

# IM Finder:透過即時通訊網路找尋問題解答

黃俊傑  
資訊工程學系  
國立中央大學  
Hugo@acnlab.csie.ncu.edu.tw

吳季偉  
資訊工程學系  
國立中央大學  
jihwei.wu@msa.hinet.net

江振瑞  
資訊工程學系  
國立中央大學  
jrjiang@csie.ncu.edu.tw

## 摘要

在今日人們運用網際網路找尋資訊或解決問題，已經成為日常生活的一部分。然而即使現在人們可以透過搜尋引擎、討論區、電子郵件等網路服務來尋求問題解答，卻仍然需要花費許多時間去搜尋、組織或是等待答案回覆。如果可以即時的找到具有足夠知識的線上回答者，將可以降低我們花在瀏覽網頁、等待討論區的回應者或是詢問朋友的時間。在這篇論文中我們介紹一個基於即時通訊網路的答案找尋系統 (IM Finder)，此系統可以透過即時通訊聯絡人所構成的社會網路，幫助找尋問題答案的詢問者，即時並精確找尋到具有足夠知識的聯絡人來回答問題。當使用者在找尋問題時，會利用其它使用者成功找到回答者的經驗以達到較好的準確度。本研究提出了一套精確並有效率地找尋適合的問題解答者的方法，並且在同儕式網路架構上實作出一個跨平台的軟體系統，以驗證運用社會網路來提升搜尋效果的成效。

**關鍵詞：**同儕式網路、社會網路、即時通訊、找尋專家、線上問題解答

## 一、緒論

在網路未普及之前，人們遇到問題可以從認識的人、專業諮詢或查閱書本以得到解答；時至今日，網際網路服務與技術發展蓬勃，人們可以應用各種網路服務如搜尋引擎、討論區、電子郵件等來尋求問題的解答，不但更方便迅速，而且內容豐富廣泛。然而運用傳統的網路服務找尋問

題的答案，使用者必須花費許多時間去搜尋、過濾、組織或等待解答，此外將問題轉換成正確的搜尋關鍵字或找到合宜的討論區是成功獲得解答的關鍵，然而這對於一般使用者而言並不是容易的事。

使用即時通訊軟體 (Instant Messenger) 直接向朋友或有能力解答問題的人詢問，是更具有個人化、整體關聯性與互動性的方法。使用者可以針對問題透過即時的互動，直接的尋求或提供答案，因此如果有一個具足夠知識的回答者，可以來幫忙回答問題，那麼將可以節省許多尋找或是等待的時間。然而每一個即時通訊使用者的聯絡人清單，通常受限於他們所熟識的人，所以即使人們可以直接的詢問聯絡人，問題的散佈仍然會被侷限在熟識的人之間。如果可以把不同使用者的人際關係串接起來，則更具搜尋效益的社會網路就會出現，也就是說使用者可以從「朋友的朋友」中搜尋有效的資源，而不是僅受限於一層的人際網路。這種遞迴式的搜尋與轉傳機制使問題不僅只在自己的聯絡人清單傳遞，而是在一個集合許多可能具有經驗和具有知識的人際網路上散佈，當然有效找尋到問題解答者的機會就會大的多。在這樣一個串連許多人的聯絡人清單而形成的人際網路裡，信任與可靠度也比較容易被建立起來，因為參與者的關係並非完全虛擬而不可知的，他們是由實際的社會人際關係轉化而成，是結合社會網路 (Social Network) 觀念而成的技術。因此我們希望可以在這種合作式的即時通訊網路，也就是一個由「朋友的朋友」所建構起來的社會網路上，能夠即時的找到有能力解答問題的線上使用者。

在這篇論文中我們提出一個建構在同儕網路(P2P Network)上的即時問題解答者搜尋系統-IM Finder, 利用搜尋使用者對特定問題的問答記錄以及來自於使用者的推薦, 來得到最有潛力回答問題的線上使用者名單, 以能夠即時提問並獲得有效的解答。與其他網路人員檔案尋索系統不同的是, IM Finder 著重於尋找能夠即時回答問題的線上使用者, 透過即時通訊系統以自然語言互動式的立即提問與提供解答。隨著系統的運作, 社會網路的結構也會發生變化, 例如某個問題的詢問者和解答者之間可能成了「直接認識的朋友」, 換句話說人際關係網路的範圍是日漸擴大的, 當然日後潛在的問題解答者也會愈來愈多。要實作IM Finder, 有二個主要的問題必須解決: (1) 針對特定問題, 要如何找到最適合的解答者? 要如何評量問題與眾多候選解答者的適配程度? (2) 如何用最少的時間和資源找到適合的解答者? 在本論文中我們設計了一套精確的程序解決這二個問題, 並實作了一個跨平台的IM Finder系統。

本篇論文的其它部份將分成以下各個章節: 第二章說明研究的相關背景知識以及文獻探討; 第三章則介紹我們所提出來的系統與方法; 第四章針對相關的議題深入討論; 第五章為系統實作; 最後第六章為結論和未來研究方向。

## 二、相關研究

我們希望能夠在基於即時通訊的社會網路上搜尋出適合的線上使用者來回答問題, 因此我們將分別介紹同樣是在即時通訊網路中搜尋使用者的兩套系統: 小世界即時通訊和同伴發現者, 以及在社會網路上搜尋的相關策略。

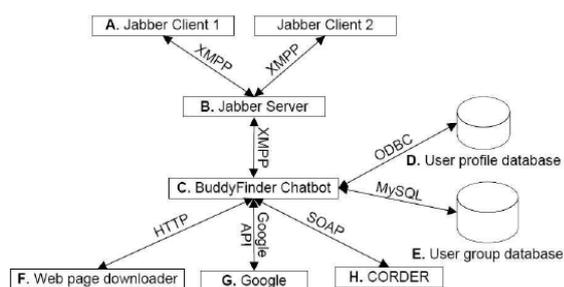
### 2.1 小世界即時通訊(SWIM)

小世界即時通訊(SWIM, Small World Instant Messenger) [2] 為一個新的即時通訊軟體, 主要著重在可以透過基於即時通訊的社會網路找尋資訊, 它具有所有一般即時通訊軟體所具備的功能, 除此之外又加入了兩個額外的功能: 第一個為SWIM除了讓使用者可以手動輸入自己的專長

和興趣之外, 也可以自動的從使用者的個人首頁、瀏覽過的書籤等建立可以代表使用者特徵的關鍵字向量; 第二個為SWIM有一個內建的推舉代理人(referral agent)負責自動處理所有查詢資訊。使用者會先把查詢送給他的推舉代理人, 之後推舉代理人會廣播這個查詢給所有他朋友的代理人, 當朋友的代理人收到查詢時, 他會先搜尋自己資料欄位的關鍵字向量, 如果沒有任何的比對成功他將會把這個查詢再轉送給他的朋友。查詢將會沿著即時通訊網路直到比對成功或是超過了原先所設定的轉傳個數。這查詢和路徑就表示了目標人物可能知道答案, 於是這兩個人就可以開始立即的交談, 或是如果回答的人沒有時間, 他們可以在之後再做討論。基本而言SWIM為一個同儕式架構, 在他們自己的資料和喜好上給予使用者完全的控制權, 同時它也考慮了隱私權的保護和分散式控制。

### 2.2 同伴發現者(BuddyFinder)

同伴發現者(BuddyFinder) [7] 主要架構在Jabber的即時通訊平台上, Jabber使用者和伺服器間主要使用稱為XMPP的XML通訊協定來做溝通, 整個系統架構如圖一。當Jabber使用者A發出一個查詢時, 此查詢將會使用XMPP的通訊協定來做傳送, 接著這個查詢會經由伺服器導向至BuddyFinder的聊天機器人上(也就是C), 負責和各個不同的模組溝通以得到朋友間排序的名單, 並且回傳這個結果。在D中的使用者個人檔案資料庫會儲存一連串的網頁連結, 一開始使用者的網頁連結會透過Google API自動的使用網頁搜尋元件, 為了可以簡單的初始化這裡利用聰明的猜(intelligent guess)找出可能相關的。所謂聰明的猜是基於使用者的電子郵件名字和它的網域名稱, 如使用者的電子郵件為m.eisenstadt@open.ac.uk 以及名字為 Marc Eisenstadt, 則系統會將 Marc Eisenstadt site:open.ac.uk送給Google, 並且以前十個網頁當作使用者的預設個人檔案。



圖一：BuddyFinder 系統架構

E是儲存使用者所屬群組的資料庫，每個使用者可以屬於不同的群組。當使用者搜尋查詢時系統會搜尋一連串朋友的個人檔案，因此即使是相同的查詢也可能因為不同的使用者而有不同的結果。F則是當做搜尋的快取，在使用者個人檔案中的網頁都會先被下載到F，如此做可以改善搜尋的效能。由於使用者可能會收到大量的排序名單，所以這裡提出了一個CORDER的演算法來計算排序的問題。CORDER主要從同時出現（Co-occurrence）、距離（Distance）、以及頻率（Frequency）三個參數來考慮，同時出現指的是兩個目標物件一同出現在同一個網頁，若一同出現的機率愈大，則會被認為兩個物件愈相關。距離指的是在同個網頁中兩個目標物件所在的位置距離，如果愈近的話會被認為愈相關。頻率指的則是如果目標物件愈常出現在網頁中的話，會被認為是比較重要的。於是排序過的名單就利用查詢和使用者名字間關係的強度來決定其相關性，而關係的強度則是取決於上述的三個參數。

### 2.3 社會網路搜尋

社會網路（Social Network）主要為一個網路社會化的概念，Barry Wellman [17]指出網際網路上的「虛擬社群」不見得一定會和「真實社群」相對立，網路的社會形態中，每個節點仍是以人為主體，網路社會化的核心概念就是實體「人」的網路化。社會網路由許多節點所構成，節點可為個人或是組織，它們由一個或多個特定的關係所連結，如價值觀，友誼，親情，衝突或金融交易，因此產生的結構往往非常複雜。彼此之間的關係由一條有向性的連線表示，是一個抽象的網路，在其最簡單的型式中，社會網路是個所有有

關係節點的地圖，可以圖型化來呈現並可用量化的方式，藉此分析個體與個體、個體與群體、群體與群體之間的互動關係及影響，以及一些其它應用，比如說它可以用來做網路中相當重要的信任度（Trust）之依據，也可以輔助一些網路社群的交流、互動等。

在社會網路中的搜尋以「小世界」（small world）[11]實驗最為大家耳熟能詳：在1960年代哈佛大學的心理學家Stanley Milgram等人發現透過自己認識的人可以成功的從內布拉斯加州寄包裹到波士頓，實驗發現即使人們只擁有一些有關他們熟識的人的局部知識，但他們卻會從熟人中挑選出他們認為最接近目的地的人做為傳送包裹的依據，並且人人皆如此。這個實驗發現任兩個人在世界上鏈結的長度不會超過6，意即最多通過6個人你就能認識任何人。這說明了社會網路為可搜尋的特性，這結果即為有名的「六度分隔理論」（six degrees of separation）。隨著網際網路的蓬勃發展，近來線上的社會網路工具和服務，在網路上極度的受到歡迎，透過軟體可以建立使用者社群的線上社會網路，在其中使用者可以和其餘擁有相同興趣或活動的人進行互動，而該服務就是提供了許多各種不同的互動與管理，例如訊息傳送、語音交談、小遊戲的進行、以及網誌和影音的分享等等。其中又以MySpace[14]和Facebook[4]擁有數千萬的活躍使用者最為知名。

即時通訊也可算為社會網路服務的一種，也可以說是架構在朋友到朋友的網路之上，其中每個使用者會有自己的聯絡人清單，可以手動輸入自己的聯絡人清單，允許訊息傳送、語音交談、群組聊天、會議服務、談話記錄以及檔案傳輸等動作。即時通訊的技術使得兩個或兩個以上的參與者，可以透過網際網路實現基於即時文字訊息的溝通，有別於以往的電子郵件非同步的訊息往來。有些系統更近一步的提供了離線訊息的服務，即使使用者離線，在下次登入時依然可以收到訊息，也可以替代傳統電子郵件的連絡。除了提供文字上的溝通以外，即時通訊系統還提供多媒體通訊服務，使用者除了文字以外還可以通過聲音，視訊等等和其他使用者進行溝通，而以上

種種的功能使的即時通訊軟體成為一種新的通訊主流。目前有許多不同的即時通訊軟體如 Windows Live Messenger[13]、Google Talk[8]和 Pidgin[15]等，即時通訊的網路可說是目前最大的數位人際網路之一，有超過上億的使用者[5]，其中隱含了大量的線上人力資源。

社會網路中搜尋的相關研究可分為找尋專家問題以及不同的搜尋策略二個方向。找專家問題又可稱為專家位置 (expertise location) 或專家找尋 (expert finder)，往往我們需要經過一連串的屬性比對才能知道。目前有一些找專家的系統被發展出來，如[18]透過搜尋電子郵件資料檔案來找到人，以及專家推薦[10]使用區域有意義的資料去推薦一組潛在的可能答案。而在社會網路中搜尋有以下不同的策略，分別有最好連結搜尋 (Best Connected Search) [1]、強/弱連結搜尋 (Strong/Weak Tie Search) [9]、漢明距離搜尋 (Hamming Distance Search) [6]和資訊氣味搜尋 (Information Scent Search) [19]等。在最好連結搜尋中節點會選擇分支度較高的節點來幫忙轉傳請求，由於分支高的原因所以有機會使的查詢可以快速的擴散，此外有研究發現在所有的網路中，這個策略並不是一定有效。在強/弱連結搜尋中，強連結表示的是從現在的使用者中，選擇日常生活裡接受到最多訊息的人做為發送請求的對象，而弱連結則是相反它會選擇接受最少訊息的人。漢明距離會挑選和現在的使用者中擁有最不相同朋友的人，做為發送請求的對象。資訊氣味搜尋會在搜尋路徑上沿路留下社會能力值 (sociability)，所以每次會挑選出和查詢比對的最高分者。

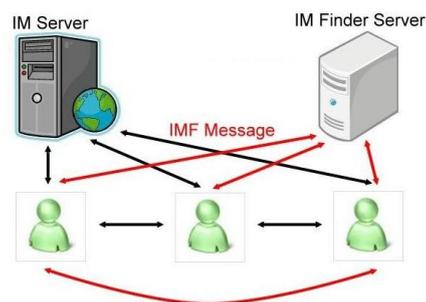
### 三、系統架構與設計

我們的目標是在即時通訊網路中找到一個在線上的最適回答者，主要的想法為每個使用者也許都只有有限的聯絡人，而一種透過聯絡人的聯絡人的搜尋方式，可以使得在找尋潛在的回答者上更為有利。搜尋的準則主要是基於在過去曾經回答過的問題，利用相似度的測量來決定查詢和他們之間的關聯。如果之前的問題沒有存在，那麼將會使用關鍵字和個人的興趣比對替代。和

專家系統較為不同的是，使用者是否具回答該問題的能力，並不完全來自於由使用者自己所提供的資料，而是根據該使用者在系統上實際回答過的問題所決定，這樣做的好處是可以透過許多人的角度以客觀的方式來決定回答者的知識資料。

爾後當使用者在系統上提出問題時，問題將會經由聯絡人鏈結散佈，去收集現在有空且曾經回答過相關問題的使用者，接著詢問者會從排序好的清單中，挑選潛在的回答者，進行即時的語音或文字交談。如果詢問者對於回答的結果滿意，那麼詢問者的問題和回答者將會被記錄下來，以便增進未來的搜尋。透過聯絡人執行遞回式的搜尋，可以有效的收集夠多回答者的個人特徵。我們假設加入「人」的自我判斷，可以比純粹的機器分析更加準確，以及相較於幫助陌生人，使用者也許會比較願意幫助熟識的人，所以在搜尋期間使用者也可以有選擇性的推薦其它適合的回答者。此外為了避免較受歡迎的人(例如回答問題的能力很強)負擔太重，當回答者在回答問題時他們將會被排除於搜尋清單中，不致於太多的問題集中在少數回答者身上。

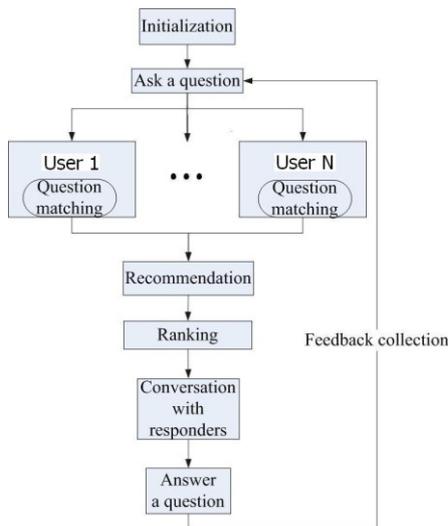
系統主要架構為同儕式 (P2P Network) 系統，但是仍然需要一部 IM Finder Server 伺服器用來做為帳號管理、用戶端的起始化以及使用者個人資料的追蹤，整個系統架構如圖二。



圖二：系統架構

原來的即時通訊使用者除了連原本的即時通訊伺服器之外，還會連上IM Finder伺服器，如此一來可以讓即使通訊使用者和非聯絡人清單上的其餘使用者進行同儕式溝通。

同儕式部份為使用者直接連線他們的聯絡人以執行真正的搜尋，回應者的搜尋則為用戶端在自己的機器上執行運算。首先每個使用者會提供一些他們願意回答的關鍵字去建立他們的個人資料，在搜尋回答者的期間，問題會先被送到由詢問者的朋友組成或由系統所提供的聯絡人清單中的人，問題將會被轉傳一個預設的深度（也就是聯絡人的層級，這裡預設是3，不過可以調整）。當聯絡人接收到問題時，將會就詢問者的問題和之前該聯絡人曾經問過的問題並成功得到答案者進行比對，此外聯絡人也可以從他們的人際網路中就目前的問題推薦一個可能的回答者。經過遞迴式的收集潛在回答者後，詢問者可以從排序好的名單中挑選以及開啟即時的交談。一旦交談結束如果詢問者覺得滿意，系統將會把問題和回答者記錄在詢問者節點上，以幫助未來其它的詢問者。我們接著描述系統主要過程如圖三，以及各個不同階段及細節：



圖三：系統執行流程

整個執行流程大致可分為問題比對、推薦、排序以及回饋收集，以下將各別論述之。

### 3.1 問題比對

詢問者將想問的問題放在系統上時，問題可能不會和之前被回答過的問題一樣，所以當聯絡人收到一個問題時，問題和每一個之前聯絡人本身所問過的問題，將會執行相似度的測量

(similarity measure)，如果量測後的相似度超過某個特定的門檻值，系統將會把聯絡人（之前的詢問者）、回答者、相似值以及問題本身加入推薦人清單中 (recommend list)。之前的詢問者也要記錄的原因，主要是將他們也視為了可能的潛在回答者，所以有助於幫助減輕較受歡迎的回答者負擔。在這裡相似度的測量是採用向量空間模型 (Vector Space Model)，此為典型的資訊擷取最常採用的技術，主要為將查詢拆成向量的詞語，再利用餘弦公式計算其查詢和曾經問過問題的角度，計算公式為：

$$Q\Delta q = \frac{\sum_{i=1}^n Qi qi}{\sqrt{\sum_{i=1}^n Qi^2} \sqrt{\sum_{i=1}^n qi^2}}$$

其中Q為查詢而q則為使用者曾經問過的問題，經過計算可以得到一個相似值。

### 3.2 推薦

有了自動的搜尋，我們認為人的判斷仍然可更準確，於是我們還在推薦人清單中加入了人的判斷，所以除了問題的比對，使用者也可以手動的推薦他們認識的朋友或自己為回答者，這也許可以增加問題被解決的機率，因為人們通常對問題會擁有比較好的一連串相關知識，甚至知道其它聯絡人的知識背景，此外當問題來自於自己的朋友或是他們所熟識的人，使用者可能也會比較願意幫忙。聯絡人推薦適當的朋友稱為推薦人，然而為了預防不當的使用者推薦不適合的使用者，每個推薦人都會有個評價值，當他們成功推薦時，評價值也會跟著上升。

### 3.3 排序

推薦清單包含了候選回答者、之前的詢問者、推薦人以及詢問者和回答者之前曾經回答過的問題的相似度，接著推薦清單的排序基於

1) 關聯度：現在和之前問題間：

$$relevance = \sum_{i=1}^n sim(Q_i)$$

其中 $Sim(Q_i)$ 為詢問者和回答者之前已經回答過的每個問題 $q$ 。

2) 信心度：推薦回答者的推薦人信譽值：

$$confidence = \sum_{i=1}^n app(introducer)_i$$

$n$ 表示推薦某個特定回答者的推薦人數量以及 $app(Introducer)_i$ 為第 $i$ 個推薦人的信譽值，因此總和表示了某推薦人可多相信的程度。

### 3.4 回饋收集

一旦詢問者和回應者間的對話結束，詢問者也許會給個評估，為了簡化這個評估我們只使用正或負的值來表示使用者的滿意程度。如果評估值為正，回答者的推薦人的評價值將會上升，回答者和所問的問題也將會被記錄在詢問者上，使得未來的搜尋更為便利。

### 3.5 增強

為了使系統能實際的運作，我們接著描述三個增進系統機制的部份。

**匿名式詢問：**由於系統主要是透過人際網路去搜尋解答，所以問題有可能會被轉傳到認識的朋友上面，但是詢問者也許不會想要讓某些朋友知道。針對隱私的考量，我們也可以讓使用者可以匿名的問問題，在該模式下詢問者的名字在轉傳的過程中，將不會出現在問題上。

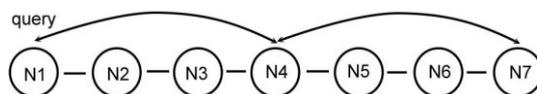
**黑名單：**惡意的使用者有可能會存在人際網路中的不同層級。為了預防惡意的使用者干擾正常的使用者操作，他們可以將惡意的使用者加入黑名單中，之後搜尋時就可以從中過濾掉可能的惡意使用者。

**自動回應：**被挑選到的回應者有可能正在忙碌而無法回答問題，因此若在這之前該回應者願意將當時的問答記錄給留下的話，系統會自動的回傳該次與其它使用者的問答記錄供詢問者參考。

## 四、討論

IM- Finder的設計主要分為中央集權式和同儕式兩個組成部份，它採同儕式設計方式主要有兩個原因 1)問題以及聯絡人的收集可能是較為隱私的資料，使用者可能只希望跟親密的朋友或是熟識的人分享，這些資料假設不用被集中化，系統就不需要只依靠在一個值得信賴的伺服器端上。 2)同儕式系統允許計算以及儲存資源可以平均被用戶端所分擔，使得高度可擴展和較可負擔的起的系統可以創建。然而某些工作如帳戶管理，或一些系統效能的資料收集，則具有安全性和公平性上的考量，因此這些工作還是被保留在伺服器端處理。但是如果所有的工作都是集中在伺服器端上，成千上萬的查詢或是發問的同時，伺服器端可能將需要大量的資源來做相似度的計算。

使用者要發送問題時，問題的轉傳深度也是個議題，在這裡系統的設定上，主要是對聯絡人清單上的所有朋友做廣播發送，由於廣播對整個網路而言會造成許多不必要的訊息浪費，而我們又希望可以讓較多的使用者可以收到問題，以增加問題被回答的機會，所以我們採用有深度限制的廣播策略，而其中訊息存活時間(TTL)設為3。因為有相關研究[20]指出當訊息存活時間為3時可得到最大的成本邊際效益，也就是可以使得整個網路的覆蓋成長率最大，而冗餘的訊息成長率最小。然而雖然訊息存活時間設為3，但系統找到的回答者仍可以超過這個深度，如圖四。



圖四：系統查詢

當N1發送查詢轉傳到N4時，假設N4在之前問過類似的查詢，並且在該次的查詢中找到N7，因此N4就會存有該次查詢的記錄，並且在收到N1的查詢時，將N7回傳回去。另外還有一些其它相關的議題討論，分述如下：

### 4.1 議題一

隨著網路的蓬勃發展，網頁的技術也不斷的更新，從1.0到2.0甚至在最近還有人提出3.0的構想。最早的1.0所有的資源以及網頁內容都掌控在伺服器手中。到了2.0時代，網頁被視為一個平台，使用者變成可以直接參與網頁內容的建立為其最主要的特徵，並且藉由人與人(People-to-People)的分享，而形成了現在大家所看到多彩多姿的網路世界，在其中人與人的互動所產生的內容，在這個平台上可以輕易的被發佈以及管理，有許多著名的web2.0的代表網站，如Flickr、Facebook、youtube，其中就分別提供了照片、影音、網誌以及檔案的分享和管理等等，而且通常這些網站都還會提供客制化的服務，這也可以說是web2.0的一項特點之一。

由於web2.0有很多的網頁內容是經由人與人之間的互動所產生，所以如果將網頁當成結點，網頁間的連結視為一種關係，則其形成的網路也可稱為社會網路的一種，因為社會網路所強調的概念就是在於人與人之間所形成的網路。只是由網頁間所形成的網路間的連結，其關係除了可能是朋友之外，還有可能會是來自於一些共同的興趣或喜好，假如可以分析這些連結的關係，進而可以取出一些跟知識相關的資料，以供為回答問題時的參考。甚至透過網路社群的力量，讓我們可以更方便的找到每個線上使用者，如此一來可以增加更多的使用者。

## 4.2 議題二

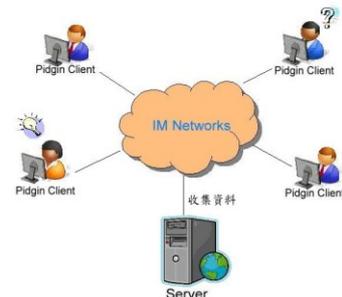
在問題比對方面，由於現階段關於語義相似度的做法仍有限，其中可以知識本體論(Ontology)以及WordNet[12]做為代表。知識本體論最早出現在哲學用語，探討的是"存在"的學問，以及萬物之間的關係，所以又可稱為存在學或本體論。在電腦資訊科學上，知識本體論指的是資料模型，並且透過物件導向的概念來描述之。簡單來說可以將資料分成各種領域，並且有個自不同的屬性，並且為其定義之間的關係，可將其視為資料描述的方法。所以透過Ontology或許可以實現語義比對上的可能性，只是Ontology的建法，通常是為了某個特定的領域而建，而且每個人所建出來的Ontology也不盡相同。

WordNet是另外一種選擇，它是由一個語言研究計畫所建立出來的語義字典，在其中字詞與字詞間若有語義上的類似關係，之間便會有連結產生，之後構成一個類似樹狀結構以供查詢用。就有研究[21]以WordNet做為基礎，進行句子與句子間的相似度比較。在這方面我們系統的焦點著重於可以開發出一個線上找人的系統，至於問題比對方面，我們期望未來能有更好的方法出現，直接可以取代現有的問題比對機制，以增加系統的效能。

## 五、系統實作

### 5.1 開發環境

我們的系統是建構在一套跨平臺的開放原始碼的即時通訊軟體平台—Pidgin[15]為其開發擴充套件(plugin)，稱為IM Finder，整個環境架構如圖五。



圖五：環境架構

Pidgin 能跨多個平台包括 Microsoft Windows、Linux、FreeBSD、Mac OS、SkyOS 以及 Trolltech's PDA system 等...，並且支援各種不同的通訊協定如 AOL Instant Messenger、ICQ、IRC (Internet Relay Chat)、Jabber、MSN Messenger 和 Yahoo Messenger 等十幾種不同的即時通訊協定。除了在 Microsoft Windows 平台之外，Pidgin 是一個最受歡迎的即時通訊軟體之一，截至目前為止 Pidgin 約有三百萬用戶[16]，它允許程式開發者可以開發自己的擴充套件，並且釋放給一般的使用者使用。Pidgin 可以在 Cygwin[3] 平台上開發，它在 Windows 作業系統上

建立一個類似Unix的環境，執行Unix的指令及眾多的應用程式。

Pidgin幾乎包含了一般即時通訊軟體的功能，例如聯絡人清單、檔案傳輸、交談記錄甚至分頁交談視窗的功能。使用者可以在Buddy List上新增聯絡人，為他們設定別名以方便記憶聯絡人的名稱。同時也支援不同的即時通訊協定，讓使用多個即時通訊協定的使用者，也可以同時登入不同的協定。

在模組上Pidgin主要可分為libpurple、Finch (GNT User Interface)以及Pidgin (GTK+ User Interface) 三個模組。其中libpurple中包含了所有實作即時通訊軟體程式會使用到的應用程式介面(API)，而Finch和Pidgin的差別，則是在於Finch使用GNT函式庫的使用者介面，Pidgin則是使用GTK+函式庫的使用者介面。

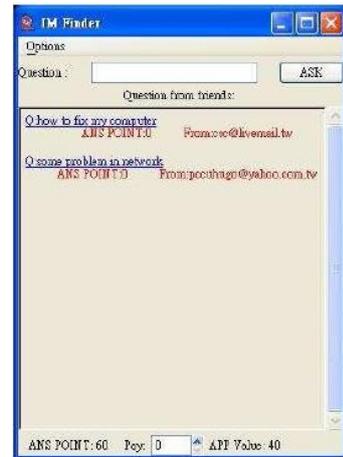
## 5.2 系統展示

在這小節中我們將介紹我們的系統。



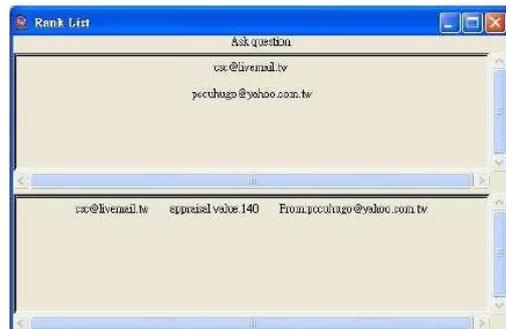
圖六：pidgin畫面

圖六為Pidgin中的朋友清單畫面，使用者可以隨意點選朋友清單中的聯絡人，並且開始進行文字的對談。



圖七：IM Finder 畫面.

圖七為IM Finder中的主畫面，使用者可以在框框中輸入欲詢問的問題之外，在下面也可以看到來自於其它使用者的問題，以及自己的評價值 (APP Value)，和決定是否要使用知識點數(ANS POINT)來誘使其它使用者回答問題。



圖八：排序畫面

圖八為收到曾回答類似問題的使用者清單以及來自朋友們的推薦，由於來自於朋友所推薦的使用者可能尚未真正回答過該問題，所以在系統設計上，分為上下兩個部份，上面是曾經回答過類似問題的使用者，下面則是來自於朋友的推薦，並會依照各自的Relevance和Confidence做排序。

## 六、結論

在這篇論文中，我們實作了一套可以讓使用者在社會網路上找人問問題的系統，我們的系統允許詢問者在即時通訊網路中透過搜尋聯絡人

所形成的社會網路，基於之前曾經回答過的問題和推薦的機制去找尋適合的回答者。它會透過分散式的方式，在自己的社會網路上去收集有關回答者的知識資料。所以即使是相同的使用者詢問相同的問題，也有可能找到不一樣的回答者，我們認為一個可以回答問題的回答者，除了對於問題基本的知識之外，可能還需具備其它的因子，例如熱心、以及溝通能力等等...換句話說，分散式的收集回答者的知識資料，也等同於是透過自身的社會網路，去建立出一個回答者回答問題的評價，而這樣的評價是較為客觀的。在系統中潛在的回答者主要來自於 (1)系統選擇，找到曾經回答過類似問題的人，以及 (2)手動推薦，透過使用者的認知去挖掘比較好的回答者。接著互動的結果將會被回饋給系統，去增進知識庫和推薦者的信譽值。隨著愈來愈多的參與者，搜尋也許會變的愈來愈準確且具效率。藉由將問題貼標籤也許可以縮窄搜尋的範圍，使得只有和某領域相關的相似值須被比較。我們期待系統的成功運作可以創造一種新的網路社交模式。

在未來的研究，本系統可以朝向一個分散式的知識系統為設計方向，也就是說每個使用者在詢問或回答過某個問題之後，可以將問題和答案分析整理並且轉換成為自己的知識庫資料，如此一來問題的詢問與回答者將轉換成累積的知識資產。最後在問題比對上若能做到語義上的相似度比較，也可拿來取代現有系統的比對方法，將有助於系統效益的提昇。

## 六、參考文獻

- [1] L.A. Adamic and Adar. E. How to search a social network. In *Social Networks*, 2005.
- [2] M Van Alstyne, J Zhang. Swim: fostering social network based information search. In *Conference on Human Factors in Computing Systems*, 2004.
- [3] Cygwin. <http://www.cygwin.com/>.
- [4] Facebook. <http://www.facebook.com>, 2007.
- [5] R. Farmer. Instant messaging – collaborative tool or educator’s nightmare! In *The North American Web-based Learning Conference*, 2003
- [6] K. Faust, D. Iacobucci, S. Wasserman. and M. Granovetter. *Social network analysis: Methods and applications*. 1994.
- [7] A Goncalves, C Denham, J Zhu, M Eisenstadt. Buddyfinder-corder: Leveraging social networks for matchmaking by opportunistic discovery. In *Semantic Web Conference*, 2005.
- [8] Google Talk. <http://www.google.com/talk/intl/zh-TW/>.
- [9] Q. Jones., L. Teryeen, S. Whittaker. Contact management Identifying contacts to support long-term communication. In *ACM Conference on Computer Supported Cooperative Work*, 2002.
- [10] D. W. McDonald and M.S Ackerman. Expertise recommender: A flexible recommendation architecture. In *Computer-Supported Cooperative Work*, 2000.
- [11] Stanley Milgram. The small world problem. *Psychology Today*, 1967.
- [12] G.A. Miller. Wordnet: A lexical database for english. In *Communications of the ACM*, 1995.
- [13] MSN. <http://tw.msn.com/>.
- [14] Myspace. <http://www.myspace.com>, 2007.
- [15] Pidgin. <http://pidgin.im/>, 2007.
- [16] Pidgin. <http://en.wikipedia.org/wiki/Pidgin>.
- [17] Barry Wellman, Janet Salaff, Dimitrina Dimitrova, Laura Garton, Milena Gulia, Caroline Haythornthwaite. Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community. *Annual Review of Sociology*, 22:213-238, August 1996.
- [18] L.Yenta Foner. A multi-agent, referral-based matchmaking system. In *International Conference on Autonomous Agents*, 1997.
- [19] B. Yu and M.P. Singh. Searching social networks. In *Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.
- [20] X Zhang H Wang. S Jiang, L Guo. Lightflood: Minimizing redundant messages and maximizing the scope of peer-to-peer search. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, 19(5):601–614, 2008.
- [21] A. Zuhair, J. D. Bandar., O. K. Crockett, Y. Li, D. McLean. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.32