CS 4650/7650
Distributional Lexical Semantics

Jacob Eisenstein[1]

October 31, 2013

[1]Some slides borrowed from Marco Baroni and Michael Collins

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$
    $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$
    $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$
- **Shallow semantics**
  - identify the key predicates and arguments in sentences
  - $[_{agent}$ Doris] **gave** $[_{goal}$ Cary] $[_{theme}$ the book].

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$ $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$
- **Shallow semantics**
  - identify the key predicates and arguments in sentences
  - $[_{agent}$ Doris$]$ **gave** $[_{goal}$ Cary$]$ $[_{theme}$ the book$]$.
- **Today: lexical semantics**

  vector-space models for the meaning of individual words

From words to meaning

A recurring theme in this course is that the mapping from words to meaning is complex.

- **Word sense disambiguation**: multiple meanings for the same form (e.g., bank)

From words to meaning

A recurring theme in this course is that the mapping from words to meaning is complex.

- **Word sense disambiguation**: multiple meanings for the same form (e.g., bank)
- **Morphological analysis**: shared semantic basis among multiple forms (e.g., speak, spoke, speaking)

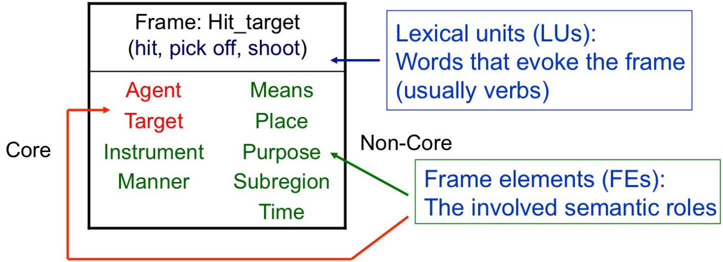
From words to meaning

A recurring theme in this course is that the mapping from words to meaning is complex.

- **Word sense disambiguation**: multiple meanings for the same form (e.g., bank)
- **Morphological analysis**: shared semantic basis among multiple forms (e.g., speak, spoke, speaking)
- Both **compositional** and **frame** semantics assume hand-crafted resources that map from words to predicates.

FrameNet

A Frame defines a set of *lexical units* and a set of *frame elements*:



[$_{\text{Agent}}$ *Kristina*] ***hit*** [$_{\text{Target}}$ *Scott*] [$_{\text{Instrument}}$ *with a baseball*] [$_{\text{Time}}$ *yesterday*].

Combinatory Categorial Grammar

In CCG semantic parsing, we assume a **lexicon** that encodes both the syntax and semantics of each word.

$$\text{opened} \vdash (S\backslash NP)/NP : \lambda x.\lambda y.\textsc{opened}(x, y)$$
$$\text{Moe's} \vdash NNP : \textsc{Moe's}$$

New words

How do we do semantic analysis of words that we've never seen before?

New words

How do we do semantic analysis of words that we've never seen before?

- A bottle of tezgüino is on the table.
- Everybody likes tezgüino.
- Tezgüino makes you drunk.
- We make tezgüino out of corn.

New words

How do we do semantic analysis of words that we've never seen before?

- A bottle of __________ is on the table.
- Everybody likes __________.
- __________ makes you drunk.
- We make __________ out of corn.

New words

How do we do semantic analysis of words that we've never seen before?

- A bottle of __________ is on the table.
- Everybody likes __________.
- __________ makes you drunk.
- We make __________ out of corn.

How well do other words fit into these contexts?

- Loud, motor oil, tortillas, choices, wine

Distributional similarity

- Words that occur in similar contexts have similar meanings. "You shall know a word by the company it keeps" (Firth 1957)
- Today we will see how to implement this idea using large datasets of unlabeled text.

Distributional similarity

- Words that occur in similar contexts have similar meanings. "You shall know a word by the company it keeps" (Firth 1957)
- Today we will see how to implement this idea using large datasets of unlabeled text.
- Why do we care about similarity?
  - **Query expansion**: search for bike, match bicycle

Distributional similarity

- Words that occur in similar contexts have similar meanings. "You shall know a word by the company it keeps" (Firth 1957)
- Today we will see how to implement this idea using large datasets of unlabeled text.
- Why do we care about similarity?
  - **Query expansion**: search for bike, match bicycle
  - **Semi-supervised learning**: use large unlabeled datasets to acquire features which are useful in supervised learning

Distributional similarity

- Words that occur in similar contexts have similar meanings.
  "You shall know a word by the company it keeps" (Firth 1957)
- Today we will see how to implement this idea using large datasets of unlabeled text.
- Why do we care about similarity?
  - **Query expansion**: search for bike, match bicycle
  - **Semi-supervised learning**: use large unlabeled datasets to acquire features which are useful in supervised learning
  - **Lexicon and thesaurus induction**: automatically expand hand-crafted lexical resources, or induce them from raw text

The vector-space model

Key idea: each word (type) is represented by a vector of contexts.

- C1: A bottle of __________ is on the table.
- C2: Everybody likes __________.
- C3: __________ makes you drunk.
- C4: We make __________ out of corn.
- ...

The vector-space model

Key idea: each word (type) is represented by a vector of contexts.

- C1: A bottle of _________ is on the table.
- C2: Everybody likes _________.
- C3: _________ makes you drunk.
- C4: We make _________ out of corn.
- ...

	C1	C2	C3	C4	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
choices	0	1	0	0	
wine	1	1	1	1	

The Vector-space model

- The "meaning" of tezgüino is represented by the vector $\{1, 1, 1, 1, \ldots\}$.
- Wine has a similar vector and therefore a similar meaning.
- The vector-space model is used in a huge range of NLP and information retrieval applications.
- Key technical questions:
  - How kinds of context should we consider?
  - How do we measure similarity?
  - How do we distinguish frequent and infrequent events?

What is “context”?

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

What is “context”?

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

What is “context”?

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

What is “context”?

The silhouette of the sun beyond a wide-open bay on the lake; the sun still glitters although evening has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

What is “context”?

The silhouette-n of the sun beyond a wide-open-a bay-n on the lake-n; the sun still glitter-v although evening-n has arrive-v in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

What is "context"?

The silhouette-n of the sun beyond a wide-open bay on the lake; the sun still glitter-v although evening- has arrived in Kuhmo. It's midsummer; the living room has its instruments and other objects in each of its corners.

What is “context”?

The silhouette-n_ppdep of the sun beyond a wide-open bay on the lake; the sun still glitter-v_subj although evening- has arrived in Kuhmo. It’s midsummer; the living room has its instruments and other objects in each of its corners.

Same corpus (BNC), different contexts (window sizes)

Nearest neighbours of *dog*

2-word window

- cat
- horse
- fox
- pet
- rabbit
- pig
- animal
- mongrel
- sheep
- pigeon

30-word window

- kennel
- puppy
- pet
- bitch
- terrier
- rottweiler
- canine
- cat
- to bark
- Alsatian

Outline

Word clustering in local context

- In the Brown et al (1992) clustering algorithm, the context is just the immediately adjacent words.
- A generative probability model:
  - Assume each word w_i has a class c_i
  - Assume a generative model
    $\log P(w) = \sum_i \log P(w_i|c_i) + \log P(c_i|c_{i-1})$
    (What does this remind you of?)

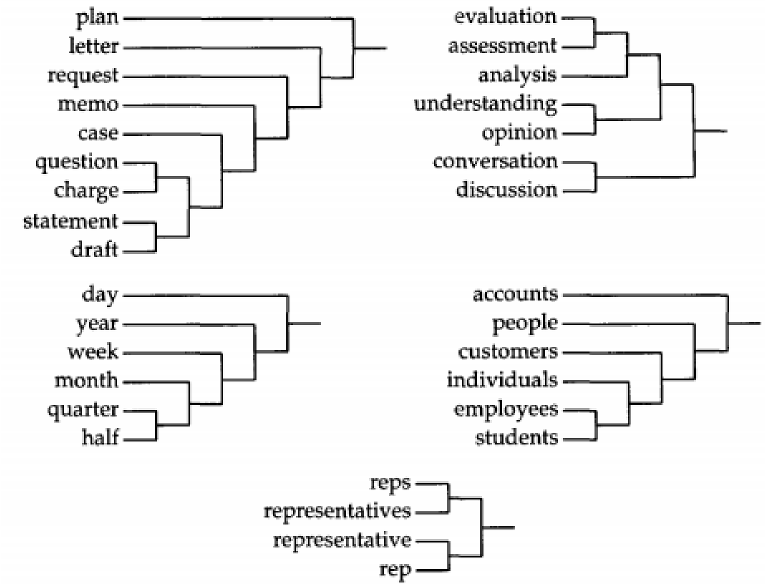
A hierarchical clustering algorithm

- Start with every word in its own cluster
- Until tired,
  - Choose two clusters c_i and c_j such that merging them will give the maximum improvement in $\log P(w)$
  - Equivalently, merge the clusters with the greatest mutual information.
- The merge path of a word describes its semantics.

Derivation

- See notes

Mutual information trees



Bit strings

- Equivalently, each word can be described by a **bit string** of branchings in the induced hierarchy.
- From Miller et al (2004):

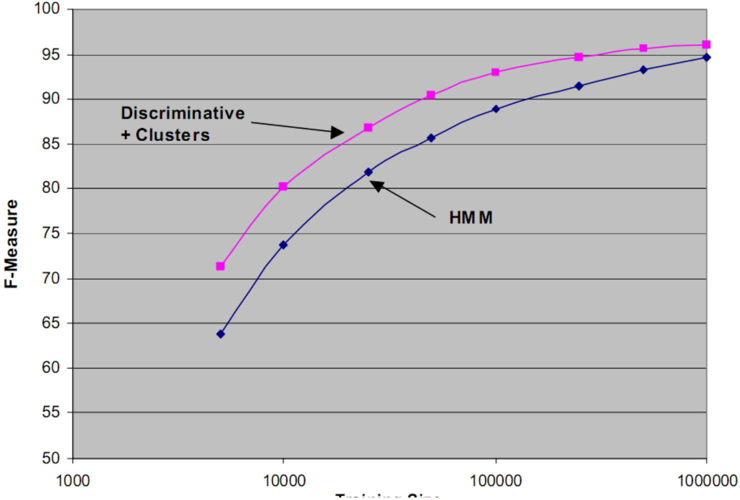
lawyer	1000001101000
newspaperman	100000110100100
stewardess	100000110100101
toxicologist	10000011010011
slang	1000001101010
babysitter	100000110101100
conspirator	1000001101011010
womanizer	1000001101011011
mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010

Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000

John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	1011100100000001100
Phillip	10111001000000011010
WILLIAM	10111001000000011011
Timothy	1011100100000001110
Terrence	10111001000000011110
Jerald	10111001000000011111
Harold	101110010000000100
Frederic	101110010000000101
Wendell	10111001000000011

- Bit strings can easily be converted into features for supervised learning.
  - Named entity tagging (Miller et al, 2004)
  - Dependency parsing (Koo et al, 2008)

Brown clusters in NER



Outline

From local to syntactic context

- Local context is contingent on syntactic decisions that may have little to do with semantics:
  - I gave Tim the ball.
  - I gave the ball to Tim.

From local to syntactic context

- Local context is contingent on syntactic decisions that may have little to do with semantics:
  - I gave Tim the ball.
  - I gave the ball to Tim.
- Using the syntactic structure of the sentence might give us a more meaningful context, yielding better clusters.

Distributional clustering of nouns

- Pereira, Tishby, and Lee, "Distributional Clustering of English Words" (ACL 1993)
  - Consider only nouns which are the direct object of verbs (using a rule-based parser).

Distributional clustering of nouns

- Pereira, Tishby, and Lee, "Distributional Clustering of English Words" (ACL 1993)
  - Consider only nouns which are the direct object of verbs (using a rule-based parser).
  - The context vector for each noun is **the count of occurences as a direct object of each verb**.

Distributional clustering of nouns

- Pereira, Tishby, and Lee, "Distributional Clustering of English Words" (ACL 1993)
  - Consider only nouns which are the direct object of verbs (using a rule-based parser).
  - The context vector for each noun is **the count of occurences as a direct object of each verb**.
  - As with Brown clustering, a class-based probability model:

$$
\begin{aligned}
\hat{p}(n, v) &= \sum_{c \in \mathcal{C}} p(c, n) p(v|c) \\
&= \sum_{c \in \mathcal{C}} p(c) p(n|c) p(v|c)
\end{aligned}
$$

where n is the noun, v is the verb, and c is the class

Distributional clustering of nouns

- Pereira, Tishby, and Lee, "Distributional Clustering of English Words" (ACL 1993)
  - Consider only nouns which are the direct object of verbs (using a rule-based parser).
  - The context vector for each noun is **the count of occurences as a direct object of each verb**.
  - As with Brown clustering, a class-based probability model:

$$\begin{aligned}\hat{p}(n, v) &= \sum_{c \in \mathcal{C}} p(c, n) p(v|c) \\ &= \sum_{c \in \mathcal{C}} p(c) p(n|c) p(v|c)\end{aligned}$$

    where n is the noun, v is the verb, and c is the class
  - Objective: find the maximum likelihood cluster centroids.

Distributional clustering from labeled dependency edges

- Dekang Lin, "Automatic Retrieval and Clustering of Similar Words" (ACL 1997)
  - Cluster all content words, not just nouns
  - Use labeled dependency edges (from a MINIPAR, a rule-based parser)
  - Contexts are counts of incoming dependency edges

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

Dependency-based word similarity

- For any pair of words i and j and relation r, we can compute:

$$P(i,j|r) = \frac{c(i,j,r)}{\sum_{i',j'} c(i',j',r)}, \qquad P(i|r) = \frac{\sum_{j'} c(i,j',r)}{\sum_{i',j'} c(i',j',r)}$$

Dependency-based word similarity

- For any pair of words i and j and relation r, we can compute:

$$P(i,j|r) = \frac{c(i,j,r)}{\sum_{i',j'} c(i',j',r)}, \qquad P(i|r) = \frac{\sum_{j'} c(i,j',r)}{\sum_{i',j'} c(i',j',r)}$$

- Let $T(i)$ be the set of pairs $\langle j, r\rangle$ such that $P(i,j|r) > P(i|r)P(j|r)$
  - $T(i)$ contains words j that are especially likely to be joined with word i in relation r.
  - Note the connection to pointwise mutual information.

Dependency-based word similarity

- For any pair of words i and j and relation r, we can compute:

$$P(i,j|r) = \frac{c(i,j,r)}{\sum_{i',j'} c(i',j',r)}, \qquad P(i|r) = \frac{\sum_{j'} c(i,j',r)}{\sum_{i',j'} c(i',j',r)}$$

- Let $T(i)$ be the set of pairs $\langle j, r\rangle$ such that $P(i,j|r) > P(i|r)P(j|r)$
  - $T(i)$ contains words j that are especially likely to be joined with word i in relation r.
  - Note the connection to pointwise mutual information.
- Similarity between u and v is defined through $T(u)$ and $T(v)$.

Quantifying similarity

- Lin considers several similarity measures for $T(u)$ and $T(v)$.
- Many of these are used widely, and are worth knowing:
  - Cosine similarity: $\frac{|T(u) \cap T(v)|}{\sqrt{|T(u)||T(v)|}}$
  - Dice similarity: $\frac{2 \times |T(u) \cap T(v)|}{|T(u)| + |T(v)|}$
  - Jaccard similarity: $\frac{|T(u) \cap T(v)|}{|T(u)| + |T(v)| - |T(u) \cap T(v)|}$

Quantifying similarity

- Lin considers several similarity measures for $T(u)$ and $T(v)$.
- Many of these are used widely, and are worth knowing:
  - Cosine similarity: $\frac{|T(u) \cap T(v)|}{\sqrt{|T(u)||T(v)|}}$
  - Dice similarity: $\frac{2 \times |T(u) \cap T(v)|}{|T(u)| + |T(v)|}$
  - Jaccard similarity: $\frac{|T(u) \cap T(v)|}{|T(u)| + |T(v)| - |T(u) \cap T(v)|}$
- Lin's metric is more complex:

$$\frac{\sum_{\langle r,w \rangle \in T(u) \cup T(v)} I(u,r,w) + I(v,r,w)}{\sum_{\langle r,w \rangle \in T(u)} I(u,r,w) + \sum_{\langle r,w \rangle \in T(v)} I(v,r,w)}$$

where $I(u,r,w)$ is the mutual information between u and w, conditioned on r.

Qualitative evaluation

Pairs of words which are each others respective nearest neighbors

Nouns		
Rank	Respective Nearest Neighbors	Similarity
1	earnings profit	0.572525
11	plan proposal	0.47475
21	employee worker	0.413936
31	battle fight	0.389776
41	airline carrier	0.370589
51	share stock	0.351294
61	rumor speculation	0.327266
71	outlay spending	0.320535
81	accident incident	0.310121
91	facility plant	0.284845
101	charge count	0.278339
111	baby infant	0.268093
121	actor actress	0.255098
131	chance likelihood	0.248942
141	catastrophe disaster	0.241986
151	fine penalty	0.237606
161	legislature parliament	0.231528
171	oil petroleum	0.227277
181	strength weakness	0.218027
191	radio television	0.215043
201	coupe sedan	0.209631

Adjective/Adverbs		
Rank	Respective Nearest Neighbors	Similarity
1	high low	0.580408
11	bad good	0.376744
21	extremely very	0.357606
31	deteriorating improving	0.332664
41	alleged suspected	0.317163
51	clerical salaried	0.305448
61	often sometimes	0.281444
71	bleak gloomy	0.275557
81	adequate inadequate	0.263136
91	affiliated merged	0.257666
101	stormy turbulent	0.252846
111	paramilitary uniformed	0.246638
121	sharp steep	0.240788
131	communist leftist	0.232518
141	indoor outdoor	0.224183
151	changed changing	0.219697
161	defensive offensive	0.211062
171	sad tragic	0.206688
181	enormously tremendously	0.199936
191	defective faulty	0.193863
201	concerned worried	0.186899

Quantitative evaluation

This method can be used to induce thesauri, which can then be compared with manually-crafted resources like WordNet and Roget's thesaurus.

	WordNet	
	average	σ_{avg}
Roget	0.178397	0.001636
sim	0.212199	0.001484
Hindle	0.204179	0.001424
Hindle_r	0.164716	0.001200
cosine	0.199402	0.001352

	Roget	
	average	σ_{avg}
WordNet	0.178397	0.001636
sim	0.149045	0.001429
Hindle	0.14663	0.001383
Hindle_r	0.115489	0.001140
cosine	0.135697	0.001275

Outline

Latent semantic analysis (LSA)

In **latent semantic analysis** (Deerwester et al., 1990), “contexts” are just the documents in which words appear.

Example of text data: Titles of Some Technical Memos

c1:	*Human* machine *interface* for ABC *computer* applications
c2:	A *survey* of *user* opinion of *computer system response time*
c3:	The *EPS user interface* management *system*
c4:	*System* and *human system* engineering testing of *EPS*
c5:	Relation of *user* perceived *response time* to error measurement
m1:	The generation of random, binary, ordered *trees*
m2:	The intersection *graph* of paths in *trees*
m3:	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4:	*Graph minors*: A *survey*

Latent semantic analysis (LSA)

In **latent semantic analysis** (Deerwester et al., 1990), “contexts” are just the documents in which words appear.

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Latent semantic analysis (LSA)

In **latent semantic analysis** (Deerwester et al., 1990), "contexts" are just the documents in which words appear.

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

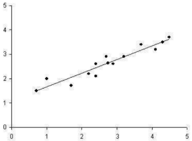
- correlation$(\text{human},\text{user}) =$ -.38
- correlation$(\text{human},\text{minors}) =$ -.29

Transforming the count matrix

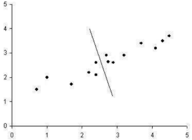
- The count matrix $\mathbf{X}$ can be huge
- In this space, similarity will be sensitive to noise.
- We'd prefer to measure similarity in a more compact space.
- Singular value decomposition (SVD): $\mathbf{X} \approx \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}$
  - $\mathbf{U}\mathbf{U}^{\mathsf{T}} = \mathbf{I}$, $\mathbf{V}\mathbf{V}^{\mathsf{T}} = \mathbf{I}$ (they are orthonormal)
  - The columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{X}\mathbf{X}^{\mathsf{T}}$.
  - The columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$.
  - $\mathbf{S}$ is a diagonal matrix containing the square roots of the eigenvalues in descending order.

Singular value decomposition (SVD)

- SVD as repeated regression on residuals:
  - fit a line to your data



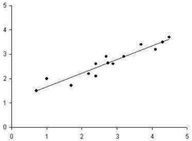
  - compute residuals
  - fit a line to the residuals



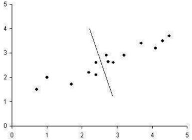
  - repeat

Singular value decomposition (SVD)

- SVD as repeated regression on residuals:
  - fit a line to your data



  - compute residuals
  - fit a line to the residuals



  - repeat

- If we fit as many lines as the smaller dimension of $\mathbf{X}$, SVD can reconstruct it exactly.
- If not, SVD forms a least-squares approximation $\hat{\mathbf{X}}$

Singular value decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Intuitively,

- **U** describes the rows (words).

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Singular value decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}$$

Intuitively,

- **U** describes the rows (words).
- $\mathbf{V}^{\mathsf{T}}$ describes the columns (documents).

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

Singular value decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathsf{T}}$$

Intuitively,

- **U** describes the rows (words).
- $\mathbf{V}^{\mathsf{T}}$ describes the columns (documents).
- **S** gives the importance of each dimension in **U** and **V**.

$$\begin{array}{ccccccccc}
3.34 & & & & & & & & \\
 & 2.54 & & & & & & & \\
 & & 2.35 & & & & & & \\
 & & & 1.64 & & & & & \\
 & & & & 1.50 & & & & \\
 & & & & & 1.31 & & & \\
 & & & & & & 0.85 & & \\
 & & & & & & & 0.56 & \\
 & & & & & & & & 0.36
\end{array}$$

Correlation in the reconstructed counts

With only two singular values, we obtain a *reduced-rank* approximation:
$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{USV}^T$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Correlation in the reconstructed counts

With only two singular values, we obtain a *reduced-rank* approximation:
$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{USV}^\mathsf{T}$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

- correlation(human,user) = .94
- correlation(human,minors) = -.83
- SVD has identified a low-dimensional basis for **X**, in which correlations are much more robust.

Title correlations

Similarly, correlation of titles in the raw counts was not informative:

Correlations between titles in raw data:

	c1	c2	c3	c4	c5	m1	m2	m3
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

		0.02	
		-0.30	0.44

Title correlations

But correlation in the reduced-rank approximation reveals the underlying structure:

Correlations in two dimensional space:

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

	0.92	
	-0.72	1.00

LSA for expanding sentiment dictionaries

Turney and Littman (2004) use LSA to expand a small sentiment dictionary.

$$\text{Semantic-orientation}(i) = \sum_{j \in \text{pos-words}} \text{sim}(u_i, u_j) - \sum_{j \in \text{neg-words}} \text{sim}(u_i, u_j)$$

- u_i is the row in the matrix $\mathbf{U}$ corresponding to word i

LSA for expanding sentiment dictionaries

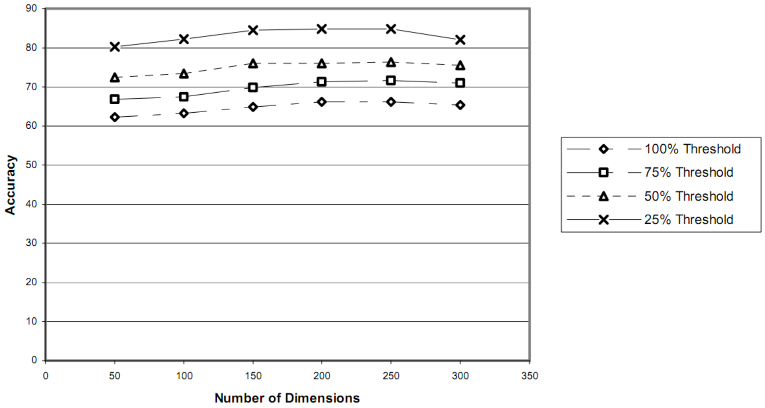
Turney and Littman (2004) use LSA to expand a small sentiment dictionary.

$$\text{Semantic-orientation}(i) = \sum_{j \in \text{pos-words}} \text{sim}(u_i, u_j) - \sum_{j \in \text{neg-words}} \text{sim}(u_i, u_j)$$

- u_i is the row in the matrix $\mathbf{U}$ corresponding to word i
- The similarity function $\text{sim}(u_i, u_j)$ is the *cosine* similarity:

$$\text{cosine}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\mathsf{T}\mathbf{y}}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

LSA for expanding sentiment dictionaries



- Dimensionality tradeoff: expressiveness for robustness
- Turney and Littman find that the ideal number of dimensions is around 250 (for this task and corpus).

LSA for automatic essay grading

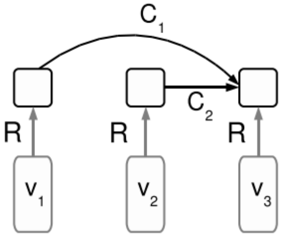
- Latent semantic analysis can be used to automatically grade test essays (Landauer et al., 1998).
- Ungraded essays are compared via cosine similarity to graded essays.
- LSA agrees with expert raters about as often as they agree with each other!
- The educational testing service (ETS) uses a combination of LSA with other features such as grammar, spelling, and repeated words (Burnstein 2003).

Limitations of LSA

- Truncated LSA gives a least-squares approximation of $\mathbf{X}$. This means that errors are **Gaussian**.
- We may prefer a bag-of-words representation:
  - Probabilistic LSA
  - Non-negative matrix factorization
  - Topic Modeling (Latent Dirichlet Allocation)
- Or we may prefer a discriminative approach...

Neural network language models

- Learn a **discriminative** model to predict the next word based on its predecessors
- Parameters are word **embeddings** **R** and transition matrix **C**. These embeddings are dense, real vectors.
- The word embeddings can be applied to semi-supervised learning (Turian et al 2010)

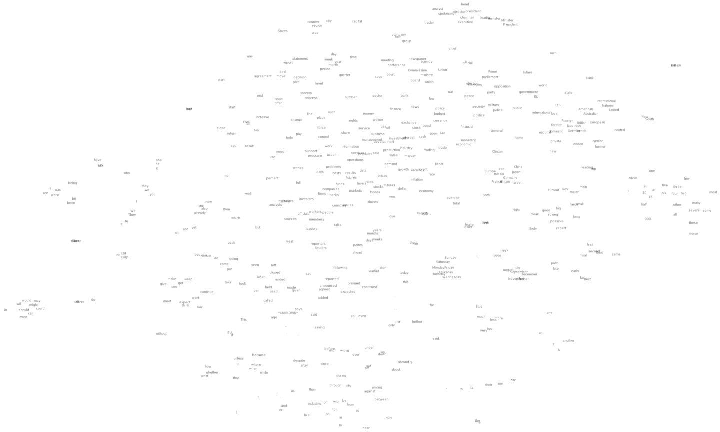


Log-bilinear language model
(Mnih and Hinton 2007)

“Neural” word embeddings, $K = 25$

“Neural” word embeddings, $K = 50$

“Neural” word embeddings, $K = 200$



Outline

Mikolov, Yih, Zweig; NAACL 2013

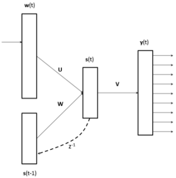
$$\mathbf{s}(t) = f(\mathbf{U}\mathbf{w}(t) + \mathbf{W}\mathbf{s}(t-1)) \quad (1)$$

$$\mathbf{y}(t) = g(\mathbf{V}\mathbf{s}(t)) \quad (2)$$

$$f(z) = \text{Logistic}(z) = \frac{1}{1+e^{-z}} \quad (3)$$

$$g(z_m) = \text{Soft-max}(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}} \quad (4)$$

$$(5)$$



- $\mathbf{w}(t)$ is a one-hot (indicator) vector for the word at token t
- $\mathbf{s}(t)$ is a dense latent vector

Mikolov, Yih, Zweig; NAACL 2013

Category	Relation	Patterns Tested	# Questions	Example
Adjectives	Base/Comparative	JJ/JJR, JJR/JJ	1000	`good:better rough:___`
Adjectives	Base/Superlative	JJ/JJS, JJS/JJ	1000	`good:best rough:___`
Adjectives	Comparative/ Superlative	JJS/JJR, JJR/JJS	1000	`better:best rougher:___`
Nouns	Singular/Plural	NN/NNS, NNS/NN	1000	`year:years law:___`
Nouns	Non-possessive/ Possessive	NN/NN_POS, NN_POS/NN	1000	`city:city's bank:___`
Verbs	Base/Past	VB/VBD, VBD/VB	1000	`see:saw return:___`
Verbs	Base/3rd Person Singular Present	VB/VBZ, VBZ/VB	1000	`see:sees return:___`
Verbs	Past/3rd Person Singular Present	VBD/VBZ, VBZ/VBD	1000	`saw:sees returned:___`

Mikolov, Yih, Zweig; NAACL 2013

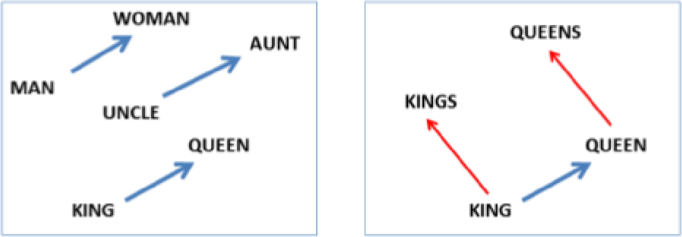


Figure 2: Left panel shows vector offsets for three word pairs illustrating the gender relation. Right panel shows a different projection, and the singular/plural relation for two words. In high-dimensional space, multiple relations can be embedded for a single word.

Given the analogy $a : b$ as $c : d$, they compute

$$\hat{d} = \arg\max_{d} \cos(u_a - u_b + u_c, u_d) \tag{6}$$

Outline

Lexical semantics in the brain

Just et al (2010) ran fMRI on subjects brains while viewing these stimuli words:

Table 1. 60 stimulus words grouped into 12 semantic categories.

Category	Exemplar 1	Exemplar 2	Exemplar 3	Exemplar 4	Exemplar 5
body parts	leg	arm	eye	foot	hand
furniture	chair	table	bed	desk	dresser
vehicles	car	airplane	train	truck	bicycle
animals	horse	dog	bear	cow	cat
kitchen utensils	glass	knife	bottle	cup	spoon
tools	chisel	hammer	screwdriver	pliers	saw
buildings	apartment	barn	house	church	igloo
building parts	window	door	chimney	closet	arch
clothing	coat	dress	shirt	skirt	pants
insects	fly	ant	bee	butterfly	beetle
vegetables	lettuce	tomato	carrot	corn	celery
man-made objects	refrigerator	key	telephone	watch	bell

doi:10.1371/journal.pone.0008622.t001

Participants were asked to think of properties of each of the words.

Factor analysis

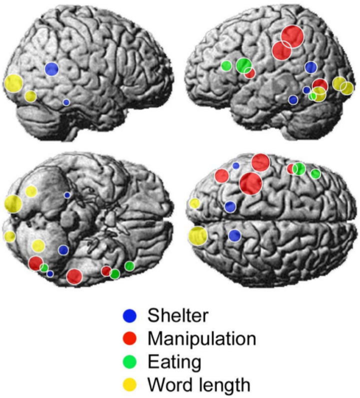
- They then identified spatial activation profiles for each word, across multiple participants.
- Factor analysis on the activation profiles identified four factors with coherent locations.

Table 2. Ten words with highest factor scores (in descending order) for each of the 4 factors.

Shelter	*Manipulation*	*Eating*	*Word length*
apartment	pliers	carrot	butterfly
church	saw	lettuce	screwdriver
train	screwdriver	tomato	telephone
house	hammer	celery	refrigerator
airplane	key	cow	bicycle
key	knife	saw	apartment
truck	bicycle	corn	dresser
door	chisel	bee	lettuce
car	spoon	glass	chimney
closet	arm	cup	airplane

Factor analysis

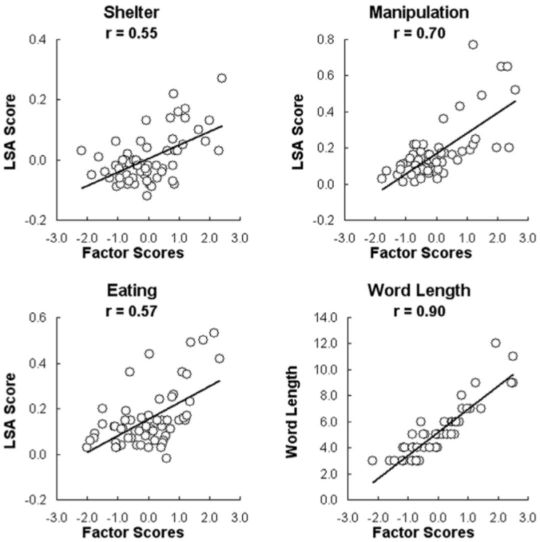
- They then identified spatial activation profiles for each word, across multiple participants.
- Factor analysis on the activation profiles identified four factors with coherent locations.



Correlation with latent semantic analysis

- The experimenters identified 5-9 additional words for each factor.
- They used LSA to measure the distance between each of the 60 stimuli factors and the factor examples.
- LSA distances were closely correlated with the factor scores of the stimuli words.

Correlation with latent semantic analysis



Outline

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$
    $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$
    $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$
- **Shallow semantics**
  - identify the key predicates and arguments in sentences
  - [$_{agent}$ Doris] **gave** [$_{goal}$ Cary] [$_{theme}$ the book].

The Semantics Roadmap

- **Compositional semantics**
  - assemble the meaning of a sentence from its components
  - What state borders Texas? $\rightarrow$
    $\lambda x.\textsc{state}(x) \wedge \textsc{borders}(x, \textsc{Texas})$
- **Shallow semantics**
  - identify the key predicates and arguments in sentences
  - $[_{agent}$ Doris$]$ **gave** $[_{goal}$ Cary$]$ $[_{theme}$ the book$]$.
- **Today: lexical semantics**
  vector-space models for the meaning of individual words

Summary of lexical semantics

- Distributional similarity is a powerful tool for understanding the relationships between words.
- The vector space model describes each word by a vector of contextual information.
- Latent semantic analysis (LSA) operates on the term-document matrix to identify a low-rank representation for both word **and** document semantics.
- Today we examined only synonymy, but there are many other lexical relations, such as *antonyms*, *part-of*, *type-of*...

Next time: discourse and reference ambiguity

- What makes a set of sentences into a coherent discourse?
- How do we resolve pronouns and other ambiguous references?