Lexical Semantics: Similarity Measures and Clustering

Today: Semantic Similarity

This parrot is no more!
It has ceased to be!
It’s expired and gone to meet its maker!
This is a late parrot!
This...is an EX-PARROT!

Beyond Dead Parrots

Automatically constricted clusters of semantically similar words (Charniak, 1997):

<table>
<thead>
<tr>
<th>Tuesday</th>
<th>Saturday</th>
<th>Sunday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>Thursday</td>
<td>Wednesday</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friday</th>
<th>Monday</th>
<th>Thursday</th>
<th>Tuesday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>guys</td>
<td>folks</td>
<td>fellows</td>
<td>CEOs</td>
</tr>
<tr>
<td>water</td>
<td>gas</td>
<td>cola</td>
<td>liquid</td>
<td>acid</td>
</tr>
<tr>
<td>that</td>
<td>the</td>
<td>the</td>
<td>the</td>
<td>the</td>
</tr>
</tbody>
</table>

head body hands eyes voice arm seat eye hair mouth

State-of-the-art Methods

Closest words for ?

| anthropology 0.275881, sociology 0.247909, comparative literature 0.245912, computer science 0.220663, political science 0.219948, zoology 0.210283, biochemistry 0.197723, mechanical engineering 0.191549, biology 0.189167, criminology 0.178423, social science 0.176762, psychology 0.171797, astronomy 0.16531, neuroscience 0.163764, psychiatry 0.163098, geology 0.158567, archaeology 0.157911, mathematics 0.157138 |
Motivation

Smoothing for statistical language models

- Two alternative guesses of speech recognizer:
  
  \[
  \text{For breakfast, she ate durian.} \\
  \text{For breakfast, she ate Dorian.}
  \]

- Our corpus contains neither “ate durian” nor “ate Dorian”

- But, our corpus contains “ate orange”, “ate banana”

Learning Similarity from Corpora

- You shall know a word by the company it keeps (Firth 1957)
  
  What is tizguino? (Nida, 1975)
  
  A bottle of tizguino is on the table.
  
  Tizguino makes you drunk.
  
  We make tizguino out of corn.

Motivation

Aid for Question-Answering and Information Retrieval

- Task: “Find documents about women astronauts”

- Problem: some documents use paraphrase of astronaut

In the history of Soviet/Russian space exploration, there have only been three Russian women cosmonauts: Valentina Tereshkova, Svetlana Savitskaya, and Elena Kondakova.
Outline

- Vector-space representation and similarity computation
  - Similarity-based Methods for LM
- Hierarchical clustering
  - Name Tagging with Word Clusters
- Computing semantic similarity using WordNet

Example 1: Next Word Representation

Brown et al. (1992)

- $C(x)$ denotes the vector of properties of $x$ (“context” of $x$)
- Assume alphabet of size $K$: $w^1, \ldots, w^K$
- $C(w) = \langle \#(w^1), \#(w^2), \ldots, \#(w^K) \rangle$, where $\#(w^i)$ is the number of times $w^i$ followed $w$ in the corpus

Example 2: Syntax-Based Representation

- The vector $C(n)$ for a noun $n$ is the distribution of verbs for which it served as direct object
- Assume (verb) alphabet of size $K$: $v^1, \ldots, v^K$
- $C(n) = \langle P(v^1|n), P(v^2|n), \ldots, P(v^K|n) \rangle$, where $P(v^i|n)$ is the probability that $v$ is a verb for which $n$ serves as a direct object
- Representation can be expanded to account for additional syntactic relations (subject, object, indirect-object)

Learning Similarity from Corpora

- Select important distributional properties of a word
- Create a vector of length $n$ for each word to be classified
- Viewing the $n$-dimensional vector as a point in an $n$-dimensional space, cluster points that are near one another
**Vector Space Model**

Each word is represented as a vector \( \vec{x} = (x_1, x_2, \ldots, x_n) \)

**Similarity Measure: Cosine**

Cosine \( \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \)

<table>
<thead>
<tr>
<th></th>
<th>cosmonaut</th>
<th>astronaut</th>
<th>moon</th>
<th>car</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soviet</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>American</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>spacewalking</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>red</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>full</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>old</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\cos(\text{cosm}, \text{astr}) = \frac{1+0+1+1+0+0+0+0+0+0}{\sqrt{1^2+0^2+1^2+0^2+0^2+0^2} \sqrt{0^2+1^2+1^2+0^2+0^2+0^2+0^2}}
\]

**Similarity Measure: Euclidean**

Euclidean \( |\vec{x}, \vec{y}| = |\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \)

<table>
<thead>
<tr>
<th></th>
<th>cosmonaut</th>
<th>astronaut</th>
<th>moon</th>
<th>car</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soviet</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>American</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>spacewalking</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>red</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>full</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>old</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
euclidian(\text{cosm}, \text{astr}) = \sqrt{(1-0)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2}
\]

**Outline**

- Vector-space representation and similarity computation
  - Similarity-based Methods for LM
- Hierarchical clustering
  - Name Tagging with Word Clusters
- Computing semantic similarity using WordNet
Smoothing for Language Modeling

- Task: estimate the probability of unseen word pairs

- Possible approaches:
  - Katz back-off scheme — utilize unigram estimates
  - Class-based methods — utilize average co-occurrence probabilities of the classes to which the two words belong
  - Similarity-based methods

Similarity-based Methods for LM

(Dagan, Lee & Pereira, 1997)

- Idea:
  1. combine estimates for the words most similar to a word \( w \)
  2. weight the evidence provided by word \( w' \) by a function of its similarity to \( w \)

- Implementation:
  - a scheme for deciding which word pairs require a similarity-based estimate
  - a method for combining information from similar words
  - a function measuring similarity between words

Discounting

\[
P(w_2|w_1) = \begin{cases} 
  P_d(w_2|w_1) & c(w_1, w_2) > 0 \\
  \alpha(w_1) P_r(w_2|w_1) & \text{otherwise}
\end{cases}
\]

- \( P_d \) Good-Turing discounted estimate
- \( \alpha(w_1) \) normalization factor
- \( P_r \) the model for probability redistribution among unseen words

Combining Evidence

Assumption: if word \( w'_1 \) is “similar” to word \( w_1 \), then \( w'_1 \) can yield information about the probability of unseen word pairs involving \( w_1 \)

\[
S(w_1) — \text{the set of words most similar to } w_1 \\
W(w_1, w'_1) — \text{similarity function} \\
P_{\text{sim}}(w_2|w_1) = \sum_{w'_1 \in S(w_1)} \frac{W(w_1, w'_1)}{N(w_1)} P(w_2|w'_1) \\
N(w_1) = \sum_{w'_1 \in S(w_1)} W(w_1, w'_1)
\]
Combining Evidence (cont.)

How to define $S(w_1)$?
- $S(w_1) = V$
- $S(w_1)$: the closest $k$ or fewer words $w_1'$ such that dissimilarity between $w_1$ and $w_1'$ is less than a threshold value $t$

Redistribution model:

$$P_2(w_2|w_1) = P_{sim}(w_2|w_1)$$

Other Probabilistic Dissimilarity Measures

- Information Radius:
  $$IRad(p, q) = D(p||\frac{p+q}{2}) + D(q||\frac{p+q}{2})$$
  - Symmetric
  - Well-defined if either $q_i > 0$ or $p_i > 0$

- $L_1$ norm:
  $$L_1(p, q) = \sum_i |p_i - q_i|$$
  - Symmetric
  - Well-defined for arbitrary $p$ and $q$

Kullback Leibler Divergence

- Definition: The KL Divergence $D(p||q)$ measures how much information is lost if we assume distribution $q$ when the true distribution is $p$
  $$D(p||q) = \sum_i p_i \log\frac{p_i}{q_i}$$

- Properties:
  - Non-negative
  - $D(p||q) = 0$ iff $p = q$
  - Not symmetric and doesn’t satisfy triangle inequality
  - If $q_i = 0$ and $p_i > 0$, then $D(p||q)$ gets infinite value

Evaluation Task: Word Disambiguation

- Task: Given a noun and two verbs, decide which verb is more likely to have this noun as a direct object
  $P(\text{plans}|\text{make})$ vs. $P(\text{plans}|\text{take})$
  $P(\text{action}|\text{make})$ vs. $P(\text{action}|\text{take})$

- Construction of candidate verb pairs:
  - generate verb-noun pairs on the test set
  - select pairs of verbs with similar frequency
  - remove all the pairs seen in the training set
Evaluation Setup

- Performance metric
  \[
  \frac{(# \text{ of incorrect choices}) + (# \text{ of ties})}{2N}
  \]
  
  \(N\) is the size of the test corpus

- Data:
  - 44m words of 1998 AP newswire
  - select 1000 most frequent nouns and their corresponding verbs
  - Training: 587833 pairs, Testing: 17152 pairs

- Baseline: Maximum Likelihood Estimator
  - Error rate: 0.5

Performance of Similarity-Based Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Katz</td>
<td>0.51</td>
</tr>
<tr>
<td>MLE</td>
<td>0.50</td>
</tr>
<tr>
<td>RandMLE</td>
<td>0.47</td>
</tr>
<tr>
<td>(L_1)MLE</td>
<td>0.27</td>
</tr>
<tr>
<td>IRadMLE</td>
<td>0.26</td>
</tr>
</tbody>
</table>

- RandMLE — Randomized combination of weights
- \(L_1\)MLE — Similarity function based on \(L_1\)
- IRadMLE — Similarity function based on IRad

Automatic Thesaurus Construction

http://www.cs.ualberta.ca/~lindek/demos/depsimdoc.htm

Closest words for president

| Leader | 0.264431, minister 0.251936, vice president 0.238359, Clinton 0.238222, chairman 0.207511, government 0.206842, Governor 0.193404, official 0.191428, Premier 0.177853, Yeltsin 0.173577, member 0.173468, foreign minister 0.171829, Mayor 0.168488, head of state 0.167166, chief 0.164998, Ambassador 0.162118, Speaker 0.161698, General 0.159422, secretary 0.156158, chief executive 0.15158 |

Problems with Corpus-based Similarity

- Low-frequency words skew the results
  - “breast-undergoing”, “childhood-phychosis”, “outflow-infundibulum”

- Semantic similarity does not imply synonymy
  - “large-small”, “heavy-light”, “shallow-coastal”

- Distributional information may not be sufficient for true semantic grouping
Outline

- Vector-space representation and similarity computation
  - Similarity-based Methods for LM
- Hierarchical clustering
  - Name Tagging with Word Clusters
- Computing semantic similarity using WordNet

Hierarchical Clustering

Greedy, bottom-up version:

- Initialization: Create a separate cluster for each object
- Each iteration: Find two most similar clusters and merge them
- Termination: All the objects are in the same cluster

Bottom-Up Hierarchical Clustering

Given: a set \( \mathcal{X} = \{x_1, \ldots, x_n\} \) of objects

A similarity function \( sim \)

for \( i := 1 \) to \( n \) do
  \( c_i := x_i \)

\( C := \{c_1, \ldots, c_n\} \)

\( j := n + 1 \)

while \( |C| > 1 \) do

  \( (c_{n_1}, c_{n_2}) := \arg\max_{(c_u, c_v) \in C \times C} sim(c_u, c_v) \)

  \( c_j := c_{n_1} \cup c_{n_2} \)

  \( C := (C - \{c_{n_1}, c_{n_2}\}) \cup \{c_j\} \)

  \( j := j + 1 \)

Agglomerative Clustering

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A B C D E
Agglomerative Clustering

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Clustering Function

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Agglomerative Clustering

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Clustering Function

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Agglomerative Clustering

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Clustering Function

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Clustering Function

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>B</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.0</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Single-link**: $CD(X, Y) = \min_{x \in X, y \in Y} D(x, y)$
- **Complete-link**: $CD(X, Y) = \max_{x \in X, y \in Y} D(x, y)$
- **Average-link**: $CD(X, Y) = \frac{1}{|X| \times |Y|} \sum_{x \in X} \sum_{y \in Y} D(x, y)$

### Evaluating Clustering Methods

- Perform **task-based evaluation**
- Test the resulting clusters **intuitively**, i.e., inspect them and see if they make sense. Not advisable.
- Have an **expert** generate clusters manually, and test the automatically generated ones against them.
- Test the clusters against a predefined **classification** if there is one.

### Clustering Function

### Outline

- Vector-space representation and similarity computation
  - Similarity-based Methods for LM
- **Hierarchical clustering**
  - Name Tagging with Word Clusters
- Computing semantic similarity using WordNet
**Named Entity Extraction as Tagging**

**INPUT:** Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

**OUTPUT:** Profits/NA soared/NA at/NA Boeing/SC Co./CC ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

NA = No entity  
SC = Start Company  
CC = Continue Company  
SL = Start Location  
CL = Continue Location  
...

**Log-Linear Models**

- We have some input domain $\mathcal{X}$, and a finite label set $\mathcal{Y}$. Aim is to provide a conditional probability $P(y \mid x)$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

- A feature is a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  
  (Often binary features or indicator functions  
  $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$).

- Say we have $m$ features $\phi_k$ for $k = 1 \ldots m$  
  $\Rightarrow$ A feature vector $\phi(x, y) \in \mathbb{R}^m$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

- We also have a **parameter vector** $\mathbf{W} \in \mathbb{R}^m$.

- We define $P(y \mid x, \mathbf{W}) = \frac{e^{\mathbf{W} \cdot \phi(x, y)}}{\sum_{y' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x, y')}}$.

**The Set of Features for POS Tagging**

- Word/tag features for all word/tag pairs, e.g.,

$$\phi_{100}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } t = \text{base} \\ 0 & \text{otherwise} \end{cases}$$

- Spelling features for all prefixes/suffixes of length $\leq 4$, e.g.,

$$\phi_{101}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in } \text{ing} \text{ and } t = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{102}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ starts with } \text{pre} \text{ and } t = \text{NN} \\ 0 & \text{otherwise} \end{cases}$$

**Tagging Performance**

(Miller, Guinness & Zamanian, 2004)

<table>
<thead>
<tr>
<th>Training Size</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>74%</td>
</tr>
<tr>
<td>150,000</td>
<td>90%</td>
</tr>
<tr>
<td>1,000,000</td>
<td>95%</td>
</tr>
</tbody>
</table>

Annotation effort:

- Annotation rate: 5000 words per hour
- 4 person-days of annotation work are required for porting a tagger to a new domain
Name Tagging with Word Clusters

- Goal: reduce the amount of training data
- Implementation:
  - Induce word clusters from a large corpus of un-annotated data
  - Incorporate cluster features in a discriminatively trained tagging model

Adding Clustering Information

How to select an appropriate level of granularity?
- Too small, and clusters provide insufficient generalization
- Too large, and they are inappropriately generalized

Use hierarchical clustering

Encoding Clustering Structure

A word is represented by a binary string
- Follow the traversal path from the root to a leaf
- Assign a 0 for each left branch, and 1 for each right branch

Sample Bit Strings

<table>
<thead>
<tr>
<th>Word</th>
<th>Bit String</th>
</tr>
</thead>
<tbody>
<tr>
<td>lawyer</td>
<td>100000110100100</td>
</tr>
<tr>
<td>newspaperman</td>
<td>100000110100100</td>
</tr>
<tr>
<td>stewardess</td>
<td>100000110100101</td>
</tr>
<tr>
<td>toxicologist</td>
<td>10000011010011</td>
</tr>
<tr>
<td>slang</td>
<td>100000110100101</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Nike</td>
<td>1011011100100101011100</td>
</tr>
<tr>
<td>Maytag</td>
<td>10110111001001010111010</td>
</tr>
<tr>
<td>Generali</td>
<td>10110111001001010111011</td>
</tr>
<tr>
<td>Gap</td>
<td>1011011100100101011110</td>
</tr>
<tr>
<td>Harley-Davidson</td>
<td>10110111001001010111110</td>
</tr>
</tbody>
</table>
Cluster Based Features

8. Tag + Pref8 of Cur Word
9. Tag + Pref2 of Cur Word
10. Tag + Pref6 of Cur Word
11. Tag + Pref20 of Cur Word
12. Tag + Pref8 of Prev Word
13. Tag + Pref2 of Prev Word
14. Tag + Pref6 of Prev Word
15. Tag + Pref20 of Prev Word
16. Tag + Pref8 of Next Word
17. Tag + Pref2 of Next Word
18. Tag + Pref6 of Next Word
19. Tag + Pref20 of Next Word

Outline

- Vector-space representation and similarity computation
  - Similarity-based Methods for LM
- Hierarchical clustering
  - Name Tagging with Word Clusters
- Computing semantic similarity using WordNet

Results

- With 50,000 words of training, the cluster-based model exceeds 90F, a level not reached by the standard model until it has 150,000 words of training.
- At 1,000,000 words of training, the cluster-based model achieves 96.08F compared to 94.72 for the HMM, a 25% reduction in error.

WordNet

- Large scale semantic lexicon for the English language
- Started in 1990 as a language project by George Miller and Christiane Fellbaum at Princeton
- As of 2006, the database contains about 150,000 words organized in over 115,000 synsets for a total of 207,000 word-sense pairs

<table>
<thead>
<tr>
<th>Category</th>
<th>Unique Forms</th>
<th>Number of Senses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>114648</td>
<td>79689</td>
</tr>
<tr>
<td>Verb</td>
<td>11306</td>
<td>13508</td>
</tr>
<tr>
<td>Adjective</td>
<td>21436</td>
<td>18563</td>
</tr>
<tr>
<td>Adverb</td>
<td>4669</td>
<td>3664</td>
</tr>
</tbody>
</table>
Word with the Corresponding Synsets

1. **water, H2O** – (binary compound that occurs at room temperature as a clear colorless odorless tasteless liquid; freezes into ice below 0 degrees centigrade and boils above 100 degrees centigrade; widely used as a solvent)

2. **body of water, water** – (the part of the earth's surface covered with water (such as a river or lake or ocean); "they invaded our territorial waters"; "they were sitting by the water's edge")

3. **water system, water supply, water** – (facility that provides a source of water; "the town debated the purification of the water supply"; "first you have to cut off the water")

4. **water** – (once thought to be one of four elements composing the universe (Empedocles))

5. **urine, piss, pee, piddle, weewee, water** – (liquid excretory product; "there was blood in his urine"; "the child had to make water")

6. **water** – (a fluid necessary for the life of most animals and plants; "he asked for a drink of water")

**Sense Distribution Statistics**

<table>
<thead>
<tr>
<th>POS</th>
<th>Monosemous</th>
<th>Polysemous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>99524</td>
<td>15124</td>
</tr>
<tr>
<td>Verb</td>
<td>6256</td>
<td>5050</td>
</tr>
<tr>
<td>Adverb</td>
<td>16103</td>
<td>5333</td>
</tr>
<tr>
<td>Adjective</td>
<td>3901</td>
<td>768</td>
</tr>
<tr>
<td>Total</td>
<td>125784</td>
<td>26275</td>
</tr>
</tbody>
</table>

**WordNet Relations**

<table>
<thead>
<tr>
<th>Relation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonymy</td>
<td>marriage, wedlock</td>
</tr>
<tr>
<td>Hyponymy/Hyperonymy</td>
<td>computer, machine</td>
</tr>
<tr>
<td>Meronymy</td>
<td>door, knob</td>
</tr>
<tr>
<td>Antonymy</td>
<td>large, small</td>
</tr>
</tbody>
</table>

Glosses: “computer (a machine for performing calculations automatically)

Links between derivationally related noun/verb pairs: “computer, computing, computed, . . . “

**Hyponymy Hierarchy**

- computer, data processor, ... — (a machine for performing calculations automatically)
- machine — (any mechanical or electrical device that performs or assists in the performance of human tasks)
- device — (an instrumentality invented for a particular purpose)
- artifact — a man-made object
- object, inanimate object, physical object — a nonliving entity
- entity — something having concrete existence; living or nonliving
Computing Semantic Similarity
Suppose you are given the following words. Your task is to group them according to how similar they are:

apple  infant  man  banana  grapefruit  baby  grape  woman

Using WordNet to Determine Similarity

<table>
<thead>
<tr>
<th>apple</th>
<th>man</th>
<th>fruit</th>
<th>produce</th>
<th>banana</th>
</tr>
</thead>
<tbody>
<tr>
<td>fruit</td>
<td>produce</td>
<td>baby</td>
<td>child, kid</td>
<td></td>
</tr>
<tr>
<td>produce</td>
<td>offspring, progeny</td>
<td>relative, relation</td>
<td>person, individual</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>man</th>
<th>female, female person</th>
</tr>
</thead>
<tbody>
<tr>
<td>fruit</td>
<td>person, individual</td>
</tr>
</tbody>
</table>

Why use WordNet?
- Quality
  - Developed and maintained by researchers
- Habit
  - Many applications are currently using WordNet
- Available software
  - SenseRelate (Pedersen et al):
    http://wn-similarity.sourceforge.com

Similarity by Path Length
Why not use WordNet?

- Incomplete (technical terms may be absent)
- The length of the paths are irregular across the hierarchies
- How to relate terms that are not in the same hierarchies?
  The “tennis problem”:
  - Player
  - Racquet
  - Ball
  - Net

Summary

- Corpus-based Similarity Computation
  - Vector Space Model
  - Similarity Measures
  - Hierarchical Clustering
- Lexicon-based Similarity Computation