一個以賽局理論為基礎的網頁主題區塊擷取算法

羅濟群 陳昌民 程鼎元
國立交通大學 國立交通大學 國立交通大學
資訊管理研究所 資訊管理研究所 資訊管理研究所
cclo@faculty.nctu.edu.tw tbh@iim.nctu.edu.tw kewas@iim.nctu.edu.tw

摘要

隨著資訊與網路科技的快速蓬勃發展，網際網路已成為目前最龐大的資料體，由於科技的進步以及使用者人數爆增，每天有數以萬計的網頁產生。要在這麼龐大的資料當中蒐集相關資料，因此，本研究提出一個以賽局理論為基礎(a Game-theoRy-based Algorithm for extracting theme-Block from a web page, GRAB)演算法能夠自動地將使用者有興趣的主題區塊自動地辨識出來並轉換成易於儲存、檢索與分析的結構化資料，對於往後的應用將會非常方便。本研究針對所設計的的演算法進行一個實作驗證與分析，實作呈現主題區塊資料擷取的範例，並以實例資料測試驗證本研究所提出的演算法運作情形。

關鍵詞：賽局理論、網頁擷取、資料擷取、主題區塊樹。

Abstract

Technology of information and network is rapid growth; the Internet has become the largest knowledge base, due to the advances in technology and the explosion in the number of users, tens of thousands of pages were produced every day. The Internet has also become the largest source of information of users. In Information-explosion Era, it is become an important research Theme to search information from such a huge of data. Accordingly, this paper presents a Game Theory-based method to extract Themeal block from web page (a Game-theoRy-based Algorithm for extracting theme-Block from a web page, GRAB), so that users can automatically retrieve the Theme block of webpage and easily convert into storage, retrieval and analysis of structured data. In this study, we design a prototype system based on GRAB algorithm, and doing the system implementation and analysis. The system implementation will show a example of theme block extraction. And we design two experiments to verify this study proposed algorithm.

Keywords: game theory, information retrieval, topic block tree .

1. 前言

隨著資訊與網路科技的快速蓬勃發展，網際網路已成為目前最龐大的資料體，由於科技的進步以及使用者人數爆增，每天有數以萬計的網頁產生，也逐漸累積成為一個內容豐富而不可忽視的資料來源。要在這麼龐大的資料當中蒐集相關資料，是相當不方便的。因此，近年來網頁資料擷取技術應用在找尋知識與資訊的變得相當重要。

因應許多新型態的網路社群出現與廣為被使用，網誌(Blog)與微網誌的資料大量地被張貼在網路上，網誌的內容可能是對某件事物的看法或使用者的評論、亦或對於商品的使用心得、甚至於是感興趣的主題，乃至於生活上的種種小事…等。使用者從原本單方向接受網際網路上的資訊，變成可以提供資訊的角色。而如何分析每天都在增加的大量部落格資料，讓有用的、主題導向的資料可以透過推薦系統(recommender systems)提供給使用者，或是進行事件偵測(event detection)、廣告行銷(targeted advertising)…等運用，就變成一個非常重要的問題。

每個網頁都是由許多主題區塊所組合而成的。所謂主題區塊，本研究定義為：站在使用者的觀點而言，區塊中的所有元件組合，皆指向同一個目的，敘述同一件事情，且網頁中描述相同主題的區塊，應當整合在一起呈現。以上圖 1 為例，方框中的區塊是“頭條新聞”，區塊中每一個元件都是在表達同一個新聞事

件，包括新聞圖片、新聞標題、新聞摘要、相關新聞連結等。

圖 1：由眾多主題區塊所組成的網頁
(資料來源：奇摩新聞網站)

從使用者的觀點來看，在瀏覽一個網站時，是以主題區塊為單位，先找到要閱讀的主要區塊後，再進一步進行詳讀。以頭條新聞區塊為例，使用者在閱讀時，心中想著要瀏覽『頭條新聞』的區域，因此網頁一呈現出來之後，目光會先掃描『頭條新聞』，然後在心中形成大致的主題區塊，如圖 1 中紅色框所選取起來的部份。接著使用者才認知到這個區塊，是由頭條新聞標題、新聞圖片、新聞標題、新聞摘要、及相關新聞所組成的。由此可知，一個網頁中包含多個主題區塊，一個主題區塊所包含的元件皆有一定的相關性、都是在講同一件事、且具有較緊密的排版關係。

因此，若能夠自動地將這些區塊辨識出來，並轉換成易於儲存、檢索與分析的結構化資料，對於往後的應用將會非常方便。其相關應用相當廣泛，大致可分為兩類：第一，可以將各個小區塊應用於手機或 PDA 等視窗較小的設備，只顯示出使用者需要的那一塊資訊就好；第二，應用於使用者個人化自訂網頁，即使用者可以從各個網頁中取出他需要的區塊，將所有區塊整合在一起，組成一個對使用者最有資訊價值的個人化網頁。

本研究後續文章結構安排如下：第二章介紹背景知識與相關研究，在第三段章述本研究所設計提出的演算法與流程，第四章說明本研究所設計系統架構與系統實作，最後是結論與未來研究工作。

2. 背景知識與相關研究探討

本章主要是針對本研究所需要的背景知識與相關研究進行介紹，首先探討 W3C 所訂定的網頁資料結構及其分類；接著是網頁資料擷取中相關研究所採用的演算法與文章相似度計算方式。

2.1 網頁資料結構

近年來已有許多對於網頁資料處理與擷取方便的研究，絕大多數研究都採用 Document Object Model (DOM) Tree 架構來表示一個網頁的結構，DOM Tree 由 W3C 所定義的一種規範，其目的在於表現單一網頁的結構性。

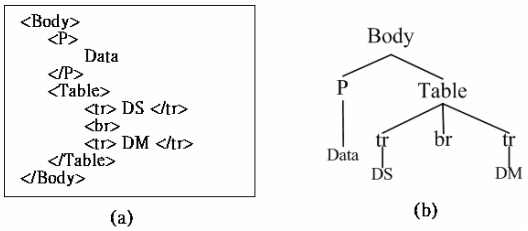

圖 2 (a)HTML 範例 (b)轉換成對應的 DOM Tree 結構

目前許多網頁資料處理與擷取相關研究都使用 DOM Tree 的架構來表現網頁結構，圖 2-a 是一個簡單的 HTML 語法範例，圖 2-b 是其對應的 DOM Tree 架構。透過樹狀的表示方式，能清楚看出結構中的上下層次關係。DOM Tree 的基本概念如下：

(1)非空標籤(Non-empty Tag)：

被<html>、</html>、<table></table>…等標籤所涵蓋的內容，都以樹狀結構中的子節點來表示，並置於 Non-empty Tag(父節點)之下。

(2) 空標籤(Empty Tag)：

、<hr>、<p>等標籤與其他位於同一個 Non-empty 下的標籤，以兄弟點節方式，存放在同一個父節點下。

另外，在 W3C 的 HTML 規格書中，HTML DOM node 可分成三種：

(1)Insignificant node：

無法在瀏覽器上顯示出來的節點。此類的節點包括只含有空白的節點、換段符號、 等等，例如註解文字、隱藏的 tag(e.g., <INPUT type=hidden>)等等。值得注意的是
並不是 Insignificant node，它是屬於 line-break node。

(2)Inline node：

會影響文字顯示外觀的 html tag。

(3)Line-break node：

既不屬於 Insignificant node，也不屬於 Inline node 的節點。

在本研究中，為了更明確地分辨節點的屬性及階層關係，因此將 W3C 定義的 Line-break node 擴充為以下兩種節點：

(1)Visible node:可以直接在瀏覽器上看到的節點，其下沒有子節點。包括以下幾個：Text node：僅包含文字內容的節點；Url node：具有超連結功能的節點，如：<a href="">；Image node：其內容為圖片、或 Flash 等能夠直接在瀏覽器上看見的圖片內容；separate node：HTML tag 本身就具有分隔效果，如:<hr>。

(2)Format node：節點內包含多個子節點，並由特定格式來呈現。包括：Free format node：包含多個 Visible node，且 Visible node 之間不相似；Area format node：本身具有區隔性，能將內容做表格式的呈現；List format node、Combined format node。

透過 W3C 所定義的標準可知，若是藉由分析處理標籤的方式可以進一步地快速擷取出網頁的結構與內容，若是將存在網頁中的結構依儲存結構可以進一步的分類為結構化網頁資料與非結構化網頁資料，如下兩小節所述：

2.1.1 結構化網頁資料

目前現有的網頁資料自動擷取方法，皆是針對結構化網頁[21]來做處理。所謂結構化網頁，是指網頁內容係由網頁伺服器在後端抓取資料庫的記錄之後，依照固定的樣板格式，再透過動態網頁技術顯示在頁面上，如圖所示，網頁中每一筆書籍資料的記錄都是由程式動態產生，因此每筆記錄的樣版格式是相同的。裡面的圖片與資料則是從後端資料庫抓取出來。

結構化網頁有兩個特性：(1)具有相同的樣式(pattern)、(2)區塊間彼此相鄰。目前的網頁區塊擷取技術，大多皆是針對結構化的網頁在系統化網頁中的主要資訊區塊與資料物件之探勘與擷取。

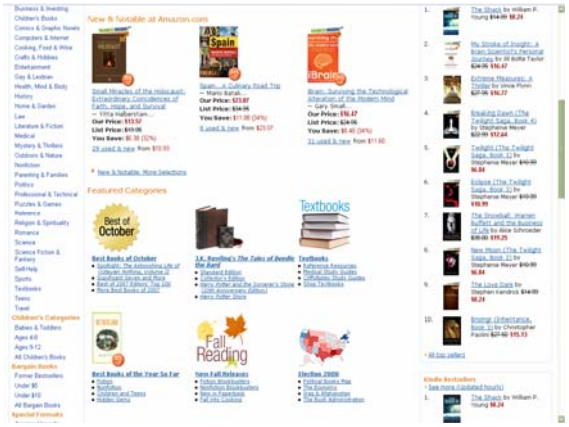

圖 3 結構化網頁

如上圖 3 中每一個粗線框起來的部分，就是網頁區塊。由此可知，結構化的網頁區塊在資料型態與架構上是相當類似的，擷取結構化區塊的技術，是先把網頁轉成 DOM Tree 架構，再從 DOM Tree 架構中找出多筆相似的子樹(sub-tree)，經由計算子樹間的相似度之後，就可以推斷出哪幾棵子樹是具有相同樣版格式(pattern)，而這些子樹就是網頁中重覆出現的重要區塊了。因此現有的方法對於系統化區塊的處理，效率及正確率都很高。

2.1.2 非結構化網頁資料

然而，只針對結構化網頁的擷取技術，並無法完全適用於真實情況。如圖 4 所示，網頁設計人員與工程師通常會為了排版好看、容易閱讀等理由，依照使用者的視察上的美化與觀感，自行設計各種大小不一的樣版格式，而分成各種樣版格式不同的區塊，因此目前仍有許多網頁是非結構化，而過去所提出的研究中尚無法處理非結構化的網頁。

圖 4 非結構化網頁

2.2 網頁資料擷取

網頁資料自動擷取的方法依據作法的不同可歸納為兩大類：(1) 機器學習 (Machine Learning) 、 (2) 自動化擷取 (Automatic Extraction)。第一類Machine Learning的方法，又稱為wrapper induction。Wrapper[25]是一種自動化的資訊擷取程式，能自動的將文件包含的資訊擷取出來，並透過單一格式顯示出來。Wrapper induction是透過監督式的機器學習方法，讓系統能夠自動根據使用者所標註的擷取部份來產生擷取規則，再進行內容的擷取，如SoftMealy[1]和STALKER**錯誤! 找不到參照來源。**所提出的系統，然而，這種方式非常耗費時間，且效率不彰。

第二類自動化擷取方法，則是屬於不須標註(Non-Labeling)的方法，能自動找出應該要擷取的部份。例如：IEPAD[12] 、DeLa[7] 、MDR[9] 、DEPTA[1][3] 等系統，皆是屬於automatic extraction的方法。其做法是認為所謂重要的資料區塊，是指多筆重覆相似的資料記錄(Data Records)，因此自動化擷取方法的目標，就是設法找出這些資料記錄[4]。

從網路上自動擷取資料後，緊接著就是要開始對資料進行分析與歸類，因此，我們將整個研究的工作細分為三個部份，分別是資料分群與相似度計算、網頁資料區塊化、網頁資料粹取，分別如下列小節所探討：

2.2.1 資料分群與相似度計算

過去在資料分群技術的研究已經有超過四十年以上的歷史，對於將相似的對象群聚集在一起，使得同一群的成員有相似屬性的分群方法。在分群方法上，MacQueen [14]提出從空間分割概念著手的K-means演算法，希望找到平衡狀態的聚類中心。I. S. Dhillon [13]提出了spherical K-means演算法，開始將空間分割的概念應用於相似的文件分群上。然而，I. S. Dhillon [13]所提出的空間分群方法，雖然可以將文件快速的分群，卻沒有辦法即時將相關的主題文件正確分群，使得這樣的批次分群無法將大量文章快速而且即時的進行分群。因此，為了因應網路的文章大量的產生，希望達到提供正確的文章分群，Tseng [22]提出了Efficient Online Spherical K-means Clustering來達到線上即時的文章正確分群。

W.H.E. Day [23]以及L. Kaufman[17]提出了階層式概念的分群演算法；將所要處理之資料集合的資料點，利用聚合或分裂的方式，將彼此相似度高的較小群集合併成較大的群集，或者將較大的群集進行分離，接著利用樹狀結構來表示群集之間的關係。學者L. Kaufman[17] 提出分裂式的分群演算法(DIANA)，採取由上而下的處理方式，一開始將所有資料點視為一個群集，不斷的依據相似度計算方式，將大群集分裂成較小的群集，直到每個資料點都成為一個獨立的群集，或著是達到某種使用者設定的終止條件為止。

學者M. Ester [19]提出了利用資料點之間的密度關係來分群的分群演算法(DBSCAN)，經由評估附近的資料點是否足夠密集、分布密度是否在界定的範圍之內，將資料集合中較密集的資料視為一個群集；DBSCAN方法可以用來過濾資料點中的雜訊。這種資料點之間的密度關系的分群方式，常應用於部落格網路的分析上。學者Shen [10]提出了在部落格中找出尋潛在朋友(latent friend)的研究，Shen利用餘弦相似度計算法 (cosine similarity-based method)；首先，將使用者的部落格文章內文經由時態還原 (stemming ）、移除停止字(stop-word)、特徵選取(feature selection）轉換成向量表示。內文包含了使用者本身所發表的文章主體(body)、文章的標題(title)，以及來自其他使用者對這篇文章的評論(comment)。向量中的每一個元素代表一個詞(term)其權重為該詞在文章中出現的次數(term frequency)。計算兩個使用者向量的餘弦值即可以得到兩個使用者的相似度。Shen [10]發現，文章發表時間對於使用者相似度的影響，認為使用者的興趣或是關心的議題會隨著時間而做改變。因此若單純的計算內文的相似度，將會忽略這樣的特性。而提出了結合時間變數的計算公式。

學者Jonathan [15]指出在計算相似度時，有兩種看待使用者發表文章的觀點，大文章和小文章觀點是單獨看待每一篇的網誌，個別的計算每一篇網誌的相似度後，再加總得到整體使用者的相似度。學者I. S. Dhillon [13]認為在計算整體的相似度時，不應該平均的加總每篇網誌的相似度。應該隨著該篇網誌對其使用者整體文章主題的集中程度(entry centrality)而給予不同的權重。例如若一位使用者的部落格中大多是關於美食的文章，則關於美食的文章在該使用者的部落格中就具有較高的權重。加上這樣的調整之後，小文章觀點在計算相似度上能有較好的表現。

2.2.2 網頁資料區塊化

IEPAD[12]是 2001 年由張嘉惠博士等人所開發的系統，係以自動化的方式產生Wrapper藉以擷取資料。IEPAD認為每一個網頁中皆含有多筆重複出現的資料，具有相同的pattern，若能找到網頁中重複出現的記錄中具有相同的 Prefix 的字串，即是所謂的 Repetitive Pattern。下一步再從這些Repetitive Pattern中找到一個滿足使用者所給的Maximal Repeat條件，接著計算每個Pattern的變異性與密度來決定此Pattern是否為真實的資料記錄，並從剩餘的Pattern推導出擷取規則。此方法對於多筆記錄的網頁擷取能力相當好，但對於網頁中只有單一筆記錄的網頁，則無法進行擷取。

DeLa[7]是在 2003 年由Wang等人所發表的系統，主要包含兩大功能：資料擷取與資料標識。Wang認為網頁中的資料記錄大多是連續且重複的，若能將網頁中所有標籤建立出一棵Suffix-tree來找出連續重複的樣式(C-Repeated Patterns)。有了連續重複的樣式，就可以用來產生Wrapper。

2.2.3 網頁資料粹取

DEPTA[1]是Liu等人在2005年所發表的系統，它改善了MDR的擷取功能，利用MDR所找出來的資料記錄中，透過Tree Alignment的方法，將資料轉換成結構化的格式，能夠存入資料庫中。為了更進一步改善計算Tree node相似度的STM (Simple Tree Matching)演算法，Liu等人在 2006 提出ESTM (Enhanced Simple Tree Matching)演算法[3]，在計算兩棵樹的最大符合(maximum matching)的時候，把節點的內容也一併加入考量。學者Y.Kim等人則是在 2007 年提出HTML Tree Matching演算法[9]，認為每個節點的內容皆不同，其權重也應該有所不同。Y.Kim等人藉由節點值(Node value)的計算，改善原本的STM方法。圖 7 為整個DEPTA系統之架構與流程

VIPS (Vision-based Page Segmentation)演算法[10][21]，是Yu、Ma等人在 2003 年提出，其概念是先把整個網頁以DOM Tree架構表示，DOM Tree中的每個節點，都代表一個虛擬區塊(Virtual Block)。但因為有些區塊太大，必須進一步做切割，分成更小的區塊。切割的方法是以背景顏色、文字數目、區塊大小等視覺上的線索來做依據，若一個區塊內的這些特徵差異性太大，就必須將區塊切割成更小。切割出許多虛擬區塊之後，接下來偵測這些區塊之間的分隔線，依照其權重值來判斷該分隔線是不是重要的。決定出分隔線之後，所有在分隔線同一邊的區塊都將合併成一個區塊，再針對這個區塊計算顆粒值(Granularity)，又稱為DoC。若區塊的DoC小於某個門檻值，就重覆進行切割，直到所有區塊都滿足門檻值為止。

3. 主題區塊樹演算法

一般的HTML網頁，可以先經由轉成XML格式之 XHTML，處理掉 HTML Tag 的缺漏問題，再清除不必要的 tag，例如 JavaScript、註解等資訊後，轉換成 DOM Tree 的樹狀架構。如下圖所示：

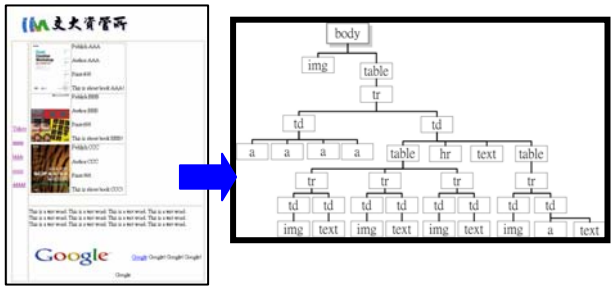

圖 5：網頁轉換成 DOM Tree 架構

DOM Tree 架構就代表網頁的階層架構，其中每一個節點都是一個 HTML Tag。但光是從 Tag 無法正確的取出網頁的資料，而是需要取出完整的子樹出來，才能得到一塊有意義的網頁區塊資訊。如圖 8 所示，若只取出<TD>這一個節點的內容，並無其他意義；若是取出包含<TD>之下的整棵子樹，就相當於擷取出這個表格的一個儲存格的內容，這就有意義了。

擷取主題區塊，較直覺的做法是事先定義一些規則[VIPS]，然後以深度為優先搜尋(Depth First Search, DFS)針對每個節點去做IF-THEN 的判斷，若判斷之後認為該節點及其底下的所有子節點可以獨立成為一個主題區塊，則以該節點為樹根，擷取出整棵子樹，所得到的一棵子樹就是一個主題區塊。然而這樣的方式太過於主觀，容易因為規則不足或例外情況等因素，造成誤判。

另一種方法則是透過計算『資訊含量』的方式。所謂資訊含量，是指每個主題區塊有包含足夠豐富的元件。主題區塊(a)包含了標題文字、新聞圖片、新聞摘要等資訊，其資訊含量就比較高；而主題區塊(b)只包含了圖片，其資

訊含量相對而言就比較低。

從一棵 DOM Tree 之中判別出哪些部份是屬於同一棵樹，是最核心也最重要的部份。本研究利用賽局理論嚴謹的數學模式，來讓每個節點做 Game 的競爭，當一個節點要判斷是否要獨立成為主題區塊時，除了計算自己的資訊含量外，也一併考慮到其下所有子節點所做之決策對它的影響。本研究透過賽局理論的方式來做判斷，有以下幾點好處：

(1)取代單純以規則判斷的缺漏，降低誤判或例外情況出現

(2)單純計算資訊含量的方式，僅針對某個節點來計算，並未考慮到該節點與其他節點之間的關係，以及其他節點做決策時，對它的影響。

(3)由於是針對網頁內容來處理，因此可以假設所有賽局中的參與者。

本研究所提出的演算法主要目的是希望能夠粹取出文章主題及其所在區塊，可分成三個主要步驟：原始網頁轉換成 DOM Tree、透過賽局理論判別主題區塊、主題的區塊進行合併、修剪。首先，將原始網頁轉換成 DOM Tree 的樹狀架構，概略的表現整個網頁結構及階層性，並且對於網頁也有初步的區隔。第二，計算 DOM Tree 裡面每一個節點的資訊含量，透過正規式的賽局(Normal Form Game)的方式，判斷該節點是否要可以自己獨立成為一個主題區塊，或者仍需要往下層繼續判斷。在這個步驟適合獨立成為主題區塊的節點，將逐一加入主題區塊樹(Topic-based Tree)，漸漸形成一棵樹。第三，調整上個步驟所產生的主題區塊樹，同主題的區塊進行合併，主題性不足的區塊，也就是小於門檻值的節點，將放入步驟二，繼續分割。最後，會得到一棵調整過後的主題區塊樹，其中的每一個節點，就是一個主題區塊，使用者可自由選擇要取得哪一塊的內容，並加以利用。將上述三步驟分別描述於下各小節：

3.1 建構 DOM Tree

首先，一個網頁會包含著各種元件，其中註解、JavaScript、CSS 樣式等非必要的資訊，要先在這個步驟中拿掉，僅留下 HTML 元件。接下來，將 HTML 轉換成符合 XML 規範的 XHTML 格式。本研究是利用 JTidy 套件來處理，JTidy 是能夠在 Java 語言上執行的 HTML Tidy，能夠自動判斷及修復容易寫錯的 HTML 標籤，不論是標籤的順序，或是標籤是否有結尾等等，接著可以透過標準的轉譯方式，將合法的 HTML 轉換成為 XML 格式的檔案。

```
<?xml version="1.0" encoding="UTF-8"?>
 <html xmlns="http://www.w3.org/1999/xhtml">
  <head>
   <meta content="HTML Tidy, see www.w3.org" name="generator"></meta>
    <title></title>
  </head>
  <body>
   <table border="1">
    <tr><td>data1</td> <td>data2</td> /tr>
    <tr><td>data3</td> <td>data4</td></tr>
   </table>
  </body>
</html>
```

圖 6 轉換成 XHTML 格式

圖 6 描述了將網頁元件處理成 XHTML 的過程。此階段產出的合法 XHTML 檔，就是一個 DOM Tree 的架構。一個合法的 XHTML 檔，會有一個唯一的根節點，其下有數個子節點，每個子節點之下還有其他子節點。相對於一棵 DOM Tree，就如同樹根、子樹節點、樹葉節點等概念。

3.2 建構主題區塊樹(Topic-based Tree)

從 DOM Tree 中挑選出主題區塊，並建構成主題區塊樹。DOM Tree 架構就代表網頁的階層架構，取出來的一棵子樹，就相當於擷取出網頁當中的某塊內容。DOM Tree 節點的分類，就是要解決這樣的問題，每一個 DOM Tree 節點都要去判斷：它是不是屬於有意義的節點？若不是，是否要繼續往下找？還是要忽略掉此節點？在這個階段中，首先會從 DOM Tree 之中挑選出適合的主題區塊，接著逐一建構成一棵主題區塊樹(Topic-based Tree)：詳細處理步驟分述如下：

Step 1：根據資訊含量、門檻值 T，透過正規形式賽局挑選出合適的主題區塊：

值得注意的是，-i 集合裡的所有節點皆選擇不獨立，則-i 的決策才為『不獨立』；若-i 集合裡有一個以上節點選擇獨立，則-i 的決策就為『獨立』。兩個 Player，每個 Player 有兩種 Action，因此會有以下四種可能：

- i 選擇獨立，-i 選擇獨立

代表節點 i 與其下所有子節點的資訊含量皆很高，在這種情況下，節點 i 的資訊含量就等於其下所有子節點的資訊含量之總和，再乘上權重。

- i 選擇獨立，-i 選擇不獨立

這種情況代表節點 i 的資訊含量夠高，而

且下所有子節點的資訊含量過低，因此所有子節點要併入節點 i。

■ i 選擇不獨立，-i 選擇獨立

這種情況代表節點 i 的資訊含量過低，不足以獨立成為主題區塊。

■ i 選擇不獨立，-i 選擇不獨立

這種情況代表節點 i 及其所有的子節點，資訊含量皆過低，沒有什麼有意義的內容，因此皆不需要獨立成為主題區塊。

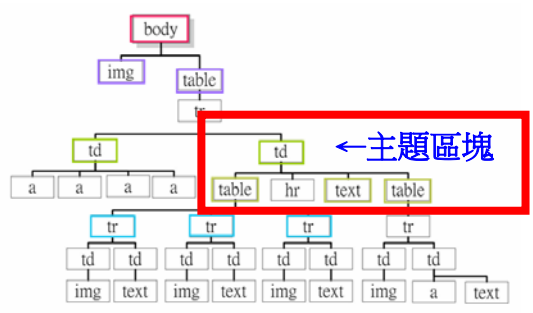

圖 6 主題區塊之判斷考量

當某節點在判斷是否要形成主題區塊時，會受到其下所有子節點的判斷所影響，而這樣就形成了一個『Game』。以圖 7 為例，當節點<td>要判斷時，並不能直接決定自己能不能獨立成為主題區塊，而是必須先看看其下子節點所做的決策。於圖 7 中，節點<td>下面的四個子節點，有三個決定要獨立成為主題區塊(table、text、table)，一個決定不獨立(hr)。

每個 Game 都有報酬矩陣，如表 1 所示。節點 i 做決策時，同時也要考慮到其下所有子節點的集合(-i)的決策。舉例來說，Node i 選擇獨立的情況下，Node –i 選擇獨立的報酬(Utility)是 5，而選擇不獨立的報酬是 4，因此 Node –i 會選擇獨立。依此類推，在本例中，最後得到的 Nash 均衡是兩者皆選擇獨立，報酬各是 5, 5。

表 1 節點報酬矩陣

		Node -i	
		獨立	不獨立
Node -i	獨立	(5,5)	(3,4)
	不獨立	(4,3)	(2,2)

在每個 Game G 中，有兩個 Player，一個是該節點 i，另一個則是其下的所有子節點的集合，我們用-i 來表示。每個 Player 皆有兩個 Action，選擇獨立，或者選擇不獨立。選擇 Action 的 Utility，以 U(i, -i)來表示。因此，整個 Game 可以用下列式來表示：

$$G = <N, A, U>$$

其中 G 代表整個 Game，N 代表 Player 的集合，A 代表 Action 的集合，U 代表 Utility 的集合。

表 2 Game Theory 與主題區塊關係

Game Theory	主題區塊擷取
Player	DOM Tree 中的節點
Action	獨立或不獨立
Utility	選擇 Action 所得到的效益

圖 8 說明了節點 i 與-i 所做的策略，可以組合出四種情況。由此我們可以發現節點 i 在計算資訊含量時，會受到其下所有子節點的影響，亦即底下想要獨立的子節點個數，若底下完全沒有子節點要獨立，則節點 i 的權利就愈大，可以併下所有的子節點；反之若有愈多子節點想要獨立，節點 i 就不見得能夠併下所有子節點了，除非它自己本身的資訊含量夠高。

圖 8 說明 DOM Tree、Topic-based Tree、及原始網頁的對應關係。在 DOM Tree 裡面的<td>節點，就對應到 Topic-based Tree 的 TB 2-1 區塊，而每個主題區塊就等同於 DOM Tree 的一棵子樹，其內容就是一個網頁區塊，即一段完整的 HTML 語法。

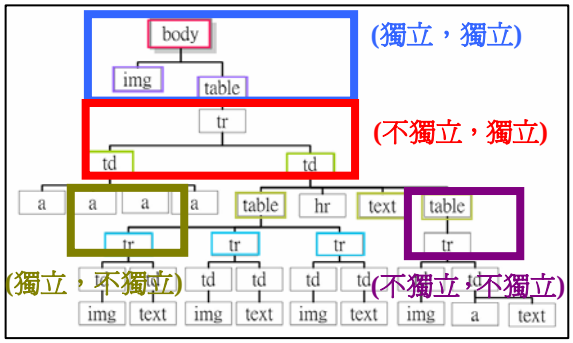

圖 7 Player i 與-i 選擇策略的四種情況

Step 2:建構出主題區塊樹

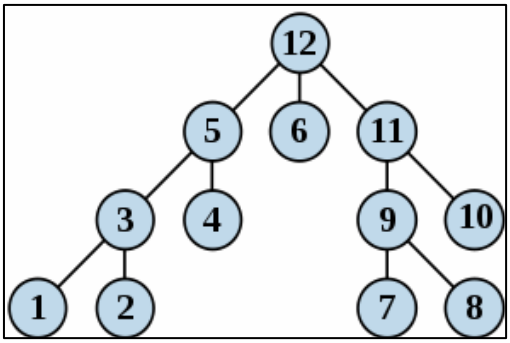

圖 8 以 DFS 進行搜尋的順序

圖 9 是以 DFS 跑過整棵 DOM Tree 的順序，數字即代表順序。挑選出來的主題區塊會先暫存起來，等到整棵 DOM Tree 跑完，所有適合的主題區塊都挑選出來之後，將最後挑選的主題區塊做為樹根，反向建置出整棵主題區塊樹。

3.3 調整主題區塊樹

一個網頁之中，會包含著許多個主題區塊。例如在前文圖 3 之中，在整個版面下方的『新聞夜報』可能就是一個主題區塊；右方的廣告也可以是一個主題區塊。因此把一個網頁切割成多個區塊後，會發現可能有某幾個區塊是同主題，或內容是相似的，但是由於位置不相鄰，而被歸類為不同的主題區塊。

再者，由於每個使用者的觀點不同，可能會主觀的認定某幾個區塊本來就應該放在一起的，因此有必要讓使用者自行決定要取得的區塊範圍及內容。針對這兩個問題，本研究利用「一般化節點」(Generalized Node)的概念，把同主題的區塊組成一個一般化節點，使得這些區塊雖然實際上是不鄰的，但邏輯上是在一起的。

這個步驟會以深先搜尋法(Depth First Search, DFS)方式逐一判斷主題區塊，基於本研究定義的門檻值 H、L 來進行調整：

(1) 若資訊含量(IC) > H，則代表該主題區塊已能充分描述某個主題，因此可以獨立成為主題區塊，不須合併或分割。

(2) 若資訊含量(IC) < L，則代表該主題區塊仍不足以充分描述某個主題，因此必須進行分割。此主題區塊會先標記起來，降低門檻值 T，經由第二個步驟再處理一次，挑選出更細緻的主題區塊。

(3) 若主題區塊的資訊含量介於 L 跟 H 之間，L ≦ IC ＜ H ，則進一步判斷是否合併：

利用計算兩兩子樹的相似度(利用本研究所提出的 TSTM 演算法)：針對相鄰的樹葉節點，計算子樹的最大相似度，再計算內容相似度，最後形成 Generalized Block。

綜合以上討論，本研究所提出之 TSTM 演算法是以 Z.Yanhong 和 L.Bing 提出的 STM 演算法以及 ESTM 演算法為基礎。STM 演算法是計算兩棵子樹的節點標籤(tag)相似度，若標籤相似則傳回 1。即使兩棵子樹的標籤、內容都相同，但資訊含量的多寡卻可能不同，也會影響到它的相似程度。一個新聞網頁中，可能同時會有摘要新聞(a)跟新聞全文(b)，雖然它們的內容相似，但資訊含量卻不同，因此若是將(a)、(b)兩個主題區塊並在一起，對使用者而言反而是一種累贅。在這種情況下，本論文會將資訊含量相對較低的區塊併入其他主題區塊，例如將許多摘要新聞整合成一塊『摘要新聞』主題區塊。

表 3 本研究所提出之 TSTM 演算法

```
Input: two subtrees
Output: the node similarity of these two trees
Alogrithm: TSTM(T1, T2)
If r1 and r2 (the roots of the two trees T1 and T2)
contain distinct symbols OR have visual conflict
then
   Return 0;
Else
   m := the number of first-level sub-trees of T1;
   n := the number of first-level sub-trees of T2;
   Initialization: M[i,0] := 0 for i=0,…,m;
   M[0,j] := 0 for j=0,…,n;
   For i=1 to m do{
      For j=1 to n do{
         M[i,j] := max(M[i,j-1],
         M[i-1,j],M[i-1,j-1]+W[i,j]),
         Where W[i,j] := TSTM(T1[i], T2[j])
         }
      }
Return (M[m,n] + AVG(IC(m), IC(n)) +
content_similarity(r1, r2))

Procedure: content_similarity(r1, r2)
   If r1 and r2 are not leaf nodes then
      Return 0;
   Else
      cs := LCS(r1.data, r2.data);
      w := the number of words contained in cs;
      m := the maximal number of words contained
in r1.data and r2.data;
return w/m;
```

4. 系統實驗與分析

針對本研究所提出的GRAB 演算法設計一個雛型系統，並進行一個系統實作與分析，實際呈現主題區塊資料擷取的實作範例。在本章，首先分別介紹所有模組的架構、資料流程及功能，接著以 10 種主題共 30 個網頁做為資料集，本研究的資料集是採用 Complete-planet網站[1]的資料。Complete-planet是目前最大的深網庫(deep web)，收集了超過 7 萬個網頁資料及搜尋引擎，其中包括 43 個主題，涵蓋真實世界所有主題領域。最後進行分析、並討論實作結果。

GRAB 演算法是針對 W3C DOM Node，考量所有 HTML Node 的特性。由於目前大多數網頁都會使用 CSS 語法來設計頁面樣式，或是嵌入許多 JavaScript 或 Flash 元件，所以這類的網頁無法完整地轉換成 DOM Tree，難以算出區塊權重，資訊含量也就難以計算出來。因此，本實驗的限制是只採用 HTML 為主的網頁，先經由人工方式，判斷網頁內容是否大部份都是 Flash 物件(超過頁面元件的 70%)，並判斷是否包含過多的 JavaScript 或 CSS(超過網頁原始碼 60%)。

本系統是採用 Microsoft Visual Studio 2005 環境，使用 C#.NET 語言開發。系統主畫面如下圖所示，系統上方可以輸入網址，或從功能表列選擇讀取 HTML 檔案。功能操作包括：(1)轉成 DOM Tree，之後再轉成主題區塊樹，計算資訊含量、(2)開始進行擷取，將賽局決策後選擇要獨立的區塊，用紅色方框顯示在網頁上、(3)輸入 ccmID 取得主題區塊內容。系統左上方是轉換出來的 DOM Tree，左下方則是轉換出來的主題區塊樹，並且將資訊含量標記上去。在主題區塊樹上按滑鼠右鍵，即可顯示出該主題區塊的 HTML 內容。

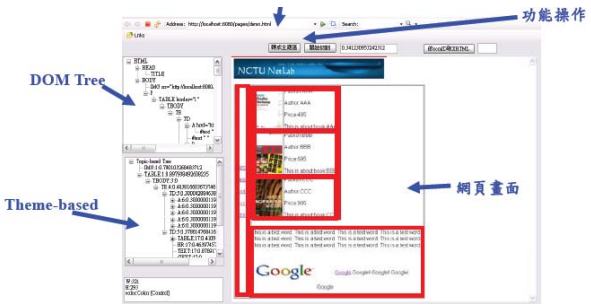

圖 9：系統主畫面

本研究所設計的系統的架構圖如下圖所示，系統共包括有 6 個模組：DOM Tree 轉換模組、主題區塊樹轉換模組、資訊含量計算模組、賽局決策模組、主題區塊樹調整模組、主題區塊樹擷取模組。

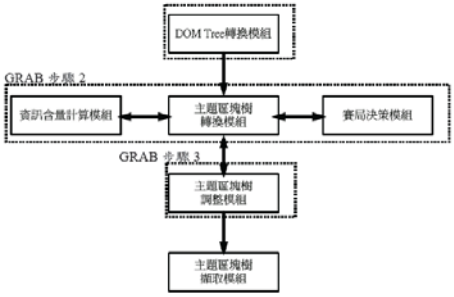

圖 10 系統架構圖

DOM Tree 轉換模組是利用一個 URL 或下載到主機的 HTML 檔案做為輸入，輸出的結果則是一棵 DOM Tree。主題區塊樹轉換模組：則是利用 XML/XHTML 文件轉換成一個主題區塊樹(Theme-based Tree)，每個節點的內容記載著要獨立成主題區塊的節點之 ccmID 值；資訊含量計算模組：則是進行某節點的資訊含量。賽局決策模組：有資訊含量的子樹，即玩家 i 與玩家-i 兩層子樹。產生一個具均衡的結果，包含兩個值，例如(0.75 , 0.635)；主題區塊樹調整模組：利用輸入一棵主題區塊樹，藉以產生一棵調整過的主題區塊樹，同樣是以 XML 文件表示；主題區塊樹擷取模組：則是利用使用者欲擷取的區塊 ccmID，產生該主題區塊的 HTML 碼。

本研究資料集中選擇 10 個領域的資料，(其中含蓋了有：Business、Education、Finance&Economics、Health、Jobs&Careers、News、Politics、Science、Sports、Travel…等領域)，並挑選 30 個網站來進行實驗。來觀察本研究所提出的 GRAB 演算法對於處理這些不同領域的效果。並與前文所提及現有三大類方法做比較，以驗證本研究所設計 GRAB 演算法，並針對處理結構化與非結構化網頁的效能來做驗證。

研究驗證 GRAB 演算法對上述所提到的 10 種不同領域的擷取效能。演算法實驗結果如表 4 所示。從圖表中可以發現，GRAB 演算法在這 10 種領域有很好的效能，平均值皆在 70%到 90%之間。其中又以 Jobs&Careers、News、與 Politics 3 個領域的效能最好，Precision 將近 90%，Recall 都接近 80%。

[1] 資料來源：Complete-planet 網站 http://www.completeplanet.com/

表 4 現有方法準確率(Precision)比較表

Topic	CorrectTB	TB	CorrectTB∩TB	Precision	Recall	F-Measure
Business	9 14 10	7 13 7	6 12 6	.857 .923 .857	.667 .857 .600	.750 .889 .706
Education	10 10 11	11 15 14	9 9 8	.818 .600 .571	.900 .900 .727	.857 .720 .640
Finance & Economics	13 14 12	18 9 10	10 8 8	.556 .889 .800	.769 .571 .667	.645 .696 .727
Health	10 20 17	11 19 22	8 17 15	.727 .895 .682	.800 .850 .882	.762 .870 .769
Jobs & Careers	17 5 7	14 4 6	12 4 5	.727 .895 .682	.800 .850 .882	.762 .872 .769
News	35 20 31	20 19 27	18 17 25	.900 .895 .926	.706 .850 .806	.774 .889 .862
Politics	8 11 5	7 10 5	6 10 4	.857 1.00 0 .800	.750 .909 .800	.800 .952 .800
Science	11 7 9	8 7 10	7 6 8	.875 .857 .800	.636 .857 .889	.737 .857 .842
Sports	5 6 10	8 6 16	5 5 8	.625 .833 .500	1.00 0 .833 .800	.769 .833 .615
Travel	11 9 8	16 7 13	10 7 7	.625 .100 .538	.909 .778 .875	.741 .875 .667

GRAB 演算法在這三個領域的準確率(Precision)都遠比現有的三類擷取方法優秀，也就是說，GRAB 所有挑選出來的區塊裡面，有90%以上都是正確的。而其中又以新聞網頁(News)與現有方法的效能差距最大，主要的原因是本演算法的第二步驟，有運用到「第三次眼球追蹤」的新聞網頁眼球移動模式，因此對於新聞網頁的效果特別好。

GRAB 有較好準確率的第二個原因，是因為有經由賽局做進一步分析，挑選出最適策略，其中大部份的均衡點是落在(獨立，不獨立)的位置。因此一些資訊含量較小的區塊就會被併掉，最後產生的主題區塊樹也更精簡準確，也能減少最後產生出來的主題區塊數量。

表 5 現有方法準確率(Precision)比較表

方法	Jobs&Carrers	News	Politics
GRAB	0.897	0.907	0.886
Machine Learning	0.780	0.734	0.883
Automatic	0.797	0.728	0.681
Rule-based	0.699	0.737	0.825

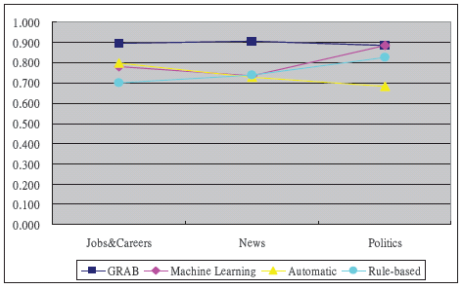

圖 11：與現有三種方法準確率比較

GRAB 演算法在這三個領域的召回率，除了在 Jobs&Careers 領域效能較低之外，其他兩個領域的效能也都優於其他三類擷取方法。這樣的結果可以說明，在所有應該要被挑選出來的區塊中，GRAB 演算法能挑出 70%~80%的正確區塊。這樣的效果跟現有三類方法差不多，且在 News 及 Politics 領域的效果比其他三類都要好。

表 6 現有方法 Recall 比較表

方法	Jobs&Carrers	News	Politics
GRAB	0.740	0.724	0.820
Machine Learning	0.713	0.676	0.723
Automatic	0.760	0.702	0.734
Rule-based	0.807	0.657	0.801

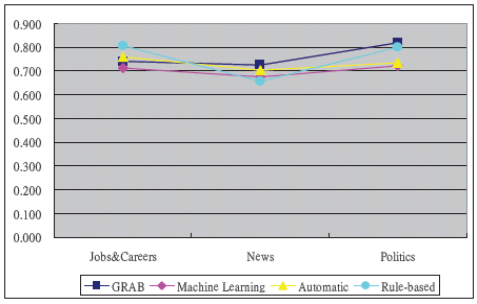

圖 12：現有三種方法 Recall 比較圖

GRAB 演算法在這三個領域的 F-Measure 皆在 80%以上，且都高於其他三類的擷取方法，證明 GRAB 演算法的效能方面的品質，是優於其他三者。

表 7 現有方法 F-Measure 比較表

方法	Jobs&Carrers	News	Politics
GRAB	0.811	0.796	0.851
Machine Learning	0.743	0.703	0.794
Automatic	0.777	0.714	0.706
Rule-based	0.747	0.691	0.812

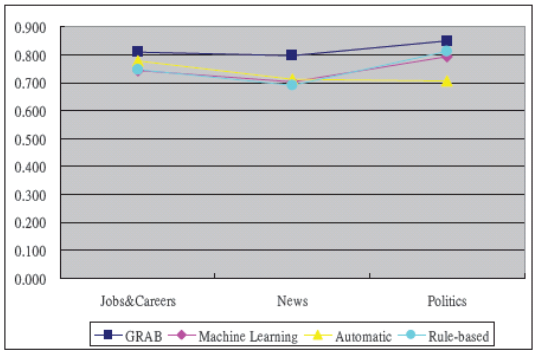

圖 13：與現有三種方法 F-Measure 比較

5. 結語與未來研究

本研究藉由提出一個以賽局為基礎的網頁主題區塊擷取演算法能夠自動地將使用者有興趣的主題區塊自動地辨識出來，並希望透過本研究所提出的方法將網頁資料轉換成易於儲存、檢索與分析的結構化資料。針對現有三大類網頁區塊擷取方法的不足，以及能處理非結構化網頁，本研究提出以賽局為基礎的主題區塊擷取方法，將網頁轉成 HTML DOM Node，並針對每個元件所在位置、特性，計算出資訊含量，再用賽局方法來決定是否要形成主題區塊。具有相似內容的主題區塊會一起呈現，並轉換成易於儲存、檢索與分析的結構化資料，便於往後在受限制畫面下的應用，例如：手機畫面呈現主題新聞。

本研究提出的 GRAB 演算法，是以 W3C 定義之 DOM Node Type 為基礎，並進一步擴充定義兩種 Node，做為計算資訊含量的基礎，結合眼球追蹤、網頁元件特性、以及賽局理論的方法，能讓資訊含量的計算更符合讀者在閱讀網頁的行為模式，賽局的方式能改變部份主題區塊原本的決策，找到一個讓兩位玩家皆滿意的決策，並能減少擷取出來的主題區塊數目，也就能減少產生主題性不足的主題區塊。

經本研究實驗結果證明，GRAB 具有不錯的效能，尤其是在新聞領域更具效果，與現有方法比較之後，也證明 GRAB 具有較好的效能，而且處理非結構化網頁的效果很好。未來，除了在能夠準確的擷取使用者有興趣的網頁主題資料之外，希望更可以建立主題之間的相關性，進而可以利用語意網路的建模方法建置使用者可能會有興趣的相關主題，提供更有效率的資料擷取與探勘的方法，本研究將未來可以繼續深入探討的的方向整理如以下幾點：

(1)擴大處理能力：

目前本論文的 GRAB 演算法，針對 HTML 網頁有很好的效果，但對於包含許多 CSS、JavaScript、<DIV>、或以 Flash 撰寫的網頁，效果就較不理想。現今也愈來愈多網頁利用 CSS、JavaScript、<DIV>標籤來建置網頁，因此未來可加強在這部份的處理。

(2)結合語意網：

除了在能夠準確的擷取使用者有興趣的網頁主題資料之外，希望更可以建立主題之間的相關性，進而可以利用語意網路的建模方法建置使用者可能會有興趣的相關主題，提供更有效率的資料擷取與探勘的方法。

(3)語意相似計算：

目前在計算主題區塊相似度的方法是採用 LCS，只考慮文字。未來可以設計成語意相似計算，讓主題區塊樹的整併可以更精確。

致謝

本研究承行政院國科會研究計劃支援。計劃編號：NSC 96-2416-H-009-008-MY3。

參考文獻

[1] [CZ07] Bi Chen, Qiankun Zhao, Bingjun Sun, and Prasenjit Mitra. Predicting blogging behavior using temporal and social networks. In proc. of the IEEE Conf. on Data Mining, pages 439-444,2007.

[2] B.Liu, R.Grossman, and Y.Zhai. Mining Data Records in Web Pages. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'03), Page 24-27, 2003.

[3] B.Liu, and Y.Zhai. Web Data Extraction Based on Partial Tree Alignment. In the Proceedings of the 14th international conference on World Wide Web, Page 76-85, 2005.

[4] B.Liu, and Y.Zhai. Structured Data Extraction from the Web Based on Partial Tree Alignment. *IEEE Transactions on knowledge and data engineering*, vol.18, no.12.

[5] C.N.Hsu, and C.C.Chang. Finite-state Transducers for Semi-Structured Text Mining. In Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Application, Page 38-49, 1999.

[6] C. N. Hsu, and M. T. Dung. Generating

Finite-state Transducer for Semi-Structured Data Extraction from the Web. Information Systems, 23(8):521-538, 1998

[7] C. N. Hsu, and C. C. Chang. Finite-state Transducers for Semi-Structured Text Mining. In Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Application, Page 38-49, 1999.

[8] C.H. Chang and S.C. Lui. IEPAD: Information Extraction Based on Pattern Discovery. In Proceedings of the 10th international conference on World Wide Web, Page:681-688, 2001.

[9] D.Cai, S.Yu, J.R Wen, and W.Y Ma. VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Research, Redmond, WA 98052.

[10] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Latent friend mining from blog data. In proc. of the IEEE Conf. on Data Mining, pages 552-561,2006.

[11] Gibbons Robert. *Game Theory for Applied Economists*. Princeton Univ Pr, Page 1-11, 1992.

[12] I.Muslea, S.Minton, and C.A.Knoblock. STALKER: Learning Extraction Rules for Semistructured Web-based Information Sources. In Proceedings of AAAI Workshop on AI and Information Integration, Pages 74081, 1998.

[13] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, Pages143–175, 2001.

[14] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Statist. Prob., Pages 281-297, 1967.

[15] Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. Retrieval and feedback models for blog feed search. In proc. of the ACM SIGIR Conf. on Research and development in information retrieval, pages 347-354,2008.

[16] J.Wang, and F.H. Lochovsky. Data Extraction and Label Assignment for Web Databases. In Proceedings of the twelfth international conference on World Wide Web, Page 187-196, 2003.

[17] L. Kaufman, P.J. Rousseeuw. Finding groups in data. an introduction to cluster analysis. John Wiley & Sons, 2002.

[18] M.Goldberg, M. Magdon-Ismailt, S. Kelley; K. Mertsalov, A locality model of the evolution of blog networks. IEEE International Conference on Intelligence and Security Informatics, Pages191-193, 2008.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. Density-connected sets and their application for trend detection in spatial databases. In Proc.1997Int. Conf. Knowledge Discovery and Data Mining, Pages 10-15, 1997.

[20] N.Kushmerick, D.S.Weld, and R.B.Doorenbos. Wrapper Induction for Information Extraction. In Intl.Joint Conference on Articial Intelligence (IJCAI), pages 729-737, 1997.

[21] S.Yu, D.Cai, J.R.Wen, and W.Y.Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, pp. 11-18, Budapest, Hungary, May 20-24, 2003.

[22] Shi Zhong. Efficient online spherical k-means clustering. In IEEE International Joint Conference on Neural Networks, volume 5, pages 3180–3185, 2005.

[23] W.H.E. Day and H. Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. J. Classification, Pages 7-24, 1984.

[24] W.Liu, X.Meng, and W.Meng. Vision-based Web Data Records Extraction. In Proceedings of the 9th SIGMOD International Workshop on Web and Databases (SIGMOD-WebDB2006), Chicago, Illinois, June 30, 2006.

[25] Yi-Feng Tseng. The Mining and Extraction of Primary Informative Blocks and Data Objects from Systematic Web Pages.

[26] Y.Kim, J.Park, T.Kim, and J.Choi. Web Information Extraction by HTML Tree Edit Distance Matching. 2007 International Conference on Convergence Information Technology.

[27] Yun Chi, Belle L. Tseng, and Junichi Tatemura. Eigen-trend: trend analysis in the blogosphere based on singular value decompositions. In proc. of the ACM Conf. on Information and Knowledge Management, pages 68-77,2006.