Last class
- Intro to lexical semantics
- Lexical semantic resources: WordNet

Today
- Word sense disambiguation
  » Dictionary-based approaches
  » Supervised machine learning methods
  » Issues for WSD evaluation

Today
- Word sense disambiguation
  » Dictionary-based approaches
  » Supervised machine learning methods
  » Issues for WSD evaluation

Given a fixed set of senses associated with a lexical item, determine which of them applies to a particular instance of the lexical item

Two fundamental approaches
- WSD occurs during semantic analysis as a side-effect of the elimination of ill-formed semantic representations

Stand-alone approach
- WSD is performed independent of, and prior to, compositional semantic analysis
- Makes minimal assumptions about what information will be available from other NLP processes
- Applicable in large-scale practical applications
Dictionary-based approaches

- Rely on machine readable dictionaries
- Initial implementation of this kind of approach is due to Michael Lesk (1986)
  - Given a word $W$ to be disambiguated in context $C$
    » Retrieve all of the sense definitions, $S$, for $W$ from the MRD
    » Compare each $s$ in $S$ to the dictionary definitions $D$ of all the remaining words $c$ in the context $C$
    » Select the sense $s$ with the most overlap with $D$ (the definitions of the context words $C$)

Example

- Word: *cone*
- Context: *pine cone*
- Sense definitions
  *pine* 1 kind of evergreen tree with needle-shaped leaves
  2 waste away through sorrow or illness
  *cone* 1 solid body which narrows to a point
  2 something of this shape whether solid or hollow
  3 fruit of certain evergreen trees

- Accuracy of 50-70% on short samples of text from *Pride and Prejudice* and an AP newswire article.

Machine learning approaches

- Machine learning methods
  - Supervised inductive learning
  - Bootstrapping
  - Unsupervised
- Emphasis is on acquiring the knowledge needed for the task from data, rather than from human analysts.

Inductive ML framework

- Examples of task $(features + class)$
- Description of context
- ML Algorithm
- Novel example $(features)$
- Learn one such classifier for each lexeme to be disambiguated
- Classifier $(program)$
- Correct word sense
- Class
Running example

An electric guitar and bass player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

1 Fish sense
2 Musical sense
3 ...

Feature vector representation

- **target**: the word to be disambiguated
- **context**: portion of the surrounding text
  - Select a “window” size
  - Tagged with part-of-speech information
  - Stemming or morphological processing
  - Possibly some partial parsing
- Convert the context (and target) into a set of features
  - Attribute-value pairs
    - Numeric, boolean, categorical, ...

Collocational features

- Encode information about the lexical inhabitants of specific positions located to the left or right of the target word.
  - E.g. the word, its root form, its part-of-speech
  - An electric guitar and bass player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

Co-occurrence features

- Encodes information about neighboring words, ignoring exact positions.
  - Select a small number of frequently used content words for use as features
    - 12 most frequent content words from a collection of bass sentences drawn from the WSJ: fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band
  - Co-occurrence vector (window of size 10)
  - Attributes: the words themselves (or their roots)
  - Values: number of times the word occurs in a region surrounding the target word

<table>
<thead>
<tr>
<th>pre2-word</th>
<th>pre2-pos</th>
<th>pre1-word</th>
<th>pre1-pos</th>
<th>fol1-word</th>
<th>fol1-pos</th>
<th>fol2-word</th>
<th>fol2-pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>guitar</td>
<td>NN1</td>
<td>and</td>
<td>CJC</td>
<td>player</td>
<td>NN1</td>
<td>stand</td>
<td>VVB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>