CS395T: Structured Models for NLP
Lecture 8: Trees 2

Greg Durrett

Some slides adapted from Dan Klein, UC Berkeley
Administrivia

- Project 1 due Thursday at 9:30am
Lexicalized and state-split constituency parsing (slides from last time)

Dependency representation

Contrast with constituency

Projectivity
Lexicalized Parsing

S(ran)
  VP(ran)
    NP(dog)
      DT(the) the
      NN(dog) dog
    VBD(ran) ran
    TO(to) to
      PP(to)
        NP(house)
          DT(the) the
          NN(house) house

Dependency Parsing

- Dependency syntax: syntactic structure is defined by dependencies
  - Head (parent, governor) connected to dependent (child, modifier)
  - Each word has exactly one parent except for the ROOT symbol
  - Dependencies must form a directed acyclic graph
Still a notion of hierarchy!

Can still derive constituents (subtrees)
Can label dependencies according to syntactic function

Major source of ambiguity is in the structure, so we focus on that more (labeling separately with a classifier works pretty well)
Constituency: several rule productions need to change
Dependency vs. Constituency: PP Attachment

- Dependency: one word (with) assigned a different parent

  the children ate the cake with a spoon

- More predicate-argument focused view of syntax

- “What’s the main verb of the sentence? What is its subject and object?” — easier to answer under dependency parsing
Dependency vs. Constituency: Coordination

- Constituency: ternary rule NP -> NP CC NP

Diagram: 
```
  NP
 / \  
NP   CC
 /    
NP    NP
 /  
NNS   NNS
|    |
dogs in
/   
NNS houses
```
```
  NP
 / 
NP PP
 /  
NNS IN NP
|   |
dogs in NNS
dogs in houses
```
```
  NP
 / 
PP IN NP
 /  
NNS in NNS
dogs in cats
```
Dependency vs. Constituency: Coordination

- Dependency: first item is the head

- Coordination is decomposed across a few arcs as opposed to being a single rule production as in constituency

- Can also choose *and* to be the head

- Both cases: headword doesn’t really represent the phrase
Stanford Dependencies

- Designed to be practically useful for relation extraction

Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas

- Standard
-Collapsed
Dependency vs. Constituency

- Dependency is often more useful in practice (models predicate argument structure)

- Slightly different representational choices:
  - PP attachment is better modeled under dependency
  - Coordination is better modeled under constituency

- Dependency parsers are easier to build: no “grammar engineering”, no unaries, easier to get structured discriminative models working well

- Dependency parsers are usually faster

- Dependencies are more universal cross-lingually
Annotate dependencies with the same representation in many languages

[Diagram showing dependencies in English, Bulgarian, Czech, and Swiss German]

http://universaldependencies.org/
What conditions have to hold for things to be tree-shaped?

Any subtree is a contiguous span of the sentence <-> tree is projective
Projectivity

- Projective $\leftrightarrow$ no “crossing” arcs
  - dogs in houses and cats
  - the dog ran to the house

- Crossing arcs:

- Extraposition: A hearing on the issue is scheduled today. is projective

credit: Language Log
Projectivity

- More extraposition

John was not as good for the job as Kate

- Time expressions can go a lot of places in sentences!

Gomez-Rodriguez et al.; Jurafsky+Martin
Projectivity

- Number of trees produceable under different formalisms

<table>
<thead>
<tr>
<th></th>
<th>Arabic</th>
<th>Czech</th>
<th>Danish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projective</td>
<td>1297 (88.8)</td>
<td>55872 (76.8)</td>
<td>4379 (84.4)</td>
</tr>
<tr>
<td>Sentences</td>
<td>1460</td>
<td>72703</td>
<td>5190</td>
</tr>
</tbody>
</table>

- Many trees in other languages are nonprojective

Pilter et al. (2013)
Projectivity

- Number of trees produceable under different formalisms

<table>
<thead>
<tr>
<th></th>
<th>Arabic</th>
<th>Czech</th>
<th>Danish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Endpoint-Crossing</td>
<td>1457 (99.8)</td>
<td>71810 (98.8)</td>
<td>5144 (99.1)</td>
</tr>
<tr>
<td>Well-nested, block degree 2</td>
<td>1458 (99.9)</td>
<td>72321 (99.5)</td>
<td>5175 (99.7)</td>
</tr>
<tr>
<td>Gap-Minding</td>
<td>1394 (95.5)</td>
<td>70695 (97.2)</td>
<td>4985 (96.1)</td>
</tr>
<tr>
<td>Projective</td>
<td>1297 (88.8)</td>
<td>55872 (76.8)</td>
<td>4379 (84.4)</td>
</tr>
<tr>
<td>Sentences</td>
<td>1460</td>
<td>72703</td>
<td>5190</td>
</tr>
</tbody>
</table>

- Many trees in other languages are nonprojective

- Some other formalisms (that are harder to parse in), most useful one is 1-Endpoint-Crossing

Pitler et al. (2013)
Projectivity

1-Endpoint-Crossing: for any edge, all edges that cross it share an endpoint.

John was not as good for the job as Kate.

- True
- False: hearing -> on

Captures most cases, still efficient parsing algorithms.