**HW6 網媒所 碩一 R04944047謝恒立**

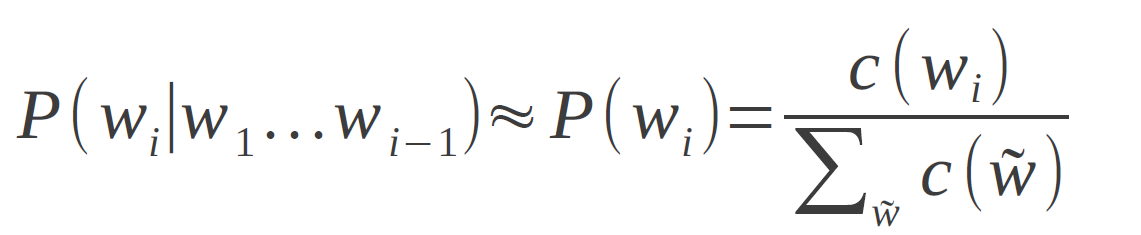
**Intro:**

這次作業要用Naive Bayes 去train Language Model, 再用EM Algorithm算mixture 的期望值

程式部分用python2.7 + sqlite完成, 並使用pypy編譯

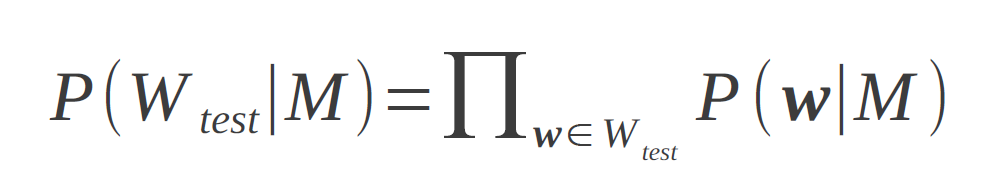
**Naïve Bayes :**

假設每個Unigram彼此間都是獨立的, 所以可以直接拿每個Unigram的probability計算



在train的時候我沒有做smooth而是等到對test data Predict時才做

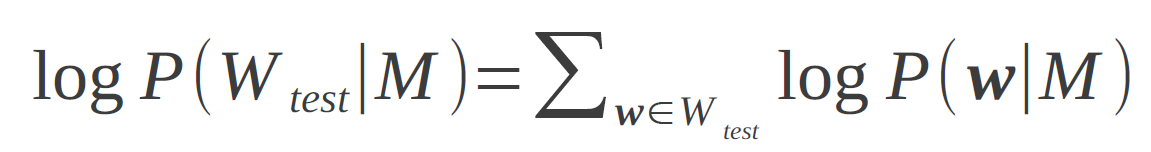
Predict時對test data的每個Unigram求Likelihood



這時P會加入smooth避免沒出現過的字為0

\hat{\theta}_{yi} = \frac{ N_{yi} + \alpha}{N_y + \alpha n} **(θ = M)**

由於機率過小不夠float存(-inf), 因此用了log



而每個doc裡面的所有unigram所算出的Log Likelihood中最大者即為該model的class

求出maxima Likelihood, 完成predict

命中率約64%

**補充:**

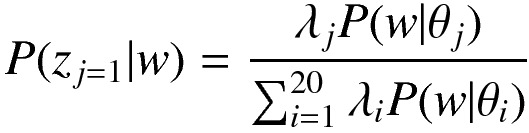
取出Unigram時, 會濾掉一些Stop Words, 另外加入一些counts過高者,

並對@還有.,符號做置換

**EM Algorithm:**

主要針對UnLabel data, 假設每個doc中的每個word都是由已知的20個topics之mixture language mode 生成

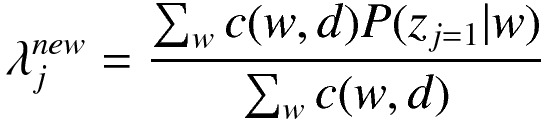
**E-step:**

 j:[1,20]

即為求Zj = 1 時的期望值

**M-step:**

Estimate **λ**

 j:[1,20]

Estimate **θ**

**結論:**

這次作業又被python錶了一次,前面在寫Naive Bayes時因為file list沒有照順序讀進來造成結果不管怎麼tune都是0.05(用猜的差不多),後來是有人建議用寫好的NB驗證一次,確認問題點,雖然花了不少時間搞svm format不過也算是上了一課

在看EM的時候與同學討論一直誤會是要用上課教的那種情況估,先validation取得labeled和mixture之間的最佳比例, 再用得到的λ計算expectation, 這樣每次iterative都要暴力validation且程式的spec有要能更動input data size, 最後還好有寄信問助教,才避免真的去搞一個推不大出來的式子

這次寫IR作業由於python比較熟了一些,寫起來問題比較少,不會像前一次那樣list滿天飛performance又差到爆,不過debug還是有些障礙, 數學的部分因為這次用Naïve和mixture model所以不用搞到複雜的distribution 且M-step有已經導好的公式所以很弱的微積分也避掉了