Language Processing with Python

Methods in Computational Linguistics I
September 3
Last Time

- Introduction to the Class
- First look at Python
- and NLTK
Today

- Language Processing with Python (Chapter 1)
- Strings
- Using NLTK
- Parsing
Natural Language Toolkit

- Text material
  - Raw text
  - Annotated Text
- Tools
  - Part of speech taggers
  - Semantic analysis
- Resources
  - WordNet, Treebanks
NLTK Demo

- Demo time!
Major CL tasks

- Part of Speech Tagging
- Parsing
- Word Net
- Named Entity Recognition
- Information Retrieval
- Sentiment Analysis
- Document Clustering

- Topic Segmentation
- Authoring
- Machine Translation
- Summarization
- Information Extraction
- Spoken Dialog Systems
- Natural Language Generation
- Word Sense Disambiguation
Part of Speech Tagging

- Task: Given a string of words, identify the parts of speech for each word.

  A man walks into a bar.
Part of Speech Tagging

- Surface level syntax.
- Primary operation
  - Parsing
  - Word Sense Disambiguation
  - Semantic Role labeling
- Segmentation
  - Discourse, Topic, Sentence
How do we do it?

• Learn from Data.
• Annotated Data:
  A man walks into a bar.
• Unlabeled Data:
  A man walks home.
  The pitcher issued four walks.
### Part of speech tagging

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>Noun</th>
<th>Verb</th>
<th>Prep</th>
<th>Adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.9</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>man</td>
<td>0.0</td>
<td>0.6</td>
<td>0.2</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>walks</td>
<td>0.0</td>
<td>0.2</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>into</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>bar</td>
<td>0.0</td>
<td>0.7</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Limitations

• Unseen tokens
• Uncommon interpretations
• Long term dependencies
Parsing

- Generate a Parse Tree from:
  - The surface form (words) of the text
  - Part of Speech Tokens
Parsing Styles

- Parse Trees

- Dependency Parsing

I gave him my address.
Context Free Grammars for Parsing

- S → VP
- S → NP VP
- NP → Det Nom
- Nom → Noun
- Nom → Adj Noun
- VP → Verb Noun
Using these rules

- Construct a parse that fits the desired text.
Limitations

• The grammar must be built by hand.
• Can’t handle ungrammatical sentences.
• Can’t resolve ambiguity.
Probabilistic Parsing

- Assign each transition a probability
- Find the parse with the greatest "likelihood"
Next Time

- Text Corpora in NLTK