**Natural Language Processing**

**Parsing IV**
Dan Klein – UC Berkeley

---

**Dependency Parsing**

- Lexicalized parsers can be seen as producing dependency trees
  - Each local binary tree corresponds to an attachment in the dependency graph

- Pure dependency parsing is only cubic \([\text{Eisner 99]}\)

**Dependency Parsing**

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

---

**Shift-Reduce Parsers**

- Another way to derive a tree:

  - Parsing
    - No useful dynamic programming search
    - Can still use beam search [Ratnaparkhi 97]

---

**Tree Insertion Grammars**

- 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**

- Start with local trees
- Can insert structure with adjunction operators
- Mildly context-sensitive
- Models long-distance dependencies naturally

**Tree-adjoining grammars**

- As well as other weird stuff that CFGs don’t capture well (e.g., cross-serial dependencies)

**TAG: Long Distance**

- 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?)

**CCG Parsing**

- John \(\vdash\) NP
- shares \(\vdash\) NP
- buys \(\vdash\) \((S(NP))\)/NP
- sleeps \(\vdash\) \(S\)/NP
- well \(\vdash\) \((S\)/NP)\/(S\)/NP

**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

- **Antecedent**
- **POS**
- **Label**
- **Count**
- **Description**

<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>VP</td>
<td></td>
<td>3549</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>VBD</td>
<td></td>
<td>1852</td>
<td></td>
</tr>
<tr>
<td>NP/PRP</td>
<td>(e.g., &quot;she is here&quot;)</td>
<td>748</td>
<td>Empty words (e.g., &quot;is&quot;, &quot;to&quot;)</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>NNP</td>
<td></td>
<td>3914</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>NN</td>
<td></td>
<td>1505</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>(e.g., &quot;in the house&quot; (&quot;in&quot;)</td>
<td>1799</td>
<td>Empty relative pronouns (e.g., &quot;in the house&quot;)</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>(e.g., &quot;in the house&quot;)</td>
<td>1799</td>
<td>Empty relative pronouns (e.g., &quot;in the house&quot;)</td>
<td></td>
</tr>
</tbody>
</table>

- **Pattern-Matching Approach**

  - [Johnson 02]

  - A Pattern-Matching Approach

  - **Details:**
    - Transitive verb marking, auxiliaries
    - Legal subtrees

  - **Rank patterns**

  - **Pre-order traversal**

  - Greedy match

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

- **Greedy match**

Top Patterns Extracted

<table>
<thead>
<tr>
<th>Count</th>
<th>Match</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>2810</td>
<td>6283</td>
<td>(a HP ((HOLE - ?) VP))</td>
</tr>
<tr>
<td>6464</td>
<td>7602</td>
<td>(LBAR (HOLE - 0) (P))</td>
</tr>
<tr>
<td>5312</td>
<td>5936</td>
<td>(LBAR ITMP - 1) (S HP ((HOLE - &quot;?&quot;)) VP))</td>
</tr>
<tr>
<td>4434</td>
<td>5037</td>
<td>(HP UP ((HOLE - ?)))</td>
</tr>
<tr>
<td>1662</td>
<td>1662</td>
<td>(LIA 2 CD ((HOLE - ?)))</td>
</tr>
<tr>
<td>1257</td>
<td>1456</td>
<td>(VP VBD ((HOLE - ?) PP))</td>
</tr>
<tr>
<td>709</td>
<td>709</td>
<td>(LIA 1 CD ((HOLE - ?)))</td>
</tr>
<tr>
<td>462</td>
<td>1209</td>
<td>(LBAR (HOLE - 1) (P HP ((HOLE - &quot;?&quot;))) VP))</td>
</tr>
<tr>
<td>418</td>
<td>655</td>
<td>((a HP ((HOLE - 0) (P HP ((HOLE - &quot;?&quot;))))))</td>
</tr>
<tr>
<td>499</td>
<td>512</td>
<td>((a HP ((HOLE - &quot;?&quot;)) (VP VBD ((HOLE - &quot;?&quot;))))))</td>
</tr>
<tr>
<td>396</td>
<td>564</td>
<td>(a HP ((HOLE - &quot;?&quot;)) (VP VBD ((HOLE - &quot;?&quot;))))</td>
</tr>
<tr>
<td>354</td>
<td>335</td>
<td>(a HP ((HOLE - &quot;?&quot;)) (VP VBD ((HOLE - &quot;?&quot;))))</td>
</tr>
<tr>
<td>246</td>
<td>275</td>
<td>(a HP ((HOLE - &quot;?&quot;)) (VP VBD ((HOLE - &quot;?&quot;))))</td>
</tr>
<tr>
<td>208</td>
<td>275</td>
<td>(a HP ((HOLE - &quot;?&quot;)) (VP VBD ((HOLE - &quot;?&quot;))))</td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Empty node</th>
<th>Section 23</th>
<th>Parser output</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS Label</td>
<td>P R</td>
<td>f</td>
</tr>
<tr>
<td>(Overall)</td>
<td>0.91 0.83</td>
<td>0.77</td>
</tr>
<tr>
<td>NP</td>
<td>0.95 0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>NP T*</td>
<td>0.93 0.88</td>
<td>0.91</td>
</tr>
<tr>
<td>0</td>
<td>0.91 0.99</td>
<td>0.96</td>
</tr>
<tr>
<td>V*</td>
<td>0.92 0.98</td>
<td>0.95</td>
</tr>
<tr>
<td>S T*</td>
<td>0.98 0.83</td>
<td>0.90</td>
</tr>
<tr>
<td>ADV N</td>
<td>0.91 0.52</td>
<td>0.66</td>
</tr>
<tr>
<td>NN</td>
<td>0.90 0.63</td>
<td>0.74</td>
</tr>
<tr>
<td>ADJ</td>
<td>0.75 0.79</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Semantic Roles

Semantic Role Labeling (SRL)

- Characterize clauses as relations with roles:
  
 _busy: She (blames the Government) for failing to do enough to help.

  Holmes would characterize this as blaming [the poor].

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

- Says 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

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 fewer roles than FrameNet (e.g. 5-20 total: agent, patient, instrument, etc.)

SRL Example

PropBank Example
PropBank Example

rotate.92: sense: shift from one thing to another
roles:
arg0: cause 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 returned their buying to other issues, traders said.
arg0: investors
ref: returned
arg1: their buying
arg3: to other issues

PropBank Example

aim.01: sense: introd, plan
roles:
Arg0: story
Arg1: plan, intro
Arg2: setting

The Central Council of Church Bell Ringers aims "trace" to improve relations with vicars.
arg0: The Central Council of Church Bell Ringers
arg1: "trace"
arg2: to improve relations with vicars

aim.02: sense: point (weapon at)
roles:
Arg0: agent
Arg1: weapon, etc.
Arg2: target
Arg3: at the elderly

Banks have been aiming packages at the elderly.

Shared Arguments

(NP-SBV) (JJ massive) (JJ internal) (NN debt)

(VP (VBN forget))

(S
(NP-SBV1 (DT the) (NN government))

(VP (TO be)

(ADVP-NNR (RB massively)))

Path Features

Path | Description
---|---
V1|PP argument/adjunct
V1|VP|S|NP | subject
V2|VP|NP | object
V3|VP|V2|S|NP | subject (embedded VP)
N2|VP|ADVP | adverbial adjunct
N2|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
  - SRL is fairly easy on gold trees
  - Harder on automatic parses

Empties and SRL