INF5820
Distributional Semantics: Extracting Meaning from Data

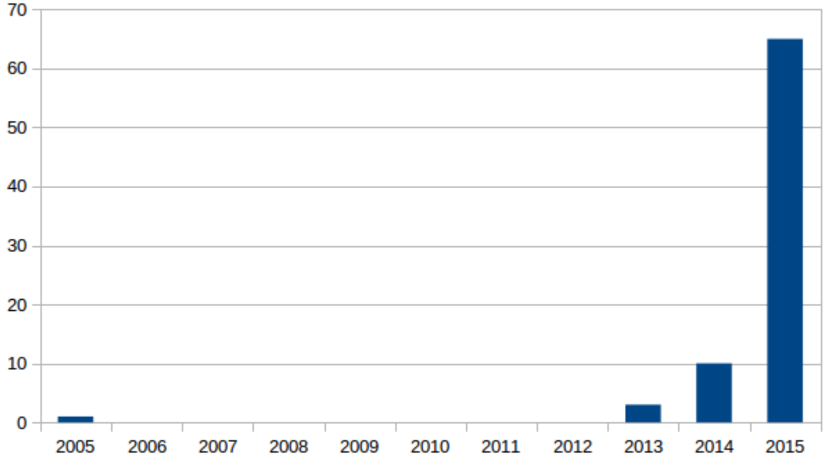
Lecture 1: Linguistic Foundations of Distributional Semantics

Andrey Kutuzov
andreku@ifi.uio.no

26 October 2016

Contents

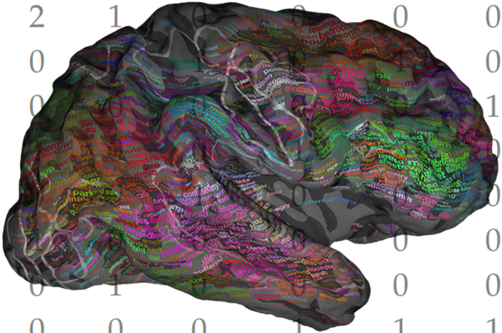
Our motivation



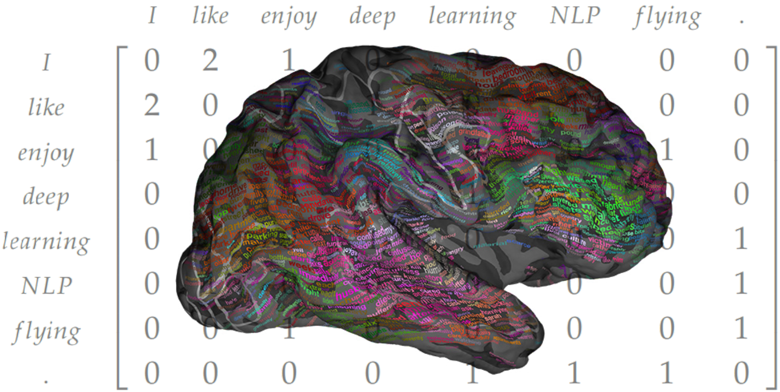
Number of publications on word embeddings in Association for Computational Linguistics Anthology (http://aclanthology.info/)

Mapping words in brain

$$
\begin{array}{c|cccccccc}
 & I & like & enjoy & deep & learning & NLP & flying & . \\
\hline
I & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
like & 2 & 0 & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & 0 & 0 \\
enjoy & 1 & 0 & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & 1 & 0 \\
deep & 0 & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & 0 & 0 \\
learning & 0 & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & 0 & \text{[illegible]} & 0 & 1 \\
NLP & 0 & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & \text{[illegible]} & 0 & 0 & 1 \\
flying & 0 & 0 & 1 & \text{[illegible]} & \text{[illegible]} & 0 & 0 & 1 \\
. & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

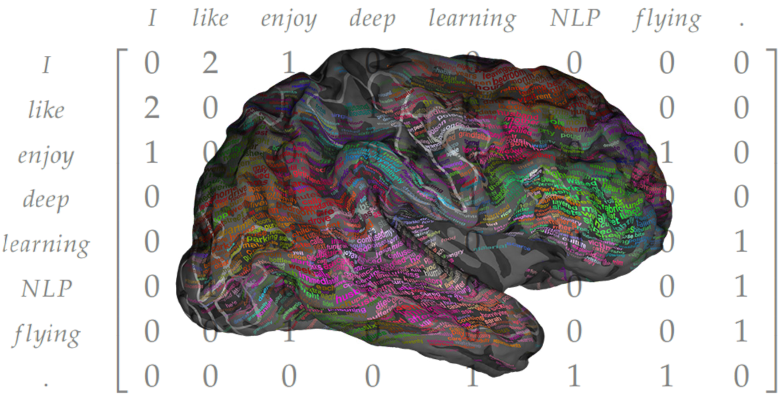


Mapping words in brain



We want a machine to imitate human brain and understand meaning of words.

Mapping words in brain



We want a machine to imitate human brain and understand meaning of words.
How we can design it?

- Vector space models of meaning (based on distributional semantics) have been here for already several decades [Turney et al., 2010].

- Vector space models of meaning (based on distributional semantics) have been here for already several decades [Turney et al., 2010].
- Recent advances in employing machine learning to train distributional models allowed them to become state-of-the-art and literally conquer the computational linguistics landscape.

- Vector space models of meaning (based on distributional semantics) have been here for already several decades [Turney et al., 2010].
- Recent advances in employing machine learning to train distributional models allowed them to become state-of-the-art and literally conquer the computational linguistics landscape.
- Now they are commonly used both in research and in large-scale industry projects (web search, opinion mining, tracing events, plagiarism detection, document collections management etc.)

- Vector space models of meaning (based on distributional semantics) have been here for already several decades [Turney et al., 2010].
- Recent advances in employing machine learning to train distributional models allowed them to become state-of-the-art and literally conquer the computational linguistics landscape.
- Now they are commonly used both in research and in large-scale industry projects (web search, opinion mining, tracing events, plagiarism detection, document collections management etc.)
- All this is based on the ability of such models to efficiently calculate semantic similarity between linguistic entities.

- Vector space models of meaning (based on distributional semantics) have been here for already several decades [Turney et al., 2010].
- Recent advances in employing machine learning to train distributional models allowed them to become state-of-the-art and literally conquer the computational linguistics landscape.
- Now they are commonly used both in research and in large-scale industry projects (web search, opinion mining, tracing events, plagiarism detection, document collections management etc.)
- All this is based on the ability of such models to efficiently calculate semantic similarity between linguistic entities.
- In this course, we will cover why and how distributional models actually work.

Contents

Simple demo

Distributional semantic models for English (and Norwegian)

http://ltr.uio.no/semvec

You can entertain yourself during the lecture :-)
Later we will look closer at the features of this service.

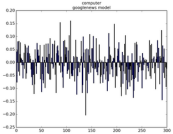
Semantic Vectors | Similar words | Visualizations | Calculator | About

What words are related to "computer" in the googlenews?

1. computers 0.79794
2. laptop 0.66405
3. laptop_computer 0.65489
4. Computer 0.64733
5. com_puter 0.60821
6. technician_Leonard_Luchko 0.56627
7. mainframes_minicomputers 0.56177
8. laptop_computers 0.55854
9. PC 0.55396
10. maker_Dell_DELL.O 0.55193

Show/hide raw vector of "computer" in model googlenews:

About the word



- Search "computer" in the Internet
- "computer" in the Wiktionary

Language Technology Group

Contents

Distributional hypothesis

Tiers of linguistic analysis

Distributional hypothesis

Tiers of linguistic analysis

Computational linguistics can comparatively easy model lower tiers of language:

- graphematics – how words are spelled

Distributional hypothesis

Tiers of linguistic analysis

Computational linguistics can comparatively easy model lower tiers of language:

- graphematics – how words are spelled
- phonetics – how words are pronounced

Distributional hypothesis

Tiers of linguistic analysis

Computational linguistics can comparatively easy model lower tiers of language:

- graphematics – how words are spelled
- phonetics – how words are pronounced
- morphology – how words inflect

Distributional hypothesis

Tiers of linguistic analysis

Computational linguistics can comparatively easy model lower tiers of language:

- graphematics – how words are spelled
- phonetics – how words are pronounced
- morphology – how words inflect
- syntax – how words interact in sentences

Distributional hypothesis

To model means to densely represent important features of some phenomenon. For example, in a phrase *'The judge sits in the court'*, the word *'judge'*:

Distributional hypothesis

To model means to densely represent important features of some phenomenon. For example, in a phrase *'The judge sits in the court'*, the word *'judge'*:

1. consists of 3 phonemes [j e j];

Distributional hypothesis

To model means to densely represent important features of some phenomenon. For example, in a phrase *'The judge sits in the court'*, the word *'judge'*:

1. consists of 3 phonemes [j e j];
2. is a singular noun in the nominative case;

Distributional hypothesis

To model means to densely represent important features of some phenomenon. For example, in a phrase *'The judge sits in the court'*, the word *'judge'*:

1. consists of 3 phonemes [j e j];
2. is a singular noun in the nominative case;
3. functions as a subject in the syntactic tree of our sentence.

Distributional hypothesis

To model means to densely represent important features of some phenomenon. For example, in a phrase *'The judge sits in the court'*, the word *'judge'*:

1. consists of 3 phonemes [j e j];
2. is a singular noun in the nominative case;
3. functions as a subject in the syntactic tree of our sentence.

Such local representations describe many important features of the word *'judge'*. But not meaning.

Distributional hypothesis

But how to represent meaning?

Distributional hypothesis

But how to represent meaning?

- Semantics is difficult to represent formally.

Distributional hypothesis

But how to represent meaning?

- Semantics is difficult to represent formally.
- We need machine-readable word representations.
- Words which are similar in their meaning should possess mathematically similar representations.

Distributional hypothesis

But how to represent meaning?

- Semantics is difficult to represent formally.
- We need machine-readable word representations.
- Words which are similar in their meaning should possess mathematically similar representations.
- *'Judge'* is similar to *'court'* but not to *'kludge'*, even though their surface form suggests the opposite.

Distributional hypothesis

But how to represent meaning?

- Semantics is difficult to represent formally.
- We need machine-readable word representations.
- Words which are similar in their meaning should possess mathematically similar representations.
- *'Judge'* is similar to *'court'* but not to *'kludge'*, even though their surface form suggests the opposite.
- Why so?

Distributional hypothesis

Arbitrariness of a linguistic sign

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.

Distributional hypothesis

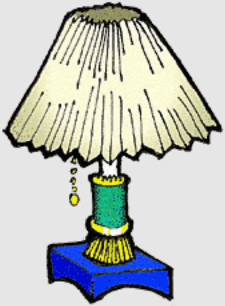
Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
‘Lantern’ concept can be expressed by any sequence of letters or sounds:

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
'Lantern' concept can be expressed by any sequence of letters or sounds:



- lantern

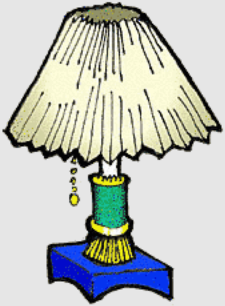
Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.

We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.

'Lantern' concept can be expressed by any sequence of letters or sounds:

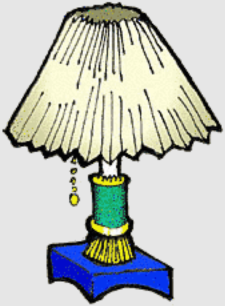


- lantern
- lykt

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
'Lantern' concept can be expressed by any sequence of letters or sounds:

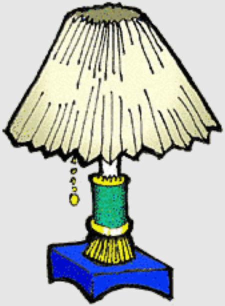


- lantern
- lykt
- лампа

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
'Lantern' concept can be expressed by any sequence of letters or sounds:

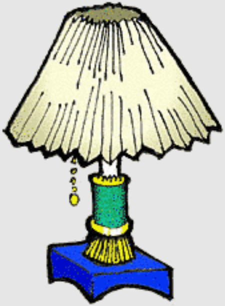


- lantern
- lykt
- лампа
- lucerna

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
'Lantern' concept can be expressed by any sequence of letters or sounds:

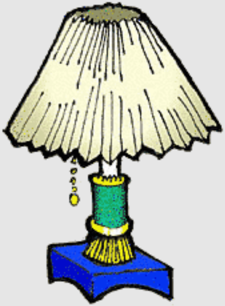


- lantern
- lykt
- лампа
- lucerna
- гэрэл

Distributional hypothesis

Arbitrariness of a linguistic sign

Unlike road signs, words do not possess a direct link between form and meaning.
We know this since Ferdinand de Saussure, and in fact structuralist theory influenced distributional approach much.
'Lantern' concept can be expressed by any sequence of letters or sounds:



- lantern
- lykt
- лампа
- lucerna
- гэрэл
- ...

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?

Distributional hypothesis

How do we know that ‘*lantern*’ and ‘*lamp*’ have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

The methods of computationally representing semantic relations in natural languages fall into two large groups:

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

The methods of computationally representing semantic relations in natural languages fall into two large groups:

1. Manually building ontologies (knowledge-based approach). Works top-down: from abstractions to real texts.

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

The methods of computationally representing semantic relations in natural languages fall into two large groups:

1. Manually building ontologies (knowledge-based approach). Works top-down: from abstractions to real texts. For example, Wordnet.

Distributional hypothesis

How do we know that '*lantern*' and '*lamp*' have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

The methods of computationally representing semantic relations in natural languages fall into two large groups:

1. Manually building ontologies (knowledge-based approach). Works top-down: from abstractions to real texts. For example, Wordnet.
2. Extracting semantics from usage patterns in text corpora (distributional approach). Works bottom-up: from real texts to abstractions.

Distributional hypothesis

How do we know that ‘*lantern*’ and ‘*lamp*’ have similar meaning? What is meaning, after all?
And how we can make computers understand this?

Possible data sources

The methods of computationally representing semantic relations in natural languages fall into two large groups:

1. Manually building ontologies (knowledge-based approach). Works top-down: from abstractions to real texts. For example, Wordnet.
2. Extracting semantics from usage patterns in text corpora (distributional approach). Works bottom-up: from real texts to abstractions.

The second approach is the topic of this course.

Distributional hypothesis

Meaning is actually a sum of contexts and distributional differences will always be enough to explain semantic differences:

- Words with similar typical contexts have similar meaning.

Distributional hypothesis

Meaning is actually a sum of contexts and distributional differences will always be enough to explain semantic differences:

- Words with similar typical contexts have similar meaning.
- The first to formulate: Ludwig Wittgenstein (1930s) and [Harris, 1954].

Distributional hypothesis

Meaning is actually a sum of contexts and distributional differences will always be enough to explain semantic differences:

- Words with similar typical contexts have similar meaning.
- The first to formulate: Ludwig Wittgenstein (1930s) and [Harris, 1954].
- *'You shall know a word by the company it keeps'* [Firth, 1957]

Distributional hypothesis

Meaning is actually a sum of contexts and distributional differences will always be enough to explain semantic differences:

- Words with similar typical contexts have similar meaning.
- The first to formulate: Ludwig Wittgenstein (1930s) and [Harris, 1954].
- *'You shall know a word by the company it keeps'* [Firth, 1957]
- Distributional semantics models (DSMs) are built upon lexical co-occurrences in a large training corpus.

Distributional hypothesis

It is important to distinguish between syntagmatic and paradigmatic relations between words.

Distributional hypothesis

It is important to distinguish between syntagmatic and paradigmatic relations between words.

- Words are in syntagmatic relation if they typically occur near each other (*'eat bread'*). It is also called first order co-occurrence.

Distributional hypothesis

It is important to distinguish between syntagmatic and paradigmatic relations between words.

- Words are in syntagmatic relation if they typically occur near each other (*'eat bread'*). It is also called first order co-occurrence.
- Syntagm is a kind of an ordered list.

Distributional hypothesis

It is important to distinguish between syntagmatic and paradigmatic relations between words.

- Words are in syntagmatic relation if they typically occur near each other (*'eat bread'*). It is also called first order co-occurrence.
- Syntagm is a kind of an ordered list.
- Words are in paradigmatic relation if the same neighbors typically occur near them (humans often *'eat'* both *'bread'* and *'butter'*). It is also called second order co-occurrence. The words in such a relation may well never actually co-occur with each other.

Distributional hypothesis

It is important to distinguish between syntagmatic and paradigmatic relations between words.

- Words are in syntagmatic relation if they typically occur near each other (*'eat bread'*). It is also called first order co-occurrence.
- Syntagm is a kind of an ordered list.
- Words are in paradigmatic relation if the same neighbors typically occur near them (humans often *'eat'* both *'bread'* and *'butter'*). It is also called second order co-occurrence. The words in such a relation may well never actually co-occur with each other.
- Paradigm is a kind of a set of substitutable entities.

We are interested mostly in paradigmatic relations (*bread* is semantically similar to *butter*, but not to '*fresh*').

Contents

Vector space models

The first and primary method of representing meaning in distributional semantics – semantic vectors.
First invented by Charles Osgood, American psychologist, in the 1950s [Osgood et al., 1964], then developed by many others.

Vector space models

In distributional semantics, meanings of particular words are represented as vectors or arrays of real values derived from frequency of their co-occurrences with other words (or other entities) in the training corpus.

Vector space models

In distributional semantics, meanings of particular words are represented as vectors or arrays of real values derived from frequency of their co-occurrences with other words (or other entities) in the training corpus.

- Words (or, more often, their lemmas) are vectors or points in multi-dimensional semantic space

Vector space models

In distributional semantics, meanings of particular words are represented as vectors or arrays of real values derived from frequency of their co-occurrences with other words (or other entities) in the training corpus.

- Words (or, more often, their lemmas) are vectors or points in multi-dimensional semantic space
- At the same time, words are also axes (dimensions) in this space (but we can use other types of contexts: documents, sentences, even characters).

Vector space models

In distributional semantics, meanings of particular words are represented as vectors or arrays of real values derived from frequency of their co-occurrences with other words (or other entities) in the training corpus.

- Words (or, more often, their lemmas) are vectors or points in multi-dimensional semantic space
- At the same time, words are also axes (dimensions) in this space (but we can use other types of contexts: documents, sentences, even characters).
- Each word A is represented with the vector $\vec{A}$. Vector dimensions or components are other words of the corpus' lexicon (B, C, $D \ldots N$). Values of components are frequencies of words co-occurrences.

In the simplest case, co-occurrences are just words occurring next to each other in the text. But contexts can be more complex!

Vector space models

A simple example of a symmetric word-word co-occurrence matrix:

	vector	**meaning**	**hamster**	**corpus**	**weasel**	**animal**
vector	0	10	0	8	0	0
meaning	10	0	1	15	0	0
hamster	0	1	0	0	20	14
corpus	8	15	0	0	0	2
weasel	0	0	20	0	0	21
animal	0	0	14	2	21	0

Vector space models

A simple example of a symmetric word-word co-occurrence matrix:

	vector	**meaning**	**hamster**	**corpus**	**weasel**	**animal**
vector	0	10	0	8	0	0
meaning	10	0	1	15	0	0
hamster	0	1	0	0	20	14
corpus	8	15	0	0	0	2
weasel	0	0	20	0	0	21
animal	0	0	14	2	21	0

We produced meaningful representations in a completely unsupervised way!

Vector space models

A simple example of a symmetric word-word co-occurrence matrix:

	vector	**meaning**	**hamster**	**corpus**	**weasel**	**animal**
vector	0	10	0	8	0	0
meaning	10	0	1	15	0	0
hamster	0	1	0	0	20	14
corpus	8	15	0	0	0	2
weasel	0	0	20	0	0	21
animal	0	0	14	2	21	0

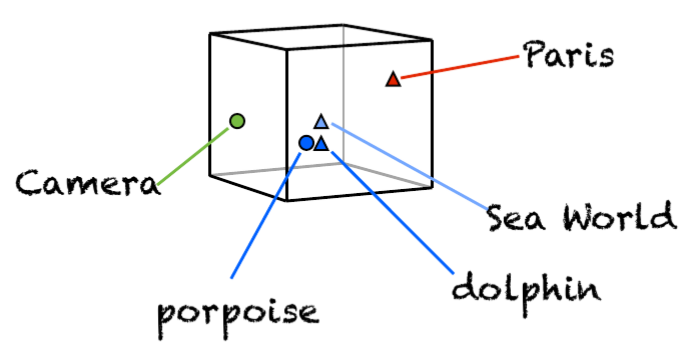
We produced meaningful representations in a completely unsupervised way!

Note how the '*animal*' vector is different from vocabulary index representations (sometimes called one-hot vectors):

'Animal': word number 1000 (or so).

Vector space models

Similar words are close to each other in the space defined by their typical co-occurrences



Vector space models

Of course one can somehow weight absolute frequency of co-occurrences to make sure that we pay less attention to ‘noise’ co-occurrences.

Vector space models

Of course one can somehow weight absolute frequency of co-occurrences to make sure that we pay less attention to 'noise' co-occurrences.
For example, Dice coefficient:

$$Dice(w, w') = \frac{2c(w, w')}{c(w) + c(w')} \quad (1)$$

where $c(w)$ – absolute frequency of w word,
$c(w')$ – absolute frequency of w' word
$c(w, w')$ – frequency of w and w' occurring together (collocation).

Vector space models

Of course one can somehow weight absolute frequency of co-occurrences to make sure that we pay less attention to 'noise' co-occurrences.
For example, Dice coefficient:

$$Dice(w, w') = \frac{2c(w, w')}{c(w) + c(w')} \tag{1}$$

where $c(w)$ – absolute frequency of w word,
$c(w')$ – absolute frequency of w' word
$c(w, w')$ – frequency of w and w' occurring together (collocation).
...or other weighting coefficients: tf-idf, log-likelihood, (positive) pointwise mutual information (PMI), etc.

Vector space models

Positive pointwise mutual information (PPMI) is the most common frequency weighting measure:

$$PPMI(w, w') = \max(\log_2 \frac{c(w, w')}{c(w) * c(w')}, 0) \tag{2}$$

where $c(w)$ – probability of w word,
$c(w')$ – probability of w' word
$c(w, w')$ – probability of w and w' occurring together.

Positive pointwise mutual information (PPMI) is the most common frequency weighting measure:

$$PPMI(w, w') = \max(\log_2 \frac{c(w, w')}{c(w) * c(w')}, 0) \tag{2}$$

where $c(w)$ – probability of w word,
$c(w')$ – probability of w' word
$c(w, w')$ – probability of w and w' occurring together.
Problem: rare words get high PPMI values. Can be alleviated by smoothing.

Vector space models

When building a co-occurrence matrix we can take into account not only immediate neighbors, but also words at some distance from our ‘focus word’:

Vector space models

When building a co-occurrence matrix we can take into account not only immediate neighbors, but also words at some distance from our ‘focus word’:

The brain is an ***organ*** *that serves as the center of the* ***nervous*** *system in all vertebrate and most invertebrate animals. The brain is located in the* ***head****, usually close to the sensory organs for senses such as* ***vision****. The brain is the most complex* ***organ*** *in a vertebrate’s body. In a human, the* ***cerebral cortex*** *contains approximately 15–33 billion* ***neurons****, each connected by* ***synapses*** *to several thousand other* ***neurons****.*

Vector space models

When building a co-occurrence matrix we can take into account not only immediate neighbors, but also words at some distance from our ‘focus word’:

*The brain is an **organ** that serves as the center of the **nervous** system in all vertebrate and most invertebrate animals. The brain is located in the **head**, usually close to the sensory organs for senses such as **vision**. The brain is the most complex **organ** in a vertebrate’s body. In a human, the **cerebral cortex** contains approximately 15–33 billion **neurons**, each connected by **synapses** to several thousand other **neurons**.*

Context width is defined at the beginning of building the matrix. Narrow windows favor ‘stricter’ semantic representations, while large windows produce more ‘associative’ models.

Vector space models

When building a co-occurrence matrix we can take into account not only immediate neighbors, but also words at some distance from our ‘focus word’:

*The brain is an **organ** that serves as the center of the **nervous** system in all vertebrate and most invertebrate animals. The brain is located in the **head**, usually close to the sensory organs for senses such as **vision**. The brain is the most complex **organ** in a vertebrate’s body. In a human, the **cerebral cortex** contains approximately 15–33 billion **neurons**, each connected by **synapses** to several thousand other **neurons**.*

Context width is defined at the beginning of building the matrix. Narrow windows favor ‘stricter’ semantic representations, while large windows produce more ‘associative’ models.

One can also change context words weights depending on the distance from the focus word, on their right or left position, or on the type of a syntactic arc between two words (*subjectof* / *objectof*)...

Vector space models

When building a co-occurrence matrix we can take into account not only immediate neighbors, but also words at some distance from our 'focus word':

The brain is an ***organ*** *that serves as the center of the* ***nervous*** *system in all vertebrate and most invertebrate animals. The brain is located in the* ***head****, usually close to the sensory organs for senses such as* ***vision****. The brain is the most complex* ***organ*** *in a vertebrate's body. In a human, the* ***cerebral cortex*** *contains approximately 15–33 billion* ***neurons****, each connected by* ***synapses*** *to several thousand other* ***neurons****.*

Context width is defined at the beginning of building the matrix. Narrow windows favor 'stricter' semantic representations, while large windows produce more 'associative' models.

One can also change context words weights depending on the distance from the focus word, on their right or left position, or on the type of a syntactic arc between two words (*subjectof* / *objectof*)...Possibilities are endless.

Contents

Calculating similarity: a first glance

Curse of dimensionality

Calculating similarity: a first glance

Curse of dimensionality

- With large corpora, we can end up with millions of dimensions (axes, words).

Calculating similarity: a first glance

Curse of dimensionality

- With large corpora, we can end up with millions of dimensions (axes, words).
- But the vectors are very sparse, most components are zero.

Calculating similarity: a first glance

Curse of dimensionality

- With large corpora, we can end up with millions of dimensions (axes, words).
- But the vectors are very sparse, most components are zero.
- One can reduce vector sizes to some reasonable values, and still retain meaningful relations between them.

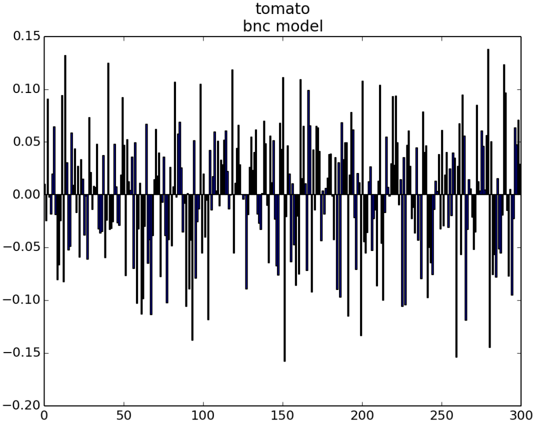
Calculating similarity: a first glance

Curse of dimensionality

- With large corpora, we can end up with millions of dimensions (axes, words).
- But the vectors are very sparse, most components are zero.
- One can reduce vector sizes to some reasonable values, and still retain meaningful relations between them.
- Such dense vectors are called 'word embeddings'.

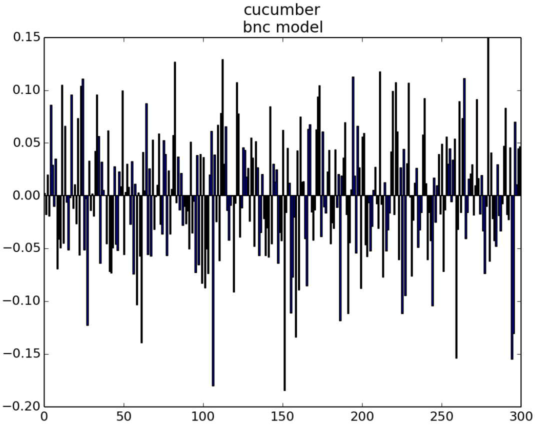
Calculating similarity: a first glance

300-D vector of ‘tomato’



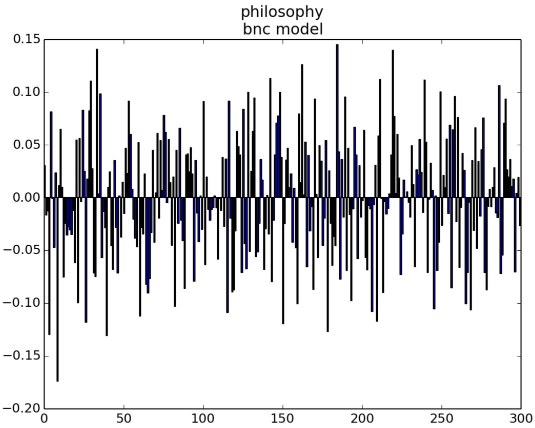
Calculating similarity: a first glance

300-D vector of 'cucumber'



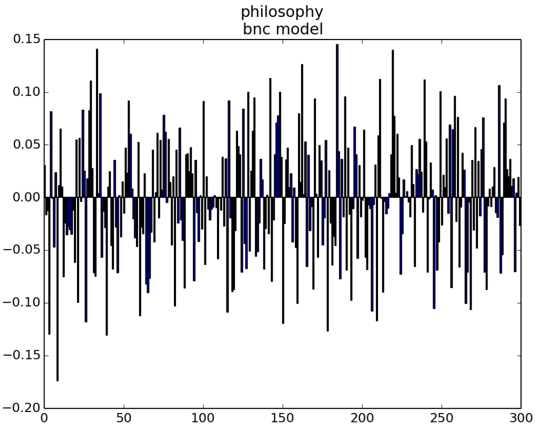
Calculating similarity: a first glance

300-D vector of 'philosophy'



Calculating similarity: a first glance

300-D vector of ‘philosophy’



Can we prove that tomatoes are more similar to cucumbers than to philosophy?

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$
$cos(tomato, cucumber) = 0.66$

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \quad (3)$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$
$cos(tomato, cucumber) = 0.66$

Cosine=1: vectors point at the same direction;

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$
$cos(tomato, cucumber) = 0.66$

Cosine=1: vectors point at the same direction;
Cosine=0: vectors are orthogonal;

Calculating similarity: a first glance

Semantic similarity between words is usually measured by the cosine of the angle between their corresponding vectors (takes values from -1 to 1).

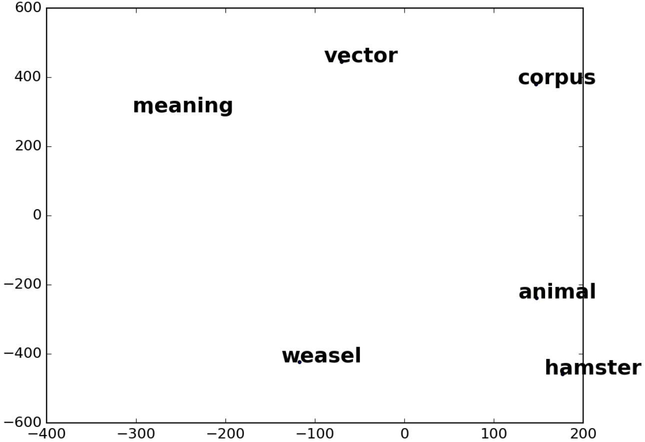
- Similarity lowers as the angle between word vectors grows.
- Similarity grows as the angle lessens.

$$cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \tag{3}$$

(dot product of unit-normalized vectors)
$cos(tomato, philosophy) = 0.09$
$cos(cucumber, philosophy) = 0.16$
$cos(tomato, cucumber) = 0.66$

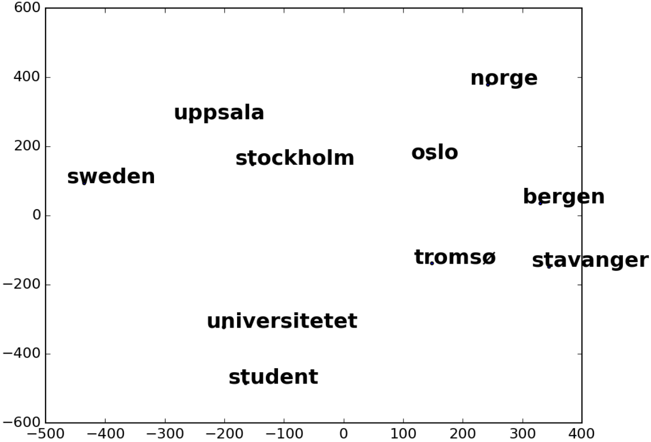
Cosine=1: vectors point at the same direction;
Cosine=0: vectors are orthogonal;
Cosine=-1: vectors point at the opposite directions.

Calculating similarity: a first glance



Embeddings reduced to 2 dimensions and visualized by t-SNE algorithm
[Van der Maaten and Hinton, 2008]

Calculating similarity: a first glance



Embeddings reduced to 2 dimensions and visualized by t-SNE algorithm
[Van der Maaten and Hinton, 2008]

Calculating similarity: a first glance

Important note

- There are several types of semantic similarity and relatedness.

Calculating similarity: a first glance

Important note

- There are several types of semantic similarity and relatedness.
- '*Gold*' and '*silver*' are semantically similar to each other but in quite a different way, than, say, '*cup*' and '*mug*'.

Calculating similarity: a first glance

Important note

- There are several types of semantic similarity and relatedness.
- '*Gold*' and '*silver*' are semantically similar to each other but in quite a different way, than, say, '*cup*' and '*mug*'.
- Can DSMs differentiate between synonyms, antonyms, meronyms, holonyms, etc?

Calculating similarity: a first glance

Important note

- There are several types of semantic similarity and relatedness.
- '*Gold*' and '*silver*' are semantically similar to each other but in quite a different way, than, say, '*cup*' and '*mug*'.
- Can DSMs differentiate between synonyms, antonyms, meronyms, holonyms, etc?
- More about this in the forthcoming lecture on practical aspects of using DSMs (including evaluation).

Contents

Summing up

Questions?

INF5820
Distributional Semantics: Extracting Meaning from Data
Lecture 1:
Linguistic Foundations of Distributional Semantics

Homework: play with
`http://ltr.uio.no/semvec`

Contents

In the next week

Main approaches to produce word embeddings

1. Point-wise mutual information (PMI) association matrices, factorized by SVD (so called *count-based models*) [Bullinaria and Levy, 2007];

In the next week

Main approaches to produce word embeddings

1. Point-wise mutual information (PMI) association matrices, factorized by SVD (so called *count-based models*) [Bullinaria and Levy, 2007];
2. *Predictive models* using artificial neural networks, introduced in [Bengio et al., 2003] and [Mikolov et al., 2013] (word2vec):

In the next week

Main approaches to produce word embeddings

1. Point-wise mutual information (PMI) association matrices, factorized by SVD (so called *count-based models*) [Bullinaria and Levy, 2007];
2. *Predictive models* using artificial neural networks, introduced in [Bengio et al., 2003] and [Mikolov et al., 2013] (word2vec):
   - Continuous Bag-of-Words (CBOW),
   - Continuous Skip-Gram (skipgram);

In the next week

Main approaches to produce word embeddings

1. Point-wise mutual information (PMI) association matrices, factorized by SVD (so called *count-based models*) [Bullinaria and Levy, 2007];
2. *Predictive models* using artificial neural networks, introduced in [Bengio et al., 2003] and [Mikolov et al., 2013] (word2vec):
   - Continuous Bag-of-Words (CBOW),
   - Continuous Skip-Gram (skipgram);
3. Global Vectors for Word Representation (GloVe) [Pennington et al., 2014];
4. ...etc

Two last approaches became super popular in the recent years and boosted almost all areas of natural language processing.

In the next week

Main approaches to produce word embeddings

1. Point-wise mutual information (PMI) association matrices, factorized by SVD (so called *count-based models*) [Bullinaria and Levy, 2007];
2. *Predictive models* using artificial neural networks, introduced in [Bengio et al., 2003] and [Mikolov et al., 2013] (word2vec):
   - Continuous Bag-of-Words (CBOW),
   - Continuous Skip-Gram (skipgram);
3. Global Vectors for Word Representation (GloVe) [Pennington et al., 2014];
4. ...etc

Two last approaches became super popular in the recent years and boosted almost all areas of natural language processing.
Their principal difference from previous methods is that they actively employ machine learning.

References I

Bengio, Y., Ducharme, R., and Vincent, P. (2003).
A neural probabilistic language model.
Journal of Machine Learning Research, 3:1137–1155.

Bullinaria, J. A. and Levy, J. P. (2007).
Extracting semantic representations from word co-occurrence statistics: A computational study.
Behavior research methods, 39(3):510–526.

Firth, J. (1957).
A synopsis of linguistic theory, 1930-1955.
Blackwell.

Harris, Z. S. (1954).
Distributional structure.
Word, 10(2-3):146–162.

References II

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
Advances in Neural Information Processing Systems 26.

Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1964).
The measurement of meaning.
University of Illinois Press.

Pennington, J., Socher, R., and Manning, C. D. (2014).
GloVe: Global vectors for word representation.
In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

References III

Turney, P., Pantel, P., et al. (2010).
From frequency to meaning: Vector space models of semantics.
Journal of artificial intelligence research, 37(1):141–188.

Van der Maaten, L. and Hinton, G. (2008).
Visualizing data using t-SNE.
Journal of Machine Learning Research, 9(2579-2605):85.