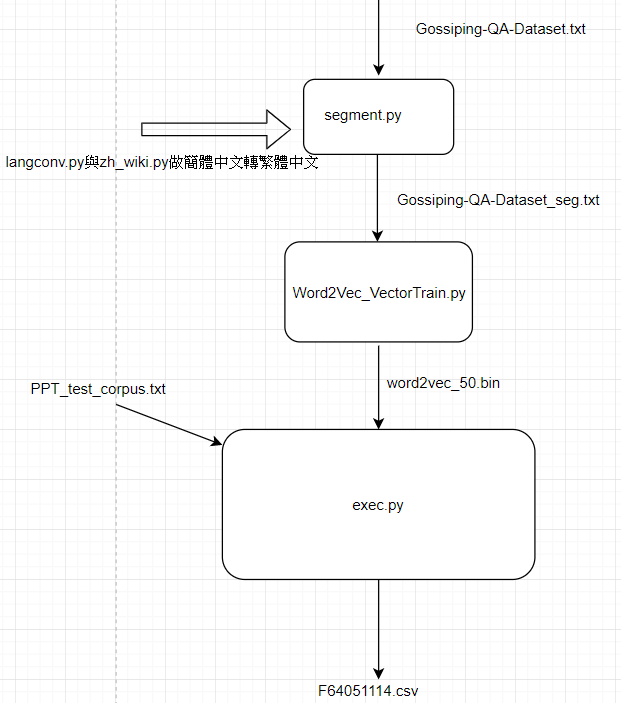
多媒體Project3程式碼說明文件

陳俊宇 學號：F64051114

* 架構圖



* 說明

資料夾內有一份fault.py，是我原本的構想，但後來發現不試用，2000題作答我是以exec.py來執行，不過這邊還是描述一下fault.py之作法。

fault.py是將所有問題都用Gossiping-QA-Dataset.txt內的問題集做相似度分析，再將相似度最高之問題找出來後，把Gossiping-QA-Dataset.txt內該問題對應之答案與4個選項做相似度分析，即可得到正確選項。但是這樣一題的作答時間大概要1分鐘，大量做題肯定無法在時間內完成，所以我做了兩層優化。第一層優化是直接對問題與所有Gossiping-QA-Dataset.txt內的問題做字串比對，如果有一模一樣的問題，那就可以直接跳過問題集相似度分析之階段。第二層優化也是類似的概念，就是直接對4個選項與Gossiping-QA-Dataset.txt內該問題之對應答案做字串比對，如果有一模一樣的答案選項，則直接輸出選項，不做相似度分析。

fault.py其實可以在助教給的500題測資正常運作，且500題只會錯4題。而不能在2000題作答是因為那2000題題目都不是從Gossiping-QA-Dataset.txt內出的(也沒有所謂語意相近)，因為我用相似度比對，將相似度分數最高的題目提取出來，相似度大概都只落在0.5甚至更低，表示這麼做找到的問題是跟原本問題幾乎沒有相關聯，更不用說使用Gossiping-QA-Dataset.txt內該問題後面知答案來作答。

而主要執行的exec.py，是使用助教給的avg\_feature\_vector() function來實作，透過問題與4個選項分別做維度與餘弦函數計算來得到最佳答案，在執行時間上也相當迅速。

另外我發現word轉vector的維度不同時，作答也不完全相同，而維度越高是否作答越精準，這我不敢保證。