Distributional Semantics

Marco Baroni and Gemma Boleda

CS 388: Natural Language Processing
Many slides, ideas and tips from Alessandro Lenci and Stefan Evert

See also:
http://wordspace.collocations.de/doku.php/course:esslili2009:start
General introductions, surveys, overviews

- Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37: 141–188
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
The distributional hypothesis

- The meaning of a word is the set of contexts in which it occurs in texts.
- *Important aspects of* the meaning of a word *are a function of* (can be approximated by) *the set of contexts in which it occurs in texts.*
He filled the *wampimuk*, passed it around and we all drunk some

We found a little, hairy *wampimuk* sleeping behind the tree
Distributional lexical semantics

- Distributional analysis in structuralist linguistics (Zellig Harris), British corpus linguistics (J.R. Firth), psychology (Miller & Charles), but not only

  “[T]he semantic properties of a lexical item are fully reflected in appropriate aspects of the relations it contracts with actual and potential contexts [...] [T]here are good reasons for a principled limitation to linguistic contexts” (Cruse 1986)

- Distributional hypothesis suggests that we can induce (aspects of the) meaning of words from texts

- This is its biggest selling point in computational linguistics: it is a “theory of meaning” that can be easily operationalized into a procedure to extract “meaning” from text corpora on a large scale
The distributional hypothesis, weak and strong
Lenci (2008)

- Weak: a quantitative method for semantic analysis and lexical resource induction
- Strong: A cognitive hypothesis about the form and origin of semantic representations
Distributional semantic models (DSMs)

Narrowing the field

- Idea of using corpus-based statistics to extract information about semantic properties of words and other linguistic units is extremely common in computational linguistics
- Here, we focus on models that:
  - Represent the meaning of words as *vectors* keeping track of the words’ distributional history
  - Focus on the notion of *semantic similarity*, measured with geometrical methods in the *space* inhabited by the distributional vectors
  - Are intended as *general-purpose* semantic models that are estimated once, and then used for various semantic tasks, and not created ad-hoc for a specific goal
    - It follows that model estimation phase is typically unsupervised
- Aka: vector/word space models, semantic spaces
Advantages of distributional semantics

Distributional semantic models are

- model of inductive *learning* for word meaning
- radically empirical
- rich
- flexible
- cheap, scalable
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Constructing the models

- Pre-process the source corpus
- Collect a co-occurrence matrix (with *distributional vectors* representing words as rows, and contextual elements of some kind as columns/dimensions)
- Transform the matrix: re-weighting raw frequencies, dimensionality reduction
- Use resulting matrix to compute word-to-word similarity
Corpus pre-processing

- Minimally, corpus must be tokenized
- POS tagging, lemmatization, dependency parsing…
- Trade-off between deeper linguistic analysis and
  - need for language-specific resources
  - possible errors introduced at each stage of the analysis
  - more parameters to tune
Distributional vectors

- Count how many times each target word occurs in a certain context
- Build vectors out of (a function of) these context occurrence counts
- Similar words will have similar vectors
Collecting context counts for target word dog

The dog barked in the park.
The owner of the dog put him on the leash since he barked.

<table>
<thead>
<tr>
<th></th>
<th>bark</th>
<th>park</th>
<th>owner</th>
<th>leash</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
The dog *barked* in the park. The owner of the dog put him on the leash since he barked.
Collecting context counts for target word dog

The dog barked in the park.
The owner of the dog put him on the leash since he barked.

<table>
<thead>
<tr>
<th>bark</th>
<th>park</th>
<th>owner</th>
<th>leash</th>
</tr>
</thead>
<tbody>
<tr>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Collecting context counts for target word **dog**

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>bark</td>
<td>++</td>
</tr>
<tr>
<td>park</td>
<td>+</td>
</tr>
<tr>
<td>owner</td>
<td>+</td>
</tr>
<tr>
<td>leash</td>
<td>+</td>
</tr>
</tbody>
</table>

The dog barked in the park. The **owner** of the **dog** put him on the leash since he barked.
Collecting context counts for target word dog

The dog barked in the park. 
The owner of the **dog** put him on the **leash** since he barked.

<table>
<thead>
<tr>
<th>word</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>bark</td>
<td>++</td>
</tr>
<tr>
<td>park</td>
<td>+</td>
</tr>
<tr>
<td>owner</td>
<td>+</td>
</tr>
<tr>
<td>leash</td>
<td>+</td>
</tr>
</tbody>
</table>
Collecting context counts for target word dog

The dog barked in the park.
The owner of the dog put him on the leash since he barked.

<table>
<thead>
<tr>
<th></th>
<th>bark</th>
<th>park</th>
<th>owner</th>
<th>leash</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
The co-occurrence matrix

<table>
<thead>
<tr>
<th></th>
<th>leash</th>
<th>walk</th>
<th>run</th>
<th>owner</th>
<th>pet</th>
<th>bark</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>cat</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>lion</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>light</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bark</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>car</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
What is “context”?

DOC1: 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”? 

Documents

**DOC1**: 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”? All words in a wide window

**DOC1**: 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”? Content words only

DOC1: 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”?  
Content words in a narrower window

DOC1: 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”?  
POS-coded content lemmas

DOC1: 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”? POS-coded content lemmas filtered by syntactic path to the target

DOC1: 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”?  
...with the syntactic path encoded as part of the context

DOC1: 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
General trends in “context engineering”

- In computational linguistics, tendency towards using more linguistically aware contexts, but “jury is still out” on their utility (Sahlgren, 2008)
  - This is at least in part task-specific
- In cognitive science trend towards broader document-/text-based contexts
  - Focus on topic detection, gist extraction, text coherence assessment, library science
  - Latent Semantic Analysis (Landauer & Dumais, 1997), Topic Models (Griffiths et al., 2007)
Contexts and dimensions
Some terminology I will use below

- **Dependency-filtered** (e.g., Padó & Lapata, 2007) vs. **dependency-linked** (e.g., Grefenstette 1994, Lin 1998, Curran & Moens 2002, Baroni and Lenci 2010)
- Both rely on output of dependency parser to identify context words that are connected to target words by interesting relations
- However, only dependency-linked models keep (parts of) the dependency path connecting target word and context word in the dimension label
Some terminology I will use below

Given input sentence: *The dog bites the postman on the street*

- both approaches might consider only *bite* as a context element for both *dog* and *postman* (because they might focus on *subj-of* and *obj-of* relations only)
- However, a dependency-filtered model will count *bite* as identical context in both cases
- whereas a dependency-linked model will count *subj-of-bite* as context of *dog* and *obj-of-bite* as context of *postman* (so, *different* contexts for the two words)
The distributional semantic framework is general enough that feature vectors can come from other sources as well, besides from corpora (or from a mixture of sources).

Obvious alternative/complementary sources are dictionaries, structured knowledge bases such as WordNet.

I am particularly interested in the possibility of merging features from text and images (“visual words”: Feng and Lapata 2010, Bruni et al. 2011, 2012).
Context weighting

- Raw context counts typically transformed into scores
- In particular, association measures to give more weight to contexts that are more significantly associated with a target word
- General idea: the less frequent the target word and (more importantly) the context element are, the higher the weight given to their observed co-occurrence count should be (because their expected chance co-occurrence frequency is low)
  - Co-occurrence with frequent context element *time* is less informative than co-occurrence with rarer *tail*
- Different measures – e.g., Mutual Information, Log Likelihood Ratio – differ with respect to how they balance raw and expectation-adjusted co-occurrence frequencies
  - Positive Point-wise Mutual Information widely used and pretty robust
Measures from information retrieval that take distribution over documents into account are also used. The basic idea is that terms that tend to occur in a few documents are more interesting than generic terms that occur all over the place.
Dimensionality reduction

- Reduce the target-word-by-context matrix to a lower dimensionality matrix (a matrix with less – linearly independent – columns/dimensions)

- Two main reasons:
  - Smoothing: capture “latent dimensions” that generalize over sparser surface dimensions (Singular Value Decomposition or SVD)
  - Efficiency/space: sometimes the matrix is so large that you don’t even want to construct it explicitly (Random Indexing)
Singular Value Decomposition

- General technique from linear algebra (essentially, the same as Principal Component Analysis, PCA)
  - Some alternatives: Independent Component Analysis, Non-negative Matrix Factorization
- Given a matrix (e.g., a word-by-context matrix) of \(m \times n\) dimensionality, construct a \(m \times k\) matrix, where \(k << n\) (and \(k < m\))
  - E.g., from a 20,000 words by 10,000 contexts matrix to a 20,000 words by 300 “latent dimensions” matrix
  - \(k\) is typically an arbitrary choice
- From linear algebra, we know that and how we can find the reduced \(m \times k\) matrix with orthogonal dimensions/columns that preserves most of the variance in the original matrix
Preserving variance

![Scatter plot with data points and axes labeled as dimension 1 and dimension 2. The variance is labeled as 1.26.]
Preserving variance
Preserving variance

\[
\begin{align*}
\text{variance} &= 0.36 \\
\end{align*}
\]
Preserving variance
Preserving variance

\[
\text{variance} = 0.72
\]
Preserving variance
Preserving variance

\[ \text{variance} = 0.9 \]
Dimensionality reduction as generalization

<table>
<thead>
<tr>
<th></th>
<th>buy</th>
<th>sell</th>
<th>dim1</th>
</tr>
</thead>
<tbody>
<tr>
<td>wine</td>
<td>31.2</td>
<td>27.3</td>
<td>41.3</td>
</tr>
<tr>
<td>beer</td>
<td>15.4</td>
<td>16.2</td>
<td>22.3</td>
</tr>
<tr>
<td>car</td>
<td>40.5</td>
<td>39.3</td>
<td>56.4</td>
</tr>
<tr>
<td>cocaine</td>
<td>3.2</td>
<td>22.3</td>
<td>18.3</td>
</tr>
</tbody>
</table>
The Singular Value Decomposition

Any $m \times n$ real-valued matrix $A$ can be factorized into 3 matrices $U\Sigma V^T$

- $U$ is a $m \times m$ orthogonal matrix ($UU^T = I$)
- $\Sigma$ is a $m \times n$ diagonal matrix, with diagonal values ordered from largest to smallest ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, where $r = \min(m, n)$)
- $V$ is a $n \times n$ orthogonal matrix ($VV^T = I$)
The Singular Value Decomposition

\[
\begin{pmatrix}
  u_{11} & u_{12} & \cdots & u_{1m} \\
  u_{21} & u_{22} & \cdots & u_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  u_{m1} & u_{m2} & \cdots & u_{mm}
\end{pmatrix}
\times
\begin{pmatrix}
  \sigma_1 & 0 & 0 & \cdots \\
  0 & \sigma_2 & 0 & \cdots \\
  0 & 0 & \sigma_3 & \cdots \\
  \vdots & \vdots & \vdots & \ddots \\
  \vdots & \vdots & \vdots & \cdots & \ddots 
\end{pmatrix}
\times
\begin{pmatrix}
  v_{11} & v_{21} & \cdots & v_{n1} \\
  v_{12} & v_{22} & \cdots & v_{n2} \\
  \vdots & \vdots & \ddots & \vdots \\
  v_{1n} & v_{2n} & \cdots & v_{nn}
\end{pmatrix}
\]
The Singular Value Decomposition
Projecting the $A$ row vectors onto the new coordinate system

$$A_{m\times n} = U_{m\times m} \Sigma_{m\times n} V_{n\times n}^T$$

- The columns of the orthogonal $V_{n\times n}$ matrix constitute a basis (coordinate system, set of axes or dimensions) for the $n$-dimensional row vectors of $A$
- The projection of a row vector $a_j$ onto axis column $v_i$ (i.e., the $v_i$ coordinate of $a_j$) is given by $a_j \cdot v_i$
- The coordinates of $a_j$ in the full $V$ coordinate system are thus given by $a_j V$, and generalizing the coordinates of all vectors projected onto the new system are given by $AV$
- $AV = U \Sigma V^T V = U \Sigma$
Reducing dimensionality

- Projecting $A$ onto the new $V$ coordinate system:
  \[ AV = U\Sigma \]

- It can be shown that, when the $A$ row vectors are represented in this new set of coordinates, variance on each $v_i$-axis is proportional to $\sigma_i^2$ (the square of the $i$-th value on the diagonal of $\Sigma$)
  - Intuitively: $U$ and $V$ are orthogonal, all the “stretching” when multiplying the matrices is done by $\Sigma$

- Given that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, if we take the coordinates on the first $k$ axes, we obtain lower dimensionality vectors that account for the maximum proportion of the original variance that we can account for with $k$ dimensions

- I.e., we compute the “truncated” projection:
  \[ A_{m \times n} V_{n \times k} = U_{m \times k} \Sigma_{k \times k} \]
The Singular Value Decomposition
Finding the component matrices

▶ Don’t try this at home!
▶ SVD draw on non-efficient operations
▶ Fortunately, there are out-of-the-box packages to compute SVD, a popular one being SVDPACK, that I use via SVDLIBC (http://tedlab.mit.edu/~dr/svdlibc/)
▶ Recently, various mathematical developments and packages to compute SVD incrementally, scaling up to very very large matrices, see e.g.: http://radimrehurek.com/gensim/
▶ See:
  http://wordspace.collocations.de/doku.php/course:esslli2009:start
▶ Very clear introduction to SVD (and PCA), with all the mathematical details I skipped here
SVD: Pros and cons

- **Pros:**
  - Good performance (in most cases)
  - At least some indication of robustness against data sparseness
  - Smoothing as generalization
  - Smoothing also useful to generalize features to words that do not co-occur with them in the corpus (e.g., spreading visually-derived features to all words)
  - Words and contexts in the same space (contexts not trivially orthogonal to each other)

- **Cons:**
  - Non-incremental (even incremental implementations allow you to add new rows, not new columns)
    - Of course, you can use $V_{n \times k}$ to project new vectors onto the same reduced space!
  - Latent dimensions are difficult to interpret
  - Does not scale up well (but see recent developments...)
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
## Contexts as vectors

<table>
<thead>
<tr>
<th></th>
<th>runs</th>
<th>legs</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>car</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Semantic space

- car (4,0)
- dog (1,4)
- cat (1,5)
Semantic similarity as angle between vectors

cat (1,5)
dog (1,4)
car (4,0)

legs

runs
Measuring angles by computing cosines

- Cosine is most common similarity measure in distributional semantics, and the most sensible one from a geometrical point of view.
- Ranges from 1 for parallel vectors (perfectly correlated words) to 0 for orthogonal (perpendicular) words/vectors.
  - It goes to -1 for parallel vectors pointing in opposite directions (perfectly inversely correlated words), as long as weighted co-occurrence matrix has negative values.
- (Angle is obtained from cosine by applying the *arc-cosine* function, but it is rarely used in computational linguistics.)
Trigonometry review

- Build a right triangle by connecting the two vectors
- Cosine is ratio of length of side adjacent to measured angle to length of hypotenuse side
- If we build triangle so that hypotenuse has length 1, cosine will equal length of adjacent side (because we divide by 1)
- I.e., in this case cosine is length of *projection* of hypotenuse on the adjacent side
Computing the cosines: preliminaries
Length and dot products

Length of a vector $\mathbf{v}$ with $n$ dimensions $v_1, v_2, \ldots, v_n$ (Pythagoras’ theorem!)

$||\mathbf{v}|| = \sqrt{\sum_{i=1}^{n} v_i^2}$
Computing the cosines: preliminaries

Orthogonal vectors

- The dot product of two orthogonal (perpendicular) vectors is 0
- To see this, note that given two vectors \( \mathbf{v} \) and \( \mathbf{w} \) forming a right angle, Pythagoras’ theorem says that
  \[
  ||\mathbf{v}||^2 + ||\mathbf{w}||^2 = ||\mathbf{v} - \mathbf{w}||^2
  \]
- But:
  \[
  ||\mathbf{v} - \mathbf{w}||^2 = \sum_{i=1}^{n} (v_i - w_i)^2 = \sum_{i=1}^{n} (v_i^2 - 2v_iw_i + w_i^2) = \\
  \sum_{i=1}^{n} v_i^2 - \sum_{i=1}^{n} 2v_iw_i + \sum_{i=1}^{n} w_i^2 = ||\mathbf{v}||^2 - 2\mathbf{v} \cdot \mathbf{w} + ||\mathbf{w}||^2
  \]
- So, for the Pythagoras’ theorem equality to hold, \( \mathbf{v} \cdot \mathbf{w} = 0 \)
Computing the cosine

- $||a|| = ||b|| = 1$
- $c = \rho b$
- $e = c - a; e \cdot b = 0$
- $(c - a) \cdot b = c \cdot b - a \cdot b = 0$
- $c \cdot b = \rho b \cdot b = \rho = a \cdot b$
- $||c|| = ||\rho b|| = \sqrt{\rho^2 b \cdot b} = \rho = a \cdot b$
Computing the cosine

- For two vectors of length 1, the cosine is given by:
  \[ ||\mathbf{c}|| = \mathbf{a} \cdot \mathbf{b} \]

- If the two vectors are not of length 1 (as it will be typically the case in DSMs), we obtain vectors of length 1 pointing in the same directions by dividing the original vectors by their lengths, obtaining:

\[
||\mathbf{c}|| = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| ||\mathbf{b}||} = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} a^2} \times \sqrt{\sum_{i=1}^{n} b^2}}
\]
Computing the cosine

Example

\[
\sum_{i=1}^{n} a_i \times b_i
\]
\[
\sqrt{\sum_{i=1}^{n} a^2 \times \sqrt{\sum_{i=1}^{n} b^2}}
\]

<table>
<thead>
<tr>
<th></th>
<th>runs</th>
<th>legs</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>cat</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>car</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

\[\text{cosine}(\text{dog,cat}) = \frac{(1 \times 1) + (4 \times 5)}{\sqrt{1^2 + 4^2} \times \sqrt{1^2 + 5^2}} = 0.9988681\]

\[\text{arc-cosine}(0.9988681) = 2.72\,\text{degrees}\]

\[\text{cosine}(\text{dog,car}) = \frac{(1 \times 4) + (4 \times 0)}{\sqrt{1^2 + 4^2} \times \sqrt{4^2 + 0^2}} = 0.2425356\]

\[\text{arc-cosine}(0.2425356) = 75.85\,\text{degrees}\]
Computing the cosine

Example

\[
\begin{array}{|c|c|}
\hline
\text{runs} & \text{legs} \\
\hline
0 & 0 \\
1 & 1 \\
2 & 2 \\
3 & 3 \\
4 & 4 \\
5 & 5 \\
6 & 6 \\
\hline
\end{array}
\]

- car (4,0)
- dog (1,4)
- cat (1,5)

- 75.85 degrees
- 2.72 degrees

\[
\frac{54}{121}
\]
Cosine intuition

- When computing the cosine, the values that two vectors have for the same dimensions (coordinates) are multiplied.
- Two vectors/words will have a high cosine if they tend to have high same-sign values for the same dimensions/contexts.
- If we center the vectors so that their mean value is 0, the cosine of the centered vectors is the same as the Pearson correlation coefficient.
- If, as it is often the case in computational linguistics, we have only nonnegative scores, and we do not center the vectors, then the cosine can only take nonnegative values, and there is no “canceling out” effect.
  - As a consequence, cosines tend to be higher than the corresponding correlation coefficients.
Other measures

- Cosines are well-defined, well understood way to measure similarity in a vector space.
- Euclidean distance (length of segment connecting end-points of vectors) is equally principled, but length-sensitive (two vectors pointing in the same direction will be very distant if one is very long, the other very short).
- Other measures based on other, often non-geometric principles (Lin’s information theoretic measure, Kullback/Leibler divergence...) bring us outside the scope of vector spaces, and their application to semantic vectors can be iffy and ad-hoc.
Outline

Introduction: The distributional hypothesis
Constructing the models
Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Recap: Constructing the models

- Pre-process the source corpus
- Collect a co-occurrence matrix (with *distributional vectors* representing words as rows, and contextual elements of some kind as columns/dimensions)
- Transform the matrix: re-weighting raw frequencies, dimensionality reduction
- Use resulting matrix to compute word-to-word similarity
Distributional similarity as semantic similarity

▶ Developers of DSMs typically want them to be “general-purpose” models of semantic similarity.
▶ These models emphasize paradigmatic similarity, i.e., words that tend to occur in the same contexts.
▶ Words that share many contexts will correspond to concepts that share many attributes (attributional similarity), i.e., concepts that are taxonomically similar:
  ▶ Synonyms (rhino/rhinoceros), antonyms and values on a scale (good/bad), co-hyponyms (rock/jazz), hyper- and hyponyms (rock/basalt).
▶ Taxonomic similarity is seen as the fundamental semantic relation, allowing categorization, generalization, inheritance.
▶ Evaluation focuses on tasks that measure taxonomic similarity.
Distributional semantics as models of word meaning
Landauer and Dumais 1997, Turney and Pantel 2010, Baroni and Lenci 2010

Distributional semantics can model
- human similarity judgments (*cord-string* vs. *cord-smile*)
- lexical priming (*hospital* primes *doctor*)
- synonymy (*zenith-pinnacle*)
- analogy (*mason* is to *stone* like *carpenter* is to *wood*)
- relation classification (*exam-anxiety*: CAUSE-EFFECT)
- text coherence
- . . .
The main problem with evaluation: Parameter Hell!

- So many parameters in tuning the models:
  - input corpus, context, counting, weighting, matrix manipulation, similarity measure
- With interactions (Erk & Padó, 2009, and others)
- And best parameters in a task might not be the best for another
- No way we can experimentally explore the parameter space
  - But see work by Bullinaria and colleagues for some systematic attempt
Nearest neighbour examples
BNC, 2-content-word-window context

<table>
<thead>
<tr>
<th>rhino</th>
<th>fall</th>
<th>rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>woodpecker</td>
<td>rise</td>
<td>lava</td>
</tr>
<tr>
<td>rhinoceros</td>
<td>increase</td>
<td>sand</td>
</tr>
<tr>
<td>swan</td>
<td>fluctuation</td>
<td>boulder</td>
</tr>
<tr>
<td>whale</td>
<td>drop</td>
<td>ice</td>
</tr>
<tr>
<td>ivory</td>
<td>decrease</td>
<td>jazz</td>
</tr>
<tr>
<td>plover</td>
<td>reduction</td>
<td>slab</td>
</tr>
<tr>
<td>elephant</td>
<td>logarithm</td>
<td>cliff</td>
</tr>
<tr>
<td>bear</td>
<td>decline</td>
<td>pop</td>
</tr>
<tr>
<td>satin</td>
<td>cut</td>
<td>basalt</td>
</tr>
<tr>
<td>sweatshirt</td>
<td>hike</td>
<td>crevice</td>
</tr>
</tbody>
</table>
### Nearest neighbour examples

**BNC, 2-content-word-window context**

<table>
<thead>
<tr>
<th>green</th>
<th>good</th>
<th>sing</th>
</tr>
</thead>
<tbody>
<tr>
<td>blue</td>
<td>bad</td>
<td>dance</td>
</tr>
<tr>
<td>yellow</td>
<td>excellent</td>
<td>whistle</td>
</tr>
<tr>
<td>brown</td>
<td>superb</td>
<td>mime</td>
</tr>
<tr>
<td>bright</td>
<td>poor</td>
<td>shout</td>
</tr>
<tr>
<td>emerald</td>
<td>improved</td>
<td>sound</td>
</tr>
<tr>
<td>grey</td>
<td>perfect</td>
<td>listen</td>
</tr>
<tr>
<td>speckled</td>
<td>clever</td>
<td>recite</td>
</tr>
<tr>
<td>greenish</td>
<td>terrific</td>
<td>play</td>
</tr>
<tr>
<td>purple</td>
<td>lucky</td>
<td>hear</td>
</tr>
<tr>
<td>gleaming</td>
<td>smashing</td>
<td>hiss</td>
</tr>
</tbody>
</table>
Some classic semantic similarity tasks

- Taking the TOEFL: synonym identification
- The Rubenstein/Goodenough norms: modeling semantic similarity judgments
- The Hodgson semantic priming data
The TOEFL synonym match task

- 80 items
- Target: *levied*
  Candidates: *imposed, believed, requested, correlated*
- In semantic space, measure angles between target and candidate context vectors, pick candidate that forms most narrow angle with target
The TOEFL synonym match task

- 80 items
- Target: *levied*
  - Candidates: *imposed, believed, requested, correlated*
- In semantic space, measure angles between target and candidate context vectors, pick candidate that forms most narrow angle with target
The TOEFL synonym match task

- 80 items
- Target: *levied*
  Candidates: *imposed, believed, requested, correlated*
- In semantic space, measure angles between target and candidate context vectors, pick candidate that forms most narrow angle with target
Human performance on the synonym match task

- Average foreign test taker: 64.5%
- Macquarie University staff (Rapp 2004):
  - Average of 5 non-natives: 86.75%
  - Average of 5 natives: 97.75%
Distributional Semantics takes the TOEFL

- **Humans:**
  - Foreign test takers: 64.5%
  - Macquarie non-natives: 86.75%
  - Macquarie natives: 97.75%

- **Machines:**
  - Classic LSA: 64.4%
  - Padó and Lapata’s dependency-filtered model: 73%
  - Rapp’s 2003 SVD-based model trained on lemmatized BNC: 92.5%

- Direct comparison in Baroni and Lenci 2010 (ukWaC+Wikipedia+BNC as training data, local MI weighting):
  - Dependency-filtered: 76.9%
  - Dependency-linked: 75.0%
  - Co-occurrence window: 69.4%
Rubenstein & Goodenough (1965)

- (Approximately) continuous similarity judgments
- 65 noun pairs rated by 51 subjects on a 0-4 similarity scale and averaged
  - E.g.: car-automobile 3.9; food-fruit 2.7; cord-smile 0.0
- (Pearson) correlation between cosine of angle between pair context vectors and the judgment averages
- State-of-the-art results:
  - Herdağdelen et al. (2009) using SVD-ed dependency-filtered model estimated on ukWaC: 80%
- Direct comparison in Baroni et al.’s experiments:
  - Co-occurrence window: 65%
  - Dependency-filtered: 57%
  - Dependency-linked: 57%
Semantic priming

- Hearing/reading a “related” prime facilitates access to a target in various lexical tasks (naming, lexical decision, reading...

- You recognize/access the word *pear* faster if you just heard/read *apple*

- Hodgson (1991) single word lexical decision task, 136 prime-target pairs
  - (I have no access to original article, rely on McDonald & Brew 2004 and Padó & Lapata 2007)
Hodgson found similar amounts of priming for different semantic relations between primes and targets (approx. 23 pairs per relation):

- synonyms (synonym): *to dread/to fear*
- antonyms (antonym): *short/tall*
- coordinates (coord): *train/truck*
- super- and subordinate pairs (supersub): *container/bottle*
- free association pairs (freeass): *dove/peace*
- phrasal associates (phrasacc): *vacant/building*
Simulating semantic priming
Methodology from McDonald & Brew, Padó & Lapata

- For each related prime-target pair:
  - measure cosine-based similarity between pair elements (e.g., *to dread/to fear*)
  - take average of cosine-based similarity of target with other primes from same relation data-set (e.g., *to value/to fear*) as measure of similarity of target with unrelated items
- Similarity between related items should be significantly higher than average similarity between unrelated items
Semantic priming results

- T-normalized differences between related and unrelated conditions (* <0.05, ** <0.01, according to paired t-tests)
- Results from Herdağdelen et al. (2009) based on SVD-ed dependency-filtered corpus, but similar patterns reported by McDonald & Brew and Padó & Lapata

<table>
<thead>
<tr>
<th>relation</th>
<th>pairs</th>
<th>t-score</th>
<th>sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>synonym</td>
<td>23</td>
<td>10.015</td>
<td>**</td>
</tr>
<tr>
<td>antonym</td>
<td>24</td>
<td>7.724</td>
<td>**</td>
</tr>
<tr>
<td>coord</td>
<td>23</td>
<td>11.157</td>
<td>**</td>
</tr>
<tr>
<td>supersub</td>
<td>21</td>
<td>10.422</td>
<td>**</td>
</tr>
<tr>
<td>freeass</td>
<td>23</td>
<td>9.299</td>
<td>**</td>
</tr>
<tr>
<td>phrasacc</td>
<td>22</td>
<td>3.532</td>
<td>*</td>
</tr>
</tbody>
</table>
Distributional semantics in complex NLP systems and applications

- Document-by-word models have been used in Information Retrieval for decades
  - DSMs might be pursued in IR within the broad topic of “semantic search”
- Commercial use for automatic essay scoring and other language evaluation related tasks
  - http://lsa.colorado.edu
Distributional semantics in complex NLP systems and applications

- Elsewhere, general-purpose DSMs not too common, nor too effective:
  - Lack of reliable, well-known out-of-the-box resources comparable to WordNet
  - “Similarity” is too vague a notion for well-defined semantic needs (cf. nearest neighbour lists above)
- However, there are more-or-less successful attempts to use general-purpose distributional semantic information at least as supplementary resource in various domains, e.g.:
  - Question answering (Tómas & Vicedo, 2007)
  - Bridging coreference resolution (Poesio et al., 1998, Versley, 2007)
  - Language modeling for speech recognition (Bellegarda, 1997)
  - Textual entailment (Zhitomirsky-Geffet and Dagan, 2009)
Distributional semantics in the humanities, social sciences, cultural studies

▶ Great potential, only partially explored
▶ E.g., Sagi et al. (2009a,b) use distributional semantics to study
  ▶ semantic broadening (dog from specific breed to “generic canine”) and narrowing (deer from “animal” to “deer”) in the history of English
  ▶ phonastemes (glance and gleam, growl and howl)
  ▶ the parallel evolution of British and American literature over two centuries
“Culture” in distributional space
Nearest neighbours in BNC-estimated model

woman

- gay
- homosexual
- lesbian
- bearded
- burly
- macho
- sexually
- man
- stocky
- to castrate

man

- policeman
- girl
- promiscuous
- woman
- compositor
- domesticity
- pregnant
- chastity
- ordination
- warrior
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Distributional semantics
Distributional meaning as co-occurrence vector

<table>
<thead>
<tr>
<th></th>
<th>planet</th>
<th>night</th>
<th>full</th>
<th>shadow</th>
<th>shine</th>
<th>crescent</th>
</tr>
</thead>
<tbody>
<tr>
<td>moon</td>
<td>10</td>
<td>22</td>
<td>43</td>
<td>16</td>
<td>29</td>
<td>12</td>
</tr>
<tr>
<td>sun</td>
<td>14</td>
<td>10</td>
<td>4</td>
<td>15</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## Distributional semantics

Distributional meaning as co-occurrence vector

<table>
<thead>
<tr>
<th></th>
<th>x729</th>
<th>x145</th>
<th>x684</th>
<th>x776</th>
<th>x998</th>
<th>x238</th>
</tr>
</thead>
<tbody>
<tr>
<td>moon</td>
<td>10</td>
<td>22</td>
<td>43</td>
<td>16</td>
<td>29</td>
<td>12</td>
</tr>
<tr>
<td>sun</td>
<td>14</td>
<td>10</td>
<td>4</td>
<td>15</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The symbol grounding problem
Interpretation vs. translation
Searle 1980, Harnad 1990

红, 紅
1. 像火或鲜血那样的颜色: "红枣 | 红日 | 面红耳赤。".
2. 借指红色的东西: "落红（指花） | 披红戴花（指红色织物）。".

google.com, “define” functionality
Cognitive Science: Word meaning is grounded
Barsalou 2008, Kiefer and Pulvermüller 2011 (overviews)
红, 红

1. 像火或鲜血那样的颜色: "红枣 | 红日 | 面红耳赤。".
2. 借指红色的东西: "落红（指花） | 披红戴花（指红色织物）。".

google.com, “define” functionality
Interpretation with perception

google.com
Classical distributional models are not grounded

Image credit: Jiming Li
Classical distributional models are not grounded

Describing tigers...

Humans (McRae et al., 2005):
- have stripes
- have teeth
- are black
- ...

State-of-the-art distributional model (Baroni et al., 2010):
- live in jungle
- can kill
- risk extinction
- ...
The meaning of a word is (can be approximated via) the set of contexts in which it occurs
Grounding distributional semantics
Multimodal models using textual and visual collocates
Bruni et al. JAIR 2014, Leong and Mihalchea IJCNLP 2011, Silberer et al. ACL 2013

<table>
<thead>
<tr>
<th></th>
<th>planet</th>
<th>night</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>moon</td>
<td>10</td>
<td>22</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>sun</td>
<td>14</td>
<td>10</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>
Multimodal models with images

words

concepts

perception

vision

action

introspection
Multimodal models

- other modalities: feature norms (Andrews et al. 2010, Roller and Schulte im Walde EMNLP 2013)
  - feature norms: *tiger* - *has stripes*...
  - manually collected...
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Bags of visual words

Motivation
Detection and description

- **Detection**: Identify the interest points, e.g. with **Harris corner detectors**

- **Description**: Extract feature vector describing area surrounding each interest point, e.g. **SIFT descriptor**

\[ \mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}] \]

\[ \mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}] \]

[Fei-Fei Li]
Visual codeword dictionary formation by clustering

Clustering / vector quantization

Cluster center = code word

Clustering / vector quantization

[Fei-Fei Li]
Vector mapping

• Nearest neighbors assignment
• K-D tree search strategy

Lecture 15 - Fei-Fei Li
Counting

[Fei-Fei Li]
Spatial pyramid representation
Lazebnik, Schmid, and Ponce, 2006, 2009
Empirical assessment
Feng and Lapata 2010

Michelle Obama fever hits the UK

In the UK on her first visit as first lady, Michelle Obama seems to be making just as big an impact. She has attracted as much interest and column inches as her husband on this London trip; creating a buzz with her dazzling outfits, her own schedule of events and her own fanbase. Outside Buckingham Palace, as crowds gathered in anticipation of the Obamas’ arrival, Mrs Obama’s star appeal was apparent.

Feng and Lapata 2010: Model learns from mixed-media documents a joint word+visual-word Topic Model

<table>
<thead>
<tr>
<th>Model</th>
<th>Word Association</th>
<th>Word Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpperBnd</td>
<td>0.400</td>
<td>0.545</td>
</tr>
<tr>
<td>MixLDA</td>
<td>0.123</td>
<td>0.318</td>
</tr>
<tr>
<td>TxtLDA</td>
<td>0.077</td>
<td>0.247</td>
</tr>
</tbody>
</table>
Empirical assessment
Bruni et al. ACL 2012, also see Bruni et al. JAIR 2014

- Bruni et al. ACL 2012: textual and visual vectors concatenated
- multimodal better at general word similarity – 0.66 vs. 0.69 (MEN dataset)
- multimodal better at modeling the meaning of color terms
  - a banana is yellow: multimodal gets 27/52 right, text only 13
  - literal vs. non-literal uses of color terms:
    - a blue uniform is blue, a blue note is not
    - text .53, multimodal .73 (complicated metric)
- more sophisticated combination of textual and visual information yields further improvements (Bruni et al. JAIR 2014)
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
The infinity of sentence meaning...
Compositionality

The meaning of an utterance is a function of the meaning of its parts and their composition rules (Frege 1892)

“gingerbread gnomes dance under the red moon”
A compositional distributional semantics for phrases and sentences?

<table>
<thead>
<tr>
<th>phrase</th>
<th>planet</th>
<th>night</th>
<th>full</th>
<th>blood</th>
<th>shine</th>
</tr>
</thead>
<tbody>
<tr>
<td>moon</td>
<td>10</td>
<td>22</td>
<td>43</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>red moon</td>
<td>12</td>
<td>21</td>
<td>40</td>
<td>20</td>
<td>28</td>
</tr>
<tr>
<td>the red moon shines</td>
<td>11</td>
<td>23</td>
<td>21</td>
<td>15</td>
<td>45</td>
</tr>
</tbody>
</table>
Outline

Introduction: The distributional hypothesis
Constructing the models
Semantic similarity as geometric distance
Evaluation
Multimodal distributional models
  Computer vision
Compositionality
  Why?
  How?
Conclusion
The unavoidability of distributional representations of phrases
What can you do with distributional representations of phrases and sentences?
Paraphrasing

Mitchell and Lapata 2010
What can you do with distributional representations of phrases and sentences?
Disambiguation

- the cucumber is rotten
- the cucumber is old
- the cucumber is ancient

Mitchell and Lapata 2008
What can you do with distributional representations of phrases and sentences? 
Semantic acceptability

Vecchi, Baroni and Zamparelli 2011
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Compositional distributional semantics

## Additive model

Mitchell and Lapata 2010, ...

<table>
<thead>
<tr>
<th></th>
<th>planet</th>
<th>night</th>
<th>blood</th>
<th>brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>15</td>
<td>3</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>moon</td>
<td>24</td>
<td>15</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>red+moon</td>
<td>39</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$0.4 \times \text{red} + 0.6 \times \text{moon}$</td>
<td>20.4</td>
<td>10.2</td>
<td>8.2</td>
<td>8</td>
</tr>
</tbody>
</table>

Weighted additive model: $\vec{p} = \alpha \vec{a} + \beta \vec{n}$
Additive model
Mitchell and Lapata 2010

<table>
<thead>
<tr>
<th></th>
<th>planet</th>
<th>night</th>
<th>blood</th>
<th>brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>15</td>
<td>3</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>moon</td>
<td>24</td>
<td>15</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>red + moon</td>
<td>39</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>0.4 × red + 0.6 × moon</td>
<td>20.4</td>
<td>10.2</td>
<td>8.2</td>
<td>8</td>
</tr>
</tbody>
</table>

weighted additive model: \[ \tilde{p} = \alpha \tilde{a} + \beta \tilde{n} \]
Additive model
Mitchell and Lapata 2010, ...

<table>
<thead>
<tr>
<th></th>
<th>planet</th>
<th>night</th>
<th>blood</th>
<th>brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>15</td>
<td>3</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>moon</td>
<td>24</td>
<td>15</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>red+moon</td>
<td>39</td>
<td>18</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>0.4×red + 0.6×moon</td>
<td>20.4</td>
<td>10.2</td>
<td>8.2</td>
<td>8</td>
</tr>
</tbody>
</table>

weighted additive model: \( \vec{p} = \alpha \vec{a} + \beta \vec{n} \)
Composition as (distributional) function application

Grefenstette, Sadrzadeh et al., Baroni and Zamparelli, Socher et al.

\[
\text{moon} = \begin{bmatrix}
\text{shine} & 301 \\
\text{eclipse} & 250 \\
\text{blood} & 93 \\
\ldots & \ldots
\end{bmatrix}
\]

\[
\text{red(moon)} = \begin{bmatrix}
\text{shine} & 11 \\
\text{eclipse} & 245 \\
\text{blood} & 90 \\
\ldots & \ldots
\end{bmatrix}
\]
Baroni and Zamparelli’s 2010 proposal

Implementing the idea of function application in a vector space

- Functions as **linear maps** between vector spaces
- Functions are matrices, function application is function-by-vector multiplication

\[ \vec{p} = \mathbf{A} \vec{n} \]
Baroni and Zamparelli’s 2010 proposal

Implementing the idea of function application in a vector space

- Functions as **linear maps** between vector spaces
- Functions are matrices, function application is function-by-vector multiplication
Baroni and Zamparelli’s 2010 proposal

Implementing the idea of function application in a vector space

▶ Functions as **linear maps** between vector spaces
▶ Functions are matrices, function application is function-by-vector multiplication

\[ \vec{p} = A \vec{n} \]

**lexical function model:**
Learning distributional composition functions

n and the moon shining i
with the moon shining s
rainbowed moon. And the
crescent moon, thrille
in a blue moon only, wi
now, the moon has risen
d now the moon rises, f
y at full moon, get up
crescent moon. Mr Angu

f a large red moon, Campana
, a blood red moon hung over
glorious red moon turning t
The round red moon, she’s
l a blood red moon emerged f
n rains, red moon blows, w
monstrous red moon had climb
. A very red moon rising is
under the red moon a vampire
Addition and lexical function
as models of adjective meaning
Addition and lexical function as models of adjective meaning

\[
\text{red} \quad \begin{array}{c|c|c|c|c} \hline \\
\hline \\
\end{array} \\
\text{moon} \quad \begin{array}{c|c|c} \hline \\
\hline \\
\end{array} \\
\text{red + moon} \quad \begin{array}{c|c|c|c|c} \hline \\
\hline \\
\end{array} \\
\]

\[
\text{moon} \quad \begin{array}{c|c|c} \hline \\
\hline x \\
\end{array} \\
\]

\[
= \text{red(moon)} \\
\]


▶ makes more explicit link with compositionality literature
▶ similarities with function-based approaches above
▶ supervised approach in which composition solution depends on annotated data from task at hand
Main points (for our purposes)

- Measure similarity of sentences taking into account not only sentence vector, but also vectors representing all constituent phrases and words
  - Map these representations to similarity matrix of fixed size, even for sentences with different lengths and structures
- Neural-network-based learning of composition function (autoencoders)
Results

- for some tasks, more sophisticated methods outperform the additive model
- but the additive model is surprisingly good
- one of the problems: lack of adequate testbeds
  - see this year’s SemEval Task 1
Outline

Introduction: The distributional hypothesis

Constructing the models

Semantic similarity as geometric distance

Evaluation

Multimodal distributional models
  Computer vision

Compositionality
  Why?
  How?

Conclusion
Some hot topics

- Compositionality in distributional semantics
- Semantic representations in context (polysemy resolution, co-composition...)
- Multimodal DSMs
- Very large DSMs
Not solved

- Parameter Hell
Build your own distributional semantic model

- corpus (several out there for several languages, see archives of the Corpora Mailing List)
- Standard linguistic pre-processing and indexing tools (TreeTagger, MaltParser, IMS CWB...)
- easy to write scripts for co-occurrence counts
  - not trivial with very large corpora. Hadoop (MapReduce algorithm) ideal for this, but often a pain in practice.
- COMPOSES webpage with link to toolkit in progress: http://clic.cimec.unitn.it/composes
- See the Links page for other toolkits!
- if you build your own matrix: Dimensionality reduction with SVDLIBC (http://tedlab.mit.edu/~dr/svdlibc/)
Distributional Semantics

Marco Baroni and Gemma Boleda

CS 388: Natural Language Processing