LING/C SC 581:

Advanced Computational Linguistics

Lecture 26

Today's Topics

- Optional Homework 11
  - make-up if you missed or bombed a homework

Or choose,

- Optional Homework 12
  - if you prefer

Last Time: nltk ptb

- `ptb.parsed_sents()` is a list of trees.
- Each tree is a Tree object.
- Tree printing:
  - `t.pretty_print()` ascii graphics
  - `t.pprint()` parentheses
  - `t.draw()` window
- Tree objects are recursively defined. Given a tree t:
  - `t.label()` category label of `t`
    - e.g. `t.label().startswith('VP')`
  - `len(t)` number of children of `t`
  - `t[i]` `i`th child of `t`
    - which could be a Tree or simply a string
  - `t.subtrees()` all possible subtrees in `t`
  - `t.pos()` list of `(word, postag)` in `t`

Homework 11 and 12 starting point

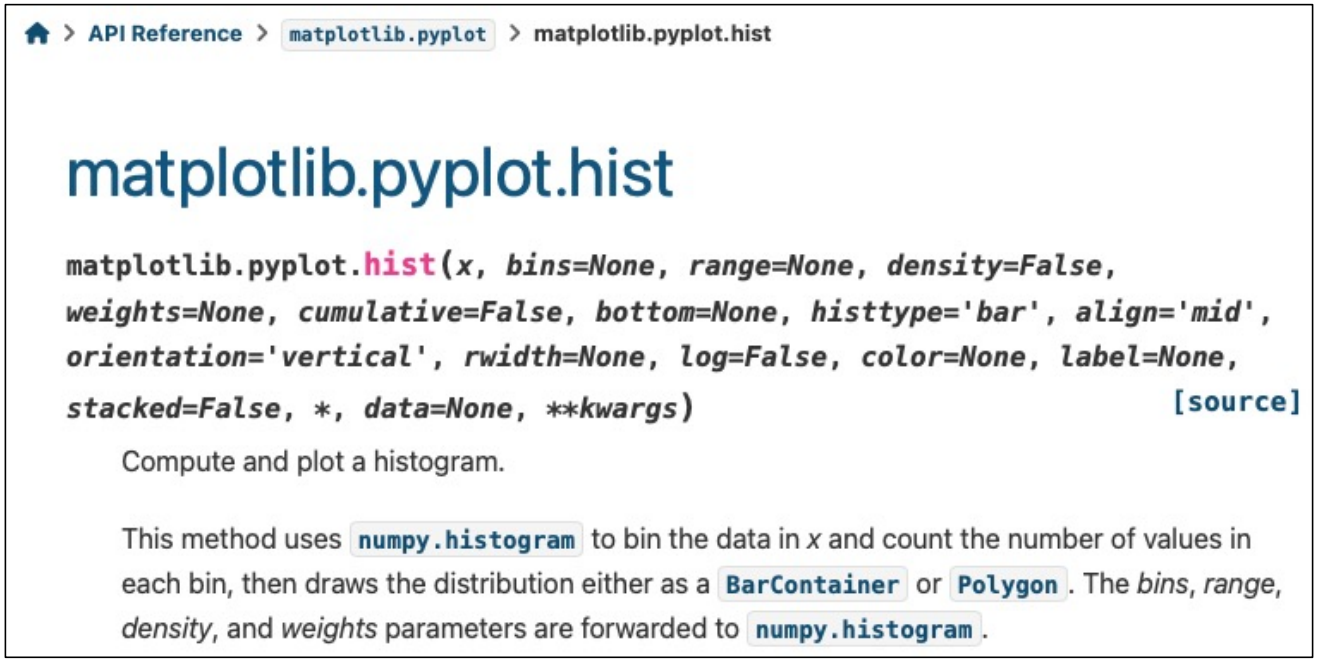
```
$ python
>>> from nltk.corpus import ptb
>>> trees = ptb.parsed_sents()
>>> len(trees)
73451
>>> trees[0].subtrees()
<generator object Tree.subtrees at 0x1434784a0>
>>> len(list(trees[0].subtrees()))
87
```

Homework 11

- Let's define the smallest and largest tree as the ones with the least and most subtrees, respectively.
- Question 1:
  - Find smallest and largest trees in ptb.
  - What are their sizes?
  - Display the trees
- Question 2:
  - plot a histogram of tree sizes in ptb
  - use `matplotlib.pyplot.hist` (*see next slide*)
  - for `hist` set `bins=range(0,450+1,25)` (*each bin is 25 wide*)
  - use `xticks(range(0,450+1,25))` to set the x-axis ticks (*every 25*)
  - Which bin has the most trees? About how many?

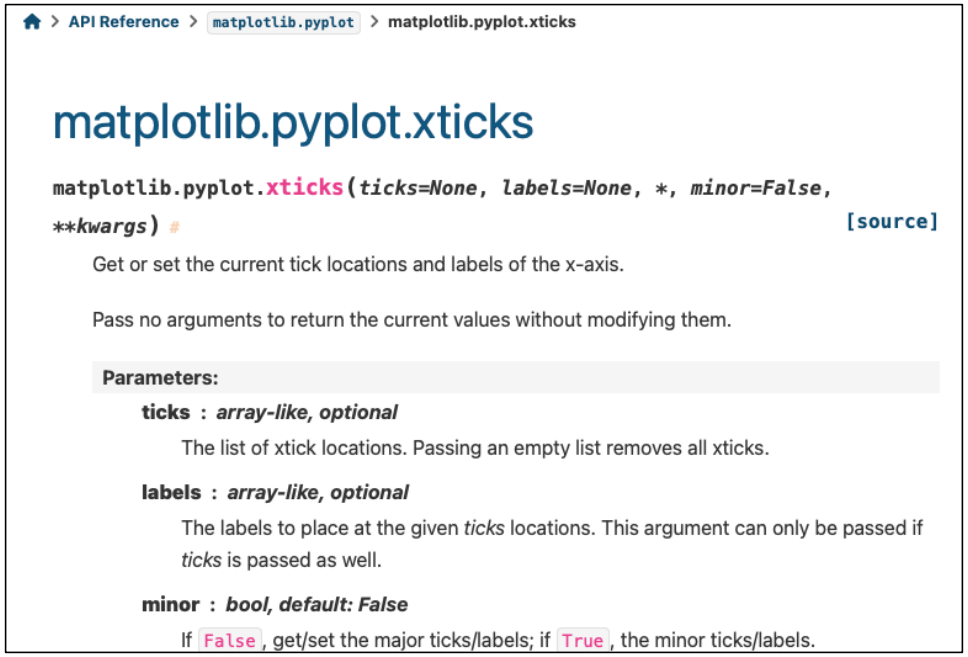
Homework 11

- https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html



Homework 11

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xticks.html



API Reference > matplotlib.pyplot > matplotlib.pyplot.xticks

matplotlib.pyplot.xticks

matplotlib.pyplot.xticks(*ticks=None*, *labels=None*, *, *minor=False*, ***kwargs*) [source]

Get or set the current tick locations and labels of the x-axis.

Pass no arguments to return the current values without modifying them.

Parameters:

ticks : *array-like, optional*

The list of xtick locations. Passing an empty list removes all xticks.

labels : *array-like, optional*

The labels to place at the given *ticks* locations. This argument can only be passed if *ticks* is passed as well.

minor : *bool, default: False*

If `False`, get/set the major ticks/labels; if `True`, the minor ticks/labels.

Homework 12

- Last time:
  - verb have **frames**
  - e.g. [$_{VP}$ [$_{VB.*}$ break][$_{NP}$ the vase]]
  - e.g. [$_{VP}$ [$_{VB.*}$ broke] [$_{PRT}$ [$_{RB}$ out]][$_{NP}$ the go codes]]
  - e.g. [$_{VP}$ [$_{VB.*}$ break][$_{NP}$ it][$_{PRT}$ [$_{RB}$ off]]]
  - PRT = syntax label (prsguid1.pdf, page 36), RB = adverb (POS label)

PRT — Particle. Category for words that should be tagged RP, as described in the *POS guidelines* [Santorini 1990], with some guidance from [Quirk et al. 1985] sections 16.3-16 in tricky ADVP vs. PRT decisions (but note that the Treebank notion of particle is somewhat different from that of Quirk et al.).

Homework 12

- *Break* is not the only verb with many **verb frames**.
- *Go* is another:
  - *go on endlessly trying to …*
  - *go far toward …*
  - *went on*
  - *going to be confidential …*
  - *went over the report …*
  - *gone overboard in stressing their significance*
  - *going to a chemistry lab*

Homework 12

- Question 1:
  - write a Python function to find all VP frames of the form:
    - [$_{VP}$ [$_{VB.*}$ go|goes|went|going|gone] …]
    - use ptb from nltk.corpus
    - how many are there?
- Question 2:
  - modify your function to return the list of labels of the sisters of *go* in the VP
  - e.g. ['NP','PP', 'SBAR-TMP'], ['S'] or [] (*nothing*),
    - *see examples in later slide*
  - How may verb frames does *go* have?
  - How many different (kinds of) frames? (Use nltk.FreqDist())
  - Which is the most common frame?
  - plot a bar chart for the top 20 frames and their counts (use .plot() from FreqDist)

Homework 12

- https://www.nltk.org/api/nltk.probability.FreqDist.html

nltk.probability.FreqDist

class nltk.probability.FreqDist [source]

Bases: `Counter`

A frequency distribution for the outcomes of an experiment. A frequency distribution records the number of times each outcome of an experiment has occurred. For example, a frequency distribution could be used to record the frequency of each word type in a document. Formally, a frequency distribution can be defined as a function mapping from each sample to the number of times that sample occurred as an outcome.

Frequency distributions are generally constructed by running a number of experiments, and incrementing the count for a sample every time it is an outcome of an experiment. For example, the following code will produce a frequency distribution that encodes how often each word occurs in a text:

```
>>> from nltk.tokenize import word_tokenize
>>> from nltk.probability import FreqDist
>>> sent = 'This is an example sentence'
>>> fdist = FreqDist()
>>> for word in word_tokenize(sent):
...     fdist[word.lower()] += 1
```

An equivalent way to do this is with the initializer:

```
>>> fdist = FreqDist(word.lower() for word in word_tokenize(sent))
```

Homework 12

- https://www.nltk.org/api/nltk.probability.FreqDist.html

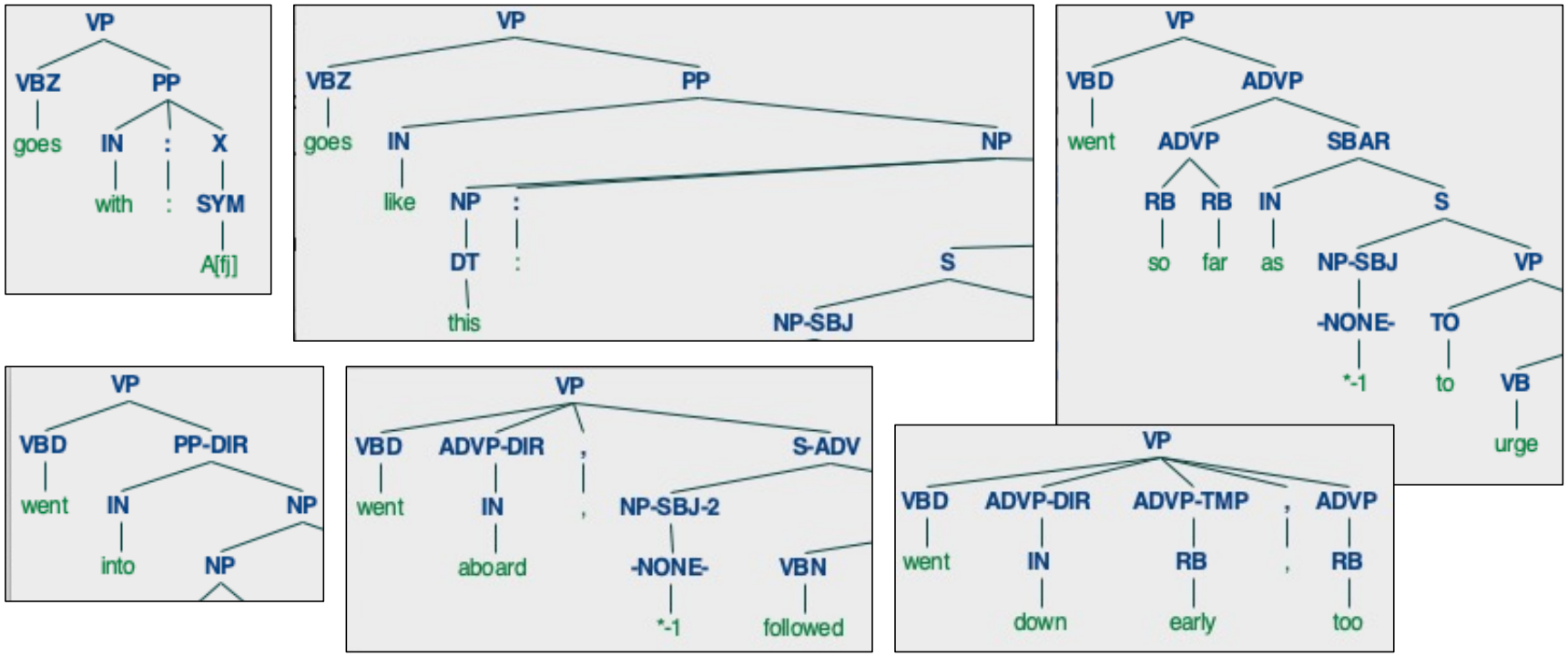
plot(*args, title='', cumulative=False, percents=False, show=True, **kwargs) [source]

Plot samples from the frequency distribution displaying the most frequent sample first. If an integer parameter is supplied, stop after this many samples have been plotted. For a cumulative plot, specify cumulative=True. Additional `**kwargs` are passed to matplotlib's plot function. (Requires Matplotlib to be installed.)

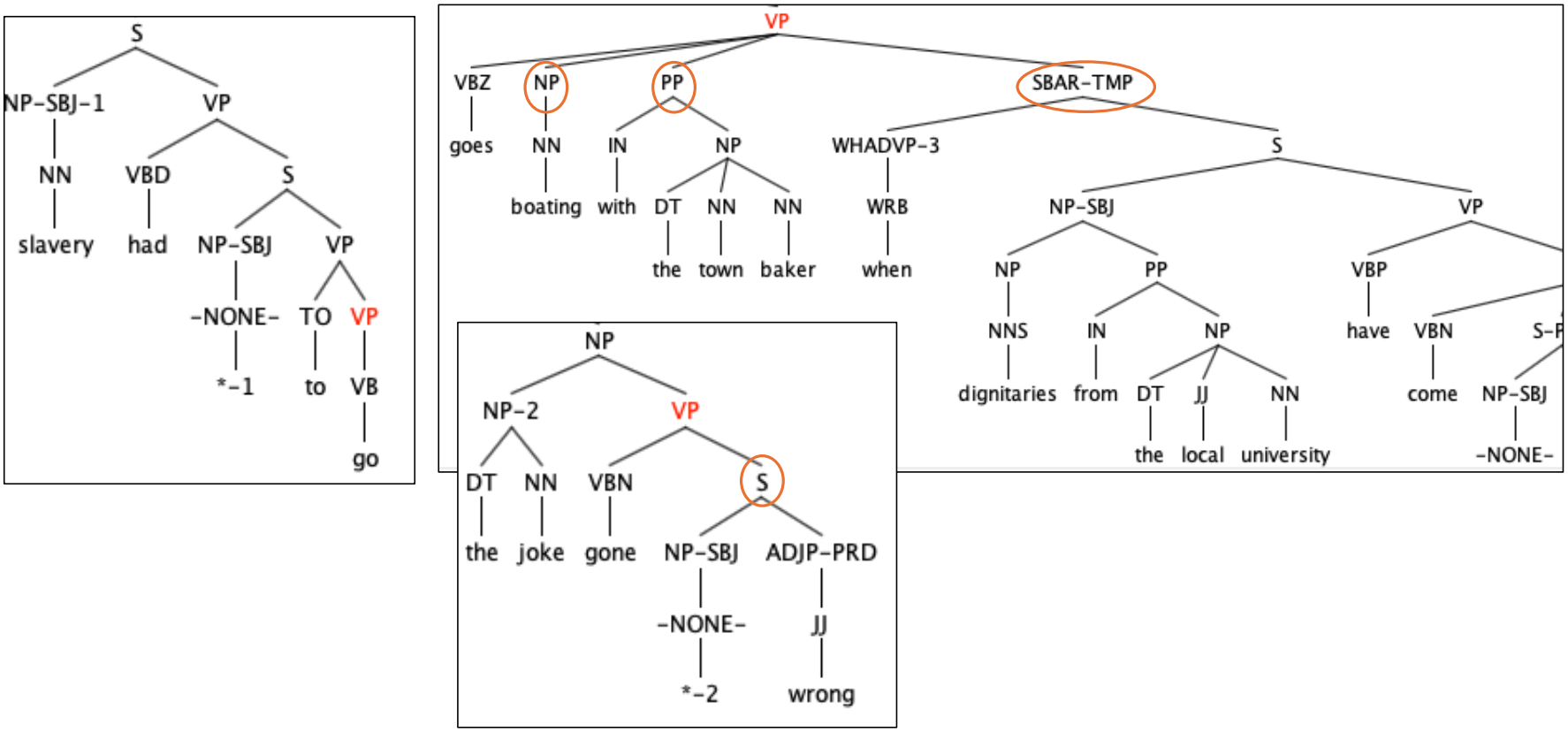
Parameters

- **title** (*str*) – The title for the graph.
- **cumulative** (*bool*) – Whether the plot is cumulative. (default = False)
- **percents** (*bool*) – Whether the plot uses percents instead of counts. (default = False)
- **show** (*bool*) – Whether to show the plot, or only return the ax.

Homework 12


VP
VBZ
PP
goes
IN
:
X
with
:
SYM
A[fj]
VP
VBZ
PP
goes
IN
NP
like
NP
:
DT
:
S
this
NP-SBJ
VP
VBD
ADVP
went
ADVP
SBAR
RB
RB
IN
S
so
far
as
NP-SBJ
VP
-NONE-
TO
*-1
to
VB
urge
VP
VBD
PP-DIR
went
IN
NP
into
NP
VP
VBD
ADVP-DIR
,
S-ADV
went
IN
,
NP-SBJ-2
aboard
-NONE-
VBN
*-1
followed
VP
VBD
ADVP-DIR
ADVP-TMP
,
ADVP
went
IN
RB
,
RB
down
early
too

Homework 12


S
NP-SBJ-1
VP
NN
VBD
S
slavery
had
NP-SBJ
VP
-NONE-
TO
VP
*-1
to
VB
go
VP
VBZ
NP
PP
SBAR-TMP
goes
NN
IN
NP
WHADVP-3
S
boating
with
DT
NN
NN
WRB
NP-SBJ
VP
the
town
baker
when
NP
PP
VBP
NNS
IN
NP
have
VBN
S-P
dignitaries
from
DT
JJ
NN
come
NP-SBJ
the
local
university
-NONE-
NP
NP-2
VP
DT
NN
VBN
S
the
joke
gone
NP-SBJ
ADJP-PRD
-NONE-
JJ
*-2
wrong