

Automatic Concept-Based Answer-Finding System

Student: Yi-Fen Chen

Advisor: Dr. Hao-Ren Ke, Dr. Wei-Pang Yang.

Institute of Computer and Information Science
National Chiao Tung University

ABSTRACT

In this thesis, we propose an automatic concept-based answer-finding system (ACAF) that exploits LSA as its core technology. Users issue their new questions into the ACAF system by natural language, and the system will return suitable answers from the question-and-answer set (QA set). To accomplish its task, ACAF employs machine learning techniques to learn three kinds of knowledge: the relationship between question keywords and answer keywords, conceptual space, and answer-type knowledge. LSA is used learn the relation matrix between question keywords and answer keywords; LSA is used construct the conceptual space as well. In addition, a probabilistic model is employed to train the relations between interrogatives and answer types. According to these three knowledge bases, ACAF calculates the similarity of a new question and the answers in the QA set. ACAF not only compares keyword-similarity but also retrieves the concepts of the new question. In this manner, an automatic answer-finding system can be promoted to the semantic level. ACAF was evaluated by using a QA set about basketball and baseball rules, and average TRDR of 83.87%, average precision of 36.4% and average recall of 44.2% were achieved. Compared to ACAF_w, the average increasing range of TRDR, precision, and recall are 17.75%, 8.53%, and 18.37% respectively.

Keywords: Automatic Answer-Finding System, Latent Semantic Analysis, Conceptual Space, Answer Type Detection

概念式自動問答探索系統

研究生: 陳意芬

指導教授: 柯皓仁博士, 楊維邦博士

國立交通大學資訊科學研究所

摘要

本論文提出一套以潛在語意分析 (LSA) 為核心技術的概念式自動問答探索系統。自動問答探索系統能讓使用者以自然語言的方式輸入新問題，系統會從歷史問答集中找出符合的答案。此套概念式自動問答探索系統首先學習三種知識庫：問題詞鍵和答案詞鍵間的關係矩陣、概念空間知識庫和答案類型判別知識庫。其中，問題詞鍵和答案詞鍵間的關係矩陣以及建構概念空間知識庫係利用潛在語意分析學習而得，而疑問詞鍵與答案類型的關係則是以一機率模組學習而得。依據此三種知識庫，概念式自動問答探索系統會比對新問題與問答集中的答案之詞鍵相似度、概念描述相似度和答案類型相似度，不僅透過詞鍵比對，更擷取出問題的概念描述，以期將自動問答探索系統提昇至語意層面。實驗中，先人工建構了關於籃球和棒球規則的問答組，概念式自動問答探索系統的效能在排序準確度 (TRDR) 評估上平均可達 83.87%，準確率平均可達 36.4%，而查全率平均可達 44.2%；較非概念式的自動問答系統的效能在排序準確度、準確率和查全率的評估上，平均增幅分別為 17.75%、8.53%、18.37%。

關鍵字：自動問答探索系統、潛在語意分析、概念空間、答案類型的判別

致謝

本論文能從無到完成，首先得感謝我的兩位指導教授，柯皓仁老師與楊維邦老師。在他們的指導下，開拓了我在資訊領域的眼界，訓練我獨立研究的能力，也培養我團隊合作時的技巧，不僅提供課業與研究方面的解惑，也在我日常生活上給予幫助與指導。

謝謝交大資科資料庫實驗室的學長姊、學弟和同學們，因為有你們的集思廣義才讓我的碩士論文得以順利進行，讓我的研究生生活增添了色彩。此外，還要感謝交大圖書館數位圖書資訊組的夥伴們在定期論文研討時給我的建議與指教。

最後，要感謝關心我的家人和朋友們，你們的支持和鼓勵讓我能全心全意地投入我的學業與研究中，謝謝你們在我情緒低落時給我安慰，謹以這篇小小的論文獻給我最愛的你們。

May 19, 2003

目錄

英文摘要	I
中文摘要	II
致謝	III
目錄	IV
圖目錄	V
表目錄	VI
方程式目錄	VII
第一章 簡介	1
第一節 自動問答探索系統	1
第二節 研究動機	5
第三節 研究目的	6
第四節 論文架構	6
第二章 相關研究工作	7
第一節 以統計技術為基礎的自動問答探索系統	8
第二節 潛在語意分析 (Latent Semantic Analysis)	12
第三節 概念空間 (Concept Space) 的建立	16
第四節 答案類型 (Answer Type) 判別	18
第三章 概念式自動問答探索系統	27
第一節 系統架構	27
第二節 問答組詞鍵關係的學習機制	30
第三節 概念空間的建構	32
第四節 答案類型判別知識的學習	38
第五節 問答探索機制	40
第四章 實驗結果分析與評估	44
第一節 實驗問答集與實驗設計	44
第二節 評估方法	45
第三節 知識庫建構評估	47
第四節 ACAF 查詢結果評估	55
第五章 結論與未來研究方向	59
第一節 結論	59
第二節 未來研究方向	60
參考文獻	62
附錄：ACAF 搜尋流程 (以常問問答集為例)	66

圖目錄

圖 1：自動問答系統架構圖	2
圖 2：自動問答探索系統架構圖	4
圖 3：相關研究工作發展	7
圖 4：利用 LSA 建構概念空間流程圖	18
圖 5：ACIRD 學習模組	20
圖 6：ACIRD 詞鍵語意網路示意圖	23
圖 7：ACAF 系統架構圖	28
圖 8：ACAF 學習架構圖	29
圖 9：問答組詞鍵關係學習機制圖	31
圖 10：ACAF 問題概念空間建構流程圖	33
圖 11：平均連結分群法之演算法	35
圖 12：ACAF 訓練問答集 (運動規則問答集)	45
圖 13：ACAF 訓練問答集 (常問問答集)	66
圖 14：ACAF _{WC} 的搜尋結果	67
圖 15：答案類型判別知識庫 (常問問答集)	68
圖 16：ACAF 的搜尋結果	68

表目錄

表格 1：NSIR 答案類型列表	24
表格 2：NSIR 疑問詞與各個答案類型的關係表	25
表格 3：NSIR 答案類型判別的 Template 實列表	25
表格 4：詞鍵相似度矩陣實例	36
表格 5：表格 4 套用平均相似度每一步驟的分群結果變化	36
表格 6：答案類型與可能出現的疑問詞鍵	40
表格 7：維度約化對問題與答案詞鍵關係建構的影響 (運動規則問答集).....	48
表格 8：維度約化對問題與答案詞鍵關係建構的影響 (一般參考問題問答集).....	48
表格 9：維度約化對問題與答案詞鍵關係建構的影響 (常問問答集).....	49
表格 10：維度約化和門檻值對概念空間建構品質的評量表 (運動規則).....	52
表格 11：維度約化和門檻值對概念空間建構品質的評量表 (一般參考問題).....	53
表格 12：維度約化和門檻值對概念空間建構品質的評量表 (交大常問問題).....	54
表格 13：答案類型知識庫建構評估表	55
表格 14：ACAF 查詢結果效能評估表 (運動規則問答集).....	56
表格 15：ACAF 查詢結果效能評估表 (一般參考問題問答集).....	57
表格 16：ACAF 查詢結果效能評估表 (常問問答集).....	57

方程式目錄

方程式 1：查詢句與文件的相似度計算	8
方程式 2：改良式 TF-IDF 模組 IDF 改善公式	9
方程式 3：問答集中問題與答案詞鍵間的交互資訊值	10
方程式 4：翻譯因子模組表示法	11
方程式 5：潛在因子計算公式	11
方程式 6：EM 學習機制	12
方程式 7：利用 LSA 計算二詞鍵的相似度	18
方程式 8：ACIRD 最明確節點的詞鍵支持值 $\sup_{t_i,C}$	21
方程式 9：ACIRD 其它節點的詞鍵支持值 $\sup_{t_i,C}$	22
方程式 10：關連性規則 $t_i \rightarrow t_j$ 的信任值 $conf_{t_i \rightarrow t_j}$ 和支持值 $\sup_{t_i \rightarrow t_j}$	22
方程式 11：ACIRD 微調分類知識	23
方程式 12：問題詞鍵 w_q 與答案詞鍵 w_a 的共同出現關係計算公式	32
方程式 13：詞鍵相似度計算公式	34
方程式 14：疑問詞鍵 t_j 的問題對某個答案類型 AT_i 的可能性	40
方程式 15：答案詞鍵 w_a 與問題 q 的相關度	41
方程式 16：問題 q 與答案 a 詞鍵間的關係強度	41
方程式 17：問題 q 與概念 c 的相似度	42
方程式 18：問題 q 與問題 q' 在概念相似度	42
方程式 19：預選答案篩選相似度	42
方程式 20：問題 q 屬於答案類型 AT 的可能性公式	42
方程式 21：問題 q 與問題 q' 在答案類型上符合強度	43
方程式 22：問題 q 和答案 a 的相似度	43
方程式 23：TRDR 計算公式	46
方程式 24：準確率計算公式	46
方程式 25：查全率計算公式	46
方程式 26：inter-cluster 計算公式	50
方程式 27：intra-cluster 計算公式	50

第一章 簡介

第一節 自動問答探索系統

隨著資訊科技的進步與數位化技術的迅速發展，有越來越多的數位化資訊藉由網路快速廣泛地傳播，讓網際網路儼然成為一個資訊的寶庫，資訊已垂手可得！透過網路雖然可以快速地獲取大量資訊，但對人們來說，面對過於龐大的資訊卻是沉重的負擔，因此過濾不需要的資訊，進而正確判斷出有用的資訊，已成為重要的課題！搜尋引擎的相關研究因而嶄露頭角。

一般資訊擷取系統（包括網際網路上的搜尋引擎[Google03] [Yahoo!03] [Yam03]）的操作模式大致如下：首先使用者輸入一筆查詢，這筆查詢通常由一些關鍵詞組成，之後資訊擷取系統就會根據使用者輸入的查詢，回傳與該查詢相關的文件，這些回傳的文件在傳統的資訊擷取系統中，就被認為和使用者資訊需求相關。此種資訊擷取系統面臨了二個主要的問題：(1)使用者必須遵循各個搜尋引擎的特殊用詞規則建構查詢；(2) 使用者的資訊需求往往是針對某一個問題的解答，資訊擷取系統所回傳的相關文件，僅僅與使用者輸入的查詢相關(例如：文件中包含查詢的關鍵詞)，但大部分卻不是問題的解答[Radev01][Radev02]。例如：當使用者想要知道「台灣最高的山峰是什麼？」時，其查詢可能包含「台灣」、「最高」、「山峰」等關鍵詞，搜尋引擎會找到這些與關鍵詞相關的文件，這些相關文件中只有一部分擁有該問題的解答，因此使用者必須經由逐一瀏覽每一篇回傳的文件後，才能找到「台灣最高的山峰是什麼？」的正確答案是「玉山」。

為了解決上述的問題，自動問答系統 (Automatic Question Answering System)的相關研究便如雨後春筍般陸續出現[Prager00] [Pasca01] [Kwok01] [Radev01] [Radev02] [Pinto02] [Dumais02]。自動問答系統能讓使用者直接使用

自然語言輸入問題，系統則從文件集中找出正確的解答。例如：當使用者輸入的問題為「台灣最高的山峰是什麼？」時，自動問答系統就會找出該問題的答案是「玉山」。

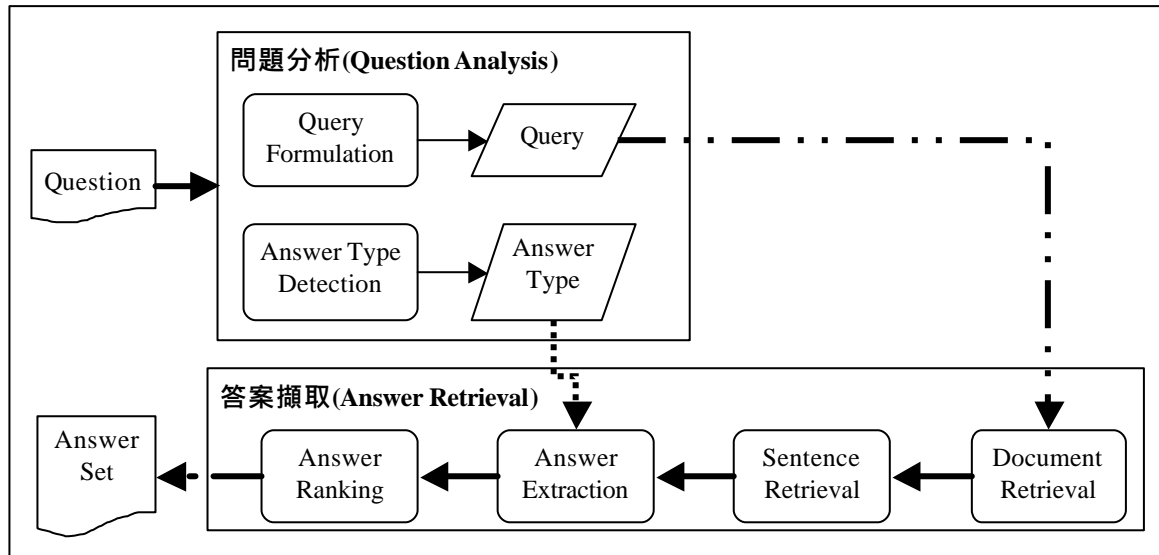


圖 1：自動問答系統架構圖

綜合自動問答系統的相關研究，可以整理出如圖 1 的自動問答系統架構圖，自動問答系統主要分為二個階段，首先是「問題分析 (Question Analysis)」階段，將使用者的問題經過查詢建構 (Query Formulation) 和答案類型判別 (Answer Type Detection)，分別產生適合資訊擷取系統的查詢與使用者問題的答案類型，這些答案類型指出問題的解答可能是「人名」、「組織名」或是「時間」等答案類型；其次是「答案擷取 (Answer Retrieval)」階段，這一個階段主要分為「文件擷取 (Document Retrieval)」、「語句擷取 (Sentence Retrieval)」、「答案抽取 (Answer Extraction)」、「答案評等 (Answer Ranking)」等步驟，這些步驟會因系統的不同而有所刪減。「文件擷取」的步驟是將之前產生的查詢輸入搜尋引擎以找出與問題相關的文件，這些文件中有一部分可能會包含問題的解答；而「語句擷取」是由每一篇相關文件中判斷出答案可能存在的語句，並找出最有可能包含答案的相關語句；在找出答案可能存在的語句後，會根據問題分析

階段所產生的答案類型，再經過「答案抽取」的步驟抽取出可能是答案的詞鍵片語；最後再經過「答案評等」使最正確的解答能在回傳的答案集中佔有較前面的排序位置，讓使用者得以快速地找到問題的解答。

另一方面，受到網際網路的衝擊，圖書館的諮詢台已由面對面服務轉變為無時間空間限制的網路服務。然而在圖書館諮詢台的知識工作者，每天仍要處理許多讀者的問題，這些問題的重覆性高，因此當使用者發問時，若能利用自動問答系統找尋曾經發問過的類似問題並回覆給使用者，而不用再讓知識工作者親自回答，便能節省知識工作者的時間，提高工作效率。

為了達成這一個目的，為圖書館諮詢台建立一個自動問答系統似乎是一個可行的辦法。但是，對圖書館的諮詢台來說，使用者提問的問題領域十分廣泛，因此為圖書館諮詢台所發展的自動問答系統應該要能處理各個領域的問題。此外，在自動問答系統相關研究中可以發現：這些自動問答系統雖然已由回答某一特殊專業領域 (Specific Domain) 的相關問題發展到可以回答一般性非專業領域 (General Domain) 的相關問題，然而所找到的答案準確率卻令人不夠滿意。因此將自動問答系統套用在圖書館諮詢台上，並不能準確地回覆使用者的問題！

為了使自動問答系統能夠適用於圖書館諮詢台，本論文輔以搜尋引擎的機制，建構一個不同於前段所提的自動問答系統，為有所分別，我們稱此系統為自動問答探索系統 (Automatic Answer-Finding System) [Berger00]。

自動問答探索系統是將圖書館諮詢台的知識工作者從前回答過的問題之回覆視為文件集，這種文件集有別於傳統資訊擷取系統的文件集，它不但包括了使用者所提問的歷史問題，更包含相對應的回覆，我們稱此文件集為問答集 (Question and Answer Set, QA Set)。而自動問答探索系統會從問答集中找出能回

答使用者新問題的回覆，這些回覆就是問答集中的答案部分。圖 2 為自動問答探索系統的架構圖，與搜尋引擎相似都是回傳答案的整篇文件內容，而不是像自動問答系統是回傳答案的關鍵詞鍵。例如：使用者輸入的問題是「台灣最高的山峰是什麼？」，自動問答系統回傳的是「玉山」這個答案的關鍵詞鍵；一般搜尋引擎回傳的是有關台灣山峰等的相關文件，至於答案只在一些文件中會出現；而自動問答探索系統回傳的是答案文件，在這篇答案文件中會有一部分會講述到台灣最高的山峰是玉山，此外還可能會有玉山的介紹或是台灣其它山峰的簡介。以圖書館諮詢台此一應用來說，我們認為自動問答探索系統是比較適合的。

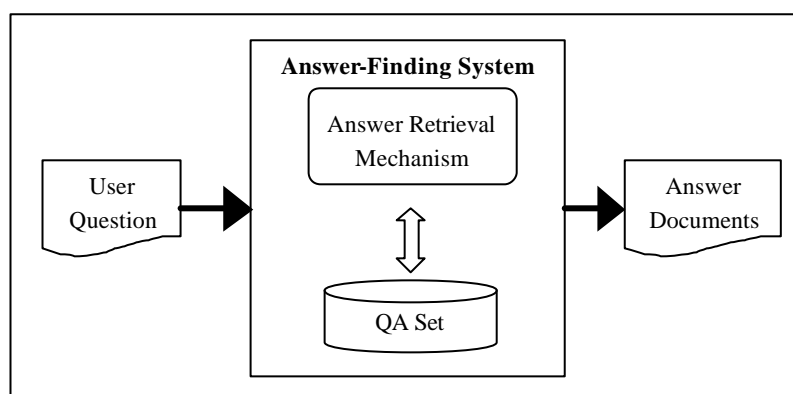


圖 2：自動問答探索系統架構圖

[Berger00]所提出的自動問答探索系統主要是探勘在問答集中問題與答案的相關性，利用使用者提出的新問題和問答集中問題與答案間的語彙關係，來擷取問答集中符合新問題用詞的答案。但是往往使用者的新問題和問答集中問題與答案在所使用的詞鍵上並不完全一樣，因此若只是參考問題與答案間的語彙關係，當新問題與問答集中的某個問題是類似的，但因使用的語彙和問答集中的問題與答案不同，就無法找到符合新問題的回覆，這會降低系統的擷取效能。故本論文除了探勘出問答集中問題與答案間的語彙關係外，還利用潛在語意分析 (Latent Semantic Analysis, LSA) 的技術導入問題的概念空間 (Concept Space)，只要新問題與問答集中的問題擁有相同的概念，問答集中的問題所相

對應的答案就是符合新問題的回覆，這樣就能克服新問題與問答集中的問題使用不同語彙的狀況。此外，我們也納入了在自動問答系統中對答案類型 (Answer Type) 判別的步驟，讓系統所找到的答案內容能符合新問題問及的意圖，例如：如果新問題是「台灣最高的山峰是什麼？」，問的是山峰的名字，那麼答案內容應該有山峰的名字，而不能只是做台灣山峰地形的介紹。

總結上述，本論文提出了一個「概念式自動問答探索系統 (Automatic Concept-Based Answer-Finding System, 簡稱 ACAF)」, ACAF 結合問題與答案問語彙關係、概念式查詢和答案類型判別三方面的知識，期能更準確地擷取出使用者新問題的解答。

第二節 研究動機

一個自動問答探索系統除了要了解使用者的問題以外，還必須了解問答集中每一組問題與答案(在這裡我們稱為問答組)所表達的知識內容，進而根據問答集的知識內容，針對每一個新問題擷取出最佳的答案。

以往自動問答探索系統的相關研究，評估答案的主要依據仍是問題與答案語彙上的比對，但我們認為要能找到問題的真正解答，必須要理解問題所詢問的概念和答案類型，進而比對問題與答案在概念傳達和答案類型的異同。藉由問題與答案詞彙的比對，再佐以傳達之概念和答案類型的異同，期望將自動問答探索系統提昇至概念式自動問答探索系統，使系統能更準確地找到使用者問題的真正解答。

另一方面，答案類型判別的相關技術主要都是處理英文問題，由於英文和中文特性上的不同，因此既存的系統若要處理中文問題和文件，就必須在答案類型判別的技术上有所更動。

本論文的研究動機就是希望設計一套自動問答探索系統，並建構概念空間和答案類型的判別知識，更改良答案類型的判別技術，使系統能應用到中文。藉由導入概念空間與答案類型的判別，使此自動問答探索系統不僅能成功運用於探索中文問題的解答，更將此系統提昇至概念式自動問答探索系統。

第三節 研究目的

本論文之研究目的在於探討自動問答探索系統的相關技術，著重如何建立概念空間，並以此概念空間為基礎建立概念式自動問答探索系統，使系統對使用者提問的問題能找到正確的解答。

本論文的主要方法是運用潛在語意分析 (Latent Semantic Analysis, LSA) 建構問答集所表達的知識內容，首先建構問答集的問題和答案的隱含語彙關係，其次是建構問題的概念空間；此外，本論文也採用了機率模組，用來判別問題的答案類型。最後，針對我們所建構的概念式自動問答系統設計實驗，以顯示概念空間在自動問答探索系統上的可行性。

第四節 論文架構

本論文首先在第二章大略介紹各項相關研究，包括自動問答探索系統、潛在語意分析、概念空間的建構技術和答案類型的判別；接著在第三章闡述我們提出的概念式自動問答系統 (Automatic Concept-Based Answer-Finding System, ACAF) 中，各主要模組的功能及採用的技術，分別介紹系統架構、學習機制和問答探索 (Answer-Finding) 機制；第四章是針對 ACAF 進行實驗，評估學習和探索效能，以證明 ACAF 的可行性；最後第五章總結本論文，並探討未來的研究發展方向。

第二章 相關研究工作

本章說明相關的研究工作。對本論文概念式問答系統採用的技術而言，相關研究工作主要分三方面：

- 自動問答探索系統：[Berger00] [Pasca01] [Dumais02] [Pinto02]
- 概念空間建立：[Park96] [Bellegarda96] [Chung99] [Aggarwal01] [Fu01] [Lin01] [Sugumaran02]
- 答案類型判別：[Prager00] [Pasca01] [Zelikovitz01] [Radev02] [Lin02]

圖 3 依年份及技術整理了相關研究的發展，而粗體的部分是 ACAF 主要採取的方法。

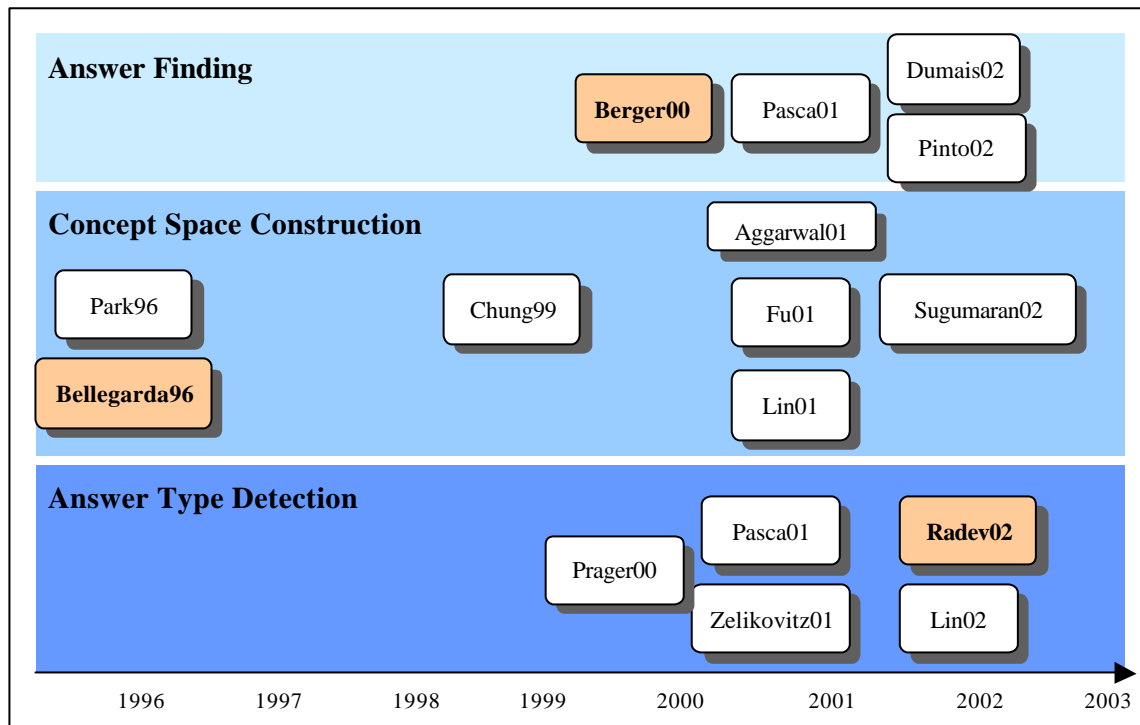


圖 3：相關研究工作發展

以下就這三方面逐一介紹，首先在第一节介紹[Berger00]以統計技術為基礎所提出的五種自動問答探索系統模組。接著第二节說明在傳統資訊擷取中探勘詞鍵潛在語意的技術：潛在語意分析 (Latent Semantic Analysis, LSA) [Landauer98]。而為了讓本論文的自動問答探索系統能結合概念搜尋，成為一個

概念式自動問答探索系統，因此第三節簡介建構概念空間與概念式搜尋引擎的相關研究。最後為了讓系統能精準地找出問題的答案，本論文導入使用者問題的答案類型判別，所以在第四節中介紹關於答案類型判別的發展。

第一節 以統計技術為基礎的自動問答探索系統

[Berger00]根據從資訊擷取所得到的經驗，針對自動問答探索系統提出了 5 種統計模組，這 5 種統計模組分別為：TF-IDF 模組、改良式 TF-IDF 模組、查詢詞鍵擴展 (Query Expansion) 模組、統計基礎翻譯 (Statistical Translation) 模組、潛在變數模組 (Latent Variable Models)，以下分別概略介紹這 5 種統計模組。

2.1.1. TF-IDF 模組

TF-IDF 模組是以資訊擷取技術為基礎，評估使用者新問題與問答集中答案的相似度，並將相似度最大的答案視為新問題的回覆。換句話說，TF-IDF 模組把使用者新提問的問題當作查詢句 q (Query)，將問答集中的每一個答案視為文件集的一篇文件 a ，利用資訊擷取技術分別將此查詢句(即新問題)和文件集的文件(即問答集中的每一個答案)以詞鍵向量空間表示，利用方程式 1(a)評量查詢句 q 與文件 a 的相似度，與查詢句 q 相似度最高的文件即為使用者新問題的最佳回覆。

$$(a) score(q, a) = \frac{\sum_{w \in q, a} I_w^2 \cdot f_q(w) \cdot f_a(w)}{\sqrt{\sum_{w \in q} f_q(w)^2 \sum_{w \in a} f_a(w)^2}}$$

$$(b) I_w = idf_w = \log \frac{N}{df_w}$$

方程式 1：查詢句與文件的相似度計算

在方程式 1 中 N 為文件集中的文件總數； df_w 是在文件集中出現詞鍵 w 的文件數； $f_q(w)$ 代表詞鍵 w 出現在查詢句 q 中的頻率； $f_a(w)$ 表示詞鍵 w 出現在文件 a 中的頻率； $score(q, a)$ 為查詢句 q 和文件 a 的相似度。

2.1.2. 改良式 TF-IDF

上述 TF-IDF 模組雖然簡單，但以問答組 (Question-Answer Pair) 為訓練集 (Training Set) 來說，依然是有改善的空間。改良式 TF-IDF 模組是由訓練集改善每一個詞鍵的 IDF 值，即前述方程式 1 中的 I_w 。改善的方法是根據問答集中的每一組問答組，對每一詞鍵 w 利用登山法 (Hill Climbing) 修正其 IDF 值，提高同時出現在問題與相對答案中的詞鍵之 IDF 值，降低沒有同時出現在問題與相對答案中的詞鍵之 IDF 值。

改善詞鍵 IDF(I_w) 值的方法是針對所有詞鍵 w ，考慮所有問題與答案同時出現詞鍵 w 的問答組，並利用方程式 2 改善詞鍵 w 的 IDF(I_w) 值。

$$(a) I_w \leftarrow I_w + r \left(\frac{2}{I_w} \right) score(q, a^*)$$

$$(b) I_w \leftarrow I_w - r \left(\frac{2}{I_w} \right) score(q, a')$$

方程式 2：改良式 TF-IDF 模組 IDF 改善公式

在方程式 2 中 r 代表學習速度參數 (Learning Rate Parameter)，其值介於 0 和 1 之間； q 是問答組的問題部分； a^* 為該問答組的答案部分；而 a' 表示其它問答組的答案； $score$ 這一個函數可利用方程式 1(a) 計算。一般情形下， I_w 會根據方程式 2(a) 修正，而方程式 2(b) 只有在 $score(q, a') > score(q, a^*)$ 時才會做調整。

經過上述登山法的學習後，每一個詞鍵的 IDF 值就能更貼切地表示該詞鍵對每一個答案集中的答案之重要性，因此使用者新問題和問答集中的答案之相

似度就能利用方程式 1(a)計算得到，而在方程式 1(a)中的 I_w 就是先前經登山法學習的 IDF 值。

2.1.3. 查詢詞鍵擴展 (Query Expansion) 模組

[Berger00]從這一個模組開始，將問答組中問題與答案所出現詞鍵間的關係納入考慮，運用這個關係來找出真正的答案。例如：若問題出現詞鍵「為什麼」，在對應的答案可能會出現詞鍵「因為」。

查詢詞鍵擴展模組是利用問答集中問題詞鍵 u 與答案詞鍵 v 間的交互資訊值 (Mutual Information) $I(u, v)$ ，交互資訊值代表問題詞鍵 u 與答案詞鍵 v 的相關性，公式如下列方程式 3：

$$(a) I(u, v) = H(p(v)) - p(u)H(p(v|u)) - p(\bar{u})H(p(v|\bar{u}))$$

$$(b) H(p) = -p \log p - (1-p) \log(1-p)$$

$$(c) p(v) = p(v \in a), p(u) = p(u \in q), p(\bar{u}) = p(u \notin q)$$

方程式 3：問答集中問題與答案詞鍵間的交互資訊值

其中， q 表示某一問答組中的問題， a 為該問答組中的答案。

查詢詞鍵擴展模組先利用方程式 3 中的 $I(u, v)$ 找出與使用者問題中所有問題詞鍵最相關的 n 個答案詞鍵，再把這些答案詞鍵加入原來使用者的問題查詢句中，增加問題查詢句在答案中可能出現的詞鍵，以提升系統的準確率。例如：若使用者問題中出現 why、site 和 windows 這些詞鍵，經過方程式 3 中的 $I(u, v)$ 可找出與這些詞鍵最相關的答案詞鍵分別為 because、http 和 Microsoft，而查詢詞鍵擴充模組會將這三個答案詞鍵加入查詢中，使原本的查詢詞鍵擴充成：why、site、windows、because、http 和 Microsoft。

2.1.4. 統計基礎翻譯模組 (Statistical Translation)

近年來統計基礎翻譯技術 (Statistical Machine Translation) 運用在解決文件擷取和文件自動摘要上有顯著的成效，因此[Berger00]運用相同的技術來解決在自動問答探索系統上的問題。[Berger00]運用問答集來學習如何由答案詞鍵轉換成問題詞鍵，例如：答案中若同時含有 at、location、place、street、directions 這些答案詞鍵，則這些答案詞鍵會一起轉換為 where 這一個問題詞鍵，藉此評量問答集中的答案成功翻譯為使用者新提問問題的可能性，可能性最高的答案即為使用者新問題的最佳回覆。在[Berger00]採用 IBM 所開發的翻譯模組技術 [Brown90]。

2.1.5. 潛在變數模組 (Latent Variable Models)

潛在變數模組利用因子模組 (Factored Model) 來表示前述問題與答案間的翻譯機制，表示法如方程式 4 所示：

$$p(w_a | w_q) = \sum_z p(w_a | z)p(z | w_q)$$

方程式 4：翻譯因子模組表示法

其中 w_q 為問題詞鍵； w_a 為答案詞鍵； z 為一潛在因子向量，這些潛在因子向量形成一集合 Z 。[Berger00]的潛在變數模組藉由 EM 演算法從問答集中學習出 $p(w_a | z)$ 和 $p(z | w_q)$ 。學習的步驟如下：

1. 任意給定 $p(q | z)$ 、 $p(a | z)$ 、 $p(w_q | z)$ 和 $p(z | w_q)$ 初始值。
2. 利用方程式 5 計算潛在因子。

$$p(z | q, a, w_q, w_a) = \frac{p(q | z)p(a | z)p(w_q | z)p(w_a | z)p(z)}{\sum_{z' \in Z} p(q | z')p(a | z')p(w_q | z')p(w_a | z')p(z')}$$

方程式 5：潛在因子計算公式

3. 針對每一問答組，利用上述方程式 5 所得到的潛在因子重新計算 $p(q|z)$ 、 $p(a|z)$ 、 $p(w_a|z)$ 和 $p(z|w_q)$ ，公式如方程式 6 所示。

$$\begin{aligned}
 p(q|z) &= \sum_{w_q, a, w_a} \lambda(q, a, w_q, w_a) / L \\
 p(a|z) &= \sum_{q, w_q, w_a} \lambda(q, a, w_q, w_a) / L \\
 p(w_q|z) &= \sum_{q, a, w_a} \lambda(q, a, w_q, w_a) / L \\
 p(w_a|z) &= \sum_{q, a, w_q} \lambda(q, a, w_q, w_a) / L \\
 \lambda(q, a, w_q, w_a) &= n(q, w_q) \cdot n(a, w_a) \cdot p(z|q, a, w_q, w_a) \\
 L &= \sum_{q, w_q, a, w_a} \lambda(q, a, w_q, w_a)
 \end{aligned}$$

方程式 6：EM 學習機制

其中 $n(q, w_q)$ 代表問題詞鍵 w_q 出現在問題 q 中的次數； $n(a, w_a)$ 代表答案詞鍵 w_a 出現在答案 a 中的次數。

由上述的學習機制，能夠找出最佳 $p(w_a|z)$ 和 $p(z|w_q)$ 的值，再利用方程式 4 就能找出符合使用者新問題的答案。

第二節 潛在語意分析 (Latent Semantic Analysis)

LSA 藉著統計的方法自動從文件集中粹取並表示詞鍵在文件中的內涵，[Landauer98]指出 LSA 能評估文件內容所隱含的知識，也能適當地表現人類在知識上的推演過程。

LSA 是以奇異值分解 (Singular Value Decomposition, SVD) 和維度約化 (Dimension Reduction) 為基礎的知識模組，SVD 是一種數學矩陣的分解技術，能將文件所隱含的知識抽象轉換到語意空間中，而維度約化能去除文件知識在語意空間中的雜訊，使 LSA 能更精確地推演出文件所隱含的知識。

2.2.1. LSA 的運作流程

LSA 的運作流程如下：

1. 將文件集中所有文件的 Context 表示為一個 Word-by-Context 矩陣 X 。
矩陣 X 中的每個元素 x_{ij} 表示詞鍵 w_i 在 Context P_j 中的出現頻率或重要性。在這裡 Context 代表具有意義的一段文字，可能為 Sentence, Paragraph, Chapter, 或 Document，視需求而定。

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \dots & \dots & \dots & \dots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix}$$

2. 將矩陣 X 經過 SVD 分解後得到三個矩陣的連乘積： $X = WSP^T$ ，其中 S 為一對角矩陣，代表語意空間 (Semantic Space)；矩陣 W 的某一行向量 (Row Vector) 為該詞鍵在語意空間的表示法；矩陣 P^T 的某一行向量為該 Context 在語意空間的表示法。
3. 利用維度約化，消除此語意空間中的雜訊，並重建矩陣 $X \sim X' = W'S'P'^T$ 。經過維度約化所得到的新語意空間 S' 能較精確地描述詞鍵與 Context 所代表的意義。

2.2.2. LSA 的實例說明

以下利用在[Landauer98]中提到的實例來說明 LSA 的運作流程。這一個例子使用 9 個技術文件標題作為 Context。其中 5 個 Context 為關於人機互動 (Human Computer Interaction, HCI) 的相關文件，分別為 c1、c2、c3、c4、c5，其它 4 個 Context 是關於數學圖形理論 (Mathematical Graph Theory) 的相關文件，分別為 m1、m2、m3、m4。

Example of text data: Titles of Some Technical Memos

[Human-Computer Interface]

- c1: *Human* machine *interface* for ABC *computer* applications
- c2: A *survey* of *user* opinion of *computer system response time*
- c3: The *EPS user interface* management *system*
- c4: *System* and *human system* engineering testing of *EPS*
- c5: Relation of *user* perceived *response time* to error measurement

[Mathematical Graph Theory]

- m1: The generation of random, binary, ordered *trees*
- m2: The intersection *graph* of paths in *trees*
- m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
- m4: *Graph minors*: A *survey*

首先，從這些 Contexts 中挑選出現 2 次以上的詞鍵(斜體的部分)，共計 12 個，將這些詞鍵和 Context 建置為一個 Word-by-Context 的矩陣 X，矩陣 X 每一列為所選出的每一個詞鍵，每一行為一個 Context。將矩陣 X 經過 SVD 後，得到三個矩陣的連乘積，分別為 W 、 S 、 P^T 。

$$X = \begin{bmatrix} & c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\ \text{human} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \text{interface} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{computer} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{user} & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{system} & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ \text{response} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{time} & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{EPS} & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \text{survey} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \text{trees} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \text{graph} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \text{minors} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} = WSP^T$$

$$W = \begin{bmatrix} 0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\ 0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\ 0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\ 0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\ 0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\ 0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\ 0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\ 0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\ 0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18 \end{bmatrix}$$

$$S = \begin{bmatrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.01 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\ 0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 & 0.08 \\ -0.95 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 & -0.03 \\ 0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 & -0.60 \\ -0.08 & -0.26 & 0.72 & -0.37 & 0.03 & -0.30 & -0.21 & 0.00 & 0.36 \\ 0.18 & -0.43 & -0.24 & 0.26 & 0.67 & -0.34 & -0.15 & 0.25 & 0.04 \\ -0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 & -0.07 \\ -0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 & -0.45 \end{bmatrix}$$

再經過維度約化，在這裡選擇新維度 (Dimension) 為 2，也就是保留矩陣 S 中最高的兩個數值，其餘的數值均設為 0，相當於只取三個矩陣 W、S、P 的前 2 列，其餘值設為 0，分別得到 W'、S'、P'^T。重建矩陣後得到矩陣 X'。

$$X' = \begin{bmatrix} & c1 & c2 & c3 & c4 & c5 & m1 & m2 & m3 & m4 \\ \text{human} & 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\ \text{interface} & 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\ \text{computer} & 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\ \text{user} & 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\ \text{system} & 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\ \text{response} & 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ \text{time} & 0.16 & 1.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ \text{EPS} & 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\ \text{survey} & 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\ \text{trees} & -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\ \text{graph} & -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\ \text{minors} & -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62 \end{bmatrix} = WS'P^T$$

比較 LSA 前後的矩陣 X 和 X' 。考慮 tree 這一個詞鍵，因 tree 並沒有出現在 m4 這個 graph theory 的標題中，但 m4 包含了 graph 和 minors 這 2 個詞鍵，因此 tree 在 m4 的權重由在原矩陣 X 的 0，在經過 SVD 和維度約化後提昇為在矩陣 X' 的 0.66。另一方面，考慮 survey 這個詞鍵，它在 m4 這一個 graph theory 標題中出現 1 次，因此在原矩陣 X 中的值為 1，經過 SVD 和維度約化後，在矩陣 X' 中的值降為 0.42，這顯示了 survey 對 m4 這個 Mathematical Graph Theory 領域標題並不重要。由此可知，LSA 能粹取出詞鍵中隱含的語意，進而能估量詞鍵對每一個 Context 的重要性。

第三節 概念空間 (Concept Space) 的建立

概念空間的建構主要分為三種方面。其一是人工建立概念空間 [Sugumaran02]；其二是建立詞鍵語意的關聯性，再經由分群自動產生概念空間 [Bellegarda96]、[Park96]、[Lin01]；其三為粹取文件的概念詞鍵，根據文件分群產生詞鍵群，並自動建構出概念空間[Fu01]。

以人工建立概念空間需耗費大量的人力，且當概念改變時也較不易變更，因此近年來有許多研究致力於自動建立概念空間。而自動建立概念空間的技術

又可分為兩種，一種是利用文件分群，另一種是利用詞鍵分群。但因為文件所描述的概念往往不只一種，所以根據文件分群來產生的概念會過於籠統。因此在概念空間的建立方面大多採用詞鍵分群來產生。

在詞鍵分群方面，首先必須建立詞鍵的語意關聯性，而建立詞鍵語意關聯性的方法主要分為二種，第一種是利用詞典 (Thesaurus) [Lin01]，如 WordNet、HowNet。在詞典中已經明確定義了詞鍵語意間的不同、相似、相反、上下位等關係，所以直接查詢詞典便能得到二詞鍵的語意關聯性。但是要為某一領域建構一個完善的詞典需要花費大量的人力與時間，這並不是一件容易的事。

建立詞鍵語意關聯性的第二種方法是藉由分析文件集產生詞鍵的語意關聯性。[Chung99]提出利用共同出現分析 (Co-occurrence Analysis) 來衡量詞鍵的語意關聯，[Chung99]認為若二詞鍵常常共同出現在同一篇文件中，則二詞鍵視為具有相似性，但若二詞鍵未在同一篇文件中共同出現，則二詞鍵就視為不具相似性。然而有時二詞鍵雖未共同出現於同一篇文件，但仍具有相似性，這二詞鍵的相似性可以藉由語意遞移性 (Transitive) 推演出，也就是若詞鍵 A 與詞鍵 B 相似，而詞鍵 B 與詞鍵 C 相似，雖然由共同出現分析將詞鍵 A 與詞鍵 C 視為不具相似性，但藉由詞鍵 B 利用語意遞移性可以推導出詞鍵 A 與詞鍵 C 具有語意關聯性。[Park96]就是利用貝氏網路 (Bayesian Network) 和語意遞移性找出二詞鍵間的語意關聯。

在另一方面，[Bellegarda96]利用了 LSA 表達詞鍵隱含語意的能力來衡量詞鍵間的語意關聯，利用此語意關聯性的衡量產生概念空間。[Bellegarda96]的運作流程如圖 4 所示。首先先將文件集表示為一個 Word-by-Document 的矩陣 K ；經過 SVD 和維度約化後，矩陣 K 變為 $K = USV^T \rightarrow U'S'V'^T = K'$ ，在 U' 中的列向量 u'_i 即為隱含語意的詞鍵表示法；根據上述所得到的詞鍵表示法再利用方程

式 7 即可計算二詞鍵的相似度；最後，套用任一分群演算法對詞鍵做分群，所得到的分群結果即為概念空間。

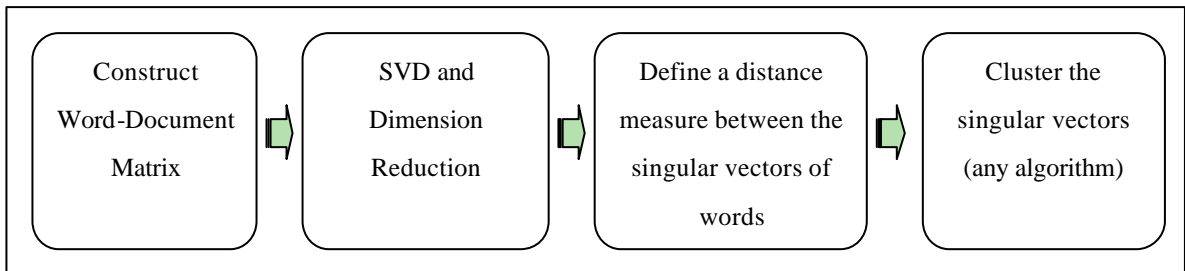


圖 4：利用 LSA 建構概念空間流程圖

$$Sim(w_i, w_j) = D(u'_i, u'_j) = \cos(u'_i S', u'_j S') = \frac{u'_i S'^2 u'_j T}{\|u'_i S'\| \|u'_j S'\|}$$

方程式 7：利用 LSA 計算二詞鍵的相似度

第四節 答案類型 (Answer Type) 判別

答案類型判別的相關研究主要分為二個方向，一為利用文件分類的方法判別答案類型，首先將每一個問題視為一篇文件，每一個答案類型視為一個分類，藉由一個學習機制，學習出每一個分類的知識，對一個新問題，經由之前學習出的分類知識，可以判別出這一個新的問題屬於哪一個答案類型；另一為問答系統 (Question Answering) 對答案類型判別上的研究工作，主要是利用語言學的一些經驗法則或是語句分析，來判斷每一個問題的答案類型，例如：如果問題中有出現「where」，那麼可以判定此問題的答案類型為「地名」。以下分別針對在答案類型判別二方面的相關研究做一些介紹。

2.4.1 文件分類的相關研究

文件分類主要是依據文件的詞鍵特徵，而要學習文件分類的特徵首先要有一個訓練集 (Training Set)，此一訓練集包含事先人工指定好類別的文件，再利

用機器學習 (Machine Learning) 的技術學習每一個文件分類所具有的詞鍵特徵。對於一篇尚未分類的新文件，只要利用這些特徵就能判斷新文件所屬的分類。近期研究中[Lin02]成功導入資料探勘 (Data Mining) 的技術來改善文件的自動分類。

[Lin02]提到在全球資訊網上搜尋資料，主要有兩種方式。一種是搜尋引擎 (Search Engines)，如 Google 和 Alta Vista；另外一種是主題式目錄分類 (Topic Directories)，如 Yahoo!。這兩種方式各有缺點，搜尋引擎主要的缺點是因為它使用字串比對，但在語言中存在一義多詞或一詞多義的情況下，找出來的資料往往有許多不是使用者所需要的，因此準確率較低。主題式目錄分類的最大缺點就是需要藉由人工來完成分類的工作，當有新網頁加入時就需要人工歸類。另外主題式目錄分類所找到的文件只包含有登錄的網頁，因此查全率偏低。

[Lin02]希望能融合這兩種方法的優點以互補彼此的缺點，所以發展出 ACIRD (Automatic Classifier for the Internet Resource Discovery) 系統，期能有效且有效率地組織、搜尋網路上的文件。在[Lin02]中主要的核心就是從主題式目錄中已分類好的文件學習每一分類的知識表示法。

ACIRD 採用監督式學習 (Supervised Learning)，其學習機制主要分為三個部分。首先從已分類的文件初步學習出分類的知識，其次再從已分類的文件探勘出詞鍵間隱含的語意 (Implicit Term Semantics) 關連性規則 (Association Rules)，最後由探勘出的詞鍵關連性規則重新調整之前所學習的分類知識。ACIRD 的學習模組如圖 5 所示。

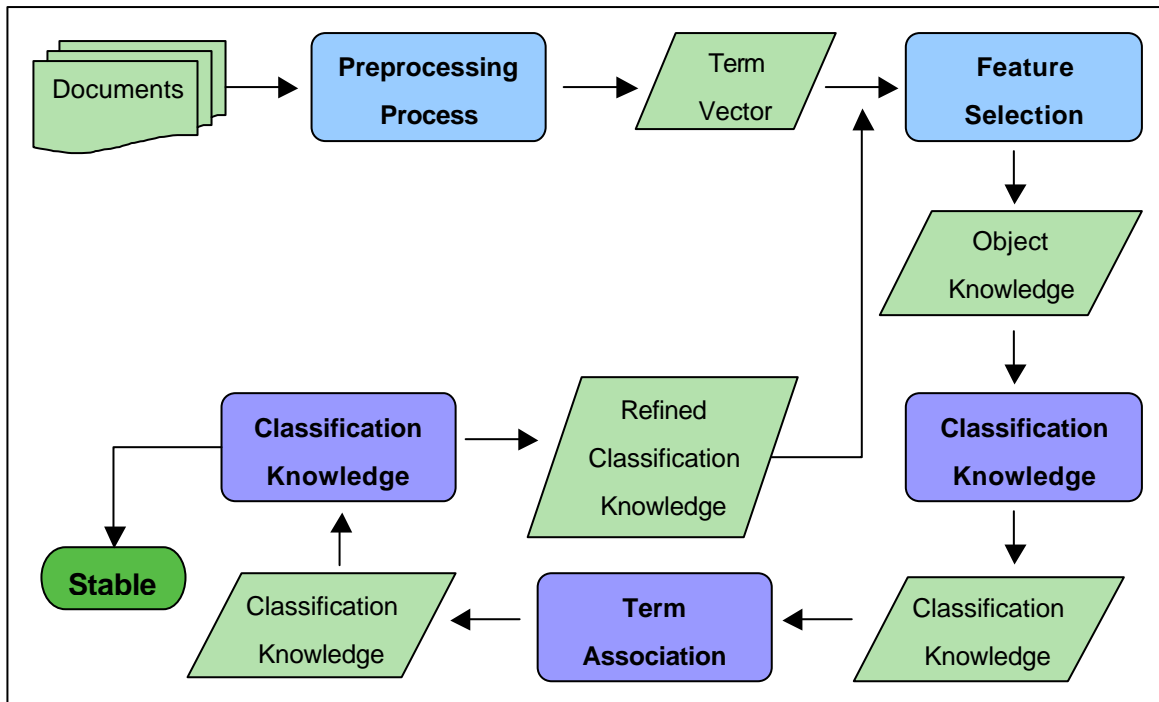


圖 5：ACIRD 學習模組

1. 前處理 (Preprocessing Process)

在經過前處理後，每一個文件都可用一個詞鍵向量來表示，向量中每一個詞鍵都有一個相對應的支持值 (Support Value) 代表這個詞鍵在文件中的重要性。

2. 特徵選取 (Feature Selection)

經過前處理後所得到文件之詞鍵向量，其維度都會很大，導致後續學習過程的複雜度增加，因此利用特徵選取來降低詞鍵向量的維度。[Lin02]利用由經驗得到的門檻值 (Threshold)，消去所有支持值低於臨界值的詞鍵，留下具有代表性的特徵(即詞鍵)。

3. 學習分類知識 (Learning Classification Knowledge)

在 ACIRD 中每一個類別 (Class) 都可用一個晶格 (Class Lattice) 來表示。在晶格中每一個節點 (Node) 都表示一個類別，而每一個父節點 (Parent Node) 表示其子節點 (Child Node) 的集合，即此晶格是代表一個階層式的分類架構。一個沒有父節點的節點稱為最籠統節點 (the Most General Node)，而一個沒有子節點的節點稱為最明確節點 (the Most Specific Node)。

ACIRD 中每一個類別的分類知識也是以一個詞鍵向量來表示，每一詞鍵所對應的支持值表示這個詞鍵在此類別的重要性。分類知識的學習機制就是要學習出代表每一個類別的詞鍵向量，其步驟如下：

- (1) 先針對每一個最明確節點來說，因為最明確節點只包含文件，所以最明確節點 C 的詞鍵向量之支持值，可由在該類別下所有文件的詞鍵向量利用方程式 8 計算得到。

$$\begin{aligned} \sup'_{t_i, C} &= \sum_{o_j} \sup_{t_i, o_j} \\ \sup_{t_i, C} &= \frac{\sup'_{t_i, C}}{\text{MAX} \{ \sup'_{t_i, C} \}} \end{aligned}$$

方程式 8：ACIRD 最明確節點的詞鍵支持值 $\sup_{t_i, C}$

其中 \sup_{t_i, o_j} 代表詞鍵 t_i 在文件 o_j 中的支持值，而 o_j 表示在類別 C 中的一個文件，而 $\sup_{t_i, C}$ 表示詞鍵 t_i 在類別 C 中的支持值。

- (2) 計算其它節點的詞鍵向量支持值。對於其它節點來說，除了可能包含子類別外，還有可能包含無法歸屬於所有子類別下的文件，因此其它節點詞鍵向量的支持值，必須考慮該類別 C 下的所有子類別和不包含在任何子類別下的所有文件。類別 C 的詞鍵向量支持值可利用方程式 9 計算得到。

$$\begin{aligned} \sup'_{t_i, C} &= \sum_{o_j} \sup_{t_i, o_j} + \sum_{c_j} |c_j| \times \sup_{t_i, c_j} \\ \sup_{t_i, C} &= \frac{\sup'_{t_i, C}}{\text{MAX} \{ \sup'_{t_i, C} \}} \end{aligned}$$

方程式 9：ACIRD 其它節點的詞鍵支持值 $\sup_{t_i, C}$

其中 o_j 是屬於類別 C 且不屬於子類別 c_j 的文件；而 $|c_j|$ 是類別 c_j 下的文件數。

4. 探勘詞鍵關連性 (Mining Term Association)

將詞鍵視為探勘時的項目 (Item)，文件中每一個隱含資訊的段落視為探勘時的一筆交易 (Transaction)，而每一個類別則作為一個交易資料庫 (Transaction Database)。ACIRD 運用資料探勘的技術探勘詞鍵的關連性規則。在這裡只考慮一對一的關連性規則，即找出像 $t_i \rightarrow t_j$ 的關連性規則，表示詞鍵 t_j 是詞鍵 t_i 的相關詞，並且利用信任值 (Confidence) 和支持值 (Support) 來估量這兩個詞鍵的相關程度，其定義如方程式 10 所示。

$$\begin{aligned} \text{conf}_{t_i \rightarrow t_j} &= \frac{df_C(t_i \cap t_j)}{df_C(t_i)} \\ \text{sup}_{t_i \rightarrow t_j} &= \frac{df_C(t_i \cap t_j)}{\|D_C\|} \end{aligned}$$

方程式 10：關連性規則 $t_i \rightarrow t_j$ 的信任值 $\text{conf}_{t_i \rightarrow t_j}$ 和支持值 $\text{sup}_{t_i \rightarrow t_j}$

其中 $df_C(t_i)$ 表示在類別 C 中出現詞鍵 t_i 的文件數， $df_C(t_i \cap t_j)$ 表示在類別 C 中同時出現詞鍵 t_i 和 t_j 的文件數， $\|D_C\|$ 為在類別 C 中的總文件數。

經過探勘詞鍵關連性後，結合先前的分類知識圖 6(a) 和詞鍵關連性圖 6(b) 可得到詞鍵語意網路示意圖如圖 6(c)。

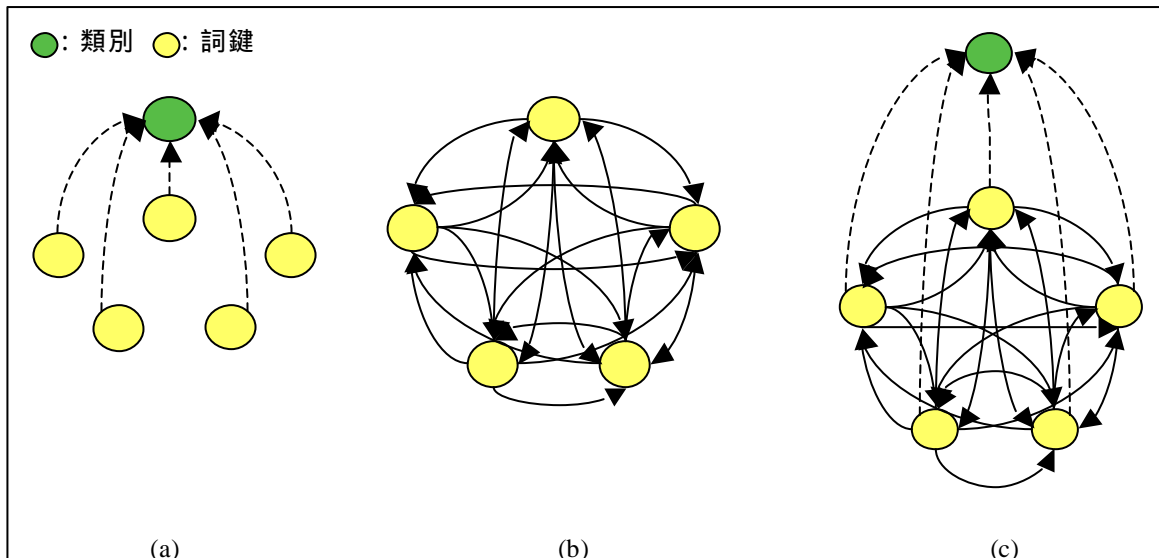


圖 6：ACIRD 詞鍵語意網路示意圖

5. 微調分類知識 (Refinement of Classification Knowledge)

利用上述所探勘出的詞鍵關連性規則，可微調先前 3 所產生的分類知識。為了決定詞鍵在類別中的支持值，必須考慮所有在詞鍵語意網路圖中該詞鍵到該類別的路徑 (Path)，並根據所有的路徑藉由方程式 11 計算出該詞在類別中的支持值。

$$\sup_{t,c}^* = \text{MAX} \left\{ \text{conf}_{t \rightarrow t_j} \times \text{conf}_{t_j \rightarrow t_k} \times \dots \times \text{conf}_{t_y \rightarrow t_z} \times \sup_{t_z,c} \right\}$$

方程式 11：ACIRD 微調分類知識

由於要根據方程式 11 計算所有詞鍵的支持值需要花費很多時間，因此 [Lin02]根據「對一個最佳的路徑來說，其子路徑也是一個最佳路徑。」，進而導入 Greedy 演算法，以加快微調分類知識的速度。

2.4.2 問題系統在答案類型判別上的相關研究

[Radev02]中提出一個網際網路的問答系統，稱之為 NSIR，使用者以自然語言 (Natural Language) 表達問題，NSIR 便從網際網路搜尋出解答，例如：一

個使用者輸入一個問題「Who was the first American in space?」, NSIR 必須找出「Alan Shepard」來回答使用者。

NSIR 的流程主要分為下列幾個步驟：首先必須先判別問題答案的類型，即判別問題所問的是人名、地名、組織名稱 等；接著利用問題所描述的關鍵字從搜尋引擎擷取出相關的文件；之後從這些相關文件抽取出可能包含答案的句子或段落；再來將這些句子或段落切割成一些片語 (Phrase)，這些片語就是可能的答案；最後將這些答案重新排序，使愈符合問題的答案能排列在回傳結果的愈前方。

NSIR 的答案類型判別係採用語言學上的一些經驗法則，所採用的答案類型如表格 1 所示。由於 NSIR 所輸入的問題為英文，因此 NSIR 在答案類型判別上主要是根據英文的疑問詞 (Wh-word)，首先由訓練資料先分析這些疑問詞與各個答案類型的關係，如表格 2 所示。由表格 2 可知當疑問詞為 When 和 Why 時，答案類型只有唯一的可能，分別是 DATE 和 REASON；當疑問詞為 Who 和 Where 時，答案類型必須使用一些簡單的樣版 (Template) 判定問題的答案類型，一些 Template 的實例如表格 3 所示。

PERSON	PLACE	DATE
NUMBER	DEFINITION	ORGANIZATION
DESCRIPTION	ABBREVIATION	KNOWNFOR
RATE	LENGTH	MONEY
REASON	DURATION	PURPOSE
NOMINAL	OTHER	

表格 1：NSIR 答案類型列表

Wh-word	Types	Wh-word	Types
Who(102)	PER(77) DES(19) ORG(6)	What / which (233)	NOM(78) PLA(27) DEF(26) PER(18)

Where(60)	PLA (54) NOM(4), ORG(2)		ORG(16) NUM(14) ABB(13) DATE(11) RATE(4) KNO(8) MON(3) PUR(2) REA(1)
When(40)	DATE(40)	How(48)	NUM(33) LEN(6) RATE(2), MON(2) DUR(3) REA(1), DES(1)
Why(1)	REA(1)		

表格 2：NSIR 疑問詞與各個答案類型的關係表

Template	Types
Who is <Person Name>	DESCRIPTION
Who (manufacture produce grow provide)	ORGANIZATION

表格 3：NSIR 答案類型判別的 Template 實例表

疑問詞為 What/Which 的答案類型判別，先要將問題做語句詞性標記 (Part of Speech Tagging, POS Tagging)，若 What/Which 是 WDT (Determiner)，則答案類型的判別是根據緊接在 What/Which 的名詞片語來做辨別，例如：「What card company sells Christmas ornaments?」，What 由語句詞性標記得知為 WDT，因此答案類型是由 card company 判斷，又 card company 為一名詞片語，所以根據該片語的最後一個名詞 company 來辨別，這個名詞稱為資訊名詞 (Information Noun)，故這個問題的答案類型為 ORGANIZATION。若 What/Which 為 WP (wh-phrase) 時，首先根據緊接在後的動詞做判斷，例如：「What caused the Lynmouth floods?」，What 經語句詞性標記得知為 WP，由 caused 所意指的答案類型可知問題的答案類型為 DEFINITION；如果動詞無法做判斷(例如動詞為 Be 動詞)，則根據動詞之後的第一個名詞片語的最後一個名詞來辨別，例如：「What is the average salary of a professional baseball player?」，其第一個名詞片語為「the average salary」，由 salary 可知問題的答案類型為 MONEY。

當疑問詞為 How 或其它情況時，則必須由問句的資訊名詞來辨別，再根據資訊名詞所意指的答案類型來做辨別。

第三章 概念式自動問答探索系統

本論文提出一套概念式自動問答探索系統 (Automatic Concept-Based Answer Finding System, ACAF System), 結合潛在語意分析 (Latent Semantic Analysis, LSA)、概念式搜尋系統與答案類型的辨別機制, 提高系統的效能。

本章首先描述 ACAF 的系統架構; 接著分別說明系統中三個主要模組的建構方式, 分別是問答組詞鍵關係的學習機制、概念空間的建立、與答案類型的判別知識; 最後闡述 ACAF 系統的問答探索機制。

第一節 系統架構

由相關研究工作中對潛在語意分析 LSA 的介紹, 我們知道 LSA 能用以描述詞鍵與語句 (Sentence) 或文件 (Document) 的隱含關係, 因此, 在本論文提出的概念式自動問答探索系統乃運用 LSA 找出問題與答案間用詞的關係, 再佐以概念式查詢與答案類型的比對, 期能更廣泛準確地找到使用者問題的解答。

圖 7 是 ACAF 的系統架構圖。當使用者將新問題 Q 輸入 ACAF 系統, ACAF 的處理流程如下:

1. 將問題 Q 斷詞切字, 得到問題 Q 的關鍵字和疑問詞鍵 (Question Words)。
2. 將問題 Q 的關鍵字和問題與答案的詞鍵關係矩陣 (W_q-by-W_a Matrix) 加以比對, 找出與問題 Q 最相關的 N 個答案詞鍵。
3. 將問題 Q 的關鍵字和問題概念 (Question Concept) 知識庫加以比對, 找到與問題最相關的 M 個問題概念。
4. 分別依據此 N 個答案詞鍵與 M 個問題概念, 找到一些候選答案。

- 根據答案類型知識庫篩選符合的答案類型，找出符合問題 Q 的真正解答。

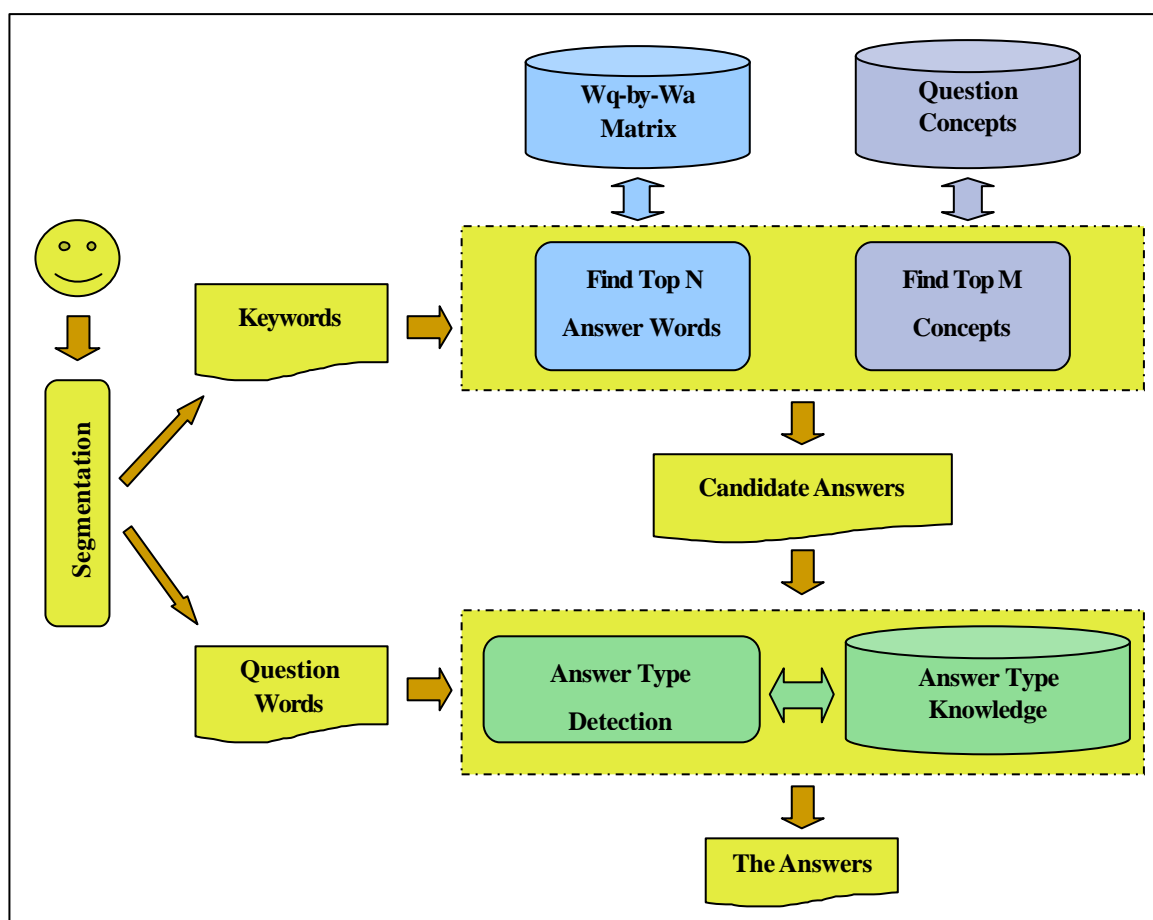


圖 7：ACAF 系統架構圖

由上述的 ACAF 系統流程可知，ACAF 系統的成功與否，取決於問題與答案詞鍵關係矩陣、問題概念知識庫和答案類型知識庫，因此在提供問答探索的服務之前，必須先從訓練資料集 (Training Data Set) 中分別學習出問題與答案詞鍵關係矩陣、問題概念知識庫和答案類型知識庫，圖 8 為 ACAF 的學習機制架構圖。

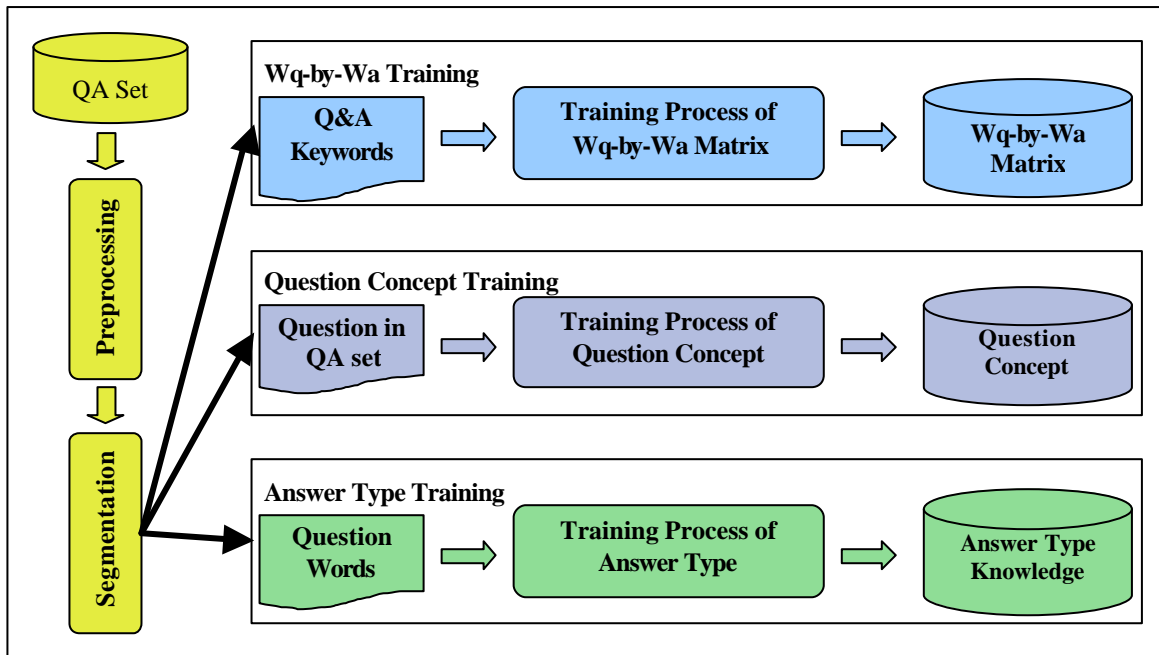


圖 8：ACAF學習架構圖

首先針對訓練資料集(即問答集, QA Set)進行前置處理 (Preprocessing), 前置處理主要是在控制用詞的一致性, 例如: 台與臺、体與體、 等不同寫法的字, 這些字的使用必須統一, 以減去問答集的雜訊, 提高後續學習的效能。

接著對問答集斷詞切字, 本論文使用的是比對詞庫的方法, 並採取最長符合 (Longest Matching) [Nei99]的策略, 同時也會消去停用詞 (Stop Word), 例如:「我是一位交通大學資訊科學系的學生。」此一句子會斷詞成:「我」、「一位」、「交通大學」、「資訊科學系」、「學生」, 其中「是」與「的」這些不具意義的詞都是停用詞, 而對「交通大學」這一關鍵詞來說, 雖然「交通」和「大學」也都有出現在詞庫中, 但是「交通大學」是由「交」字開始符合詞庫的最長關鍵詞, 所以會切割為「交通大學」一詞, 而不會斷為二詞「交通」和「大學」。問答集中問題與答案經斷詞切字後所得到的關鍵詞, 會輸入問題詞鍵與答案詞鍵關係的訓練流程, 以學習出問題詞鍵與答案詞鍵的關係。

另一方面，在斷詞切字的處理過程中，除了一般關鍵詞外，因 ACAF 必須從問題中學習出答案類型，所以雖然在一般的斷詞切字中，會把像「為什麼」、「何時」、「哪些」等的疑問詞鍵視為停用詞，本論文仍是將這些疑問詞鍵保留，以便學習答案類型與疑問詞鍵的關係。

至於 ACAF 學習機制中的問答組詞鍵關係、問題概念知識庫和答案類型知識庫，分別在以下三節中闡述。

第二節 問答組詞鍵關係的學習機制

在[Berger00]中統計基礎翻譯模組 (Statistical Translation) 和潛在變數模組 (Latent Variable Models)，是利用前置工作所學習出問題與答案的詞鍵關係，從問答集中找出符合使用者問題的答案。此外根據 LSA 在傳統資訊擷取中的成功經驗，LSA 能探勘出詞鍵與 Context 間的隱含關係。因此我們認為若能善加利用 LSA，就可提高學習問答組詞鍵隱含關係的效能，進而改善探索的準確率 (Precision)。

LSA 是利用一個 Word-by-Context 的矩陣學習出詞鍵與 context 的隱含語意，因此如果利用問題詞鍵 (w_q) 與答案詞鍵 (w_a) 的共同出現關係建立一個 W_q -by- W_a 的矩陣 M ，再套入 LSA，經過奇異值分解 (SVD) 與維度約化 (Dimension Reduction) 後，即可得到問題詞鍵與答案詞鍵的隱含關係矩陣 M' 。問答組詞鍵關係的學習機制如圖 9 所示。

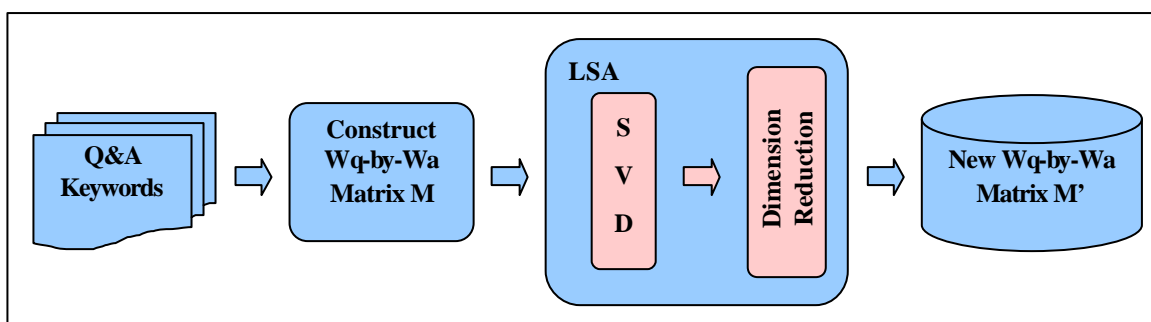


圖 9：問答組詞鍵關係學習機制圖

假設共有 n 組問答組，以 $\langle q_i, a_i \rangle$ 表示第 i 組問答組， q_i 表示問題， a_i 表示答案。要建立一個代表問題詞鍵 W_q 與答案詞鍵 W_a 之間共同出現關係的 W_q -by- W_a 矩陣 M ，其步驟如下：

首先，先將所有的問題詞鍵 w_q 以 n 維的問題向量表示， $w_q = [f_{wq_1}, f_{wq_2}, \dots, f_{wq_n}]$ ，其中 f_{wq_i} 代表 w_q 在 q_i 中出現的頻率，同理所有的答案詞鍵 w_a 亦可以 n 維的答案向量表示， $w_a = [f_{wa_1}, f_{wa_2}, \dots, f_{wa_n}]$ ，其中 f_{wa_i} 代表 w_a 在 a_i 中出現的頻率。

接下來利用方程式 12 計算問題詞鍵 w_q 與答案詞鍵 w_a 的共同出現關係 $m_{w_q w_a}$ ，其中 $|kthanswer|$ 表示第 k 組問答組中答案的長度比例。在計算 $m_{w_q w_a}$ 時除以第 i 組問答組的答案長度比例 $|kthanswer|$ 的原因是：因為問題的用詞方面通常都較為簡短，但是答案的部分有時會因為有較詳細的說明而長度較長；此外，由觀察訓練資料集發現，當答案較為簡短時，問題與答案間的詞鍵關係較強，然而當答案較長時，由於答案的用字充斥著與問題較不相關的說明關鍵詞，因此問題詞鍵與大部分的答案詞鍵關係較弱，故除以答案長度比例以顯現問題與答案間的詞鍵關係會隨著共同出現的問答組之答案長度比例而有所不同。

$$m_{w_q w_a} = \frac{\sum_{k=1}^n f_{w_q k} \times f_{w_a k}}{|kth\ answer|}$$

$$|kth\ answer| = 1 + \frac{\sum f_{w_a k}}{10}$$

方程式 12：問題詞鍵 w_q 與答案詞鍵 w_a 的共同出現關係計算公式

最後利用每一個問題詞鍵與答案詞鍵共同出現的關係(如方程式 12 所示)

即可建立一個 W_q -by- W_a 矩陣 M ，如下：

$$M = \begin{bmatrix} m_{w_{q1} w_{a1}} & m_{w_{q1} w_{a2}} & \Lambda & m_{w_{q1} w_{aN}} \\ m_{w_{q2} w_{a1}} & m_{w_{q2} w_{a2}} & \Lambda & m_{w_{q2} w_{aN}} \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ m_{w_{qM} w_{a1}} & m_{w_{qM} w_{a2}} & \Lambda & m_{w_{qM} w_{aN}} \end{bmatrix}$$

藉由問題詞鍵和答案詞鍵共同出現之關係建立詞鍵關係矩陣 M 後，透過 LSA 之奇異值分解和維度約化此二步驟，得到詞鍵隱含關係矩陣 M' ，矩陣 M' 中的每一元素 $m'_{w_q w_a}$ 表示問題詞鍵 w_q 與答案詞鍵 w_a 之隱含關係。

第三節 概念空間的建構

本節描述問答集中問題之概念空間 (Conceptual Space) 的建構方法。在問答集中可能有與新問題類似的問題，但因新問題的詞鍵與問答集中的詞鍵並不會完全一樣，此時若僅依據由問答集學習出的問題與答案詞鍵間的關係，並不能完全推導出符合新問題的答案可能出現的詞鍵。有鑑於此，我們希望建立問題的概念空間，讓新問題中所使用的詞鍵，能透過問題概念空間，從問答集中找到符合新問題所描述的概念之問答組，而這些問答組中的答案部分就是新問題可能的答案。

在 ACAF 中利用一些詞鍵來表示一個概念，例如：利用「回教」、「伊斯蘭教」等詞來描述「回教」此一概念，也就是說若新問題是與「伊斯蘭教」相關，而由所建立的概念空間可知「伊斯蘭教」與「回教」是同一個概念，因此除了從問答集中尋找由「伊斯蘭教」可能導出的答案詞鍵外，也會搜尋「回教」可能推導出的答案詞鍵。

ACAF 中問題概念空間的建構流程如圖 10 所示，我們同樣採用 LSA 的學習模組，希望能藉 LSA 先找出詞鍵與文件的隱含關係，以提高詞鍵相似度的正確性，進而改善利用詞鍵分群所建構的問題概念空間！

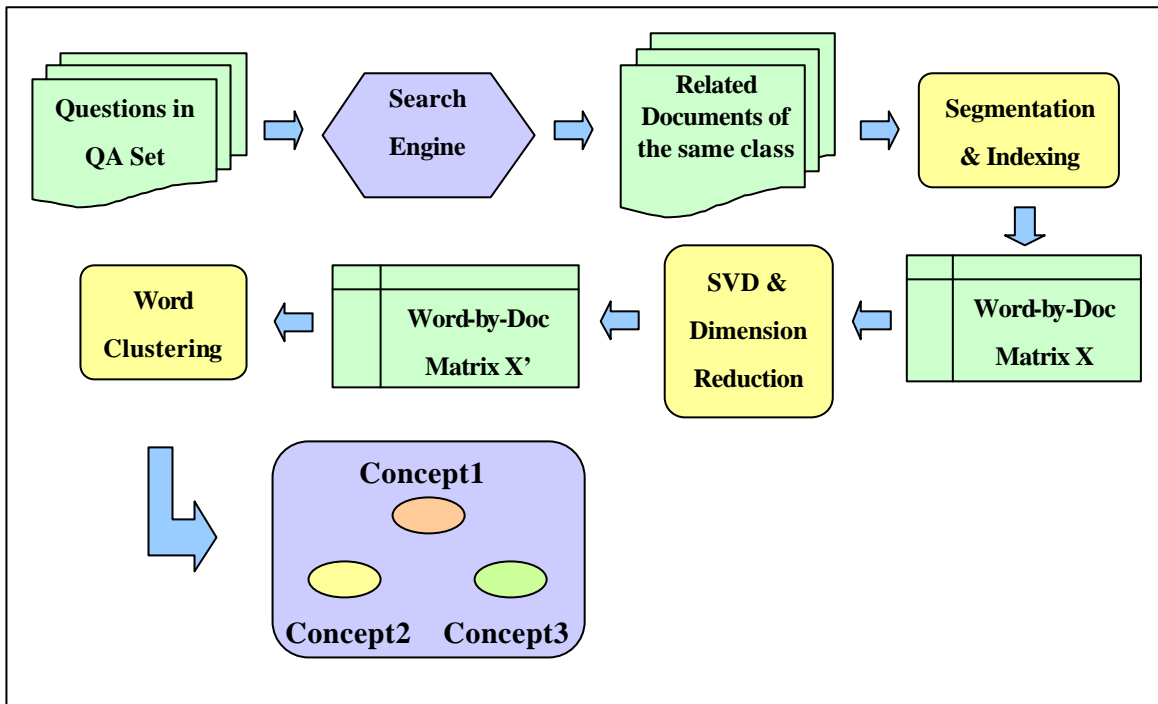


圖 10：ACAF 問題概念空間建構流程圖

由於問答集中問題的字數通常都較少，若直接由問題來建立問題的概念空間，此概念空間會較不完整。有鑑於此，我們首先透過網路上的搜尋引擎收集建構概念空間的文件集，實作上是採用 Google [Google03]。先把問題的關鍵字利用 AND 形成查詢字串，再將查詢透過 Google 為每一個問題找到 5 篇相關文件，並將問題與相關文件合併為一個文件集，每一個問題與每一篇相關文件都

是一篇文件，將此文件集經過斷詞切字 (Segmentation) 和計算權重並建立索引後，產生 Word-By-Document 的矩陣。經 LSA 重建矩陣後，用此矩陣的列向量表示相對應的詞鍵，再採用分群法將詞鍵分群，分群結果中的每一個群就代表一個概念。

本論文建構概念空間所使用的分群法是採用平均連結分群法 (Average-Link Clustering) [Gose96]，演算法如圖 11 所示，斜體字代表變數，*Clusters* 代表分群結果，並利用集合表示，例如： w_1 、 w_2 為一群，而 w_3 、 w_4 為一群，則分群結果 $Clusters = \{\{w_1, w_2\}, \{w_3, w_4\}\}$ 。其詳細步驟如下：首先計算兩兩詞鍵的相似度 (Similarity)，公式如方程式 13 所示，並將高於某一門檻值 (Threshold) 的相似度由高至低排序。接著將相似度最高的兩詞鍵 w_i 和 w_j ，併為一群；接下來處理相似度次高的兩詞鍵 w_k 和 w_l ，判斷 w_k 是否能加入現有群中含有詞鍵 w_l 的所有群，此外亦要判斷 w_l 是否能加入現有群中含有詞鍵 w_k 的所有群，若兩詞鍵 w_k 和 w_l 其中有一詞鍵不能加入現有群中，則兩詞鍵 w_k 和 w_l 再形成新的一群。之後再處理相似度第三高的兩詞鍵，處理方法與上述相同，並以此類推。

$$sim_{w_i, w_j} = \frac{\overline{w}_i \cdot \overline{w}_j}{|\overline{w}_i| |\overline{w}_j|}$$

方程式 13：詞鍵相似度計算公式

在方程式 13 中， \overline{w}_i 代表在 LSA 重建的矩陣中，與詞鍵 w_i 對應的列向量表示法，分子的部分為兩向量的內積，而 $|\overline{w}_i|$ 表示 \overline{w}_i 的向量長度。因此當兩詞鍵對文件集中所有文件的重要性愈相似，則該二詞鍵的相似度就越高。

而要判斷一詞鍵 w 是否能加入某一群 C 中，主要是比較詞鍵 w 與群 C 中所有詞鍵的平均相似度，若平均相似度大於事先定義的門檻值，則詞鍵 w 可以加入群 C 中，否則詞鍵無法加入群 C 中。

分群法中與平均連結分群法相似的技术還包括完整連結分群法 (Complete-Link Clustering) [Jain99]和單一連結分群法 (Single-Link Clustering) [Jain99]，此二種分群法與平均連結分群法的不同點在於：判斷某詞鍵 w 是否能加入某一群 C 時，完整連結分群法是比較詞鍵 w 與群 C 中所有詞鍵的相似度，若相似度均大於門檻值，則詞鍵 w 可以加入群 C 中；而單一連結分群法是比較詞鍵 w 與群 C 中所有詞鍵的相似度，只要有一相似度大於門檻值，則詞鍵 w 可以加入群 C 中。由於使用完整連結分群法所得到的概念會過於瑣碎，而使用單一連結分群法所得到的概念又太過籠統，因此本論文採用折衷的技术－平均連結分群法。

- | |
|--|
| <ol style="list-style-type: none"> 1. Input: word vectors and threshold T ; Output Variable: $Clusters = \phi$ 2. Count Similarities between words. 3. $Sims[]$? Sort the Similarities bigger than T. 4. for $i=0$ To Size-of ($Sims[]$) <ol style="list-style-type: none"> (1) Let $Sims[i]$ is the similarity of w_j and w_k. (2) $NewCluster = -1$; $Changes = 0$ (3) \forall cluster c in $Clusters$ and w_k in c,
 AVE = the average of the similarities of w_j and all words in c
 if ($AVE \geq T$) then w_j joins c and $Changes = Changes + 1$
 if ($Changes > 0$) then $NewCluster = NewCluster + 1$ and $Changes = 0$ (4) \forall cluster c in $Clusters$ and w_j in c,
 AVE = the average of the similarities of w_k and all words in c
 if ($AVE \geq T$) then w_k joins c and $Changes = Changes + 1$
 if ($Changes > 0$) then $NewCluster = NewCluster + 1$ (5) if ($NewCluster < 2$) then
 New cluster $c = \{w_j, w_k\}$
 c joins $Clusters$ |
|--|

圖 11：平均連結分群法之演算法

以下以一個實例來說明平均連結分群法的每一個步驟。

表格 4 為詞鍵的相似度矩陣，假設事先定義的門檻值為 0.5，表格 4 中灰底的部分為詞鍵相似度大於門檻值的詞鍵組。首先提取詞鍵間相似度大於 0.5 的詞鍵組，根據相似度由大至小分別為(A, F)、(A, E)、(B, F)、(A, D)、(A, C)、(B, D)、(C, D)，依序考慮這此詞鍵組能不能加入現有群中，起始的分群結果為一空集合。

	A	B	C	D	E	F
A		.3	.5	.6	.8	.9
B			.4	.5	.1	.8
C				.3	.5	.2
D					.4	.1
E						.3
F						

表格 4：詞鍵相似度矩陣實例

表格 5 為表格 4 套用平均連結分群法每一個步驟分群結果的變化，粗體字的部分為新產生的分群或發生變化的分群結果。詳細的分群步驟描述如下：

步驟	分群結果
一	{A, F}
二	{A, E, F}
三	{A, E, F}、 {B, F}
四	{A, E, F}、{B, F}、 {A, D}
五	{A, E, F}、{B, F}、{A, D}、 {A, C}
六	{A, E, F}、{B, F}、{A, D}、{A, C}、 {B, D}
七	{A, E, F}、{B, F}、{A, D}、 {A, C, E} 、{B, D}

表格 5：表格 4 套用平均相似度每一步驟的分群結果變化

第一步：考慮(A, F)，因現在分群結果為空集合，故{A, F}自成一群。

第二步：考慮(A, E)，現在分群中僅有一群{A, F}，因此判斷 E 是否能加入{A, F} 中，先計算 E 與此群{A, F} 的平均相似度， $Ave(Sim_{AE}, Sim_{EF}) = (0.8 + 0.3) / 2 = 0.55$ ，因平均相似度大於門檻值 0.5，所以 E 能成功加入現有群{A, F}中形成{A, E, F}。

第三步：考慮(B, F)，現在分群中僅有一群{A, E, F}，因此判斷 B 是否能加入{A, E, F} 中，先計算 B 與此群{A, E, F} 的平均相似度， $Ave(Sim_{AB}, Sim_{BE}, Sim_{BF}) = 0.4$ ，因平均相似度小於門檻值 0.5，所以 B 不能加入現有群{A, E, F}中。此外現在的分群結果中沒有包含 B 的分群，因此{B, F} 形成一群。現在分群結果有二群，{A, E, F}、{B, F}。

第四步：考慮(A, D)，現有分群中包含 A 的分群僅有一群{A, E, F}，因此判斷 D 是否能加入{A, E, F}中，先計算 D 與此群{A, E, F}的平均相似度， $Ave(Sim_{AD}, Sim_{DE}, Sim_{DF}) = 0.367$ ，因平均相似度小於門檻值 0.5，所以 D 不能加入現有群{A, E, F}中。此外現在的分群結果中沒有包含 D 的分群，因此{A, D} 形成一群。現在分群結果有三群，{A, E, F}、{B, F}、{A, D}。

第五步：考慮(A, C)，現有分群中包含 A 的分群有二群{A, E, F}和{A, D}，因此先判斷 C 是否能加入{A, E, F}中，計算 C 與此群{A, E, F}的平均相似度， $Ave(Sim_{AC}, Sim_{CE}, Sim_{CF}) = 0.4$ ，因平均相似度小於門檻值 0.5，所以 C 不能加入現有群{A, E, F}中。再判斷 C 是否能加入{A, D}中，計算 C 與此群{A, D}的平均相似度， $Ave(Sim_{AC}, Sim_{CD}) = 0.4$ ，因平均相似度小於門檻值 0.5，所以 C 不能加入現有群{A, D}中。此外現在的分群結果中沒有包含 C 的分群，因此{A, C} 形成一群。現在分群結果有四群，{A, E, F}、{B, F}、{A, D}、{A, C}。

第六步：考慮(B, D)，現有分群中包含 B 的分群僅有一群{B, F}，因此判斷 D 是否能加入{B, F} 中，先計算 D 與此群{B, F} 的平均相似度，

$Ave(Sim_{BD}, Sim_{DF}) = 0.3$ ，因平均相似度小於門檻值 0.5，所以 D 不能加入現有群 {B, F} 中。另一方面，現在的分群結果中包含 D 的分群，僅有一群 {A, D}，因此判斷 B 是否能加入 {A, D} 中，先計算 B 與此群 {A, D} 的平均相似度， $Ave(Sim_{AB}, Sim_{BD}) = 0.4$ ，因平均相似度小於門檻值 0.5，所以 B 不能加入現有群 {A, D} 中。因此 {B, D} 形成一群。現在分群結果有五群，{A, E, F}、{B, F}、{A, D}、{A, C}、{B, D}。

第七步：考慮最後一對詞鍵組(C, E)，現有分群中包含 C 的分群僅有一群 {A, C}，因此判斷 E 是否能加入 {A, C} 中，先計算 E 與此群 {A, C} 的平均相似度， $Ave(Sim_{AE}, Sim_{CE}) = 0.65$ ，因平均相似度大於門檻值 0.5，所以 E 能成功加入現有群 {A, C} 中形成新的 {A, C, E}，現在分群結果仍有五群，{A, E, F}、{B, F}、{A, D}、{A, C, E}、{B, D}。另一方面，現在的分群結果中包含 E 的分群，有二群 {A, E, F} 和 {A, C, E}，因此判斷 C 是否能加入 {A, E, F} 中，先計算 C 與此群 {A, E, F} 的平均相似度， $Ave(Sim_{AC}, Sim_{CE}, Sim_{CF}) = 0.4$ ，因平均相似度小於門檻值 0.5，所以 C 不能加入現有群 {A, E, F} 中。故最後利用平均相似度分群法得到的分群結果為：{A, E, F}、{B, F}、{A, D}、{A, C, E}、{B, D}。

第四節 答案類型判別知識的學習

由答案類型判別的相關研究中可以發現，利用文件分類的技術來判別答案的類型，對於每一篇文件需要有較多的字數，然而對於我們要處理的問題來說，問題的描述通常都較為精短，因此較不適合，故在答案類型的判別方面，本論文參考[Radev02]中的作法，但因[Radev02]所處理的是英文的問題，然而 ACAF 回答的是中文的問題，而對中文的問題來說，若能抓取一些較重要的疑問詞鍵，例如：「為什麼」、「如何」、「多少」等，大部分都能直接做出答案類型的判斷，雖然這些疑問詞鍵對於問題與答案詞鍵關係和概念空間的建立是沒有意義的

(通常我們會將這些疑問詞鍵視為停用詞)，但是在答案類型判別方面，卻是決定性的答案類型特徵。

ACAF 將問題分為 9 種答案類型，並採用機率經驗法則 (Probabilistic Heuristic Rule)，利用問題中可能出現的疑問詞鍵來辨識問題的類型。每一個答案類型與可能出現的疑問詞鍵如表格 6 所示。其中「NOWord」代表問題中不包含任何疑問詞鍵的一個標記，而[一 十]表示在該位置出現一到十其中一個，或者不出現，例如「哪[一 十]位」表示「哪一位」、「哪二位」、...、「哪十位」這些疑問詞鍵。

此外，由表格 6 可知對於某些疑問詞鍵其所屬的答案類型是唯一的，例如：如果問題中出現「何時」，那麼該問題的答案類型就是「時間」；而對於其它疑問詞鍵其所屬的答案類型並不唯一，例如：如果問題中出現「為何」，那麼該問題就有「原因」、「描述性」、「人名」、「專有名詞」這些可能的答案題型，因此在判斷問題的答案類型以前，必須要先學習疑問詞鍵對於答案類型的可能性。

要學習判別答案類型也就是要先學習疑問詞鍵對於每一個答案類型的可
能性，學習流程如下(參見圖 8)：

1. 對問答集中的每一個問題人工設定好其答案類型。
2. 將問答集中的問題部分做斷詞切字，找出疑問詞鍵，而當一個問題沒有出現疑問詞鍵時，我們會給這一個問題一個特殊的疑問詞鍵「NOWord」。
3. 計算每一個疑問詞鍵 t_j 對每個答案類型 AT_i 的可能性，在這裡我們是計算出現某疑問詞鍵 t_j 的問題屬於某個答案類型 AT_i 的機率，計算公式如方程式 14 所示，其中 $freq(t_j \in AT_i)$ 表示答案類型為 AT_i 的問題

中，疑問詞鍵 t_j 出現的頻率，而 $totfreq(t_j)$ 代表在所有問答集的問題中，疑問詞鍵 t_j 出現的頻率。

$$P(t_j, AT_i) = \frac{freq(t_j \in AT_i)}{totfreq(t_j)}$$

方程式 14：疑問詞鍵 t_j 的問題對某個答案類型 AT_i 的可能性

代號	答案類型	可能出現的疑問詞鍵
0	原因	為何、為什麼、起因。
1	時間	何時、什麼時間、什麼時候、時間、年、月、日、何年、何月、何日、誕辰、哪[一 十]天、分、分鐘、多少、幾。
2	數字	多少、幾、NoWord。
3	描述性	何謂、由來、意義、什麼、為何、是否、起源、來源、NoWord。
4	地名	哪裡、什麼地方、何處、NoWord
5	人名	誰、為何、何[二 九]者、何人、哪些人、人物、NoWord
6	專有名詞	哪些、為何、什麼、情況、何者、NoWord。
7	HowToDo	如何、怎麼辦。
8	全文出處	出處、全文、出自何處。

表格 6：答案類型與可能出現的疑問詞鍵

第五節 問答探索機制

ACAF 系統主要的功能是要從問答集中找到最符合新問題的答案。本論文以三個方面來考量新問題 q 與問答集中歷史答案 a 的相似度，其一為利用先前探勘出的問題與答案詞鍵間的關係，來衡量問題 q 與答案 a 詞鍵間的關係強度；其二為問題 q 與答案 a 在問答集中相對應的問題 q' 在概念上的描述相似度；其三為問題 q 與問題 q' 在答案類型上符合的強度。最後 ACAF 計算使用者的問題 q 與問答集中歷史答案 a 的相似度就會綜合評量這三種相似度。

在衡量問題 q 與答案 a 詞鍵間的關係強度方面，首先先計算答案 a 中每一個答案詞鍵 w_a 與問題 q 的相關度，如方程式 15 所示，也就是由先前所學習出的問題與答案詞鍵關係矩陣中找出答案詞鍵 w_a 與問題 q 中所有的關鍵詞鍵 w_q 的關係強度 $m_{w_q w_a}$ ，並將這些詞鍵關係強度加總，其和就是問題 q 與答案詞鍵 w_a 的關係強度。

$$score_{w_a q} = \sum_{w_q \in q} m_{w_q w_a}$$

方程式 15：答案詞鍵 w_a 與問題 q 的相關度

接著藉由 $score_{w_a q}$ 來衡量問題 q 與答案 a 詞鍵間的關係強度，如方程式 16 所示，也就是先計算答案 a 中每一個詞鍵 w_a 在問題 q 中的關係強度 $score_{w_a q}$ 和在答案 a 中的權重 $weight_{w_a, a}$ 之乘積，再加總這些乘積，所得到的值就是問題 q 與答案 a 詞鍵間的關係強度；即問題 q 與答案 a 詞鍵間的關係強度取決於所有詞鍵 w_a 與問題 q 的關係強度和在答案 a 中的權重。其中詞鍵 w_a 在答案 a 中的權重以 w_a 在答案 a 中的出現頻率表示。

$$SimW_{qa} = p(a | q) = \sum_{w_a \in a} p(w_a | q) = \sum_{w_a \in a} score_{w_a q} \times weight_{w_a, a}$$

$$weight_{w_a, a} = freq(w_a \in a)$$

方程式 16：問題 q 與答案 a 詞鍵間的關係強度

此外，假設答案 a 在問答集中相對應的問題為 q' ，則問題 q 與問題 q' 在概念上的相似度計算方面，首先計算問題 q 與某一概念 c 的相似度，公式如方程式 17 所示，此相似度計算考量了所有同時出現在問題 q 和概念 c 中的詞鍵 w ，問題 q 與概念 c 的相似度就是這些詞鍵在概念 c 中的權重平均，其中詞鍵 w 在概念 c 的權重 $weight_{wc}$ 就是詞鍵 w 與在概念 c 中的其它詞鍵 w' 的平均相似度。

$$Sim_{qc} = Ave_{w \in q, w \in c} (weight_{wc})$$

$$weight_{wc} = Ave_{w' \neq w, w' \in c} (Sim_{w,w'})$$

方程式 17：問題 q 與概念 c 的相似度

根據方程式 17，我們可以找出問題 q 與問題 q' 最具代表性的 M 個概念，分別以 $C_q=(c_{q1}, c_{q2}, \dots, c_{qM})$ 和 $C_{q'}=(c_{q'1}, c_{q'2}, \dots, c_{q'M})$ 來表示，並利用方程式 18 來計算出問題 q 與問題 q' 在概念描述上的相似度。

$$SimC_{qq'} = \sum_{c \in C_q, c \in C_{q'}} Sim_{qc} \cdot Sim_{q'c}$$

方程式 18：問題 q 與問題 q' 在概念相似度

當我們知道問題 q 與答案 a 詞鍵間的關係強度，以及問題 q 與答案 a 在問答集中相對應的問題 q' 在概念上的相似度後，ACAF 根據方程式 19 先計算在問答集中所有答案與問題的篩選相似度，並擷取出此相似度大於 0 的所有答案，這些答案就是符合新問題 q 的預選答案(如圖 7 中的 Candidate Answers)，再將這些預選答案經過答案類型的符合強度計算。

$$SimWC_{qa} = w_w \times SimW_{qa} + w_c \times SimC_{qq'}$$

方程式 19：預選答案篩選相似度

問題 q 與問題 q' 在答案類型上符合強度方面，先考量問題 q 中出現的疑問詞鍵 t_q ，根據之前學習到有關疑問詞鍵 t_q 對於每一個答案類型 AT 的可能性，我們可以得到問題 q 屬於某答案類型 AT 的可能性 $Poss_{q,AT}$ ，如方程式 20 所示。

$$Poss_{q,AT} = \sum_{t_q \in q} P(t_q, AT)$$

方程式 20：問題 q 屬於答案類型 AT 的可能性公式

假設問題 q' 的答案類型為 AT，根據方程式 21 可以計算出新問題 q 與答案 a 在問答集中對應的問題 q' 在答案類型上符合的強度。

$$SimT_{qq'} = \begin{cases} Poss_{q,AT} + 0.5 \times \sum_{AT \neq AT'} Poss_{q,AT'} & \text{if } AT = 3 \\ Poss_{q,AT} + 0.3 \times Poss_{q,3} & \text{otherwise} \end{cases}$$

方程式 21：問題 q 與問題 q' 在答案類型上符合強度

由於當某個問題其答案類型為「描述性」時(其代號為 3)，其答案應會包羅萬象，因此當問題 q' 的答案類型為「描述性」時，其答案內容可能可以回答其它答案類型的新問題，而當問題 q' 的答案類型不是「描述性」，且新問題 q 的答案類型若有可能是「描述性」時，那麼新問題 q 的答案就有可能包含問答集中問題 q' 的答案 a。因此在方程式 21 中，評量問題 q 與問題 q' 在答案類型符合強度，除了考量問題 q 與問題 q' 符合的答案類型外，還會考慮二問題答案類型可能為「描述性」的特性。

總結上述三方面衡量方式，要計算一個問題 q 與問答集中某一個答案 a 的相似度，只要將上述三個相似度做一個加權總和，設 w_w 、 w_C 和 w_T 分別代表「問題 q 與答案 a 詞鍵間的關係強度」、「問題 q 與答案 a 在問答集中相對應的問題 q' 在概念上的描述相似度」和「問題 q 與問題 q' 在答案類型上的符合強度」的加權權重，根據此三個加權權重以及方程式 16、方程式 18 和方程式 21 即可得到問題 q 與答案 a 的相似度，如方程式 22 所示，進而依據此相似度即能找到與使用者所提問問題的最符合答案！

$$Sim_{qa} = w_w \times SimW_{qa} + w_C \times SimC_{qq'} + w_T \times SimT_{qq'}$$

方程式 22：問題 q 和答案 a 的相似度

第四章 實驗結果分析與評估

本章說明 ACAF 所用技術的實驗結果和 ACAF 的系統效能。第一節簡介實驗所使用的問答集，並說明所設計的實驗；第二節介紹實驗所採用的評估方法，第三節評估 ACAF 建構知識庫的品質，第四節討論針對新問題查詢結果的效能評量。

第一節 實驗問答集與實驗設計

在訓練問答集部分，我們採用三種特性不同的問答集，其一是參考國際籃球規則和棒球規則分別設計 20 組問答組，構成訓練問答集共 40 組（此問答集稱為運動規則問答集），其部分內容如圖 12 所示。平均每一個問題中含有 23.025 個字、4.575 個詞鍵，而每一個答案含有 61.075 個字、11.625 個詞鍵。其二是收集大學圖書館的一般參考問題（此問答集稱為一般參考問題問答集），共計 150 組問答組，平均每一個問題中含有 9.593 個字、3.56 個詞鍵，而每一個答案含有 493.533 個字、163.067 個詞鍵。其三是以太南大學浩然圖書館的館內常問題為我們的問答集（此問答集稱為常問問答集），共計 34 組問答組，平均每一個問題中含有 15.882 個字、3.265 個詞鍵，而每一個答案含有 112.794 個字、19.853 個詞鍵。

此外，因在訓練問答集中原本並沒有相對應的答案類型，所以，訓練問答集中的每組問答組先經由人工設定其答案類型，以便學習出疑問詞鍵與答案類型間的關連性。

此三種問答集內容特徵主要不同點在於三問答集之問題內容所涵蓋的範圍，運動規則問答集之內容所涵蓋的範圍較小，常問問答集之內容次之，而一般參考問題問答集之內容所涵蓋的範圍則較零散。

QID	Q	QAnswer	classno	AnswerType
1	在什麼情況下計時器會停止計時呢？	1、任何投球或罰球中斷時。2、起球時。一位罰球手罰球時，球離開手不能0:00	000	6
2	一球隊的球員離球場有多久時間？	一、當一球員在籃球場上獲得球權後，該球隊必須在二十四秒內投球進籃。0:00	000	3
3	進攻隊的一球員投球時，防守隊球員離球有多久？	發出投球時。當：1、一球員離球離水平面上正向下落的球。2、一球員在球離0:00	000	3
4	當球在起球時，有一防守球員離球時，被罰罰則？	這種情形叫侵手攔球，這是一種違例。干擾球發生在投球時。當：1、球正離起球0:00	000	3
5	進攻隊的一球員投球時，防守隊球員離球有多久？	防守隊發生攔球或攔球攔球的違例，則違例隊受罰：1、若在二分區籃筐區域投球0:00	000	3
6	在罰球時，有什麼功用？	罰球當一球員在籃球場上獲得球權後，該球隊必須在二十四秒內投球進籃。0:00	000	7
7	有規定進攻隊球員在什麼時間內要把球投進籃筐？	當一球員在球場上獲得球權後，該球隊必須在八秒之內投球進籃。0:00	000	1
8	在籃球比賽進行中，當二十四秒計時器超時時，該隊一隊隊員控制球。該隊立即停止比賽。投球權歸到超時時已投球時的一隊。0:00	0:00	000	7
9	當球在起球時，有一球員離球時，被罰罰則？	一、進攻隊違例，場分不計。除非此罰球罰了，則由對隊於罰球線外發射0:00	000	7
10	在籃球比賽時，會聽人說到籃球場上，要如何投球？	投球或罰球，觸及球場一位球員或裁判，該球員或裁判身體任何部分與球場接觸時。0:00	000	7
11	籃球比賽中，幾次會被判出界？	無論幾次入犯規或持球犯規，球員犯滿五次犯規時，應被判出界。0:00	000	2
12	球員犯滿五次犯規後，有規定多久時間內必須出球場？	球員犯滿五次犯規後，必須於三十秒內被判出界。0:00	000	1
13	一個球員已經犯滿五次犯規了，但是他又犯規了，該怎麼辦？	應立即在罰球線上發射「罰」。0:00	000	7
14	一個籃球比賽的成員包含哪些？	球員、教練員、裁判、助理教練以及球隊有關人員。0:00	000	6
15	打籃球時，有那些的侵入犯規，這些犯規包含哪些？	阻擋、攔人、背球非法防守、攔人、非法攔球、不合法的手部動作、攔人等。0:00	000	6
16	一場籃球賽中，場上球員共有多少？	每一球隊各有球員五人，共計十人。0:00	000	2
17	球隊請求暫停，有多少時間可以判用？	每次暫停給予一分鐘的時間。0:00	000	5
18	球隊的哪些成員可以離場暫停？	球隊的教練、助理教練或尚未上場的球員可以請求離場暫停時，時間為一分鐘。0:00	000	1
19	在場人犯規中有一種叫阻擋，如何判罰？	一位隊員中的球員，正企圖攔球，與一位定位或向其運動的對手發生身體接觸。0:00	000	7
20	籃球規則中有一個違例叫作非法攔球，什麼叫非法攔球？	一位球員以危險姿勢對付，同時：1、發生身體接觸時，攔球球員為非法攔球。2、在0:00	000	3
21	一名防守球員在起球之前，如何判罰？	這就是攔球犯規，即防守手手中無球時，非法攔球球員的行為。這叫攔球犯規。0:01	001	3
22	在籃球比賽中，打雷打斷比賽，什麼情況叫界外球？	1、球停止在本場區一處，或本場區三邊之間的界內處。2、球在界內區觸及隊員。0:01	001	3
23	在籃球比賽中，打雷打斷比賽，應如何判罰？	1、球停止在本場區一處，或本場區三邊之間的界外處。2、球在界內處觸及隊員。0:01	001	3
24	在籃球比賽中，哪些情況下的球會被判定為好球？	1、打雷停頓時。2、打雷沒有停頓，但球在場中。3、沒有好球或一好球。0:01	001	6
25	在籃球比賽中，哪些情況下的球會被判定為好球？	一、進攻隊投球。2、進攻隊投球時，攔球球員犯規時，裁判員可以判0:01	001	6
26	在籃球比賽中，防守球員只要攔球二名進攻球員出界？	防守的球隊以危險動作並且沒有發生攔球，則進攻隊的二名球員出界。0:01	001	3
27	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
28	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
29	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
30	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
31	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
32	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
33	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
34	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6
35	在籃球比賽中，攔球犯規的判罰？	攔球犯規：他應判出界，再把球傳向一處投球出界處。沒有攔球：他應打0:01	001	6

圖 12：ACAF 訓練問答集 (運動規則問答集)

此外，為了測試 ACAF 的效能，我們針對籃球和棒球規則各設計 5 題題目，構成測試問題共 10 題，而針對一般參考問題問答集和常問問答集亦設計 10 題測試問題，並分別以人工判斷在三訓練問答集中符合測試問題的答案。

在 ACAF 實驗設計方面，我們分為知識庫品質評估和查詢結果效能評估。在知識庫品質評估上，主要是以質化方法分析 ACAF 在學習過程中所產生的三個知識庫，而在查詢結果效能評估上，則是對上述之測試問題在 ACAF 的查詢結果做效能的估量。

第二節 評估方法

在實驗的評估方法上，我們採用 3 種評量方式，分別為排序準確度 (Total Reciprocal Document Rank, TRDR)、準確率 (Precision)、查全率 (Recall)。

TRDR 的衡量是為了確保正確的答案在系統回傳的答案中能有較高的排序，讓使用者能更快速地找到正確的答案，公式如方程式 23 所示，其中 $rank_i$ 表示第 i 個正確解答在回傳的答案中排序的位置。

$$TRDR = \sum_{i=1}^N \frac{1}{rank_i}$$

方程式 23：TRDR 計算公式[Radev02]

此外我們也採用在傳統資訊擷取上常被用來衡量搜尋系統效能的兩個準則，其一為準確率，公式如方程式 24，這是衡量在所有回傳的答案中，正確答案的比率，其中 N 表示回傳答案的個數， n 為在回傳的答案中正確的答案個數；另一為查全率，公式為方程式 25，這是衡量在使用者問題的真正解答中，有多少比率的答案能被找出來，其中 $|R|$ 表示對使用者問題來說，真正解答的個數。

$$\text{Precision} = \frac{n}{N}$$

方程式 24：準確率計算公式[Ricardo99]

$$\text{Recall} = \frac{n}{|R|}$$

方程式 25：查全率計算公式[Ricardo99]

舉例來說，若某系統 A 對問題 q 回傳的答案共有 5 個，其中第 1 個答案和第 3 個答案都是正確解答，則評量系統 A 的 TRDR 為 $\frac{1}{1} + \frac{1}{3} = 1.33$ ，若另一個系統 B 對問題 q 的回傳答案一樣也有 5 個，但是第 4 個和第 5 個答案才是正確解答，則評量此系統 B 的 TRDR 為 $\frac{1}{4} + \frac{1}{5} = 0.45$ ，由這個例子可以知道兩系統雖然都找到二個正確的答案，二系統的準確率都是 $\frac{2}{5} = 0.4$ ，但是對使用者來說，使用系統 A 能較快找到正確的解答，亦即對使用者來說，TRDR 值較高的系統，能較快速找出使用者問題的答案。

第三節 知識庫建構評估

本節主要是探討 ACAF 中三個知識庫的建構品質評估，分別是：問題與答案詞鍵關係的建構品質評估、概念空間的建構品質評估和答案類型知識庫建構品質評估。

4.3.1 問題與答案詞鍵關係的建構品質評估

ACAF 採用 LSA 的技術學習問題與答案詞鍵間的關係。由第二章第二節可知 LSA 主要分為二個步驟：(1) 將詞鍵關係矩陣予以奇異值分解，(2) 約化空間維度。其中不同程度的維度約化會影響 LSA 的學習成效，因此，為了瞭解維度約化的程度對建構問題與答案詞鍵關係的影響，在這裡我們直接考量不同程度的維度約化對問答探索系統查詢效能的作用。另一方面，為了降低問答探索系統本身在查詢效能上的變化因子，我們將 ACAF 系統簡化成只考量問題與答案詞鍵間的關係，也就是將方程式 22 中的 w_c 和 w_t 設為 0，而 w_w 設為 1，此簡化後的系統稱為 ACAF_w。

在問題與答案詞鍵關係的品質評估方面，首先分別將前述三問答集的 10 題測試問題送進 ACAF_w 中，再針對查詢結果做 TRDR、準確率和查全率的分析，以判斷維度約化對問題與答案詞鍵關係建構的影響。運動規則問答集的實驗結果如表格 7 所示，表格 8 顯示一般參考問題問答集的實驗結果，而表格 9 為常問問答集的實驗結果。表格中 DR 表示維度約化的程度，若矩陣經奇異值分解後 S 的 Rank 為 n，則 10% 的維度約化表示 S' 的 Rank 就是 $n \times 10\%$ ，DR 為 100% 則代表沒有經過維度約化，即原來的 W_q -by- W_a 矩陣；而表格中的每一個數值為所有測試問題的分析平均值。例如：表格中 10% 維度約化的 ACAF_w 之 TRDR 分析值為 0.710863095，其代表 10 題測試問題的 TRDR 平均值為 0.710863095。

DR \	TRDR	準確率 (Precision)	查全率 (Recall)
10%	0.710863095	0.317361111	0.514136905
20%	0.68907967	0.286160714	0.51860119
30%	0.635470779	0.310267857	0.420386905
40%	0.61344697	0.296291209	0.338244048
50%	0.64469697	0.318315018	0.348660714
60%	0.634280303	0.312957875	0.348660714
70%	0.655113636	0.320398352	0.359077381
80%	0.778125	0.33974359	0.390327381
90%	0.71264881	0.34672619	0.372470238
100%	0.645982143	0.31547619	0.351636905

表格 7：維度約化對問題與答案詞鍵關係建構的影響 (運動規則問答集)

DR \	TRDR	準確率 (Precision)	查全率 (Recall)
10%	0.25	0.075	0.051851852
20%	0.583333333	0.263888889	0.301851852
30%	0.638888889	0.275	0.335185185
40%	0.638888889	0.247222222	0.335185185
50%	0.638888889	0.275	0.335185185
60%	0.638888889	0.275	0.335185185
70%	0.638888889	0.275	0.335185185
80%	0.638888889	0.275	0.335185185
90%	0.638888889	0.275	0.335185185
100%	0.638888889	0.275	0.335185185

表格 8：維度約化對問題與答案詞鍵關係建構的影響 (一般參考問題問答集)

DR \	TRDR	準確率 (Precision)	查全率 (Recall)
10%	0.041666667	0.033333333	0.333333333
20%	0.111111111	0.041666667	0.333333333
30%	0.333333333	0.047619048	0.333333333
40%	0.333333333	0.047619048	0.333333333
50%	0.388888889	0.095238095	0.444444444
60%	0.388888889	0.089285714	0.444444444
70%	0.375	0.084656085	0.444444444

80%	0.375	0.084656085	0.444444444
90%	0.375	0.084656085	0.444444444
100%	0.375	0.084656085	0.444444444

表格 9：維度約化對問題與答案詞鍵關係建構的影響 (常問問答集)

由表格 7 可知：經 70% 90%的維度約化後，重建之問題詞鍵與答案詞鍵關係矩陣，能使 $ACAF_w$ 有較佳的系統效能。TRDR 值由原來的 0.64598 提高至 0.65511 0.77813，最高的增幅可達 20%；準確率從原來的 0.31548 提高到 0.32040 0.34673，增幅最高可達約 10%；而查全率也由原來的 0.35164 提昇為 0.35908 0.39033，增幅最大可到約 10%。

此外，經 10%的維度約化後，重建之問題詞鍵與答案詞鍵關係矩陣，使 $ACAF_w$ 的 TRDR 值提昇至 0.711、準確率提高為 0.3174、查全率也改善為 0.5141，相較於 70% 90%的維度約化，雖然其系統效能稍佳，但仔細觀察 10%維度約化後之重建矩陣發現：此矩陣過度約化，導致此矩陣包含過多的雜訊，使其表示之詞鍵關係已過度失真。因此，以問題詞鍵與答案詞鍵關係矩陣表達的意義看來，10%維度約化後，重建之問題詞鍵與答案詞鍵關係矩陣，儘管能 $ACAF_w$ 的系統效能，但已失去其代表的意義。

由表格 8 可知：經過維度約化後，重建之問題詞鍵與答案詞鍵關係矩陣，無法提高 $ACAF_w$ 的系統效能，這是因為一般參考問題問答集中每一組問答組所屬之領域的差異過大，造成 LSA 的學習效能不佳！

由表格 9 可知：經 50% 60%的維度約化後，重建之問題詞鍵與答案詞鍵關係矩陣，能使 $ACAF_w$ 有較佳的系統效能。TRDR 值由原來的 0.375 提高至 0.389，增幅為 3.73%；準確率從原來的 0.0847 提高到 0.0893 0.0952，增幅最高可達約 12.4%；而查全率則維持在 0.444。

由上述的實驗結果可得到：只要能控制問答集涵蓋的範圍，使其不要太過於零散，則運用 LSA 的技術就能成功的推導出問題詞鍵與答案詞鍵的關係，利用此詞鍵間的關係也順利地提昇問答探索系統的效能。

4.3.2 概念空間的建構品質評估

在概念空間的建構方面，ACAF 也是利用 LSA 的技術，同樣地，不同程度的維度約化也會影響建構概念空間的品質。又因為 ACAF 是利用分群法來建構每一個概念，所以不同的門檻值也會建構出不同品質的概念空間。

對於概念空間的品質評量，我們採用群間相似度 (inter-cluster) 和群內相似度 (intra-cluster)。群間相似度是衡量每一個概念與其它概念的相似程度，公式如方程式 26(a)所示；而群內相似度是評估在同一概念中詞鍵間的相似程度，公式如方程式 27(a)所示。因此我們可以知道：「群間相似度的值越低、群內相似度的值越高」表示該概念空間的品質就越精良。

$$(a) \text{ inter - cluster}(CS) = AVE_{\bar{c}_i, \bar{c}_j \in CS} (Sim_{\bar{c}_i, \bar{c}_j})$$

$$(b) \text{ inter - cluster}_{SD}(CS) = SD_{\bar{c}_i, \bar{c}_j \in CS} (Sim_{\bar{c}_i, \bar{c}_j})$$

方程式 26：inter-cluster 計算公式

$$(a) \text{ intra - cluster}(CS) = AVE_{c \in CS} (AVE_{w_i, w_j \in c} (Sim_{w_i, w_j}))$$

$$(b) \text{ intra - cluster}_{SD}(CS) = AVE_{c \in CS} (SD_{w_i, w_j \in c} (Sim_{w_i, w_j}))$$

方程式 27：intra-cluster 計算公式

在方程式 26(a)與方程式 27(a)中，AVE 表示平均 (Average)，而 CS 表示所建構出來的概念空間， \bar{c}_i 和 \bar{c}_j 表示概念 c_i 和 c_j 的群中心，群中心通常是利用在該概念中最具代表性的詞鍵來表示，或是以該概念中所有詞鍵表示法的平均來表示，在這裡我們採用第一種方式，是利用該概念中與其他詞鍵平均相似度

最高的詞鍵來表示該概念的群中心。另一方面，因概念的群中心也是一個詞鍵，因此二群中心的相似度 $Sim_{\vec{c}_i, \vec{c}_j}$ 的計算方式可直接套用方程式 13 即可。

除了使用平均來度量概念空間的群間相似度和群內相似度外，本論文另外採用了標準差來做為計算群間相似度和群內相似度時相似度分佈的指標，分別如方程式 26(b)和方程式 27(b)所示，其中 SD 表示標準差 (Standard Deviation)。加入這兩個公式的評量是因為經過 LSA 後詞鍵表示法的向量元素值會有所改變，此時單單只是看平均的話，會因向量的每一個元素值同時變大而相似度計算會跟著變大，所以群間相似度和群內相似度也隨著變大，這時候加上方程式 26(b)和方程式 27(b)有關標準差的考量，就能評估這些兩兩的相似度的分散程度。當同時考量方程式 26(a)、(b)和方程式 27(a)、(b)後，我們可以知道：對於一個品質精良的概念空間，其概念間相似度的值都會很小，所以群間相似度的值應該很小，而且 $inter-cluster_{SD}$ 的值也會較低；另一方面，每一個概念中兩兩詞鍵相似度的值都會很大，所以群內相似度的值應該很大，而且 $intra-cluster_{SD}$ 的值也會較低。

表格 10、表格 11 和表格 12 為分別以運動規則、一般參考問題和常問題為問答集，顯示不同的維度約化和不同的門檻值對概念空間建構品質的影響，表格中 Threshold 代表平均連結分群法的門檻值，這裡考量了三種不同的門檻值，分別為 0.6、0.7 和 0.8；此外，intra 表示 intra-cluster； $intra_{SD}$ 代表 $intra-cluster_{SD}$ ；inter 表示 inter-cluster；而 $inter_{SD}$ 代表 $inter-cluster_{SD}$ ；表格中淺色的部分表示比原矩陣有較佳的結果。

Analysis \ DR		5%	10%	20%	30%	40%	50%	60%	70%	100%
		5%	10%	20%	30%	40%	50%	60%	70%	100%
Threshold=0.8	intra	0.8618	0.8101	0.6329	0.5458	0.4860	0.4493	0.4322	0.4279	0.3746
	$intra_{SD}$	0.8295	0.6132	0.4322	0.3563	0.3173	0.2930	0.2798	0.2721	0.2081
	inter	0.2931	0.1674	0.1075	0.0918	0.0827	0.0766	0.0741	0.0735	0.0836

	inters_{SD}	0.3036	0.2402	0.1810	0.1614	0.1513	0.1442	0.1414	0.1406	0.1451
Threshold=0.7	intra	0.7826	0.7975	0.7743	0.7274	0.6638	0.6332	0.6176	0.6178	0.5768
	intra_{SD}	0.7897	0.7308	0.6008	0.5143	0.4654	0.4182	0.3977	0.3943	0.3247
	inter	0.3715	0.1785	0.0930	0.0821	0.0743	0.0679	0.0660	0.0660	0.0787
	inters_{SD}	0.3074	0.2597	0.1815	0.1615	0.1504	0.1416	0.1382	0.1389	0.1445
Threshold=0.6	intra	0.7046	0.7158	0.7330	0.7341	0.7192	0.7096	0.7012	0.6977	0.6725
	intra_{SD}	0.7168	0.7063	0.6745	0.6465	0.5954	0.5634	0.5369	0.5313	0.4728
	inter	0.4550	0.2056	0.0934	0.0708	0.0612	0.0578	0.0563	0.0570	0.0709
	inters_{SD}	0.2963	0.2884	0.1934	0.1640	0.1481	0.1411	0.1376	0.1377	0.1459

表格 10：維度約化和不同的門檻值對概念空間建構品質的評量表 (運動規則)

由表格 10 可知：Word-By-Document 矩陣經過 LSA 後所得到的詞鍵表示法應用在概念空間的建構上，所得到的 intra-cluster 評量結果都比原來未經過 LSA 有較好的結果。當門檻值為 0.8 時，intra-cluster 評估平均增幅在 55.01% 左右；在門檻值為 0.7 時，intra-cluster 評估平均增幅約 21.67%；而門檻值為 0.6 時，intra-cluster 評估平均增幅在 6.24% 左右。

以 inter-cluster 評量來說，在門檻值為 0.8 時，只有 40% 70% 有較好的結果，平均改善程度約 8.25%；當門檻值為 0.7 時，只有 40% 70% 有較好的結果，平均改善程度在 12.86% 左右；而門檻值為 0.6 時，只有 30% 70% 有較好的結果，平均改善程度約 14.50%。

若再輔以 intra-cluster_{SD} 和 inter-cluster_{SD} 綜合評估下，可以得到：當門檻值為 0.8 或 0.7、維度約化為 50% 70% 時，所建的概念空間有較佳之品質評估；當門檻值為 0.6、維度約化為 40% 70% 時，所建的概念空間有較佳之品質評估。因此，在 intra-cluster 和 inter-cluster 的評估上，LSA 能改進概念空間的建構品質。

若由概念空間的內容來看，以 40% 和 50% 之維度約化和門檻值為 0.7 所建構的概念空間為例，「驅逐」和「出場」二詞鍵在沒有維度約化和門檻值為 0.7

的概念空間中，分別屬於不同概念，但是在 40% 和 50% 的維度約化和門檻值為 0.7 的概念空間中，這二個詞鍵是屬於同一個概念。而我們知道在運動規則「驅逐」某人和請某人「出場」是代表相同的意思，因此這二個詞應屬於相同的概念。由此可知：利用 LSA 能改良概念空間的建構品質。

DR \ Analysis		5%	10%	20%	30%	40%	50%	60%	70%	100%	
		Threshold=0.8		intra	0.2583	0.4107	0.5275	0.5924	0.6623	0.6734	0.6274
		intra _{SD}	0.2509	0.3908	0.4914	0.5107	0.5210	0.5159	0.4512	0.3986	0.3795
Threshold=0.7		inter	0.0037	0.0037	0.0024	0.0020	0.0022	0.0023	0.0021	0.0019	0.0016
		inter _{SD}	0.0582	0.0611	0.0478	0.0432	0.0415	0.0417	0.0391	0.0353	0.0308
Threshold=0.6		intra	0.2515	0.4032	0.5329	0.6124	0.7094	0.7477	0.7181	0.6728	0.6451
		intra _{SD}	0.2487	0.3936	0.5070	0.5503	0.5844	0.5907	0.5241	0.4705	0.4582
Threshold=0.6		inter	0.0035	0.0036	0.0023	0.0017	0.0019	0.0019	0.0018	0.0015	0.0011
		inter _{SD}	0.0575	0.0604	0.0474	0.0414	0.0402	0.0400	0.0370	0.0323	0.0254
Threshold=0.6		intra	0.2441	0.3951	0.5206	0.6023	0.7054	0.7542	0.7295	0.6931	0.6724
		intra _{SD}	0.2460	0.3890	0.5064	0.5609	0.6221	0.6297	0.5696	0.5159	0.5080
Threshold=0.6		inter	0.0035	0.0036	0.0023	0.0016	0.0017	0.0018	0.0016	0.0012	0.0007
		inter _{SD}	0.0578	0.0606	0.0478	0.0407	0.0390	0.0393	0.0358	0.0296	0.0209

表格 11：維度約化和不同的門檻值對概念空間建構品質的評量表（一般參考問題）

由表格 11 可知：Word-By-Document 矩陣經過 LSA 後所得到的詞鍵表示法應用在概念空間的建構上，40% 70% 的維度約化所得到的 intra-cluster 評量結果比原來未經過 LSA 有較好的結果。當門檻值為 0.8 時，intra-cluster 評估平均增幅在 16.8% 左右；在門檻值為 0.7 時，intra-cluster 評估平均增幅約 6.7%；而門檻值為 0.6 時，intra-cluster 評估平均增幅在 4.8% 左右。但一般參考問題問答集所建構的概念空間，在 inter-cluster 的評估上反而得到較差的數值。追究其原因，亦是因為一般參考問題問答集所涵蓋之範圍過於零散，導致 LSA 的學習效能不彰！

DR \ Analysis		20%	30%	40%	50%	60%	70%	80%	90%	100%
---------------	--	-----	-----	-----	-----	-----	-----	-----	-----	------

Threshold=0.8	intra	0.8399	0.7999	0.7413	0.6810	0.6112	0.5653	0.5518	0.3995	0.3926
	intra _{SD}	0.7620	0.6968	0.6147	0.5596	0.5155	0.4855	0.4491	0.2686	0.2679
	inter	0.0960	0.0558	0.0303	0.022	0.0191	0.0171	0.0161	0.0193	0.0191
	inter _{SD}	0.2103	0.1672	0.1308	0.1111	0.1004	0.0932	0.0898	0.0878	0.0875
Threshold=0.7	intra	0.8223	0.8381	0.8213	0.7990	0.7714	0.7440	0.7273	0.6354	0.6356
	intra _{SD}	0.7868	0.7788	0.7347	0.6777	0.6485	0.6134	0.5794	0.4153	0.4126
	inter	0.0906	0.0539	0.0276	0.0190	0.0160	0.0144	0.0138	0.0170	0.0170
	inter _{SD}	0.2164	0.1731	0.1326	0.1109	0.0995	0.0913	0.0878	0.0859	0.0855
Threshold=0.6	intra	0.7653	0.7857	0.7952	0.7909	0.7835	0.7698	0.7572	0.7044	0.7045
	intra _{SD}	0.7634	0.7894	0.7715	0.7435	0.7346	0.7271	0.6939	0.5791	0.5718
	inter	0.0937	0.0546	0.0270	0.0168	0.0136	0.0118	0.0112	0.0129	0.0131
	inter _{SD}	0.2231	0.1799	0.1363	0.1117	0.0984	0.0908	0.0868	0.0790	0.0781

表格 12：維度約化和不同的門檻值對概念空間建構品質的評量表（常問問題）

由表格 12 可知：Word-By-Document 矩陣經過 LSA 後所得到的詞鍵表示法應用在概念空間的建構上，20% 80%的維度約化所得到的 intra-cluster 評量結果比原來未經過 LSA 有較好的結果。當門檻值為 0.8 時，intra-cluster 評估平均增幅在 29.2%左右；在門檻值為 0.7 時，intra-cluster 評估平均增幅約 15.4%；而門檻值為 0.6 時，intra-cluster 評估平均增幅在 7.4%左右。

以 inter-cluster 評量來說，在門檻值為 0.8 時，只有 60% 80%有較好的結果，平均改善程度約 8.73%；當門檻值為 0.7 時，只有 60% 80%有較好的結果，平均改善程度在 13.3%左右；而門檻值為 0.6 時，只有 70% 90%有較好的結果，平均改善程度約 8.65%。

綜合上述評估，只要能控制訓練問答集涵蓋的範圍，使其不要過於零散，則在 intra-cluster 和 inter-cluster 的評估上，LSA 能改進概念空間的建構品質。

4.3.3 答案類型知識庫建構評估

在 ACAF 的答案類型知識庫中記錄了疑問詞鍵對於每一個答案類型的可能性，為了評量本論文建置疑問詞鍵對答案類型的可能性，我們根據學習機制中

得到的答案類型知識庫，計算出每一個測試問題的可能答案類型，並比較先前人工所指定的答案類型，計算出答案判別的正確率，如表格 13 所示。表格中測試問題 QN-M 表示第 N 個問題，以人工判斷的答案類型之類型代號為 M；而 ACAF 判斷之可能答案類型 M(w)代表 ACAF 判斷答案類型為 M，其權重為 w。由表格 13 可知，利用簡單的機率模組，ACAF 判斷新問題的答案類型之平均正確率可達 66.82%。

測試問題	ACAF 判斷之可能答案類型	正確率
Q1 - 7	7(1)	1.0
Q2 - 3	1(0.6)、3(0.4)	0.4
Q3 - 3	3(1)	1.0
Q4 - 3	3(0.875)、6(0.125)	0.875
Q5 - 2	1(0.5)、2(0.5)	0.5
Q6 - 3	3(1)	1.0
Q7 - 3	1(0.6)、3(0.4)	0.4
Q8 - 3	1(0.6)、3(0.4)	0.4
Q9 - 1	1(0.25)、2(0.25)、3(0.2)、6(0.3)	0.25
Q10 - 6	5(0.142857)、6(0.857143)	0.857143
		0.6682

表格 13：答案類型知識庫建構評估表

第四節 ACAF 查詢結果評估

這一節主要是討論 ACAF 對於使用者新問題的查詢結果，利用第二節提到的 TRDR、準確率和查全率三個評估方法來證明概念空間和答案類型在問答探索系統上的可行性。

在實驗的設計上，本論文以傳統的資訊擷取技術為基準 (Base Line)，並將 ACAF 分為 3 個階段來評估 ACAF 系統在 3 個階段的改善程度。在資訊擷取技術方面，先將測試問題視為查詢字串，經過斷詞切字後直接比對先前對訓練問答集所建置的索引，找到符合測試問題的答案。本論文針對訓練問答集的 3 種

不同範圍建置索引，分別是問題 (Question)、答案 (Answer) 和問答 (Q & A) 部分。在建置問題索引時，把訓練問答集中的每一問題視為一篇文件，因此訓練問答集的問題部分就形成了資訊擷取中的文件集；而答案索引在建置時，是將訓練問答集中的每一答案視為一篇文件，所以訓練問答集的答案部分就構成了資訊擷取中的文件集；至於建構問答索引時，則是把訓練問答集中的每一個問題和相對應的答案(即問答組)視為一篇文件，故訓練問答集中的所有問題和相對應的答案就組成了資訊擷取中的文件集。確立了文件集後，再套用資訊擷取的技術，即可建構該文件集的索引。

在 ACAF 系統方面，本論文考量三個階段的 ACAF 系統。首先在第一個階段中，僅根據問題與答案詞鍵間的關係，來衡量測試問題與訓練問答集中的答案相似度，為了標記方便，稱此一系統為 ACAF_w；其次在第二階段，除了考量問題與答案詞鍵間的關係外，還加入了概念空間的概念描述，讓此問答探索系統同時考慮測試問題和訓練問答集中的答案在詞鍵間的關係和概念描述的正確性，稱此系統為 ACAF_{wc}；最後，在第三階段中將 ACAF_{wc} 系統加入了答案類型的判別，即本論文所發展的系統 ACAF。表格 14、表格 15 和表格 16 分別為 ACAF 以運動規則、一般參考問題和常問問題為問答集針對先前所述之測試問題的查詢結果效能評估表。

		TRDR	準確率 (Precision)	查全率 (Recall)
資訊擷取 (IR)	Question	0.3542	0.1979	0.1210
	Answer	0.3125	0.1458	0.0595
	Q & A	0.1250	0.0833	0.0417
ACAF _w		0.7781	0.3397	0.3903
ACAF _{wc}		0.7893	0.3595	0.4082
ACAF		0.8372	0.3679	0.4332

表格 14：ACAF 查詢結果效能評估表 (運動規則問答集)

		TRDR	準確率 (Precision)	查全率 (Recall)
資訊擷取 (IR)	Question	0.5833	0.2778	0.3019
	Answer	0.5000	0.2500	0.0796
	Q & A	0.8333	0.3611	0.3296
ACAF _w		0.6389	0.275	0.3352
ACAF _{wc}		0.8889	0.3583	0.3630
ACAF		0.8889	0.4167	0.4463

表格 15：ACAF 查詢結果效能評估表 (一般參考問題問答集)

		TRDR	準確率 (Precision)	查全率 (Recall)
資訊擷取 (IR)	Question	0.1667	0.0833	0.0152
	Answer	0.2500	0.1389	0.1875
	Q & A	0.1667	0.0833	0.0152
ACAF _w		0.3889	0.0893	0.4444
ACAF _{wc}		0.8889	0.3750	0.5051
ACAF		0.9000	0.3472	0.5051

表格 16：ACAF 查詢結果效能評估表 (常問問答集)

由表格 14 可知：以運動規則為問答集，在運用資訊擷取技術的三個系統(分別為 Question、Answer 和 Q&A)中，利用問題索引的查詢效能在 TRDR、準確率和查全率的評估有較好的表現；由表格 15 可知：以一般參考問題為問答集，在運用資訊擷取技術的三個系統中，利用問答索引的查詢效能在 TRDR、準確率和查全率的評估有較好的表現；由表格 16 可知：以常問問題為問答集，在運用資訊擷取技術的三個系統中，利用答案索引的查詢效能在 TRDR、準確率和查全率的評估有較好的表現。然而相較於 3 個階段的 ACAF 系統，ACAF_w、ACAF_{wc} 和 ACAF 在三個問答集的評估中，均遠優於資訊擷取系統。

以運動規則為問答集之詞鍵間關係的實驗可知：70% 80%的維度約化能得到較佳的詞鍵間關係矩陣，此外，依概念空間的實驗結果，ACAF_{wc} 和 ACAF

採用門檻值為 0.6 和 40%的維度約化所建構之概念空間。由表格 14 可知：不論是 TRDR、準確率或是查全率評估， $ACAF_{WC}$ 的查詢效能優於 $ACAF_W$ ，而 $ACAF$ 又優於 $ACAF_{WC}$ 。

以一般參考問題為問答集之詞鍵間關係和概念空間的實驗可知：雖然 LSA 並不能改善一般參考問題的問題詞鍵和答案詞鍵間關係矩陣亦不能改良概念空間的品質，但由表格 15 可知：不論是 TRDR、準確率或是查全率評估， $ACAF_{WC}$ 的查詢效能優於 $ACAF_W$ ，而 $ACAF$ 又優於 $ACAF_{WC}$ 。

以常問問題為問答集之詞鍵間關係的實驗可知：50% 60%的維度約化能得到較佳的詞鍵間關係矩陣，此外，依概念空間的實驗結果， $ACAF_{WC}$ 和 $ACAF$ 採用門檻值為 0.7 和 60%的維度約化所建構之概念空間。由表格 16 可知： $ACAF$ 在 TRDR 的評量上有最佳的查詢效能， $ACAF_{WC}$ 次之，而 $ACAF_W$ 與前二系統相較，其 TRDR 只有 0.3889，而加入概念的比對後，增幅高達 129%；另在準確率的評估上， $ACAF_{WC}$ 有最佳的查詢效能， $ACAF$ 次之，而 $ACAF_W$ 則僅有 0.0893；最後在查全率的估量上， $ACAF_W$ 加入概念比對之 $ACAF_{WC}$ ，其查全率由 0.4444 提昇至 0.5051，而 $ACAF_{WC}$ 加入答案類型的判別之 $ACAF$ ，其查全率仍維持 0.5051。

由上述三種問答集之查詢效能可知：導入概念空間的概念描述和答案類型的判別，不僅能使回傳的結果更精確，也能找到較多相關答案。

第五章 結論與未來研究方向

本章總結本論文並說明未來的研究方向。第一節討論本論文所開發的概念式自動問答探索系統將概念空間和答案類型判別納入自動問答探索系統的效益、優點和限制；第二節說明未來可能的研究方向。

第一節 結論

本論文結合了問題詞鍵與答案詞鍵間的關係、問題的概念空間以及答案類型判別的方法，成功地發展一套概念式自動問答探索系統。ACAF 主要運用 LSA 探勘問題詞鍵和答案詞鍵間的隱含關係，並學習出詞鍵在文件中隱含的語意關係，以建立問題的概念空間。此外，利用一套簡單的機率模組，學習疑問詞鍵對某答案類型的可能性。

在實驗的過程中發現：若問答集中的每一個問題與彼此間所描述的領域差異性過大，且每一個關鍵詞鍵僅出現於一問答組中，則 LSA 推導的問題詞鍵與答案詞鍵間之隱含關係，無法改善 $ACAF_w$ 的系統效能；此外，利用 LSA 建立的問題概念空間，也因從搜尋引擎收集到的相關文件集中，文件彼此間描述的領域差異性過大，導致無法改善此概念空間的品質。

然而，若控制問答集中問題描述的領域差異性，由實驗結果可以知道，LSA 能推演出問題詞鍵與答案詞鍵間的隱含關係，使 $ACAF_w$ 系統的擷取效能不管是在準確率、查全率或是 TRDR 的評估上都有明顯的改進。此外利用 LSA 在推導文件中詞鍵的隱含語意之能力，ACAF 針對問題建立概念空間，使自動問答探索系統能解決多詞一義的問題，讓系統能處理使用者在問題描述上的變異性，將系統提昇至語意層面，而由實驗也證明了加入概念的描述評估，提昇了 $ACAF_{wc}$ 的擷取效益。最後，利用簡單的機率模組將 $ACAF_{wc}$ 系統加入答案類型的判別功能，針對可能的答案做篩選，也讓最後的概念式自動問答探索系統

(ACAF System) 在擷取效能評估上優於之前的系統，讓系統所找到的答案更符合使用者的需求！

第二節 未來研究方向

本論文利用 LSA 推導出問題詞鍵和答案詞鍵的隱含關係，用來搜尋問題的可能答案詞鍵，也運用 LSA 推演文件集中詞鍵間的語意關係，用來建立問題的概念空間，此外也導入簡單的機率模組，計算疑問詞鍵對答案類型的可能性，用來判別問題的答案類型。藉由結合這三方面的技術，不僅使本論文所開發的 ACAF 系統提昇到知識的層級，也提高系統的擷取效能。

以下提出我們的未來研究發展方向：

1. 在 ACAF 的探索機制，本論文只考慮了問題詞鍵和答案詞鍵間出現的關係，但以翻譯模組的技術考量，在做兩種語言之翻譯時還要考慮先後的順序，因此，未來擬加入問題詞鍵和答案詞鍵出現的先後順序的評量，以增加 ACAF 的系統搜尋效能。
2. 在斷詞切字方面，本論文僅比對詞庫切出可能的詞鍵，然而中文詞鍵間的分界是模糊不清的，利用比對詞庫的方法並不能正確地斷詞，因此斷詞切字的結果可能會有錯誤，進而導致問題詞鍵與答案詞鍵間關係的推導發生錯誤，也影響了概念空間的建構。未來我們將導入中文語句結構的分析，增加斷詞的正確性。
3. 在 LSA 的分析過程中，維度約化是 LSA 的成功關鍵，在實驗時，是利用各種維度約化的可能性尋找最佳的維度約化。未來，擬藉由分析原始資料，找出資料特性與最佳維度約化間的關係，減少尋找最佳維度約化的時間。

4. 在答案類型判別上，本論文僅利用問題的疑問詞鍵做判斷。未來，會加入問題的語句結構之分析，以增加答案類型的判別正確率。



參考文獻

1. [Aggarwal01] C.C. Aggarwal and P.S. Yu, “On Effective Conceptual Indexing and Similarity Search in Text Data,” *Proc. IEEE International Conference on Data Mining*, pp. 3-10, 2001.
2. [Bellegarda96] J.R. Bellegarda, J.W. Butzberger, Y.L. Chow, N.B. Coccaro, and D. Naik, “A Novel Word Clustering Algorithm Based On Latent Semantic Analysis,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 172-175, 1996.
3. [Berger00] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, “Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding,” *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 192-199, 2000.
4. [Brown90] P. Brown, J. Cocke, S.D. Pietra, V.D. Pietra, F. Jelinek, J. Lafferty, R. Mercer and P. Roossin, “A Statistical approach to machine translation,” *Computational Linguistics*, vol. 16, No. 2, pp. 79-85, 1990.
5. [Chung99] M. Chung, Q. He, K. Powell and B. Schatz, “Semantic Indexing for a Complete Subject Discipline,” *Proc. of the fourth ACM conference on Digital libraries*, pp. 39-48, 1999.
6. [Dumais02] S. Dumais, M. Banko, E. Brill, J. Lin and A. Ng, “Web Question Answering: Is More Always Better?,” *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 291-298, 2002.
7. [Fu01] W. Fu, B. Wu, Q. He and Z. Shi, “Text Document Clustering and the Space of Concept on Text Document Automatically Generated,” *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing, 2001 International*

- Conferences on*, vol. 3, pp. 107-112, 2001.
8. [Gose96] E. Gose, R. Johnsonbaugh and S. Jost, *Pattern Recognition and Image Analysis*. Prentice Hall, 1996.
 9. [Halkidi02] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "Clustering Validity Checking Methods: Part 1," *ACM SIGMOD Record*, vol. 31, No. 3, pp. 19-27, 2002.
 10. [Jain99] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, No. 3, pp. 264-323, 1999.
 11. [Kwok01] C. Kwok, O. Etzioni and D.S. Weld, "Scaling Question Answering to the Web," *ACM Trans. on Information Systems*, vol. 19, No. 3, pp. 242-262, 2001.
 12. [Landauer98] T.K. Landauer, P.W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, pp. 259-284, 1998.
 13. [Lin01] D. Lin and P. Pantel, "Induction of Semantic Classes from Natural Language Text," *Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 317-322, 2001.
 14. [Lin02] S.H. Lin, M.C. Chen, J.M. Ho and Y.M. Huang, "ACIRD: Intelligent Internet Document Organization and Retrieval," *IEEE Trans. On Knowledge and Data Engineering*, vol. 14, No. 3, pp. 599-614, 2002.
 15. [Google03] <http://www.google.com>, Google, 2003.
 16. [Nei99] J.Y. Nie, F. Ren, "Chinese information retrieval: using characters or words?," *Information Processing & Management*, vol. 35, No. 4, pp. 443-462, 1999.
 17. [Park96] C.P. Park and K.S. Choi, "Automatic Thesaurus Construction Using Bayesian Networks," *Information Processing and Management*, vol. 32, No. 5, pp. 543-553, 1996.

18. [Pasca01] M. Pasca and S.M. Harabagiu, "Answer Mining from On-Line Documents," *Proc. of the ACL-2001 Workshop on Open-Domain Question Answering*, pp. 38-45, 2001.
19. [Pasca01] M.A. Pasca and S.M. Harabagiu, "High Performance Question/Answering," *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 366-374, 2001.
20. [Pinto02] D. Pinto, M. Branstein, R. Coleman, W.B. Croft, M. King, W. Li and X. Wei, "QuASM: A System for Question Answering Using Semi-Structured Data," *Proc. of the 2nd ACM/IEEE-CS joint conference on Digital Libraries*, pp. 46-55, 2002
21. [Prager00] J. Prager, E. Brown and A. Coden, "Question-Answering by Predictive Annotation," *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 184-191, 2000.
22. [Radev01] D. Radev, H. Qi, Z. Zheng, B.G. Sasha, Z. Zhang, W. Fan and J. Prager, "Mining the Web for Answers to Natural Language Questions," *Proc. of the 10th International Conference on Information and Knowledge Management*, pp. 143-150, 2001.
23. [Radev02] D. Radev, W. Fan, H. Qi, H. Wu and A. Grewal, "Probabilistic Question Answering," *Proc. of the 11th World Wide Web Conference*, pp. 408-419, 2002.
24. [Ricardo99] B.Y. Ricardo, R.N. Berthier, *Modern Information Retrieval*. Addison-Wesley, 1999.
25. [Sugumaran02] V. Sugumaran and V.C. Storey, "Ontologies for Conceptual Modeling: their Creation, Use, and Management," *Data and Knowledge*

- Engineering*, vol. 42, No.3, pp. 251-271, 2002.
26. [Yahoo!03] <http://tw.yahoo.com>, Yahoo! Taiwan Inc., 2003.
 27. [Yam03] <http://www.yam.com>, Yam Digital Technology, 2003.
 28. [Yang00] Y. Yang, T. Ault, T. Pierce and C.W. Lattimer, "Improving Text Categorization Methods for Event Tracking," *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 65-72, 2000.
 29. [Zelikovitz01] S. Zelikovitz and H. Hirsh, "Using LSI for Text Classification in the Presence of Background Text," *Proc. of the 10th International ACM Conference on Information and Knowledge Management*, pp. 113-118, 2001.

附錄：ACAF 搜尋流程 (以常問問答集為例)

在此附錄中以一測試問題來說明整個 ACAF 的搜尋流程。在訓練問答集的部分採用常問問答集 (如圖 13 所示), QNum 為問答組的編號, QContent 為問題內容, QReplyCont 為答案內容, AnswerType 為問題的答案類型。

QNum	QContent	QReplyCont	AnswerType
1	海運可否申請借書證?	亞太教職員工及學生之圖書借書證可以借書, 有章者, 請至圖書館二樓借書證台辦理借書證, 其借書用款併入該教職員工等生之借書用款, 有問題者請電(03)5712121轉52636或52658。	3
2	校友及其眷屬可否借書?	校友可逕向本校校友會辦理校友證。憑「校友證」即可至圖書館辦理借書, 最多可借10本, 借書期限30天, 得以續借一次, 並可至本館資訊中心使用視聽媒體。校友配偶及滿十八歲之直系親屬, 經校友同意可申請專屬借書證, 借書用款計入校友借書用款。校友(含眷屬)借書逾期違者, 每天每本罰金5元, 其他借閱規定, 均比照本館相關借閱閱覽規定辦理。本校校友會在圖書館地下一樓, 電話: (03)5712121轉54459。	3
3	請問預約書到館多久後去借, 會被取消預約書的借書權利?	預約書到館通知書後, 保留七天, 如在七天內未借出, 則將借下一位預約的讀者依序上至書庫。	1
4	借閱的圖書何時可以辦理續借?	讀者續借圖書可在到期日前五天內(含到期日當日)至借閱書櫃台或線上辦理續借, 但需符合下列三個條件: 1. 讀者目前所借圖書未逾期。2. 未被凍結借借。3. 所要續借的圖書未被其他人預約。	1
5	我要如何才能「再借書」已借書一次的圖書?	圖書館為了維護每位讀者對資料使用公平機會, 所以規定每位讀者對同一本圖書只有借借一次的權利, 但若借借一次尚無法看完, 想再借借一次, 建議您可以先將該圖書歸還, 若當時無人預約, 可重新再「借書」一次。	7
6	尚未註冊的進修研究生如何在圖書館借書?	本校進修研究生, 在尚未註冊前, 可至圖書館網頁上圖書館服務項目下點選「研究所新生臨時借書號碼申請」, 填寫相關資料(姓名、學號、身份证号、e-mail address, 請必要填寫), 即可向系所助理領取臨時借書號碼, 將號碼貼在附有照片的證件上, 即可至本館借書。若於註冊前離校, 需歸還所有圖書。	7
7	校外讀者可否借書?	凡年滿二十歲之市民, 均可至本館二樓借書證台申請市民借書證, 每年年費2000元, 保證金5000元, 或可找本校正式教職員工當保證人即可免保證金, 若有問題請電(03)5712121轉52636或52658。	3
8	清華大學、中央大學、陽明大學師生可否借書?	本館與清華大學、中央大學、陽明大學圖書館實施一體適用, 四校師生均可憑自己本校服務證或學生證至對方借書, 借書用款別用, 借書期限30天, 可續借一次。	3

圖 13：ACAF 訓練問答集 (常問問答集)

以測試問題「如何延後所借的書之到期日?」為例, 此測試問題的相關答案為圖 13 中 QNum 為 4、5 和 32 的三篇答案文件。測試問題經過斷詞切字後, 得到關鍵詞鍵為「延後」、「到期日」, 疑問詞鍵為「如何」。因「延後」和「到期日」並沒有出現在訓練問答集中, 因此找不到與問題關鍵詞鍵相關的答案詞鍵, 故 ACAF_w 找不到任何符合的答案, TRDR、準確率和查全率的值均為 0!

利用「延後」和「到期日」, 參考問題的概念空間可以找到包含二詞鍵的概念有 5 個概念, 分別為 Concept81、Concept308、Concept318、Concept347、

Concept143，因此加入了概念的比對，就能找到符合的答案(搜尋結果如圖 14 所示)，TRDR、準確率和查全率分別提升至 1.5、1 和 0.67。



圖 14：ACAF_{WC}的搜尋結果

利用疑問詞鍵「如何」，參考答案類型判別知識庫(如圖 15 所示)，得到此測試問題可能的答案類型代號為 7，因此加上答案類型判別的 ACAF 系統對此測試問題所找到的答案如圖 16 所示。ACAF 系統會將 ACAF_{WC} 擷取出的答案做答案類型的篩選，因此對 QNum 為 5 的答案文件，其答案類型符合測試問題的答案類型，因此 QNum 為 5 的答案文件對測試問題的權重仍為 1；而對 QNum 為 4 的答案文件來說，其答案類型代號為 1，因此 QNum 為 4 的答案文件對測試問題的權重降為 0.73。儘管如此，因回傳的文件仍為 QNum5 和 QNum4，因此 TRDR、準確率和查全率仍然不變。

Term	AnswerType	probability
為什麼	0	1
何時	1	1
多久	1	1
可否	3	1
是否	3	1
何處	4	1
哪些人	5	1
誰	5	1
如何	7	1
怎麼辦	7	1
人物	5	1
什麼	3	0.5
什麼	6	0.5
什麼地方	4	1
什麼時候	1	1
什麼時間	1	1

圖 15：答案類型判別知識庫 (常問問答集)

搜尋結果

答案類型：7

排名	QNum	問題	答案	權重
1	5	我要如何才能「再續借」已續借一次的圖書？	圖書館為了維護每位讀者對資料使用公平機會，所以設定每位讀者對同一本圖書只有續借一次的權利，但若續借一次尚無法看完，想再續借一次，建議您可以先將該圖書歸還，若當時無人預約，可重新再「借書」一次。	1
2	4	借閱的圖書何時可以辦理續借？	讀者續借圖書可在到期日前五天內（含到期日當日）至借還書櫃台或線上辦理續借，但需符合下列三個條件：1. 讀者目前所借圖書皆未逾期。2. 未被凍結停借。3. 所要續借的圖書未被其他人預約。	0.732716800086

[回首頁](#)

圖 16：ACAF 的搜尋結果