Lecture 6

Ashwini Vaidya

22/4

Tagging

```
>>> from nltk.corpus import gutenberg
>>> gutenberg.fileids()
sample_corpus= gutenberg.raw('carroll-alice.txt')


>>> alice_tokens = nltk.word_tokenize(sample_corpus.lower())

Alternatively,
>>> alice_tokens2= nltk.corpus.gutenberg.words('carroll-alice.txt')

>>> alice_tagged = nltk.pos_tag(alice_tokens)

For help with tags, look at
>>>nltk.help.upenn_tagset('RB')
```

The tagged corpus gives us a list of **tuples**. These are another kind of data structure meant for grouping values that are similar in some way. Tuples are **immutable** like strings. Slicing syntax applies to tuples as well e.g.

```
 >>> alice_tagged[1]
>>>alice_tagged[1][1]
>>> alice_tagged[1][1] = 'KL'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

```
>>> for word, pos in alice_tagged:
>>>     alice_r.append((pos,word))
>>> cfd = nltk.ConditionalFreqDist(alice_r)
<ConditionalFreqDist with 41 conditions>

>>> cfd.conditions()
['PRP$', 'VBG', 'VBD', '``', 'VBN', ',', "''", 'VBP', 'WDT', 'JJ', 'WP', 'VBZ', 'DT',

>>> for tag in sorted(cfd.conditions()):
...         print tag, sorted(cfd[tag],key=lambda x:cfd[tag][x])
...
>>> for tag in sorted(cfd.conditions()):
...         print tag, sorted(cfd[tag],key=lambda x:cfd[tag][x],reverse='True')[:10]
...
```