In [1]:

```
import numpy as np
import pickle
import matplotlib.pyplot as plt
import copy
import random
```

Loading data from pkl file

In [2]:

```
with open('alice_parsed.pkl','rb') as f:
    u = pickle._Unpickler(f)
    u.encoding = 'latin1'
    data = u.load()
count, next_word_count = data[0], data[1]
```

Q2(b) i

In [3]:

```
def getWordProbability(word, count=count, next_word_count = next_word_count):
    return count[word]/sum(count.values())

# Testing
getWordProbability('rabbit')
```

Out[3]:

```
0.0016590000754090944
```

Q2(b) ii

Conditional Probability

In [4]:

```
def getConditionalProbability(x, y, count=count, next_word_count = next_word_count):
    word = x
    nextWord = y

    if nextWord not in next_word_count[word]:
        return 0

    nextWordGivenWordCount = next_word_count[word][nextWord]
    nextWordAll = sum(next_word_count[word].values())
    return nextWordGivenWordCount/nextWordAll

# Testing
getConditionalProbability('rabbit','just')
```

Out[4]:

```
0.022727272727272728
```

Q2(c) iii

From Bayes' theorem

$P(A \mid B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

$A, B =$ events

$P(A \mid B) \quad =$ probability of A given B

$P(B \mid A) \quad =$ probability of B given A

$P(A), P(B) =$ the independent probabilities of A and B

Here,

A = nextWord

B = word

$P(nextWord \mid word) = \frac{P(word|nextWord) \cdot P(nextWord)}{P(word)}$

In [7]:

```
def predict(word, topk, count=count, next_word_count = next_word_count):

    possibleNextWords = next_word_count[word]
    ans = []
    pWord = getWordProbability(word)
    for nextWord in possibleNextWords.keys():
        pNextWord = getWordProbability(nextWord)
        bayesEstimate =  getConditionalProbability(word, nextWord) * getWordProbability(nextWord)/ pWor
d
        ans.append((nextWord, bayesEstimate))
    topk = min(len(possibleNextWords), topk)
    return [(k,v) for k, v in sorted(ans, key=lambda item: item[1], reverse = True)][:topk]
#     return [(k) for k, v in sorted(ans, key=lambda item: item[1], reverse = True)][:topk]

print ("word most likely to follow 'a' is: " ,predict('a',1)[0])
print ("word most likely to follow 'the' is: " ,predict('the',1)[0])
print ("word most likely to follow 'splendidly' is: " ,predict('splendidly',1)[0])
print ("word most likely to follow 'exclaimed' is: " ,predict('exclaimed',1)[0])
```

```
word most likely to follow 'a' is:  ('little', 0.019377904182022114)
word most likely to follow 'the' is:  ('queen', 0.001647382599763552)
word most likely to follow 'splendidly' is:  ('dressed', 1.0)
word most likely to follow 'exclaimed' is:  ('alice', 32.083333333333336)
```