DSP hw3 report

b05902109 資工三 柯上優

Environment and Execution

- environment：CSIE workstation
- mapping.py
  - Compile and execution

    ```
    # 方法一
    make map
    # 方法二
    python3 mapping.py [sourse file] [destination file]
    ```

  - Require package：none。
- mydisambig.cpp
  - Compile

    ```
    make
    ```

  - Execute

    ```
    # 方法一
    make run
    # 方法二
    ./mydisambig -text testdata/example.txt -map ZhuYin-Big5.map -lm bigram.lm -
    order 2
    ```

  - Require package： srilm，連結方法如同作業要求中修改makefile參數與編譯sourse code的方法。
  - 注意：請務必按照順序輸入參數，我沒有處理arguement的輸入，順序錯一定會core dump。

What I Have Done

- 建立二維陣列，第一維是string的每一個字，第二維是假如字為注音(不確定是哪個字)，則放入每個可能的中文字。
- 注意前後有放入開始與終止符<s>、</s>。
- 由前往後，對於任兩行，找到並記錄，對於給定前面的最佳解，下一個最可能的中文字。
- 在Viterbi時還需要多判斷機率最大的結尾字，但是由於我們有終止符</s>，所以直接從它開始back tracking，找到最好的解(完整中文字串)。
- 在這裡介紹一些重要的class和methods，也歡迎助教將這些指令分享給其他修課學生，減少他們寫作業的負擔。

```
#include "Vocab.h"
VocabString string[len];
unsigned length = Vocab::parseWords(char* str_src, VocabString* str_des, MAX_LENGTH);
```

- 由於Cpp的string或是char*都無法處理中文字，所以Vocab.h提供了專門處理中文字的資料型態。
- 將ASCII字串str_src轉成可以使用的VocabString，回傳字串長度(以中文字為單位，空白省略)。

```
#include "VocabMap.h"
VocabMap map(Vocab zhuyin, Vocab big5)
map.read(FILE*);

VocabIndex idx = zhuyin.getIndex(VocabString chr);
VocabString chr = zhuyin.getWord(VocabIndex idx);

VocabMapIter iter(map, zhuyin.getIndex(VocabString chr);
iter.init();
iter.next(VocabIndex idx, Prob prob);
```

- 如同Cpp原本的map，這種結構能將含有中文字的檔案變成一個字典。
- 提供雙向查找，可以找到用Index找字和用字找index。
- zhuyin和big5使用相同的VocabIndex，對不同的Vocab使用getWord可分別得到注音與中文。
- 若有一注音對應到多個中文字，可以使用VocabMapIter將其對應的中文字一個個拿出來，idx會存取中文字在map內的index，Prob會有其機率(假如有)。

```
#include "Ngram.h"
VocabMap n_gram(Vocab voabulary, int order)
n_gram.read(FILE*);

VocabIndex idx = vocabulary.getIndex(VocabString chr);
n_gram.wordProb(VocabIndex idx1, VocabIndex* idx2);
```

- 如同Cpp原本的map，這種結構能將含有中文字的檔案與其機率變成一個字典。
- 一樣有getIndex的methods。
- 使用wordProb得到 $P(W_i = c_{idx1} | W_{i-1} = c_{idx2})$ 。