台北大學電機系

專題企劃書

# 建立社交論壇文章情緒文字分析系統
# 運用即時資料作分析

學生:康富城

學號:410187042

指導老師:林嘉洋 教授

# 大綱

研究背景:

近年來對於大數據分析的相關技術已成為現在科學的主流，小企業在看到許多大企業(Google,阿里巴巴,facebook)成功的例子上，紛紛相繼投入尋求這方面技術之人才。在政府公開資料網站中,近年來癌症的預防(健康)、果物的進口出口（飲食)、高速公路路況（交通）等種種生活相關資訊，人民能有效地做為參考去改善以及避免。文字探勘為數據分析的其中一環，對於客戶在各個交易論壇、拍賣網站上所留下來的一字一句，都可能是影響其他客戶對產品評價的基礎。如何快速並有效的掌握客戶對產品的評價，改善其中的缺陷，往往能在同類型的商品中鶴立雞群。而對於社交網站、聊天論壇裡，面對每天不同的相關主題，如果能夠使用有效的方法，探討大家對某個主題的熱烈以及好壞程度，並作出統計分析呈現，簡單的便利帶來的是更多的人潮。學者 Thomas.H.Davenport 曾在著書標題寫上”分析技術之競爭:新科學時代的贏家”學習大數據分析已是不可阻擋之趨勢。

繼近期將投稿至 IEEE itpro 的論文: ***Increasing Vehicle Sales by Leveraging Text Mining and Sentimental Analysis*** 後，本人將嘗試著對**系統作出改良**。原本僅止在 R 軟體及其介面開發 Shiny 作出呈現系統，將利用目前最受歡迎的 Nodejs 框架去做伺服器的開發，此外，對於原本 Mysql 資料庫無法作即時資料存取的部分，改變成 Rethink 資料庫去做存取改善。

研究內容:

本論文將介紹文字探勘之**系統開發**:結合 **R 軟體**以及 **Node.js** 的開發系統**作前後端的開發**，並連接能作即時訊息(Real-time data,RTD)的資料庫 **Rethinkdb**，嘗試著快速有效率的對於論壇與社交網站的文章分析，作出一個有各式統計圖片的數據分析網站。使用者可以自行選擇關鍵字以及想要搜尋的論壇網站，對特定的人事物時間做仔細的解析。本部分分為**兩個部分**，第一部分為對於軟體的介紹與同類型軟體的比較，第二個部分為整體的流程圖與分配解析。

PART.1:

R 軟體程式語言自從 1995 年被 Ross Ihaka 與 Robert 發明以來，近年成為各大業界膾炙人口的搶手貨，有學者曾說過:”The closer you are to statistics,research and data science,the more you might prefer R.”R 在商業數據分析、統計圖繪製呈現相當出色，區區的一兩行就能造出相當的模組。也因為是開放式資源(open source),許多使用

者貢獻了各式各樣的 R package 幫助程式之開發，更對於大數據之分析相較同類型程式語言有更優秀的方法去解析。

Python 是常常與 R 作比較的開放式資源程式語言，它主打著更有效率，更簡單的程式編譯與演算法，同樣吸引了許多人去學習。Python 擁有更彈性的空間去做設計，對於網頁應用開發表現相當亮眼，也同樣的支持著資料分析。常常有人會問，為何先選擇 R 軟體而不選擇 Python，藉此做了一個表格，簡單比較與 R 軟體與 Python 程式語言之優劣，並解釋選擇 R 的理由。

| | R 語言 | Python 語言 |
|---|---|---|
| 適合方向 | 資料分析，統計及繪製圖表 | 高效能，系統開發，網頁應用程式 |
| 上手難度 | 起頭較難，需要一些程式基礎。 | 主打新人可以簡單上手。 |
| 數據處理能力 | 有龐大的套件支持，對於簡單的資料可以直接分析。 | 較少套件,近年來逐漸解決這方面的問題。不過基本的處理就得先下載套件。 |
| 分析程式碼 | 簡單幾行即可寫出統計模型。 | 利用各種演算法去實踐模型。 |
| 使用人數 | 相比較少 | 相比較多 |

表 1. R 與 Python 之比較

比較起來，因為在程式語言上有了一定的基礎，對於資料的分析希望以更簡潔的程式碼即可呈現繪製模型。但不可否認的是，python 在效能處理方面仍是略勝一籌，在未來會學習 python 程式語言，以便未來做系統改善的動作。

Node.js 是一個網頁的伺服器，也是一個應用程式之框架，它採用的是 Google V8 Javascript Engine ，一個效能相當優秀的引擎。自它在 2009 年被揭露後，受到了全世界企業的注目，各大知名企業紛紛投入人才去研究。而它如此受關注的背後，是藉由著 Javascript event-driven 之特性，讓他在後端的程式開發收益良多，造成即時訊息應用的開發簡化，不僅如此，它更讓過去一向讓人詬病的 Javascript 這個腳本語言受到應有的重視。Javascript 採用非同步概念，簡單來說也就是能把一些

工作交給後台背景處理，在開啟網頁時，不必等所有的工作處理完即可先呈現基本樣本。Node.js 能支援第三方的模組(Third-party Module)，方便開發者使用他人已完成的模組去實作功能。

PART.2:

下圖為這次研究計畫所規劃之系統流程圖：

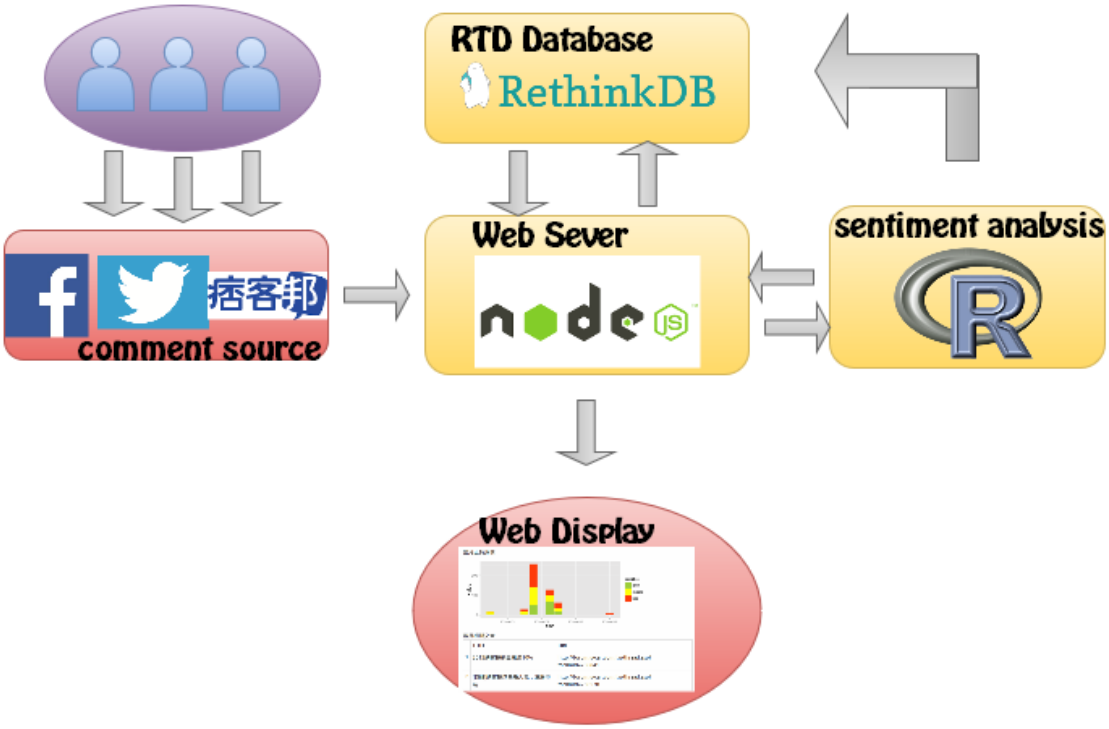


圖 1. 系統總流程圖

以下將介紹有關製作的解說:

1. 以取部落格為例，將一篇篇與關鍵字相關的文章找到後，藉由 node.js 寫出一個基本的爬蟲程式，配合連接 RethinkDB 分為時間、作者、內文三部分先存入，此程式將會持續在一定時間內更新，得到最新的資料。

2. 接下來使 node.js 與 R 軟體作連接的動作，從資料庫中取出變數，結合 R 軟體與其下套件(Rvest)作分詞之斷字斷句，完成後再回存至資料庫。同時在另一方面，利用隱馬克夫模型去對字詞間的關係作分類，找出關鍵詞特徵所相關的情緒字詞分成正面、中立、負面三種，再存入資料庫。

3. R 再次藉由 node.js 取得分詞過後完成的詞彙後，利用套件(ggplot2 等)作統計圖與折線圖,回傳給 node.js 在頁面上顯示模型圖表、相關文章網址及大家是否對此表達好壞偏向的表格。

4. 隨著使用者想找的關鍵字變多，可以呈現作比較分析關鍵字互相在一段期間內所被討論的次數，作出折線圖、長條圖等作比較。

5. 使用者可以從網頁介面端輸入想要找的關鍵詞去搜尋，也可直接按近期熱門關鍵字去察看比較。另外使用者可自行決定圖表的呈現多寡，可以支持多個關鍵字之比較。

## 未來研究:

儘管這篇是基於上一篇論文之系統作強化與改善，在情緒分析字詞的部分，仍然有些準確度的問題存在。相信這是一直以來都在改善的部分，未來我在這方面會結合機器學習作自動分辨，並想辦法提高分詞之準確率。

研究內容有提到 Python 的高效能，對於此優勢可以將一些資料的運算統計放入 Python 來計算，R 則將利用傳過來的資料呈現圖表，作進一步之改良。

## 結論:

這個系統開發的目的，是希望能給予使用者有用的資訊，節省他們在網路上用人工搜查的方式所花費的時間，也希望可以使用到商品銷售上，讓企業有效的知道自己商品的特性，長處的推廣與短處的改善。

隨著新的技術與方法不斷被發現後，對於這系統的改良方向是可以無限延伸的，近來不只要對文字，對**圖片、聲音等非結構化資料之分析**也相繼崛起，活在這大數據時代，自然對 3Vs(volume, variety, velocity)的要求也就永無止境。在未來本人將追求新的知識與技術，分析法的改善與演算法的改良，並多嘗試不同的程式去補強自己的知識視野。

## 參考文獻:


[1]. Choosing R or Python for data analysis? An infographic

http://blog.datacamp.com/r-or-python-for-data-analysis/

[2]. Comparing Python and R for Data Science

http://blog.dominodatalab.com/comparing-python-and-r-for-data-science/

[3]. An Absolute Beginner's Guide to Node.js

http://blog.modulus.io/absolute-beginners-guide-to-nodejs

[4].粉丝日志 从零开始 nodejs 系列文章

http://blog.fens.me/series-nodejs/

[5]. Bo-Yan Tseng and Fu-Chen Kang.“Increasing Vehicle Sales by Leveraging Text Mining and Sentimental Analysis”

[6]. Shruti Kohli and Himani Singal.“Data Analysis with R” 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing

[7]. Andreas Pfeiffer and Maria Grazia Pia. “Data analysis with R in an experimental physics environment.”

[8].錢逢祥、蔡正崇、林政毅著 ,不一樣的 Node.js 用 Javascript 打造高效能的前後台網頁程式