新聞資料搜尋系統架構：以飛航新聞為例

王泰期　陳建良　姜順貿　黃孟琦
國家實驗研究院 國家高速網路與計算中心
taichi@narlabs.org.tw c00cmw00@narlabs.org.tw 1903026@narlabs.org.tw
1803094@narlabs.org.tw

摘要

由於過去針對新聞檢索的網頁，像是 Google news 或是 Bing news，都包含了過多的資訊以致無法收斂企業級用戶所關注的目標新聞，其搜尋的功能也限縮於關鍵字本身，本系統研究針對飛機商情分析的需求，建構了一套新聞檢索的系統架構，包含新聞蒐整、關鍵字提取、特殊數值欄位提取、新聞摘要等等功能，同時也將這些資訊網頁化，可藉由簡易的網頁搜索來達到快速有效的新聞收整，可加速新聞資訊的整理並提供公司有效的應對依據。

關鍵詞：新聞爬蟲、關鍵字提取、文章摘要、新聞搜索系統

Abstract

According to the past news searching web pages, like Google News or Bing News, the web pages contain too much information to capture the target contents. In addition, these searching engines are limited by their searching bars and can only search for the keywords. Due to these reasons, this research focuses on developing a news search system, including parsing and organizing news, extracting keywords and numeric digits of special regular expressions, and news abstract based on the demand for airplane business. In the same time, we also develop web pages for displaying the system we constructed. Based on this system, the researchers can easily search the news and can efficiently organize and provide the most important information for business.

Keywords: news crawler, keyword extraction, text summarization, search system of news.

1. 研究介紹與研究目的

由於目前網路發達資訊眾多，各種不同來源的資訊不同，身為一個持續需要關注產業發展的商情分析人員，每日必須閱讀大量的國際相關新聞，來判斷哪些情報是重要的資訊，相當耗費人力與時間，因此希望建立自動新聞彙整機制，來減少人工彙整新聞與快速判斷資訊重要性，透過文字探勘的技術來蒐整並快速找到重要的新聞對一般公司的商業營運來說至關重要([1]-[2])。但是，一個新聞彙整平台[3]，該收集什麼樣的資訊才能符合不同產業的商情需求呢？身為一個商情分析的人員，需要一個平台提供哪些必要功能呢？而作為一個系統開發的角度來說，系統設計上又該有什麼樣的考量呢？經訪談與了解，對一個商情分析人員來說會遇到以下幾個問題，(1) 通常文章來源眾多，需要一一閱讀文章才知道重點文章(2) 真正需要看的文章數量不多，需要有良好的篩選機制與介面 (3) 文章裡的文字過多，重要句子或關鍵字不多，卻需要完整的閱讀才能判斷重要性。為了優先解決這幾個問題，團隊設計了一套做文字探勘的工具與介面，來提供快速的文字資訊彙整，藉此減少相關人員的消耗，又同時能夠以 AI 與大數據的技術來提供如專業人員般的判斷，透過良好設計的資料檢視與搜尋介面，讓相關使用資料的人員可以快速的找到關注的新聞來做為製作報表與報告重要的訊息來源。

其中過去主流的媒體新聞，通常只有新聞類別和關鍵字的搜尋[4]，如大多數人常用的 Google news 或是 Bing news，本研究主要提供了原始資料的加值，包含透過專家字典設定關鍵字的功能，因為商業研究上需要擷取新聞提及的交易金額，快速進行文章摘要的處理等等技術，我們嘗試在這個新聞彙整平台上，架構這些文章處理技術，做為未來文章彙整平台的重要功能。

2. 資料來源與資料彙整

為達成使用者的需求，最重要的項目與難處就在於資料的彙整，不同領域的人對新聞的需求也大有不同，不同標的的專業用詞差異也相當大，因此資料的收整與使用者的新增詞彙會決定一個新聞探勘系統的效能與正確度。

2.1 新聞爬文

確定新聞來源是做新聞彙整最重要的第一個步驟，當資料限縮的條件下，就能夠大幅提升目標資料搜尋的準確度，因此在本研究的案例中，以飛航相關的商業情資做資料探索，設計上由於需要快速的進行爬文與資料彙整，因此採用 AWS 的 Lambda Service 進行平行化的爬文，基礎爬文機制如圖1所示

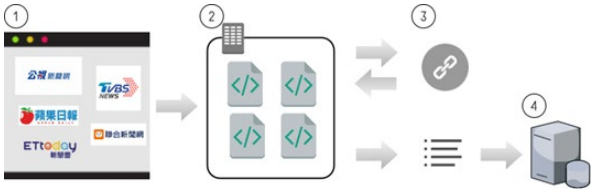

圖 1 基本爬蟲架構

step1: 獲得網站網址
step2:取得網址的 HTML
step3:對網址的 HTML 進行解析

step3.1 解析結果若為網址，則重回步驟二
step3.2 解析結果若為新聞內容，執行步驟四
step4:將資料儲存進本地資料庫

2.2 新聞主要欄位資訊

選定新聞做為這個系統的原始資料來源的原因是新聞是屬於專業的文章，有固定且為人所知的基本架構，而且國際也有類似的標準，通常一則新聞包含新聞來源、新聞類別、作者、發表時間、本文、原始新聞網址等等資訊，藉由這樣的資訊我們得以跨新聞平台與類別進行資料彙整，也能夠提供基本的搜尋介面。

2.3 專業字典

執行新聞彙整的專案中，很重要的元素就是專業知識的整合，對系統開發人員來說，不可能知道所有領域的專業知識，因此彙整專業知識的來源就是透過字典的資訊，透過專家提供的字典資訊可以有效的萃取關鍵字，並將這些關鍵字做為系統中重要的搜尋資訊，這次的研究中，我們使用的是飛航的新聞，因此彙整了部分的飛機專有名詞，譬如 Boeing 747 MAX、Airbus 等等特殊的用詞，還有對應的飛航公司與工廠等等資訊，藉此可以快速搜尋到競爭公司或是上下游廠商的相關訊息，可以快速在產品線上與經營策略上做應對與調整。

2.4 文章金額提取

對一個商業新聞彙整的重點來說，文章內牽涉到的交易金額是重要的一環，因此特別萃取了文章內提及的金額數字，雖然看起來不容易，實際檢視了新聞的內容後，發現一般新聞使用金額的表示是有跡可循的，特別容易以金錢符號開始或結束，譬如美金符號$與歐元符號€，可以針對這些符號進行數值提取，詳細方法請參閱3.1節。

2.5 新聞摘要

由於新聞內容有時候很長，對於新聞閱讀人員來說，通常必須花一些時間來整理該文章的關鍵內容，相當耗時耗力，因此若能針對本文的內容做快速的重點整理，會協助減少重要文章判讀所需花的時間，也加速商情人員做文章的彙整，因此本研究導入文章摘要的分析功能，進行文章摘要處理，詳細作法請參閱3.2。

2.6 資料庫

系統使用 mysql 8.0.20。為了減少 query 資料時用過多的 join，除了關鍵字，查詢的新聞資料皆儲存於同一表格之內(請參見表1)。

表 1 查詢的新聞資料欄位

欄位	類型	說明
nid	int	編號 自動遞增
dt_publish	varchar(4096)	新聞發佈時間
sys_time	datetime	系統爬蟲時間
title	longtext	標題
content	longtext	文章
author	longtext	作者
url	longtext	網址來源
img	longtext	內文圖片網址
Source	varchar(4096)	報社來源
category	varchar(4096)	新聞分類
key_sentence	longtext	摘要
star	tinyint [0]	標記重要與否
updated_at	datetime	更新時間
predict	int [0]	預測
display	int [0]	是否顯示
money_i1	int	1000萬以下
money_i2	int	1000萬至1億
money_i3	int	1億以上

其中，新聞發佈時間 dt_publish 因文章時區有所不同，所以直接依據文章上的時間作抓取，並記錄當下爬取的時間 sys_time。摘要 key_sentence 的部份抓取文章中判斷五個最重要的句子，並在展示原始文章時，反白表示。money_i1 標記為1000萬以下、money_i2標記為1000萬至1億和 money_i3 1億以上，一篇文章中可能包含多個金額，因此開三個欄位分別做標註，只要文章中有包含其中一個金額，就符合條件。這裡利用開三個欄位的方式，而不另開一張表格儲存文章內所有的數字，主要是為先行判斷數字是為金額和幣值先將解析文章內所有數字，搜尋特定關鍵字$ 、million..等。如：16 million 或 $ 2 billion。判斷數字的大小及類型。最後在回寫到 新聞類型表格。這不用做正規化的處理，主要是為了加快使用者查詢的時間(請參見表2之欄位對應)。

表 2 查詢的金額資料表格

欄位	類型	說明
aid	int	流水編號
nid	int	新聞編號
keyword	varchar(4096)	價格關鍵字
number	float NULL	關鍵字數字部份
unit	varchar	單位
amount	float	轉成統一數字
dict_name	varchar	數

由於所有新聞的標籤數目不同，不能用查詢金額的方式處理，因此還是必須以正規化方式切出獨立表格，儲存標籤(如表3)。本文處理查詢方式可分成兩種，一種是先查詢新聞資料，然後再

查詢標籤，一種是以標籤為查詢條件，新聞資料必須Join標籤，找出符合的標籤，然後再查詢新聞。因為索引條件清楚，不會產生額外的查詢時間及不需要的資料。

表 3 查詢的標籤表格

欄位	類型	說明
aid	bigint	流水編號
nid	bigint	新聞編號
keyword	text	關鍵字
dict_name	text	關鍵字類別

3. 研究方法

本文主要有兩項需解決的任務，分別是自動化文章摘要(automatic text summarization)與關鍵資訊擷取，以下分開介紹，下面先介紹關鍵資訊提取，再介紹自動化文章摘要的方式。

3.1 關鍵資訊提取

一般文章的重點資訊，離不開人事時地物，而資訊的提取，可分為兩種，一種是簡單，但明確的關鍵字萃取，有無法取代且表示鮮明的關鍵字出現在文章，看到關鍵字就可以了解文章部分傳達訊息、主題，一般來說是專業用語，例如:HIV(Human immunodeficiency virus)、Boeing、aerospace 等，另一種是關鍵短語，透過專屬動詞與名詞搭配，能獲得更明確的資訊，例如:Vistara **signs** $2.4 billion engine and maintenance **contract** with CFM。透過合作夥伴在航空業的專業知識，羅列相關專有名詞與可能使用動詞的字典，利用前處理與斷句後，判斷各專有名詞、動詞是否有出現的方式，得到最終的提取資訊。除了列出的重要關鍵字和句型以外，另外對金錢和數量也做了相關的抓取，因為某公司訂多少架、什麼機型的飛機，或是什麼公司簽了多少金額的合約，買了哪家公司多少金額的機型，對合作夥伴來說，皆是文章中至關重要的資訊，下面將詳述其具體過程，與處理方式，得到最終的結果。

text preprocessing

文字探勘中，大部分皆會做前處理，去除雜訊，讓精準度提高，更多前處理方法可以參考Vijayarani, Ilamathi, and Nithya (2015)[5]。

lowercase or uppercase

我們需要統一英文的大小寫表示，在電腦判讀會認為Text和text是不同的單字，需要讓電腦知道這些字代表的意義相同。

remove number or replace converting words into number

阿拉伯數字與英文數字只是表示法不同，意義是相同的，如若沒有轉換，電腦會認為是不同的單字，準確度會下降，例如:2 和 two 意義是等價。如果數字在目前任務並不重要，可以刪除，讓變數減少，更為精簡需要的資訊。

removing stop words

停止詞(stop words)代表非常常用的單字，像是"a","the","an","is","on",…。這些字並沒有太多意義，容易成為雜訊，混淆真正重要單字，所以需刪除停止詞。

Stemming or Lemmatization

Stemming和Lemmatization兩個方法皆是減少單字變異性的方式。Stemming 透過找尋單詞的詞根(root)，還原單詞，有可能產生不在英文辭典裡的單字，例如:amusing、amusement 皆還原成 amus。而lemmatization透過辭典與形態分析，解析單詞的結尾，還原成字典的原型。

Stemming 使用上較為自由，不需要額外的字典，能快速適用於新語言，而lemmatization需要有充足有效的辭典，才有較好的效果。本次研究因為是英語語料，已經有充足有效的辭典，使用lemmatization 去做單詞還原，會有更好的效果。

具體流程

本研究兩種功能皆有建立，第一種透過合作夥伴給的專業字典，拆分成不同的字典，如國家、公司、時間等，接下來依照字典的重要性順序，刪除相同的用字，避免重複在網頁上顯示標籤(tag)。關鍵字提取，值得注意的是不對文章做任何前處理，因為大小寫本身有其代表特別意義，通常為專有名詞(例如:Apple)，透過斷詞，假如文章有出現字典中的詞，則記錄下來。使用者可以在網頁上點選感興趣的關鍵字(tag)，找到其他文章包含此關鍵字。第二種則是透過大量文章，找尋重要且獨特的動詞，以及合作夥伴提供感興趣的名詞辭典，對文章進行前處理，還原各詞性後，斷句。各句依照各情況採取不同的動作:

(i)該句沒有重要動詞→跳過

(ii)該句有重要動詞，但沒有重要名詞→跳過

(iii)該句有重要動詞與名詞→解析

本研究使用的資料為航空業新聞，與合作夥伴討論後，客戶希望能找出獲得訂單、取消訂單與交貨等相關資訊，當取消(cancel、scrap、lose…)動詞與重要名詞(Boeing 747,A320neo,…)同時出現，此句有極大可能是在描述這個機型(或公司)被取消了多少訂單。此外為了避免抓錯數量，會先將年份數字取代掉(1900~2100，史上最大一筆訂單430架飛機，離千還有一段距離，較不會誤刪)與刪除百分比等非架數或金錢的數字。抓取金錢分成兩部分，分別是金錢符號如:$、€、¥等為第一部分，並記錄兌換美金的匯率，可用近期的平均匯率做代表。第二部分則是金錢的英文或縮寫如:million、billion、bln 等。文章內出現金錢符合以下格式，

皆會被記錄。
(i)第一部分+數字(如:$ 500)
(ii)數字+第二部分(5 million)
(iii)第一部分+數字+第二部分($500 million)
缺少的預設值，第一部分為美金($)，第二部分為dollar。
最後透過匯率轉換，得到文章描述的粗估美金金額，如$1 mln=1*10^6、€10 ≈ 1.1473 * 10 * 1 = 11.473(以20200701~20200730平均匯率舉例)。

3.2 自動化文章摘要

如何做出自動化文章摘要的主要想法是來自R. Mihalcea and Tarau (2004) [6]，此論文透過計算文章兩兩句子共同出現單字的頻率，得到句子之間的相似度，再用多種 graph-based ranking [7-8] 方式得到句子的重要性評分，依分數高低排序，取出前 N 句，得到最終的結果。而與本文主要差別在於，使用 Mikolov et al. (2013)提出的單字向量化(word to vector)的方式，將句子向量化，用餘弦距離計算出兩兩句子的相似度，取代共同詞頻，以期能得到更精準的句子間相似程度。下面簡單介紹使用到的方法與詳述具體流程。

word2vec

Word2vec[9]是2013年 Google 公司釋出的開源演算法，除了能讓單詞投影到向量空間外，同時保留詞與詞之間的"相似性"，例如:

倫敦之於英國 <-> 北京之於中國
男人之於女人 <-> 國王之於皇后

有別於詞袋(bag of word)方式，會考量字詞的順序性，會透過加權的方式，考慮鄰近使用的字詞，如果兩個字詞附近的字都很相近的話，代表兩個字本身的相似程度高，向量空間表示就會相近。訓練過程中就會不斷調整字詞之間的權重，最後得到的權重，就是 word2vec 的值。

textrank

Textrank[6]的核心來自 Brin and Page (1998)[10]提出的 pagerank 演算法。pagerank 演算法原應用在搜尋引擎，是透過推薦或投票的方式，得到排序。當網址被其他網站推薦(內文有原網址超連結)，就會得到推薦網站的重要性分數。得到越多重要的網站推薦，分數越高，此外還考量網站本身內文提到超連結的個數，超連結越少，每個連結分到的重要性分數就越高。經過不斷的迭代，得到的綜合推薦評分，即代表這網站的重要性有多高。

為了讓文字能運用在 graph-based algorithm，必須建造文字之間的圖形關聯圖，邊與邊是共同出現的文字或其他有意義的關聯性。根據不同的使用場景，使用不同的文字單元(單字、句子…)。Mihalcea and Tarau (2004)定義句子是節點，兩兩之間的關聯性為方程式(1)，即可使用 pagerank 計算各句的重要性，轉成 textrank 演算法。

Similarity and cosine similarity

$$\text{similarity}(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)} \quad (1)$$

這裡S_i=$(w_{i0}, w_{i1}, \dots, w_{in})$與$S_j$=$(w_{j0}, w_{j1}, \dots, w_{jn})$分別代表文本第 i 個句子與第 j 個句子，$w_k$代表詞彙庫第 k 個單字，詞彙庫是代表全部文檔經過斷詞後有出現的所有單字集合，全共 n 個單詞。

cosine similarity 是其中一種文字分類最常用的距離計算方式，本文使用 cosine similarity 計算兩兩句子的距離。公式定義如下:

$$\cos(\vec{V_i}, \vec{V_j}) = \frac{\vec{V_i} \cdot \vec{V_j}}{|\vec{V_i}||\vec{V_j}|} = \frac{\sum_{t=0}^{n} v_{it} * v_{jt}}{\sqrt{\sum_{t=0}^{n} (v_{it}{}^2) * \sum_{t=0}^{n} (v_{jt}{}^2)}} \quad (2)$$

這裡V_i=$(v_{i0}, v_{i1}, \dots, v_{in})$與$V_j$=$(v_{j0}, v_{j1}, \dots, v_{jn})$分別代表兩個 n 維的向量。

具體流程

在時間與資料皆有限的情況與缺少專家標註摘要的前提下，選擇使用無監督式演算法，得到最終摘要。首先對收集的所有文章做前處理，包括英文統一小寫，去除停止詞，詞形還原(lemmatization)，刪除所有詞頻小於某個 threshold 的字詞等。之後用 python 的 glove 套件[9]對清理過後的文章做 word2vec 的訓練，得到各單詞對應的詞性量，並記錄各單詞與對應的詞向量值，如表4。

表 4 訓練 word2vec 模型後，單詞與對應的詞向量示意表。V_i代表詞向量的第 i 維

word	V_1	V_2	……	V_{100}
worry	0.4	5	……	1
care	0.36	0.44	……	0.79
home	0	0	……	-0.3
⋮	⋮	⋮	⋮	⋮
sleepy	-0.01	0.005	……	-0.001

輸入一篇文章時，對文章做前處理後，使用 python nltk 套件中的 sent_tokenize 做文章斷句，使用上述紀錄詞向量的表格，取各句各單詞對應的詞向量平均作為句子代表向量。接下來用 cosine similarity(方程式(2))計算各句的相似度。最後各句當作節點，節點互相的連結是相似度的值，就能使用 Textrank，得到最終各句的重要性，取出前五高的值，依照原始句子出現的順序，組成摘要，即為最終結果。

4. 系統展示:以飛航相關新聞為例

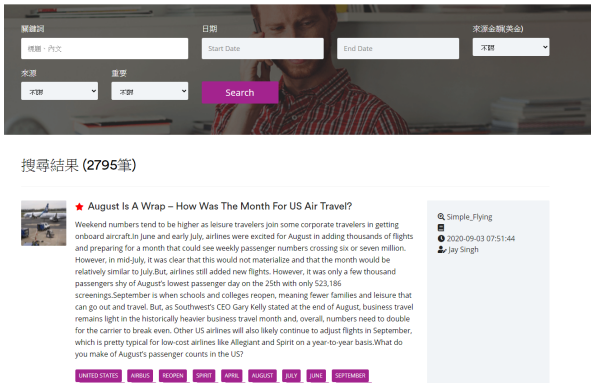

圖2. 基本網頁搜尋介面

基本搜尋圖示如圖2所示，在整體的搜尋部分主要提供搜尋關鍵字、新聞開始與結束時間、內文所提到的金額、資料來源、是否為重要新聞等等欄位進行搜尋，基本文章列表部分，可顯示所有篩選後的筆數，來源、發佈時間、作者等資訊列於右側，文章區塊列上文章標題、內文摘要、關鍵字、縮圖、與重要性標示。

本文的部份，從標題的部分開啟連結與內文，內文將展示除了摘要以外的本文，本文中的關鍵句增添底色用於標示重要句，最後可再由標題的部分連結到原始資料來源。

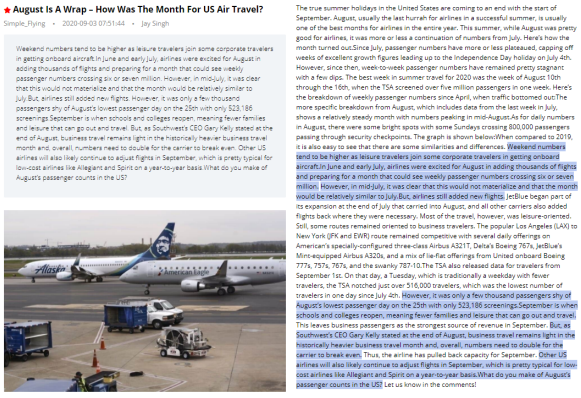

圖3. 文章全文展示

5. 結論與未來規劃

整體規劃來說，有別於單一來源新聞網站的展示，作為一個客製化文章匯集的網站，重點將放在專業知識的匯集，需要了解產業的需求與關鍵字，從中簡化資料庫設計與網頁搜尋架構，本研究可作為文字收整平台的草案，再根據不同需求來做調整。

另外，文字探勘的需求通常會延續到資料分類的需求[11]，唯本測試案由於起步較慢，資料量較少也缺乏人力標記，因此辨識能力不夠完善，另外也表示待解決的問題是，當資料量不足的情況如何提升分類的準確率問題，也是未來相關研究必須要討論的重點；在文章資料收集足夠之後，可能將會規劃語音與文字辨識的需求，也是主要發展路線之一[12]；同時可以預期到的是，隨著時間的累積，資料量的膨脹可能是非常迅速的，資料庫的結構與格式也會面臨到挑戰[13-14]，也將是我們未來研究的重點。

參考文獻

[1] Z. Ma, O. R. Sheng and G. Pant, "Discovering company revenue relations from news: A network approach." Decision Support Systems, Vol.47, No.4, pp.408-414, 2009.

[2] Z. Ma, G. Pant and O. R. Sheng, "Network-based Approach to Mining Competitor Relationships from Online News." International Conference on Information Systems, 2009.

[3] 王乙涵，新聞策展平台內容產製研究－以「關鍵評論網專題」為例，世新大學，台北市，2017.

[4] D. Giomelakis and A. Veglis. "Investigating search engine optimization factors in media websites: The case of Greece." Digital journalism, vol. 4.3, pp. 379-400, 2016.

[5] S. Vijayarani, J. Ilamathi, and M. Nithya. "Preprocessing techniques for text mining-an overview." International Journal of Computer Science & Communication Networks, Vol. 5.1, pp. 7-16, 2015.

[6] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411, 2004.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality." In NIPS, pp. 3111–3119, 2013.

[8] R. Mihalcea, "Graph-based ranking algorithms for sentence extractin, applied to text summarization," In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL 2004) (companion volume), Barcelona, Spain, 2004.

[9] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12, 2014.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, The Pagerank Citation Ranking: Bringing Order to the web, technical report, Stanford University, Stanford, CA, 1998.

[11] P. Han, S. Shen, D. Wang, Y. Liu, "The influence of word normalization in English document clustering," In Proceedings of IEEE International Conference on Computer Science and Automation Engineering (CSAE 2012), pp. 116–120, 2012.

[12] D. Jurafsky, and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition. Prentice Hall, 2008.

[13] A. H. Tan. "Text mining: The state of the art and the challenges," In: Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases. sn, pp. 65-70, 1999.

[14] K. L. Sumathy, and M. Chidambaram, "Text mining: concepts, applications, tools and issues-an overview," International Journal of Computer Applications, 80(4), 2013.