Machine Learning HW4 Report

ML2016 HW4 Unsupervised Learning

## A. Analyze the most common words in the clusters

a. TF-IDF Top 25 after filtering stop-words (title_StackOverflow.txt)

| | | | | |
|---|---|---|---|---|
| wordpress | visual | data | spring | linq |
| oracle | matlab | haskell | way | apache |
| magento | bash | drupal | hibernate | sharepoint |
| excel | mac | ajax | get | svn |
| file | scala | studio | use | qt |

b. TF-IDF Top 25 before filtering stop-words (title_StackOverflow.txt)

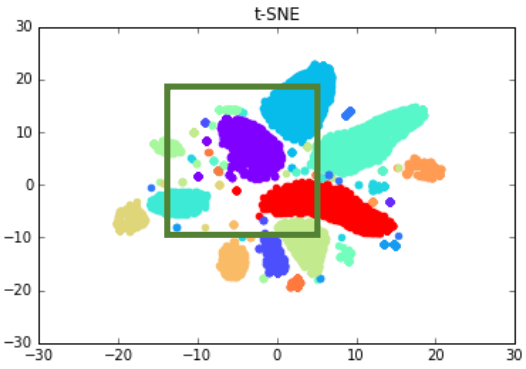
| | | | | |
|---|---|---|---|---|
| how | and | for | wordpress | with |
| to | using | spring | linq | of |
| do | the | can | magento | a |
| drupal | i | hibernate | is | in |
| excel | an | on | what | from |

c. Remark

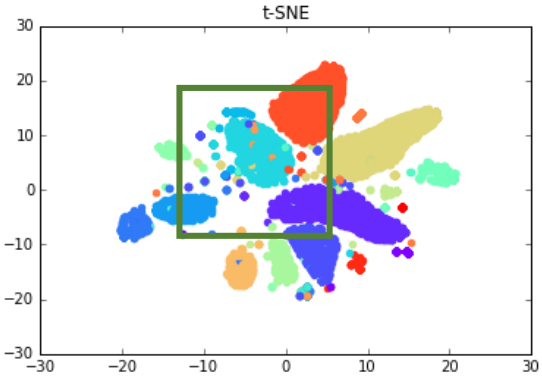
1. stop-word : from nltk.corpus (English)
2. blue word : actual tag words

## B. Visualize the data

a. My prediction




b. True Label



c. Remark

1. 由於我會先將 title 分成兩類：有/沒有 Top 20(見 A)字在句子中，針對有者(約 13000 筆)，使用 K-mean 分類，沒有者(約 7000 筆)則隨機分類。為了不影響視覺，上述圖型僅考慮前者。
2. 由於記憶體限制，上述圖型使用 t-SNE，僅畫出 13000 筆資料(有 Top 20 字者)之前 6000 筆。
3. 由兩張圖可以得知，分類效果還不錯(除了少部分如綠色框框中的錯誤)，其中原特徵向量為 20 維度

**C. Compare different feature extraction methods**

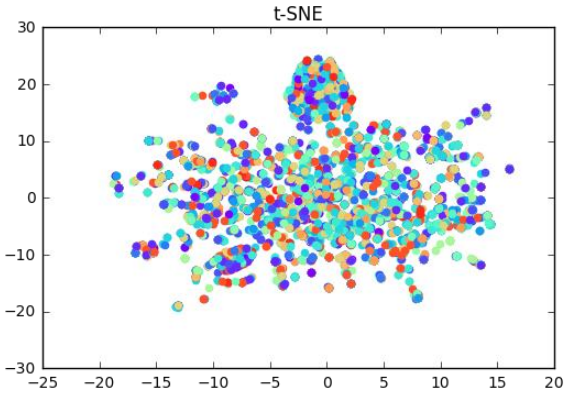
a. TF-IDF with stop-words removing and different vector size

測試過使用 20, 50, 100, 150 維的向量當作 feature，結果是 20 維最好，可能的原因可由 A 看出，經過 TF-IDF 的篩選，Top 20 的字可能涵蓋了大部分實際的 tag，使用太多維度可能會增加 noise 而降低準確度。

| 20 | 50 | 100 | 150 |
|---|---|---|---|
| 0.28559 | 0.20279 | 0.15042 | 0.05318 |

b. TF-IDF without removing stop-words

若不去除 stop-words，而其餘方法皆與 a 相同，向量將會含有過多雜訊，無法分類(如下圖所示)。




c. Bag-of-word (simply count word showing up times)

與 b 道理相似，將會把"the"等重複出現但無意義的字加入 feature，使雜訊過多。

d. Document vector

有嘗試過使用 gensim 的 doc2vec 來找出句子的向量，但可能是因為參數不會調整導致結果不好(或是讓 model 看太多無關緊要的句子)，儘管 model 似乎能夠幫忙將一些字連結在一起(如 model 可以判斷'svn'與'repository'很接近等)

**D. Try different cluster numbers and compare them. You can compare the scores and also visualize the data**

由於我的方法有 1/3 以上的答案是隨機分配，由分數來呈現有時不太準確，不過大致上可以看出 clustering 數量在 18~26 間表現大致上差不多。

Ex: 分群數與 public set scores

| 14 | 18 | 20 | 26 | 30 |
|---|---|---|---|---|
| 0.39590 | 0.48192 | 0.48595 | 0.48207 | 0.47749 |

N>=18 的主要原因大概可以從 A 分析，由於取 Top 20 當作 feature，有一些關鍵字(tag)可能被遺漏在外面(只有找到 18 個 tag)，導致某些句子沒有關鍵字(即上述 7000 筆資料)，故剩下的 13000 筆大部分應該存在於 18 個 clustering。而 N 略大於 20 有時候可能會好的原因，可能是所取的 feature 不太好，導致有一些 outliers，稍微增加目標群數有一定機率可以避開。N 太大的話，會使原本應該要同群的 data 被分開，導致結果變得不理想。