R08944035 呂翊愷

Digital Speech Processing Homework #3

What I observed:

在切割mapping和測資的字串的時候，原本在想Big5的一個字會是2 bytes的char，該怎麼存這樣子的字。後來發現以空白符號用strtok來切割每個Big5字，用一個char指標存回傳的char陣列，再將char陣列轉換成string型態的字串就好，所以我在之後的Viterbi記錄最可能的字的時候也都統一都用string型態來做處理，解決在處理中文字字串時的麻煩。

在算bigram的機率的時候，我將第一個字和第二個字在一起的機率利用language model算出來，然後再和第一個字出現的機率相乘，原本不覺得這樣有什麼問題，但後來發現language model算出來的機率是以log_{10}為底的數字，而如果要將log裡面的兩數相乘，在log外面的時候必須是相加的，因此我把兩個機率改成用相加的方式。

在實作Viterbi algorithm的時候，我把所有bigram的機率算出來。為了找出機率最大的字，在比大小的時候，忘記language model算出來的機率是以log_{10}為底的數字，所以會是負的，但是我把比大小的初始值設成最小的正值，因此這樣怎麼找都找不到最大值。在這個地方卡了很久，後來把language model算出的bigram印出來才得以發現這個錯誤，把比大小的初始值設成最小的負值才修正完成。

What I have done:

首先，我先把Big5-ZhuYin.map轉換成ZhuYin-Big5.map。這個部分我依照助教的提醒，在讀取檔案和寫入檔案的時候，皆使用'big5-hkscs'的編碼方式。在切割字串的時候，我利用換行符號、空白符號、斜線符號來切割注音以及對應的字，並用python中的dict()來把每個字依照他的注音來分類並輸出成ZhuYin-Big5.map。

在mydisambig的部分，在切割測資的字串後，利用ngram指令train出來的language model，從頭到尾算出每個接下來可能的字bigram的機率。接著在backtracking的時候，從後面到前面，接連將最大機率的字記錄下來，並輸出到檔案，就完成了這次的作業。