基於階層架構資訊及關鍵詞語義擴展的階層式目錄整合研究

洪誠澤　陳英祥　吳秉蓉　楊正仁
元智大學資訊工程學系
Department of Computer Science and Engineering
Yuan Ze University
{chris,sean,pjwu,czyang }@syslab.cse.yzu.edu.tw

摘要

目錄整合的議題近年來受到不少研究的注意。針對攤平式與階層式的分類目錄，分別有不同研究利用來源端目錄所隱含的目錄資訊，有效的提升整合的精確度效能。然而在目前的文獻回顧裡，我們尚未看到使用外部語義庫的資訊來提升階層式目錄整合效能的相關研究。在本論文中，我們探討如何利用外部語義庫與目錄階層架構關係的資訊，使得目錄整合效能可以進一步被提升。在初步實驗中，我們使用最大熵（Maximum Entropy）模型來實作 KSE-ME 整合機制，並用實際的 Web 目錄來進行測試，同時與使用支持向量機核心的 ECI-SVM 一起評估。實驗的結果顯示如果能同時利用階層架構資訊及關鍵詞語義擴展資訊，將可得到良好的整合效果。

關鍵詞：階層式目錄整合，最大熵模型，關鍵詞語義擴展，階層架構索引典資訊

一、緒論

在資訊的處理與組織上，常常會以階層式目錄的架構來分類相關的資訊。而在不同的環境中，也常需要將不同的目錄內容整合在一起。例如，在電子商務市場上，一些大型的網路書店，像亞馬遜書店(Amazon) [1]，需要整合一些下游的零售商的產品目錄，以提供客戶更多元的網路購物選擇。一些大型的商業公司，也須要與它下游廠商提供的零件與產品目錄進行整合，以加強企業內部資源整合。在學術領域上，一些聯合性的學術數位資源網站也會需要整合多個數位圖書館的數位內容目錄，以提供相關研究學者使用這些學術資源。此外，一般使用者最常使用的入口網站與新聞網站，將來也會面臨和其他同質性網站進行目錄內容整合的需求。

從各式分類目錄的實際應用環境顯示，透過分類目錄呈現資訊內容的方式，確實需要一套有效的機制，以提供網路資訊的準確整合和交換。然而，由於網際網路上分類目錄的資訊量愈加龐大與內容日趨廣泛，採用人工整合的方式不僅耗時費力且成本過高，長期維護下來，非常不符合經濟效益。此外，若目錄的規模不斷地成長，人工的整合方法也無法滿足實際的需求。有鑑於此，面對如此日趨龐大的目錄資訊，如何建立準確有效的自動化目錄整合機制，已成爲目錄整合的重要議題。

過往已經有許多研究討論如何進行目錄整合[3,6,9,10,13,14,16,17]。然而當中有許多研究偏重在單純的攤平式目錄整合機制之上，討論將目錄類別全部扁平化以後，整合到攤平式的目的目錄中，並不考慮直接整合到階層式目錄中[3,10,14,16,17]。因此這樣的過程中無法考慮到目的目錄中，類別與子類別之間的從屬關係。由於許多實際的目錄皆是階層式的架構，因此階層式目錄整合機制很需要進一步探討。

過往階層式文件分類研究顯示[8,11]，利用階層式架構確實能有效的加強原本攤平式分類的準確性。其中，McCallum 等人發現在階層式架構下，利用機率模型與聚斂(shrinkage)的方法，可以有效地提升文件分類的準確性。他們的實驗結果並顯示當資料集中的特徵資訊數量足夠多時，階層式架構的分類效果明顯高於採用攤平式架構[11]。然而，他們的研究也顯示，shrinkage 的效果並非絕對能提升階層式分類的準確性。當訓練資料量較大時，反而可能造成某些目錄的準確率下降[11]。

在階層式目錄整合的相關研究上，有不少相關研究皆證實階層式架構在目錄整合上的好處[6,7,9,13]。其中，Doan 等人透過擷取階層式架構中，相鄰節點的相關資訊特徵，來加強本體知識的語意對應[7]。Rajan 等人則是採用最大相似可能性（maximum likelihood）模型，並且充分地討論階層式目錄整合的各種對應情形[13]。他們的方法中更進一步提出，可以將來源目錄的類別新增到目的目錄中，建立新的階層式架構。Chen 等人和 Ho 等人並利用階層式架構索引含的索引典關係（thesaurus），以支持向量機（Support Vector Machines）建立起一個整合機制 ECI-SVM，有效地提升階層式目錄整合的成效[6,9]。

然而過往階層式目錄整合研究只注意到利用階層式架構的好處，我們發現事實上目錄中的語義關係可以進一步被用來加強整合效果。在[15]的研究中顯示，利用外部語義庫（例如 WordNet）自動建立文件叢集的標題時，能有效地建立具語義概念代表性的主題標籤。因此在本研究中，基於過往 ECI-SVM 的研究，探討如何利用[15]的研究成果，找出可以代表文件的關鍵詞，並透過外部語義庫來擴展出輔助語義資訊（Keyword Semantic Expansion），提升目錄整合的效能。

由於統計物理模中的最大熵模型（Maximum Entropy Model，ME）可以將所有特徵的機率條件一起考慮，預測結果不容易受到單獨表現不顯著的資料條件所影響，因此過往研究顯示 ME 在自然語言處理與文件分類上有很好的表現[4,5,12]。另一方面，如果資料中擁有代表性的明顯的機率條件，其最後的結果卻會依據此具代表性的條件來提升整合的效果。因此，我們在研究中採用 ME 做爲階層式目錄整合的模型，並與過往研究中的 ECI-SVM 來分析比較整合效能。我們分別實作以 ECI 爲基礎，但分類器改爲 ME 的 ECI-ME，以及在 ECI-ME 之上加入關鍵字語義擴展（Keyword Semantic Expansion）的 KSE-ME。我們以兩個實際的 Web 階層目錄，進行目錄整合實驗。實驗結果顯示，在階層式架構下，ECI-ME 的準確率表現平均優於 ECI-SVM，並且引用 InfoMap [2]的外部語義庫資訊，KSE-ME 能進一步加強階層式目錄整合的整體成效。

本論文其餘的章節安排如下：第二節將介紹過往目錄整合相關研究，以及外部語義庫來加強文件資訊的研究。在第三節中，我們將簡略介紹最大熵模型分類器，以及如何將關鍵詞語義擴展運用在 ECI 的方法之中。第四節將說明我們的實驗結果。最後，第五節是本研究的結論。

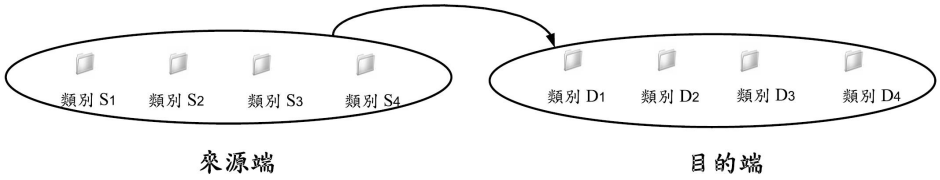
二、相關研究

（一）、目錄整合

目錄進行整合時，最簡單的方式是用基本的文件分類方式來進行，然而過往研究發現可以利用其他相關的資訊來有效提升目錄整合的準確性。以下我們將目錄整合分爲攤平式目錄整合與階層式目錄整合，然後依這二大類分別介紹相關研究。

1、攤平式目錄整合

在目錄整合的研究中，最初僅考慮單純的攤平式目錄整合。如圖一所示，來源端的目錄與目的端的目錄中，分別包含了類別 $S_1,\ldots, S_m$ 以及 $D_1,\ldots,D_n$，這些類別之間相互沒有關連，也沒有上下之間的階層關係。在大多數的早期相關研究中，如果所處理的類別之間有目錄階層關係，便會將這些類別的子類別全部合併成如圖一的攤平式架構來處理。並不討論上下之間的階層關係。



圖一、攤平式目錄整合

例如在 2001 年，Agrawal 和 Srikant 所提出的 Enhanced Naive Bayes (ENB)方法[3]。這個方法主要是挖掘出原始目錄架構裡所具備的隱藏資訊，再利用機率模型中的 NB 分類器(Naive Bayes classifier)，將這些架構隱藏資訊學習出來，最後利用 NB 分類器分析這些資訊，用以提高整合的準確率。實驗的結果顯示，只利用 NB 的原本方法來進行目錄整合時，所達到的正確率爲 47.4%。但是透過 Agrawal 和 Srikant 所提出的 ENB 方法，卻能提升到 61.7%。證明隱含資訊的使用能有效提昇目錄整合正確性。但在他們研究中，僅略微提到可以擴展成階層式的整合方式，卻沒有進一步討論相關作法。

Sarawagi 等人在 2003 年提出一個交互學習(Cross-training, CT) 的方法[14] 來改進目錄整合的正確性，並利用 SVM 分類器和 Expectation Maximization (EM) 分類器來實作整合機制。CT 利用交互學習的方式，將來源端的目錄資訊先擷取出來加入目的端的分類器，如此一來目的端的目錄便可學習來源端的目錄資訊，並加強目錄整合的效果。在他們的實驗中，實驗結果顯示 CT 的機制能夠有效的改進 EM 分類器。但對於 SVM 分類器而言，卻只能對約半數的目錄有改進的效果，不能全面地加強目錄整合的準確性。而他們所討論的整合對象，也還都是攤平式的目錄。

除了上述的相關研究之外，後續還有許多相關研究在探討攤平式目錄整合的應用，提出不同的輔助演算法[10,17]。在 2004 年，Lee 和 Zhang 使用 SVM 並且配合 cluster shrinkage 的方法，以及利用 co-bootstrapping 等兩種機制，來加強攤平式目錄整合的效果。根據他們的實驗結果顯示，這些方法皆能有效地提升攤平式目錄整合的準確性。然而，由於這些加強方法都是基於攤平式的方法來進行目錄整合，並沒有討論如何針對階層式目錄來直接整合。

Wu 等人在 2005 年，利用來源目錄的階層式架構資訊，並使用最大熵模型來進行目錄整合 [16]。其研究結果顯示，整合的效果明顯地改善原有的 ENB 方法。在這個研究當中，主要改進的方法，就是透過來源目錄的階層式架構資訊加強攤平式目錄整合的效果。雖然此研究只使用了目錄的階層架構資訊，並未討論如何對階層式目錄進行整合，但是從他們的研究結果顯示，階層架構資訊確實可以被用來提升目錄整合的效能。

雖然已有研究討論如何使用階層架構資訊，然而它們仍是對攤平式目錄來整合，並未進一步討論如何有效地直接整合階層式目錄。因此雖然這些研究顯示來源端目錄資訊可以提升整合效能，但是對於階層式目錄整合的實際應用卻缺乏討論。因此，如何將兩

個階層式目錄，在不經過扁平化的過程而直接整合，將是一個重要的研究議題。

2、階層式目錄整合

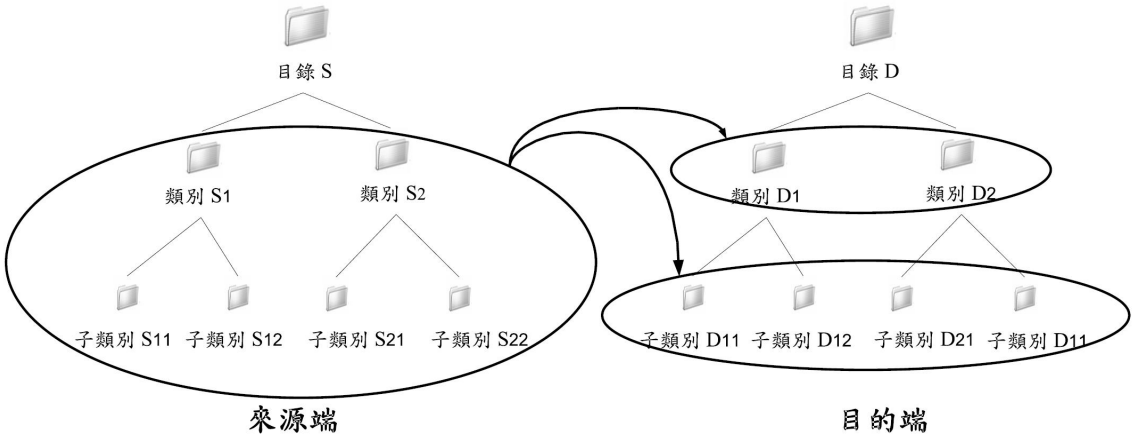
針對目錄整合問題，我們在這裡所討論的階層式目錄整合，與攤平式目錄整合的最大差別，就在於這些階層式目錄並不先被扁平化成攤平式目錄，而是直接進行兩個目錄的整合。如圖二，目錄中的文件存在子類別當中。在進行整合時，會先參考到子類別的父類別資訊，也就是 S_1 或 S_2 類別。在整合至目的端時，會依據文件及階層資訊的特性，可能被整合到類別目錄 D_1 或 D_2 當中，也有可能被整合到子類別目錄當中。

在 2005 年，Rajan 等人提出了一個兩階段的對應與整合方法，並且討論了階層式目錄整合中的四種情形[13]。Rajan 等人有效地利用階層式架構與 maximum likelihood 的方法來提升目錄整合的成效。他們的實驗結果顯示，maximum likelihood 在一對多的階層式架構目錄整合效能非常顯著，並且此研究也顯示目的目錄的階層式架構在目錄整合上能有效地幫助準確率的提昇。

在 2006 年時，Chen 等人[6]和 Ho 等人[9]陸續地進一步探討如何從來源端目錄與目的目錄中，從他們的階層關係中抽取出類似於索引典(thesaurus)的詞彙關係，加強階層式目錄整合的成效。在他們的研究中，提出了一個 ECI 的階層目錄整合方式，採用 SVM 分類器並配合 one-against-rest 的分類機制，讓原本是二元分類法的 SVM 模型能解決多類別的問題。Ho 等人的實驗結果進一步顯示，在 Yahoo！和 Google 上的五個實際的階層式目錄中，利用來源目錄與目的目錄階層式結構的關係，在大多數的攤平式和階層式目錄下，皆能明顯地提升目錄整合的成效[9]。在本研究中，我們並基於 ECI 的方式，另外再考慮加上語義資訊，來提升整合效能。

（二）、外部語義庫

在 2006 年，Tseng 等人[15]提出了一個在相關叢聚的文件中，找出較具有代表性的特徵詞，並從中運用這些特徵詞到外部的語義庫 WordNet 尋找出最具代表性的一個詞，來表示這些特徵詞。Tseng 等人應用 Correlation Coefficient (CC)特徵詞的篩選方法，並且透過所提出的機制來自動選擇 WordNet 相關特徵詞中合適的上位詞，以代表相關文件的標題。例如：找出的特徵詞是 apple、banana、orange，作者進一步利用 WordNet 上位詞關係，找出 fruit 這個詞來當作標籤。實驗結果顯示，此方法可以自動地選擇出



圖二、階層式目錄整合

合適的特徵詞與適切的叢聚文件標題。

從過往階層式目錄整合的研究中，我們發現若能適當地運用來源目錄與目的目錄之間的階層式架構關係，並且透過合適的語義詞典將兩階層式目錄的語義關係加強，將能有效地提升階層式目錄整合的準確性。因此，本論文進一步探討如何結合來源目錄與目的目錄的階層式架構關係，並參考 Tseng 等人[15]的研究，運用合適的外部語義庫來擷取相關語義詞彙與叢集標籤，以加強階層式目錄整合的準確性。

三、階層目錄整合方法

（一）、問題描述

在目錄整合的過程中，我們預設使用者乃是針對兩個同質性比較高的目錄來進行整合。所謂「同質性較高」是指兩個目錄具有類似的分類意義，且兩者之間有部分文件是相同的。兩個目錄分別是來源端目錄 S 與目的端目錄 D。S 當中有 n 個相似而不同的類別目錄 $S_1,S_2,...,S_n$ ，而每一個類別底下或者有最多 m 個子類別目錄 $S_{11},S_{12},...,S_{1m}$，其中 m 是該層子類別的最大值。其他層次以此類推。另一方面，D 當中有 p 個相似而不同的類別目錄 $D_1,D_2,...,D_p$，而每一個類別底下或者有最多 q 個子類別目錄 $D_{11},D_{12},...,D_{1q}$。其他層次以此類推。

階層式目錄整合的目的，就是將來源端目錄 S 底下的子類別目錄當中的所有文件能夠正確整合到目的端目錄 D 底下的子類別目錄當中。例如將 Google 目錄的 autos 類別中的文件，整合到 Yahoo!目錄 automotive 的類別裡。在這裡有幾個整合上的議題需要特別討論。首先，由於在真實的目錄環境裡，一份文件可能會被歸類在一個以上的目錄類別中。針對這種情況，我們採用 one-against-rest 的分類機制，以保留真實目錄環境中一份文件可以同時存在兩個以上的類別的情形。第二，兩個實際的目錄架構很有可能並不一樣，爲了簡化評估上的複雜度，因此我們在本研究中，先討論對稱架構上的目錄整合問題，也就是將兩個目錄先簡約成相同的對稱架構，進行階層式目錄整合上的討論。

（二）、最大熵模型

我們使用 Maximum Entropy (ME) 模型 [5]爲整合機制的分類器核心。ME 是一個統計物理學模型技術，用來測量既有資料最大複雜度的條件機率。ME 模型能測量所有可能的分佈，並呈現出最貼近已知條件的資料分佈情形，在文件分類上的運用都能夠有相當不錯的效能[12]。在本研究中，ME 主要是用來計算來源目錄內的文件被整合到目的目錄中的機率分佈。

1、機率分佈的 Entropy

在 Maximum Entropy 模型中，Entropy 用來表示在一個不確定性的情況下，機率分佈的複雜度。對一個 alphabet 集合 X，若它有一組機率分佈模型 $P(x) = \{p(x_1), p(x_2), \ldots, p(x_n)\}, x_i \in X$，則對機率 p 而言，它的 Entropy $H(p)$ 被定義爲：

$$H(p) = -\sum_{\forall x_i \in X} p(x_i) \log p(x_i) \qquad (3.1)$$

如果考慮一組條件機率分佈模型 $p(y|x)$的 Conditional Entropy，則可定義爲：

$$H(p) = -\sum_{\forall x \in X} \tilde{p}(x) p(y \mid x) \log p(y \mid x) \qquad (3.2)$$

其中 $\tilde{p}(x)$ 是由實際現象所觀察出之經驗機率（empirical probability）。

2、Maximum Entropy 的限制

在找出一組使得熵值爲最大的機率前，必須定義出特徵函式（feature function）*f*(*x*,*y*)來表示所要觀察的現象。例如式(3.3)就是一個常見的二元表示法。若其中 *x* 爲特徵詞，*y* 代表文件的集合，則 *f*(*x*,*y*)=1 時所滿足的條件是表示特徵詞 *x* 出現在文件集合 *y* 中。

$$f(x, y) = \begin{cases} 1 & \text{如果(x,y)滿足條件} \\ 0 & \text{其它} \end{cases} \qquad (3.3)$$

針對所給予的資料集，我們可依需求決定出 *f* (*x*,*y*)之滿足條件。在此資料集中，我們希望針對經驗機率 $\tilde{p}(x, y)$ 算出特徵期望（feature expectation），如式(3.4)。但實際觀察所得到的條件分佈則如式(3.5)的近似函式，其中 $\tilde{p}(x)$ 是由訓練文件所觀察出之經驗機率。特徵期望應與經驗期望一致，因此必須滿足 $E_p\{f\} \equiv E_{\tilde{p}}\{f\}$ 之限制。此外，每一組計算出來的機率值總合必須爲 1，如式(3.6)。

$$E_p\{f\} \equiv \sum_{x,y} \tilde{p}(x, y) f(x, y) \qquad (3.4)$$

$$E_{\tilde{p}}\{f\} \equiv \sum_{x,y} \tilde{p}(x) p(y \mid x) f(x, y) \qquad (3.5)$$

$$\sum_{y} p(y \mid x) = 1 \qquad (3.6)$$

3、最大熵模型之解

Maximum Entropy 的原理是從一個受限制之條件機率分佈集合 *C* 中，找出一個機率模型 p^*，使 Entropy 得到最大值。式(3.7) 的 p^* 即爲 Maximum Entropy 的解法。

$$p^* = \arg\max_{p \in C} H(p) \qquad (3.7)$$

因此只要確定出 p^* 就可得到 Maximum Entropy。從 p^* 的式(3.7)與 Entropy 本身的兩個限制，帶入 Lagrange Multipliers 來處理（推演過程可參考[5]），可以得到式(3.8)來計算文件分類的條件機率。透過 Maximum Entropy 計算 *p*(*y*|*x*)的機率值，其中 *y* 是文件集合，*x* 爲特徵詞的集合，f_i (*x*,*y*)表示是第 *i* 個特徵函式，*z*(*x*)的計算如式(3.9)。

$$p(y \mid x) = \frac{1}{z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x, y) \right) \qquad (3.8)$$

$$z(x) = \sum_{y} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x, y) \right) \qquad (3.9)$$

因此只要計算出最合理的 λ 值，就可以得到最大的 Entropy 的機率值。機率模型中的每個特徵詞都會有一個 λ 值，其權重由一個 Improved Iterative scaling（IIS）的方程

式計算所得。主要是爲了要使每個 λ 值滿足以下方程式，相關內容可參考[4]。圖三爲 IIS 的演算法，在一開始的時候，給 λ_i 一些隨機產生趨近於 0 的數值。接著在迴圈中重複作微分的動作，直到結果收斂爲 0。因此，IIS 演算法的中需要調整 λ_i 的值，使其滿足微分式能等於 0。當微分結果收斂爲 0 時，就將 λ_i 代入 $\lambda_i + \delta_i$，並產生預測的 λ_i 值與最大熵值。

IIS Algorithm

1.Start with some value for each λ_i

2.Repeat until convergence:

Find each δ_i by solve the equation: $\frac{\partial B(\Delta)}{\partial \delta_i} = 0$

Set $\lambda_i \leftarrow \lambda_i + \delta_i$

圖三、Improved Iterative scaling演算法

期望值方程式(3.9)的功能是要滿足每個條件機率，使式(3.8)計算出來的值，在式(3.6 加總爲 1。因此 Maximum Entropy 分類器能夠在目錄整合時，保證每個特徵詞能夠滿足機率值總和爲 1。

（三）、關鍵詞語義擴展

1、目錄整合流程

目錄整合的流程如圖四。目錄整合的步驟包含網頁文件處理，特徵詞選取。然後從特徵值中進行關鍵詞的語義擴展，自外部語義庫中取出適當的上位詞（hypernyms），並加入階層架構的索引典資訊，與文件原有特徵詞共同組合成擴展文件特徵。最後將此擴展文件特徵轉換爲 ME 分類器格式，進行目錄整合。以下將進一步說明各個步驟。

2、文件處理

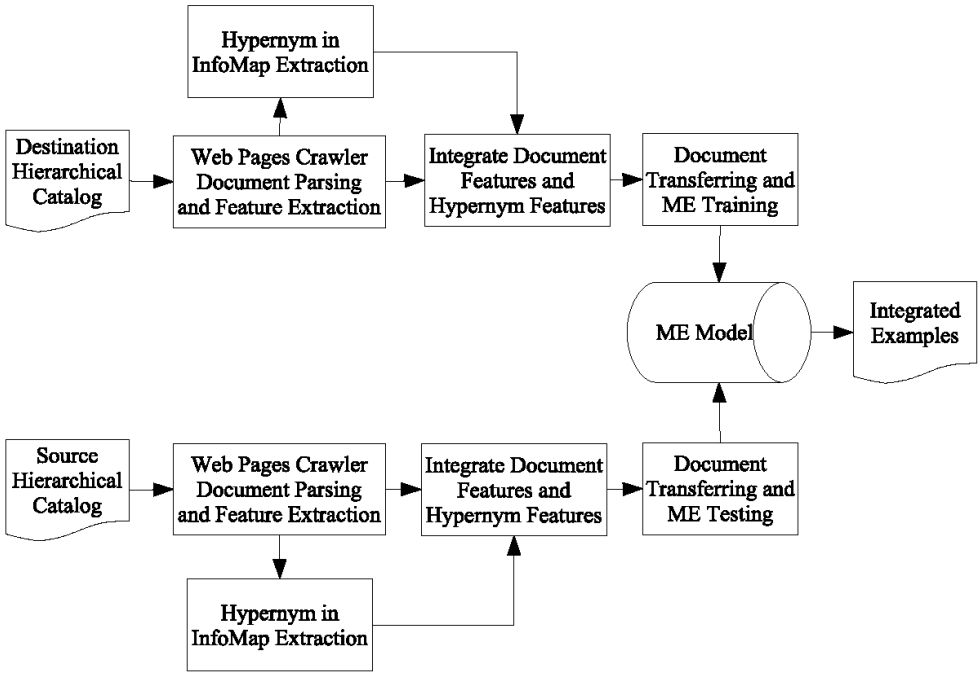
在此步驟中，我們將對 Web 網頁文件做一些前置處理。這些前置處理包含：移除 Web 網頁文件中的 HTML 標籤，script 網頁執行碼和文字。移除文章中的 stop word，並用 Porter 的演算法對每個詞做 stemming 處理。處理後的單詞做爲文件的特徵詞。

3、文件中特徵詞權重計算

特徵詞權重的計算方式，一般在實作上有 TF-IDF、TF 與 TF/*sum*(TF)等方法。我們考慮到 TF-IDF 在計算時會因目錄的變動而常常需要更新，在實際使用上將會花費許多計算時間，因此在本研究中，我們使用如式 (3.10)來計算 TF/*sum*(TF) 特徵詞權重計算。

$$f_i = \frac{TF(w_i, d)}{\sum_{i=0}^{n} TF(w_i, d)} \qquad (3.10)$$

其中 w_i 表示是第 i 個特徵詞，d 爲文件，$TF(w_i,d)$爲 w_i 這單字或片語在文件 d 中所出現的頻率，總共有 n 個特徵詞。



圖四、引用外部語義庫進行階層式目錄整合流程圖

4、類別關鍵詞選取

如果將文件中所有的特徵詞都將其上位詞加入，將會使文件充滿過多無關重要的資訊而影響整合效果。因此在進行語義擴展的時候，必須先對文件中的特徵詞篩選出重要的關鍵詞，再針對這些具有代表性的關鍵詞來進行語義擴充，也就是將他們的上位詞加入特徵詞集合當中，來輔助整合效果。同時，爲避免在同一類別中，不同文件之間的關鍵詞仍可能存有分歧，因此我們針對一個目錄類別來選取可以代表該類別的關鍵詞。

在過往研究中可以發現，以 Correlation Coefficient 的方式，可以擷取到具有代表性的關鍵詞 [15]。因此我們利用 Correlation Coefficient 的方式來抽取關鍵詞，再進行語義擴展。也就是是針對同一個目錄底下所有文件中的特徵詞，透過底下方程式(3.11)計算出能夠代表此目錄當中每一個特徵詞的權重。在 (3.11) 式當中的 TP 表示類別 C 之文件含有特徵詞 T 的文件數，FP 表示 C 以外其他類別之文件含有特徵詞 T 的文件數，FN 表示 C 以外其他類別之文件不含有特徵詞 T 的文件數，TN 表示類別 C 之文件不含有特徵詞 T 的文件數表示在目錄底下有包含或者沒有包含此特徵詞的正負向文件個數。所算出來的 Co(T,C) 表示在類別 C 中特徵詞 T 與類別 C 的 Correlation Coefficient。透過 Correlation Coefficient 方法可以精確的選擇出一些針對此目錄較具代表性的特徵詞。

$$\mathrm{Co}(\mathrm{T},\mathrm{C}) = \frac{(\mathrm{TP}\times\mathrm{TN}+\mathrm{FN}\times\mathrm{FP})}{\sqrt{(\mathrm{TP}+\mathrm{FN})(\mathrm{FP}+\mathrm{TN})(\mathrm{TP}+\mathrm{FP})(\mathrm{FN}+\mathrm{TN})}} \qquad (3.11)$$

5、關鍵詞語義擴展

經由以上式 (3.11) 所計算出來的權重，反映出該特徵詞在該類別中的關連性，因此我們擷取出權重最高的 5 個特徵詞，視爲可以代表該類別的關鍵詞，查詢他們在外部

語義庫 InfoMap 的上位詞資訊，進而將這些上位詞算出權重後加入特徵值空間中。例如某一目錄類別最高的 5 個特徵詞為 output, signal, circuit, input, frequency，透過 InfoMap，我們可得到 signal, signaling, sign, communication, abstraction, relation 等 6 個上位詞，再用下式 (3.12) 計算出第 i 個上位詞的權重 HW_i。由這些權重，我們可以對一個文件 d 來決定它的上位詞特徵向量 H_d。

$$HW_i = \frac{HF_i}{\sum_{i=0}^{n} HF_i} \qquad (3.12)$$

其中 HF_i 代表從這 5 個特徵詞所取出的所有 n 個上位詞中，第 i 個上位詞出現的頻率。

6、階層架構的索引典資訊計算

在過往研究中發現，階層目錄中每一類別的標籤資訊架構，例如 Google 目錄中的 “recreation/autos/” 該類別的說明，可視為該目錄的一個索引典（thesaurus），在目錄整合上相當有幫助 [6,9]。此資訊可以是該類別的名稱，亦或者是該類別說明。在實驗中，我們以該類別的說明來組成其索引典。

因此，針對一個文件 d，我們將它在目錄的索引典特徵向量 L_d，與經由關鍵詞語義擴展後所得的上位詞特徵向量 H_d，以及原先文件中的特徵向量 F_d 一起整合，如式 (3.13) 所示。此外，透過 λ 、α 來取得權重的平衡，此方法所計算出來的特徵向量 FE_d，即可加以提升整合正確性。當中的 λ 主要是調整 [6,9] 所提出加入階層式目錄資訊。如表一中，文件所存在的目錄資訊權重為 $1/2^1$，再上一層的目錄資訊權重為 $1/2^2$，以指數遞減的方法來計算。另一方面，α 是調整文件當中本身特徵詞與從外部語義庫所取出的上位詞資訊。將每個特徵詞的權重計算出來，在目錄整合時，可以依照較高權重的特徵詞加強來源端的資訊，進而提升整合效能。

$$FE_d = \lambda \frac{L_i}{\sum_{i=0}^{n} L_i} L_d + (1-\lambda)\left[\alpha \times H_d + (1-\alpha) \times F_d\right] \qquad (3.13)$$

表一、階層式目錄標籤權重

Hierarchical	**Label Weight**
Document Level L_0	$1/2^0$
One Level Upper L_1	$1/2^1$
Two Level Upper L_2	$1/2^2$
.	.
.	.
.	.
nLevel Upper L_n	$1/2^n$

7、目錄整合

最後，將上述的特徵值轉換成 Maximum Entropy 模型分類器的資料格式，進行目錄整合工作。每一份文件都會有自己本身的目錄資訊和＜*feature*, *weight*＞配對而成立。因此每一份文件被定義為＜*line*＞=＜*target*＞＜*feature*＞： ＜*value*＞...＜*feature*＞：＜*value*＞。當中的 target 定義為 1（表示是正向文件）或是－1（表示是負向文件），＜*feature*

＞為文件當中的特徵詞，＜*value*＞為特徵詞的權重。

四、目錄整合實驗

為了瞭解這些改進方式的效能，我們以真實的目錄來進行實驗，分別自 Google 和 Yahoo!取得部份目錄。在實驗中，我們所採用的外部語義庫是 InfoMap。在實驗上，我們依據 ECI 方法 [9] ，並加以運用到 SVM (ECI-SVM) 與 ME (ECI-ME) 分類器中。此外，在 ECI-ME 上，再進行關鍵詞語義擴展 (KSE-ME)，以下將進一步說明各項細節。

（一）、實驗環境

1、資料集

在實驗的部份，我們從兩個實際目錄：Google 和 Yahoo!，分別取得 5 個分項階層目錄來當作資料集，表二和表三是這些階層目錄的根節點目錄名稱。表四是這 5 個階層目錄當中所擷取下來文件的數量和類別數量。其數量依 5 個目錄 Autos、Movies、Outdoors、Photo 和 Software 來區分。

表二、Yahoo!中目錄的根節點位址

Category	**Matched URL**
Autos	http://dir.yahoo.com/recreation/automotive/
Movies	http://dir.yahoo.com/entertainment/movies and film/
Outdoors	http://dir.yahoo.com/recreation/outdoors/
Photos	http://dir.yahoo.com/arts/visual arts/photography/
Software	http://dir.yahoo.com/computers and internet/software/

表三、Google 中目錄的根節點位址

Category	**Matched URL**
Autos	http://dir.google.com/top/recreation/autos/
Movies	http://dir.google.com/top/arts/movies/
Outdoors	http://dir.google.com/top/recreation/outdoors/
Photos	http://dir.google.com/top/arts/photography/
Software	http://dir.google.com/top/computers/software/

透過每個網頁目錄節點中所包含的次目錄和對外連結，依此建立目錄之間的上下層關係和向外的連結。整合的過程會依向外的連結，取得文件。以 Google 目錄為例，我們進行實驗的文件當中，差集的部份｜Google－Yahoo!｜是訓練文件（G-Y），交集的部份｜Google∩Yahoo!｜是測試文件（G Test）。

表四、實驗中所用到的目錄類別數，以及訓練與測試文件數量

	Yahoo!			**Google**		
	Y-G	Y Class	Y Test	G-Y	G Class	G Test
Autos	1681	24	436	1096	12	426
Movies	7255	27	1344	5188	26	1422
Outdoors	1579	19	210	2396	16	208
Photo	1304	23	218	615	9	235
Software	1876	15	710	5829	27	641
Total	13965	108	2918	15124	90	2932

2、測量方式

在實驗當中我們的測量整合時的精確率 (P，precision)與召回率 (R，recall)，以及 $F_1 = 2PR/(P+R)$，來比較不同系統的成效。在目前實驗中，我們允許一份文件可以被整合到多個目錄類別中。因此 Precision 的算法是（正確分類的文件數/所有分至該類的文件數），Recall 的算法是（正確分類的文件數/應該分至該類的文件數）。Recall 也可看成是過往研究中的整合精確度。

除此之外，我們也同時考量多個類別的整體表現，因此也使用 micro-average 與 macro-average 兩種平均方法。Micro-average 由於是全部文件一起累加統計，不分類別，因此容易受到佔大多數的大件類別影響。相對地，Macro-average 考慮每個類別的成效後再做平均，因此容易受到大量的小類別而影響。

3、外部語義庫

在目前研究中，我們使用的外部語義庫是 InfoMap [2] 來尋找上位詞。事實上，其他的外部語義庫也可以使用，例如 WordNet，然而在過往研究中發現，使用 InfoMap 所擴展的結果與使用 WordNet 的結果所得的成效相當接近 [15]。因此換用 WordNet 可能也會有與目前類似的分類表現。未來在我們的研究計畫中，預備將進一步探討外部語義庫的品質對於整合品質的影響。

4、分類器

在分類器上，我們使用 SVM^{light} 來實作 ECI-SVM 的整合機制，使用 linear kernel 以及預設的參數，版本為 5.00 版。ME 分類器則使用 Edinburgh 大學的 ME 工具。所使用的 ME 工具的版本為 20041229 版。

（二）、實驗結果與討論

在實驗中，我們設定不同的 λ 值。為了解不同 λ 值的影響，在來源目錄中，我們將 λ_s 值設定從 0.00 到 1.00，而在目的目錄中，λ_d 值設定為 0.00，0.01，0.05 分別來看它們的成效如何。如此取的原因是，在過往研究中發現 λ 值過大其實會將 Recall 值降低 [6,9]。在我們實驗中也確實有如此情況。另一個參數 α 值，我們目前只作了一些初步測試，由於篇幅的關係，此處僅報告 α=0.1 的情況(KSE-ME)。在表八與表九的結果中，我們可以看到隨著 λ_s 的增加，Macro-Recall 與 Micro-Recall 都同時下降。

表六到表十一顯示由 Google 到 Yahoo!階層式目錄的效能。在表六與表七當中，可發現 Macro-Precision 與 Micro-Precision 的效果都不夠良好。這是因為在實驗中，我們允許一份文件可以分至多個目錄類別所致。所以在 λ_s 值較低時，由於缺乏階層架構索引典的輔助，False-positive 的比例會普遍升高，使 Precision 表現不佳。但我們可以看出，隨著 λ_s 值提高，False-positive 的比例逐步下降，使 Precision 逐步升高。同時，我們也可以發現，採用關鍵詞語義擴展的 KSE-ME，在 Precision 上有最好的表現。

在表八與表九中，我們可以發現，KSE-ME 在 Recall 的表現上，普遍都要比 ECI-ME 為佳。至於 ECI-SVM，雖然其 Precision 的表現不如 ECI-ME 與 KSE-ME，但在 Recall 的表現上，反而普遍有最好的表現。但可從表中看出，當 λ_s=0.05 這個常在實驗中使用的值時，KSE-ME 依然有領先的表現。

若從 F_1 的表現上來看，表十和表十一顯示 KSE-ME 都有不錯的成效，比 ECI-ME 與 ECI-SVM 的 F_1 表現都來得好。因此，從目前初步的實驗可以得知，利用階層架構索引典資訊以及關鍵詞語義擴展，階層式目錄整合的效能可以有效的提升。

表六、階層式目錄整合的 Macro-Precision

		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
macroP		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
	0.00	1.08%	1.09%	1.24%	1.11%	1.12%	1.21%	1.32%	1.32%	1.42%
	0.10		1.18%	3.12%		1.28%	2.77%		2.36%	9.05%
	0.20		1.34%	8.37%		1.74%	14.80%		4.93%	26.55%
	0.30		1.82%	13.18%		3.04%	30.04%		10.09%	36.34%
	0.40		3.50%	15.32%		5.98%	35.96%		17.62%	42.42%
λ_s	0.50		6.87%	16.25%		12.98%	38.62%		22.82%	47.55%
	0.60		9.45%	16.81%		20.91%	40.42%		30.08%	49.37%
	0.70		12.68%	17.20%		23.85%	41.34%		33.96%	50.12%
	0.80		14.69%	17.52%		24.95%	41.81%		35.31%	51.01%
	0.90		16.86%	17.90%		24.38%	42.17%		35.52%	51.96%
	1.00		17.64%	17.93%		24.93%	42.40%		36.05%	52.72%

表七、階層式目錄整合的 Micro-Precision

		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
microP		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
	0.00	1.08%	1.09%	1.23%	1.11%	1.12%	1.18%	1.29%	1.30%	1.40%
	0.10		1.18%	2.43%		1.28%	2.26%		1.75%	4.80%
	0.20		1.34%	6.34%		1.71%	9.82%		2.84%	18.52%
	0.30		1.80%	9.86%		2.84%	23.47%		4.81%	29.10%
	0.40		3.29%	11.33%		5.58%	30.47%		6.94%	35.71%
λ_s	0.50		6.21%	12.24%		10.93%	34.10%		11.32%	39.81%
	0.60		8.90%	12.87%		18.58%	37.24%		19.35%	40.81%
	0.70		11.71%	13.46%		20.91%	38.79%		25.17%	41.34%
	0.80		13.27%	13.95%		21.36%	39.32%		26.97%	41.97%
	0.90		14.76%	14.26%		20.29%	39.67%		27.37%	42.41%
	1.00		15.27%	14.33%		20.33%	39.92%		28.13%	43.61%

表八、階層式目錄整合的 Macro-Recall

		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
macroR		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
	0.00	94.66%	94.66%	88.48%	95.80%	96.05%	94.42%	97.43%	97.69%	97.69%
	0.05		95.83%	94.31%		95.96%	92.09%		96.99%	94.80%
	0.10		95.67%	92.44%		92.98%	85.16%		93.80%	89.98%
	0.20		92.41%	86.91%		87.57%	79.65%		89.21%	83.12%
	0.30		89.58%	83.66%		84.40%	76.92%		85.06%	81.02%
	0.40		86.54%	81.54%		81.89%	76.40%		81.68%	78.32%
λ_s	0.50		84.11%	80.12%		79.85%	75.54%		78.66%	76.99%
	0.60		80.72%	79.28%		78.20%	74.87%		76.79%	76.46%
	0.70		79.37%	78.53%		76.70%	74.35%		75.42%	76.12%
	0.80		77.69%	77.89%		76.22%	73.49%		74.08%	76.07%
	0.90		77.18%	77.82%		74.67%	72.94%		73.49%	76.00%
	1.00		76.56%	77.79%		73.22%	72.83%		73.31%	75.93%

表九、階層式目錄整合的 Micro-Recall

microR		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
λ_s	0.00	94.76%	94.79%	87.91%	96.20%	96.44%	94.59%	97.26%	97.40%	97.60%
	0.05		96.23%	95.14%		96.85%	94.35%		97.33%	96.27%
	0.10		96.37%	93.63%		95.51%	90.00%		95.68%	92.36%
	0.20		94.69%	89.69%		91.88%	85.41%		91.92%	86.06%
	0.30		92.74%	87.39%		88.69%	82.46%		87.87%	83.59%
	0.40		90.10%	85.58%		86.47%	82.01%		84.65%	81.02%
	0.50		88.08%	84.52%		84.96%	80.92%		81.67%	79.86%
	0.60		85.17%	83.66%		83.86%	80.40%		79.72%	79.48%
	0.70		83.56%	82.91%		82.87%	79.82%		78.07%	79.10%
	0.80		81.71%	82.19%		82.53%	79.24%		77.05%	79.07%
	0.90		81.06%	82.05%		81.64%	78.66%		76.50%	78.93%
	1.00		80.54%	82.01%		79.38%	78.59%		76.26%	78.76%

表十、階層式目錄整合的 Macro-F_1

macroF		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
λ_s	0.00	2.13%	2.15%	2.44%	2.19%	2.21%	2.38%	2.60%	2.61%	2.81%
	0.10		2.32%	5.92%		2.52%	5.27%		4.60%	16.44%
	0.20		2.62%	14.59%		3.39%	23.59%		9.34%	40.24%
	0.30		3.54%	21.99%		5.80%	41.84%		18.04%	50.17%
	0.40		6.65%	24.85%		11.01%	47.65%		28.99%	55.03%
	0.50		12.42%	25.99%		21.91%	50.03%		35.37%	58.79%
	0.60		16.57%	26.66%		32.55%	51.53%		43.23%	60.00%
	0.70		21.62%	27.14%		35.89%	52.26%		46.83%	60.44%
	0.80		24.34%	27.52%		37.01%	52.48%		47.82%	61.07%
	0.90		26.99%	28.00%		36.04%	52.67%		47.89%	61.72%
	1.00		27.81%	28.05%		36.36%	52.83%		48.34%	62.23%

表十一、階層式目錄整合的 Micro-F_1

microF		λ_d								
		ECI-SVM			ECI-ME			KSE-ME		
		0.00	0.01	0.05	0.00	0.01	0.05	0.00	0.01	0.05
λ_s	0.00	2.14%	2.16%	2.43%	2.19%	2.22%	2.33%	2.54%	2.56%	2.77%
	0.10		2.33%	4.74%		2.53%	4.40%		3.44%	9.13%
	0.20		2.63%	11.85%		3.35%	17.61%		5.51%	30.48%
	0.30		3.54%	17.72%		5.50%	36.54%		9.13%	43.17%
	0.40		6.35%	20.01%		10.48%	44.43%		12.83%	49.57%
	0.50		11.60%	21.38%		19.37%	47.98%		19.88%	53.13%
	0.60		16.11%	22.31%		30.42%	50.90%		31.14%	53.93%
	0.70		20.54%	23.16%		33.40%	52.21%		38.07%	54.30%
	0.80		22.84%	23.86%		33.93%	52.56%		39.95%	54.83%
	0.90		24.97%	24.30%		32.50%	52.74%		40.31%	55.17%
	1.00		25.67%	24.40%		32.37%	52.94%		41.10%	56.13%

五、結論

隨著網路資訊蓬勃發展與快速整合和交換的需求，目錄整合在許多領域已經成爲重要的議題，在過往研究中，已從初步的攤平式目錄整合，逐漸深入到階層式目錄整合的討論。因此在本論文中，我們針對階層式目錄整合，再進一步討論整合效能加強的方式。我們架構於之前研究的階層架構索引典資訊的加強方法，另外提出使用關鍵詞語義擴展

的方式，來進一步增進階層式目錄整合效能。

在我們的初步實驗中可以看出，使用階層架構索引典資訊與關鍵詞語義擴展這兩個方式的 KSE-ME，在 Precision 上有很好的表現，在 Recall 上雖不能普遍比 ECI-SVM 來得好，但也還是普遍比 ECI-ME 來得佳。在綜合考量的 F_1 的評估上，KSE-ME 具有最好的表現。

從我們目前的研究成果可以發現，如何在不降低 Recall 表現的同時，還能夠減少 False-Positive 的整合技術，仍待進一步討論。如此，當 Precision 也提高的時候，也將是階層式目錄整合技術趨於成熟，能夠運用在實際環境中的時候。未來我們也計畫進一步討論外部語義庫的品質，對於整合效能影響的關鍵因素，期待能探索出有效提升目錄整合效能的方式。

致謝

本研究感謝國科會計畫 NSC-96-2221-E-155-067 的支助，並感謝論文審查委員寶貴的建議。

參考文獻

[1] http://www.amazon.com.

[2] "Information mapping project," Computational Semantics Laboratory, Standford University. [Online]. Available: http://infomap.stanford.edu/.

[3] R. Agrawal and R. Srikant. "On Integrating Catalogs." In *Proceedings of the 10th WWW Conference. (WWW10)*, pp. 603–612, Hong Kong, May 2001.

[4] A. Berger. "The improved Iterative Scaling Algorithm: A Gentle Introduction." *Technical report*, 1997.

[5] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. "A Maximum Entropy Approach to Natural Language Processing." *Computational Linguistics,* pp. 39–71, 1996.

[6] I.-X. Chen, J.-C. Ho, and C.-Z. Yang. "On Hierarchical Web Catalog Integration with Conceptual Relationships in Thesaurus." In *Proceedings of the 29th International ACM SIGIR (SIGIR 2006)*, pp. 635–636, Seattle, Washington, USA, 2006.

[7] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. "Learning to Map between Ontologies on the Semantic Web." In *Proceedings of the 11th WWW Conf. (WWW2002)*, pp.662–673, Honolulu, Hawaii, 2002.

[8] S. Dumais, and H. Chen. "Hierarchical Classification of Web Content." In *Proceedings of the 23rd Annual ACM Conf. on Research and Development in Information Retrieval (SIGIR'00)*, pp.256–263, Athens, Greece, 2000.

[9] J.-C. Ho, I.-X. Chen, and C.-Z. Yang. "Learning to Integrate Web Catalogs with Conceptual Relationships in Hierarchical Thesaurus." In *Proceedings of the 3rd Asia Information Retrieval (AIRS 2006)*, pp. 217–229, Singapore, 2006.

[10] W. S. Lee and D. Zhang. "Web Taxonomy Integration through Co-Bootstrapping." In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pp. 410–417, 2004.

[11] A. MaCallum, R. Rosenfeld, T. Mitchell, and A. Ng. "Improving Text Classification by Shrinkage in a Hierarchy of Classes." In *Proceedings of the 15th International Conference on Machine Learning (ICML-98),* pp. 359–367, Madison, Wisconsin, 1998.

[12] K. Nigam, J. Lafferty, and A. McCallum. "Using Maximum Entropy for Text Classification." In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61–67, Oct. 1999.

[13] S. Rajan, K. Punera, and J. Ghosh. "A Maximum Likelihood Framework for Integrating Taxonomies." In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pp.856–861*,* Pittsburgh, Pennsylvania.

[14] S. Sarawagi, S. Chakrabarti, and S. Godbole. "Cross-Training: Learning Probabilistic Mappings between Topics." In *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 177–186, 2003.

[15] Y.-H. Tseng, C.-J Lin, H.-H Chen, and Y.-I. Lin. "Toward Generic Title Generation for Clustered Documents." In *Proceedings of the 3rd Asia Information Retrieval (AIRS 2006)*, pp. 145–157, 2006.

[16] C.-W. Wu, T.-H. Tsai, and W.-L. Hsu, "Learning to Integrate Web Taxonomies with Fine-Grained Relations: A Case Study Using Maximum Entropy Model." In *Proceedings of the 2nd Asia Information Retrieval Symposium 2005 (AIRS 2005)*, pp. 190–205, Jeju Island, Korea, Oct. 2005.

[17] D. Zhang and W.S. Lee. "Web Taxonomy Integration using Support Vector Machines." In *Proceedings of the 13th WWW Conference (WWW2004),* pp.472–481, New York, NY, May 2004.