks (context:

- Continuous Bag-Of-Words (CBOW)
- Skipgram

Two training regimes

- Hierachical softmax
- Negative sampling

https://code.google.com/archive/p/word2vec/

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013 [url]

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013 [url]

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	**context (c)**	**label**
cat	small	1
cat	furry	1
cat	car	0
...	...	...

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	**context (c)**	**label**
cat	small	1
cat	furry	1
cat	car	0
…	…	…

1. **Create examples**
   - Positive examples: Target word and neighboring context
   - Negative examples: Target word and randomly sampled words from the lexicon (*negative sampling)*
2. Train a **logistic regression** model to distinguish between the positive and negative examples
3. The resulting **weights** are the embeddings!

Word2Vec: skipgram overview

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	context (c)	label
cat	small	1
cat	furry	1
cat	car	0
…	…	…

Embedding vectors are essentially a byproduct!

1. **Create examples**
   - Positive examples: Target word and neighboring context
   - Negative examples: Target word and randomly sampled words from the lexicon (*negative sampling)*
2. Train a **logistic regression** model to distinguish between the positive and negative examples
3. The resulting **weights** are the embeddings!

Word2Vec: skipgram

The domestic **cat** is a small, typically furry carnivorous mammal
$c1$ $c2$ w $c3$ $c4$ $c5$ $c6$ $c7$

We have **target** words (*cat*) and **context** words (here: window=5).

The probability that c is a real context word:

$$P(+|w,c)$$

The probability that c is not a real context word:

$$P(-|w,c)$$

See also: 6.8 of Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

Word2Vec: skipgram

Intuition: A word c is likely to occur near the target if its embedding is similar to the target embedding.

$$\approx w \cdot c$$

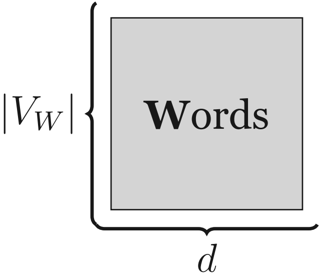
Turn this into a probability using the sigmoid function

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

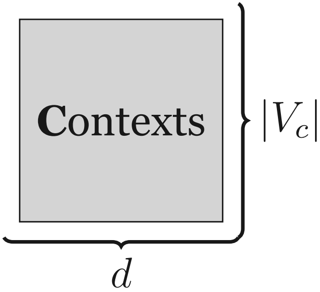
See also: 6.8 of Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

Word2Vec

Words:
Each word w is represented as a d-dimensional vector.



Contexts:
Each word w is represented as a d-dimensional vector.



All vectors are initialized with random weights.

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

During training:

- *Maximize* the similarity between the embeddings of the target word and context words from the positive examples
- *Minimize* the similarity between the embeddings of the target word and context words from the negative examples

Word2vec: skipgram (learning)

We **start** with random embedding vectors.

During training:

- *Maximize* the similarity between the embeddings of the target word and context words from the positive examples
- *Minimize* the similarity between the embeddings of the target word and context words from the negative examples

After training:

- frequent word-context pairs in data: $w \cdot c$ high
- not word-context pairs in data: $w \cdot c$ low

So: Words occurring in same contexts are close to each other

Word2vec: skipgram (learning)

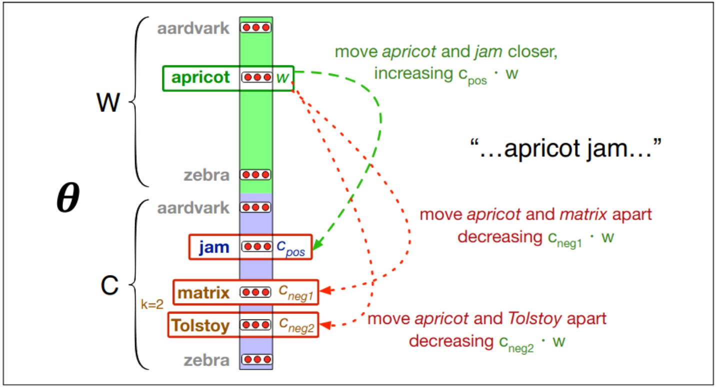


Figure: Figure 6.14 from Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin
https://web.stanford.edu/~jurafsky/slp3/

fastText

Limitation of word2vec: Can't handle unknown words :(

fastText is very similar to word2vec, but each word is **represented as a bag of character n-grams** (+ the word itself). $\leq$ and $\geq$ mark word boundaries.

Example: *where* with $n = 3$: `<wh, whe, her, ere, re>` and `<where>`

Representation of a word: The sum of the vector representations of its n-grams.

Enriching Word Vectors with Subword Information, Bojanowski et al., TACL 2017, [url], software: https://fasttext.cc/

GloVe

- First create a *global word-word co-occurrence matrix* (how frequent pairs of words occur with each other). Requires a pass through the entire corpus at the start!
- Training objective: learn word embeddings so that their dot products equals the log of the words' co-occurrence probability.

GloVe: Global Vectors for Word Representation, Pennington et al., EMNLP 2015 [url], software `https://nlp.stanford.edu/projects/glove/`

Pre-trained embeddings

- I want to build a system to solve a task (e.g. sentiment analysis)
  - Use pre-trained embedddings. Should I fine-tune?
    - Lots of data: yes
    - Just a small dataset: no
- Analysis (e.g. bias, semantic change)
  - Train embeddings from scratch

Agenda

- ~~What are word embeddings?~~
- ~~How do we learn word embeddings?~~
- Properties of word embeddings
- Evaluation
- Biases in word embeddings
- Application: analyzing semantic change

Properties of word embeddings

Properties of word embeddings

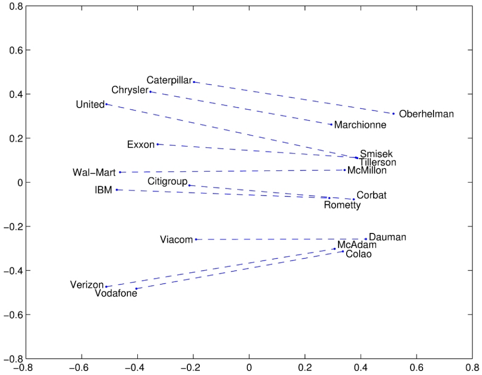


Figure: company - ceo

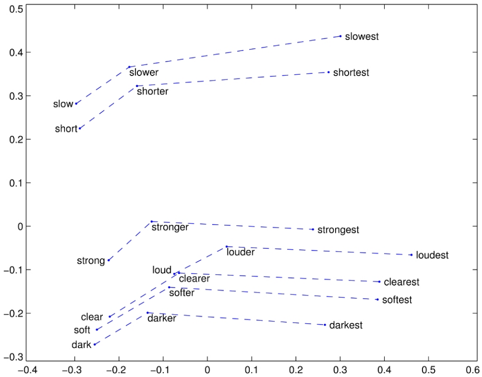


Figure: comparative - superlative

Source: https://nlp.stanford.edu/projects/glove/

Properties of word embeddings: analogies

We can look at analogies in the vector space, for example:
$king - man + woman \approx queen$

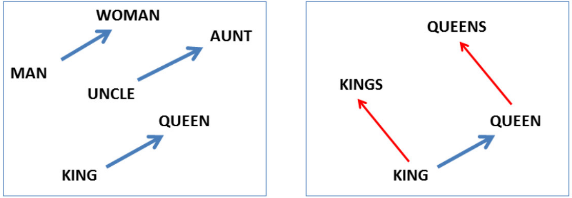
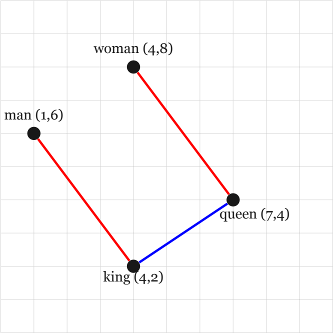


Figure: Figure 2 from Linguistic Regularities in Continuous Space Word Representations, Mikolov et al. NAACL 2013 [url]

Properties of word embeddings: analogies

We can look at analogies in the vector space, for example:
king - man + woman ≈ *queen*




```
king-man = [4,2] - [1,6] = [3,-4]
king-man + woman = [3,-4] + [4,8] = [7,4]
```

Stability of embeddings

Many factors can have an effect on the training (corpus size, presence/absence of documents, etc...). How *stable* are embeddings?

Measures of stability: One simple method is looking at the overlap between nearest neighbors in an embedding space

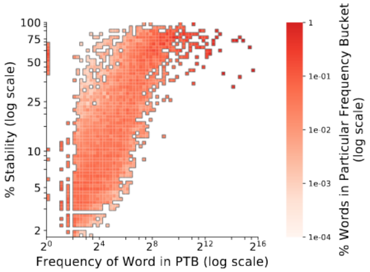


Figure: *word2vec* embeddings: lower frequency words have lower stability and higher frequency words have higher stability (Figure 1 from Wendlandt et al. 2018)

Factors Influencing the Surprising Instability of Word Embeddings, Wendlandt et al., NAACL 2018 [url]

Stability of embeddings

Antoniak and Mimno et al. 2018:

- The training corpus is only a sample!
- But: they were sensitive to the presence of specific documents
- “*with smaller corpora comes greater variability, and we recommend that practitioners use bootstrap sampling to generate an ensemble of word embeddings for each sub-corpus and present both the mean and variability of any summary statistics*”

Evaluating the Stability of Embedding-based Word Similarities, Antoniak and Mimno, TACL 2018 [url]

recap!

Design decision: context

The distributional hypothesis: Words that occur in similar contexts tend to have similar meanings.

recap!

Design decision: context

The distributional hypothesis: Words that occur in similar contexts tend to have similar meanings.

How do we define our **context**?

Context

`Australian scientist` **`discovers`** `star with telescope`

context window = 1

Context

`Australian scientist` **`discovers`** `star with telescope`

context window = 2

Context

`Australian scientist` **`discovers`** `star with telescope`

context window = sentence

Context

`Australian scientist` **`discovers`** `star with telescope`

context window = sentence

Smaller contexts → syntactic properties
Large contexts → semantic/topical properties

Example Levy and Golbert, ACL 2014 for *hogwarts*:
window=2: *evernight* and *sunnydale* vs. window=5: *dumbledore*, *hallows*

(Levy and Golbert, ACL 2014; Melamud, NAACL 2016; and others)

Agenda

- ~~What are word embeddings?~~
- ~~How do we learn word embeddings?~~
- ~~Properties of word embeddings~~
- Evaluation
- Biases in word embeddings
- Application: analyzing semantic change

Evaluation

Evaluation

Types of evaluation

1. Extrinstic evaluation
2. Intrinsic evaluation

Evaluation

Types of evaluation

1. Extrinstic evaluation
2. Intrinsic evaluation

Evaluation based on performance on *external* tasks (e.g., part of speech tagging, sentiment analysis)

I.e. plug in different embeddings into the same NLP system and measure difference in task performance.

Evaluation

Types of evaluation

1. Extrinstic evaluation
2. Intrinsic evaluation

Evaluations based on *only* the embeddings.

0.12	. . .	-0.2

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

Input: Dataset with relatedness or similarity scores for pairs of words.

Goal High (pearson or spearman) correlation between scores and the cosine similarity of the embeddings for the two words.

Example from *WordSim353*:
wood and *forest*: 7.73
money and *cash*: 9.15
month and *hotel*: 1.81

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

```
Base/3rd Person Singular Present
see:sees return: ?

Singular/Plural
year:years law: ?

Meronyms
player:team fish: ?

UK city county
york:yorkshire Exeter: ?
```

(Mikolov et al. 2013 [url]; Gladkova et al. 2016 [url])

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

This method is referred to by Levy and Goldberg (2014) as **3COSADD**
$\mathbf{a} - \mathbf{a}^* \approx \mathbf{b} - \mathbf{b}^*$. We can find $\mathbf{b}^*$ as follows:

$$\underset{\mathbf{b}^* \in V}{\operatorname{argmax}}\, cos(\mathbf{b}^*, \mathbf{b} - \mathbf{a} + \mathbf{a}^*)$$

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

This method is referred to by Levy and Goldberg (2014) as **3COSADD**
$\mathbf{a} - \mathbf{a}^* \approx \mathbf{b} - \mathbf{b}^*$. We can find $\mathbf{b}^*$ as follows:

$$\operatorname*{argmax}_{\mathbf{b}^* \in V} cos(\mathbf{b}^*, \mathbf{b} - \mathbf{a} + \mathbf{a}^*)$$

Linzen 2016 notes that results can be misleading: The offsets are often very small, so that often just the nearest neighbor to $\mathbf{b}$ is returned.
Control setting: Just return the nearest neighbor of $\mathbf{b}$

Issues in evaluating semantic spaces using word analogies, Tal Linzen. 2016 [url]

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

Cluster the words based on their embeddings and compare them against a known categorization.

Evaluation methods for unsupervised word embeddings, Schnabel et al. EMNLP 2015 [url]

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

Are words in the neighborhood of the *query* word mutually related? Present four words (query word + two close neighbors + intruder). Task: identify the intruder (e.g. Turkers).

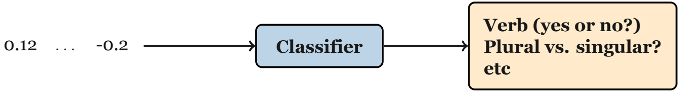
Example: a) *finally*; (b) *eventually*; (c) *immediately*; (d) *put*

Evaluation methods for unsupervised word embeddings, Schnabel et al. EMNLP 2015 [url]

Intrinsic evaluation

- Similarity
- Analogies
- Clustering
- Coherence
- Probing classifiers

Also called *diagnostic classifiers*



Mostly used to evaluate sentence embeddings, but sometimes also used for analyzing word embeddings.

But, be careful! Performance might seem high, but classifier might learn other signals (e.g. word frequency, part of speech classes) than what you focus on.

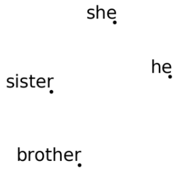
What you can cram into a single $&!# vector: Probing sentence embeddings for linguistic properties, Conneau et al., ACL 2018 [url]

Agenda

- ~~What are word embeddings?~~
- ~~How do we learn word embeddings?~~
- ~~Properties of word embeddings~~
- ~~Evaluation~~
- Biases in word embeddings
- Application: analyzing semantic change

Biases in word embeddings

Biases in word embeddings



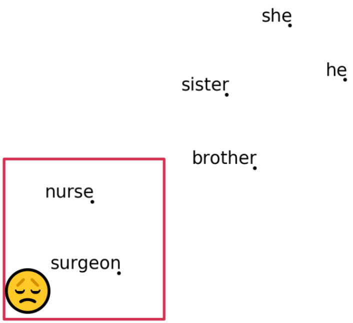
Measuring gender bias:

- To assess NLP models and investigate the impact of 'bias mitigation' techniques
- To study societal trends

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Biases in word embeddings



Pre-trained GloVe model on Twitter

Measuring gender bias:

- To assess NLP models and investigate the impact of 'bias mitigation' techniques
- To study societal trends

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Biases reflected in analogy tasks

Biases reflected in analogy tasks:

man is to *computer programmer* as *woman* is to ? : x = homemaker
father is to *doctor* as *mother* is to ? : x = nurse

Note: Input words are excluded as possible answers! (see also Nissim et al. 2020 [url])

Compare: gender-specific words (e.g., *brother*, *businesswoman*) vs. *gender-neutral* words (e.g. *nurse*, *teacher*).

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi, et al. NIPS 2016 [url]

Biases in word embeddings

Find gender analogies. We want to find pairs that are parallel to the seed direction and its words should be close to each other.

$$S_{(a,b)}(\mathrm{x,y}) = cos(\mathrm{a} - \mathrm{b}, x - y) \quad if \quad \|x - y\|_2 \leq \delta$$

embedding$_{she}$ embedding$_{he}$ L_2 distance

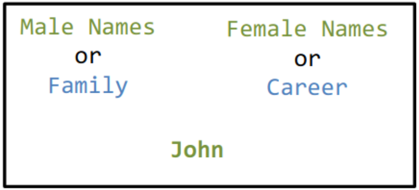
Gender appropriate she-he analogies
queen–king
sister–brother
ovarian cancer–prostate cancer
mother–father
convent–monastery

Gender stereotype she-he analogies
nurse–surgeon
sassy–snappy
cupcakes–pizzas
lovely–brilliant
vocalist–guitarist

Bolukbasi et al. look at 300-dimensional embeddings from w2vec Google news corpus.

Word-Embedding Association Test

- The Implicit Association Test (IAT) is based on response times and has been widely used.



Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Word-Embedding Association Test

- The Implicit Association Test (IAT) is based on response times and has been widely used.
- Word-Embedding Association Test (WEAT) by Caliskan et al: use the cosine similarity between pairs of vectors as analogous to reaction time in the IAT

Were able to replicate well-known IAT findings!

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Word-Embedding Association Test

Let X and Y be two sets of **target words** of equal size;
Let A, B be the two sets of **attribute words**.
For a given target word w we get a score:

$$s(w, A, B) = mean_{a \in A} cos(\vec{w}, \vec{a}) - mean_{b \in B} cos(\vec{w}, \vec{b})$$

Target words X—flowers: aster, clover, hyacinth, crocus, rose, ...
Target words Y—insects: ant, caterpillar, flea, spider, bedbug, ...
Attribute words A—pleasant: freedom, love, peace, cheer, ...
Attribute words B—unpleasant: abuse, crash, filth, murder, divorce,...

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Word-Embedding Association Test

Let X and Y be two sets of **target words** of equal size;
Let A, B be the two sets of **attribute words**.
For a given target word w we get a score:

$$s(w, A, B) = mean_{a \in A} cos(\vec{w}, \vec{a}) - mean_{b \in B} cos(\vec{w}, \vec{b})$$

Target words X—math: math, algebra, numbers, calculus, ...
Target words Y—arts: poetry, art, dance, literature, ...
Attribute words A—male: male, man, boy, brother, he, him, ...
Attribute words B—female: female, woman, girl, sister, she, her,..

Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Word-Embedding Association Test

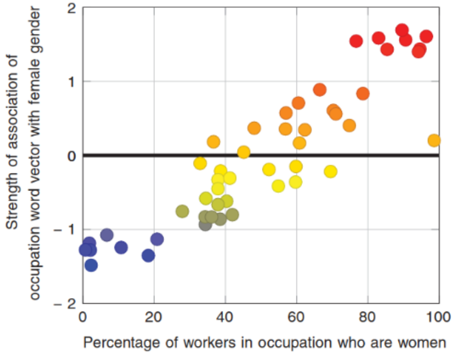


Fig. 1. Occupation-gender association. Pearson's correlation coefficient $\rho = 0.90$ with $P < 10^{-18}$.

Figure from: Semantics derived automatically from language corpora contain human-like biases, Caliskan, Bryson, Narayanan, Science 2017 [url]

Perpetuation of bias in sentiment analysis

> *"I had tried building an algorithm for sentiment analysis based on word embeddings [..]. When I applied it to restaurant reviews, I found it was ranking Mexican restaurants lower. The reason was not reflected in the star ratings or actual text of the reviews. It's not that people don't like Mexican food.* ***The reason was that the system had learned the word "Mexican" from reading the Web****."*

(emphasis mine)

http://blog.conceptnet.io/posts/2017/conceptnet-numberbatch-17-04-better-less-stereotyped-word-vectors/

Agenda

- ~~What are word embeddings?~~
- ~~How do we learn word embeddings?~~
- ~~Properties of word embeddings~~
- ~~Evaluation~~
- ~~Biases in word embeddings~~
- Application: analyzing semantic change

Application:
analysis of semantic change

Applications: Semantic change

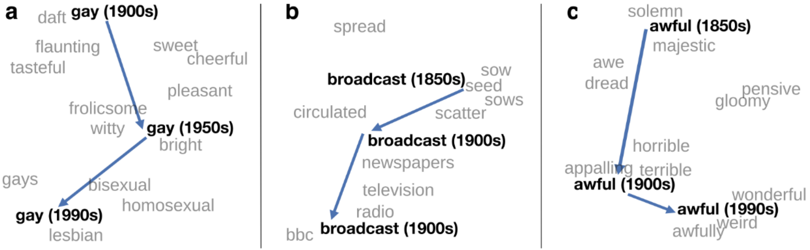
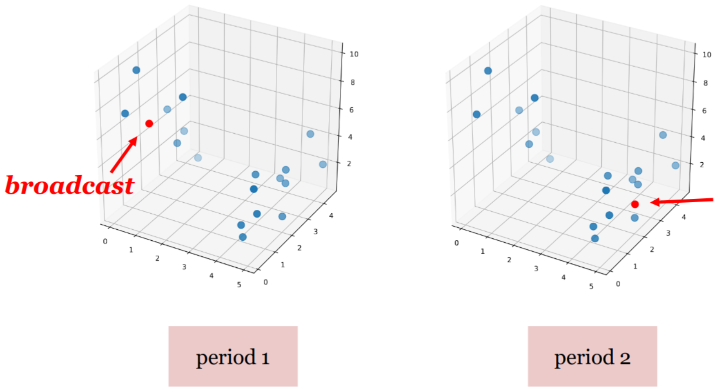


Figure 1. from Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change, Hamilton et al., ACL 2016 [url]

Tracking change in embedding space



Dong Nguyen (2021)

Semantic change in social media

lit

CBS News
@CBSNews

New York City's Rockefeller Center Christmas tree lit up for the holidays Wednesday night 🎄🗽

The College Board
@CollegeBoard

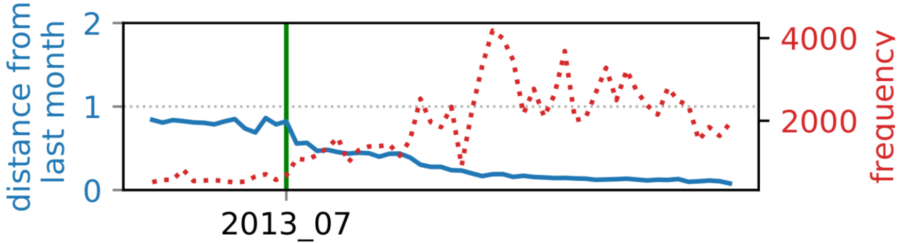
Good luck to all the AP students taking their AP Chemistry, AP Spanish Lit, AP German, and AP Psychology Exams today! ⚗️ 📖 🇩🇪 🧠

Lewis Hamilton
@LewisHamilton

Roscoe's birthday party last night was lit 🔥

Semantic change: *glo*

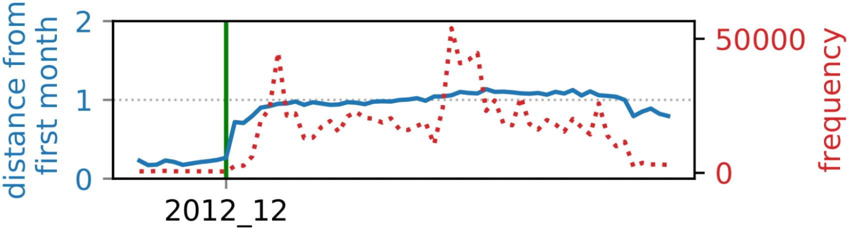
August 2013 rapper Chief Keef
released "Gotta Glo Up One Day"



P. Shoemark*, F. F. Liza*, D. Nguyen, S. A. Hale, B. McGillivray. Room to glo: A systematic comparison of semantic change detection approaches with word embeddings, EMNLP 2019 [url]

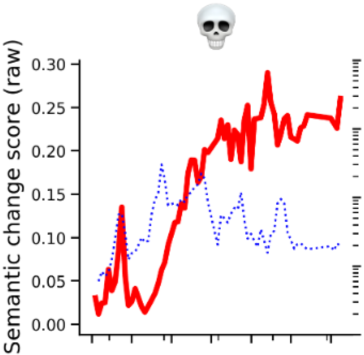
Semantic change: *vine*

Video hosting service was launched in January 2013



P. Shoemark*, F. F. Liza*, D. Nguyen, S. A. Hale, B. McGillivray. Room to glo: A systematic comparison of semantic change detection approaches with word embeddings, EMNLP 2019 [url]

Semantic change: emojis



2012: *zombie, corpse, bury, undead, murder*
2013–: *lmao* and similar terms.

A. Robertson, F. Ferdousi Liza, D. Nguyen, B. McGillivray, S. A. Hale. Semantic Journeys: Quantifying Change in Emoji Meaning from 2012–2018, 4th International Workshop on Emoji Understanding and Applications in Social Media 2021 [url]

Resources

Resources

Readings:

- *Contextual Word Representations: Putting Words into Computers*, Noah A. Smith, 2020 https://cacm.acm.org/magazines/2020/6/245162-contextual-word-representations/fulltext
- *Vector Semantics and Embeddings (Chapter 6)*, Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin, 2020 https://web.stanford.edu/~jurafsky/slp3/

Video's:

- *Stanford CS224N: NLP with Deep Learning | Winter 2019 | Lecture 1 – Introduction and Word Vectors* (and lecture 2): https://www.youtube.com/watch?v=8rXD5-xhemo
- video's by Jordan Boyd-Graber, e.g. *Understanding Word2Vec* https://www.youtube.com/watch?v=QyrUentbkvw and others

Resources: blogposts

- *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)* by Jay Alammar
  http://jalammar.github.io/illustrated-bert/ (2018)
- *The Illustrated Word2vec* by Jay Alammar
  http://jalammar.github.io/illustrated-word2vec/ (2019)
- *Generalized Language Models* by Lilian Weng
  https://lilianweng.github.io/lil-log/2019/01/31/generalized-language-models.html

Software

- **word2vec**: gensim (https://radimrehurek.com/gensim/) and official implementation (https://code.google.com/archive/p/word2vec/).
- **fasttext**: official implementation (https://fasttext.cc/)
- **GloVe**: official implementation (https://nlp.stanford.edu/projects/glove/)