# Distributional and compositional semantics

Distributional word clustering

Selectional preferences

Compositional semantics

Compositional distributional semantics

# Outline.

Distributional word clustering

Selectional preferences

Compositional semantics

Compositional distributional semantics

# Clustering

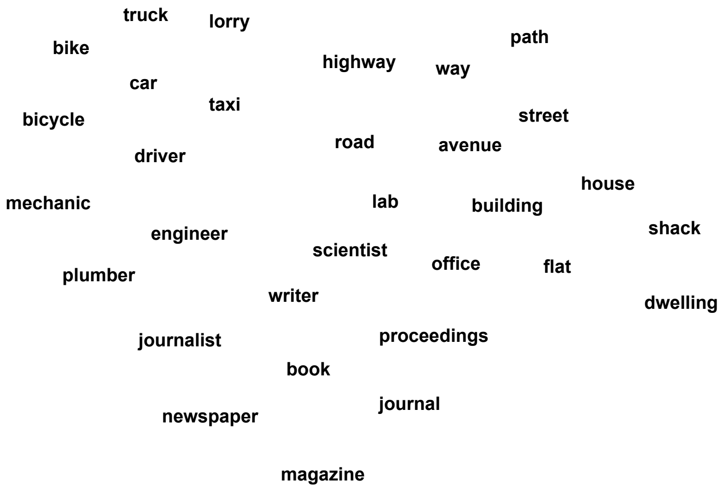
- clustering techniques group objects into clusters
- similar objects in the same cluster, dissimilar objects in different clusters
- allows us to obtain generalisations over the data
- widely used in various NLP tasks:
  - semantics (e.g. word clustering);
  - summarization (e.g. sentence clustering);
  - text mining (e.g. document clustering).

# Distributional word clustering

We will:

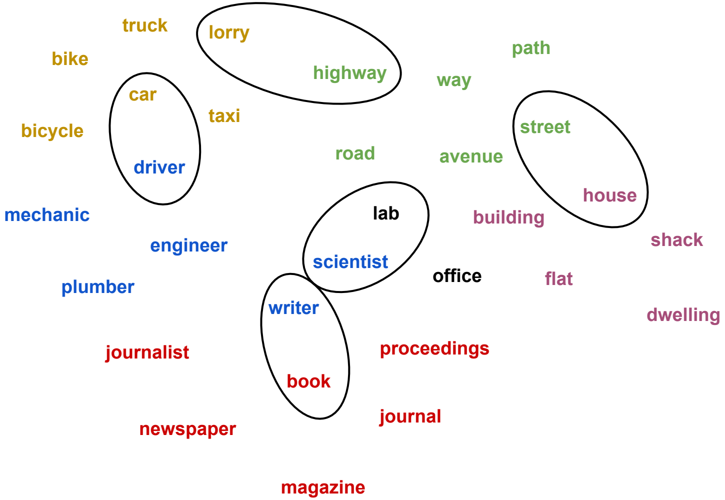
- cluster words based on the contexts in which they occur
- assumption: words with similar meanings occur in similar contexts, i.e. are distributionally similar
- we will consider noun clustering as an example
- cluster 2000 nouns – most frequent in the British National Corpus
- into 200 clusters

# Clustering nouns



# Clustering nouns

# Clustering nouns



# Feature vectors

- can use different kinds of context as features for clustering
  - window based context
  - parsed or unparsed
  - syntactic dependencies
- different types of context yield different results
- Example experiment: use verbs that take the noun as a direct object or a subject as features for clustering
- Feature vectors: verb lemmas, indexed by dependency type, e.g. subject or direct object
- Feature values: corpus frequencies

# Extracting feature vectors: Examples

| **tree (Dobj)** | **crop (Dobj)** | **tree (Subj)** | **crop (Subj)** |
|---|---|---|---|
| 85 plant_v | 76 grow_v | 131 grow_v | 78 grow_v |
| 82 climb_v | 44 produce_v | 49 plant_v | 23 yield_v |
| 48 see_v | 16 harvest_v | 40 stand_v | 10 sow_v |
| 46 cut_v | 12 plant_v | 26 fell_v | 9 fail_v |
| 27 fall_v | 10 ensure_v | 25 look_v | 8 plant_v |
| 26 like_v | 10 cut_v | 23 make_v | 7 spray_v |
| 23 make_v | 9 yield_v | 22 surround_v | 7 come_v |
| 23 grow_v | 9 protect_v | 21 show_v | 6 produce_v |
| 22 use_v | 9 destroy_v | 20 seem_v | 6 feed_v |
| 22 round_v | 7 spray_v | 20 overhang_v | 6 cut_v |
| 20 get_v | 7 lose_v | 20 fall_v | 5 sell_v |
| 18 hit_v | 6 sell_v | 19 cut_v | 5 make_v |
| 18 fell_v | 6 get_v | 18 take_v | 5 include_v |
| 18 bark_v | 5 support_v | 18 go_v | 5 harvest_v |
| 17 want_v | 5 see_v | 18 become_v | 4 follow_v |
| 16 leave_v | 5 raise_v | 17 line_v | 3 ripen_v |
| ... | ... | ... | ... |

# Feature vectors: Examples

**tree**
131 grow_v_Subj
85 plant_v_Dobj
82 climb_v_Dobj
49 plant_v_Subj
48 see_v_Dobj
46 cut_v_Dobj
40 stand_v_Subj
27 fall_v_Dobj
26 like_v_Dobj
26 fell_v_Subj
25 look_v_Subj
23 make_v_Subj
23 make_v_Dobj
23 grow_v_Dobj
22 use_v_Dobj
22 surround_v_Subj
22 round_v_Dobj
20 overhang_v_Subj
...

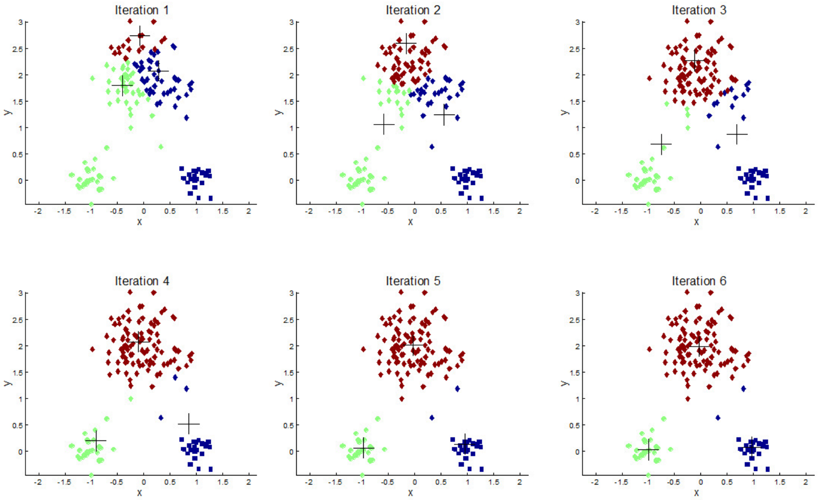
**crop**
78 grow_v_Subj
76 grow_v_Dobj
44 produce_v_Dobj
23 yield_v_Subj
16 harvest_v_Dobj
12 plant_v_Dobj
10 sow_v_Subj
10 ensure_v_Dobj
10 cut_v_Dobj
9 yield_v_Dobj
9 protect_v_Dobj
9 fail_v_Subj
9 destroy_v_Dobj
8 plant_v_Subj
7 spray_v_Subj
7 spray_v_Dobj
7 lose_v_Dobj
6 feed_v_Subj
...

# Clustering algorithms, K-means

- many clustering algorithms are available
- example algorithm: K-means clustering
  - given a set of $N$ data points $\{x_1, x_2, ..., x_N\}$
  - partition the data points into $K$ clusters $C = \{C_1, C_2, ..., C_K\}$
  - minimize the sum of the squares of the distances of each data point to the cluster mean vector $\mu_i$:

$$\arg\min_{C} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{1}$$

# K-means clustering



# Noun clusters

| |
|---|
| tree crop flower plant root leaf seed rose wood grain stem forest garden |
| consent permission concession injunction licence approval |
| lifetime quarter period century succession stage generation decade phase interval future |
| subsidy compensation damages allowance payment pension grant |
| carriage bike vehicle train truck lorry coach taxi |
| official officer inspector journalist detective constable police policeman reporter |
| girl other woman child person people |
| length past mile metre distance inch yard |
| tide breeze flood wind rain storm weather wave current heat |
| sister daughter parent relative lover cousin friend wife mother husband brother father |

# We can also cluster verbs...

| |
|---|
| sparkle glow widen flash flare gleam darken narrow flicker shine blaze bulge |
| gulp drain stir empty pour sip spill swallow drink pollute seep flow drip purify ooze pump bubble splash ripple simmer boil tread |
| polish clean scrape scrub soak |
| kick hurl push fling throw pull drag haul |
| rise fall shrink drop double fluctuate dwindle decline plunge decrease soar tumble surge spiral boom |
| initiate inhibit aid halt trace track speed obstruct impede accelerate slow stimulate hinder block |
| work escape fight head ride fly arrive travel come run go slip move |

# Uses of word clustering in NLP

- Noun and verb clustering are typically used in NLP
- for lexical acquisition tasks
  - to automatically create large-scale lexical resources to support other NLP tasks
  - e.g. selectional preferences
- dimensionality reduction for other statistical models
  - to reduce negative effects of sparse data

# Outline.

Distributional word clustering

Selectional preferences

Compositional semantics

Compositional distributional semantics

# Selectional preferences

- Selectional preferences are the semantic constraints that a predicate places onto its arguments.
- i.e. certain classes of entities are more likely to fill the predicate's argument slot than others.

The authors wrote a new paper.

Watch out, the cat is eating your sausage!

*The carrot ate the keys.

*The law sang a driveway.

# Selectional preferences

- Selectional preferences are the semantic constraints that a predicate places onto its arguments.
- i.e. certain classes of entities are more likely to fill the predicate's argument slot than others.

The authors wrote a new paper.

Watch out, the cat is eating your sausage!

*The carrot ate the keys.

*The law sang a driveway.

# Selectional preferences

- Selectional preferences are the semantic constraints that a predicate places onto its arguments.
- i.e. certain classes of entities are more likely to fill the predicate's argument slot than others.

The authors wrote a new paper.

Watch out, the cat is eating your sausage!

*The carrot ate the keys.

*The law sang a driveway.

# Selectional preferences

- Selectional preferences are the semantic constraints that a predicate places onto its arguments.
- i.e. certain classes of entities are more likely to fill the predicate's argument slot than others.

The authors wrote a new paper.

Watch out, the cat is eating your sausage!

*The carrot ate the keys.

*The law sang a driveway.

# Selectional preferences

- Selectional preferences are the semantic constraints that a predicate places onto its arguments.
- i.e. certain classes of entities are more likely to fill the predicate's argument slot than others.

The authors wrote a new paper.

Watch out, the cat is eating your sausage!

*The carrot ate the keys.

*The law sang a driveway.

# Learning selectional preferences from distributions

[animate] eat [food]

[person] sing [song]

[person] read [book]

1. Need to define a set of argument classes that can fill the argument slot of the predicate

   e.g. use noun clusters for this purpose
2. Need to quantify the level of association of a particular verb with a particular noun class

# Selectional preference model

Phillip Resnik, 1997. *Selectional Preference and Sense Disambiguation*

## Selectional preference strength

$$S_R(v) = D_{KL}(P(c|v)||P(c)) = \sum_c P(c|v) \log \frac{P(c|v)}{P(c)}$$

$D_{KL}$ is Kullback–Leibler divergence

## Selectional association

$$A_R(v,c) = \frac{1}{S_R(v)} P(c|v) \log \frac{P(c|v)}{P(c)}$$

$P(c)$ is the prior probability of the noun class;
$P(c|v)$ its posterior probability given the verb; $R$ is the grammatical relation

# Calculating probabilities

$$P(c) = \frac{f(c)}{\sum_k f(c_k)},$$

$$P(c|v) = \frac{f(v,c)}{f(v)},$$

$$f(c) = \sum_{n_i \in c} f(n_i)$$

$f(v,c)$: frequency of verb $v$ co-occurring with the noun class $c$
$f(v)$: total frequency of verb $v$ with all noun classes
$f(c)$: total frequency of the noun class $c$

# Selectional preferences of ***kill*** (Dobj)

0.38 girl other woman child person people
0.20 being species sheep animal creature horse baby human fish male lamb bird rabbit female insect cattle mouse monster
0.19 sister daughter parent relative lover cousin friend wife mother husband brother father
0.04 thousand citizen inhabitant resident minority youngster refugee peasant miner hundred
0.0378 gene tissue cell particle fragment bacterium protein acid complex compound molecule organism
0.0336 fleet soldier knight force rebel guard troops crew army pilot
0.0335 official officer inspector journalist detective constable police policeman reporter
0.0322 victim bull teenager prisoner hero gang enemy rider offender youth killer thief driver defender hell
0.0136 week month year
...

# Selectional preferences of ***drink*** (Dobj)

0.5831 drink coffee champagne pint wine beer
0.2778 drop tear sweat paint blood water juice
0.1084 mixture salt dose ingredient sugar substance drug milk cream alcohol fibre chemical
0.0515 brush bowl bucket receiver barrel dish glass container plate basket bottle tray
0.0069 couple minute night morning hour time evening afternoon
0.0041 stability efficiency security prospects health welfare survival safety
0.0025 recording music tape song tune radio guitar trick album football organ stuff
0.0005 rage excitement panic anger terror flame laughter
0.0004 ball shot kick arrow stroke bullet punch bomb shell blow missile
0.0003 lunch dinner breakfast meal
...

# Selectional preferences of ***cut*** (Dobj)

0.2845 expenditure cost risk expense emission budget spending
0.1527 dividend price rate premium rent rating salary wages
0.0832 employment investment growth supplies sale import export production consumption traffic input spread supply flow
0.0738 potato apple slice food cake meat bread fruit
0.0407 stitch brick metal bone strip cluster coffin stone piece tile fabric rock layer remains block
0.0379 excess deficit inflation unemployment pollution inequality poverty delay discrimination symptom shortage
0.0366 tree crop flower plant root leaf seed rose wood grain stem forest garden
0.0330 tail collar strand skirt trousers hair curtain sleeve
0.0244 rope hook cable wire thread ring knot belt chain string
...

# Different senses of ***run***

The children **ran** to the store
If you see this man, **run**!
Service **runs** all the way to Cranbury
She is **running** a relief operation in Sudan
the story or argument **runs** as follows
Does this old car still **run** well?
Interest rates **run** from 5 to 10 percent
Who’s **running** for treasurer this year?
They **ran** the tapes over and over again
These dresses **run** small

# Selectional preferences of ***run*** (Subj)

0.2125 drop tear sweat paint blood water juice
0.1665 technology architecture program system product version interface software tool computer network processor chip package
0.1657 tunnel road path trail lane route track street bridge
0.1166 carriage bike vehicle train truck lorry coach taxi
0.0919 tide breeze flood wind rain storm weather wave current heat
0.0865 tube lock tank circuit joint filter battery engine device disk furniture machine mine seal equipment machinery wheel motor slide disc instrument
0.0792 ocean canal stream bath river waters pond pool lake
0.0497 rope hook cable wire thread ring knot belt chain string
0.0469 arrangement policy measure reform proposal project programme scheme plan course
0.0352 week month year
0.0351 couple minute night morning hour time evening afternoon

# Selectional preferences of ***run*** (continued)

0.0341 criticism appeal charge application allegation claim objection suggestion case complaint
0.0253 championship open tournament league final round race match competition game contest
0.0218 desire hostility anxiety passion doubt fear curiosity enthusiasm impulse instinct emotion feeling suspicion
0.0183 expenditure cost risk expense emission budget spending
0.0136 competitor rival team club champion star winner squad county player liverpool partner leeds
0.0102 being species sheep animal creature horse baby human fish male lamb bird rabbit female insect cattle mouse monster
...

# Uses of selectional preferences

Widely used in NLP as a source of lexical information:

- Word sense induction and disambiguation
- Parsing (resolving ambiguous attachments)
- Identifying figurative language and idioms
- Paraphrasing and paraphrase detection
- Natural language inference (e.g. in the entailment identification task)
- etc.

We looked at verbs, other parts of speech (e.g. adjectives) can have preferences too.

# Outline.

Distributional word clustering

Selectional preferences

Compositional semantics

Compositional distributional semantics

# Compositional semantics

- Principle of Compositionality: meaning of each whole phrase derivable from meaning of its parts.
- Sentence structure conveys some meaning
- Formal semantics: sentence meaning as logical form

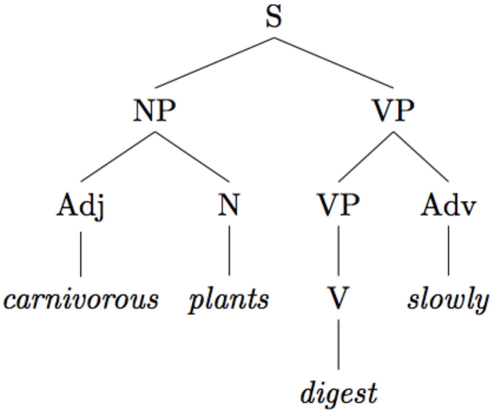
*Kitty chased Rover.*
*Rover was chased by Kitty.*

$$\exists x, y[\text{chase}'(x, y) \wedge \text{Kitty}'(x) \wedge \text{Rover}'(y)]$$

or $\text{chase}'(k, r)$ if $k$ and $r$ are constants (*Kitty* and *Rover*)

- Deep grammars: model semantics alongside syntax, one semantic composition rule per syntax rule

# Compositional semantics alongside syntax



# Semantic composition is non-trivial

- Similar syntactic structures may have different meanings:

  *it barks*
  *it rains; it snows* – *pleonastic pronouns*

- Different syntactic structures may have the same meaning:

  *Kim seems to sleep.*
  *It seems that Kim sleeps.*

- Not all phrases are interpreted compositionally, e.g. idioms:

  *red tape*
  *kick the bucket*

  but they can be interpreted compositionally too, so we can not simply block them.

# Semantic composition is non-trivial

- Elliptical constructions where additional meaning arises through composition, e.g. logical metonymy:

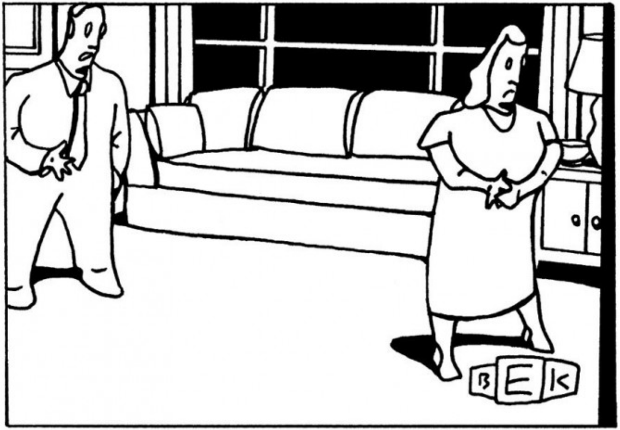
  *fast programmer*
  *fast plane*

- Meaning transfer and additional connotations that arise through composition, e.g. metaphor

  *I cant **buy** this story.*
  *This sum will **buy** you a ride on the train.*

- Recursion

# Recursion




*"Of course I care about how you imagined I thought you perceived I wanted you to feel."*

# Outline.

Distributional word clustering

Selectional preferences

Compositional semantics

Compositional distributional semantics

# Compositional distributional semantics

Can distributional semantics be extended to account for the meaning of phrases and sentences?

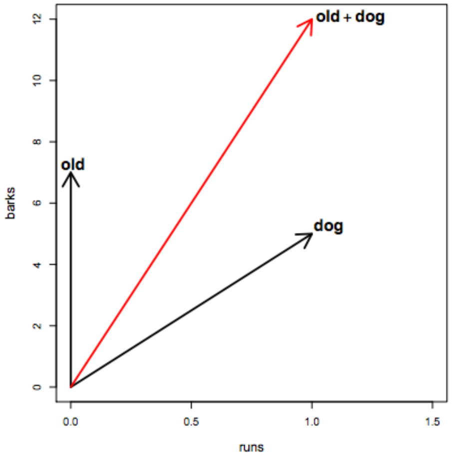
- Language can have an infinite number of sentences, given a limited vocabulary
- So we can not learn vectors for all phrases and sentences
- and need to do composition in a distributional space

# 1. Vector mixture models

Mitchell and Lapata, 2010.
*Composition in Distributional Models of Semantics*

Models:

- Additive
- Multiplicative



# Additive and multiplicative models

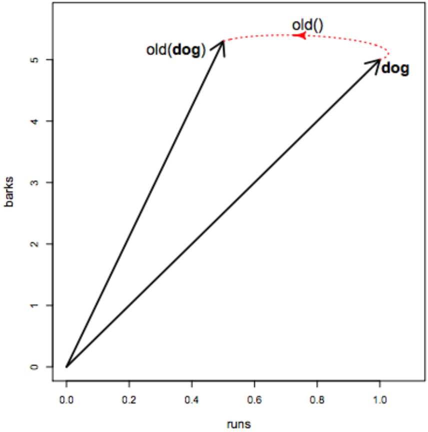
| | | | | additive | | multiplicative | |
|---|---|---|---|---|---|---|---|
| | **dog** | **cat** | **old** | **old + dog** | **old + cat** | **old ⊙ dog** | **old ⊙ cat** |
| runs | 1 | 4 | 0 | 1 | 4 | 0 | 0 |
| barks | 5 | 0 | 7 | 12 | 7 | 35 | 0 |

- correlate with human similarity judgments about adjective-noun, noun-noun, verb-noun and noun-verb pairs
- but... commutative, hence do not account for word order *John hit the ball* = *The ball hit John*!
- more suitable for modelling content words, would not port well to function words:
  e.g. *some dogs; lice and dogs; lice on dogs*

# 2. Lexical function models

Distinguish between:

- words whose meaning is directly determined by their distributional behaviour, e.g. nouns
- words that act as functions transforming the distributional profile of other words, e.g., verbs, adjectives and prepositions



# Lexical function models

Baroni and Zamparelli, 2010. *Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space.*

Adjectives as lexical functions

$$old\ dog = F_{old}(dog)$$

- Adjectives are parameter matrices ($\Theta_{old}$ , $\Theta_{furry}$, etc.).
- Nouns are vectors (house, dog, etc.).
- Composition is simply $old\ dog = \Theta_{old} \times dog$.

# Learning adjective matrices

1. Obtain vector $n_j$ for each noun $n_j$ in lexicon.
2. Collect adjective noun pairs $(a_i, n_j)$ from corpus.
3. Obtain vector $h_{ij}$ of each bi-gram $(a_i, n_j)$
4. The set of tuples $\{(n_j, h_{ij})\}_j$ is a dataset $D_i$ for adj. $a_i$
5. Learn matrix $\Theta_i$ from $D_i$ using linear regression.

| **OLD** | runs | barks |
|---|---|---|
| runs | 0.5 | 0 |
| barks | 0.3 | 1 |

$\times$

| | **dog** |
|---|---|
| runs | 1 |
| barks | 5 |

$=$

| | **OLD(dog)** |
|---|---|
| runs | $(0.5 \times 1) + (0 \times 5) = 0.5$ |
| barks | $(0.3 \times 1) + (5 \times 1) = 5.3$ |