

Agent-Based Social Network Search Mechanism 以代理人機制為基礎的社群網路資料搜尋之研究

李佳菁 賴宥呈 陳明僑
國立臺東大學資訊工程系

maius531@yahoo.com.tw, keith54105@yahoo.com.tw, joechen@nttu.edu.tw

摘要

社群網路服務幾乎是民眾日常生活裡無可或缺的服務，社群網路服務提供線上的人際關係網，隨著社群網站裡的人際關係越來越龐大，所獲得的資訊也越來越多，使用者被大量的資訊淹沒，結果真正重要的資訊卻不容易取得，因此，本論文提出代理人機制負責收集我們在社群網站上相關的互動資料，考慮網路社群裡發佈的 status、photo、blog，以及其他回應的方式和頻率，我們建立社群網路的關係密切程度函數，再使用線性規劃的方式得出關係密切程度的最大值，當使用者進行資料搜尋時，代入關係密切程度的數值進行資料搜尋的重要因素，最後再經由視覺化時間軸的方式呈現，提供使用者精確的搜尋結果、以及易於瀏覽的呈現方式。

關鍵詞：關係密切程度、時間軸、社群網站、代理人

Abstract

Social Network Service (SNS) plays an increasing role in people's lives. Social Network Service enables Internet users to interact with family, friends, or colleagues, to share their personal information, and to build a social relationship. With the rapid growth of network users, the quantity of data also increased tremendously, the search of this increasing amount of contents become more complicated and difficult. Therefore, this paper proposes an Agent-Based search mechanism to collect status, photo, and blog interactive content, response time and frequency. The mechanism uses linear programming approach to maximize the relationship function as the correlation of search results. Finally, the paper provides a timeline visualization methodology to show efficiently and effectively the result for users.

Keywords: Social Networks, Multi-Agent System, Search Mechanism, Relationship Function

1. 前言

社群網路服務(Social Network Service, SNS)對於大多數的民眾而言，幾乎是每日生活裡無可或缺的存在，透過網路在虛擬的社群裡和家人、朋友以

及同事們，彼此分享個人的想法、感受，藉由這種分享和互動的過程，建立新世代的社群關係。近年來提供社群網路服務最著名的就是 Facebook，Facebook 之所以大受歡迎，在於它創新的想法，讓使用者以新的方式將彼此聯繫在一起，它藉由使用者自我描述的 profile，建立彼此之間的關聯，再進而建立個人的社群網路，由於這些關聯是取決於不同的因素，可能是朋友的關係，也可能是具有相同的嗜好、或是喜歡同一個偶像，甚至是相同政治信仰的陌生人。

隨著社群網路使用者快速增加，隨之而來所產生的資料量也跟著急遽增加，要處理如此龐大的資料量，就變得相當複雜與困難。因此，在一個以使用者為導向的社群網路，使用者應該不需盲目的摸索或查詢，就可以讓使用者快速而精確的找到所需的訊息，這需要一個有效的運作機制，才能夠在社群網路龐大的資料裡，萃取出使用者真正需要的資訊。

同時，社群網路的持續成長，其組成結構也越來越複雜，每位使用者之間原本所存在一個或多個特定的相互依存的關聯(例如：朋友、親戚、共同嗜好、信仰、知識等等)，也會因應社群網路的成長而更加複雜，於是產生更多間接、隱藏且容易被忽略的資訊與關係，因此，思考如何設計一個能夠有效發現這些隱含資訊與關係的機制是有其必要性的。

本論文的目的有二：一是設計適合社群網路的搜尋機制(Search Mechanism)，讓使用者可以快速、有效且準確地搜尋明確所需的資訊；二是以時間軸的方式將前項搜尋結果以視覺化的方式呈現。

2. 文獻探討

2.1 社群網路服務

社群網路服務 (Social Network Service, SNS) 主要是提供一群擁有相同興趣與活動的使用者可以在網路上建立的線上社群。這類服務是透過網際網路、提供使用者各種聯繫、交流的互動網路，例如即時訊息、即時通、電子郵件等，他們通常通過朋友、親戚或是同事將人際網路展延出去。

大多數的社群網路會提供多種方式讓使用者

彼此互動，例如：訊息分享、心情分享、聊天、影音或檔案分享、部落格、討論群組等，社群網路為資訊的交流與分享提供了新的途徑。對於現代的網路使用者而言，使用社群網路服務幾乎成為每天的生活必需項目之一。目前在世界上有許多提供社群網路服務的網站，知名的包括 Facebook、MySpace、Orkut、Twitter、Flickr 等等。

早期的社群網路服務大多數是提供使用者透過聊天室進行交流，隨著 Blog 等新的網路社交工具的出現，使用者可以透過網站建立個人首頁來分享喜愛的資訊。2002 年至 2004 年間，世界上三大最受歡迎的社群網路服務類網站是 Friendster、MySpace、Bebo。在 2005 年之際 MySpace 成為了世上最大的社群網路服務網站。甚至傳聞當時其頁面存取量超越了著名的搜尋引擎--Google。2006 年，Facebook 提供 API 允許第三方使用、開發相關的應用程式，使得 Facebook 隨後一躍成為全球使用者量增長最快的網站。

社群服務網站的發展驗證了「六度分隔理論」(Six Degrees of Separation) [1]，即「在人際關聯脈絡方面，你必然可以透過不超出六位中間人，間接與世上任意一個人產生聯繫」。個體的社交圈會不斷地擴大和重疊，並在最終形成大的社群網路。在此類透過對「朋友的朋友是朋友」原則的實作，而得到發展的線上社群網路中，Friendster 是一個典型的代表。透過社群網路服務，我們與朋友保持直接的聯繫，建立起大的人際關係，以 Facebook 為例，它提供的尋找使用者的工具，幫助使用者尋到失去聯絡的朋友們。另外，在社群網站裡的使用者，通常有很多志趣相同並互相熟悉的使用者群組，相對於網路上的其他廣告而言，商家若在社群網站裡針對特定使用者群組打廣告更有效果。

2.2 多代理人系統

多代理人系統是一個快速成長的研究領域，也是成長最快速的 IT 產業之一。代理人 (software agent, agent) 本身是一組程式，主要的目的是希望可以代替使用者運作，產生自主性。多代理人系統 (Multi-Agent System, MAS) 是一個分散式系統，由一組彼此互動的代理人所組成。由於代理人的自主性，所以多代理人系統並不限在一個特定的系統裡，只要是支援代理人的系統都可以執行。

代理人是具有溝通 (communication) 與合作 (collaboration/cooperation) 的機制。國際間有許多訂定代理人技術標準的單位或組織，例如：FIPA (Foundation for Intelligent Physical Agents)[2] 與 OMG (Object Management Group) [3]。

一般代理人所具備的特質如下：

1. Adaptivity：具備學習能力並能隨經驗增長。
2. Autonomy：具有主動性，朝達成目標的方向下有自主能力採取啟發性的行為。
3. Collaborative behavior：具備可以與其他代理

人合作並一起達成目標的能力與行為。

4. Inferential capability：具備推論的能力。
5. Knowledge-level communication ability：在與其他代理人溝通時，具備使用人性化語言與溝通方式的能力。
6. Mobility：具備機動引導之能力，可以從一個 host 移至另一個。
7. Personality：具備個性化之特質。
8. Reactivity：具備選擇性感測與反應的能力。
9. Temporal continuity：具備保持身分扮演一致性的特質並能夠長時間的維持。

代理人具有能力可自主地完成特定的任務，然而，當代理人面對一些複雜的問題時，例如處理分散式資料或是與分散式控制的系統互動時，就需採用多代理人系統 (Multi-Agent System, MAS)。其中每個代理人都具備不同的特性功能，在完成某一個共同目標的過程中扮演不同的角色。多代理人系統是透過非緊密式連結 (weakly coupled) 所形成的網狀系統，在不同環節裡包含很多不同的個別解決方案 (problem solver)，而這些 problem solver 結合後的解決問題的能力，遠大於個體的 solver。因此，多代理人系統包含多種自動化 component 組合而成，其具備下列的特性：

1. 每個代理人不具備完整可以獨力解決問題的能力
2. 採用分散式的控制系統
3. 代理人之間共用與分享同一個資料庫
4. 採用不同步的計算方式

基於以上的探討，多代理人系統特質裡的 adaptivity, autonomy, personality 等，使得多代理人系統非常適合處理社群網路的相關問題，例如：

1. 多代理人系統的 adaptivity 特性，使得代理人可以依據使用者的興趣而改變，採取以目標為導向的行動，並且提供使用者更合適且個人化的資訊服務。
2. 多代理人系統的 autonomy 特性，使得代理人可以獨立地搜尋資訊，不用使用者的直接介入，如此就可以完成主動式的資訊服務，節省使用者的時間跟精力。
3. 多代理人系統的 Personality 以及彼此間溝通的特性，使得具有相同興趣與嗜好的代理人可以相互溝通，並提供使用者更多更好的資訊。
4. 多代理人系統的學習能力，使得代理人可以透過互動與使用者學習，以至於能感知使用者興趣的改變，進而調整 interest model，這也是個人化資訊服務的以實現的重要關鍵[4]。

2.3 社群搜尋機制

隨著社群網路資料量的日益增加，要處理龐大的資料會變得相對複雜與困難。雖然每個人每天都會透過社群網站得到所需的資料，然而關於如何改善搜尋以及搜尋品質的問題，尤其是如何即時地提

供使用者有效率及有用的資訊，是一個需要持續努力的議題。重要的關鍵並不是搜尋不到資料，而是搜尋引擎可能無法快速提供使用者真正想要或是有興趣的資料，因此，以目前而言，社群網路的搜尋，仍是一件很費時費力的工作。

搜尋引擎的缺點包括：一個 query 可能會傳回大量的回覆結果，以及缺乏個人化(Personalization)，也就是通常對於相同的一個 query，針對不同的使用者應該會依據不同的使用者特質，會有因人而異的結果[5](如圖 1)。上述的第一個缺點可以透過使用者對搜尋引擎重新輸入合適的 query 而得到改善，這需要使用者某種程度的直覺與經驗。另外，對於缺乏個人化這部份，就是實質地針對不同的個人提供不同的 query 結果，而不僅僅只是針對關鍵字進行回應而已。



圖 1 個人化網頁

有許多 meta-search 引擎提供了主要的個人化服務。例如：Query Server, Profusion, Infogrid, Mamma, Ixquick 等，讓使用者可以自行定義要使用的搜尋引擎[6]。Excalibur 專案[7]提出一個個人化的 meta-search 引擎，可以隱含的擷取使用者的喜好並且使用 Naïve Bayesian 分類器做 ranking 與相似度量測(resemblance measure)。Radovanovic 等人[8]提出一個稱之為 CatS 的 meta-search 引擎，利用文字分類技術(text classification)改善搜尋結果，使用經由 dmoz Open Directory topic 階層所推演出的 topic 樹顯示出來，供使用者瀏覽。在 Captain Nemo 專案[9]提出一個具有個人化階層式 search space 的全功能(full-functional)的 meta-search 引擎，可以根據個人化擷取模式與呈現風格去擷取與顯示搜尋結果。WebNaut 專案[10]提出一個以多代理人為基礎的 meta-search 引擎，其包含一組互連的代理人以及一個 meta-genetic 演算法，藉以學習使用者的嗜好與個人化搜尋結果。Kanteev 等人[11]提出一個多代理人的文字瞭解系統(text understanding system)，使用語意瞭解內容並且產生語意描述(semantic descriptor)，並且運用儲存於本體(ontology)內的問題領域知識。

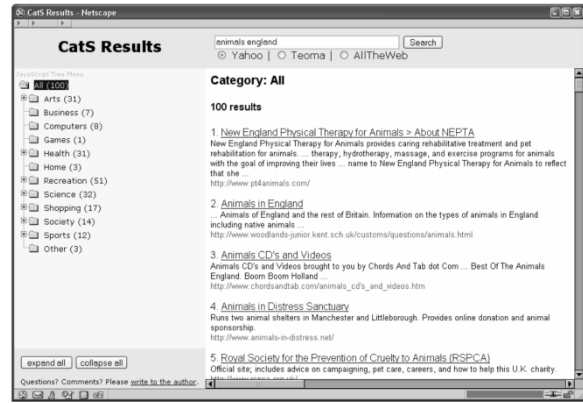


圖 2 Cats Meta-Search 引擎[8]

除此之外，推薦系統(recommender system)也可被視為有效存取有用資訊的重要工具之一。它們可分成 content-based, collaborative filter 以及 hybrid system 三大類，以下大略說明之：

1. Content-based system：藉由分析之前所瀏覽的網頁所得到的資訊，以及使用經由瀏覽相似內容的網頁，產生推薦內容。
2. Collaborative filtering system：經由計算不同使用者的相似度，將相似度相近的使用者所存取過的內容提供給使用者。
3. Hybrid system：適當的結合以上兩種方法。

Wei 等人[12]提出一個 market-based 推薦系統。它提供一個可以代表使用者的多代理人系統，讓推薦內容可以顯示在 sidebar 上。其他參與的代理人也會按順序的顯示在 sidebar 上，而 agent-initiator 會選擇的內容提供給使用者。提供的選項如果被使用者接受，就會得到獎賞。此代理人系統藉由採取多重階層的推薦方法做出最好的推薦內容。Yu 等人[13]提出一個多代理人的推薦系統，他們提出一個 MARS 知識管理轉送系統負責配置代理人給使用者，代理人藉由互動產生使用者所提出問題的答案。這個系統使用預先定義好的本體(ontology)，由各代理人一起分享，如此就可以很輕易的分享知識，共同解決問題。

Degemmis 等人[14]提出了一個結合 collaborative filtering 與學習使用者 profile 技術的推薦系統，此外，這個系統也結合了 collaborative 與 content-based 策略，藉由使用者的知識可以呈現在 profile 內，以及 collaborative filtering 演算法減少推薦產生的時間。Chau 等人[15]提出一個 collaborative 多代理人網頁探勘系統—Collaborative Spiders，藉由不同型態的代理人依據排程各司其職的擷取網頁內容、執行 post-retrieval 分析、與使用者互動、分享使用者搜尋的資訊、執行 profile 配對以及實施擷取與分析工作。

Zhu 等人[16]提出一個利用使用者行為模式去預測相關網頁的推薦系統—WebICLite，並將瀏覽網頁概念化為搜尋一個特定定義好的資訊需求(need)，而將這些需求識別為使用者瀏覽的網頁與作用在其他的動作(action)，並且參考與比較一些識別

information-need-revealing 樣式的特定演算法，並依序使用這些演算法用於推論使用者所使用的搜尋引擎的資訊需求。Macedo 等人[17]提出一個應用推薦系統除了支援使用者網頁瀏覽，也幫助推薦其他尋求推薦的使用者。而這個系統也允許使用者基於隱私權的考量，隱藏該使用者所搜尋的網頁一段時間。

3. 研究方法

本論文的目標是經由建構於社群網路的多代理人系統為基礎，進行收集與處理各使用者相關的資訊與喜好，設計一套適合社群網路的搜尋機制(Search Mechanism)，便於使用者可以快速、有效且準確地搜尋明確所需的資訊，進而開發一套發現機制(Discovery Mechanism)強化與發現出隱含不易發現的資訊。

多代理人系統所具備的 adaptivity, autonomy, social, personality 等特質，非常適合處理社群網路的相關問題。因此我們提出一個如圖 3 所示的多代理人系統架構，其中包含了 3 種型態的代理人：圖形使用者介面代理人(Graphic User Interface Agent, GA)、使用者代理人(User Agent, UA)、以及社群代理人(Social Agent, SA)。

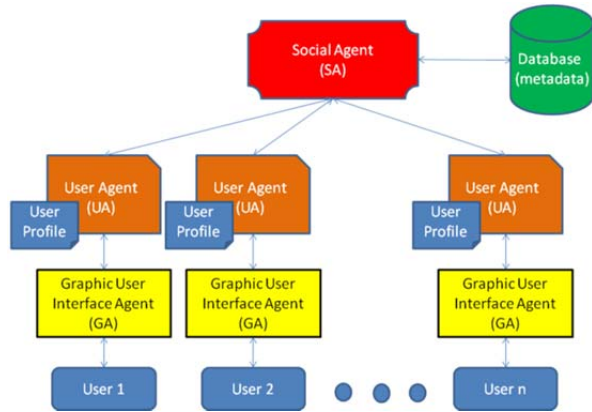


圖 3 多代理人系統架構

每位使用者都有一個 UA 與它所伴隨的 GA。在多代理人系統中，UA 就是設計為代表使用者。GA 的功能就是提供使用者與其 UA 之間的介面。GA 會提供一個圖形化介面，讓使用者透過系統所提供的功能與其他使用者溝通，並且使用圖形化介面顯示相關的資訊與個人化的訊息給使用者。而 SA 的角色就是提供社群網路服務所需的所有支援與協助，不像其他兩種型態的代理人(每一個使用者都會被分配一個 GA 與 UA)，SA 不屬於特定的人，對整個系統而言，它是一個管理者(manager)的角色，SA 統籌處理所有關於儲存、管理現存所有的使用者群組、協調所有系統中 UA 之間交換的訊息等。

社群網站的使用者原本就基於相同的興趣、共同的喜好、或是某些密切的關係，而在網路上群聚在一起的使用者，因此，本論文基於二個主要的基礎：物以類聚、共同的話題，建置本論文的系統架

構圖如圖 4 所示，主要內容包括：

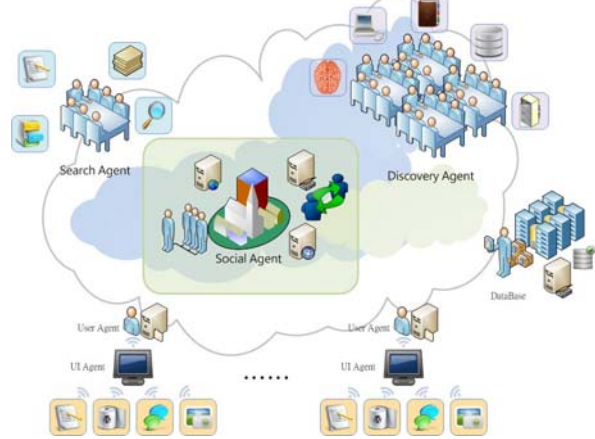


圖 4 系統架構圖

3.1 代理人服務機制之關係搜尋法

多代理人系統(Multi-Agent System, MAS)具備適應用於社群網路的重要特質，由代理人學習得知使用者與其他使用者之間的友好關係狀態，並依彼此的關係不斷演進而因應需要進行調整，代理人之間也會藉由合作與溝通完成任務。社群網路多代理人機制所開發建置的 UI Agent、User Agent 與 Social Agent，藉由代理人間相互的通訊與合作，收集各使用者相關的資訊與互動的密切程度，也藉由回饋機制，學習、改變與調整相關的個人化資料模型。

以社群網站最多人使用的 facebook 為例，facebook 的使用者可以透過 facebook 的設定，將某些關係指定為『家人』、『摯友』、『點頭之交』、『大專院校』、『高中』或者任職的公司，代理人以原有 facebook 的分類為基礎，再將使用者與不同使用者、不同關係之間的互動狀態隨時記錄下來，藉此分析並得知與我們互動最為密切的朋友或是群組。

假設以向量 r 來表示 facebook 裡設定的朋友關係(摯友、特定名單、家人關係、未分類名單...等)的係數，另外，向量 x_{ij} 其中的 i 代表發佈的近況、相片、部落格，其中的 j 表示對 i 按讚、訊息回應、同時按讚與訊息回應的次數，以 s_k 表示針對發佈的近況按讚、訊息回應、同時按讚與訊息回應的係數，以 p_k 表示針對發佈的相片按讚、訊息回應、同時按讚與訊息回應的係數，以 b_k 表示針對發佈的部落格按讚、訊息回應、同時按讚與訊息回應的係數，我們得到使用者的關係密切程度函數可以表示如下：

$$\begin{aligned} S &= s_1x_{11} + s_2x_{12} + s_3x_{13} \\ P &= p_1x_{21} + p_2x_{22} + p_3x_{23} \\ B &= b_1x_{31} + b_2x_{32} + b_3x_{33} \\ R &= rS + rP + rB \end{aligned}$$

我們再透過線性規劃的方式來找出它的最大值，如下：

$$\text{Maximize } R = rS + rP + rB$$

Subject to

$$\begin{aligned} s_1x_{11} + s_2x_{12} + s_3x_{13} &= S \\ p_1x_{21} + p_2x_{22} + p_3x_{23} &= P \\ b_1x_{31} + b_2x_{32} + b_3x_{33} &= B \\ x_{i1}, x_{i2}, x_{i3} &\geq 0 \\ s_1, s_2, s_3 &\geq 0 \\ p_1, p_2, p_3 &\geq 0 \\ b_1, b_2, b_3 &\geq 0 \\ r_1, r_2, \dots, r_n &\geq 0 \\ S \geq 0, P \geq 0, B \geq 0 \end{aligned}$$

由關係密切程度函數所得出的數值越大的，表示關係程度越密切，數值越小的表示關係程度越疏遠，因此，當使用者搜尋資料時，將以此為主要排序依據，由關係程度最密切的方向開始，依序往關係程度較疏的方向進行尋找，所搜尋到的資料內容也是關係比較密切的內容或資料來源。

3.2 時間軸之視覺化顯示

facebook 目前並未提供針對內容進行搜尋的功能，而且，以 facebook 現有的資訊呈現方式來看，主要分為『人氣動態』、『最新動態』二種，由於社群網站的型態，其資料更新的頻率非常快速，因此，facebook 預設是以『人氣動態』的方式呈現，將目前比較多人在回應的內容顯示在比較上方，主要目的在於吸引社群使用者的目光焦點，但是當使用者要尋找某些資料時，就只能使用捲軸往下拖曳，以眼球搜尋的方式自行尋找資料。

本論文除了搜尋使用者所發表的訊息內容，提供使用者可以隨時搜尋過往的資訊，另外一個主要目標在於提供使用者快速、有效且準確的搜尋機制，使用者不需要盲目的摸索與探詢、重複嘗試著不同的關鍵字，就能夠快速且準確地找到所需的資訊。因此我們發展適合社群網路之搜尋機制，藉由前述建立的使用者間的關係狀態，同時，在不同的時間點，社群裡討論的話題或是觀注的焦點會不斷的變換，但是以較為微觀的角度來看，在某些時間的區間，社群裡所談論的話題應偏向某些特定的主題，因此，我們基於前項的關係密切程度下，導入時間群聚的概念，觀察與使用者密切關係的群組裡，在某些時間區間裡較多的討論主題，藉以提高或命中使用者在搜尋時所期望出現的搜尋結果。因此，最容易且最有效的方式即以視覺化呈現再結合時間軸的方式，使用者可以在時間軸的呈現結果裡，一目瞭然搜尋結果的密集程度，越密集的就表示越有可能就是使用者要的搜尋結果。

4. 系統實作

4.1 關係密切程度函數

依據關係密切程度方式，可以將使用者彼此互動的狀況，帶入以下公式計算而得到其值：

$$\text{Maximize } R = r_iS + r_iP + r_iB$$

Subject to

$$\begin{aligned} 0.1x_{11} + 0.6x_{12} + 1.0x_{13} &= S \\ 0.1x_{21} + 0.6x_{22} + 1.0x_{23} &= P \\ 0.1x_{31} + 0.6x_{32} + 1.0x_{33} &= B \end{aligned}$$

以下說明各係數的狀況及代表的權重：

r_j 的係數表示如下：

$$r_1=1.0, \text{ 摯友}$$

$$r_2=0.8, \text{ 特定名單}$$

$$r_3=0.5, \text{ 家人關係}$$

$$r_4=0.2, \text{ 未分類名單}$$

$$s_1, p_1, b_1=0.1, \text{ 按讚的權重}$$

$$s_2, p_2, b_2=0.6, \text{ 以訊息回應的權重}$$

$$s_3, p_3, b_3=1.0, \text{ 同時按讚及訊息回應的權重}$$

最後計算出來的數值越大表示彼此互動的關係越密切，數值越小表示彼此之間互動的關係越少，透過關係密切程度函數所得到的值，做為系統搜尋資料時的主要依據。

4.2 時間軸之視覺化顯示

當網頁內容超過一頁時，大多會以縱向的方式呈現，不過，對於時間軸的概念而言，我們會比較習慣以橫向的方式，以 XY 二維平面來看，X 軸是時間，X 軸的左方是比較過去的時間，右方則是比較新的時間，因此，我們將前項搜尋的結果以橫向時間軸的方式呈現。

此部份的技術，我們將使用以 BSD license 釋出的 SIMILE Timeline Project，使用 JavaScript 方式，將資料以橫向時間軸方式呈現，同時，還可以具有大小二種不同的時間刻度，使用者可以直接滑動大小不同刻度的時間軸，如圖 5 所示，進行搜尋結果的瀏覽。



圖 5 搜尋結果以時間軸之視覺化方式呈現

5. 結論

社群網站的使用者快速增加，隨之而來所產生的資料量也跟著急遽增加，在一個以使用者為導向的資料搜尋，將變成非常困難的任務，本論文透過

代理人機制，定期收集使用者在網路社群裡互動的狀態，例如個人 status 的更新、發佈新的照片，以及其它好友的回應狀況，將狀態與內容儲存於本論文所建置的資料庫，將前述的彼此互動的狀態，透過關係密切程度函數經由線性規劃方法，計算出社群網站裡使用者之間的最大關係密切程度，當使用者透過本論文所建置的系統進行資料搜尋時，系統將得到的關係密切程度導入搜尋資料的動作，因此，藉以確保搜尋的結果將與使用者具有較密切的關係。最後，系統以視覺化時間軸的方式將結果呈現，提供使用者一目瞭然的搜尋結果。

本論文所提出的系統，可以提供使用者在社群網路龐大的資料進行搜尋時，萃取出與使用者密切相關的資訊。

致謝

本論文由國家科學委員會補助部分研究經費，計畫編號 NSC100-2221-E-143-001、NSC101-2221-E-143-003 及 NSC102-2218-E-143-001。

參考文獻

- [1] X. Ke, "A Social Networking Services System Based on the "Six Degrees of Separation" Theory and Damping Factors," Proceedings of Second International Conference on Future Networks (ICFN '10), pp.438-441, 2010.
- [2] FIPA, <http://www.fipa.org>
- [3] OMG, <http://www.omg.org>
- [4] J. Chen, "Research on Individualized Information Retrieval System Based on Multi-Agent," Shanghai Jiao Tong University, Master's Thesis, December 1, 2006.
- [5] M. Gori and I. Witten, "The Bubble of Web Visibility," Communications of the ACM, vol. 48, no. 3, March 2005.
- [6] S. Souldator, T. Dalamagas and T. Sellis, "Sailing the Web with Captain Nemo: A Personalized Meta-Search Engine," Proceedings of the ICML 2005 Workshop: Learning in Web Search (LWS '05), Bonn, Germany, Aug, 2005.
- [7] L. Yuen, M. Chang, Y. K. Lai and C. K. Poon, "Excalibur: A Personalized Meta-Search Engine," Proceedings of 28th Annual International Computer Software and Applications Conference (COMPSAC '04), vol. 2, pp.49-50, 2004.
- [8] M. Radovanovic and M. Ivanovic, "Cats: A Classification-Powered Meta-Search Engine," Advances in Web Intelligence and Data Mining, Springer-Verlag, vol. 23, pp.191-200, 2006.
- [9] S. Souldatos, T. Dalamagas and T. Sellis, "Captain Nemo: A Meta-Search Engine with Personalized Hierarchical Search Space," Informatics, vol. 30, pp.173-182, 2006.
- [10] N. Zacharis and T. Panayiotopoulos, "SpiderServer: The Meta-Search Engine of WebNaut," Proceedings of the 2nd Hellenic Conference on Artificial Intelligence, pp.475-86, 2002.
- [11] M. Kanteev, I. Minakov, G. Rzevski, P. Skovelev and S. Volman, "Multi-Agent Meta-Search Engine Based on Domain Ontology," Springer, vol. 4476, pp. 269-274, July 2007.
- [12] Y. Z. Wei, L. Moreau and N. R. Jennings, "Recommender System: A Market-Based Design," Proceedings of the second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '2003), pp.600-607, New York, NY, USA, 2003.
- [13] B. Yu and M. P. Singh, "An Agent-Based Approach to Knowledge Management," Proceedings of the eleventh International Conference on Information and Knowledge Management, pp.642-644, New York, NY, USA, 2002.
- [14] M. Degemmis, P. Lops, G. Semeraro, M. F. Costabile, O. Licchelli and S. Guida, "A Hybrid Collaborative Recommender System Based on User Profiles," Proceedings of ICEIS, pp.162-169, 2004.
- [15] M. Chau, D. Zeng, H. Chen, M. Huang and D. Hendriawan, "Design and Evaluation of A Multi-Agent Collaborative Web Mining System," Decision. Support System, vol. 35, no. 1, pp.167-183, 2003.
- [16] T. Zhu, R. Greiner, G. Haubl, B. Price and K. Jewell, "Behavior-Based Recommender Systems for Web Content," Proceedings of the 10th International Conference on Intelligent User Interfaces, New York, NY, USA, 2005.
- [17] A. A. Macedo, K. N. Truong, J. Antonio and M. Pimentel, "Automatically Sharing Web Experiences through A Hyperdocument Recommender System," Proceedings of the fourteenth ACM Conference on Hypertext and Hypermedia, pp.48-56, New York, NY, USA, 2003.
- [18] Wikipedia, "Linear programming", http://en.wikipedia.org/wiki/Linear_programming
- [19] Simile Widgets, <http://www.simile-widgets.org/timeline/>