激! 火力全開 - 網軍攻防視覺化系統

廖宜珊 R06725043　方心成 R06725045　何智誠 R07725027　鄧鈞岱 R07725030

1. PURPOSE

近年來“台灣政治”長期以來一直是媒體高度關注的議題，從前的時代大家習慣藉由如電視新聞、廣播節目來被動接收每天所發生之議題，雖然達到關心時事的目的，相對卻被無形的電視台立場束縛住。近年來隨著網路及線上平台蓬勃發展，越來越多人能夠藉由 PTT、FACEBOOK 等平台做到真正的議題參與，在發表針砭時事的高見同時，也能與其他網友相互討論，從而激盪出更多元的想法與觀點。不知不覺，這股力量漸漸展現出其價值所在，2014 年的太陽花學運正是從 PTT 平台中反對服貿議題的文章發展成影響國家政策的事件，同年間一位台大醫生柯文哲也受惠於他在網路平台被廣為推崇的聲勢與串聯支持下，最後推翻長期以來的藍綠之爭而以無黨籍當選市場。

至此，線上選舉活動開始備受重視，各政黨和候選人都試圖透過網路來拉近與選民的距離，一方面希望多多宣傳自己外，另外一方面這種積極的做法也能打擊對手。直到最近九合一選舉期間，由於競爭過於激烈而衍伸出是否存在所謂的「網軍 – 發表與議題本身無關之言論，透過抹黑及仇恨言論影響意識形態」角色，對於候選人與議題的也從尊重、包容、友善的討論轉而變成互相懷疑、嘲諷、攻擊的純粹對立局面。然而這種情景是否發生大多依靠主觀判斷，或沒有積極證據的推論，也無從得知台灣在政治議題上是否會因為選舉因素導致火藥味上升。

因此我們希望利用上課所學，對來自網友們的留言進行情感分類，依據內容字眼來判斷此句話是否帶有負面情緒，並配合網頁技術建立一個視覺化的平台，指出到底攻擊量實際有多少？藍營還是綠營被攻擊的比較多？反映出特定時間中各政黨人物粉絲專頁底下的文章所獲得之情緒狀況，一方面以數據為基礎回答網軍之相關問題，另一方面以較輕鬆的方式展現出潛藏在言論自由背後日趨嚴重的問題。

2. SOLUTUION

根據上述所提及之動機與目的，我們希望能夠爬取 Facebook 的 PO 文留言作為資料來源，以現行的分類器與機器學習方法建立出情緒分析模型後作為判斷標準，最後持續針對每天出現的新資料以模型擬合之，並以視覺化方式呈現於網頁。我們主要以 Python 語言實作程式，運用資料爬蟲以及文字探勘課程所學與機器學習演算法建立各系統功能，以下我們將詳述整個專案的實作流程與技術內容。

2.1 爬蟲部分

在建立情緒分析的模型前，必須要有資料來源來幫助建立分類器，討論後決定爬取 2018/11/24 選舉前 10 天內政治人物的 Facebook 粉絲專頁 PO 文留言作為訓練集資料，包含當時聲勢最為龐大、各陣營最具代表性的韓國瑜、柯文哲、蔡英文、時代力量(立委)作為目標。由於 Facebook 本身提供的 API 限制過多，我們選用 Selenium 作為爬蟲開發工具，Selenium 能夠模擬瀏覽器真實操作網頁，對於含有較多 JavaScript 語言設計的網頁能夠搜索到更完整之資訊。

2.2 資料前處理

我們利用爬蟲所蒐集之政治人物粉絲團貼文底下所有留言評論來進行分類模型建置，而在模型建置前，必須將留言依據其語句情感實作標籤判斷，接著處理斷詞與停止字眼，最後透過卡方篩選出較有影響力之詞彙來進行模型訓練與分類預測。

2.2.1 資料標籤-人工萃取法

在資料標籤部份，原本預計採用「情感字典法」來做語句情感分析，透過線上既有字典（如: NTUSD...等）來自動判斷語句正、負面情緒，並透過語句所得綜合分數計算來標籤分類，然而，我們認為此方法彈性較低，且既有字典過於制式化，容易會有誤判情況發生，尤其是參雜複數負面字眼的正面語句，像是:

「不不不~蔡總統才是我們的未來!」

在情感字典法中，我們假設「不」這負面字眼分數為 -0.8，而這語句開頭包含連續三個「不」使得綜合計算後該語句所累積之負面積分高，致其容易被分類標籤為負面語句，但其整體字面上來看應該是屬於正面語句。

因此，我們選擇「監督式機器學習法」為分類方法，並搭配「人工萃取法」作資料標籤處理，將蒐集而來之選前評論留言語句採用人工方式給予情感標籤，逐一地標記共 **12000** 則正面/中立/負面情緒，又由於對評論的判別具有一定的主觀與複雜性，過程中盡量只考慮單純語句整體之正負面評價，並未考慮到該留言正、負面評價的對象是誰，此階段處理截圖：

72	加油 你好棒, 我老家台南,在高雄工作 全家族10幾張票都是	3
73	看今天其選場還是有不少人還沒覺醒，請大家記得催票拉票	3
74	白癡吳x義，不要再用謾罵影射了，這種方式不但讓對手有	1
75	我們要過好生活，我們不要當低收入戶領補助，泯黨又開始 讓我這個小老百姓來告訴妳，這代表妳們執政無能，代表妳 打死我都不要，因為我有自尊	2
76	願上帝讓您有如摩西帶領以色列人過紅海 禱告韓國瑜11/24帶領高雄市民邁向偉大的未來	3
77	韓國瑜主委 精神 誠心祝愿和祝福你 1124翻轉高雄成功 大家	3
78	謝長庭搞假錄音帶、陳菊弄假走路工，都不是誠實的人，而	2
79	為今晚辯論會祝福！太可怕了要去三立😭😭😭為什麼一定	2
80	我是一個外飄的台灣人 我其實不喜歡台灣政治 但是韓總 讓	3
81	感動...今天經過學校..正播放著國歌及國旗歌...好久了……沒 不論勝不勝選……您已啟動台灣失去已久的...愛國熱情...迫	2

圖 1. 人工萃取法資料標籤結果

此外，用於呈現視覺化的測試資料，我們爬有 2018/09 至 2019/01 約 25 萬筆 Facebook 留言，來自於目前台灣政壇上較有知名度的政治人物（詳見系統架構&網頁內容）。

2.2.2 Tokenize & Stopword

無論使用哪種特徵做為模型依據，都需要針對資料作前處理，目前市面上最普遍之

Python 中文斷詞套件為 Jieba 結巴，但由於我們的專案大量包含“台灣政治”相關文字語句，從結果來看無法滿足特定專有詞彙，例如處理國民黨若不添加特定規則可能切為“國民”、“黨”，又如“柯 p”詞彙原始套件設無法成功保留，必須耗費大量時間解決。

最後我們使用了 **UdicOpenData** – 中興大學普及資料與智慧運算實驗室提供的開放資料，該專案提供來自台灣 PTT 論壇上爬取的大量鄉民留言，並結合 Jieba 套件原有優點，最後釋出成用以 tokenize 與 filter stopword 之 Python 套件，簡易測試如下:

- **原句為:**

備受關注的「蔡柯會」在 13 日下午登場，台北市長柯 P 致詞強調，「大巨蛋」需要中央幫忙解套，蔡英文則指出，今天會談後將指示行政院以專案處理

- **Tokenize & Stopword:**

備受|關注|蔡柯會|下午|登場|台北|市長|柯 P|致詞|強調|大巨蛋|需要|中央|幫忙|解套|蔡英文|指出|會談|後將|指示|行政院|專案處理

2.2.3 特徵篩選-卡方

初步排除出現頻率過高的不重要詞彙後，接續依據「bag-of-words」、「bag-of-words+bi-grams」、「tf-idf」等方式來使用「卡方（Chi-square）」做重要詞彙的挑選。

屬性篩選前，在前述三種方式下的模型表現如表 1.，可以看到在 B-NB、M-NB 模型中採用 bag-of-words+bi-gram 的方式準確度最高，在 SVC 則是三種方式表現一樣，而在 L-SVC、Nu-SVC、LR 模型中採用 tf-idf 詞頻方式的準確度較高且表現較好，尤其是 NuSVC。

	SVC	L-SVC	Nu-SVC	B-NB	M-NB	LR
bag-of-words	0.6990	0.8530	0.8133	0.8300	0.8524	0.8633
bag-of-words + bi-gram	0.6990	0.8599	0.8059	0.8311	0.8547	0.8621
tf-idf	0.6990	0.8685	0.8759	0.8300	0.8346	0.8639

表 1. 屬性篩選前各分類器表現

反之，屬性篩選後，在採用 tf-idf 方式下依序挑選重要度排名前 500~3000 名的重要詞彙訓練出來之模型表現如下表，可以看到就隨著挑選詞彙數目上升，L-SVC、Nu-SVC、M-NB、LR 模型準確度皆有略為增長之趨勢，但整體準確度並沒有明顯優於屬性篩選前，因此在本研究中，屬性篩選並沒有明顯的效果。

	Feature Selection	SVC	L-SVC	Nu-SVC	B-NB	M-NB	LR
tf-idf	500	0.6990	0.8380	0.8346	0.8329	0.8139	0.8369
tf-idf	1000	0.6990	0.8512	0.8530	0.8311	0.8260	0.8443
tf-idf	1500	0.7048	0.8570	0.8535	0.8357	0.8369	0.8489
tf-idf	2000	0.6990	0.8553	0.8581	0.8420	0.8369	0.8524
tf-idf	2500	0.6990	0.8667	0.8604	0.8478	0.8380	0.8553
tf-idf	3000	0.6990	0.8679	0.8759	0.8329	0.8375	0.8644

表 2. 屬性篩選後各分類器表現

2.3 監督式分類模型（系統使用）

將前述處理過後之語句詞彙來做分類模型之訓練，在模型訓練部份，我們採用 1、(-1)兩數值來分別代表正、反面評價作為模型應變數(Y)，而相對應之語句組成詞彙則為模型自變數(X)，兩兩搭配來反覆地建置與訓練模型，而本研究中，我們將嘗試訓練各種常見

的監督式機器學習模型（分類器），像是屬於 SVM 分類模式下的 SVC、 LinearSVC、NuSVC；屬於 NB（Naïve Bayes）分類模式下的 BernouliNB、MutinomialNB；二元變數線性回歸 LogisticRegression，從中挑選表現最佳者(準確度最高)作為本研究系統最後使用之模型。

2.3.1 SVC、 LinearSVC、NuSVC

SVM (Support Vector Machine)支持向量機下主要有兩種模式：分類模式 SVC (Support Vector Classification)、回歸模式 SVR (Support Vector Regression) ；而在 Python 的 sklearn 套件中針對分類模式 SVC 有三種模式- SVC、LinearSVC、NuSVC，基本上運作大同小異，僅有些許參數設定上的差異存在:

- SVC、NuSVC 方法基本上一致，唯一區別在於損失函數的度量方式不同
  ⇨ NuSVC nu 參數 V.S. SVC C 參數：
  - nu 參數：誤差上限與支持向量下限
  - C 參數：penality 懲罰係(error_term)
- LinearSVC 在懲罰係數上有 penalty 參數可設定，即正規化參數，有 L1、L2 可選擇，loss 參數為損失函數設定，有 L1 和 L2 的設定 hinge 和 squared_hinge

2.3.2 BernouliNB、MutinomialNB

NB(Naïve Bayes)貝式分類器為透過機率值的計算來進行分類預測，據其機率計算方式不同而有兩種主要形式，單純考慮字詞存在與出現與否之機率的 BernouliNB；或是字詞出現頻率之機率的 MutinomialNB，在 Python 實作上亦可以透過 sklearn 套件來做模型之訓練與建置，alpha 參數可以設定 smoothing 與否來避免零機率的發生。

2.3.3 LogisticRegression

由於本研究應變數為二元型態，即由(1、-1)分別來代表該語句言論是屬於正面評論、反面評論，因此，該變數非為連續型而無法採用一般線性回歸，採用適用於二元變數之邏輯式回歸比較好，透過給定的關鍵詞彙，預測分類出該語句屬於 1(正向)、-1(反向)，在 Python 則可以透過同樣的 sklearn 套件來達成。

此外，我們也將透過 CNN 卷積神經網路的訓練與預測作為本研究之比較模型，並在後面將介紹並探討其與系統使用之模型的結果差異與分析比較。

2.4 卷積神經網路模型(CNN)

本模型建構是以 Yoon Kim 於 2014 年發表的論文為參照，分析並重建 CNN 模型，前述或者往常使用的文章表示方式大多是使用詞袋方法，考慮每個詞出現的頻率而不考慮詞的先後順序問題，這裡會先將詞依據 gensim 的方法將詞轉為向量表示，再透過搭建的 CNN 模型對其進行預測。

2.4.1 Word Embedding

本研究採用 gensim 的詞轉向量方法，預訓練的資料集採用兩種不同來源供後續分析，兩者分別是維基百科繁中 33 萬篇文章與 Facebook 政治人物粉絲專頁下 25 萬則前處

理後的留言，對其皆使用 gensim word2Cec 詞轉向量產生模型當作不同 word embedding 模型基準。Word2Vec 共同參數如以下:

維度 size=300 /截斷詞頻 min_count=10 / 訓練演算法 sg=0 (CBOW) / 其餘皆為預設

最終我們會依照表現較好的 word embedding 資料來源當作我們後續分析的基準。

2.4.2 CNN 架構

引入 Yoon Kim 論文中 CNN 結構如圖：

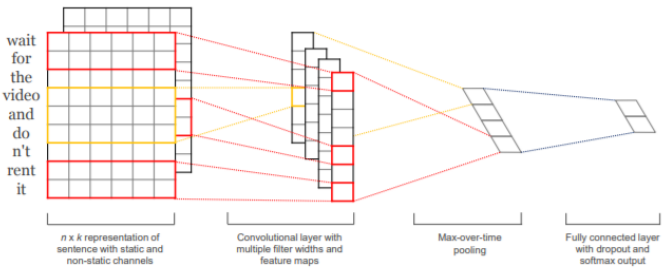

圖 2. CNN 一般架構圖

輸入層代表將一句話轉換成一個二維圖像，每一排是一個詞 word2vec 的向量，縱向就代表這句話斷詞處理後詞出現的順序，因此每張圖像的 size 便為：

最長語句詞數 * word embedding 維度

其他模型相關描述：

- 一句話若不足最長語句詞數就採 zero padding
- 卷積後的卷積層則是一個寬度為 1 的特徵地圖
- 使用 Max Pooling，一個特徵地圖只選一個最大值留下，被認為是這個 kernel 卷積後留下最重要的特徵
- 使用帶 dropout 的全鏈接層與 softmax 完成 CNN model 的建構

3. SYSTEM OUTCOME

系統的結果主要以視覺化表現，呈現出使用者搜尋的特定時段內，各政黨之各政治人物的聲量變化情況。

➔若想細看系統 DEMO 影片可以前往 https://www.youtube.com/watch?v=Urxdlzm97Fw

圖 3. 測試資料正負面情緒分類圖

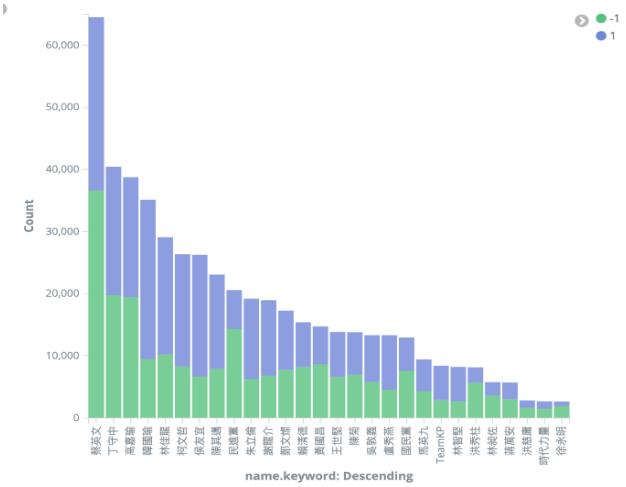

圖 4. 後端資料庫索引截圖

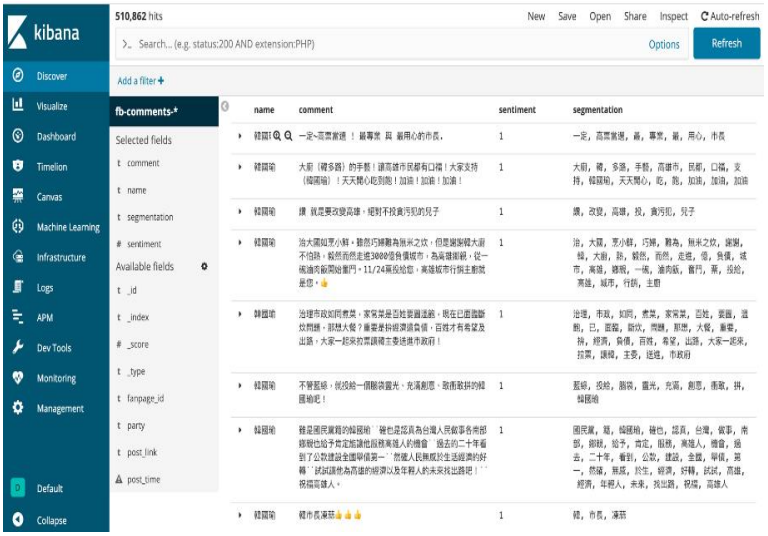

圖 5. 前端網頁介面截圖

3.1 系統架構

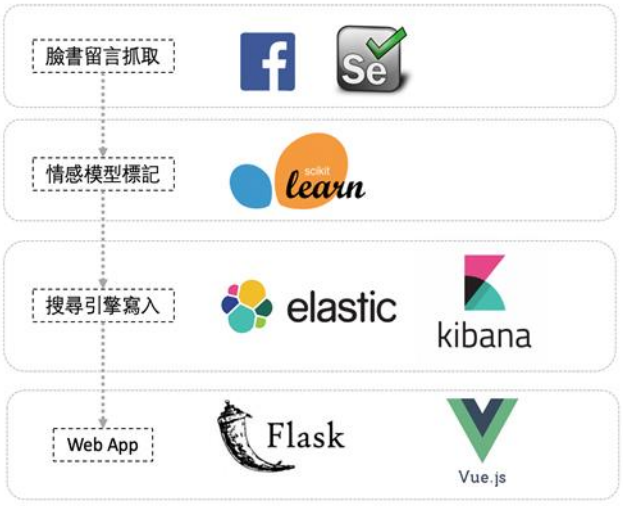

圖 6. 本研究整體系統架構圖

在留言抓取的部分，我們鎖定 28 位政治人物的粉絲專頁，抓取文章下方留言，最終資料範圍為自 2018 年 9 月到 2019 年 1 月 13 號，並使用訓練好的模型進行情緒標記。我們將標記好的留言存入 Elasticsearch 的資料庫裡，其唯一高效能的文本搜尋引擎，並搭配 Kibana 作為資料庫的可視化介面，方便我們查看各候選人的標記結果，以及有效率的使用 aggregation 的方式，根據留言時間、政黨、候選人及情緒，統整留言結果。Web 後端 Server 負責接收前端傳來的時間區段，然後從資料庫取得該區段的所有留言後，編排成前端動畫影格的格式回傳，前端在接收後便開始根據正負情感的留言量開始播放。

3.2 系統採用模型

在本段落我們將比較前述之六種常見監督式學習模型- SVC、 LinearSVC、NuSVC、BernouliNB、MutinomialNB、LogisticRegression，找出分類預測表現最好的模型來作為本研究系統模型使用，讓預測分析結果更具可信度與可靠性。

Feature/Classifier	Feature Selection	svc	Lsvc	Nsvc	Bnb	Mnb	lr
bag-of-words	500	0.715106	0.825388	0.808156	0.829408	0.826536	0.835727
bag-of-word + bi-gram	500	0.715106	0.821941	0.802412	0.823090	0.821941	0.831706
tf-idf	500	0.699024	0.838024	0.834578	0.832855	0.813900	0.836875
bag-of-words	1000	0.708788	0.834003	0.805284	0.828834	0.834578	0.847789
bag-of-word + bi-gram	1000	0.708214	0.828834	0.801838	0.815623	0.831706	0.835152
tf-idf	1000	0.699024	0.851235	0.852958	0.831132	0.825962	0.844342
bag-of-words	1500	0.704193	0.850086	0.814474	0.834003	0.843768	0.856979
bag-of-word + bi-gram	1500	0.704767	0.837450	0.796669	0.811603	0.830557	0.847789
tf-idf	1500	0.699024	0.856979	0.853532	0.835727	0.836875	0.848937
bag-of-words	2000	0.701321	0.859851	0.811603	0.834003	0.847789	0.862148
bag-of-word + bi-gram	2000	0.702470	0.842619	0.805859	0.812177	0.836301	0.854681
tf-idf	2000	0.699024	0.855256	0.858128	0.842045	0.836875	0.852384
bag-of-words	2500	0.699024	0.857553	0.816198	0.843768	0.851235	0.865020
bag-of-word + bi-gram	2500	0.699598	0.847789	0.809879	0.825388	0.845491	0.858128
tf-idf	2500	0.699024	0.866743	0.860425	0.847789	0.838024	0.855256
bag-of-words	3000	0.699024	0.858702	0.812751	0.834003	0.851809	0.866169
bag-of-word + bi-gram	3000	0.699024	0.856979	0.805284	0.819644	0.846640	0.863297
tf-idf	3000	0.699024	0.867892	0.875933	0.832855	0.837450	0.864446

表 3. 各分類器屬性篩選後表現分析表

綜合前面所述與上表 3.，可以看到：(1) 採用 tf-idf 詞頻方式的結果普遍優於 bag-of-words 詞袋方式 (2) 屬性變數篩選前後模型差異不大，可說是沒有明顯效果，但對於某些分類器（尤其是在 NuSVC）來說準確度仍有些許提升 (3) 隨著篩選詞彙數量上升，模型準確度會有些微成長趨勢存在。因此，我們挑選在 tf-idf 詞頻方式屬性挑選

3000 個重要詞彙所訓練出來之 NuSVC 分類器來作為本研究系統使用模型，可以看到分類預測準確度達到約莫 0.8759 左右。而在後續，將會將之與 CNN 神經網路模型來作分析比較。

3.3 模型分析

我們將針對第二章節提到的模型進行成效分析，並呈現比較後之結果。

3.3.1 監督式分類模型

此章節針對系統採用之文章表示方法與分類模型（Tf-Idf+nuSVC）進行分析。

首先雖然根據分類結果準確度如同上述達到接近 0.87 水準，但因為採取詞頻方式對文章進行轉換，且所蒐集資料來源為 2018 年台灣選舉前十天，可以大略估計其結果會因此而產生部分影響，因此我們分別以當時較被和不被民眾喜愛的政治人物搭配負面、正面的詞，去觀察分類模型所得出的結果，如下：

留言	斷詞、處理	正、負	對、錯
蔡英文一直都有在幫台灣做事的!	'蔡英文' '幫台灣' '做事'	負面	錯
蔡英文蔡英文蔡英文蔡英文加油	'蔡英文', '蔡英文', '蔡英文', '蔡英文', '加油'	負面	錯
蔡英文加油	'蔡英文', '加油'	正面	對
韓國瑜很爛，還要給他四年，完了	'韓國瑜' '很爛' '還要' '給他' '四年' '完'	正面	錯
陳建錦的任期做的很棒，這幾年都在替台灣默默耕耘	'陳建錦' '任期' '做' '很棒' '幾年' '台灣' '默默耕耘'	正面	對

表 4. NuSVC 系統使用模型分析比較表

以選舉當時的聲量來看，韓國瑜的網路正面聲量極大；而蔡英文網路聲量偏負，因此即使韓國瑜接續一個很負面的詞；蔡英文接續一個很正面的詞，模型依然判斷錯誤。總之，模型分類結果與我們猜想結果相近，依照詞頻處理文本所得的結果不考慮上下文，卻因詞本身的正、負效果極大影響了分類器的預測結果，雖然在選前幾天的資料會有較佳的表現，但政治人物的聲量卻不是恆定而是隨著話題、時間變動，且這也是我們專案希望呈現的正確結果，因此，為了呈現更準確且恆定的預測，以下將與參考之論文中的 CNN 架構進行比較。

3.3.2 卷積神經網路模型(CNN)

依照分類結果的 f1-score 與準確率，我們挑選 25 萬則留言生成 word2vec 的 word embedding 結果，再搭配 CNN 架構作為預測模型，而(Facebook 留言 gensim word2vec + CNN)模型的準確率為 0.864。

以下比較系統使用模型(Tf-Idf+nuSVC)與 CNN 模型(gensim word2vec + CNN)對前面表格之相同語料進行分類結果比較：

留言	斷詞、處理	正負(系統)	對錯(系統)	正負(CNN)	對錯(CNN)
蔡英文一直都有在幫台灣做事的!	'蔡英文' '幫台灣' '做事'	負面	錯	負面	錯
蔡英文蔡英文蔡英文蔡英文加油	'蔡英文', '蔡英文', '蔡英文', '蔡英文', '加油'	負面	錯	正面	對
蔡英文加油	'蔡英文', '加油'	正面	對	正面	對
韓國瑜很爛，還要給他四年，完了	'韓國瑜' '很爛' '還要' '給他' '四年' '完'	正面	錯	負面	對
陳建錦的任期做的很棒，這幾年都在替台灣默默耕耘	'陳建錦' '任期' '做' '很棒' '幾年' '台灣' '默默耕耘'	正面	對	正面	對

表 5. 系統使用模型與 CNN 模型分析比較表

從表中對照可以發現，使用 CNN 的模型在同語料中除了第一則留言分錯外，其餘分類表現皆較系統使用模型正確，可推因為

CNN 模型會考慮語句的上下文關係，而不是只依賴於詞頻而分類；雖然對選前十天我們人工標記的結果分類之準確率依舊略為低於詞頻為主的模型，但我們相信以長久來說以 CNN 為概念的模型表現會較為穩定。

4. CONCLUSION

本研究提供之系統能針對網路特定人物之留言而視覺化呈現該人物之網路聲量，使關心網路即時聲量，或想分析事件與行為對網路聲量是否有影響之使用者可藉由系統結果做為參考，判斷自身的網路聲量。

由於政治人物的聲量隨事件、時間不斷變動，要有更穩健的模型就必須不斷有新的標記資料餵入，直到分類結果較為穩定為止，由此可知大量的人工成本是必須；但我們專案四位成員時間有限，算是較可惜的部份。最後，資料標記的部份即使成員間先行達成一定共識，但對於正、負面留言的判斷依舊過於主觀，若有更大量人力與時間，每人分類相同文章並對實行統計實驗(取眾數、檢定、投票等...)應會是較佳的標記資料。

5. REFERENCES

https://scikitlearn.org/stable/supervised_learning.html

[Convolutional Neural Networks for Sentence Classification]([1408.5882] (EMNLP 2014) DOI: http://aclweb.org/anthology/D14-1181

✧ **本研究系統 DEMO 影片可以參照**
https://www.youtube.com/watch?v=Urxdlzm97Fw

6. WORKLOAD

✓ 廖宜珊：

資料標籤、監督式分類模型研究、模型改進研究、報告撰寫

✓ 方心成：

資料標籤、CNN 模型研究、模型分析研究、報告撰寫

✓ 何智誠：

資料爬蟲、資料標籤、整體系統網站架設、報告撰寫

✓ 鄧鈞岱：

資料爬蟲、資料標籤、監督式分類模型研究、報告撰寫