男女歌手用詞差異分析

第四組

古貿昌、余興祐、施文捷、許家誠

2020-01-12

R 語言與資料科學導論

目錄

1. 簡介

歌詞是語言中非常有趣的一部份，情感表述的方式比起日常用語來說更為豐富多變，我們對於用資料科學處理歌詞會有甚麼樣的結果感到好奇，決定了主題要與歌詞相關。

社會的壓力或是期望，歌詞都會反映出來，不管內容是順從還是反抗社會。雖然歌詞的表現形式很容易讓意思模糊不清，或是充滿晦澀的象徵，有可能正是這種沒有清楚界線的文字更能表現出平常隱藏在表象下的真實。我們認為性別是一個分類歌手簡單直觀的標籤，也蠻可能有明顯的表現差異，所以最後決定蒐集男女歌詞並做出分析。

2. 方法

2.1. 資料取得

稍微修改了網路上的的爬蟲 code，用 python 從魔鏡歌詞網上爬取華語熱門排行男歌手和女歌手中前 80 名歌手的所有歌詞，最後取得了男女各約 14000 首歌的歌詞。

2.2. 前處理

爬取資料之後，因魔鏡歌詞網的格式有點混亂，還包含許多意義不明的空白頁，讓我們在整理資料時遇到了一些麻煩。透過正規表示式的搜尋方法，將

特定的且不必要的內文資料移除，並手動刪除了檔案大小為 0kB 的檔案，以便進行接下來斷詞的步驟。

斷詞的部份我們使用 jeibaR 進行。其中停用詞表是讓我們感到困擾的因素之一。若使用一般的停用詞表，會刪掉代名詞和連接詞等比較沒有具體意義的詞彙；然而一開始我們認為男女用詞上的差距可能就隱藏在這些詞中。我們先試了比較簡單的停詞表，但是結果難以看出所以然，最後決定修改了網路上比較齊全的停詞表自製了一份。計算斷完詞後的資料後，用結果製成詞頻表並比較男女歌手之間的分別。

2.3. 原始碼運作說明

```
import requests
from bs4 import BeautifulSoup
import re
import os
from random import random
import time

#魔鏡歌詞網熱門華語男歌手的 url
url = 'https://mojim.com/twza1.htm'

#取得根目錄
curDir = os.getcwd()

#取得所有男歌手的頁面
resp = requests.get(url)
soup = BeautifulSoup(resp.text, 'html.parser')
block = soup.find(class_="s_listA").find_all('a')

#創建資料夾
if not os.path.exists('超過 100 首'):
```

```
        os.mkdir('超過 100 首')
        print("Directory top100 Created ")
    else:
        print("Directory top100 already exists")
    curDir += "/超過 100 首"
    os.chdir(curDir)
    if not os.path.exists('男'):
        os.mkdir('男')
        print("Directory  男  Created ")
    else:
        print("Directory  男  already exists")
    if not os.path.exists('女'):
        os.mkdir('女')
        print("Directory  女  Created ")
    else:
        print("Directory  女  already exists")
    curDir += "/男"
    os.chdir(curDir)

    #將歌詞寫入檔案
    def writeFile(i):
        #參數 i: bs4 中的<a> tag 物件
        #取得歌詞內文
        j = i.get('href')
        sub_resp = requests.get("https://mojim.com" + j)
        sub_soup = BeautifulSoup(sub_resp.text, 'html.parser')
        sub_name = sub_soup.find('dd','fsZx3')
        title = i.get('title').split('  歌')[0]
        if '/' in title:
            title=title.replace("/", "")
        path = title+ ".txt"
        f = open(path, 'w')
        if(sub_name!=None):

            s=str(sub_name)
            start = re.search("<br/><br/>",s).start()
            s=s[start:]
            sub_name = BeautifulSoup(s, 'html.parser')
```

```
            #文本處理，用正規表示刪除一些不必要的字彙
            if re.search("\[(([a-z]|\d|\:|\.)+|([a-z]+:\S{1,4}))\]",
            sub_name.text)!=None:
                end = re.search("\[(([a-z]|\d|\:|\.)+|([a-z]+:\S{1,4}))\]",
            sub_name.text).start()
                write = sub_name.text[:end]
                write = write.replace('更多更詳盡歌詞 在 ※ Mojim.com   魔鏡歌詞網',"")
                write = re.sub(r"感謝.{5,10}(修正|提供)歌詞", "", write)
                f.write(write)
            else:
                write = sub_name.text.replace('更多更詳盡歌詞 在 ※ Mojim.com   \
            魔鏡歌詞網',"")
                write = re.sub(r"感謝.{5,20}(修正|提供)歌詞", "", write)
                f.write(write)


for i in block:
    j = i.get('href')
    resp = requests.get("https://mojim.com" + j)
    #歌手網址
    soup = BeautifulSoup(resp.text, 'html.parser')
    block3= soup.find_all('dd', 'hb2')
    block4 = soup.find_all('dd','hb3')

    for i in block3:
        song_left = i.find_all('span','hc3')
        for j in song_left:
            sl = j.find_all('a')
            for k in sl:
                writeFile(k)
        song_right = i.find_all('span','hc4')
        for j in song_right:
            sr = j.find_all('a')
            for k in sr:
                writeFile(k)


    for i in block4:
```

```
            song_left = i.find_all('span','hc3')
            for j in song_left:
                sl = j.find_all('a')
                for k in sl:
                    writeFile(k)
            song_right = i.find_all('span','hc4')
            for j in song_right:
                sr = j.find_all('a')
                for k in sr:
                    writeFile(k)

        #random delay  避免被魔境網拒絕訪問
        delay=round(random()*100)
        print(delay)
        time.sleep(delay)
```

```
library(jiebaR)
library(stringr)

# Get txt file paths
fps_m <- list.files('male', full.names = T)
fps_f <- list.files('female', full.names = T)

# Initialize jiebaR
seg <- worker()

# determining song names
name_m <- vector('character', length(fps_m))

for (i in seq_along(fps_m)) {
  name_m[i] <- basename(fps_m[i])
}

name_f <- vector('character', length(fps_f))

for (i in seq_along(fps_f)) {
```

```
  name_f[i] <- basename(fps_f[i])
}

# determining lyrics
lyric_m <- vector('character', length(fps_m))

for (i in seq_along(fps_m)) {
  l_m <- readLines(fps_m[i], encoding = 'UTF-8') %>% str_squish()
  segged_m <- segment(l_m, seg)
  lyric_m[i] <- paste0(segged_m, collapse = ' ')
}

lyric_f <- vector('character', length(fps_f))

for (i in seq_along(fps_f)) {
  l_f <- readLines(fps_f[i], encoding = 'UTF-8') %>% str_squish()
  segged_f <- segment(l_f, seg)
  lyric_f[i] <- paste0(segged_f, collapse = ' ')
}

# Combine results into dfs
male_df <- tibble::tibble(編號 = seq_along(fps_m), 性別 = '男', 歌名 = name_m,
歌詞 = lyric_m)
female_df <- tibble::tibble(編號 = seq_along(fps_f), 性別 = '女', 歌名 = name_f,
歌詞 = lyric_f)
```

```
library(tidytext)
library(dplyr)
library(readxl)

stopwords <- read_excel('ch_stop_words.xlsx')

tidy_male <- male_df %>%
  unnest_tokens(output = '用詞', input = '歌詞', token = 'regex', pattern = '
') %>%
  anti_join(get_stopwords(), by = c('用詞' = 'word')) %>%
  anti_join(stopwords, by = c('用詞' = '停用詞'))
```

```
freq_male <- tidy_male %>%
  group_by(用詞) %>%
  summarize(個數 = n()) %>%
  arrange(desc(個數)) %>%
  print()
```

```
    tidy_female <- female_df %>%
      unnest_tokens(output = '用詞', input = '歌詞', token = 'regex', pattern =
' ') %>%
      anti_join(get_stopwords(), by = c('用詞' = 'word')) %>%
      anti_join(stopwords, by = c('用詞' = '停用詞'))

    freq_female <- tidy_female %>%
      group_by(用詞) %>%
      summarize(個數 = n()) %>%
      arrange(desc(個數)) %>%
      print()
```

```
    library(ggplot2)

    freq_male %>%
      top_n(20, 個數) %>%
      ggplot() +
        geom_bar(aes(reorder(用詞, +個數), 個數), stat = 'identity') +
        coord_flip() +
        labs(title = '男歌手高頻用詞')
```

```
    freq_female %>%
      top_n(20, 個數) %>%
      ggplot() +
        geom_bar(aes(reorder(用詞, +個數), 個數), stat = 'identity') +
        coord_flip() +
        labs(title = '女歌手高頻用詞')
```

3. 結果

3.1. 數據

下列是使用在 jieba 套件的 GitHub 中隨附的停用詞表所產生出的分析結果：

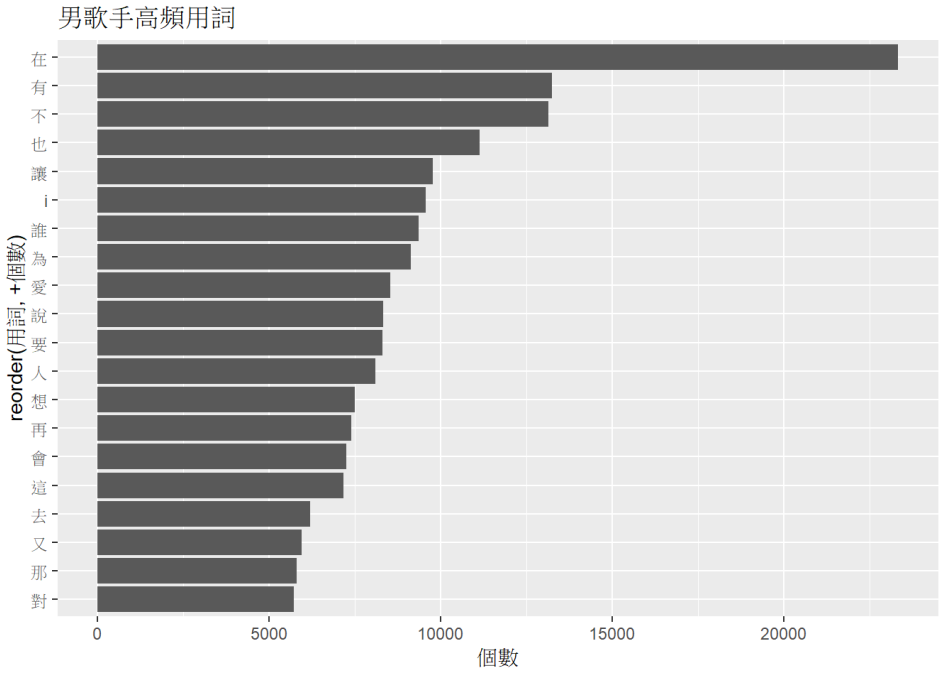

圖 1：男歌手高頻用詞長條圖（前 20 名） - 簡易停用詞版

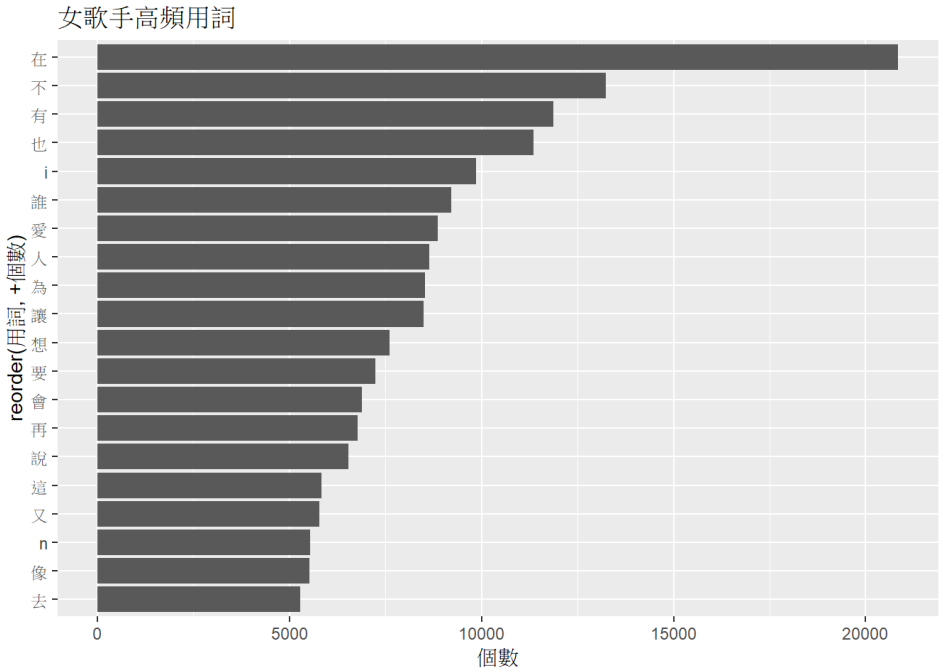

圖 2：女歌手高頻用詞長條圖（前 20 名） - 簡易停用詞版

圖 3：男歌手用詞文字雲 - 簡易停用詞版

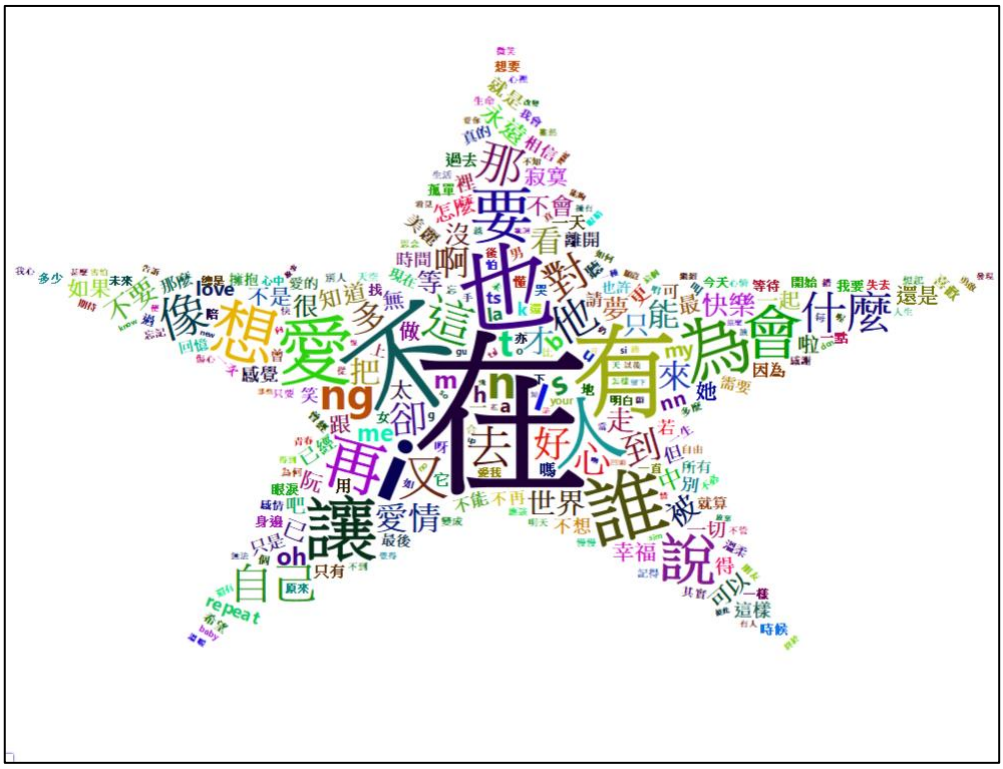

圖 4：女歌手用詞文字雲 - 簡易停用詞版

由長條圖及文字雲中可以發現，jieba 隨附的簡易版停用詞表在歌詞中的表現，很難看出這兩組資料的高頻用詞中有什麼樣的區別。於是我們使用了另

外一份修訂過的停用詞表，結果如下：

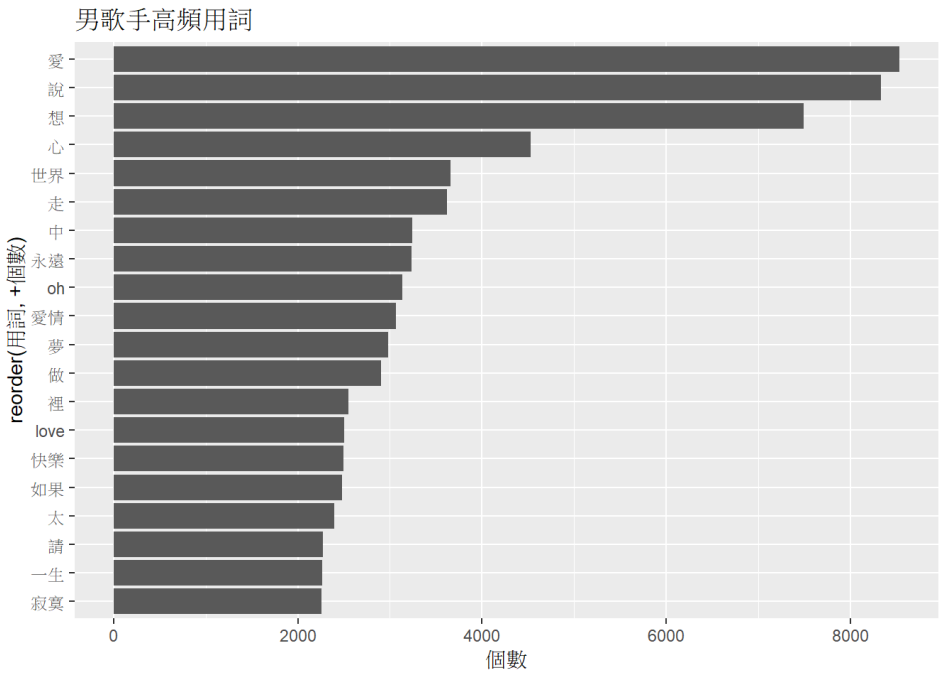

圖 5：男歌手高頻用詞長條圖（前 20 名） - 修訂停用詞版

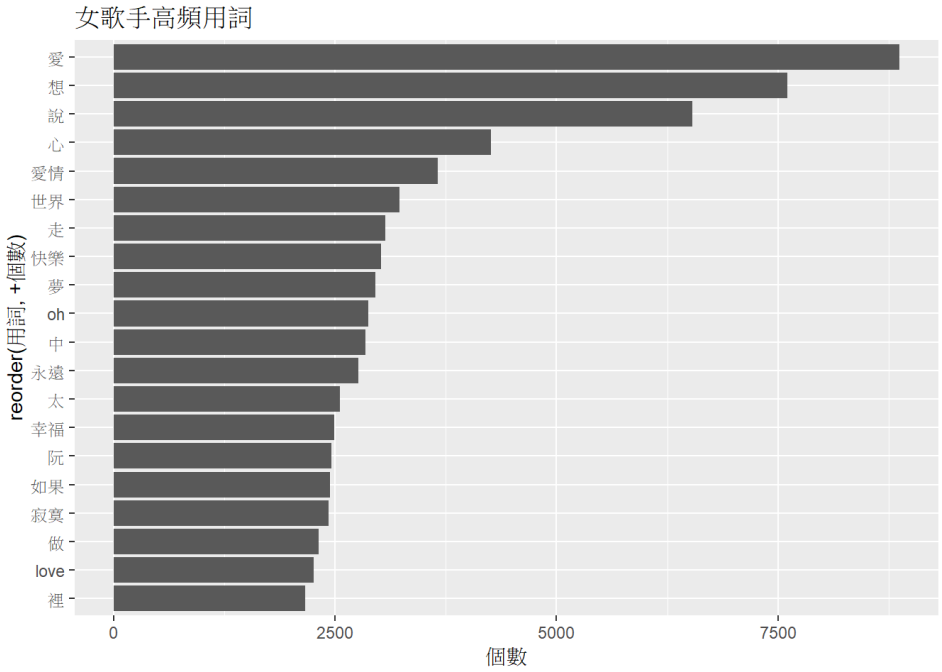

圖 6：女歌手高頻用詞長條圖（前 20 名） - 修訂停用詞版

圖 7：男歌手用詞文字雲 - 修訂停用詞版

圖 8：女歌手用詞文字雲 - 修訂停用詞版

以修訂版的停用詞表來分析，男歌手的歌詞中總共有 1232244 個詞（tokens），其中有 99402 個不重復的詞（types）；女歌手的歌詞中則分則有 1143804 個 tokens 和 89816 個 types。

觀察修訂版的長修圖及文字雲可以發現，男女用詞在大部份的高頻率的字詞上相似度很高，而在有些出現頻率稍低的詞則會出現男女分別。例如「一生」只在男生的歌有較高的出現率，而「孤單」則幾乎只在女生的歌中出現（但是男生的歌詞中有不少「孤獨」）。

高頻用詞中，最高的幾項出現的總次數非常高，「愛」這個字在男女歌手的詞中都出現了超過 8000 次。一樣是高頻用詞，男女的「說」和「想」出現頻率剛好顛倒。

女生的歌詞中「阮」的出現頻率很高，應該是台語的「我們」，表示歌詞中混入了不少台語歌，不能確定這對數據會不會有很大的影響。

3.2. 分析

有些平常使用的詞彙，本身可能帶有一些平時不容易察覺的性別意識在其中。例如歌詞中女性會使用「孤單」，而男性則偏好「孤獨」，就是很有趣的一點，雖然意思相近，孤單和孤獨乍聽之下也沒有明顯的男女氣質分別，歌手使用上卻仍各有偏好。

在差異之外有點令人訝異的是男女在大部分詞彙運用上保持非常高的一致性，出現明顯差異的地方都在整體出現率較低的詞彙上。一致性最明顯的地方，男女對「愛」的興趣都非常濃厚，如果去除「說」和「想」這兩個主題沒這麼明確的字，「愛」的出現頻率在男女用詞都足足領先了第二名一倍，28000 條歌中出現了 16000 次。

總而言之，就大方向來說，男女歌手的用詞相當一致，而大方向之下有許多雖然較微小，但不該忽略的重要差異。

4. 討論與貢獻

一開始本來想做情歌相關的研究，用機器學習分辨情歌和非情歌，然而在非情歌的定義上遇到了困難，情歌之外的歌曲領域不容易找到一個有代表性的領域可以與之對照，再加上同一條歌在不同情形下可能是或不是情歌，導致我們很難取得大量公正的對照組資料，而將目標改為觀察男女情歌的用詞差異，比較不會產生定義（是否為情歌的）問題。

而分群的依據，最後決定是以歌手性別、而不是作詞人或作曲人的性別做為歌曲分類的方式，一來是魔鏡歌詞網的資料格式不一，歌手資訊相比於作詞、作曲人的資訊取得更為方便，二來以歌曲的內容來說，作詞、作曲人在創作時依然會以歌手的角度或心境下來描繪，歌手在將歌曲收錄至專輯或單曲

時，也會衡量該歌曲是否符合自己的風格等等，所以我們認為，以歌手性別做為分群依據應是合理的。

這次的結果應能作為研究社會性別差異的素材。例如結論中提到的「孤單」和「孤獨」的頻率差異，在辭典解釋上，「孤獨」偏向社會狀態，而「孤單」則是表達心理上的感受，我們推測在撰寫歌詞時， 就算歌唱本身是一種鼓勵情緒表達的藝術，仍會受到「男性不應該輕易表達（脆弱）情緒」這種印象所影響。歌詞的表現手法多變，直白的、含蓄的、熱情的、晦澀的，雖然意思常常因此變的不明確，但也因此容易規避社會框加的束縛，更接近內心真正的想法。

在差異之外有點令人訝異的是男女在大部分詞彙運用上保持非常高的一致性，出現明顯差異的地方都在整體出現率較低的詞彙上。也許可以解釋成，其實男女之間的差異沒有想像中的遙遠。

我們這詞的研究也有不少不足之處。首先，停詞表並沒有很嚴謹的被決定，而停詞表示這個研究非常重要的一環，會極大幅的影響研究結果，如果時間充足，應該對停詞表多做闡述。其次，只根據詞頻來分類男女歌手略顯粗糙，詞語之間的關聯性也很有可能有重要差別。最後，我們的資料本身不夠乾淨，有混到不少台語歌跟少量日語和英文，因為我們是根據歌手去爬歌詞。雖然有手動整理了一下，但或許有更好的資料爬取或整理方式。

5. 分工

余興祐：歌詞爬蟲、資料清理

古貿昌：歌詞斷詞、詞頻統計、長條圖與文字雲製圖

施文捷：海報製作、書面報告文字

許家誠：書面報告文字、排版