LING/C SC 581:

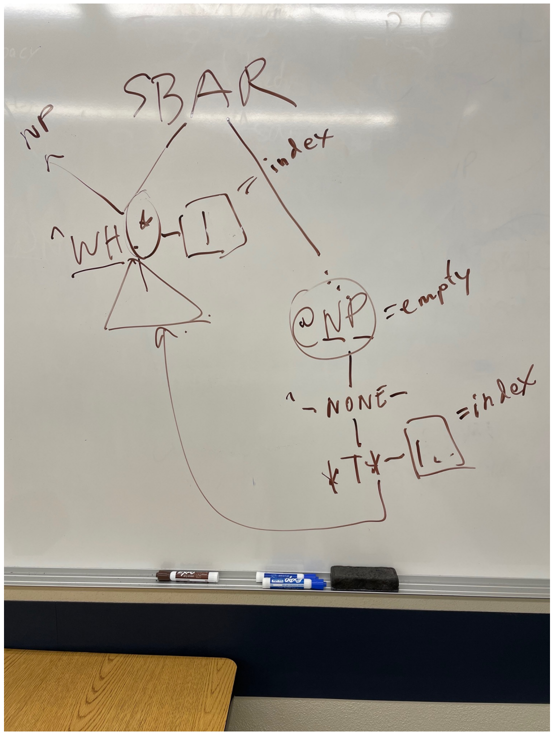
Advanced Computational Linguistics

Lecture 19

Today's Topic

- Last time we discussed the definition of c-command
- Let's put it to use!
- Homework 9
  - 6 Questions
  - we start doing Q1 live in class …
  - last one is for extra credit

Wh-movement



- Recall the in-class explanation of the tregex code from *Lecture 17*?

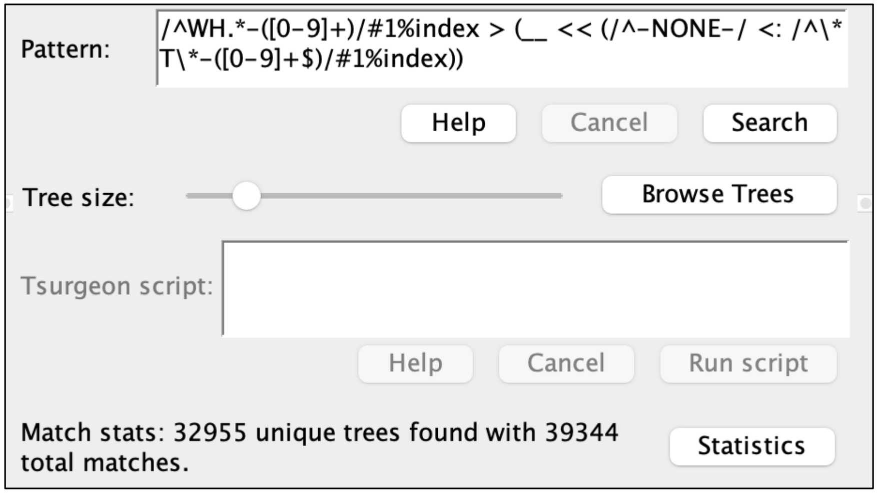
```
@SBAR < /^WH.*-([0-9]+)$/#1%index
<< (__ < (/^-NONE-/ < /^\*T\*-([0-
9]+)$/#1%index))
```

- Let's revise this code slightly for the following slides:

```
/^WH.*-([0-9]+)/#1%index > (__ <<
(/^-NONE-/ <: /^\*T\*-([0-
9]+$)/#1%index))
```

  - this will highlight the WH node in red.

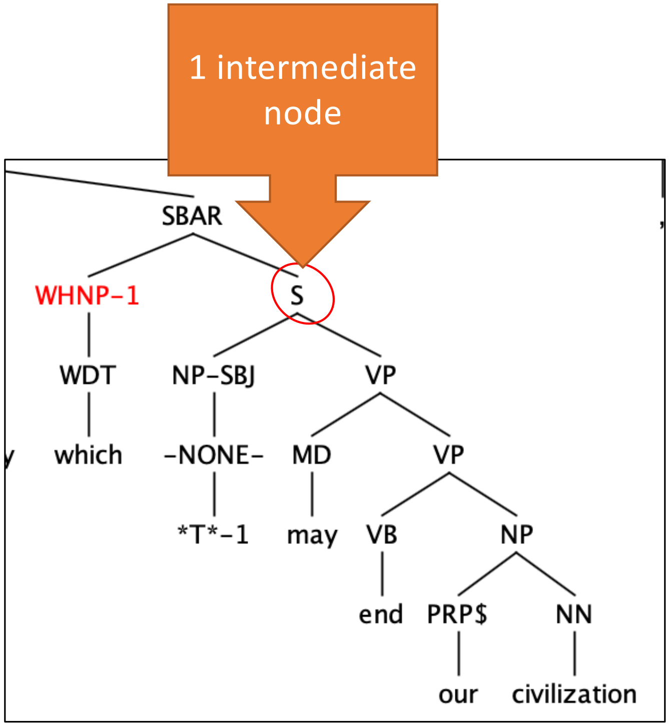
Wh-movement: example



Pattern:

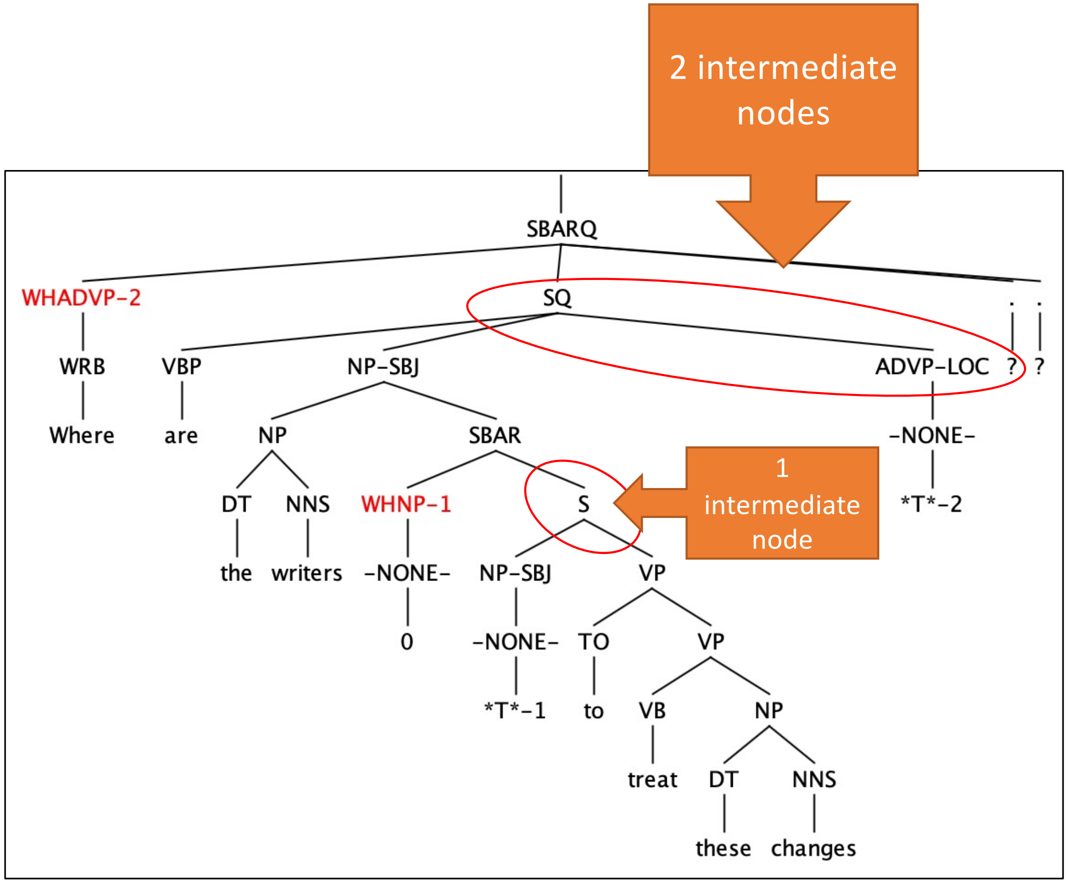
```
/^WH.*-([0-9]+)/#1%index > (__ << (/^-NONE-/ <: /^\*
T\*-([0-9]+$)/#1%index))
```

Match stats: 32955 unique trees found with 39344 total matches.

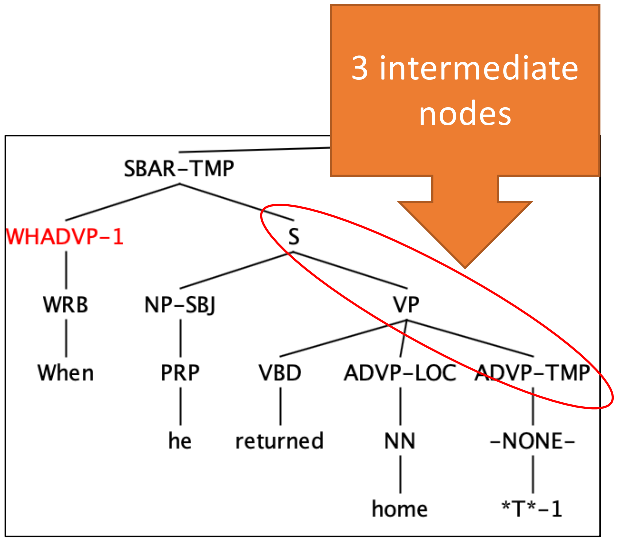


1 intermediate node

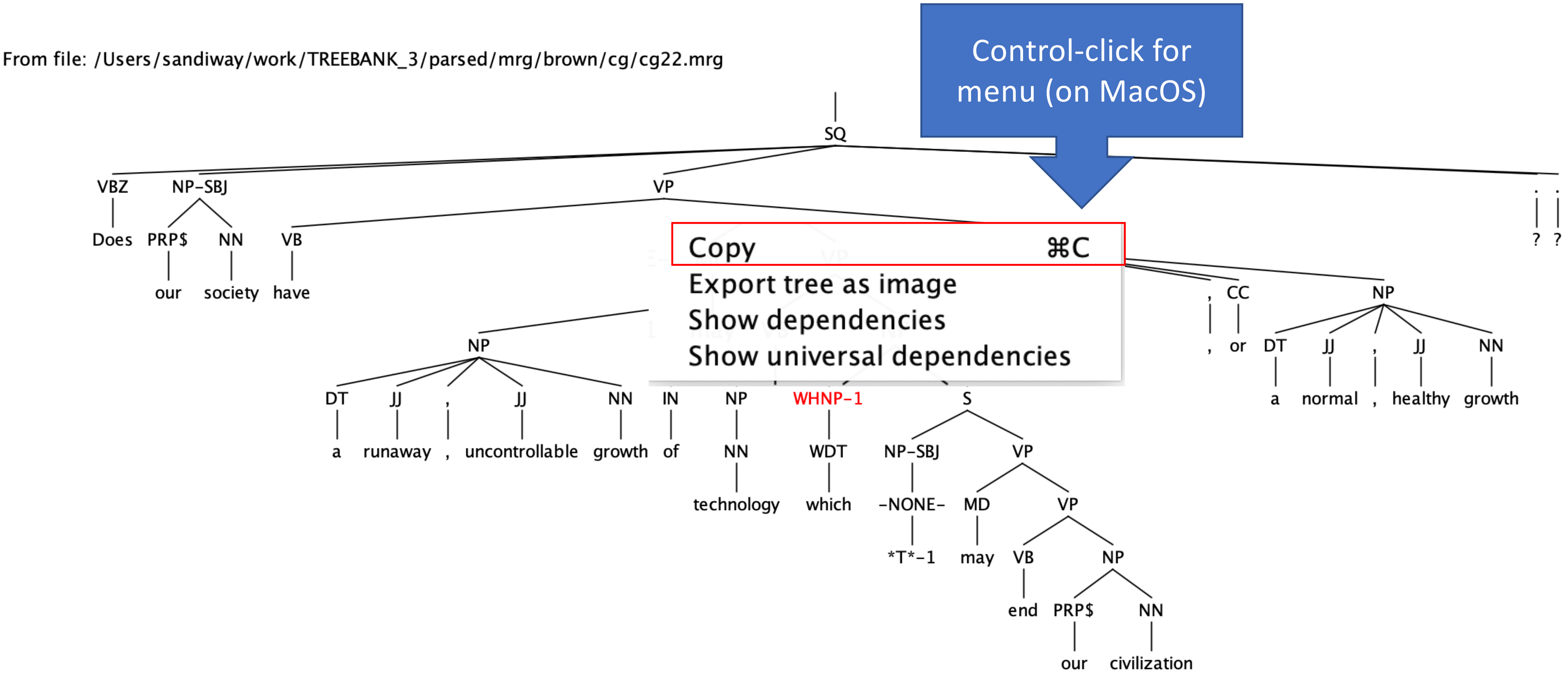
Wh-movement


2 intermediate nodes

1 intermediate node


3 intermediate nodes

Wh-movement: example



From file: /Users/sandiway/work/TREEBANK_3/parsed/mrg/brown/cg/cg22.mrg

Control-click for menu (on MacOS)

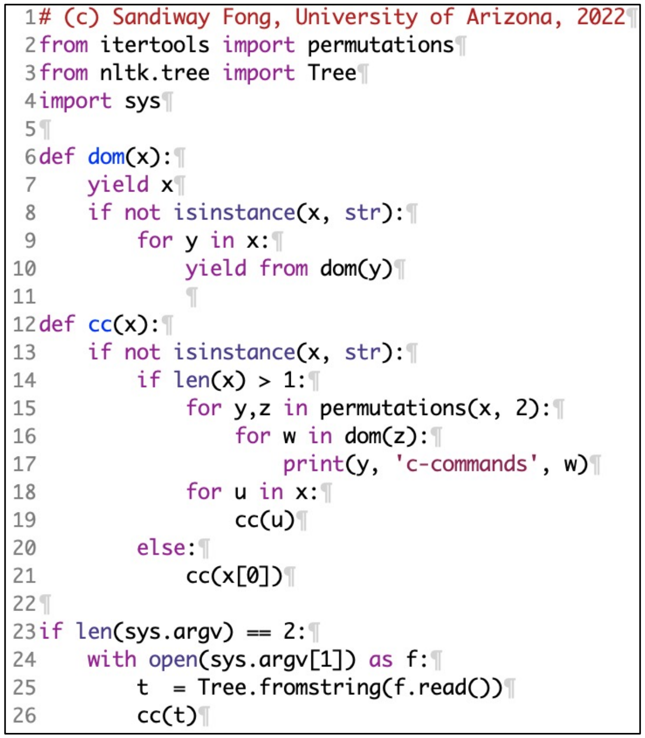
Wh-movement: example

Saved in file: tregex-whtrace-ex1.mrg

Pasted:

```
( (SQ (VBZ Does) tregex-whtrace-ex1.mrg
    (NP-SBJ (PRP$ our) (NN society))
    (VP (VB have)
      (NP
        (NP
          (NP (DT a) (JJ runaway) (, ,) (JJ uncontrollable) (NN growth))
          (PP (IN of)
            (NP (NN technology)))
          (SBAR
            (WHNP-1 (WDT which))
            (S
              (NP-SBJ (-NONE- *T*-1))
              (VP (MD may)
                (VP (VB end)
                  (NP (PRP$ our) (NN civilization)))))))
        (, ,)
        (CC or)
        (NP (DT a) (JJ normal) (, ,) (JJ healthy) (NN growth))))
    (. ?) (. ?)))
```

Wh-movement: example



```python
# (c) Sandiway Fong, University of Arizona, 2022
from itertools import permutations
from nltk.tree import Tree
import sys

def dom(x):
    yield x
    if not isinstance(x, str):
        for y in x:
            yield from dom(y)
            
def cc(x):
    if not isinstance(x, str):
        if len(x) > 1:
            for y,z in permutations(x, 2):
                for w in dom(z):
                    print(y, 'c-commands', w)
            for u in x:
                cc(u)
        else:
            cc(x[0])

if len(sys.argv) == 2:
    with open(sys.argv[1]) as f:
        t  = Tree.fromstring(f.read())
        cc(t)
```

ccommand2f.py

- Usage:
  - python ccommand2f.py tregex-whtrace-ex1.mrg

```
(VBZ Does) c-commands (NP-SBJ (PRP$ our) (NN society))
(VBZ Does) c-commands (PRP$ our)
(VBZ Does) c-commands our
(VBZ Does) c-commands (NN society)
(VBZ Does) c-commands society
(VBZ Does) c-commands (VP
  (VB have)
  (NP
    (NP
      (NP (DT a) (JJ runaway) (, ,) (JJ uncontrollable) (NN growth))
      (PP (IN of) (NP (NN technology)))
      (SBAR
        (WHNP-1 (WDT which))
        (S
          (NP-SBJ (-NONE- *T*-1))
          (VP
            (MD may)
            (VP (VB end) (NP (PRP$ our) (NN civilization)))))))
    (, ,)
    (CC or)
    (NP (DT a) (JJ normal) (, ,) (JJ healthy) (NN growth))))
(VBZ Does) c-commands (VB have)
(VBZ Does) c-commands have
…
etc.
```

Homework 9: Question 1

Using Python regex,

- i.e. `import re` and `re.search()`, `re.match()`, etc.,

modify program `ccommand2f.py` to filter out c-commander non-matches with `^WH.*-[0-9]+$` from the tregex search command:

- `/^WH.*-([0-9]+)/#1%index > (__ << (/^-NONE-/ <: /^\*T\*-([0-9]+$)/#1%index))`

i.e.

- `python ccommand2fq1.py tregex-whtrace-ex1.mrg`

```
(VBZ Does) c-commands (NP-SBJ (PRP$ our) (NN society))
(VBZ Does) c-commands (PRP$ our)
(VBZ Does) c-commands our
…
(WHNP-1 (WDT which)) c-commands (S
  (NP-SBJ (-NONE- *T*-1))
  (VP (MD may) (VP (VB end) (NP (PRP$ our) (NN civilization)))))
(WHNP-1 (WDT which)) c-commands (NP-SBJ (-NONE- *T*-1))
(WHNP-1 (WDT which)) c-commands (-NONE- *T*-1)
(WHNP-1 (WDT which)) c-commands *T*-1
…
```

you code should not print these!

Note: two cases: 1) Tree(Label, […]), and 2) *T*-1 or word
1), you want to extract the Tree label for regex matching: e.g. *subtree*.label() (see last lecture)
2), you just match directly against the string

Homework 9: Question 1

- How many matching c-commands relations did you get?
- Show your output and code
- **Update**: *let's start this in class today*

Homework 9: Question 2

- Let’s modify your code from the answer to Q1 one step further.
- Now also filter out non-matches with `^\*T\*-[0-9]+$` from the tregex search command:
  - `/^WH.*-([0-9]+)/#1%index > (__ << (/^-NONE-/ <: /^\*T\*-([0-9]+$)/#1%index))`

i.e.

  - `python ccommand2fq2.py tregex-whtrace-ex1.mrg`

```
(WHNP-1 (WDT which)) c-commands (S
  (NP-SBJ (-NONE- *T*-1))
  (VP (MD may) (VP (VB end) (NP (PRP$ our) (NN civilization)))))
(WHNP-1 (WDT which)) c-commands (NP-SBJ (-NONE- *T*-1))
(WHNP-1 (WDT which)) c-commands (-NONE- *T*-1)
(WHNP-1 (WDT which)) c-commands *T*-1
```

you code should not print these!

This matches!

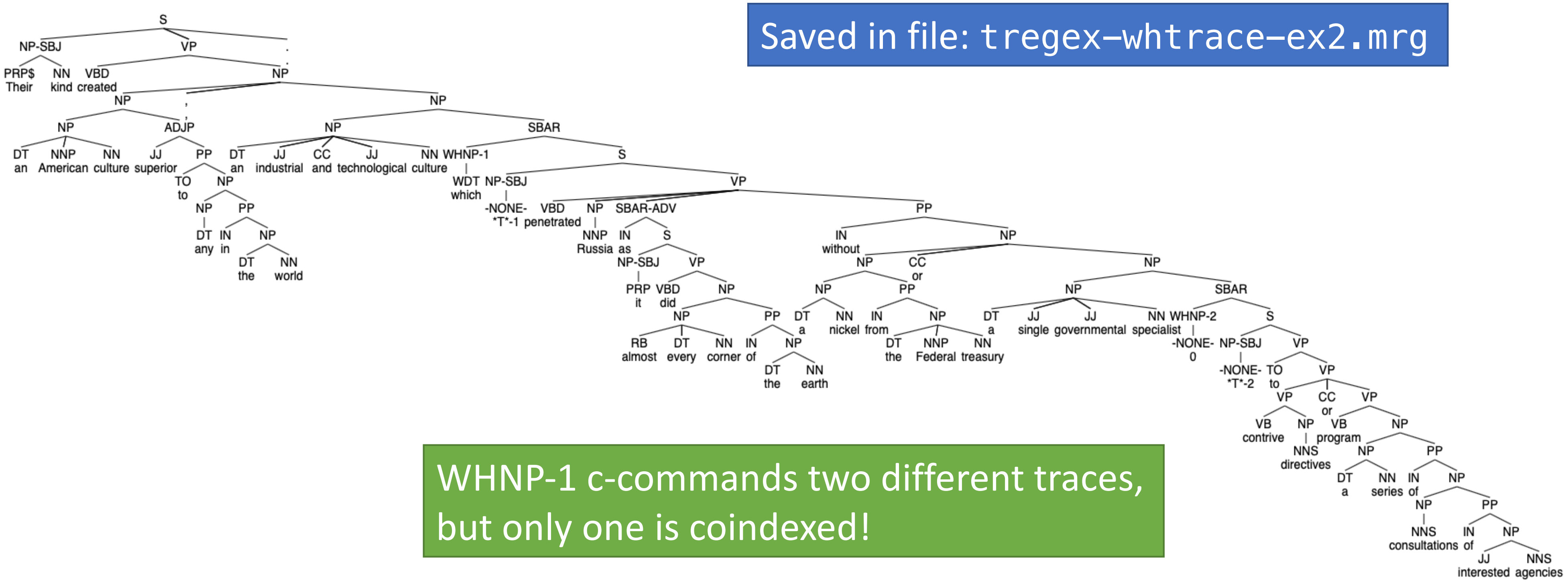
Homework 9: Question 2

- How many matching c-commands relations do you get now?
- Show your output and code

Homework 9: Question 3

- Let’s modify your code from the answer to Q2 another step further.
- Impose the constraint that the indices have to be the same
- Example:
  - `(WHNP-1 (WDT which)) c-commands *T*-1`
- **Hint**: recall tregex search expression
  - `/^WH.*-([0-9]+)/#1%index > (__ << (/^-NONE-/ <: /^\*T\*-([0-9]+$)/#1%index))`
  - `m = re.search(regex, string)`
  - `m.group(1)` gets group 1 string from regex match
- Test your code on example (see next slide):
  - `tregex-whtrace-ex2.mrg`
- Show your output and code

Homework 9: Question 3



Saved in file: `tregex-whtrace-ex2.mrg`

WHNP-1 c-commands two different traces, but only one is coindexed!

Homework 9: Question 4

Let's run the code over the entire ptb dataset!

  - `python –i ccommand2fq3.py` drops you into the Python interpreter
  - `from nltk.corpus import ptb`

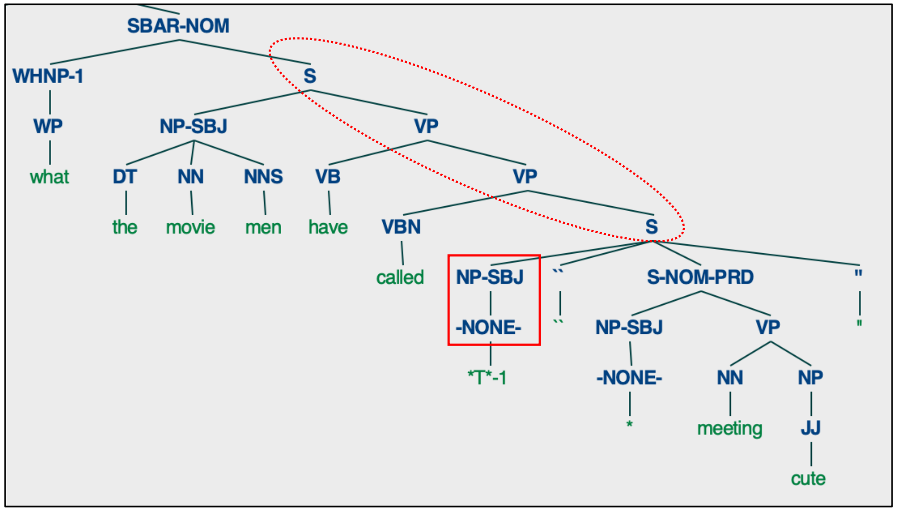
- recall method `.parsed_sents()` gives all the trees
- Modify your code for Q3 to count the total number of matching WH-antecedent/trace pairs in the corpus.
- Give your answer and source code

Homework 9: Question 5

Modify your code to compute the distance between the antecedent and its trace.

- Find the tree with the biggest distance in the ptb corpus.
- Show your code and output (show the tree)
- **Hint:**
  - modify the function `dom` to count depth
  - Note: `dom` is recursive
  - use an extra parameter in the function call to increment the depth each time around

Distance = 6 nodes intervening



Homework 9: Question 6

Extra Credit

- Modify your code to plot the histogram for the WH-antecedent to trace depth over the entire ptb corpus
- What is the average depth?
- **Hint**:
    - `import matplotlib.pyplot as plt`
    - `plt.hist(dist, bins=range(1,max))`
    - `plt.xticks(range(1,max))`
    - `plt.show()`