IR HW3 資管碩二 R05725034 張鑑霖

1. 執行環境 & 作業系統

   Jupyter & win7

2. 程式語言

   Python3

3. 作業處理邏輯說明

建立前處理後的每篇文章，存在 doc_tf/

建立 training data (class 1~13)的 dictionary

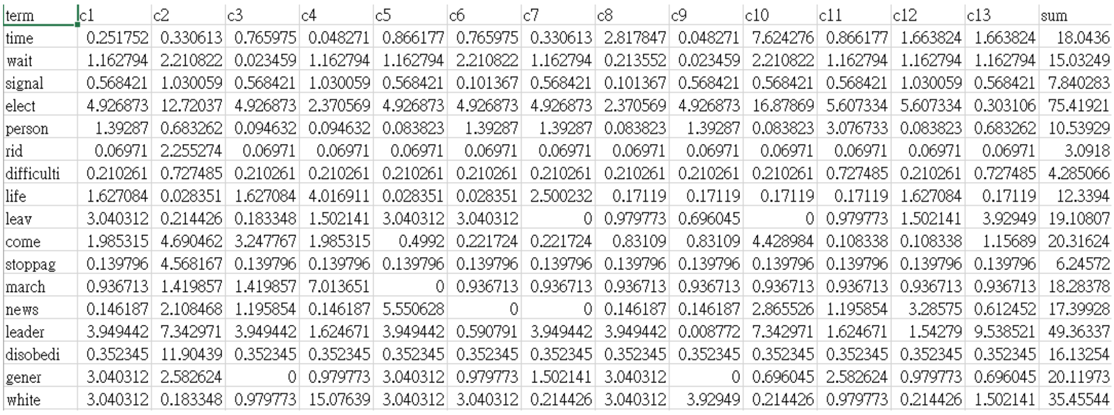
前處理：

對此 dictionary 的每個 term 算它在每個 class 的 likelihood ratio

Likelihood ratio: n11 = term 在 class 中出現次數，n10 = 15-n11

n01 = term 沒在 class 中出現次數，n00 = 180 – n01，其結果大概如下圖，

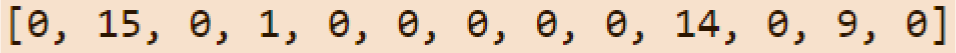
最後我們家總後排序取前 500 個當 term



term	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	sum
time	0.251752	0.330613	0.765975	0.048271	0.866177	0.765975	0.330613	2.817847	0.048271	7.624276	0.866177	1.663824	1.663824	18.0436
wait	1.162794	2.210822	0.023459	1.162794	1.162794	2.210822	1.162794	0.213552	0.023459	2.210822	1.162794	1.162794	1.162794	15.03249
signal	0.568421	1.030059	0.568421	1.030059	0.568421	0.101367	0.568421	0.101367	0.568421	0.568421	0.568421	1.030059	0.568421	7.840283
elect	4.926873	12.72037	4.926873	2.370569	4.926873	4.926873	4.926873	2.370569	4.926873	16.87869	5.607334	5.607334	0.303106	75.41921
person	1.39287	0.683262	0.094632	0.094632	0.083823	1.39287	1.39287	0.083823	1.39287	0.083823	3.076733	0.083823	0.683262	10.53929
rid	0.06971	2.255274	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	0.06971	3.0918
difficulti	0.210261	0.727485	0.210261	0.210261	0.210261	0.210261	0.210261	0.210261	0.210261	0.210261	0.727485	0.210261	0.727485	4.285066
life	1.627084	0.028351	1.627084	4.016911	0.028351	0.028351	2.500232	0.17119	0.17119	0.17119	0.17119	1.627084	0.17119	12.3394
leav	3.040312	0.214426	0.183348	1.502141	3.040312	3.040312	0	0.979773	0.696045	0	0.979773	1.502141	3.92949	19.10807
come	1.985315	4.690462	3.247767	1.985315	0.4992	0.221724	0.221724	0.83109	0.83109	4.428984	0.108338	0.108338	1.15689	20.31624
stoppag	0.139796	4.568167	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	0.139796	6.24572
march	0.936713	1.419857	1.419857	7.013651	0	0.936713	0.936713	0.936713	0.936713	0.936713	0.936713	0.936713	0.936713	18.28378
news	0.146187	2.108468	1.195854	0.146187	5.550628	0	0	0.146187	0.146187	2.865526	1.195854	3.28575	0.612452	17.39928
leader	3.949442	7.342971	3.949442	1.624671	3.949442	0.590791	3.949442	3.949442	0.008772	7.342971	1.624671	1.54279	9.538521	49.36337
disobedi	0.352345	11.90439	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	0.352345	16.13254
gener	3.040312	2.582624	0	0.979773	3.040312	0.979773	1.502141	3.040312	0	0.696045	2.582624	0.979773	0.696045	20.11973
white	3.040312	0.183348	0.979773	15.07639	3.040312	3.040312	0.214426	3.040312	3.92949	0.214426	0.979773	0.214426	1.502141	35.45544

Training:

開啟文章，根據每個 class 的 training 文章裡面的 term 算出現的次數，例如 opposite 在各個 class 的出現次數，有了這個就可以根據此數據算出機率每個 term 對應到每個 class 的機率。



```
[0, 15, 0, 1, 0, 0, 0, 0, 0, 14, 0, 9, 0]
```

Testing:

最後每篇文章的 class 分數都從 0 開始，並根據加上上面 train 出來的機率，最大的就是哪個 class