Week 7 : Digital humanities

Q2. Give one problem when applying the direct estimation as suggested by Mary. Provide two drawbacks related to Laplace smoothing.

Problem ocuure if any word not present in the word tokens then related probability will zero for corresponding word

Drawbacks of laplace smoothing:

- If word type id infinite then too much probability mass is shifted towards unseen n-grams
- Probability of rare (or unseen) n-grams is overestimated
- All unseen n-grams are smoothed in the same way

Q3. Generate the bigrams of tokens. Provide the ten most frequent bigrams from tweets written by a woman or a man.

In [1]:

```python
# Loading data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
import nltk
from nltk.tokenize import word_tokenize
```

In [2]:

```python
aListOfTweets = []
myInputFile = open('./corpus/tweets.female.txt', 'r', encoding="utf8")
aLine = myInputFile.readline()
IDs=[]
while aLine:
    myLine = aLine.split('\t')
    IDs.append(myLine[0])

    aFinalLine = ''
    for aSubLine in myLine[3:]:
        aSubLine = re.sub('\n', '', aSubLine)
        aFinalLine += aSubLine + ' '
    aListOfTweets.append(aFinalLine)
    aLine = myInputFile.readline()

print(aListOfTweets[0], IDs[1])
```

```
alex is too nice for love island :(   0
```

In [3]:

```
b_gram = []
for total_words in aListOfTweets:
    b_gram.extend(list(nltk.bigrams(total_words.split())))


frequency = nltk.FreqDist(b_gram)
dict_sorted = {k: v for k, v in sorted(frequency.items(), key=lambda item: item[1
bgram_10_words_f = list(dict_sorted.keys())[:20]

print("Top 20 bigram tweets and their counts by female are: ")
for key in bgram_10_words_f:
    print(key,dict_sorted[key])
```

```
Top 20 bigram tweets and their counts by female are:
('rt', '@') 39059
("'", 's') 10165
('’', 's') 6098
("'", 't') 5077
('i', "'") 4834
('.', 'urllink') 4350
('in', 'the') 4206
('’', 't') 3644
('of', 'the') 3620
('…', 'urllink') 3122
('urllink', '…') 3112
(':', 'urllink') 3029
("'", 'm') 2931
('it', "'") 2807
('i', '’') 2775
('urllink', 'urllink') 2594
('.', 'i') 2460
('on', 'the') 2340
('.', '#') 2302
('to', 'be') 2239
```

In [4]:

```
aListOfTweets_male = []
myInputFile_male = open('./corpus/tweets.male.txt', 'r', encoding="utf8")
aLine = myInputFile_male.readline()
IDs_male=[]
while aLine:
    myLine = aLine.split('\t')
    IDs_male.append(myLine[0])

    aFinalLine = ''
    for aSubLine in myLine[3:]:
        aSubLine = re.sub('\n', '', aSubLine)
        aFinalLine += aSubLine + ' '
    aListOfTweets_male.append(aFinalLine)
    aLine = myInputFile_male.readline()

print(aListOfTweets_male[0], IDs_male[1])
```

```
@ jennycastle 96 ahaha last time acting reckless 😂😂   6
```

```
In [5]: b_gram = []
        for total_words in aListOfTweets_male:
            b_gram.extend(list(nltk.bigrams(total_words.split())))


        frequency = nltk.FreqDist(b_gram)
        dict_sorted = {k: v for k, v in sorted(frequency.items(), key=lambda item: item[1
        bgram_10_words_m = list(dict_sorted.keys())[:20]

        print("Top 20 bigram tweets and their counts by male are: ")
        for key in bgram_10_words_m:
            print(key, dict_sorted[key])
```

```
Top 20 bigram tweets and their counts by male are:
('rt', '@') 30619
("'", 's') 12366
('.', 'urllink') 6800
("'", 't') 6759
('’', 's') 4948
('in', 'the') 4514
('i', "'") 4382
('of', 'the') 4206
('it', "'") 3596
('.', '#') 3050
('on', 'the') 2718
('urllink', '…') 2585
('’', 't') 2567
('.', 'i') 2510
('for', 'the') 2435
("'", 'm') 2401
('don', "'") 2209
('this', 'is') 2182
(':', 'urllink') 2115
('to', 'be') 2112
```

Discussion

In both male and female bigrams, the common tweets such ('rt', '@'), ("'", 's'), (".", 'urllink') etc. have been observed since it is mandated by the Twitter to follow that template. As far as I can see the top tweets are discriminative enough to classify the gender from them