Distributional Semantics

Magnus Sahlgren

Pavia, 11 September 2012

Distributional Semantics

Course web page:

`http://www.gavagai.se/distributional_semantics.php`

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

This course:

- Theoretical foundations (*the distributional hypothesis*)
- Basic concepts (*co-occurrence matrix, distributional vectors*)
- Distributional semantic models (*LSA, HAL, Random Indexing*)
- Applications (*lexical semantics, cognitive modelling, data mining*)
- Research questions (*semantic relations, compositionality*)

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

- What are the underlying assumptions and theories?
- What are the main models, and how do they differ?
- How to build and use a model
- How to evaluate a model
- What can we use the models for?
- What are the main research questions at the moment?

Distributional Semantics

Theoretical basis
The Distributional Hypothesis

Representational framework
Vector Space

Distributional Semantics

Theoretical basis
The Distributional Hypothesis

Representational framework
Vector Space

Distributional Semantics

Theoretical basis
The Distributional Hypothesis

Representational framework
Vector Space

The Distributional Hypothesis

Extracting *semantics* from the *distributions* of terms

Distributional Semantics

Firth:
"you shall know a word by the company it keeps"

Wittgenstein:
"the meaning of a word is its use in language"

?:
"words with similar distributions have similar meanings"

=

The Distributional Hypothesis

Distributional Semantics

Firth:
"you shall know a word by the company it keeps"

Wittgenstein:
"the meaning of a word is its use in language"

?:
"words with similar distributions have similar meanings"

=
The Distributional Hypothesis

Distributional Semantics

Firth:
"you shall know a word by the company it keeps"

Wittgenstein:
"the meaning of a word is its use in language"

?:
"words with similar distributions have similar meanings"

=
The Distributional Hypothesis

Distributional Semantics

Firth:
"you shall know a word by the company it keeps"

Wittgenstein:
"the meaning of a word is its use in language"

?:
"words with similar distributions have similar meanings"

=
The Distributional Hypothesis

The Distributional Hypothesis

Usually motivated by Zellig Harris' distributional methodology:

- Linguistics can only deal with what is internal to language
- Linguistic explanans is distributional facts

The Distributional Hypothesis

Usually motivated by Zellig Harris' distributional methodology:

- Linguistics can only deal with what is internal to language
- Linguistic explanans is distributional facts

The Distributional Hypothesis

Usually motivated by Zellig Harris' distributional methodology:

- Linguistics can only deal with what is internal to language
- Linguistic explanans is distributional facts

The Distributional Hypothesis

Harris' distributional methodology is the *discovery procedure* by which we can quantify distributional similarities between linguistic entities

But why would such distributional similarities indicate *semantic* similarity? (And what does "meaning" mean anyway?)

The Distributional Hypothesis

Harris' distributional methodology is the *discovery procedure* by which we can quantify distributional similarities between linguistic entities

But why would such distributional similarities indicate *semantic* similarity? (And what does "meaning" mean anyway?)

The Distributional Hypothesis

If linguistics is to deal with meaning, it can only do so through distributional analysis

Differential, not referential

Structuralist legacy — Bloomfield, Saussure

The Distributional Hypothesis

If linguistics is to deal with meaning, it can only do so through distributional analysis

Differential, not referential

Structuralist legacy — Bloomfield, Saussure

The Distributional Hypothesis

If linguistics is to deal with meaning, it can only do so through distributional analysis

Differential, not referential

Structuralist legacy — Bloomfield, Saussure

The Distributional Hypothesis

Structuralism:

- Language as a system
- Signs (e.g. words) are identified by their functional differences within this system
- Two kinds: syntagmatic and paradigmatic

The Distributional Hypothesis

Structuralism:

- Language as a system
- Signs (e.g. words) are identified by their functional differences within this system
- Two kinds: syntagmatic and paradigmatic

The Distributional Hypothesis

Structuralism:

- Language as a system
- Signs (e.g. words) are identified by their functional differences within this system
- Two kinds: syntagmatic and paradigmatic

The Distributional Hypothesis

Structuralism:

- Language as a system
- Signs (e.g. words) are identified by their functional differences within this system
- Two kinds: syntagmatic and paradigmatic

The Distributional Hypothesis

Syntagmatic relations:

- Words that co-occur
- Combinatorial relations (phrases, sentences, etc.)

The Distributional Hypothesis

Syntagmatic relations:

- Words that co-occur
- Combinatorial relations (phrases, sentences, etc.)

The Distributional Hypothesis

Syntagmatic relations:

- Words that co-occur
- Combinatorial relations (phrases, sentences, etc.)

The Distributional Hypothesis

Paradigmatic relations:

- Words that *do not* co-occur
- Substitutional relations (co-occur with same *other* words)

The Distributional Hypothesis

Paradigmatic relations:

- Words that *do not* co-occur
- Substitutional relations (co-occur with same *other* words)

The Distributional Hypothesis

Paradigmatic relations:

- Words that *do not* co-occur
- Substitutional relations (co-occur with same *other* words)

The Distributional Hypothesis

Example:

I	drink	coffee
you	sip	tea
they	gulp	cocoa

The Distributional Hypothesis

Syntagmatic relations:

I	drink	coffee
you	sip	tea
they	gulp	cocoa

The Distributional Hypothesis

Paradigmatic relations:

I	drink	coffee
you	sip	tea
they	gulp	cocoa

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. "good" / "great")
- Antonyms (e.g. "good" / "bad")
- Hypernyms (e.g. "dog" / "animal")
- Hyponyms (e.g. "dog" / "poodle")
- Meronymy (e.g. "finger" / "hand")

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. “good” / “great”)
- Antonyms (e.g. “good” / “bad”)
- Hypernyms (e.g. “dog” / “animal”)
- Hyponyms (e.g. “dog” / “poodle”)
- Meronymy (e.g. “finger” / “hand”)

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. "good" / "great")
- Antonyms (e.g. "good" / "bad")
- Hypernyms (e.g. "dog" / "animal")
- Hyponyms (e.g. "dog" / "poodle")
- Meronymy (e.g. "finger" / "hand")

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. “good” / “great”)
- Antonyms (e.g. “good” / “bad”)
- Hypernyms (e.g. “dog” / “animal”)
- Hyponyms (e.g. “dog” / “poodle”)
- Meronymy (e.g. “finger” / “hand”)

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. "good" / "great")
- Antonyms (e.g. "good" / "bad")
- Hypernyms (e.g. "dog" / "animal")
- Hyponyms (e.g. "dog" / "poodle")
- Meronymy (e.g. "finger" / "hand")

The Distributional Hypothesis

Paradigmatic relations:

- Synonyms (e.g. “good” / “great”)
- Antonyms (e.g. “good” / “bad”)
- Hypernyms (e.g. “dog” / “animal”)
- Hyponyms (e.g. “dog” / “poodle”)
- Meronymy (e.g. “finger” / “hand”)

The Distributional Hypothesis

"drink" and "coffee" have a syntagmatic relation if they co-occur

"coffee" and "tea" have a paradigmatic relation if they co-occur with the same *other* words (e.g. "drink")

The Distributional Hypothesis

“drink” and “coffee” have a syntagmatic relation if they co-occur

“coffee” and “tea” have a paradigmatic relation if they co-occur with the same *other* words (e.g. “drink”)

The Distributional Hypothesis

To summarize:

Harris: If linguistics is to deal with meaning, it can only do so through distributional analysis

Structuralism: Meaning = relations between words

Saussure: Two types of relations: syntagmatic and paradigmatic

The Distributional Hypothesis

To summarize:

Harris: If linguistics is to deal with meaning, it can only do so through distributional analysis

Structuralism: Meaning = relations between words

Saussure: Two types of relations: syntagmatic and paradigmatic

The Distributional Hypothesis

To summarize:

Harris: If linguistics is to deal with meaning, it can only do so through distributional analysis

Structuralism: Meaning = relations between words

Saussure: Two types of relations: syntagmatic and paradigmatic

The Distributional Hypothesis

To summarize:

- **Harris:** If linguistics is to deal with meaning, it can only do so through distributional analysis
- **Structuralism:** Meaning = relations between words
- **Saussure:** Two types of relations: syntagmatic and paradigmatic

Distributional Semantics

Extracting *semantics* from the *distributions* of terms

=

Extracting *syntagmatic or paradigmatic relations* between words from *co-occurrences or second-order co-occurrences*

Distributional Semantics

Extracting *semantics* from the *distributions* of terms

=

Extracting *syntagmatic or paradigmatic relations* between words from *co-occurrences or second-order co-occurrences*

Distributional Semantics

The Distributional Hypothesis makes sense from a linguistic perspective

Does it also make sense from a cognitive perspective?

Distributional Semantics

The Distributional Hypothesis makes sense from a linguistic perspective

Does it also make sense from a cognitive perspective?

Distributional Semantics

We (i.e. humans) obviously *can* learn new words based on contextual cues

`He filled the wapimuk with the substance, passed it around and we all drunk some`

`We found a little, hairy wapimuk sleeping behind the tree`

Distributional Semantics

We (i.e. humans) obviously *can* learn new words based on contextual cues

`He filled the wapimuk with the substance, passed it around and we all drunk some`

`We found a little, hairy wapimuk sleeping behind the tree`

Distributional Semantics

We (i.e. humans) obviously *can* learn new words based on contextual cues

`He filled the wapimuk with the substance, passed it around and we all drunk some`

`We found a little, hairy wapimuk sleeping behind the tree`

Distributional Semantics

We encounter new words everyday...

but rarely ask for definitions

The absence of definitions in normal discourse is a sign of communicative prosperity

Distributional Semantics

We encounter new words everyday...

but rarely ask for definitions

The absence of definitions in normal discourse is a sign of communicative prosperity

Distributional Semantics

We encounter new words everyday...

but rarely ask for definitions

The absence of definitions in normal discourse is a sign of communicative prosperity

Distributional Semantics

Abstract terms (Vigliocco et al. 2009)
`justice`, `tax`, etc.

Concrete terms for which we have no direct experience of the referents
`aardvark`, `cyclotron`, etc.

Distributional Semantics

Abstract terms (Vigliocco et al. 2009)
`justice`, `tax`, etc.

Concrete terms for which we have no direct experience of the referents
`aardvark`, `cyclotron`, etc.

Distributional Semantics

Congenitally blind individuals show normal competence of color terms and visual perception verbs (Landau & Gleitman 1985)

Distributional Semantics

The *cognitive representation* of a word is some abstraction or generalization derived from the contexts in which the word has been encountered (Miller & Charles 1991)

Distributional Semantics

A contextual representation may include extra-linguistic contexts

De facto, context is equated with linguistic context

- Practical reason — it is easy to collect linguistic contexts (from corpora) and to process them
- Theoretical reason — it is possible to investigate the role of linguistic distributions in shaping word meaning

Distributional Semantics

A contextual representation may include extra-linguistic contexts

De facto, context is equated with linguistic context

- Practical reason — it is easy to collect linguistic contexts (from corpora) and to process them
- Theoretical reason — it is possible to investigate the role of linguistic distributions in shaping word meaning

Distributional Semantics

A contextual representation may include extra-linguistic contexts

De facto, context is equated with linguistic context

- Practical reason — it is easy to collect linguistic contexts (from corpora) and to process them
- Theoretical reason — it is possible to investigate the role of linguistic distributions in shaping word meaning

Distributional Semantics

A contextual representation may include extra-linguistic contexts

De facto, context is equated with linguistic context

- Practical reason — it is easy to collect linguistic contexts (from corpora) and to process them
- Theoretical reason — it is possible to investigate the role of linguistic distributions in shaping word meaning

Distributional Semantics

Weak Distributional Hypothesis

A quantitative method for semantic analysis and lexical resource induction

Meaning (whatever this might be) is reflected in distribution

Distributional Semantics

Weak Distributional Hypothesis

A quantitative method for semantic analysis and lexical resource induction

Meaning (whatever this might be) is reflected in distribution

Distributional Semantics

Weak Distributional Hypothesis

A quantitative method for semantic analysis and lexical resource induction

Meaning (whatever this might be) is reflected in distribution

Distributional Semantics

Strong Distributional Hypothesis

A cognitive hypothesis about the form and origin of semantic representations

Word distributions in context have a specific causal role in the formation of the semantic representation for that word

The distributional properties of words in linguistic contexts explains human semantic behavior (e.g. judgment of semantic similarity)

Distributional Semantics

Strong Distributional Hypothesis

A cognitive hypothesis about the form and origin of semantic representations

Word distributions in context have a specific causal role in the formation of the semantic representation for that word

The distributional properties of words in linguistic contexts explains human semantic behavior (e.g. judgment of semantic similarity)

Distributional Semantics

Strong Distributional Hypothesis

A cognitive hypothesis about the form and origin of semantic representations

Word distributions in context have a specific causal role in the formation of the semantic representation for that word

The distributional properties of words in linguistic contexts explains human semantic behavior (e.g. judgment of semantic similarity)

Distributional Semantics

Strong Distributional Hypothesis

A cognitive hypothesis about the form and origin of semantic representations

Word distributions in context have a specific causal role in the formation of the semantic representation for that word

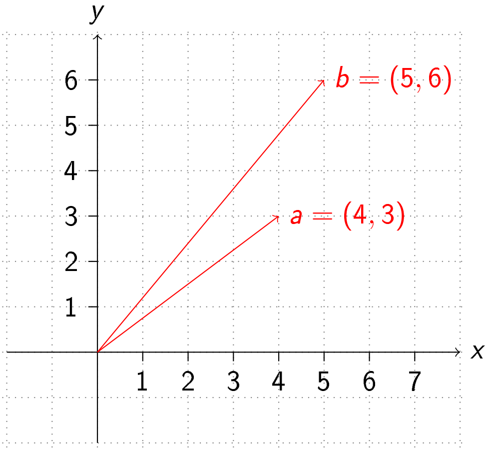
The distributional properties of words in linguistic contexts explains human semantic behavior (e.g. judgment of semantic similarity)

Distributional Semantics

Theoretical basis
The Distributional Hypothesis

Representational framework
Vector Space

Vector space



Vector space

$$A_{2,2} = \begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} (4 & 3) \\ (5 & 6) \end{bmatrix} \end{array}$$

Vector space

$$A_{2,2} = \begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{bmatrix} (4 & 3) \\ (5 & 6) \end{bmatrix} \end{array}$$

Vector space

$$A_{2,2} = \begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} a \\ b \end{array} & \left[\begin{array}{c} (4 \quad 3) \\ (5 \quad 6) \end{array} \right] \end{array}$$

Vector space

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{bmatrix}$$

Vector space

High-dimensional vector spaces (i.e. dimensionalities on the order of thousands)

Mathematically straightforward

Warning: high-dimensional spaces can be counterintuitive!

Vector space

High-dimensional vector spaces (i.e. dimensionalities on the order of thousands)

Mathematically straightforward

Warning: high-dimensional spaces can be counterintuitive!

Vector space

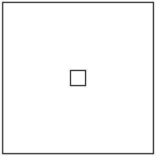
High-dimensional vector spaces (i.e. dimensionalities on the order of thousands)

Mathematically straightforward

Warning: high-dimensional spaces can be counterintuitive!

Vector space

Imagine two nested cubes:

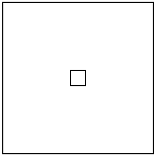


How big must the inner cube be in order to cover 1% of the area of the outer cube?

Answer: 10% of the side length ($0.1 \times 0.1 = 0.01$)

Vector space

Imagine two nested cubes:

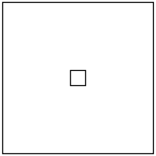


How big must the inner cube be in order to cover 1% of the area of the outer cube?

Answer: 10% of the side length $(0.1 \times 0.1 = 0.01)$

Vector space

Imagine two nested cubes:



How big must the inner cube be in order to cover 1% of the area of the outer cube?

Answer: 10% of the side length ($0.1 \times 0.1 = 0.01$)

Vector space

For 3-dimensional cubes, the inner cube must have $\approx 21\%$ of the side length of the outer cube ($0.21 \times 0.21 \times 0.21 \approx 0.01$)

For n-dimensional cubes, the inner cube must have $0.01^{\frac{1}{n}}$ of the side length

Vector space

For 3-dimensional cubes, the inner cube must have $\approx 21\%$ of the side length of the outer cube ($0.21 \times 0.21 \times 0.21 \approx 0.01$)

For n-dimensional cubes, the inner cube must have $0.01^{\frac{1}{n}}$ of the side length

Vector space

For $n = 1\ 000$ that is 99.5%

That is, if the outer 1 000-dimensional cube has sides 2 units long, and the inner 1 000-dimensional cube has sides 1.99 units long, the outer cube will still contain *100 times* more volume

Vector space

For $n = 1\ 000$ that is 99.5%

That is, if the outer 1 000-dimensional cube has sides 2 units long, and the inner 1 000-dimensional cube has sides 1.99 units long, the outer cube will still contain *100 times* more volume

Distributional Semantics

Distributional semantics + vector space

=

Word Space Models

Distributional Semantics

Distributional semantics + vector space

=

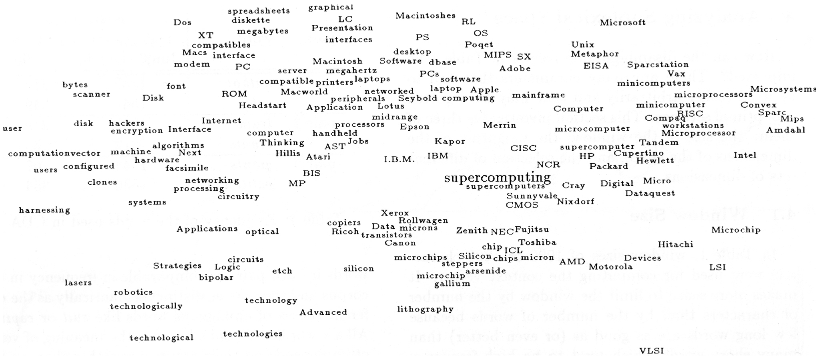
Word Space Models

Word Space

"Vector similarity is the only information present in Word Space: semantically related words are close, unrelated words are distant."

(Hinrich Schütze: Word space, 1993)

Word Space



(Schütze: Dimensions of Meaning, 1992)

Word Space (A Brief History)

Manually defined features

Psychology: Osgood (1950s)

Human ratings over contrastive adjective-pairs

	small – large	bald – furry	docile – dangerous
mouse	2	6	1
rat	2	6	4

Word Space (A Brief History)

Manually defined features

Psychology: Osgood (1950s)

Human ratings over contrastive adjective-pairs

	small – large	bald – furry	docile – dangerous
mouse	2	6	1
rat	2	6	4

Word Space (A Brief History)

Manually defined features

Psychology: Osgood (1950s)

Human ratings over contrastive adjective-pairs

	small – large	bald – furry	docile – dangerous
mouse	2	6	1
rat	2	6	4

Word Space (A Brief History)

Manually defined features

Connectionism, AI: Waltz & Pollack (1980s), Gallant (1990s), Gärdenfors (2000s)

	human	machine	politics	...
astronomer	+2	-1	-1	...

Word Space (A Brief History)

Manually defined features

Connectionism, AI: Waltz & Pollack (1980s), Gallant (1990s), Gärdenfors (2000s)

	human	machine	politics	...
astronomer	+2	-1	-1	...

Word Space (A Brief History)

Manually defined features

Pros: Expert knowledge

Cons: Bias; how many features are enough?

Word Space (A Brief History)

Manually defined features

Pros: Expert knowledge

Cons: Bias; how many features are enough?

Word Space (A Brief History)

Manually defined features

Pros: Expert knowledge

Cons: Bias; how many features are enough?

Word Space

Automatic methods: count co-occurrences (c.f. the distributional hypothesis)

- Co-occurrence matrix
- Distributional vector

Word Space

Automatic methods: count co-occurrences (c.f. the distributional hypothesis)

- Co-occurrence matrix
- Distributional vector

Word Space

Automatic methods: count co-occurrences (c.f. the distributional hypothesis)

- Co-occurrence matrix
- Distributional vector

The Co-occurrence Matrix

$$
A_{m,n} = \begin{array}{cc} & \begin{array}{ccccc} c_1 & c_2 & c_3 & \cdots & c_n \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} & \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{bmatrix} \end{array}
$$

- w are the m (target) word types
- c are the n contexts in the data
- and the cells are word-context co-occurrence frequencies

The Co-occurrence Matrix

$$A_{m,n} = \begin{array}{c c} & \begin{array}{ccccc} c_1 & c_2 & c_3 & \dots & c_n \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} & \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \dots & a_{m,n} \end{bmatrix} \end{array}$$

- w are the m (target) word types
- c are the n contexts in the data
- and the cells are word-context co-occurrence frequencies

The Co-occurrence Matrix

$$A_{m,n} = \begin{array}{c c} & \begin{array}{ccccc} c_1 & c_2 & c_3 & \cdots & c_n \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} & \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{bmatrix} \end{array}$$

- w are the m (target) word types
- c are the n contexts in the data
- and the cells are word-context co-occurrence frequencies

The Co-occurrence Matrix

$$A_{m,n} = \begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} c_1 & c_2 & c_3 & \cdots & c_n \end{array} \\ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & a_{m,3} & \cdots & a_{m,n} \end{bmatrix} \end{array}$$

- w are the m (target) word types
- c are the n contexts in the data
- and the cells are word-context co-occurrence frequencies

The Co-occurrence Matrix

$$A_{m,n} = \begin{array}{c c} & \begin{array}{ccccc} c_1 & c_2 & c_3 & \dots & c_n \end{array} \\ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} & \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 1 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

- w are the m (target) word types
- c are the n contexts in the data
- and the cells are word-context co-occurrence frequencies

Distributional Vectors

$$A_{m,n} = \begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} c_1 & c_2 & c_3 & \dots & c_n \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 1 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

Rows are n-dimensional distributional vectors

Words that occur in similar contexts get similar distributional vectors

So how should we define "context"?

Distributional Vectors

$$A_{m,n} = \begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} c_1 & c_2 & c_3 & \dots & c_n \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 1 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

Rows are n-dimensional distributional vectors

Words that occur in similar contexts get similar distributional vectors

So how should we define "context"?

Distributional Vectors

$$A_{m,n} = \begin{array}{c} \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} c_1 & c_2 & c_3 & \dots & c_n \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 1 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

Rows are n-dimensional distributional vectors

Words that occur in similar contexts get similar distributional vectors

So how should we define "context"?

Words-by-region Matrices

$$A_{m,n} = \begin{array}{c} \\ \textit{drink} \\ w_2 \\ \textit{coffee} \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} r_1 & r_2 & r_3 & \dots & r_n \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ 0 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

A region is a text unit (document, paragraph, sentence, etc.)

r_2 = *"I drink coffee"*

Syntagmatic similarity: words that co-occur in the same region

Words-by-region Matrices

$$A_{m,n} = \begin{array}{l} \\ \textit{drink} \\ w_2 \\ \textit{coffee} \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} r_1 & r_2 & r_3 & \dots & r_n \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ 0 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

A region is a text unit (document, paragraph, sentence, etc.)
r_2 = *"I drink coffee"*

Syntagmatic similarity: words that co-occur in the same region

Words-by-region Matrices

$$A_{m,n} = \begin{array}{l} \\ drink \\ w_2 \\ coffee \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} r_1 & r_2 & r_3 & \ldots & r_n \end{array} \\ \left[\begin{array}{ccccc} 0 & 1 & 0 & \ldots & 1 \\ 1 & 0 & 1 & \ldots & 0 \\ 0 & 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \ldots & 0 \end{array}\right] \end{array}$$

A region is a text unit (document, paragraph, sentence, etc.)
r_2 = *"I drink coffee"*

Syntagmatic similarity: words that co-occur in the same region

Words-by-words Matrices

$$A_{m,n} = \begin{array}{c} \\ \textit{coffee} \\ w_2 \\ \textit{tea} \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & \textit{drink} & w_3 & \ldots & w_m \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \ldots & 1 \\ 1 & 0 & 2 & \ldots & 0 \\ 0 & 2 & 0 & \ldots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \ldots & 0 \end{bmatrix} \end{array}$$

Paradigmatic similarity: words that co-occur with the same *other* words

Syntagmatic similarity by looking at the individual dimensions

Words-by-words Matrices

$$A_{m,n} = \begin{array}{l} \\ coffee \\ w_2 \\ tea \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & drink & w_3 & \ldots & w_m \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \ldots & 1 \\ 1 & 0 & 2 & \ldots & 0 \\ 0 & 2 & 0 & \ldots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \ldots & 0 \end{bmatrix} \end{array}$$

Paradigmatic similarity: words that co-occur with the same *other* words

Syntagmatic similarity by looking at the individual dimensions

Words-by-words Matrices

$$A_{m,n} = \begin{array}{l} \\ coffee \\ w_2 \\ tea \\ \vdots \\ w_m \end{array} \begin{array}{c} \begin{array}{ccccc} w_1 & drink & w_3 & \dots & w_m \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 2 & \dots & 0 \\ 0 & 2 & 0 & \dots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \end{array}$$

Paradigmatic similarity: words that co-occur with the same *other* words

Syntagmatic similarity by looking at the individual dimensions

Context window

The region within which co-occurrences are collected

Parameters:

- Size
- Direction
- Weighting

Context window

The region within which co-occurrences are collected

Parameters:

- Size
- Direction
- Weighting

Context window

The region within which co-occurrences are collected

Parameters:

- Size
- Direction
- Weighting

Context window

The region within which co-occurrences are collected

Parameters:

- Size
- Direction
- Weighting

Context window

The region within which co-occurrences are collected

Parameters:

- Size
- Direction
- Weighting

Context window

[I drink very strong] coffee [at the cafe down the street]

1 1 1 1 1 1 1 1 1 1

An entire sentence as a flat context window

Context window

```
[I drink very strong] coffee [at the cafe down the street ]
 1    1     1      1              1   1    1    1    1      1
```

An entire sentence as a flat context window

Context window

```
I  drink [very strong] coffee [at the ] cafe down the street
           1      1                1    1
```

A 2+2-sized flat context window

Context window

```
[I drink very strong] coffee [at the cafe down ] the street
 1    2     3      4             4   3    2     1
```

A 4+4-sized *distance weighted* context window

Context window

`I drink very strong coffee [at the cafe down the street]`

I	drink	very	strong	coffee	[at	the	cafe	down	the	street]
					1	0.9	0.8	0.7	0.6	0.5

A 0+6-sized distance weighted context window

Context window

```
I [drink very strong] coffee [at the cafe] down the street
     1      0      3                0    0     1
```

A 3+3-sized distance weighted context window that ignores stop words

Distributional vectors

Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

`[12,0,234,92,1,0,87,525,0,0,1,2,0,8129,1,0,51,0,235...]`

And then what?

Distributional vectors

Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

`[12,0,234,92,1,0,87,525,0,0,1,2,0,8129,1,0,51,0,235...]`

And then what?

Distributional vectors

Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

`[12,0,234,92,1,0,87,525,0,0,1,2,0,8129,1,0,51,0,235...]`

And then what?

Distributional vectors

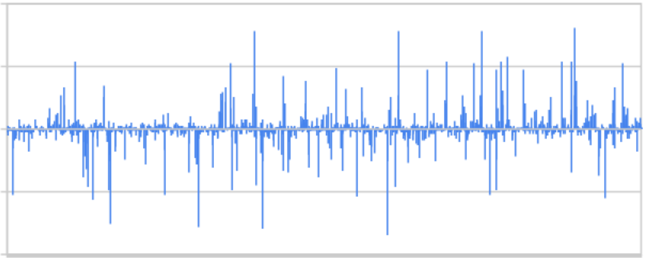
Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

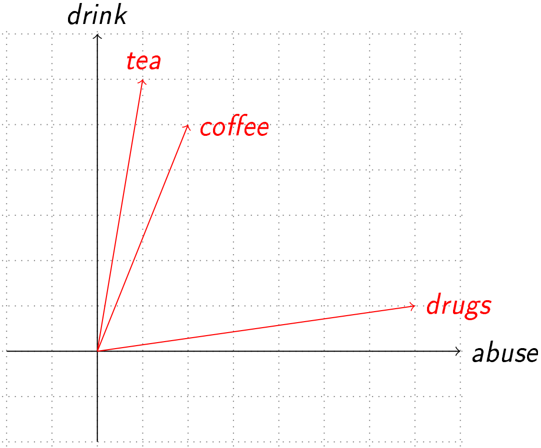
`[12,0,234,92,1,0,87,525,0,0,1,2,0,8129,1,0,51,0,235...]`

And then what?

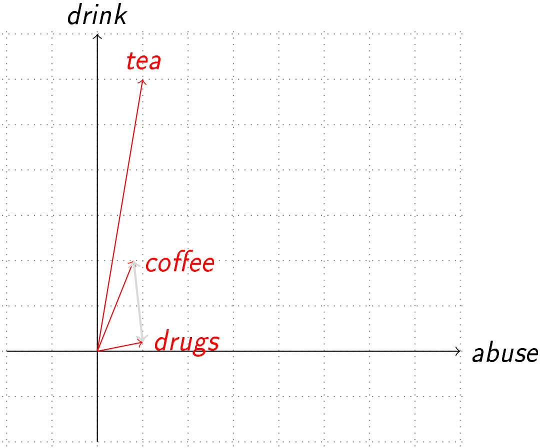
The Meaning of Life



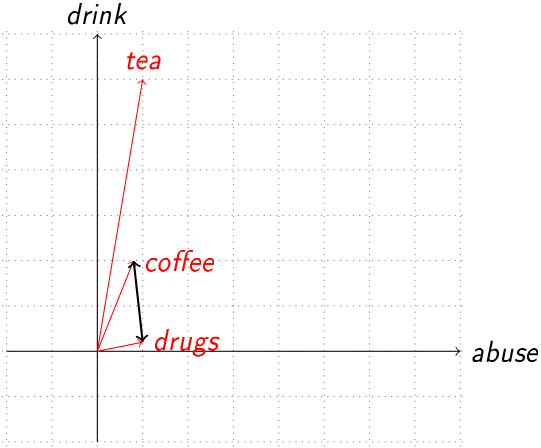
Vector similarity



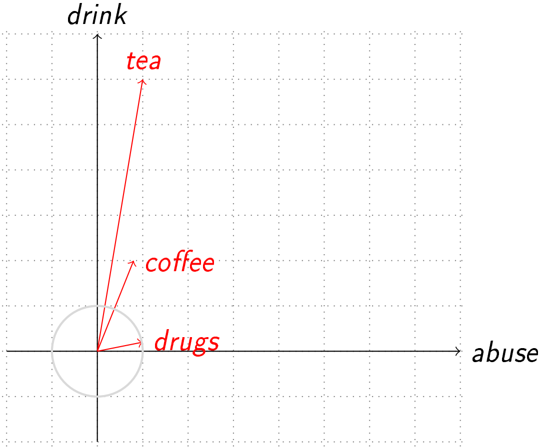
Vector similarity



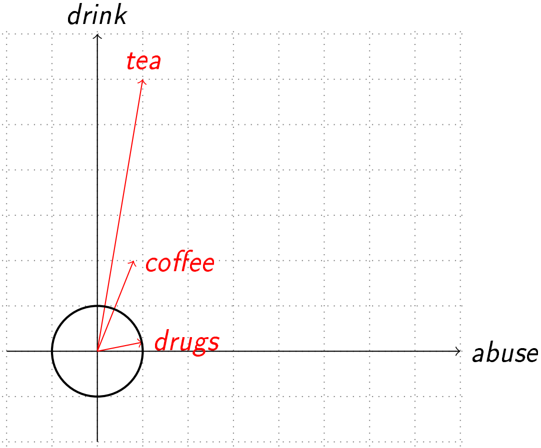
Vector similarity



Vector similarity



Vector similarity



Vector similarity

Minkowski distance: $\left(\sum_{i=1}^{n} |x_i - y_i|^N\right)^{\frac{1}{N}}$

- City-Block (or Manhattan) distance: $N = 1$
- Euclidean distance: $N = 2$
- Chebyshev distance: $N \to \infty$

Scalar product: $x \cdot y = x_1 y_1 + x_2 y_2 + ... + x_n y_n$

Cosine: $\frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Vector similarity

Minkowski distance: $\left(\sum_{i=1}^{n} |x_i - y_i|^N\right)^{\frac{1}{N}}$

- City-Block (or Manhattan) distance: $N = 1$
- Euclidean distance: $N = 2$
- Chebyshev distance: $N \to \infty$

Scalar product: $x \cdot y = x_1 y_1 + x_2 y_2 + ... + x_n y_n$

Cosine: $\frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Vector similarity

Minkowski distance: $\left(\sum_{i=1}^{n} |x_i - y_i|^N\right)^{\frac{1}{N}}$

- City-Block (or Manhattan) distance: $N = 1$
- Euclidean distance: $N = 2$
- Chebyshev distance: $N \to \infty$

Scalar product: $x \cdot y = x_1 y_1 + x_2 y_2 + ... + x_n y_n$

Cosine: $\frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Vector similarity

Minkowski distance: $\left(\sum_{i=1}^{n} |x_i - y_i|^N\right)^{\frac{1}{N}}$

- City-Block (or Manhattan) distance: $N = 1$
- Euclidean distance: $N = 2$
- Chebyshev distance: $N \rightarrow \infty$

Scalar product: $x \cdot y = x_1 y_1 + x_2 y_2 + ... + x_n y_n$

Cosine: $\frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

Distributional vectors

Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

And then what?

Compute similarity between such vectors

Distributional vectors

Count how many times each target word occurs in a certain context

Collect (a function of) these frequency counts in vectors

And then what?

Compute similarity between such vectors

Nearest Neighbors

Nearest neighbor search: extracting the k nearest neighbors to a target word

1. compute the cosine similarity between the context vector of the target word and the context vectors of all other words in the word space
2. sort the resulting similarities
3. return only the top-k words

Nearest Neighbors

Nearest neighbor search: extracting the k nearest neighbors to a target word

1. compute the cosine similarity between the context vector of the target word and the context vectors of all other words in the word space
2. sort the resulting similarities
3. return only the top-k words

Nearest Neighbors

Nearest neighbor search: extracting the k nearest neighbors to a target word

1. compute the cosine similarity between the context vector of the target word and the context vectors of all other words in the word space
2. sort the resulting similarities
3. return only the top-k words

Nearest Neighbors

Nearest neighbor search: extracting the k nearest neighbors to a target word

1. compute the cosine similarity between the context vector of the target word and the context vectors of all other words in the word space
2. sort the resulting similarities
3. return only the top-k words

Nearest Neighbors

dog	
dog	1.000000
cat	0.774165
horse	0.668386
fox	0.648134
pet	0.626650
rabbit	0.615840
pig	0.570504
animal	0.566003
mongrel	0.560158
sheep	0.551973
pigeon	0.547442
deer	0.534663
rat	0.531442
bird	0.527370

red	
red	1.000000
yellow	0.824409
white	0.789056
brown	0.723576
grey	0.720103
blue	0.700047
pink	0.672573
black	0.671302
shiny	0.661379
purple	0.633858
striped	0.619801
dark	0.610804
gleam	0.603501
palea	0.595221

Building a word space: step-by-step

The "linguistic" steps

`I drank very strong "Arabica" coffees, at the cafe.`

- Pre-process a corpus (optional, but recommended)

Building a word space: step-by-step

The "linguistic" steps

`I drank very strong ‘‘Arabica’’ coffees, at the cafe.`

- Pre-process a corpus (optional, but recommended)

Building a word space: step-by-step

The "linguistic" steps

`I drank very strong Arabica coffees at the cafe`

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization,

Building a word space: step-by-step

The "linguistic" steps

`i drank very strong arabica coffees at the cafe`

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing,

Building a word space: step-by-step

The "linguistic" steps

```
i drink very strong arabica coffee at the cafe
```

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)

Building a word space: step-by-step

The "linguistic" steps

`i drink very strong arabica coffee at the cafe`

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up

Building a word space: step-by-step

The "linguistic" steps

`i/PN drink/VB very/AV strong/AJ arabica/NN coffee/NN at/PR the/DT cafe/NN`

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up
  (PoS tagging,

Building a word space: step-by-step

The "linguistic" steps

```
(S (NP i/PN) (VP drink/VB (NP (AP very/AV strong/AJ) arabica/NN
coffee/NN) (PP at/PR (NP the/DT cafe/NN))))
```

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up
  (PoS tagging, syntactic parsing,

Building a word space: step-by-step

The "linguistic" steps

```
(S (NP i/PN) (VP drink/VB (NP (AP very/AV strong/AJ) arabica/NN
coffee/NN) (PP at/PR (NP the/DT cafe/NN))))
```

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up
  (PoS tagging, syntactic parsing, named entity recognition...)

Building a word space: step-by-step

The "linguistic" steps

```
(S (NP i/PN) (VP drink/VB (NP (AP very/AV strong/AJ) arabica/NN
coffee/NN) (PP at/PR (NP the/DT cafe/NN))))
```

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up
  (PoS tagging, syntactic parsing, named entity recognition...)
- Select (the target words and) the linguistic contexts
  (e.g. text region, words...)

Building a word space: step-by-step

The "linguistic" steps

```
(S (NP i/PN) (VP drink/VB (NP (AP very/AV strong/AJ) arabica/NN
coffee/NN) (PP at/PR (NP the/DT cafe/NN))))
```

- Pre-process a corpus (optional, but recommended)
  (e.g. tokenization, downcasing, stemming/lemmatization...)
- Linguistic mark-up
  (PoS tagging, syntactic parsing, named entity recognition...)
- Select (the target words and) the linguistic contexts
  (e.g. text region, words...)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space: step-by-step

The "mathematical" steps

- Count the target-context co-occurrences
- Weight the contexts (optional, but recommended)
  (e.g. raw co-occurrence frequency, entropy, association measures...)
- Reduce the matrix dimensions (optional)
  (dimension reduction method: context selection, variance, SVD, RI...)
  (dimensionality)
- Compute vector similarities (do nearest neighbor search) between distributional vectors
  (e.g. cosine, euclidean distance, etc.)

Building a word space

Tools:

- **GSDM** Guile Sparse Distributed Memory (`Guile`)
- **S-Space** https://github.com/fozziethebeat/S-Space (`Java`)
- **SemanticVectors** http://code.google.com/p/semanticvectors/ (`Java`)
- **GenSim** http://radimrehurek.com/gensim/ (`Python`)
- **Word2word** http://www.indiana.edu/ semantic/word2word/ (`Java`)

Building a word space

Tools:

- **GSDM** Guile Sparse Distributed Memory (`Guile`)
- **S-Space** https://github.com/fozziethebeat/S-Space (`Java`)
- **SemanticVectors** http://code.google.com/p/semanticvectors/ (`Java`)
- **GenSim** http://radimrehurek.com/gensim/ (`Python`)
- **Word2word** http://www.indiana.edu/ semantic/word2word/ (`Java`)

Building a word space

Tools:

GSDM	Guile Sparse Distributed Memory (Guile)
S-Space	https://github.com/fozziethebeat/S-Space (Java)
SemanticVectors	http://code.google.com/p/semanticvectors/ (Java)
GenSim	http://radimrehurek.com/gensim/ (Python)
Word2word	http://www.indiana.edu/ semantic/word2word/ (Java)

Building a word space

Tools:

GSDM	Guile Sparse Distributed Memory (`Guile`)
S-Space	https://github.com/fozziethebeat/S-Space (`Java`)
SemanticVectors	http://code.google.com/p/semanticvectors/ (`Java`)
GenSim	http://radimrehurek.com/gensim/ (`Python`)
Word2word	http://www.indiana.edu/ semantic/word2word/ (`Java`)

Building a word space

Tools:

- **GSDM** Guile Sparse Distributed Memory (`Guile`)
- **S-Space** https://github.com/fozziethebeat/S-Space (`Java`)
- **SemanticVectors** http://code.google.com/p/semanticvectors/ (`Java`)
- **GenSim** http://radimrehurek.com/gensim/ (`Python`)
- **Word2word** http://www.indiana.edu/ semantic/word2word/ (`Java`)

Building a word space

Tools:

GSDM	Guile Sparse Distributed Memory (`Guile`)
S-Space	https://github.com/fozziethebeat/S-Space (`Java`)
SemanticVectors	http://code.google.com/p/semanticvectors/ (`Java`)
GenSim	http://radimrehurek.com/gensim/ (`Python`)
Word2word	http://www.indiana.edu/ semantic/word2word/ (`Java`)

Lab

Use GSDM to:

- Build a words-by-documents model
- Build a words-by-words model
- Extract similar words

Lab

Use GSDM to:

- Build a words-by-documents model
- Build a words-by-words model
- Extract similar words

Lab

Use GSDM to:

- Build a words-by-documents model
- Build a words-by-words model
- Extract similar words

Lab

Use GSDM to:

- Build a words-by-documents model
- Build a words-by-words model
- Extract similar words