Unsupervised Grammar Induction?
- Start with raw text, learn syntactic structure
- Some have argued that learning syntax from positive data alone is impossible:
  - Gold, 1967: Non-identifiability in the limit
  - Chomsky, 1980: The poverty of the stimulus
- Many others have felt it should be possible:
  - Lari and Young, 1990
  - Carroll and Charniak, 1992
  - Alex Clark, 2001
  - Mark Paskin, 2001
- ... and many more, but it hasn’t worked well

American Structuralists
- Methods of automatically doing linguistic induction (if stated in a somewhat informal, non-computational manner) were the basis of the proposed 'discovery procedures' argued for by structuralists in the 1950s:
  - "Many linguists felt that the procedures had been so well worked out that computers could take over the drudgery of linguistic analysis. The time was near at hand when all one would have to do would be to punch the data into the computer and out would come the grammar!" (Newmeyer 1986)

Unsupervised Learning
- Systems take raw data and automatically detect patterns
- Why unsupervised learning?
  - More data than annotation
  - Insights into machine learning, clustering
  - Kids learn some aspects of language entirely without supervision
- Here: unsupervised learning
  - Work purely from the forms of the utterances
  - Syntactic learnability: Neither assume nor exploit prior meaning or grounding [cf. Feldman et al.]

Supervised Learning
- Systems duplicate correct analyses from training data
- Hand-annotation of data
  - Time-consuming
  - Expensive
  - Hard to adapt for new purposes (tasks, languages, domains, etc.)
  - Corpus availability drives research, not tasks!
- Example: Penn Treebank
  - 50K Sentences
  - Hand-parsed over several years

Gold (1967) vs. Horning (1969)
- Gold: no superfinite class of languages (e.g., regular or context-free languages, etc.) is learnable without negative examples.
  - Certain conditions: nearly arbitrary sequence of examples; only constraint is that no sentence may be withheld from the learner indefinitely.
- Still regularly cited as bedrock for innatist linguistics
  - "The Poverty of the Stimulus" (Chomsky 1965)
- Responses suggested by Gold:
  - Subtle, covert negative evidence ← Some recent claims
  - Inmate knowledge shrinks language class ← Chomsky
  - Assumption about presentation of examples is too general ← e.g., probabilistic language model
Gold (1967) vs. Horning (1969)

- If texts are generated by some stochastic process, then we can take advantage of how often something occurs to drive language acquisition.
- In particular, as time goes by without us having seen something, we have more evidence that it either doesn't happen or happens very, very rarely in the language.
- Implicit negative evidence: you can't withhold common stuff.
- Horning: stochastic context-free languages are learnable from only positive examples.
- But Horning's proof is enumerative, rather than providing a plausible grammar learning method. (See, e.g., Rohde and Plaut 1999 for discussion)

Poverty of the stimulus?

- There's a lot of stimulus: Baayen estimate of 200 million words by adulthood.
- c. 1,000,000 sentences a year.
- Much more sign of other poverties:
  - Poverty of imagination
  - Poverty of model building
  - Poverty of understanding of (machine) learning among linguists.
- At any rate, an obvious way to investigate this question is empirically....

NLP

- This is also an engineering problem.
- Because of the very modest success of grammar induction systems, practical work has largely abandoned induction, and is dependent on large hand-built resources.
- Means that we're barely learning. Instead, getting human to externalize knowledge.
- It's scarcely machine learning at all....
- Machine learning should be looking at feature/representation induction.

Idea: Lexical Affinity Models

- Words select other words on syntactic grounds.
- Link up pairs with high mutual information.
- (Foster, 1998, MIT PhD: Greedy linkage)
- (Fasskin, 2001, NIPS; Iterative relaxation with EM)
- Evaluation: compare linked pairs to a gold standard.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fasskin, 2001</td>
<td>39.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem: Non-Syntactic Affinity</th>
</tr>
</thead>
</table>

- Mutual information between words does not necessarily indicate syntactic selection.
  - congress narrowly passed the amended bill
  - expect brushbacks but no beanballs
  - a new year begins in new york

Idea: Word Classes

- Individual words like congress are enwined with semantic facts about the world.
- Syntactic classes, like NOUN and ADVERB are bleached of word-specific semantics.
- Automatic word classes more likely to look like DAYS-OF-WEEK or PERSON-NAMES.
- We could build dependency models over word classes. [cf. Carroll and Charniak, 1992]

- congress narrowly passed the amended bill
Problems: Word Class Models

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Carroll and Charniak 92</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN NOUN VERB</td>
<td>41.7</td>
<td></td>
</tr>
<tr>
<td>NOUN NOUN VERB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stock prices fell</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stock prices fell</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Issues:
- Too simple a design — doesn’t exploit context.
- Too complex a design — entirely supervised.
- Too many words of stock prices fell.

Bias: Using more sophisticated dependency representations

Classifiers

<table>
<thead>
<tr>
<th></th>
<th>Distance</th>
<th>Local Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paskin 01</td>
<td>x</td>
<td>Pr(u</td>
</tr>
<tr>
<td>Carroll &amp; Charniak 92</td>
<td>x</td>
<td>Pr(c(u)</td>
</tr>
<tr>
<td>Our Model (DMV)</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Adjacent Words

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Paskin 01</td>
<td>55.9</td>
</tr>
<tr>
<td>Carroll &amp; Charniak 92</td>
<td>63.6</td>
</tr>
</tbody>
</table>

Grammar Structure search: Greedy-Merge

Agglomerative clustering of part of speech sequences
- At each step merge two closest sequences.
- Merging can create recursion.

DT NN
DT JJ NN → z1 → DT NN
z1 → DT JJ NN
Create a category

NN
JJ NN → NN → JJ NN
z1 → DT NN
1. Add to a category
z1 → DT [JJ NN] z1 → DT [NN]

2. Re-analyze rules

Learned Grammars (NP)

N-bar or zero determiner NP
Proper NP
zNN → NN | NNS
zNN → JJ zNN
zNN → zNN zNN
zNP → DT zNN
zNP → zNN NP
zNN → zNP zNN

NP with determiner
zNP → DT zNN
zNP → PRPS zNN

Categories have been hand re-labeled based on closest treebank category.

Problems with Structure Search

Greedy systems make local constituency decisions.
- Chunk prep article → X or
- Chunk article noun → Y?
- Decide heuristically!
- Many heuristics are often right, but
- Early errors stay around

In the end, constituents have to tie up into parses!
- CRul na-o-overlap constraint.
- Requires comparing full parses.
- Not heuristic.

Idea: Learn PCFGs with EM (Parameter Search)

Classic experiments on learning PCFGs with Expectation-Maximization [Lari and Young, 1990]

\[ \begin{array}{c}
X_1, X_2, \ldots, X_n \\
X_j \\
X_k
\end{array} \]

- Full binary grammar over + symbols
- Parse randomly at first
- Re-estimate rule probabilities off parses
- Repeat
- Their conclusion: it doesn’t work at all
Example Parse: PCFG EM

Treebank Parse  DEP-PCFG Parse

Linguistic constituency

- EM assumptions/greediness may not work:
  - The important tests for linguistic constituency emphasize external distribution
  - Not internal constituency  (e.g., Radford 1988)
- Why? Look at Noun Phrases:

  you  <s>   --- saw me
  the man  <s>  He saw ---  </s>
  a cat with a limp  the elephant sat on ---
  three black dogs  My gosh!  ---  </s>
  Don’t look alike  All appear in these
  (and other) contexts

Other Approaches

- Some recent work in learning constituency:
  - [Adriaans, 99] Language grammars aren’t general PCFGs
  - [Clark, 01] Mutual-information filters detect constituents, then an MLD-guided search assembles them
  - [van Zaalen, 00] Finds low-diff-distance sentence pairs and extracts their differences
  - GB/Minimality  No results that I’ve seen
- Evaluation: fraction of nodes in gold trees correctly posited in proposed trees (unlabeled recall)

<table>
<thead>
<tr>
<th></th>
<th>Adriaans, 1999</th>
<th>Clark, 2001</th>
<th>van Zaalen, 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16.8</td>
<td>34.6</td>
<td>33.6</td>
</tr>
</tbody>
</table>

Right-Branching Baseline

- English trees tend to be right-branching, not balanced
- they were unwilling to agree to new terms
- A simple (English-specific) baseline is to choose the right-branching structure for each sentence

<table>
<thead>
<tr>
<th></th>
<th>van Zaalen, 00</th>
<th>Right-Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>33.6</td>
<td>46.4</td>
</tr>
</tbody>
</table>

Inspiration: Distributional Clustering for capturing linguistic classes

- Item tokens occur in specific contexts.
- Item types are identified with their context distribution.
- Distance between items defined by distance between their contexts.
- Used for clustering.

Learning Language Distributionally

<table>
<thead>
<tr>
<th>Item</th>
<th>Context</th>
<th>Learned</th>
<th>Paraphrasis</th>
</tr>
</thead>
<tbody>
<tr>
<td>words</td>
<td>left, right</td>
<td>parts-of-speech</td>
<td>Finch et al. 95, Schütze 95, Clark 00</td>
</tr>
<tr>
<td>words</td>
<td>content words</td>
<td>semantic fields</td>
<td>Pereira et al. 93, Dagan et al. 97</td>
</tr>
</tbody>
</table>

... it generally “pretty much” works:
  - parts-of-speech are partially semantic
  - constituent types require some filtering
**Idea: Distributional Syntax?**

- Can we use distributional clustering for learning syntax?

<table>
<thead>
<tr>
<th>Span</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>fell in september</td>
<td>payrolls ... ●</td>
</tr>
<tr>
<td>payrolls fell in</td>
<td>factory ... sept</td>
</tr>
</tbody>
</table>

**Problem: Identifying Constituents**

Distribution classes are easy to find...

... but figuring out which are constituents is hard.

**Constituent–Context Model (CCM)**

\[
P(S|T) = \prod_{(i,t) \in S} P(f(i,t) \mid +) \prod_{(i,t) \in S} P(\{ \_ \} \mid +) \prod_{(i,t) \in S} P(f(i,t) \mid -) \prod_{(i,t) \in S} P(\{ \_ \} \mid -)
\]

factory payrolls fell in september

**Parameter Estimation with EM**

1. **Re-Parse (E-Step)**
2. **Posterior Bracketing Distributions**
3. **Constituency Model Parameters**
   - \( P(c \mid b) \)
   - \( P(w_i \ldots w_j \mid c) \)
   - \( P(w_{i+1} \ldots w_{j+1} \mid c) \)

4. **Update Model (M-Step)**

**Results: Constituency**

<table>
<thead>
<tr>
<th>Right-Branch</th>
<th>Treebank Parse</th>
<th>CCM Parse</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.0</td>
<td><img src="image" alt="Treebank Parse" /></td>
<td><img src="image" alt="CCM Parse" /></td>
</tr>
</tbody>
</table>

**Spectrum of Systematic Errors**

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Inside NPs</th>
<th>Possessives</th>
<th>Verb groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM</td>
<td>the lazy cat</td>
<td>John’s cat</td>
<td>will be there</td>
</tr>
<tr>
<td>Treebank</td>
<td>the lazy cat</td>
<td>John’s cat</td>
<td>will be there</td>
</tr>
</tbody>
</table>

**CCM Right?**

Yes | Maybe | No

But the worst errors are the non-systematic ones! (~25%)
Combining These Models

- General case: [Klein and Manning, 03]
  - Multiple scoring functions over a single family of tree or sequence structures
  - Each function scores some sub-projection
  - We want to find the best structure according to the sum or product of the individual functions
  - Can do A* search with bounds given by components

Factored A* Estimates

- If \( w \) factors over projections \( \{\pi_i\} \), then for a path \( T \):
  \[
  w(T) = \sum_i w_i(\pi_i(T))
  \]
- Factored scores have a natural A* estimate:

Combining the two models

### Dependency Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>DMV</th>
<th>CCM + DMV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>45.6</td>
<td>62.7</td>
<td>64.7</td>
</tr>
</tbody>
</table>

### Constituency Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>CCM</th>
<th>CCM + DMV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>39.4</td>
<td>81.0</td>
<td>88.0</td>
</tr>
</tbody>
</table>

- Supervised PCFG constituency recall is 92.8
- Qualitative improvements:
  - Subject-verb groups gone, modifier placement improved

Crosslinguistic applicability of the learning algorithm

<table>
<thead>
<tr>
<th></th>
<th>English (7422 sentences)</th>
<th>Constituency Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Baseline</td>
<td>CCM + DMV</td>
</tr>
<tr>
<td>German (2175 sentences)</td>
<td>49.6</td>
<td>89.7</td>
</tr>
<tr>
<td>Chinese (2473 sentences)</td>
<td>35.5</td>
<td>46.7</td>
</tr>
</tbody>
</table>

Maybe grammar induction is possible after all?

That's it folks!

- Reminders:
  - Get those final projects done!
    - Must be submitted by next Friday
    - Mini-presentations in class next Wednesday