Lecture 18: Semantic Roles

Dan Klein – UC Berkeley

Includes examples from Johnson, Jurafsky and Gildea, Palmer

**Dependency Parsing**

- Lexicalized parsers can be seen as producing *dependency trees*

- Each local binary tree corresponds to an attachment in the dependency graph
Dependency Parsing

- Pure dependency parsing is only cubic [Eisner 99]

- Some work on non-projective dependencies
  - Common in, e.g. Czech parsing
  - Can do with MST algorithms [McDonald and Pereira 05]

Parse Reranking

- Assume the number of parses is very small
- We can represent each parse T as an arbitrary feature vector $\varphi(T)$
  - Typically, all local rules are features
  - Also non-local features, like how right-branching the overall tree is
  - [Charniak and Johnson 05] gives a rich set of features
Parse Reranking

- Since the number of parses is no longer huge
  - Can enumerate all parses efficiently
  - Can use simple machine learning methods to score trees
  - E.g. maxent reranking: learn a binary classifier over trees where:
    - The top candidates are positive
    - All others are negative
    - Rank trees by $P(+/T)$

- The best parsing numbers are from reranking systems

Shift-Reduce Parsers

- Another way to derive a tree:

- Parsing
  - No useful dynamic programming search
  - Can still use beam search [Ratnaparkhi 97]
Data-oriented parsing:

- Rewrite large (possibly lexicalized) subtrees in a single step

- Formally, a tree-insertion grammar
- Derivational ambiguity whether subtrees were generated atomically or compositionally
- Most probable parse is NP-complete

TIG: Insertion
Derivational Representations

- Generative derivational models:
  
  \[ P(D) = \prod_{d_i \in D} P(d_i|d_0 \ldots d_{i-1}) \]

- How is a PCFG a generative derivational model?

- Distinction between parses and parse derivations.

  \[ P(T) = \sum_{D: D \rightarrow T} P(D) \]

- How could there be multiple derivations?

Tree-adjoining grammars

- Start with local trees
- Can insert structure with adjunction operators
- Mildly context-sensitive
- Models long-distance dependencies naturally
- ... as well as other weird stuff that CFGs don't capture well (e.g. cross-serial dependencies)
TAG: Adjunction

NP
D↓N
man

N
AdjN*
tall

TAG: Long Distance

S
V
does
NP
Bill
think

S
V
S*

S
NP
VP

S
NP
VP

S
V
who

S
NP
VP

S
V
who

S
V
likes

S
VP

S
V
Harry

S
V

S
V
NP

S
V

S
V

S
V

S
V

S
V

CCG Parsing

- Combinatory Categorial Grammar
  - Fully (mono-) lexicalized grammar
  - Categories encode argument sequences
  - Very closely related to the lambda calculus (more later)
  - Can have spurious ambiguities (why?)

\[
\begin{align*}
  John &\vdash NP \\
  shares &\vdash NP \\
  buys &\vdash (S\backslash NP)/NP \\
  sleeps &\vdash S\backslash NP \\
  well &\vdash (S\backslash NP)\backslash(S\backslash NP)
\end{align*}
\]

\[
\begin{array}{c}
  S \\
  \downarrow \begin{array}{c}
    \text{NP} \\
    \text{S}\backslash\text{NP}
  \end{array}
\end{array}
\]

\[
\begin{array}{c}
  \text{John} \\
  \text{(S}\backslash\text{NP})/\text{NP} \\
  \text{NP} \\
  \downarrow \begin{array}{c}
    \text{buys} \\
    \text{shares}
  \end{array}
\end{array}
\]

Statistical Semantics?

- Last time:
  - Syntactic trees + lexical translations $\rightarrow$ (unambiguous) logical statements
  - Symbolic deep (?) semantics
  - Often used as part of a logical NLP interface or in database / dialog systems
  - Applications like question answering and information extraction often don't need such expressiveness

- Today:
  - Statistically extracting shallow semantics
  - Semantic role labeling
  - Coreference resolution
Semantic Role Labeling (SRL)

- Characterize clauses as relations with roles:

  \[ \text{Judge She} \text{ blames [Evaluate the Government [Reason for failing to do enough to help]]} \]

  Holman would characterize this as blaming [Evaluate the poor].

  The letter quotes Black as saying that [Judge, white and Navajo ranchers] misrepresent their livestock losses and blame [Reason everything] on coyotes.

- Want to more than which NP is the subject (but not much more):
- Relations like subject are syntactic, relations like agent or message are semantic
- Typical pipeline:
  - Parse, then label roles
  - Almost all errors locked in by parser
  - Really, SRL is quite a lot easier than parsing

SRL Example
PropBank / FrameNet

- FrameNet: roles shared between verbs
- PropBank: each verb has its own roles
- PropBank more used, because it's layered over the treebank (and so has greater coverage, plus parses)
- Note: some linguistic theories postulate even fewer roles than FrameNet (e.g. 5-20 total: agent, patient, instrument, etc.)

PropBank Example

fall.01 sense: move downward
roles: Arg1: thing falling
Arg2: extent, distance fallen
Arg3: start point
Arg4: end point

Sales fell to $251.2 million from $278.7 million.
arg1: Sales
rel: fell
arg4: to $251.2 million
arg3: from $278.7 million
PropBank Example

rotate.02  sense: shift from one thing to another
   roles:  Arg0:  causer of shift
           Arg1:  thing being changed
           Arg2:  old thing
           Arg3:  new thing

Many of Wednesday’s winners were losers yesterday as investors quickly took profits and rotated their buying to other issues, traders said.  (wsj_1723)
   arg0:  investors
   rel:   rotated
   arg1:  their buying
   arg3:  to other issues

PropBank Example

aim.01  sense: intend, plan
   roles:  Arg0:  aimer, planner
           Arg1:  plan, intent

The Central Council of Church Bell Ringers aims *trace* to improve relations with vicars.  (wsj_0089)
   arg0:  The Central Council of Church Bell Ringers
   rel:   aims
   arg1:  *trace* to improve relations with vicars

aim.02  sense: point (weapon) at
   roles:  Arg0:  aimer
           Arg1:  weapon, etc.
           Arg2:  target

Banks have been aiming packages at the elderly.
   arg0:  Banks
   rel:   aiming
   arg1:  packages
   arg2:  at the elderly
Shared Arguments

(NP-SBJ (JJ massive) (JJ internal) (NN debt))
(VP (VBZ has))
(VP (VBN forced))
(S
  (NP-SBJ-1 (DT the) (NN government))
  (VP
    (VP (TO to))
    (VP (VB borrow))
    (ADVP-MNR (RB massively))...  

Path Features

Path | Description                        
-----|------------------------------------
VB|VP|PP    | PP argument/adjunct  
VB|VP|S|NP   | subject  
VB|VP|NP    | object   
VB|VP|VP|S|NP   | subject (embedded VP)  
VB|VP|ADVP  | adverbial adjunct  
NN|NP|NP|PP   | prepositional complement of noun
Results

- **Features:**
  - Path from target to filler
  - Filler’s syntactic type, headword, case
  - Target’s identity
  - Sentence voice, etc.
  - Lots of other second-order features

- **Gold vs parsed source trees**

<table>
<thead>
<tr>
<th>CORE</th>
<th>ARGMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Acc.</td>
</tr>
<tr>
<td>92.2</td>
<td>80.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CORE</th>
<th>ARGMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Acc.</td>
</tr>
<tr>
<td>84.1</td>
<td>66.5</td>
</tr>
</tbody>
</table>

- SRL is fairly easy on gold trees
- Harder on automatic parses

---

Interaction with Empty Elements
Empty Elements

- In the PTB, three kinds of empty elements:
  - Null items (usually complementizers)
  - Dislocation (WH-traces, topicalization, relative clause and heavy NP extraposition)
  - Control (raising, passives, control, shared argumentation)

- Need to reconstruct these (and resolve any indexation)

Example: English
Example: German

Types of Empties

<table>
<thead>
<tr>
<th>Antecedent</th>
<th>POS</th>
<th>Label</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>NP</td>
<td>*</td>
<td>18,334</td>
<td>NP trace (e.g., Sam was seen)</td>
</tr>
<tr>
<td></td>
<td>NP</td>
<td>*</td>
<td>9,812</td>
<td>NP PRO (e.g., * to sleep is nice)</td>
</tr>
<tr>
<td>WHNP</td>
<td>NP</td>
<td><em>T</em></td>
<td>8,620</td>
<td>WH trace (e.g., the woman who you saw)</td>
</tr>
<tr>
<td></td>
<td><em>U</em></td>
<td>7,478</td>
<td></td>
<td>Empty units (e.g., $25 *U&quot;)</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td><em>T</em></td>
<td>5,635</td>
<td>Empty complementizers (e.g., Sam said to Sasha someone)</td>
</tr>
<tr>
<td>WHADVVP</td>
<td>ADVP</td>
<td><em>T</em></td>
<td>4,063</td>
<td>Moved clauses (e.g., Sam had to go, Sasha explained)</td>
</tr>
<tr>
<td>SBAR</td>
<td></td>
<td></td>
<td>2,492</td>
<td>WH-trace (e.g., Sam explained how to leave)</td>
</tr>
<tr>
<td>WHNP</td>
<td></td>
<td></td>
<td>2,033</td>
<td>Empty clauses (e.g., Sam had to go, Sasha explained)</td>
</tr>
<tr>
<td>WHADVVP</td>
<td></td>
<td></td>
<td>1,759</td>
<td>Empty relative pronouns (e.g., the woman we saw)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>573</td>
<td>Empty relative pronouns (e.g., no reason 0 to leave)</td>
</tr>
</tbody>
</table>
A Pattern-Matching Approach

- [Johnson 02]

Pattern-Matching Details

- Something like transformation-based learning
- Extract patterns
  - Details: transitive verb marking, auxiliaries
  - Details: legal subtrees
- Rank patterns
  - Pruning ranking: by correct / match rate
  - Application priority: by depth
- Pre-order traversal
- Greedy match
Top Patterns Extracted

<table>
<thead>
<tr>
<th>Count</th>
<th>Match</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>5816</td>
<td>6233</td>
<td>(S (NP (¬NONE- *) VP)</td>
</tr>
<tr>
<td>5005</td>
<td>7895</td>
<td>(SBAR (¬NONE- 0) S )</td>
</tr>
<tr>
<td>5312</td>
<td>5338</td>
<td>(SBAR WHNP-1 (S (NP (¬NONE- T*-1)) VP))</td>
</tr>
<tr>
<td>4434</td>
<td>5317</td>
<td>(NP CP (¬NONE- U*))</td>
</tr>
<tr>
<td>1652</td>
<td>1682</td>
<td>(NP S CD (¬NONE- U*))</td>
</tr>
<tr>
<td>1222</td>
<td>1593</td>
<td>(VP VBNz (NP (¬NONE- *)) PP)</td>
</tr>
<tr>
<td>700</td>
<td>700</td>
<td>(ADJP CP (¬NONE- U*))</td>
</tr>
<tr>
<td>662</td>
<td>1219</td>
<td>(SBAR WHNP-1 (¬NONE- 0) (S (NP (¬NONE- T*-1)) VP))</td>
</tr>
<tr>
<td>618</td>
<td>635</td>
<td>(S S-1 , NP (VP VBD (SBAR (¬NONE- 0) (S (¬NONE- T*-1)))) .)</td>
</tr>
<tr>
<td>499</td>
<td>512</td>
<td>(SINV &quot; &quot; S-1 , &quot; (VP VBD (S (¬NONE- T*-1))) NE .)</td>
</tr>
<tr>
<td>361</td>
<td>369</td>
<td>(SINV &quot; &quot; S-1 , &quot; (VP VBD (S (¬NONE- T*-1))) NP .)</td>
</tr>
<tr>
<td>352</td>
<td>320</td>
<td>(S NP-1 (VP VBD (S (NP (¬NONE- T*-1)) VP)))</td>
</tr>
<tr>
<td>346</td>
<td>273</td>
<td>(S NP-1 (VP AUX (VP VBNz (NP (¬NONE- T*-1)) PP)))</td>
</tr>
<tr>
<td>322</td>
<td>467</td>
<td>(VP VBDz (NP (¬NONE- *)) PP)</td>
</tr>
<tr>
<td>269</td>
<td>275</td>
<td>(S &quot; &quot; S-1 , &quot; NP (VP VBD (S (¬NONE- T*-1))) .)</td>
</tr>
</tbody>
</table>

Results

<table>
<thead>
<tr>
<th>Empty node POS</th>
<th>Section 23 P</th>
<th>R</th>
<th>f</th>
<th>Parser output P</th>
<th>R</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Overall)</td>
<td>0.93</td>
<td>0.83</td>
<td>0.88</td>
<td>0.85</td>
<td>0.74</td>
<td>0.79</td>
</tr>
<tr>
<td>NP *</td>
<td>0.95</td>
<td>0.87</td>
<td>0.91</td>
<td>0.86</td>
<td>0.79</td>
<td>0.82</td>
</tr>
<tr>
<td>NP <em>T</em></td>
<td>0.93</td>
<td>0.88</td>
<td>0.91</td>
<td>0.85</td>
<td>0.77</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.94</td>
<td>0.99</td>
<td>0.96</td>
<td>0.89</td>
<td>0.88</td>
</tr>
<tr>
<td>S <em>U</em></td>
<td>0.92</td>
<td>0.98</td>
<td>0.95</td>
<td>0.87</td>
<td>0.96</td>
<td>0.92</td>
</tr>
<tr>
<td>S <em>T</em></td>
<td>0.98</td>
<td>0.83</td>
<td>0.90</td>
<td>0.97</td>
<td>0.81</td>
<td>0.88</td>
</tr>
<tr>
<td>ADJP <em>T</em></td>
<td>0.91</td>
<td>0.52</td>
<td>0.66</td>
<td>0.84</td>
<td>0.42</td>
<td>0.56</td>
</tr>
<tr>
<td>SBAR</td>
<td>0.90</td>
<td>0.63</td>
<td>0.74</td>
<td>0.88</td>
<td>0.58</td>
<td>0.70</td>
</tr>
<tr>
<td>WHNP 0</td>
<td>0.75</td>
<td>0.79</td>
<td>0.77</td>
<td>0.48</td>
<td>0.46</td>
<td>0.47</td>
</tr>
</tbody>
</table>
A Machine-Learning Approach

- [Levy and Manning 04]
- Build two classifiers:
  - First one predicts where empties go
  - Second one predicts if/where they are bound
  - Use syntactic features similar to SRL (paths, categories, heads, etc)

<table>
<thead>
<tr>
<th></th>
<th>Performance on gold trees</th>
<th>Performance on parsed trees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ID</td>
<td>Rel</td>
</tr>
<tr>
<td>WSJ (full)</td>
<td>92.0</td>
<td>82.9</td>
</tr>
<tr>
<td>WSJ (sem)</td>
<td>92.3</td>
<td>79.5</td>
</tr>
<tr>
<td>NEGRA</td>
<td>72.9</td>
<td>64.6</td>
</tr>
</tbody>
</table>