Corpora in NLTK
Methods in Computational Linguistics I
Lesson 3
Last Time

- Text Processing
Today

- Text Corpora in NLTK
- for loops
- stored python scripts
- WordNet
What is a corpus?

- A (relatively) unprocessed set of data
- Penn Treebank
- WSJ
- WordNet?
- Switchboard
How are corpora used in CL?

- Corpora are the lifeblood of computational linguistics research.
- Collection and Dissemination of Corpora
- Descriptive Statistics
- Statistical Modeling
- Consistent Evaluation
Collection and Dissemination

- Collecting language material is a valuable research outcome.
- Endangered languages
- Hard-to-find data
  - Authentic emotion
  - Proprietary information
  - fMRI
- Articulatory data
Descriptive Statistics

- Describe some linguistic phenomenon using statistics.
- How likely is the word “tweet” to be a verb?
- Is a sentence more likely to start with the word “the” or “baboon”?
- What language use phenomena are associated with positive product reviews?
Statistical Modeling

- Use statistics to make predictions about unseen language data.
- Train a statistical model based on part of a corpus.
- Evaluate on unseen material.
- Models are task specific.
Consistent Evaluation

- Different approaches to a common task can be evaluated on common material.
- “Shared Tasks”
- This allows for “state-of-the-art” results to be established and verified.
Using Corpora with NLTK and Python

- NLTK includes many corpora
- Books of the bible
- Public domain books (Project Gutenberg)
- News
- Web chat
- Blogs
- Some non-English material as well.
NLTK Corpus Demos

- What corpora are included
- conditional frequency distributions
- for loops
- python scripts
  - calling python from outside the interpreter
NLTK resources

• Corpora often also include data annotations
• NLTK has a variety of methods to access corpus annotations.
  • corpus.sents()
  • Author
  • Part of speech (POS) tags
  • Parse tree information
  • Word net annotations
  • Discourse information
  • etc.
• We'll come back to these as needed.
<table>
<thead>
<tr>
<th>Major CL tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Part of Speech Tagging</td>
</tr>
<tr>
<td>• Parsing</td>
</tr>
<tr>
<td>• Word Net</td>
</tr>
<tr>
<td>• Named Entity Recognition</td>
</tr>
<tr>
<td>• Information Retrieval</td>
</tr>
<tr>
<td>• Sentiment Analysis</td>
</tr>
<tr>
<td>• Document Clustering</td>
</tr>
<tr>
<td>• Topic Segmentation</td>
</tr>
<tr>
<td>• Authoring</td>
</tr>
<tr>
<td>• Machine Translation</td>
</tr>
<tr>
<td>• Summarization</td>
</tr>
<tr>
<td>• Information Extraction</td>
</tr>
<tr>
<td>• Spoken Dialog Systems</td>
</tr>
<tr>
<td>• Natural Language Generation</td>
</tr>
<tr>
<td>• Word Sense Disambiguation</td>
</tr>
</tbody>
</table>
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 John My Address.
Context Free Grammars for Parsing

- \( S \rightarrow VP \)
- \( S \rightarrow NP \ VP \)
- \( NP \rightarrow \text{Det} \ Nom \)
- \( Nom \rightarrow \text{Noun} \)
- \( Nom \rightarrow \text{Adj} \ Nom \)
- \( VP \rightarrow \text{Verb} \ Nom \)
- \( \text{Det} \rightarrow \text{“A”, “The”} \)
- \( \text{Noun} \rightarrow \text{“I”, “John”, “Address”} \)
- \( \text{Verb} \rightarrow \text{“Gave”} \)
- \( \text{Adj} \rightarrow \text{“My”, “Blue”} \)
- \( \text{Adv} \rightarrow \text{“Quickly”} \)
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

• Functions, Lists and Tuples