R語言與資料科學導論

書面報告

主旨：你的同溫層，暖嗎？

以Cosine Similarity探討Dcard、PTT 兩大社群平台討論內容之一致性

第七組 @匿名yunchiapig
許芸嘉 陳柄瑞 陳韋傑 陳信德

目錄

一、動機

迴聲室效應(Echo Chamber Effect)，形容在封閉的環境中，相似意見的聲音不斷重複，也就是所謂的「同溫層效應」。在現今網際網路越來越發達的社會，社群網站的使用越來越頻繁，再加上個人化推薦的技術被頻繁應用於使用者資料，用戶所接觸到的資料也更受到侷限，加深了同溫層效應。

因此我們想要透過課堂上學到的Cosine Similarity方法，來比較在兩大社群平台PTT與Dcard各個板上之同溫層厚度有多厚，因為研究限制，我們只各選擇了四個板來比較，分別是八卦板、美妝板、女孩板、台大板。

二、研究問題

平台	PTT	Dcard
看板名稱	八卦 (Gossiping)	時事 (trending)
	女人 (WomenTalk)	女孩 (girl)
	美容 (MakeUp)	美妝 (makeup)
	臺大 (NTU)	臺灣大學 (ntu)

我們將PTT及Dcard中，性質類似的看板放在一起比較 ：

1. 比較PTT八卦板 (Gossiping) 與Dcard時事板 (trending)，在同一時間區段內，看板文章之Cosine Similarity。

2. 比較PTT女人板 (WomenTalk) 與Dcard女孩板 (girl)，在同一時間區段內，看板文章之Cosine Similarity。

3. 比較PTT美容板 (MakeUp) 與Dcard美妝板 (makeup)，在同一時間區段內，看板文章之Cosine Similarity。

4. 比較PTT臺大板 (NTU) 與Dcard臺灣大學板 (ntu)，在同一時間區段內，看板文章之Cosine Similarity。

5. 比較PTT / Dcard的八卦板 (Gossiping) / 時事板 (trending)、女人板 (WomenTalk) / 女孩板 (girl)、美容板 (MakeUp) / 美妝板 (makeup)、臺大板 (NTU) / 臺灣大學板 (ntu)，不同看板文章之Cosine Similarity。

三、假設

對於研究的結果，我們先做了一些假設：

1. PTT八卦板 (Gossiping) 與Dcard時事板 (trending)：
   Dcard因為使用者多為大學生，年齡層較接近，推論學生族群討論之內容會較類似，因此一致性會較PTT高。

2. PTT女人板 (WomenTalk) 與Dcard女孩板 (girl)：
   女孩們討論的話題應該本身就較類似，因此推論整體的一致性會較高。至於Dcard跟PTT的差異應該不大，Dcard可能因年齡相近，一致性稍較PTT高一些。

3. PTT美容板 (MakeUp) 與Dcard美妝板 (makeup)：
   因網路上討論美妝的用詞、內容都大同小異，因此推論整題的一致性也會較高。而美妝板在Dcard的討論度大於PTT，因此推論Dcard一致性會較高。

4. PTT臺大板 (NTU) 與Dcard臺灣大學板 (ntu)：
   臺大板類似於圍繞著臺大話題的八卦板，同溫層更厚，推測一致性會很高。且Dcard討論度較高，推測一致性也會較PTT高。

我們認為，Dcard之使用者年齡層較接近，相較使用者年齡層分布很廣的PTT，Dcard的一致性會更高。

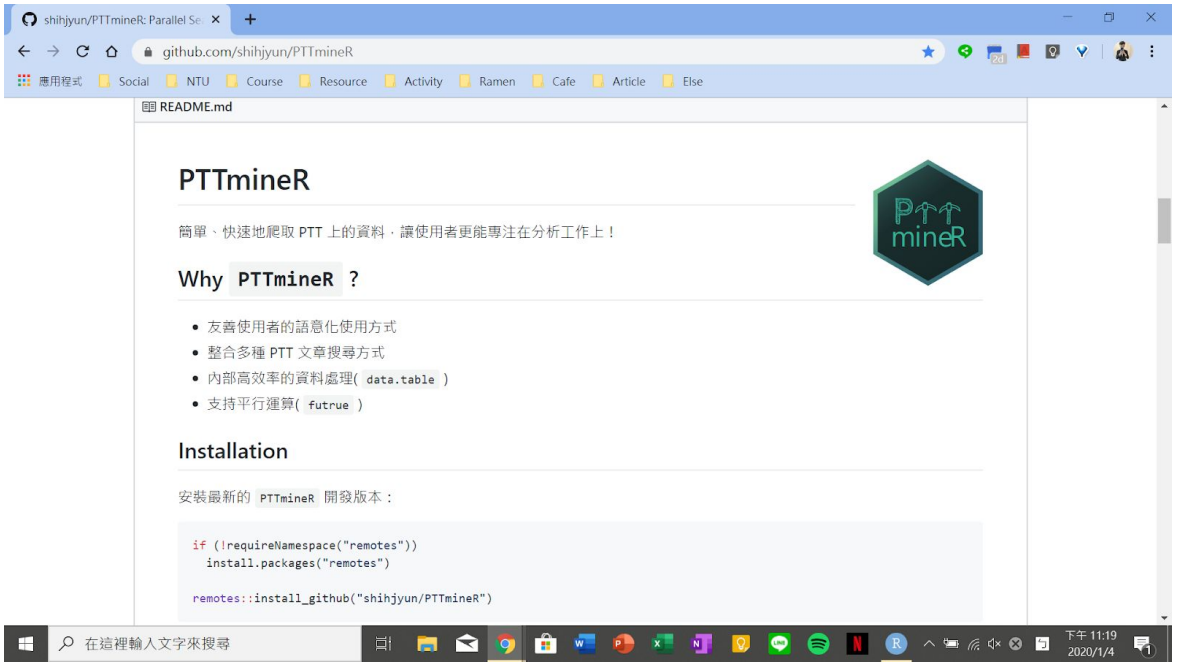
5. 跨看板的比較：
   我們推測美容/美妝板會最高，因為美妝的主題明確、用詞較特定，而依序遞減為臺大/臺灣大學板、女人/女孩板、八卦/時事板，其中八卦/時事板因討論內容較多較複雜，因此我們認為一致性會最低。

一致性	PTT < Dcard
最低	八卦(Gossiping) < 時事(trending)
次低	女人(WomenTalk) < 女孩(girl)
最高	美容(MakeUp) < 美妝(makeup)
次高	臺大(NTU) < 臺灣大學(ntu)

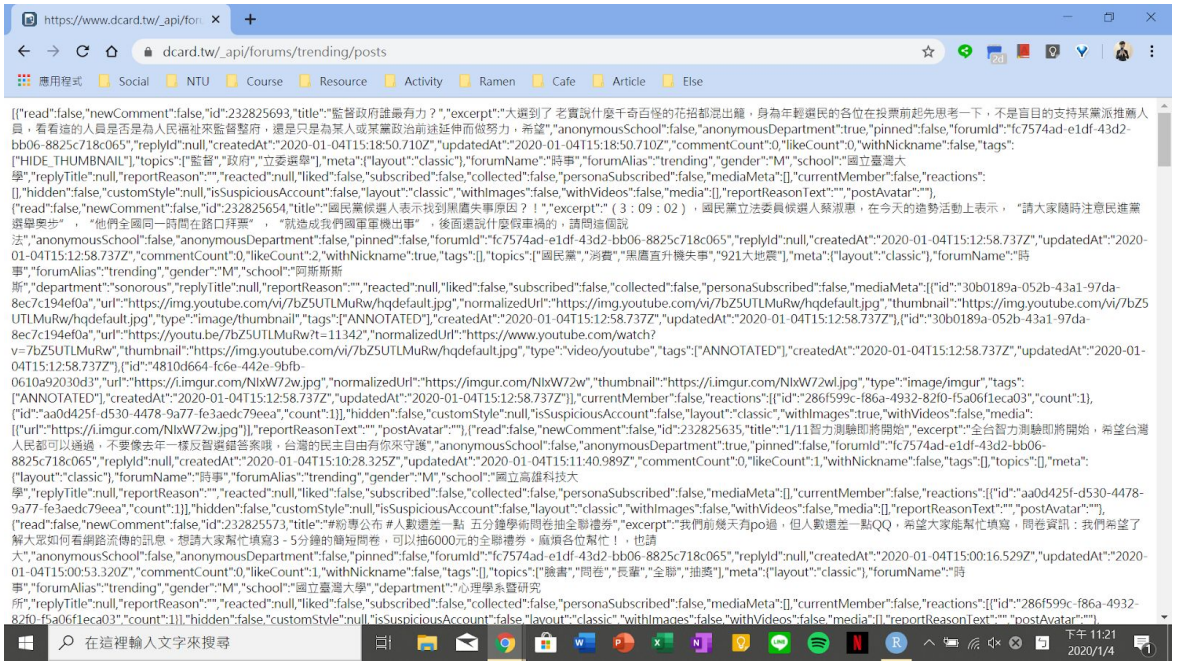
四、研究方法

(1)資料取得

在資料取得方面，PTT我們直接用現成的套件PTTmineR幫我們做資料的爬取；Dcard我們使用R的httr套件爬取Dcard的公開api。



公開套件PTTmineR



Dcard的公開api

```{r}
# PTT  八卦版爬到之前Dcard所爬到186篇文最早的時間（即1/3 確保資料同為相同時段）
mine_ptt(ptt.miner = rookie_miner2,
         board = "Gossiping",
         min.date = "2020-01-03")
```

```
PTTmineR mining from ptt on your setting ...Item 2 has 0 rows but longest item has 1; filled with
NAItem 3 has 0 rows but longest item has 1; filled with NAItem 4 has 0 rows but longest item has 1;
filled with NAItem 5 has 0 rows but longest item has 1; filled with NA條件的長度 > 1，因此只能用其第一元素t
longest item has 1; filled with NAItem 3 has 0 rows but longest item has 1; filled with NAItem 4 has 0
rows but longest item has 1; filled with NAItem 5 has 0 rows but longest item has 1; filled with NA
```

使用PTTmineR爬取八卦版文章

```{r}
# 爬1000篇最新文章，將1000篇文章之id存入向量"idxs"
library(stringr)
library(httr)
idxs <- vector(mode = "character",length = 200)

req <- GET('https://www.dcard.tw/',
           path = c("_api","forums","trending","posts?popular=false&limit=100"))
idx <- content(req)

for (i in 1:100){
  idxs[i] <- idx[[i]]$id
}
id_end <- idxs[100]

i <- 101
for(loop in 2:2)
{
  str <- paste0("https://www.dcard.tw/_api/forums/trending/posts?popular=false&limit=100&before=", id_end)
  req <- GET(str)
  idx <- content(req)
  for(count in 1:100)
  {
  idxs[i] <- idx[[count]]$id
  id_end <- idxs[i]
  i <- i + 1
  }
}

```

使用httr套件爬取Dcard api
(因api限制一次只能爬100篇，因此我們運用迴圈爬取)

(2)抽樣

一開始，我們原本希望可以爬到整個十二月的文章，一方面文章篇數較多較具可信度，一方面因為時間區段固定，便不會因為兩版時間區段不同造成討論主題不同，進而影響到相似度。

但我們很快就遇到了問題，由於每個版的屬性以及討論的東西本身就不同，因此每個版的發文頻率也不同。舉例來說，PTT八卦版非常熱門，一天就可以發到一千篇；相較之下，Dcard的時事版一天大概只有一百篇文章，且受限於電腦設備，我們也

無法將PTT八卦版所有十二月的文章都爬下來。在思考和不斷測試過後，我們決定這樣解決每個版文章數不同的問題：

看板	八卦/時事		女人/女孩		美容/美妝		臺大/臺灣大學	
平台	PTT	Dcard	PTT	Dcard	PTT	Dcard	PTT	Dcard
時間	1/3 ~ 1/4		12/7 ~ 1/3		12/19 ~ 1/4		9/24~1/3	
原始篇數	2201	186	3410	1000	211	1000	601	1000
抽樣篇數	186	186	500	500	211	211	500	500

受限於電腦設備，在最熱門的PTT八卦版我們只爬了兩天的文章，總共有2201篇，並在Dcard上爬了相同時間區段的文章，共186篇。其他三個領域我們都是將Dcard固定在一千篇，再以Dcard的時間區段去PTT爬對應時間區段的文章，不過在PTT美容版則是遇到文章過少的問題(只有211篇)。

最後再將所爬到的文章進行隨機抽樣，我們認為透過隨機抽樣我們可以看到較多元的文章，也較不會有某幾天熱門事件被大肆討論造成相似度過高的問題。

(3)斷詞

利用R的jiebaR套件對文章內容進行斷詞。

根據斷詞結果，我們挑出未被jiebaR正確斷開的詞語(網紅、蔡佩軒、阿滴英文)，建立語料庫，再重新進行斷詞。

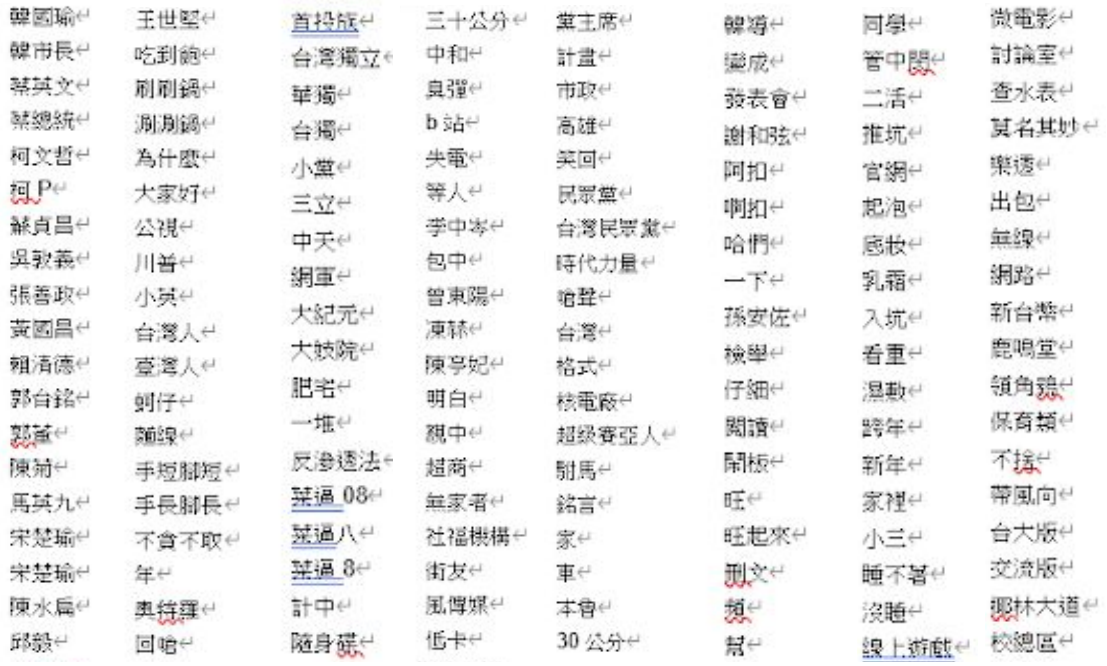
content
韓粉 又 爆 盜用 ！「 破 億 網紅 蔡佩軒 」影片 整支 被 搬...
Youtuber「 阿滴英文 」 日前 號召 28 名 包括 千千 、 博 恩...
此 篇文章 為 轉貼 文章 ， 請 更新 至 最新 版本 觀看 完整 ...
https : / / youtu . be / aZ5znkF _ Umw 阿滴英文 提醒 大家 ...

針對每個版，我們都透過閱讀前20篇文章的斷詞結果，持續修正語料庫。最後再使用最新的語料庫進行斷詞。我們也發現每修正一篇內文，斷詞錯誤的發生次數急遽下降，到後來已經幾乎沒有錯誤。

```{r}
# 對爬文結果斷詞
# ptt_tb[i]會出現第i篇文章的斷詞結果,文章被空白鍵斷開
library(jiebaR)
library(quanteda)

seg <- worker(symbol = T, bylines = F, user = "words.txt")

for (i in 1:length(dcard_tb$content)){
  dcard_tb$content[i] <-
  dcard_tb$content[i] %>% segment(seg) %>% paste(collapse = " ")
}
```

```
Loading required package: jiebaRD
Package version: 1.5.2
Parallel computing: 2 of 4 threads used.
See https://quanteda.io for tutorials and examples.

Attaching package: 'quanteda'

The following object is masked from 'package:utils':

    View
```

先使用jiebaR進行斷詞，透過持續修正語料庫(words.txt)來達到精準的斷詞



語料庫(部分)

```r
# 將斷出的詞存入變數"dcard_toks"
# dcard_toks[i]會出現第i篇文章被斷出來的所有詞彙
dcard_corp <- corpus(dcard_tb, text_field = "content")
dcard_toks <- tokens(dcard_corp, what = "fasterword" , remove_punct = TRUE,
                     remove_numbers = T, remove_url = T)
```

使用quenteda的corpus跟token來處理

其中token這個函數提供了刪除標點符號、url跟數字的功能，對於分析相似度提供了不少幫助(因標點、url多無意義，數字也不是我們主要分析相似度的對象)。而在檢查時我們也注意到，token會再把我們切好的詞語再切一次，我們也苦惱了一陣子，後來發現將他的what參數設為fasterword，便能解決這個問題。

(4)將斷詞取聯集

```r
# 將每篇文章的詞彙取聯集，存入變數"dcard_all"
dcard_all <- vector("character",100000)
for (i in 1:186){
  dcard_all <- union(dcard_all,dcard_toks[[i]])
}
```

```r
# 查看斷詞聯集的大小
length(dcard_all)
```

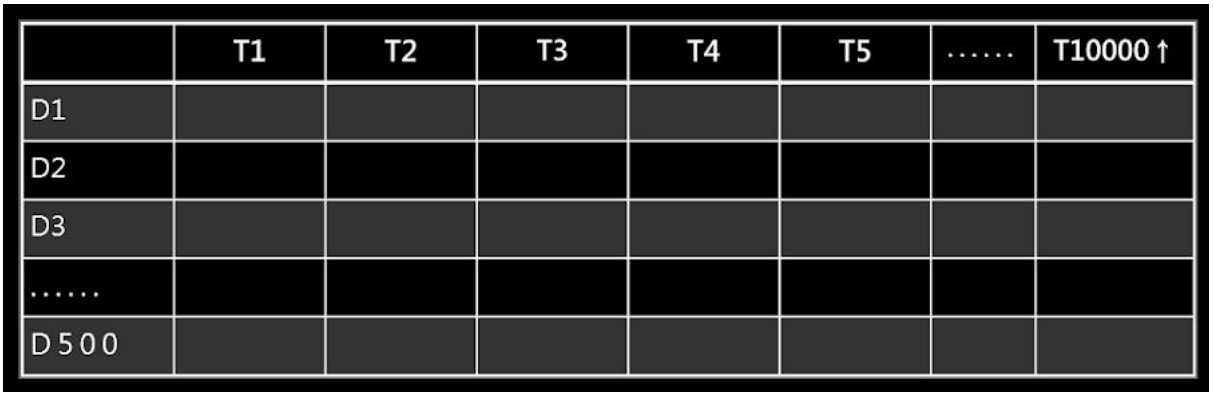
```
[1] 9739
```

使用內建的union函數將該版上所有文章的斷詞取聯集存入dcard_all。

(5)建立DTM

```r
# 製作出dtm
dcard_dtm <- matrix(nrow = 186, ncol = length(dcard_all))

for (i in 1:186){
  dcard_num  <- vector("numeric",length(dcard_all))
  temp = dcard_toks[[i]]
  for (j in 1:length(dcard_all))
  {
     for(k in 1:length(temp))
     {
       if(temp[k] == dcard_all[j]){
         dcard_num[j] <- dcard_num[j] + 1
       }
     }
  }
  dcard_num <- dcard_num/sum(dcard_num)
  dcard_dtm[i,] <- dcard_num
}
```

建立DTM



	T1	T2	T3	T4	T5	.....	T10000↑
D1							
D2							
D3							
.....							
D500							

DTM示意圖，其中每列代表一篇文章、每欄代表一個詞

(6)計算cosine similarity

```r
# 自訂Cosine Similarity之函數
cos_sim <- function(x,y)
{
  result <- x %*% y /(sqrt(x %*% x)*(sqrt(y %*% y)))
  return(result[1])
}
```

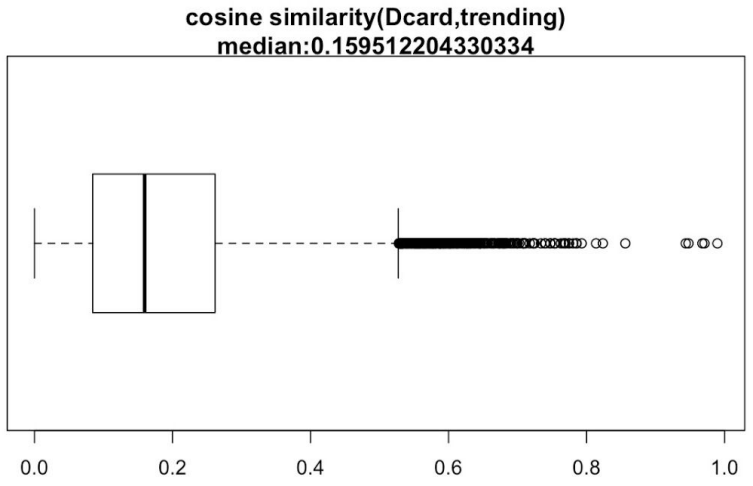
定義cosine similarity 函數 = $\frac{xy}{\sqrt{x^2}\sqrt{y^2}}$

```r
# 任兩篇文章計算其Cosine Similarity
dcard_n <- 186
dcard_cosine_similarity <- vector("numeric", dcard_n*(dcard_n-1)/2 )
dcard_index <- 1

for(i in 1:(dcard_n-1)){
  for(j in (i+1):dcard_n){
    dcard_cosine_similarity[dcard_index] <- cos_sim(dcard_dtm[i,],dcard_dtm[j,])
    dcard_index <- dcard_index + 1
  }
}
```

將板上所有文章每取兩篇計算一次cosine similarity(共 C_2^n 次)，並將結果存入dcard_cosine_similarity矩陣。

(7)作圖

```r
# 對所有的Cosine Similarity作圖
title <- paste0("cosine similarity(Dcard,trending)\nmedian:",median(dcard_cosine_similarity))
dcard_outlier_values <- boxplot.stats(dcard_cosine_similarity)$out
boxplot(dcard_cosine_similarity, main= title,
        horizontal=TRUE)

```



對所有計算出的cosine similarity繪製盒狀圖，使用boxplot函數

選用盒狀圖的原因，是因為我們最後要看的值是中位數，因為平均數會受極端值的影響，故我們選用中位數，較能反映我們的研究結果。

```r
# 看outlier的數量
length(dcard_outlier_values)

```

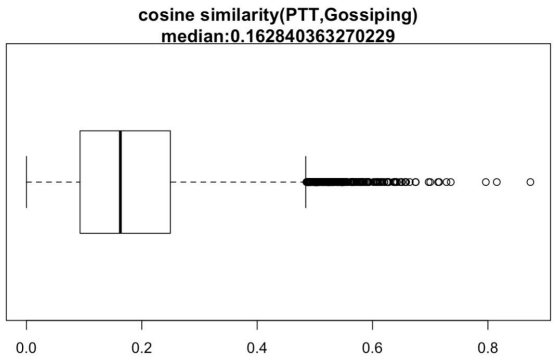
```
[1] 415
```

同時我們也算出outliner的數量(上圖裡點的數量)

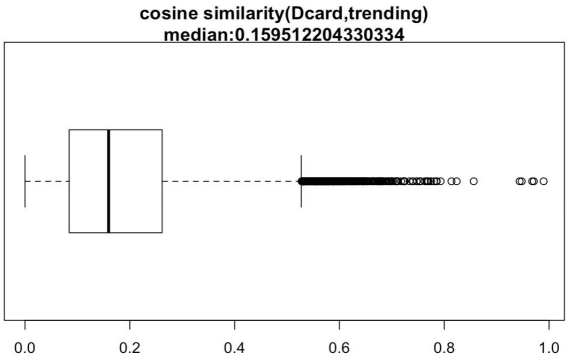
五、研究結果

1. PTT八卦板 (Gossiping) 與Dcard時事板 (trending)

PTT-Gossiping：

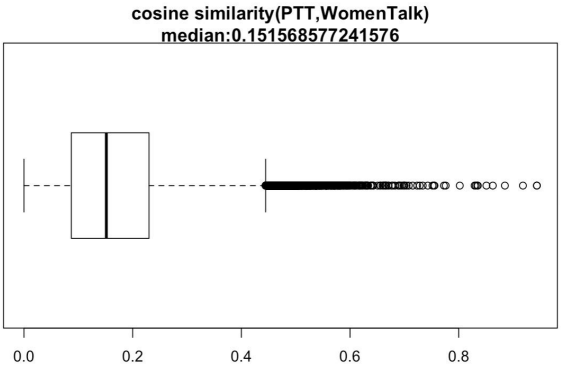


Dcard- Trending：

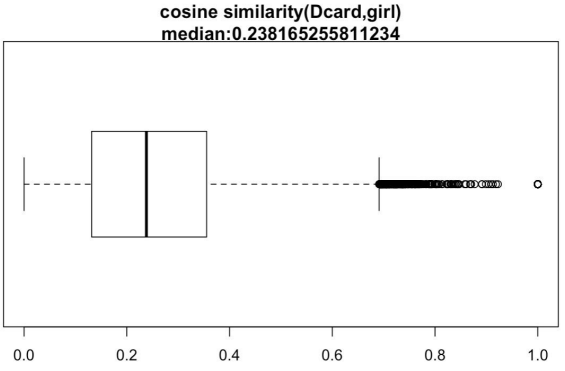


2. PTT女人板 (WomenTalk) 與Dcard女孩板 (girl)

PTT-WomenTalk：

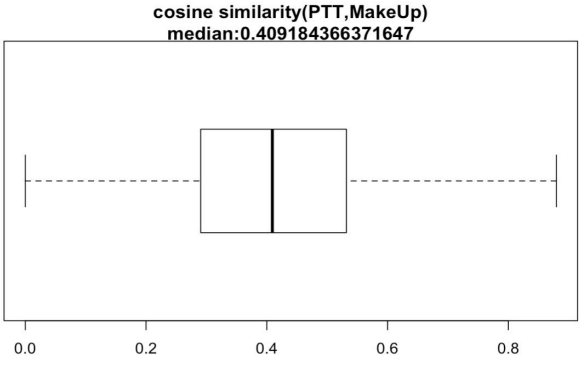


Dcard-Girl：

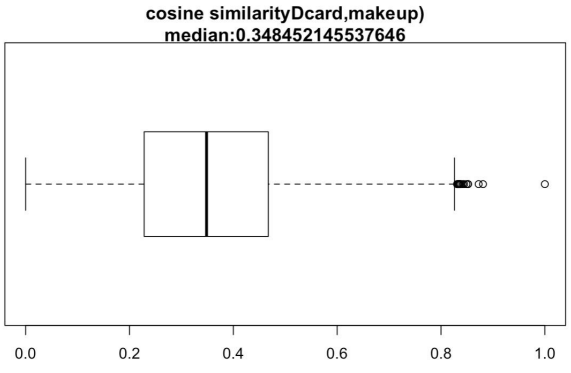


3. PTT美容板 (MakeUp) 與Dcard美妝板 (makeup)

PTT-Makeup：

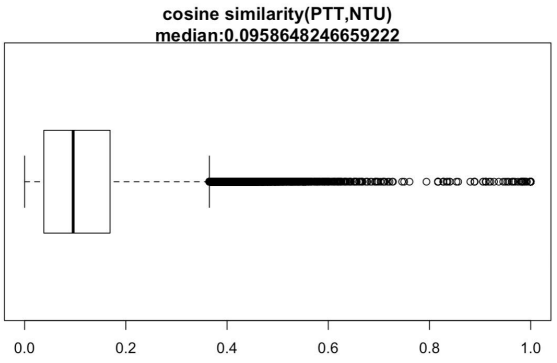


Dcard-Makeup：

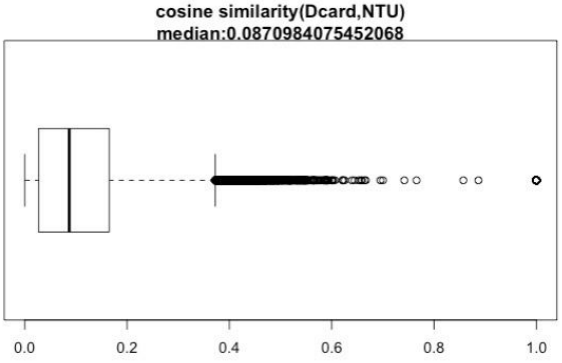


4. PTT臺大板 (NTU) 與Dcard臺灣大學板 (ntu)

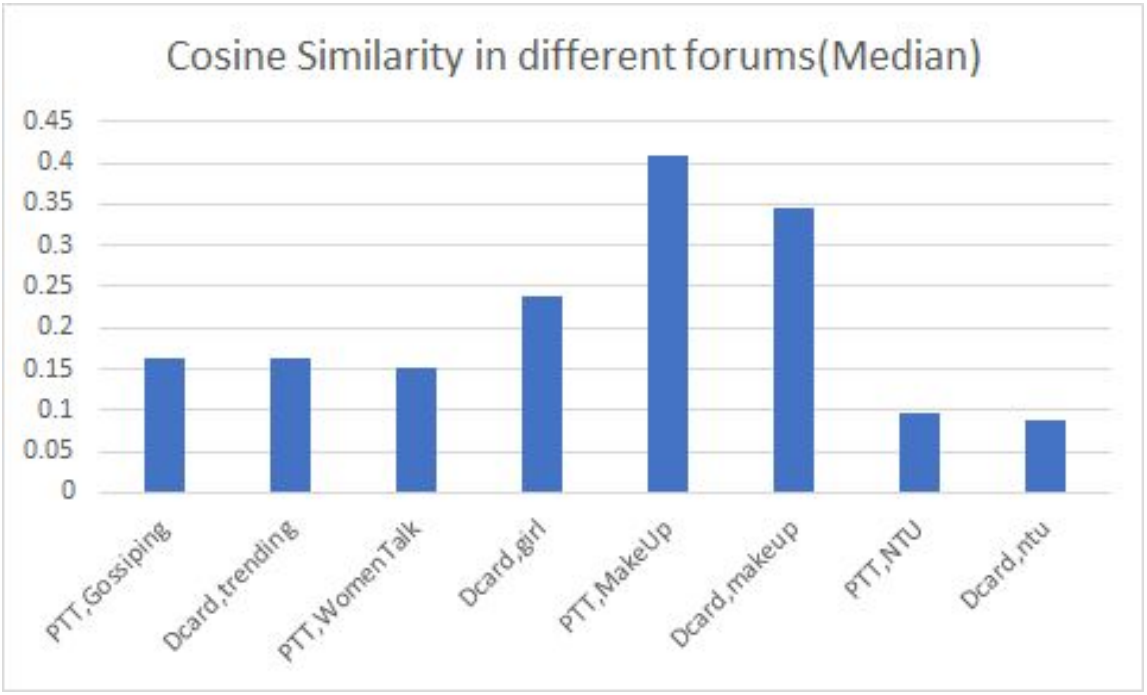
PTT-NTU：



Dcard-NTU：



看板	八卦/時事		女人/女孩		美容/美妝		臺大/臺灣大學	
平台	PTT	Dcard	PTT	Dcard	PTT	Dcard	PTT	Dcard
時間	1/3 ~ 1/4		12/7 ~ 1/3		12/19 ~ 1/4		9/24~1/3	
斷詞數	8435	9739	10423	13011	11592	11631	16157	7610
Cosine Similarity median	0.16284	0.16382	0.15157	0.23817	0.40881	0.34630	0.09586	0.08710



六、結論

1. PTT八卦板 (Gossiping) 與Dcard時事板 (trending)：

    八卦、時事板由關注多元議題的群眾組成，由結果可以看到兩者的cosine similarity中位數分別是0.16284和0.16382，相當接近，因此可以說兩看板討論內容的相異度差不多。

    與假設(Dcard的同溫層會較厚)不符，我們推測可能的原因是兩個看板都著重討論多元的議題，因此使用者也較多元，沒有其中一方同溫層較厚的現象。

2. PTT女人板 (WomenTalk) 與Dcard女孩板 (girl)：

    PTT女人板與Dcard女孩板的cosine similarity中位數分別是0.15157和0.23817，可以發現Dcard的同溫層明顯比較厚，與假設相符。

3. PTT美容板 (MakeUp) 與Dcard美妝板 (makeup)：

    PTT美容板與Dcard美妝板的cosine similarity中位數分別是0.40881和0.34630，可以發現PTT討論的內容一致性較高。

    此結果與假設相反。根據我們身為使用者的經驗推測，Dcard美妝版相較PTT美容板普遍發文內容較詳細，篇幅也相對比較長，所以每篇文使用到的詞語較多，在斷詞量相當的情形下，一致性會較低。

4. PTT臺大板 (NTU) 與Dcard臺灣大學板 (ntu)：

    PTT臺大板與Dcard臺灣大學板的cosine similarity中位數分別是0.09586和0.08710，可以發現兩者的內文一致性差不多低，而PTT略高於Dcard。

    此結果與與假設(Dcard一致性較高)不符。我們認為因為兩者的cosine similarity都太低，因此兩者的比較我們不做討論。而我們認為造成cosine similarity低的原因有：每日發文篇數太少且篇幅短、台大學生多使用臉書ntu交流板、台大板缺乏明確的主題定位，使得使用者不會直接聯想到台大板，再加上台大學生常見面，其實可以直接分享訊息。此外，PTT臺大板區因有眾多子板(各院皆有自己的子板、NTUtalk下分表特、黑特、帥哥板等等)，推測這也是

造成發文數量稀少的原因之一(大家都不會到NTU板上，而會到與文章主題有關的板，像是管院、表特.....)。

5. 不同看板間之一致性比較：

研究結果可知cosine similarity中位數： 美妝>女孩>八卦>台大板，與假設(美妝>台大>女孩>八卦)不符。其中台大板與我們預期相差最多，推測是由第四點中提到的原因造成發文數量過少以及討論度過低，主題分散的情況下自然相似度就不會很高。

七、研究限制

在做這份報告的時候，我們遇到了一些問題：

1. 網址、英文、表情符號的使用
2. PTT「引述」功能
3. 錯字
4. 特殊用語(鄉民用語)
5. 無意義用語(你我他、然後、總之...)
6. 不符看板主題的文章
7. 資料量過大、設備效能不足
8. Dcard API限制大量爬蟲
9. 文章內容在圖片上

這些問題目前我們仍沒有辦法解決，且這些問題通常都會對相似度分析帶來影響(網址為沒有意義的亂碼、引述會高相似度、不符看板主題的文章不應被納入考量)，若這些問題能夠妥善的處理的話，相信我們的研究會更加的具有可信度。

八、未來展望

至於我們的研究還能如何延伸呢？目前我們想到的是，可以做「文章分類器」。也就是說，假設今天有人寫了一篇文章，我們可以透過計算相似度的方式，尋找與該文章最相符的看板，並將其看板推薦給使用者。

如此一來，能減少文章內容不符該板主題的問題。此外，對於使用量較低的板(例如：臺大板)，透過推薦能夠導引部分文章至該板，使得使用者較少的板不會越來越「邊緣化」；而若推薦器也很少將文章推薦至該板，則可能代表該板因為主題不明確等問題，也可以考慮直接廢除。

九、參考資料

1. PTTmineR https://github.com/shihjyun/PTTmineR
2. Dcard爬蟲簡介 https://levirve.github.io/blog/2016/Dccard-crawler/