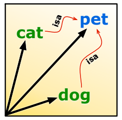
Hands-on Distributional Semantics

Part 1: Introduction

Stephanie Evert[1] & Gabriella Lapesa[2]
with Alessandro Lenci[3] and Marco Baroni[4]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
[2]University of Stuttgart, Germany
[3]University of Pisa, Italy [4]University of Trento, Italy

http://wordspace.collocations.de/doku.php/course:esslli2021:start



What is distributional semantics?

- A corpus-based approach to the representation of meaning based on a very simple intuition: **distributional hypothesis**
  - ☞ similar context $\Longleftrightarrow$ similar meaning
- An empirical method that produces usage-based lexical entries for words, which to the computer look like this:
  - $(10, 0, 0, 0, 0, 100, 40)$
  - $(-1.3, 1.4, 0.4, -0.2, 1.3, 2.7, -0.001)$
- Closely related to neuronal word embeddings
- Maths behind it can be complicated . . .
  - . . . but you can apply DS to many research questions with existing software packages if you understand the basic concepts clearly
  - ☞ Beware of the black box problem!

Goals of this course

1. Introduce the basic concepts of **distributional semantics** (DS) and – at the same time – teach you to take your own steps into DS with the `wordspace` package for R
2. Show you what can be done with DS in two domains of interdisciplinary application, including hands-on exercises
   - Linguistic Theory
     - ★ Motivation: test theories, enlarge scope of investigation
     - ★ Challenge: operationalization (theoretical concepts → empirical properties)
   - Cognitive modeling
     - ★ Motivation: corpus data are behavioural data after all
     - ★ Challenge: continuous variables, large vocabularies
3. Equip you with the “coordinates” to navigate the current DS literature beyond the scope of this course

Today’s plan

Introduction
- The distributional hypothesis
- Distributional semantic models
- DSM and semantic similarity
- Course Outline

Getting practical
- Software and further information
- R as a (toy) laboratory

Outline

Introduction
- The distributional hypothesis
- Distributional semantic models
- DSM and semantic similarity
- Course Outline

Getting practical
- Software and further information
- R as a (toy) laboratory

Meaning & distribution

- "Die Bedeutung eines Wortes liegt in seinem Gebrauch."
  — Ludwig Wittgenstein
  - ☞ meaning = use = distribution in language
- "You shall know a word by the company it keeps!"
  — J. R. Firth (1957)
  - ☞ distribution = collocations = habitual word combinations
- Distributional hypothesis: difference of meaning correlates with difference of distribution (Zellig Harris 1954)
  - ☞ semantic distance
- "What people know when they say that they know a word is not how to recite its dictionary definition – they know how to use it [. . .] in everyday discourse." (Miller 1986)

What is the meaning of "**bardiwac**"?

Can we infer meaning from usage?

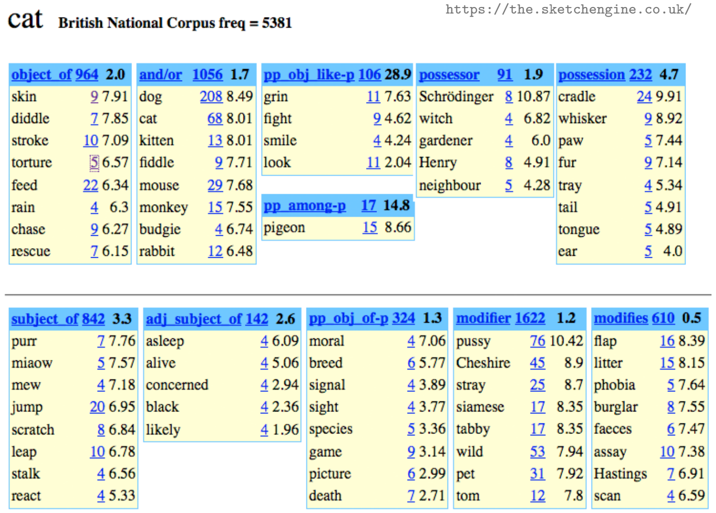
- He handed her her glass of bardiwac.
- Beef dishes are made to complement the bardiwacs.
- Nigel staggered to his feet, face flushed from too much bardiwac.
- Malbec, one of the lesser-known bardiwac grapes, responds well to Australia's sunshine.
- I dined off bread and cheese and this excellent bardiwac.
- The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.
- ☞ bardiwac is a heavy red alcoholic beverage made from grapes

All examples from British National Corpus (handpicked and slightly edited).

Word sketch of "**cat**"

Can we infer meaning from collocations (as Firth suggests)?

https://the.sketchengine.co.uk/



cat **British National Corpus freq = 5381**

object_of	964	2.0
skin	9	7.91
diddle	7	7.85
stroke	10	7.09
torture	5	6.57
feed	22	6.34
rain	4	6.3
chase	9	6.27
rescue	7	6.15

and/or	1056	1.7
dog	208	8.49
cat	68	8.01
kitten	13	8.01
fiddle	9	7.71
mouse	29	7.68
monkey	15	7.55
budgie	4	6.74
rabbit	12	6.48

pp_obj_like-p	106	28.9
grin	11	7.63
fight	9	4.62
smile	4	4.24
look	11	2.04

pp_among-p	17	14.8
pigeon	15	8.66

possessor	91	1.9
Schrödinger	8	10.87
witch	4	6.82
gardener	4	6.0
Henry	8	4.91
neighbour	5	4.28

possession	232	4.7
cradle	24	9.91
whisker	9	8.92
paw	5	7.44
fur	9	7.14
tray	4	5.34
tail	5	4.91
tongue	5	4.89
ear	5	4.0

subject_of	842	3.3
purr	7	7.76
miaow	5	7.57
mew	4	7.18
jump	20	6.95
scratch	8	6.84
leap	10	6.78
stalk	4	6.56
react	4	5.33

adj_subject_of	142	2.6
asleep	4	6.09
alive	4	5.06
concerned	4	2.94
black	4	2.36
likely	4	1.96

pp_obj_of-p	324	1.3
moral	4	7.06
breed	6	5.77
signal	4	3.89
sight	4	3.77
species	5	3.36
game	9	3.14
picture	6	2.99
death	7	2.71

modifier	1622	1.2
pussy	76	10.42
Cheshire	45	8.9
stray	25	8.7
siamese	17	8.35
tabby	17	8.35
wild	53	7.94
pet	31	7.92
tom	12	7.8

modifies	610	0.5
flap	16	8.39
litter	15	8.15
phobia	5	7.64
burglar	8	7.55
faeces	6	7.47
assay	10	7.38
Hastings	7	6.91
scan	4	6.59

A thought experiment: deciphering hieroglyphs

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

A thought experiment: deciphering hieroglyphs

(knife)		**51**	**20**	**84**	**0**	**3**	**0**
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

sim(,) = 0.770

A thought experiment: deciphering hieroglyphs

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		**12**	**17**	**3**	**2**	**9**	**27**
(banana)		11	2	2	0	18	0

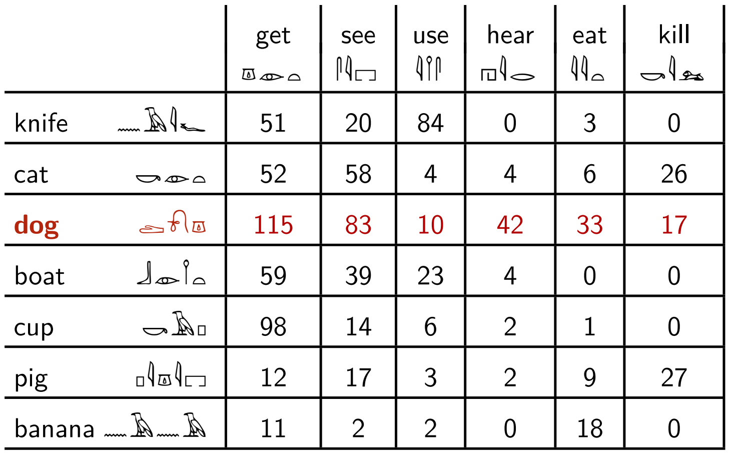
sim(,) = 0.939

A thought experiment: deciphering hieroglyphs

(knife)		51	20	84	0	3	0
(cat)		**52**	**58**	**4**	**4**	**6**	**26**
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

sim(,) = 0.961

English as seen by the computer . . .



	get	see	use	hear	eat	kill
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
dog	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

verb-object counts from British National Corpus

Geometric interpretation

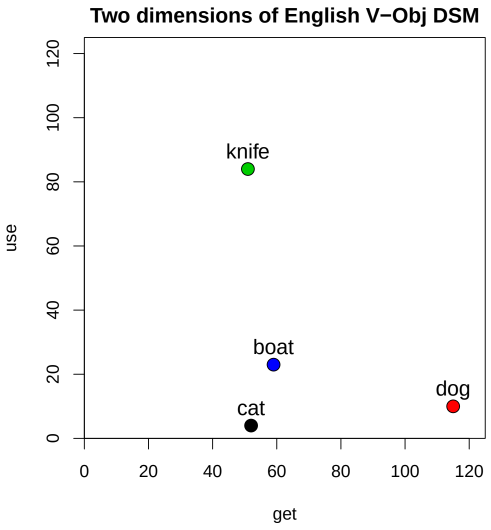
- row vector $\mathbf{x}_{dog}$ describes usage of word *dog* in the corpus
- can be seen as coordinates of point in *n*-dimensional Euclidean space

	get	see	use	hear	eat	kill
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
dog	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

co-occurrence matrix M

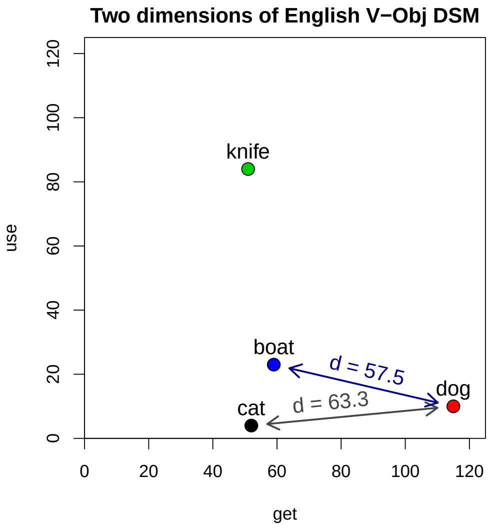
Geometric interpretation

- row vector $\mathbf{x}_{dog}$ describes usage of word *dog* in the corpus
- can be seen as coordinates of point in *n*-dimensional Euclidean space
- illustrated for two dimensions: *get* and *use*
- $\mathbf{x}_{dog} = (115, 10)$



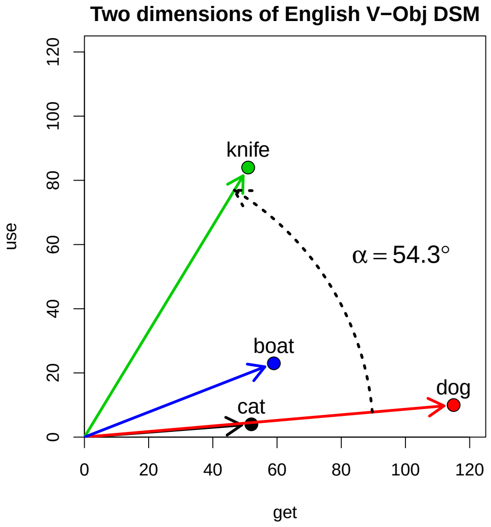
Geometric interpretation

- similarity = spatial proximity (Euclidean dist.)
- location depends on frequency of noun ($f_{dog} \approx 2.7 \cdot f_{cat}$)



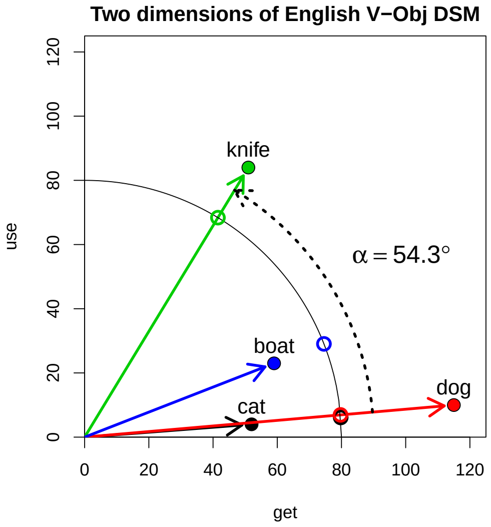
Geometric interpretation

- vector can also be understood as arrow from origin
- direction more important than location
- use angle α as distance measure



Geometric interpretation

- vector can also be understood as arrow from origin
- direction more important than location
- use angle α as distance measure
- or normalise length $\|\mathbf{x}_{\text{dog}}\|$ of arrow



Outline

Introduction

The distributional hypothesis

Distributional semantic models

DSM and semantic similarity

Course Outline

Getting practical

Software and further information

R as a (toy) laboratory

General definition of DSMs

A **distributional semantic model** (DSM) is a scaled and/or transformed co-occurrence matrix **M**, such that each row **x** represents the distribution of a target term across contexts.

	get	see	use	hear	eat	kill
knife	0.027	-0.024	0.206	-0.022	-0.044	-0.042
cat	0.031	0.143	-0.243	-0.015	-0.009	0.131
dog	-0.026	0.021	-0.212	0.064	0.013	0.014
boat	-0.022	0.009	-0.044	-0.040	-0.074	-0.042
cup	-0.014	-0.173	-0.249	-0.099	-0.119	-0.042
pig	-0.069	0.094	-0.158	0.000	0.094	0.265
banana	0.047	-0.139	-0.104	-0.022	0.267	-0.042

Term = word, lemma, phrase, morpheme, word pair, . . .

Nearest neighbours

DSM based on verb-object relations from BNC, reduced to 100 dim. with SVD

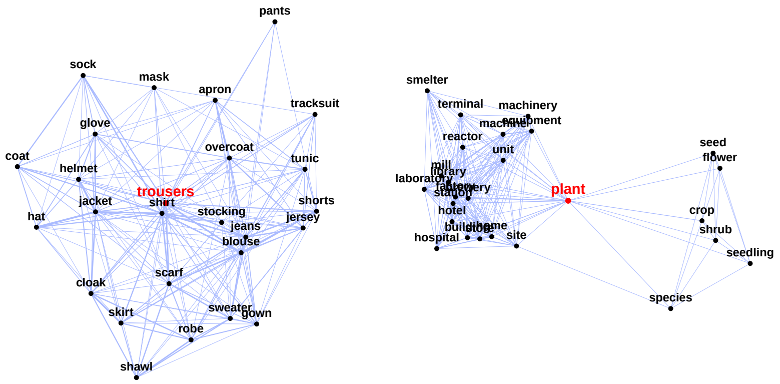
Neighbours of **trousers** (cosine angle):

- ☞ shirt (18.5), blouse (21.9), scarf (23.4), jeans (24.7), skirt (25.9), sock (26.2), shorts (26.3), jacket (27.8), glove (28.1), coat (28.8), cloak (28.9), hat (29.1), tunic (29.3), overcoat (29.4), pants (29.8), helmet (30.4), apron (30.5), robe (30.6), mask (30.8), tracksuit (31.0), jersey (31.6), shawl (31.6), . . .

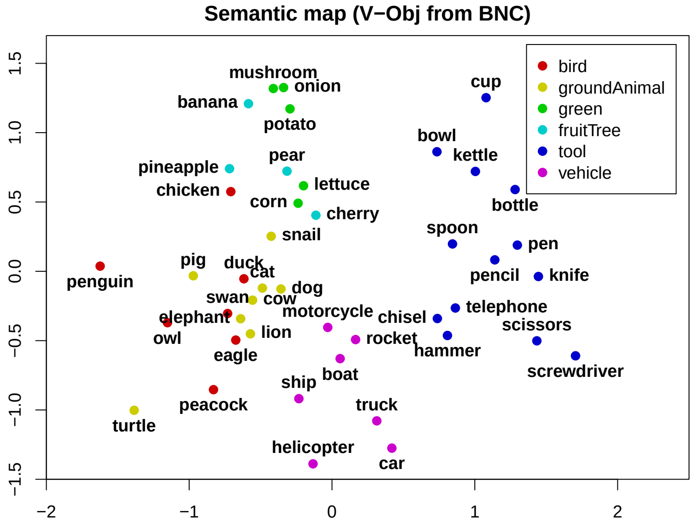
Neighbours of **rage** (cosine angle):

- ☞ anger (28.5), fury (32.5), sadness (37.0), disgust (37.4), emotion (39.0), jealousy (40.0), grief (40.4), irritation (40.7), revulsion (40.7), scorn (40.7), panic (40.8), bitterness (41.6), resentment (41.8), indignation (41.9), excitement (42.0), hatred (42.5), envy (42.8), disappointment (42.9), . . .

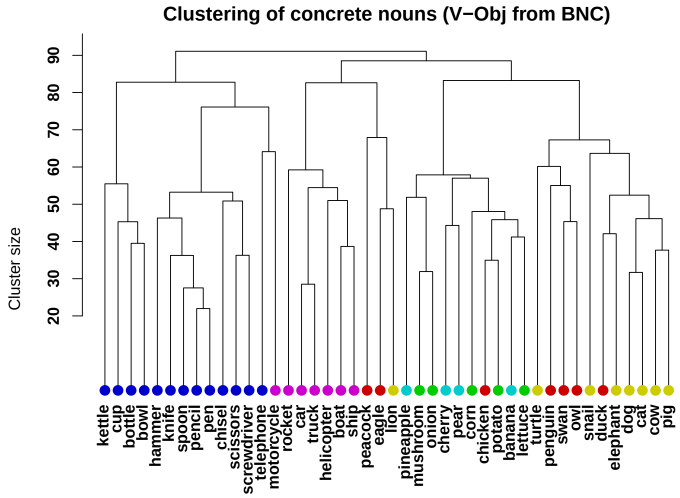
Nearest neighbours with similarity graph



Semantic maps



Clustering



DSM vectors as word embeddings

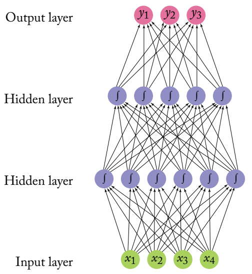
DSM vector as sub-symbolic meaning representation

- feature vector for machine learning algorithm
- input for neural network
- such distributed representations are known as **embeddings**

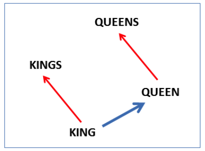
☞ embeddings $\not\approx$ distributional

Computation in semantic space

- find meaningful subdimensions in DSM space (➜ correlation)
- linear operations on vectors



(Goldberg 2017, Fig. 4.2)



(Mikolov *et al.* 2013, Fig. 2)

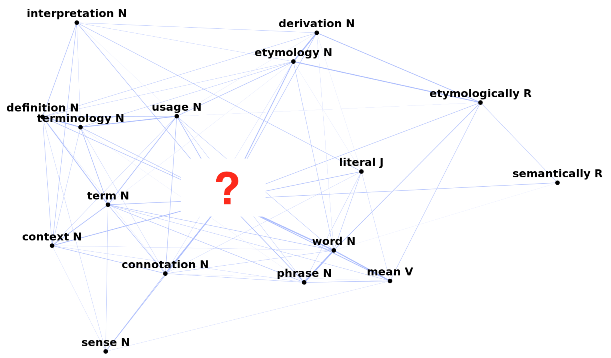
Outline

Introduction
The distributional hypothesis
Distributional semantic models
DSM and semantic similarity
Course Outline

Getting practical
Software and further information
R as a (toy) laboratory

Inverse distributional semantics

Which word “bought” the same contexts as the ones displayed in this graph?



. . . look at the neighbors:
is there one notion of similarity “to rule them all”?

Distributional similarity as semantic similarity

- DSM similarity as a quantitative notion
  - if **a** is closer to **b** than to **c** in the distributional vector space, then *a* is more semantically similar to *b* than to *c*
- DSM similarity as a graded notion, differently from categorical nature of most theoretical accounts
- DSM similarity as the empirical correlate of a heterogeneous set of phenomena

  ... which we may want to tease apart!
- DSM similarity is symmetric – cognition is not

  ... can we fix this?

Characterizing DSM similarity

- DSMs are thought to represent taxonomic similarity
  - words that tend to occur in the same contexts
- Words that share many contexts share many properties (attributes) and are thus taxonomically/ontologically similar
  - synonyms (*rhino/rhinoceros*)
  - antonyms and values on a scale (*good/bad*)
  - co-hyponyms (*rock/jazz*)
  - hyper- and hyponyms (*rock/basalt*)
- Taxonomic similarity is seen as the fundamental semantic relation organising the vocabulary of a language, allowing categorization, generalization and inheritance...

Is distributional similarity *just* taxonomic?

Nearest DSM neighbors have different types of semantic relations.

car (BNC, L2/R2 span)

- van co-hyponym
- vehicle hyperonym
- truck co-hyponym
- motorcycle co-hyponym
- driver related entity
- motor part
- lorry co-hyponym
- motorist related entity
- cavalier hyponym
- bike co-hyponym

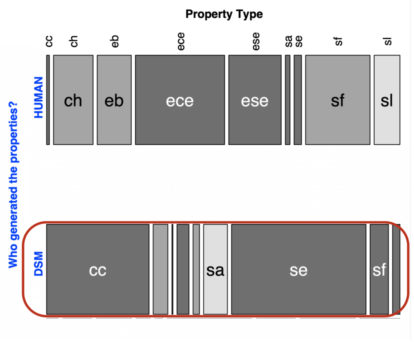
car (BNC, L30/R30 span)

- drive function
- park typical action
- bonnet part
- windscreen part
- hatchback part
- headlight part
- jaguar hyponym
- garage location
- cavalier hyponym
- tyre part

http://clic.cimec.unitn.it/infomap-query/

Is distributional similarity *just* taxonomic?

Manual annotation: what are the properties of *car*? Humans vs DSM



Taxonomic category:

cc	(co-)hyponym	*truck*
ch	hypernym	*vehicle*

Properties of entity:

eb	typical behaviour	
ece	ext. component	wheel
ese	surf. property	smooth

Situationally associated:

sa	action	*park*
se	other entity	*traffic light*
sf	function	*drive*
sl	location	*garage*
sp	participant	*driver*

Task: humans: given a word, generate properties; DSM), generate top 10 neighbors. Items: 44 concrete English nouns (Baroni & Lenci 2008).

DSM similarities: terminological coordinates

Attributional similarity vs. Semantic relatedness

- Attributional similarity (← taxonomical) – two words sharing a large number of salient features (attributes)
  - synonymy (*car/automobile*)
  - co-hyponymy (*car/van/truck*)
  - hyperonymy (*car/vehicle)*
    - ★ Problem: subset/superset, need ad-hoc measures (distributional inclusion cf. Lenci & Benotto (2012))
  - antonymy (*hot/cold*)
    - ★ Problem: they are the opposite of similar, and yet...
- Semantic relatedness (Budanitsky & Hirst 2006) – two words semantically associated without necessarily being similar
  - function (*car/drive*)
  - meronymy (*car/tyre*)
  - location (*car/road*)
  - attribute (*car/fast*)

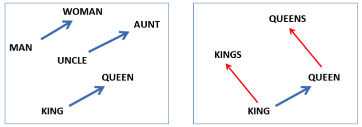
Why similar in DSMs? They co-occur → share contexts

DSM similarities: terminological coordinates

Attributional vs. Relational Similarity

- Attributional similarity (← taxonomical) – two words sharing a large number of salient features (attributes)
  - synonymy (*car/automobile*)
  - co-hyponymy (*car/van/truck*)
  - hyperonymy (*car/vehicle)*
- Relational similarity (Turney 2006) – similar relation between pairs of words (analogy)
  - *policeman* : *gun* :: *teacher* : *book*
  - *mason* : *stone* :: *carpenter* : *wood*
  - *traffic* : *street* :: *water* : *riverbed*

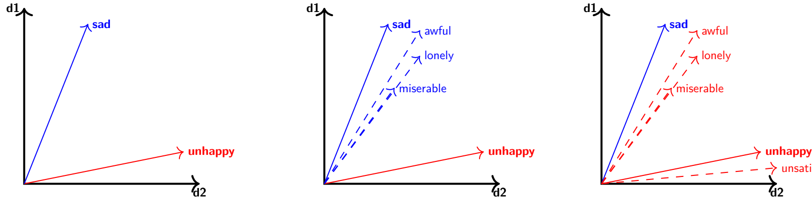
  ... textbook example of neural embeddings application



Problem: symmetry in DSM similarity

The symmetry assumption does not fit all phenomena

Solution: neighbor rank can capture (potential) asymmetries



dist(sad, unhappy) = dist(unhappy, sad) rank(sad, unhappy) = 4 rank(unhappy, sad) = 5

- Motivation: cognitive processes are notoriously asymmetric
- Advantage: rank makes similarity predictions comparable across models and is applicable to different distance measures
- Interpretation: rank controls for differences in density in the semantic space

Outline

Introduction
The distributional hypothesis
Distributional semantic models
DSM and semantic similarity
Course Outline

Getting practical
Software and further information
R as a (toy) laboratory

Day 1: Introduction

Summing up what we learnt

- A DSM is a **matrix**, which contains
  - ... targets: **rows**
  - ... contexts: **columns**
  - ... co-occurrence scores (or fancier versions of co-occurrence) for target/context pairs: **matrix cells**
- The row corresponding to a target (**vector**) is the best *approximation* we have for it its meaning
  - Goal: make comparisons (recall the hieroglyphs)
    - ⋆ Similarity as context overlap
- Geometric interpretation: vectors as coordinates in space
  - Similarity as distance
  - Neighbors reveal the semantic nuances a DSM is capturing
  - Visualization: neighbor maps
  - Neighbor rank as a way to get asymmetric similarity predictions

Roadmap: First steps in distributional semantics

- **Day 2: Building a DSM, step by step**
  - DSM parameters: formal definition & taxonomy
  - Collecting co-occurrence data: what counts as a context?
  - Mathematical operations on the DSM vectors
  - Computing distances/similarities
  - Practice: building DSMs and exploring parameters
- **Day 3: Which meaning is a DSM capturing (if any?)**
  - Evaluation: conceptual coordinates
  - Standard evaluation tasks: multiple choice, prediction of similarity ratings, clustering
  - Narrowing down similarity: classifying semantic relations
  - Practice: evaluation of selected tasks

Roadmap: Interdisciplinary applications

- **Day 4: DS beyond NLP – Linguistic theory**
  - Linguistic exploitation of distributional representations
  - A textbook challenge for DSMs: polysemy
  - Success stories: semantic compositionality (belown and above word level), morphological transparency, argument structure
  - Issues: not all words have a (straightforward) DS meaning
  - Practice: word sense disambiguation & modeling of morphological derivation
- **Day 5: DS beyond NLP – Cognitive modelling**
  - DSMs for cognitive modeling: general issues
  - Free association norms as a window into the organization of the mental lexicon
  - Predicting free associations with DSMs
  - Practice: combine DSMs with first-order co-occurrence in the FAST free association task

Outline

Introduction

- The distributional hypothesis
- Distributional semantic models
- DSM and semantic similarity
- Course Outline

Getting practical

- Software and further information
- R as a (toy) laboratory

Some applications in computational linguistics

- Query expansion in information retrieval (Grefenstette 1994)
- Unsupervised part-of-speech induction (Schütze 1995)
- Word sense disambiguation (Schütze 1998; Rapp 2004b)
- Thesaurus compilation (Lin 1998; Rapp 2004a)
- Attachment disambiguation (Pantel & Lin 2000)
- Probabilistic language models (Bengio *et al.* 2003)
- Translation equivalents (Sahlgren & Karlgren 2005)
- Ontology & wordnet expansion (Pantel *et al.* 2009)
- Language change (Sagi *et al.* 2009; Hamilton *et al.* 2016)
- Multiword expressions (Kiela & Clark 2013)
- Analogies (Turney 2013; Gladkova *et al.* 2016)
- Sentiment analysis (Rothe & Schütze 2016; Yu *et al.* 2017)

☞ Input representation for neural networks & machine learning

Software packages

Infomap NLP	C	*classical LSA-style DSM*
HiDEx	C++	*re-implementation of the HAL model (Lund & Burgess 1996)*
SemanticVectors	Java	*scalable architecture based on random indexing representation*
S-Space	Java	*complex object-oriented framework*
JoBimText	Java	*UIMA / Hadoop framework*
Gensim	Python	*complex framework, focus on parallelization and out-of-core algorithms*
Vecto	Python	*framework for count & predict models*
DISSECT	Python	*user-friendly, designed for research on compositional semantics*
`wordspace`	R	*interactive research laboratory, but scales to real-life data sets*
`text2vec`	R	*GloVe embeddings & topic models*

click on package name to open Web page

Further information

- Handouts & other materials available from wordspace wiki at

  `http://wordspace.collocations.de/`

  ☞ based on joint work with Marco Baroni and Alessandro Lenci
- Tutorial is open source (CC), and can be downloaded from

  `http://r-forge.r-project.org/projects/wordspace/`

Further information

- Review papers on distributional semantics:

  Turney, Peter D. and Pantel, Patrick (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37, 141–188.

  Erk, Katrin (2012). Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6-1, 635–653.

  Boleda, Gemma (2020). Distributional Semantics and Linguistic Theory. *Annual Review of Linguistics*, 6-1, 213–234.
- We should be working on a textbook *Distributional Semantics* for *Synthesis Lectures on HLT* (Morgan & Claypool)

Outline

Introduction

- The distributional hypothesis
- Distributional semantic models
- DSM and semantic similarity
- Course Outline

Getting practical

- Software and further information
- R as a (toy) laboratory

Prepare to get your hands dirty . . .

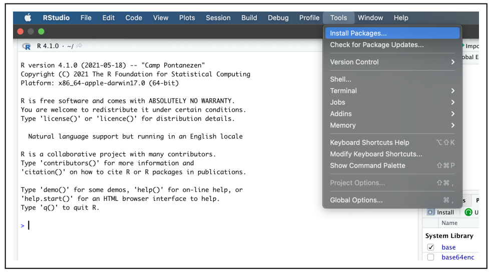
- We will use the statistical programming environment **R** as a toy laboratory in this tutorial
  - ☞ but one that scales to real-life applications

Software installation

- **R** version 4.0 or newer from `http://www.r-project.org/`
- RStudio from `http://www.rstudio.com/`
- R packages from CRAN (through RStudio menu)
  - `sparsesvd`, `wordspace`
  - recommended: `e1071`, `rsparse`, `Rtsne`, `uwot`
  - optional: `tm`, `quanteda`, `data.table`, `wordcloud`, `shiny`, `spacyr`, `udpipe`, `coreNLP`
- Get data sets, precompiled DSMs and `wordspaceEval` package (with some non-public data sets) from http://wordspace.collocations.de/doku.php/course:material

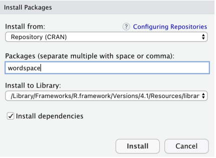
Prepare to get your hands dirty . . .

Installing `wordspace` and `wordspaceEval` in RStudio



```
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

`wordspace`

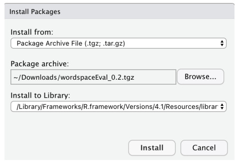


Install from: Repository (CRAN)

Packages (separate multiple with space or comma): wordspace

Install to Library: /Library/Frameworks/R.framework/Versions/4.1/Resources/librar

☑ Install dependencies



Install from: Package Archive File (.tgz; .tar.gz)

Package archive: ~/Downloads/wordspaceEval_0.2.tgz

Install to Library: /Library/Frameworks/R.framework/Versions/4.1/Resources/librar

`wordspaceEval`

Prepare to get your hands dirty . . .

Setting up a working directory and RStudio project

- Create a separate **directory** (folder) for this course
  - subdirectory `models` for pre-compiled DSMs (large files)
  - subdirectory `data` for other data files
- Recommended: set up **RStudio project** for the course
  - click *New Project* (top right corner), then *Existing Directory*
  - choose the course directory you've just created
  - this will be set as your R working directory within the project!
  - ☞ you can easily switch between different RStudio projects
- Alternatively: set working directory at start of session
  - e.g. `setwd("~gabriella/Desktop/ESSLLI22")`
- Work with **R scripts** rather than in interactive console
  - RStudio: add *R Script* from drop-down menu in top left corner
  - we provide example scripts for each hands-on session (+extras)

First steps in R

Start each session by loading the wordspace package.

```
> library(wordspace)
```

The package includes various example data sets, some of which should look familiar to you.

```
> DSM_HieroglyphsMatrix
       get see use hear eat kill
knife   51  20  84    0   3    0
cat     52  58   4    4   6   26
dog    115  83  10   42  33   17
boat    59  39  23    4   0    0
cup     98  14   6    2   1    0
pig     12  17   3    2   9   27
banana  11   2   2    0  18    0
```

Term-term matrix

Term-term matrix records co-occurrence frequencies with feature terms for each target term

```
> DSM_TermTermMatrix
```

	breed	tail	feed	kill	important	explain	likely
cat	83	17	7	37	–	1	-x1-
dog	561	13	30	60	1	2	4
animal	42	10	109	134	13	5	5
time	19	9	29	117	81	34	109
reason	1	–	2	14	68	140	47
cause	–	1	–	4	55	34	55
effect	–	–	1	6	60	35	17

Term-context matrix

Term-context matrix records frequency of term in each individual context (e.g. sentence, document, Web page, encyclopaedia article)

```
> DSM_TermContextMatrix
```

	Felidae	Pet	Feral	Bloat	Philosophy	Kant	Back pain
cat	10	10	7	–	–	–	–
dog	–	10	4	11	–	–	–
animal	2	15	10	2	–	–	–
time	1	–	–	–	2	1	–
reason	–	1	–	–	1	4	1
cause	–	–	–	2	1	2	6
effect	–	–	–	1	–	1	–

Playing with a larger model

Term-term matrix, dimensionality-reduced, built from Web texts for target words in the format *lemma_POS* (e.g. banana_N)

```
> DSM_Vectors
> View(DSM_Vectors)
```

Let's inspect some nearest neighbors:

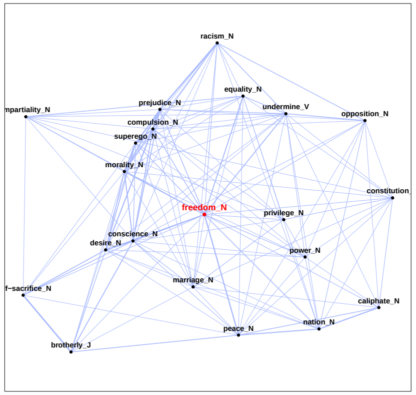
```
> nearest.neighbours(DSM_Vectors, "banana_N", n=4)
   coconut_N  pineapple_N watermelon_N       bean_N
    10.86118     12.60826     13.35160     13.79671
```

```
> nearest.neighbours(DSM_Vectors, "freedom_N", n=4)
     peace_N    morality_N   equality_N conscience_N
    30.13420     34.18397     34.23418     34.23894
```

Playing with a larger model

Or create a semantic map for a word we are interested in:

```
> plot(nearest.neighbours(DSM_Vectors, "freedom_N", n=20,
                          dist.matrix=TRUE))
```



… and with an even larger model

You can download several **large pre-compiled DSMs** from the course wiki, which represent different parameters of the co-occurrence matrix (➜ part 2).

- e.g. `WP500_DepFilter_Lemma.rda`
- download this file to subdirectory `models`

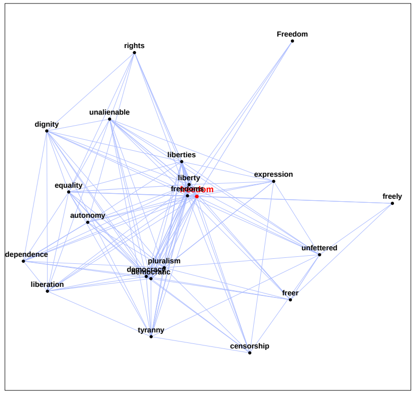
```
> load("models/WP500_DepFilter_Lemma.rda", verbose=TRUE)
Loading objects:
  WP500_DepFilter_Lemma
> model <- WP500_DepFilter_Lemma # assign to a shorter name
```

Now try the semantic map again:

```
> plot(nearest.neighbours(model, "freedom_N", n=20,
                          dist.matrix=TRUE))
```

Freedom in a neural embedding model: `word2vec`

```
> load("GoogleNews300_wf200k.rda", verbose=TRUE)
> embeddings <- GoogleNews300_wf200k.rda
> plot(nearest.neighbours(embeddings, "freedom_N", n=20,
                          dist.matrix=TRUE))
```



Bonus: Recreate the hieroglyphs example

```
# apply log-transformation to de-skew co-occurrence frequencies
> M <- log2(DSM_HieroglyphsMatrix + 1) # see part 2
> round(M, 3)

# compute semantic distance (cosine similarity)
> pair.distances("dog", "cat", M, convert=FALSE)
  dog/cat
0.9610952

# find nearest neighbours
> nearest.neighbours(M, "dog", n=3)
     cat      pig      cup
16.03458 20.08826 31.77784

> plot(nearest.neighbours(M, "dog", n=5, dist.matrix=TRUE))
```

Explorations

While you wait for part 2,
you can explore some DSM similarity networks online:

- https://corpora.linguistik.uni-erlangen.de/shiny/wordspace/
- built in R with `wordspace` and `shiny`

References I

Baroni, Marco and Lenci, Alessandro (2008). Concepts and properties in word spaces. *Italian Journal of Linguistics*, **20**(1).

Bengio, Yoshua; Ducharme, Réjean; Vincent, Pascal; Jauvin, Christian (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, **3**, 1137–1155.

Boleda, Gemma (2020). Distributional semantics and linguistic theory. *Annual Review of Linguistics*, **6**(1), 213–234.

Budanitsky, Alexander and Hirst, Graeme (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, **32**(1), 13–47.

Erk, Katrin (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, **6**(10), 635–653.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–55. In *Studies in linguistic analysis*, pages 1–32. The Philological Society, Oxford.

Gladkova, Anna; Drozd, Aleksandr; Matsuoka, Satoshi (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California.

References II

Goldberg, Yoav (2017). *Neural Network Methods for Natural Language Processing*. Number 37 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Grefenstette, Gregory (1994). *Explorations in Automatic Thesaurus Discovery*, volume 278 of *Kluwer International Series in Engineering and Computer Science*. Springer, Berlin, New York.

Hamilton, William L.; Leskovec, Jure; Jurafsky, Dan (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany.

Harris, Zellig (1954). Distributional structure. *Word*, **10**(23), 146–162.

Kiela, Douwe and Clark, Stephen (2013). Detecting compositionality of multi-word expressions using nearest neighbours in vector space models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1427–1432, Seattle, WA.

References III

Lenci, Alessandro and Benotto, Giulia (2012). Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

Lin, Dekang (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-ACL 1998)*, pages 768–774, Montreal, Canada.

Lund, Kevin and Burgess, Curt (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, **28**(2), 203–208.

Mikolov, Tomas; Yih, Wen-tau; Zweig, Geoffrey (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, GA.

Miller, George A. (1986). Dictionaries in the mind. *Language and Cognitive Processes*, **1**, 171–185.

References IV

Pantel, Patrick and Lin, Dekang (2000). An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hongkong, China.

Pantel, Patrick; Crestan, Eric; Borkovsky, Arkady; Popescu, Ana-Maria; Vyas, Vishnu (2009). Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 938–947, Singapore.

Rapp, Reinhard (2004a). A freely available automatically generated thesaurus of related words. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 395–398.

Rapp, Reinhard (2004b). A practical solution to the problem of automatic word sense induction. In *Proceedings of the ACL-2004 Interactive Posters and Demonstrations Sessions*, pages 194–197, Barcelona, Spain. Association for Computational Linguistics.

Rothe, Sascha and Schütze, Hinrich (2016). Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–517, Berlin, Germany.

References V

Sagi, Eyal; Kaufmann, Stefan; Clark, Brady (2009). Semantic density analysis: Comparing word meaning across time and phonetic space. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics (GEMS)*, pages 104–111, Athens, Greece.

Sahlgren, Magnus and Karlgren, Jussi (2005). Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, **11**, 327–341.

Schütze, Hinrich (1995). Distributional part-of-speech tagging. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1995)*, pages 141–148.

Schütze, Hinrich (1998). Automatic word sense discrimination. *Computational Linguistics*, **24**(1), 97–123.

Turney, Peter D. (2006). Similarity of semantic relations. *Computational Linguistics*, **32**(3), 379–416.

Turney, Peter D. (2013). Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, **1**, 353–366.

Turney, Peter D. and Pantel, Patrick (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, **37**, 141–188.

References VI

Yu, Liang-Chih; Wang, Jin; Lai, K. Robert; Zhang, Xuejie (2017). Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539, Copenhagen, Denmark.