L114 Lexical Semantics

Session 4: Semantic Spaces and Semantic Similarity

Simone Teufel

Natural Language and Information Processing (NLIP) Group
UNIVERSITY OF CAMBRIDGE

Simone.Teufel@cl.cam.ac.uk

(Slides after Stefan Evert)

2013/2014

Distributional Semantic Spaces

- We want to automatically determine how "similar" two words are.
- Distributional hypothesis of word meaning:
  - "Die Bedeutung eines Wortes liegt in seinem Gebrauch."
    –Ludwig Wittgenstein
  - "You shall know a word by the company it keeps."
    J.R. Firth (1957)
- Represent a word by its syntagmatic and paradigmatic affinities, and you have captured its meaning.
- Today: how to create models that do that (and that can be used for many NLP applications)
- Apart from the Distributional Measures treated here, there are also Thesaurus-based Methods (cf. JM chapter 20.6)

What is ⊠ similar to?

	■	⥀	∇	★	♭	↫
♦	51	20	84	0	3	0
⊞	52	58	4	4	6	26
⊠	115	83	10	42	33	17
♠	59	39	23	4	0	0
♡	98	14	6	2	1	0
♢	12	17	3	2	9	27
♣	11	2	2	0	18	0

What is ⊠ similar to?

	■	⥀	∇	★	♭	↫
♦	51	20	84	0	3	0
⊞	52	58	4	4	6	26
⊠	115	83	10	42	33	17
♠	59	39	23	4	0	0
♡	98	14	6	2	1	0
♢	12	17	3	2	9	27
♣	11	2	2	0	18	0

$$\mathrm{sim}(\boxtimes, \blacklozenge) = 0.770$$
$$\mathrm{sim}(\boxtimes, \diamondsuit) = 0.939$$
$$\mathrm{sim}(\boxtimes, \boxplus) = 0.961$$

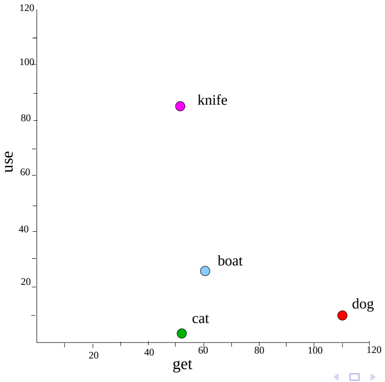
What it really looks like

	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

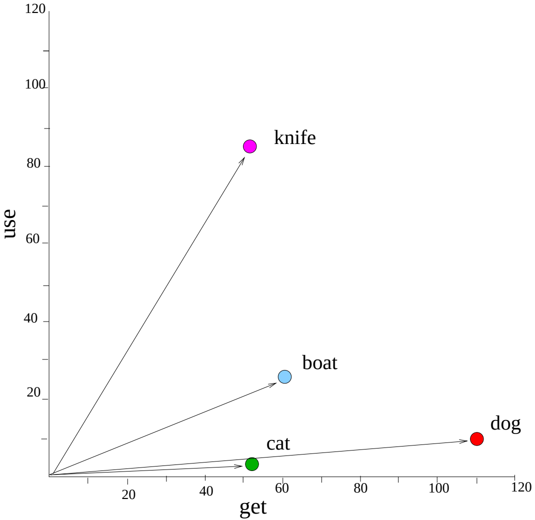
- Row vector x_{dog} describes the usage of the word *dog* in the corpus.

Geometric interpretation

- Row vector can be seen as coordinates of point/vector "dog" in n-dimensional Euclidean space
- Illustrated with two dimensions, *get* and *use*. $x_{dog} = (115, 10)$



Cosign of Vector angles in Semantic Space



Variations of (Distributional) Semantic Space

- What we looked at so far was one particular semantic space: V-obj. term–term matrix with frequency counts.
- There are many alternative types of semantic spaces.
- Definition of DSM (Distributional Semantic Model): a scaled and/or transformed co-occurrence Matrix M such that each row x represents a distribution of a target term across contexts

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between
3. Type of context (co-occurrence, dependency relations (structured, lexicalised?), . . .)

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between
3. Type of context (co-occurrence, dependency relations (structured, lexicalised?), . . .)
4. Feature scaling/term weighting

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between
3. Type of context (co-occurrence, dependency relations (structured, lexicalised?), . . .)
4. Feature scaling/term weighting
5. Normalisation of rows/columns

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between
3. Type of context (co-occurrence, dependency relations (structured, lexicalised?), . . .)
4. Feature scaling/term weighting
5. Normalisation of rows/columns
6. Compression/Dimensionality Reduction

Dimensions of Distributional Semantic Models

1. Linguistic Pre-processing: definition of a term
2. Size of context in Term–Context matrix: Context can be document, or term, or anything in between
3. Type of context (co-occurrence, dependency relations (structured, lexicalised?), . . .)
4. Feature scaling/term weighting
5. Normalisation of rows/columns
6. Compression/Dimensionality Reduction
7. Proximity measure chosen

Linguistic Preprocessing

- Tokenisation
- POS-tagging (*light*/N vs *light*/A vs *light*/V)
- Stemming/lemmatisation
  - *go, goes, went, gone, going* $\rightarrow$ *go*
- Dependency parsing or shallow syntactic chunking

Effect of Linguistic Preprocessing

Nearest Neighbours of *walk* (BNC):

Word forms
stroll
walking
walked
go
path
drive
ride
wander
sprinted
sauntered

Lemmatised forms
hurry
stroll
stride
trudge
amble
wander
walk-NN
walking
retrace
scuttle

(Semantic space above is defined by (head of) subject — verb)

Term–document vs term–term matrices

- In Information Retrieval, the "context" is always exactly one document.
- This results in term–document matrices (called the "Vector Space Model")
- This allows us to measure the similarity of words with sets of words (e.g., documents vs. queries in IR).
- Term–document matrices are sparse

	doc1	doc2	doc3	doc4	doc5	doc6	doc7	doc8
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	0	0	1	0
information	0	0	1	1	1	0	1	0
arts	1	0	0	0	0	0	0	0

Context Type

- But in Lexical semantics, different contexts can be used.
- Some possibilities:
  - Context term appears in same fixed window
  - Context term is member in same linguistic unit as target (e.g., paragraph, turn in conversation)
  - Context term is linked to target term by a syntactic dependency (e.g.,subject, modifier)

Nearest neighbours of *car* and *dog* (BNC)

2-word window	
car	dog
van	cat
vehicle	horse
truck	fox
motorcycle	pet
driver	rabbit
motor	pig
lorry	animal
motorist	mongrel
cavalier	sheep
bike	pigeon

30-word window	
car	dog
drive	kennel
park	puppy
bonnet	pet
windscreen	bitch
hatchback	terrier
headlight	rottweiler
jaguar	canine
garage	cat
cavalier	to bark
tyre	Alsatian

Nearest neighbours of *car* and *dog* (BNC)

2-word window	
car	dog
van	cat
vehicle	horse
truck	fox
motorcycle	pet
driver	rabbit
motor	pig
lorry	animal
motorist	mongrel
cavalier	sheep
bike	pigeon

30-word window	
car	dog
drive	kennel
park	puppy
bonnet	pet
windscreen	bitch
hatchback	terrier
headlight	rottweiler
jaguar	canine
garage	cat
cavalier	to bark
tyre	Alsatian

Tendency:

paradigmatically related

syntagmatically related

Semantic Similarity vs. Relatedness

There are at least two dimensions of word associations:

- Semantic Similarity (aka paradigmatic relatedness): two words sharing a high number of salient features (attributes)
  - (near) synonymy (*car–automobile*)
  - hyperonymy (*car–vehicle*)
  - co-hyponymy (*car–van–lorry–bike*)
- Semantic Relatedness (aka syntagmatic relatedness): two words semantically associated without being necessarily similar
  - function (*car–drive*)
  - meronymy (*car–tyre*)
  - location (*car–road*)
  - attribute (*car–fast*)
  - other (*car–petrol*)

Nearest Neighbours of *car* (BNC)

2-word window	
van	co-hyponym
vehicle	hyperonym
truck	co-hyponym
motorcycle	co-hyponym
driver	related entity
motor	meronym
lorry	co-hyponym
motorist	related entity
cavalier	hyponym
bike	co-hyponym

30-word window	
drive	function
park	typical action
bonnet	meronym
windscreen	meronym
hatchback	meronym
headlight	meronym
jaguar	hyponym
garage	location
cavalier	hyponym
tyre	meronym

Evaluating Distributional Similarity Intrinsically

Intrinsic means by direct comparison to the right answer

- Compare to human association norms, e.g., Rubenstein and Goodenough (1965) – 65 word pairs
  - Scoring on a scale of 0–4
  - stable and replicable
    - car–automobile 3.9
    - food–fruit 2.7
    - cord–smile 0.0
  - Miller and Charles (1991) – 30 word pairs
- Simulate semantic priming data
  - Hearing/reading a "related" prime facilitates access to a target in various lexical tasks (naming, lexical decision, reading)
  - The word *pear* is recognised/accessed faster if it is heard/read after *apple*.
- Compare to thesaurus(es), using precision and recall
  - Curran (2003) found Dice, Jaccard and t-score association metric to work best

Evaluating Distributional Similarity Extrinsically

Extrinsic means measure success of end-to-end application that uses DS.

- Synonym tasks and other language tests (Landauer and Dumais 1997; Turney et al. 2003), e.g. TOEFL test
  - Which of 4 multiple choices is correct synonym of a test word?
  - Target: **levied**
    Candidates: *imposed, believed, requested, correlated*
- Detection of malapropism (contextual misspellings): "It is minus 15, and then there is the **windscreen** factor on top of that." (Jones and Martin 1997)
- PP-attachment disambiguation (Pantel 2000)
- Query expansion in information retrieval (Salton, Wang and Yang 1975, Grefenstette 1994)

More Extrinsic Evaluations for Distributional Similarity

- Automatic thesaurus extraction and expansion (Grefenstette 1994, Lin 1998, Pantel 2000, Rapp 2004)
- Classification of 44 concrete nouns (ESSLLI 2008 competition) (animals: bird vs. ground; tools, vehicles, plants: fruit vs vegetables)
- WSD (Schuetze 1998) and WS ranking (McCarthy et al. 2004)
- Text segmentation (Choi, Wiemer-Hastings and Moore, 2001)
- Unsupervised part-of-speech induction (Schuetze 1995)
- Many other tasks in computational semantics: entailment detection, noun compound interpretation, detection of idioms, ...

TOEFL test

Lexicalised grammatical relations (Lin 1998)

subj-of, absorb	1
subj-of, adapt	1
subj-of, behave	1
...	
pobj-of, inside	16
pobj-of, into	30
...	
nmod-of, abnormality	3
nmod-of, anemia	8
nmod-of, architecture	1
...	
obj-of, attack	6
obj-of, call	11
obj-of, come from	3
obj-of, decorate	2
...	
nmod, bacteria	3
nmod, body	2
nmod, bone marrow	2

Context word: **cell**; frequency counts from 64-Million word corpus.

Structured vs. Unstructured Dependencies

A dog bites a man. The man's dog bites a dog. A dog bites a man.

unstructured	bite
dog	4
man	2

structured	bite-subj	bite-obj
dog	3	1
man	0	2

Pado and Lapata (2007) investigate dependency-based semantic spaces in detail; they weight the relative importance of different syntactic structures.

Feature Scaling

- How can we discount less important features?
- Two solutions:
  - If they occur in few contexts overall, they must be important
    - Zipf's law; TF*IDF
  - If they co-occur with our target word more than expected, they must be important
    - Association metrics

Zipf's Law

Most frequent words in a large language sample, with frequencies:

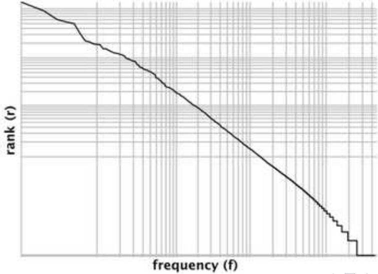
Rank	English (BNC)		German
1	the	61847	der
2	of	29391	die
3	and	26817	und
4	a	21626	in
5	in	18214	den
6	to	16284	von
7	it	10875	zu
8	is	9982	das
9	to	9343	mit
10	was	9236	sich
11	I	8875	des
12	for	8412	auf
13	that	7308	für
14	you	6954	ist
15	he	6810	im

Zipf's Law

Zipf's Law: The frequency rank of a word is reciprocally proportional to its frequency:

$$freq(word_i) \sim \frac{1}{i} freq(word_1)$$

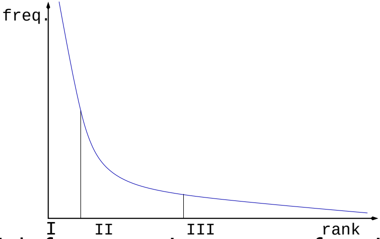
($word_i$ is the ith most frequent word of the language)
Plotting a Zipfian distribution on a log-scale:



Other collections (allegedly) obeying Zipf's law

- Sizes of settlements
- Frequency of access to web pages
- Income distributions amongst top earning 3% individuals
- Korean family names
- Size of earth quakes
- Word senses per word
- Notes in musical performances
- ...

Zipf's law as motivation for Term Weighting



- **Zone I**: High frequency items, e.g., function words, carry little semantics. (Top 135 types account for 50% of tokens in Brown corpus.)
- **Zone II**: Mid-frequency items, best indicators of semantics of the co-occurring word.
- **Zone III**: Low frequency words tend to be overspecific (e.g., "Uni7ed", "super-noninteresting", "87-year-old", "0.07685")

Term Weighting

- Not all terms describe a document equally well
- Terms which are frequent in a document are better:

$$tf_{w,d} = freq_{w,d}$$

- Terms that are overall rare in the document collection are better:

$$idf_{w,D} = log\frac{|D|}{n_{w,D}}$$

$$tfidf_{w,d,D} = tf_{w,d} \times idf_{w,D}$$

- Improvement: Normalize by term frequency of most frequent term in document

$$norm_tf_{w,d} = \frac{freq_{w,d}}{max_{l \in d} freq_{l,d}}$$

TF*IDF, formulae

$tfidf_{w,d,D}$	TFIDF weight of word w in document d in document collection D.
$tf_{w,d}$	Term frequency of word w in document d
$norm_t f_{w,d}$	Normalized term frequency of word w in document d
$idf_{w,D}$	Inverse document frequency of word w in document collection D
$n_{w,D}$	Number of documents in document colletion D which contain word w
$max_{l \in d} freq_{l,d}$	Maximum term frequency of any word in document d

Example: TF*IDF

Document set contains N=30,000 documents

Term	tf	$n_{w,D}$	TF*IDF
the	312	28,799	5.55
in	179	26,452	9.78
general	136	179	302.50
fact	131	231	276.87
explosives	63	98	156.61
nations	45	142	104.62
1	44	2,435	47.99
haven	37	227	78.48
2-year-old	1	4	3.88

$\text{IDF(“the”)} = \log\left(\frac{30{,}000}{28{,}799}\right) = 0.0178$
$\text{TF*IDF(“the”)} = 312 \cdot 0.0178 = 5.55$

Association measures: weighting co-occurrences

How surprised should we be to see context term associated with the target word?
Expected co-occurrence frequency:

$$f_{exp} = \frac{f_1 \cdot f_2}{N}$$

	eat	get	hear	kill	see	use
boat	7.0	52.4	7.3	9.5	31.2	17.6
cat	8.4	62.8	8.8	11.4	37.5	21.1
cup	6.8	50.7	7.1	9.2	30.2	17.0

...

PMI

Pointwise Mutual Information (PMI) compares observed vs. expected frequency of a word combination:

$$PMI(word_1, word_2) = log_2 \frac{f_{obs}}{f_{exp}} = log_2 \frac{N \cdot f_{obs}}{f_1 \cdot f_2}$$

$word_2$	$word_1$	f_{obs}	f_2	f_1	PMI
dog	small	855	33,338	490,580	3.96
dog	domesticated	29	33,338	918	6.85
dog	sgjkj	1	33,338	1	10.31

Disadvantage: PMI overrates combinations involving rare terms. Log-likelihood ratio (Dunning 1993) and several other metrics correct for this.

Another Association Metric: t-score

t-score:

$$assoc_{t-test}(w_1, w_2) = \frac{f_{obs} - f_{exp}}{\sqrt{f_{obs}}}$$

How many standard deviations is f_{obs} away from expected value (f_{exp})?

	eat	get	hear	kill	see	use
knife	-2.95	-2.10	-9.23	-11.97	-4.26	6.70
cat	-0.92	-1.49	-2.13	2.82	2.67	-7.65
dog	2.76	-0.99	3.73	-1.35	0.87	-9.71
boat	-7.03	0.86	-1.48	-9.47	1.23	1.11
cup	-4.11	4.76	-2.93	-9.17	-4.20	-4.17
pig	1.60	-4.80	-1.21	4.10	-0.12	-3.42

Distance metrics

- **Manhattan Distance**: (Levenshtein Distance, L1 norm)

$$distance_{manhattan}(\vec{x}, \vec{y}) = \sum_{i=1}^{N} |x_i - y_i|$$

- **Euclidean Distance**: (L2 norm)

$$distance_{euclidean}(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

	boat	cat	cup	dog	knife
cat	1.56				
cup	0.73	1.43			
dog	1.53	0.84	1.30		
knife	0.77	1.70	0.93	1.73	
pig	1.80	0.80	1.74	1.10	1.69

Similarity Metrics

- **Cosine**: (normalisation by vector lengths)

$$sim_{cosine}(\vec{x},\vec{y}) = \frac{\vec{x}\vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{N} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}}$$

- **Jaccard** (Grefenstette, 1994):

$$sim_{jacc}(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{N} min(x_i, y_i)}{\sum_{i=1}^{N} max(x_i, y_i)}$$

- **Dice Coefficient** (Curran, 2003):

$$sim_{dice}(\vec{x},\vec{y}) = \frac{2\sum_{i=1}^{N} min(x_i, y_i)}{\sum_{i=1}^{N}(x_i + y_i)}$$

Information-Theoretic Association Measures

How similar two words are depends on how much their distributions diverge from each other.

- **Kuhlback-Leibler Divergence**

$$D(P||Q) = \sum_x P(x) log \frac{P(x)}{Q(x)}$$

Unfortunately, KL is undefined when Q(x) = 0 and P(x) $\neq$ 0, which is frequent. Therefore:

- **Jensen-Shannon Divergence**

$$sim_{JS}(\vec{x}||\vec{y}) = D(\vec{x}|\frac{\vec{x}+\vec{y}}{2}) + D(\vec{y}|\frac{\vec{x}+\vec{y}}{2})$$

Example: Lin's Online Similarity Tool

hope (N)		**hope (V)**		**brief (A)**		**brief (N)**	
optimism	0.141	would like	0.158	lengthy	0.256	legal brief	0.139
chance	0.137	wish	0.140	hour-long	0.191	affidavit	0.103
expectation	0.137	plan	0.139	short	0.174	filing	0.0983
prospect	0.126	say	0.137	extended	0.163	petition	0.0865
dream	0.119	believe	0.135	frequent	0.163	document	0.0835
desire	0.118	think	0.133	recent	0.158	argument	0.0832
fear	0.116	agree	0.130	short-lived	0.155	letter	0.0786
effort	0.111	wonder	0.130	prolonged	0.149	rebuttal	0.0778
confidence	0.109	try	0.127	week-long	0.149	memo	0.0768
promise	0.108	decide	0.125	occasional	0.146	article	0.0758

all MINIPAR relations used; $assoc_{Lin}$ used; similarity metric from Lin(98) used.

LSA

- Vectors in standard vector space are very sparse
- Orthogonal dimensions clearly wrong for near-synonyms *canine–dog*
- Different word senses are conflated into the same dimension
- One way to solve this: **dimensionality reduction**
- Hypothesis for LSA (Latent Semantic Analysis; Landauer): true semantic space has fewer dimensions than number of words observed.
- Extra dimensions are noise. Dropping them brings out **latent** semantic space

Linear Algebra: a reminder

- Eigenvalues λ and eigenvectors $\vec{x}$ of a matrix **A**:
  $\mathbf{A}\ \vec{x} = \lambda\vec{x}$
- Example:

$$\mathbf{A} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 4 \end{pmatrix} \Rightarrow \vec{x_1} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \vec{x_2} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \vec{x_3} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\lambda_1 = 9; \lambda_2 = 4; \lambda_3 = 2$$

- Eigenvalues are determined by solving the polynomial

$$det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

I is unit matrix (diagonal consists of 1s, 0s otherwise)

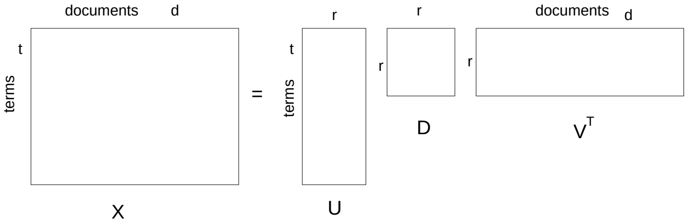
Eigenvector Decomposition

- We can decompose any square matrix C into 3 matrices

$$C = Q\Lambda Q^{-1}$$

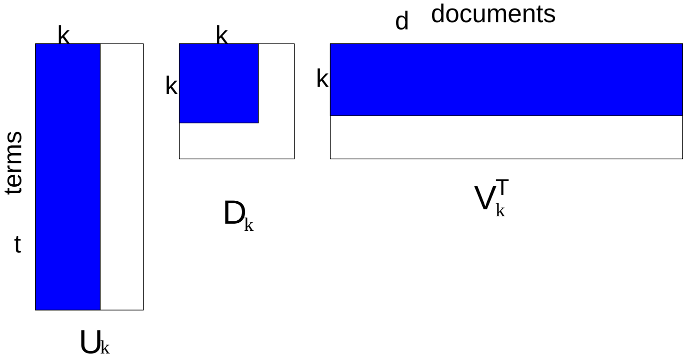
  such that Q represents the eigenvectors, and eigenvalues are listed in descending order in matrix Λ.
- Rectangular matrices need SVD (Singular Value Decomposition) for similar decomposition, because they have left and right singular vectors rather than eigenvectors.
- Left singular vectors of A are eigenvectors of AA^T.
- Right singular vectors of A are eigenvectors of A^TA.

Singular Value Decomposition



- r: rank of matrix; t: no of terms; d: no of documents
- D contains singular values (square roots of common eigenvalues for U and V) in descending order
- U contains left singular vectors of X in same ordering
- V contains right singular vectors of X in same ordering

Singular Value Decomposition



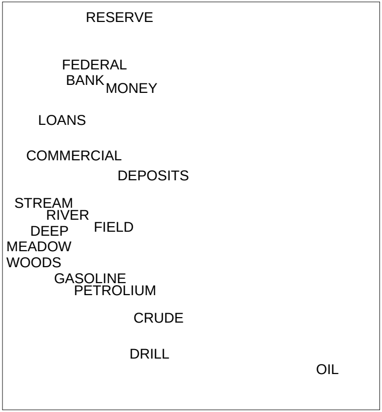
- Keep only first k (most dominant) singular values in D
- This results in two latent semantic spaces:
  - Reduced U_k represents terms in topic/concept space
  - Reduced V_k represents documents in topic/concept space collection

Dimensionality Reduction

Similarity calculations in LSI:

- Term–term similarity: $U_k D_k$
- Document–document similarity: $V_k D_k$
- Matrix D_k scales axes for comparison across spaces

Example: first 2 dimensions



from Griffiths, Steyvers, Tenenbaum (2007)

TOEFL test again

- **levied** vs *imposed, believed, requested, correlated*
- LSA: 64.5% correct; real applicants: 64.5%; native speakers 97.75% (Rapp, 2004)
- Can also explain human learning rate.
  - 40K-100K words known by age 20: 7-15 new words each day; one new word is learned in each paragraph.
  - But: experiments show only 5-10% successful learning of novel words
  - L&D hypothesize that reading provides knowledge about other words not present in immediate text.
  - Simulations show: direct learning gains 0.0007 words per word encountered. Indirect learning gains 0.15 words per article → 10 new words per day

Reading

- Jurafsky and Martin, chapters 20.7 (Word Similarity: Distributional Methods);
- Dekang Lin (1998), Automatic Retrieval and Clustering of Similar Words, ACL-98.

Further Reading

- Pado and Lapata (2007). Dependency-based Construction of Semantic Spaces. *Computational Linguistics.*
- Griffiths, Steyvers, Tenenbaum (2007). Topics in Semantic Representation. *Psychological Review*, 114(2):211.
- Landauer and Dumais (1997). A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211.