

以多主題建置個人化解答與專家之推薦系統-以Yahoo知識+為例

顏以豪
成功大學資訊管理研究
所研究生
david2300409@gmail.com

王惠嘉
成功大學資訊管理研究
所副教授
hcwang@mail.ncku.edu.tw

摘要

網際網路的快速發展，網路幾乎已經成為大家尋找資訊、學習知識的主要管道，許多知識性質的社群網路也隨之興起，以互動式問答的方式成為人們尋求與分享資訊的主要平台；然而，當平台的知識量急速增加，資訊過載的問題也造成了使用者的不便，學習者不易在平台中獲得所需資訊，知識提供者面對過多重複性的問題也失去了回答的意願。

因此，本研究將專家推薦系統結合文件多分類分析、投票模型及向量擴張技術，針對使用者的問題個人化推薦平台中的可能知識提供者與解答，並以知識平台 Yahoo! 知識+ 為例，分析平台中的問題與使用者資訊，提出一個同時提供解答與專家的個人化推薦系統，並以 MRR 驗證推薦準確度。
關鍵詞：專家搜尋、資訊擷取、個人化推薦、社群網路、投票模型

1. 緒論

網路科技的發達，網路上的資源越來越豐富，人們的知識大量的轉換成編碼後的資訊以各種形式存在於網路空間中，例如：論文、報告及任何人們所發表的文字資料[16]，編碼後的文字資料也代表了每個人的專業知識或者有興趣的議題，例如：當某位使用者在程式設計領域的知識平台發表文章進行分享，也就意味著該使用者對於程式設計的領域擁有其專業知識或者對於程式設計的知識有所需求，需要尋求平台中的其他專家協助。

隨著社群網路的興起，各種領域紛紛建立該領域專屬的分享平台，以提供有興趣、喜好、經驗、專長相似的群眾聚集在

一起，以一個共同的空間進行資訊的分享或情感的交流；然而，資訊快速的累積使得人們要從大量的資訊中得到自己需要的部分變得相當困難，造成了資訊過載的問題。

知識平台也是一種社群網路，隨著平台的領域不同，使用者可以在平台中找到相關的知識解決問題，而知識平台上的資訊交流來自於使用者間的發問與回答，發問者會在平台上提出自己的問題並等待其他專家們的回答，隨著問答紀錄的累積，同樣有著資訊過載的問題，重複發表的問題目過多，專家們需要花費時間去搜尋，發問者獲得解答的時間相對地增加，因此，如何從這龐大的知識庫中準確找出所需的資訊，以解決使用者的問題提供個人化的服務，成為一個急需解決的問題，Liang, Yang, Chen, & Ku[13]也提到要提供個人化的服務，對使用者的了解與正確的推薦是非常重要的。

在過去的研究中，個人化的推薦系統應用的層面相當廣泛，例如：網頁[8,21]、電視[3]、音樂[7]、電子商務[16]等人們平日會接觸到的服務都已經有很多的研究提出各種分析使用者喜好的方法，而在知識領域方面，Liang et al. [13]以語意網配合蔓延激發模式 (Spreading Activation Model, SAM) 推論使用者所用的真正意圖；Liao et al.[14]針對使用者建立其知識本體 (Ontology)；Yimam-Seid & Kobsa[26]也提出了以下五種使用者可能需要尋找專家的情境：

1. 需要未被文件化的資訊
2. 有無法解決的問題需要專家的協助
3. 善用其他成員的專業
4. 針對不了解的資訊需要專業人員進行解說

5. 使用者期望與人互動而非單純與文件和電腦互動

在專家推薦系統的研究中, Macdonald & Ounis[18]利用組織中成員的文件資料建立組織內部的專家搜尋機制, 提出文件中的人名與文件內容知識的關聯性; Tseng & Weng[22] 則利用學習性的論壇(Programmer Club)中每位學習者(learner)與專家的發表文章配合網站架構建立程式設計能力的本體論, 並針對學習者所提出的問題找出值得信賴的專家。

雖然過去研究提出了許多分析使用者及個人化的方法, 但在過去的推薦系統中多半都偏重顯性知識的推薦, 例如: 文件、網頁、文獻等, 但是單單提供顯性知識未能有效地解決使用者的問題, 而在過去的專家推薦的文獻中[15,16,18,22,23], 仍然有以下缺點需要改善:

1. 偏重顯性或者隱性的知識推薦, 鮮少研究同時提供顯性及隱性知識的同時推薦, 然而, 使用者可能需要實際專家的介入才能解決問題。
2. 在針對問答行知識平台的專家推薦文獻中, 僅把問題考慮在單一領域, 然而實際的發問很可能同時包含多個領域, 或者單一領域但不同類型的問題, 如圖 1 中的問題應該同時屬於法律相關問題與拍賣相關問題, 但在普遍的問答系統中, 都會被歸類在單一分類。
3. 在專家推薦的方法中, 都以使用者所發布的文字資訊作為使用者輪廓(User Profile), 並未考慮使用者學習的效應, 如圖 1 中所示的問題範例中使用者選擇了自己滿意的答案作為最佳解答, 表示發問者的問題獲得了解答, 因此當以後有同樣的問題發生時, 該發問者應該也可以提供相關的回應, 表示使用者在問答的過程中, 知識是有所累積的, 然而如此的學習效果在過去的文獻未被提起, 因此會造成最終被推薦的專家都是發文頻率較高的專家, 反而造成這些專家的負擔, 此外, 在分析使用者的問題時, 也並未加入個人化資訊的考量, 導致在內隱知識的推薦上, 並沒有辦法

針對使用者真正需要的資訊給予專家的推薦。

4. 每位專家都有可能有不只一項的專長, 若單純以單一的關鍵詞向量表示每位專家, 那麼該專家的部分專長可能會因此被忽略。

有鑑於上述的因素, 本研究針對問答型的知識平台提出同時提供解答與專家的個人化推薦服務, 以便於使用者在系統的解答推薦上並未解決其問題時, 仍然可以透過詢問相關專業人士以解決其問題, 而透過分析知識平台中每位發問者與回答者的文字資料, 利用資料分類、投票模型與向量擴張技術提供發問者可能的解答與可回答的專家, 並主動提供專家可能有能力解決的問題, 以降低專家需要額外搜尋可回答的問題的負擔。



圖 1 多分類的問題範例

2. 參考文獻

2.1 知識分享

「知識分享」為社群中主要的連結點, 社群成員以問答為主軸, 在虛擬知識社群中交換並分享知識[9,11], Leenders et al. [12]曾提及虛擬社群成員為分享彼此的想法, 利用網路平台集成一個虛擬團隊而創造新知識; 虛擬社群之交流多以文字傳達並儲存於虛擬社群之分享空間, 以利其他社群之成員觀看或討論, 進而再交流與創造知識。

Armstrong 及 Hagel[2]認為隨著資訊科技的進步與發展，網路上的虛擬社群已成為主要知識分享的平台。Wasko & Faraj[24]也認為虛擬社群中的成員主要是因為對於某些知識有所需求，而產生知識分享，也有學者認為知識是一種流量(flow)的概念，對知識擁有者來說，這是一種選擇性「推」的過程；而對知識接受者則是「拉」的過程，在推與拉的過程中，知識分享的行為於焉產生[20]。

而在知識的定義上，許多研究都有不同的表達方式及定義。Zack[27]認為知識是將眾人的經驗或推理作有系統的累積，使之成為有價值的內容，而 Bock, Zmud, Kim, & Lee[3]則將知識定義為解決問題的必要資訊；而關於知識的類型，可分為內隱知識(tacit)及外顯知識(explicit)兩種；內隱知識指的是個人的、難以形式化與溝通的知識，相對的，外顯知識指的是可形式化、可用言語傳達的知識。以知識的分類方式來看，虛擬知識社群所傳達的知識內容應是以外顯知識為主。

2.2 資料的分類與分群

資料的分類與分群已應用在許多的領域，如：醫學、產品[6]、電子郵件[1]、文件[8]等，其資料探勘的方法也有很多種，Wu et al.[25]列出了十個最具影響力的演算法，分別為：C4.5、K-Means、SVM、Apriori、EM、PageRank、AdaBoost、kNN、Naive Bayes、及 CART，其中 SVM 也被認為是最穩健、最正確的分類演算法。

SVM 是一種基於統計的理論所建立的分類方法，由 Vapnik 學者於 1995 年提出，起初是針對二元分類所設計的一種分類法，而現在在很多領域中已經延伸至多分類的應用[19]，SVM 多分類主要分為兩種，第一種是 One-Versus-One(OVO) SVM，針對任兩個類別的資料分別訓練，訓練類別 i 與類別 j 可視為最佳化問題如式 1：

$$\begin{aligned} \text{Minimize } & \frac{1}{2} \sum_{m=1}^N \sum_{n=1}^N y_m y_n K(x_m, x_n) \alpha_m^{ij} \alpha_n^{ij} - \sum_{m=1}^N \alpha_m^{ij} \\ \text{subject to } & \sum_{m=1}^N y_m \alpha_m^{ij} = 0, 0 \leq \alpha_m^{ij} \leq C \quad \text{式 1} \end{aligned}$$

另一種是 One-Versus-Rest(OVR) SVM，其只建立 k 個 SVM 分類器，針對第 i 類的分類器的訓練資料選取上將第 i 類的樣本做為正樣本，其餘的樣本皆為負樣本，進一步求解式 2：

$$\begin{aligned} \text{Minimize } & \frac{1}{2} \sum_{m=1}^N \sum_{n=1}^N y_m y_n K(x_m, x_n) \alpha_m^i \alpha_n^i - \sum_{m=1}^N \alpha_m^i \\ \text{subject to } & \sum_{m=1}^N y_m \alpha_m^i = 0, 0 \leq \alpha_m^i \leq C \quad \text{式 2} \end{aligned}$$

在過去研究文件分群的應用上，Vertommen, Janssens, De Moor, & Duflou[23]利用 Agglomerative Hierarchical Clustering(AHC)判斷每位學者不同的專長；Lai & Liu[10]則將 AHC 做知識文件的分群；Tseng & Weng[22]利用分群方法將知識平台中類似議題的問題做分群，且認為問題所討論的議題數量不定，因此適合利用分群方法將討論類似議題的問題做分群。

2.3 推薦系統

推薦系統的主要目的在於解決資訊過載(Information overload)的問題[17]，利用使用者過去的資料進而推估未來可能的興趣與喜好並過濾掉不必要的資訊，因此，在過去推薦系統的研究中，往往會加入個人化的概念進一步改善回傳結果的準確性，而在資訊過濾的方法上大致可分為三類[13]，分別為規則式的過濾技術(Rule-based Filtering)、以內容為基礎的過濾方法(Content-based Filtering)及協同過濾技術(Collaborative Filtering)，其中使用最廣泛的為協同過濾技術。

自電子商務興起之後，推薦系統的需求日漸增加，應用的層面也相當個廣泛，例如：書籍、新聞、音樂、網頁、產品等，甚至社群網站中的虛擬商品也漸漸提供了個人化推薦的服務[17]，此外，在知識領域方面，Liu et al. [15]認為每位知識工作者(Knowledge workers)的知識都存在於他們定義的資料夾中(Folder)，透過他們所擁有的文件及對文件的評分，找出相似的使用者並從中找出使用者可能有興趣的文件進行推薦，達到知識分享；Lai 與 Liu 學者[16]

將知識流探勘(Knowledge flow mining)的概念、協同過濾技術與分群技術進行整合，並加入時間因素探討使用者不同的時期有著不同的知識需求，因此針對不同的使用者的知識需求推薦所需的文件；Liao et al. [14]利用圖書館讀者的借閱紀錄建立每個讀者的知識本體，並推薦其他的書籍給使用者；Tseng & Weng[22]利用互動式的問答系統 ProgrammerClub 的問答紀錄資料找出值得信賴的專家給學習者，以網站的架構以及問題的類型建立問題的本體，同時考量專家的領域 (Domain)、信賴度 (Trustworthy) 及上線狀況 (Availability) 等因素去對專家做評分；Macdonald & Ounis[18]利用 TREC 2005 & 2006 Enterprise Track 的資料，從使用者相關的文件中發掘每一位使用者的知識，針對是使用者所下的 Query 對文件做評分，再以投票的方法將文件的評分與使用者輪廓 (User Profile) 整合成對每一位專家的評分並推薦給查詢者，該學者也提出在專家推薦的系統上，Precision 是個重要的衡量指標。

3. 研究方法

本研究主要在於探討解答與專家的個人化推薦，使用者的知識與專業大多編碼成文字資料[16]，例如：論文、報告、書籍等任何由使用者所發表的文字資訊皆可視為使用者本身內隱知識外顯化的一種轉換，因此，本研究提出研究架構如圖 2，將從現有知識平台的問答內容中發掘每位使用者的專長及有興趣的領域，在解答的推薦上，從現有平台中推薦可能的解答給發問者；在專家上推薦可能有能力解決發問者問題的其他使用者，並主動式邀請專家協助問題的回答，縮短發問者等待的時間，加速知識的流動。

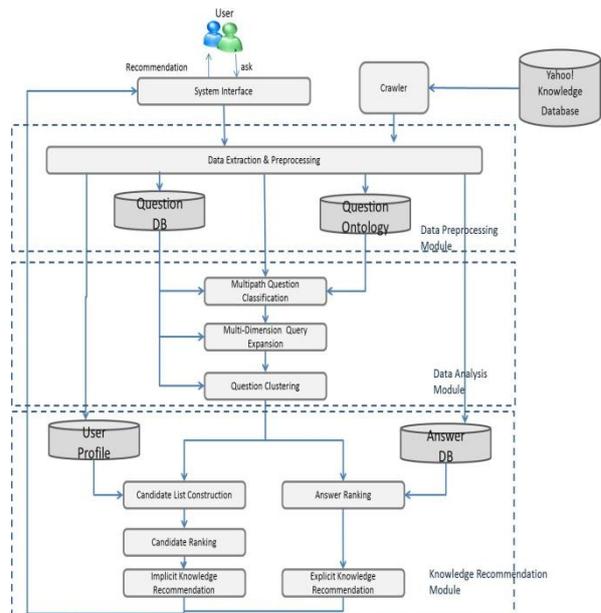


圖 2 研究架構

3.1 資料蒐集與前處理

在知識平台的選取上，本研究以 Yahoo! 奇摩知識+為例，該平台為類型較廣泛的知識平台，亦即發問者所發布的問題範圍較廣，且是一個中文為主的知識平台，因此，本研究在資料蒐集與前處理階段蒐集 Yahoo! 奇摩知識+中所需資訊並儲存於資料庫中，針對中文的自然語言以中研院 CKIP 進行文字的切割、詞性標定 (POS)、冗字去除、向量轉換的處理以用於後續的多分類器進行判斷。

一般互動式問答的知識平台已解決的問題樣本有以下資訊：發問者、發問類型、發問標題、問題內容、最佳解答、最佳解答回答者、其他回答、其他回答者，本研究假設當一位發問者的問題獲得了最佳解答後，發問者本身會學習到該類题目的相關知識，並且具有累積效應，因此選用已有最佳解答的問題集做為本研究的主要資料集 S ，此外，本研究研究目的之一在於發掘過去類似的發問紀錄以避免重複發問的情形，因此將針對發問的標題及問題內容視為發問者目前的問題 Q 並進行資料前處理，資料前處理的目的在於過濾掉文字內容中不必要的資訊，並將擷取出的關鍵字詞視為該問題的主要特徵向量 \vec{Q} ，並採用

TF 作為文字權重計算方法，其原因在於在使用者的發問中，去除冗字之後，比較常出現的字詞應該比較重要，並不會因為其他使用者也使用者樣的字詞發問，該字詞就失去重要性，而若採用 TF-IDF 計算，就會有以上問題，因此使用 TF 作為權重的計算方法。

3.2 問題分類

本研究在問題的類型上採用 Yahoo!奇摩知識+所制定的類型，該平台中問題的類型屬於階層式(Hierarchical)的類型分布如圖 3 所示，在每個問題的分類上都經過使用者自行選取，然而，往往一個問題不一定屬於單一領域，類似這樣有可能跨領域的問題若是以過去單一分類法處理，勢必會遺失掉部分的資訊，那麼一來便會導致該問題無法得到全部的答案，Diez et al. [6]針對架構型分類提出 H-SVM 作為分類器並且考慮階層間的相互關係，也驗證該方法的效果在樹狀架構均勻的情況下是優於 SVM 的，也改良了 OVR SVM 在選取訓練資料上選取過多負樣本的缺點，但是該方法中所使用的 Top-Down 一致性檢驗機制在不同類別包含相同文件集的情況下，例如：“電腦認證”與“電腦”同樣連結到“問題集 3”，會造成分類上判斷的錯誤，因此，本研究在分類上將參考 Diez et al.[6]的 H-SVM 樣本選取的概念並提出新的檢驗機制 Bottom-Up Multipath Evaluation (BUME)改善這個問題，本節將說明分類器的訓練樣本的選取及檢驗機制。

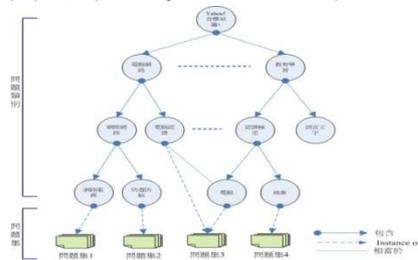


圖 3 Yahoo!奇摩知識+的問題分類架構圖

3.2.1 H-SVM 分類器資料選取與模型訓練

為了進行問題的多分類判斷

(Multipath)，參考了 Diez et al.(2010)所提出之多分類模型，在既定的分類架構上每一個類別節點(Node)都是一個分類器(Classifier)，令訓練集(Training set) $S = \{(\vec{q}_1, \vec{c}_1), \dots, (\vec{q}_n, \vec{c}_n)\}$ ，參數說明如下：

\vec{q}_i = 第 i 個問題輸入向量
 \vec{c}_i = 第 i 個樣本分類結果輸出向量， $\vec{c}_i \in \{-1, +1\}^k$
 n = 問題總數
 k = 總類別數

而在分類器學習時正負樣本的選取會從訓練集S中進一步挑選而非全部丟入訓練每一個分類器，在第j個分類器上，正樣本選取S中屬於類別j的樣本，即 $S_j^+ = \{(\vec{q}_i, \vec{c}_i) : c_{i,j} = +1\}$ ，負樣本則選取屬於類別j的父節點(par(j))但是不屬於類別j的樣本作為訓練的負樣本，即 $S_j^- = \{(\vec{q}_i, \vec{c}_i) : c_{i,par(j)} = +1 \wedge c_{i,j} = -1\}$ ，取得每個節點正負訓練樣本後，則以 Weka 中的 LibSVM 進一步訓練本研究之架構型的多分類模型。

3.2.2 Bottom-Up Multipath Evaluation (BUME)

當取得架構型的多分類模型後，將問題向量 \vec{Q} 利用該模型可在每一個節點分類器取得分類結果輸出向量 \vec{c}_Q ，為了確定分類的結果能夠符合樹狀結構，避免發生父類別與子類別分類結果不一致的情況，即 $\vec{c}_{Q,par(j)} = -1$ 但 $\vec{c}_{Q,j} = +1$ ，過去針對階層式分類(Hierarchical Classification)的檢驗最常使用的是 Top-Down 的檢驗機制，然而針對本研究有可能出現子節點擁有共同問題集的狀況 Top-Down 的檢驗機制會導致分類上的錯誤，因此，本研究針對這種狀況提出新的檢驗機制 Bottom-Up Multipath Evaluation(BUME)，演算法如圖 4 所示。

```

Input : Possible Class Result ( $\overline{Q, c_{Q_{possible\ leaf}}}$ )
Output : Final Multi-Class Result ( $\overline{Q, \overline{c_Q}}$ )
Step 1: For Each Possible Leaf Class in  $\overline{c_{Q_{possible\ leaf}}}$ 
Step 2: Insert  $c_{Q_{possible\ leaf, i}}$  Into NextClassQueue
Step 3: For j = 0 to NextClassQueue.Size()
Step 4: UpperResult = NextClassQueue.UpperClassification
Step 5: IF UpperResult IS TRUE THEN
Step 6: NextClassQueue <- UpperClass
Step 7: End IF
Step 8: Next
Step 9: IF NextClassQueue.LastElement IS Root THEN
Step 10: Insert  $c_{Q_{possible\ leaf, i}}$  into Final Multi-Class Result ( $\overline{Q, \overline{c_Q}}$ )
Step 11: End IF
Step 12: NextClassQueue = NULL
Step 13: Next
Step 14: Return Final Multi-Class Result ( $\overline{Q, \overline{c_Q}}$ )

```

圖 4 BUME 演算法

在 BUME 中， $\overline{c_{Q_{possible\ leaf}}}$ 表示每個樣本所有可能的子類別，Final Multi-Class Result ($\overline{Q, \overline{c_Q}}$) 表示問題 Q 所判斷的所有分類中符合階層架構分類準則的類別，也就是從葉節點(leaf node)到根節點(Root Node)的分類結果皆為 TRUE，而每一個節點間都存在著連結關係，如圖 3 所示，當一個樣本丟入樹狀分類會得到一組子節點之分類結果，假設樣本的分類結果為”電腦”，從圖 3 可得到相關分類得到相關的 In-Out Link Table 如表 1，先針對所分類到的葉節點檢查其 In-Link 是否有其中一個 Node 的分類為 TRUE，以圖 3 為例，即為檢查”認證檢定”、”電腦網路”之分類結果，如果任一個 In-Link 的節點中之分類結果為 TRUE，則將其加入節點檢查序列 NextClass，在最後得到的 NextClass 中，如果最後一個被加入的節點為一根節點，則表示此子節點必定存在一條分類結果全部為 TRUE 的路徑，因此將其加入最終多分類結果 Final Multi-Class Result ($\overline{Q, \overline{c_Q}}$)，透過 BUME 演算法後，即可得到問題目前的多分類判斷 $\overline{c_Q}$ 。

表 1 In-Out Link Table 範例

Node	In-Link	Out-Link
Root	Null	電腦網路、教育學習
電腦網路	Root	網際網路、電腦認證
教育學習	Root	認證檢定、語言文字

認證檢定	教育學習	電腦、商業
電腦	認證檢定	問題集三
電腦認證	電腦網路	問題集三

3.3 問題向量擴張

當問題得到分類結果之後，問題 Q 可能被分類至不同類別，以錯誤!找不到參照來源。的問題為例，該問題可能被分類至”生活法律”與”稅務”兩種類別，即 $c_{Q,生活法律}=+1$ 與 $c_{Q,稅務}=+1$ ，因此在問題向量擴張的階段，要進一步針對每個可能類別建立其問題向量得到 $\overline{Q_{生活法律}}$ 與 $\overline{Q_{稅務}}$ ，並結合使用者過去類似的發問紀錄，進一步精確的描述問題，流程如圖 5。



圖 5 問題向量擴張流程圖

首先，先利用發問者的原始問題的關鍵字透過 Yahoo API 在判定的分類底下進行查詢，並各自取前五個回傳結果，再利用使用者過去的發問紀錄，分別作為向量擴張的文件集合，並分別利用 TF 計算建立各自分類向量，擴張後的向量建立流程如圖 6 所示，以得到問題 Q 在每個類別的問題向量。



圖 6 問題向量擴張範例圖

3.4 問題分群

在建立完問題向量擴張後，在同一分類底下的問題仍然可能包含許多的議題，例如在”C++”的分類底下，有些問題可能是問物件導向的概念，有些問題可能是問迴圈的

概念，問題的分群主要目的在於將同一分類下類似議題的問題分成一群，然而每個分類並沒有明確定義實際共有多少個議題 [22]，因此本研究將利用 Single-Link 的 Agglomerative Hierarchical Clustering(AHC) 配合門檻值的設定，自動將過去發問紀錄中，與目前問題討論類似議題的歷史紀錄找出來，再從中建立候選答案清單與候選專家，而透過 AHC 分群後可得到的類似的問題及其相似度，其中問題之相似度的計算採用 Cosine 相似度計算，如式 3 所示：

$$sim(Q, q_i) = \frac{\bar{Q} \cdot \bar{q}_i}{\|\bar{Q}\| \|\bar{q}_i\|} \quad \text{式 3}$$

其中 $sim(Q, q_i)$ 表示目前所提出的問題 Q 與已存在於資料集 S 中的問題 q_i 的相似度， \bar{Q} 及 \bar{q}_i 為問題 Q 與 q_i 之文字向量，將從這些資訊進一步得到以下四個資訊並以 SimilarQuestionData 表示四種資訊的集合：

- (1) 類似的問題及相似度 $sim(Q, q_i)$ ：記錄所有類似的問題與原問題的相似度。
- (2) 類似問題的最佳解答(Candidate Answer, CA)：記錄所有在(1)中的最佳解答。
- (3) 類似問題的發問者(Learning Profile, LP)：記錄所有在(1)中的發問者，用於發現經過教育的潛在知識提供者。
- (4) 最佳解答的回答者(Best Answer Profile, BAP)：記錄所有在(2)中的回答者。

3.5 解答之評分與推薦

本研究在解答的推薦上，針對目前發問者的問題 Q 找經過分群的類似問題 q 中找出目前問題 Q 相似的問題，並將類似問題 q 中的最佳解答視為解決問題 Q 的答案，並考慮問題的可信任程度後推薦給發問者，本研究假設知識提供者在該領域獲得最佳解答次數越高，則該知識提供者所提供的答案可信任的程度也越高，對於候選答案的評分上，利用 AHC 對於問題的分群時所計算的相似度，然而，當兩個問題相似程

度過低但是信任程度過高或者相似程度高但是信任程度過低的情況下皆有可能將錯誤的答案給予過高的評分，因此將該相似度 $sim(Q, q_i)$ 作為候選答案的評分並與信任分數 $TrustValue(CA_i)$ 加以結合，如下式：

$$Score(CA_i) = \frac{2}{\frac{1}{sim(Q, q_i)} + \frac{1}{TrustValue(CA_i)}} \quad \text{式 4}$$

CA_i 是類似問題 q_i 的最佳解答， $TrustValue(CA_i)$ 是該候選解答回答者在該領域下得到最佳解答的總次數，該式子所計算出來之值在兩者不平衡的情形下，較小值對於影響最後結果的影響會較大，以避免將錯誤的資訊給予過高的評分，最後依據 $Score(CA_i)$ 排序並推薦前 K 個可能的答案給使用者。

3.6 專家之評分與推薦

在專家的評分上利用投票(Vote)的概念進一步將原本候選答案的評分整合成專家的評分，每一位候選專家 u_i 從自己回答過的最佳解答的問題所獲得的分數為 $Score_{BAP}(u_i)$ ，本研究稱之為最佳解答輪廓(BAP)的評分，從自己詢問過的且有獲得最佳解答的問題所獲得的分數為 $Score_{LP}(u_i)$ ，本研究稱之為學習輪廓(LP)的評分，透過 α 、 β 依偏好進行調整以得到候選專家最終的評分，評分方式如下列式子 5 所示：

$$Score(u_i) = \alpha \times Score_{BAP}(u_i) + \beta \times Score_{LP}(u_i), \\ i = 1 \text{ to } n \wedge \alpha + \beta = 1 \wedge 0 < \alpha, \beta < 1 \quad \text{式 5}$$

$$Score_{BAP}(u_i) = MAX(Score(CA_i): CA_i \in CA(Q) \cap BAP(u_i)) \quad \text{式 6}$$

$$Score_{LP}(u_i) = \sum_{CA_i \in CA(Q) \cap LP(u_i)} Score(CA_i) \quad \text{式 7}$$

式子說明：

u_i = 第 i 個專家;

CA_i = 與第 i 個專家相關的候選解答

$CH(Q)$ = 問題 Q 所找出的候選答案;

$Score(CA_i)$ = 候選答案的評分

$Score_{BAP}(u_i)$ = 第 i 個專家最佳解答輪廓的評分;

$Score_{LP}(u_i)$ = 第 i 個專家解答學習輪廓的評分;

$BAP(u_i)$ = 第 i 個專家所回答過最佳解答的問題集合;

$LP(u_i)$ = 第 i 個專家過去發問且有得到最佳解答的問題集合;

而本研究在最佳解答輪廓的評分上之所以取最大值是因為希望從每位專家所發表的最佳答案中，取出最具代表性的作為專家在最佳解答輪廓的評分，而在學習效應方面，則是基於本研究假設，當專家過去所得到的最佳解答越多，所累積的相關知識應該也越豐富，因此使用總和的方式加總過去學到的知識，然而為了避免過去學習的知識影響大於專家自己發表的最佳解答，使用參數控制兩者間的重要性並整合專家的隱性知識與學習到的知識，最後針對不同領域的專家分別推薦五位給發問者作為詢問或討論的參考。

3.7 小結

本研究結合多分類於解答與專家的搜尋，利用每一位使用者所發布的最佳解答作為候選解答的主要來源，並同時考慮每一位發問者在發問之後選取了最佳解答的學習效應，以避免過多的問題都推薦在特定的專家造成專家的負擔，也結合了分析使用者過去紀錄的諮詢以提供發問者個人化的解答以及專家推薦，並主動邀請相關的專家針對使用者提出的問題進行協助，期望增加專家們協助回答問題的動機，提供每位發問者準確的回答，使其能夠迅速地得到解答。

4. 實驗驗證

本研究之開發環境為 Linux 10.04、MySQL 5.0.67、PHP 5.3.5、JAVA、Weka、LibSVM 及 CKIP，並採用 Yahoo! 知識+ 2009/11 至 2010/10 之問答紀錄作為主要資料及，所包含問題之類別共有 496 個類別，所蒐集之文件總數為 2,160,729 筆，所蒐集

之有效文件總數約為 1,925,042 筆。

實驗方面採用 2009/11 內資料作為系統模型之訓練樣本以建置本研究之多主題個人化解答與專家推薦系統，並與 Yahoo! 知識+所提供之”解答搜尋系統”以及”分類助人王”進行 MRR 之比較。

實驗一：

此實驗進行本研究權重參數測試，以 MRR 比較不同的參數設定 ($W_{current}$ 、 W_{past} 、 W_{sim} 、 $InvolveTrustValue$) 對於推薦結果的影響，並比較在問題推問階段考量解答可信度與否對於推薦成效的影響，最後，與 Yahoo! 奇摩知識+之搜尋系統分別進行 Top 5、Top 10 比較。

實驗方法為在推薦階段參數的設定區分為以下 6 種情境：

1. QueryOnly with Answer Trust Value ($W_{current}=1, W_{past}=0, W_{sim}=0, InvolveTrustValue=1$): 考量發問者的問題關鍵字資訊以及相關候選解答的可信度進行本研究之推薦模型。
2. QueryOnly without Answer Trust Value ($W_{current}=1, W_{past}=0, W_{sim}=0, InvolveTrustValue=0$): 只考量發問者的問題關鍵字資訊進行本研究之推薦模型。
3. Query+CB+CF with Answer Trust Value ($W_{current}=0.5, W_{past}=0.4, W_{sim}=0.1, InvolveTrustValue=1$): 此情境為本研究之主要方法，綜合發問者的問題關鍵字、發問者過去的類似發問紀錄及其他使用者類似發問之關鍵字詞等 3 種因素並考量相關候選解答的可信度進行本研究之推薦模型。
4. Query+CB+CF without Answer Trust Value ($W_{current}=0.5, W_{past}=0.4, W_{sim}=0.1, InvolveTrustValue=0$): 綜合目前發問者的問題關鍵字、發問者過去的類似發問紀錄及其他使用者類似發問之關鍵字詞等 3 種因素進行本研究之推薦模型。
5. Query+CF with Answer Trust Value ($W_{current}=0.5, W_{past}=0, W_{sim}=0.5, InvolveTrustValue=1$): 綜合目前發問者的問題關鍵字及其他使用者類似發問之關鍵字詞等 2 種因素並考量相關

候選解答的可信度進行本研究之推薦模型。

6. Query+CF without Answer Trust Value($W_{current}=0.5, W_{past}=0, W_{sim}=0.5, InvolveTrustValue=0$): 綜合目前發問者的問題關鍵字及其他使用者類似發問之關鍵字詞等 2 種因素進行本研究之推薦模型。

針對每種情境以及 Yahoo! 奇摩知識+之搜尋系統在 Top-5 及 Top-10 的推薦情況下，分別針對實驗一中的測試資料進行解答的推薦，在推薦 Top-5 及 Top-10 的情況下，在圖 7 中可發現本研究的六種情境下的 MRR 值皆優於 Yahoo! 奇摩知識+之搜尋系統，進一步針對每種情境進行 MRR 之成對 T 檢定如

表 2 所示，可以發現無論是 Top-5 或者 Top-10，本研究所提出之六種情境的推薦結果皆顯著優於 Yahoo! 奇摩知識+之搜尋系統，且又以情境三的推薦結果最佳，當使用者提出問題後考量過去的歷史資訊以及相關類別的類似字詞可以有效的提升解答推薦的準確性，並且能將正確的答案以較高的排序進行推薦，再進一步討論考量答案可信度與否對於 MRR 的影響而進行成對 T 檢定，從表 3 可以發現在 Top-5 略無顯著差異，而在 Top-10 的情況下，考量了答案可信度的 MRR 值皆顯著優於為未考量答案可信度的 MRR 值，也就是在推薦解答模型中，考量每個解答的可信任程度對於最後的推薦正確性有顯著的改善。

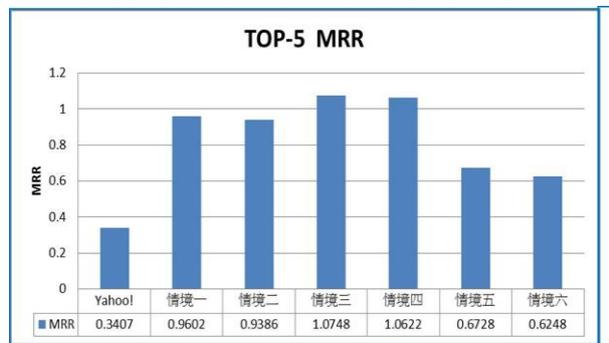


圖 7 六種情境與 Yahoo! 知識+在之 Top-5 及 Top-10 平均 MRR 比較圖

表 2 六種情景與 Yahoo! 之成對 T 檢定之比較表

Top-5 六種情境與 Yahoo! 之 MRR 成對 T 檢定 (信心水準=95%)						
	情境一	情境二	情境三	情境四	情境五	情境六
Yahoo!	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0021*
情境一		0.1395	0.0483*	0.0584	0.0037*	0.0013*
情境二			0.0320*	0.0343*	0.0080*	0.0024*
情境三				0.3591	0.0000*	0.0000*
情境四					0.0001*	0.0000*
情境五						0.1355

Top-10 六種情境與 Yahoo! 之 MRR 成對 T 檢定 (信心水準=95%)						
	情境一	情境二	情境三	情境四	情境五	情境六
Yahoo!	0.0000*	0.0000*	0.0000*	0.0000*	0.0021*	0.0046*
情境一		0.0810	0.0383*	0.0856	0.0037*	0.0003*
情境二			0.0188*	0.0372*	0.0020*	0.0005*
情境三				0.1208	0.0000*	0.0000*
情境四					0.0000*	0.0000*
情境五						0.1149

*表示顯著差異

表 3 在 Top-5 & Top-10 對於考量答案可信度與否對於 MRR 影響之檢定

考量答案可信度考量檢定之比較 (信心水準=95%)						
	Top-5	Mean	P-value	Top-10	Mean	P-value
考量答案可信度		0.9026	0.0789	考量答案可信度	1.0640	0.0176*
不考量答案可信度		0.8752		不考量答案可信度	1.0271	

最後，從實驗一中的比較可發現採用結合使用者 Query、歷史發問紀錄、類似發問紀錄及答案的可信度能夠有效改善解答的推薦結果，因此將採用情境三之參數組合進行實驗二。

實驗二：

在實驗二中，將採用實驗一情境三的參數組合 ($W_{current}=0.5, W_{past}=0.4, W_{sim}=0.1, InvolveTrustValue=1$) 搭配 α 及 β 兩個參數設定以下四種情境：

1. Focus on BAP ($\alpha=1$ 及 $\beta=0$): 在專家推薦的分數計算上，以每位專家過去回答的最佳解答紀錄所獲得的分數做為主要的評分依據。
2. Focus on Both ($\alpha=0.5$ 及 $\beta=0.5$): 在專家推薦的分數計算上，同時考量每位專家過去回答的最佳紀錄以及發問紀錄所獲得的分數做為主要的評分依據，此情境為本研究所提出之情境。
3. Focus on LP ($\alpha=0$ 及 $\beta=1$): 在專家推薦的分數計算上，以每位專家過去發問的紀錄經過學習效應後所獲得的分數做為主要的評分依據。

4. Yahoo!分類知識王：以每個類別下回答最佳解答次數最多的前五位以及前十位使用者做為推薦依據。

藉由以上四種情境比較考量使用者的學習效應是否會影響專家推薦的準確性，並且比較前三種情境下與 Yahoo! 奇摩知識+的分類助人王中所有專家推薦分布的情形，探討透過本研究所提出的模型是否在能夠不影響推薦效能的情況下平衡每一位專家所需回答的問題數量。

本實驗驗證的方法為比較四種情境在總題數 1000 題中每位專家平均被推薦的次數，其中情境二為本研究所提出的模型，從表 4 可以發現本研究所提出的情境二每位專家備平均回答題數為 3.52，與其他三種情境比較可以發現在本研究所提出的情境下，每一位專家被邀請回答問題的次數最低，接著針對前三種情境進一步比較 MRR 值是否有顯著的差異，討論是否在平均了每位專家的回答量後不能維持同樣的推薦效能，從

表 5 中可以發現情境一與情境二的 MRR 值並沒有明顯的差別，從表 6 中的檢定中也可以發現情境一與情境二並沒有顯著差異，而情境二與情境三間則有顯著性的差異，因此驗證了本研究所提出的使用者學習效應的考量能在不降低專家推薦的正確率下平均每一位專家的負擔。

表 4 三種專家推薦情境與 Yahoo!分類知識王的專家平均回答題數以及單一專家被推薦的最大值比較表



表 5 情境一至情境三 Top-5 與 Top-10 之 MRR 比較表

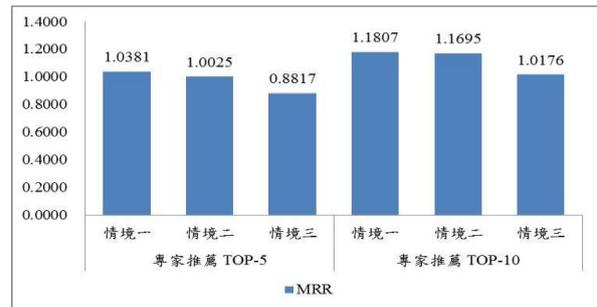


表 6 Top-5 與 Top-10 專家推薦情境檢定比較表

三種專家推薦情境之推薦效能檢定表					
專家推薦 TOP-5			專家推薦 TOP-10		
P-value	情境二	情境三	P-value	情境二	情境三
情境一	0.5245	0.0334*	情境一	0.4311	0.0489*
情境二		0.0002*	情境二		0.0001*

5. 結論

本研究主要在於建置一個不同於以往推薦流程的個人化的解答與專家推薦系統，同時提供發問者可能已存在的解答及相關領域的專家，並在所提供的解答無法滿足發問者的情況下，進一步提供相關的專家以供諮詢，在專家方面，則將真正與該位專家領域相關的問題主要邀請協助回答，節省專家主動搜尋的時間也避免其回覆過多重複性的發問後降低再次回覆的意願。

本研究提出 BUMe 的多分類檢驗機制有效改善過去研究在文件分類的缺失，利用多分類及類似主題分群結合個人化技術並考量解答可信程度與專家學習效應的方法進一步精準地分析發問者所需要的完整資訊，改善現有知識平台關鍵字查詢的缺點，提升解答推薦的效能並在維持同樣的專家推薦效能上降低每位專家的負擔。

在系統效能的驗證方面，本研究蒐集 Yahoo! 知識+的一年內(2009 / 11 / 01 ~ 2010 / 10 / 31)的問答紀錄中所有的問題、最佳解答及相關的發問者與回答者進行系統驗證與比較，在實驗一，比較對象分別為 Yahoo!知識+的關鍵字搜尋以及本研究所提出之六種情境，其中情境三為本研究所提出之模型，從實驗數據中可發現本研究所提出之六種情境在解答上的推薦效能皆顯著優於 Yahoo!知識+的關鍵字搜尋，在六種情境中又以情境三的推薦效能最佳，而在考量答案可信度因子的比較分析中也可發現，在 Top-10 的推薦結果中，考量答案可信度的推薦效能顯著優於未考量可信度的推薦效能，也證明本研究所提出的結合答案可信度的因子能夠取得較佳的推薦結果；在實驗二中，則提出四種比較情境，其中情境二為本研究所提出之模型，實驗數據中也發現本研究在考量專家本身知識與學習效應的情況下，能夠維持同樣的推薦水準並降低每一位專家回答上的負擔。

最後，本研究在考量了問題多分類、類似主題分群、個人化技術、解答可信程度與專家學習效應

的因子後，有效提升解答推薦的效能並降低每一位專家回答上的負擔，並針對發問者與專家分別設計主動地告知流程，達到即時解決發問者問題與並提供諮詢對象，並主動要求專家進入協助，降低每位專家回答過多重複問題的負擔。

參考文獻

- [1] Appavu, S., Rajaram, R., Muthupandian, M., Athiappan, G., & Kashmeera, K. S. Data mining based intelligent analysis of threatening e-mail, “*Knowledge-Based Systems*”, Vol. 22, 2009, pp. 392-393.
- [2] Armstrong, A. & Hagel, J. “The real value of on-line communities,” *Harvard Business Review*, Vol. 10, 1996, pp. 134-&.
- [3] Blanco-Fernandez, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrera, M., Lopez-Nores, M., Garcia-Duque, J., Fernandez-Vilas, A., Diaz-Redondo, R. P., Bermejo-Munoz, J. “A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems,” *Knowledge-Based Systems*, Vol. 21, 2008, pp. 305-320.
- [4] Bock, G. W., Zmud, R. W., Kim, Y. G., & Lee, J. N. “Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate,” *MIS Quarterly*, Vol. 29, 2005, pp. 87-111.
- [5] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. “Modeling wine preferences by data mining from physicochemical properties,” *Decision Support Systems*, Vol. 47, 2009, pp. 547-553.
- [6] Diez, J., del Coz, J. J., & Bahamonde, A. “A semi-dependent decomposition approach to learn hierarchical classifiers,” *Pattern Recognition*, Vol. 43, 2010, pp. 3795-3804.
- [7] Kim, J. W., Lee, K. M., Shaw, M. J., Chang, H. L., Nelson, M., & Easley, R. F. “A preference scoring technique for personalized advertisements on Internet storefronts,” *Mathematical and Computer Modeling*, Vol. 44, 2006, pp. 3-15.
- [8] Kim, K. J. & Cho, S. B. “Personalized mining of web documents using link structures and fuzzy concept networks,” *Applied Soft Computing*, Vol. 7, 2007, pp. 398-410.
- [9] Komito, L. “The net as a foraging society: Flexible communities,” *Information Society*, Vol. 14, 1998, pp. 97-106.
- [10] Lai, C. H. & Liu, D. R. “Integrating knowledge flow mining and collaborative filtering to support document recommendation,” *The Journal of Systems and Software*, Vol. 82, 2009, pp. 2023-2037.
- [11] Lee, F. S. L., Vogel, D., & Limayem, M. “Virtual community informatics: A review and research agenda,” *The Journal of Information Technology Theory and Applications (JITTA)*, Vol. 5, 2003, pp. 47-61.
- [12] Leenders, R., van Engelen, J. M. L., & Kratzer, J. “Virtuality, communication, and new product team creativity: a social network perspective,” *Journal of Engineering and Technology Management*, Vol. 20, 2003, pp. 69-92.
- [13] Liang, T. P., Yang, Y. F., Chen, D. N., & Ku, Y. C. “A semantic-expansion approach to personalized knowledge recommendation,” *Decision Support Systems*, Vol. 45, 2008, pp. 401-412.
- [14] Liao, S. C., Kao, K. F., Liao, I. E., Chen, H. L., & Huang, S. O. “PORE: a personal ontology recommender system for digital libraries,” *The Electronic Library*, Vol. 27, 2009, pp. 496-508.
- [15] Liu, D. R., Lai, C. H., & Huang, C. W. “Document recommendation for knowledge sharing in personal folder environments,” *The Journal of Systems and Software*, Vol. 81, 2008, pp. 1377-1388.
- [16] Liu, D. R., Lai, C. H., & Lee, W. J. “A hybrid of sequential rules and collaborative filtering for product recommendation,” *Information Sciences*, Vol. 179, 2009, pp. 3505-3519.
- [17] Liu, F. & Lee, H. J. “Use of social network information to enhance collaborative filtering performance,” *Expert Systems with Applications*, Vol. 37, 2010, pp. 4772-4778.
- [18] Macdonald, C. & Ounis, I. “Searching for expertise: Experiments with the Voting Model,” *The Computer Journal*, Vol. 52, 2010, pp. 729-748.
- [19] Malathi, V., Marimuthu, N. S., & Baskar, S. “A comprehensive evaluation of multicategory classification method for fault classification in series compensated transmission line,” *Neural Comput & Applic*, Vol. 19, 2010, pp. 595-600.
- [20] Ryu, S., Ho, S. H., & Han, I. “Knowledge sharing behavior of physicians in hospitals,” *Expert Systems with Applications*, Vol. 25, 2003, pp. 113-122.
- [21] Stamou, S. & Ntoulas, A. “Search personalization through query and page topical analysis,” *User Model User-Adap Inter*, Vol. 19, 2009, pp. 5-33.
- [22] Tseng, S. S. & Weng, J. F. “Finding trustworthy experts to help problem solving on the programming learning forum,” *Interactive Learning Environments*, Vol. 18, 2010, pp. 81-99.
- [23] Vertommen, J., Janssens, F., De Moor, B., & Duflou, J. R. “Multiple-vector user profiles in support of knowledge sharing,” *Information Sciences*, Vol. 178, 2008, pp. 3333-3346.
- [24] Wasko, M. M., & Faraj, S. “It is what one does: why people participate and help others in electronic communities of practice,” *Journal of*

- Strategic Information Systems*, Vol. 9, 2000, pp. 155-173.
- [25] Wu, X. D., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., & Steinberg, D. "Top 10 algorithms in data mining," *Knowledge and Information Systems*, Vol. 14, 2008, pp.1-37.
- [26] Yimam-Seid, D. & Kobsa, A. "Expert finding systems for organizations: problem and domain analysis and the DEMOIR approach," *Journal of Organizational Computing and Electronic Commerce*, Vol. 13, 2003.
- [27] Zack, M. H. "Managing codified knowledge," *Sloan Management Review*, Vol. 40(4), 1999, pp. 45-58.