IR Final Project

Yossip

R03725009	R03725031	R03725032	R03725039
楊天怡	施岱伶	廖書萱	林劭軒

1. Purpose

從前想要知道大眾對於名人的評價可能要透過電話訪問，或是問卷調查，也就是說想要得到大眾的回饋，必須要主動去蒐集資訊、透過比較麻煩的手段，耗費人力及時間才有辦法做到。然而現今由於網路社群日益發達，許多熱門的平台都提供大眾留言評論的功能，可以藉由網路直接接觸大眾對某個人事物真正想法及喜好程度。所以我們希望藉由抽取部分社群網站的評論，獲得第一手大眾對於名人的評價及印象，以作為名人行銷時的輔助工具。舉例來說，在歌手發行新專輯、播出新 MV、或是電影偶像劇推出時，製作商或經紀公司也許會想要了解聽眾、觀眾對於該影片或影集的評價，根據結果來決定下一步如何進行行銷。

目前先將目標鎖定在歌手，而一個歌手的形象多半來自於他所發行的歌曲，所以我們選定 Youtube 作為分析的平台，藉由網友對影片的評論，來找出大眾對於該影片或該歌手的喜好程度，並了解他們在網友心目中的形象。尤其在發行 MV 新歌時，我們觀察到在 MV 底下引起熱議的留言多半精確地點出該歌曲或歌手的特色。

當然評論中也有很多和歌曲無關的評論，尤其是具爭議性的國內外歌手，例如蔡依林的新專輯中，評論包含兩岸議題、藍綠議題、以及中韓問題、釣魚台問題等等，當然也包括一些和她形象相近歌手的比較。因而發現一首歌、一位歌手可以牽涉這麼多層面的討論，對此我們將其視為未來可以列入考量的特色：各個不同歌手的特性與背景，會延伸出甚麼不同的問題？當一位歌手下方留言出現很多次另一位歌手的名子，是否兩位歌手之間有關聯？

2. Solution

從前段得知我們這次 project 的兩個目標：

1. 找出明星在網友心目中的形象
2. 找出明星在網友中的喜好程度

而我們實作上的呈現方式為：

(1) 找出關鍵字製作關鍵字雲

高點閱率影片會有較多網友參與評論，而做出來的結果也會較準確。因此我們選擇一些知名的國內外歌手，並挑選這些歌手在 Youtube 上高點閱率影片作為探討對象。我們的做法是擷取影片中的評論，經中英文不同的斷字方式後，利用 tf

來判斷該關鍵字的權重，製作成關鍵字雲。

※關鍵字權重不用 tf-idf 的原因在於 df 在這裡指得是在多少個影片評論出現，高的 df 值對於描述一位歌手是有利的；若是分析單一影片的話，也許 tf-idf 就是較適合的選擇。

(2) 分辨正負項評論

要找出喜好程度，除了使用者本身 rating 外，我們希望能將留言分為正向評論及反向評論，然後計算出這個明星的評論正負值，以了解該明星在大眾心目中的聲望以及支持度。若是 Positive 指數很高，則代表大眾對於高明星的印象普遍較佳；若 Negative 指數很高，則代表有著較不好的印象與評價；若是分數趨近於 0，則可能是大眾對於該明星沒有特別的喜好程度，或是該明星是個具有爭議性的人物，導致每個人對他的觀感好壞差距甚遠。

Detail implementation

- **擷取評論（Python）**

  我們透過 Youtube 提供的 API，根據使用者給予的歌手名、前幾個搜尋結果和前幾個評論數值，用相對的 URL 得到 Youtube 產生的 Json。

  ※意思為從 Youtube 上挑選出搜尋某明星會跳出的前 n 個影片，擷取影片下前 n 個受歡迎的評論（n 由使用者決定）。

- **斷詞（Python）**

  我們將中英文的歌手分開處理，若為中文名字則自動使用中文方式擷取；若歌手為英文名字則判斷以英文的方式擷取評論。

  中文斷詞：Jieba 組件，因為大陸人開發，簡繁體都可以使用。

  中文 stopword：選擇通用詞庫並自行增加台灣較常使用的詞彙。

  英文斷詞：根據空格擷取 terms，但因關鍵字完整性未做 stemming。

  英文 stopword：使用老師提供的 stopword，手動加入某些單字。

- **關鍵字雲（Javascript、Python）**

  擷取評論後，後端 python 利用斷字組件找出 terms，並刪除沒有意義的 stopword，使用 tf 為指標排序 terms 的重要性。

  由後端將關鍵字 dictionary 丟置前端，經過排列的關鍵字由 d3.js（javascript 的 library）中的 d3-cloud 來接收 keyword list 並根據不同權重畫出對應的文字雲。

- **建立網頁介面**

  使用 Python 語言來撰寫 web 介面，運用 Django 框架建 web。在使用者 query 後會產生三個頁面：關鍵字雲、重要評論關鍵字及評論正反分類。

- **留言分類**

  利用 Chinese corpus 正反面詞庫，並加入自行訓練的一些常用語言，判斷留言對於歌手是正面或反面的評價。中英文分別判斷。若留言中出現一次正片詞彙，則視為+1，一個負面詞彙則為-1，總和在除上留言總字數。依此判斷出該則留言的正負面分數總和。

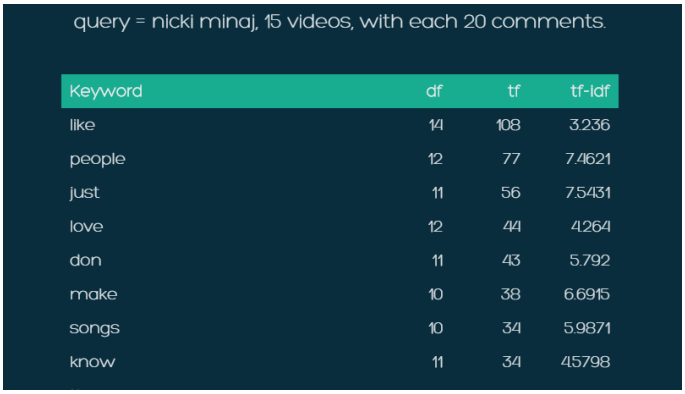
- **Bubble chart**
  由 static json file 讀入已事先分析過的部分歌手資訊，加入使用者搜尋的歌手後，並利用 d3.js 套件來製作 bubble chart。目前設定舊有對照組是該歌手前 20 個影片的所有評論。
- **建立行動版 APP**
  希望能夠建立一個更方便使用者操作的介面。
  使用 java android 來撰寫 APP，接收網頁端產生的圖。使用如下 query:（q 為歌手名，v 為影片數，c 為評論數，c=all 代表全部評論）
  http://ourIP/IRproject/forApp/?=q=”nicki+minaj”&v=10&c=10

3. System outcomes

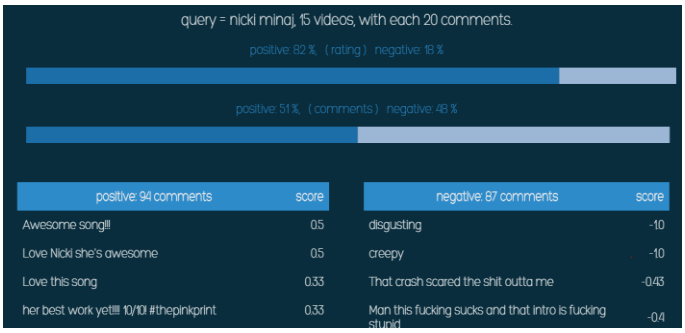
(1) 網頁版

使用者可以自行輸入想要查詢的明星，並且可以設定想要搜尋的影片個數，以及留言個數。可以瀏覽關鍵字雲、關鍵字權重以及留言分類。

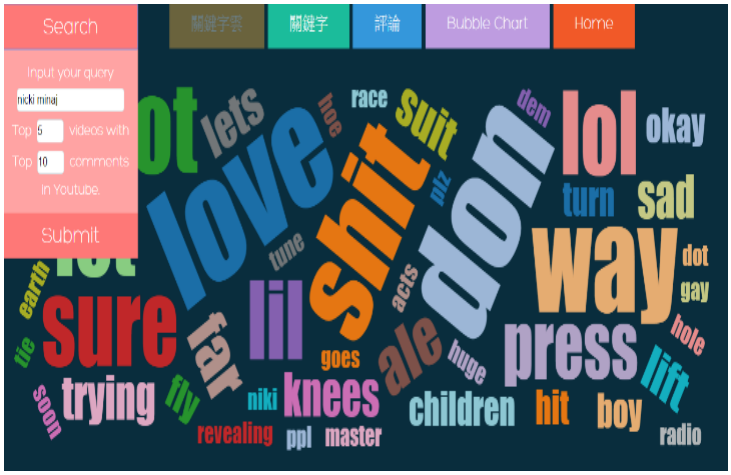
➢ 關鍵字雲：權重大的關鍵字會比例較大，反之不重要的則越小



➢ 留言分類：以長條圖呈現正面反面評論的比例，並以左右兩欄分別顯示分數前幾高的正面評論以及反面評論。

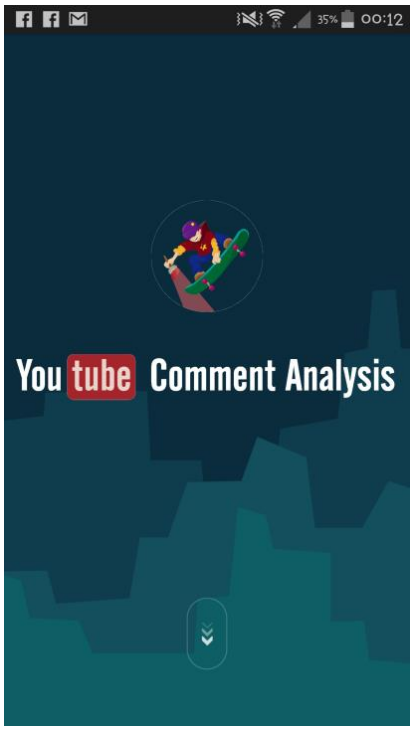


➢ Bubble chart：不同的歌手各自是一個 bubble，bubble 的大小代表該名歌手前 n 部影片的平均觀看次數(已存成靜態 JSON 檔供程式讀取），而 bubble 左右半圓分別按照比例顯示正反面評論數。可以比較搜尋的歌手與其他在線歌手的熱門和受歡迎程度。



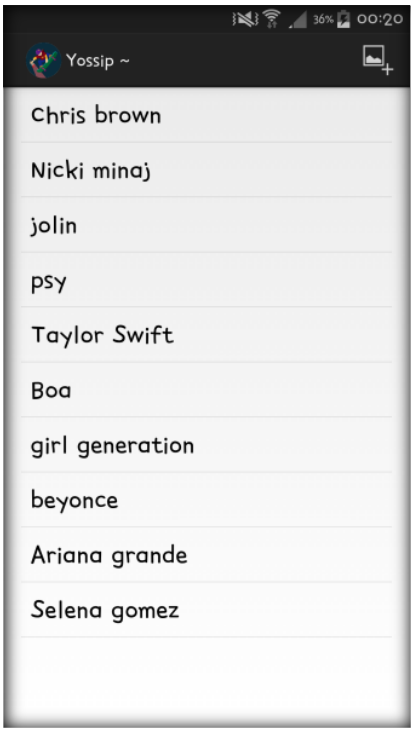
(2) 行動版

使用者可以自行輸入想要查詢的明星，並且可以設定想要搜尋的影片個數，以及留言個數。可瀏覽關鍵字雲，並查看之前所查詢的紀錄。

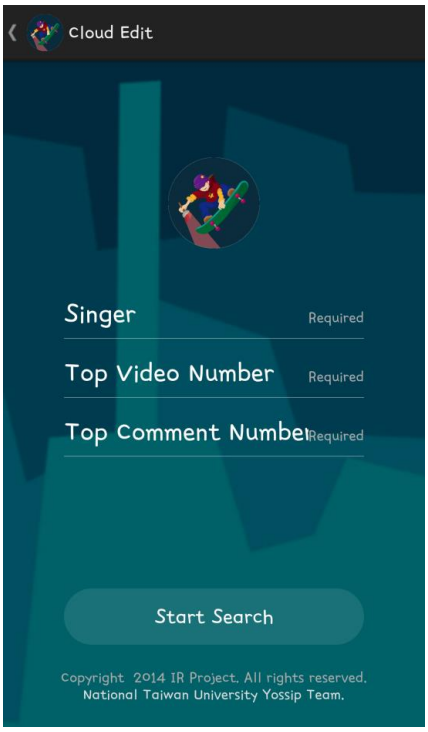


起始介面

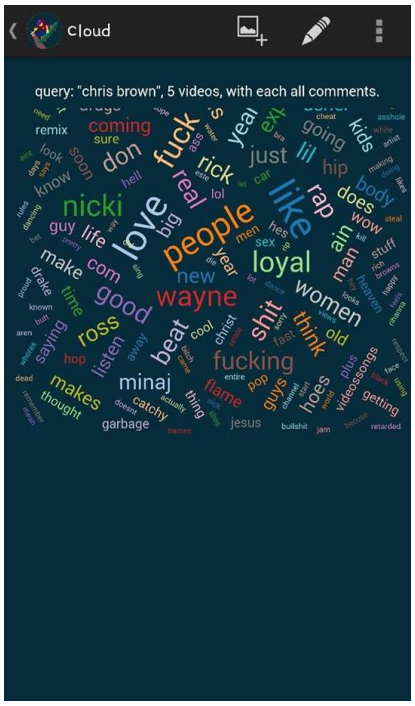
Loading Animation



過去檢索 List View



新增雲頁面



Word cloud

4. Conclusions

我們建立的這個系統，可以讓使用者可以輕易地了解大眾對於明星的印象，使用關鍵字雲可以一目了然甚麼是對於明星最顯著的評價，不僅有娛樂效果，也可以具有商業上的用處。

在商業上，我們認為有以下幾個運用及延伸

- **唱片公司**的角度：一個歌手在發行唱片時最希望的就是能夠及時掌握大眾的喜好程度，本程式可以在 MV 放上 youtube 後，及時對於評論做出一個統計，並找出關鍵字，了解多數人對於新歌的想法。藉此可以找出後續的主打歌，或是調整 MV 的拍攝方式。
- **經紀公司**的角度：可以透過此關鍵字雲或是評論分類，來判斷他們為此歌手打造的形象，是否有深植到大眾的心，或是是否討觀眾的喜愛。若發現主打的形象與評論中大眾口中的形象相差甚遠，可以考慮是否要轉型，或是在該方面做更多的包裝宣傳。
- **公司找產品代言人**：即可利用本程式來了解該藝人在大眾心目中的形象，藉此來考量是否符合該產品想要給大眾的印象，此一人是否與產品的概念有所差距。
- **明星自我評估**：藉此了解網路上對於他的評價如何，或是粉絲期待他的走向如何，可以依此做一些改善或調整，或是做為未來發展方向的一個參考。

Issue

(1) 受限於斷字和中文本身文法問題，我們對於中文的正反面判斷還無法達到很好的精準度。而不論中英文，反面評論的判斷誤判率比正面高很多，例如：This song make me cry.會被分成反面評論，但實際上並不是。

(2) 選取關鍵字的方式單純從 tf 判斷可能不足，如何對關鍵字作排序可以根據分析的對象而有不同的方式。

(3) 因 Youtube API 只顯示評論而未顯示回覆評論的評論，在收集資訊上可能因此漏掉一些資訊，另外排序是按照評論回應數（但我們認為評論本身按讚數也很重要）。

Future Work

(1) 根據中文邏輯讓中文斷字更加精準，而英文考量到評論有些有趣的片語型關鍵字，也應嘗試除了空格外的斷字方法。

(2) 目前都沒有針對關鍵字詞性作分析，如果能夠判斷每個字的詞性，不管是正反面詞彙的記分和尋找關鍵字都會有幫助。正反面詞彙可能因為前面的副詞而有更多的權重，或是根據該字出現的位置記分也可能有所不同。（如：doping as f*ck 這類的評論會被歸為負面）

(3) 因考量到使用者的某些用字習慣，評論使用的字通常都是流行語，有很多反諷的用法，或者該字已經演變成別的意思，這種情況下字典必須有所調整。

(4) 歌手常常被互相比較，Youtube 下方的評論很容易成為"戰場"，當某位歌手一直在另一位歌手影片下方出現，代表兩位歌手可能有相似之處（或誹聞），這對於分析某位歌手也是個有利的資訊。（可以用 cosine similarity 之類的方法來分析相似度）

(5) 未使用資料庫存取資料而是等使用者下query後才去分析JSON檔，可能較不穩定且速度較慢（但訊息即時）。為了增加效能，除了修改程式外也可以嘗試透過其他方式存取資料。

5. Member's workload

廖書萱	App 圖片設計、字庫分類、上台報告
林劭軒	擷取評論、關鍵字雲、評論分類、網頁技術
楊天怡	文件、語意字庫分類整理、上台報告
施岱伶	Android APP 製作、App Demo

6. References

[1] Jieba, https://github.com/fxsjy/jieba

[2]Word_cloud, https://github.com/amueller/word_cloud

[3]D3-cloud, https://github.com/jasondavies/d3-cloud

[4]D3 bubble chart, http://bl.ocks.org/mbostock/4063269

[5]D3 bubble chart example, http://www.nytimes.com/interactive/2012/09/06/us/politics/convention-word-counts.html?_r=0

[6] Chinese corpus, sentiment analysis' data https://github.com/chagge/chinese-corpus/tree/master/emotion-dic/sentiment

[7] English corpus, twitter sentiment analysis' data https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107/tree/master/data/opinion-lexicon-English

[8] Django Girls 學習指南, http://djangogirlstaipei.gitbooks.io/django-girls-taipei-tutorial/content/

[9] Android github: https://github.com/iLanguage/iLanguageCloud