私 立 東 海 大 學

資訊工程與科學研究所
碩 士 論 文

以階層式詞義網路為基礎的中文文件分析及其效能評估

指導教授：呂芳懌博士
研 究 生：施 政 瑋

中 華 民 國 九 十 二 年 七 月

中文摘要

本論文係以中文詞彙所蘊含的概念（concept）為基礎，探討詞彙之間的相互關係（relationship），並將中文同義詞集合（synset）依照詞彙概念，建立一個以同義詞結構為基礎的詞庫。之後配合各種資訊檢索技術（information retrieval），包括中文斷詞與文法規則、自然語言處理、查詢處理與延伸、關鍵詞索引建置（index construction）、文件分類（document clustering）等，設計出一套以概念為檢索條件的文件分析法，除了可以自然語言輸入查詢語句，以剖析關鍵詞的基礎之外，也能以詞義鏈（lexical chain）的方式進行全文檢索。本論文採用向量空間模組（Vector space model ）作為比較其績效之基準，我們將以特定領域的文件及所用術語，對文件內容作檢索並依相似程度排序，最後以召回度（recall）及精確度（precision）作為評估兩者之間效率及正確性的指標。

關鍵詞：資訊檢索、文件檢索、同義詞集合、相似性比對。

Abstract

In this paper we research the mutual semantic relationship between terms via term concepts. We collect Chinese synonyms for building a synonyms thesaurus, and analyze documents with information retrieval tech like Chinese word tagging, natural language processing（NLP）, query processing, index constructing, etc. and build a semantic based document retrieval system.. It not only can handle keyword-based query, it can analyze documents by using lexical chain instead of keyword search to improve the accuracy in document retrieval. Besides, we evaluate retrieval performance with vector space model. Finally, we examine the system with document in certain domain and evaluate performance of the algorithm with recall and precision degree.

Keyword：Information retrieval, Document retrieval, Synset, Similarity comparison.

誌謝

首先最感謝的是我的指導教授呂芳懌老師，在我碩士班的修業期間，除了給予我論文研究方面的創意啟發以及寫作技巧的指導外，在表達能力、實事求是的精神與以及嚴謹的研究態度更是我所學習到最寶貴的資產，而林宜隆老師、陳澤雄老師、黃其泮老師、賴森堂老師，在論文口試時給予我許多寶貴的意見，使我的論文能更完整。

其次要感謝的，是所上的同學及學弟們，包括義樺、俊維、嘉鴻、清健、真真、仁傑、坤義、國榮、耀中、國煒、以及其他一起努力、一起打拼的夥伴，在他們身上，我學習到許多的知識及道理，更感受到來自他們的真摯友誼，藉由相互的鼓勵，使我能更積極的面對漫長的研究生活。

最後，我想要感謝我的家人，是他們在背後一直默默的支持我、容忍我，我才能順利的完成研究所的學業，迎向下一階段的挑戰。

僅以此表達最深的感謝

施政瑋　于　東海大學
七月　九日　二００三

目錄

圖目錄

表目錄

第 1 章　緒論

1.1　研究動機與目的

電腦的發明，可說是人類文明在二十世紀最重要的進展。半個世紀以來，靠著電腦強大的計算及儲存能力，人類得以處理許多自身能力所不能及的問題。而電腦的普及，使得人類的生活品質一日千里。尤其在網路興起之後，電腦更成為最新最方便的傳播媒介，人們可以藉由網路自由表達意見、傳遞資料及分享知識，而不再只是被動地經由傳播媒體單向地接收資訊，一方面，又因無時間空間限制的特性，網路儼然已成為人類最大的資料來源。

照理說，網路上的資訊越豐富，對於人類的貢獻也會越大，但事實上並非如此。舉一個圖書館的例子，圖書館除了藏書以外，對每一本藏書的書目索引及分類，才是讓使用者能輕易找到所需圖書的關鍵；試想一個沒有書目索引的圖書館，因為無法快速找到所需的資料，就算藏書量再多，對於讀者的意義不大。現今網路就有些類似的情況，網路資源的無限度地擴張，使得要找到正確且切合需求資料的困難度大幅提高。雖然網路資源不比圖書館館藏，有著具體的範圍，其增長的速度更是超出我們的想像，但若是能提供一個資訊索引的機制，對使用者來說，無疑可以節省大量的時間及精力，於是，各類型的資料搜尋系統因應而生。各資料搜尋系統的原理雖有若干差異，但皆以事先擷取及分析資料內容的方式，分類其涵蓋範圍內的所有資料。等到使用者查詢時，即回傳所有符合需求的資料位址，讓使用者能快速找尋到需要的資訊。而本研究的目的亦以此為基礎，發展出一套以詞義概念為基礎的搜尋方式，做法是將所有詞彙依其意義分群，並以關鍵詞所蘊含的概念取代關鍵詞，計算各概念在文件中所佔的比重，以為查詢的依據，希望能以「詞義」的角度，去除不符合查詢的文件，以得到更高的精確度；並應用資訊檢索（information retrieval）技術，將資料庫中的文章，以使用者提出的查詢文字進行概念上的比對，再依相似程度排序，讓使用者能快速而精確的找到所需要的資訊。

1.2 相關研究

現今查詢系統，包括各大搜尋引擎的查詢方式，皆是由使用者對搜尋引擎輸入一個或數個關鍵字（key word），搜尋系統內部對於資料的分類也是以資料內含的關鍵字，作為分類的依據。然而，在系統所蒐集的資料快速增加的情況下，搜尋引擎回傳給使用者的查詢結果，往往也不再是使用者可以一一檢視瀏覽的數量。資料數量一旦大量增加，則可能出現兩個問題：

一、 由於使用者只以數個關鍵字作為查詢條件，符合使用者提出關鍵字的網頁數量可能超出預期，其中一大部分卻無參考價值。

二、 切合使用者需求的網頁於查詢結果中所佔比例減小，即查詢正確率降低。

事實上，我們可以發現，上述的兩個問題皆是由於以關鍵字作為查詢條件所帶來的後遺症。其實就使用者來說，能以口語字句作為查詢條件是最方便的，然而現今的資訊技術仍無法完全解析人類語言及其含義。不得已只好採用關鍵詞，但關鍵詞往往不能完整地代表使用者的查詢需求，以太過廣義的關鍵詞進行查詢，符合的網頁數量當然會大幅增加。造成使用上的困擾。

儘管如此，我們還是可以其他方式彌補關鍵詞查詢的不足。事實上，這方面的研究已成為資訊檢索發展的趨勢。其中最重要的方向，就是對於語意的研究。部分學者專家利用文件中各詞彙的排列及分布，表達文章主旨相關的主要概念。這一類的技術包括詞彙的配對出現（term co-occurrence）、關鍵詞的分類（term clustering）、文件結構的分析（document structure analysis）等；此外，以知識庫的方式，模擬人類意識運作，而以概念（concept）為基礎，所串聯的知識結構與知識庫系統，也是其中一個重要的研究課題。

1.3 章節編排

本論文的編排方式如下：第一章為序論及簡介；第二章介紹中文詞庫以及模糊化詞義概念網路；第三章敘述文件處理的方式以及文件索引的建置；第四章說明查詢處理的流程以及文件比對的演算法則；第五章介紹系統時作的設計、實驗參數配置以及實驗結果；第六章為結論以及未來發展。

第 2 章　中文詞庫與模糊化詞義概念網路

中文文件與英文文件結構最大的不同，在於後者雖為一連串詞彙的集合，但詞彙之間皆以空格或標點符號區隔；前者則是由眾多具意義的詞彙連接而成，詞彙之間並無明確的分界標示，因而同一文句可能出現數種的排列組合方式，最著名的例子莫過於明朝徐文長的「下雨天留客天留我不留」，其中「下雨天留客，天留我不留」與「下雨天，留客天，留我不？留！」兩種斷詞法其意義完全不同，而事實上，除了上述兩種讀法以外，還另有五種斷詞方式亦有其意義【1】；中文處理比起其他歐系語言的困難處，正是在於要如何找出正確的詞彙組合，也就是中文斷（分）詞，不正確的斷詞，往往會將文義誤導至完全不相干或相悖的方向。

2.1 統計斷詞與詞庫斷詞

一般而言，處理中文斷詞的方法，大致上可分為統計法（statistical method）、詞庫法（dictionary-based method）、語法分析法（syntax-base method）、及概念分析（conceptual method）四種【2】，介紹如下：

1. 統計斷詞：統計式斷詞需要蒐集一定數量的文件構成一個大的語料庫（corpus），再由字彙的組成趨向，統計、學習與發現詞彙的界限。由於中文詞是由一個以上的字所組成，具意義的詞其組成字彙配對出現的頻率通常會比不具意義的字組配對來的高，故我們由語料庫中統計配對出現的字彙，出現頻率較高的字組即視為具意義的詞彙，並以這些詞彙作為之後文章斷詞的依據，舉例來說，假定有兩字 a 及 b，在文件中發生的次數分別是 f(a) 與 f(b)，同時發生且 b 緊接於 a 之後的次數是 f(a,b)。則我們可以此來判定「ab」為一有效詞的可能性：

$$C = \frac{f(a,b)}{f(a)\cdot f(b)}$$

C 可以看做是兩個字同時並緊鄰出現的趨向【3】。C 越大，則代表「ab」組成一詞的可能性越高。因此只要統計語料庫中所有字彙相互的排列組合次數，並計算 C 值，即可依此推論何種組合為有效詞。

2. 詞庫斷詞：採用詞庫法必須預先蒐集常用的詞彙，以建立一個有效詞詞庫，並以此詞庫所收錄的字詞與文件中的字組逐步比對，以分辨文件內何種字組排列於中文使用上具有意義，並排除不具意義字組出現的可能。然而，詞庫內容是否完備，會直接影響斷詞結果的正確性。
3. 語法分析：語法分析需分析語料庫中各詞彙的辭類及排列方式，歸納出常見的排列規則後作為文法，再依此文法規則對文件內容進行斷詞。
4. 概念分析：概念分析是以語義角度對文件進行斷詞，故對於各辭彙的意義必須詳加定義。

上述的斷詞法中，語法分析以及概念分析仍不足以單獨作為斷詞的依據，故通常只作為輔助性法則，一般還是以統計式及詞庫斷詞為主。

統計斷詞的優點是不需事先準備詞庫，且遇到詞庫中未蒐集的詞彙時，也能將其斷出。缺點則是容易將一些出現頻率大但無意義的字組當成有效詞，或是忽略出現頻率小但具重要意義的詞彙，因此正確率較為遜色；而詞庫斷詞與統計式斷詞相反，幾乎沒有詞彙判別正確率的問題，但卻必須花費時間蒐集及建立詞庫，當然，未收錄於詞庫中的詞彙會被當作無效字處理。本系統在考量斷詞正確率對於關鍵詞檢索過程及檢索結果的影響，故採用詞庫斷詞法，再配合斷詞法則，以語意及語法結構的角度，彌補詞庫斷詞法的缺點，斷詞法則詳見第三章。

我們以預先蒐集的所有的中文有效詞，建構一個有效詞詞庫（目前我們所蒐集的詞彙已超過十五萬個）。使用詞庫斷詞的系統相當的多，大多數的搜尋引擎，如 google、openfind 等皆採用之。主要原因是搜尋引擎的使用者皆採用關鍵詞查詢，採用詞庫斷詞算是最直接的處理方法，當然，各大搜尋引擎都會研發其他的演算法來提昇搜尋的效率及正確性，如 google 的 pagerank、openfind 的 polyrank【4，5】等等，但其檢索機制還是以關鍵字在文件中出現的頻率及其位置為基礎。

2.2 有效詞分類

有效詞詞庫中詳細記錄各有效詞的詞類。因為文法結構及用字的不同，中文與英文的詞類分法也不相同，英文等歐系語言有所謂的字根（stem），依字根可產生與其意義相關的各種詞類變化，例如 general 可產生 generalize、generally、generalist 等。而中文有許多詞彙是同型異類的，如「散步」若以英文的詞類分法可以當名詞也可以當動詞，為避免上述的情形，中文詞彙必須以不同的詞類分法，才能避免詞類的混淆。

在中研院詞庫小組的「中文詞類分析」【6】中，將中文詞分為以下八種：

1. 體詞（N）：包括名詞（例如，物品、時間、地點）、定詞（例如，這個、其他、一些等）、量詞（例如，一「本」書、三「根」棍子、一絲一毫）、代名詞、方位詞等。
2. 述詞（V）：大致可分為動作詞（例如，走、使用、散步）、狀態詞（例如，高大、開心、擅長等）及其他蘊含明確訊息卻無主體的詞彙。
3. 副詞（D）：具數量、評價、程度、方式、態度、否定、疑問意義之詞彙，例如，一共（數量副詞）、居然（評價副詞）、有點（程度副詞）。
4. 非謂形容詞（A）：指純形容詞（即不做他用的形容詞），用於修飾名詞，陽性、野生、真正等。
5. 介詞（P）：介詞多置於述詞前後作為引述或修飾用，例如，他「被」毆打、「以」人為鏡等。
6. 連接詞（C）：用以連接兩個（以上）的詞使成一單位，例如，我「和」你、咖啡「或」茶。
7. 語助詞（T）：常置於句子或詞組之後，表示語氣。例如，你好「嗎」、如此「罷了」、什麼「來著」。
8. 感嘆詞（I）：表示情緒或態度，與句子無關，是完全獨立的詞類，例如，「哎呀」、「喔」、「哀哉」等。

上述是中文詞類的大致分法，事實上在各詞類中還有許多細分的法則，只是我們可以發現，這八種詞類中，以體詞及述詞所代表的訊息最具意義，其次是非謂形容詞，至於其他的詞類在文件中大致上都只負責修飾、連接、表達語氣或態度的功能。值得注意的是副詞中的否定副詞（分類代碼為 Dc），所代表的否定語氣，對於文件內容有決定性

的影響，故不可輕易忽略之，而是要將其視為具訊息的詞類。我們在文字處理過程中，除了斷詞以外，也將標定各詞彙的分類，以作為語法語意分析的參考。

2.3 同義詞集合詞庫

假設只使用一般的有效詞詞庫進行斷詞作業，則可能碰到下列情形：

1. 文章撰寫時為考慮到前後文通順及字詞典雅，往往會使用一些同義詞（synonyms），以避免同一詞彙出現頻率過高。所謂同義詞就是指具相同意義的不同詞彙。由於文字資訊檢索是以文件中有效詞出現的頻率作為判別文章語意的依據，若未將同義詞的影響列入考量，在判別文章內容大意時可能會出現誤差。
2. 除了同義詞以外，詞彙之間還存在如相似、反義等關係，這些詞與詞之間的關聯使得每一詞彙在文件中並非獨立（independent）；以往的資訊檢索方式是將各關鍵詞的出現次數全部計算，並沒有考量詞彙間的相互關係對於文件主旨的影響。

為解決上述問題，我們進行資訊檢索時，除了依據傳統做法考慮關鍵詞出現頻率外，也加入語意（semantics）的觀念。目前資訊技術對於語意這種抽象概念的處理能力十分有限，而詞義的處理與分析上也一直是資訊技術發展的目標。其中由美國普林斯頓大學所發展的「Wordnet」系統【7】，已初步具備了概念（concept）的結構。基本上 Wordnet 是一個詞庫，與一般詞庫不同的是，構成 Wordnet 的主體並非是字詞本身，而是字詞所蘊含的意義。在 Wordnet 中所有詞彙係以「詞義」分類，相同意義的詞彙會被分在同一個集合中，此集合稱為「Synset」，即同義詞集合（synonym set）。而具有多種意義的詞彙，則會同時出現在不同的同義詞集合中；除了同義字以外，Wordnet 系統也對詞彙之間的關係，依照詞性不同做統整，舉例來說，在 fast 與 slow 之間存在反義詞關係，而與 quick、rapid、swift 等詞之間存在相似關係。

有鑒於同義詞集合概念在詞義處理上的優點，本研究採用此方式作為文件處理的方法。首先整理出一些特定領域的詞彙，並依其概念，將所有同義詞彙收集成一同義詞集合，再設定詞彙之間的關聯。圖 2.1 中我們可以看到具相同意義的詞彙會先構成一個同義詞集合，例如，「客人」與「來賓」、「主人」與「東道主」等，之後再依語意，將具

關係的兩個同義詞集合連接起來，並註明關係的種類，包括，同義、相似、屬於其中之一等等。

一篇文章中，並不是所有的詞彙都需要詞義處理。如前述，我們平日所使用的語言中，有許多不嚴重影響詞義的詞彙，包括介詞、冠詞、語助詞等。為維護同義詞庫的查詢績效，同時節省不必要的文字處理時間，沒有必要將其列入同義詞庫的範圍。基於此項考量，Wordnet 系統的做法是去除不具意義的詞類，僅保留較具意義者，包括名詞、動詞、形容詞與副詞，再將所蒐集的詞彙依其在不同領域所代表的意義分類，並註明與其他詞彙間的關係，例如，名詞之間具有分類關係、形容詞有語氣強弱之分、動詞有行動種類等等；而在本系統中，因為詞彙分類的不同，我們便只針對體詞、述詞、及否定副詞三種較具意義的詞類建立同義詞集合。

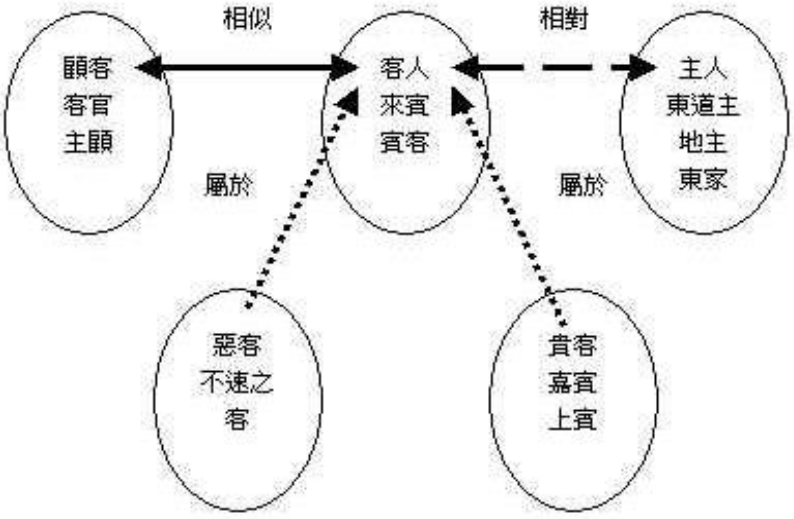


圖 2.1 Synset 示意圖

在建構同義詞詞庫時，為了考慮到每一個詞彙蘊含意義的完整性，我們不僅要列出各詞彙字面上的意義，也要整理該意義所衍生的慣用語、隱喻、別稱、特別術語等非正式用法，例如：盜版軟體光碟常被稱為大補帖、泡麵、麵等等。這些非正式用法往往會隨著社會的發展而有所更動，故我們傾向由人力來建構，並視實際情形隨時做必要的更新。

此外， Wordnet 僅提供詞彙之間是否具有關聯，對於其程度大小則無著墨；為能更

明確的表示出詞彙之間的關係及其相關程度，我們以模糊概念網路（fuzzy concept network）【8，9】的技術，將詞彙依詞義分群，以建構一個模糊化詞義概念網路（fuzzy term concept network，FTCN）。其做法首先以詞義概念為單位來分群各詞彙，並將各種概念相互之間的關係分為數種不同種類，如相似、相反等，而該關係的程度則由人為訂立一個參數值，來說明其程度大小，例如，「金融」與「財政」兩詞彙之間為相似關係，程度可能達 0.8，「蘋果」與「水果」之間存在「屬於其中之一」的關係，程度達 0.9。

2.4 模糊化詞義概念網路

首先定義模糊化詞義概念網路中的要件：

<定義>模糊化概念網路：一個以 $G(N,E)$ 表示之圖形，其中 $N=\{N_1,N_2,\Lambda,N_l\}$，$N_i$（$i=1,2,\Lambda,l$）為一概念；$E=\{E_{ij}|E_{ij}$ 為從 N_i 指向 N_j 的有向邊，$1\le i,j\le l,i\ne j\}$，而以 $L_{i\to j}$ 表示 N_i 與 N_j 之間的關係種類，以 $m_{ij}\in[0,1]$ 表示 $L_{i\to j}$ 的程度大小。m_{ij} 值越大則代表 N_i 與 N_j 之間的相關度越強。任兩個同義詞集合之間皆可能存在關係鏈結，而形成一網狀結構（見圖 2.2），稱為模糊化概念網路。

<定義>孤立：一個模糊化概念網路 $G(N,E)$，$N=\{N_1,N_2,\Lambda,N_l\}$ 為節點集合、$E=\{E_{ij}|E_{ij}$ 為從 N_i 指向 N_j 的有向邊，$1\le i,j\le l,i\ne j\}$；若 $N_k\in N$，而 $E_{kj}\notin E$，$1\le j\le l$，$k\ne j$，則 N_k 於 G 中為孤立（isolated）節點。

<定義>入關係鏈結：一個模糊化概念網路 $G(N,E)$，$N=\{N_1,N_2,\Lambda,N_l\}$ 為節點集合、$E=\{E_{ij}|E_{ij}$ 為從 N_i 指向 N_j 的有向邊，$1\le i,j\le l,i\ne j\}$；若存在 $E_{kj}\in E$，則 E_{kj} 為 N_j 的入關係鏈結（in-let relationship link）。

<定義>出關係鏈結：一個模糊化概念網路 $G(N,E)$，$N=\{N_1,N_2,\Lambda,N_l\}$ 為節點集合、$E=\{E_{ij}|E_{ij}$ 為從 N_i 指向 N_j 的有向邊，$1\le i,j\le l,i\ne j\}$；若存在 $E_{kj}\in E$，則 E_{kj} 為 N_k 的出關係鏈結（out-let relationship link）。

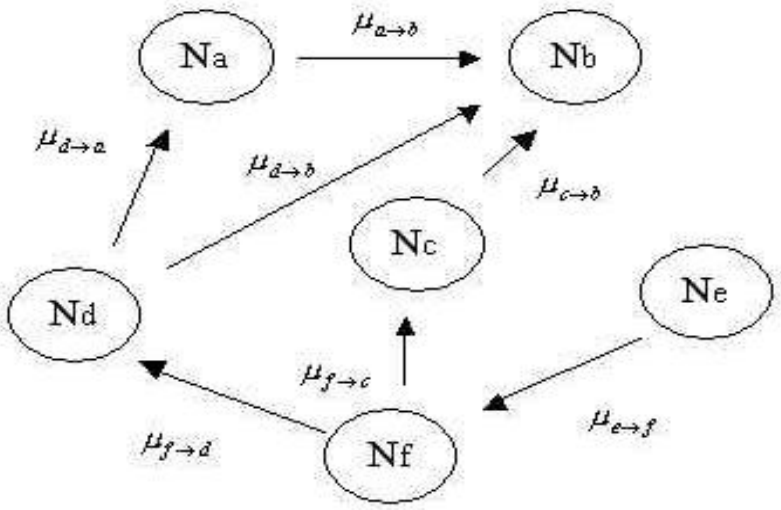


圖 2.2 概念網路結構圖

在兩節點 N_i 與 N_j 之間存在關係 $L_{i\to j}$ 的例子，在生活中處處可見；舉例來說，假設 N_i 代表「芭樂」，N_j 代表「番石榴」，因為兩者就一般人的認知是相近甚至是相同的東西，故 $L_{i\to j}$ 可解釋為兩者之間的相似性，明顯是一雙向關係。但假使 N_i 代表「水果」，N_j 代表「蘋果」；對「蘋果」來說，「水果」具有較廣義的解釋，此處 $L_{i\to j}$ 可解釋為「蘋果是水果的一種」；若將這句話反過來變成「水果是蘋果的一種」，則顯然不合邏輯；其他模糊化詞義概念網路的研究中，有些對於兩詞義概念之間是以相似程度定義相互關係【10，11】，但兩詞義概念之相似性顯然為雙向關係才有意義，故在處理詞彙之間的關係時，必須以某概念為基準，再定義各出關係鏈結，如此方能獲得雙向關係。

2.4.1 關係（Relationship）

在 Chen 及 Horng【12】的研究中，他們以四種參數定義詞義概念之間的關係（relationship），並以此判定文件相似度。此四種關係分別為：

- 正相關（positive association）：泛指兩概念的相似程度。凡意義類似或是具高度相關者皆屬之。唯任一概念不可概括另一方。
- 負相關（negative association）：代表該詞義概念的反義（antonym）或補義（complementary）。任意兩無相關的概念並不存在此關係。

- **廣義（generalization）：當一概念 A 在語意上可完全地包括另一概念 B 時，則對 B 而言，A 具廣義關係。**
- **狹義（specialization）：為廣義的逆關係。**

換言之，是將詞義概念之間的關係分為相似、相反、以及範圍大小，並以此判別文件之間的相似性。但是在詞義概念之間並不只存在上述關係而已。舉例來說，「汽車」與「輪胎」在分類上並不屬於同一類，兩者之間也沒有相似性，但「輪胎是汽車的一部份」的關係不容忽視；再者，提到「鳥類」，就會想到「卵生」，但兩者純粹是特質的聯想，在詞類上根本無關。因此為了避免缺憾，我們加入三項關係：整體（Holonyms）、細部（Meronyms）、及屬性（Attribute），相關定義如下：

<定義>本質（essence）：指事物本身所固有的，決定性質、面貌和發展的根本。

<定義>整體關係：兩概念C_a與C_b之間若存在關係$L_{a\to b}$，且C_a為C_b的組成要素（意即若是C_b缺少C_a的部份，C_b的本質會有所改變或不完整，例如，電腦缺少中央處理器、汽車缺少引擎），但兩者在分類上並不屬於同一類，則$L_{a\to b}$視為整體關係，而以$H_{a\to b}$表示。

<定義>細部關係：兩概念C_a與C_b之間若存在關係$L_{a\to b}$，且C_b為C_a的組成要素，但兩者在分類上並不屬於同一類，則$L_{a\to b}$視為細部關係，而以$M_{a\to b}$表示。

<定義>屬性關係：兩概念C_a與C_b之間若存在關係$L_{a\to b}$，且C_a與C_b並無上述其他關係存在，但C_a為C_b的特性之一（此處的特性，指的是事物的狀態、功能等，用來描述關於該事物的事實），則$L_{a\to b}$為屬性關係，而以$A_{a\to b}$表示。

本質在此指的就是該事物所擁有的特性及功能，此屬性可由其上層（廣義關係）繼承而來。換句話說，下層（狹義關係）本質的一部份來自上層所有概念的屬性集合；當然，下層關係也會依照實際情形增加該上層所缺少的本質。例如，「企鵝」屬於「鳥類」，「企鵝」與「鳥類」之間互為廣義/狹義的關係，「企鵝」所擁有的屬性有「有羽毛」與「會游泳」等，其中「有羽毛」繼承自該上層「鳥類」，而「會游泳」則屬於企鵝本身的特性；由此可知在概念網路中位於越狹義的概念，通常其本質及屬性也會相對的增多。若C_a與C_b為正相關，則$L_{a\to b}$以$P_{a\to b}$表示，負相關以$N_{a\to b}$表示，廣義以$G_{a\to b}$表示，而狹義則以$S_{a\to b}$表示。

另外兩個例子如下：「汽車」屬於交通工具，故汽車的特性是「能移動」，功能則是「代步」「載物」等，倘若將汽車的輪胎拿掉只剩車殼，則汽車就失去原本能跑的功能

與特性；「香蕉」屬於水果，特性是「可以吃」，一般也是將它當作食品。香蕉是有皮的，但我們把皮剝掉，香蕉仍然是香蕉，「可以吃」的特性並不會改變，事實上香蕉是要剝皮才能吃，換言之，少了「皮」的屬性，香蕉的本質並未改變。因此我們在決定 C_a 與 C_b 是否存在整體關係時，必須看 C_b 是否會因為缺少了由其上層 C_a 所繼承而來的屬性，而在本質上有所改變。相對於整體關係，細部關係就沒有所謂屬性繼承的限制，但也因為如此，在建構細部關係時主觀意識的影響將要比整體關係大的多，為避免因主觀意識而產生的歧異，我們不主動訂定細部關係，而是藉由整體關係的逆向關係代替。

<定義>逆向關係：兩概念 C_a 與 C_b 之間存在關係 $L_{a\to b}$，且 $L_{a\to b}$ 亦存在，則 $L_{b\to a}$ 即為 $L_{a\to b}$ 之逆向關係，$m_{ba} = m_{ab}$。

舉例來說，「香蕉屬於水果」是正確的，「有種水果叫香蕉」亦正確，水果之於香蕉 $L_{香蕉\to水果}$ 為廣義關係，則 $L_{水果\to香蕉}$ 為狹義關係，顯然廣義與狹義關係皆成對出現且方向相反，$L_{水果\to香蕉}$ 即為 $L_{香蕉\to水果}$ 的逆向關係；而整體與細部關係間亦互為逆向關係。在訂立整體/細部以及廣義/狹義關係時，我們盡量避免邏輯上的爭議（例如，「白馬不是馬」這句話，在這裡不適用，白馬將視為馬的一種，就如同香蕉是水果的一種一樣），僅就一般通用的思考模式去建構詞義概念的相互關係。

同義詞集合之間的所有關係中，除屬性關係外，其餘的在詞義概念網路中，皆有逆向關係的存在，舉例來說，若兩同義詞集合 a 及 b 之間存在廣義關係 $G_{a\to b}$，則狹義關係 $S_{b\to a}$ 必同時存在，此時對 a 而言，$G_{a\to b}$ 為出關係鏈結，$S_{b\to a}$ 則入關係鏈結，對 b 而言則正好相反；屬性關係是單獨存在，有其不可逆性，定義如下：

<定義>屬性關係的不可逆性：假設兩同義詞集合 a 及 b 之間存在屬性關係 $A_{a\to b}$，則不存在 $A_{b\to a}$。

至於屬性關係的判定，應以較易界定以及客觀的為主，不易界定、特例、或是太主觀認定的特質則不在討論範圍之內，舉例來說，對一般人而言，牛與豬是一部份食用肉的主要來源，雖然有若干國家及地區並不吃牛肉或豬肉，但我們仍將兩者視為食品，故「可食用」仍為牛肉與豬肉的屬性。一般而言，兩正相關概念視為有若干共通的屬性；與整體/細部關係相同，狹義概念將繼承（inherit）其廣義概念的屬性關係。

詞義概念網路中以一關係鏈結串連兩相異且具有關聯的同義詞集合，但並非所有的同義詞集合之間皆存在關係鏈結。因此對於兩不相鄰同義詞集合之間的關係判定，則需

要關係合成（relationship combination）如圖 2.3（a），AC 兩點之間在詞義概念網路中並無直接的關係連結，但可經由 B 點來聯繫，AB 之間存在 X 關係，BC 之間存在 Y 關係，則 AC 之間的關係，可以 X 與 Y 關係合成後得到的 Z 關係表示之（如圖 2.3（b））。因此，關係合成結果取決於兩同義詞集合之間的關係種類，就圖 2.2 的例子來說，假設 $m_{e\to f}$ 為正相關關係，$m_{f\to d}$ 間存在負相關關係，因為 N_e 與 N_f 可能在某些屬性上相同，N_f 與 N_d 則在某些屬性上相反。而對 N_e 來說，與 N_f 相同的屬性，可能也會因而和 N_d 形成負相關關係；又假設 $m_{e\to f}$ 為整體關係，$m_{f\to d}$ 間存在細部關係，N_e 雖為 N_f 的一部份，但並不一定是唯一的部分，換句話說，對 N_f 存在細部關係的 N_d，與 N_e 不一定相同或相似，有可能完全沒有關聯，例如，輪胎與方向盤雖然皆為汽車零組件，但兩者間並無關係。因為如此，在 N_e 與 N_d 之間我們便以無關係來表示。任兩節點之間的關係若以上述邏輯推算。依此邏輯所定義出的合成關係如表 2.1 所示。

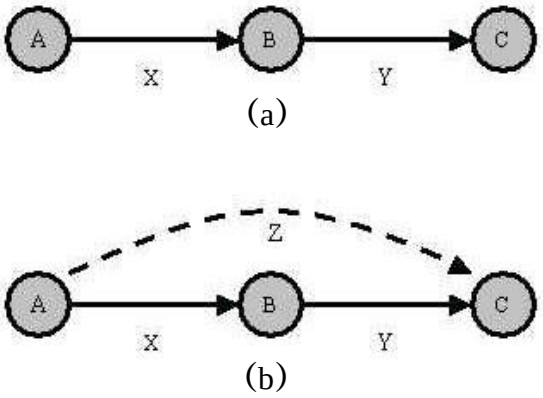


圖 2.3 關係合成示意圖

表 1 概念關係合成對應表

	P	N	G	S	H	M	A	Z
P	P	N	G	S	H	M	A	Z
N	N	P	N	N	Z	Z	Z	Z
G	G	N	G	P	H	M	Z	Z
S	S	N	P	S	H	M	A	Z
H	H	Z	H	H	H	Z	Z	Z
M	M	Z	M	M	Z	M	Z	Z
A	A	Z	Z	A	Z	Z	A	Z
Z	Z	Z	Z	Z	Z	Z	Z	Z

P : Positive Association

N : Negative Association

G : Generalization

S : Specialization

H : Holonym

M : Meronym

A : Attribute

Z : None

將所有關係依合成優先的順位（priority**）排列的結果為：**

1. **無關係（以「-」表示）。**
2. **負相關。**
3. **整體及細部關係。**
4. **廣義及狹義關係。**
5. **正相關關係。**

我們可以發現，定義越清楚的關係，順位也就越高。正相關由於其模糊的定義（「相似」「相關」僅僅是一個抽象形容詞，標準極難界定），故其順位為最後。至於「屬性」關係，因為是以附加性質存在，故除了屬性的繼承外，與其他關係之間的順位大小並不能加以比較。

在概念網路中，原則上兩相鄰同義詞集合之間的關係都會由人工加以定義，非相鄰的同義詞集合關係則以概念關係合成定義之，但當檢視兩非相鄰同義詞集合間的關係時，則可能遭遇下列兩種狀況：

1. **缺少兩非相鄰同義詞集合之間的關係資料。**
2. **兩非相鄰同義詞集合間的連結路徑並非唯一。**

如前述，各相關的同義詞集合之間是由有向關係所連接，由於一個概念網路通常不是完全圖（complete graph**），故任兩個節點，可能不存在有向關係。**

<定義>可抵達節點：一個模糊化概念網路 $G(N,E)$，$N=\{N_1,N_2,\Lambda\ ,N_l\}$ **為節點集合、** $E=\{E_{ij}|E_{ij}$ **為從** N_i **指向** N_j **的有向邊，** $1\le i,j\le l,i\ne j\}$，**若存在一關係組合** $\{E_{ab}\}\subseteq E$ **可讓** N_a **經由** $\{E_{ab}\}$ **連結到** N_b，**其中** $N_a,N_b\in N$，**即** N_b **為** N_a **可抵達節點**（reachable node），**若** E_{ab} **不存在，** N_b **則為** N_a **之不可抵達節點**（unreachable node）。

因此，若兩同義詞集合間互為不可抵達節點，則沒有必要再對其進行關係合成，另一方面，若兩者之間存在兩條以上的路徑，則這些路徑就必須相互競爭，以決定一最佳路徑，此部份將於後續章節說明之。

關係結合的流程如下：

$G(N,E)$**中兩概念** C_a **與** C_b **之間若不存在** $L_{a\to b}$ **之直接關係，但至少存在一路徑** $P=\{L_{a\to c1},L_{c1\to c2},\Lambda\ ,L_{ci\to b}\}$，**即** C_b **為** C_a **之可抵達節點，則：**

1. **令** $cx=c1$，$cy=c2$。
2. **依表** 2.1 **所示，進行** $L_{a\to cx}$ **及** $L_{c1\to cy}$ **之間的關係合成以決定** $L_{a\to cy}$。
3. **如果** $cy\ne b$ **且** $L_{cy\to cz}\in P$，**令** $cx=cy$，$cy=cz$ **重複步驟** 2，**否則停止**。

所有同義詞集合之間的關係以上述方式定義後，可以下列方式表示：

令無屬性「-」為 Z，**而同義詞庫中存在同義詞集合** $S=\{s_1,s_2,s_3,\Lambda\ ,s_n\}$，**則關係矩陣** R：

$$R=\begin{array}{c} \\ s_1\\ s_2\\ s_3\\ \vdots\\ s_n\end{array}\begin{array}{c}\begin{array}{cccccc} s_1 & s_2 & s_3 & \cdots & s_n\end{array}\\ \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n}\\ r_{21} & r_{22} & r_{23} & \cdots & r_{2n}\\ r_{31} & r_{32} & r_{33} & \cdots & r_{3n}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ r_{n1} & r_{n2} & r_{n3} & \cdots & r_{nn}\end{bmatrix}\end{array}$$

其中 $r_{ij}\in\{P,N,G,S,H,M,A,Z\}$，$1\le i,j\le n$，$r_{ij}=P$ when $i=j$。

2.4.2 相關程度（Relevant）

若兩同義詞集合之間存在兩條以上的可抵達路徑，則必須決定一最佳路徑來描述兩同義詞集合之間的關係，因此有必要在同義詞集合間的關聯加上某種加權值。另一方

面，假設節點ｂ為節點ａ的可抵達節點，則表示節點ｂ與節點ａ之間存在某種關係。但只以「有」或「沒有」來定義兩節點之間的關係，可能太過籠統。舉例來說，麻雀、金絲雀、及企鵝皆屬鳥類，但就相似程度而言，麻雀與金絲雀顯然高於與麻雀與企鵝，若使單純的以相似或不相似來定義任兩概念之間的關係，可能會造成語意上的誤判。因此在兩同義詞集合之間，除了定義其關係種類以外，我們也將定義該關係的程度大小；以圖 2.2 為例，由於其中的 m 值只代表兩節點之間的關係，並未包括相關程度的資訊，因此加入相關程度的模糊化詞義概念網路應該加以修改。如圖 2.4 所示，節點之間的參數由 $m_{i\to j}$ 變為 $m_{i\to j}(r,re)$，其中 r 代表關係種類，re 表示該關係的相關程度。

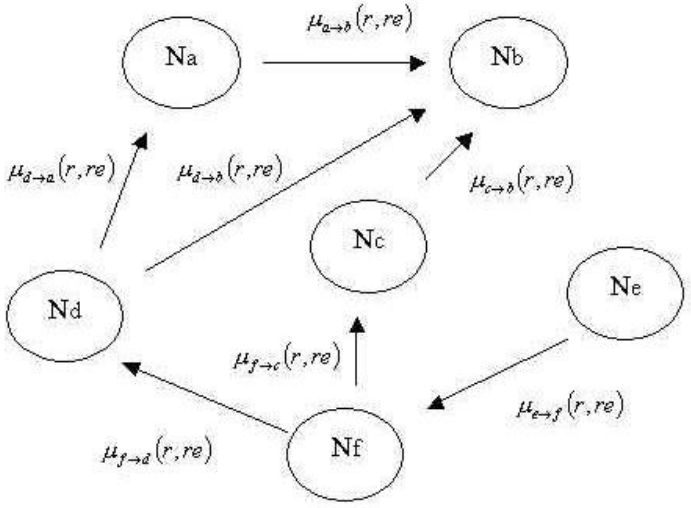


圖 2.4　加入相關程度後的模糊化詞義概念網路

在建構詞義概念網路中同義詞集合之間的關係及其程度時，我們並不逐一定義所有同義詞集合的組合，而是依詞義分類，串聯較為直觀且不易混淆的概念，在概念網路中無直接相連的兩同義詞集合 C_1C_2 之間的相關程度值，並不會在人工建構的階段被定義，因此與 C_1C_2 間關係相同，相關程度也必須經過相關程度合成來獲得；如圖 2.5（a）中，$L_{A\to C}$ 只能間接由 $L_{A\to B}$ 與 $L_{B\to C}$ 獲得，$m_{AC}=m_{AB}\times m_{BC}$。此外，由於兩同義詞集合間可能存在有一條以上的路徑，如圖 2.4（b）所示，$L_{M\to P}$ 可由 $L_{M\to N}$ 及 $L_{N\to P}$，或是 $L_{M\to O}$ 及 $L_{O\to P}$ 推算出來；就語意的角度來說，存在一條以上的路徑，表示該概念 C_1 可由一種以上的聯想過程聯想到另一概念 C_2，至於該以哪一條路徑來決定 C_1C_2 之間的關係，方法如下：

任兩同義詞集合之間的相關程度判定過程：

1. **由** S_i **出發到達其可抵達節點** S_j **之間至少存在一條路徑** P_{ij}**，而** P_{ij} **中包含同義詞集合節點依序為** S_1, S_2, Λ, S_n**，即** $P_{ij} = (E_{i,1}, E_{1,2}, \Lambda, E_{n-1,n}, E_{n,j})$**；** RE_{ij} **為** S_i **與** S_j **在以** P **為路徑的相關程度，** $RE_{ij} = [0,1]$**，則：**

$$RE_{i,j} = Min\ (RE_{i,1}, RE_{1,2}, RE_{2,3}, \Lambda, RE_{n-1,n}, RE_{n,j})$$

2. **若** S_i **與** S_j **之間共存在** m **（**$m \geq 1$**）條路徑** $P_{S_i \to S_j} = \{P_1, P_2, \Lambda, P_m\}$**，** $P_{S_i \to S_j}$ **的相關程度為** $RE_{P_{S_i \to S_j}} = \{RE_{P_1}, RE_{P_2}, \Lambda, RE_{P_m}\}$**，則：**

$$RE_{i,j} = MAX\ (RE_{P_1}, RE_{P_2}, \Lambda, RE_{P_m})$$

即相關程度最大的路徑為 S_i **與** S_j **之間的最佳路徑**（best path）。

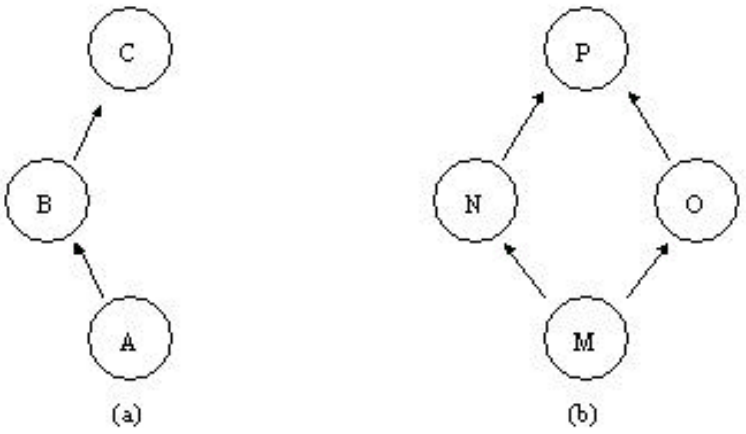


圖 2.5 兩非相鄰同義詞集合關係示意圖

同義詞庫中存在同義詞集合 $S = \{s_1, s_2, s_3, \Lambda, s_n\}$**之相關程度矩陣** RE **為：**

$$RE = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{array} \begin{array}{c} \begin{array}{ccccc} S_1 & S_2 & S_3 & \cdots & S_n \end{array} \\ \begin{bmatrix} RE_{11} & RE_{12} & RE_{12} & \cdots & RE_{1n} \\ RE_{21} & RE_{22} & RE_{23} & \cdots & RE_{2n} \\ RE_{31} & RE_{32} & RE_{33} & \cdots & RE_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ RE_{n1} & RE_{n2} & RE_{n3} & \cdots & RE_{nm} \end{bmatrix} \end{array}$$

其中 $RE_{ij} = [0,1]$**，** $1 \leq i, j \leq n$**，** $RE_{ij} = 1$ when $i = j$。

2.4.3 階層距離（Layer distance）

所有對任一同義詞集合 n_i 而言，可抵達的節點，都需經一個或以上的有向關係，經過的有向關係數量越多，表示 n_i 與目標節點 n_t 間的距離越遠，也代表兩者之間語義上的差距越大；基本上，n_i 與 n_t 之間的連結關係是取決於兩者之間的聯想方向。由於概念網路是一網狀結構，若兩節點之間必須經過數個有向關係連結，則可能會因路徑中的聯想方式不同，而在語意上有著不同程度的差別，我們以「轎車」、「計程車」、及「輪胎」為例，「轎車」與「計程車」皆屬於交通工具，計程車即轎車改裝而成，在概念上差距不大，雖然「輪胎」為兩者皆具備的零件，一般人也能由汽車聯想到輪胎，但與前兩者根本是不同層級的概念，在語義上的差距顯然超過「轎車」及「計程車」。由此可知，隨著兩同義詞集合之間連結關係的不同，有必要調整語義關係之間的階層差距。

階層的觀念主要是存在於廣義/狹義關係上，在日常生活中，我們習慣將事物分門別類，例如，機車、汽車、飛機、船屬於交通工具；香蕉、蘋果、葡萄、柳丁屬於水果等。以上述例子來看，「飛機」屬於「交通工具」中的一種，但以「交通工具」的角度來看，「飛機」與「汽車」都是其中的一類。若將此一想法延伸到所有事物，可以發現就分類學的角度來看，我們可將每個概念依其層級排列並形成若干樹狀結構。這裡的層級指的是概念所包含的範圍，範圍越大，層級也就越高。圖 2.5 中，BC 是 A 下一層的概念，而 BC 兩概念分別包含了其他的概念，對 A 來說，BC 兩點在分類上是同一層的，而 DEFG 因位於 BC 的下一層，所以也在同一階層，以此類推，即可大致定義所有 A 所包含概念的階層。Wordnet 系統將所有概念依其意義，分成十三大項，每一大項都代表一個概念的樹狀結構。然而，若將所有的概念依其涵蓋範圍分層級，由於各分類定義不一，各分類中相同的層級所包含事物的差距可能很大，且要如何以客觀標準訂定各層級的範圍，也是另一個難題。因此本研究將不對所有概念定義其絕對階層，而是以相對階層距離的方式，來判定兩概念之關的關係。以圖 2.6 為例，對於 A 來說，H 位於下三層之處，故 A 與 H 之間的階層距離為 3，以相對階層距離不但可避免階層定義標準不一所造成的謬誤，更能以分類的角度，來分析概念之間的關係。

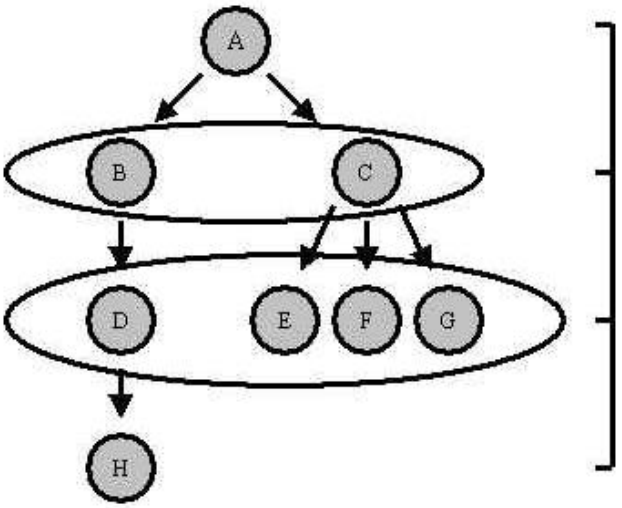


圖 2.6 概念層級示意圖

上述的階層距離乃針對整體/細部關係定義，而其他關係之間的階層距離亦須以此為基準定義之。我們發現，若將所有關係依階層距離由小到大排列的結果為：

1. 正相關。
2. 負相關。
3. 廣義及狹義。
4. 整體及細部。
5. 無關係（以「-」表示）。

正、負相關強調的是概念的相似或相反，故在語義上的差距不大；廣義與狹義關係在分類上有著明確的階層關係，因此作為分層的基準；而整體與細部關係在語義上並無相似之處，同時在分類上大都也不屬同一類，絕大部分整體與細部關係的建立皆以聯想的方式來串聯之，例如，汽車與輪胎。因此，除了無關係外，兩同義詞集合間若存在整體與細部關係，表示兩者在分類層級上有一定差距。我們以廣義與狹義關係在分類上的階層為基準，在詞義概念網路上每經一層廣義或狹義關係，則階層值 l 增加 1，正、負相關關係由於較常出現在同層次的概念之間，故 l 值小於 1，而整體或細部關係往往要經過一層以上的聯想關係，故 l 值需大於 1，以凸顯出概念上的差距。我們在定義兩同義詞集合 S_a 與 S_b 之間的階層距離 ld_{ab} 時，首先找出 S_a 與 S_b 之間的路徑 P，再找出 P 上所有節點之間的階層距離，P 上的 l 值總和即為 ld_{ab} 值。

<定義>階層距離：假設同義詞集合 S_b 為同義詞集合 S_a 為可抵達節點，S_a 到 S_b 存在一路徑 $P_{ab}=\{E_{a1},E_{12},\Lambda,E_{(n-2)b}\}$，計由 $n-1$ 個關係連結組成，$LAY=\{lay_1,lay_2,\Lambda,lay_{n-1}\}$ 為對應 P_{ab} 的階層值，則 S_a 與 S_b 之間的階層距離 ld_{ab}

$$ld_{ab}=\sum_{1}^{n-1} lay_i$$

若 S_b 為 S_a 之不可抵達之節點，則 $LD_{a\to b}=\infty$。

在定義階層距離 $ld_{a,b}$ 時，與兩同義詞集合之間決定相關程度時類似，都可能面臨到兩同義詞集合之間階層距離不唯一的問題，此時我們需找出兩同義詞集合 S_a 與 S_b 間的最佳路徑 P_{ab}，再以之推導出 S_a 與 S_b 之間的階層距離。

所有同義詞集合 $S=\{s_1,s_2,\Lambda,s_n\}$ 間的階層距離以階層距離矩陣 LD 表示：

$$LD=\begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{array}\begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & s_3 & \cdots & s_n \end{array} \\ \begin{bmatrix} ld_{11} & ld_{12} & ld_{13} & \cdots & ld_{1n} \\ ld_{21} & ld_{22} & ld_{23} & \cdots & ld_{2n} \\ ld_{31} & ld_{32} & ld_{33} & \cdots & ld_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ ld_{n1} & ld_{n2} & ld_{n3} & \cdots & ld_{nn} \end{bmatrix} \end{array}$$

其中 $d_{ij}\in N$，$1\le i,j\le n$，$ld_{ij}=0$ when $i=j$。

2.4.4 建立詞義概念網路的原則

詞義概念網路結合中文詞庫，並詳細定義網路節點之間的關係、相關程度、以及距離之後，就可以應用於文字處理、查詢延伸、與相似性比對等資訊檢索的核心部分。然而，要想建立一個能涵蓋所有知識的系統，幾乎不可能實現，在知識結構的完整性（completeness）無法達成的情況下，盡量提高詞義概念的正確率（accuracy），就顯得非常重要。值得注意的是概念網路的內容，並沒有一個明確的規範來評量其正確性，而是端看其內容是否符合一般人的認知。舉例來說，香蕉的屬性是黃色，這是一般人可接受的事實，雖然有些香蕉經過特殊方式的培育而結出其他顏色的果實，但是並不足以改變在一般人印象中香蕉是黃色的事實。另外還有一個需要注意的是資料一致性

（consistence）的問題，一致性的問題常會發生在不同觀點的人判斷相同事物時，所得到結論的差異。舉例來說，一般人覺得很類似的兩件衣服，若由服裝設計師來看，則可能因為其設計款式、材質、織法、品牌、搭配方式等，而對其作出「完全不同」的評價，這樣的差異源自於時空背景以及專業知識的不同，若是沒有一個統一的標準，在事物概念上的判斷極可能產生謬誤。因此，在建立詞義概念網路時，必須以統計的方式，蒐集多人對同一事物的認知，再整理出一個對一般人較能接受的定義，如此可使因立場觀點不同產生的認知誤差減到最低。

第 3 章　文字處理及索引

本章討論文件及查詢語言的內容處理，與文件內含關鍵詞的索引建置。

3.1 文字處理

搜尋系統在提供服務前，必須預先蒐集文件，處理並分析其內容，並依照文件內各有效詞彙於文件內所在位置建立索引以供查詢。因此，搜尋結果的正確率，與文字處理的精確程度有很大的關係。大抵上中文的文字處理可分為下列步驟：

1. 斷句：斷句的目的是為了將一個文章段落切割成較小的單位—句子，以利於後續的處理。由於中文句與句之間皆以標點符號分隔，故先以標點符號斷句。
2. 斷詞：斷詞的目的是要將一個完整句子，依其意義找出句子中詞彙的適當排列方式。如前述，斷詞必須配合常用詞彙的統計數據或詞庫系統，對句子作適當的處理。
3. 斷詞驗證：斷詞驗證是為了避免不適當的斷詞排列方式影響文字處理精確度。常見的斷詞驗證方式係以常用文法分析已斷詞之文句，剔除不符語法的斷詞排列，提供正確的斷詞結果及詞類分析。

以下將逐一探討。

3.1.1 斷句及中文標點符號

中文標點符號計有為句號、逗號、頓號、分號、冒號、引號、問號、驚嘆號、破折號、夾註號、刪節號、書名號、專名號、音節號等共 14 種【13】，各符號表示見表 3.1。有些標點符號表示語氣的終止或轉折（例如：分號、句點等）；有些用以連接兩句子或當作詞彙的媒介（例如：逗號，頓號等）。斷句時注重的是該標點符號是否會讓文句的語義告一段落。在所有標點符號中，除了書名號、專名號、音節號、頓號及引號以外，其他都有段落的作用。其中因為書名號與專名號必須依附於文字，而音節號主要用於敘述人名或地名等專有名詞，對文句的段落影響不大。真正在斷句處理上需要注意的唯有

頓號及引號。頓號是表示語氣的停頓，常用於列舉事物，或依序列出性質相近詞類相同之詞彙，以示強調或表示多元化特性，但不希望造成語氣的轉折或形成段落時，例如，「中文標點符號計有為句號、逗號、頓號、分號、冒號、引號、問號、驚嘆號、破折號、夾註號、刪節號、書名號、專名號、音節號等」；另一例是「整齊、清潔、簡單、樸素、迅速、確實」。這些頓號，只是要分隔兩種事物以避免混淆，對於句子的完整性並無影響，故斷句時可忽略之，惟進行斷詞時，頓號必為斷詞點。

表 2　中文標點符號

句號	。	夾註號	（）——
逗號	，	驚嘆號	！
頓號	、	破折號	——
分號	；	刪節號	……
冒號	：	書名號	〈〉
引號	「」『』	專名號	＿＿
問號	？	音界號	‧

引號大致上有兩種用法，其一是強調該詞彙於文件中的語氣或用以表示專業術語，如：中華男籃換血面臨「陣痛期」；另一種用法是在轉述某人所說或寫出的話語及文字，例如，小明說：「這麼醜的書包我才不要。」我們可以發現，前者雖然將「陣痛期」一詞以引號括註，但其目的只是為表示該詞於此文中的特別用法或引用典故，若將引號去掉，並不影響文章的意義；而後者由於是轉述一完整句子，引號內的文字無法與前後文合併，此種用法在引號之前會有一冒號，表示以下字句為轉述，不可與上下文合併，因為冒號已有段落字句的功用，故仍可將引號忽略，唯斷詞時，該引號原所在地也是一個斷詞點。除頓號及引號外，遇到其他具段落作用的標點符號，如句號、逗號等，則視為該句子的結束點。

3.1.2 斷詞法則

所謂「詞」是指具有獨立意義，且扮演特定語法功能的字串，要如何由一連串的中文字中找出合於文法及語意的字串組合，實際上非常困難，原因包括：

1. 中文詞類包括體詞、述詞、副詞、連接詞、語助詞【14】等等，每種詞類皆有不同的結構。
2. 體詞與述詞皆有複合詞組，例如，「下午兩點多」即為「下午」和「量點多」兩述詞之複合詞組。
3. 詞尾衍生範圍的不確定性，例如，台北市、台北市政建設、台北市政府等，看到台北市後，無法確認詞尾的詞性。
4. 詞彙長度的判別困難，例如，國民大會代表可作為一個獨立詞，也可分成「國民大會」以及「代表」，或是「國民」「大會」「代表」。

因此除了詳細的詞庫資料外，還需要訂定一些斷詞時須注意的法則，當遭遇詞庫中未收錄的詞彙時，以其作為斷詞的依據，以下敘述之。

目前在國內對於中文語言處理的研究眾多，其中中華民國計算語言學學會在中文斷詞方面之研究已有多年經驗，在 1995 年該學會接受中央標準局的委託，擬定了「資訊處理用中文分詞標準草案」【15】，對於中文語言處理訂定了良好基礎。其中對於中文斷詞的規範，採語意及語法角度，設立兩基本原則及六項輔助原則，兩項基本原則為：

1. 語意無法由組成成分之語意直接相加而得到之字串，視為一分詞單位。例如，「吃飯」是由「吃」與「飯」的語意相加而來，但「吃虧」一詞一旦分開即失去其意義。
2. 詞類無法由組成成分直接得到，視為一分詞單位。例如，「難看」、「好聽」兩詞的詞類，無法由「難」、「看」或「好」、「聽」的組合獲得，故不可將其斷開。

六項輔助原則為：

1. 有明顯分隔標記應該切分之。例如，「吃了一頓飯」，由於「吃飯」一詞之間有其他的詞存在，故不得不將「吃飯」一詞斷開。
2. 附著語素盡量和前後詞合為一個分詞單位。例如，「數位化」一詞，「化」是依附在「數位」之後，故不將其斷開。
3. 使用頻率高或共現率高的字串盡量視為一個分詞單位。例如，「吃飯」、「等

車」之類常用詞，便不需將其分開。

4. 雙音節結構之偏正式動詞盡量視為一個分詞單位。例如，「清洗」為動詞型態且符合雙音節結構，雖不一定是常用詞，但仍將其視為一詞。
5. 雙音節加單音節之偏正式名詞盡量視為一個分詞單位。例如，「互動性」一詞可切為「互動」及「性」兩詞，但考慮到其音節結構以及結尾字與首字之間的緊密關係，我們不將其斷開。
6. 內部結構複雜之詞盡量切分之。例如「國立台灣藝術教育館」雖為專有名詞，但因為太過冗長，故將其斷為「國立」「台灣」「藝術」「教育館」。

上述法則原則上是在斷詞過程中，遭遇詞庫無收錄的字組或是無法判別詞性的字組時的處理方式，若是詞庫中已收錄的字詞，則採用 n-gram 技術，以最大法則（maximum matching）方式斷詞。所謂 n-gram 是由 n 個字所組成的字組，若 n =2，則稱為 bigram，也就是二元詞，n=3 即形成 trigram，即三元詞，依此類推；而最大法則是將文件內容，先以較大的 n 值進行斷詞，再依序將 n 減少，也就是說，文件內容中字數較多的詞將優先被找出。舉例來說，根據中研院所建立的中文語料庫，文件中若提到「類風濕關節炎」，則會當作一個詞，因這是一個專有名詞且已收錄於詞庫，就算是詞庫中還收錄了「關節炎」、「關節」、「風濕」、「類」等詞彙，也不須將其拆開。這正是上述斷詞原則所強調的精神，再以「國民大會」為例，在詞庫中除了「國民大會」，還另外收錄了「國民」與「大會」兩詞，但就詞義的角度來看，「國民」加上「大會」並不等於「國民大會」。事實上，只要詞庫內容夠齊全，斷出的結果大致上都有不錯的效果。

3.1.3 詞類判別

基本上在斷詞結束後，各詞彙的詞類皆會被標記，作為判別關鍵字以及建立詞彙索引的依據。舉例來說，「目前在國內對於中文語言處理的研究眾多」這句話，以中研院詞庫小組所發展的中文自動斷詞系統 1.0 版斷詞後的結果為：

目前　在　國內　對於　中文　語言　處理　的　研究　眾多

若標記各詞彙的詞類，則為：

目前（Nd）　在（P）　國內（Nc）　對於（P）　中文（Na）　語言（Na）　處理（VC）

的（DE） 研究（Na） 眾多（VH）

其中屬於體詞的詞類為 Nd（時間名詞）、Nc（地方名詞）、Na（名詞）；屬於述詞的有 VC（動作單賓述詞）、VH（狀態不及物述詞）；其他如 P（介詞）、DE（副詞）在語意上並無太大意義，事實上可依詞類過濾不具意義的詞彙，以加速文件索引的建制。

以下是文字處理的演算法：

```
Begin
    /*去除標點符號*/
    依序讀入文件字元 A
    if A = 標點符號 then
      if （A = 引號 and A 之前不存在冒號） or A = 頓號 then
          delete A 並標記其位置，以為一斷詞點；
      else A = 句號 //（所有具段落字句用途的標點符號，皆視為句號）；
    依序儲存 A 字元於暫存檔 B；
End.
Begin
    /*斷詞處理*/
    由暫存檔 B 依序讀入字元 C 於暫存區 M；
    若 C=句號 then 停止；
   for (n = 詞庫收錄最大詞數；n =1；n - -)
    {
       w = 1；
        if M 中的第 w 字元到第 w+n-1 字元為 n 元詞 then
            {斷出此字串連同其詞類一併記錄於暫存檔 S
           w = w + n}
           else w = w + 1；
        until w = 字串的倒數第 n -1 個字元；}
        End
```

3.2 詞彙索引

要從文件集合中找到含有特定詞彙的文件有兩種方法，一是循序搜尋法（sequential search），即將集合內所有文件從頭到尾逐字作字串的比對，這是最直接的做法，但是當文件數量增加（例如：檢索網頁），搜尋所花的時間甚長且不切實際。二是建立各詞彙的索引（index），這也是最常見的方法，索引是建構在文字上的資料結構，依搜尋演算法的不同而有多種型態，例如，樹狀（tree）、堆積（heap）等，目的是替代循序搜尋，以最少的時間找出欲搜尋關鍵詞的所在位置，雖然建立索引需要額外的儲存空間，還必須維護索引的完整，但因比循序搜尋節省時間，一般檢索系統皆採用之。原則上檢索系統會先蒐集文件，分析文件內容之後，將具意義的關鍵詞建立索引，以下說明之。

3.2.1 去除無效字

文件中並非所有的字詞都具有意義，為了文義的流暢，我們常使用一些語助詞、介詞、代詞，或一些無關緊要的形容詞、副詞等，這些字詞稱為無效字（stopword）。常見的例子，如「這個」、「那個」、「而」、「乎」等，均是為了修飾文詞，所以在各種不同主題的文件上，都可以發現他們的存在，因而不能做為查詢的條件。事實上，一份文件中，無效字的比例可能達到 80%以上【16】，因此，去除這些無效字，對於建構關鍵詞索引，可省下大量的空間及處理時間，其中空間的使用，往往可以節省 40%以上。

如前述，一篇文章中，體詞、述詞及否定副詞所包含的訊息比較具有意義，因此詞類可以當作判別詞彙有效詞或無效詞的依據。本研究也只為此三類詞彙建構索引。

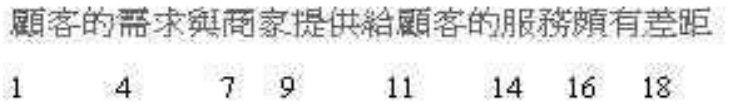
3.2.2 關鍵詞索引的建置

在使用者提出查詢關鍵詞之後，系統會檢視資料庫中的文件，以找出那些文件含有查詢關鍵詞及其在各文件中出現的次數，以判別該關鍵詞於各文件中所佔的權重。

目前建構關鍵詞索引較常見的演算法有下列數種：

1. 反轉檔（Inverted file）：反轉檔的做法是標示各關鍵詞出現在文件中的位置（offset），例如，「小明和我一起去學校」，「小明」是由第一個字元開始，「學校」則是由第八個字元開始。將所有文件內含關鍵字的位置標出後，將同一關

鍵詞的出現位置集合起來，故在反轉檔內，需要儲存兩種資料，見圖 3.1，一是各關鍵詞列表，另一個則是各關鍵詞於文中出現的位置。



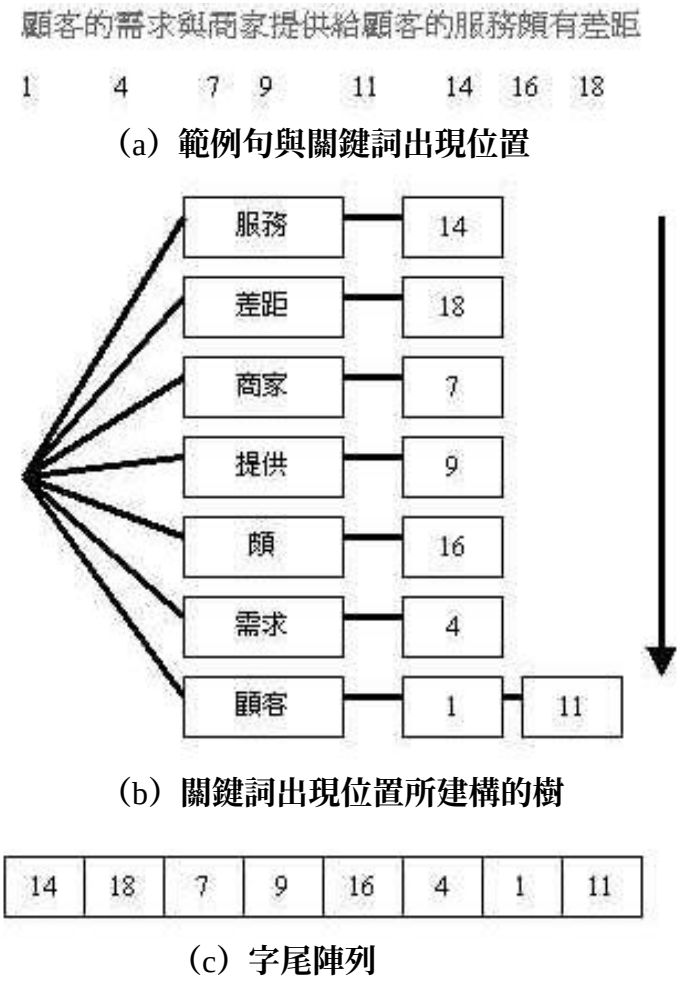
（a）範例句與關鍵詞出現位置

顧客	1,11
需求	4
商家	7
提供	9
服務	14
頗	16
差距	18

（b）關鍵詞列表與關鍵詞位置

圖 3.1 反轉檔

2. 字尾陣列（Suffix array)：字尾陣列的形成必須由字尾樹（Suffix tree）的結構所獲得。將所有關鍵詞以某演算法排序之後，以與反轉檔相同的方式將其出現位置排出，若是有相同詞首的詞彙，則共用其詞首，並將除去詞首後所剩餘的部分依相同方法排序，所有關鍵詞將形成一樹狀結構；見圖 3.2，其葉節點即為各關鍵詞於文件中出現的相對位置。此樹狀結構即為字尾樹。但以字尾樹來作為索引所佔的儲存空間過大，為了節省時間，我們依順序記錄所有的葉節點值，故可將一樹狀結構簡化成一個陣列（array)，此陣列即為字尾陣列；在進行搜尋時，只要依照該關鍵詞於陣列中的記錄位置，即可得知該詞彙於文中出現過的所有位置。



（a）範例句與關鍵詞出現位置

（b）關鍵詞出現位置所建構的樹

（c）字尾陣列

圖 3.2 字尾陣列（以詞首筆劃數量排序）

3. **識別檔**（Signature file）：**識別檔是將一篇文件切成若干大小相等的區塊，並將所有的關鍵詞分別以識別碼表示，舉例來說，假設有** abcd **四個關鍵詞，其識別碼則以四位元表示，分別為** 0001、0010、0100、1000，**若區塊** A **中出現** a **與** c **兩詞，則區塊** A **的關鍵詞資料紀錄為** 0101，**即詞** a **的識別碼** 0001 **及詞** c **的識別碼** 0100 **的和。我們只要檢視各區塊的關鍵詞紀錄，即能了解各關鍵詞於文中的分布情形（見圖** 3.3)。

顧客的需求與商家	提供給顧客的服務	頗有差距
0000111	0011001	1100000

(a) 範例句中各區段關鍵詞出現位置編碼

顧客	0000001
需求	0000010
商家	0000100
提供	0001000
服務	0010000
頗	0100000
差距	1000000

(b) 各關鍵詞編碼

圖 3.3 識別檔（七個關鍵辭需七位元識別碼）

上述三個方式各有優缺點，反轉檔因為內含所有關鍵詞的資料，故可直接對應任何種類的查詢，但其所需的儲存空間也最大；字尾陣列最省空間，但必須找出較為可行的關鍵詞排序法，同時字尾陣列由於需要經過對照表才能找到對應的詞彙，查詢時所需的資料隨機存取能力較為薄弱；而識別檔所表現的是各關鍵詞於特定區塊出現與否，但同一關鍵詞的出現頻率則無法表示。因此，我們以反轉檔為基礎，設計一演算法，以建立文件所含關鍵詞的索引。此外，在確定文件中各詞彙之位置之後，如何有系統的儲存，以利於查詢時的隨機存取，也是影響系統效能的重要關鍵。我們以下列的步驟儲存詞彙索引資料：

1. 蒐集該文件中所有具有相同詞首的關鍵詞並存放於相近之處，例如「電視機」、「電冰箱」、「電梯」的詞首「電」相同，若是詞首相同字數越多，則存放距離越近。
2. 將所有的中文詞依詞首的 ASCII 碼排序，詞首第一個字相同，再依第二個字排序，若又相同，則依第三個字排序，依此類推。
3. 將所有經排序後的中文詞依序紀錄成關鍵詞列表，並以各詞的第一個字建立一定位點，具相同詞首的詞彙分配在同一定位點範圍內，第一個字相同者，

便依第二個字建立次級定位點，並依此類推直到所有詞彙存在唯一定位點為止。

4. 計算各詞彙於文中所出現的位置，並對應關鍵詞列表順序紀錄之以為關鍵詞配置表。

反轉檔的資料結構可以記錄關鍵詞出現一次以上的情形，詞首定位點也加快了特定關鍵詞搜尋時隨機存取磁碟的速度。系統進行關鍵詞查詢時，只須先由該詞的第一個字開始依其指標搜尋，舉例來說，欲搜尋「國民大會」一詞時，則過濾出開頭為「國」的所有詞彙，再由其中挑出第二個字為「民」的詞彙，依此類推即可快速找到該詞彙於文中的出現位置。

3.3 文件叢集

隨著資訊量的增加，資料叢集（data clustering）的技術現今已廣泛應用於各領域中，例如：決策支援（decision-making）、機械學習（machine-learning）、資料採礦（data mining）、圖像辨識（pattern recognition）、文件檢索等【17】。資料叢集的意義，是依據資料之間的相似性分類，以利於分析處理。由於資料庫蒐集眾多文件，若是將所有文件任意存放，進行相類似文件之搜尋或相似性比對時，可能會因漫無目的找尋對象，而延長處理的時間，資料叢集便可避免此問題。

叢集大致上可分為下列步驟【18】：

1. 選取特徵樣本（feature pattern）：選取資料中具代表性或特徵的部分，例如，重要關鍵詞。
2. 定義特徵樣本相似性：在適當的領域中，決定以何種演算法則衡量及表示具相同特徵資料之間的關係。
3. 分群（grouping）：將具相同特徵的資料配置在同一類別中。

在本研究中，特徵樣本意味著各文件中具有特徵性質的詞彙，而相似性則是指文件之間各具特徵詞彙的相符合程度。以下將依序介紹之。

3.3.1 特徵詞彙的選取

基本上文件內出現的詞彙，扣除掉較不具意義的詞類以外，還需考慮以下的情形：

1. 令 D 為一文件集合，當 D 中的文件絕大多數都含有詞彙 t 時，則表示在 D 中：
   (1) t 的分布太過廣泛，不足以成為能代表文件特徵的詞彙。這類詞彙常見如：避免、整理、出現等。
   (2) t 本身的詞義太過籠統，可作為大範圍的類別名稱，也就是階層式概念中，屬於較上層者，以此類詞作為區分 D 中文件的依據幫助不大。例如：倫理、資訊、經濟等通常用於大分類的詞彙。
   (3) D 原本就已經是屬於某特定領域的文件集合，而 t 恰好為該領域的重要代表詞彙，例如，網路之於通訊類的文件，此時 t 可能可以成為 D 所屬類別之特徵詞彙，但無法以 t 細分 D 中的文件。
2. 令 D 為一文件集合，當詞彙 t 只出現在 D 中的極少數文件時，則表示在 D 中：
   (1) t 的詞義過於狹隘。
   (2) t 為極為罕見的詞彙。
   (3) t 的意義與 D 的領域差距太遠。

為避免上述情形，我們預先採用向量空間法，計算時設法將出現範圍過大或過小的詞彙的影響減到最小，換言之，即加重特徵詞於叢集過程中的權重。計算公式如下：

令一文件集合 $D=\{d_1,d_2,\Lambda\ ,d_N\}$，$T=\{t_1,t_2,\Lambda\ ,t_M\}$ 為 D 中所出現的有效詞集合，對於文件 $d_j\in D$ 而言，$t_i\in T$ 的權重 $w_{i,j}$ 為：

$$w_{i,j}=\frac{tf_{i,j}}{tf_{i,j}+0.5+1.5\dfrac{nt_j}{AVG(nt_j)}}\times\frac{\log\left(\dfrac{N+0.5}{n_i}\right)}{\log(N+1)} \quad 【19】$$

其中，$tf_{i,j}$ 為 t_i 在 d_j 中出現的次數，

nt_j 為 d_j 中的詞彙總數量，

$AVG(nt_j)$ 為 D 中詞彙數平均值，

N 為 D 中文件的總數量，

n_i 為 D 中含有 t_i 的文件數量。

$\frac{\log\left(\frac{N+0.5}{n_i}\right)}{\log(N+1)}$為反文件頻率項（inverse document frequency, IDF）。

此公式中，$tf_{i,j}$值的大小會決定t_i的權重，若$tf_{i,j} >> 1$，t_i權重會趨近於 1，反之，若$tf_{i,j}$接近 1，則nt_j與$AVG(nt_j)$的比值對t_i的權重$w_{i,j}$值有決定性的影響。

算出$w_{i,j}$之後，我們只需計算各詞彙的加權總和，就可以決定要以哪些詞彙來作為叢集的特徵詞，計算方法如下：

$$W_i = \sum_{j=1}^{n_i} w_{i,j}$$

W_i值越高，則代表t_i在D中越具代表性，我們可以設定一個下限值e，若$W_i < e$，則將t_i剔除於特徵詞集合以外，反之則為特徵詞彙。

3.3.2 文件相似性與分群

文件叢集（document clustering）的方法大致可分為兩大類：階層法（hierarchical method）及分割法（partitional method）。階層法是將所有文件先依照某些特性/特徵來分類（類別可能是事先定義或是臨時決定），每一類別再依其某些特徵或範圍大小持續地細分成若干子類別，各子類別又繼續細分，而成為一樹狀結構；而分割法是將整個文件集合分割成若干子集合，各子集合皆有其特徵（就文件而言，可能是關鍵詞在文件 d 中出現頻率、或是兩相異關鍵詞同時出現在 d 中的機率、也可能是同性質詞彙之間在 d 中的距離等），各文件依其內容劃分到特徵與其相似/相近的集合中，再藉由不斷修正各子集合的特徵屬性，來調整文件子集合的範圍。以常見的平方差法（squared error algorithm）為例，我們首先將各文件以二維圖形表示其特徵差異（如圖 3.4），進行叢集時，再將文件劃分為若干區域，各文件歸屬於何區域端看該文件與各區域之間的歐幾里得距離（Euclidean distance）大小，距離越小，表示叢集資料與分割區域之間的相似性越高，歐幾里得距離e的計算公式如下：

$$e^2(D,C) = \sum_{j=1}^{m}\sum_{i=1}^{n}\left\| d_i^{(j)} - c_j \right\|^2$$

其中 D 為文件，C 為某分類區域，$CI = \{c_1, c_2, \Lambda, c_m\}$為 C 的屬性集合，即

CI 為C 的特徵項，c_i 可為一詞彙、字串、字句、文章段落、或是文件本身；。$d_i^{(j)}$ 為文件 D 的某特徵項 d_i 以 C 的特徵項 CI 為基準所評判出的權重，也就是說，$d_i^{(j)}$ 是計算 D 針對 CI 中所有特徵的程度，而 $d_i^{(j)} - c_j$ 就是 D 與 C 在 CI 上的差距，圖 3.5 (a) 中，文件雖明顯分為兩類，但重疊區域的文件及為劃入區域內者，則無法明確的歸類，如能調整 C 的範圍，便可得到較為適當的文件分類方式（如圖 3.5（b））。

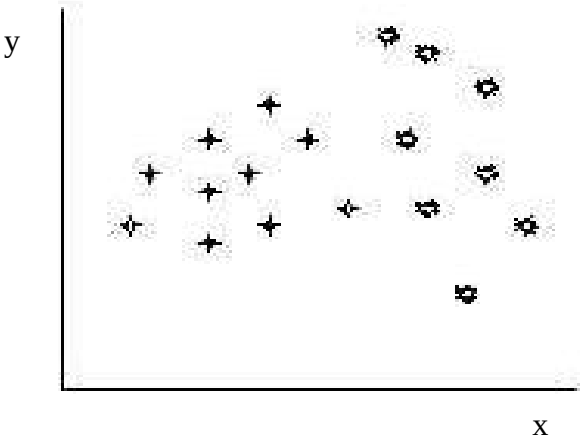


圖 3.4 文件的分布

C-Mean algorithm 為典型的分割法【20】，做法是先將文件分為 C 個叢集，並預設每個叢集的「中心」，所謂「中心」指的是該叢集的特徵趨向，再以向量型式表示；其次，依各文件與各叢集中心的距離，歸入最近的叢集中，之後以遞迴的方式，不斷的調整叢集的「中心」位置，使得所有文件與各叢集中心之間的距離總合能收斂到一穩定值，此時文件 d_j 距離叢集 c_i 的中心最近，c_i 即為 d_j 的歸屬叢集。但這裡就產生了一個問題，特定文件的特徵向量與各叢集質心之間的距離是相對的，也就是說，就算是 d_j 經計算應該歸屬於 c_i，但 d_j 與其他叢集的關係也不應忽略，事實上 d_j 歸屬於 c_i 只是相對比較下的結果，採用太過刻板（hard）的叢集方式反而會使 d_j 與其他叢集間關係的資訊流失。為反映此層關係，我們採用以模糊（fuzzy）為基礎的 C-Mean 演算法，作為判別文件關係及分群的準則。

模糊 C-Mean 叢集法（fuzzy c-mean clustering method）【21】主要還是以歐幾里得距離的計算為基礎，並加入一個歸屬係數 l，l 以二維矩陣表示，分別定義各文件與各個叢集之間的歸屬度，假設一文件 d 與某叢集 c 的歸屬度 $l_{d,c}$ 越大，代表 d 歸屬於 c 時，能

有較佳的分類效果，此時為使得 c 所代表的特徵能更貼近該叢集所屬文件的特徵，c 的中心特徵會朝 d 的特徵修正，試圖將 cd 兩者之間的歐幾里得距離最小化；在修正叢集中心後，$l_{d,c}$ 值也會增加。模糊 C-Mean 叢集法的公式及其相關定義如下：

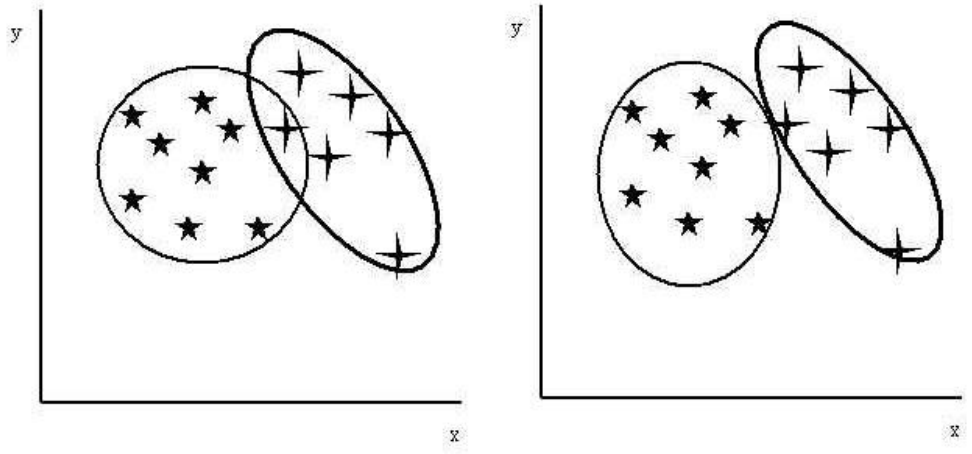


圖 3.5 分類區域的調整

令一文件集合 $D=\{d_1,d_2,\Lambda\ ,d_n\}$，今欲將 D 分為 a 個叢集 $C=\{c_1,c_2,\Lambda\ ,c_a\}$，則計算 d_i 與叢集 c_k，$1\le k\le a$，間距離的公式為：

$$J_m=\sum_{k=1}^{a}\sum_{i=1}^{n}\left(l_{k,i}\right)^m\left\|d_i-V_k\right\|^2$$

其中，m 為一固定常數，V_k，$1\le k\le a$，為 C_k 的質心，且

$$V_k=\frac{\sum_{i=1}^{n}\left(l_{k,i}\right)^m\cdot d_i}{\sum_{i=1}^{n}\left(l_{k,i}\right)^m}$$

其中，$d_i\in D$，$l_{k,i}=l_k(d_i)$ 為 d_i 與 C_k 之間的歸屬度，$\sum_k l_{k,i}=1$，且 $\forall l_{k,i}\ge 0$，而

$$l_{k,i}=\sum_{j=1}^{a}\left(\frac{\left\|d_i-V_k\right\|}{\left\|d_i-V_j\right\|}\right)^{-2/(m-1)},\quad 1\le k\le a,\ 1\le i\le n$$

由於 V_k 及 $l_{k,i}$ 是相互參考的兩參數，故必須經過許多次的調整才能找到適當值，其最終目的是設法將 J_m 最小化 (minimize)。J_m 的最小化意味著 $l_{k,i}$ 必須為極大值，此時 V_k 會達到一個穩定狀態而不再變動，之後便可由 $l_{k,i}$ 來了解文件與各叢集之間的歸屬程

度，若 $l_{k,i}$ 趨近於 1，則 d_i 的特徵也會趨近於 C_k 的分群原則。

模糊 C-Mean 叢集法的分群步驟如下：

1. 決定文件分類後的叢集數量。
2. 隨機預設文件各叢集之間的歸屬度 $l_{k,i}$， $0 \le l_{k,i} \le 1$。
3. 依 $l_{k,i}$ 計算出各叢集的質心 V_k。
4. 以 V_k 重新計算歸屬度 $m_{k,i}$
5. 重複步驟 3 及步驟 4，直到 J_m 收斂於某值，或小於預先設定的下限值 d 。（此時 $l_{k,i}$ 的值已漸趨穩定。）

3.3.3 新增／刪除文件

新增文件時叢集處理原則上有兩種做法，一是重新調整叢集的動作，二是將該文件併入已存在且較接近的叢集中（如圖 3.6）。從時間花費來看，第一種做法顯然較無效率，然而在新增的文件日漸增多的情況下，實質上的叢集質心可能已逐漸改變，若不重新調整，則失去文件預先分類的意義。故實作上，我們應該在文件增加數量超過特定比例時，應對資料庫中的文件集合重新分群，並求取各叢集的中心，重新分類之。唯文章數量及涵蓋範圍增加，單一叢集可能包含太多的文件，同時涵蓋主題也可能過於廣泛，導致失去預先分群的意義；反之，若文件數量減小，以致單一叢集內的文件數量過少，叢集的作用相對來說也會減少，因此在文件數量大量變化之時，叢集數量也必須加以調整。

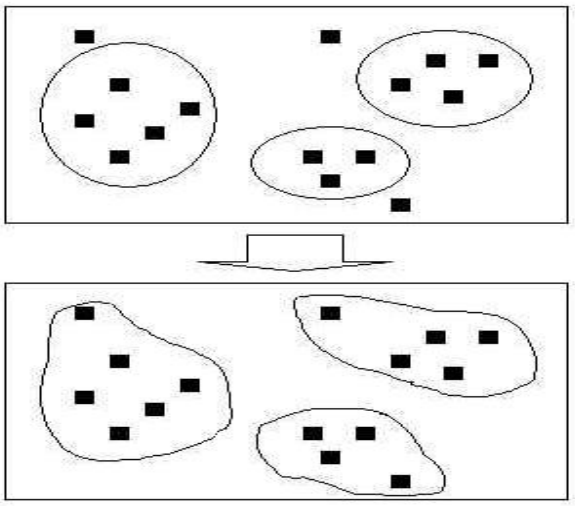


圖 3.6 新增文件後的重新分類

以下是以反轉檔建立文件索引的演算法：

Begin

/*去除無效字*/

讀入詞彙 t，其詞類標記為 A

if A = 體詞、述詞、否定副詞 then

將 t 存入關鍵詞彙資料檔 k；

else delete t（所有詞彙其詞類不為體詞、述詞、否定副詞的，皆視為無效詞刪除之）；

End.

Begin

/*索引建置*/

由關鍵詞彙資料檔 k 依序讀入關鍵詞 C 於暫存區 M；

若 C = null then 停止；

紀錄 C 於索引檔中的詞彙欄位，紀錄 C 的位置 L 於索引檔中的位置欄位

End

第 4 章　查詢處理及文件比對

查詢處理系統可說是資訊檢索系統的重心，系統接受使用者提出的查詢需求，再將符合查詢條件的文件或段落依序排列，並回傳給使用者，至於要如何明確的使系統了解使用者的需求，實為資訊檢索的一大課題。

我們依查詢方式的不同，將查詢分為關鍵詞查詢（key-word query）及全文檢索（full-text retrieval）兩部分。關鍵詞查詢係從使用者提出的查詢語句中，擷取符合使用者查詢概念的關鍵詞，作為查詢條件所進行的處理之謂。而全文檢索則是由使用者指定某一篇文件作為查詢條件，再從資料庫中找出與其語義相似的其他文件。並依其相似程度排序以為查詢結果。

4.1 關鍵詞查詢

以最簡單的方式找到需要的資料，是每個系統使用者所渴望的。但不管是網際網路上的搜尋引擎，或是圖書館中的館藏查詢等，目前都必須將查詢需求，以一個或多個「關鍵詞」表示。但是過於籠統的關鍵詞，符合條件的網頁往往很多，使用者難以一一檢視其內容。假若能以自然語言（natural language）來描述查詢，概念上應該更為精準。

4.1.1 自然語言分析

目前電腦還無法完全解析人類語言的意涵，因此如何由自然語言中分析出具代表性的詞句，是查詢處理的關鍵。本研究擬以一個句子中的述詞、體詞及否定副詞作為查詢條件，從查詢語句過濾出這些詞類的過程如下：

1. 處理查詢語句經斷詞及詞類判別，並保留其中之體詞、述詞、與副詞等詞彙。
2. 刪除「否定副詞」外的所有副詞，並將緊接於否定副詞後面的述詞或體詞標記「not」屬性，之後刪除該否定副詞。
3. 剩餘者皆作為查詢之關鍵詞，並根據使用者所輸入之要求，將各關鍵詞之間以 and、or 或 not 查詢條件。

4.1.2 自然語言的查詢延伸

當使用者欲查詢的資料是具體的事物，如台北市政府、ICRT 等，以關鍵詞檢索的搜尋系統通常都能提供不錯的查詢結果，然而，如前述，當查詢條件過於籠統及抽象，或是使用者本身也不知道查詢的確切對象時，符合條件的資料數目往往相當龐大，其內容也未必能切合查詢目標。因此，在使用者無法以數個精確的字詞來表達所要查詢的事物時，同義詞集合及詞義概念網路便可由詞義的角度，來修正此種誤差。利用事先已建置完成的概念網路，可以關鍵詞之間的關係及其相關程度，合理的延伸查詢條件，俾找出欲查詢但並未提出的查詢概念。而越符合查詢條件的文件排名越前面，以便於使用者修正查詢的方向，提高查詢的招回率（recall ratio，越高表示得到所得答案越完整），或讓使用者依排名順序瀏覽其內容。舉例來說，使用者想要了解調整攝影景深的技巧，於是輸入「攝影」以及「景深」兩關鍵詞，系統可依此推導出相關的詞彙，例如，「光圈」、「焦距」等作為延伸的關鍵辭，連同原有的一起加入查詢，使查詢條件更加完備，系統搜尋結果更加精確。

加入延伸關鍵詞的原因如下：

1. 自然語言通常是 context sensitive，其任一詞彙 T 不可完全視為一獨立的語義單元，T 與其他詞彙在詞義上可能存在某種關聯，如，同義（相同或相似）、反義、概括（廣義）、特例（狹義）等等；其關聯程度大小也依詞義而有所不同。
2. 不是使用者直接輸入之查詢條件 q，但與使用者所輸入條件相似或相關的詞彙，可能在某種程度上也符合使用者欲查詢的意向；舉例來說，使用者輸入「足球」，與之相關的詞彙，例如，守門員、自由球等，可能也要連帶地列入考量，取決條件則是與原查詢條件之間的相關程度的大小。
3. 資料庫中之某文章 D 之內容中並未出現 q，但卻有與 q 相似或相關的字詞，則 D 也可以列入查詢結果。

因此，在使用者提出查詢 q 之後，我們便以 q 為中心，在詞義概念網路中沿著同義詞集合之間的關係鏈結，檢視各相關同義詞集合，並以詞義概念網路中所規範的參數：關係、相關程度、及階層距離，計算查詢延伸商數（query extension quantity）：

<定義>查詢延伸商數：有一同義詞集合 a 與使用者輸入之查詢條件 $q = \{q_1, q_2, q_3, \Lambda\ , q_n\}$，

則a的查詢延伸商數$E_{q,a}$為：

$$E_{q,a} = \sum_{i=1}^{n} E_{q_i,a}$$

其中，a與q_i之間的相關程度為$re_{q_i,a}$，階層距離為$d_{q_i,a}$，而

$$E_{q_i,a} = \begin{cases} re_{q_i,a}^{\,d_{q_i,a}/t}, & if \quad E_{q_i,a} > e \\ 0, & if \quad E_{q_i,a} \le e \end{cases}$$

其中，t為調整參數，$t \neq 0$。

$E_{q,a}$（$E_{q_i,a}$）的大小代表a與q（q_i）在概念上相似的程度；$E_{q_i,q_i} = 1$；當$E_{q_i,a} < e$，則a會被視為與q_i之間的相似性不足，而不列入延伸查詢的範圍，其中，e為某一臨界值時，在此我們設e為0.1。列入延伸關鍵字詞的a將以$E_{q,a}$值作為其權重。由於計算$E_{q,a}$需要檢視q中概念與a之間所有的相關程度及概念層級，故整體計算的時間複雜度為$O(n)$。

查詢延伸處理的過程如下：

1. 找出使用者輸入各關鍵詞$q = \{q_1, q_2, \Lambda, q_n\}$在詞義概念網路$C$中所屬的同義詞集合$S = \{s_1, s_2, \Lambda, s_m\}$。
2. 計算所有在C中與S（任一元素）相鄰的同義詞集合a的$E_{q,a}$值。
3. 令$S = S \cup \{E_{q,a} \mid E_{q,a} \ge e\}$。
4. 重複步驟二和步驟三，直到不再有新的a加入S為止，S即為實際上用來查詢的查詢條件。
5. 將$E_{q,a}$代入相似性比對演算法，以作為a於文件比對時的加權。

以向量空間法（vector space model）【22】套用$E_{q,a}$計算相似性的過程如下：

今有一文件d_j及一查詢q，詞彙集合$K = \{k_1, k_2, \Lambda, k_t\} = \{d_j$中所出現的關鍵詞$\} \cup \{q$中出現的關鍵詞$\}$，令$w_{i,j}$為$k_i \in K$於$d_j$中的加權值（即$k_i$於$d_j$中的重要性），$w_{i,q}$為$k_i \in K$於$q$中的加權值（即$k_i$於$q$中的重要性），換言之，$d_j$可以$(w_{1,j}, w_{2,j}, \Lambda, w_{t,j})$表示；$q$可以$(w_{1,q}, w_{2,q}, \Lambda, w_{t,q})$表示，則$d_j$與$q$之相似度$sim(d_j, q)$：

$$sim\left(d_j, q\right) = \frac{\vec{d}_j \bullet \vec{q}}{\left|d_j\right| \times |q|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q} \times e}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

其中，若$k_i \in q$，則令$w_{i,j} = f_{i,j} \times idf_i$，且$e = 1$；

若$k_i \notin q$，則$w_{i,j} = f_{i,j} \times idf_i$，且$e = E_{q,i}$。

$$f_{i,j} = \frac{k_i\text{於}d_j\text{中出現的次數}}{Max\left(k_i\text{於}d_j\text{中出現的次數},\ i = 1,2,\Lambda\ ,t\right)},$$

idf_i = inverse document frequency for term k_i。

以關鍵詞作為檢索條件的文件搜尋，若採用查詢延伸的方法，應可提高查詢正確率，我們將在後續章節作深入的探討。

4.2 全文檢索

進行全文檢索時，使用者須先指定一標的文件G；若G並不在文件資料庫中，則新增之；否則直接存取G之關鍵詞索引，再進行檢索。由於此處我們取得的查詢關鍵詞，並不只是文件中所含的詞彙t，而是在詞義概念網路中，與t位於同一個或鄰近t且相關程度大於臨界值之同義詞集合的詞彙；因為每個詞彙都代表一種（含）以上的意義，每個意義又分別代表一個語意概念，這些語意概念可經由一連串的思考推演而獲得相互間之關係。而一篇文件的摘要可能是由許多諸如此類的概念關係所串聯而成，假若能將文件中所出現的概念串聯起來，雖不一定能完全代表整篇文章的大綱，但至少能拼湊出文章所敘述的大部分意義，這種概念的串聯稱為詞義鏈（Lexical chain）【23】。

4.2.1 詞義鏈

詞義鏈是近幾年興起，用以表達文章大意的概念串聯結構，本研究將以該結構中之「概念集合」作為查詢比對的依據。圖 4.1（a）中的文件所出現的「電腦」「網路」「網際網路」「數位」等詞彙，可藉由其概念之間的相互關係，相互連結成圖 4.1 之圖形（graph）。由於這些詞彙的概念大致上不脫資訊網路的範疇，我們可以由此得知該文件討論的主題應與資訊網路的概念有關。先前提到同義詞集合是以「詞義」作為分類的依據，詞彙之間可能存在某特定關係，例如反義或相似等，這些關係依程度大小，我們將

其分為極強（extra-strong)、強（strong)、弱（weak）三個等級【24】，極強關係只存在於兩完全相同的詞彙之間，即其詞彙與語義均相同；強關係則出現在下列情形：

甲、屬於相同同義詞集合中的詞彙：例如，摩托車之於機車。

乙、互為反義或具相似意義的詞彙：例如，強壯之於瘦弱。

丙、兩詞彙在分類上互為廣義/狹義關係：例如，學校之於公立學校。

剩餘者皆為弱關係；一個詞義鏈 L，可視為該文件的「摘要」之一；假設一個文件 d 中共有 m 個詞義鏈 $LC=\{LC_1,LC_2,\Lambda,LC_m\}$，表示該文件係由 m 個概念群所組成，而 LC_i 中計有 n 個同義詞集合，即 $S_i=\{s_{i1},s_{i2},\Lambda,s_{in}\}$，$d$ 的同義詞集合 $S=S_1\cup S_2\cup\Lambda\cup S_m$，則詞義鏈 LC_i 在文件 d 的加權 $w_{LC_i}(1\le i\le m)$，和屬於 LC_i 的所有同義詞集合 S_i 在 d 中出現的次數總和有關，即：

$$w_{LC_i}=\frac{\sum_{j=1}^{n}f_{ij}}{\sum_{l=1}^{m}\sum_{j=1}^{k}f_{lj}}$$

其中，f_{ij} 為 $s_{ij}(s_{ij}\in S_i)$在 d 中出現之次數，k 為 LC_l 中所有同義詞集合之數量，即 $|S_l|\,(1\le l\le m)$。也就是說，LC_i 中的辭彙出現的次數越多，LC 於 d 中被強調的情形越明顯。

以詞義鏈代替關鍵詞，能以「概念集合」的角度分析文件之間的相似性。一個詞義鏈 LC 中的詞彙 $x\in LC$ 與詞彙 $y\notin LC$ 所屬同義詞集合假設分別為 A 與 B，而 LC 是否將 B 加入以延伸其範圍，則需視 x 與 y 之間的關聯程度而定。為避免詞義鏈無限度的擴張，我們需設定一臨界值(threshold) e，唯有 y（或 B）與 x（或 A）之間的相關程度超過 e，B 才被允許加入 LC；在此同時，我們也必須設定一個有效範圍，避免詞義鏈延伸過於廣泛。因此，處理一篇文件 d 所含詞義鏈的演算法如下：

詞義鏈 $LS_d=\{L_1,L_2,\Lambda,L_k\}$，其中，$L_i\cap L_j=f$，$i\ne j$，而要找出 $L_i=\{s_{i1},s_{i2},\Lambda,s_{im}\}$，則應找出 L_i 中各同義詞集合 s_{ij}，s_{ij} 則是由 d 中比較具代表性的詞彙 $T=(t_1,t_2,\Lambda,t_n)$所屬的同義詞集合 $S=(s_1,s_2,\Lambda,s_n)$中挑出，$n=\sum_{i=1,k}mi$，其中 m_i 為 L_i 中之同義詞集合之數量。其原則是：若是可以由 L_i 中的某個同義詞集合 s_i 連結到 d 中的某詞彙 t_a（$t_a\subseteq T$）所屬的同義詞集合 s_a，且 s_i 與 s_a 之間的關係強度大於 e，則將 s_a 所屬詞彙（同義詞集合）加

入 L_i。總結上述的定義，文件內含詞彙串聯的步驟如下：

1. 去除文件 D 中的無效詞後，留下關鍵詞 $T=\{t_1,t_2,\Lambda,t_n\}$，並找出 T 所對應之同義詞集合 $S=\{s_1,s_2,\Lambda,s_m\}$，$m \le n$。
2. 設定串聯標準 e 及串聯有效階層距離 ld。其中 $e \in [0,1]$，$ld \in$ 正整數。
3. 令 $LC(s_i)=\{s_i\}$，$i=1,2,\Lambda,m$。
4. 於詞義概念網路 CN 中，檢視 s_i 與 s_j 的相關程度 re_{ij}。其中 $\forall s_i \exists s_j$，$1 \le j \le m$，$i \ne j$，若 $re_{ij} \ge e$，$LC(s_i) \cap LC(s_j)=f$，且 s_i 與 s_j 之間的階層距離 $ld_{ij} < ld$，則 $LC(s_i)=LC(s_i) \cup LC(s_j)$。

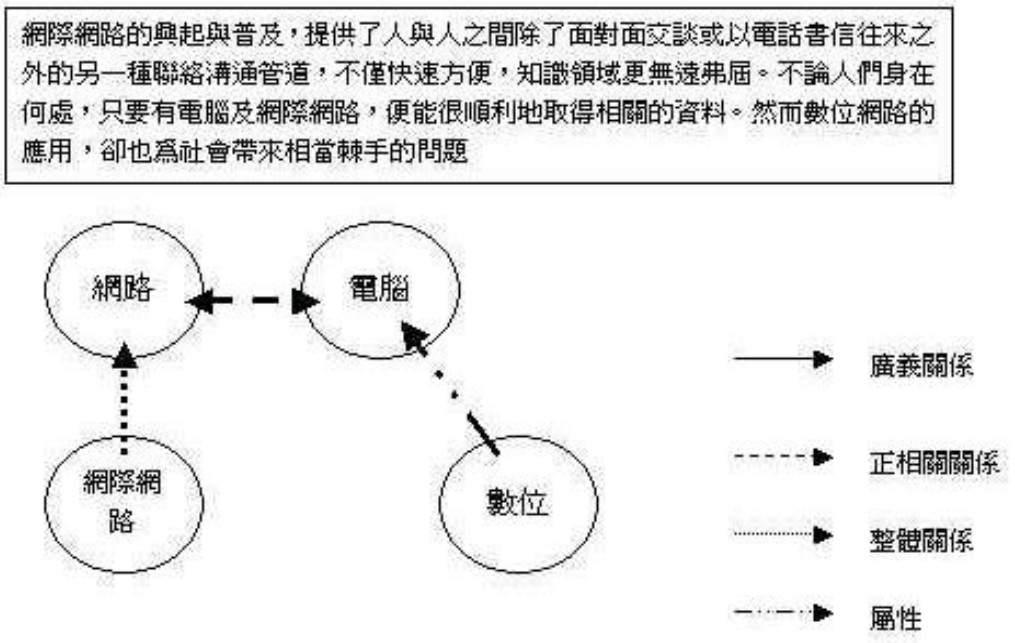


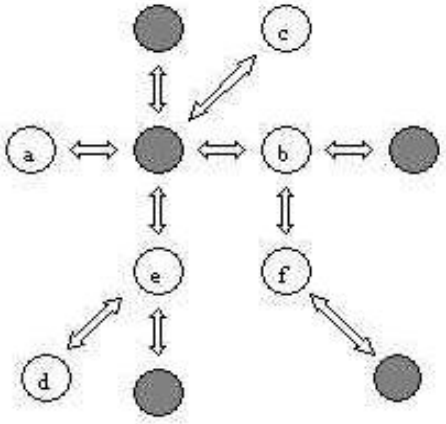
圖 4.1 詞義鏈的例子

4.2.2 詞義鏈與隱含概念

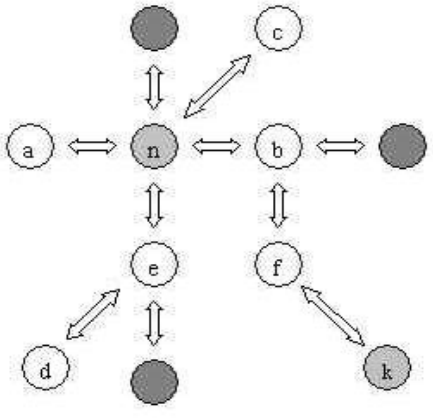
詞義鏈是將文件 D 中曾出現的詞彙，依照詞義概念網路 CN 中的相互關係，而相關程度強的詞彙串聯而成。若存在某些同義詞集合 C 並未在 D 中出現，但在詞義概念網路中卻與 D 中之詞義鏈 L 關係密切，即和 L 中某概念 f 之間的相關程度 $re_{Lf} \ge e$，則 C 應是 L 的的隱含概念，應加入 L 中，以為 L 之延伸。例如，D 中曾出現 a、b、c、d、e、f

等六個詞彙，在 CN 上的分布亦假設如圖 4.2(b)所示，且相互之間的相關程度都有一定水準以上，意即，$re_{ij} \in \{a, b, c, d, e, f\}$，換言之，此六個詞彙可能是隸屬於同一思路的範疇，我們將此六個詞彙納入同一個詞義鏈 L 內。在此同時，CN 中有一概念 n 與 L 部分成員相關程度高，我們認為 n 應加入 L 中。另外，若圖 4.2（b）中之節點 k 雖不在 D 中，但 $re_{kf} \geq e$，則依 L 之延伸演算法，亦應加入 L 中。

要找出文件 D 中未出現但應加入其詞義鏈 L 的詞彙，應先將 D 中比較具代表性的詞彙 $T = (t_1, t_2, \Lambda, t_n)$ 挑出，再檢視 CN 中 C 對應的各個同義詞集合 $S = (s_1, s_2, \Lambda, s_n)$，並以詞義鏈演算法找出 m 個詞義鏈 $LS = (L_1, L_2, \Lambda, L_m)$。若是可由 D 中某些詞彙 t_p，$1 \leq p \leq n$，$t_p \in T(s_p \in S)$ 聯想到 s_i，則將 s_i 加入 s_p 所屬之 L_r，$1 \leq r \leq m$，就如同圖 4.2，由 a、b、c、d、e、f 等六個詞彙聯想到 n 一樣。但此處要考慮同義詞集合之間的連結方向。如圖 4.3 中，n 的入關係鏈結 $IRL = \{a, b, d, e, f\}$，出關係鏈結 $ORL = \{a, b, c, d, e\}$。唯圖 4.2（b）中，k 若為 f 的屬性，我們以 k 所屬同義詞集合的入關係鏈結平均相關度來作為判別詞義鏈延伸的依據，茲定義如下：



（a）六個詞彙在詞義概念網路上之分布



（b）k 亦應加入 a~f 所在的詞義鏈中

圖 4.2 從詞義概念網路推導出文章應隱含 n 與 k 的概念

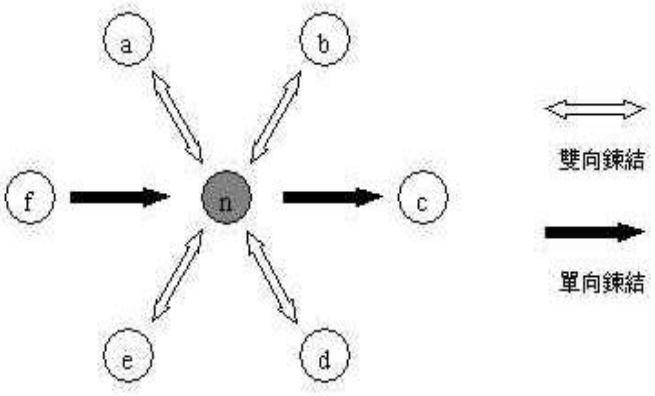


圖 4.3 入關係鏈結及出關係鏈結（c 為 n 的屬性，n 為 f 的屬性）

<定義>入關係鏈結平均相關度（In-let relationship link Average Degree，*IAR*）：**假設在詞義概念網路中，有** t **個同義詞集合** $S=\{s_1, s_2, \Lambda, s_t\}$ **指向同義詞集合** n**，無論其種類為何，** R **與** n **之間的相關程度** $RE_n=\{re_{1n}, re_{2n}, \Lambda, re_{tn}\}$：**則** n **的入關係鏈結平均相關度**

$$IAR_n = \frac{\sum_{i=1}^{t} re_{i,n}}{t}$$

每個同義詞集合在詞義概念網路 *CN* **中的** *IAR* **值是固定的，只有在** *CN* **有所更動時才會改變。同義詞集合** n **的** *IAR* **大，代表由其他同義詞集合** S **指向** n **的連結，具有較高的相關性，這代表我們可以很容易的由** S **聯想到** n。**反之，則可能是下列幾種情形：**

1. n **的概念過於廣泛，並不專屬於某特定領域。例如，方向，政府等。**
2. n **的概念過於獨特，只專屬某特定領域。與其他同義詞集合之間並無共通的概念，例如，超氧化物歧化脢**(Superoxide Dismutase，SOD)。

基於以上的結論，若有文件 D **中一詞彙集合** T **與** n **之間的平均相關度（稱為** $IAR_{T,n}$**）比** IAR_n **值大，就代表在整個詞義概念網路中，** T **是所有與** n **有關係的同義詞集合中，相關程度較高的一群。也代表** T **與** n **可能是同一領域的詞彙，因此可以將** n **加入** T **的詞義鏈中。計算** $IAR_{T,n}$ **的方法如下：**

文件 D 中存在關鍵詞集合 $T=\{t_1,t_2,\Lambda,t_m\}$，$T$ 所對應之同義詞集合指向 n 的相關程度 $RE_{T,n}=\{re_{t_1n},re_{t_2n},\Lambda,re_{t_mn}\}$，$m\le t$，則在 D 中 n 的入關係鏈結平均相關度：

$$IAR_{T,n}=\frac{\sum_{i=1}^{m} re_{t_i,n}}{t}。$$

若 $IAR_n < IAR_{T,n}$，即使 $n\notin T$，n 仍需視為由 D 產生之詞義鏈的一部份。

因此，在建構詞義鏈完成之後，詞義鏈延伸的串聯步驟如下：

1. 參考 CN，計算 CN 所有同義詞集合 $s_n\in CN$ 的入關係鏈結平均相關度 IAR_n。
2. 以建構完成的詞義鏈集合 $LC=\{LC_1,LC_2,\Lambda,LC_k\}$ 為基礎，找出 CN 中，與 LC_i，$i=1,2,\Lambda,k$ 內各同義詞集合之階層距離小於 ld 的所有同義詞集合 $S_e=\{s_1,s_2,\Lambda,s_p\}$。
3. 計算 s_j，$j=1,2,\Lambda,p$ 與 LC_i 中所有同義詞集合的入關係鏈結平均相關度 IAR_{ij}。
4. 將 s_j 的入關係鏈結平均相關度 IAR_j 與 IAR_{ij} 加以比較，若 $IAR_j < IAR_{ij}$，則 $LC_i=LC_i\cup s_j$。

在串聯檢驗完成後，將以各詞義鏈出現的頻率，來凸顯該語意概念於文件中被強調的程度。此時容易發生兩種情形：

1. 假設文件 D 中共 n 個詞彙會生成 m 個詞義鏈 $\{LC_1,LC_2,\Lambda,LC_m\}$。當 e 過小或 ld 過大，使得 $n>>m$，其極端是 $m=1$，這些詞義鏈 LC_i 的組成元素可能已廣泛涉獵各領域，不再是單一主題，此時，容易因 LC_i 包含概念的雜亂，而無法明確的判斷 D 的內容所描述的重點或核心議題為何。
2. 反之，若是 e 過大或 ld 過小，則 LC_i 中包含的同義詞集合數量太少，其極端是 $m=n$，此時每一詞義鏈只包含一個概念，難以表達一個完整的主題敘述，效果與採用單一關鍵詞類似，無法表現出以詞義鏈為「摘要」為檢索方式的優點。

另外，因為詞義鏈不像詞彙是獨立存在，各詞義鏈之間若有部分交集，例如，圖 4.4 中，文件 A 的兩個詞義鏈 A1 和 A2，同時與文件 B 的兩個詞義鏈 B1B2 有部分交集，此時因為 A1A2 以及 B1B2 中有部分成員是重複的，無法就單一詞義鏈的角度來比較其在兩文件的比重大小。對於這個問題，我們以適度提高 e 及／或降低 ld 來應對，e 提高

表示同義詞集合之間要具備更強的相關程度才能結合成詞義鏈，降低 ld 表示要減少每一詞義鏈中概念的數量。意即，設法讓 $A_i \cap B_j = f$ ， $i, j = 1,2$ 。原有的詞義鏈結構會因此縮小，若兩文件詞義鏈仍有部分交集的情形，則將所有交集的詞義鏈視為一個新的詞義鏈。因此，如何適當調整 e 及 ld 的大小，對於文件檢索的精確性有很大的影響。

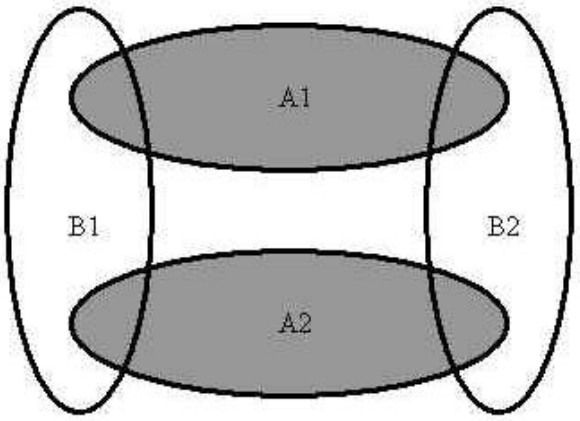


圖 4.4 兩文件之詞義鏈部分交集

全文檢索查詢處理流程如下：

1. 由使用者以某特定文件 D 作為查詢條件。
2. 文件資料庫中是否存在 D，若不存在，則新增此文件之後，再進行下一步，否則直接進行步驟 3。
3. 將 D 的關鍵詞索引調出。
4. 以 D 中所含的關鍵詞集合 $T = \{t_1, t_2, \Lambda, t_n\}$ ，找到其同義詞集合 $S = \{s_1, s_2, \Lambda, s_m\}$ ， $m \le n$ ，產生成詞義鏈集合 $LC = \{L_1, L_2, \Lambda, L_k\}$ ， $k \le m$ 。
5. 當出現詞義鏈相互部分交集的情形時，適度提高 e 或降低 ld ，直到不再出現部分交集為止。
6. 統計 D 中各詞義鏈 L_i 的出現次數 $Count(L_i)$ 。
7. 以 LC 查詢其他與 D 同一分類的文件 $E = \{e_1, e_2, \Lambda, e_p\}$ ，並統計 L_i ， $i = 1,2,\Lambda, k$ 在這些 e_j ， $j = 1,2,\Lambda, p$ 中出現的頻率 f_{ij} 。

之後，將 $Count(L_i)$ 和 f_{ij} ， $i = 1,2,\Lambda, k$ ， $j = 1,2,\Lambda, p$ 交由比對系統，做 D 與 e_j 的相

似性比對。

4.3 文件的比對

所謂文件的比對就是以比對演算法，找出最符合查詢條件的文件，並依相似程度排序。查詢條件是指使用者輸入的關鍵詞，或經全文檢索及詞義鏈處理之文件。我們將查詢條件的出現頻率，代入演算法中以計算文件與文件之間的相似性，或文件與關鍵詞查詢之間的符合程度。

本研究將採用向量空間模組（Vector space model ）分析計算的結果，並對其結果作準確性的驗證。以下分別簡述向量空間模組的原理：

4.3.1 向量空間模組

向量空間法是將文件 d 中所含的詞彙以向量 $\vec{d} = \{V_1, V_2, \Lambda, V_n\}$ 表示，每個向量元素 V_i，$i = 1,2,\Lambda, n$ 即代表一個關鍵詞 t_i，元素值 $|V_i|$ 則取決於 t_i 在 d 中的代表性，例如，在 d 中出現的次數。$\vec{d}$ 則代表 d 的文意走向，再以兩文件（d_i, d_j）的向量內積值來計算相似程度 $sim(d_i, d_j)$：

$$sim(d_i, d_j) = \frac{\vec{d}_i \bullet \vec{d}_j}{|\vec{d}_i| \times |\vec{d}_j|} = \frac{\sum_{k=1}^{t} w_{k,i} \times w_{k,j}}{\sqrt{\sum_{k=1}^{t} w_{k,i}^2} \times \sqrt{\sum_{k=1}^{t} w_{k,j}^2}}$$

其中，$\vec{d}_i$ 及 $\vec{d}_j$ 代表文件 d_i 及 d_j 的向量，k 表示在 d_i 或 d_j 中出現過關鍵詞的個數集合，至於 $w_{k,i}$ 與 $w_{k,j}$ 分別代表某關鍵詞 k_a 於 d_i 及 d_j 中出現的次數乘上文件逆頻率值（inverse document frequency，IDF），此參數代表 k_a 於此類文件中的代表性大小。至於 t 的認定方式如下：假設 d_i 及 d_j 中的關鍵詞集合分別是 $T_{d_i} = \{t_1, t_2, \Lambda, t_n\}$ 以及 $T_{d_j} = \{t_1, t_2, \Lambda, t_m\}$，$T_{d_i d_j} = T_{d_i} \cup T_{d_j} = \{t_1, t_2, \Lambda, t_p\}$，即 p 是 d_i 及 d_j 中關鍵詞的數量，其中一部份 t 只在 d_i 或 d_j 中出現，另一部份則同時出現在 d_i 與 d_j 中。

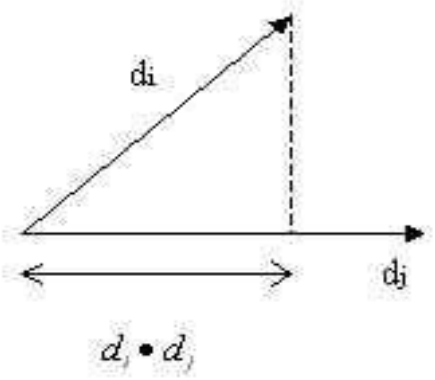


圖 4.5 向量內積

向量空間模組雖然有以下幾項缺點：

1. 以語意的角度，詞彙與詞彙之間的關係並非是完全獨立，而可能存在若干相關性，故將不同的詞彙視為相互垂直的向量，容易產生文義上的誤判。
2. 異詞同義的情形，例如「進監獄」與「進牢房」意義相同，但會被視為意義不同的詞彙。
3. 同詞異義的情形，這在英文中較為常見，但是中文處理也有此項問題，例如，「獵戶」一般指的是獵人，但從星象學來看，「獵戶」則是星座的名稱。

但我們以先前提到的「synset」同義詞集合，已可由語意角度，考量字詞之間的相關性並計算其相關程度解決之。向量空間模組是計算文件與查詢中所有元素的相似度，故其時間複雜度約為 $O(n^2)$。

4.3.2 相關反饋

所謂反饋（feedback）指的是透過使用者或是先前所得的結果，對系統作修正與調整，以產生更為精確結果的過程。由於資訊檢索的過程中，我們所得到的結果與文件之間實際的相互關係與相關程度不見得完全符合，大抵上查詢結果集合（answer set，AS）與相關文件集合（relevant document set，RS）會形成四個子集合，包括 $s_1 \not\subset AS$ 且 $s_1 \subseteq RS$、$s_2 \subseteq AS$ 且 $s_2 \not\subset RS$、$s_3 \subseteq AS$ 且 $s_3 \subseteq RS$、與 $s_{41} \not\subset AS$ 且 $s_4 \not\subset RS$，見圖 4.6。我們以反饋的方式，盡量使查詢結果集合與相關文件集合相契合。

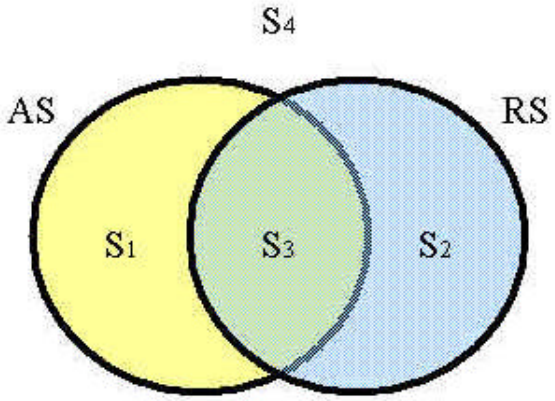


圖 4.6 查詢集合關係圖

理論上我們若能加強 s_3 部分文件的加權，並削弱 s_4 文件的權重，就能使查詢更為精確而切合實際，這個過程稱為相關回饋（relevant feedback）。但相關文件集合我們無法事先得知，目前的作法除了以人工進行查詢修正外，就只能以現有的查詢結果為依據來修正查詢。以下介紹回饋步驟。

向量空間模組的相關回饋常見的做法如下：

假設由原查詢 $\vec{q}$ 所得到的文件集合 D_r， D_n 為與 $\vec{q}$ 不相關之文件集合，則經過反饋後的查詢 $\vec{q}_e$ 有下列三種方式：

(1) Standard Rochio：$\vec{q}_e = a\vec{q} + \frac{b}{|D_r|}\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{g}{|D_n|}\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$ 【26】

(2) Ide Dec Hi：$\vec{q}_e = a\vec{q} + b\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - g\max_{non-relevant}(\vec{d}_j)$ 【27】

(3) Ide Regular：$\vec{q}_e = a\vec{q} + b\sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - g\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$ 【28】

$|D_r|$、$|D_n|$ 分別為 D_r 與 D_n 集合內的文件數量。

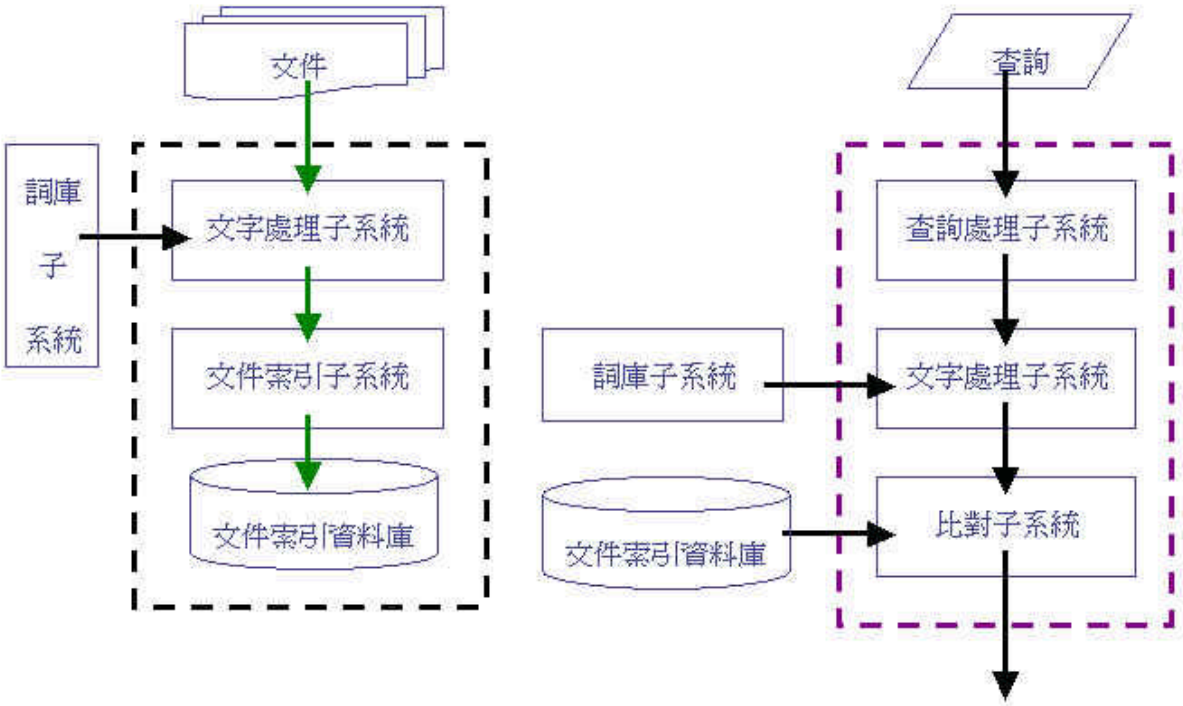
a 、 b 、 g ：調整常數。

$\max_{non-relevant}(\vec{d}_j)$：與查詢最不相關的文件向量。

得到 $\vec{q}_e$ 後，系統會重新以 $\vec{q}_e$ 為 $\vec{q}$ 重新查詢，而重複上述的過程直到查詢結果收斂於一固定值。事實上，上述三個方法一般而言所得效果大致相近，我們將取 Standard Rochio 的方式進行相關反饋的工作。

第 5 章　系統架構與實驗結果

本研究所提出的實作系統模組流程圖如圖 5.1 所示：



（a）前置作業　　（b）使用者查詢

圖 5.1　系統處理流程圖

本系統流程可分為前置作業及使用者查詢兩部分。

1. 前置作業：請參考圖 5.1（a），所有儲存在系統內的文件，都必須經過文字處理模組，進行斷詞及文法結構的分析。之後這些經斷詞的文件內容，會由文件索引系統，記錄文件中每一個具關鍵意義的字詞於文章中所在的相對位置，以為字詞的索引。並將該索引儲存於索引資料庫中。同時，也會藉由統計各文件內含各關鍵字詞出現的頻率，並將具相似內容的文件分類存放。
2. 使用者查詢：請參考圖 5.1（b），使用者所提出的查詢語言，會由查詢處理系統解析蘊含在查詢語言內的查詢條件，對系統提出查詢。系統會由文件索引

中，找出含有查詢條件關鍵詞的文件，在取得文件中該關鍵詞出現次數之後，依照各相似度計算的演算法，將最符合查詢條件的文件依序排列，供使用者參考。

將系統依功能細分成下列子系統，其名稱與功能簡述如下：

1. 詞庫子系統：包括一有效詞詞庫及一同義詞庫，其內容依照第二章所定義的辭庫內容加以實作，有效詞詞庫將有效詞提供給文字處理子系統，同義詞庫則提供同義詞集合資料，以作為語法分析的依據。
2. 文字處理子系統：為針對中文文件或查詢，按照第三章所定義的文字處理方式，包括斷句、斷詞、擷取有效詞等模組的的實作。
3. 文件索引子系統：乃依據第三章之索引建置部分，將資料庫中文件建立反轉檔索引，並依其屬性分類。
4. 查詢處理子系統：依據第四章之查詢處理所描述的細節，選擇使用者欲進行的查詢模式，擷取使用者所輸入之關鍵字或查詢文件，並延伸其查詢條件。
5. 比對子系統：依據第四章之定義，將查詢處理子系統提供的關鍵詞或詞義鏈資訊，以兩種不同的比對方式，進行相似性比對，並排序其結果。

3. 系統核心的作業流程接下來將針對系統處理程序，以各子系統為單元說明之。

5.1 詞庫子系統

詞庫子系統包含兩大部分，第一部份是有效詞詞庫，該詞庫蒐集所有可能的中文有效詞資料。有效詞詞庫依詞彙所含的字數，分為一元詞至八元詞共八類，各類中收錄中文常出現的詞彙共 13,3342 個，並詳細記錄各詞彙的詞類。至於字數超過八以上的詞彙，因為較為罕見，在此不加以考慮。第二部份是中文同義詞詞庫，該詞庫是由詞彙及關係鏈結組成的網狀結構，稱為詞義概念網路，其中內含 94 個詞彙於 87 個概念之中，概念之間依照第二章所規範的相關定義，將其間的關係種類加以分類，並記錄其關係程度值及其他相關資訊，以作為查詢延伸時的依據。圖 5.2 為詞庫子系統的架構圖。

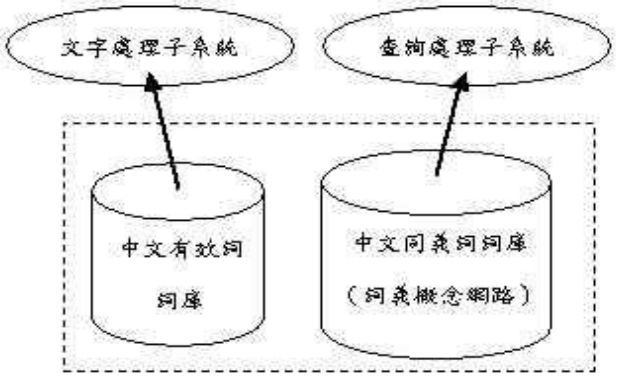


圖 5.2 詞庫子系統

詞庫內容的蒐集皆以人工方式進行，在中文有效詞部分，本系統採用由中研院所提供的的中文語料庫，統計詞彙出現次數以擷取詞彙，並配合微軟視窗內建於新注音輸入法的辭庫，建構出一有效詞詞庫。至於中文同義詞詞庫的建構，由於中文詞數量龐大，若以所有中文詞彙為基礎建立詞義概念網路，工程浩大，故在此我們只選取特定的領域，以該領域的術語（terminology）建立詞義概念網路，並以檢索該領域的文件及關鍵字驗證其結果，若是成效顯著，將再逐步擴大中文同義詞詞庫的涵蓋範圍。

5.2 文字處理子系統

文字處理子系統，乃配合先前所建置的詞庫子系統，對文件的文法結構，作適當斷句、斷詞、判別詞類後，提供其他子系統作後續的處理。如圖 5.3 所示，欲新增於資料庫中，且需要進行分析的文件，或是查詢處理子系統欲分析的查詢語言讀入後，參照標點符號資料庫，將文字部份以句子為單位分開後，再由詞庫系統所提供的有效詞資料，斷出句子內的有效詞。其後標明各有效詞之詞類，並將結果回傳給提出分析要求的系統。

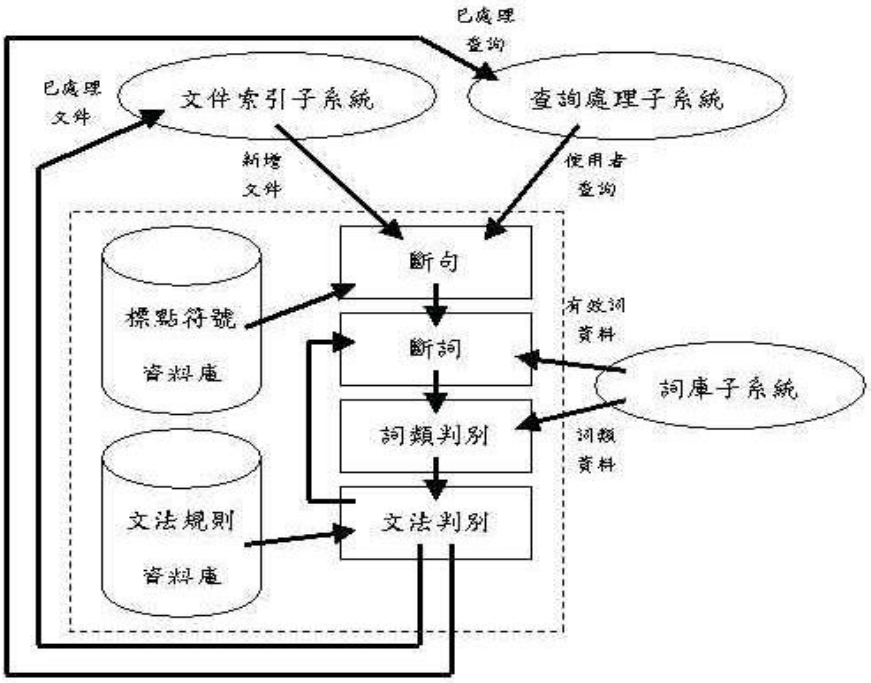


圖 5.3 文字處理子系統

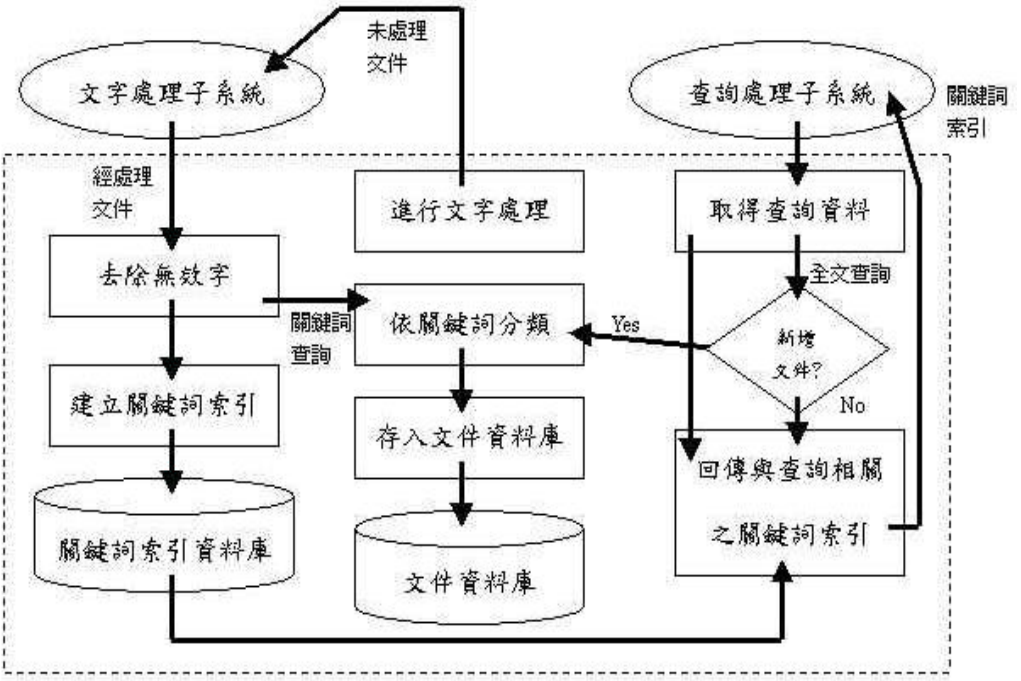


圖 5.4 文件索引子系統

5.3 文件索引子系統

本子系統主要負責以下工作：

1. 在文件資料庫中新增文件。
2. 回應由查詢處理子系統所提出的關鍵詞索引。

新增文件會發生在兩個情形下：

1. 使用者進行全文分析查詢時，由於全文檢索的查詢條件是一特定文件，當系統中並無該文件的資料時，則系統會要求使用者輸入該文件並建立該文件所屬的索引等資料。
2. 系統定時的資料擴增。

系統新增文件時，首先會查看文件資料庫中是否存在該文件，若是則停止；否則系統將該文件存於文件資料庫中，同時將其內容傳送給文字處理子系統，進行斷詞及詞類分析。之後文字處理子系統會回傳經處理後的文件內容。如圖 5.4 所示，系統在接收由文字處理子系統傳回的文件後，即依詞類消除其無效字，接著一方面將所有文件中曾出現的關鍵詞，依其在文件中出現的位置，以第三章所說明的方式建立關鍵詞索引，並以文件為單位，儲存於關鍵詞索引資料庫中。另一方面，已去除無效字的文件，便可依文件中關鍵詞的出現方式，將其依與其他文件的相似性預先分群，也就是將主題類似的文件存放在一起，以利於檢索的效率，之後將文件本文放在文件資料庫中，即完成新增文件程序。

文件索引子系統在接收查詢處理子系統的提供的查詢條件時，會先判斷查詢條件是自然語言語法或是某特定文件，若是前者，則回傳查詢所指定的關鍵詞索引資料，若是文件，則會先判別該文件是否為新文件，若是，則進行新增文件之程序，否則回傳該文件所含關鍵詞的索引。

5.4 查詢處理子系統

為考慮到使用者的需求，查詢系統分為關鍵詞查詢及文件查詢兩部分。關鍵詞分析

查詢負責從使用者提出的自然語言查詢語句中，擷取符合使用者查詢概念的關鍵詞，再以這些關鍵詞進行查詢。而全文分析系統查詢即為使用者指定某一篇文件，再由資料庫中找出與其相似的其他文件。系統架構見圖 5.5。

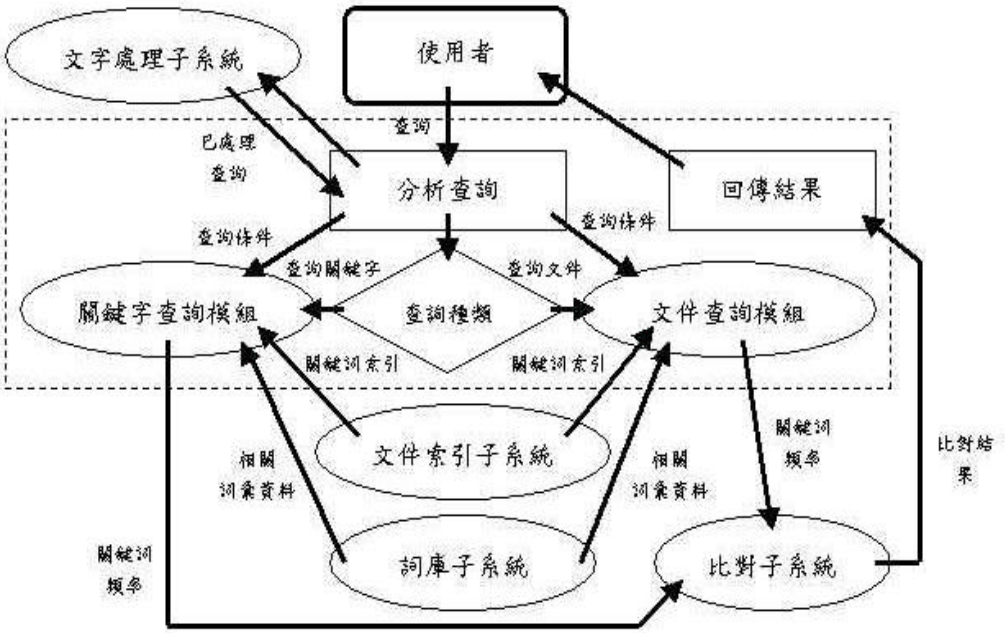


圖 5.5 查詢處理子系統

（1）自然語言分析模組：

本模組將使用者輸入的自然語言查詢，分析其查詢涵義後，與詞庫系統內的同義詞詞庫連結，找出同義或相關的詞彙，並計算其查詢延伸商數，剔除查詢延伸商數未達特定臨界值的相關詞彙，其他的則加入查詢關鍵詞中作為查詢條件，之後索引資料庫將查詢條件的詞彙索引傳給比對系統進行查詢比對。模組架構圖如下：

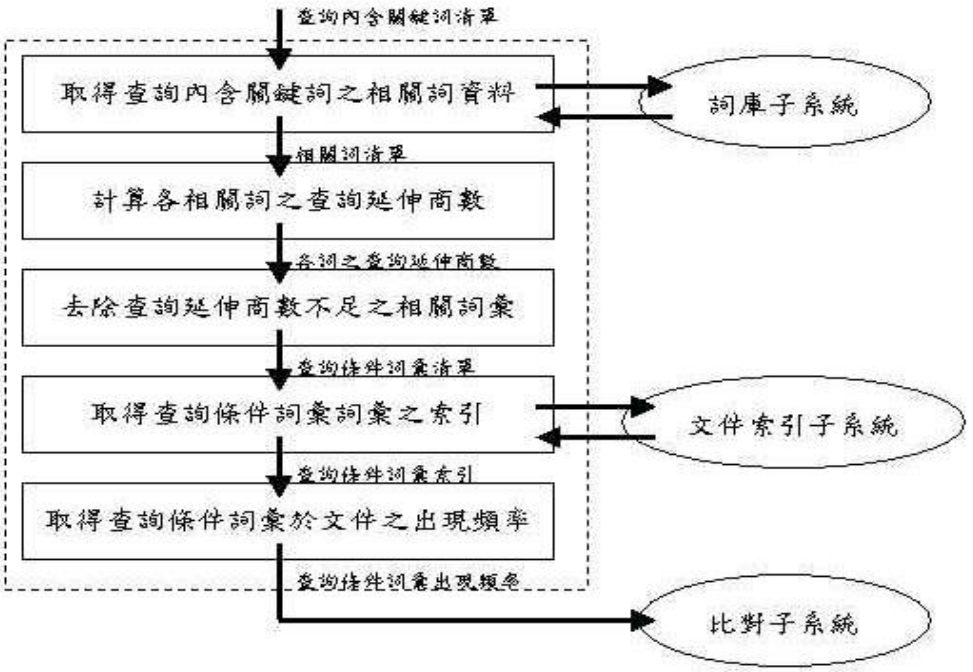


圖 5.6 自然語言分析模組

（2）文件分析模組：

若使用者選擇以全文分析模組，則須先指定一標的文件；系統會先在關鍵詞索引資料庫中，取得該文件所屬關鍵詞的資料，並向詞庫系統取得相關辦文件內未出現的詞彙資料。由於全文檢索是以詞義鏈代替詞彙，作為判別相似性的單位，故必須先找出該文件所有的詞義鏈，並進行詞義鏈串聯的工作，包括原本存在於文件內以及鏈結平均相關度達標準以上的詞彙，皆會依詞義相互鏈結。在確定各詞彙所屬詞義鏈之後，索引資料庫亦會將詞義鏈中的詞彙索引資料，傳給比對系統進行查詢比對。文件分析查詢模組架構如圖 5.7：

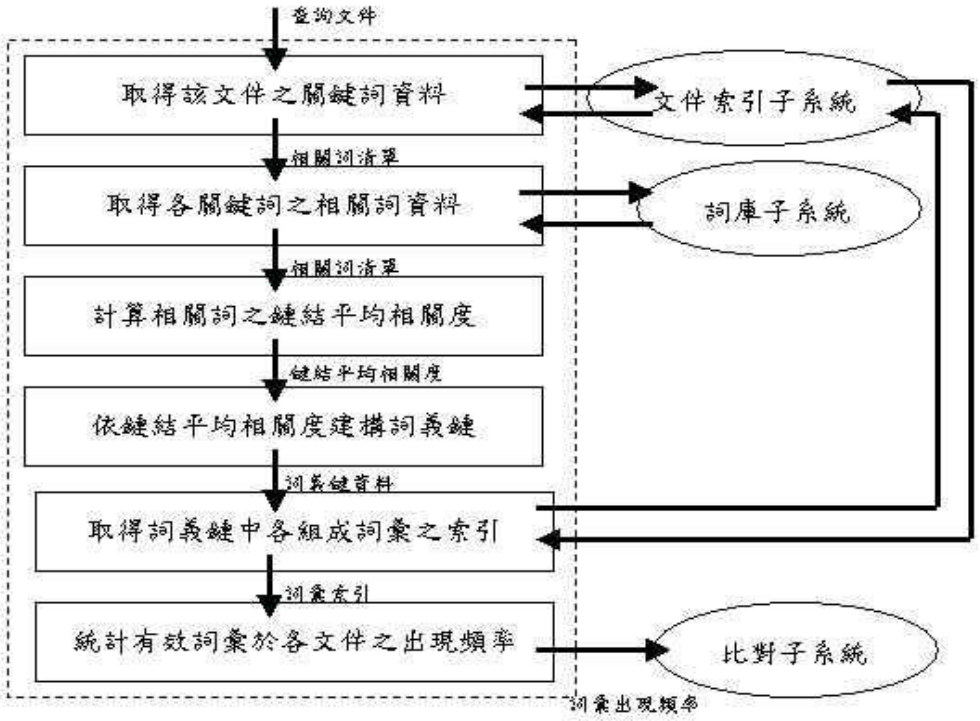


圖 5.7 文件分析模組

5.5 比對子系統

比對子系統負責接收來自查詢處理子系統的查詢資訊，包括文件內含詞彙以及透過同義詞詞庫所衍生的相關詞彙，或是詞義鏈的出現頻率，經由相似性比對的演算，找出最符合查詢條件的文件，並依相似程度排序。

在本子系統中，我們依照第四章所介紹的內容計算文件之間的相似性，如圖 5.8 所示，本子系統在接收來自查詢處理子系統的資料後，會由比對模組加以分析比對，得到的結果再回傳給查詢處理子系統以回報給使用者。

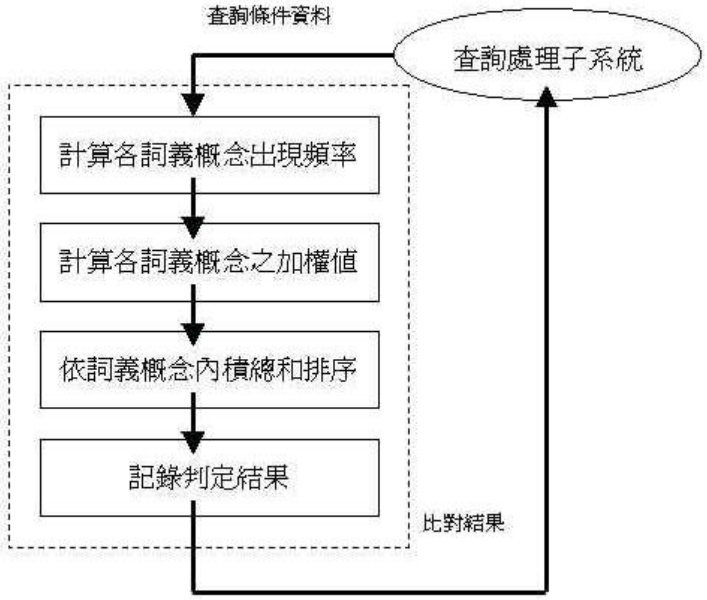


圖 5.8 比對子系統

5.6 系統實驗設定

本研究中，我們蒐集了聯合新聞網與中時電子報四、五月份的體育新聞共一千篇，同時，我們選定「籃球」與「NBA」的字詞建構同義詞詞庫（見附錄一），並將其加入實驗中，以未使用詞義概念網路的比對結果為對照組，使用詞義概念網路的比對結果為實驗組，分別將相似性比較結果排名在前的文件資料取出，觀察使用詞義概念網路及詞義鏈對文件檢索招回度及精確度的影響。實驗取樣方式如下：

1. **由文件集合中任意選取一符合「籃球」與「NBA」主題的標的文件（**Target document**）** d **。**
2. **由文件集合中，隨機取出** 100 **篇成一測試文件集合（**testing document set**）** D **，與** d **作相似性比對，並依相似性將結果排序。**
3. **將排序的結果分別取前** 5**、**10**、**15**、**20**、**25**、及** 30 **篇，觀察其與** d **之主題是否吻合，同時標記** D **中吻合** d **主題的文件數量。**
4. **檢視** D **，假設所有與** d **相似的文件數量為** C_{all} **，取樣中與** d **相似的文件數量為** C_{retr} **，則：**

$$\text{Re}\,call_d = \frac{C_{retr}}{C_{all}}$$

5. 取樣中被系統判定與 d 相似的文件數量為 C_{ret}，則：

$$\text{Pr}\,ecision_d = \frac{C_{retr}}{C_{ret}}, \quad C_{ret} \geq C_{retr}$$

由於判定文件是否相似的工作是由人工進行，太過於細分文件內涵的概念，結果往往會因人為的主觀而有所差異，為避免此情況發生，判定文件之間的相似性依據時，主要是以檢視文件中的概念是否包含「籃球」與「NBA」兩主題為準則。

本研究將以向量空間法，分別就使用詞義概念網路與否進行交叉比對，並分析其數據結果。詳細實驗規格如下：

- 查詢延伸商數 E_q 中作為調整參數的 t 值設為 1，臨界值 e 設為 0.5。
- 詞義鏈中，串聯標準 e 設為 0.5，串聯有效階層距離 ld 設為 3。
- 使用平台為 P4 – 1.8G，512mb ram，Windows 2000 Professional Service Pack 3。

5.7 實驗結果

首先針對同一文件 d 進行相似性比對，實驗一之結果如下列所示：

表 3　實驗 1-1 不使用詞義概念網路及詞義鏈之實驗結果

總文件數目	取樣數目	Recall（%）	Precision（%）	實際符合主題文數目
100	5	26.6（4/15）	80（4/5）	15
100	10	60（9/15）	90（9/10）	15
100	15	80（12/15）	80（12/15）	15
100	20	86.6（13/15）	65（13/20）	15
100	25	86.6（13/15）	52（13/25）	15
100	30	93.3（14/15）	46.7（14/30）	15

表 4　實驗 1-2 使用詞義概念網路及詞義鏈之實驗結果

總文件數目	取樣數目	Recall（%）	Precision（%）	實際符合主題文數目
100	5	33.3（5/15）	100（5/5）	15

100	10	66.7（10/15）	100（10/10）	15
100	15	100（15/15）	100（15/15）	15
100	20	100（15/15）	75（15/20）	15
100	25	100（13/15）	60（15/25）	15
100	30	100（14/15）	50（15/30）	15

將上述資訊繪製成折線圖如圖 5.9 **所示：**

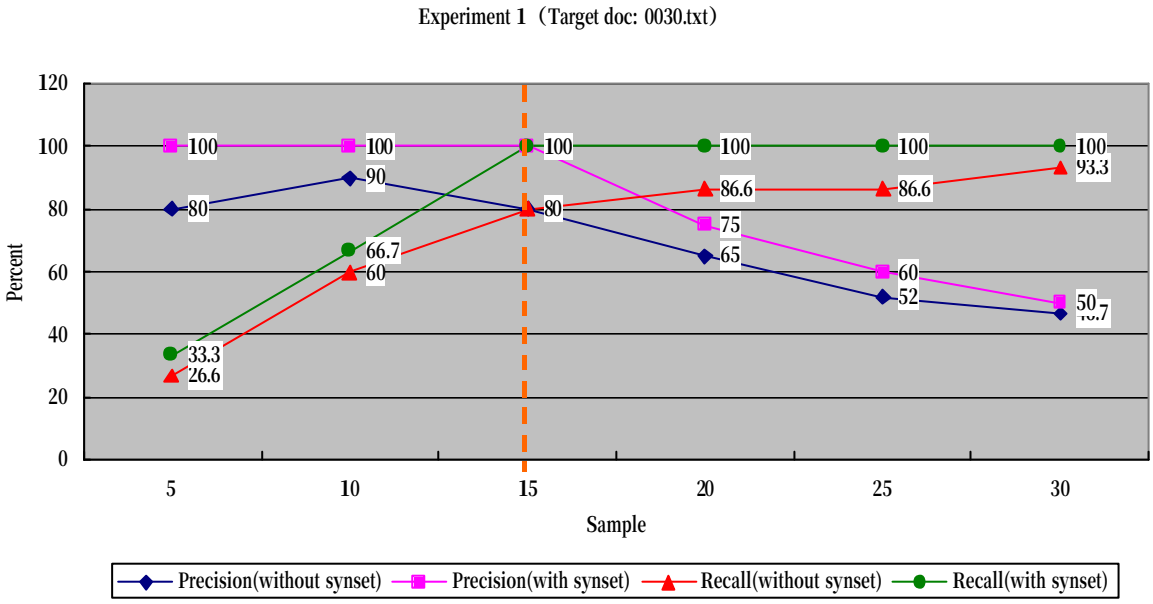


圖 5.9 **實驗一之** Precision **及** Recall

由實驗數據可知，在標的文件 *d* **與實驗文件集合** *D* **皆相同的情況下，使用詞義概念網路以及詞義鏈進行檢索，與不使用詞義概念網路以及詞義鏈，單以向量空間模組進行檢索，無論取樣數目為何，在招回度（**recall**）及精準度（**precision**）上皆有一定程度的提昇，若我們重複數次實驗，將文件集合** *D* **中所含真正符合** *d* **主題的文件數目作為取樣數目，即** recall = precision **的情形，則可歸納成下表。**

表 5 六次實驗在 Precision=Recall **的情況下之結果**

實驗編號	*D* 文件數目	取樣數目	Recall（without synset）	Recall（with synset）	Precision（without synset）	Precision（with synset）	實際符合文件數目
1	100	15	80（12/15）	100（15/15）	80（12/15）	100（15/15）	15

2	100	18	77.8（14/18）	100（18/18）	77.8（14/18）	100（18/18）	18
3	100	21	85.7（18/21）	95.2（20/21）	85.7（18/21）	95.2（20/21）	21
4	100	36	86.1（31/36）	97.2（35/36）	86.1（31/36）	97.2（35/36）	36
5	100	26	69.2（18/26）	84.6（22/26）	69.2（18/26）	84.6（22/26）	26
6	100	18	72.2（13/18）	88.9（16/18）	72.2（13/18）	88.9（16/18）	18

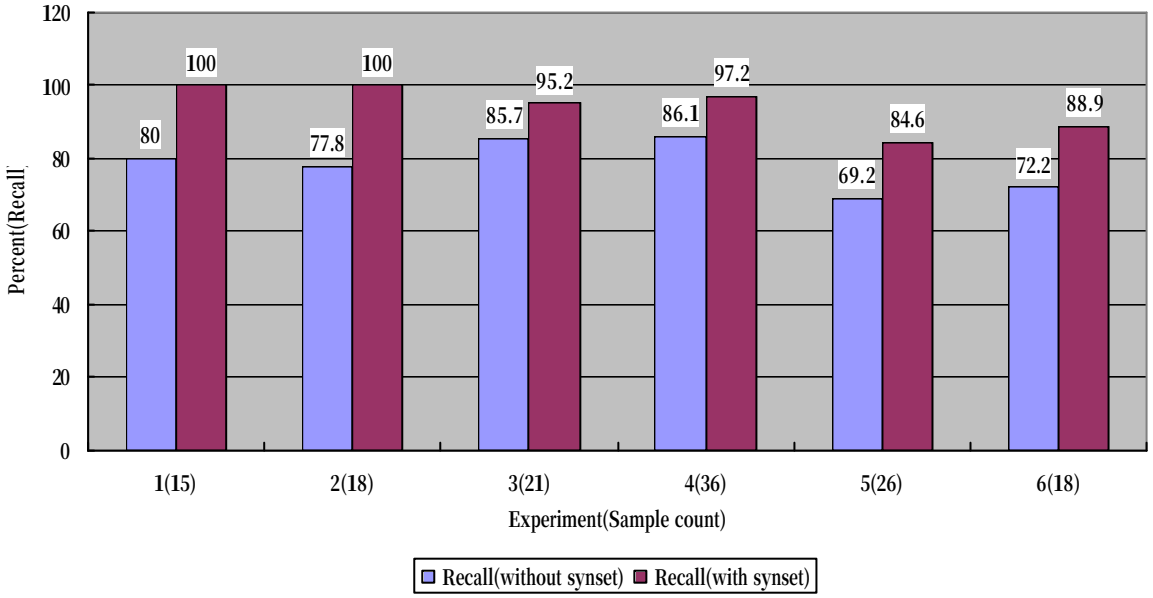


圖 5.10 六次實驗在 Precision=Recall 的情況下之招回度比較

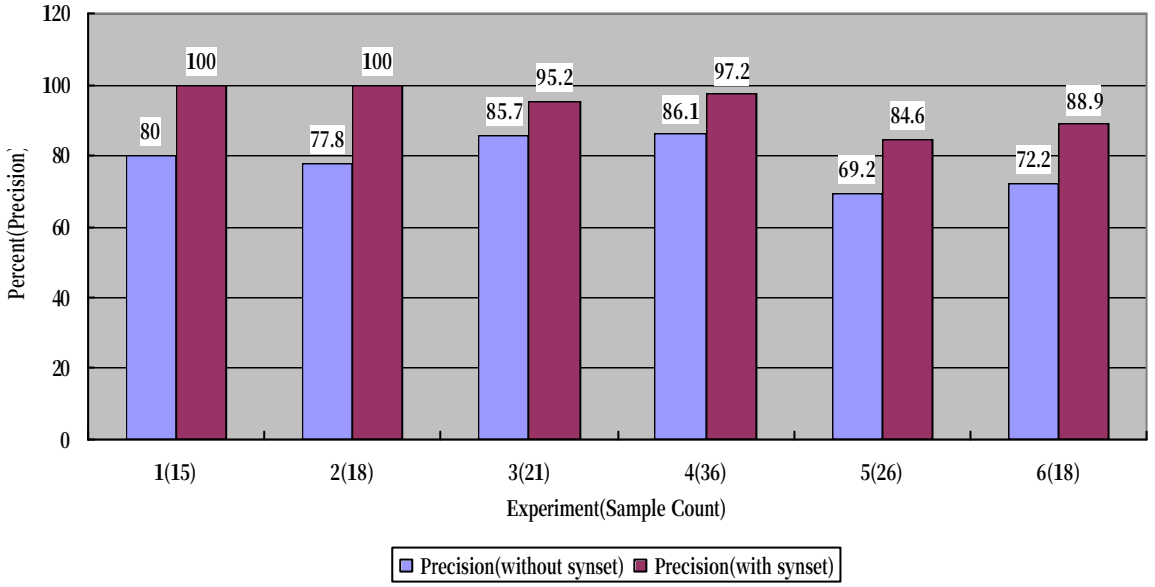


圖 5.11 六次實驗在 Precision=Recall 的情況下之精確度比較

另外，我們又選定幾篇其他關於籃球及 NBA 主題的文件作為標的文件，以下是實驗的結果：

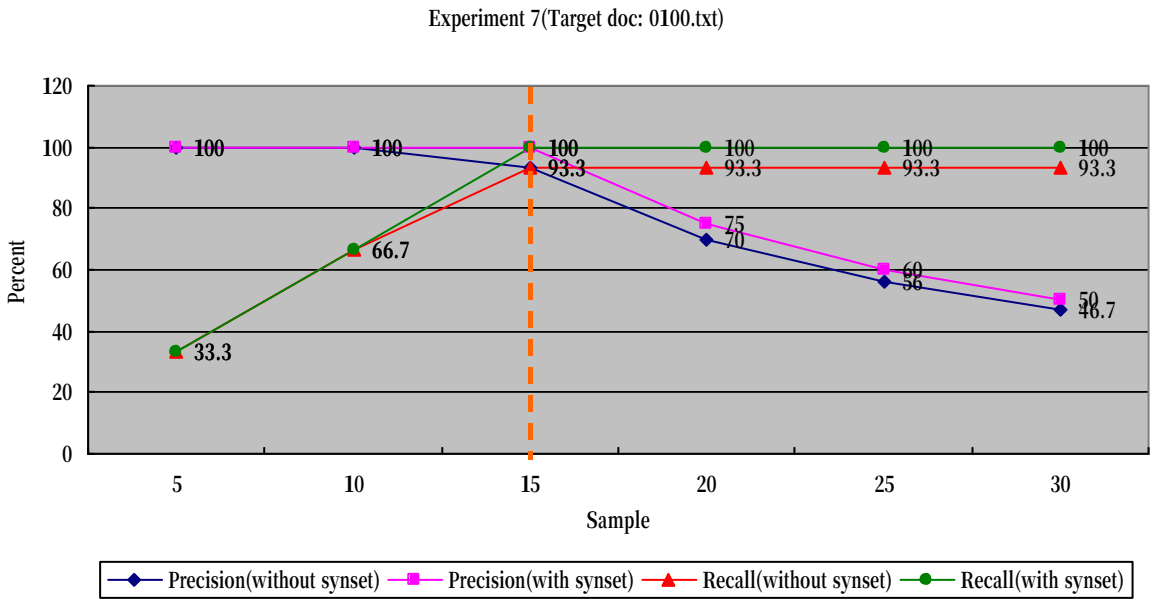


圖 5.12 以不同文件作為標的文件的 Precision 及 Recall（標的文件：0100.txt）

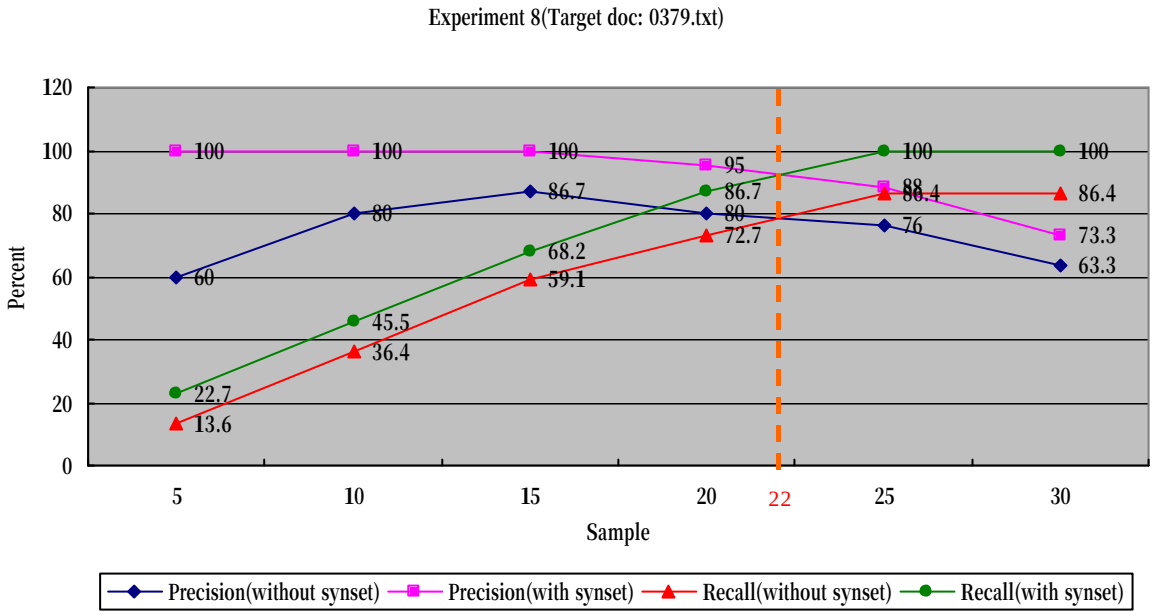


圖 5.13 以不同文件作為標的文件的 Precision 及 Recall（標的文件：0379.txt）

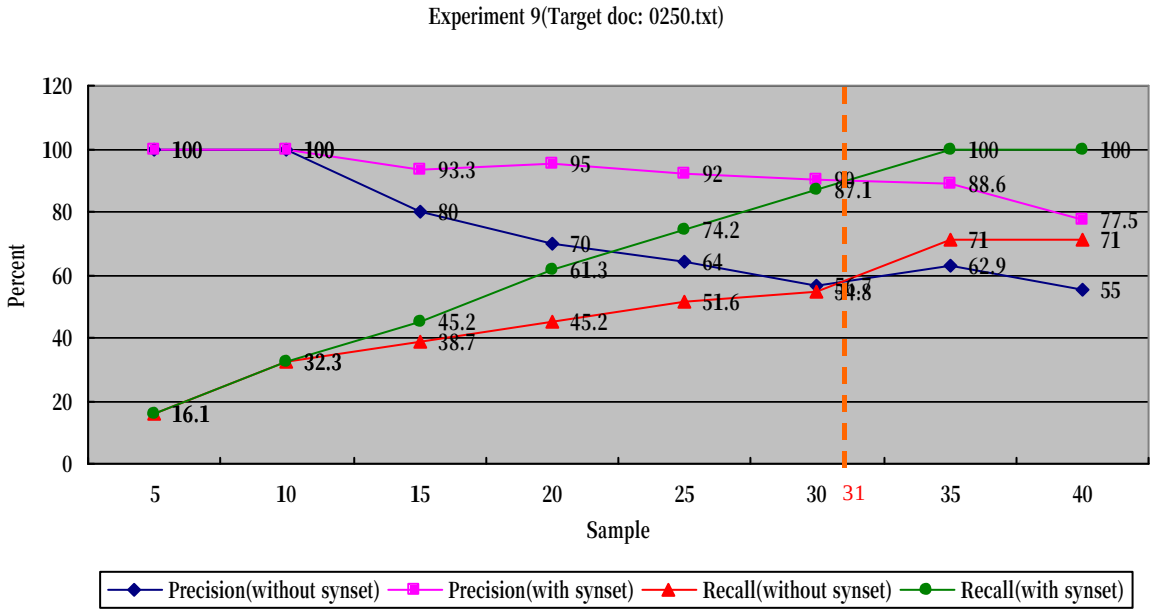


圖 5.14 以不同文件作為標的文件的 Precision 及 Recall（標的文件：0250.txt）

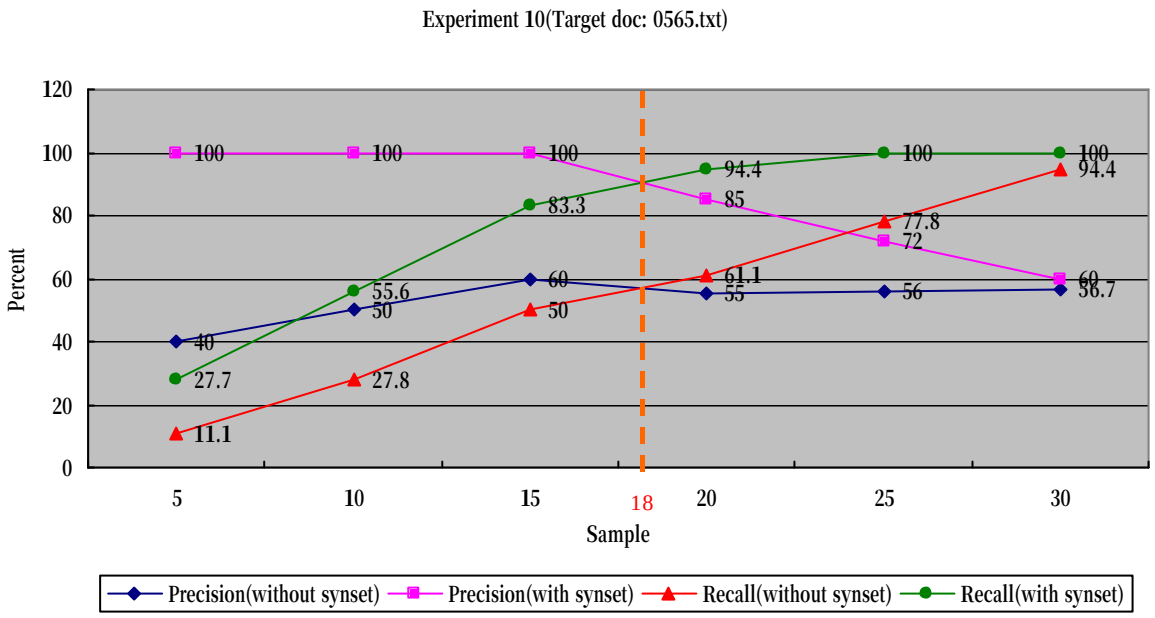


圖 5.15 以不同文件作為標的文件的 Precision 及 Recall（標的文件：0565.txt）

以上均以 100 篇文件為檢索範圍的實驗數據中，不管標的文件為何，只要該文件中存在有關於籃球以及 NBA 的概念，使用詞義概念網路及詞義鏈的正確率皆比未使用同義詞詞庫及詞義鏈時要來的優越。我們若將測試文件集合中的文件數量提昇至 200、300、400、500 篇，重複以上的實驗過程所得到的實驗結果如下：

表 6 較大範圍檢索實驗結果

實驗編號	*D* 文件數目	取樣數目	Recall（without synset）	Recall（with synset）	Precision（without synset）	Precision（with synset）	找出符合（without synset）	找出符合（with synset）
1	200	59	79.7（47/59）	91.5（54/59）	79.7（47/59）	91.5（54/59）	47	54
2	300	70	57.1（40/70）	90（63/70）	57.1（40/70）	90（63/70）	40	63
3	400	101	84.1（85/101）	91.1（92/101）	84.1（85/101）	91.1（92/101）	85	92
4	500	129	72.1（93/129）	91.5（118/129）	72.1（93/129）	91.5（118/129）	93	118

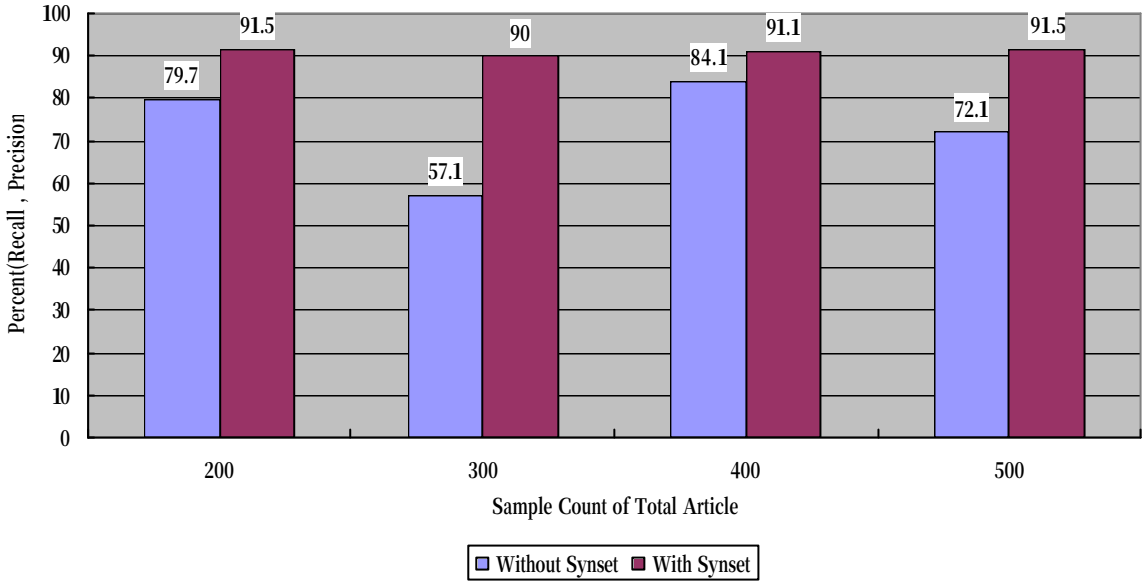


圖 5.16 較大範圍檢索實驗結果比較（長條圖）

下圖為測試文件集合文件數量所需處理時間的比較圖。

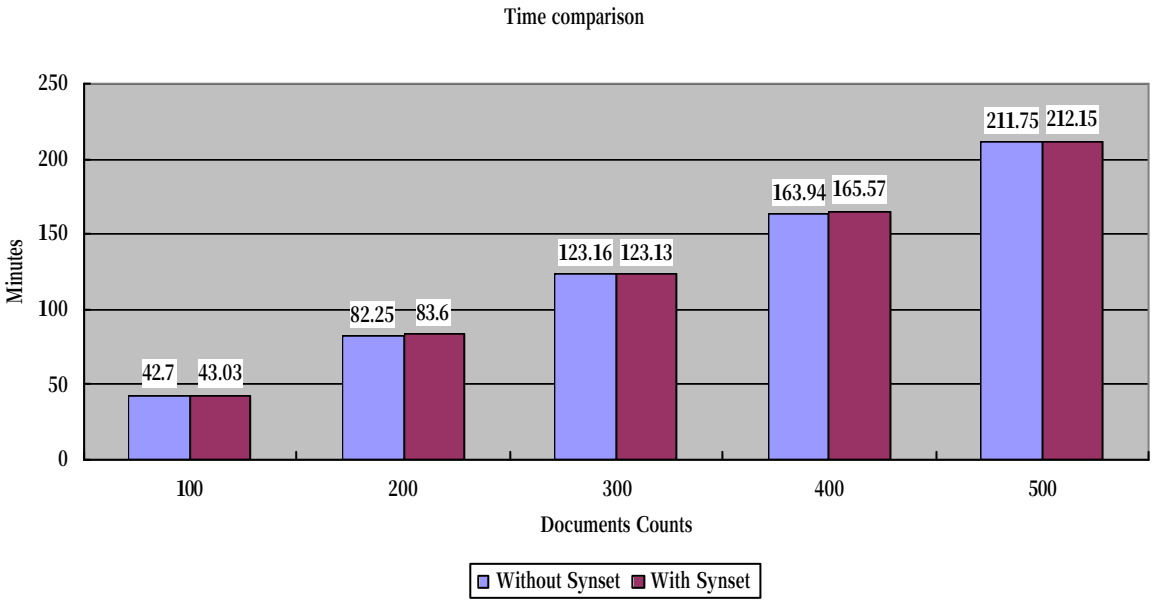


圖 5.17 測試文件集合文件數量所需處理時間的比較

為驗證本系統在不同領域的文件檢索仍保持其正確性，我們另外蒐集了中時電子報財經股市版的文件兩百篇，並以台灣證卷交易所公佈歸類於電子工業的上市公司名稱共

253 家，將其建立成詞義概念網路，並依前述實驗的規格進行相似性比對，所得結果如下：

表 7　不使用詞義概念網路及詞義鏈於財經股市新聞領域之實驗結果

總文件數目	取樣數目	Recall（%）	Precision（%）	實際符合主題文數目
100	5	12.1（4/33）	80（4/5）	33
100	10	12.1（4/33）	40（4/10）	33
100	15	12.1（4/33）	26.7（4/15）	33
100	20	12.1（4/33）	20（4/20）	33
100	25	12.1（4/33）	16（4/25）	33
100	30	12.1（4/33）	13.3（4/30）	33

表 8　使用詞義概念網路及詞義鏈於財經股市新聞領域之實驗結果

總文件數目	取樣數目	Recall（%）	Precision（%）	實際符合主題文數目
100	5	15.2（5/33）	100（5/5）	33
100	10	30.3（10/33）	100（10/10）	33
100	15	42.4（14/33）	93.3（14/15）	33
100	20	51.5（17/33）	85（17/20）	33
100	25	60.6（20/33）	80（20/25）	33
100	30	69.7（23/33）	73.3（23/30）	33

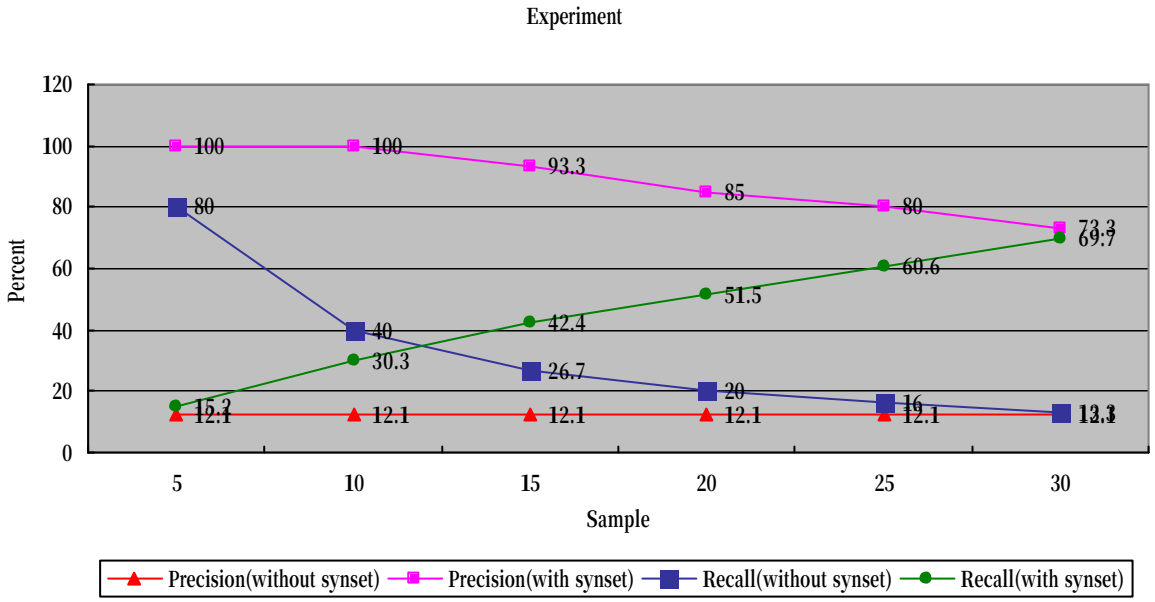


圖 5.18 使用詞義概念網路及詞義鏈與否於財經股市新聞領域之實驗結果比較

由以上實驗結果中可知，在測試文件集合文件數量增加的情況下，使用同義詞詞庫及詞義鏈進行檢索，不管招回度以及精確度仍舊能有一定程度的提昇，而且其正確率表現十分的平穩，約略在百分之九十上下。值得注意的是，在實驗過程中，使用同義詞詞庫及詞義鏈與否對時間的使用上並無太大差別（時間複雜度皆為 $O(n^2)$），圖 5.17 顯示，系統比對時間的花費，大致與測試文件集合文件數量成正比，而使用詞義概念網路以及詞義鏈與否，在處理時間上並無顯著的差距。另外，在圖 5.18 的實驗中，由於文件內容大都是集中在特定公司的盤勢分析或市場分析，採用關鍵詞彙進行檢索不易找出類似領域公司的資料，因此，精確度以及招回度表現皆不理想，若是採用詞義概念網路及詞義鏈，則不會出現這樣的問題，當然，若是詞義概念網路的內容更為豐富，成果應該更加理想。

第 6 章　結論與未來發展

整體來說，實驗結果與預期相近，以詞義概念為條件的方式進行文件檢索，在其餘條件皆相同的情況下，比起向量空間模組所得到的結果，在招回度以及精確度上都有顯著的改善，在處理時間的花費上則幾乎完全相同。證明了以詞義概念為基礎的檢索法，應用於全文檢索的比對上，確實可有效的強調文件內涵的關鍵概念，以提昇檢索的正確性。然而這樣的結果，還是有幾點值得探討的地方，以下列舉之：

1. 以本研究所提出的方式進行檢索，固然在檢索結果的準確性上有著一定程度的提昇，但事實上，若想要將準確率提升到令人滿意的境界，所需的並不只是資料處理前端眼演算模組的改良，更重要的是後端資料是否對各領域概念知識的涵蓋範圍及正確率已達一定水準。唯有在後端資料完備的情形之下，整個檢索模組才能發揮出最大的作用。在蒐集後端資料這方面，確實還需要更多的努力。
2. 在定義同義詞之間的關係以及相關程度時，太容易參雜過多的主觀意識，這對於針對一般人使用的文件檢索模組正確率會有負面的影響。因此，在制訂同義詞集合時，我們應以更公正客觀的方式，針對各個概念找出一般人皆能接受的定義。
3. 同義詞集合之間關係種類的訂定之根據為何，則是另一個需要討論的問題。兩概念是根據何種原因判定為同反義、廣狹義，整體細部，目前看來並無一統一的定義方式。是否應將所有的概念，如同知網（hownet）一般，將其概念內容切分成不同的子概念，再依據切分出的各個子概念相互比較後，定義其父概念之間的關係。舉例來說，若把「好吃」一詞的概念切分為「味道」「好、佳」兩子概念，則與「好吃」相關的其他概念其子概念便必須符合上述格式，假設「難吃」的概念可切分為「味道」「差、劣」兩子概念，由於「好、佳」以及「差、劣」互為反義關係，則「難吃」可定義為「好吃」的反義詞。若能以這樣的方式訂立概念之間的關係，相信對於語意分析的精確度以及定義上會有相當大的進展。
4. 本研究實作所所定的領域是集中在籃球以及 NBA 範圍的文件，藉由詞義鏈及

同義詞詞庫的作用可有效的強調文件中符合該領域的概念，若是拓展其領域，亦或是增加不同領域的概念，只要加入該領域的相關資料（同義詞集合）即可。惟增加檢索領域時，各領域的相關資料必須做更清楚的界定，以避免領域間概念的相互混淆。

此外，考慮到文件內各章節的概念之間可能存在有概念推導的關係，也就是說，文件中較位置較前的章節點出的概念，會由後續的章節詳細描述的情形。若能將各章節的概念依其因果關係串聯成一個概念流程（concept flow），則概念流程亦可作為文件檢索時各文件的特徵，以進行模擬人類思考路線為基礎的文件檢索。甚而組合各個概念流程，以建構出該文件的摘要，這也是未來可以努力的研究方向。

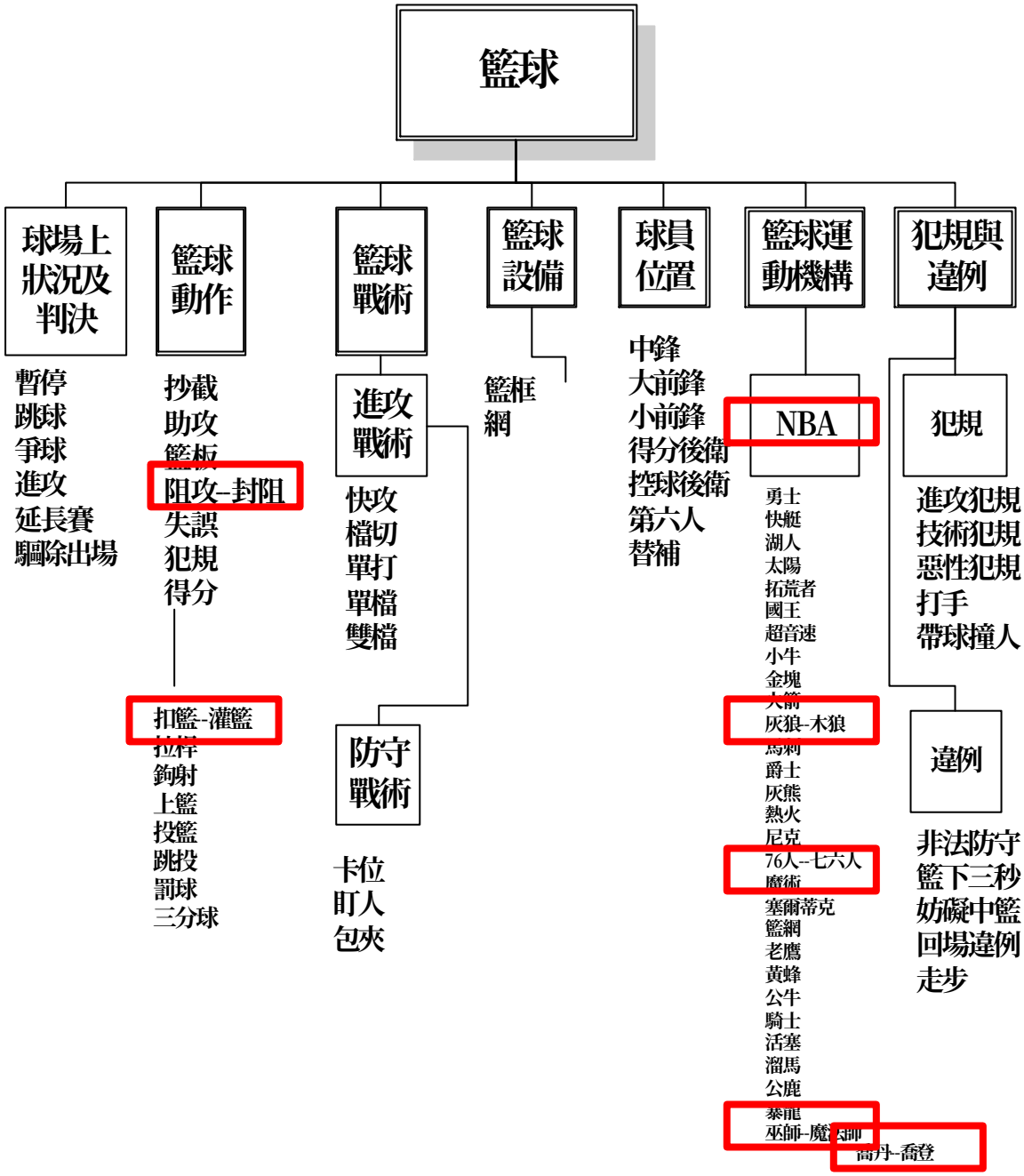
Reference

1. 劉玉琛，標點符號用法，66 年 4 月，國語日報出版社
2. C. H. Chi、C. Ding and Andrew Lim, Word segmentation and recognition for web document framework, ACM CIKM'99，November 1999 Page(s):458-465.
3. Sproat R and Shih C, A statistical method for finding word boundaries in Chinese text, Computer proceeding of Chinese and oriental language, 1990.
4. www.google.com/technology/
5. www.openfind.com.tw
6. 中文詞類分析，中華民國計算語言學學會。
7. Christiane Fellbaum, Wordnet, MIT Press, 1999.
8. Shyi-Ming Chen and Jeng-Yih Wang, Document retrieval using knowledge-based fuzzy information retrieval techniques, Systems, Man and Cybernetics, IEEE Transactions on , Volume: 25 Issue: 5 , May 1995 , Page(s):793-803.
9. Shyi-Ming Chen、Yih-Jen Horng and Chia-Hoang Lee, Document retrieval using fuzzy-valued concept networks, Systems, Man and Cybernetics, Part B, IEEE Transactions on , Volume: 31 Issue: 1 , Feb 2001 , Page(s):111-118.
10. 1 *Jiang, J.; Conrath, D.;* Multi-word complex concept retrieval via lexical semantic similarity, Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on , 31 Oct.-3 Nov. 1999 , Page(s):407-414.
11. *Song, W.W.; Cheung, D.; Tan, C.J.;* A semantic similarity approach to electronic document modeling and integration, Web Information Systems Engineering, 2000. Proceedings of the First International Conference on , Volume: 1 , June 2000 , Page(s):116-124 vol.1.
12. Shyi-Ming Chen ＆ Yih-Jen Horng；Fuzzy query processing for document retrieval based on extended fuzzy concept networks, Systems, Man and Cybernetics, Part B, IEEE Transactions on , Volume: 29 Issue: 1 , Feb.

1999，Page(s):96-104.
13. 重定標點符號手冊，教育部國語推行委員會，1987
14. 「搜」文解字－中文詞界研究與資訊用分詞標準，中華民國計算語言學學會。
15. http://godel.iis.sinica.edu.tw/ROCLING/project.htm
16. Ricardo Baeza-Yates, Berthier Ribeiro-Neto Modern information retrieval, Addison Wesley, 1999.
17. A.K. Jain, M.N. Murty, and P.J. Flynn；Data clustering：a review；ACM Computing surveys Vol.31 No.3 September, 1999.
18. A.K. Jain, R.C. Dubes;Algorithms for clustering data;Prentice-Hill, 1988.
19. Anton Leuski, Evaluating document clustering for interactive information retrieval, ACM CIKM'01, November 2001. Page(s):33-40.
20. J. McQreen；Some method for classification and analysis of multivariate observation；Proceeding of the fifth Berkeley symposium on mathematical statistics and probability. 1967，Page(s):281-297.
21. J.C. Bezdek；Pattern recognition with fuzzy objective function algorithm；Plemun Press, New York, 1981.
22. G. Salton and M.J. Mcgill. Introduction to modern informationretrieval. McGraw-Hill Books Co., 1983.
23. J. Morris and G. Hirst, Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 17，1991. Page(s):21-48.
24. *Green, S.J.;* Building hypertext links by computing semantic similarity, Knowledge and Data Engineering, IEEE Transactions on，Volume: 11 Issue: 5，Sept.-Oct. 1999，Page(s):713-730.
25. S. K. M. Wang、W. Ziarko、and P. C. N. Wong, Generalized vector space model in information retrieval. In Proc. 8th ACM SIGIR Conference on Research and Development in information retrieval, 1985，New York, USA，Page(s):18-25.
26. J. J. Rocchio, Relevance feedback in information retrieval. In G. Salton, editor, The smart retrieval system – experiments in automatic document processing. Prentice Hall inc., 1971
27. E. Ide. New experiments in relevance feedback. In In G. Salton, editor, The smart retrieval system, 1971.
28. G. Salton and M.J. Mcgill. Introduction to modern information retrieval.

McGraw-Hill Books Co., 1983.

附錄一　同義詞詞庫之內容（標記的部分為同義詞集合）



附錄二　檢索標的文件

NBA 季後賽 如果少了喬丹、姚明？
【記者黃顯祐】

東區的公鹿與巫師，西區的太陽與火箭，那一隊能搭上分區季後賽末班車，將是 NBA 本季例行賽最後半個月的焦點之一。

東、西區「老八」之爭，對 NBA 球迷而言是抱著迎新送舊的情懷，因為巫師「老飛人」喬丹的最後一季如果沒有飛進季後賽，喬丹迷難免惋惜，而火箭大陸中鋒姚明無法助隊打季後賽，喜愛姚明的球迷也會悵然若失。

落後公鹿一場勝差的巫師，如果被摒除在季後賽之外，代表著喬丹的球員生涯只剩九場比賽，四十歲的他似乎百般不願以此再度高掛球鞋，但他率領巫師奮戰，顯示寶刀未老，也讓球迷印象深刻。

公鹿也有非得進季後賽的壓力，特別是季中送走了射手艾倫，使原本的「鐵三角」徹底瓦解，換來超音速隊後衛裴頓與卡塞爾並肩作戰，如果無法闖進季後賽，公鹿全隊一定會大震盪。

火箭本季成為客場最受歡迎的球隊之一，主要是衝著姚明而來，美國球迷的好奇，加上華人社會的支持，使姚明的 NBA「菜鳥季」顯得多采多姿，但他的壓力也接踵而至，上周三場平均只上場廿點三分鐘，得六點三分、五個籃板，命中率只有兩成三。

雖然姚明昨天打出近一個月的代表作，但球隊還是輸球，最後八場例行賽必須全力以赴，否則前途不樂觀，即使火箭闖進季後賽可能在首輪就打包，但少了姚明的季後賽，將使大陸「移動長城」只有巴特爾的馬刺隊在季後賽力爭上游，而巴特爾又鮮少上場，華人市場將受影響。

附錄三　實驗一之文件正確率分布

Ranking	Without synset	With synset
1	0030.txt	0079.txt
2	0054.txt	0054.txt
3	0082.txt	0053.txt
4	0048.txt	0030.txt
5	0034.txt	0009.txt
6	0009.txt	0031.txt
7	0079.txt	0010.txt
8	0031.txt	0097.txt
9	0010.txt	0068.txt
10	0047.txt	0047.txt
11	0053.txt	0048.txt
12	0038.txt	0013.txt
13	0068.txt	0029.txt
14	0011.txt	0100.txt
15	0013.txt	0034.txt
16	0029.txt	0082.txt
17	0039.txt	0052.txt
18	0024.txt	0025.txt
19	0025.txt	0026.txt
20	0026.txt	0058.txt
21	0094.txt	0089.txt
22	0095.txt	0024.txt
23	0099.txt	0094.txt
24	0041.txt	0095.txt
25	0052.txt	0099.txt
26	0058.txt	0041.txt
27	0059.txt	0059.txt
28	0089.txt	0016.txt
29	0002.txt	0061txt
30	0097.txt	0027.txt
31	0100.txt	0030.txt

共 100 篇文章，其中有 15 篇符合查詢條件，加網底字為正確結果，圖中是取相似度前 31 名的結果