```
 1 #Jiaxu Zhu
 2 #CSE250A hw3.5
 3 from math import log
 4
 5 class TokenEntry:
 6     totalCount = 0
 7     def __init__(self, index, unigram, bigram):
 8         self.index = index
 9         self.unigram = unigram
10         self.bigram = bigram
11
12 def pu(token):
13     if not token in tokenDict:
14         token = '<UNK>'
15     return tokenDict[token].unigram / TokenEntry.totalCount
16
17 def pb(token1, token2):
18     if not token1 in tokenDict:
19         token1 = '<UNK>'
20     if not token2 in tokenDict:
21         token2 = '<UNK>'
22     if not token2 in tokenDict[token1].bigram:
23         #print 'Not in Corpus %s %s' % (token1, token2)
24         return 0
25     else:
26         return tokenDict[token1].bigram[token2] / tokenDict[token1].unigram
27
28 def pm(token1, token2, l):
29     return l * pu(token2) + (1-l) * pb(token1, token2)
30
31 def lu(sentence):
32     tokens = sentence.upper().strip('\n').split(' ')
33     p = 1
34     for token in tokens:
35         p *= pu(token)
36     return log(p)
37
38 def lb(sentence):
39     tokens = sentence.upper().strip('\n').split(' ')
40     p = 1
41     for i in range(0, len(tokens)):
42         if i == 0:
43             p *= pb('<s>', tokens[i])
44         else:
45             p *= pb(tokens[i-1], tokens[i])
46     return log(p)
47
48 def lm(sentence, l):
49     tokens = sentence.upper().strip('\n').split(' ')
50     p = 1
51     for i in range(0, len(tokens)):
52         if i == 0:
53             p *= pm('<s>', tokens[i], l)
54         else:
55             p *= pm(tokens[i-1], tokens[i], l)
56     return log(p)
```

```
 57
 58
 59 unigram = [];
 60 tokenList =[]
 61 tokenDict = {}
 62 tokenFile = open('vocab.txt', 'r')
 63 index = 0
 64 for token in tokenFile.readlines():
 65     token = token.strip('\n');
 66     tokenList.append(token)
 67     tokenDict[token] = TokenEntry(index, 0, {})
 68
 69 unigramFile = open('unigram.txt', 'r')
 70 for line in unigramFile.readlines():
 71     tokenDict[tokenList[index]].unigram = float(line)
 72     TokenEntry.totalCount += tokenDict[tokenList[index]].unigram
 73     index += 1
 74
 75 bigramFile = open('bigram.txt', 'r')
 76 for line in bigramFile.readlines():
 77     line = line.split('\t');
 78     index1 = int(line[0]) - 1
 79     index2 = int(line[1]) - 1
 80     count = float(line[2])
 81     tokenDict[tokenList[index1]].bigram[tokenList[index2]] = count
 82
 83 for token in tokenList:
 84     if token[0] == 'M':
 85         print '%s & %f \\\\' % (token, tokenDict[token].pu())
 86         print '\\hline'
 87
 88 b = sorted(tokenDict['THE'].bigram.items(), key=lambda x: x[1], reverse=True)
 89 for iter in range(0, 10):
 90     print '%s & %f \\\\' % (b[iter][0], b[iter][1] / tokenDict['THE'].unigram)
 91     print '\\hline'
 92
 93 print lu('The stock market fell by one hundred points last week')
 94 print lb('The stock market fell by one hundred points last week')
 95
 96 print lu('The sixteen officials sold fire insurance')
 97 print lb('The sixteen officials sold fire insurance')
 98
 99 for l in range(1,101):
100     print lm('The sixteen officials sold fire insurance', float(l)/100)
```