ML_hw4 Report

電機四 B02504086 陳品融

1. Analyze the most common words in the clusters.

TF:

of	to	in	with	from
using	to	for	how	on
and	linq	is	spring	visual
hibernate	drupal	from	wordpress	to

TF-IDF without stopword:

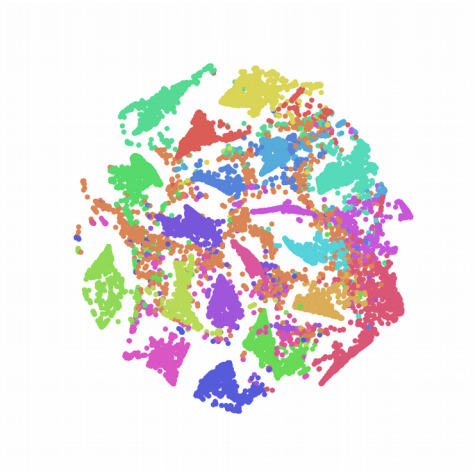
of	to	with	how	drupal
for	magento	on	is	visual
sharepoint	hibernate	haskell	wordpress	and
ajax	the	in	linq	matlab

TF-IDF with stopword:

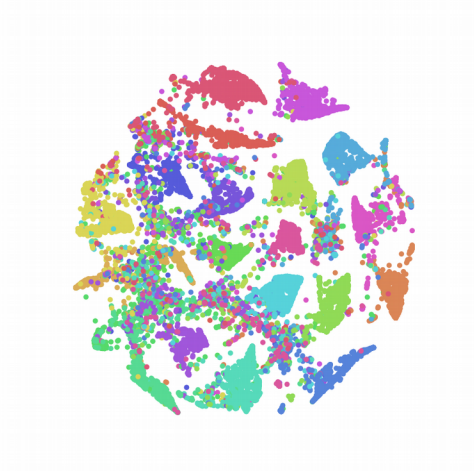
bash	oracle	qt	matlab	drupal
magento	haskell	visual	scala	sharepoint
apache	mac	excel	qt	svn
spring	hibernate	linq	ajax	wordpress

從上面的表我們可以發現，只使用 term frequency 的話，大部分群最常見的字都是一些無關緊要的介系詞，而使用 inverse document frequency 之後，情況有所改善，然而還是有一半的比例是無關的字。最後，如果再加入 sklearn 內建的 stopword set 以及自行觀察出的 stopword，會得到非常好的效果，只有 mac 這一個字不在原本的 tags 中。而其實 mac 跟真正的答案 osx 有著高度的相關性，所以我們可說 TF-IDF with stopword 這個方法確實可以讓我們做出有意義的分群。

2. Visualize the data by projecting onto 2-D space.



Labeled by cluster predictions



Labeled by true labels

由上圖可發現，K-means 所分出來的 cluster 大致上能界定每個群的區域及邊界在哪裡，只有少數地方分布較雜亂無法辨別。至於根據 true labels 所做的分群，我們可以明顯發現每群之間的距離拉得比較開，然而還是有少數的點凌亂重疊在一起，我認爲這是因爲有些 title 內其實包含了兩個 tag 以上，因此本質上很難用一般 cluster 的方法對它做好分群。我想，也就是因爲如此，使得當 num_cluster 設爲 20 時，K-means 在 precision 的表現上比較差(見第四點討論)。

3. Compare different feature extraction methods.

(1)PCA

num_cluster=20	Stopword	F-Measure
BoW	True	0.25744
	False	0.13116
TF-IDF	True	0.29801
	False	0.26166

(2)LSA

num_cluster=20	Stopword	F-Measure
BoW	True	0.58305
	False	0.14355
TF-IDF	True	0.64475
	False	0.31632

由上面的表可發現，影響 perfomace 最大的是降維方式與是否使用 Stopword。我認爲 pca 的 score 會與 LSA 差距如此大是因爲 pca 在做的事是把 data 投影到 variance 最大的維度上，但它背後需假設 data 是 multivariant gaussian distribution，而這個假設對這次的 task 可能不成立。至於爲何加了 stopword performance 會上升這麼多，而加了 idf 卻只讓 F-measure 微幅增加，我認爲是因爲我只把 title 拿進來做 cluster。由於 title 裡的字其實已經大部分都算是關鍵字，且這次的 data 都是從 stack overflow 取下來的，每個 title 間的字彼此有蠻大的相關性與重複性，因此使得即便是重要的字其 TF-IDF 也會被往下拉。而手動加入 stopword set 則可以直接了當地把常見無關緊要的介系詞，助動詞等通通濾掉，所以 performance 能夠有明顯的進步。

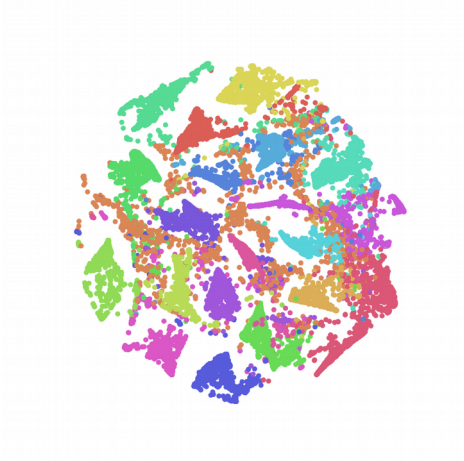
4. Try different cluster numbers and compare them.

num_cluster	20	40	60	80	85	100
F-Measure	0.64475	0.83545	0.85848	0.87532	0.87273	0.86872

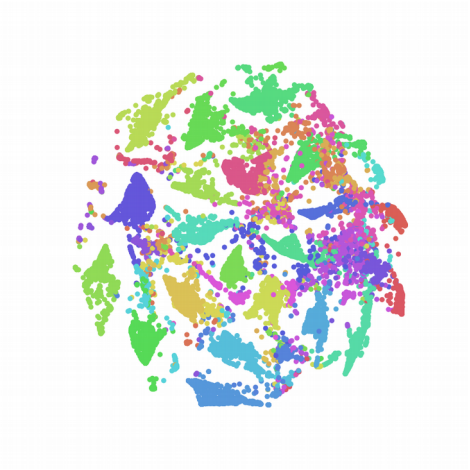
F-Measure: $P = TP/(TP+FP)$, $R=TP/(TP+FN)$, $F_\beta=(1+\beta)^2PR/(\beta^2P+R)$

根據上面圖表，我們可發現隨著 cluster 數目的增加，F-Measure 一開始會有顯著的提升，之後成長幅度趨緩，最後開始下降。我認爲這是由於增加 cluster 數目會使得原本不同 cluster 的成員被分配到相同 cluster 的機率降低，也就是 false positve 會減少，進而讓 precision

分數變高。雖然增加 cluster 數相反的會讓 false negative 增加，使得 recall 下降，然而由於 F-Measure 的計算會使 precision 比重較重，因此整體而言 F-Measure 還是會上升。



num_cluster=20



num_cluster=40

從上圖可以發現，新分出的 cluster 主要都是在原本 num_cluster=20 下分布較爲零散重疊的地方，因此增加 cluster 確實可以讓 precision 上升。而其實 num_cluster=40 仍有許多群無法被分辨，這代表著還有空間可以再增加 cluster 數使得 precision 及 F-Measure 繼續上升。