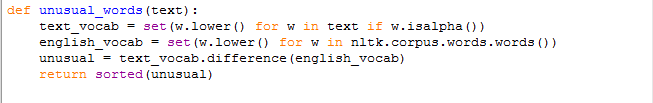
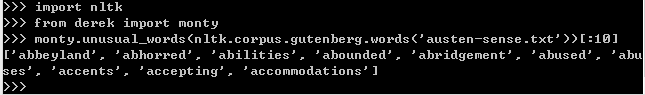
词典或词典资源是一个词和/或短语及其相关信息的集合。例如词性和词意定义等相关信息。词典资源附属于文本，而且通常在文本的基础上创建和丰富。词汇表和词频都是词典资源。

词项包括了词目（词条）和其他附加信息（如词意和词性）。两个含义不同但拼写相同的词条被称为同音异义词。

词汇列表语料库

NLTK中包含一些仅仅包含词汇列表的语料库。词汇语料库是UNIX中的/usr/dict/words文件，被一些拼写检查程序所使用。

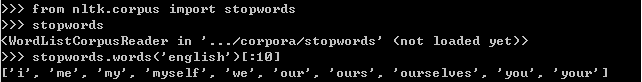




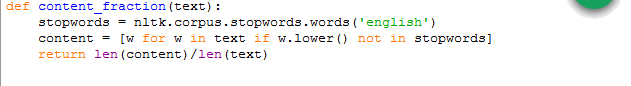
在这个例子中，我们将一本书中的单词与词汇语料库中的单词比较，选出书中独有的一些词，这些词便是罕见词或者是拼写错误的词。

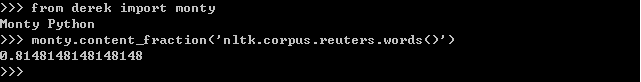
与聊天语料库比较，结果更为明显。

这里要介绍一个非常有用的语料库，叫做停用语料库，包括的是高频词汇，如the,to和and，这些词在分析是毫无用处而数量巨大，十分讨厌。



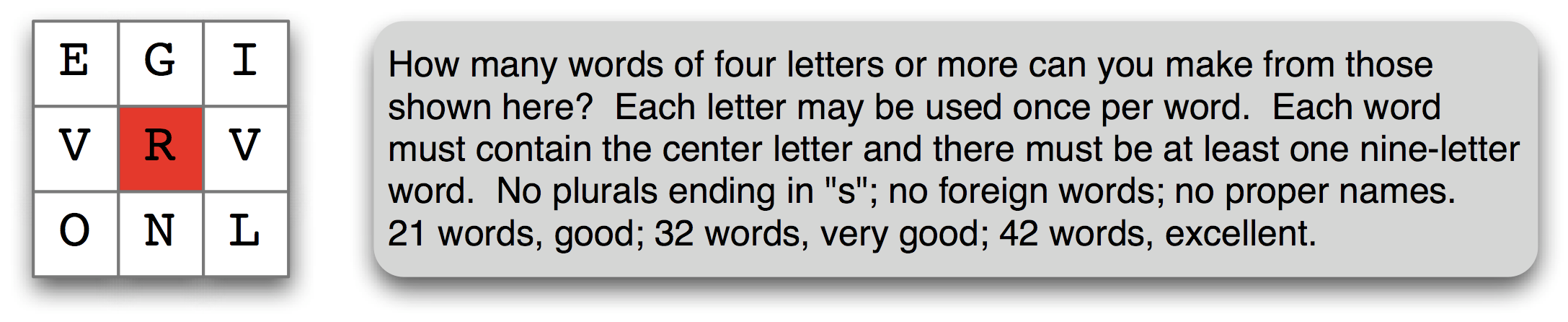
可惜没有中文，实在遗憾





通过stopwords词库，我们删去了约20%的词。

词汇列表可以用来完成下面的猜词游戏



单词必须包含R，而且九宫格中的每个字母只能用一次。

puzzle\_letters = nltk.FreqDist('egivrvonl')



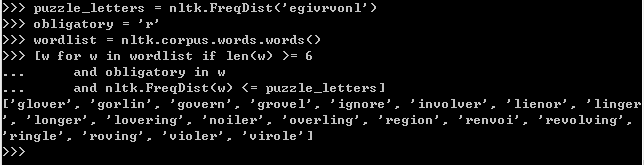
obligatory = 'r'

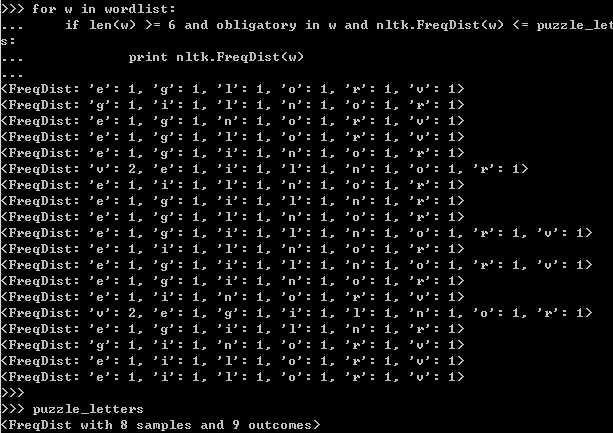
wordlist = nltk.corpus.words.words()

[w for w in wordlist if len(w) >= 6

and obligatory in w

and nltk.FreqDist(w) <= puzzle\_letters]





被选中的每个词都包含r而且每个字母的出现频次都小于等于九宫格

另外一个词汇列表是名字语料库，包含8000个按性别分类的名字。分性别存放。

下面我们查找一下男女通用的名字

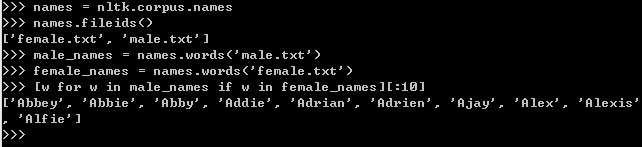
names = nltk.corpus.names

names.fileids()

male\_names = names.words('male.txt')

female\_names = names.words('female.txt')

[w for w in male\_names if w in female\_names][:10]



我们再来分析一下以某个字母结尾的名字与性别的关系

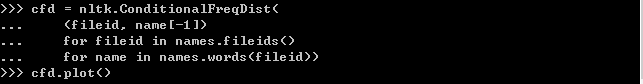
cfd = nltk.ConditionalFreqDist(

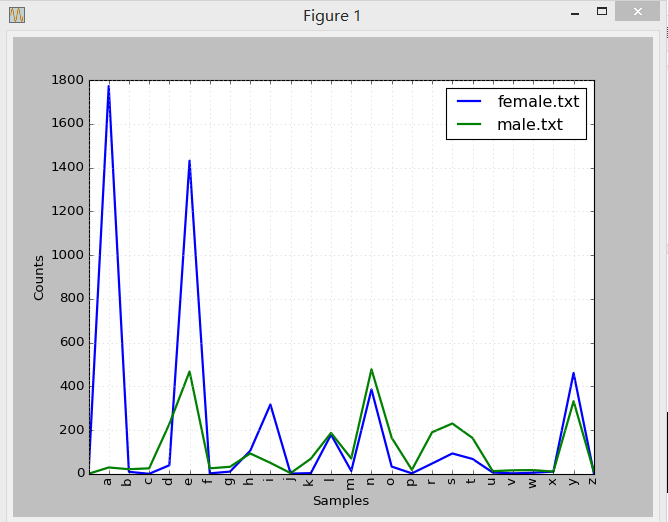
(fileid, name[-1])

for fileid in names.fileids()

for name in names.words(fileid))

cfd.plot()





由此图可以看出，以a结尾的几乎都是女性。

发音的词典

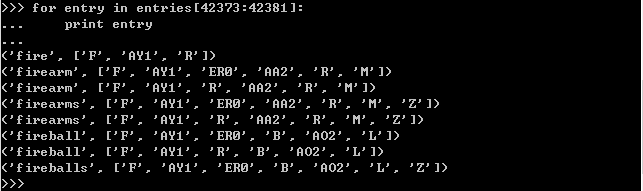
nltk中包含美国英语的CMU发音词典，它是为语音合成器而设计的。

entries = nltk.corpus.cmudict.entries()

len(entries)

for entry in entries[39943:39951]

print entry



fire有两个发音（美国英语）：单音节FAY1R和双音节FAY1ER0。CMU发音词典中的符号是从Arpabet（[美国](http://baike.haosou.com/doc/2188339.html)高级研究计划署（Advanced Research Project Agency）的简称）来的。

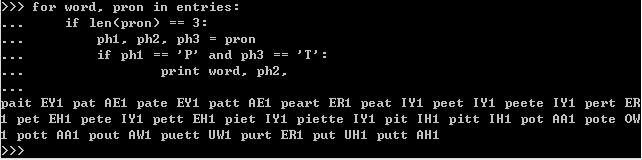
for word, pron in entries:

if len(pron) == 3:

ph1, ph2, ph3 = pron

if ph1 == 'P' and ph3 == 'T':

print word, ph2,

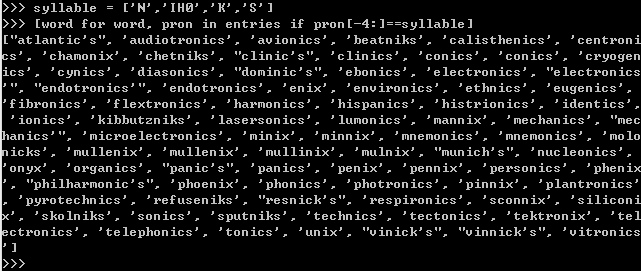


这个例子找出了三个音节，以音素P开头，以T结尾的单词。

我们再来找一下发音结尾语nicks相似的词汇

syllable = ['N','IH0','K','S']

[word for word, pron in entries if pron[-4:]==syllable]



**>>> [w for w, pron in entries if pron[-1] == 'M' and w[-1] == 'n']  
['autumn', 'column', 'condemn', 'damn', 'goddamn', 'hymn', 'solemn']  
>>> sorted(set(w[:2] for w, pron in entries if pron[0] == 'N' and w[0] != 'n'))  
['gn', 'kn', 'mn', 'pn']**

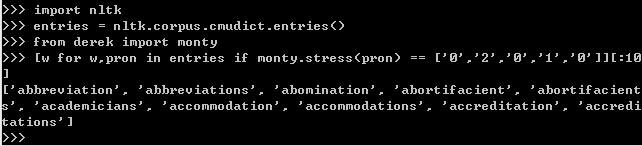
发音和拼读可以不匹配

音素包含数字表示主重音（1），次重音（2）和无重音（0）.

def stress(pron):

return [char for phone in pron for char in phone if char.isdigit()]

这里看出python的语言实在简洁强大，双层嵌套循环。



找出所有音素P开头的三音素词，以第一个和最后一个音素来分组。

entries = nltk.corpus.cmudict.entries()

p3=[(pron[0]+'-'+pron[2],word) for (word,pron) in entries if pron[0] == 'P' and len(pron)==3]

cfd = nltk.ConditionalFreqDist(p3)

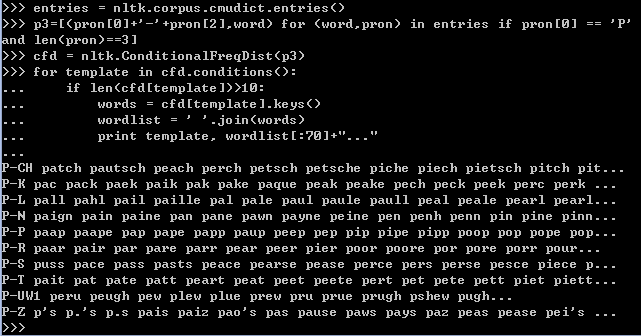
for template in cfd.conditions():

if len(cfd[template])>10:

words = cfd[template].keys()

wordlist = ' '.join(words)

print template, wordlist[:70]+"..."



词典数据结构

prondict = nltk.corpus.cmudict.dict()

prondict['fire']



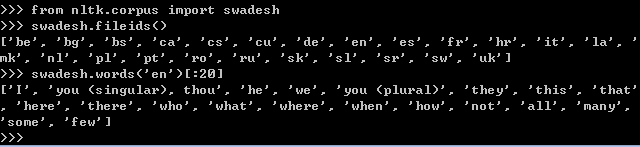
比较词表

nltk包含了Swadesh wordlists，包括集中语言的约200个常用词的列表。语言标识符使用ISO639双字码。

from nltk.corpus import swadesh

swadesh.fileids()

swadesh.words('en')[:20]



可以使用entries方法来指定一个语言链表来访问多语言中的同源词。

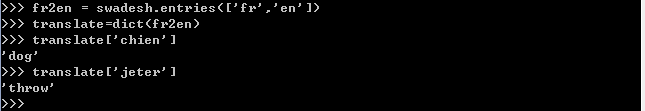
这样可以做出一个翻译词典用。

fr2en = swadesh.entries(['fr','en'])

translate=dict(fr2en)

translate['chien']

translate['jeter']



toolbox

这个工具有点问题，用不了。