以正規概念分析為基礎之本體論自動擴展機制

林群貿 李明哲 王宗一
國立成功大學工程科學系
n9695117@mail.ncku.edu.tw

摘要

隨著數位學習領域的標準趨向統一化，大部分的數位教材內容、學習元件(Learning Object)皆是以 IEEE 所制定的「學習物件後設資料」（learning objects metadata; LOM）來描述學習元件。我們可以輕易的在網路上搜尋到許多符合國際標準的學習元件，而這些學習元件可以一再的被不同的教學者重組、再利用。然而，隨著科技日新月異，在數位學習領域裡，新的學習概念也逐漸增多。

因此本研究提出一個以正規概念分析為基礎之本體論自動擴展機制，主要著重在分析 LOM (Learning Object Metadata)欄位的特性，再配合一個經過改良的 TF-IDF(Term Frequency-Inverse Document Frequency)資料前處理方法－Location weight TF-IDF(LTF-IDF)所擷取出重要的關鍵詞，接著藉由本研究提出的學習概念擷取機制，判斷是否有新學習概念之形成，並配合領域專家所建立的本體論(Ontology)，將新學習元件概念做新增的動作。

關鍵詞：本體論、數位學習、正規概念分析、學習物件後設資料

一、 緒論

近年來，由於電腦的普及與全球資訊網(WWW)的蓬勃發展，所以越來越多的資訊都能夠經過上網使用搜尋引擎而被大量被取得, 然而這種資訊的擴充速度將逐漸地失去掌控，導致資訊爆炸的時代隨之來臨。然而，在教學方面，許多老師們紛紛使用數位教材取代傳統的教學，因此在關於數位學習(e-Learning)這方面的研究也越來越被受到重視。

所以，如何有效地將網路上收集到的這些數位教材加以處理分類過後，再將之轉變整合成一個有系統的學習資料集是相當的重要的。所以也開始有許多人對於學習資料做自動分類及學習路徑建構等相關的研究。

然而隨著科技日新月異，在數位學習領域裡，當要學習的課程越來越多，需要學習的概念也逐漸增多，就需要更多的專家來制定這些學習的課程概念，如何自動擷取課程的學習概念，成為一門重要的課題，因此本論文提出一個以正規概念分析為基礎之本體論自動擴展機制，來幫助我們將網路上所收集的學習元件做學習概念自動化的擷取及本體論的擴展。

二、 相關研究

2.1 本體論基本定義

本體論(Ontology)源自哲學理論，其意義是有系統的解釋存在的現象。主要探討存在現象的一切現實事物的基本特徵。本體論定義了一個主題領域的構成詞彙，包含詞彙間基本條件與關係，以及延伸詞彙所定義的基本條件與關係的法則。本體論是一種正規化的(formal)、明確的(explicit)、概念化的(conceptualization)、分享的(share)描述[2]。知識本體是一種明確的且概念化

描述的邏輯理論[3]

當我們要使用本體論來描述某特定領域下的知識時，本體論便是由概念(Concept 或 Class)、屬性(Attribute、Property 或 Slot)、實例(Instance)與關係(Relation)等元素所組合而成[4]。

2.2 關鍵字擷取

擷取文件重要資訊以進行文件自動分類或文件管理之相關研究中，其中最具代表性之文件特徵資訊為「文件關鍵字」；因此，過去諸多文件自動分類研究乃以文件關鍵字為概念特徵。有些關鍵字足以代表文件中重要概念；所以許多研究者提出自動擷取文件關鍵字之方法，以擷取文件中具代表性之關鍵字，利於文件後續之內容分析。我們將關鍵字擷取方法區分為「文法剖析」、「詞庫比對」與「統計分析」三種方法:

(1) 文法剖析法

透過自然語言處理技術的文法剖析程式，剖析出文件中的名詞片語，再運用一些自然語言處理相關的方法與準則，過濾掉不適合的詞彙。其結果幾乎也都是有意義的名詞片語，但大部份的剖析程式，需要藉助已經建立的詞典或語料庫，因此其缺點無法擷取所有關鍵字（因為受限於詞庫規模）。除此之外，有些文法剖析法甚至只能剖析合乎文法的完整文句，使得書目、標題等資料裡的關鍵詞或特殊專有名詞，無法被擷取出來。

(2) 詞庫比對法

關鍵字擷取技術中以「詞庫比對法」所擷取之詞彙正確性最高，此乃因直接比對詞庫中之正確詞彙，可保證所擷取之詞彙皆為正確合理之詞彙；但其缺點為無法擷取所有關鍵字（因受限於詞庫規模）。詞庫比對之前需先建立詞庫，關於詞庫建立之相關研究，具有詞彙間關聯之詞庫，利用詞庫中與輸入關鍵字高度相關之其他關鍵字，查詢時能一併搜尋並回應相同概念資料。

(3) 統計分析法

透過對文件的分析，累積足夠的統計參數後，再將統計參數符合某些條件的片語擷取出來。最簡單的統計參數是計數詞彙發生的頻率，即詞頻，將詞頻落在某一範圍的詞彙取出。由於沒有用到詞庫或語料庫，會有擷取錯誤的情況發生，得到無意義或不合法的詞彙。此外，統計參數不足的關鍵詞無法被選到。然而其優點是較不受語文國別與句型的限制，而且可以擷取出未曾被詞庫、語料庫網羅的專業用語、新生詞彙與專有名稱等片語。

2.3 FCA 正規化概念分析

FCA 正規化概念分析(Formal Concept Analysis)是一種從資料集合(Data sets)中發現概念結構(Conceptual structures)的資料分析理論，在 1982 年由 Rudolf Wille 提出了這個方法之後， FCA 目前已經快速的發展並應用到許多領域如：醫學、心理學、音樂學、語言學、資料庫、圖書館學、資訊科學、軟體工程、生態學及其它領域。Priss(2003)指出「在資訊科學的領域中，FCA 也有許多的應用：FCA 運用在數學方格上可以用來解釋分類系統。正式的分類系統可以根據關係之間的一致性來分析。」

2.3.1 概念點陣(Concept Lattices)

概念點陣繪製出一個最上方之具體概念至下方特殊概念排序的圖形，最上方的最大子概念，稱為上確界 (Supremum)；最下方的最小子概念，稱為下確界(Infimum)。圖 1 是表 1 所呈現出概念全文的概念點陣，全文中若有註記為「X」，意味著與同一列的物件跟同一欄的 Slots 有關聯，其中上確界包含了屬性‘Fish’ 的所有物件集合，向下移動得到一個較特殊概念—包含物件集合{‘Fred’，‘Bob’， ‘Mel’} 及屬性集合{‘Fish’， ‘Chicken’}，呈現出較少的物件分配到較多的屬性。進一步分析整個概念點陣，它提供了更多隱藏在資料間概念的關聯 (Tam，2004)。

表 1 概念本文

I		屬性			
		Fish	Beef	Pork	Chicken
物件	Fred	×			×
	Jess	×	×	×	
	Bob	×		×	×
	Mel	×		×	×

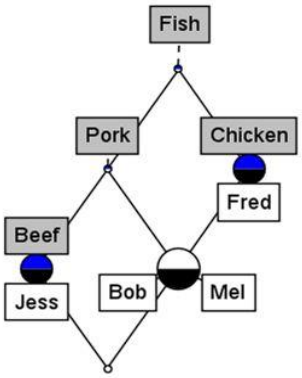


圖 1 概念點陣

三、 學習概念探勘機制

3.1 系統架構

- **LOM 剖析器(LOM Parser)**

本模組主要的功能為用來剖析學習元件後設資料(LOM)，並分析欄位的特性，找出有助於判斷分類的關鍵詞(Term)。

- **重要關鍵字擷取模組(Important Term Set Extraction Module)**

剖析學習元件後設資料後，經過斷字移除(StopWords Remove)、Stemming Algorithm 處理後，找出有助於判斷分類的關鍵詞(Term)，建立 Important Term Set，包含 Title、keyword、Description 等資訊，其中 Title、Keyword、Description 欄位隱含著許多有助於分類的關鍵詞，因此我們從這些欄位中取出和 Ontology 有對應的關鍵詞，並個別建立 Title Term Vector、Keyword Term Vector、Description Term Weigh Vector。

Term	superclass	subclass
Frequency	1	1

Title Term Vector

Term	superclass	subclass
Frequency	1	1

Keyword Term Vector

Term	object	inherit	class	superclass	subclass
Frequency	2	1	6	1	1

Description Term Vector

- **輔助關鍵字擷取模組(Assistant Term Set Extraction Module)**

除了擷取出 Important Term Set 外，我們會在再從學習元件的 LOM 中找出有助於分類的欄位，並建立 Assistant Term Set。我們將這些關鍵詞存入 Assistant Term Set 作為輔助後續的分類判斷。像是學習元件後設資料中的 technical 類別主要是描述此學習元件的技術需求和特性，以一個介紹 JAVA 程式語言的學習元件為例，在其 LOM 中的 technical 欄位，含有 J2SE Development Kit 5.0、Jbuilder…等關鍵詞，藉由這些資訊，我們可以得知這個學習物件跟 Java 程式語言有一定的關係。我們將這些關鍵詞存入 Assistant Term Set 作為輔助後續的分類判斷。

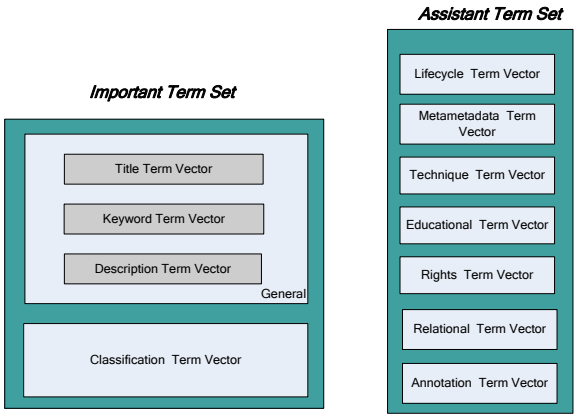


圖 2 重要關鍵詞集合、輔助關鍵詞集合

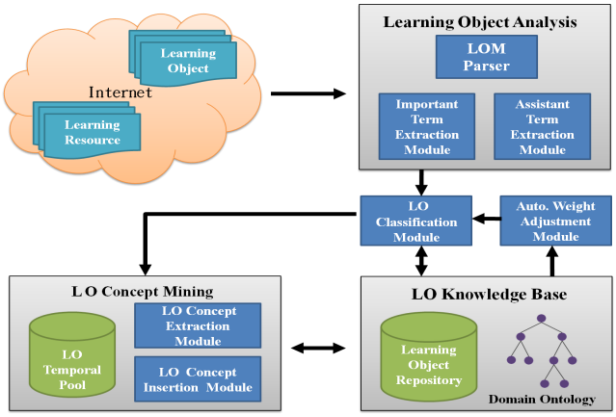


圖 3 系統架構

- **學習元件分類模組(LO Classification Module)**

根據 Important Term Set、Assistant Term Set 所提供的關鍵詞(Term)與本體論的資訊，將學習元件做概念分數(TCS, Total Concept Score)的計算，並制定兩個 TCS 門檻值 λ1(下界)與 λ2(上界)，利用 λ1 與 λ2 來過濾出與本 ontology 有些許程度相關的學習元件，以便作新學習元件概念之擷取，至於兩個 TCS 門檻值 λ1 與 λ2 值的制定，將在實驗設計與結果分析，會有詳細的介紹。

- **自動化權重調整模組(Automatic Weight adjustment Module)**

本模組隱含機器學習(Machine Learning)的概念，主要是分析 Ontology 中每個類別(Concept)裡已分類學習元件是因為哪幾個類別關鍵詞(Concept Term)的對應(match)而分類至此類別，並自動將這些類別關鍵詞的權重提高，以表示這些類別關鍵詞對此類別的重要度。

- **學習概念擷取模組(LO Concept Extraction Module)**

剖析學習元件的 Important Term Set，並分析 Title 、Keyword、Description 這三個欄位的特性，因為這三個欄位與學習元件內容有較高的相關性。找出這三個欄位中有描述學習元件概念的關鍵詞(Term)，將這些關鍵詞(Term)計算其出現的詞頻，並利用一個以 TF-IDF 為基礎的 LTF-IDF 方法，找出那些在學習元件中足以被視為概念的重要關鍵字，進而擷取出學習概念。

- **學習概念新增模組(LO Concept Insertion Module)**

擷取出的學習概念，先定義學習概念的關鍵詞集合，並利用 Jaccard co-efficient 相似度計算學習概念與領域知識本體(ontology)中每個概念之相似度，找出與學習概念關係最接近的概念，再利用 FCA(Formal Concept Analysis)分析學習概念與領域知識本體(ontology)的階層關係，並將學習概念新增到適合的概念下。

- **領域本體論(Domain Ontology)**

由領域專家所建立，用來輔助學習元件的分類，例如 Java Learning Object Ontology[5]、ACM Computing Classification Ontology[6]。

3.2 學習元件分類模組

本模組會利用重要關鍵字和輔助關鍵字擷取模組所收集到的 Important Term Set 和 Assistant Term Set，去計算學習元件與 ontology 的概念分數(TCS, Total Concept Score)。我們會去制定兩個 TCS 門檻值 λ1(下界)與 λ2(上界)，利用 λ1 與 λ2 來過濾出與可能含有新學習概念的學習元件，以便作新學習概念之擷取。設定好 λ1 與 λ2 的值，我們可以將 TCS 與 λ1、λ2 的關係分成以下三種情況討論：

Case 1. 學習元件之 TCS 小於 λ1，表示此學習元件與本 ontology 無關，則不做任何處理。

Case 2. 若欲過濾的學習元件之 TCS 大於 λ1，但小於 λ2，表示此學習元件與本 ontology 有些許程度的相關且可能含有新學習概念，這種情況也是本論文所要研究的重點。

Case 3. 若欲過濾的學習元件之 TCS 大於 λ2，表示此學習元件與本 ontology 有高度的相關，則直接進行分類。

首先我們先定義學習元件分類演算法所使用到的相關名詞：

(1)Important Term & Concept Term

在 Important Term Set 中的關鍵詞(Term)皆稱為 Important Term，而在 Ontology 中每個類別(Concept)皆有一組關鍵詞用來代表此類別，我們稱此組關鍵詞為類別關鍵詞(Concept Term)。

(2)BCS(Basic Concept Score)

對一個類別而言，只要 Important Term Set 中有任一個關鍵詞對應(match)到類別的類別關鍵詞，我們稱此類別為 Basic Concept，而此類別與 Important Term Set 的對應總分，稱為 BCS。主要是計算一個學習元件後設資料與 Basic Concept 的對應程度，對應分數越高的 Basic Concept 表示此類別越有可能成為這個學習元件的所屬分類，算分的依據是考量每個對應關鍵詞(Match Term)在學習元件後設資料的重要度(LOM Term Weight)和對應關鍵詞在類別的重要度(Concept Term Weight)。

$$BCS^i = \sum_{j=1}^{n} (TW + KW + DW + CW) \times CTW_j^i \quad \textbf{(公式 1)}$$

- LOM Term Weight：

  表示對應關鍵詞在此學習元件後設資料的重要度(權重)。一個關鍵詞對於學習元件的重要度取決於這個關鍵詞是從學習元件後設資料中的哪個欄位擷取出來和關鍵詞在學習元件後設資料的出現頻率而決定的。例如一個關鍵詞是從學習元件後設資料的Title欄位中擷取出來的，那麼它的重要度就比一個關鍵詞從Description欄位中擷取出來的重要度還高。我們針對學習元件後設資料中不同的欄位保留不同的參數，以調整權重。

  $TW = \alpha \times$ Term Frequency in Title Term Vector

  $KW = \beta \times$ Term Frequency in Keyword Term Vector

  $DW = \gamma \times$ Term Frequency in Description Term Vector

  $CW = \delta \times$ Term Frequency in Classification Term Vector

- Concept Term Weight(CTW_j^i)：

  Ontology中，每一個類別皆有一組類別關鍵詞(Concept Terms)用來代表此類別，CTW_j^i 表示第i個類別的第j個類別關鍵詞的權重。CTW_j^i 越高表示此關鍵詞對類別的重要度越高，也越能代表這個類別。

$$MTF_{CT_j^i} = \frac{MatchFre_{CT_j^i}}{\sum_{x=1}^{n} MatchFre_{CT_x^i}} \quad \textbf{(公式 2)}$$

  - $MatchFre_{CT_j^i}$：表示 Ontology 中，第 i 個類別(Concept)的 **Matched Fre**quency Array 中第 j 欄位的內含值。即求出被分類至這個類別底下的學習元件有多少個和第 j 個類別關鍵詞對應。
  - $\sum_{x=1}^{n} MatchFre_{CT_x^i}$：加總 i 類別裡所有類別關鍵詞的 Matched Frequency。

- Inverse Concept Frequency：一個類別關鍵詞屬於越多的類別，表示這個類別關鍵詞較不具代表性，它的 ICF 值較低。相反的，當一個類別關鍵詞屬於較少的類別時，表示這個類別較具代表性，對於所屬的類別，重要度越高。

$$ICF_{CT_j^i} = \log \frac{\#\text{ of concepts in ontology}}{\#\text{of concepts that have } CT_j^i} \quad \textbf{(公式 3)}$$

  - $\#\text{of concepts in ontology}$：表示 Ontology 中的類別個數
  - $\#\text{of concepts that have } CT_j^i$：表示有

多少個類別包含 CT_j^i

W_j^i：表示類別關鍵詞 j 在類別 i 的權重。當 MTF 越高且 ICF 越高，則權重越高。

$$W_j^i = MTF_{CT_j^i} \times ICF_{CT_j^i} \quad \textbf{(公式 4)}$$

- Concept Term Weight CTW_j^i：表示正規化後的類別關鍵詞權重，將權重值限定在 0~1 之間

$$CTW_j^i = \frac{MTF_{CT_j^i} \times ICF_{CT_j^i}}{\sqrt{\sum_{x=1}^{n}\left(MTF_{CT_x^i} \times ICF_{CT_x^i}\right)^2}} \quad \textbf{(公式 5)}$$

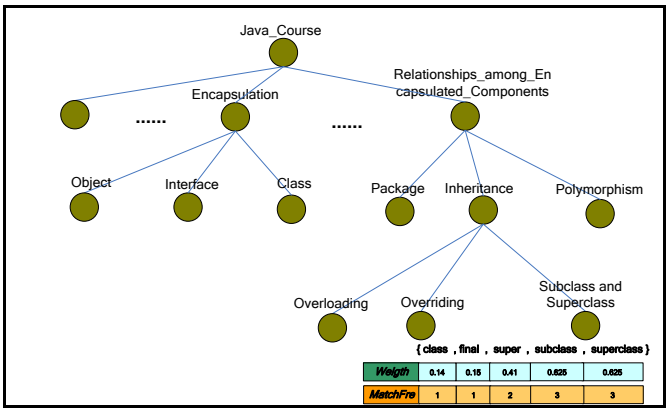


圖 4 概念權重計算範例

我們以圖為例，superclass 這個關鍵詞的 *CTW* 的計算方式如以以下過程所示:

- $$MTF_{CT_j^i} = \frac{MatchFre_{CT_j^i}}{\sum_{x=1}^{n} MatchFre_{CT_x^i}} = \frac{3}{10} = 0.3$$

- $$ICF_{CT_j^i} = \log \frac{\#\text{ of concepts in ontology}}{\#\text{ of concepts that have } CT_j^i} = \log \frac{170}{1} = 2.23$$

- $$CTW_j^i = \frac{TF_{CT_j^i} \times ICF_{CT_j^i}}{\sqrt{\sum_{x=1}^{n}\left(TF_{CT_x^i} \times ICF_{CT_x^i}\right)^2}} = \frac{0.3 \times 2.23}{1.069} = 0.625$$

因此我們可以得知，superclass 這個關鍵詞的 *CTW* 為 0.625。

(3)Assistant Term Match Score

與BCS的算法相同，ATMS計算出Assistant Term Set與類別的命中分數

$$ATMS^i = \sum_{j=1}^{n} ATW \times CTW_j^i \quad \textbf{(公式6)}$$

$$ATW = \sigma \times \text{Term Frequency in Assistant Term Vector}$$

(4)Candidate Concept Score

$$CCS^i = BCS^i + ATMS^i \quad \textbf{(公式7)}$$

(5)Normalized Candidate Concept Score

將Candidate Concept Score正規化，使分數介於0~1之間(n表示候選類別的個數)，具有有降階的作用。

$$NCCS^i = \frac{CCS^i}{\sqrt{(\sum_{j=1}^{n} CCS^j)^2}} \quad \textbf{(公式8)}$$

(6)Hierarchical Impact Score

類別和類別在Ontology上若在相同的Hierarchical Path上，表示這兩個類別具有一定程度的關係，我們給予互相的影響分數。其中m表示候選類別中，有m個候選類別與第i個類別，在相同的Hierarchical Path上。

$$HIS^i = \sum_{k=1}^{m} \frac{NCCS^i \times NCCS^k}{number\ of\ hops\ from\ i\ to\ k} \quad \textbf{(公式9)}$$

(7)Total Concept Score

求出每個候選類別的總分，TCS最高的即為學習元件的所屬類別

$$TCS^i = NCCS^i + HIS^i \quad \textbf{(公式 10)}$$

(8)將 TCS 與 λ1 和 λ2 兩個門檻值做比較，會有以下三種情況

(i). 將所有候選 Concept 的 TCS 由高至低排序，找出 TCS 分數最高的 Concept 作為目標 Concept，若 TCS 分數高於 λ2，則將學習元件放入此 Concept。

(ii). 若 TCS 分數最高的 Concept 之 TCS 分數與介於 λ1 和 λ2 之間，則將學習元件放入學習元件暫存庫中暫存，由概念擷取模組做學習概念的擷取。

(iii). 若TCS分數最高的Concept之TCS分數小於λ1，表示此學習元件與本ontology無關，則不做任何處理。

3.3 概念擷取模組

本模組會先取出學習元件後設資料的 Important Term Set，包含 Title、keyword、Description 等資訊，接著將學習元件利用 LTF-IDF 方法(一個改良的 TFIDF 方法)，擷取出重要的關鍵詞。相關研究證實出現於文件中前段與後段之文句，因具有描述主題與總結主題之詞彙，故此兩部分之詞彙其重要性較高。根據一個詞(term)出現的位置，可以去判斷詞的重要程度[7]。所以本研究根據詞的位置，給予詞不同的權重，並結合了 TF-IDF 方法，產生了一個 LTF-IDF(Location weight TF-IDF)，希望可以提高擷取關鍵詞的精確度。 LTF-IDF(Location weight TF-IDF)計算公式，如公式 11 所示：

$$W_i = LTF(t_i, d) * IDF(t_i) \quad \textbf{(公式 11)}$$

如式子(3-11)中W_i為計算詞彙t_i的權重，其中$LTF(t_i, d)$為詞彙t_i在文件 d 中所出現的位置加權詞頻，$LTF(t_i, d)$計算方式如公式 12 所示：

$$LTF(t_i, d) = TF(t_i, d_1) + TF(t_i, d_f) + TF(t_i, d) \quad \textbf{(公式12)}$$

如公式12中$TF(t_i, d_1)$為詞彙t_i在文件d中第一段落所出現的頻率，$TF(t_i, d_f)$為詞彙t_i在文件d中最後一段落所出現的頻率，$TF(t_i, d)$為詞彙t_i在文件d中所出現的頻率，公式11中$IDF(t_i)$如公式12：

$$IDF(t_i) = log\frac{|D|}{|\{d_i \epsilon D\}|}) \quad \textbf{(公式13)}$$

如公式13中|D|為所有文件的總篇數，$|\{d_i \epsilon D\}|$為有出現詞彙t_i的文件篇數。

學習概念擷取流程 ：

1. 當學習元件暫存庫裡，學習元件的數量累積至一定數量時，分析每篇學習元件的 LOM(Learning Object Metadata)。
2. 擷取 LOM 裡的 Important Term，並計算 Important Term 裡每個字詞(Term)的 LTF-IDF 分數，並將每篇 LTF-IDF 分數最高的關鍵字列出來。
3. 若有兩篇以上的學習元件之 LTF-IDF 分數最高的關鍵字，為同一字的話，將具有相同關鍵字的學習元件，視為一**候選概念**的集合。
4. 當候選概念(Candidate Concepts)集合中，學習元件數量累積超過一定數量時，則將候選概念視為正式概念。
5. 定義正式概念的概念關鍵字集合，再將正式概念送到學習概念儲存庫裡，由概念新增模組做概念新增的動作。

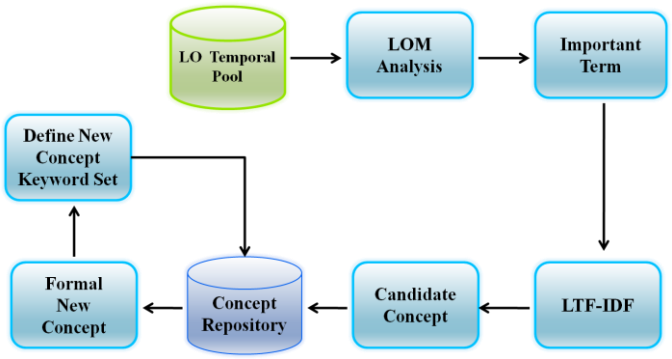


圖 4 概念擷取流程圖

3.4 概念新增模組

當我們擷取出一個正式概念後，必須找出本體論中與此正式概念關係最接近的概念，因此如何尋找出最接近的概念變成一門重要的課題。所以本論文提出了利用 Jaccard co-efficient 相似度計算公式，去計算概念之間的相似度。Jaccard 係數是在衡量資料交集(Transaction Data Set)時最為廣泛使用的相似度量測標準，計算本體論中的每個概念與此正式概念的相似度，再將這些和正式概念關係相近的概念與正式概念，利用 FCA(Formal Concept Analysis)建構出概念點陣(Concept Lattices)，判斷正式概念與本體論的階層關係，並將此正式概念新增至本體論中。

Jaccard co-efficient 相似度計算公式:

$$\text{Jaccard co} - \text{efficient} = \frac{|X \cap Y|}{|X \cup Y|} \qquad \textbf{(公式14)}$$

如式子(3-14)，若有 X、Y 兩個集合，X={a} Y={a ,b} ，則 X 與 Y 之相似度為 1/2。

我們將定義後的 New Concept 的關鍵字集合，與 JLOO 裡的每個概念之關鍵字集合做 Jaccard co-efficient 相似度的計算。 我們以一個實例來說明：

假設有一正式概念叫 **Superclass and subclass**，欲新增至 JLOO，其關鍵字集合為【 subclass, superclass, super, final, class, Inheritance, extend】，如圖 5 所示。

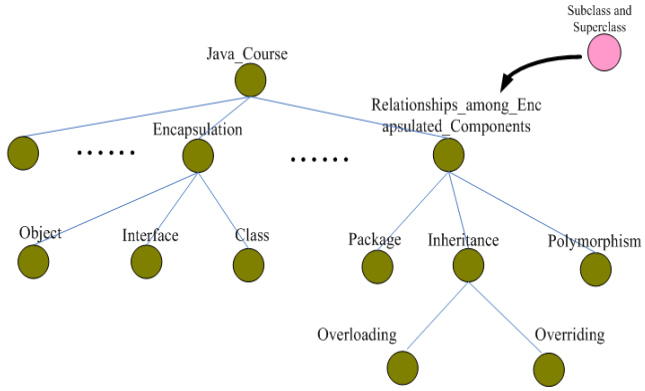


圖 5 新增正式概念之示意圖

我們將 superclass and subclass 其關鍵字集合與 JLOO 裡的其他概念的關鍵字集合，做 Jaccard co-efficient 相似度的計算，結果如表 2 所示。

表 2 概念間相似度與關鍵字集合

概念名稱	相似度	關鍵字集合
inheritance	0.285	inheritance, extend
overriding	0.25	overriding, inheritance, extend
overload	0.25	overload, inheritance, extend
constructors	0.125	constructor , class
abstract_class	0.125	abstract , class
class	0.111	class, instance, encapsulation

由表 2，我們可以得知，在 JLOO 中與 superclass and subclass 關係比較接近的概念，以及與 superclass and subclass 的相似度，並利用這些概念的關鍵字集合，如表 2，當作 FCA(Formal Concept Analysis) 中建構概念本文(Concept Context)的屬性，如圖 6，進而建構出概念點陣(Concept Lattice) ，如圖 7，可以得知 superclass and subclas 這個概念應該新增至 inheritance 此概念下。

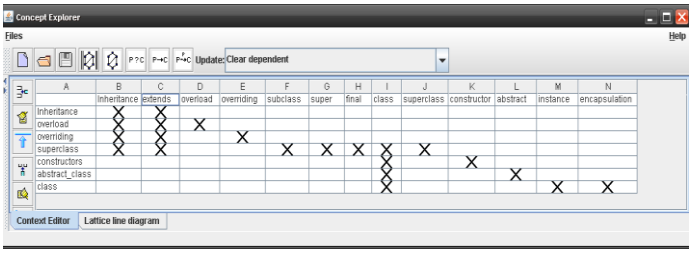


圖 6 概念本文(Concept Context)

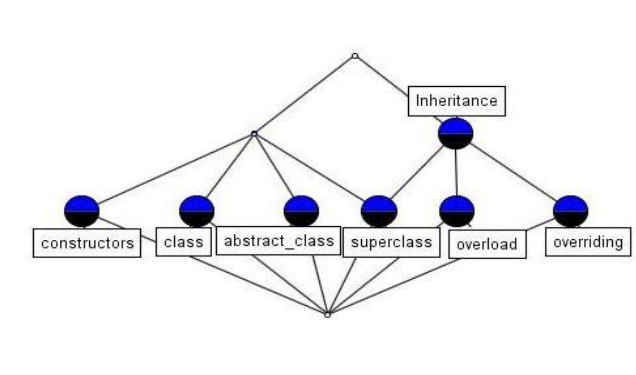


圖 7 概念點陣(Concept Lattice)

如何將正式概念新增回本體論：

1. 將定義後的 New Concept 的關鍵字集合，與 JLOO 裡的每個概念之關鍵字集合做 Jaccard co-efficient 相似度的計算。
2. 利用 Jaccard co-efficient 相似度，挑選出與正式概念關係接近的概念。
3. 將這些概念的關鍵字集合，與正式概念的關鍵字集合當作每個概念的屬性，並建立出概念本文。
4. 利用概念本文建構出概念點陣(Concept Lattices)，分析正式概念與本體論之間的階層關係。
5. 將正式概念新增到適當的本體論概念下。

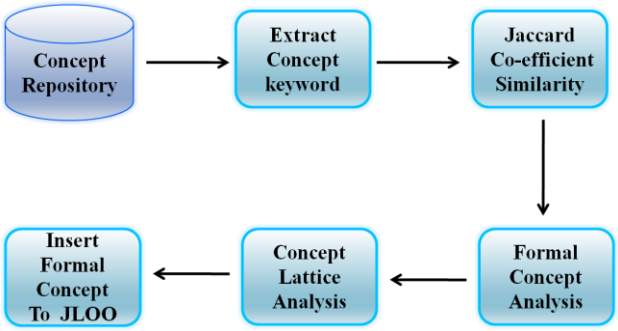


圖 8 新增正式概念流程圖

四、 實驗設計與結果分析

實驗一:制定出兩個 TCS 門檻值λ1(下界)與λ2(上界)的值。

(1). 收集與 Java 課程無關的學習元件 120 篇
(2). 收集與 Java 課程相關的學習元件 120 篇
(3). 收集與 Java 課程相關但學習概念並不存在 JLOO 裡的學習元件 120 篇。

利用(1)和(2)，與 Java 程式設計課程相關和無關的學習元件各 120 篇，分別計算出其 TCS(Total Concept Score)平均值，如表 3 所示：

表 3 TCS 平均值

	JLOO相關(120篇)	JLOO無關(120篇)
TCS平均值	1.27	0.27

由以上表格得知， λ1 與λ2 的值應介於 0.27 和 1.27 之間。分別利用(1)~(3)這些學習元件分別來測試λ1 的過濾準確率 P(λ1) 、 λ2 的過濾準確率 P(λ2)、(λ1、λ2)區間的過濾準確率 P(λ1、λ2)，進而求出整體過濾準確率，公式如下所示：

$$整體過濾準確率(precision)=\frac{P(\lambda1)+P(\lambda2)+P(\lambda1、\lambda2)}{3}$$

如圖 8 實驗結果，λ1 與 λ2 區間大小的選擇，區間太大會有誤判的情況，區間太小會造成準確率太低的情況，根據本研究實驗測試，λ1 與 λ2 區間為 0.6 會有較好的成效，如圖 4-3 所示，當 λ1=0.4 而 λ2=1.0 時，過濾準確率為 72%，所以(λ1 , λ2)=(0.4 , 1.0)，較為恰當。

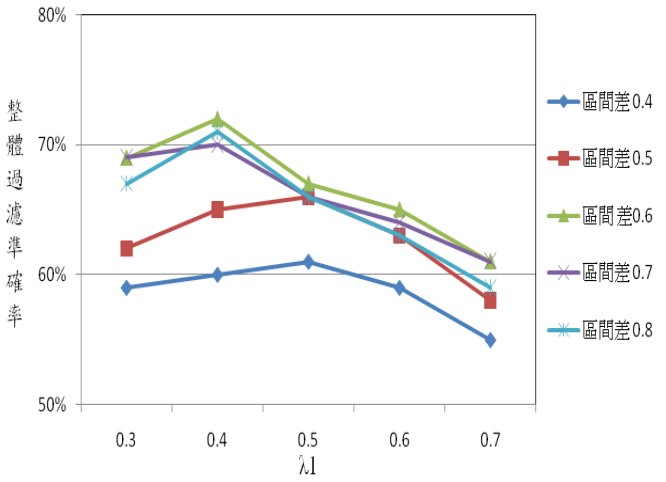


圖 8 實驗一結果

實驗二:本研究方法(LTF-IDF)與其他關鍵字擷取方法(TF-IDF)之評比

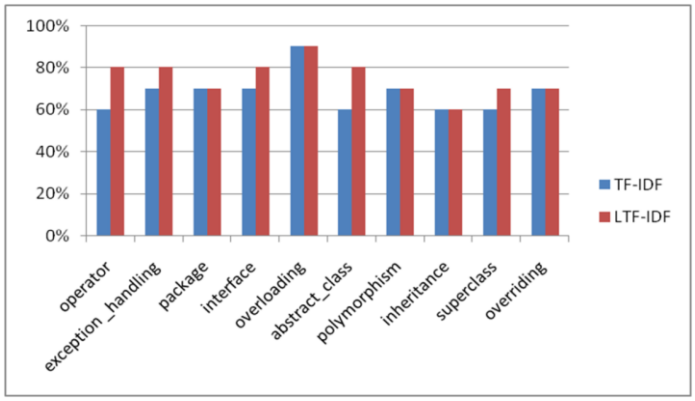
使用 Java Learning Object Ontology (JLOO) [10]中所提供的 100 份 Java 學習元件，共包含了 10 個學習概念，每個概念各 10 份學習元件。當作實驗的輸入目標資料。接著使用 LTF-IDF 以及 TF-IDF 這兩種關鍵字擷取方法，分別計算它們所產生的最重要的關鍵字，與經過領域專家定義的正確關鍵字做比對，是否正確。

而這些正確的關鍵字集合，是經由專家用人工的方法從 JLOO 的 100 份 Java 學習元件所挑出的具課程代表性的關鍵字，並且再使用資料挖掘中很常被使用的準確率(Precision)來加以觀察。準確率表示所輸出的資料集合中，正確資料的比例是多少。準確率 (Precision)計算公式如下：

$$Precision = \frac{正確擷取出關鍵字之文件數}{所有文件數} \quad \textbf{(公式15)}$$

如圖 9 所示，若僅使用 TF-IDF 的方法來尋找關鍵字的話，將會使得所得的準確率較低，由於這些重要的關鍵字大多是出現在文章的前後段落部分，因此本研究方法提出了 LTF-IDF 方法來尋找關鍵字，去加重前後段落關鍵字的權重，使它的準確率能有效的提升。

表 3 TF-IDF 與 LTF-IDF 方法擷取概念之比較



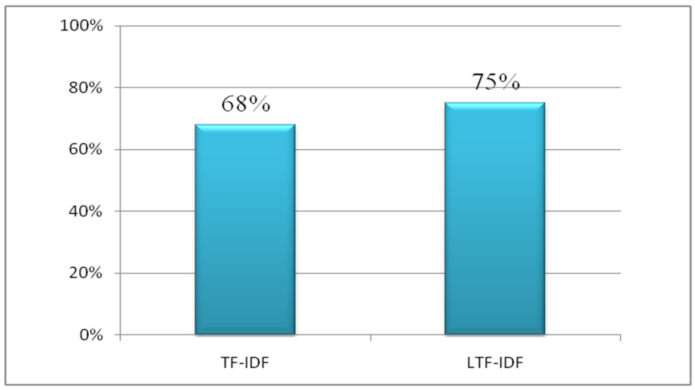


圖 9 LTF-IDF 與 TF-IDF 擷取概念精確度之比較

實驗三: 本研究方法與其他的文件分類方法成果之比較。

將本研究方法(Automatic Ontology Expansion Mechanism)，分別對 VSM (以向量空間模型為基礎之分類方法)、Latifur R. Khan [8]所提出的音樂物件的分類方法以及陳偉洲[1]所提出的以 Java Learning Object Ontology 為基礎的學習元件分類法(OALOC) 做一個比較。其中，OALOC 與 Latifur 皆是以一個已知架構為基礎的分類方法，而本研究以及 VSM 則是僅根據對學習元件本身內容的分析而求得想要的結果。

如圖10所示，是本研究方法與其他的文件分類方法所做的一個比較表，隨著學習概念的數量增加，本研究方法在分類準確度有逐漸的提升，主要的原因是本研究所提出的機制會自動產生一個概念類別，相較於其他的方法，本研究的分類準確度較高。

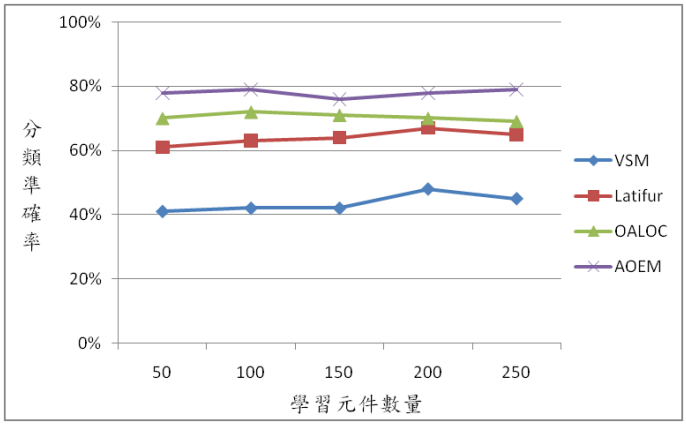


圖 10 分類準確率

五、結論與未來展望

由於電腦的普及與全球資訊網的蓬勃發展，使用者經由網路來搜尋自己有興趣的課題，而達成學習的目的，已經是現今最常見的趨勢之一。雖然使用網路搜尋適合的教材片段、學習元件，來取代傳統的獨立製作教材方式，可以節省人力和成本的花費。然而，由於這些網路上的素材通常都是分散且組織零散的，難以達到再利用的目的。

而且，面對這些豐富、大量的學習元件，教學者卻必須以人工的方式定義學習元件的主題概念，非常浪費人力與成本。因此，本論文提出一個以正規概念分析為基礎之本體論自動擴展機制，藉由 LTF-IDF 找出那些能夠代表學習元件的主要關鍵字，當做學習概念，再使用 Jaccard co-efficient 相似度計算公式來計算學習概念與本體論概念之間的關聯程度，之後再將學習概念新增利用本研究所提出的機制，的確能有效的擷取出學習概念，並能找出與學習概念關係相近的概念，自動地擴展本體論。將有助於使用者，管理及學習所需要的知識。

根據實驗後的結果發現，本研究所提出的學習概念擷取機制，雖然能夠有效的擷取出學習概念，但對於學習概念新增回本體論方面上，在本體論中『概念階層』的判斷，也還有進步的空間。而且當人為建構本體論的時候，會依據個人主觀的因素來作建構，因此每個人建構出來的本體論都不太一樣，在建立『概念階層』的時候，我們無法明確的得知新的『概念』是不是就是最底層的『概念』，以後會不會有新的『概念』衍生出來。 所以在建立『概念階層』的時候，不論是利用手動或是自動的方式，都很難做到完全的正確。

除此之外，由於本研究是屬於比較一般化的方法，前處理的部份也僅用了以統計為基礎的方法計算出重要的學習元件關鍵字，所以在關鍵字萃取方面的準確率無法盡善盡美。

在未來的研究中，如果能再加上自然語言處理對關鍵字之間的關聯性加以分析，以及能夠分辨出學習元件關鍵字之間的相異領域的方法，相信能夠得到更佳的結果。

六、致謝

本研究承蒙國科會計畫
NSC95-2221-E-006-158-MY3經費部分補助，特此感謝。

參考文獻

[1] 陳偉洲, "基於本體論之學習元件自動分類演算法",成功大學工程科學研究所, 2006.

[2] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout, "ENABLING TECHNOLOGY FOR

[3] F. Sebastiani, “Machine Learning in Automated Text Categorization”, ACM Computing Surveys, Vol.34, No.1, March 2002, pp.1-47.

[4] N.F.Noy and D.L.Mcguinness, ”Ontology Development 101: A Guide to Creating Your First Ontology,” Stanford Knowledge System Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880,Mar.2001.

[5] M. C. Lee, D. Y. Ye, and T. I. Wang, ”Java Learning Object Ontology”, The 5th IEEE International Conference on Advanced Learning Technologies, pp.538-542, July 2005, Kaohsiung, Taiwan.

[6] The ACM Computing Classification System [1998Version], http://www1.acm.org/class/1998/.

[7] F. Chen and K. Han and G. Chen, ”An Approach to Sentence-Selection-Based Text Summarization” , Oct. 2002

[8] Khan, L., "Ontology-based Information Selection," Ph.D. Dissertation, Department of Computer Science, University of Southern California, 2000.