LING/C SC 581:
Advanced Computational Linguistics

Lecture 12

Today's Topics

Did everyone manage to install the structural probes program?

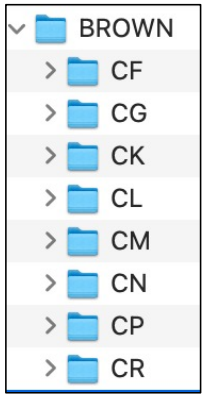
- Using the structural probes program
- Case study: variety of passives
- Homework 6
  - due next Monday midnight

ptb

```
>>> import nltk
>>> from nltk.corpus import ptb
>>> ptb.fileids()
```

```
['BROWN/CF/CF01.MRG', 'BROWN/CF/CF02.MRG', 'BROWN/CF/CF03.MRG', 'BROWN/CF/CF04.MRG',
 'BROWN/CF/CF05.MRG', 'BROWN/CF/CF06.MRG', 'BROWN/CF/CF07.MRG', 'BROWN/CF/CF08.MRG',
 'BROWN/CF/CF09.MRG', 'BROWN/CF/CF10.MRG', 'BROWN/CF/CF11.MRG', 'BROWN/CF/CF12.MRG',
 'BROWN/CE/CE13.MRG', 'BROWN/CE/CE14.MRG', 'BROWN/CE/CE15.MRG', 'BROWN/CE/CE16.MRG',
 'BROWN/CF/CF17.MRG', 'BROWN/CF/CF18.MRG', 'BROWN/CF/CF19.MRG', 'BROWN/CF/CF20.MRG',
 'BROWN/CF/CF21.MRG', 'BROWN/CF/CF22.MRG', 'BROWN/CF/CF23.MRG', 'BROWN/CF/CF24.MRG',
 'BROWN/CE/CE25.MRG', 'BROWN/CE/CE26.MRG', 'BROWN/CE/CE27.MRG', 'BROWN/CE/CE28.MRG',
 'BROWN/CF/CF29.MRG', 'BROWN/CF/CF30.MRG', 'BROWN/CF/CF31.MRG', 'BROWN/CF/CF32.MRG',
 'BROWN/CG/CG01.MRG', 'BROWN/CG/CG02.MRG', 'BROWN/CG/CG03.MRG', 'BROWN/CG/CG04.MRG',
 'BROWN/CG/CG05.MRG', 'BROWN/CG/CG06.MRG', 'BROWN/CG/CG07.MRG', 'BROWN/CG/CG08.MRG',
 'BROWN/CG/CG09.MRG', 'BROWN/CG/CG10.MRG', 'BROWN/CG/CG11.MRG', 'BROWN/CG/CG12.MRG',
 'BROWN/CG/CG13.MRG', 'BROWN/CG/CG14.MRG', 'BROWN/CG/CG15.MRG', 'BROWN/CG/CG16.MRG',
 'BROWN/CG/CG17.MRG', 'BROWN/CG/CG18.MRG', 'BROWN/CG/CG19.MRG', 'BROWN/CG/CG20.MRG',
 'BROWN/CG/CG21.MRG', 'BROWN/CG/CG22.MRG', 'BROWN/CG/CG23.MRG', 'BROWN/CG/CG24.MRG',
 'BROWN/CG/CG25.MRG', 'BROWN/CG/CG26.MRG', 'BROWN/CG/CG27.MRG', 'BROWN/CG/CG28.MRG',
 'BROWN/CG/CG29.MRG', 'BROWN/CG/CG30.MRG', 'BROWN/CG/CG31.MRG', 'BROWN/CG/CG32.MRG',
 'BROWN/CG/CG33.MRG', 'BROWN/CG/CG34.MRG', 'BROWN/CG/CG35.MRG', 'BROWN/CG/CG36.MRG',
 'BROWN/CK/CK01.MRG', 'BROWN/CK/CK02.MRG', 'BROWN/CK/CK03.MRG', 'BROWN/CK/CK04.MRG',
 'BROWN/CK/CK05.MRG', 'BROWN/CK/CK06.MRG', 'BROWN/CK/CK07.MRG', 'BROWN/CK/CK08.MRG',
 'BROWN/CK/CK09.MRG', 'BROWN/CK/CK10.MRG', 'BROWN/CK/CK11.MRG', 'BROWN/CK/CK12.MRG',
 'BROWN/CK/CK13.MRG', 'BROWN/CK/CK14.MRG', 'BROWN/CK/CK15.MRG', 'BROWN/CK/CK16.MRG',
 'BROWN/CK/CK17.MRG', 'BROWN/CK/CK18.MRG', 'BROWN/CK/CK19.MRG', 'BROWN/CK/CK20.MRG',
 'BROWN/CK/CK21.MRG', 'BROWN/CK/CK22.MRG', 'BROWN/CK/CK23.MRG', 'BROWN/CK/CK24.MRG',
 'BROWN/CK/CK25.MRG', 'BROWN/CK/CK27.MRG', 'BROWN/CK/CK28.MRG', 'BROWN/CK/CK29.MRG',
 'BROWN/CL/CL01.MRG', 'BROWN/CL/CL02.MRG', 'BROWN/CL/CL03.MRG', 'BROWN/CL/CL04.MRG',
 'BROWN/CL/CL05.MRG', 'BROWN/CL/CL06.MRG', 'BROWN/CL/CL07.MRG', 'BROWN/CL/CL08.MRG',
 'BROWN/CL/CL09.MRG', 'BROWN/CL/CL10.MRG', 'BROWN/CL/CL11.MRG', 'BROWN/CL/CL12.MRG',
 'BROWN/CL/CL13.MRG', 'BROWN/CL/CL14.MRG', 'BROWN/CL/CL15.MRG', 'BROWN/CL/CL16.MRG',
 'BROWN/CL/CL17.MRG', 'BROWN/CL/CL18.MRG', 'BROWN/CL/CL19.MRG', 'BROWN/CL/CL20.MRG',
 'BROWN/CL/CL21.MRG', 'BROWN/CL/CL22.MRG', 'BROWN/CL/CL23.MRG', 'BROWN/CL/CL24.MRG',
 'BROWN/CM/CM01.MRG', 'BROWN/CM/CM02.MRG', 'BROWN/CM/CM03.MRG', 'BROWN/CM/CM04.MRG',
 'BROWN/CM/CM05.MRG', 'BROWN/CM/CM06.MRG', 'BROWN/CN/CN01.MRG', 'BROWN/CN/CN02.MRG',
 'BROWN/CN/CN03.MRG', 'BROWN/CN/CN04.MRG', 'BROWN/CN/CN05.MRG', 'BROWN/CN/CN06.MRG',
 'BROWN/CN/CN07.MRG', 'BROWN/CN/CN08.MRG', 'BROWN/CN/CN09.MRG', 'BROWN/CN/CN10.MRG',
 'BROWN/CN/CN11.MRG', 'BROWN/CN/CN12.MRG', 'BROWN/CN/CN13.MRG', 'BROWN/CN/CN14.MRG',
 'BROWN/CN/CN15.MRG', 'BROWN/CN/CN16.MRG', 'BROWN/CN/CN17.MRG', 'BROWN/CN/CN18.MRG',
 'BROWN/CN/CN19.MRG', 'BROWN/CN/CN20.MRG', 'BROWN/CN/CN21.MRG', 'BROWN/CN/CN22.MRG',
 'BROWN/CN/CN23.MRG', 'BROWN/CN/CN24.MRG', 'BROWN/CN/CN25.MRG', 'BROWN/CN/CN26.MRG',
 'BROWN/CN/CN27.MRG', 'BROWN/CN/CN28.MRG', 'BROWN/CN/CN29.MRG', 'BROWN/CP/CP01.MRG',
 'BROWN/CP/CP02.MRG', 'BROWN/CP/CP03.MRG', 'BROWN/CP/CP04.MRG', 'BROWN/CP/CP05.MRG',
 'BROWN/CP/CP06.MRG', 'BROWN/CP/CP07.MRG', 'BROWN/CP/CP08.MRG', 'BROWN/CP/CP09.MRG',
 'BROWN/CP/CP10.MRG', 'BROWN/CP/CP11.MRG', 'BROWN/CP/CP12.MRG', 'BROWN/CP/CP13.MRG',
 'BROWN/CP/CP14.MRG', 'BROWN/CP/CP15.MRG', 'BROWN/CP/CP16.MRG', 'BROWN/CP/CP17.MRG',
 'BROWN/CP/CP18.MRG', 'BROWN/CP/CP19.MRG', 'BROWN/CP/CP20.MRG', 'BROWN/CP/CP21.MRG',
 'BROWN/CP/CP22.MRG', 'BROWN/CP/CP23.MRG', 'BROWN/CP/CP25.MRG', 'BROWN/CP/CP26.MRG',
 'BROWN/CP/CP27.MRG', 'BROWN/CP/CP28.MRG', 'BROWN/CP/CP29.MRG', 'BROWN/CR/CR01.MRG',
 'BROWN/CR/CR02.MRG', 'BROWN/CR/CR03.MRG', 'BROWN/CR/CR04.MRG', 'BROWN/CR/CR05.MRG',
 'BROWN/CR/CR06.MRG', 'BROWN/CR/CR07.MRG', 'BROWN/CR/CR08.MRG', 'BROWN/CR/CR09.MRG',
…
```

192



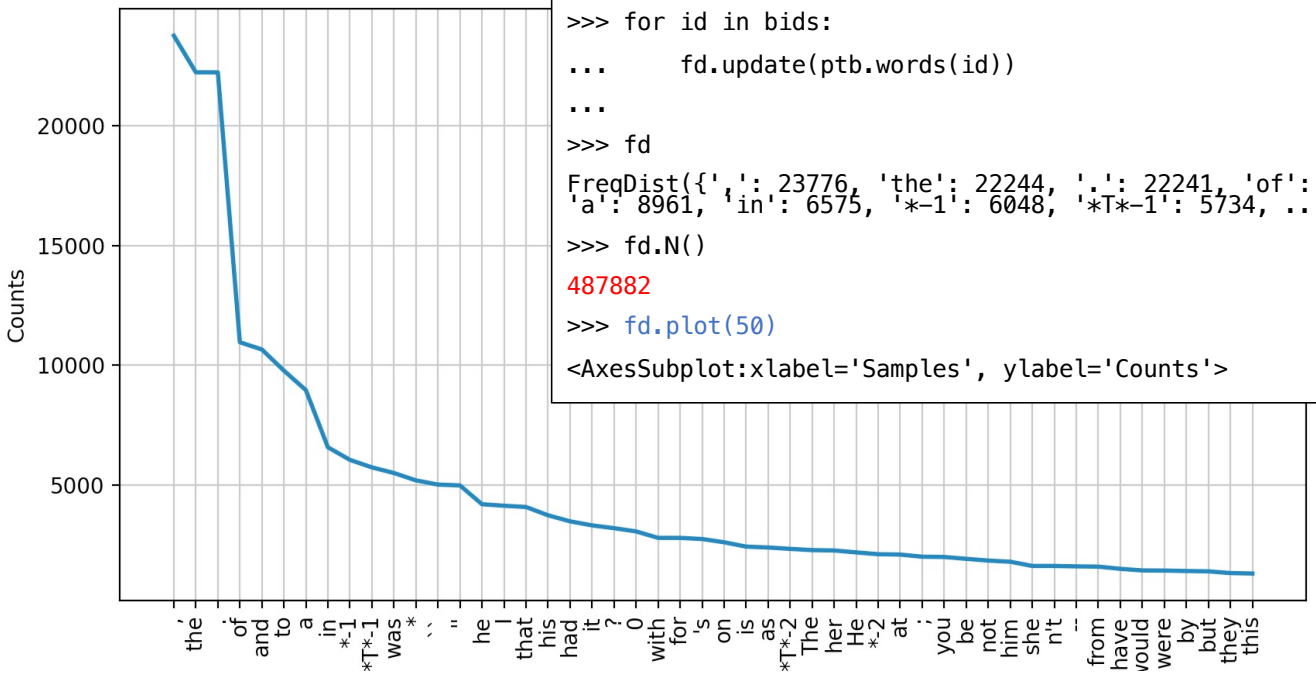

```
['WSJ/00/WSJ_0004.MRG', 'WSJ/00/WSJ_0005.MRG', 'WSJ/00/WSJ_0006.MRG', 'WSJ/00/WSJ_0007.MRG',
 'WSJ/00/WSJ_0008.MRG', 'WSJ/00/WSJ_0009.MRG', 'WSJ/00/WSJ_0010.MRG', 'WSJ/00/WSJ_0011.MRG',
 'WSJ/00/WSJ_0012.MRG', 'WSJ/00/WSJ_0013.MRG', 'WSJ/00/WSJ_0014.MRG', 'WSJ/00/WSJ_0015.MRG',
 'WSJ/00/WSJ_0016.MRG', 'WSJ/00/WSJ_0017.MRG', 'WSJ/00/WSJ_0018.MRG', 'WSJ/00/WSJ_0019.MRG',
 'WSJ/00/WSJ_0020.MRG', 'WSJ/00/WSJ_0021.MRG', 'WSJ/00/WSJ_0022.MRG', 'WSJ/00/WSJ_0023.MRG',
 'WSJ/00/WSJ_0024.MRG', 'WSJ/00/WSJ_0025.MRG', 'WSJ/00/WSJ_0026.MRG', 'WSJ/00/WSJ_0027.MRG',
 'WSJ/00/WSJ_0028.MRG', 'WSJ/00/WSJ_0029.MRG', 'WSJ/00/WSJ_0030.MRG', 'WSJ/00/WSJ_0031.MRG',
 'WSJ/00/WSJ_0032.MRG', 'WSJ/00/WSJ_0033.MRG', 'WSJ/00/WSJ_0034.MRG', 'WSJ/00/WSJ_0035.MRG',
 'WSJ/00/WSJ_0036.MRG', 'WSJ/00/WSJ_0037.MRG', 'WSJ/00/WSJ_0038.MRG',
…
 'WSJ/23/WSJ_2398.MRG', 'WSJ/23/WSJ_2399.MRG', 'WSJ/24/WSJ_2400.MRG', 'WSJ/24/WSJ_2401.MRG',
 'WSJ/24/WSJ_2402.MRG', 'WSJ/24/WSJ_2403.MRG', 'WSJ/24/WSJ_2404.MRG', 'WSJ/24/WSJ_2405.MRG',
 'WSJ/24/WSJ_2406.MRG', 'WSJ/24/WSJ_2407.MRG', 'WSJ/24/WSJ_2408.MRG', 'WSJ/24/WSJ_2409.MRG',
 'WSJ/24/WSJ_2410.MRG', 'WSJ/24/WSJ_2411.MRG', 'WSJ/24/WSJ_2412.MRG', 'WSJ/24/WSJ_2413.MRG',
 'WSJ/24/WSJ_2414.MRG', 'WSJ/24/WSJ_2415.MRG', 'WSJ/24/WSJ_2416.MRG', 'WSJ/24/WSJ_2417.MRG',
 'WSJ/24/WSJ_2418.MRG', 'WSJ/24/WSJ_2419.MRG', 'WSJ/24/WSJ_2420.MRG', 'WSJ/24/WSJ_2421.MRG',
 'WSJ/24/WSJ_2422.MRG', 'WSJ/24/WSJ_2423.MRG', 'WSJ/24/WSJ_2424.MRG', 'WSJ/24/WSJ_2425.MRG',
 'WSJ/24/WSJ_2426.MRG', 'WSJ/24/WSJ_2427.MRG', 'WSJ/24/WSJ_2428.MRG', 'WSJ/24/WSJ_2429.MRG',
 'WSJ/24/WSJ_2430.MRG', 'WSJ/24/WSJ_2431.MRG', 'WSJ/24/WSJ_2432.MRG', 'WSJ/24/WSJ_2433.MRG',
 'WSJ/24/WSJ_2434.MRG', 'WSJ/24/WSJ_2435.MRG', 'WSJ/24/WSJ_2436.MRG', 'WSJ/24/WSJ_2437.MRG',
 'WSJ/24/WSJ_2438.MRG', 'WSJ/24/WSJ_2439.MRG', 'WSJ/24/WSJ_2440.MRG', 'WSJ/24/WSJ_2441.MRG',
 'WSJ/24/WSJ_2442.MRG', 'WSJ/24/WSJ_2443.MRG', 'WSJ/24/WSJ_2444.MRG', 'WSJ/24/WSJ_2445.MRG',
 'WSJ/24/WSJ_2446.MRG', 'WSJ/24/WSJ_2447.MRG', 'WSJ/24/WSJ_2448.MRG', 'WSJ/24/WSJ_2449.MRG',
 'WSJ/24/WSJ_2450.MRG', 'WSJ/24/WSJ_2451.MRG', 'WSJ/24/WSJ_2452.MRG', 'WSJ/24/WSJ_2453.MRG',
 'WSJ/24/WSJ_2454.MRG']
```

2312

ptb

Brown corpus file ids

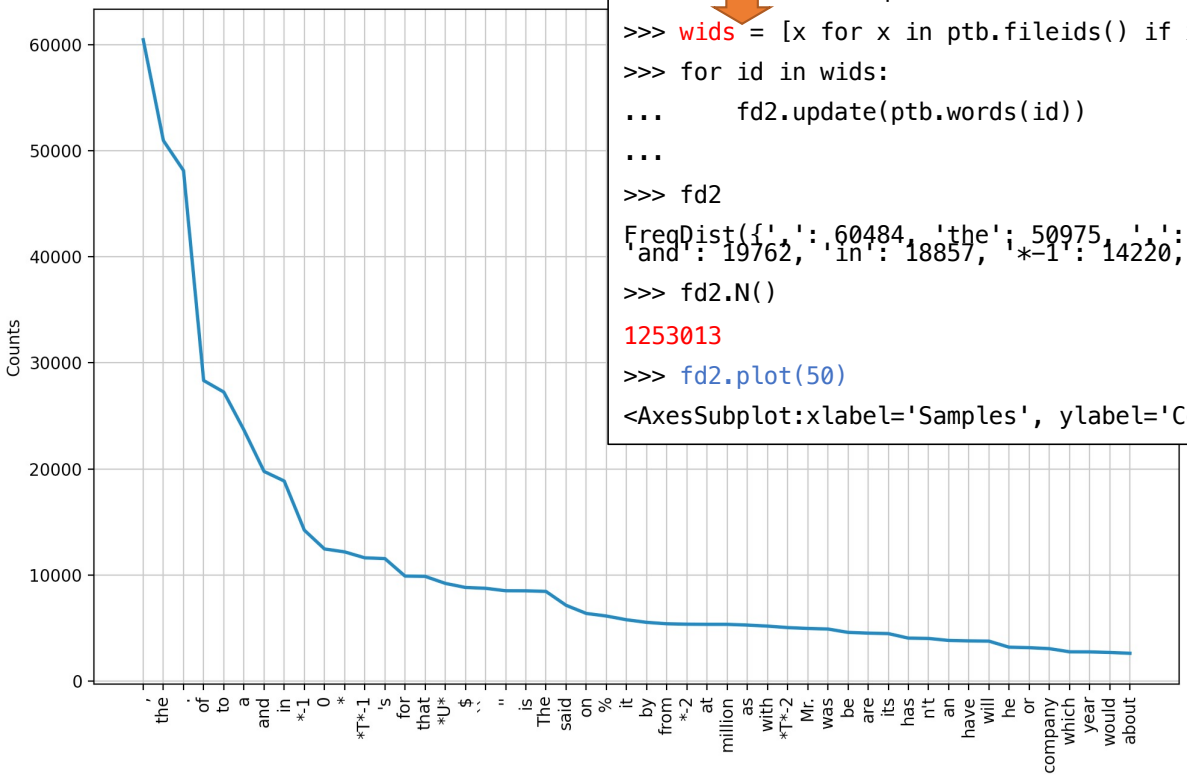
```
>>> bids = [x for x in ptb.fileids() if x.startswith('BROWN')]
>>> len(bids)
192
>>> fd = nltk.FreqDist()
>>> for id in bids:
...     fd.update(ptb.words(id))
...
>>> fd
FreqDist({',': 23776, 'the': 22244, '.': 22241, 'of': 10964, 'and': 10661, 'to': 9778,
'a': 8961, 'in': 6575, '*-1': 6048, '*T*-1': 5734, ...})
>>> fd.N()
487882
>>> fd.plot(50)
<AxesSubplot:xlabel='Samples', ylabel='Counts'>
```



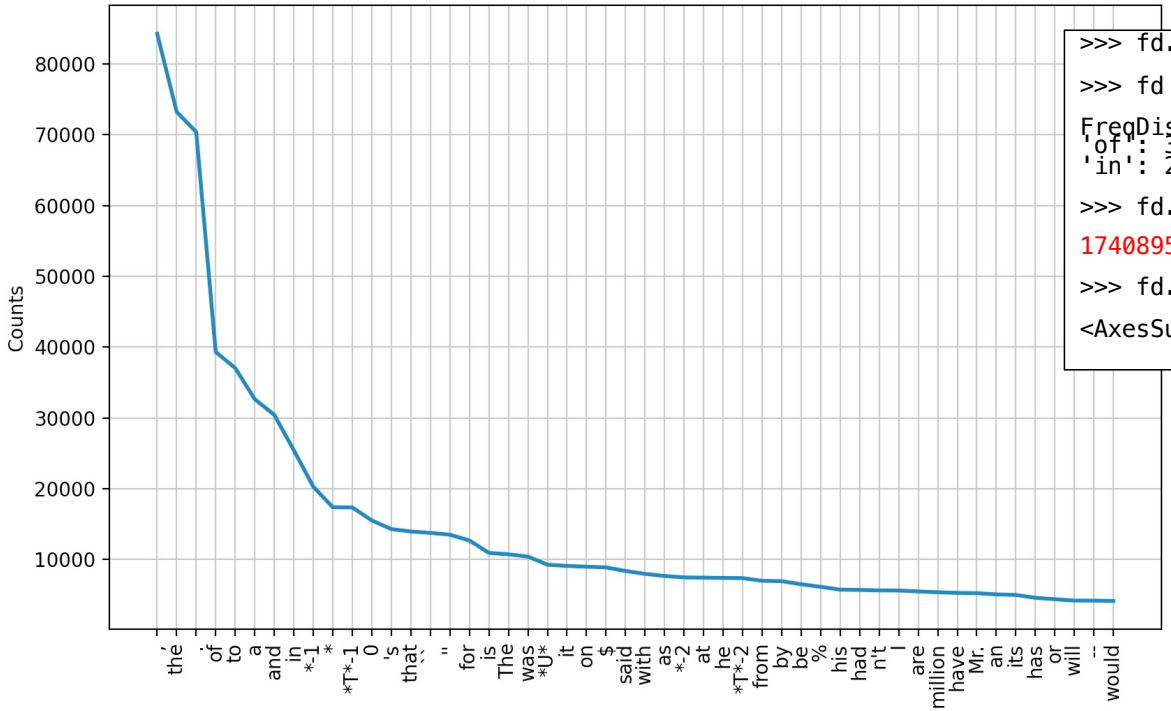
ptb

WSJ corpus file ids

```
>>> fd2 = nltk.FreqDist()
>>> wids = [x for x in ptb.fileids() if x.startswith('WSJ')]
>>> for id in wids:
...     fd2.update(ptb.words(id))
...
>>> fd2
FreqDist({',': 60484, 'the': 50975, '.': 48144, 'of': 28338, 'to': 27249, 'a': 23673,
'and': 19762, 'in': 18857, '*-1': 14220, '0': 12447, ...})
>>> fd2.N()
1253013
>>> fd2.plot(50)
<AxesSubplot:xlabel='Samples', ylabel='Counts'>
```



ptb



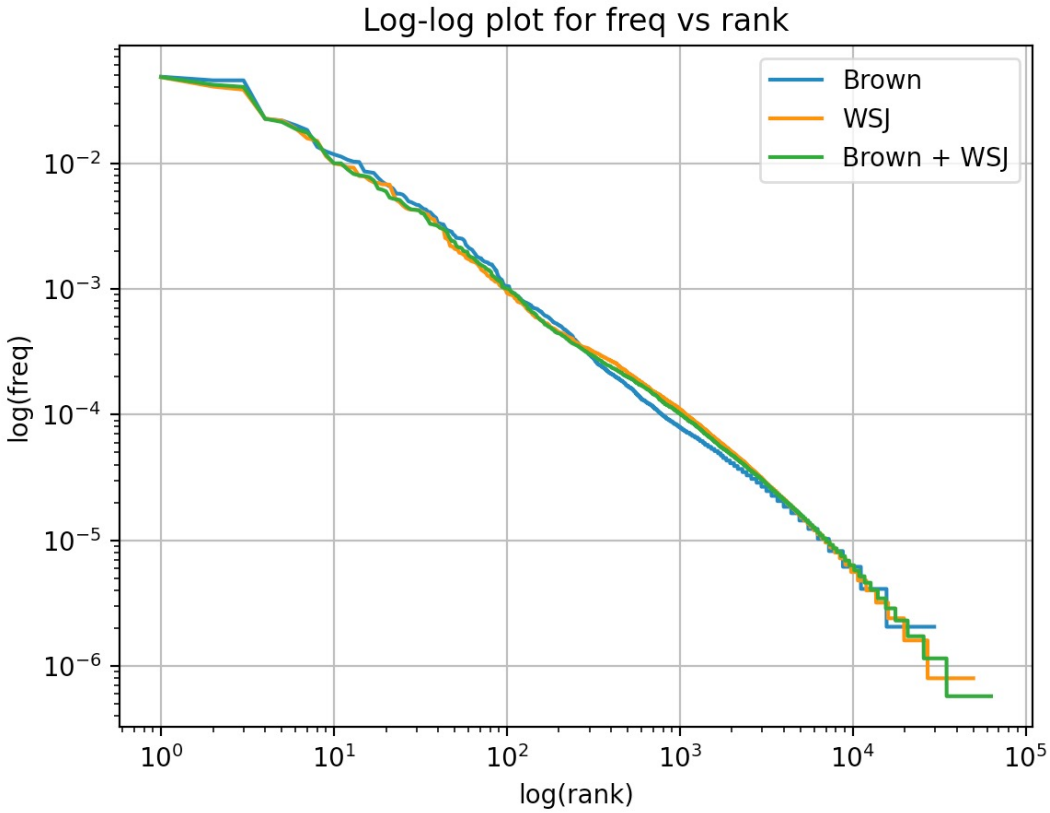

```
>>> fd.update(fd2)
>>> fd
FreqDist({',': 84260, 'the': 73219, '.': 70385,
'of': 39302, 'to': 37027, 'a': 32634, 'and': 30423,
'in': 25432, '*-1': 20268, '*': 17363, ...})
>>> fd.N()
1740895
>>> fd.plot(50)
<AxesSubplot:xlabel='Samples', ylabel='Counts'>
```

ptb

```
>>> wws = []
>>> for id in wids:
...     wws.extend(ptb.words(id))
...
>>> bws = []
>>> for id in bids:
...     bws.extend(ptb.words(id))
...
>>> len(bws)
487882
>>> len(wws)
1253013
>>> from zipf import *
>>> fig()
>>> plot(bws, "Brown")
>>> plot(wws, "WSJ")
>>> plot(bws+wws, "Brown + WSJ")
>>> plt.legend()
<matplotlib.legend.Legend object at 0x1278c4520>
>>> plt.show()
```

WSJ words

Brown words



ptb

On course website: `zipf.py`

```
1# Sandiway Fong (c) University of Arizona 2019¶
2# simple function to plot Zipf's Law¶
3# assumes matplotlib¶
4from collections import Counter¶
5from math import log, log10¶
6import matplotlib.pyplot as plt¶
7¶
8def plot(tokens, text):¶
9    size = len(tokens)¶
10    c = Counter()¶
11    for token in tokens:¶
12        c[token] += 1¶
13    mc = c.most_common()¶
14    ranks = [x for x in range(1,len(mc)+1)]¶
15    freq = [item[1]/size for item in mc]¶
16    plt.plot(ranks,freq, label=text)¶
```

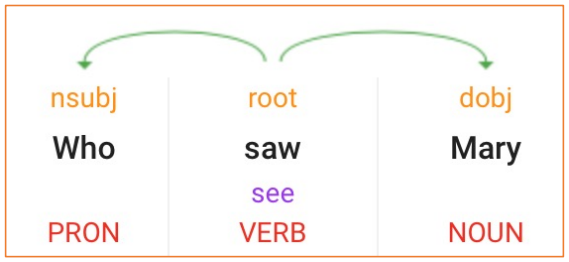
```
17¶
18def fig():¶
19    plt.figure(1)¶
20    plt.xscale('log')¶
21    plt.xlabel('log(rank)')¶
22    plt.yscale('log')¶
23    plt.ylabel('log(freq)')¶
24    plt.grid(True)¶
25    plt.title('Log-log plot for freq vs rank')¶
```

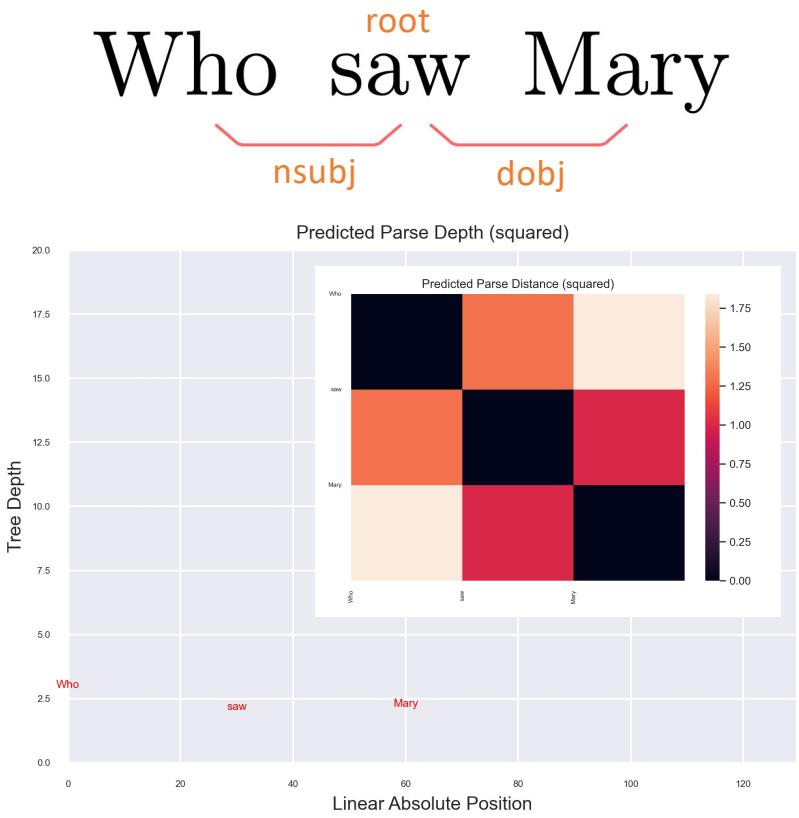
structural probes

Run a pretrained structural probe on `BERT-large` quickly on the command line.

It's easy to get predictions on a sentence (or file of sentences) using our demo script and the pre-trained structural probes we release. We use `pytorch-pretrained-bert` to get BERT subword embeddings for each sentence; it should be installed during setup of the repository.

Make sure you've run `download_example.sh`; this will download two probe parameter files to `example/data/`. Also make sure you've installed all dependencies. One is a distance probe on the 16th hidden layer of BERT large, and the other is a depth probe on the same layer. The configuration file `example/demo-bert.yaml` has the right paths already plugged in; just pipe text into the demo file, as follows:





Testing a range of sentences

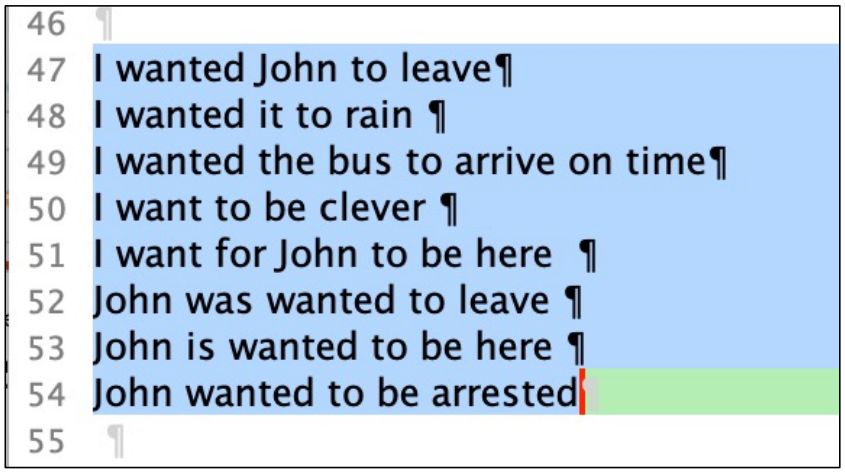
- Examples:
  - assuming you are in the (1st level) `structural-probes` directory on the Terminal:
    - there are further subdirectories structural-probes, example, scripts, doc-assets
    - `run_demo.py` is inside `structural-probes/structural-probes`
    - `demo-bert.yaml` is inside `structural-probes/example`

1. `echo "the bombing of the cities is a crime\nthe bombing of the cities are a crime" | python3 structural-probes/run_demo.py example/demo-bert.yaml`
2. `sed -n '207,221p' l\&u1.txt | python3 structural-probes/run_demo.py example/demo-bert.yaml`

Linux and Mac: **sed** selection

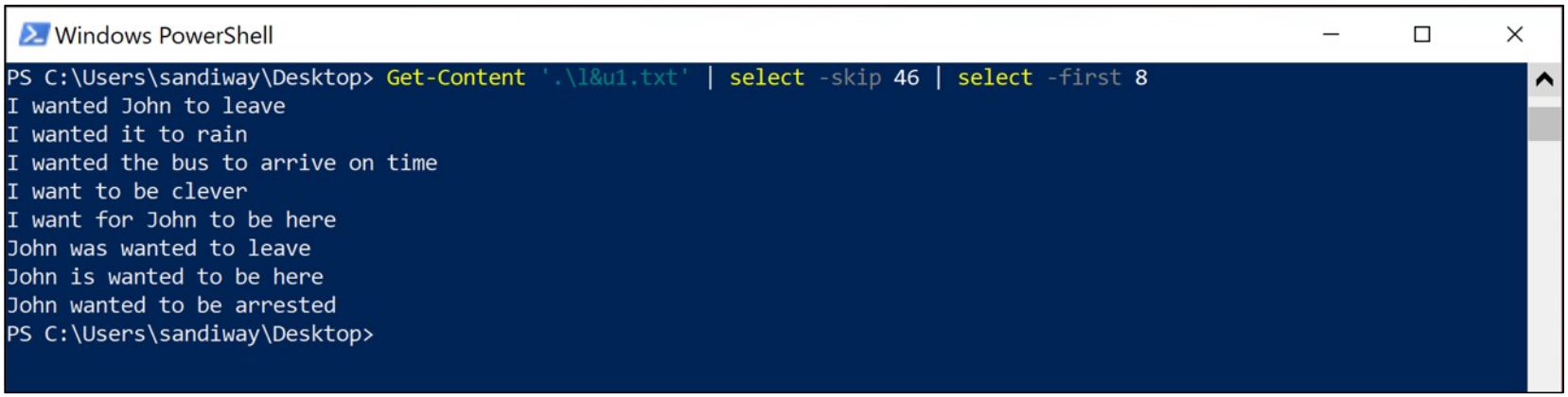
- Example:

```
sed –n '47,54p' l\&u1.txt
I wanted John to leave
I wanted it to rain
I wanted the bus to arrive on time
I want to be clever
I want for John to be here
John was wanted to leave
John is wanted to be here
John wanted to be arrested
```



Windows 10: `sed` selection functionality

- Powershell cmdlets:
  - Get-Content *filename* | Select -Skip *n*
  - Select -First *n*
  - Select -Last *n*



Passives

- **Examples**:
  1. John was arrested
  2. Who was arrested
  3. John was believed to have been arrested (*exceptional passivization*)
  4. It is believed that John was arrested
  5. That John was arrested is believed by everyone
  6. John had insults shouted at him (*indirect passive*, Emonds 2007)

Homework 6

Question 1

- **Raising verb *seems***:
  - John is happy
  - John *seems* happy
  - It *seems* John is happy
  - It *seems* that John is happy
  - John *seems* to be happy

- Run these examples on Google Natural Language (Google NL) and the structural probes (SP) program.
- Assume the SP unlabeled relations are labeled with the most likely dependency labels
- What is different between the SP and Google NL analyzes?

Homework 6

Question 2

- suppose the lower clause is:
  - John saw Mary
- and the raising predicate is *seemed*:
  - John seemed to see Mary
  - It seems (that) John saw Mary

- Does the SP program do better on these examples?

Homework 6

Question 3

- Consider object *wh*-question formation:
  - Who did John see
  - Who did John think (that) Mary saw
  - Who did John (that) say Bill (that) thought Mary saw

- Which does better, the SP program or Google NL?

Homework 6

Question 4

- Consider subject *wh*-question formation:
  - Who saw Mary
  - Who did John think saw Mary
  - Who did John say (that) Bill thought saw Mary
- Which does better, the SP program or Google NL?