Information Retrieval and Applications

HW04-Pseudo Relevance Feedback Model

四資工四甲B10515047陳牧凡

這次的作業需要實作的演算法PRF最大的特色在於考慮query的準確性，因為在做資料檢索中，information need 和query未必是一致的，因此我們需要將query進行修正，讓其能夠符合information need。我採取的修正方法是Rocchio演算法。

$$\vec{q}' = \alpha \cdot \vec{q} + \beta \cdot \frac{1}{|R_q|} \cdot \left(\sum_{d_j \in R_q} \vec{d_j} \right)$$

首先需要取得該query的Rq(relevance document)，也就是相關性比較高的文件，那麼取得的方式我是使用Kmeans clustering，將文章和query進行分類:

```
from sklearn.cluster import KMeans
print('......calculating kmeans clustering......')
kmeans = KMeans(n_clusters=36, random_state=0).fit(totalTF)
```

分群後就可以得到每個query各自的Rq，再透過調參數取得較佳的q'解。

將query修正後再計算和document的相似度。我使用的方法是之前作業做過的BM25。

- To sum up, the BM25 model can be written as:

$$im_{BM25}(d_j, q) \equiv \sum_{w_i \in \{d_j \cap q\}} \frac{(K_1 + 1) \times tf_{i,j}}{K_1 \left[(1 - b) + b \times \frac{len(d_j)}{avg_{doclen}} \right] + tf_{i,j}} \times \frac{(K_3 + 1) \times tf_{i,q}}{K_3 + tf_{i,q}} \times log \left(\frac{N - n_i + 0.5}{n_i + 0.5} \right.$$

Tunable Parameters
Document Length Normalization
Term Frequency
Inverse Document Frequency

在這之前我也有嘗試過Tri-mixture model，會使用到HW3的EM演算法，不過當初在做的時候就沒有寫得很好，因此在這次的作業中效果也不是很優秀，所以就沒有繼續採用了。

參考資料:

The 2014 Conference on Computational Linguistics and Speech Processing ROCLING 2014, pp. 3-20 © The Association for Computational Linguistics and Chinese Language Processing