第二十三屆全國資訊安全會議(CISC 2013) Cryptology and Information Security Conference 2013

媽媽臉書過濾器
Facebook Privacy Filter

邱奕斌[1] 蘇庭昱[2] 許晏峻[3] 蔡宗翰[4] 吳育松[5]
安全與系統實驗室，國立交通大學資訊工程系
t3706408@gmail.com[1] a1060229@hotmail.com[2] head2568.cs96@g2.nctu.edu.tw[3]
smartPG@gmail.com[4] hankwu@g2.nctu.edu.tw[5]

摘 要

Facebook（臉書）是目前最熱門的社群網站之一，使用者可於其上發布自己想法或是想說的話抒發情緒，藉由塗鴉牆上的互動擴展社交，使得臉書至今已成為與人溝通交流的一個重要管道。然而在這樣廣大的社群網路之中，使用者有時在不注意的情況下會發布一些不適合被所有好友觀看的特定訊息，而造成個人隱私被侵犯的疑慮或是產生不必要的誤會。在本研究中，我們利用自然語言處理與資料探勘等技術，搭配臉書訊息存取控制機制研發出了「媽媽臉書過濾器」來保護臉書發文的隱私性，在操作上會自動對訊息進行分類以及權限設定，且系統是實作於發文者端因此對於觀看者端的系統環境沒有特別的要求與限制。

1. 研究簡介

臉書是目前最熱門的社群網站之一，擁有廣大的使用群，其最大特色為「塗鴉牆」與「動態消息」，使用者可以隨時發佈自己想法或是想說的話，所發佈的資訊即為動態消息且會顯示在自己塗鴉牆上，而擁有查看此動態權限的使用者即可即時查看並做回應等等，因此成為了許多人抒發情緒與擴展社交的媒介，而隨著 Facebook 的蓬勃發展，雖然使我們多了一個與人溝通交流的管道，但也卻衍生出了不少問題，而我們的研究即是針對隱私方面實作出一系統防止問題發生。

媽媽臉書過濾器(Facebook Privacy Filter)是一套結合發文者端瀏覽器插件以及伺服器端訊息內容分析機制的過濾系統，使用者透過 Browser Extension 發布 Facebook 塗鴉牆訊息，該系統能將使用者輸入的訊息加以辨識並分類，藉由訊息的分類類別給予適合的群組閱讀的權限，這些合適的群組可以閱讀使用者輸入的原始訊息，而其他未被授權的群組，系統會產生一則較為正常的訊息給這些群組閱讀，此 Browser Extension 只有在發文的時候才需要用到，觀看者並不需要安裝任何特殊元件，所以對方並不會注意到使用者是透過此系統來發文。

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

2. 研究動機

在現實社會中已經出現過許多使用臉書時所造成的個人隱私疑慮的相關事件，像是新聞報導過美國一些雇主在應徵員工時，會查閱應徵者的臉書帳號，雖然是想藉由臉書瞭解應徵者的生活與個性，但仍然產生了一些隱私上的爭議[1]。而對於我們這些一般青少年來說，此種類似情形也會發生在我們身上，不同的是，我們所不希望涉入我們生活的對象卻是：我們的父母，前一陣子 YouTube 上有個影片(圖 1)以十分搞笑的方式呈現出此問題的解決之道並且造成廣大迴響，此解決之道即是:「媽媽臉書過濾器」，影片中構思出此種工具可以將使用者所發佈的訊息做過濾，使得愛子心切、對於小孩生活細節處處均要插一手的媽媽所查看到的不是我們原始動態或是圖片，而是另一截然不同且適合給媽媽看到的訊息，雖然此影片目的主要是取悅觀眾，但這概念卻讓我們覺得是個很好的動機，因此我們想真正將之實作出來，並增強其功能性，且這種情形不只是發生在父母與小孩、雇主與員工之間，朋友之間也會發生，所以我們想如果影片中的過濾器能被更廣泛的實作出來，那我們在網路上的隱私將能獲得更多的保障。

圖 1 媽媽臉書過濾器[2]

3. 設計訴求以及核心議題

我們在系統設計上希望能做到隱密性、便利性和修改彈性三方面。隱密性是指使用者透過此系統發文時，訊息可以受到保護，避免一些訊息給不適當的人所看見，而且系統是只有發文者才須安裝，

這樣也可避免其他人提前發現此系統的存在。便利性是希望使用者在發文的時候不需要有太多的準備，只需要將想要發的訊息打好，我們的系統便會經由人工智慧自行判斷訊息與群組的對應關係，並且將原訊息和修改樣本訊息設定好適合的群組自動發布到 Facebook 塗鴉牆上，而不需要使用者去決定哪些人可以看到原始訊息。修改彈性則是指當使用者某天希望讓某些人閱讀訊息的權限升高或降低時，不需要將過去發布的訊息一個個作權限的修改，只需要修改群組內好友名單即可完成，減少發布訊息後若要再做修改的麻煩。

在隱密性和便利性上面，有一個共通必須要完成的挑戰就是判斷訊息是屬於哪種概念，我們目前將訊息分成六種不同的概念，每一種概念都有各自和群組的對應關係，為了要能由系統自動判斷訊息所屬的觀念，避免不適合的人看到使用者發布的訊息，這是我們在這整個系統中最主要要處理的問題。

4. 系統設計

使用者在使用過濾系統前，須先在個人 Facebook 上建立四個群組，分別是”Bosom Friends”、”Normal Friends”、”Lover”以及”Parents”。在使用者發布動態後，系統會先判斷適合觀看訊息的群組有哪些，再產生出一個訊息副本給其他的群組，且**此機制並非建立在查看者端而是在發佈動態者端實作**，因此**不管查看者用哪台電腦或是哪個瀏覽器皆只能看到我們所決定給他查看的訊息**。例如使用者發布一則訊息包含一些情緒字眼或是不雅字樣，系統判定此則訊息只適合給”Bosom Friends”群組裡的好友看到，而對於另外的群組，系統會產生一個一般化訊息供他們查看，因此使用者不必擔心發出的訊息是否會冒犯到他人。

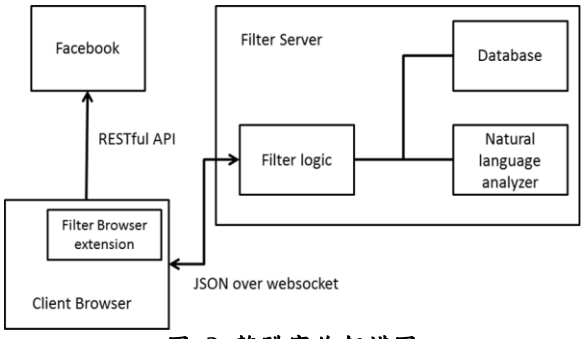

圖 2 整體實作架構圖

上圖圖 2 是我們過濾系統的架構，使用者在 client 端透過 extension 發出的訊息會先被送到 server 端做處理，送進來的訊息會先交給自然語言分析器作分析，判斷訊息屬於哪個概念，如此便可猜測哪些群組適合觀看此訊息，再從樣本資料庫中隨機取出替代訊息給其他不適合觀看原始訊息的群組觀看，處理完後的訊息會送回 client 端，經由 extension 設定好訊息與群組的對應關係後，再發布至使用者 Facebook 塗鴉牆，以下詳細說明各元件的功能：

4.1 Filter Browser Extension

我們將 client 端透過 Chrome extension[3]實作，extension 主要提供了使用者介面進行發布動態訊息的動作，並且將動態訊息以 WebSocket[4]的方式傳送給 Filter Server，等待 server 回傳處理完的訊息與其對應的群組資料，再以 Facebook API[5]依照這些群組設定將訊息發布至使用者塗鴉牆上。

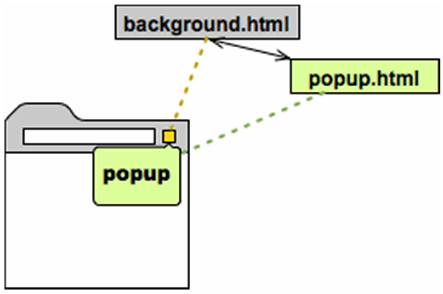

圖 3 Chrome Extension UI pages

Client 端的 extension 包含 popup 和 background 兩個模組，如圖 3。popup 頁面如圖 4，主要提供發布動態訊息的介面，操作就如同在 Facebook 網頁上發布動態訊息。而 background 頁面則為背景程式，用於與後端的 Filter Server 溝通。

圖 4 Filter Browser Extension Popup 頁面

4.1.1 Popup 頁面

Popup 頁面提供了使用者發文的介面，我們運用 localStorage[6]來和 background 程式共用資訊，包括 Facebook API 的 access token、群組名單還有一些 extension 的設定，並且藉由 Chrome API 來通知 background 處理發文的訊息。

在介面上，我們也提供包含連結的動態訊息，就如同 Facebook 發文介面一般，經 Javascript 知道使用者輸入了網址的內容，以正規表達式抓出

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

了網址部份，再藉由 Facebook API 把網址傳送給 Facebook 取得網址內容資訊，最後模擬 Facebook 頁面的呈現方式提供給使用者，如圖 4 右方所示。在 Popup 頁面中的”*自動重整 Facebook 頁面*”選項提供成功發布之後自動重新載入 Facebook 頁面的功能。”*開啟發文過濾*”選項則是提供過濾功能，不勾選即等同於一般發文。當按下”留言”時，整個動態訊息將會利用 Chrome API 由 popup 頁面傳送至 background，接著由 background 來負責處理這些訊息。

4.1.2 Background 頁面

Background 為 extension 的背景程式，由數個事件(在流程圖中，以灰色方塊表示)構成來負責各種工作包括取得 Facebook API 的 access token、和 Filter Server 進行連線溝通、取得群組資料、發布動態訊息至 Facebook 頁面等。

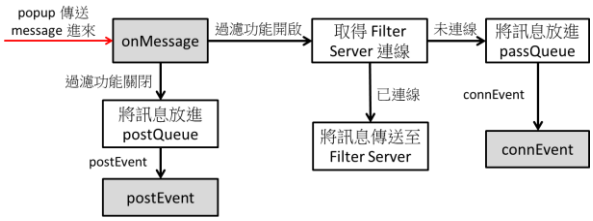

圖 5 Background 收到來自 popup 的訊息

當使用者從 popup 做出發文動作時，將會傳送訊息到 background 而觸發 Chrome 的 onMessage 事件，如圖 5。這裡依照過濾功能的關閉和開啟，分別做直接發布動態訊息至 Facebook(觸發事件 postEvent)和將訊息傳送到 Filter Server 做處理。

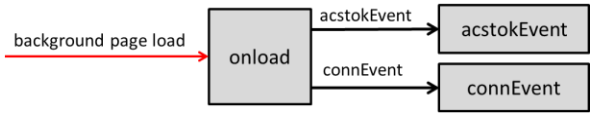

圖 6 Background page onload event

Background 的初始部分，如圖 6，當 extension 第一次啟用時會載入 background 頁面，其後會觸發兩個事件：acstokEvent(流程如圖 7)和 connEvent(流程如圖 8)，分別完成取得 access token 和連線至 Filter Server 的動作。

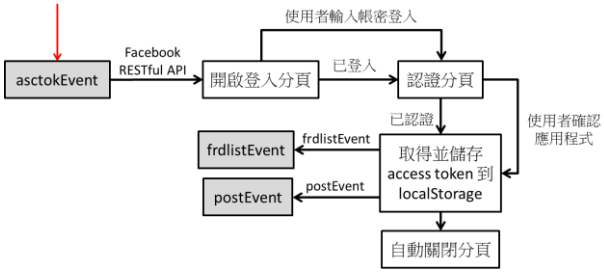

圖 7 acstokEvent 流程

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

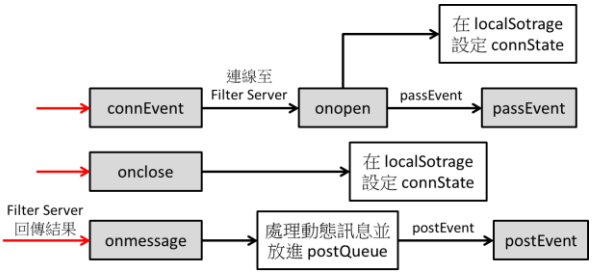

圖 8 connEvent 流程和 WebSocket 相關事件處理

取得 access token 後，會觸發 frdlistEvent(圖 9)來和 Facebook 取得群組，並且將群組資訊儲存進 localStorage，方便之後發布動態訊息至特定群組。

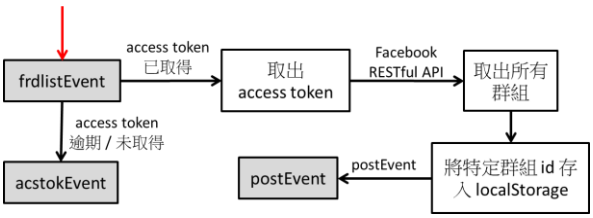

圖 9 frdlistEvent 流程

圖 10 是和 Filter Server 相關的事件，當連線建立或重新建立後將會觸發 passEvent 事件(圖 10)，目的在於將未處理之訊息傳送到 Filter Server 處理。

當 Filter Server 回傳動態訊息過濾後的結果，將會觸發 WebSocket 的 onmessage 事件，觸發後會將這些結果放進預備發布至 Facebook 中的佇列(postQueue)，再觸發 postEvent 事件(如圖 11)進行發布。

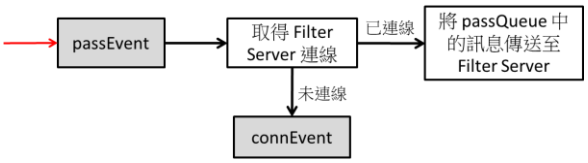

圖 10 passEvent 流程

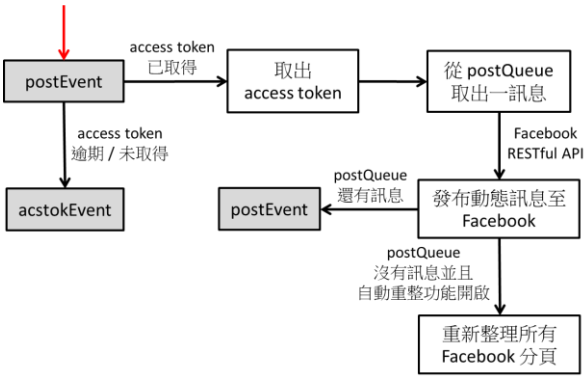

圖 11 postEvent 流程

4.1.3 訊息格式

Popup 和 background 間的溝通、background 和 Filter Server 溝通的訊息格式我們都使用原生 JavaScript 就能夠處理的 JSON[7]；而目前訊息尚未經過加密，往後考慮到資訊安全議題亦可由加密的方式進行傳送。

表 1. Background 傳至 Filter Server 之訊息格式

```
{
msg: 動態訊息內容,
link: {
      url: 連結網址,
      name: 連結名稱,
      caption: 連結標題,
      description: 連結描述
     },
}
```

表 2. Server 回傳至 background 的結果格式

```
[
      {
      msg: 動態訊息內容,
      link: 連結物件(同 request 的 link 格式),
      group: ['群組 1', '群組 2', ……]
      },
      {
      格式同上物件
      },
      ……
]
```

Browser extension background 頁面傳送至 Filter Server 的訊息格式如表 1 所示。其中 link 欄位由 Facebook API 提供。

而由 Filter Server 回傳至 background 的分析結果訊息格式如表 2 所示。最外面是一個陣列，其中每個元素都是一個文章的物件，代表要發出去的文章，每個文章物件比 request 多了一個 group 欄位，group 是一個由群組名稱組組成的陣列。

4.2 Filter Server

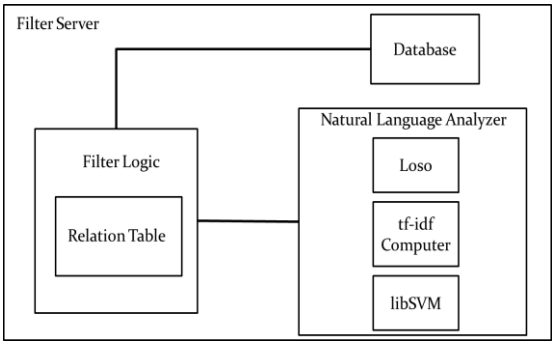

圖 12 Filter Server 內部架構圖

Filter server 內部主要分成三個部分(圖 12)，第一個部分是 filter logic，filter logic 主要的工作是當後面的自然語言分析器分析完訊息後，利用 relation table 決定可觀看訊息的群組。Natural language analyzer 的內部有三個功能，分別是 Loso[8]——作中文分詞、tf-idf[9] computer——計算詞組特徵值、libSVM[10]——判斷訊息所屬的語意概念。最後還有一個 database，此樣本資料庫是儲存替換訊息供 filter logic 選取。

4.2.1 Filter logic

這部分為整個 filter server 運作的主要流程核心。起初當使用者發佈訊息時，server 端將接收 browser extension 所取得的訊息，再交由 natural language analyzer 做訊息分類，對於分類的結果給予適合的群組觀看的權限，而對於其他未授權的群組，系統會從樣本訊息資料庫去隨機選取一則訊息供其觀看，再將原始訊息和樣本訊息以及其分別對應的群組關係傳回給 client 端做發布的動作。另外如果使用者發布的訊息為一多媒體連結，我們可以接收到連結的一些訊息，如網址和網站名稱及縮圖等資訊，也能納入分析的範圍內。

4.2.2 Natural Language Analyzer

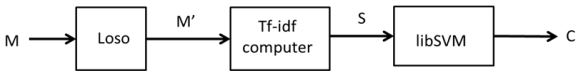

圖 13 Natural Language Analyzer 運作流程

上圖是 Natural language analyzer 的整體運作流程，其中用到的一些元件稍後會做介紹。在此我們先大致介紹上圖的運作流程，extension 端傳送訊息 M 到 server 的 Natural language analyzer，analyzer 將輸入的訊息透過 Loso 作分詞，得到分詞後的結果 M'，算出各詞組的 tf-idf 值，集合成一個特殊的格式 S 再傳給 libSVM，經由 libSVM 做 predict 判斷該訊息所屬的語意概念 C，有了語意概念 C 就能決定該訊息所能被觀看到的群組權限，如下圖圖 14 所示。

語意概念	Normal Friends	Bosom Friends	Parents	Lover
一般訊息	○	○	○	○
情話		○		○
吃喝玩樂	○	○		○
購物敗家	○	○		○
情緒字眼		○		
情色內容		○		

圖 14 群組與語意概念對應表

◆ **Loso**

Loso 是一個用 python 實作出來的中文分詞系統，他原本是為了改進 Plurk 的中文分詞搜尋而開

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

發的系統。Loso 決定分詞的演算法是基於 Hidden Markov Model[11]，該演算法可由輸入資料庫計算出特定輸出序列的機率，因此能猜測可能的詞組組成，以達到分詞的目的。而我們在使用該系統上，我們蒐集許多長篇小說以及時事新聞作為語彙資料庫，將此資料庫輸入 Loso 處理，如此之後使用者輸入的訊息經由 Loso 處理後，便可完成分詞的動作。例如：使用者輸入“明天開始就要進行網路上超有效減肥計畫”，經由 Loso 處理後，完成的結果會是“明天”“開始”“就”“要”“進行”“網路”“上”“超”“有效”“減肥”“計畫”(圖 15)。

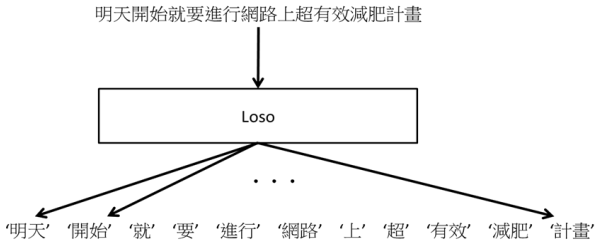

圖 15 Loso 分詞結果範例

◆ **Tf-idf computer**

TF-IDF（term frequency－inverse document frequency）是一種用於資訊檢索與文本挖掘的常用加權技術。它是一種統計方法，用以評估一字詞對於一個文件集或一個語料庫中的其中一份文件的重要程度。字詞的重要性隨著它在文件中出現的次數成正比增加，但同時會隨著它在語料庫中出現的頻率成反比下降。TF-IDF 加權的各種形式常被搜索引擎應用，作為文件與用戶查詢之間相關程度的度量或評級。

Term frequency tf(t,d)最簡單的表示法就是直接算出某一個 term t 在一個文件 d 中出現的次數，我們這裡使用的是正規化的 frequency，是用 t 在 d 中出現的次數除以文件 d 中出現頻率最高的 term w 的次數，算式如圖 16：

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

圖 16 tf 運算式

Inverse document frequency 則是用來判斷一個 term t 在全部的文本中出現的頻率，計算方式為全部文本的數量 D 除以在所有文本中出現 term t 的文本個數，再整體取 log，算式如圖 17：

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

圖 17 idf 運算式

而我們此處的 tf-idf 值是以 tf(t,d) 乘上 idf(t,D)，所以最後 tf-idf 的算式為圖 18：

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

圖 18 tfidf 運算式

由於要計算 tf-idf 時需要有所有分詞和它們出現的次數，所以在做 Loso 的同時，當每一篇文章被輸入到 Loso 裡面時，該篇文章裡的所有分詞都會被存入一個字典檔裡，並且會計算出現的次數，值得注意的是出現次數並不是每出現一次就要計算一次，而是一篇文章的同一個分詞只會計算一次，在有了每個分詞在所有文章出現的次數之後，就可利用上面的公式計算出使用者輸入訊息裡每個分詞的 tf-idf 值，之後再用特殊格式傳到 libSVM 作後續處理，關於特殊格式會在 libSVM 中做說明。

◆ **libSVM**

SVM (Support Vector Machine)屬於一般化線性分類器，是在統計學習理論的基礎上發展起來的一種學習演算法，簡而言之它是個起源跟類神經網路有點像的系統，不過現今最常拿來就是做分類，這也是我們在 filter 裡運用的功能。在 libSVM 裡面，我們用到了 train 和 predict 兩種功能，train 會接受特定格式的輸入，產生一個 Model 檔供 predict 使用，而此特定格式我們是以各詞組的 tf-idf 作為特徵值，它是一種統計方法，用以評估一個詞組對於一個資料庫中其中一份文件的重要程度，而 libSVM 輸入資料的特殊格式，如下頁圖 19 所示，其中 label 代表種類的代號，即我們前面提到的語意概念，index 代表詞組的索引，而 value 就是 tf-idf 特徵值。在我們系統中，我們自定義了六種語意概念，六種概念分別為：一般訊息、情話、吃喝玩樂、購物敗家、情緒字眼和情色內容，並各自有資料庫供 SVM 作 training。而 predict 能將使用者輸入的訊息，依據 model 將之分類到六個語意概念之一。

```
[label] [index1]:[value1] [index2]:[value2] ...
[label] [index1]:[value1] [index2]:[value2] ...
.
.
```

圖 19 輸入 libSVM 特殊格式

圖 20 是用“*明天開始就要進行網路上超有效減肥計畫*”當作 input 通過 tf-idf computer 之後，傳給 libSVM 的特殊格式。於圖中，最前端的數字代表此訊息所屬的語意概念，由此可知“*明天開始就要進行網路上超有效減肥計畫*”是屬於概念一，也就是一般訊息，在後面一段是“28:1.07522163362”，冒號前面的 28 表示原訊息裡的第一個分詞“明天”在分詞的字典檔中是第 28 個分詞，而在經過 tf-idf computer 的計算後得出

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

的特徵值則是 1.07522163362，將整個訊息裡面每個分詞都找出它們在字典檔的 index 和它們的特徵值，如此便是一個訊息經過 tf-idf computer 計算後得出的 libSVM 輸入格式。

```
1 28:1.07522163362 34:1.14002375478 101:1.39830177003 175
:0.763572661946 347:0.952438209038 351:1.23444496739 516:
1.80078940645 2903:2.30139175702 3369:1.93341497172 4610:
2.60242175268 8496:2.60242175268
```

圖 20 libSVM 輸入訊息範例

4.2.3 Database

此處儲存了語意資料庫和替代訊息的樣本資料庫，下圖圖 21 即為整體內容架構。此處資料庫內的資料是以大量的 txt 檔做儲存，在做自然語言分析的時候只需要以 txt 檔的方式就可存取，資料庫有 3 種，第一種是 Loso 的 training data，這些 training data 就是各類型的文章和新聞，供 Loso 計算詞組出現頻率以完成中文分詞。第二種是 libSVM 的語意概念資料庫，目前共有六種，每一個資料庫裡都有大量符合該語意概念的訊息或文章，可做後續的增減。最後一種是替代訊息的樣本資料庫，裡面存的是較為一般化的訊息，可以給所有群組的人觀看，當有需要使用替代訊息時，系統便會從中隨機取出一個當作替代訊息。

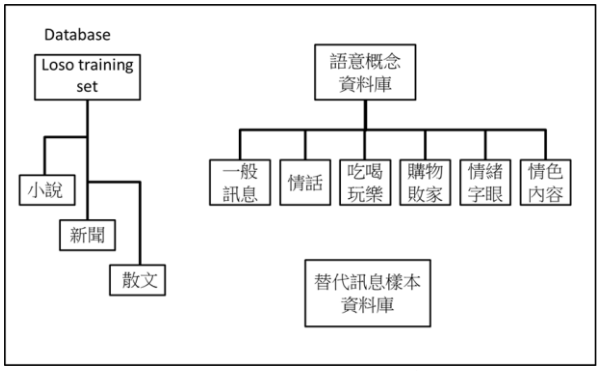

圖 21 Database 內部架構圖

4.3 運作流程範例

在正式使用 filter 前需要先做一些準備動作，第一件事就是先安裝過濾器，由於我們的系統是建立在 Chrome 瀏覽器上面，所以使用者必須先在 Chrome 瀏覽器上安裝我們的 extension，先選 Chrome 的工具→擴充功能呼叫出擴充功能頁面，點選上方的載入未封裝擴充功能按鈕，選取 filter 的資料夾或壓縮檔即可，若是成功安裝後會在擴充功能欄裡看到 filter 資訊出現，如圖 22 所示。

圖 22 成功安裝擴充功能

另外使用者還需要先做一項準備事宜，就是使用者需要在他自己的 Facebook 的好友群組中先設立四個群組，分別是”Bosom Friends”、”Normal Friends”、”Parents”跟”Lover”(圖 23)，這裡會需要使用者自行設立群組和選擇群組內的好友名單是為了達到在第三節中提到的修改彈性，由於群組是使用者自行設立的，所以名單內容會比較符合使用者的需求，後續的修改也較方便，這四個群組設立完成後便可開始發布動態了。

圖 23 事先將好友分為四群組

作為範例假設今天我們要發布一則動態訊息，內容是一則線上遊戲《英雄聯盟》的宣傳影片多媒體連結，如圖 24 所示：

圖 24 發布動態消息

Browser extension 接收訊息後會以 JSON 的格式送給 filter server，server 收到後會擷取相關資料，即影片名稱及其描述，將這些資訊傳給 Loso

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

作中文分詞，算出 tf-idf 特徵值後，再以特殊格式交由 libSVM 做分類，此例中分類的結果為語意概念 3，就是指吃喝玩樂類，而此類訊息不適合給 Parents 群組裡的好友看到，因此需從樣本資料庫中選取替代訊息，而在最後分別以不同的群組權限回傳給 client 端。

此範例中，由於我們傳遞的訊息是一個網址連結，系統會取網址的連結名稱，先將連結名稱裡的英文和符號去除後，再通過 Loso 處理，得到的分詞結果為”英雄”、”聯盟”、”暗殺星”、”宣傳”、”片”，再將分詞經過 tf-idf computer 計算得到的結果用 libSVM 輸入格式表示後，結果如圖 25 所示。

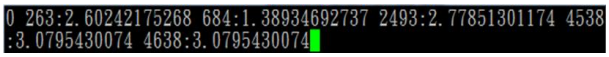

圖 25 計算 tfidf 後結果

發布之後在使用者塗鴉牆上看到的結果就如圖 26 所示，只有 Parents 群組的好友看到的訊息不是使用者發布的原始連結，其他群組的人就可以看到原始的連結。

圖 26 使用者塗鴉牆上的結果

5. 研究成果展示

5.1 Filter Browser Extension 安裝順序圖解

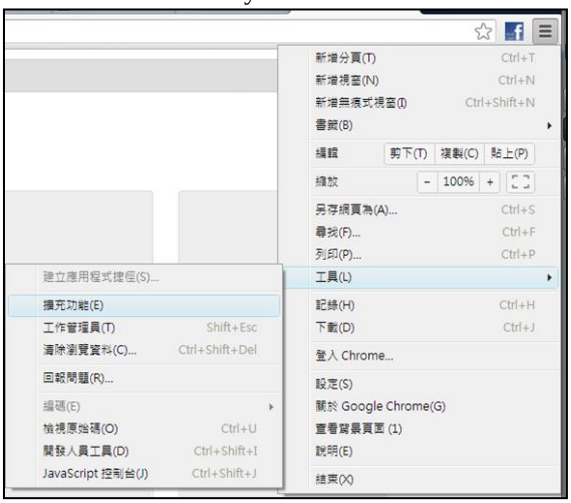

圖 27. 由 Chrome 右上方選取”擴充功能”

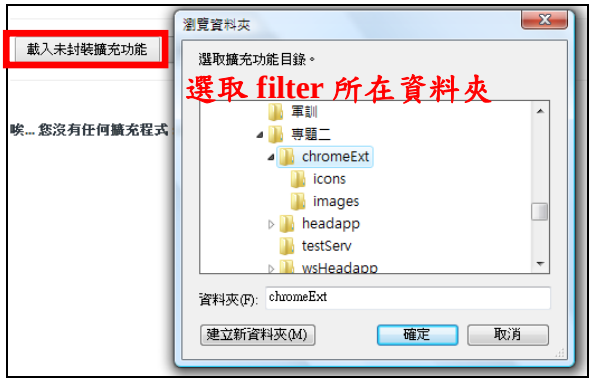

圖 28. 在”載入未封裝擴充功能”中選取 filter browser extension

圖 29 符合以上圖式即表示成功安裝 filter

安裝完 browser extension 後還需要分類好友至四個群組，進入朋友介面後點選”建立新名單”新增群組，共有四個群組，名稱必須一樣，分別為 "Bosom Friends"、"Normal Friends"、"Parents"、”Lover”，以下以 Bosom Friends 為例。

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

圖 30 建立”Bosom Friends”群組

5.2 Filter 使用介面

圖 31 在 extension 中央空白處輸入欲發布訊息

5.3 各功能成果事例

圖 32 到圖 35 為我們對各種訊息類型所做的過濾測試結果，各事例分別有一原始訊息以及另一替代訊息成功發布，可以看到我們除了能夠處理文字訊息之外，也能對網頁連結與多媒體連結作分析，並將這些訊息做適當的分類，在結果中可以看到原始訊息皆正確分類至適宜觀看此訊息的好友，而另一替代訊息，則提供給其他群組觀看。

圖 32 過濾包含不雅字樣文字訊息結果

圖 33 過濾包含情話文字訊息結果

圖 34 過濾購物網頁連結結果

圖 35 過濾包含情色多媒體連結結果

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

6. 實驗成果

6.1 準確度

libSVM 對訊息做分類出來的結果可能發生誤判的情形，其中一個原因是丟給 libSVM 的 training data 不及使用者所發布的訊息內容來的廣泛，因此我們做了準確度的實驗，來觀察 training set 數量與準確度的關係。在實驗中我們替六個類別各自蒐集了 200 份相異的使用者訊息，一共 1200 筆測資，分別取其中的 12.5%(各 25 筆)、25%(各 50 筆)、37.5%(各 75 筆)以及 50%(各 100 筆)作為 training set，並拿這些 training 完的 module 來對全部 1200 筆訊息做分類，判斷分類正確性所得到的結果如圖 36 所示。

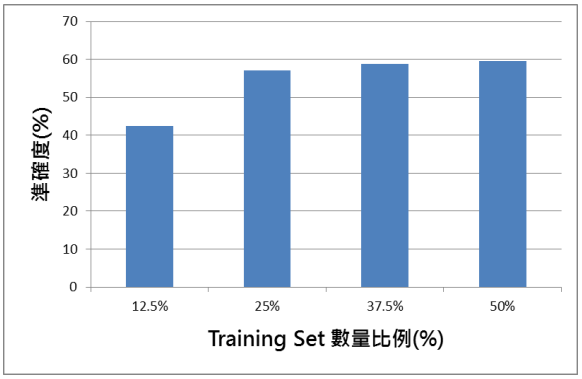

圖 36 準確度實驗結果

其中當 training set 僅有全部訊息的 12.5%時所測得的準確度為 42.2%，而若將 training set 數量增加則準確度可以達到近 60%，但也發現到一現象就是準確度接下來就沒有明顯的提升，猜測可能的原因應該是，當其他類別裡的 training set 增多時，也會讓其他可能看起來不足以代表該特殊類別的詞組變成分類的依據，舉例來說，如果在”情話”類別中除了大量出現”親愛的”、”想你”這類很明顯的分類依據之外，還經常出現了”出去玩”這項詞組，當資料量一大時，可能使得 libSVM 在做分類時會不知道這類訊息是該分類到”玩樂”類別中還是”情話”類別，因此反而降低了分類的準確度。

6.2 反應時間

另外我們也在反應時間這部分做了實驗，在這部分，由於主要的處理是在 server 端，且與 server 端之間的連線品質則是視情況而定，因此我們著重在實驗 server 端的處理時間，也就是收到一訊息後，須要花多久時間才能做完正確的分類並擇一替代訊息以及配好各自對應的群組再回傳給 client 端，當然這處理時間會與”訊息長度”成正比關係，訊息愈長，在 Loso 部分需要分詞的內容跟著增加，libSVM 中做 predict 也會需要查看更多詞組，因此處理時間自然會跟著增加，而在實際實驗中，各長度段落實驗二十次取平均，得到的數據如圖 37 顯示，若是較為簡短的訊息，大概字數在 20 字以內，處理時間平均為 3ms 左右，而若訊息字數在 100 字以內，處理時間約為 10ms 以內，至於訊息內容量更大的情況，大約三四百字左右的訊息，所需的處理時間則約需要花費 25ms，雖然這反應時間或許在訊息量大時會顯得有點冗長，但大部分的 Facebook 使用者所發布的訊息都為簡短的心情特寫，出現內容量大的訊息的頻率不高，因此我們認為這反應時間的結果是我們能夠接受的。

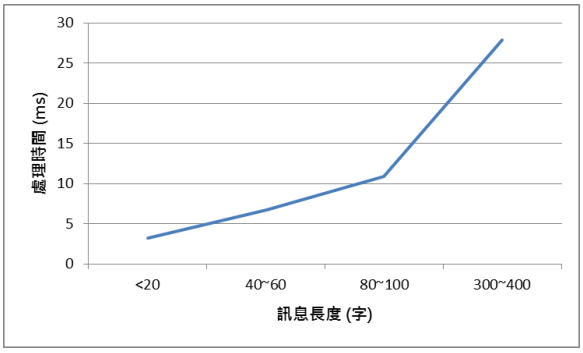

圖 37 反應時間實驗結果

7. 待解決議題

目前在分類上面面臨兩個問題，第一個是在 Loso 作中文分詞上，有些專有名詞或是特定的詞彙可能因為資料庫的不足而無法正確分割詞組，以致在之後的分類上造成誤判的結果。另一個問題是在 libSVM 的 training data 上面，由於概念可能包含的訊息過於廣泛，且大部分臉書使用者發布的訊息較為簡短，因此能拿來做分類的依據略微不足，導致 libSVM 分類出來的結果可能不盡理想。下頁圖 38 即是一個分類失敗的例子，訊息中"快打旋風"是一個遊戲的名稱，但是因為在 Loso 的資料庫中可能沒有出現這個詞彙，導致無法對這四個字進行分詞。另外”一整晚”這三個字會被 Loso 分做同一個辭彙，但在 libSVM 中能被分作”吃喝玩樂”類別的關鍵詞彙只有”玩”這個詞，而此訊息中最重要的”玩了一整晚”由於”一整晚”不一定只在”吃喝玩樂”類別中出現，所以此訊息沒有被分到”吃喝玩樂”，而被分到”一般訊息”中。

圖 38 玩樂訊息分類失敗至一般訊息範例

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費

這兩個問題只能透過不斷擴大資料庫來改善。另外臉書的使用者除了文字訊息和多媒體連結之外，還可能分享圖片，但我們還沒有對圖片的處理部分做深入的研究，因此目前沒辦法對圖片作處理。

8. 結論

為了達到讓使用者在使用臉書社群網站時能夠替自己增加一些隱私性，我們設計並實作出了「媽媽臉書過濾器」系統，其功能是能將使用者欲發布至塗鴉牆上的訊息進行內容及語意分析，然後透過自動對應臉書訊息存取控制的設定來達到保護臉書發文隱私的目的。

系統分類訊息的方法，是先以 Loso (基於 Hidden Markov Model)此中文分詞系統將訊息做分詞，再以經由 libSVM 做完 training 的 modules 對訊息做分類。

本系統的一大特點在於結合臉書現有的訊息存取控制機制來達到保護訊息隱私的效果。使用本系統時，僅需要在發布訊息階段使用配有本系統所附之插件的瀏覽器進行發文。爾後所發布的訊息不管是用哪台電腦、或是各種不同的瀏覽器（包括沒有裝本系統插件的瀏覽器）觀看，隱私均會受到保護。

9. 論文銘謝

在此感謝趨勢科技公司透過 101 年度經濟部工業局「雲端資安技術研發-強化社群媒體使用安全產學合作」資助本計畫之研究經費。

參考文獻

[1] Kashmir Hill, “Forbes: Forbes: Facebook Can Tell You If A Person Is Worth Hiring”, (http://www.forbes.com/sites/kashmirhill/2012/03/05/facebook-can-tell-you-if-a-person-is-worth-hiring/)

[2] YouTube, “**媽媽臉書過濾器**”, (http://www.youtube.com/watch?v=i7ZUcfdn_OA)

[3] Chrome Developers, “ Google Chrome Extensions”, (http://developer.chrome.com/extensions/index.html)

[4] Wikipedia, “WebSocket”, (http://en.wikipedia.org/wiki/WebSocket)

[5] Facebook Developers, “API reference”, (http://developers.facebook.com/docs/reference/apis/)

[6] Wikipedia, “Web storage”, (http://en.wikipedia.org/wiki/Web_storage)

[7] Wikipedia, “JSON”, (http://en.wikipedia.org/wiki/JSON)

[8] Victor Lin, “Loso”, (http://opensource.plurk.com/Loso_Chinese_Segmentation_System/)

[9] Wikipedia, “tf-idf”, (http://en.wikipedia.org/wiki/Tfidf)

[10] Chih-Chung Chang & Chih-Jen Lin, “A Library for Support Vector Machines”, (http://www.csie.ntu.edu.tw/~cjlin/libsvm/)

[11] Wikipedia, “Hidden Markov model”, (http://en.wikipedia.org/wiki/Hidden_Markov_model)

* 本研究接受趨勢科技公司經由 101 年度經濟部工業局雲端資安技術研發-強化社群媒體使用安全產學合作計畫經費