P76084091 資工所 李宗樺

系統環境：

作業系統：fedora 29

主程式：Django

實做題目：

針對文件進行Zipf distribution分析。

使用Porter's algorithm 做詞性歸一化，並分析歸一化後文章文字的內容以及其對zipf distribution 之影響。

實做edit distance，對輸入的內容做拼字正確檢查。

實做方法：

基於上一次的作業新增功能。

- Zipf distribution：
    1. 讀取文章，做tokenize。
    2. 移除特殊符號，減少無效字對zipf distribution影響。
    3. 利用python set（集合，相同字視為同一集合）計算詞頻。
    4. 以詞頻排序並輸出最高的30個字，原因是我只想關注重要的字且畫圖後的資料分佈較容易分析。
- Porter's algorithm：
    1. 2. 步驟和上方相同

    3. 在計算詞頻之前，所有字做Porter stemer，詞性歸一化會將同一個字不同詞性的變化轉換成一個字（ex: detect & detection & detected --> detect）。

    4. 以詞頻排序並輸出。
- Edit Distance：
    1. 先從所有pubmed資料集的內文做tokenize，目的是先做出一個辭典當作 edit distance 的答案。
    2. Edit Distance 以 javascript 撰寫，目的是在使用者輸入時即時做錯字偵測，用二維陣列實做DP，把Edit Distance 小於3的字當作備選的正確字，供使用者參考。

問題與討論：

- 做完 Porter's algorithm，在進行 Zipf distribution，因為詞性歸一化的關係，某些的字的字頻會上升改變在zipf distribution的排序。
- 目前 Edit Distance 沒有對備選字做排序，所以最像的字不一定會在最上方，這部份是接下來可以在改進的部份。