LING/C SC 581: Advanced Computational Linguistics

Lecture 11
Feb 22nd
Today's Topics

• Treebanks and Statistical parsers
• Install the Bikel-Collins Parser
Relevance of Treebanks

• Statistical parsers typically construct syntactic phrase structure
  • they’re trained on Treebank corpora like the Penn Treebank

• **Note:** some use dependency graphs, not trees
Berkeley Parser

Note: no empty categories, or subtags ...
Stanford Parser

Please enter a sentence to be parsed:
Which report did you file without reading?

Language: English

Sample Sentence

Parse

(ROOT
  (S BARQ
    (WHNP
      (WHNP (WDT Which))
      (NP (NN report)))
    (SQ (VBD did))
    (NP (PRP you))
    (VP (VB file))
    (PP (IN without)
      (NP (NN reading))))
  (.
  ?))
)

Typed dependencies

det(report-2, Which-1)
dobj(file-5, report-2)
aux(file-5, did-3)
nsubj(file-5, you-4)
prep_without(file-5, reading-7)
Statistical Parsers

• Don’t recover fully-annotated trees
  • not trained using nodes with indices or empty (−NONE−) nodes
  • not trained using functional tags, e.g. −SBJ, −PRD

• Therefore they don’t fully parse the PTB
Parsers trained on the Treebank

• SBAR can be forced by the presence of an overt relative pronoun, but note there is no subject gap:
Parsers trained on the Treebank

- Probabilities are estimated from frequency information of each node given surrounding context (e.g. parent node, or the word that heads the node)
- Still these systems have enormous problems with prepositional phrase (PP) attachment

**Examples:** (borrowed from Igor Malioutov)

- A *boy with a telescope kissed Mary on the lips*
- *Mary was kissed by a boy with a telescope on the lips*
  - PP *with a telescope* should adjoin to the noun phrase (NP) *a boy*
  - PP *on the lips* should adjoin to the verb phrase (VP) headed by *kiss*
Active/passive sentences

• Examples using the Stanford Parser:

  Your query
  
  A boy with a telescope kissed Mary on the lips.

  Your query
  
  Mary was kissed by a boy with a telescope on the lips.

Both active and passive sentences are parsed incorrectly
Active/passive sentences

• Examples using the Stanford Parser:

Your query
Mary was kissed by a boy with a telescope on the lips.

Parse

(ROOT
  (S
    (NP (NNP Mary))
    (VP (VBD was)
      (VP (VBN kissed)
        (PP (IN by)
          (NP
            (NP (DT a) (NN boy))
            (PP (IN with)
              (NP
                (NP (DT a) (NN telescope))
                (PP (IN on)
                  (NP (DT the) (NNS lips))))))))))

Both active and passive sentences are parsed incorrectly
Active/passive sentences

• Examples:

**Your query**

*A boy with a telescope kissed Mary on the lips.*

Typed dependencies, collapsed

- det(boy-2, A-1)
- nsubj(kissed-6, boy-2)
- det(telescope-5, a-4)
- prep_with(boy-2, telescope-5)
- root(ROOT-0, kissed-6)
- dobj(kissed-6, Mary-7)
- det(lips-10, the-9)
- **prep_on(Mary-7, lips-10)**

**Your query**

*Mary was kissed by a boy with a telescope on the lips.*

Typed dependencies, collapsed

- nsubjpass(kissed-3, Mary-1)
- auxpass(kissed-3, was-2)
- root(ROOT-0, kissed-3)
- det(boy-6, a-5)
- agent(kissed-3, boy-6)
- det(telescope-9, a-8)
- prep_with(boy-6, telescope-9)
- det(lips-12, the-11)
- **prep_on(telescope-9, lips-12)**

**X on the lips modifies Mary**

**X on the lips modifies telescope**
Treebank Rules

• Just how many rules are there in the WSJ treebank?

• What’s the most common POS tag?
• What’s the most common syntax rule?
Penn Treebank

• How many grammar rules are there in the treebank?

Treebank Grammar Rules

<table>
<thead>
<tr>
<th>Total number of rules:</th>
<th>978,873</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of different rules:</td>
<td>31,338</td>
</tr>
</tbody>
</table>

Top 5:
1. S → NP-SBJ VP
2. PP → IN NP
3. NP-SBJ → -NONE-
4. NP → NP PP
5. NP → DT NN
Treebank

Treebank Grammar Rules

Percentage

Rule
Treebank

Treebank Grammar Rules

Total # of rules: 978,873
# of different rules: 17,554
Treebank

Treebank Grammar Rules

Rules

Percentage
Penn Treebank

• How about POS tags?

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>355,039</td>
<td>28.3%</td>
</tr>
<tr>
<td>V</td>
<td>154,975</td>
<td>12.4%</td>
</tr>
</tbody>
</table>

WSJ POS tag frequencies

**Total number of tags: 1,253,013**

**Top 5:**
1. NN  singular common noun
2. IN  preposition
3. NNP singular proper noun
4. DT  determiner
5. JJ  adjective
Using the Treebank

- What is the grammar of the Treebank?
  - We can extract the phrase structure rules used, and
  - count the frequency of rules, and construct a stochastic parser
Using the Treebank

• Breakthrough in parsing accuracy with \textit{lexicalized} trees
  • think of expanding the nonterminal names to include head information and the words that are at the leaves of the subtrees.

\textbf{Figure 8}
A fully lexicalized tree. The \textit{VP} node is the head child of \textit{S}.
Bikel Collins Parser

Michael Collins’ parsing models (Collins, 1996; Collins, 1997; Collins, 1999) have been quite influential in the field of natural language processing. Not only did they achieve new performance benchmarks on parsing the Penn Treebank (Marcus, Santorini, and Marcinkiewicz, 1993),

• Java re-implementation of Collins’ parser (originally in C)
• Paper
• Software
  • http://www.cis.upenn.edu/~dbikel/software.html#stat-parser (page no longer exists)
Bikel Collins

• Download and install Dan Bikel’s parser
  • dbp.zip (on course homepage)
Bikel Collins

• Training the parser with the WSJ PTB
• See guide
  • userguide/guide.pdf

  (a) If you have access to the Penn Treebank, execute the following commands to create the file /tmp/wsj-02-21.obj.gz:

  i. cd <Penn Treebank home>/combined/wsj
  ii. cat 0[2-9]/*.mrg 1[0-9]/*.mrg 2[01]/*.mrg > /tmp/wsj-02-21.mrg
  iii. train 800 <dbparser home>/settings/collins.properties \ /tmp/wsj-02-21.mrg

  directory: TREEBANK_3/parsed/mrg/wsj
  chapters 02-21: create one single .mrg file
  events: wsj-02-21.obj.gz
Bikel Collins

• Settings:
**Bikel Collins**

**Parsing**

- Command

  `parse <max. heap> <settings> <derived data file> \ <input file>`

- Input file format (sentences)

  The input file should have one of two Lisp-style formats:

  1. `((word1 (pos1)) (word2 (pos2)) ... (wordN (posN)))`

  2. `(word1 word2 ... wordN)` Here is the first sentence of Section 00 of the WSJ Penn Treebank in Format 1:

    `((Pierre (NNP)) (Vinken (NNP)) (, (,)) (61 (CD)) (years (NNS)) (old (JJ)) (, (,)) (will (MD)) (join (VB)) (the (DT)) (board (NN)) (as (IN)) (a (DT)) (nonexecutive (JJ)) (director (NN)) (Nov. (NNP)) (29 (CD)) (. (.))))`
Bikel Collins

• Verify the trainer and parser work on your machine
Bikel Collins

• File: `bin/parse` is a shell script that sets up program parameters and calls `java`

```bash
#!/ figuring out our location
define dir = 'dirname $0'
define scriptDir = 'cd $dir ; echo $cmd'
define parent = 'cd $scriptDir/.. ; echo $cmd'

# set java
set java = (java -server)
set class = danbikel.parser.Parser
set maxHeap = $1
set minHeap = $maxHeap
set settingsFile = $2
set settings = "-Dparser.settingsFile=$settingsFile"
set jarFile = $parent/dbparse.jar
set classpath = $jarFile
if ($?CLASSPATH) then
  set classpath = $jarFile:$CLASSPATH
endif
set derivedDataFile = $3
set inFile = $4
set command = ($java -Xms$minHeap\m -Xmx$maxHeap\m -cp $classpath $settings \ $class -is $derivedDataFile -sa $inFile)

echo "" exec echo "Executing command..." base echo "$command" exec echo "" exec
```
5.2 Parsing

The Java class used for parsing is danbikel.parser.Parser. A typical usage is as follows (assumes you have dbparser.jar in your class path):

```
java -Xms400m -Xmx400m -Dparser.settingsFile=<settings> \n    danbikel.parser.Parser -is <derived data file> \n    -sa <sentence input file>
```

You can see its full usage by executing

```
java danbikel.parser.Parser -help
```
Bikel Collins

• **File: bin/train** is another shell script

```bash
#set java = java
set java = (java -server)
set class = danbikel.parser.Trainer
set maxHeap = $1
set minHeap = $maxHeap
set settingsFile = $2
set settings = "-Dparser.settingsFile=$settingsFile"
set jarFile = $parent/dbparser.jar
set classpath = $jarFile
if ($?CLASSPATH) then
  set classpath = $jarFile:$CLASSPATH
endif
set inFile = $3
set oFile = inFile:r.observed.gz
set odFile = inFile:r.obj.gz

set command = ($java -Xms$minHeap\n-Xmx$maxHeap\n -cp $classpath $settings \n-class -i inFile -o oFile -od odFile)

echo ""
echo Executing command
echo "\"$command""
echo ""
$command
```
5.1 Training

The Java class used for training is danbikel.parser.Trainer. A typical usage is as follows (assumes you have dbparser.jar in your class path):

```
java -Xms800m -Xmx800m -Dparser.settingsFile=<settings> \
    danbikel.parser.Trainer -i <training file> \
    -o <observed file> -od <derived data file>
```

You can see its full usage by executing

```
java danbikel.parser.Trainer -help
```

If either <observed file> or <derived data file> ends with the extension .gz it is automatically compressed before being written to disk.

• Relevant WSJ PTB files

<table>
<thead>
<tr>
<th>File</th>
<th>Date/Time</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsj-02-21.mrg</td>
<td>Dec 13, 2006 3:14 PM</td>
<td>27.5 MB</td>
</tr>
<tr>
<td>wsj-02-21.obj.gz</td>
<td>Dec 12, 2006 5:16 PM</td>
<td>22.2 MB</td>
</tr>
<tr>
<td>wsj-02-21.observed</td>
<td>Mar 12, 2009 1:09 PM</td>
<td>91.8 MB</td>
</tr>
<tr>
<td>wsj-02-21.observed.gz</td>
<td>Dec 13, 2006 3:07 PM</td>
<td>13 MB</td>
</tr>
</tbody>
</table>