Administrivia

• We need to reboot the homework (*my fault*: explained below)
• You get full credit for the Homework (treat as done)
• Let's redo the homework as new Homework 7
Preparing the Training Data

![Training data size graph]

Number of sentences vs Number of sections
Goal: get the graph

• What does the graph for the F-score (Labeled Recall/Labeled Precision) look like?

```
F-score

# of sentences/sections used for training

report your results
```
A problem with the Homework

- **Symptom:**
  - Lots of nulls instead of parses returned by Bikel-Collins

- **Cause:**
  - Bikel-Collins doesn't do so well when the POS tags are not supplied
  - also fails to tag --: ( --)

```markdown
1920 [\([\text{NNP} +\text{unknown}+] \text{ (NNP Oregon)}\)]
```

- **Solution:**
  - Let's supply the gold tags
Initial fix: nevalb.c

- **e n** number of errors to kill (default=10)
- **MAX_ERROR 10** in COLLINS.PRM (100 errors ok)

- Modify evalb.c to not skip the null parses
- nevalb.c counts them as if the parser had simply returned \((W_1 \ldots W_n)\)
- this kills the recall for those cases (but- well - precision is 100%)
- Unfortunately, the number of nulls is impossible to ignore – despite increasing the number of sections dedicated to training ...
Creating section 23 input

Format now is ((Word₁ (Tag₁)) .. (Wordₙ (Tagₙ)))
Creating section 23 input

Can use `.tagged_sents()` from the ptb corpus that we already have in nltk

```python
import nltk
from nltk.corpus import ptb
# cat wsj_[0-1][0-9].gold wsj_2[0-2].gold | wc -l
# 45446
# see also lectures 4 and 7
excluded = set(["-NONE-"])
sec23_raw = ptb.tagged_sents(categories=['news'])[45446:45446+2416]
sec23 = [[wt for wt in sent if wt[1] not in excluded] for sent in sec23_raw]
for sent in sec23:
    print("("+".join('{{}} {{{}}}'.format(x[0],x[1]) for x in sent)+")")
```

`extract_23.py`

```bash
python3 extract_23.py > wsj_23.lsp
```
Splitting section 23 input

Manageable chunks of 650 sentences each?

```
$ python3 extract_23.py > wsj_23.lsp
$ split -l 650 wsj_23.lsp wsj_23-
$ wc -l wsj_23-*
 650 wsj_23-aa
 650 wsj_23-ab
 650 wsj_23-ac
 466 wsj_23-ad
 2416 wsj_23-words_only
 4832 total
```

```
$ split -l 101 wsj_23.lsp wsj_23-
$ wc -l wsj_23-*
 101 wsj_23-aa
 101 wsj_23-ab
 101 wsj_23-ac
 101 wsj_23-ad
 101 wsj_23-ae
 101 wsj_23-af
 101 wsj_23-ag
 101 wsj_23-ah
 101 wsj_23-ai
 101 wsj_23-aj
 101 wsj_23-ak
 101 wsj_23-al
```

```
dbp/bin/parse 1500 dbp/settings/collins.properties ptb/wsj-12.obj.gz ptb/wsj_23-aa
```

```
danbikel.parser.Parser -is ptb/wsj-12.obj.gz -sa ptb/wsj_23-aa
```

```
danbikel.parser.Parser -is ptb/wsj-12.obj.gz -sa ptb/wsj_23-aa
```

```
```
Parsing

• Platform-dependent: about 20 mins; 109 sentences in (5W CPU)
• Problem is memory used
  1. don't want to activate garbage collection
  2. don't want to activate memory compression
Re-combine section 23 output

- `cat wsj_23-a[a-w].parsed > wsj_23-12.parsed`

Then do `evalb`:

```
$ EVALB/nevalb -p EVALB/COLLINS.prm -e 100 ptb/wsj_23.gold ptb/wsj_23-12.parsed
```
Repeat parsing how many times?

- The following schedule is 20 repeats
- But you may use more sparse repeats, depending on your access to fast laptop(s) or lab machines
- Also see next slide
Only sentences ≤ 40 words long

```python
import sys
if len(sys.argv) != 2:
    print('Usage: python3 del_gt40.py file')
    sys.exit(1)

f = open('wsj_23.txt','r')
for line in f:
    i = 0
    gt40 = []
    l = len(line.strip().split())
    if l > 40:
        gt40.append(i)
        i += 1
    s = set(gt40)
print('Stats: {} out of {}, remain {}'.format(len(gt40),i,i-len(gt40)))
print('Create: 40wsj_23'.format(sys.argv[1]))
f.close()
```

Assume: `wsj_23.txt (supplied on course webpage)`
Only sentences <= 40 words long

40wsj_23--aa
Only sentences <= 40 words long

cat 40wsj_23-a[a-w].parsed > 40wsj_23-12.parsed

EVALB/nevalb -p EVALB/COLLINS.prm -e 100 ptb/40wsj_23-200.gold ptb/40wsj_23-200.parsed

```
1 dbp/bin/parse 1500 dbp/settings/collins.properties ptb/wsj-12.obj.gz ptb/40wsj_23-aa
2 dbp/bin/parse 1500 dbp/settings/collins.properties ptb/wsj-12.obj.gz ptb/40wsj_23-ab
3 dbp/bin/parse 1500 dbp/settings/collins.properties ptb/wsj-12.obj.gz ptb/40wsj_23-ac
4 dbp/bin/parse 1500 dbp/settings/collins.properties ptb/wsj-12.obj.gz ptb/40wsj_23-ad
```

Just testing 1st 200 on nevalb

<table>
<thead>
<tr>
<th>-- All --</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sentence</td>
<td>200</td>
</tr>
<tr>
<td>Number of Error sentence</td>
<td>0</td>
</tr>
<tr>
<td>Number of Skip sentence</td>
<td>0</td>
</tr>
<tr>
<td>Number of Valid sentence</td>
<td>200</td>
</tr>
<tr>
<td>Bracketing Recall</td>
<td>63.07</td>
</tr>
<tr>
<td>Bracketing Precision</td>
<td>67.47</td>
</tr>
<tr>
<td>Bracketing FMeasure</td>
<td>65.20</td>
</tr>
</tbody>
</table>