Distributional approaches to semantic analysis

Diarmuid Ó Séaghdha

Natural Language and Information Processing Group
Computer Laboratory
University of Cambridge
do242@cam.ac.uk

HIT-MSRA Summer Workshop on Human Language Technology
August 2011
http://www.cl.cam.ac.uk/~do242/Teaching/HIT-MSRA-2011/


UNIVERSITY OF CAMBRIDGE

Course structure

Part 1: **Introduction and overview**

Part 2: **Fundamentals of distributional methods**
Applications: estimating lexical similarity and relatedness

Part 3: **Dimensionality reduction and topic modelling**
Applications: text-to-text similarity, selectional preferences

Part 4: **Learning about entities and relations**
Applications: entity set expansion, taxonomy construction, relation extraction

The latest slides will always be at `http://www.cl.cam.ac.uk/~do242/Teaching/HIT-MSRA-2011/` (and shortly afterwards on the summer school website).

What is computational semantics?

- In linguistics, semantics is the study of meaning, or how the components of language (words and phrases) correspond to concepts in the communicator's mind.
- In Natural Language Processing, computational semantics is the study of automated methods for acquiring and using knowledge about linguistic meaning.
- Some fundamental questions we are interested in:
  - *Do two words have similar or dissimilar meanings?*
  - *What classes of entities are referred to in language?*
  - *What relations typically hold between these entities?*
  - *What relations hold between entities in a particular text?*
- Computational semantics sometimes overlaps with what researchers in other areas call "language modelling" or "information extraction".

Course focus

- The focus of this course is on "distributional" approaches to semantics, i.e. methods that extract semantic information from the way words behave in text corpora.
- We won't spend much time discussing methods that rely solely on manually constructed resources such as WordNet or HowNet.
- There will also be little discussion of structured prediction problems such as semantic role labelling or semantic parsing; these would require a lecture course of their own!

Measuring semantic similarity

- A fundamental task for semantic models is to predict how similar two words' meanings are.
- Why is this important?
  - Test of the quality of the semantic model
  - Answer questions about the possibility of emulating human behaviour with NLP techniques
  - Applications: similarity-based smoothing, spelling correction, query expansion, word clustering, entity set expansion, learning sentiment lexicons, paraphrasing...

Similarity and relatedness

- Similarity is often correlated with substitutability: if I replace word w in a sentence with w', to what degree is the meaning preserved?

cup

mug

- *Cups* and *mugs* are very similar in terms of appearance and function.

Similarity and relatedness

- Relatedness is a less strict condition than similarity:

cup

tea

- *Cups* and *tea* have very different appearance and function, but their functions are related.

Evaluating models of semantics

- Assume we have built some computational model of meaning; how can we tell whether our model is a good one? How do we compare it to an alternative model?
- We could test the ability of our model to "recreate" real-life text data. This is a standard approach in language modelling, but some interesting studies have shown that it is not necessarily predictive of semantic quality (Chang et al., 2009).
- One idea is to test whether an existing application (with a well-defined evaluation method) is improved by incorporating semantic knowledge provided by our model. For example, can we improve parsing or machine translation?
- Alternatively, we can trust humans as experts on semantics; we collect semantic judgements from human judges and compare our model's predictions to those judgements.

Similarity-based smoothing

- Many NLP systems learn from patterns in annotated or unannotated text corpora.
- These patterns can be very sparse: bilexical dependencies in parsing, higher-order n-grams, even unigram features in small annotated datasets.
- One approach is to smooth the estimate for a pattern with those for semantically similar patterns, on the assumption that they will have similar behaviour with respect to the task at hand, e.g.:

$$f(w) = \sum_{w' \in W} sim(w, w') f(w) \tag{1}$$

Human judgements of similarity

- Rubenstein and Goodenough (1965) collected similarity ratings for 65 pairs of nouns using a scale 0-4:

automobile	*car*	*3.92*
magician	*wizard*	*3.21*
car	*journey*	*1.55*
automobile	*wizard*	*0.11*

- If we build a computational system that predicts similarity between words, we can evaluate it by measuring the correlation between its predictions and human judgements.

Your assignment (Part I)

- The Rubenstein and Goodenough pairs give us a set of similarity judgements for English.
- We are going to collect a similar set of judgements for Chinese.
- Please download the file at this URL:
  `http://www.cl.cam.ac.uk/~do242/Teaching/HIT-MSRA-2011/harbin_chinese_pairs.txt`
- You will see a list of word pairs; for each pair, decide how similar the two words are on a scale of 0-4, where 0 means not similar at all and 4 means identical, and enter your decision on the same line.
- Send your completed lists by email to `do242@cam.ac.uk`; include "Chinese word pairs" in the subject line.

Fundamentals of distributional semantics

The distributional hypothesis

The word space model

Association measures

Latent Semantic Analysis

Large data semantics

What is *tezgüino*?

- Imagine that *tezgüino* is a rare English word, and you saw the word used in the following sentences:
  1. A bottle of *tezgüino* is on the table.
  2. Everyone likes *tezgüino*.
  3. *Tezgüino* makes you drunk.
  4. We make *tezgüino* out of corn.

(Lin, 1998a)

- Can you guess what *tezgüino* means?
- What kind of things do you expect will be similar to *tezgüino*?

The distributional hypothesis

- Two words are expected to be semantically similar if they have similar co-occurrence behaviour in observed text.
- Harris (1954): "If we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C."
- Frith (1957): "You shall know a word by the company it keeps."
- This principle is known as the *distributional hypothesis*.
- In order to apply this hypothesis, we must specify what we mean by "co-occurrence behaviour" and how to measure "similar co-occurrence behaviour".

Words = vectors

- A very popular framework for lexical semantics is the *vector space model*. Essentially, we define a feature mapping $\phi : V \rightarrow \mathbb{R}^k$ from vocabulary items to vectors.
- Directly inspired by the vector space model of documents in Information Retrieval.
- Let V be the vocabulary of target terms, D be a corpus of documents and C be a set of context items.
- We associate each term w in our vocabulary V with a vector of real numbers $\mathbf{w} \in \mathbb{R}^k$, where each basis element of the vector space corresponds to a context item $c \in C$, so $k = |C|$.
- For now, we will assume that the value of the jth entry in the vector $\mathbf{w}_i$ is $w_{ij} = \mathit{frequency}(w_i, c_j)$ in the corpus D.

Context types

- Different types of context (or different feature mappings) induce different kinds of semantic spaces.
- Some important classes of context definitions:
  - Document context: The context for w consists of the document in which it appears.
  - Window context: The context for w consists of all words that are within n words to its left or right.
  - Syntactic context: The context for w consists of all words connected to w by a syntactic path.
- For maximal flexibility we define context items as pairs (r, x), where $r \in R$ is a relation and either $x \in V$ or $x \in D$. This allows us to make distinctions between, e.g., "w_1 is to the left of w_2" and "w_1 is to the right of w_2" or between "w_1 is the subject of w_2" and "w_1 is the direct object of w_2".

Know your corpus

- The nature and quality of any distributional model depends on the corpus from which it is learned.
- Corpora containing different registers and genres will produce different models: consider the uses of the word *mouse* in a text about computers and a text about biology.
- In order to focus the distributional model on semantically relevant information it is often useful to preprocess a corpus with one or more "cleaning steps", including tokenisation, lemmatisation, stopword removal and part-of-speech tagging. It may also be necessary to parse the corpus.
- There can a tradeoff between the size of the corpus and the amount of preprocessing that is feasible - it's not possible to parse the World Wide Web.

The "football" corpus

Document 1

I played soccer until I was 13. Mum was a bit nervous about letting me and my twin brother play rugby.

Document 2

Soccer is played on a rectangular field of grass or green artificial turf, with a goal in the middle of each of the short ends. The object of the game is to score by driving the ball into the opposing goal.

Document 3

11 soccer players kick off against their 11 opponents.

Document 4

Rugby is based on running with the ball in hand. Rugby is played with an oval-shaped ball on a field up to 100 metres long and 70 metres wide with H-shaped goal posts on each goal line.

The "football" corpus - lemmatised

Document 1

i play soccer until i be 13 mum be a bit nervous about let i and my twin brother play rugby

Document 2

soccer be play on a rectangular field of grass or green artificial turf with a goal in the middle of each of the short end the object of the game be to score by drive the ball into the opposing goal

Document 3

11 soccer player kick off against their 11 opponent

Document 4

rugby be base on running with the ball in hand rugby be play with an oval-shaped ball on a field up to 100 metre long and 70 metre wide with H-shaped goal post on each goal line

The "football" corpus - stopwords removed

Document 1

i play soccer until i be 13 mum be a bit nervous about let i and my twin brother play rugby

Document 2

soccer be play on a rectangular field of grass or green artificial turf with a goal in the middle of each of the short end the object of the game be to score by drive the ball into the opposing goal

Document 3

11 soccer player kick off against their 11 opponent

Document 4

rugby be base on running with the ball in hand rugby be play with an oval-shaped ball on a field up to 100 metre long and 70 metre wide with H-shaped goal post on each goal line

Document context

- Only one relation type: w_{ij} counts the number of times word w_i occurs in document d_j.

	Doc 1	Doc 2	Doc 3	Doc 4
soccer	1	1	1	0
rugby	1	0	0	1
ball	0	1	0	1
play	2	1	0	1
player	0	0	1	0
field	0	1	0	1
goal	0	2	0	2
brother	1	0	0	0
⋮				

Window context

- In the basic model, only one relation type: w_{ij} counts the number of times word w_i occurs within an n-word "window" of word w_j.

  soccer be play on a rectangular field of grass or green artificial turf with a **goal** *in the middle of each of the short end*

- Narrower windows tend to highlight similarity, wider windows favour semantic relatedness.
- It is possible to expand the set of relation types by taking into account positional information, e.g. $R = \{\text{left-of}, \text{right-of}\}$.

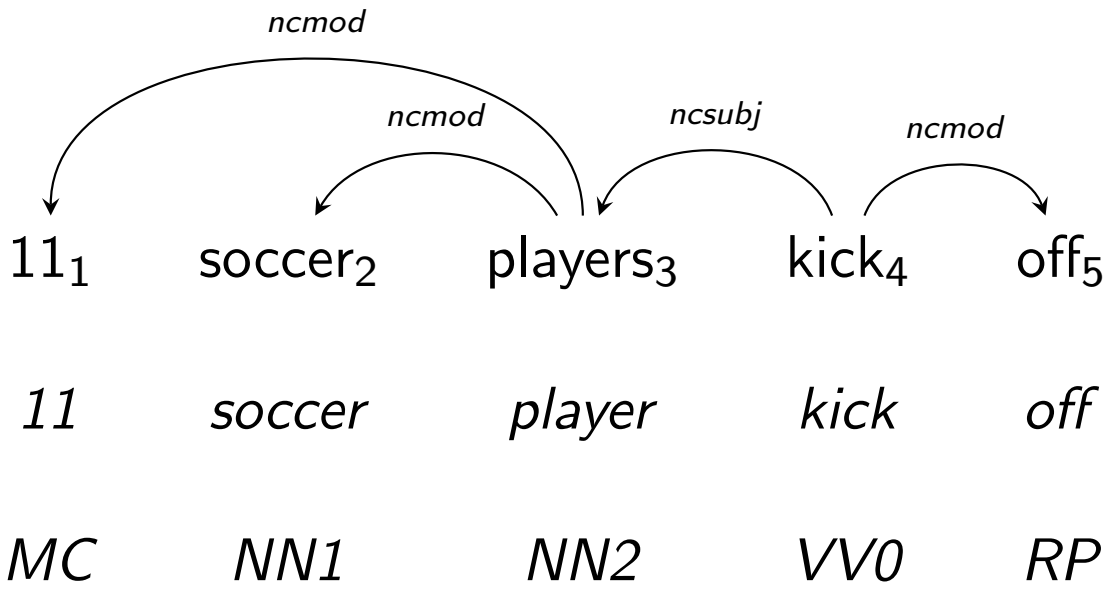
Window context

	soccer	rugby	ball	play	player	field	goal	...
soccer	0	1	0	2	1	0	0	
rugby	1	0	0	1	0	0	0	
ball	0	0	0	1	0	1	1	
play	2	1	1	0	0	1	0	
player	1	0	0	0	0	0	0	
field	0	0	1	1	0	0	0	
goal	0	0	1	0	0	0	0	
brother	0	1	0	0	0	0	0	
⋮								

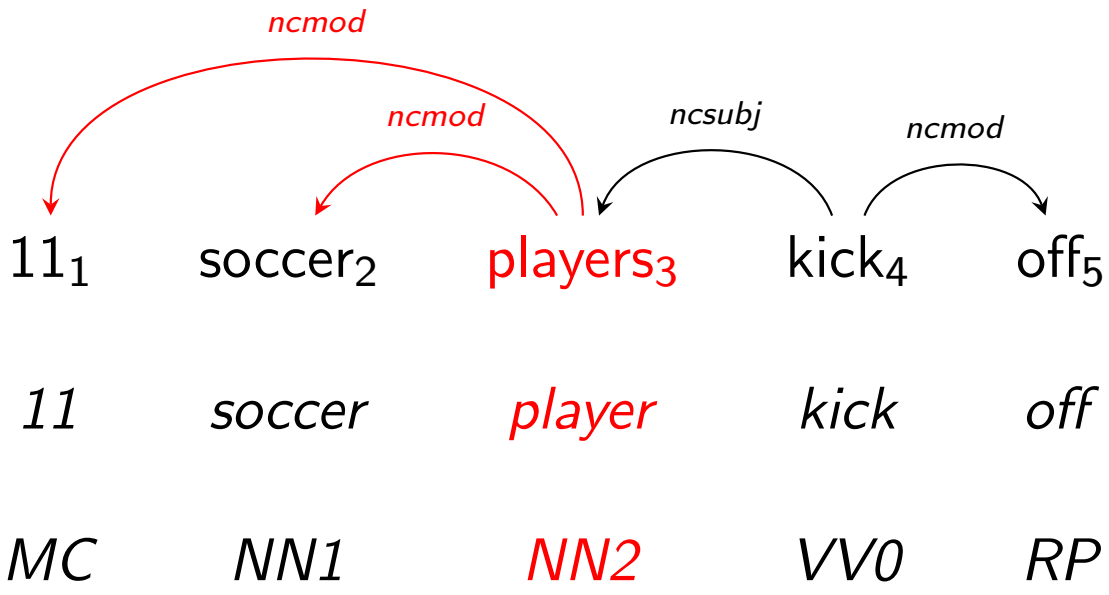
Syntactic context

- When we use syntactic context, each context type (r, w_j) corresponds to the pairing of a syntactic relation r and a word w_j.
- Syntactic word space models require a parsed corpus as input. Typically a parser is chosen that produces (labelled) dependency output, as this is easier to use for semantic analysis than constituency tree output.
- Popular dependency formats include RASP (Briscoe et al., 2006) (used in these lectures), MINIPAR (Lin, 1998b) and the Stanford format (de Marneffe et al., 2006).
- If, for example, we adopt the RASP format, R may be the set of dependency labels $\{$ncsubj, dobj, iobj, ncmod,$\ldots\}$ or a subset of all such labels.
- Not all dependencies are useful; we may want to ignore determiner and punctuation relations.

Syntactic context



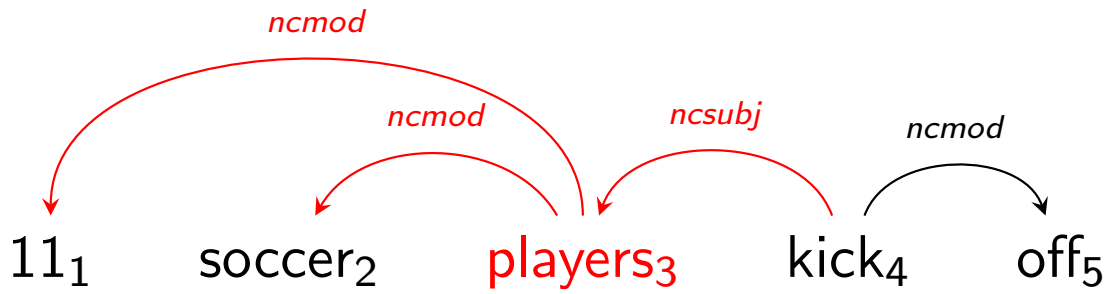
Syntactic context



Context set of *players*:

$C_3 = \{(ncmod, 11), (ncmod, soccer)\}$

Syntactic context

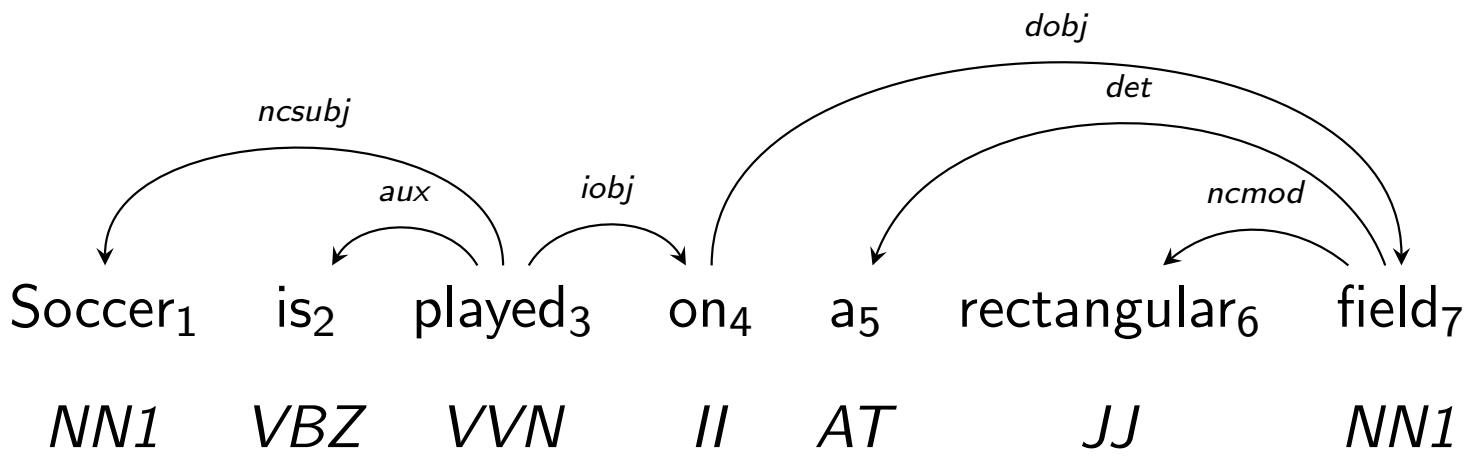


Context set of *players*:

$C_3 = \{(ncmod, 11),$
$(ncmod, soccer),$
$(ncsubj^{-1}, kick)\}$

Syntactic context

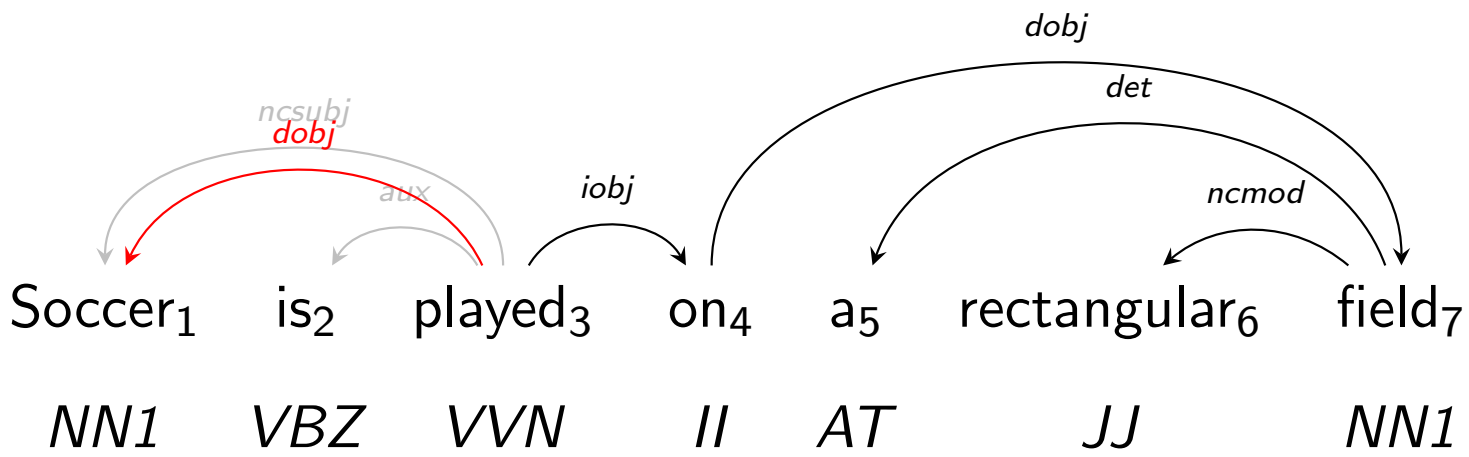
- It can be useful to postprocess parsed sentences with parser- and language-specific rules that add or replace edges:



Syntactic context

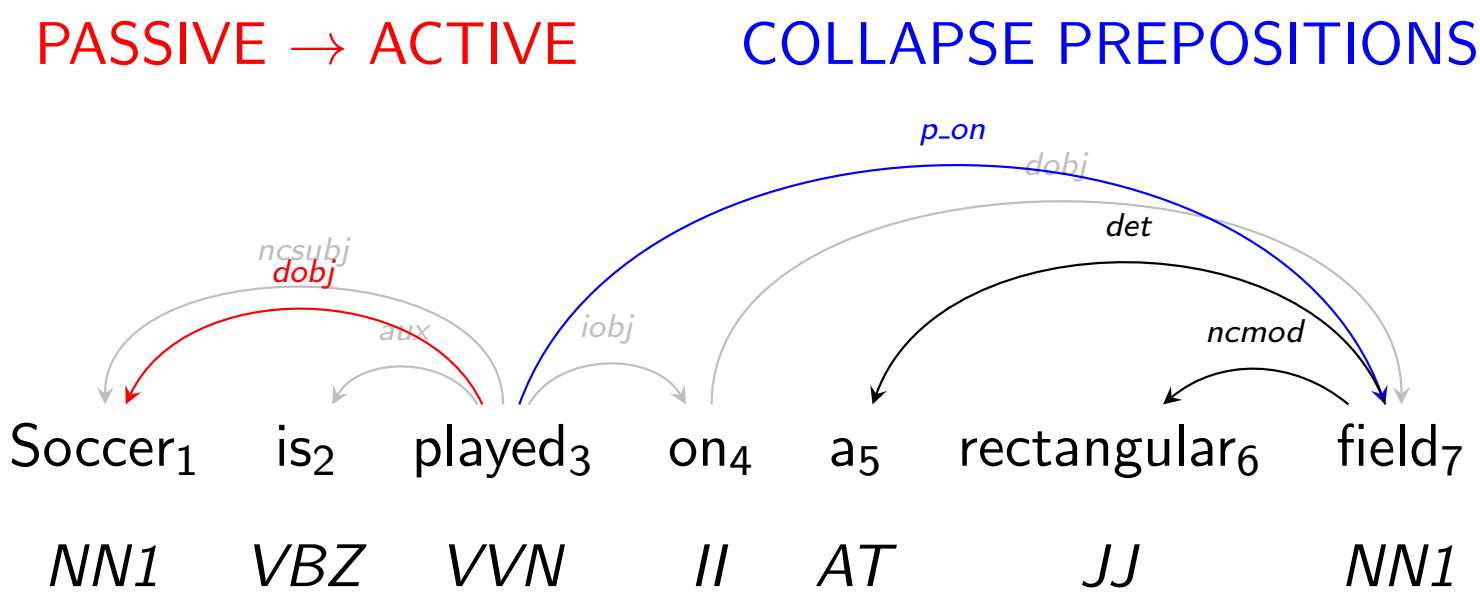
- It can be useful to postprocess parsed sentences with parser- and language-specific rules that add or replace edges:

PASSIVE → ACTIVE



Syntactic context

- It can be useful to postprocess parsed sentences with parser- and language-specific rules that add or replace edges:



Syntactic context

	dobj soccer	dobj rugby	dobj^{-1} play	p_with ball	p_with^{-1} play	p_on field	p_on^{-1} play	...
soccer	0	0	2	0	0	0	0	
rugby	0	0	2	0	0	0	0	
ball	0	0	0	0	1	0	0	
play	2	2	0	1	0	1	0	
player	0	0	0	0	0	0	0	
field	0	0	0	0	0	0	2	
goal	0	0	0	0	0	0	0	
brother	0	0	0	0	0	0	0	
⋮								

Similarity and distance

- Assumption: proximity in word space correlates with similarity in meaning
- Cosine similarity is a standard way of computing closeness between vectors:

$$\text{Cosine}(\mathbf{v_1}, \mathbf{v_2}) = \frac{\mathbf{v_1}{\mathbf{v_2}}^T}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|} \tag{2}$$

$$= \frac{\sum_i^k v_{1i} v_{2i}}{\sqrt{\sum_{i=1}^k v_{1i}^2}\sqrt{\sum_{i=1}^k v_{2i}^2}} \tag{3}$$

- Equivalent to dot product of L_2-normalised vectors; not affected by magnitude.
- Cosine is 0 between orthogonal vectors, 1 if $v_1 = \alpha v_2, \alpha > 0$.

Similarity and distance

- A standard measure of distance (or dissimilarity) in $\mathbb{R}^k$ is the L_2 or Euclidean distance:

$$L_2(\mathbf{v_1}, \mathbf{v_2}) = \sqrt{\sum_{i=1}^{k}(v_{1i} - v_{2i})^2}$$

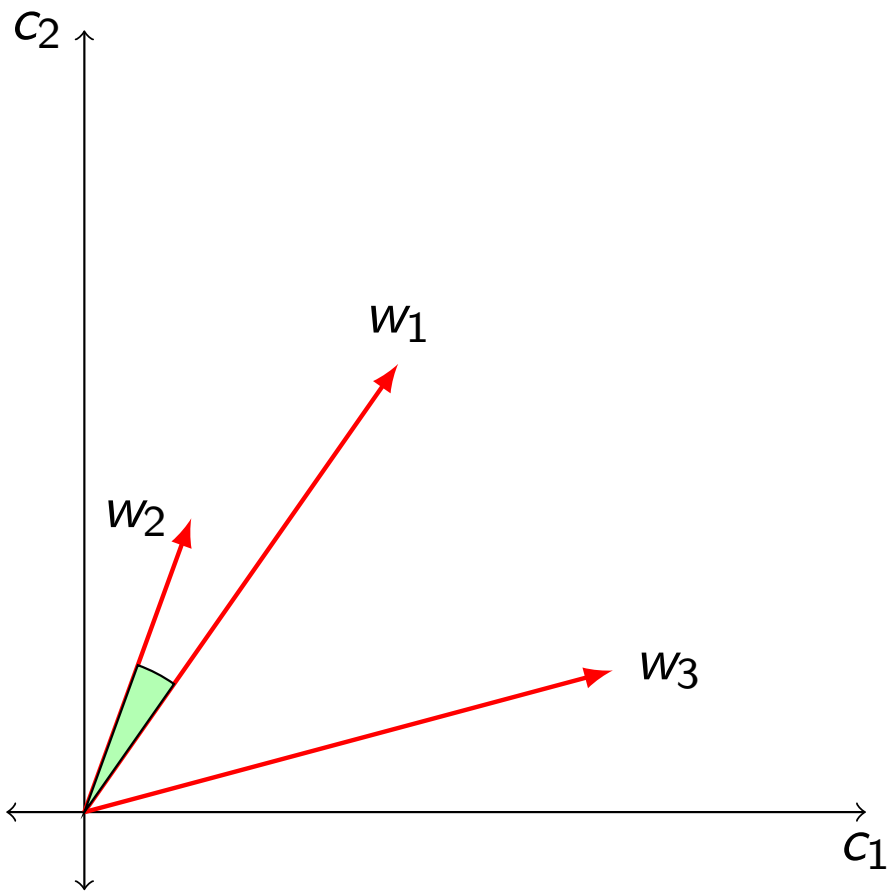
- Note that when $\|\mathbf{v_1}\| = 1, \|\mathbf{v_2}\| = 1$,

$$(L_2(\mathbf{v_1}, \mathbf{v_2}))^2 = 2 - 2 * \text{Cosine}(\mathbf{v_1}, \mathbf{v_2})$$

- Another similarity measure that can be derived from Euclidean distance is the *Gaussian RBF kernel* often used in Support Vector Machine classification:

$$RBF(\mathbf{v_1}, \mathbf{v_2}) = \exp(-\beta * (L_2(\mathbf{v_1}, \mathbf{v_2})^2))$$

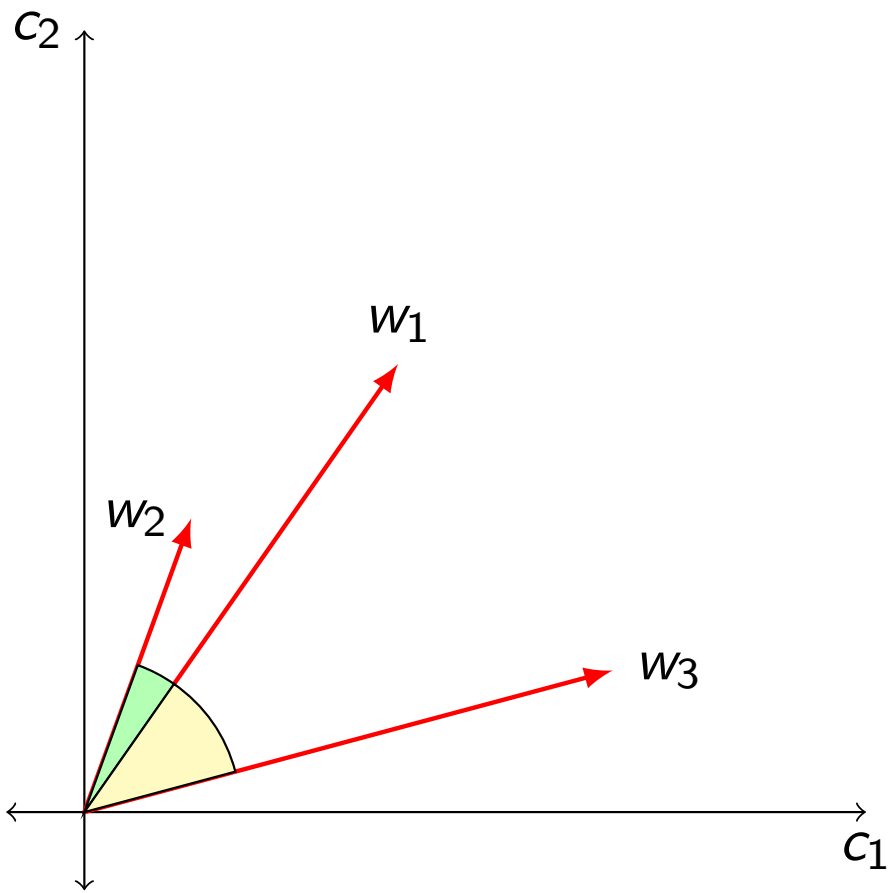
Cosine example



θ_{12} is the angle between the vectors w_1 and w_2.

$$\theta_{12} = 15^\circ$$
$$\cos(\theta_{12}) = 0.966$$

Cosine example



θ_{12} is the angle between the vectors w_1 and w_2.

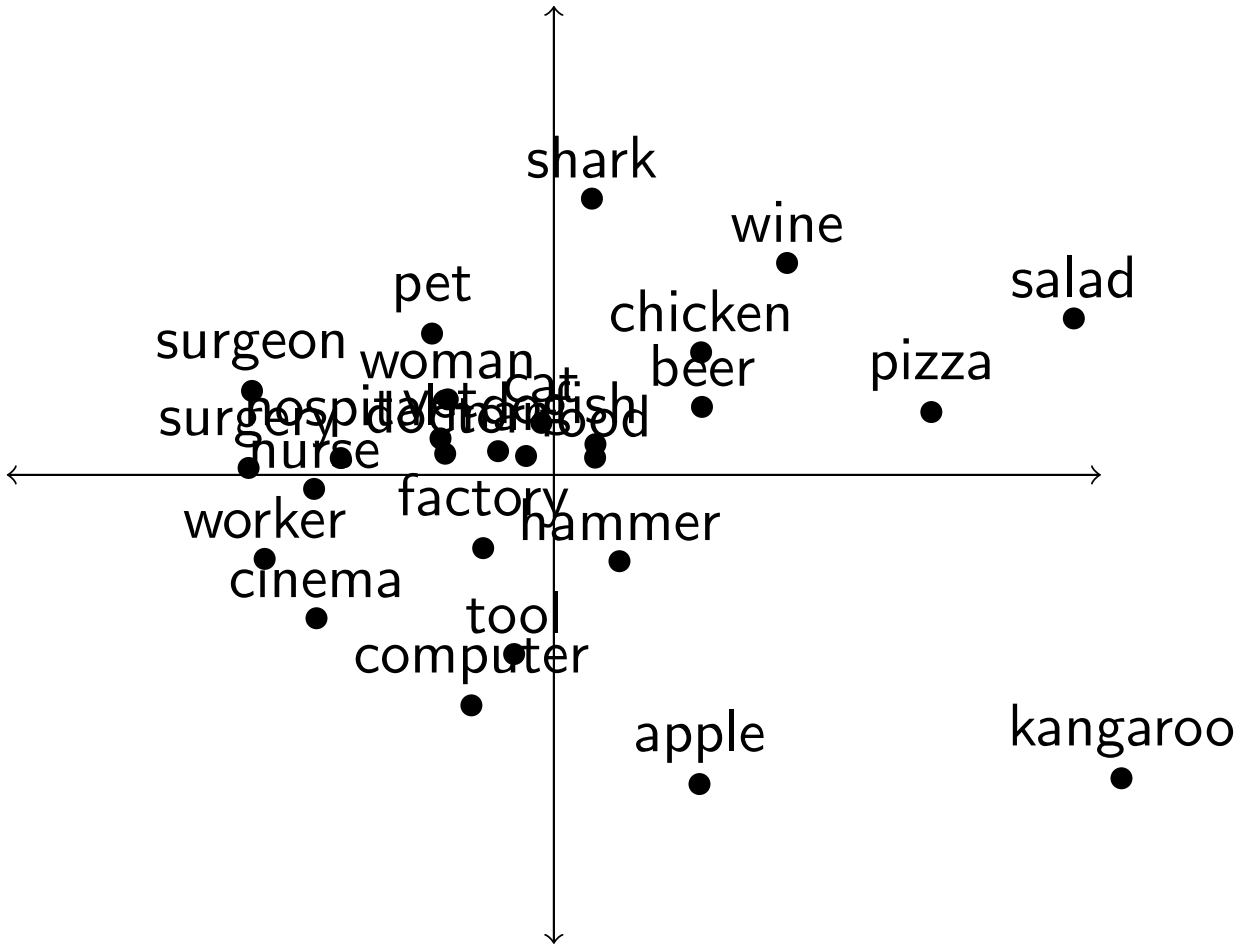
$$\theta_{12} = 15^\circ$$
$$\cos(\theta_{12}) = 0.966$$

θ_{13} is the angle between the vectors w_1 and w_3.

$$\theta_{13} = 40^\circ$$
$$\cos(\theta_{13}) = 0.766$$

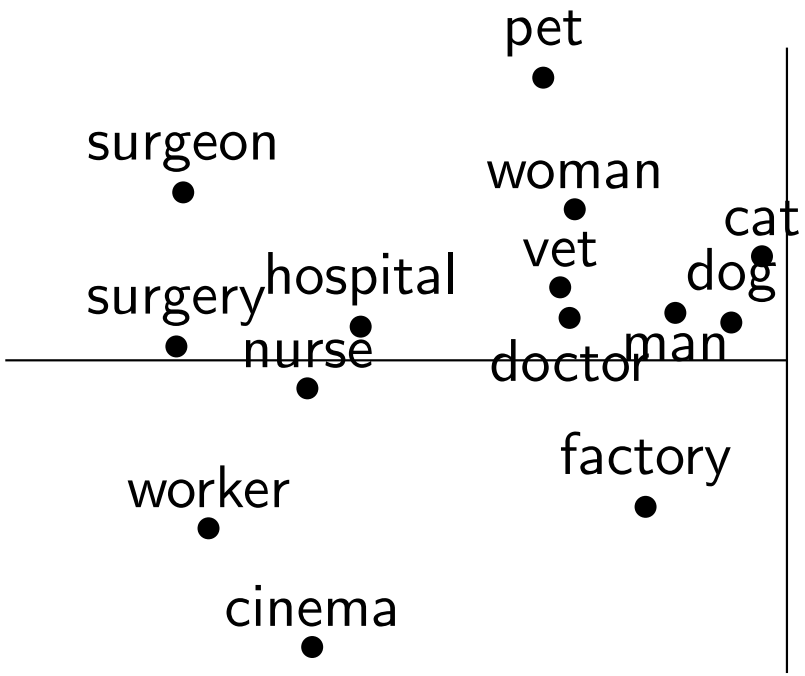
Window space visualisation

British National Corpus, Window size = 5, top 5000 context words



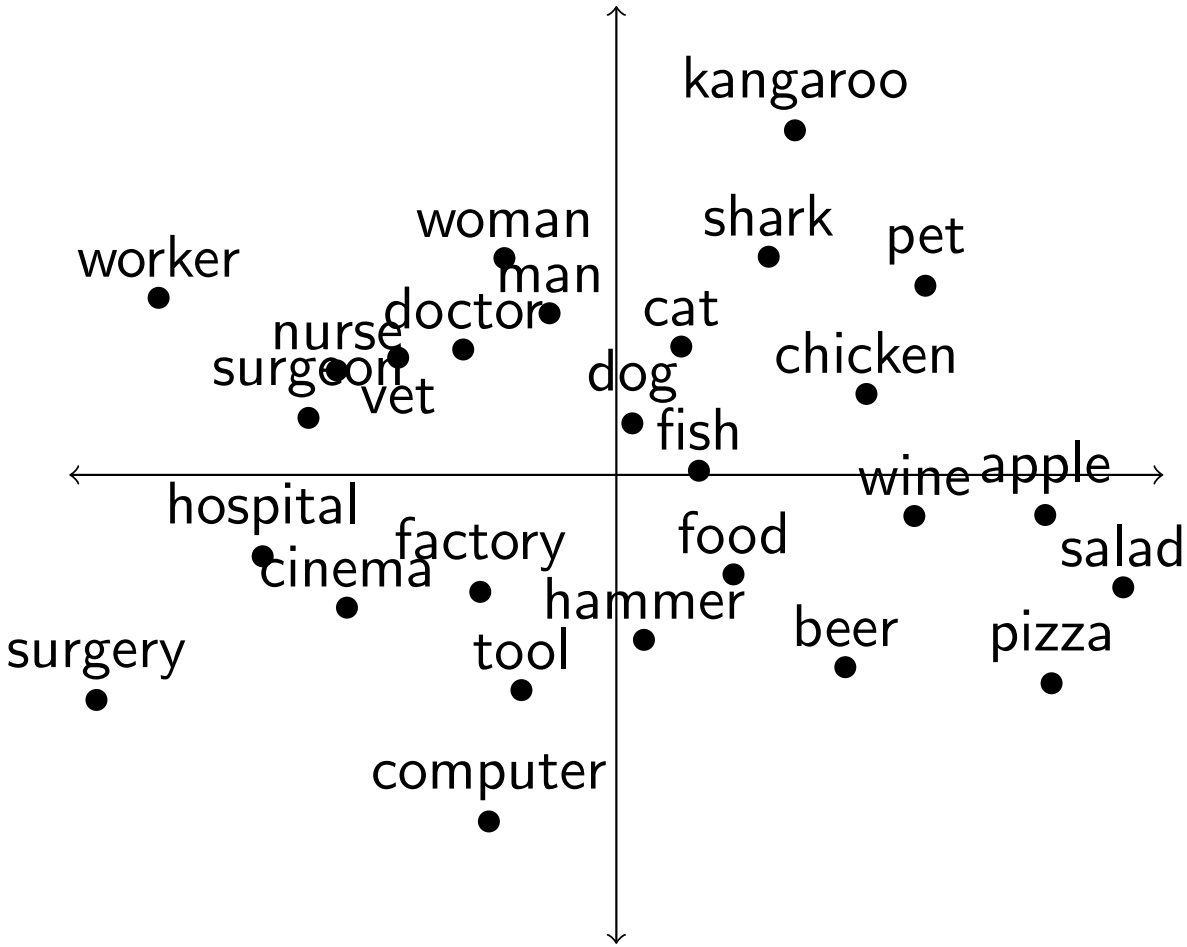
Window space visualisation

British National Corpus, Window size = 5, top 5000 context words



Syntactic space visualisation

British National Corpus, top 5000 dependencies



Halfway Summary

- We have introduced the field of computational semantics and considered some fundamental questions in this field.
- The distributional hypothesis gives us the theoretical basis for building semantic models.
- One of the most important semantic models is the vector space model of word meaning, which allows us to apply well-known techniques from linear algebra to model phenomena such as lexical similarity.
- We have seen how different ways of mapping words to vectors can affect the properties of distributional models.

Reweighting with association measures

- Using raw frequency information runs the risk of allowing frequent context types to dominate the vector comparison: very frequent context types may have high co-occurrence counts for every word in V.
- Association measures take into account the marginal frequencies of a word w and a context item c, as well as the corpus size N, to compute the statistical strength of the association between w and c.
- *How much higher/lower is the observed co-occurrence frequency of w and c than the frequency one would expect from the marginal frequencies of w and c?*

Association measures

Observed frequencies:

	y	$\neg y$
x	O_{11}	O_{12}
$\neg x$	O_{21}	O_{22}

$$O_{11} = f_{xy}$$
$$O_{12} = f_y - f_{xy}$$
$$O_{21} = f_x - f_{xy}$$
$$O_{22} = f_y + f_x - f_{xy}$$
$$N = \sum_{i,j} O_{ij}$$

Expected frequencies:

	y	$\neg y$
x	E_{11}	E_{12}
$\neg x$	E_{21}	E_{22}

$$E_{11} = \frac{f_x f_y}{N}$$
$$E_{12} = \frac{f_x f_y}{N}$$
$$E_{21} = \frac{f_x f_{\neg y}}{N}$$
$$E_{22} = \frac{f_{\neg x} f_{\neg y}}{N}$$

Some popular association measures

$$\text{PMI} = \log \frac{O_{11}}{E_{11}}$$

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

$$\text{t-score} = \frac{O_{11} - E_{11}}{\sqrt{O_{11}}}$$

$$\text{Log-likelihood} = 2 \sum_{i,j} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

- For a comprehensive account of association measures, see Evert (2004).

Association measure example

Top features (BNC, 5-word window):

Raw frequencies

farmer: *small, local, will, would, say, wife, do, . . .*

doctor: *say, see, do, will, nurse, patient, tell, . . .*

hospital: *general, take, patient, go, London, service, where, . . .*

Association measure example

Top features (BNC, 5-word window):

Raw frequencies

farmer: *small, local, will, would, say, wife, do, . . .*
doctor: *say, see, do, will, nurse, patient, tell, . . .*
hospital: *general, take, patient, go, London, service, where, . . .*

Transformed frequencies (t-score)

farmer: *part-time, sheep, peasant, tenant, wife, crop, . . .*
doctor: *nurse, junior, prescribe, consult, patient, surgery,. . .*
hospital: *psychiatric, memorial, discharge, admission, clinic, . . .*

Association measures as similarity measures

- Turney (2001) proposes a similarity measure *PMI-IR* based on the PMI association measure and co-occurrences obtained by submitting queries to Web search engines:

$$\text{PMI-IR}(w_1, w_2) = \log \frac{Hits(w_1 \wedge w_2)}{Hits(w_1) * Hits(w_2)}$$

  (note that we have lost the normalising term N compared to standard PMI)
- In principle, any association measure and source of co-occurrence frequencies can be used.
- PMI-IR measures the statistical dependency between the appearance of w_1 and the appearance of w_2 in a particular context; therefore it is more appropriate to describe it as a measure of semantic relatedness than one of semantic similarity.

Example results for measuring semantic similarity

- We train semantic space models on the British National Corpus (∼90 million words) and compare model predictions with the human judgements collected by Rubenstein and Goodenough (1965) using Spearman's rank correlation ρ.
- For each model, we ignore all co-occurrence frequencies $f_{ij} < 3$ and prune all but the 10,000 most frequent features.

Contexts	Raw frequencies	
5-word window	0.53	
dependencies	0.65	

- Using raw frequences, the window-based method is swamped by frequent non-discriminative terms.

Example results for measuring semantic similarity

- We train semantic space models on the British National Corpus (∼90 million words) and compare model predictions with the human judgements collected by Rubenstein and Goodenough (1965) using Spearman's rank correlation ρ.
- For each model, we ignore all co-occurrence frequencies $f_{ij} < 3$ and prune all but the 10,000 most frequent features.

Contexts	Raw frequencies	Transformed (t-test)
5-word window	0.53	0.68
dependencies	0.65	0.70

- Using raw frequences, the window-based method is swamped by frequent non-discriminative terms.
- The filtering effect of the t-test transformation allows the window-based contexts to come close to the performance of the consistently effective syntactic model.

Application: Text-to-text similarity

- How similar or related are these two texts?

Text 1

I travelled to Beijing by plane. It is a beautiful city with many attractions for tourists.

Text 2

I flew in to Beijing on Tuesday. The Chinese capital is really attractive, there's so much to do here and I hope to visit again.

- Note that they have few exact vocabulary matches, but multiple related terms. Simple word matching will give a low similarity score, even though we as humans recognise that the texts describe similar contexts.

Latent Semantic Analysis

- *Latent Semantic Analysis (LSA)* was introduced by Deerwester et al. (1990) as a method for document management and retrieval. It has since become a standard tool for distributional semantics.
- The core motivation behind LSA is that using word matching to compute similarity between documents (or words) ignores "latent" conceptual aspects of language.
- LSA attempts to discover a data representation that has much lower dimension than the original feature space but preserves the most important aspects of the data.
- The mathematical technique behind LSA is also known as *Principal Components Analyis*, a well-established tool in many areas of statistical learning.
- Our discussion of LSA assumes a word space model of the kind we built in the previous section.

The LSA algorithm I

Step 1: LSA takes as input a co-occurrence matrix X, where cell x_{ij} contains the co-occurrence frequency of word w_i and context c_j. X has dimension $V \times C$. This matrix is transformed in two steps:

i. Each x_{ij} is replaced by $\log(x_{ij} + 1)$.
ii. All entries in row $\mathbf{x}_i$ are divided by the entropy H of that word's co-occurrence distribution:

$$H = -\sum_j \frac{x_{ij}}{\sum_{j'} x_{ij'}} \log \frac{x_{ij}}{\sum_{j'} x_{ij'}}$$

Step i reduces the relative effect of single large co-occurrence counts. Step ii reduces the effect of words that have a more uniform co-occurrence distribution.

The LSA algorithm II

Step 2: The transformed data matrix $\tilde{X}$ is then decomposed using the *Singular Value Decomposition (SVD)*:

$$\tilde{X} = U\Sigma V^T$$

where U and V are orthonormal matrices (all columns are orthogonal and have unit length) and Σ is a diagonal matrix of *singular values* $\sigma_1, \ldots, \sigma_n$. By convention the components of U, V and T are ordered so that $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_n$.

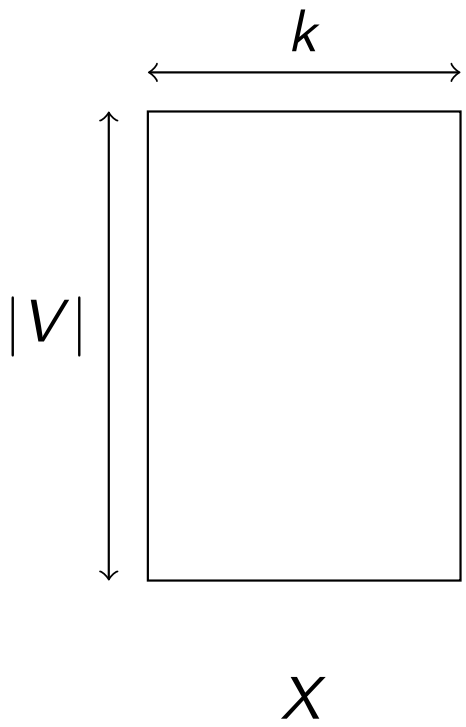
The LSA algorithm III

Step 3: In order to reduce the dimensionality of the data, we keep only the first l singular values and the corresponding columns of U and V:

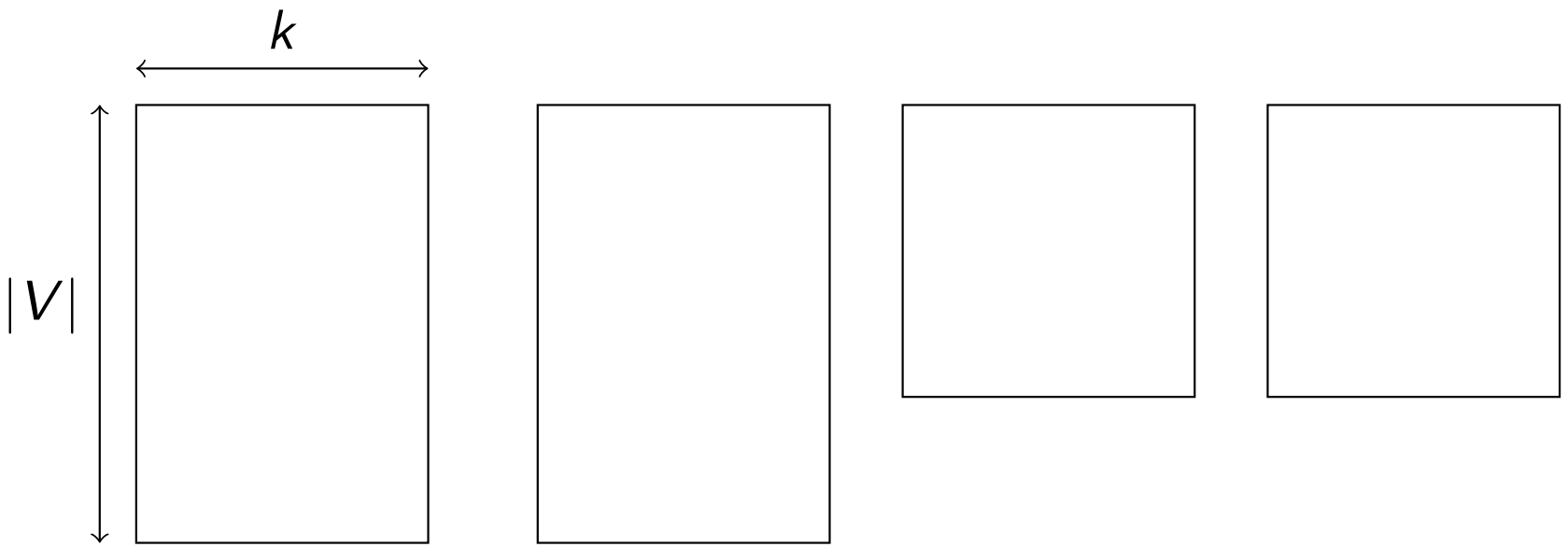
$$\hat{X}_l = U_{1:l}\Sigma_{1:l}V_{1:l}^T$$

The new vector for word w_i in the new reduced vector space is given by the ith row of $U_{1:l}$. The columns of V "explain" how the basis elements of the new space correspond to linear combinations of the basis elements of the original feature space.

LSA in pictures

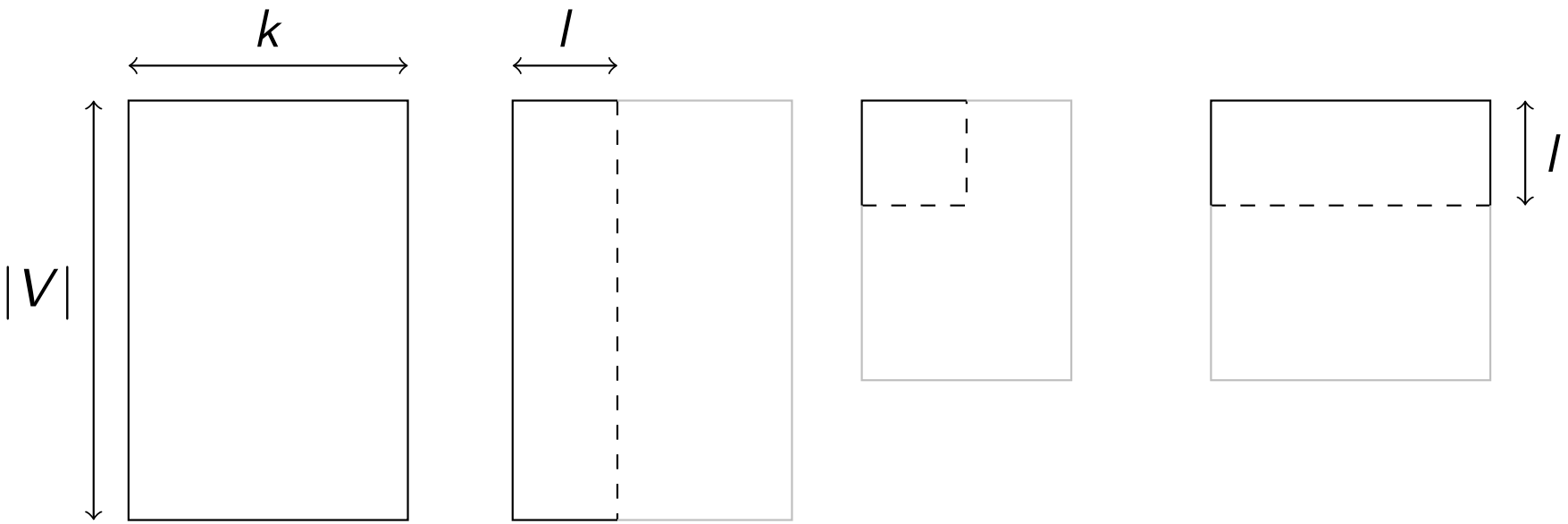


LSA in pictures



$$X = U \times \Sigma \times V^T$$

LSA in pictures



$$X \approx U_{1:l} \times \Sigma_{1:l} \times V^T_{1:l}$$

LSA example components

Topic 1	Topic 2	Topic 3	Topic 4
back	technique	crohn's	obscurity
cried	simple	biliary	alkali
looked	characteristics	bladder	brink
suddenly	method	colorectal	detriment
quietly	techniques	chronic	cyclic
laughing	easily	gastrointestinal	flicking
sighed	limitations	bowel	levers
watched	readily	cardiac	needles

(Widdows, 2004)

A famous LSA experiment

- The Test of English as a Foreign Language (TOEFL) synonym test requires learners of English to identify synonymous words in the presence of non-synonym distractors:

  You will find the office at the main **intersection**.
  (a) place
  (b) crossroads
  (c) roundabout
  (d) building

- Landauer and Dumais (1997) train an LSA model on 4.6m words of text from Grolier's Academic American Encyclopedia, a reference book for young adults.
- On a set of 80 multiple choice TOEFL questions, LSA scores 64.5% accuracy, equivalent to the average scores of applicants to US universities from non-English-speaking countries.

A famous LSA experiment

- The Test of English as a Foreign Language (TOEFL) synonym test requires learners of English to identify synonymous words in the presence of non-synonym distractors:

  You will find the office at the main **intersection**.
  (a) place
  (b) crossroads
  (c) roundabout
  (d) building

- Landauer and Dumais (1997) train an LSA model on 4.6m words of text from Grolier's Academic American Encyclopedia, a reference book for young adults.
- On a set of 80 multiple choice TOEFL questions, LSA scores 64.5% accuracy, equivalent to the average scores of applicants to US universities from non-English-speaking countries.

Application: Text-to-text similarity

- Recall the text-to-text similarity problem: How similar or related are these two texts?

Text 1

I travelled to Beijing by plane. It is a beautiful city with many attractions for tourists.

Text 2

I flew in to Beijing on Tuesday. The Chinese capital is really inviting, there's so much to do here and I hope to visit again.

- "Bag of words" vectors for Texts 1 and 2 will have very low similarity as they share very few non-zero features. Using LSA to project the texts onto lower-dimensional vectors should draw out their conceptual similarity.

Text-to-text similarity

- As an alternative, Mihalcea et al. (2006) propose using word similarity measures as building blocks in a measure of text similarity:

$$sim(T_1, T_2) = \frac{\sum_{w_1 \in T_1} \max_{w_2 \in T_2} sim(w_1, w_2) * idf(w_1)}{2 \sum_{w_1 \in T_1} idf(w_1)} + \frac{\sum_{w_2 \in T_2} \max_{w_1 \in T_1} sim(w_2, w_1) * idf(w_2)}{2 \sum_{w_2 \in T_2} idf(w_2)}$$

  where $idf(w) = \frac{1}{df(w)}$ is the inverse document frequency of w.

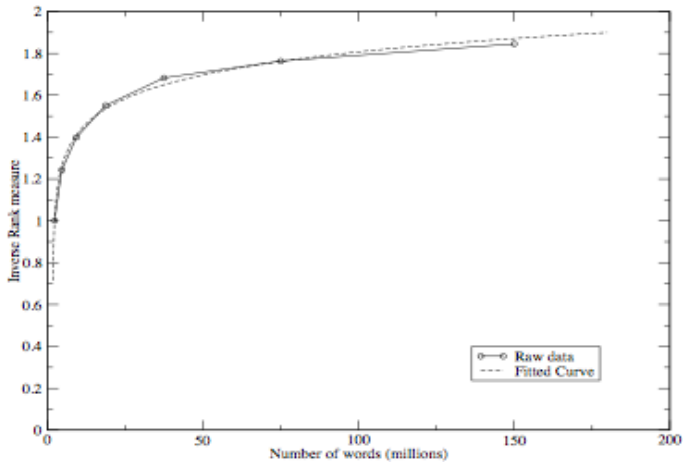
- Any similarity measure between words can be used (Mihalcea et al. investigate corpus-based and WordNet-based measures); two that perform well are PMI-IR and LSA.

LSA - Why does it work?

- The singular vales σ_i correspond the amount of variance in the data that is associated with the direction $\mathbf{v}_i$.
- If we assume that the directions of high variance correspond to interesting aspects of the data and directions of low variance are uninteresting "noise", then selecting the largest components "cleans" the data.
- Another assumption is that the space of meanings is smaller than the space of words (there are many ways to express the same meaning). LSA identifies the linear subspace of the data that is closest to the full dataset for any choice of l.
- It has been claimed that processes in human cognition have a function very similar to dimensionality reduction (Seung and Lee, 2000; Landauer and Dumais, 1997).

How much data do we need for distributional semantics?

- Word features are sparse, co-occurrence features are even sparser.
- In general, more data is better:



(Curran, 2003)

How much data do we need for distributional semantics?

- Word features are sparse, co-occurrence features are even sparser.
- In general, more data is better (but mind the quality):

Corpus	Tokens (m)	Types (m)	R-Prec
Wikipedia	721	34	0.315
Web004	8,717	22	0.264
Web020	43,588	108	0.356
Web100	217,940	542	0.404

(Pantel et al., 2009)

The Web as corpus

- In theory, the World Wide Web contains as much text data as we might ever need. However:
  - Massive datasets are non-trivial to store and process, especially if parsing is required.
  - Internet data can be very noisy in terms of both format (often messy HTML) and content (spam).
  - Some useful resources for Web-as-corpus semantics:
    - Google n-gram corpora containing frequency counts for 1- to 5-grams, available from the Linguistic Data Consortium
      English: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13`
      Chinese: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2010T06`
    - WaCky project: billion-word fulltext corpora for various languages
      `http://wacky.sslmit.unibo.it/doku.php?id=corpora`

Web-scale distributional semantics - an exact approach

- Similarity measures that treat each pair of points and each dimension independently (like cosine) are trivial to parallelise.
- Pantel et al. (2009) describe using MapReduce to compute exact similarity values for a Web corpus of 2×10^{11} tokens, "*pairwise similarity between 500 million terms is computed in 50 hours using 200 quad-core nodes*".
- This requires a lot of computing power - maybe only practical if you work for Google or Microsoft.
- On the other hand, the calculations only have to be performed once and can be done offline.

Web-scale distributional semantics - approximate approaches I

- Approximate approaches to large-scale distributional semantics are based on the idea that we are willing to tolerate some amount of error in our representation in order to process more data more efficiently.
- Ravichandran et al. (2005) describe a technique based on *Locality Sensitive Hashing (LSH)*, where we use a set of random hash functions that preserve cosine similarity: similar word vectors are likely to be hashed close together.
- Given d randomly sampled hash functions $h_1, \ldots, h_d$ and a word vector $\mathbf{w}$, $\mathbf{w}$ is mapped to a new binary vector $\mathbf{h}(\mathbf{w}) = (h_1(\mathbf{w}), \ldots, h_d(\mathbf{w}))$. The cosine between words is approximated by the Hamming distance betwen their LSH vectors, which can be computed very efficiently.

Web-scale distributional semantics - approximate approaches II

- Assuming that $d \ll k$, this reduces the $O(n^2 k)$ task of computing a full similarity matrix to an $O(nk)$ task. To reduce the error in the approximation we can increase d.
- For similar approaches, see also Van Durme and Lall (2010) and Goyal and Daumé III (2011).
- Also related is the *random indexing* method of dimensionality reduction Kanerva et al. (2000).

Web-scale distributional semantics - a simpler approach

- Keller and Lapata (2003) and Lapata and Keller (2004) show that useful semantic information can be extracted simply from entering combinations of words into a search engine and observing the page counts returned.
- If we have a flexible query language we can submit queries such as w_1 *NEAR* w_2 or w_1 * * * w_2.
- This approach can be surprisingly successful for tasks such as selectional preference prediction and compound noun paraphrasing.
- Web querying is only suitable as a solution for tasks that look like language modelling (*how frequently do I see these words together?*).
- Note also that search engine results are not always transparent and API usage limits make it hard to construct a general distributional model.

Summary

- We have introduced the Distributional Hypothesis and seen how it can be implemented to discover semantic information from text data.
- The vector space model of meaning gives us a way to compare the distributional profiles of words and is a fundamental building block in many NLP applications.
- The vector space model is extremely flexible; among the parameters we have considered are the definition of context types and the use of association measures.
- Dimensionality reduction can help us find "hidden structure" in the data and improve our ability to compare related items.
- A current focus of intense research is the ability of distributional models to "scale up" to data as big as the Web.