國立臺灣師範大學
資訊工程研究所碩士論文

指導教授：柯佳伶　博士

社群標籤系統中查詢結果標籤階層式組織

技術之研究

Hierarchical Tag Organization for Browsing Query Results on Social Tagging Systems

研究生：邱俊嘉　撰

中華民國　一百零二　年　七　月

摘要

社群標籤系統中查詢結果標籤階層式組織技術之研究

邱俊嘉

本論文以標籤資源為研究資料，考慮使用者在以查詢字於社群標籤資源中進行搜尋，探討如何從搜尋結果物件的標籤找出有效篩選物件的標籤字，並自動組織成概念階層架構，以方便使用者進行進一步選取所需物件。我們從包含查詢字為標籤的物件中，以這些物件包含的標籤當作候選標籤字，從中挑選出與查詢字相關度較高的前 k 個標籤作為代表標籤。我們以人為給定有上下概念包含關係的標籤配對組合為訓練資料，根據個別標籤字在資料物件的多種出現特徵，利用 Rank-SVM 模型學習判別語意概念高低排序模型。此外，同樣以人為給定具語意包含關係及不具語意包含關係的兩類標籤配對為訓練資料，根據標籤配對中兩個標籤在資料庫中出現情況所計算出的多種特徵，運用 SVM 模型學習出判斷兩個標籤是否有語意包含關係的分類模型。我們將查詢結果代表標籤字及其特徵輸入排序模型中進行語意概念廣度的排序。依照其排序結果之順序一一加入概念架構中，再由分類模型判斷每一個新加入概念架構的代表標籤可作為在概念架構中那些標籤下的子概念，建立出標籤概念階層式架構。實驗結果顯示，本論文方法所挑選的代表標籤字並進行建立語意階層式架構，能夠有好的查詢效果；同時本論文提出的階層式架構建立方法也能找出具語意包含關係的標籤架構。

關鍵字: 社群標籤資源、查詢標籤推薦、階層式架構

Abstract

Hierarchical Tag Organization for Browsing Query Results on Social Tagging Systems

By

Jiun-Jia Chiou

This thesis considers the scenario that users give short queries to search the resources with tags. In order to help users find the required resources efficiently, our goal is to study how to find the tags used for further filtering the objects in the query results and construct a concept hierarchy for these tags automatically. At first, we find out the query results which consist of all the objects with tag sets containing the query terms. All the tags of these objects are called the candidate tags. Among these candidate tags, we select the top-*k* tags whose relatedness with the query is the highest, which are called the representative tags. In the offline-processing, according the various features of tags, a collection of tag pairs that have relationships of semantic containment is used as training data to learn the concept-abstraction sorting model by using Rank-SVM. In addition, based on the co-occurring features between a pair of tags got from the corpus, we use SVM to construct a classification model for deciding whether a tag represents a sub-concept of another tag. Then the representative tags and their features are inputted to the concept sorting model to get a sorted list according to their degrees of concept abstraction. Each tag in the sorted list is added into the concept hierarchy of tags one by one. The constructed classification model is used to decide whether a newly added representative tag can serve as a sub-concept of the other tags existing in the concept hierarchy. The experimental results show that performing the proposed representative tag selection method before constructing the concept hierarchy of tags can improve the effectiveness of searching. Furthermore, the proposed method of constructing concept hierarchical of tags can find a good result with level-wise sematic relationships among the representative tags.

Keywords: Social-tagging resources, query tag recommendation, hierarchical architecture

誌謝

本論文能夠順利完成，首先誠摯地感謝我的指導教授—柯佳伶老師。老師悉心的教導使我在研究過程上能夠順利進行，適切的討論並指點我正確的方向，使我在這兩年中對於資料探勘領域的知識和實作技術獲益匪淺。因此，在老師給予的叮嚀及鼓勵下，不論是討論研究的細節及可行的方法或是心靈方面的培養，皆令我在學術研究上留下許多深刻且精采美好的體驗與成長。在論文撰寫的過程中，老師不厭其煩地修正論文內容，讓我把論文寫的更臻完善。十分感激老師在這段時間的指導，僅以此誌謝感謝老師的辛勞。另外也感謝林宜鴻教授與徐嘉連教授在百忙之中撥空擔任我的口試委員，對於我的研究提供許多寶貴的建議。

在碩士班的日子裡，所結識的夥伴都是幫助我成長的重要人物。感謝光庭、昇宏和柏先學長在日常生活的種種關懷以及在研究上的問題給予幫忙、解惑。在這過程中也少不了同學們的互相體諒與幫忙，我的同袍戰友們—爾剛、奕智。無論是修課、進行研究或是在生活上，總是能帶來歡樂的氣氛並共同努力，很開心能與你們成為同學！另外碩一學弟妹楨喻、舜宸、懿萱、張崴當然也不能忘記，在心情起伏的日子裡幫我打氣並且也不遺餘力的幫助我進行實驗，妳們的幫忙我銘記在心。

最後由衷感謝我的家人，總是在我文思困頓時，給我的支持與關懷。感謝你們讓我看見父母對子女無私的愛並且提供我無虞的環境，讓我能順利完成碩士學業！

邱俊嘉　僅識

於國立台灣師範大學資訊工程研究所

2013 年 7 月 31 日

目錄

附表目錄

附圖目錄

第一章 緒論

1.1 研究動機及目的

近來社群網站平台上(例如 Flickr, del.icio.us , CiteULike 等),讓使用者可以對上傳分享物件給予標籤已經成為一種趨勢。因此使用者可以在社群網路分享的平台上，透過物件具有的標籤當作查詢關鍵字，系統回傳的結果為所標示標籤集中包含查詢關鍵字的資料物件(e.g.相片、文章、影片..等)。然而由於有些字彙的語意相當廣泛，相同的標籤字可能代表多種意思。當使用者給定查詢關鍵字進行搜尋，系統回傳的結果可能混雜著許多不同概念意涵的資料。多數使用者所下查詢中的字數相當少，當這查詢中的查詢字涵蓋的語意廣泛時，容易發生回傳的結果過多，需要花費大量的時間瀏覽一個清單頁面(多項資料物件)，無法讓使用者有效率找到需要的資料。若能將查詢結果中包含的標籤進行概念分類，將有助於使用者瀏覽篩選所需的資料。以查詢字 apple 為例,它所表達主題內容可能包含:1. 水果 2. 蘋果公司產品 3. 紐約市(Big apple)等不同意涵，如果系統能提供區分各主題的標籤字，則使用者能夠選定適合的標籤字加入查詢，能夠更加清楚地表達使用者的搜尋意圖，進一步進行查詢結果的篩選，以有效減少查詢結果回傳數量。

至於要提供那些標籤字，能夠有效區分查詢結果中不同概念的資料物件，是我們需要考量的。以圖 1.1 所示:當所下的查詢字為"apple"，與查詢字同時出現的字有許多，若能從這些字中找出代表標籤並依其語意架構組織起來，則可有效幫助使用者用來篩選查詢結果。例如: 富士(fuji)、五爪(reddelicious)兩個品種的蘋果歸屬於水果(fruit)的子概念，而 phone、iphone5、ipadmini 則屬於蘋果公司產品(product)的子概念， nyc，newyork 則屬於 city 的子概念，因此可組織成如圖 1.2 所示之概念架構。

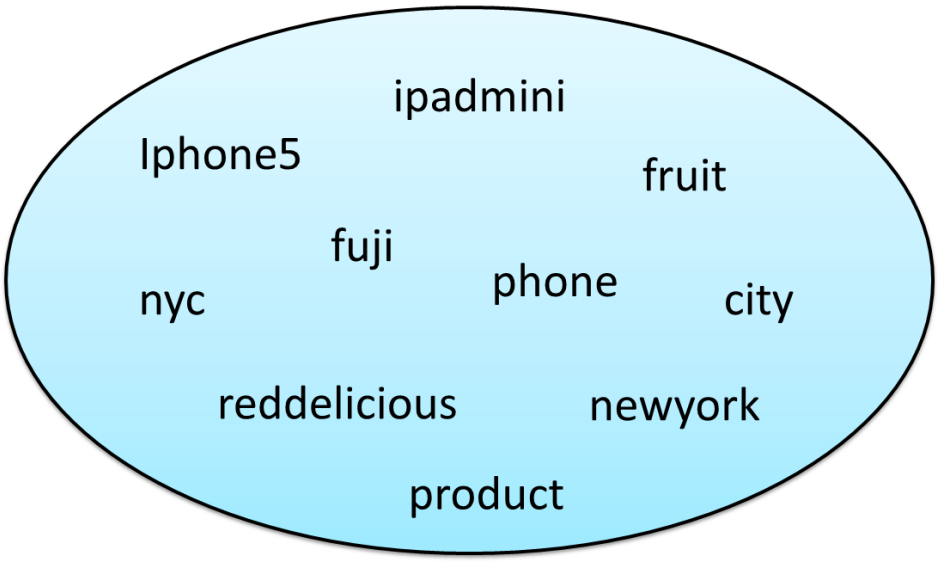


圖 1.1 查詢字"apple"回傳之標籤結果範例

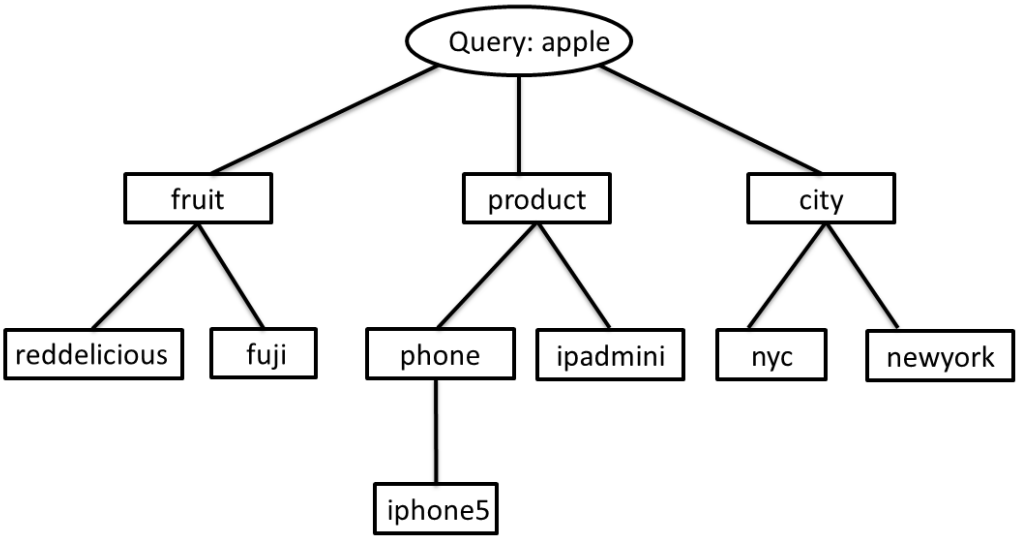


圖 1.2 組織查詢字"apple"回傳的標籤結果

以此種階層性概念架構組織查詢結果物件中的代表標籤，可讓使用者選定代表特定主題概念的標籤，快速地找到相關資料，如同一種面向查詢(faceted search)所提供查詢方式，使用者可選擇概念架構中不同層級的標籤，有效地縮小查詢結果範圍。

本論文之研究目的在於如何利用查詢結果物件中的標籤，從中建議一些可加入查詢中篩選查詢結果的代表標籤字。這些建議的代表標籤字必須考慮所涵蓋的查詢結果物件數目不能太少，且彼此所涵蓋的查詢結果物件重複性不能太高。此外，本論文將針對這些查詢字結果代表標籤，研究如何在不利用外部資源的情況下，分析其在資料物件中出現的特徵，自動建立可有效輔助查詢結果篩選的標籤字概念架構。

1.2 研究的範圍與限制

本論文研究所考慮的資料是具有標籤的資料物件。社交標籤系統平台上的資料，目前只考慮以英文為標籤內容的資料。使用者所給定的查詢為一個包含一個以上的關鍵字所成的集合，查詢結果中的回傳資料物件是該物件的標籤必須包含使用者給定查詢中的所有相同關鍵字。

本論文將研究如何從查詢結果物件所包含的標籤字中，選出代表性標籤字，並自動建立這些代表標籤字的概念架構。因此本論文的研究重點可分成兩個部份:

<1>提出從查詢結果中選取代表標籤的評估方法

<2>提出有效快速建立代表標籤概念架構的建立方法

1.3 論文方法

根據研究目標，本論文所提出方法主要分成兩個部分：查詢結果代表標籤選取方法及代表標籤概念架構建立方法。

<1> 查詢結果代表標籤選取方法

本論文先將包含查詢字資料物件所具有的標籤找出來，根據其在查詢結果中的出現頻率，以及其在標籤資源資料庫上所出現的頻率反比為依據，計算其與查詢字的相關性高低，來挑選出與查詢字相關性高前 k 名的標籤出來，作為查詢結果代表標籤。

<2> 代表標籤概念架構建立方法

我們以人為給定有上下概念語意包含關係的標籤配對組合(e.g. "Country-USA"或是"Country-China")為訓練資料，根據個別標籤字在資料物件的多種出現特徵，利用 Rank-SVM 模型(Support Vector Machine for Ranking Model)學習判別語意概念高低排序模型。此外，我們同樣以人為給定具語意包含關係及不具語意包含關係的兩類標籤配對為訓練資料，根據標籤配對中兩個標籤在資料庫中出現情況所計算出的多種特徵，運用 SVM 模型(Support Vector Machine Model)學習出判斷兩個標籤是否有語意包含關係的分類模型。

所有查詢結果代表標籤字及其特徵將輸入所建立語意概念高低排序的模型來進行語意概念廣度的排序。依照其語意廣度排序結果的順序(概念愈廣的排在愈前面)一一加入概念架構，由兩個標籤間是否有語意包含關係的分類模型，判斷每

一個新加入概念架構的代表標籤可作為在概念架構中哪些標籤下的子概念，以"Top-down"之方式建立出階層式樹狀結構。

為評估本論文所提出方法的效果，我們利用公開且免費的 API 以網路爬蟲(web crawler)方式抓取在 Flickr 中使用者的分享照片及其所標註的標籤作為實驗資料物件。實驗分成兩部分，第一部分以系統化測試，評估所提出方法所建立概念標籤架構的不同層標籤用在查詢結果篩選的效果，我們以下列三項評估標準做為比較查詢篩選效果的優劣，分別是覆蓋率(coverage)、重複率(overlap)以及選擇性(selectivity)，並與相關研究方法所得結果進行比較。第二部分則是以問卷的方式評估階層式標籤架構的有效性(effectiveness)，讓使用者判斷系統推薦的代表標籤字與使用者給予的查詢字是否相關，且系統所建構標籤階層式架構中的上下包含關係是否具查詢條件特殊化(specialization)的語意。

1.4 論文架構

本論文以下章節內容簡介如下:第二章為相關文獻探討，第三章為本論文方法之系統架構與處理流程。第四章說明代表標籤字之挑選方式，第五章詳細描述我們如何建立查詢結果代表標籤的階層式架構，第六章以實驗結果評估本論文所提出方法的執行效果並加以分析討論，最後在第七章提出總結及未來研究方向。

第二章 文獻探討

近來有許多研究討論標籤的特性、相關的處理技術與應用。以下我們將依序介紹與本論文相關的研究，可分成社群標籤產生方式與呈現方法、以標籤輔助查詢之技術以及瀏覽式面向查詢技術。

2.1 社群標籤產生方式與應用

社群標籤產生方式主要可分作專家分類(Taxonomy)及大眾分類(Folksonomy)兩種。根據[1]提及到專家分類是由專業的人士來定義標籤資源分類的項目，使用者需要參考這些分類項目，以指定特定的類別去進行標籤標註、搜尋標籤字及其資料物件內容，因此此方式缺乏大眾所追求對於物件自由標註標籤之理念且專家與一般使用者對於一個字詞可能有多方解讀，所以容易發生無法符合使用者所需求的資料之情形；反觀當採用大眾分類則強調讓使用者按照自我觀感去對想要描述的物件進行下標籤的動作，對於編輯、產生、修改等行為相當容易。

在知名的社群網站平台(e.g. Flickr, del.icio.us)也是使用大眾分類規則來供使用者任意對於物件進行標註，所以現今考量的重點主要圍繞著如何有效地將這一些自由標註標籤的資源進行分析討論及應用，希望透過資料物件的標籤分析標籤語意，將一群的資料物件之摘要資訊顯示出來亦或是用以分類、群聚資料物件來

做處理。並在[2]提及到大眾分類法，一般會擁有三個重要的角色在，分別為資源(resources)、標籤(tags)、使用者(users)，用以做預測分析與討論。因此我們使用大眾分類(Folksonomy)方式所獲取的標籤資源中，其包含眾多使用者對於資源(資料物件)給予標籤標記的資訊來作為主要的研究對象。

由於大眾分類法的標籤產生方式比較自由，所以標籤可能在語意上，會有一詞多義的問行產生。所以在研究上，標籤雲(tag cloud)是常見的一種將一群社群標籤資源中的主題呈現出來之方式。舉例來說，當我們從社群標籤網站中取得一個標籤雲，會以不同的字型大小來顯示它的重要性，而這字體大小則是依據頻率出現的多寡來決定字體大小，也就是說出現頻率高→字體大、出現頻率低→字體小的視覺化效果，進而判斷對於每一個物件是否具有相關及擁有某種程度上的重要性存在，此方式是最直覺且容易實現的衡量基準。因此標籤雲可顯示出最主要的主題為何，使用者可透過選取來得到資料，至於該如何挑選出好的標籤呈現在標籤雲中是值得研究探討的。[4]將目標設定在如何對一個查詢所得到的結果集合中，挑選出適合當作摘要結果內容的標籤。作者訂定了許多評估一組標籤集合是否適合選為標籤雲的評分方法，包含考慮<**1**> 標籤集合中的標籤涵蓋之資料筆數、<**2**> 標籤集合中不同標籤涵蓋資料的重複程度、<**3**> 標籤集合中的凝聚力(Cohesiveness)–以計算標籤集合中包含這些標籤字的資料物件彼此的相似度值來表達該集合中的這些標籤字彼此關聯性的高低、<**4**> 在標籤集合中的標籤與原先所下查詢字的關聯性以及<**5**> 標籤集合中的標籤之普及度(Popularity)，也就是在

查詢字搜尋結果集合中的出現物件次數等特徵，作者再依不同評估標準提供對應的演算法，挑選出可幫助使用者了解查詢結果摘要的標籤集。

雖然在選取各項評估方法有多方面的考量，但是大多普遍常見的仍使用到標籤字之出現次數來當作一種重要特徵。為了與傳統方法以頻率排名方式做比較，因此[3]提出許多方法以達到標籤雲(tag cloud)查詢方式的效能增進。作者認為除了利用頻率的方式外，是否仍有其他方法可以有更佳的效果存在，因此作者舉出了一些策略來實施、比較，包括 **<1>** 對於出現頻率來挑選標籤，直接計算各個標籤出現在多少個資料物件中、 **<2>** TF-IDF 之分數-即統計特定的一個字詞在一個資料物件中出現的次數，倘若該字詞出現在該資料物件過於頻繁，則該字詞的重要性也相對降低的設計理念，以及 **<3>** 將標籤表示成圖形結構中的節點，再對圖形架構進行 random walk 計算出各節點所對應標籤的重要性，另外也考慮 **<4>** 標籤間的歧異性(diversity)和 **<5>** 標籤在物件之標籤列中的排名位置，標籤的排名愈前面，則分數高，反之分數低。以上所述的五種計算分數方式，來分別評估各種挑選標籤雲的方法運用在資料查詢、瀏覽、及群組推薦上的效果好壞，在實驗發現使用歧異性或是標籤出現在標籤列的排名位置相較於以頻率多寡為考量的方法，有所增進標籤雲查詢方式的效能。

2.2 以標籤階層式架構輔助查詢之技術

而我們認為需要組織起標籤字間的關係，因而用以輔助使用者查詢的採用方式為階層式標籤架構。

雖然標籤雲能夠顯示出標籤的被使用率多寡，能夠透過字體大小來區分出重要程度，但卻無法看出標籤彼此間的語意關係。標籤若能以樹狀的方式呈現並具備[階層關係]的特性，則可顯出標籤彼此間的語意包含關係，為廣泛或是較為狹義的標籤概念（concept）。舉例來說，Sport 可能涵蓋著國家籃球協會(nba)、國家橄欖球聯盟(nfl)…等多項運動比賽項目。我們認為若將標籤字間的關係組織起來，能有效地輔助使用者查詢進行查詢，因此我們以建立語意階層式標籤架構為主要研究目標。

在許多針對如何有效地輔助使用者下查詢字 (query)，找到所要的目標物件之研究上，絕大部分著重在於如何清楚地得知使用者搜尋的意圖，給予使用者明確的推薦字詞來縮小回傳結果的範圍。因而在查詢字推薦(query recommendation)的研究一般有兩種做法，(1)可利用從網頁蒐集資料，從網頁內容利用機率的分析找尋與查詢字相關的字詞[15][16]，或是(2)透過使用者之前使用過的關鍵字來當成搜尋紀錄(search logs)，可以知道那些查詢字是比較經常被使用，亦或是修改原來的查詢字並且而新查詢字來取代的關鍵字，如表 2.1 所示也可稱作擴展字(expand keyword)就能當成推薦字[17][18]。

表 2.1 搜尋紀錄之範例

Query Modification	**Pattern**
ladies song → ladies lyrics	song→ lyrics
university map → college map	university→ college
university map →university location	map→ location

根據上述的推薦的標籤字挑選方式外，因而衍伸出階層式架構之研究探討。[17]運用了上述的第二種類型(參考搜尋紀錄)以及[5]使用了上述兩種類型的結合(以搜尋紀錄為基礎而衍伸的機率模型) ，兩者皆是在找出推薦的標籤字後，將這些標籤字語意關係組織起來並建構出語意階層式架構。

[7]探討將標籤架構當作引導使用者瀏覽查詢標籤系統的階層式資料目錄是否有用，比較了不同的演算法建立的標籤架構。演算法有把資料物件視作圖形中的節點，計算每一節點彼此間的相似程度值來進行組織建立的作法。此外其他的做法也有運用了分群演算法進行標籤架構的建立方式。經過分析發現雖然標籤架構理論上可支援瀏覽查詢，但架構中同一層的子類別也必須要有限制，否則過多的子類別數並不容易讓使用者一一瀏覽及挑選，這一點若以使用者角度來看是相當直觀且必要的。因而如何建構出有效幫助使用者獲取具體需要的資料是相當重要的。

過去的研究中，對於語意階層式架構又可分成兩種組織架構的方式來幫助使用者達到查詢意圖，分別為<1>分析標籤字間的語意關係進行建立以及<2>將物件進行分群後，並從每一群的資料物件找出代表標籤字進行建立的作法，於以下兩小節作說明。

2.2.1 以標籤字間的語意關係之建立方式

[6]提出一個演算法，將標籤間的語意關係以一個樹狀結構的分類階層式架構顯示。其做法首先藉由概念廣泛程度之排名(ranking)來決定加入階層式架構的順序，因此該架構可以顯示出標籤的語意概念上下對應關係，愈上層的標籤表示其語意概念較廣，愈下層的標籤表示其語意概念較為狹義。而樹狀結構中兩個標籤是否存在著相連邊則以它們會被同時用來標記同一個物件的可能性來決定，因而作者提出了三項特徵，分別為標籤字出現頻率、涵蓋該標籤的資料物件個數以及標籤對於主題之亂度值(entropy)。當系統在建立階層式架構時，以每一次加入一個新的節點-標籤字 t_j，此時需要先計算與目前所建立階層式架構中的每一個標籤 t_i 的距離，距離 $d(t_i, t_j)$即為架構中標籤 t_i 到標籤 t_j所經過的權重邊值的加總。以圖 2.1 為例，假設左圖為目前已建立的階層式架構 THA，並同時附有各個邊的權重值-綜合以上三項特徵值得出。因此會先計算 t_1 到 t_2、t_1 到 t_3、t_2 到 t_3 兩兩的距離，$d(t_1, t_2)=0.3+0.4=0.7$、$d(t_1, t_3)=0.1$、$d(t_2, t_3)=0.4+0.3+0.1=0.8$。並且將所有的距離作加總(0.7+0.1+0.8)當作 THA 所花費的建立成本(THA 建立成本值

為 1.6)。右圖顯示當有一個新加入的標籤 t_4 時，所需作的考量。假設標籤 t_4 接在 THA 中的各標籤下的權重值為 0.2，則在依序的計算 t_4 與 t_i 的距離 $d(t_4, t_1)$、$d(t_4, t_2)$、$d(t_4, t_3)$，將這些值加總後加上左圖 THA 建立的成本值(1.6)，即可表示當標籤 t_4 加入 THA 時所需花費的建立成本。例如，當標籤 t_4 接在 t_1 底下，則 $d(t_4, t_1)$=0.2、$d(t_4, t_2)$ =0.9、$d(t_4, t_3)$ =0.5，t_4 加入 THA 時所需花費的建立成本共是 1.6+0.2+0.9+0.5=3.2。接著針對標籤 t_4 接在其他 t_i 的可能都分別計算，取出花費的建立階層式架構成本最低即為標籤 t_4 最適合擺放的位置。

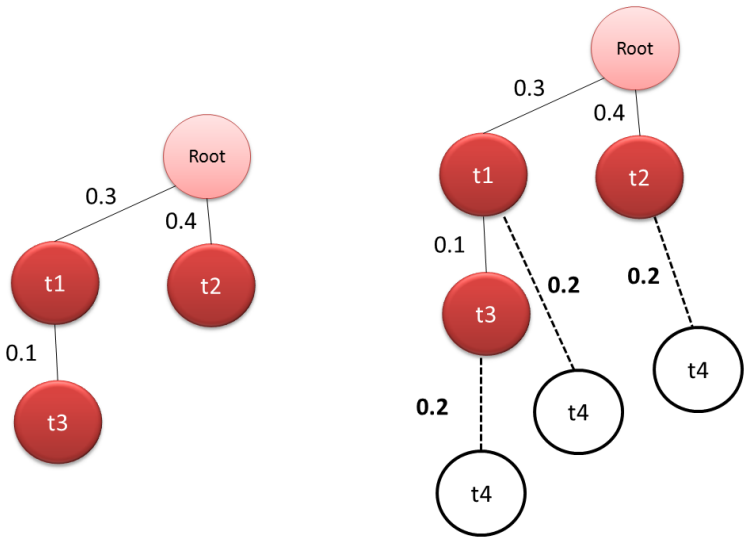


圖 2.1 概念階層式架構之範例

因此此篇所採用的方式為在處理過程中，根據以上的範例敘述，作者認為每一個標籤所取出的特徵值，其兩兩標籤的特徵值之差是作為判別語意概念關係之依據。當差值愈小則認為這兩標籤是有語意包含關係，因此每回合依照此依據將標籤加入到階層式架構中。同時也把偏頗樹(skew tree)的情況去除掉，接著將標籤架構的結果運用在標籤推薦，結合標籤語意概念層次、標籤分類的相似性以及

對應資料內容進行標籤推薦。不過作者所認定其特徵值之差的做法並非可以絕對顯示出概念廣度的包含關係，因此我們將在後續實驗部分，將本論文與此參考文獻的階層式架構之建立方法做比較。

在[5]主要是針對學校網域的資料進行探討，因為網頁內容變化性較低，所以採取以上所述的兩個方法-(1)網頁內容利用機率的分析找尋與查詢字相關的字詞，以及(2)透過使用者之前使用過的關鍵字來當成搜尋紀錄(search logs)，獲取可推薦字詞，預測出是否有歸納(Subsumption)關係在，所謂的歸納關係代表著一個標籤 t_j 為另一個標籤 t_i 的子概念，因此標籤 t_j 被歸納於標籤 t_i 之下。以兩兩標籤字組(t_i, t_j)判定具備歸納關係與否，所建立成階層化(hierarchy)的架構來進行分析。

2.2.2 階層式分群之建立方式

當我們想要對搜尋標籤資訊建立一個標籤架構，而方法的實現是可以根據外部資源(Wikipedia ,ODP 等)事前預設要找到的語意關聯來建立之。以將資料物件進行分群(Cluster)並從中挑選該群的代表標籤字之方法為例，可將資料物件中的標籤集合作語意上具有相似的概念的字詞進行比較，依照不同的概念產生出多個群聚。

而[12]開發出一項具個人化的搜尋引擎平台"ClusteringWiki"，強調結果以個人化之方式呈現，並以類似階層式目錄方式(將以一查詢字的回傳結果進行分群)。一般傳統上的搜尋引擎只列出相關文件，文字過度繁複使得使用者還得去判別每一個內容是否為使用者所需再進一步瀏覽，倘若我們可藉由分群後的結果將這些

結果做資訊篩選之動作(為了避免資訊過載的問題)，則可讓使用者更加明確地找到需要的資料。

此外，也有處理時間、地點、物體等語意類別之參考，物件可能被附上的標籤會有年份、月份、節慶的字眼出現，或是處理照片事件(大多在拍攝時是在描述當時發生的事件情況)之分類，則可運用外部資源(Wikipedia、ODP(Open Directory Project)、Wordnet…等)來進行標籤的組織架構[13][14]，為另一種階層式架構架立方式。

綜合上述對於輔助使用者查詢的方式簡介，不論是以比較兩兩標籤的語意關係建立階層式架構還是以資料物件進行分群，再將標籤對應的資料物件進行組織的作法上，皆是著重於如何建立出幫助使用者瀏覽查詢結果的階層式架構為最終目的。而目前本論文所使用的方式則是以建立起具備語意關係的階層式架構來輔助使用者檢索資料，並將[6]提出的階層式架構建立演算法作為比較對象。

2.3 瀏覽式面向查詢

階層式架構的重要性所在，可用於輔助使用者在對於查詢結果以類似動態的方式去挑選系統推薦的標籤字。而這種呈現方式即為面相搜尋的概念，因而在許多熱門拍賣購物網站平台，皆提供面向搜尋方式來讓消費者查看尋找購物網站中的商品，依照欲購買的商品的特徵屬性值清單快速地找到目標，而不需要一一瀏覽各筆商品。例如我們想要找尋一台電腦，而以遊戲、效能好為關鍵字做查詢，則系統會列出一些符合的產品清單,而同時也列出螢幕大小、品牌、影像卡、CPU、

作業系統...等多種面向(facet)特徵來供使用者進行篩選。不過大部分的面向查詢系統，都需要將面向特徵視為已知，也就是事前定義好一些特徵屬性，但如此一來容易產生無法滿足使用者不同篩選需求的問題。因此[9]作者想解決這些問題與限制，讓使用者對任一主題(例如商品類別、文件內容主題分類..等)能利用關鍵字更有彈性地描述每一個面向，希望能採用非監督式(unsupervised)的學習方法，使用機率的方式來擷取出多個面向的概觀，以達到能夠滿足使用者所需要的資訊內容(產品、文章..等)之目的。

[8]則想對關鍵字的查詢結果，動態選出使用者可能感興趣的屬性，並以像線上分析處理(OLAP)中執行瀏覽探勘的概念呈現給使用者。依照使用者給予預期想要達到的結果與系統給予的結果作比較，其會令使用者感到意外的程度高低，定義所謂的"興趣程度值"(interestingness)。作者提出了一個同時採用資料的文字內容及結構屬性來得到動態的多面向搜尋系統。

[10] 提出一個稱作 TEXplorer 的系統，將具備結構化的文件資料庫如產品評論意見資料視為一個多維度的文件資料庫，將關鍵字查詢進行文件排序之處理和 OLAP 的資料整合(aggregation)及資料探勘功能做統整。一般在購物網站上，使用者想找到欲購買的相關產品，會透過固定的清單選項(商品分類)來進行搜索，然而對於某些產品，以筆記型電腦為例，除了具有許多的屬性欄位(attribute): 品牌、CPU、RAM、螢幕大小等的規格，還伴隨著許多的商品評論(評價)。TEXplore 系統，會先依商品評論和關鍵字所算出的相關分數(分別針對所有的屬性欄位

(attribute)計算分數)提供可篩選出其商品評論和相關性高的資料物件之屬性及屬性值,藉由瀏覽引導的方式,階層式的幫助使用者找到和查詢相關的資料及文件。[11]則把個人化的概念加入面向查詢概念之中，作者提供一個可根據使用者的喜好自動挑選面向及面向屬性值於查詢介面。允許使用者對於查詢結果進行評分的動作。此方式可以幫助系統作為查詢結果之參考，進而回傳更貼近使用者所需要的面相查詢結果。

綜合上述對於面向查詢的研究參考資料，面向查詢這種搜尋方式不僅整合了文字搜尋跟結構化查詢，並且根據使用者所選擇的面向可以當作後續瀏覽內容的參考依據。因此提供查詢結果物件標籤的概念階層式架構供使用者選擇階層式架構中的代表標籤字，可提供使用者類似面相搜尋的方式逐漸減少搜尋回傳結果。

第三章 系統架構與流程

本論文系統架構主要分成三部分，第一個部分為包含查詢字之資料物件搜尋，第二個部分為選取代表標籤字，第三個部分則將代表標籤字先遵行語意概念廣度大小排名，並依排名順序加入建立出標籤概念階層式架構。

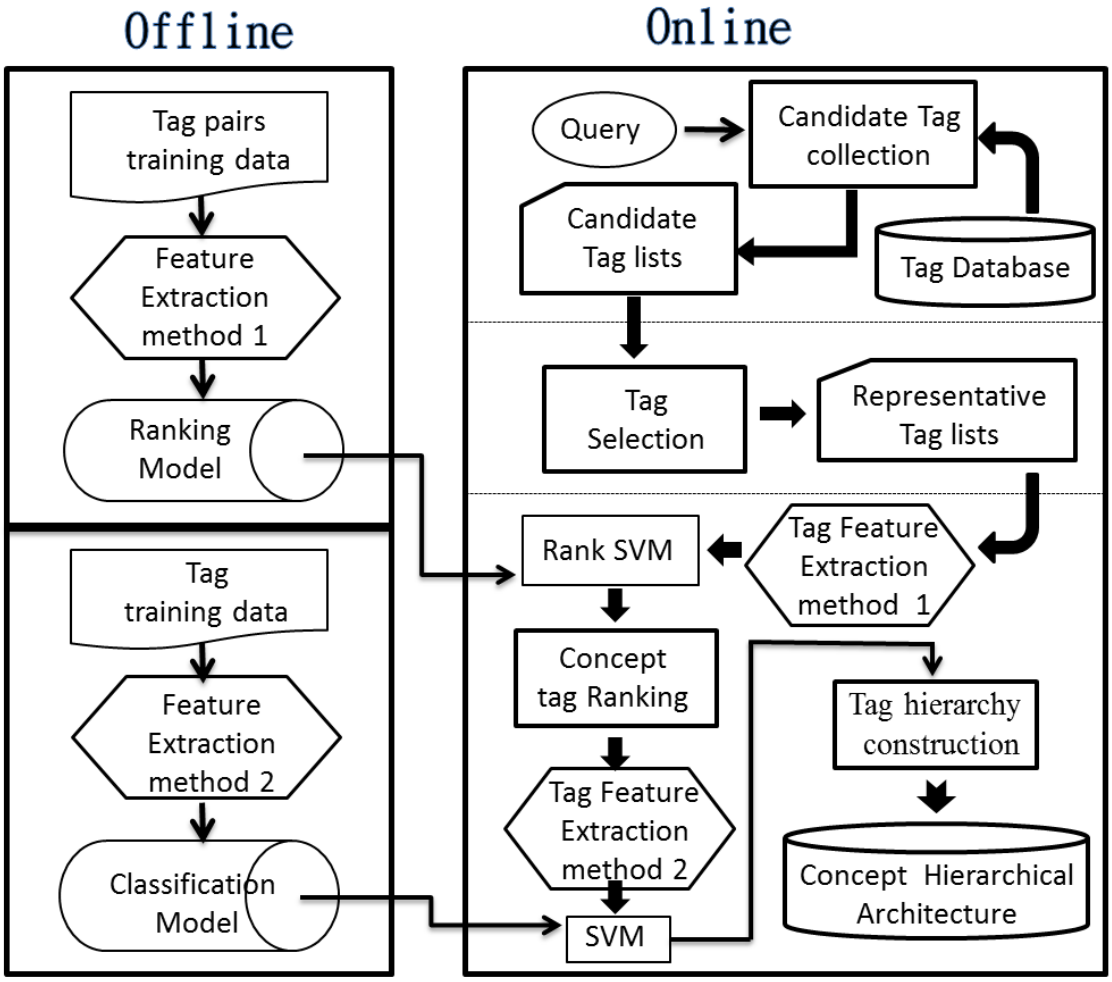


圖 3.1 系統架構圖

系統架構中分成線上處理(Online processing)和離線訓練(Offline training)兩大部份。如圖 3.1 所示。

線上處理可分為：蒐集候選標籤字 (Candidate tag collection)、挑選代表標籤字以及概念階層式架構之建立三大處理步驟。而第三步驟又可細分為標籤字概念廣度排名評估(Concept Tag Ranking)以及代表標籤字關係之建立(Tag Hierarchy Construction)，如圖 3.2 所示，以下將逐項說明。

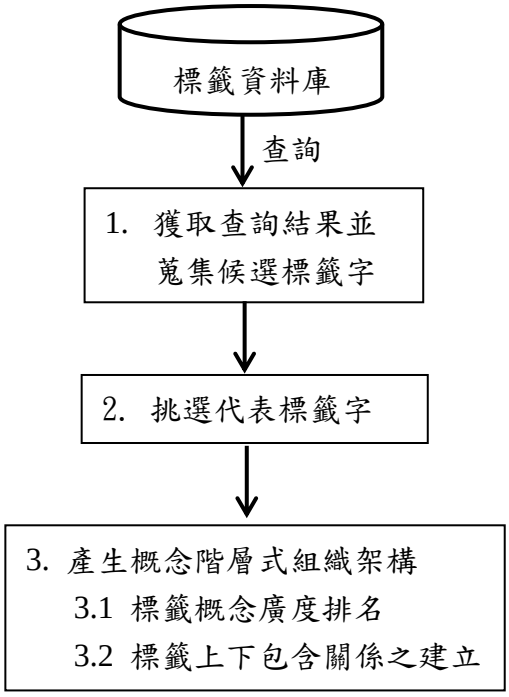


圖 3.2 系統線上處理流程圖

圖 3.2 所示為系統流程圖，以下將逐項說明。

1. 由於我們的目的在幫助使用者篩選查詢結果，因此必須先把涵蓋查詢字的資料物件挑選出來。蒐集這些查詢結果的標籤字形成候選標籤字，接著再做後續的篩選處理。
2. 在找出與查詢字一同出現的候選標籤字後，有些字可能是一些具不明確意涵的字，或是出現過於頻繁而非重要字。所以必須進行代表標籤字挑選的處理，

取出代表標籤字。

3. 經過上述步驟，蒐集到代表標籤字集合。標籤階層式架構初始為空，系統會將代表標籤字先進行語意概念廣度排序，再依序根據標籤和當時標籤階層式架構中已存的標籤是否有上下包含關係的判定，加入到階層式架構中的適當位置。

在步驟三中為了進行標籤概念廣度的評估以及上下包含關係之判定，因此需要在離線處理先訓練排名模型(ranking model)和分類模型(classification model)兩個處理單元來輔助線上處理階段之判斷。以下將對於此兩項工作進行說明:

(1)建立排名模型- 給定多組含有概念上下關係的標籤字對，並且對每一組標籤字取出特徵值後，運用 Rank-SVM 工具建立概念廣度排名模型。該建立模型將用來對代表標籤字進行語意概念廣度排序。

(2)建立分類模型- 給定具有語意上下關係和不具有語意上下關係的標籤字對，並取出其多項特徵為訓練資料，以 SVM 工具進行分類學習，產生用來判斷一組標籤間是否有概念上下關係的分類模型。而該模型將用來在建立標籤階層式架構時，判斷一個代表標籤字是否允許加入在另一個代表標籤字下。

第四章 代表標籤字之選取

本章將介紹如何對標籤查詢結果蒐集候選標籤字集合，以及代表標籤字選取的處理方法。

4.1 蒐集候選標籤字集合

首先，我們令 TDB 表示一個具標籤資源的資料庫，當中儲存許多資料物件(object)，每個資料物件 o 有一個對應的標籤集合，以 o.tagset 表示，一個標籤集合可為多個標籤字組合而成。例如表 4.1 所示，編號 464511629 的資料物件對應的標籤集合為{cat, kitty, dog, fight, battle}。

當使用者所下查詢 q 為{cat}時，以表 4.1 所示範例會找出 4 個物件，編號依序為{ 464511629, 31377556, 2528462725, 74682438 }，而這些物件編號我們定義為在 TDB 中涵蓋查詢字 q 的物件集合，以 O_q 表示。

O_q 中的物件之標籤集的聯集並去除 q 中的標籤形成一個候選標籤字集合，以 CT_q 表示，也就是 $CT_q = \bigcup_{o \in O_q} o.tagset - q$ 。

表 4.1 標籤資源範例

物件編號	標籤集合
464511629	cat kitty dog fight battle
172425284	dog battle chase
31377556	home cat pet dog
2528462725	baby ball cat kitty
74682438	cat pet kitty footprints

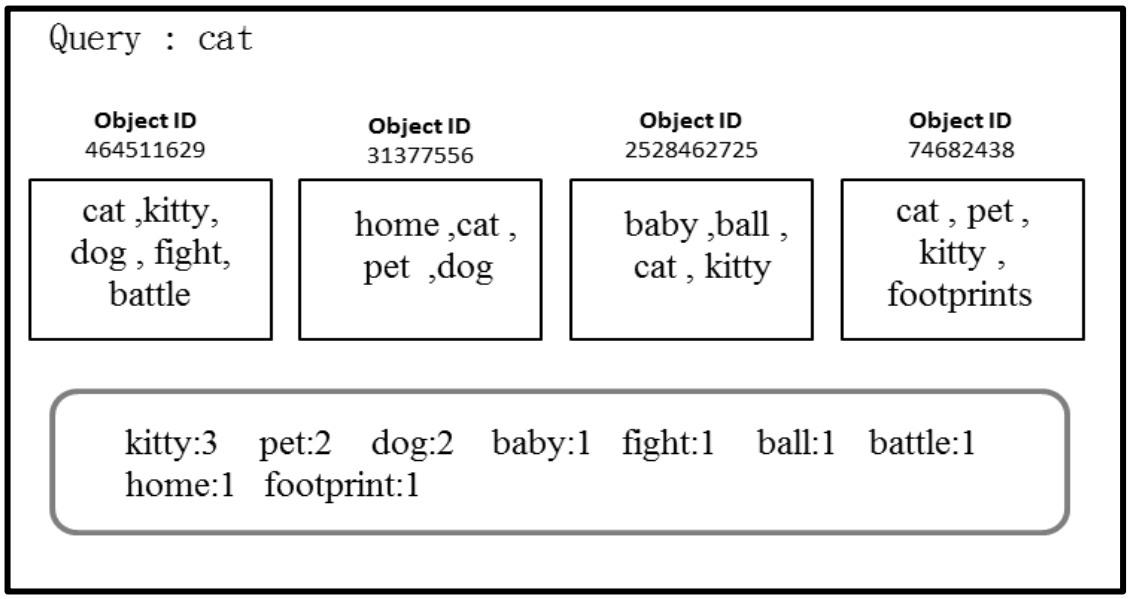


圖 4.1 標籤出現頻率範例

以查詢{cat}為例，從表 4.1 標籤資源範例中可以找到四個標籤集合包含{cat}的物件，分別為 Object 編號 464511629 , 31377556 , 2528462725 , 74682438 的物件。將這些物件的標籤集進行聯集，所得到的候選標籤字 CT_q 集合為{kitty、pet、dog、baby、fight、ball、battle、home、footprint}。

在這些候選標籤字當中可能存在一些錯別字或無意義的字，因此我們會統計各個候選標籤字在 O_q 出現的次數，並設定一個門檻值來做前處理的篩選動作。在此我們限定在 O_q 中出現小於 10 筆的標籤字，從候選標籤字 CT_q 集合中篩除。

4.2 代表標籤字挑選辦法

在資訊檢索方法中，一個字在一個文章中出現的次數，可用來反應其在該文章中的重要性，但若這個字在太多文章中皆出現，表示其語意特定性不高，又應降低其對該文章的重要性，也就是 TF-IDF 的設計概念。運用此概念，我們認為在 O_q 中出現較頻繁的標籤字彙表示其在 O_q 中的代表性較高，但這些頻繁出現在 O_q 中的標籤字，若於整個標籤資源資料庫(TDB)出現頻率也太高，就應該降低其在 O_q 中的代表性。因此我們設計一個計算候選標籤 t_i 在 O_q 中的代表性分數算式，如算式 1 所示。

$$r\_score(t_i, O_q, TDB) = p(t_i|q) * log_2(\frac{1}{p(t_i)}) \quad \text{(算式 1)}$$

其中 t_i 代表在 CT_q 中的一個候選標籤字。以圖 4.1 為例，共有 9 個候選標籤字。$p(t_i|q)$表示該候選標籤字 t_i 在 O_q 集合之物件的標籤集中出現的機率值，也就是 $p(t_i|q) = |\{o|o \in O_q \wedge t_i \in o.tagset\}| / |O_q|$。$p(t_i)$表示該候選標籤字 t_i 在 TDB 中出現在物件的標籤集中出現的機率值，也就是 $p(t_i) = |\{o|o \in TDB \wedge t_i \in o.tagset\}| / |TDB|$。我們在此階段從候選標籤字中挑選出 r_score 分數前 50 名的標籤字，稱為 O_q 的代表標籤，做為後續建立語意概念架構的標籤。

舉例來說，當使用者下了查詢{cat}，在 $O_{\{cat\}}$ 中找到的出現頻率高可能除了"

pet "、" kitty "等和 cat 相關常出現的標籤字，還可能出現如" 2005 "、"canon"等與主題無關而在標籤資源資料庫經常出現的標籤字。假設上述提及到的四個候選標籤字在 $O_{\{cat\}}$中的出現次數統計如下: 若" pet "在 $O_{\{cat\}}$中出現了十五次，我們定義為$freq(\{pet\}|\{cat\})$=15。因此$freq(\{kitty\}|\{cat\})$=20、$freq$({2005}|{cat})=30、$freq$({canon}|{cat})=30。包含查詢字{cat}的資料物件個數為 100，也就是$|O_{\{cat\}}|$=100。則在 O_q 集合中依序含有這四個候選標籤字的機率值如下:$p(\{pet\}|\{cat\})$=0.15、$p(\{kitty\}|\{cat\})$=0.2、p({2005}|{cat})=0.3、p({canon}|{cat})=0.3。而在 TDB 中四個候選標籤字的出現次數統計如下: $freq(pet)$=100、$freq(kitty)$=50、$freq$(2005)=1500、$freq(canon)$=1500。倘若 TDB 中共有 10,000 筆資料物件，也就是$|TDB|$ =10,000。則在 TDB 集合中依序含有這四個候選標籤字的比例值如下:$p(pet)$= 0.01、$p(kitty)$=0.005、p(2005)=0.15、p(canon)=0.15。套用至算式 1 我們可以得到這四個候選標籤字的 r_score 值，分別為 $r\_score(pet, O_{\{cat\}}, TDB)$ =0.3 、 $r\_score(kitty, O_{\{cat\}}, TDB)$ =0.4602 、 $r\_score(2005, O_{\{cat\}}, TDB)$= 0.24717 、$r\_score(canon, O_{\{cat\}}, TDB)$= 0.24717。因此計算出候選標籤字的重要程度順序為"kitty" > "pet" >"2005"和"canon"。

第五章 查詢結果標籤階層式架構之建立

本章將介紹本論文提出之標籤概念階層式架構建立的基本概念，再分別介紹如何對查詢結果的代表標籤字進行語意概念排名，及自動建立標籤概念階層式架構。

5.1 標籤概念階層式架構建立方法之概念敘述

經過第四章所述方法的處理找出代表標籤字後，本方法會將這些代表標籤字先進行概念上的廣度評估。圖 5.1 所示為示意範例來表達藉由概念廣度評估排序欲達到的目標。

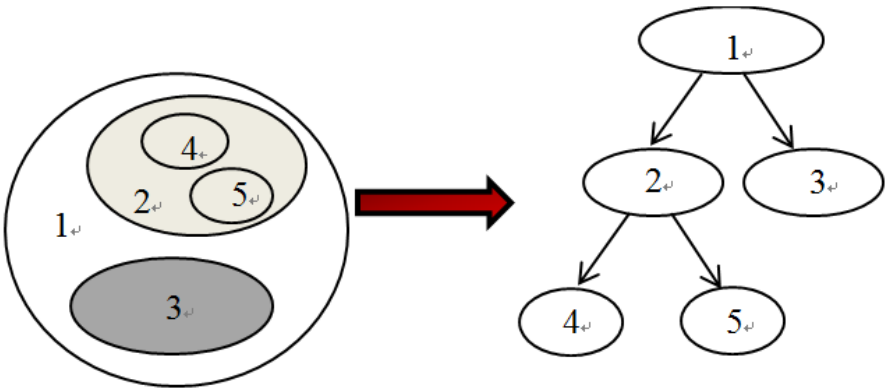


圖 5.1 概念之架構範例

如圖所示共有 5 個編號為 1~5 的標籤字，圖 5.1 左以面積大小及面積包含情況表示各標籤字的語意概念廣度及語意包含關係，若標籤字間有語意包含關係，則依其面積的包含關係，表示其語意廣度排序。因此在圖中可顯示(一) 1>2>4、(二) 1>2>5、(三) 1>3 ，因此當對應到以概念階層式架構顯示則如圖 5.1 右所示。舉例來說，以"apple"作為查詢，用人為判斷希望其查詢結果標籤應該組織成如圖 5.2

所示-為查詢結果標籤概念階層式架構所呈現的一部分結果。當中的"city"、"food"、"leopard"，是屬於較廣義的標籤字。而"fruit"、"newyork"、"bigapple"、"macbook"則屬於較為具體且明確的標籤字。因此如何建立出階層式架構具有概念上下包含關係的效果為主要目標。此外，建立的過程中對代表標籤字進行概念廣度排序的目的是要讓具有上下包含關係的標籤 t_a 和 t_b，若 t_a 的概念比 t_b 的概念廣，則在概念廣度排序結果中 t_a 必須出現在 t_b 之前。

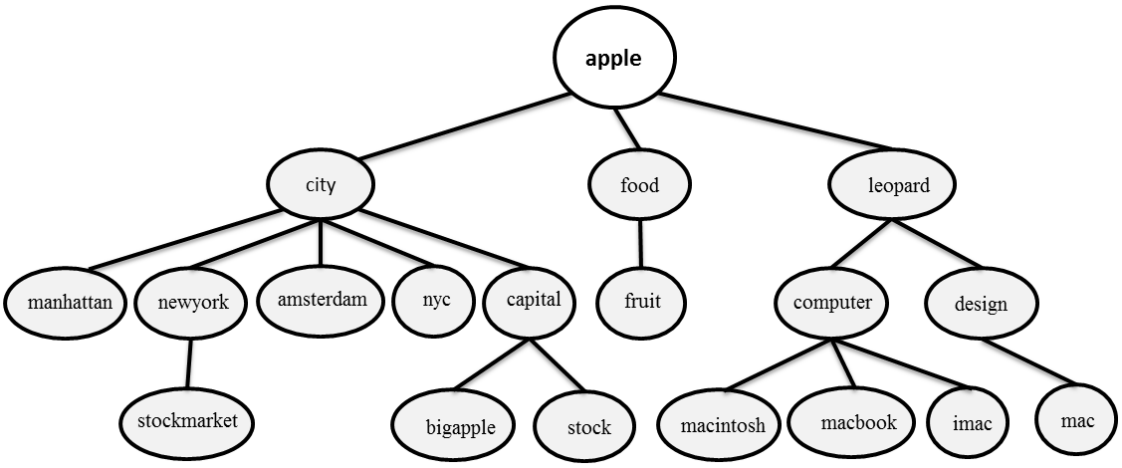


圖 5.2 查詢"apple"結果標籤字概念階層式架構的部分結果呈現

若能先找出代表標籤字的概念廣度排序結果，在建構概念階層式架構時，只需依序加入產生概念階層式架構中的節點，後面加入的代表標籤字便可限定只能放在前面已加入的代表標籤字節點底下。因此僅需判斷欲加入至樹狀架構的代表標籤字是否可成為已加入的代表標籤字節點下的概念字，而不需在已建立架構中和所有已存的代表標籤節點兩兩判斷語意關係並尋找可能插入位置。因此本方法在建立標籤概念階層式架構前先對代表標籤字進行概念廣度排序，之後再採用分類方法，決定一個代表標籤字是否具有為父節點與子節點的關係。

5.2 標籤字的語意概念廣度評估

我們對代表標籤字進行概念廣度預測之處理，是為了便於後續建立階層式結構可依照此排序更加有效率地將較廣概念之標籤放置在上層 level，而較低階的概念位於下層 level。以表 5.1 來看，共有 7 個與查詢字{apple}相關的標籤字，尚未使用 Rank-SVM 之代表標籤清單，及使用了 Ranking model 後之排序結果，分別如圖 5.3 左及右所示。可發現到在概念上較廣的概念詞{Food,Fruit,Color}會被排在前半段。

以下將一一介紹我們所使用作為概念排行判斷之依據的特徵:包括有標籤出現物件數(# of objects)、標籤字亂度(tag entropy)、及 Kullback-Leibler 分歧度(KL divergence)三種特徵。

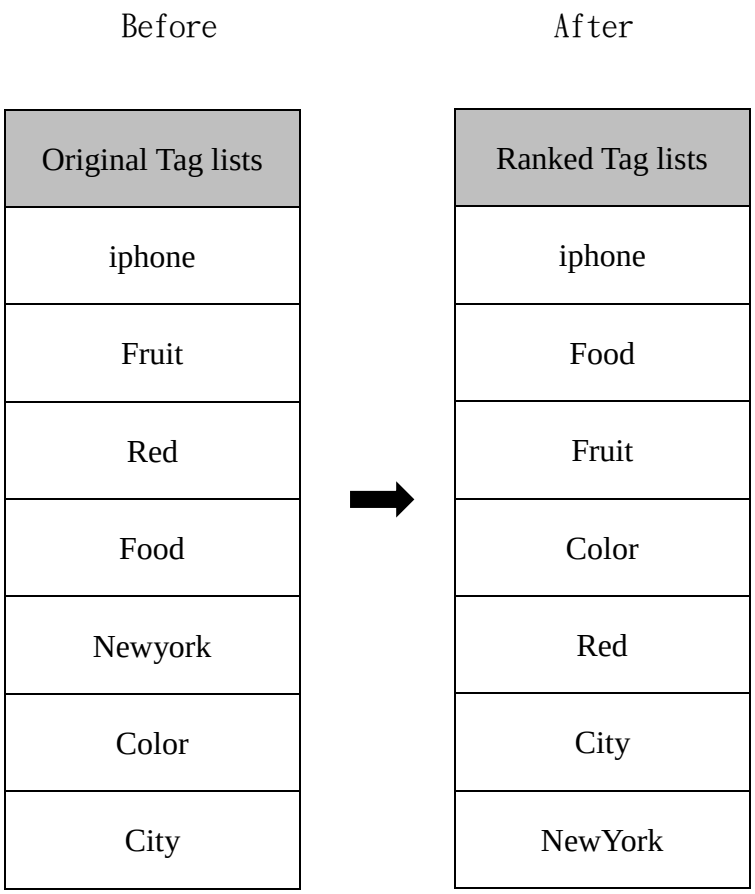


圖 5.3 排名模型之訓練資料

5.2.1 排序模型特徵擷取

<1>. 標籤出現之物件數

一個標籤字被標註在不同資料物件次數之多寡，通常可代表著該標籤字的語意廣泛程度。因此對一個代表標籤字 t 第一個特徵值 num_obj(t)的定義如下:

$$num\_obj(t) = |obj(t)| \qquad \text{（算式 2）}$$

$$obj(t) = \{o \mid o \in TDB \wedge t \in o.tagset\}$$

如算式 2 所示，$obj(t)$為一個包含代表標籤字 t 的資料物件集合，此特徵值表示有哪些資料物件 o 的標籤集合 o.tagset 包含 t 的資料物件個數。

<2>. 標籤字主題分佈亂度值

H(t) 表示一個標籤字 t 在 TDB 中不同主題分佈的亂度值。由於資料物件沒有明確主題，我們選定 TDB 中出現頻率前 100 高的標籤字(如表 5.1 所示)當作主題關鍵字。接著計算每一個代表標籤字 t 和這 100 個主題關鍵字中各個關鍵字同時出現的機率值分佈。以概念較廣的標籤字來說，和各主題關鍵字一起出現的機率值可能差不多，因此我們可得出該代表標籤字具有較高的亂度值。反之，若一個標籤字的概念較為具體明確，只和幾個特定的主題關鍵字同時出現的機率值較高，其代表標籤字在 TDB 中不同主題出現機率值分佈的亂度值會呈現較低的情形。

因此我們採用以算式 3 來計算代表標籤字 t 在 100 個主題關鍵字的分佈亂度值 H(t)，作為其第二個特徵值，定義如下:

$$\mathrm{H}(t): -\sum_{i=1}^{100} p\left(topic_i \mid t\right) \log p(topic_i \mid t) \text{ , } 1 \leqq i \leqq 100 \text{ (算式 3)}$$

$$p\left(topic_i \mid t\right): \frac{|obj(t,topic_i)|}{\sum_{i=1}^{100} |obj(t,topic_i)|} \quad \text{(算式 4)}$$

其中 $topic_i$ 表示第 i 個主題關鍵字，i 從 1 到 100。$p(topic_i|t)$表示標籤字 t 與第 i 個主題關鍵字同時出現之機率值。其算法是:分子為標籤字 t 與第 i 個主題標籤字 $topic_i$ 共同出現的物件數($|obj(t,topic_i)|$)，分母為與各主題標籤字同時出現物件數之總和($\sum_{i=1}^{100} |obj(t,topic_i)|$)，如算式 4 所示。

表 5.1 出現頻率前 100 個的標籤字

nature	people	orange	2006	france
sky	reflection	sun	uk	london
blue	sea	ocean	rock	house
water	art	winter	lake	specanimal
cloud	portrait	interestingness	wildlife	cute
nikon	night	blackandwhite	car	asia
canon	architecture	boat	pink	cat
light	yellow	bird	park	nyc
green	beach	urban	england	newyork
red	travel	snow	photoshop	grass
tree	city	geotagged	i500	wow
landscape	macro	beautiful	old	abandoned
bravo	girl	shadow	film	field
hdr	california	canada	woman	magicdonkey
color	street	window	italy	topv111
sunset	black	colour	photography	spring
explore	usa	photo	2008	mywinner
flower	mountain	europe	germany	japan
white	building	summer	topf25	sand
animal	2007	river	man	silhouette

‹3›. Kullback-Leibler 分歧度

我們認為出現特定代表標籤字 t 的資料物件是否影響各主題關鍵字之出現分佈情況，可以作為判斷 t 的語意廣泛程度的參考依據。對於每個代表標籤字 t，TDB 中的資料物件可分成: 1. 包含該標籤字 t 的資料物件集合以及 2. 未包含該標籤字 t 的資料物件集合，分別稱作 TDB 中 t 的包含集合(contain set)與非包含集合(not contain set)。以表 5.3 為例，假如 t 為"dog"，則可以找到涵蓋此標籤字的物件編號有{001,003,004}，稱為包含 dog 的資料物件集合；而物件編號{002,005}稱為非包含 dog 的資料物件集合。

KL 分歧度可用來衡量在相同事件中的兩個機率分佈之差異。因此我們將前述取得的 100 個主題關鍵字視為事件，評估包含一個代表標籤字 t 的資料物件集

合以及未包含代表標籤字 t 的資料物件集合之主題分佈差異。KL 值越高，這意味著包含該代表標籤字集合和未包含該代表標籤字集合的分佈狀況較不一致，此代表關鍵字 t 具備一定的分佈影響，可能在一個或一個以上的主題關鍵字上足以凸顯出整體的分佈情形之差異，所以在概念上較屬於為狹義且具體明確的標籤字。反之，KL 值越低，則代表著包含該代表標籤字集合和未包含該代表標籤字集合的分佈狀況是比較一致的，此特定的代表關鍵字對主題分佈上較沒有影響力，無法凸顯出各主題整體的分佈情形之差異，所以在概念上較屬於為廣義且較不具體明確的標籤字。

因此對一個代表標籤字 t 計算包含 t 和不包含 t 的資料物件集，評估是否出現 t 對各主題關鍵字之出現次數分佈差異來作為第三個特徵值 KL(t)。KL (t) 定義如以下算式 5:

$$KL(t)=\sum_{topic}\left(P(t_i,topic_j)*\log\frac{P(t_i,topic_j)}{Q(t_i,topic_j)}\right) \qquad \text{(算式 5)}$$

$$P(t_i,topic_j)=\frac{|obj(t_i,topic_j)|}{\sum_{j=1}^{100}|obj(t_i,topic_j)|} \qquad 1\leqq j\leqq 100 \qquad \text{（算式 6）}$$

$$Q(t_i,topic_j)=\frac{|obj(topic_j)-obj(t_i,topic_j)|}{\sum_{j=1}^{100}|obj(topic_j)-obj(t_i,topic_j)|} \qquad 1\leqq j\leqq 100 \quad \text{（算式 7）}$$

其 P 代表著包含特定代表標籤 t_i 的資料物件集合，而 Q 代表著未包含特定代表標籤 t_i 的資料物件集合。算式 6 的分子$|obj(t_i, topic_j)|$表示包含代表標籤字 t_i 的物件集合中含有特定主體關鍵字 $topic_j$ 的資料物件個數。算式 6 的分母則為 t_i 與這 100

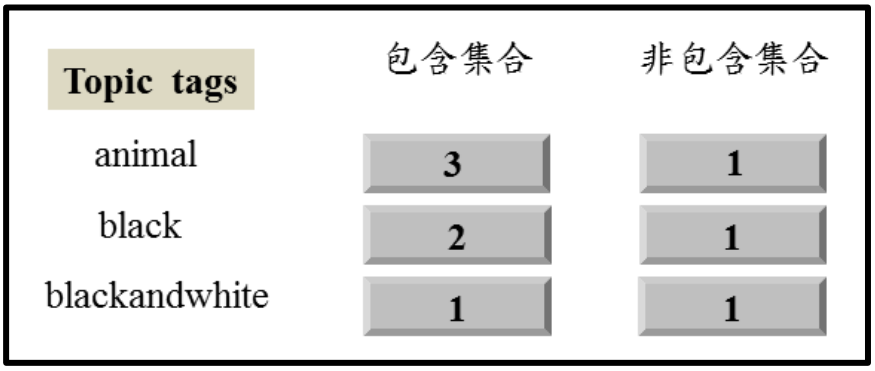
個主題關鍵字中各關鍵字同時出現的資料物件之數目總和。因此$P(t_i, topic_j)$代表標籤集合中出現*ti*的資料物件中會出現主題關鍵字$topic_j$的機率值。$Q(t_i, topic_j)$則表示未包含 *ti* 的資料物件中會出現主題關鍵字 $topic_j$ 的機率值。以下舉一範例來說明之 KL divergence 的特徵值計算。

表 5.2 資料物件及其對應的標籤集合之範例

物件編號	標籤集合
001	animal black dog
002	aniaml blackandwhite nikon
003	animal dog pet small
004	animal black blackandwhite dog pet
005	black pet picture

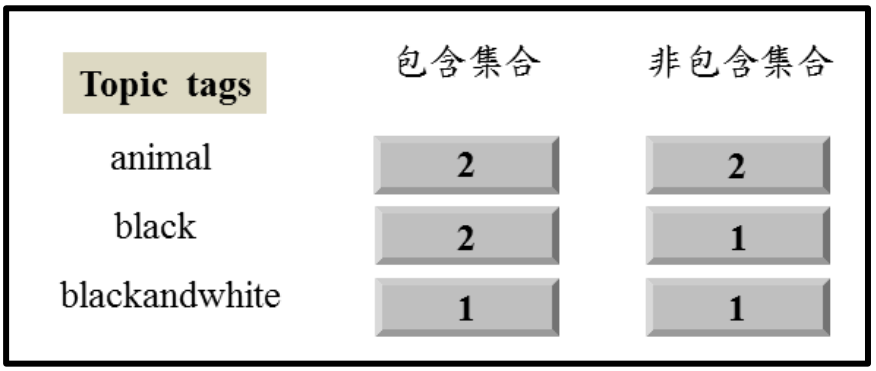
[**範例 5.1**]

如表 5.2 所呈現，假設"animal"、"black"、"blackandwhite"為主題關鍵字，則我們依序對特定標籤字"dog"及"pet"做 KL(dog)和 KL(pet)的計算。圖 5.4 中，顯示包含 dog 的資料物件集合中與非包含 dog 的資料物集合中各主題關鍵字的出現次數統計，圖 5.5 則顯示是否出現"pet"這個標籤字的各主題關鍵字出現次數統計。根據圖 5.4，計算 $KL(dog) = \frac{3}{6}log(\frac{3/6}{1/3}) + \frac{2}{6}log(\frac{2/6}{1/3}) + \frac{1}{6}log(\frac{1/6}{1/3})$ =0.03787。根據圖 5.5，計算 $KL(pet) = \frac{2}{5}log(\frac{2/5}{2/4}) + \frac{2}{5}log(\frac{2/5}{1/4}) + \frac{1}{5}log(\frac{1/5}{1/4})$ =0.0235。此例符合我們對較廣義的標籤字會得到較高 KL 特徵值的推測。



Topic tags	包含集合	非包含集合
animal	3	1
black	2	1
blackandwhite	1	1

圖 5.4 在標籤字"dog"對應兩集合與各主題關鍵字的相同物件數之統計



Topic tags	包含集合	非包含集合
animal	2	2
black	2	1
blackandwhite	1	1

圖 5.5 在標籤字"pet"對應兩集合與各主題關鍵字的相同物件數之統計

5.2.2 產生排名模型之訓練資料

為了對代表標籤字進行概念廣度排序，我們必須先離線進行排名模組的訓練，使用 Rank-SVM[1]來建立出概念廣度 Learning to Rank 的模型。Rank-SVM 的訓練資料為給定多組有概念上下包含關係的標籤字配對,訓練資料格式如圖 5.6 所示，其中包含多個標籤配對組合(tag pairs) ，最前面的數字 - 2:代表著該代表標籤字語意概念較高、1:代表著該代表標籤字語意概念較低，並給定每一個代表標籤字被計算出的特徵值。如此一來 Rank-SVM 可以學習每一組概念廣度的大小關係，例

[1] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

如圖 5.6 中 animal > deer、color > black。從中找出具有概念上下包含關係特徵值的規律性進行訓練。因此將訓練資料放入 Rank-SVM 工具中自動產生出排名模型。當取得 k 個代表標籤字時，排名模型會依照每個代表標籤字計算出的三項特徵值，將概念性較廣的標籤排名於其子概念標籤之前，依照此方式來進行概念廣度的整體大小排名。

```
# tag_pair 1 <animal,deer>
2 qid:1 1:0.0354174 2:0.3496462 3:0.223689
1 qid:1 1:0.006071  2:0.2522163 3:0.0287409

# tag_pair 2 <animal,elephant>
2 qid:2 1:0.0354174 2:0.3496462 3:0.223689
1 qid:2 1:0.0057483 2:0.2382628 3:0.0255497

# tag_pair 3 <animal,cow>
2 qid:3 1:0.0354174 2:0.3496462 3:0.223689
1 qid:3 1:0.0056297 2:0.2844673 3:0.0301872

# tag_pair 4 <color,black>
2 qid:4 1:0.0426196 2:0.5227198 3:0.4320452
1 qid:4 1:0.0274698 2:0.4662348 3:0.2562738

# tag_pair 5 <color,orange>
2 qid:5 1:0.0426196 2:0.5227198 3:0.4320452
1 qid:5 1:0.023483  2:0.4923399 3:0.2383827

# tag_pair 6 <color,pink>
2 qid:6 1:0.0426196 2:0.5227198 3:0.4320452
1 qid:6 1:0.0176123 2:0.4047292 3:0.1462712
```

圖 5.6 排名模型之訓練資料

我們使用 ODP(Open Directory Project)資源來建立訓練資料。ODP 是對於網頁內容連結的一個開放式目錄網站，是由來自多處的使用者依同維護與建置的社群網站。依照網頁的內容可分作 16 大項的類別概念，有 Art(藝術)、Business(商業)、Computers(電腦)、Games(遊戲)、Health(健康)、Home(家園)、Kids and Teens(兒童與青少年)、News(新聞)、Recreation(休閒娛樂)、Reference(參考資料)、Regional(區域)、Science(科學)、Shopping(購物)、Society(社會)、Sports(體

育運動)、World(世界)，如圖 5.7 所呈現的 ODP 首頁。我們從 TDB 的資料物件的標籤中，找出在 ODP 分類架構中，擁有概念上下包含關係的字。如果其中一個字詞在 ODP 中被歸屬於另一字詞的子概念中，則作為語意概念上下關係的標籤字對訓練資料。我們以此方式建立了 163 組標籤字配對並算出每個標籤字的三個特徵值作為 Rank-SVM 的訓練資料。

圖 5.7 開放式目錄網站 ODP - 16 項類別概念之呈現

5.2.3 代表標籤字概念廣泛程度排名之處理流程

根據輸入的每一個代表標籤字，先取得 5.2.1 小節所述的三項特徵，再經由事先產生的訓料資料放入 Rank-SVM 中自動訓練出的排名模型，最終產生依語意廣泛程度的代表標籤字由廣度大到廣度小排序結果。

5.3 階層式結構之建立

對於 5.2 節處理後依概念廣度排序之代表標籤結果，接下來我們會依序進行插入建立概念階層式架構處理。在將標籤 t_j 加入概念階層式架構時，必須檢查現有概念階層式架構中有那些節點對應的標籤 t_i 與標籤 t_j 具有語意上下包含關係。我們將採用分類方法來判別標籤 t_j 是否為標籤 t_i 語意概念下的概念。

對於一個要新加入概念階層式架構的代表標籤字 t_j 與已經於語意階層式架構中的每一節點對應之標籤 t_i，判斷 t_j 是否為 t_i 語意子概念，我們將此問題視為一個分類問題，系統先取出能表示兩個標籤字出現關聯的特徵值，我們再採用 SVM 模型進行分類。以下三小節我們將分別介紹我們以何種特徵訓練出分類模型、訓練資料產生方式，以及建立標籤階層式架構之處理步驟。

5.3.1 分類模型特徵擷取

在分類模型中，我們共採用了三大類特徵來判斷兩標籤字間是否擁有語意包含關係。可分為：共同出現關聯性(Co_Occurrence)、相對出現頻率(Relative Frequency)、以及出現互斥相關性 (Exclusive_Occurrence)。

<1>. 共同出現關聯性

此特徵以兩個標籤字t_1和t_2在 TDB 中出現在相同一個物件的個數，計算相對於t_1和t_2個別出現物件的總數的比例值，顯示兩個標籤的共同出現關聯性。當包含標籤 t_1 的資料物件集合和包含標籤 t_2 的資料物件集合完全相同時，表示兩者的共同出現關聯程度值為最高。因此第一項分類特徵$Co_Occurence(t_1, t_2)$定義如下:

$$Co\_Occurence(t_1, t_2) = \frac{2 * |Obj(t_1) \cap Obj(t_2)|}{|Obj(t_1)| + |Obj(t_2)|} \quad \text{(算式 7)}$$

其中 *Obj(t_i)*表示以 t_i 做查詢字找到的資料物件集合，*i* 為 1 或 2。其值域介於 0 到 1 之間。

［**範例 5.2**］

假設兩個標籤字詞 t_1 與 t_2 分別為"pet"與"dog"，並且各自當作查詢字於標籤資源資料庫 TDB 中搜尋，並從系統所回傳的資料物件中統計包含"pet"的物件是否包含"dog"，以及包含"dog"的物件是否包含"pet"，其結果如表 5.3 所示。從中可得知包含"pet"的資料物件個數為 10，包含"dog"的資料物件個數為 8，同時包含"dog"和"pet"的資料物件個數為 6，代入算式 7可得這組標籤字的第一項特徵值:

$Co\_Occurence(\text{"dog"}, \text{"pet"}) = \frac{2*6}{10+8} = 0.67$ 。

表 5.3 包含標籤字 dog 及包含 pet 資料物件的個數統計

tag_2 (pet) \ tag_1 (dog)	包含 dog 的資料物件	不包含 dog 的資料物件	物件總計
包含 pet 的資料物件	**6**	**4**	**10**
不包含 pet 的資料物件	**2**	**8**	**10**
物件總計	**8**	**12**	

<2>. 相對出現頻率

給定兩個標籤字 t_1 和 t_2，若 t_1 和 t_2 間具有語意包含的關係，且標籤 t_1 的概念較標籤 t_2 廣，則在含有標籤 t_2 的資料物件中可能同時含有標籤 t_1。但是在含有標籤 t_1 的資料物件中卻未必含有標籤 t_2。因此我們將以上兩種發生情形的相對出現頻率(relative frequency)考慮進來，作為第二項和第三項分類特徵以便用於在判斷語意包含關係上，如下列算式 8 及算式 9。

$$rel\_freq(t_2|t_1) = \frac{|Obj(t_1) \cap Obj(t_2)|}{|Obj(t_1)|} \qquad \text{(算式 8)}$$

$$rel\_freq(t_1|t_2) = \frac{|Obj(t_2) \cap Obj(t_1)|}{|Obj(t_2)|} \qquad \text{(算式 9)}$$

由於 t_1 和 t_2 的順序不同會造成相對出現頻率會有所不同，因此我們會計算在包含標籤 t_1 的資料物件中含有標籤 t_2 的條件機率值，以及在資料物件包含標籤 t_2 的情況下含有標籤 t_1 的條件機率值。若 t_1 為 t_2 的上層概念，其$rel_freq(t_1|t_2)$應

具有較高值，表示標籤 t_1 會常出現於在包含標籤 t_2 的資料物件中。但是 $rel_freq(t_2|t_1)$會得到較小值，表示因為 t_1 的概念較廣，因此具有標籤 t_1 的資料物件中不一定會含有標籤 t_2。

如範例 5.2 為例，以標籤 t_1 和標籤 t_2 分別為"pet"和"dog"來說，通常我們可以在含有 dog 的資料物件中，裏頭也同時含有 pet 這個標籤字眼。而含有 pet 的資料物件中，因為有多種可能的寵物，因此未必會含有 dog 這個標籤字眼。範例 5.2 中包含"pet"的資料物件個數為 10 而包含"dog"的資料物件個數為 8 且同時包含"dog"和"pet"的資料物件個數為 6，代入算式 8 和 9 可得這兩個標籤字的第二及三項特徵值如下:

$rel\_freq(\text{"dog"}|\text{"pet"}) = \frac{6}{6+4} = 0.6$ 及 $rel\_freq(\text{"pet"}|\text{"dog"}) = \frac{6}{6+2} = 0.75$。

此外，為了同時考慮 t_1 和 t_2 的相互包含情況，我們把第二項和第三項特徵值之差值來作為第四項分類特徵值，第四項分類特徵$Rel_freq_difference(t_1, t_2)$定義如下:

$$Rel\_freq\_difference(t_1, t_2) = \ Abs(rel\_freq(t_1|t_2) - rel\_freq(t_2|t_1))$$

(算式 10)

Abs 表示將其值取絕對值。差距較大的，可視為這一組代表標籤字對< t_1, t_2>可能具有語意包含的關係，因為當包含標籤 t_2 的資料物件中含有標籤 t_1 的機率遠大於包含標籤 t_1 的資料物件中含有標籤 t_2 的機率，則可得知標籤 t_2 很可能為標籤 t_1 的子概念。而差距較小的，則意味著包含標籤 t_2 的資料物件中含有標籤

t_1 的機率與包含標籤 t_1 的資料物件中含有標籤 t_2 的機率差不多，則標籤 t_2 未必為標籤 t_1 的子概念。

以範例 5.2 為例，當給定標籤 t_1 為"pet"和標籤 t_2 為"dog"，經系統計算出分類特徵二和分類特徵三後，代入算式 10 可得這兩個標籤字的第四項特徵值:

$Rel\_freq\_difference$("pet","dog") = | 0.75 - 0.6 | = 0.15

<3>. 出現互斥相關性

此項特徵考慮在含有標籤 t_1 的資料物件中沒有出現標籤 t_2 之物件數，相較於含有標籤 t_2 的資料物件中沒有出現標籤 t_1 之物件數的比例值，作為判斷 t_1 和 t_2 之語意包含關係的分類特徵。因此第五項分類特徵Exclusive_percent(t_1, t_2)定義如下:

$$\text{Exclusive\_Occurrence}(t_1, t_2) = \frac{|\text{obj}(\neg t_2 | t_1)| + \alpha}{|\text{obj}(\neg t_1 | t_2)| + \alpha} \qquad \text{(算式 11)}$$

obj$(\neg t_2 | t_1)$表示 TDB 中包含標籤 t_1 但不包含標籤 t_2 的資料物件集合，obj$(\neg t_1 | t_2)$表示 TDB 中包含標籤 t_2 但不包含標籤 t_1 的資料物件集合。當標籤 t_2 為標籤 t_1 的子概念時，包含 t_1 的資料物件數應該比包含 t_2 的資料物件數多，且因為出現 t_1 的物件未必出現 t_2，|obj$(\neg t_2 | t_1)$|有可能大。但出現 t_2 的物件通常要有 t_1，|obj$(\neg t_1 | t_2)$|應該小，所以其所得互斥相關性數值應該為大。為了避免分母|obj$(\neg t_1 | t_2)$|有數值為 0 的情形，所以在分子及分母各加上一常數α，我們將其以設定為 1。

［**範例 5.3**］

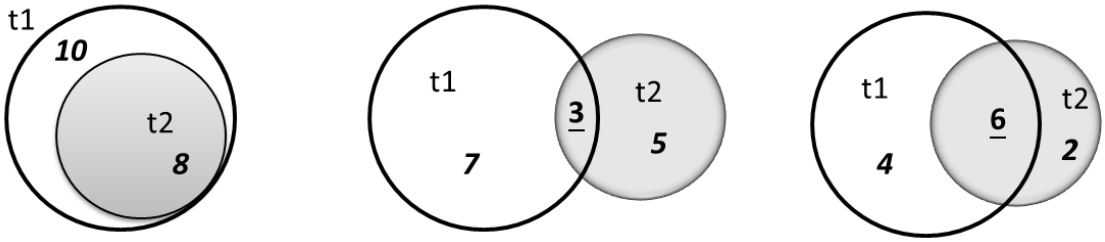
假設代表標籤字 t_1 及 t_2 在 TDB 出現物件數，分別為 10 個與 8 個。考慮包含 t_1 和 t_2 的資料物件的三種可能情況，如圖 5.8 中的(一)、(二)、(三)。以下我們針對三種情形套用至我們的公式 11 作計算:

圖(一): Exclusive_percent(t_1, t_2) = $\frac{2+1}{0+1}$ = 3。

圖(二): Exclusive_percent(t_1, t_2) = $\frac{7+1}{5+1}$ = 1.33

圖(三): Exclusive_percent(t_1, t_2) = $\frac{4+1}{2+1}$ = 1.66

在圖(一)中相較於圖(二)及圖(三)可顯示此兩代表標籤的資料物件集合具有包含關係。而圖(二)中兩者最不具有包含關係。圖(三)中 t_2 雖然未被 t_1 完全包含，但 t_2 仍有大部分被 t_1 所包含，此例顯示以上所算出 Exclusive_percent(t_1, t_2)能顯出兩個標籤 t_1 及 t_2 對應出現物件集合的包含關係程度。



（一）　　（二）　　（三）

圖 5.8 標籤字 t_1、t_2 出現互斥相關性之特徵範例

綜合上述所介紹的特徵，對於代表標籤字 t_2 是否可加入目前標籤階層式架構的標籤 t_1 下時，系統會計算出 5 種特徵值，用以作為後續的標籤字間是否包含關係之分類判斷。以下表 5.5 所示為我們採用的分類模型之特徵整理。

表 5.4 分類模型之特徵清單

特徵編號	特徵名稱	特徵函式
1	共同出現關聯程度	$Co\_Occurence(t_1, t_2)$
2	相對出現頻率($t_1 \rightarrow t_2$)	$rel\_freq(t_2 \mid t_1)$
3	相對出現頻率($t_2 \rightarrow t_1$)	$rel\_freq(t_1 \mid t_2)$
4	相對出現頻率差	$Rel\_freq\_difference(t_1, t_2)$
5	出現互斥相關性	Exclusive_Occurrence(t_1, t_2)

5.3.2 分類模型之訓練資料

給定多組有語意包含關係的標籤字配對，訓練資料的格式輸入內容如圖 5.9 所示，最前頭的數字 1 代表有語意包含關係、2 代表沒有語意包含關係，並給定標籤字配對被計算出的五個特徵值進行學習訓練，輸入 SVM[2]工具自動產生出分類模型(Classification model)。

關於分類模型的訓練資料，我們採用同 5.2.1 小節中描述的 ODP 為依據。從 TDB 的資料物件之標籤中選取了 131 組的標籤字配對作為訓練資料，其中具有語意上下包含關係的配對有 78 組，而無上下包含關係的配對則有 58 組。比照蒐集到的資料集合，依照 ODP 網頁所對應的概念分類方式，從具有語意上下包含關係的字且在 TDB 內可以找到該標籤的情形下，挑選為用以訓練的包含關係標籤字對，若不存在語意上下包含關係的字則作為非包含關係標籤字對。

```
city_london
1 1:0.086956 2:0.071428 3:0.111111 4:0.039682 5:1.60606

city_nyc
1 1:0.069767 2:0.053571 3:0.1      4:0.046428 5:1.928571

city_tokyo
1 1:0.028571 2:0.017857 3:0.071428 4:0.053571 5:4.0

sky_water
2 1:0.15     2:0.146853 3:0.153284 4:0.006431 5:1.051282

sun_ocean
2 1:0.117647 2:0.133333 3:0.105263 4:0.02807  5:1.3
```

圖 5.9 分類模型之訓練資料

[2] http://ntu.csie.org/~piaip/svm/svm_tutorial.html

5.3.3 標籤字間包含關係之建立

假設根據排名模型排序的代表標籤字結果清單為<t_1 ,t_2 ,t_3 , ……, t_n>，系統會依照這個順序，先取代表標籤字 t_1 開始建立節點(node)。接著取代表標籤字 t_2 代表標籤字運用分類模型判斷是否可接在 t_1 標籤字底下而成為子點(child node)，接著再取出代表標籤字 t_3 並考慮 t_3 運用分類模型判斷是否可接在 t_1 或 t_2 標籤字底下而成為子節點，後續的代表標籤字同樣依此方式處理。

以下詳述本系統建立階層式樹狀架構的處理流程。給定已經過語意概念廣度排序過的代表標籤字清單 *RTL*=<t_1 ,t_2 ,t_3 , ……, t_n>及分類模型 *CM*，系統將根據每一代表標籤字t_i，依照下述步驟進行概念階層式標籤架構的建立。建立樹狀架構之完整演算法如下演算法1。演算法中Line 2到Line 3為初始值設定，Line 4到Line 12則為建立概念階層式架構之步驟流程。

步驟<1>: 從已進行過語意概念廣度排序的代表標籤字清單，取得t_1的標籤字(Line2)，將其建立一個節點存於$G_{hierarchy}$中的root節點下(Line 3)

步驟<2>: 依序讀取代表標籤字t_i ($2 \leqq i \leqq n$)，進行在標籤階層式架構中建置節點之動作(Line 4)，直到已將t_n 加入於$G_{hierarchy}$中為止。

步驟<2.1>:

比對目前已建置於$G_{hierarchy}$中各節點對應的代表標籤字t_j (Line 5), 利用分類模型一次放入一組代表標籤字對(t_i , t_j)，判斷此兩個代表標籤字間有無語意包含關係。

步驟<2.2>:

若分類模型判斷該組代表標籤字對具有語意包含關係(Line 6)，則對代表標籤t_i產生一個節點，並建立父節點t_j 與子節點t_i 的關係鏈結(Line 7)。否則，額外建立一個獨立節點於$G_{hierarchy}$中的root節點之下(Line 9)。

步驟<3>: 輸出 $G_{hierarchy}$ 為概念階層式標籤架構 (Line13)。

Algorithm 1 Build the relation between tags

Input: ranked tag list *RTL* , classification model *CM*.

BEGIN
  t_1=FetchFirst(*RTL*);
  $G_{hierarchy}$. addChild (root,t_1);
    FOR EACH tag t_i in RTL-{t_1} **DO**
      FOR EACH tag t_j in $G_{hierarchy}$ **DO**
        IF *CM*(t_i , t_j)=True **THEN**
          $G_{hierarchy}$.addChild(t_j , t_i);
        ELSE
          $G_{hierarchy}$. addChild (root,t_i);
        END IF
        END FOR
    END FOR
  OUTPUT concept hierarchy tag architecture $G_{hierarchy}$;
END

演算法 1 建立標籤字之關係

［**範例 5.4**］

圖 5.9 顯示以本方法建立概念階層式架構之流程，在 5.2 節中圖 5.3 之右側顯示以概念廣度排名模型對代表標籤字排序之結果。排序後的代表標籤字< t_1 ,t_2 ,t_3 , ……, t_7 >依序為: "iphone"、"food"、"fruit"、"color"、"red"、"city"、"newyork"。一開始先插入標籤字"iphone"，接著系統取出"food"並判斷"food"可否接在"iphone"之下，虛線代表可考慮插入的節點位置，如圖中編號(2)所示。若分類模型判定"food"為"iphone"的子概念則"food"插入於"iphone"之下，否則就將"food"放置在 root 節點之下。倘若此時已建立好的概念階層式架構為圖中編號(3)，接下來系統再取出"fruit"並且運用分類模型判定"fruit"是否為"iphone"或"food"底下的子概念，

若分類結果為【是】則" fruit "插入於"iphone"或"food"之下，若【不是】則放置在 root 節點之下，如圖中編號(4)所示為可能插入的節點位置。而後續的標籤字依照前面的標籤字之處理方式，依序加入概念階層式架構中，直到所有排名清單中的標籤字都加入到概念階層式架構中為止。

Rank	1	2	3	4	5	6	7
	iphone	food	fruit	color	red	city	newyork

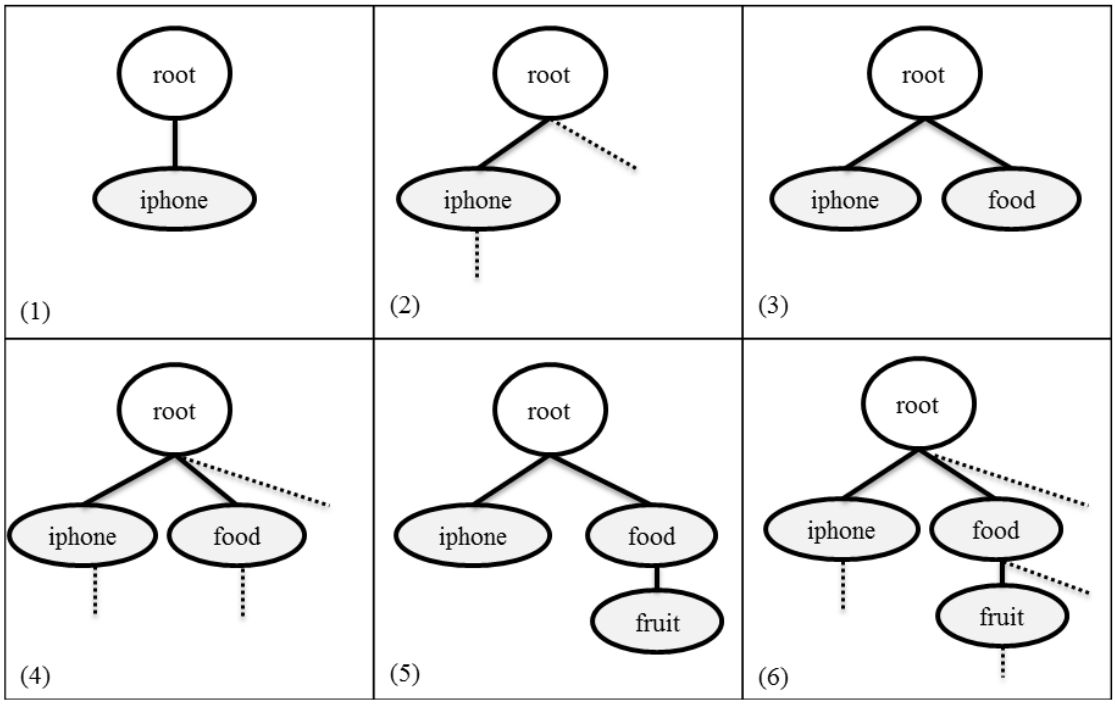


圖 5.10 階層式樹狀結構建立流程示意圖

第六章 實驗結果與討論

本論文實驗分成兩部份進行，第一部分為評估挑選代表標籤的步驟對於所建立標籤架構提供查詢瀏覽的效果，第二部分則評估階層式標籤架構的有效性(effectiveness)。以下將詳細介紹實驗資料來源以及環境設定、實驗測試資料、各部分實驗方法及實驗評估結果。

6.1 實驗資料來源及環境設定

6.1.1 實驗資料來源

本論文以新加坡國立大學 T.-S.Chua 及 J.Tang 等人從社群網站 Flickr 中蒐集而來的標籤資源集合[3]為實驗資料集。該資料集根據專家所制定的 81 個概念，從超過5,000名使用者所分享的相片中取得269,648張的標籤資訊，當中共有425,059個不同的標籤字。

[3] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

6.1.2 資料前處理

由於使用者對於資料物件進行標註時所使用的標籤字，其單複數標籤字所意指的事物是相同的。因此，本論文採用的是波特原型化演算法(Porter Stemming Algorithm)，先對標籤字進行單複數原型化處理。

6.1.3 實驗環境設定

本實驗實作所使用的環境為執行 Window 7 作業系統的個人電腦，硬體配備為 Intel Core i7 處理器及 8G DDR3 記憶體。實作所採用的程式語言為 JAVA 程式語言，並於 NetBeans 開發環境中進行編譯。

6.2 評估查詢結果標籤階層式架構之效果

此部分的實驗將評估挑選代表標籤的步驟對於所建立標籤階層式架構提供查詢瀏覽的效果。此外，我們也將本論文所提出的 CTC (Classification-based Tag hierarchy Construction method)演算法建立的概念階層式架構與[6] Y. Song 提出 MDL 演算法所得結果進行比較。這兩部分評估將分別在**[實驗 1.1]**及**[實驗 1.2]**中進行。

6.2.1 系統測試資料

本論文採用電腦隨機挑選單一個標籤字作為測試查詢，我們在挑選查詢字時選定在標籤資源集合 TDB 中出現頻率大於 50 者，以確保其查詢所得資料物件數達一定數量以上。

為考慮不同出現頻率標籤字之查詢所得資料物件數量不同，有可能影響標籤概念階層建構效果,因此我們首先計算標籤資源集合 TDB 中各標籤字出現頻率，將其標籤依照出現的頻率高低訂定三種範圍，分別為"高頻率標籤字 (出現頻率範圍為>1000)"、"中頻率標籤字(出現頻率範圍為 150~1000)"以及"低頻率標籤字(出現頻率範圍為 50~150)"。並統計各個出現頻率範圍對應的標籤字個數，如表 6.1 所示。再依照此三種出現頻率範圍的標籤字個數總數比例約 1:4:8 之比例，在個頻率範圍的標籤字各取出 30、120、及 250 個標籤字作為測試查詢字，總數為 400 個測試查詢字來進行實驗評估。

表 6.1 三種出現頻率範圍的標籤字個數統計

	出現頻率範圍	此範圍內的標籤字個數
高頻率標籤字	**>1000**	**788**
中頻率標籤字	**150~1000**	**3269**
低頻率標籤字	**50~150**	**6333**

6.2.2 實驗評估方法

系統為一個查詢 q 所找到的資料物件集合 O_q，其建立好的標籤概念階層式架構，評估其用於進一步查詢 O_q 中的物件之查詢效果，可包括以下幾個考量:

(1) 覆蓋率-是否可涵蓋 O_q 中的大部分資料物件

(2) 重複程度-採用不同標籤篩選得到的資料物件是否不同

(3) 選擇性-用來找到特定一個資料物件的標籤是否可有效篩除其他大量資料物件

我們認為一個好的階層式架構之建立，其代表標籤集應該覆蓋率高、重複程度低、選擇性高。

<1> 覆蓋率

覆蓋率是要評估運用概念階層式架構對 O_q 進行查詢，階層式架構的每一層有多少比例的資料可以被找出來。因而我們將針對第一層的代表標籤字集合 L_1、第二層的代表標籤字集合 L_2 以及第三層的代表標籤字集合 L_3 所對應的資料物件之聯集進行個數統計，並且與包含查詢字的資料物件集合 O_q 中的物件個數作比較。希望藉此評估系統推薦出來的代表標籤字能涵蓋多少比例的查詢結果的資料物件。因此我們以覆蓋率(coverage)為一評估與分析的準則，以下分別對於不同層的代表標籤字集合計算，依序如算式 14、15、16 所示。

O_q 中包含任一個第一層代表標籤字的資料物件比例值，以算式 12 計算出。

$$coverage(L_1) = \frac{\left| \bigcup_{t_i \in L_1} obj(q, t_i) \right|}{\left| O_q \right|} \qquad \text{(算式 12)}$$

O_q中包含任一個第一層或第二層代表標籤字的資料物件之比例值以算式 13 計算出。

$$coverage(L_2) = \frac{\left| \bigcup_{t_i \in L_1 \cup L_2} obj(q, t_i) \right|}{\left| O_q \right|} \qquad \text{(算式 13)}$$

O_q中包含任一個代表標籤字的資料物件的比例值以算式 14 計算出。

$$coverage(L_3) = \frac{\left| \bigcup_{t_i \in L_1 \cup L_2 \cup L_3} obj(q, t_i) \right|}{\left| O_q \right|} \qquad \text{(算式 14)}$$

上述三個算式中L_1、L_2、L_3分別表示第一層、第二層和第三層的代表標籤字集合。t_i為概念階層式架構中的代表標籤字。$obj(q, t_i)$表示為包含查詢字 q 及代表標籤字 t_i的資料物件集合。

算式 12、13、14 表示在查詢結果 O_q 中，在每一層不包含該層以上層級代表標籤所找到的資料物件集之情況下，計算該層代表標籤所能找到的資料物件集個數，摒除以 O_q 中的資料總數，此值介於 0 到 1 之間，愈接近 1 代表涵蓋 O_q 中愈多資料物件。

<2> 重複程度

重複程度是評估在階層式架構的每一層中的代表標籤字，兩兩代表標籤字所找到資料物件的重複程度。我們希望能夠有效篩選查詢結果且覆蓋率能夠越高的狀況下，也希望這一些代表標籤字所能找到的資料物件彼此間的重複性不要太高。避免使用者在選定不同的代表標籤字，卻回傳相同的資料物件。因此重複程度(overlap)為我們的另一評估重點，採取的方式係以各層分開作計算。對於特定一層的代表標籤字 t_j，其與該層的其他代表標籤字 t_i 統計出在 O_q 中共同的資料物件個數，除以 O_q 中出現 t_j 的資料物件個數，算出重複出現的比例值。並且以此方式將同一層的所有代表標籤字和其他標籤皆計算出重複程度後取平均值，如以下算式 15 所示。

$$overlap(L_k) = \frac{\left| \sum_{t_i, t_j \in L_k} \frac{|obj(q,t_i) \cap obj(q,t_j)|}{|obj(q,t_j)|} \right|}{|L_k| * (|L_k| - 1)}, i \neq j \qquad \text{(算式 15)}$$

L_k表示為第 k 層的代表標籤字集合，我們建立的架構共有三層，因此 k 可以設定為 1 或 2 或 3。$overlap(L_k)$值愈低代表第 k 層兩兩代表標籤字所找出的資料物件較少有重複的現象。

<3> 選擇性

選擇性是評估 O_q 中每個物件 o，考慮在 O_q 中有多少資料物件無法被任一個可涵蓋 o 的代表標籤字集所涵蓋，並計算其在 O_q 中的比例值。此值表示有多少比例的資料物件可和物件 o 有效區分開。對每個物件算出此比例值，最後再計算整體平均值，此值愈大表示概念階層式架構中特定一層的代表標籤集之過濾篩選效果愈好。當使用者下一個查詢時，系統應該輔助使用者找出感興趣且具體的資料物件，所以會希望盡可能篩除掉一些不符合使用者搜尋意圖的資料物件。我們希望對於能找到特定資料物件的標籤不要涵蓋其他太多物件，如此一來使用者才能依照選定的代表標籤字，較快速找出所需資料。因此，選擇性(selectivity)為我們第三個評估查詢結果之語意階層式架構效果的測量值，如算式 16 所示。

$$selectivity(L_k) = \frac{\sum_{o_i \in O_q} \frac{|\{o_j \mid o_j \in O_q \wedge o_j \notin \bigcup_{t \in (L_k \cap o_i.tagset)} obj(t)\}|}{|O_q|}}{\left|O_q\right|}, i \neq j \text{^} k = \{1,2,3\} \quad \text{(算式 16)}$$

$o_i.tagset$表示 O_i 所包含的標籤字集合，$o_j.tagset$ 亦同。L_k 則表示第 k 層的所有標籤字集合。算式 16 針對包含查詢字的資料物件集合O_q，從中一次挑選一個資料物件 O_i 並同時找出該資料物件 O_i 所擁有的標籤字集合 $o_i.tagset$。將這些標籤字與特定一層的所有代表標籤字集 L_k 比對，找出有哪一些代表標籤字 $L_k \cap o_i.tagset$是存在於被選定的此資料物件o_i中的標籤集合內。接著再從其他包含查詢字的資料物件，計算有多少個物件 O_j 未具有$L_k \cap o_i.tagset$中任一個標籤，

其所占 O_q 的平均比例值，再計算整體平均值。此值愈高其代表著特定一層的代表標籤字集可有效篩除大量的資料物件。

根據上述的三種評估準則，我們希望其查詢結果階層式標籤架構能夠符合"覆蓋率高、重複率低且選擇性高"的特性，視為我們的目標。

6.2.3 實驗評估結果

[實驗 1.1] 評估挑選代表標籤的步驟對於查詢結果之影響

為了比較使用代表性標籤選取步驟的效果，因此分別所採用兩種方法為:

(一) 使用 4.2 節所介紹 r_score 方法計算出候選標籤字相關程度值，從中取出 r_score 分數最高的前五十名之候選標籤字作為代表標籤字，再將這些代表標籤字進行標籤階層式架構之建立依據。我們稱此方法為 Using filtering strategy (簡稱成 FS)。

(二) 在不採代表性標籤選取的方法，我們將 O_q.tagset 的所有候選標籤字輸入概念廣泛程度排名模型進行排序，取出概念廣泛程度最高的前五十名之標籤字作為後續標籤階層式架構之建立依據。我們稱此方法為 No filtering strategy (簡稱成 NFS) 。

同時我們想比較前述的三種類型查詢字(高、中、低頻率標籤字)，針對 FS 及 NFS 兩種做法，評估其建構出語意標籤架構中不同階層所對應的代表標籤字集合，依序列出三項評估數據。因此為了以各個評估標準來觀察 FS 及 NFS 對於所建立標籤概念階層式架構提供查詢瀏覽的效果並且進行三種類型查詢字的實驗結果

分析討論。圖 6.1、6.2、6.3 依序為對於覆蓋率、重複程度、選擇性的評估結果。

首先在圖 6.1 中，我們可以得知當使用高頻率查詢字，其 FS 和 NFS 兩種挑選標籤字的方法所建立出的階層式架構的覆蓋率差距較大，而使用低頻率查詢字時，FS 和 NFS 的覆蓋率差距則較小。

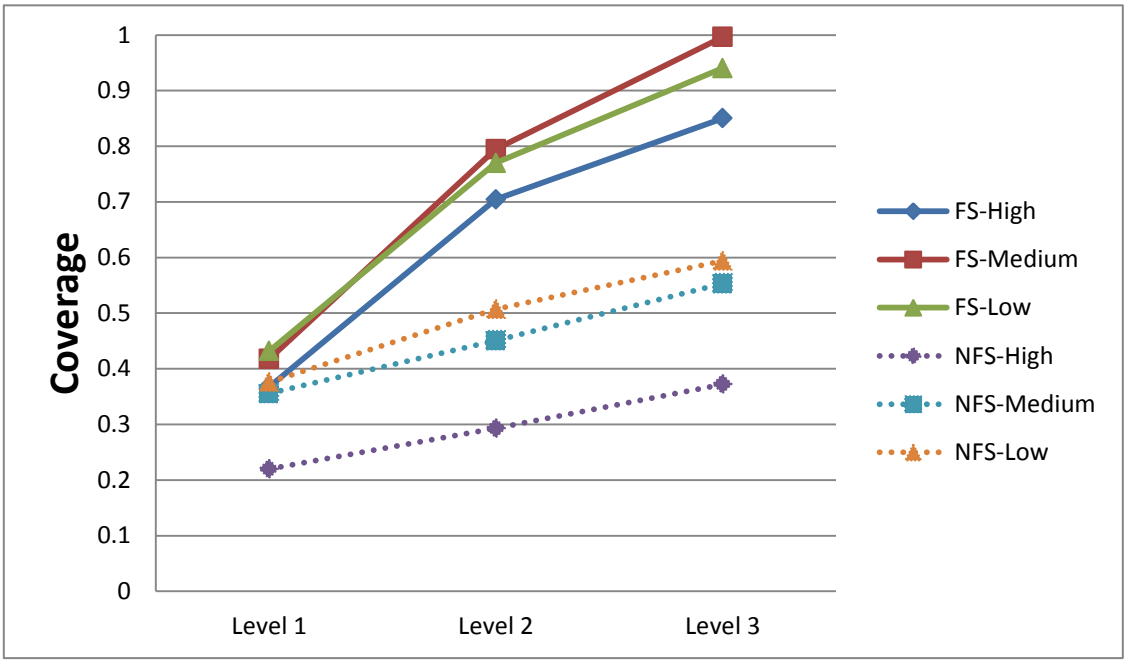


圖 6.1 採用不同挑選代表標籤字策略的標籤架構-階層累積覆蓋率折線圖

接著圖 6.2 則可以看出使用三種類型的查詢字，在重複程度的評估上，和層級的關係並不明顯，FS 方法所得數值略高於 NFS 方法。在包含查詢字的資料物件少(Low frequency queries)的情況下，使用 FS 方法時，其挑選出的代表標籤與查詢字同時出現的相關性高，且因為找到的資料物件個數比較少，導致查出的物件重複程度較高。

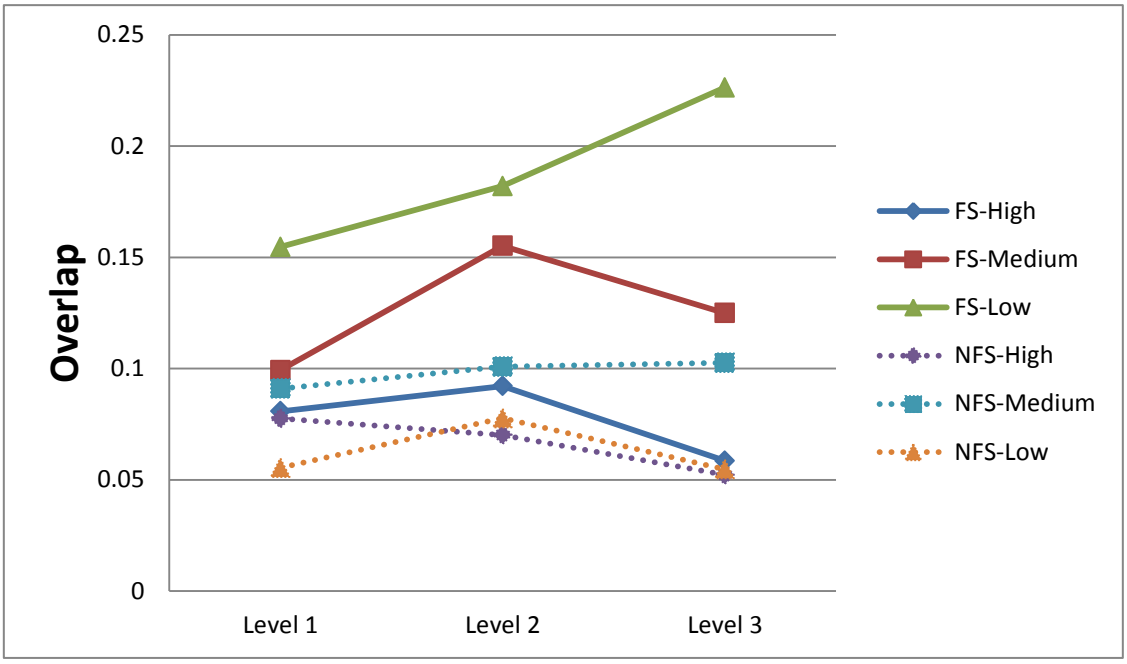


圖 6.2 採用不同挑選代表標籤字策略的標籤架構-重複程度之階層分佈圖

圖 6.3 顯示是否採用挑選代表標籤字的篩選策略對於不同階層中選擇性的影響並不這麼明顯，但是當我們以低頻率查詢字做搜尋並採用 FS 方式時，其效果相較於 NFS 方法是比較差的。主要原因為我們選取與查詢字具較高相關性的代表標籤字集合，當 O_q 集合中物件個數較少，以具有高關聯性的字詞作為篩選條件，仍會找出 O_q 中大部分的資料物件，因此在篩除資料的效果上較不明顯。

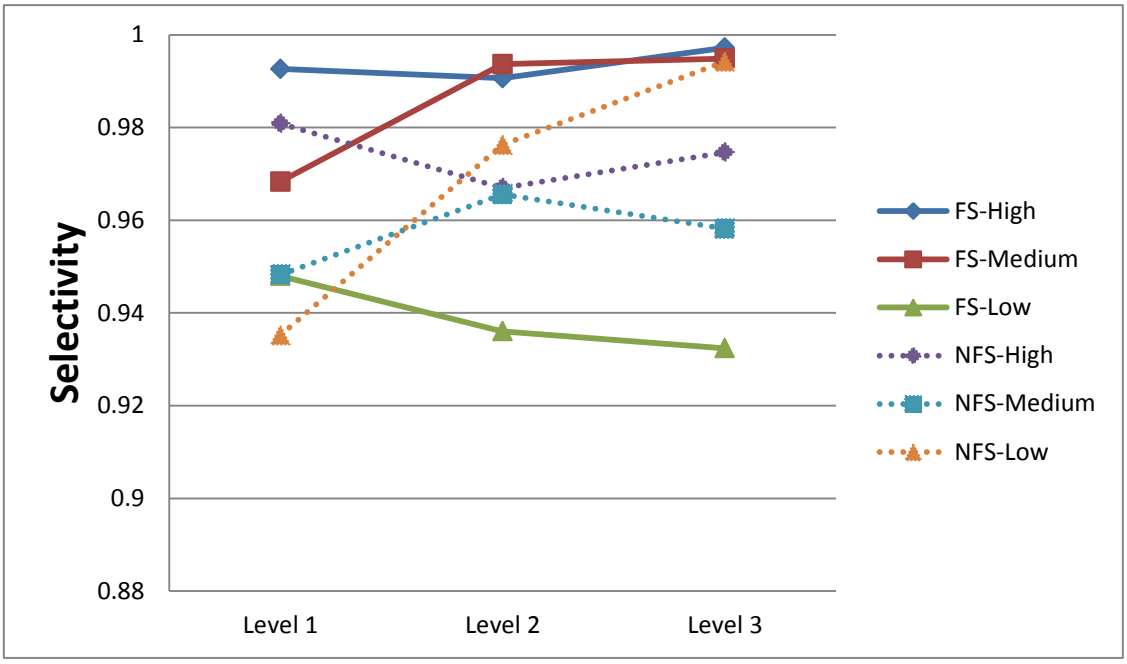


圖 6.3 採用不同挑選代表標籤字策略的標籤架構-選擇性之階層分佈圖

表 6.2 採用不同挑選代表標籤字策略的標籤架構之整體評估

	FS	NFS
Ave_Coverage	0.31966	0.16885
Ave_Overlap	0.13044	0.09986
Ave_Selectivity	0.97264	0.96668

綜合考量*coverage*、*overlap*、*selectivity*三項平均數據(如表 6.2)，雖然在 overlap 的值 FS 方法是略高於 NFS 方法，但在 coverage 上 FS 方法明顯高於 NFS 方法，而在 Selectivity 上兩者效果差不多。因此在整體上仍可說明使用 FS 的挑選代表標籤字方法是有必要的。可證實有使用 r_score 的篩選機制在整體上是優於僅僅考慮語意廣度的排序來進行挑選的作法。

以下分析三種不同類型查詢字的實驗結果:

i. 高頻率查詢字

覆蓋率的實驗結果,可以看出有使用篩選機制所建立出來的階層式架構(FS),從第一層到第三層皆優於未使用篩選機制的方法(NFS)。重複程度的實驗結果,可得知兩種方法在重複程度上,雖然 NFS 的方法略低於採用 FS 的方法,但其差異沒有相當明顯。選擇性的實驗結果,FS 的方法優於 NFS 的方法。

ii. 中頻率查詢字

覆蓋率的實驗結果,可以看出有使用篩選機制所建立出來的階層式架構(FS),從第一層到第三層皆優於未使用篩選機制的方法(NFS),在第二層和第三層尤為明顯。重複程度的實驗結果,可得知兩種方法在重複程度上, NFS 的方法略低於採用 FS 的方法。選擇性的實驗結果,FS 的方法則優於 NFS 的方法。

iii. 低頻率查詢字

覆蓋率的實驗結果,可以看出有使用篩選機制所建立出來的階層式架構(FS),各階層皆優於未使用篩選機制的方法(NFS)。重複程度的實驗結果,顯示 NFS 的方法比採用 FS 的方法得到較低的 coverage,可以解釋的原因為隨著包含查詢字的資料物件的減少,導致我們所使用 r_score 來進行篩選標籤字的效果無法彰顯出來。選擇性的實驗結果,僅在第一階層 FS 方法是略高於 NFS 方法,而在第二、三層則 NFS 是有較高的 selectivity。

[實驗 1.2]　評估不同的階層式架構建立方法之比較

本實驗分別採用本論文提出的 CTC 演算法所建立的概念階層式標籤架構與[6]提出的 MDL 演算法之概念階層式架構建立方式，比較兩者所建立的概念階層式架構對於查詢瀏覽的效果。

如同實驗 1.1 的方式採用三種出現頻率範圍共 400 個查詢字，並使用 FS 方式挑選代表標籤後，針對我們提出的 CTC 演算法之架構建立方法及[6]使用的 MDL 演算法架構建立方法兩種做法，評估其建構出語意標籤架構中不同階層所對應的代表標籤字集合，依序算出三項評估數據。

我們以各個評估標準來觀察 CTC 演算法及 MDL 演算法的階層式架構建立方法，其建立出的階層式架構能否有效輔助使用者進行搜尋資料。圖 6.4、6.5、6.6 依序對於覆蓋率、重複程度、選擇性來進行統整分析。以三種不同類型的查詢字搭配不同的挑選代表標籤字方式之組合，系統對於每種實驗的配對組合進行各層的評估。

首先在圖 6.4 中，我們可以得知在第一階層時，其覆蓋率會隨著包含查詢字的資料物件個數減少而相對地上升。並且可得知使用低頻率查詢字，由於覆蓋率高，其查詢結果的標籤概念階層式架構可以不需要太多層。而高頻率查詢字，因為覆蓋率不高且表達的語意概念較廣，需要提供較多的查詢概念才會比較完整。此外，我們仍可從覆蓋率的數據中得知我們提出的建構架構方法 CTC 較 MDL 的建構架構方法可得到較高 coverage 值。

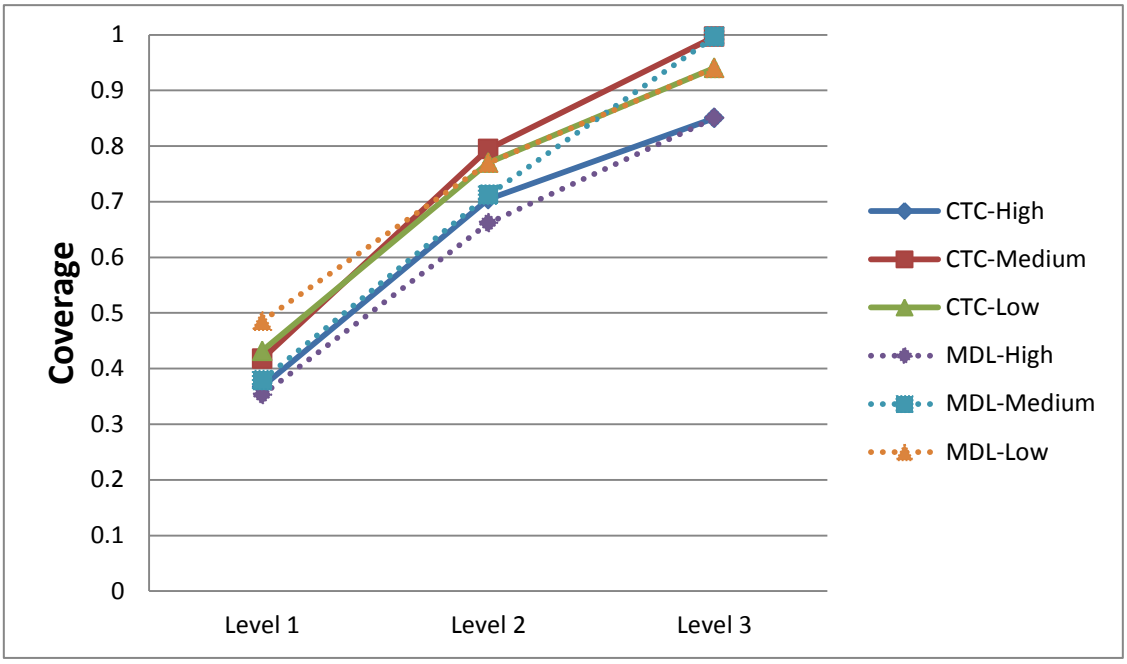


圖 6.4 採用不同建立階層式架構方法-階層累積覆蓋率折線圖

接著在圖 6.5 中，不同類型的查詢字有著明顯的差異，對於高頻率查詢字系統所組織的階層式架構，其同一階層下的代表標籤字重複程度並不高。而對於低頻率查詢字系統所組織的階層式架構，其同一階層下的代表標籤字重複程度就來的較高。主要原因是包含高頻率查詢字的資料物件個數多，其查詢結果可供使用者瀏覽的資料可能較多，相對來說資料不容易相同。整體在重複程度上，仍可從數據顯示階層式架構建立方法，CTC 演算法較 MDL 演算法有較低的 overlap 值。

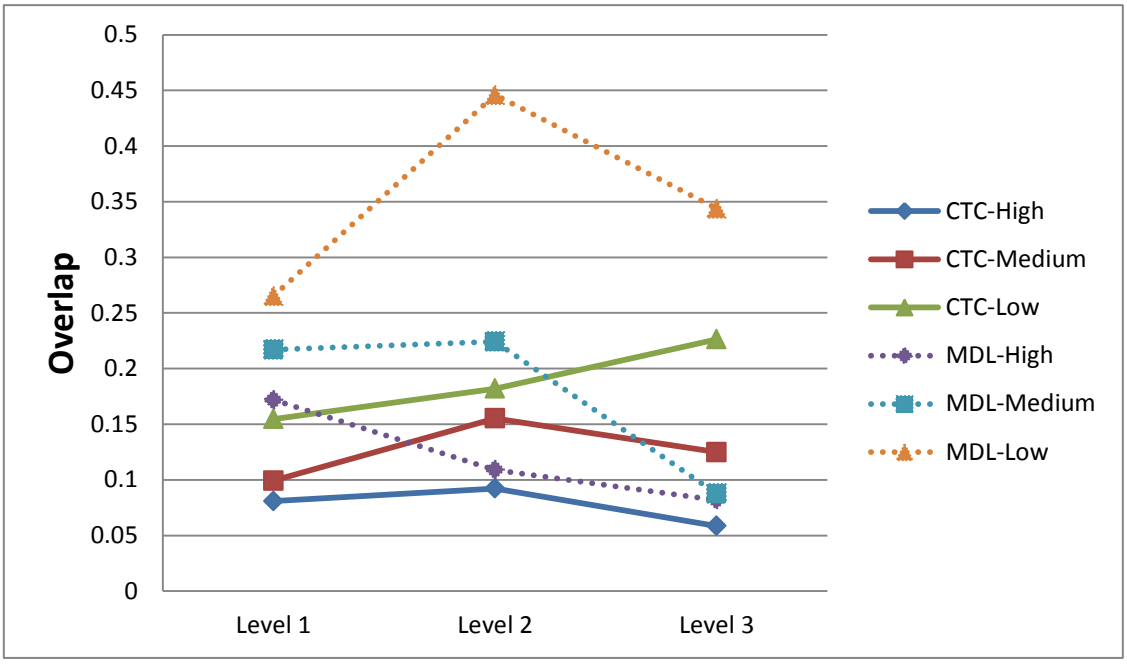


圖 6.5 採用不同建立階層式架構方法-重複程度之階層分佈圖

圖 6.6 為選擇性值實驗數值，對於三項不同類型的查詢字，其查詢結果的階層式架構，可以得知本論文使用的建立方式 CTC 都比 MDL 建立方式有較高的 selectivity。當中其選擇性值的變化會因不同類型的查詢字而有所變化。倘若依序使用高頻率查詢字、中頻率查詢字以及低頻率查詢字，其選擇性值會依序降低。原因為當我們所選取與查詢字具備較高相關性的代表標籤字集合進行組織與建立時，由於包含高頻率查詢字的資料物件數多，因而讓使用者在瀏覽資料時能夠忽略較多可能被使用者認定為不需要的資料，藉此有效地達到符合使用者所需要的具體資料之目的。因此，若使用低頻率查詢字來進行搜尋時，包含該類型查詢字的資料物件數較少，則其過濾的資料幅度也就較低(選擇性)。此外，我們可以得知選擇性通常與重複程度有反比關係。當特定一層的代表標籤集的重複重度較低時，其選擇性也就較高 - 可篩除的資料量相對較多。

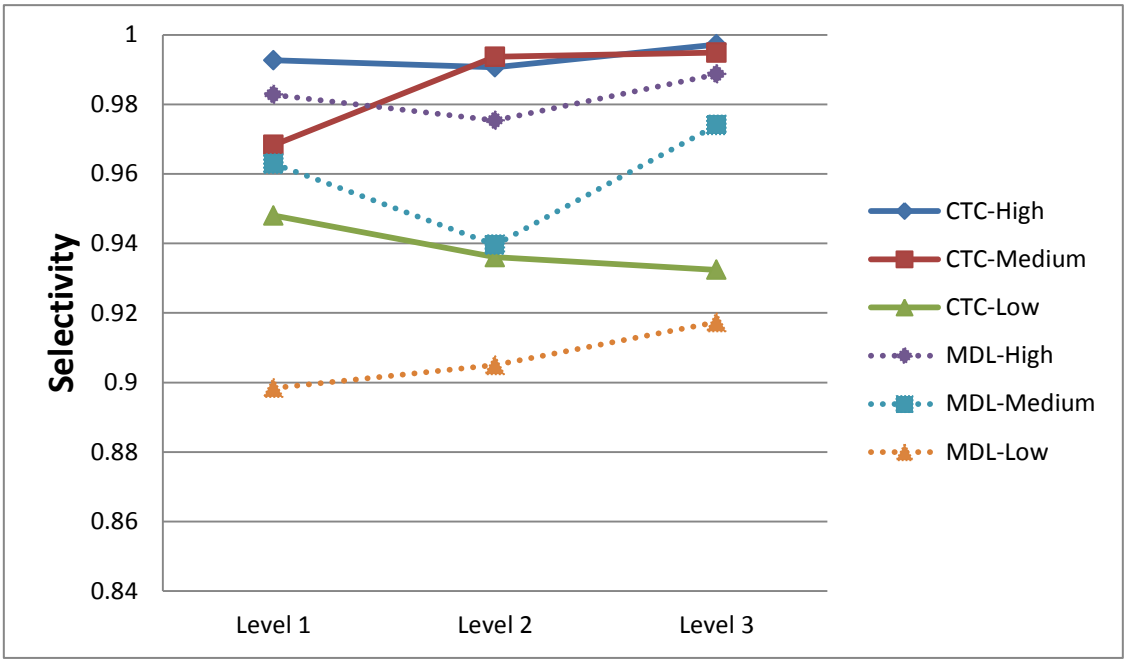


圖 6.6 採用不同建立階層式架構方法-選擇性之階層分佈圖

表 6.3 採用不同建立階層式架構方法的標籤架構之整體評估

	CTC	MDL
Ave_Coverage	0.31966	0.31966
Ave_Overlap	0.13044	0.21632
Ave_Selectivity	0.97264	0.94934

根據*coverage*、*overlap*、*selectivity*三項平均數據(如表 6.3) 數據顯示，因為挑選的代表標籤字皆是使用 R_Score 篩選機制，所以整體的平均覆蓋率會是相同的。但是不論在重複程度以及選擇性上，本論文提出的建構方式 CTC 演算法的查詢效果可顯示出比 MDL 演算法建構方式可得到較好的結果。

6.3 評估階層式標籤架構的有效性

本部份的實驗目的在於評估系統挑選出的代表標籤字是否與使用者所下的查詢關鍵字相關，以及階層式標籤架構中所提供上下包含關係的標籤組(t_i,t_j)是否符合語意概念且 t_j 能用來輔助使用者篩選概念 t_i 的資料物件中更具體的概念。這兩部分評估將分別在**[實驗 2.1]**及**[實驗 2.2]**中進行。

本實驗找了四位受試者，以表 6.2 的問卷形式，系統會回傳各階層的代表標籤字以及歸類於每個代表標籤字 t_i 之下的子概念標籤字集合。使用者需評估的包括以下兩部分:

(1) 判斷代表標籤字 t_i 是否與查詢字 q 有關。舉例而言，若"apple"為查詢字，"city"為標籤階層式架構中的一個標籤，則請使用者判斷是否與查詢字"apple" 有語意相關關 ，若有關則標記 "1"，否則標記為"0"。

(2) 判斷代表標籤字 t_i 底下的各個子概念標籤 t_j 是否是與其具備語意上下關係，也就是 t_j 可幫助查詢時指出概念 t_i 下更明確的子概念。具有此特性的標籤字對標記"1"，否則標記為"0"。

舉例來說，"city"底下的子概念標籤集合有"nyc","big","store"，這樣的查詢字是否有意義地且具備語意上下關係。

- "nyc"因為可將原先的 city 概念指定特定城市(nyc – new york city)所以是有意義的，且 nyc 是一個 city 的子概念，因此標定為"1"。
- " big"若認為可解釋成在找大城市或是描述城市的子概念,則可標"1" ，

但測試者也可能主觀覺得視為沒意義的則標"0"。

表 6.4 語意階層式架構評分問卷範例

Query: apple		
Level_1 :		
city	Rel(1)/Irrel(0)	
Sub-tagset		上下包含關係
nyc		Yes (1)/ NO (0)
big		Yes (1)/ NO (0)
store		Yes (1)/ NO (0)

6.3.1 測試資料來源

本實驗共採用了 9 個查詢字，依照 6.2.1 小節三種頻率範圍（高、中、低頻率)的標籤字當作查詢，我們分別給予使用者每一種種類型各 3 個查詢字來進行評估。表 6.3 為我們所進行實驗的查詢字清單。

表 6.5 用以實驗的查詢字清單

高頻率查詢字	中頻率查詢字	低頻率查詢字
animal	apple	obama
car	kitten	story
rock	asia	musician

6.3.2 實驗評估方法

系統為一個查詢 q 找到資料物件集合O_q並從O_q挑選出代表標籤字集，評估其是否與查詢 q 有關以及建立完成的概念階層式架構中所提供上下包含關係的代表標籤字組是否具有語意概念且有效地篩選資料物件幫助使用者找到更具體的資

料物件。針對以上兩點，我們以平均精確度來進行評估。一個好的代表標籤字之挑選方法，其平均精確值應該要愈高。而能夠顯示出代表標籤字的語意上下包含關係的階層式架構，其平均精確值也會愈高。平均精確度之定義如下所示:

$$precision(q) = \frac{\#\ related\ \ tags}{\#\ total\ evaluated\ tags} \qquad \text{(算式 17)}$$

$$Average\_precision(Q) = \frac{\sum_i precision(q_i)}{|Q|} \quad , q_i \in Q \quad \text{(算式 18)}$$

$\#\ related\ \ tags$表示為被使用者判斷為與查詢字相關的代表標籤字個數。

$\#\ total\ evaluated\ tags$是以查詢字 q 進行查詢時，系統所挑選的代表標籤字個數。

判斷相關與否的平均精確度是當以查詢字 q 查詢時且經由使用者標記後，我們計算系統推薦的每一個代表標籤字是否相關的一項資訊檢索評估方法，如算式 17 所示。接著計算我們的九個實驗查詢字之精確值平均，如算式 18 所示。

至於語意上下包含關係的精確值計算則以範例 6.1 舉例說明。

［**範例 6.1**］

以表 6.4 為例，當下一查詢 q，系統回傳兩個代表標籤字 t_1 和 t_2，各自底下子概念標籤分別為{ t_{11}, t_{12}, t_{13}}及{ t_{21}, t_{22}}，使用者判定語意概念包含關係與否於 Judger vote 欄位所示。對於標籤 t_1 而言，標籤 t_{11} 和標籤 t_{12} 被認為是標籤 t_1 的子概念，標籤 t_{13} 被認為並非是標籤 t_1 的子概念。對於標籤 t_2 而言，標籤 t_{21} 認為是標籤 t_2 子概念，標籤 t_{21} 被認為並非是標籤 t_2 的子概念。因此精確度為 **3/5**。之後計算了我們的九個實驗查詢字之精確值取平均，即為平均精確值。

表 6.6 計算語意包含關係之平均精確值範例

Representative tags	Sub-tagset	Judger vote
$\mathbf{t_1}$	t_{11}	Yes
	t_{12}	Yes
	t_{13}	No
$\mathbf{t_2}$	t_{21}	Yes
	t_{22}	No

6.3.3 實驗評估結果

[**實驗 2.1**]　評估代表標籤字是否與查詢關鍵字相關

在這個實驗中我們分別採用 FS 及 NFS 方法取得代表標籤字集合建立出的概念性階層架構，評估對於當使用者進行查詢時合者可提供較好的查詢結果。而在此部分實驗則改採用人為判斷的評估方式，來進行 average precision 的實驗評估。

圖 6.7 顯示使用了 r_score 篩選機制(FS)的做法以及未使用 FS 篩選機制(NFS)的做法，經過四位受試者評估九個查詢字結果找出代表標籤的 average precision 數值。FS 所挑選出來的代表標籤字被受試者認為與查詢字相關的數目較多，而 NFS 所挑選出來的代表標籤字普遍被受試者認為與查詢字相關的數目則較少，因而可以顯示 r_score 的篩選步驟的必要性。

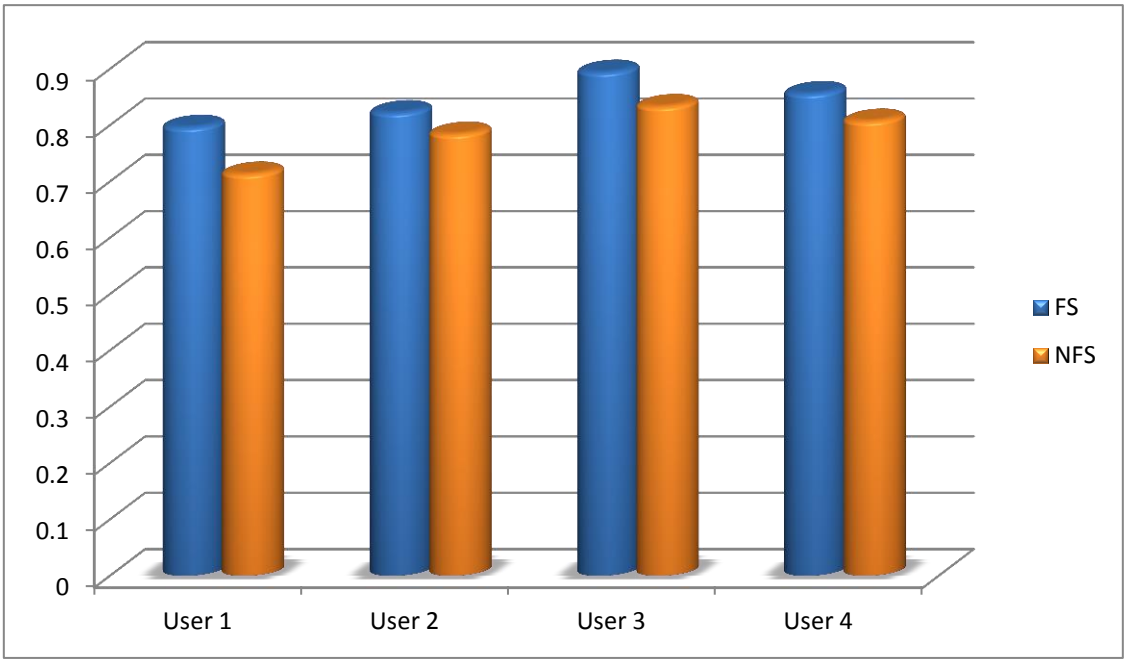


圖 6.7 不同挑選代表標籤字方法的評估(Average_Precision)比較結果

[**實驗 2.2**] 評估階層式架構中之上下包含關係的標籤組是否符合語意包含關係

此實驗主要是判定不同建立階層式架構的方法中的上下包含關係是否具查詢條件特殊化(specialization)的語意。

圖 6.8 顯示了對於使用不同的語意階層式架構建立方法，對於九個查詢結果經過四位受試者評估其組織的各階層代表標籤及其底下的子概念標籤式是否具有語意包含關係的 average precision 數值。數據顯示本論文提出的建立方式 CTC 在語意包含關係的平均精確度較高於 MDL-Tree 所建立階層式架構的方法，表示受試者認為我們所提出的建立方式(CYC)，其代表標籤的語意關係組織較具有概念的上下包含關係。

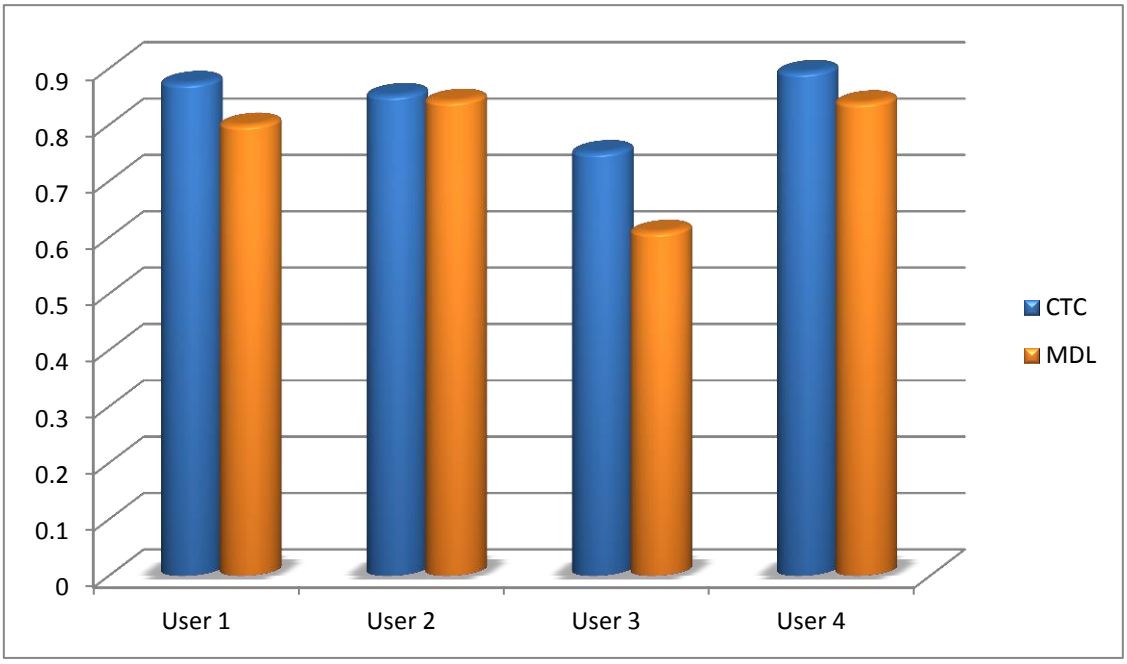


圖 6.8 不同階層式架構建立方法的評估(Average_Precision)比較結果

第七章 結論與未來研究方向

7.1 結論

本論文針對標籤資源提出一個對於查詢結果標籤集自動建立概念階層式架構的方法。首先系統針對使用者給予的查詢從標籤資源中找出包含查詢字的資料物件，從這些物件中蒐集到候選標籤字集合進行關聯代表性值 r_score 的分數計算，再依照關聯代表性值高低挑選分數最高的前 K 個標籤字作為代表標籤字。接著再將這些代表標籤字進行語意概念的處理並且組織建立成標籤概念階層式架構。

實驗部分評估了代表標籤字的挑選方法以及概念階層式架構建立演算法。對於建立好的語意階層式架構，當使用者下一查詢所找到的資料物件集合中，評估在該物件集合中想進一步以代表標籤字作為查詢搜尋資料時，其查詢結果的效果好壞。我們針對不同的代表標籤字挑選方法並且與[6]所建立的概念階層式架構進行比較分析。根據實驗的評估結果顯示，在系統化測試上本論文方法所挑選的代表標籤字並進行建立語意階層式架構，其覆蓋率、重複程度以及選擇性皆能顯示出有好的查詢效果。而在問卷評估上，本論文提出的階層式架構建立演算法也能找出較具備概念上下包含關係的標籤架構。

7.2 未來研究方向

本論文對於標籤字的概念廣度大小判別所使用的特徵值計算方式可以再做進一步改進，例如主題關鍵字的挑選，希望實質找出具備代表性的主題字，而不僅僅只考慮出現頻率次數的多寡作為參考。此外，不論是排名模型亦或是分類模型的特徵仍可進行增添，皆是可以再做深入的探討。

另外，個人化標籤階層式架構的建立是可行的，根據使用者標記標籤的喜好及其下查詢字的意圖作為系統所推薦的標籤字集之一項參考，接著考量推薦的每一個標籤字之概念廣度，再進而建立出標籤階層式架構。當使用者進行查詢時，可以藉由該階層式架構更精確地幫助使用者找到所需資訊，使之更加完整且實用。

參考文獻

[1] R. Binzabiah and S. Wade, “Proposed Method to Build an Ontology Based on Folksonomy,” in Proceedings of the 2012 International Conference on Information Society (i-Society), 441 - 446 , June 2012.

[2] S. Lohmann and P. Diaz, ”Representing and Visualizing Folksonomies as Graph-A Reference Model,” in Proceedings of International Working Conference on Advanced Visual Interfaces(AVI), 2012.

[3] D. Skoutas and M. Alrifai, “Tag Clouds Revisited,” in Proceedings of the 20th international conference on Information and knowledge management(CIKM), 2011.

[4] P. Venetis, G. Koutrika and H. Garcia-Molina, “On the Selection of Tags for Tag Clouds,” in Proceedings of the fourth ACM international conference on Web search and data mining(WSDM), 2011.

[5] L. Adeyanju, D. Song, M-D. Albakour, U. Kruschwitz, A.D. Roeck and M. Fasil, ”Adaptation of the Concept Hierarchy Model with Search Logs for Query Recommendation on Intranets” in Proceedings of 35th international conference on Research and development in information retrieval(SIGIR), 2012.

[6] Y. Song, B. Qiu and U. Farooq, “Hierarchical Tag Visualization and Application for Tag Recommendations,”in Proceedings of the 20th international conference on Information and knowledge management(CIKM), 2011.

[7] D. Helic and M. Strohmaier,”Building Directories for Social Tagging Systems,” in Proceedings of the 20th ACM international conference on Information and knowledge management(CIKM), 2011.

[8] D. Dash, J. Rao, N. Megiddo, A. Ailamaki and G. Lohman, ”Dynamic Faceted Search for Discovery-driven Analysis,” in Proceedings of the 17th conference on Information and knowledge management(CIKM), 2008.

[9] X. Ling, Q. Mei, C. X. Zhai and B. Schatz, ”Mining Multi-Faceted Overviews of Arbitrary Topics in a Text Collection,” in Proceedings of the 14th international conference on Knowledge discovery and data mining(SIGKDD), 2008.

[10] B. Zhao, X. Lin, B. Ding and J. Han, “TEXplorer: Keyword-based Object Search and Exploration in Multidimensional Text Databases,” in Proceedings of the 20th ACM international conference on Information and knowledge management(CIKM), 2011.

[11] J. Koren, Y. Zhang and X. Liu, “Personalized Interactive Faceted Search,” in Proceedings of the 17th international conference on World Wide Web(WWW), 2008.

[12] D.C. Anastasiu, B.J. Gao and D. Buttler, “A Framework for Personalized and Collaborative Clustering of Search Results” in Proceedings of the 20th ACM international conference on Information and knowledge management(CIKM), 2011.

[13] S. Overall, B. Sigurbjornsson and R. van Zwol, “Classifying Tag Using Open Content Resources,” in Proceedings of the Second ACM International Conference on Web Search and Data Mining(WSDM), 2009.

[14] C.S. Firan, M. Georgescu, W. Nejdl and R. Paiu, ”Bring Order to Your Photos: Event-Driven Classification of Flickr Images Based on Social Knowledge,” in Proceedings of the 19th ACM international conference on Information and knowledge management(CIKM), 2010.

[15] V. Dang and R.W. Croft, ” Query Reformulation Using Anchor Text,” in Proceedings of the third ACM international conference on Web search and data mining(WSDM), 2010.

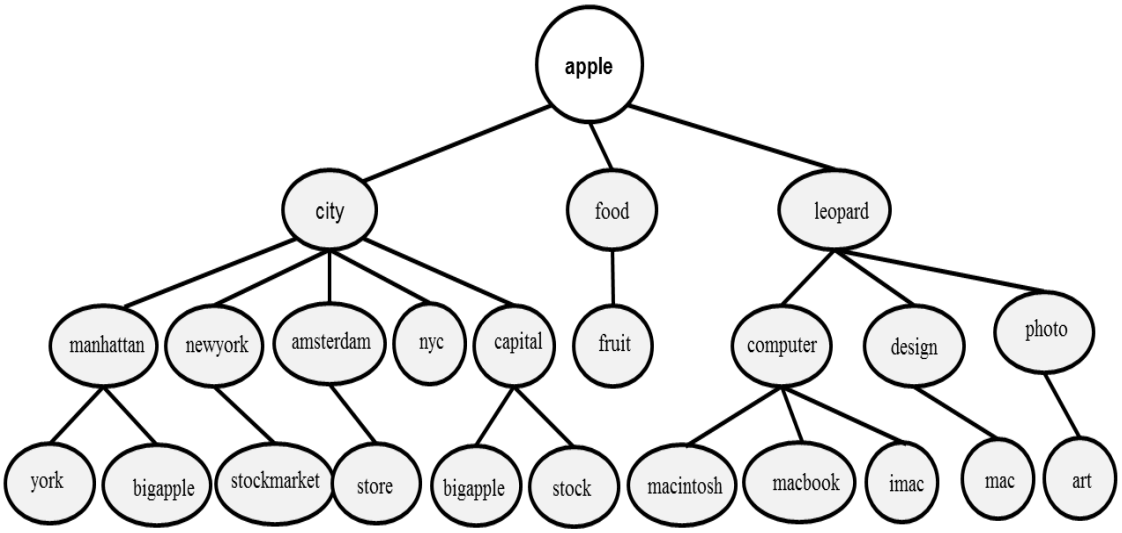
[16] M. Sanderson and B. Croft, “Deriving concept hierarchies from text,” in Proceedings of the 22nd annual international ACM conference on Research and development in information retrieval(SIGIR), 1999.

[17] R. Baeza-Yates and A. Tiberi, “Extracting Semantic Relations from Query Logs, ” in Proceedings of the 13th ACM international conference on Knowledge discovery and data mining(KDD), 2007.

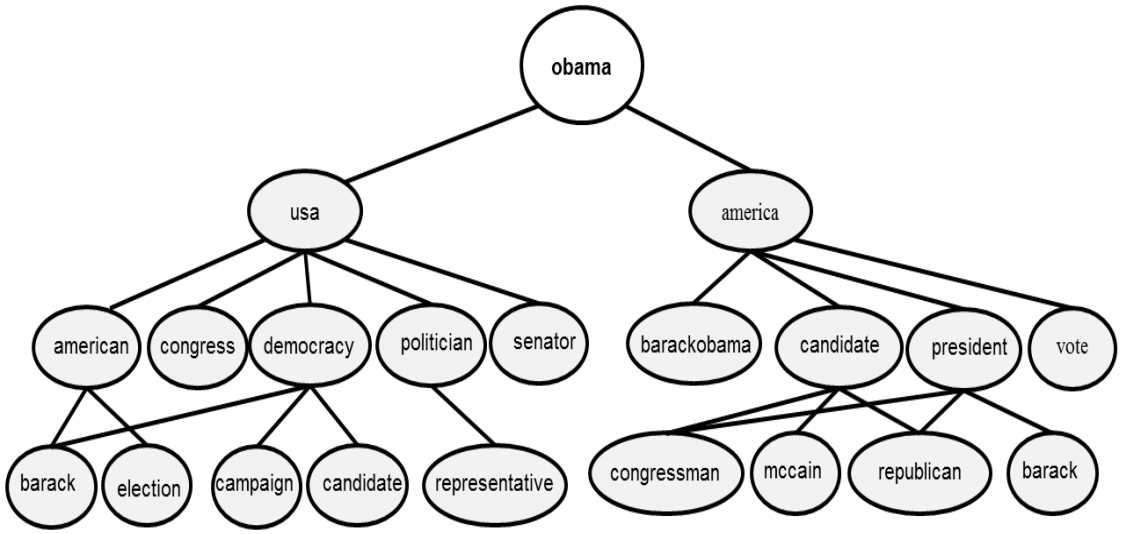
[18] R. Baraglia, F. Cacheda, V. Carneiro, D. Fernandez, V. Formoso, R. Perego and F. Silvestri, “Search Shortcuts: a New Approach to the Recommendation of Queries,” in Proceedings of the third ACM conference on Recommender systems(RecSys), 2009.

附錄

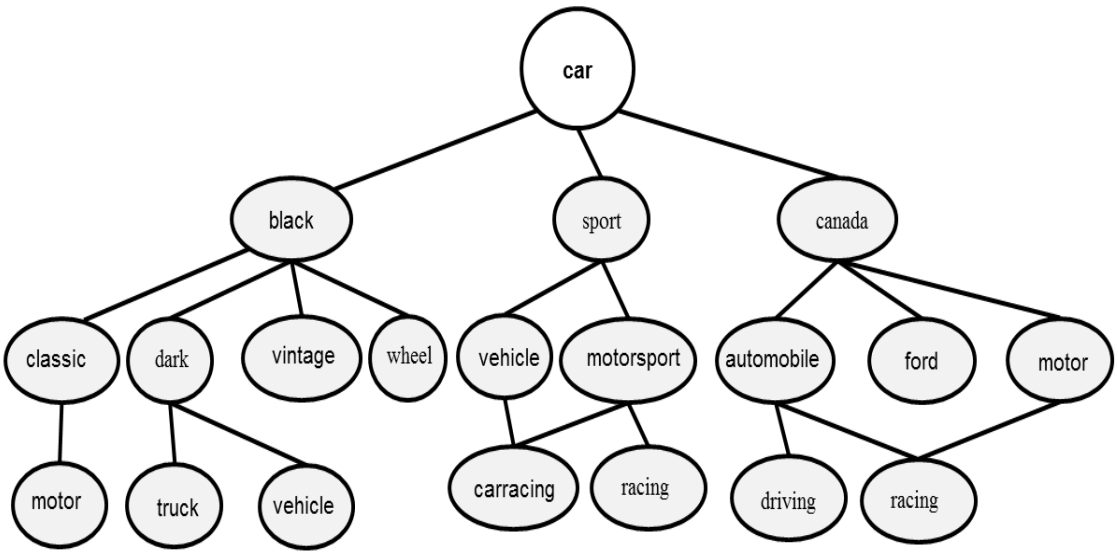
查詢字"apple"之階層架構



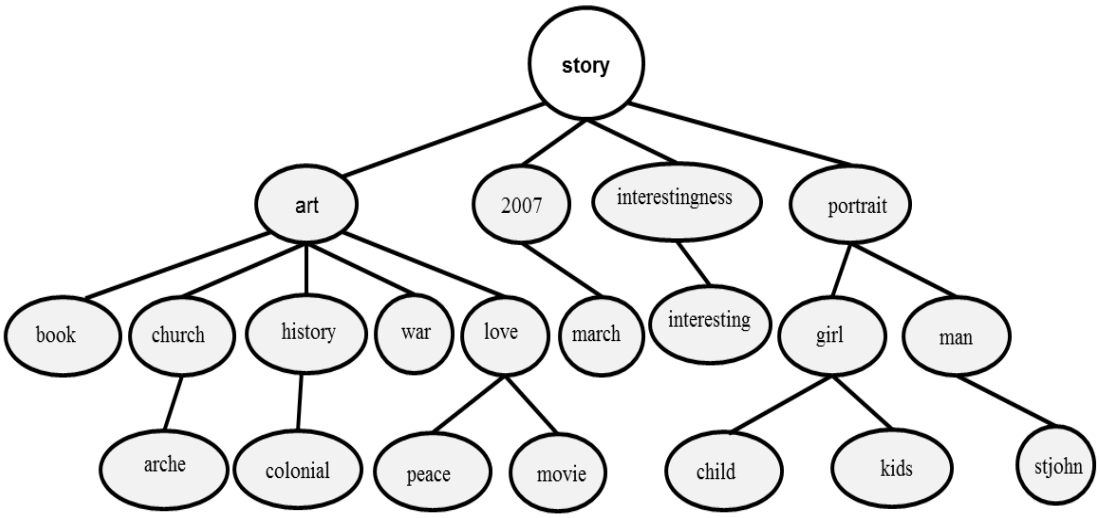
查詢字"obama"之階層架構



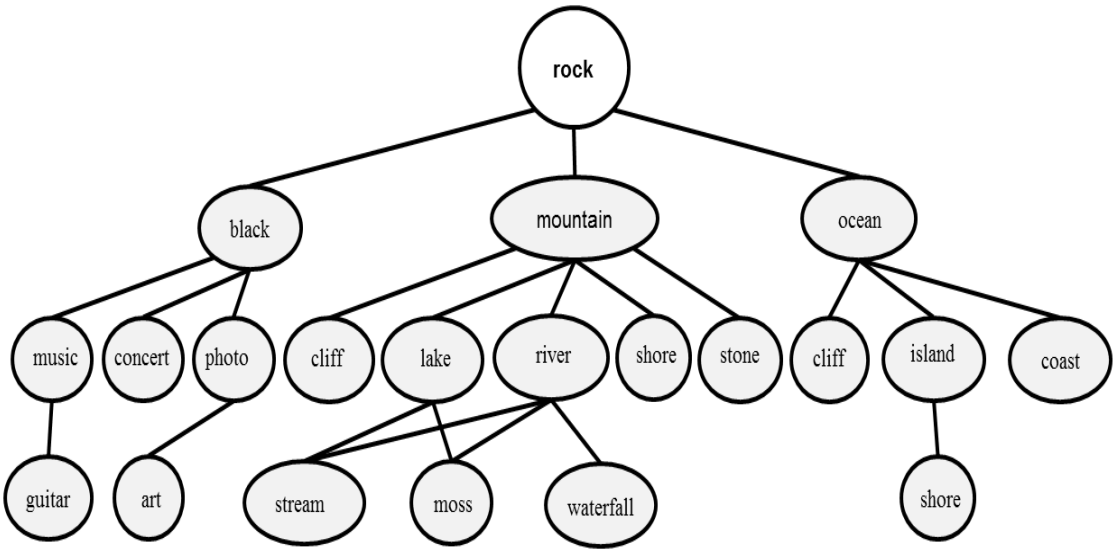
查詢字"car"之階層架構



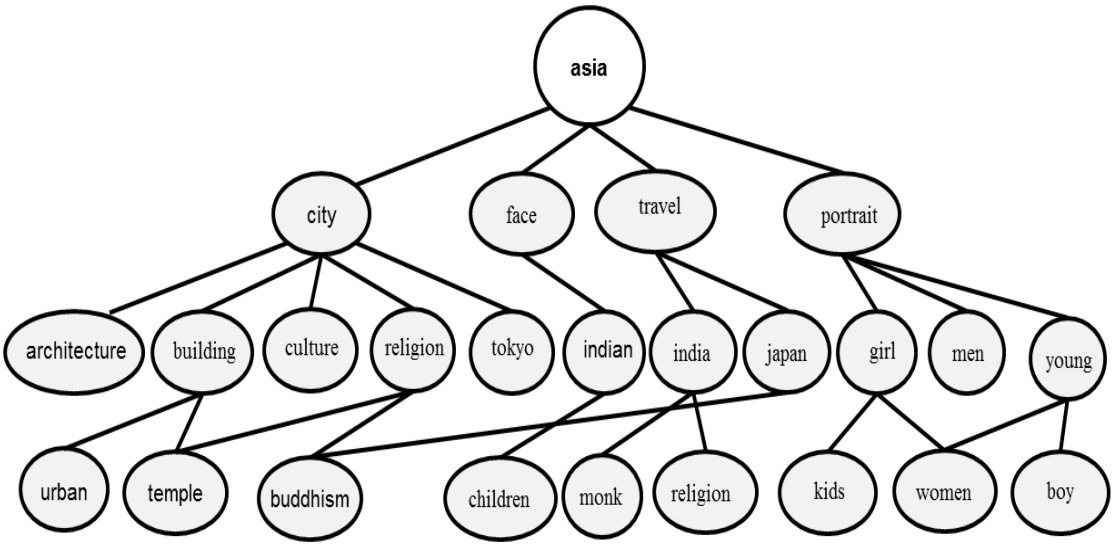
查詢字"story"之階層架構



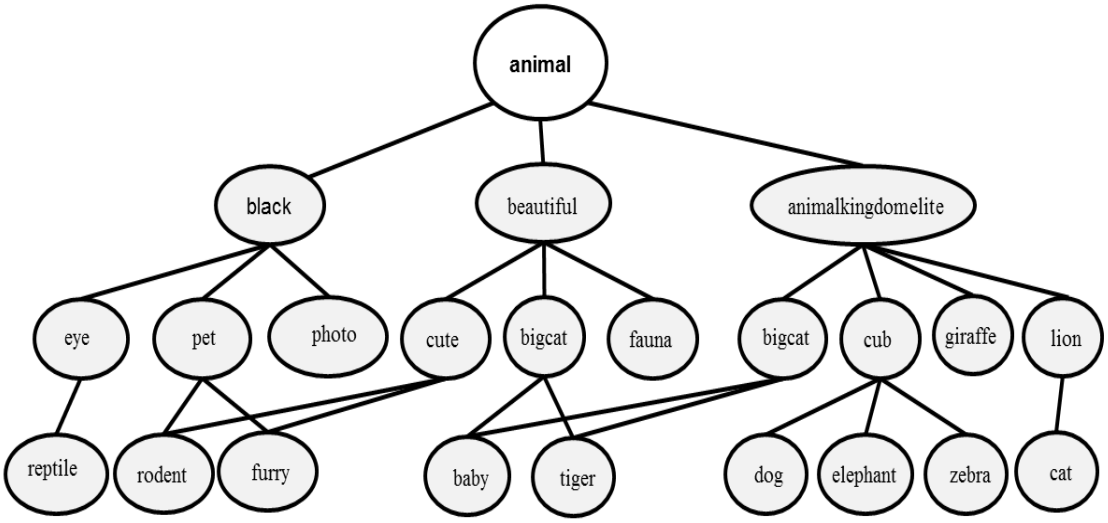
查詢字"rock"之階層架構



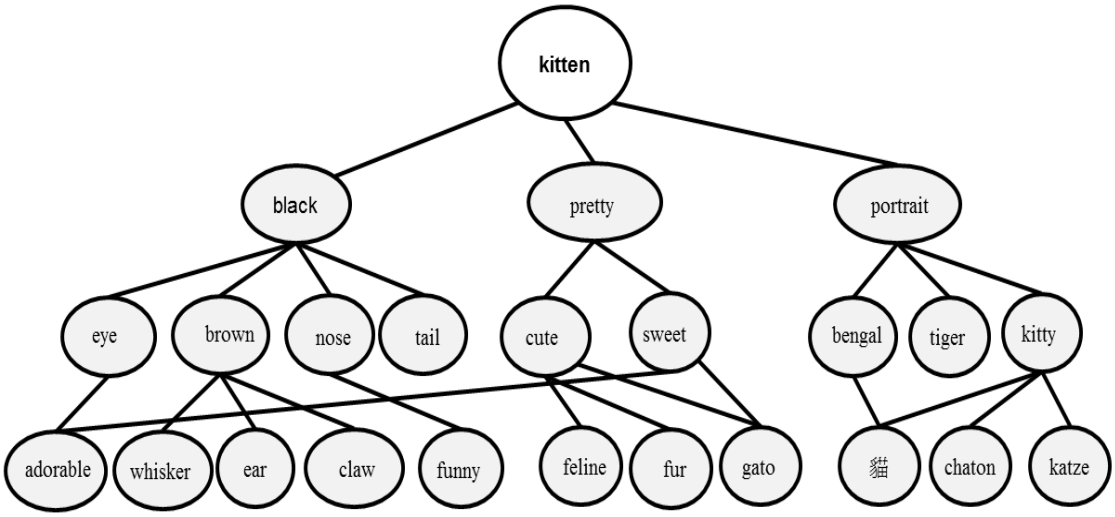
查詢字"asia"之階層架構



查詢字"animal"之階層架構



查詢字"kitten"之階層架構



查詢字"musician"之階層架構