國立高雄師範大學

軟體工程系

專題製作報告

以 WordCloud 實現大數據與視覺化應用分析
-網路媒體解析器

指導教授: 李文廷 博士

學　　生: 鍾弘浩　撰

中 華 民 國 112 年 4 月

目錄

壹、前言

一、研究動機

隨著科技的不斷進步和數位化的普及，我每天早上吃早餐時喜歡喝咖啡配新聞，數據量呈指數級增長。大數據與視覺化研究可以幫助人們更好地處理和分析海量數據，從而更好地理解和利用這些數據。大數據時代需要更多的資訊技術人才，以處理數據分析、機器學習和人工智慧等技術。透過大數據與視覺化的研究，人們可以更好地掌握數據分析和數據可視化的技能，從而更好地滿足企業和社會的需求。大數據與視覺化的研究可以幫助人們更好地理解和利用數據，以更精確地做出決策。從而提高企業的效率和競爭力，因此，做出這項專題是針對大數據與視覺化之應用在網路媒體，這樣可以在海量文章中加快對文章主題內容的理解以及接收資訊的正確性。此項專題可以讓民眾加快閱讀文章以及了解時事主題。並且提高對報導的內容的質量還有價值，幫著廣告商和市場營銷人員更好分析消費者行為和趨勢，置地更有效的廣告和市場經營策略，幫助記者和編輯可以更好運用以及了解讀者需求和反應，優化內容和報導，讓文章內容更有說服力。

二、研究目的

研究目的有以下幾點:

1、探究大數據與視覺化技術在網路媒體上的應用，如何幫助記者和編輯更好地進行新聞報導和內容製作，提高報導和內容的質量和價值。

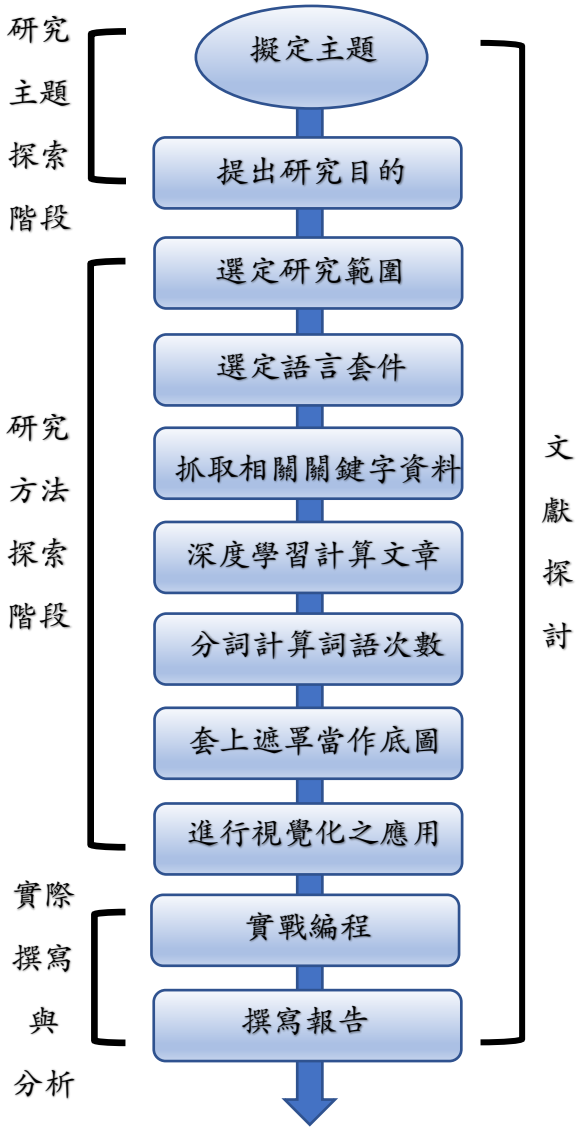
2、研究大數據與視覺化技術在網路媒體上的應用，如何幫助廣告商和市場營銷人員更好地分析消費者行為和趨勢，制定更有效的廣告和市場營銷略。

3、探討大數據與視覺化技術如何應用於網路媒體的數據分析和數據可視化，從而幫助記者和編輯更好地了解讀者需求和反應，進而優化內容和報導。

4、透過大數據分析，可以獲取更多的數據，並且從中發現複雜的模式和關係，以更好地預測未來趨勢和做出更明智的決策。

5、幫助人們發現以前未知的關係和趨勢，對於推動科學、技術和社會發展都有著重要的作用。

三、研究流程



貳、文獻探討

一、Search Analytics API now supports Discover, Google News, and Regex（來源：Google Search Central Blog）

The Google Search Console Performance reports already show data about Search, Discover, and Google News to site owners that have this type of traffic. Since we launched the Discover and Google News performance reports, we've been receiving

requests from users to also add these stats to the Search Analytics API. We are happy to announce that this is happening today.

The searchType parameter, which previously enabled you to filter API calls by news, video, image, and web, will be renamed to type and will support two additional parameters: discover (for Google Discover) and googleNews (for Google News). Please note that while we renamed the parameter to type, we're still supporting the old name searchType.

二、如何使用 python 製作文字雲（來源：Eric Cheng）

基本型：英文

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

# Read the whole text.
txtfile = "c:/test-wordcloud/cnn.txt" # 剛才下載存的文字檔
text = open(txtfile,"r",encoding="utf-8").read()

# Generate a word cloud image
wordcloud = WordCloud().generate(text)

# 繪圖
plt.figure()
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

增加 Mask：英文

```
from wordcloud import WordCloud, STOPWORDS
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Read the whole text.
txtfile = "c:/test-wordcloud/cnn.txt"  # 剛才下載存的文字檔
pngfile = "c:/test-wordcloud/cloud.jpg"  # 剛才下載存的底圖
text = open(txtfile,"r",encoding="utf-8").read()
alice_mask = np.array(Image.open(pngfile))

# Generate a word cloud image
wordcloud = WordCloud(background_color="white", mask=alice_mask, con

# 繪圖
plt.figure()
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

三、Python - 知名 Jieba 中文斷詞工具教學（來源：Kenny's Blog)

官方分類法說明:

標籤	含意	標籤	含意	標籤	含意	標籤	含意
n	普通名词	f	方位名词	s	处所名词	t	时间
nr	人名	ns	地名	nt	机构名	nw	作品名
nz	其他专名	v	普通动词	vd	动副词	vn	名动词
a	形容词	ad	副形词	an	名形词	d	副词
m	数量词	q	量词	r	代词	p	介词
c	连词	u	助词	xc	其他虚词	w	标点符号
PER	人名	LOC	地名	ORG	机构名	TIME	时间

Jieba 斷詞主要是结合:

規則斷詞:

主要是透過詞典，在對句子進行斷詞的時候，將句子的每個字與詞典中的詞進行匹配，找到則斷詞，否則無法斷詞。

統計斷詞:

主要是看如果相連的字在不同的文本中出現的次數越多，就推斷這相連的字很可能就是一個詞。因此就可以利用字與字相鄰出现的頻率來做統計。當高於某一個臨界值時，便可認為此字組是一個詞語。

Jieba 斷詞模式

精確模式：將句子最精確的切開，適合文本分析

全模式：把句子中所有的可以成詞的詞語都斷出来，速度非常快。

搜索引擎模式：在精確模式的基礎上，對長的詞語再次切分，提高召回率，適合用於搜索引擎分詞。

操作方式:

透過 jieba.cut () 來進行斷詞，cut_all 參數為 True 的話為全模式，預設為 False，也就是精確模式 jibea.cut_for_search () 是搜索引擎模式 cut ()、cur_for_search () 返回的結構都是一个可迭代的 generator，因此使用 for 迴圈來取得每個斷詞。

四、ImageColorGenerator(來源：API Reference)

class **wordcloud.ImageColorGenerator**(*image, default_color=None*) [source]

Color generator based on a color image.

Generates colors based on an RGB image. A word will be colored using the mean color of the enclosing rectangle in the color image.

After construction, the object acts as a callable that can be passed as color_func to the word cloud constructor or to the recolor method.

Parameters: image : *nd-array, shape (height, width, 3)*

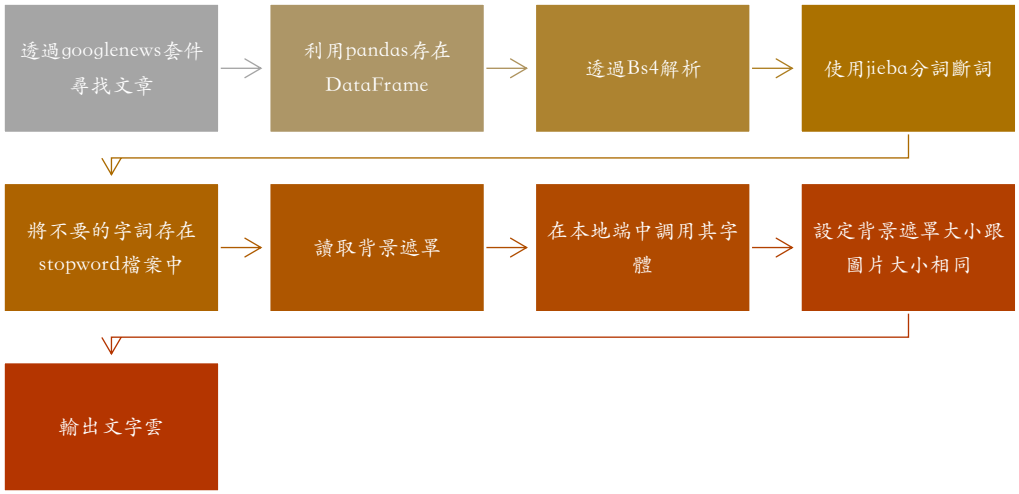
Image to use to generate word colors. Alpha channels are ignored. This should be the same size as the canvas. for the wordcloud.

default_color : *tuple or None, default=None*

Fallback colour to use if the canvas is larger than the image, in the format (r, g, b). If None, raise ValueError instead.

參、研究方法

一、研究架構



二、程式片段說明

一、SSL 用法以及順利連接的主要因素

這行代碼的作用是在 Python 中創建一個默認的 HTTPS 上下文，並且使用未驗證的 SSL 證書進行連接。

通過設置 ssl._create_default_https_context 為 ssl._create_unverified_context，我們告訴 Python 在創建 HTTPS 連接時，使用一個未經驗證的 SSL 上下文。這樣做的效果是，Python 將不再驗證服務器的 SSL 證書，可以繼續連接服務器，但這也帶來了一定的安全風險，因為我們無法確保我們正在連接到的服務器的真實性和安全性。

```
ssl._create_default_https_context = ssl._create_unverified_context
```

二、Googlenews 的查詢套件

1.創建 GoogleNews 的實例對象。

2.設置搜索結果的語言為中文。

3.設置搜索結果的時間範圍為最近一天。

4.設置編碼格式為 UTF-8，以支持中文字符。

5.清除之前的搜索結果和設置，以便進行新的搜索。

6.通過用戶輸入獲取要搜索的關鍵字。

7.使用用戶輸入的關鍵字進行搜索。

8.獲取搜索結果的所有數據。

9.從搜索結果中提取純文本內容。

10.獲取搜索結果中的鏈接。

11.輸出一個空行，用於分隔搜索結果的顯示。

```
googlenews = GoogleNews()

googlenews.setlang('cn')
googlenews.setperiod('d')
googlenews.setencode('utf-8')
googlenews.clear()

x = input("請輸入要搜尋的關鍵字，將為你搜集相關字詞內容:")
googlenews.search(x)

alldata = googlenews.result()
result = googlenews.gettext()
links = googlenews.get_links()

print()
```

三、透過 pandas 套件創建一個 DataFrame

整體而言，這段代碼的目標是將通過 Google 新聞搜索獲取到的新聞內容和鏈接進行打印和整理。通過遍歷列表，可以逐個打印新聞內容和鏈接。然後，使用 Pandas 庫創建一個 DataFrame 對象，將新聞內容和鏈接以表格的形式存儲起來。最後，從 DataFrame 中提取第一個鏈接，並將其打印出來。

```
for n in range(len(result)):
    print(result[n])
    print(links[n])

df = pd.DataFrame(
    {
        '標題': result,
        '連結': links
    })

url = df['連結'][0]

print(url)
```

四、HTTP 請求，用於向服務器傳遞客戶端

在這段代碼中，user_agent 字典中的鍵是 'User-Agent'，值是一個具體的 User-Agent 字符串。這個 User-Agent 字符串描述了一個使用 Chrome 瀏覽器（版本號為 86.0.4240.111）在 Mac OS X 10.15.7 操作系統上的客戶端。該字符串模擬了一個常見的瀏覽器 User-Agent，可以用於向服務器發送請求時偽裝成使用特定瀏覽器和操作系統的訪問者。

使用自定義的 User-Agent 字符串可以用於模擬特定客戶端環境，訪問某些網站可能需要特定的 User-Agent 才能獲得正確的響應或內容。在這種情況下，可以將自定義的 User-Agent 字符串添加到請求頭部中，以達到偽裝的目的。

```
user_agent = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrom

r = requests.get(url, headers=user_agent)
r.encoding = "utf-8"
web_content = r.text
soup = BeautifulSoup(web_content, 'html.parser')
```

五、將所有文本內容連接成一個字符串

這段代碼的目標是從一個名為 articleContent 的列表中獲取每個元素的文本內容，並將所有文本內容連接成一個字符串。通過循環遍歷並提取元素的文本內容，將每個內容添加到列表中，然後使用 '\n'.join(article) 將列表中的內容連接起來，形成一個包含所有元素文本內容的字符串。這樣可以方便地對文本內容進行進一步處理或顯示。

```
article = []
for p in articleContent:
    article.append(p.text)

articleAll = '\n'.join(article)
```

六、jieba 進行分詞斷詞

這段代碼的目標是使用 jieba 庫對中文文本進行分詞處理。通過加載自定義的用戶詞典，可以識別特定的詞彙。然後，對文本進行預處理，去除或替換一些特殊字符。最後，使用 jieba 庫的分詞方法對處理後的文本進行分詞，將其切分為一個個詞語，方便進行後續的文本分析或處理。

```
jieba.load_userdict('/Users/zhonghonghao/googlenews-wordCloud/dict.txt.big.txt')

d = articleAll.replace('!', '').replace('/', "").replace('《', '').replace('》', '').replace('）', '').replace(
    '「', '').replace('」', '').replace('（', '').replace('）', '').replace('！', '').replace('？', '').replace('
                                                                                                    '').
    '▲', '').replace('…', '').replace('：', '')


jieba.setLogLevel(10)

Sentence = jieba.cut_for_search(d)
```

七、詞頻計算

這段代碼的目標是統計經過分詞處理後的文本中每個詞語的出現次數，並輸出結果。通過遍歷分詞後的文本，對每個詞語進行判斷和計數。如果詞語已存在於 terms 字典中，則增加其計數；否則，在 terms 字典中新增該詞語並設置初始計數為 1。最後，使用 Counter(terms) 統計詞語出現次數，並打印結果。

```
terms = {}
for sentence in Sentence:
    if sentence in stopwords:
        continue

    if sentence in terms:
        terms[sentence] += 1
    else:
        terms[sentence] = 1

print(Counter(terms))


artDf = pd.DataFrame.from_dict(terms, orient='index', columns=['詞頻'])
artDf.sort_values(by=['詞頻'], ascending=False)
```

八、生成文字雲

這段代碼的目標是基於給定的文本內容 articleAll，生成一個詞云圖像。通過加載遮罩圖像，並根據圖像的顏色生成器對詞云進行著色。最後，使用 Matplotlib 庫顯示生成的詞云圖像

```
img = "color-0"
img_path = "/Users/zhonghonghao/googlenews-wordCloud/%s.png" % img

mask_color = np.array(Image.open(img_path))
mask_color = mask_color[::3, ::3]
mask_image = mask_color.copy()
mask_image[mask_image.sum(axis=2) == 0] = 255

edges = np.mean([gaussian_gradient_magnitude(mask_color[:, :, i] / 255., 2) for i in range(3)], axis=0)
mask_image[edges > .08] = 255

wc = WordCloud(font_path="/Users/zhonghonghao/Downloads/ThePeakFontBeta_V0_101/ThePeakFontBeta_V0_101.ttf",
               mask=mask_color,
               max_font_size=35,
               max_words=4000,
               stopwords=stopwords,
               margin=0,
               relative_scaling=0,
               )

wc.generate(articleAll)
image_colors = ImageColorGenerator(mask_color)
wc.recolor(color_func=image_colors)

plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.figure(figsize=(25, 25))
plt.show()
```

肆、研究分析與結果

一、系統流程

查詢關鍵字 → 連上網際網路 → 進行關鍵字搜索 → 進行大數據搜索 → 進行斷詞 → 計算詞頻 → 視覺畫出來

二、輸出分析

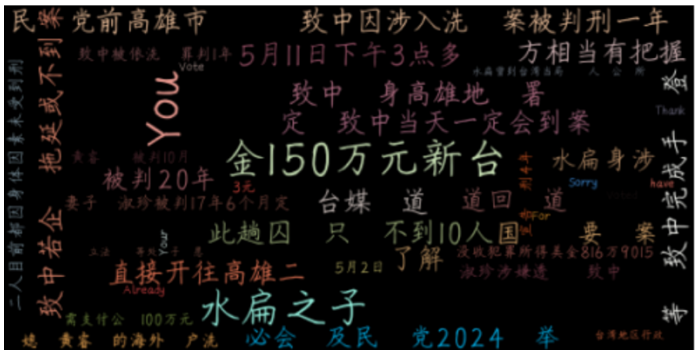
一、馬英九

二、習近平

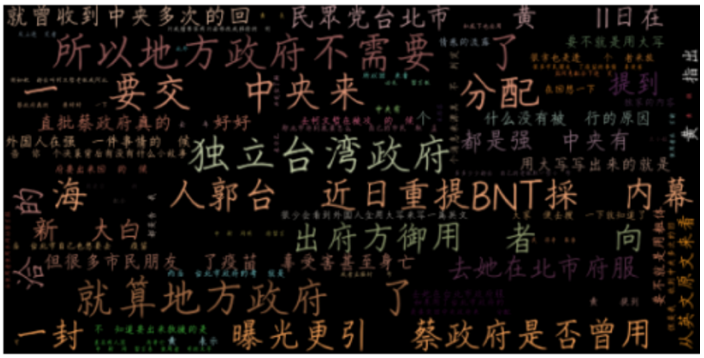
三、普發 6000

四、GoogleAI

五、陳致中



六、柯文哲



伍、研究結論與建議

一、研究結論:

本專題探討了大數據與視覺化技術在網路媒體上的應用，以及其對新聞報導、內容製作、廣告和市場營銷的影響。以下是我們的研究結論:

1. 大數據與視覺化技術能夠幫助記者和編輯更好地進行新聞報導和內容製作。通過分析大量的數據，如社交媒體數據、用戶行為數據等，記者和編輯可以更好地了解讀者的需求和反應，從而優化內容和報導的質量和價值。

2. 大數據與視覺化技術對廣告商和市場營銷人員也有重要作用。通過分析消費者行為和趨勢的數據，廣告商和市場營銷人員可以制定更有效的廣告和市場營銷策略，並更好地了解目標受眾的需求和喜好。

3. 在網路媒體的數據分析和數據可視化方面，大數據與視覺化技術可以幫助記者和編輯更好地了解讀者需求和反應。透過數據分析和可視化工具，他們可以快速且清晰地呈現數據，發現數據中的模式和關係，從而更好地優化內容和報導。

4. 大數據分析的應用還可以幫助預測未來趨勢和做出更明智的決策。通過分析大量的數據，我們可以發現隱藏在其中的複雜模式和關係，從而提前預測未來的趨勢和變化，並做出相應的戰略和決策。

二、研究建議:

基於我們的研究結果，我們提出以下幾點建議:

1. 媒體機構應該加強對大數據和視覺化技術的培訓和應用。培養記者和編輯的數據分析和視覺化能力，使他們能夠更好地利用大數據和視覺化技術來提高報導和內容的質量和價值。

2. 廣告商和市場營銷人員應該重視大數據和視覺化技術在消費者行為和趨勢分析中的應用。投資並利用數據分析和可視化工具, 更好地了解目標受眾的需求和喜好, 從而制定更有效的廣告和市場營銷策略。

3. 網路媒體應該加強數據分析和數據可視化的能力。提供強大的數據分析工具和可視化平台，使記者和編輯能夠更快速、準確地分析數據，發現故事背後的數據，並將其融入報導和內容製作中。

4. 繼續研究和發展大數據分析和視覺化技術。隨著數據的不斷增長和技術的不斷進步，我們需要持續關注新的數據分析方法和視覺化技術，以應對日益複雜和多變的網路媒體環境。

陸、參考文獻

1. Wang, F., Chen, L., & Huang, M. (2016). Big data analytics in online social networks: A comprehensive survey. ACM Computing Surveys (CSUR), 49(1), 1-36. https://dl.acm.org/doi/10.1145/3150226

2. Lee, J. H., & Kim, H. W. (2017). How do social media analytics support crisis decision making? An analysis of crisis tweets during the 2013 Boston Marathon bombings. Government Information Quarterly, 34(3), 442-454. https://www.researchgate.net/publication/310440818_Image_use_in_social_network_communication_A_case_study_of_tweets_on_the_Boston_marathon_bombing

3. Bughin, J., Manyika, J., & Woetzel, J. (2017). A strategy for the era of artificial intelligence. Harvard Business Review, 95(6), 50-59. https://hbr.org/2017/07/the-business-of-artificial-intelligence

4. Liu, Y., Huang, X., An, A., & Yu, X. (2018). Big data analytics in social media: A review. Social Network Analysis and Mining, 8(1), 1-18. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7553883/

5. Kohavi, R., & Provost, F. (2018). Trustworthy online controlled experiments: Five puzzling outcomes explained. ACM SIGKDD Explorations Newsletter, 20(1), 36-41. https://notes.stephenholiday.com/Five-Puzzling-Outcomes.pdf

6. Agrawal, R., & Srikant, R. (2020). Data mining: Concepts and techniques. Elsevier. https://www.sciencedirect.com/science/article/pii/S0957417420306059

7. Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. IEEE Transactions on Knowledge and Data Engineering, 26(1), 97-107. https://ieeexplore.ieee.org/document/6547630

8. Ward, M. O., & Grinstein, G. G. (2015). Interactive data visualization: Foundations, techniques, and applications. CRC Press. https://www.taylorfrancis.com/books/mono/10.1201/b18379/interactive-data-visualization-daniel-keim-matthew-ward-georges-grinstein

9. Cairo, A. (2019). How charts lie: Getting smarter about visual information. WW Norton & Company.

https://www.books.com.tw/products/F016464142

10. Google Search Central Blog 指出：Search Analytics API now supports Discover, Google News, and Regex

https://developers.google.com/search/blog/2021/10/search-analytics-discover-gnews?hl=zh-tw

11. Eric Cheng 指出：如何使用 python 製作文字雲

https://tech.havocfuture.tw/blog/python-wordcloud-jieba

12. Kenny's Blog 指出：Python - 知名 Jieba 中文斷詞工具教學

https://blog.kennycoder.io/2020/02/12/Python-知名 Jieba 中文斷詞工具教學/

13. API Reference 指出：ImageColorGenerator

https://amueller.github.io/word_cloud/references.html