Chapter 5

1. What are the most common adverbs in the brown corpus (categories="news")? Please sort all the adverbs by frequency, with the most frequent ones first. (Please use the universal tagset)

In [1]:
```
import nltk
from nltk.corpus import brown
from nltk import FreqDist

# get the tagged words for news category
brown_news_tagged = brown.tagged_words(categories='news', tagset='universal')
adverbs = [w[0] for w in brown_news_tagged if w[1] == 'ADV']

# printing only the top 25
print(FreqDist(adverbs).most_common(25))
```

```
[('not', 254), ('when', 128), ('also', 120), ('now', 76), ('as', 75), ('here', 67), ('where', 58), ('then',
56), ('back', 55), ('about', 49), ('more', 49), ('only', 48), ('even', 48), ('so', 47), ('most', 45), ('wel
l', 41), ('When', 41), ('just', 41), ('never', 38), ('p.m.', 38), ('however', 37), ('too', 37), ('how', 3
7), ('still', 36), ('ago', 36)]
```

2. What are the part-of-speech tags before the word "news" in the brown corpus (categories="news")? (Please use the universal tagset)

In [2]:
```
from nltk import bigrams

# get the tagged words for news category
brown_news_tagged = brown.tagged_words(categories='news', tagset='universal')

# pair of (word, tag) that comes before the word news
print(sorted(set(a for (a, b) in bigrams(brown_news_tagged) if b[0] == 'news')))
```

```
[('League', 'NOUN'), ('Romantic', 'ADJ'), ('The', 'DET'), ('a', 'DET'), ('first', 'ADJ'), ('foreign', 'AD
J'), ('good', 'ADJ'), ('made', 'VERB'), ('startling', 'ADJ'), ('the', 'DET')]
```

3. What are the words that are highly ambiguous as to their part-of-speech tags ((i.e. the word has more than 3 pos tags) in the brown corpus (categories=”reviews”). (Please use the universal tagset)

In [3]:
```
from nltk import ConditionalFreqDist

# get the tagged words for reviews category from brown corpus
brown_reviews_tagged = brown.tagged_words(categories='reviews', tagset='universal')
data = ConditionalFreqDist((w.lower(), t) for (w, t) in brown_reviews_tagged)

# run a for loop on the data such that we include only cases where more than 3 pos are observed
for word in sorted(data.conditions()):
    if len(data[word]) > 3:
        tags = [t for (t, _) in data[word].most_common()]
        print(word, data[word].most_common(10))
```

```
close [('ADJ', 6), ('ADV', 4), ('VERB', 1), ('NOUN', 1)]
that [('ADP', 167), ('PRON', 115), ('DET', 65), ('ADV', 1)]
```

4.Train a unigram tagger on the brown corpus (categories="humor"). a) Split the data into training and testing dataset- training on the 95% of data and testing on the remaining 5%. b) Evaluate the performance of this tagger. c) Use this tagger to tag some new text ['this','is','a','NLP','class']. d) Observe that some words are not assigned a tag. Explain why not? (Please do not use the universal tagset)

In [4]:
```
from nltk.tag import UnigramTagger

# get the tagged words for the category humor
brown_tagged_sents = brown.tagged_sents(categories='humor')

# a) split the data
limit = round(0.95 * len(brown_tagged_sents))
training_data, test_data = brown_tagged_sents[ :limit], brown_tagged_sents[limit: ]
unigram_tagger = UnigramTagger(training_data)

#b) evaluate the performance of the tagger
print("Performance of the tagger is", round((unigram_tagger.evaluate(test_data)*100),2), "%")
```

```
Performance of the tagger is 70.33 %
```

In [5]:
```
#c) initiating the tagger with a new text
unigram_tagger.tag(['this','is','a','NLP','class'])

# d) Explanation to what we see from ouptut in part c)
# Ans: The words never appeared in the training text, and therefore the tagger can't speculate the word's
```

Out[5]:
```
[('this', 'DT'), ('is', 'BEZ'), ('a', 'AT'), ('NLP', None), ('class', 'NN')]
```

5.Explore the nps_chat corpus and find out what part-of-speech tags occur before a noun, with the most frequent ones first.(Please use the universal tagset)

In [6]:
```
# create taggeed words for nps_chat corpus
nps_chat_tagged = nltk.bigrams(nltk.corpus.nps_chat.tagged_words(tagset = 'universal'))

# define the condition for post tags before noun
pos_noun = [a[1] for (a,b) in nps_chat_tagged if b[1] == 'NOUN']

# print results
print(FreqDist(pos_noun).most_common())
```

```
[('X', 2558), ('DET', 1308), ('NOUN', 1262), ('VERB', 947), ('ADJ', 823), ('PRON', 676), ('.', 636), ('AD
P', 567), ('CONJ', 238), ('NUM', 223), ('ADV', 218), ('PRT', 189)]
```

6. Explore the brown corpus (categories="romance") to find out all tags starting with VB and its associated (word, frequency) pairs (no more than 6 pairs). (Please do not use the universal tagset)

For example, one of the outputs should look like:

VBG [('going', 59), ('looking', 36), ('trying', 23), ('thinking', 21), ('watching', 20), ('taking', 19)]

In [7]:
```
# writing a functiont that takes all tags and words where the tag starts with VB
def findtags(tag_prefix, tagged_text):
    cfd = nltk.ConditionalFreqDist((t, w) for (w, t) in tagged_text if t.startswith(tag_prefix))
    return dict((t, cfd[t].most_common(6)) for t in cfd.conditions())

# get the tagged words for romance category and the tag_prefix in a dictioanry
tagdict = findtags('VB', brown.tagged_words(categories='romance'))

# print the values
for tag in sorted(tagdict):
    print(tag, tagdict[tag])
```

```
VB [('get', 92), ('know', 88), ('go', 76), ('see', 74), ('take', 62), ('say', 59)]
VB+PPO [("Let's", 10), ("let's", 5)]
VBD [('said', 318), ('went', 82), ('thought', 80), ('came', 75), ('knew', 69), ('looked', 68)]
VBG [('going', 59), ('looking', 36), ('trying', 23), ('thinking', 21), ('watching', 20), ('taking', 19)]
VBG+TO [('gonna', 4)]
VBG-TL [("Racin'", 1), ('Dancing', 1), ('Surviving', 1)]
VBN [('got', 36), ('come', 29), ('done', 29), ('gone', 25), ('seen', 20), ('made', 20)]
VBN+TO [('gotta', 1)]
VBN-TL [('United', 3), ('Armed', 1), ('Forked', 1)]
VBZ [('says', 7), ('wants', 7), ('goes', 5), ('gets', 4), ('thinks', 4), ('makes', 4)]
```

7. Write programs to process the Brown Corpus and find answers to the following questions (Please do not use the universal tagset):

a.Which nouns are more common in their plural form (e.g. tag='NNS'), rather than their singular form (e.g.tag='NN')? (Only consider regular plurals, formed with the -s suffix.)

b.What do the 10 most frequent tags represent in the Brown Corpus? Please output the tags and explain.

In [8]:
```
# a)

# list all the singular and plural words
singular = set([w.lower() for (w, t) in brown.tagged_words() if t == 'NN'])
plurals = set([w.lower() for (w, t) in brown.tagged_words() if t == 'NNS'])

# check for the common regular plurals only with -s suffix
common = [a for a in singular if a + "s" in plurals]

# get FreqDist for singular nouns and plural nouns
singular_fd = FreqDist(w.lower() for (w, _) in brown.tagged_words() if w in singular)
plurals_fd = FreqDist(w.lower() for (w, _) in brown.tagged_words() if w in plurals)

# find out which words are more common in the plural form
common_plurals = [(plurals_fd[i + 's'], i + 's', singular_fd[i], i) for i in common if plurals_fd[i + 's']

# print the common plural - top 25
sorted(common_plurals, reverse = True)[:25]
# it first shows the count of each plural form with the plural word, and
# then the count of singular form with the singular word
```

Out[8]:
```
[(943, 'years', 649, 'year'),
 (391, 'eyes', 119, 'eye'),
 (361, 'things', 331, 'thing'),
 (312, 'members', 133, 'member'),
 (291, 'means', 199, 'mean'),
 (269, 'words', 261, 'word'),
 (204, 'students', 109, 'student'),
 (193, 'minutes', 54, 'minute'),
 (188, 'months', 130, 'month'),
 (179, 'conditions', 89, 'condition'),
 (173, 'hours', 145, 'hour'),
 (169, 'miles', 42, 'mile'),
 (160, 'terms', 79, 'term'),
 (150, 'friends', 125, 'friend'),
 (138, 'methods', 137, 'method'),
 (125, 'sales', 44, 'sale'),
 (115, 'arms', 91, 'arm'),
 (106, 'leaders', 69, 'leader'),
 (103, 'elements', 52, 'element'),
 (102, 'factors', 71, 'factor'),
```

```
 (99, 'events', 81, 'event'),
 (98, 'techniques', 58, 'technique'),
 (97, 'dollars', 43, 'dollar'),
 (96, 'institutions', 38, 'institution'),
 (95, 'trees', 56, 'tree')]
```

In [9]:

```
# I also verified if it works or not and picked examples like days which doesn't show up because the singu
# 'day' is more common.

print(singular_fd['day'])
print(plurals_fd['days'])

# here singular form is more common and appeared 623 times over 377 times for plural form and hence this g
```

```
623
377
```

In [10]:

```
# b)

# condition for tags which in brown tagged words. Then we print the 10 most common records
fd = nltk.FreqDist([t for (_, t) in brown.tagged_words()])
print(fd.most_common(10))
```

```
[('NN', 152470), ('IN', 120557), ('AT', 97959), ('JJ', 64028), ('.', 60638), (',', 58156), ('NNS', 55110),
('CC', 37718), ('RB', 36464), ('NP', 34476)]
```

8. Write code to search the Brown Corpus for particular words and phrases according to tags, to answer the following questions (please do not use the universal tagset):

a. Produce an alphabetically sorted list of the distinct words tagged as MD.

b. Identify three-word prepositional phrases of the form IN + AT + NN (eg. in the lab).

In [11]:
```
# a)

# Just one condition for distinct words tagged as MD
print(sorted(set([w[0] for w in brown.tagged_words() if w[1] == 'MD'])))
```

```
['Can', 'Could', 'May', 'Might', 'Must', 'Ought', 'Shall', 'Should', 'Will', 'Would', "c'n", 'can', 'cold
e', 'could', 'dare', 'kin', 'maht', 'mai', 'may', 'maye', 'mayst', 'might', 'must', 'need', 'ought', 'shal
l', 'should', 'shuld', 'shulde', 'wil', 'will', 'wilt', 'wod', 'wold', 'wolde', 'would']
```

In [12]:
```
# b)

# 3 word prepostitional phrases of the form IN + AT + NN
def process(sentence):
    for (w1,t1), (w2,t2), (w3,t3) in nltk.trigrams(sentence):
        if (t1 == 'IN' and t2 == 'AT' and t3 == 'NN'):
            print(w1, w2, w3)
```

In [13]:
```
# print the phrases
for tagged_sent in brown.tagged_sents():
    process(tagged_sent)
```

```
of the election
for the manner
in the election
to the end
on a number
of the law
through the welfare
in the state
with the exception
in the future
in the appointment
in a manner
for the purpose
at the jail
for the mayor
than a year
on the petition
from the audience
for a state
```

9. Use a default dictionary and itemgetter (n) to sort the most frequent tags used in the brown corpus (categories="reviews"). Please first convert the tags into the universal tags.

In [14]:
```
from operator import itemgetter

# get the tagged words for brown corpus from review category
brown_reviews_tagged = [t for (w,t) in brown.tagged_words(categories="reviews", tagset = 'universal')]

# use the default dictionary from nltk as asked in the ques to set a counter
counts = nltk.defaultdict(int)
for t in brown_reviews_tagged:
    counts[t] += 1

# print the tags with their frequency using itemgetter(1) as the key
print(sorted(counts.items(), key=itemgetter(1), reverse=True))
```

```
[('NOUN', 10528), ('VERB', 5478), ('.', 5354), ('ADP', 4832), ('DET', 4720), ('ADJ', 3554), ('ADV', 2083),
('CONJ', 1453), ('PRON', 1246), ('PRT', 870), ('NUM', 477), ('X', 109)]
```

10. Explore the brown corpus (categories="learned") to find out the most 200 frequent words and store their most likely tags. We can then use this information as the model for a "lookup tagger" (an NLTK UnigramTagger). If the words are not among the 200 most frequent words, we would like to assign the default tag of "NN" to them. Then use this lookup tagger to tag a new sentence of your own.

In [15]:
```
# create variables to store frequently distribution and conditionally frequently distributed tags for lear
fd  = FreqDist(brown.words(categories='learned'))
cfd = ConditionalFreqDist(brown.tagged_words(categories='learned'))

# creating a dictionary for possible tags for 200 most common words and then checking the same with unigra
likely_tags     = dict((word, cfd[word].max()) for (word, _) in fd.most_common(200))
baseline_tagger = nltk.UnigramTagger(model=likely_tags)

# assign the default tag of NN to them
baseline_tagger = nltk.UnigramTagger(model=likely_tags, backoff=nltk.DefaultTagger('NN'))
```

In [16]:
```
# running the experiment for a new example
sent = 'Today is such a nice sunny day'
print(baseline_tagger.tag(sent.split()))
```

```
[('Today', 'NN'), ('is', 'BEZ'), ('such', 'JJ'), ('a', 'AT'), ('nice', 'NN'), ('sunny', 'NN'), ('day', 'N
N')]
```