Unsupervised Language Learning: Representation Learning for NLP

Katia Shutova

ILLC
University of Amsterdam

3 April 2018

Taught by...

- Lecturers: Katia Shutova and Wilker Aziz

- Teaching assistant: Samira Abnar

Lecture 1: Introduction

Overview of the course

Distributional semantics

Count-based models

Similarity

Distributional word clustering

Overview of the course

- This course is about learning meaning representations
  - Methods for learning meaning representations from linguistic data
  - Analysis of meaning representations learnt
  - Applications
- This is a research seminar
  - Lectures
  - You will present and critique research papers,
  - implement and evaluate representation learning methods
  - and analyse their behaviour

Overview of the course

We will cover the following topics:

- Introduction to distributional semantics
- Learning word and phrase representations – deep learning
- Learning word representations – Bayesian learning
- Multilingual word representations
- Multimodal word representations (language and vision)
- Applications: NLP and neuroscience

Assessment

Work in groups of 2.

- Presentation and participation (20%)
  - Present 1 paper per group in class
- Practical assignments, assessed by reports
  1. Analysis of the properties of word representations (10%)
  2. Implement 3 representation learning methods (20%)
  3. Evaluate in the context of external NLP applications – final report (50%)

More information at the first lab session on Thursday, 5 April.

Also note:

Course materials and more info:
`https://uva-slpl.github.io/ull/`

Contact

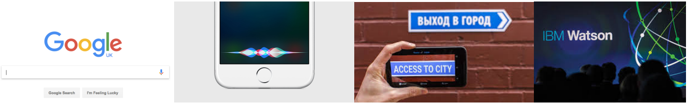
- Main contact – Samira: `s.abnar@uva.nl`
- Katia: `e.shutova@uva.nl`
- Wilker: `w.aziz@uva.nl`

Email Samira by Thursday, 5 April with details of your group.

- names of the students
- their email addresses
- subject: ULL group assignment

Natural Language Processing

Many popular applications



- Information retrieval
- Machine translation
- Question answering
- Dialogue systems
- Sentiment analysis
- Recently: fact checking etc.

Why is NLP difficult?

Similar strings mean different things, different strings mean the same thing.

- Synonymy: different strings can mean the same thing

  The King's speech *gave the much needed reassurance to his people.*
  His majesty's address *reassured the crowds.*

- Ambiguity: same strings can mean different things

  His majesty's address *reassured the crowds.*
  His majesty's address *is Buckingham Palace, London SW1A 1AA.*

Why is NLP difficult?

Similar strings mean different things, different strings mean the same thing.

- Synonymy: different strings can mean the same thing

  The King's speech *gave the much needed reassurance to his people.*
  His majesty's address *reassured the crowds.*

- Ambiguity: same strings can mean different things

  His majesty's address *reassured the crowds.*
  His majesty's address *is Buckingham Palace, London SW1A 1AA.*

Why is NLP difficult?

Similar strings mean different things, different strings mean the same thing.

- Synonymy: different strings can mean the same thing

  The King's speech *gave the much needed reassurance to his people.*
  His majesty's address *reassured the crowds.*

- Ambiguity: same strings can mean different things

  His majesty's address *reassured the crowds.*
  His majesty's address *is Buckingham Palace, London SW1A 1AA.*

Why is NLP difficult?

Similar strings mean different things, different strings mean the same thing.

- Synonymy: different strings can mean the same thing

  The King's speech *gave the much needed reassurance to his people.*
  His majesty's address *reassured the crowds.*

- Ambiguity: same strings can mean different things

  His majesty's address *reassured the crowds.*
  His majesty's address *is Buckingham Palace, London SW1A 1AA.*

Why is NLP difficult?

Similar strings mean different things, different strings mean the same thing.

- Synonymy: different strings can mean the same thing

  The King's speech *gave the much needed reassurance to his people.*
  His majesty's address *reassured the crowds.*

- Ambiguity: same strings can mean different things

  His majesty's address *reassured the crowds.*
  His majesty's address *is Buckingham Palace, London SW1A 1AA.*

Wouldn't it be better if . . . ?

The properties which make natural language difficult to process are essential to human communication:

- Flexible
- Learnable, but expressive and compact
- Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- Ambiguity is mostly local (for humans)
- resolved by immediate context
- but requires world knowledge

Wouldn't it be better if . . . ?

The properties which make natural language difficult to process are essential to human communication:

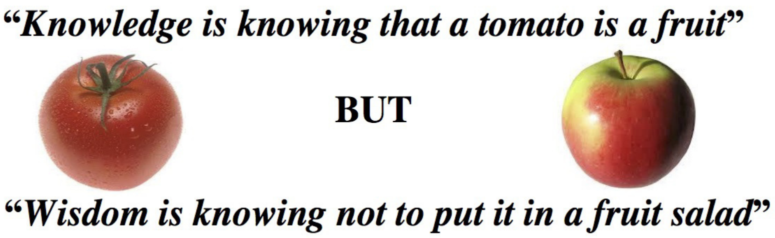
- Flexible
- Learnable, but expressive and compact
- Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- Ambiguity is mostly local (for humans)
- resolved by immediate context
- but requires world knowledge

World knowledge...



- Impossible to hand-code at a large-scale
- *either* limited domain applications
- *or* learn approximations from the data

Distributional hypothesis

You shall know a word by the company it keeps (Firth)

The meaning of a word is defined by the way it is used (Wittgenstein).

it was authentic scrumpy, rather sharp and very strong

we could taste a famous local product — scrumpy

spending hours in the pub drinking scrumpy

Cornish Scrumpy Medium Dry. £19.28 - Case

Distributional hypothesis

You shall know a word by the company it keeps (Firth)

The meaning of a word is defined by the way it is used (Wittgenstein).

it was authentic scrumpy, rather sharp and very strong

we could taste a famous local product — scrumpy

spending hours in the pub drinking scrumpy

Cornish Scrumpy Medium Dry. £19.28 - Case

Distributional hypothesis

You shall know a word by the company it keeps (Firth)

The meaning of a word is defined by the way it is used (Wittgenstein).

it was authentic scrumpy, rather sharp and very strong

we could taste a famous local product — scrumpy

spending hours in the pub drinking scrumpy

Cornish Scrumpy Medium Dry. £19.28 - Case

Distributional hypothesis

You shall know a word by the company it keeps (Firth)

The meaning of a word is defined by the way it is used (Wittgenstein).

it was authentic scrumpy, rather sharp and very strong

we could taste a famous local product — scrumpy

spending hours in the pub drinking scrumpy

Cornish Scrumpy Medium Dry. £19.28 - Case

Distributional hypothesis

You shall know a word by the company it keeps (Firth)

The meaning of a word is defined by the way it is used (Wittgenstein).

it was authentic scrumpy, rather sharp and very strong

we could taste a famous local product — scrumpy

spending hours in the pub drinking scrumpy

Cornish Scrumpy Medium Dry. £19.28 - Case

Scrumpy


WESTONS
CIDERS INSPIRED BY TRADITION
SCRUMPY
CLOUDY CIDER
Alc 7.5% Vol

Distributional hypothesis

This leads to the distributional hypothesis about word meaning:

- the context surrounding a given word provides information about its meaning;
- words are similar if they share similar linguistic contexts;
- semantic similarity $\approx$ distributional similarity.

Distributional semantics

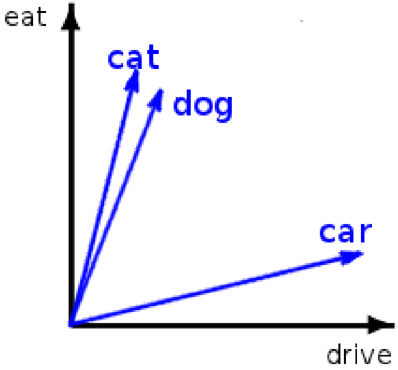
Distributional semantics: family of techniques for representing word meaning based on (linguistic) contexts of use.

1. Count-based models:
   - Vector space models
   - dimensions correspond to elements in the context
   - words are represented as vectors, or higher-order tensors
2. Prediction models:
   - Train a model to predict plausible contexts for a word
   - learn word representations in the process

Count-based approaches: the general intuition

- The **semantic space** has dimensions which correspond to possible contexts – features.
- For our purposes, a distribution can be seen as a point in that space (the vector being defined with respect to the origin of that space).
- *scrumpy* [...pub 0.8, drink 0.7, strong 0.4, joke 0.2, mansion 0.02, zebra 0.1...]

Vectors



Feature matrix

	feature$_1$	feature$_2$	...	feature$_n$
word$_1$	$f_{1,1}$	$f_{2,1}$		$f_{n,1}$
word$_2$	$f_{1,2}$	$f_{2,2}$		$f_{n,2}$
...				
word$_m$	$f_{1,m}$	$f_{2,m}$		$f_{n,m}$

The notion of context

1 Word windows (unfiltered): *n* words on either side of the lexical item.
Example: n=2 (5 words window):

*| The prime **minister** acknowledged the | question.*

minister [the 2, prime 1, acknowledged 1, question 0]

Context

2 Word windows (filtered): *n* words on either side removing some words (e.g. function words, some very frequent content words). Stop-list or by POS-tag.
Example: n=2 (5 words window), stop-list:

| *The prime **minister** acknowledged the* | *question.*

minister [prime 1, acknowledged 1, question 0]

Context

3 Lexeme window (filtered or unfiltered); as above but using stems.
Example: n=2 (5 words window), stop-list:

> *| The prime **minister** acknowledged the | question.*

minister [prime 1, acknowledge 1, question 0]

Context

4 Dependencies (directed links between heads and dependents). Context for a lexical item is the dependency structure it belongs to (various definitions).

Example:

*The prime **minister** acknowledged the question.*

minister [prime_a 1, acknowledge_v 1]

minister [prime_a_mod 1, acknowledge_v_subj 1]

minister [prime_a 1, acknowledge_v+question_n 1]

Parsed vs unparsed data: examples

word (unparsed)	**word (parsed)**
meaning_n	or_c+phrase_n
derive_v	and_c+phrase_n
dictionary_n	syllable_n+of_p
pronounce_v	play_n+on_p
phrase_n	etymology_n+of_p
latin_j	portmanteau_n+of_p
ipa_n	and_c+deed_n
verb_n	meaning_n+of_p
mean_v	from_p+language_n
hebrew_n	pron_rel_+utter_v
usage_n	for_p+word_n
literally_r	in_p+sentence_n

Dependency vectors

word (Subj)	**word (Dobj)**
come_v	use_v
mean_v	say_v
go_v	hear_v
speak_v	take_v
make_v	speak_v
say_v	find_v
seem_v	get_v
follow_v	remember_v
give_v	read_v
describe_v	write_v
get_v	utter_v
appear_v	know_v
begin_v	understand_v
sound_v	believe_v
occur_v	choose_v

Context weighting

- Binary model: if context *c* co-occurs with word *w*, value of vector $\vec{w}$ for dimension *c* is 1, 0 otherwise.

  *... [a long long long **example** for a distributional semantics] model... (n=4)*

  ... {a 1} {dog 0} {long 1} {sell 0} {semantics 1}...
- Basic frequency model: the value of vector $\vec{w}$ for dimension *c* is the number of times that *c* co-occurs with *w*.

  *... [a long long long **example** for a distributional semantics] model... (n=4)*

  ... {a 2} {dog 0} {long 3} {sell 0} {semantics 1}...

Characteristic model

- Weights given to the vector components express how *characteristic* a given context is for word w.
- Pointwise Mutual Information (PMI)

$$PMI(w,c) = \log \frac{P(w,c)}{P(w)P(c)} = \log \frac{P(w)P(c|w)}{P(w)P(c)} = \log \frac{P(c|w)}{P(c)}$$

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

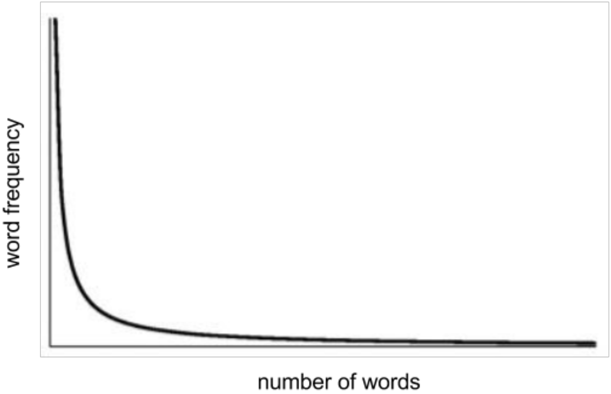
$$PMI(w,c) = \log \frac{f(w,c)\sum_k f(c_k)}{f(w)f(c)}$$

$f(w,c)$: frequency of word w in context c
$f(w)$: frequency of word w in all contexts
$f(c)$: frequency of context c

What semantic space?

- Entire vocabulary.
  - + All information included – even rare contexts
  - - Inefficient (100,000s dimensions). Noisy (e.g. *002.png|thumb|right|200px|graph_n*). Sparse
- Top *n* words with highest frequencies.
  - + More efficient (2000-10000 dimensions). Only 'real' words included.
  - - May miss out on infrequent but relevant contexts.

Word frequency: Zipfian distribution



What semantic space?

- Entire vocabulary.
  - + All information included – even rare contexts
  - - Inefficient (100,000s dimensions). Noisy (e.g. *002.png|thumb|right|200px|graph_n*). Sparse.
- Top *n* words with highest frequencies.
  - + More efficient (2000-10000 dimensions). Only 'real' words included.
  - - May miss out on infrequent but relevant contexts.

What semantic space?

- Singular Value Decomposition (LSA): the number of dimensions is reduced by exploiting redundancies in the data.
  - + Very efficient (200-500 dimensions). Captures generalisations in the data.
  - - SVD matrices are not interpretable.

Experimental corpus

- Dump of entire English Wikipedia, parsed with the English Resource Grammar producing dependencies.
- Dependencies include:
  - For nouns: head verbs (+ any other argument of the verb), modifying adjectives, head prepositions (+ any other argument of the preposition).
    e.g. cat: chase_v+mouse_n, black_a, of_p+neighbour_n
  - For verbs: arguments (NPs and PPs), adverbial modifiers.
    e.g. eat: cat_n+mouse_n, in_p+kitchen_n, fast_a
  - For adjectives: modified nouns; head prepositions (+ any other argument of the preposition)
    e.g. black: cat_n, at_p+dog_n

System description

- Semantic space: top 100,000 contexts.
- Weighting: normalised PMI (Bouma 2007).

An example noun

- *language*:

0.54::other+than_p()+English_n
0.53::English_n+as_p()
0.52::English_n+be_v
0.49::english_a
0.48::and_c+literature_n
0.48::people_n+speak_v
0.47::French_n+be_v
0.46::Spanish_n+be_v
0.46::and_c+dialects_n
0.45::grammar_n+of_p()
0.45::foreign_a
0.45::germanic_a
0.44::German_n+be_v

0.44::of_p()+instruction_n
0.44::speaker_n+of_p()
0.42::pron_rel_+speak_v
0.42::colon_v+English_n
0.42::be_v+English_n
0.42::language_n+be_v
0.42::and_c+culture_n
0.41::arabic_a
0.41::dialects_n+of_p()
0.40::percent_n+speak_v
0.39::spanish_a
0.39::welsh_a
0.39::tonal_a

An example adjective

- *academic*:

0.52::Decathlon_n
0.51::excellence_n
0.45::dishonesty_n
0.45::rigor_n
0.43::achievement_n
0.42::discipline_n
0.40::vice_president_n+for_p()
0.39::institution_n
0.39::credentials_n
0.38::journal_n
0.37::journal_n+be_v
0.37::vocational_a
0.37::student_n+achieve_v
0.36::athletic_a
0.36::reputation_n+for_p()
0.35::regalia_n
0.35::program_n
0.35::freedom_n
0.35::student_n+with_p()
0.35::curriculum_n
0.34::standard_n
0.34::at_p()+institution_n
0.34::career_n
0.34::Career_n
0.33::dress_n
0.33::scholarship_n
0.33::prepare_v+student_n
0.33::qualification_n

Corpus choice

- As much data as possible?
  - British National Corpus (BNC): 100 m words
  - Wikipedia: 897 m words
  - UKWac: 2 bn words
  - ...
- In general preferable, *but*:
  - More data is not necessarily the data you want.
  - More data is not necessarily realistic from a psycholinguistic point of view. We perhaps encounter 50,000 words a day. BNC = 5 years' text exposure.

Data sparsity

- Distribution for *unicycle*, as obtained from Wikipedia.

0.45::motorized_a
0.40::pron_rel_+ride_v
0.24::for_p()+entertainment_n
0.24::half_n+be_v
0.24::unwieldy_a
0.23::earn_v+point_n
0.22::pron_rel_+crash_v
0.19::man_n+on_p()
0.19::on_p()+stage_n
0.19::position_n+on_p()

0.17::slip_v
0.16::and_c+1_n
0.16::autonomous_a
0.16::balance_v
0.13::tall_a
0.12::fast_a
0.11::red_a
0.07::come_v
0.06::high_a

Polysemy

- Distribution for *pot*, as obtained from Wikipedia.

0.57::melt_v
0.44::pron_rel_+smoke_v
0.43::of_p()+gold_n
0.41::porous_a
0.40::of_p()+tea_n
0.39::player_n+win_v
0.39::money_n+in_p()
0.38::of_p()+coffee_n
0.33::amount_n+in_p()
0.33::ceramic_a
0.33::hot_a

0.32::boil_v
0.31::bowl_n+and_c
0.31::ingredient_n+in_p()
0.30::plant_n+in_p()
0.30::simmer_v
0.29::pot_n+and_c
0.28::bottom_n+of_p()
0.28::of_p()+flower_n
0.28::of_p()+water_n
0.28::food_n+in_p()

Polysemy

- Some researchers incorporate word sense disambiguation techniques.
- But most assume a single space for each word: can perhaps think of subspaces corresponding to senses.
- Graded rather than absolute notion of polysemy.

Idiomatic expressions

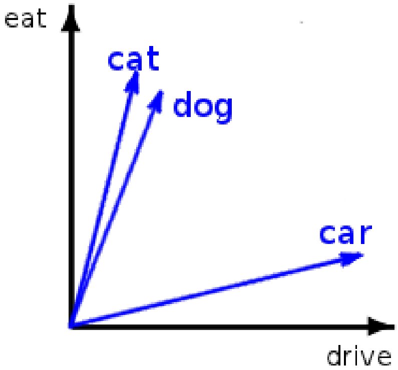
- Distribution for *time*, as obtained from Wikipedia.

0.46::of_p()+death_n
0.45::same_a
0.45::1_n+at_p(temp)
0.45::Nick_n+of_p()
0.42::spare_a
0.42::playoffs_n+for_p()
0.42::of_p()+retirement_n
0.41::of_p()+release_n
0.40::pron_rel_+spend_v
0.39::sand_n+of_p()
0.39::pron_rel_+waste_v

0.38::place_n+around_p()
0.38::of_p()+arrival_n
0.38::of_p()+completion_n
0.37::after_p()+time_n
0.37::of_p()+arrest_n
0.37::country_n+at_p()
0.37::age_n+at_p()
0.37::space_n+and_c
0.37::in_p()+career_n
0.37::world_n+at_p()

Calculating similarity in a distributional space

- Distributions are vectors, so distance can be calculated.



Measuring similarity

- Cosine:

$$cos(\theta) = \frac{\sum v1_k * v2_k}{\sqrt{\sum v1_k^2} * \sqrt{\sum v2_k^2}} \tag{1}$$

- The cosine measure calculates the angle between two vectors and is therefore length-independent. This is important, as frequent words have longer vectors than less frequent ones.
- Other measures include Jaccard, Euclidean distance etc.

The scale of similarity: some examples

house – building 0.43
gem – jewel 0.31
capitalism – communism 0.29
motorcycle – bike 0.29
test – exam 0.27
school – student 0.25
singer – academic 0.17
horse – farm 0.13
man –accident 0.09
tree – auction 0.02
cat –county 0.007

Words most similar to *cat*

as chosen from the 5000 most frequent nouns in Wikipedia.

1 cat	0.29 human	0.25 woman	0.22 monster
0.45 dog	0.29 goat	0.25 fish	0.22 people
0.36 animal	0.28 snake	0.24 squirrel	0.22 tiger
0.34 rat	0.28 bear	0.24 dragon	0.22 mammal
0.33 rabbit	0.28 man	0.24 frog	0.21 bat
0.33 pig	0.28 cow	0.23 baby	0.21 duck
0.31 monkey	0.26 fox	0.23 child	0.21 cattle
0.31 bird	0.26 girl	0.23 lion	0.21 dinosaur
0.30 horse	0.26 sheep	0.23 person	0.21 character
0.29 mouse	0.26 boy	0.23 pet	0.21 kid
0.29 wolf	0.26 elephant	0.23 lizard	0.21 turtle
0.29 creature	0.25 deer	0.23 chicken	0.20 robot

But what is similarity?

- In distributional semantics, very broad notion: synonyms, near-synonyms, hyponyms, taxonomical siblings, antonyms, etc.
- Correlates with a psychological reality.
- Test via correlation with human judgments on a test set:
  - Miller & Charles (1991)
  - WordSim
  - MEN
  - SimLex

Miller & Charles 1991

3.92 automobile-car
3.84 journey-voyage
3.84 gem-jewel
3.76 boy-lad
3.7 coast-shore
3.61 asylum-madhouse
3.5 magician-wizard
3.42 midday-noon
3.11 furnace-stove
3.08 food-fruit
3.05 bird-cock
2.97 bird-crane
2.95 implement-tool
2.82 brother-monk
1.68 crane-implement
1.66 brother-lad
1.16 car-journey
1.1 monk-oracle
0.89 food-rooster
0.87 coast-hill
0.84 forest-graveyard
0.55 monk-slave
0.42 lad-wizard
0.42 coast-forest
0.13 cord-smile
0.11 glass-magician
0.08 rooster-voyage
0.08 noon-string

- Distributional systems, reported correlations 0.8 or more.

TOEFL synonym test

Test of English as a Foreign Language: task is to find the best match to a word:

Prompt:		levied
Choices:	(a)	imposed
	(b)	believed
	(c)	requested
	(d)	correlated
Solution:	(a)	imposed

- Non-native English speakers applying to college in US reported to average 65%
- Best corpus-based results are 100%

Distributional methods are a usage representation

- Distributions are a good conceptual representation if you believe that ‘the meaning of a word is given by its usage’.
- Corpus-dependent, culture-dependent, register-dependent.
  Example: similarity between *policeman* and *cop*: 0.23

Distribution for *policeman*

policeman

0.59::ball_n+poss_rel
0.48::and_c+civilian_n
0.42::soldier_n+and_c
0.41::and_c+soldier_n
0.38::secret_a
0.37::people_n+include_v
0.37::corrupt_a
0.36::uniformed_a
0.35::uniform_n+poss_rel
0.35::civilian_n+and_c
0.31::iraqi_a
0.31::lot_n+poss_rel
0.31::chechen_a
0.30::laugh_v
0.29::and_c+criminal_n

0.28::incompetent_a
0.28::pron_rel_+shoot_v
0.28::hat_n+poss_rel
0.28::terrorist_n+and_c
0.27::and_c+crowd_n
0.27::military_a
0.27::helmet_n+poss_rel
0.27::father_n+be_v
0.26::on_p()+duty_n
0.25::salary_n+poss_rel
0.25::on_p()+horseback_n
0.25::armed_a
0.24::and_c+nurse_n
0.24::job_n+as_p()
0.24::open_v+fire_n

Distribution for *cop*

cop
0.45::crooked_a
0.45::corrupt_a
0.44::maniac_a
0.38::dirty_a
0.37::honest_a
0.36::uniformed_a
0.35::tough_a
0.33::pron_rel_+call_v
0.32::funky_a
0.32::bad_a
0.29::veteran_a
0.29::and_c+robot_n
0.28::and_c+criminal_n
0.28::bogus_a
0.28::talk_v+to_p()+pron_rel_

0.27::investigate_v+murder_n
0.26::on_p()+force_n
0.25::parody_n+of_p()
0.25::Mason_n+and_c
0.25::pron_rel_+kill_v
0.25::racist_a
0.24::addicted_a
0.23::gritty_a
0.23::and_c+interference_n
0.23::arrive_v
0.23::and_c+detective_n
0.22::look_v+way_n
0.22::dead_a
0.22::pron_rel_+stab_v
0.21::pron_rel_+evade_v

The similarity of synonyms

- Similarity between *egglant/aubergine*: 0.11
  Relatively low cosine. Partly due to frequency (222 for *eggplant*, 56 for *aubergine*).
- Similarity between *policeman/cop*: 0.23
- Similarity between *city/town*: 0.73

In general, true synonymy does not correspond to higher similarity scores than near-synonymy.

Similarity of antonyms

- Similarities between:
  - cold/hot 0.29
  - dead/alive 0.24
  - large/small 0.68
  - colonel/general 0.33

Identifying antonyms

- Antonyms have high distributional similarity: hard to distinguish from near-synonyms purely by distributions.
- Identification by heuristics applied to pairs of highly similar distributions.
- For instance, antonyms are frequently coordinated while synonyms are not:
  - a selection of cold and hot drinks
  - wanted dead or alive

Distributions and knowledge

What kind of information do distributions encode?

- lexical knowledge
- world knowledge
- boundary between the two is blurry
- no perceptual knowledge

Distributions are partial lexical semantic representations, but useful and theoretically interesting.

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

truck lorry
bike
path
highway way
car
taxi
bicycle
street
road avenue
driver
house
mechanic lab building
shack
engineer
scientist
office flat
plumber
writer
dwelling
journalist proceedings
book
journal
newspaper
magazine

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
...

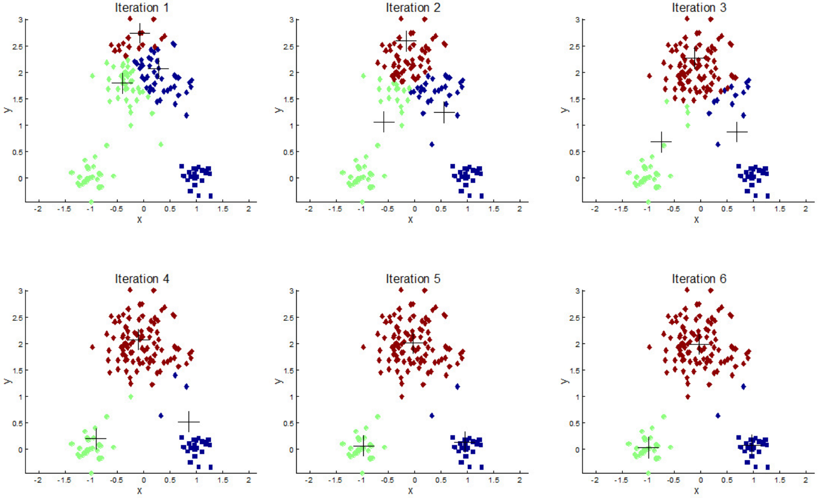
crop
78 grow_v_Subj
76 grow_v_Dobj
44 produce_v_Dobj
23 yield_v_Subj
16 harvest_v_Dobj
12 plant_v_Dobj
10 sow_v_Subj
10 ensure_v_Dobj
10 cut_v_Dobj
9 yield_v_Dobj
9 protect_v_Dobj
9 fail_v_Subj
9 destroy_v_Dobj
8 plant_v_Subj
7 spray_v_Subj
7 spray_v_Dobj
7 lose_v_Dobj
6 feed_v_Subj
...

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 μ_i:

$$\arg\min_C \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \quad (2)$$

K-means clustering

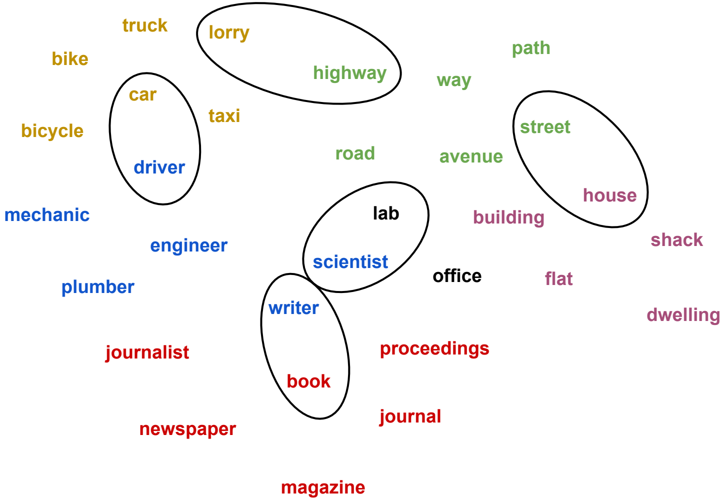


Noun clusters

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

Clustering nouns

Clustering nouns



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

Widely used in NLP as a source of lexical information:

- Word sense induction and disambiguation
- Modelling predicate-argument structure (e.g. semantic roles)
- Identifying figurative language and idioms
- Paraphrasing and paraphrase detection
- Used in applications directly, e.g. machine translation, information retrieval etc.