cse250a_hw1

October 9, 2018

```
In [ ]: import numpy as np
        import math
        import pandas as pd
        import string
```

Part(a)

```
In [183]: pw = pd.read_csv('hw1_word_counts_05.txt', sep=" ",header=None)
          pw.columns = ['word','count']
```

```
In [184]: # Calculate P(w)
          pw.loc[:,'P(w)'] = pw['count'].apply(lambda x: x/sum(pw['count']))
          pw.sort_values('P(w)', ascending=False).head(15)
```

Out[184]:

	word	count	P(w)
5821	THREE	273077	0.035627
5102	SEVEN	178842	0.023333
1684	EIGHT	165764	0.021626
6403	WOULD	159875	0.020858
18	ABOUT	157448	0.020542
5804	THEIR	145434	0.018974
6320	WHICH	142146	0.018545
73	AFTER	110102	0.014365
1975	FIRST	109957	0.014346
1947	FIFTY	106869	0.013943
4158	OTHER	106052	0.013836
2073	FORTY	94951	0.012388
6457	YEARS	88900	0.011598
5806	THERE	86502	0.011286
5250	SIXTY	73086	0.009535

```
In [174]: pw.sort_values('P(w)', ascending=True).head(14)
```

Out[174]:

	word	count	P(w)
3554	MAPCO	6	7.827935e-07
712	BOSAK	6	7.827935e-07
895	CAIXA	6	7.827935e-07
4160	OTTIS	6	7.827935e-07

```
5985  TROUP      6  7.827935e-07
1107  CLEFT      7  9.132590e-07
2041  FOAMY      7  9.132590e-07
977   CCAIR      7  9.132590e-07
5093  SERNA      7  9.132590e-07
6443  YALOM      7  9.132590e-07
5872  TOCOR      7  9.132590e-07
3978  NIAID      7  9.132590e-07
4266  PAXON      7  9.132590e-07
1842  FABRI      7  9.132590e-07
```

In [185]:
```
# function to calculate P(E|w)
def p_e_given_w(word,correct,incorrect):
    out = 1
    s   = set()
    for i in range(len(correct)):
        if correct[i]==' ':
            s.add(word[i])
        elif correct[i]!=word[i]:
            out = out*0
        else:
            out = out*1

    if bool(set(correct).intersection(s)):
            out = out*0

    for j in range(len(incorrect)):
        if incorrect[j] in word:
            out = out*0
        else:
            out = out*1

    return out
```

In [186]:
```
# calculation of P(W|E)
correct = [' ',' ',' ',' ',' ']
incorrect = ['A','I',' ',' ',' ']

pw.loc[:,'P(E|w)'] = pw['word'].apply(lambda x: p_e_given_w(x,correct,incorrect))
```

In [187]:
```
def product(x):
    return x[2]*x[3]
pw.loc[:,'P(E|w)*P(w)'] = pw.apply(product, axis=1)
```

In [188]:
```
pw.loc[:,'P(w|E)'] = pw['P(E|w)*P(w)'].apply(lambda x: x/sum(pw['P(E|w)*P(w)']))
pw.head()
```

Out[188]:

	word	count	P(w)	P(E|w)	P(E|w)*P(w)	P(w|E)
0	AARON	413	0.000054	0	0.0	0.0

```
1  ABABA    199  0.000026       0          0.0     0.0
2  ABACK     64  0.000008       0          0.0     0.0
3  ABATE     69  0.000009       0          0.0     0.0
4  ABBAS    290  0.000038       0          0.0     0.0
```

In [189]:
```
word = pw['word']
list_char = list(string.ascii_uppercase)
pwe = pw['P(w|E)']

def l_in_word(l,word,correct):
    if set(l).issubset(set(correct)):
        return 0
    elif l in word:
        return 1
    else:
        return 0

ple = [0] * 26
i = 0
for ch in list_char:
    for k in range(len(word)):
        ple[i] += l_in_word(ch,word[k],correct)*pwe[k]
    i += 1
```

In [190]:
```
max_value = max(ple)
max_index = ple.index(max_value)
print('P(Li = l|E):', max_value)
print('Best next guess is:', list_char[max_index])
```

```
P(Li = l|E): 0.6213518619180538
Best next guess is: E
```