Q56084048 蔡沛蓁

使用工具：前端(Html+Js)

後端(Python)

一、 資料蒐集：(關鍵字：**dengue**)

Xml—使用 **pubmed** 有提供全文瀏覽的 book 分別撈出字數**短**(200 字)**中**(約 2000 字)**長**(約 8000 字)各**一**篇，以及**同批作者**寫作出的 book **兩**篇。

Json—使用 **twitter** 中字詞在 **20-100** 間的資料(約 10 篇)。

二、 資料處理：

原：標題、內文、字元、字串、字串頻率

新加入：

1. 搜尋模式：**複合字**搜尋。
2. 內文顯示：內文 label 拆解、字串頻率排序、歸依化字串頻率及排序、Zipf Distribution 圖表(原始字串頻率+歸依化字串頻率)。
3. 其他顯示：複合字的 **edit distance** 及**錯字修正**。

三、 相關文獻顯示：

Zipf Distribution 圖表：使用 **chart.js** 中的 line chart 繪製，data 為排序後的字串頻率。原本使用 mpld3 方式直接從 python 打包丟入 html 發現運算太耗時而作罷。

使用 **edit distance** 演算法，算出兩個字詞間的 edit distance。原始寫法造成迴圈過深而跑不出來，但限制迴圈次數又會算不出來的問題。

使用 **autocorrect** 演算法，推薦出**複合字**可能的**修正字詞**。

四、 問題討論：

Zipf Distribution 圖表：很像一個反比例的圖像。齊夫定律中表示--**第 n 常見的頻率是最常見頻率出現次數的 1/n**。在資料處理中，我發現字數越多越符合這個定律，字數太少(約 150 字以下)看不太出來。

歸依化前後比較：我覺得前 1/5 的字詞都是不太有意義的(Ex. and the..)中間的 2/5~3/5 較容易因歸依化而有異動。

此外，同批作者的文章會有特定慣用詞，出現頻率較高。