陳林志、陳大仁、葉國暉、吳忠澄（2015），『使用語意模型分析線上部落格文件』，*中華民國資訊管理學報*，第二十二卷，第三期，頁 273-316。

使用語意模型分析線上部落格文件

陳林志*
國立東華大學資訊管理學系

陳大仁
國立臺中科技大學資訊管理系

葉國暉
國立東華大學資訊管理學系

吳忠澄
國立東華大學資訊管理學系

摘要

近年來線上部落格成長的速度如同其它社群網站一樣迅速。一般而言，我們使用不同部落格搜尋引擎（例如：Technorati、Blogpulse、Google Blog Search）搜尋那些我們最感興趣之部落格貼文；一般而言，當我們從部落格搜尋引擎進行搜尋時，很可能會面臨到同義詞（兩個字詞形態不同，但語意相同）及一詞多義（一個字詞有不同的意義）問題。在本論文裡，我們使用兩個語意分析模型：潛在語意分析（LSA）及機率潛在語意分析（PLSA），去解決上述兩個問題。LSA 使用奇異值分解（SVD）技術，去擷取字詞間存在之同義詞關係；PLSA 則可解決一詞多義並明確區分字詞間的不同含意和不同用法。根據模擬的結果，我們認為語意分析模型可增進部落格搜尋引擎的效能。

關鍵詞：語意分析、部落格搜尋、潛在語意分析、機率潛在語意分析、奇異值分解

* 本文通訊作者。電子郵件信箱：lcchen@mail.ndhu.edu.tw
2014/03/13 投稿；2014/09/16 修訂；2014/11/17 接受

Chen, L.C., Chen, D.R., Yeh K.H.and Wu, C.C. (2015), 'Using the Semantic Models to Analyze the Online Blog Posts', *Journal of Information Management*, Vol. 22, No. 3, pp. 273-316.

Using the Semantic Models to Analyze the Online Blog Posts

Lin-Chih Chen*
Department of Information Management, National Dong Hwa University

Da-Ren Chen
Department of Information Management,National Taichung University of Science and Technology

Kuo-Hui Yeh
Department of Information Management, National Dong Hwa University

Chung-Cheng Wu
Department of Information Management, National Dong Hwa University

Abstract

Purpose－In recent years, the online blogging community is growing bigger as the social network service. Generally, we have used various blog search engines, such as Technorati, Blogpulse, and Google Blog Search, to find the blog post most appropriate for what we are seeking.

Design/methodology/approach－In this paper, we use two semantic analysis models, Latent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analysis (PLSA), to deal with these two problems.

Findings－According to the results of simulation analysis, we conclude that the semantic analysis models can effectively be applied to the blog search engine.

Research limitations/implications－We have encountered synonym (two terms

* Corresponding author. Email: lcchen@mail.ndhu.edu.tw
2014/03/13 received; 2014/09/16 revised; 2014/11/17 accepted

are syntactically different but semantically interchangeable expressions) and polysemy (a term has different meanings) problems when we search from the blog search engine.

Practical implications－LSA uses a Singular Value Decomposition (SVD) technique to capture the synonym relationships between terms. PLSA can deal with the problem of polysemy and can explicitly distinguish between different meanings and different types of term usage.

Originality/value－We claim that the semantic analysis models can effectively improve the performance of blog search engine.

Keywords: semantic analysis, blog search, latent semantic analysis, probabilistic latent semantic analysis, singular value decomposition.

壹、導論

隨著科技的日新月異，自從美國國家科學基金會（NSF）在 1995 年開放網際網路（internet）供商業使用後，網際網路的發展變得相當迅速，至今已趨成熟。現今網際網路所提供的服務已經不勝枚舉，人們可以透過網際網路互相溝通、交換訊息。

早期的網際網路稱為 Web 1.0，僅實現了單向的資訊傳播，使用者可從傳統的網頁取得所想要的資訊，直到 DiNucci（1999）使用 Web 2.0 這個詞彙，Web 2.0 的概念才逐漸被提出來，直至 2004 年，Web 2.0 才真正實現。Web 2.0 是網際網路運用的新時代，網際網路成為新的分享平台，網頁內容經由每位使用者的參與，進而產生個人化內容，並允許人際間交流及互動。

Web 2.0 對電腦工業來說是一種商業革命，起因於人們開始把網際網路當成交易平台，並企圖在此平台上建立通往成功的準則（O'Reilly 2005）。IBM 的社群網路分析師 Judicibus（2008），提出了另一個 Web 2.0 的定義，主要著重在社群互動和架構的建立，他認為 Web 2.0 是一個架構在知識分享的環境之上，透過人際互動所產生出的內容，並經由服務導向架構中的程式，在這個環境被發佈、管理和使用。

Web 2.0 在近幾年來透過網際網路應用，進而促進網際網路上人際間之資訊交換和協同合作，其模式更加以使用者為中心。典型的網際網路應用包含有：網路社群、網路應用程式、社群網站、部落格、維基百科等等。

然而 Web 2.0 最廣為人知的應用便是部落格（Blog），而部落格的原文是 Weblog，或縮寫成 Blog。Weblog 這個字眼最早是由 Jon Barger 於 1997 年左右提出（Wikipedia 2013），其原始意思是一種以 Web 作為呈現媒介的（個人）log，相當於電子佈告欄系統（bulletin board system, BBS）上的個人版。到了 1999 年左右 Peter Merholz 開始把 Weblog 縮寫成 Blog（Merholz. 1999）。部落格又稱為網路日誌，是一種經由個人管理，不定期發表新的文章、圖片、或影片的網頁，亦即記錄使用者的生活（Nardi et al. 2004）。

線上部落格文件係由部落客（blogger）藉由部落格服務所發表的文章，這些發表的文章可能具有相近的主題、或由同一人或同一群人所撰寫，傳統的部落格文件內容通常充斥著超連結，但現今的部落格文章以描述生活記事、個人評論為主（Jeong & Oh 2012）。

部落格上的文章通常根據時間由新到舊排序，使用者可按照自己的喜好為所發表的文章做分類及標籤，而每一篇文章都可以和來自世界各地的網友進行互動。部落格的類別可以由下列兩種方式進行區分：(1)作者身分、(2)主題類別，見表 1。

表 1：部落格類別的區分方式

類別	概述
以作者身分區分	個人部落格：是個人持續性的日記或評論，也是最常見的部落格，知名部落客的部落格可以吸引很多讀者引起人們的共鳴。 公司部落格：部落格大部分可作為私人用途，亦可作為商業用途，用於公司內部，可以增進聯繫，或者對外作為行銷、品牌推廣、公共關連等目的，稱為公司部落格。
以部落格主題區分	一些部落格集中於某一個特定的課題，如政治部落格、視訊部落格（podcast）、圖片部落格、時尚部落格等等。

資料來源：本研究

部落格發展至今影響了世界，儘管近年來社群網站（social network service）的崛起，部落格的服務依然有它的優勢，編寫部落格有許多優點，例如：記錄生活點滴、傳達個人思想、抒發個人情緒、交流專業知識、企業宣傳文化等等。在多家部落格業者提供的服務之下，世界上已經有太多的部落格，於是衍生出部落格搜尋引擎之服務，例如 Google Blog Search（2014）、Sysomos（2014）、Technorati（2014）等等。

由於部落格搜尋引擎是近幾年才發展的，技術尚未成熟，另外部落格的文件型態又跟一般網頁文件性質有一定的差異，其主要原因在於部落格文件是由網路使用者所自行輸入，難免會有許多隱含的資訊無法被搜尋引擎擷取到，也因為資訊過載（information overload），使用者要找到想要尋找的部落格文件越加困難。因此，我們需要一套機制輔助使用者去尋找到他真正想要的部落格文件，節省使用者在部落格搜尋上所花費的搜尋成本、並解決使用者資訊過載的問題。

利用搜尋引擎查詢資訊已經是人們獲取訊息最重要的工具之一，而部落格的資訊隨著長年發展下來，資訊量也越來越大，故有部落格搜尋服務之需求，其中最熱門的部落格搜尋引擎為 Google 開發的 Google Blog Search。

傳統的資訊檢索的方法可以分為兩個基本的大方向，一為更接近使用者的需求，另一個為資料模型的考量。在符合使用者需求方向，主要考量到下列兩個課題：(1)個人化資訊檢索，針對個別使用者需求進行檢索、(2)特定領域的文件檢索，針對某些特定文件進行分析討論。而在資料模型方向，主要著重在下列兩個課題：(1)注重檢索效能，即演算法是否能提供較好的檢索結果，(2)注重模型的計算效率，希望以較少的計算時間或儲存空間。

在我們搜尋部落格文件時，常會面臨到兩個問題：同義詞（synonymy）及一詞多義（polysemy）。例如：(1)同義詞：例如國際籃球協會與 NBA 或是 National

Basketball Association、PS3 和 PlayStation 3、姊姊與姐姐或是 Older sister，上述各個字詞之間呈現的方式不同但它們都是表達同樣的意思，故為同義詞；(2)一詞多義：例如我們的查詢字詞為 Apple，查詢結果可能有水果類的蘋果以及蘋果電腦公司等相關頁面；查詢字詞為 Java，查詢結果可能有爪哇咖啡以及 Java 程式語言等相關頁面。

從潛在語意的層面來看，字詞（term）是一份文件（document）中最基本組成元素，而字詞和文件之間有一層潛在的語意關係，我們稱之為主題（topic）。人們在寫文章時，首先想到的是文章的主題，然後才根據主題選擇合適的字詞來表達自己的觀點。

針對潛在語意關係的問題，學者們提出了以下語意模型：潛在語意分析（latent semanticanalysis, LSA）（Deerwester et al. 1990）、機率潛在語意分析（probabilistic latent semantic analysis, PLSA）（Hofmann 1999）。LSA 和 PLSA 以字詞和文件之間擁有潛在語意關係的概念做為出發點，而 PLSA 使用機率模型的方式強化 LSA 潛在主題（latent topic）的概念，更符合文件的特性。

Mishne 與 Rijke（2006）指出，部落格搜尋和一般的網頁搜尋不同，部落格搜尋將追蹤命名實體作為首要目標，並且經由部落格的主題定位部落格。由於目前的部落格搜尋引擎並沒有針對潛在語意關係進行探討，而 LSA 和 PLSA 符合學者對於部落格文件隱含主題的概念，因此我們採用 LSA、PLSA 為主要研究方法。

本研究考慮到部落格文件中字詞間所隱含的語意關係，以及字詞和文件間所存在之潛在主題關係，將 LSA 及 PLSA 語意模型應用於字詞和部落格文件之間的處理，希望能夠提昇部落格文件檢索的效能，並且建議適用於部落格文件檢索的語意模型方法。本研究有以下目的：

- 提出適合部落格文件檢索系統的語意模型
- 解決部落格文件的同義詞和一詞多義問題
- 使用語意模型後以便提昇部落格文件檢索效能
- 提出成本效能概念，以便加速 PLSA 的執行效率

在我們論文後續部份，我們首先討論與本論文有關之文獻，接下來討論本論文之研究方法，然後探討相關實驗數據，最後以未來研究方向總結本文。

貳、文獻探討

一、潛在語意分析

潛在語意分析（latent semantic analysis, LSA）（Deerwester et al. 1990），是以數學統計為基礎的知識模型，視為向量空間模型的一種延伸，以奇異值分解（singular value decomposition, SVD）（Golub & Reinsch 1970）和維度約化（dimension

reduction）（Deerwester et al. 1990）為核心之語意推導模型，LSA 不僅僅是依照文件中字詞出現的頻率及位置計算出兩篇文件的相似度，而且還會從一篇文章中某些特定概念的詞，擷取出來並重新呈現（Dumais 2005）。

LSA 主要是從一個已知文件及相對應之詞典，採用詞袋（bag of words）模型假設，我們可以將資料集表示為一個 $w_j \times d_i$ 的二維矩陣，其中$<w_j, d_i>$表示字詞 w_j 在文件 d_i 中出現的次數。LSA 的基本思想是，將文件從稀疏的高維語句空間映射到一個低維的向量空間，我們稱之為隱含語意空間（latent semantic space），而其中映射的方式採用 SVD 及維度約化來達成。

在其運算過程中，原始的二維矩陣會利用 SVD 分解技術，將其分解成三個二維矩陣，其中兩組為奇異向量（singular vector），另一組為保存奇異值（singular value）的對角矩陣（diagonal matrix）。兩個奇異向量分別表示字詞及文件對應至隱含語意空間之表達，對角矩陣表示該原始二維矩陣所保留之隱含語意空間。透過過濾雜訊（或稱維度約化）的技術，亦即去除語意空間中某些較不重要的語意，保留某些重要語意空間，重新獲得去除雜訊後之新語意空間，並將上述兩個奇異向量及新語意空間重向計算矩陣乘積後，得到維度約化後之潛在語意矩陣。由於維度約化後之潛在語意矩陣具有去除雜訊語意之新空間，因此能夠正確推理更深層次的隱含語意關係（McInerney et al. 2012）。

LSA 模型廣泛的應用於資訊檢索（information retrieval, IR）領域（Landauer et al. 2013）；Evangelopoulos（2013）討論許多 LSA 應用，諸如：語言學、心理學、認知科學、教育學等；Kuo、Shan 與 Lee （2013）針對影音資料使用 LSA 來建議適當的背景音樂；針對考試命題中簡答題類型之答案評估有學者（Klein et al. 2011; Lintean et al. 2010）使用 LSA 進行評估答案的正確與否；針對大量使用者文件以彙總形式呈現等問題，Ozsoy、Alpaslan 與 Cicekli（2011）利用 LSA 進行文件整理，達成文件彙總的目的；針對搜尋引擎排名函數，Luh、Yang 與 Huang（2012）以 LSA 及基因演算法，推估搜尋引擎的排名演算法。LSA 特色如下（Landauer et al. 2013）：

1. LSA 假設經過 SVD 後所得到的對角矩陣，其所代表的意義是整份文件的語意空間。所謂的語意空間就是文件中每個字詞的定義空間，也就是說，每個字詞可以透過這個語意空間的定位來得到真正代表的意思。
2. 為了要將語意空間的真正維度定義出來，LSA 需要經過維度約化來重建最後的字詞-文件矩陣，亦即使用 VSM（vector space model）矩陣的方式呈現。
3. 經由維度約化後的語意空間更可精確描述字詞與文件內容所代表的意義。
4. 相較於使用外在資源以達到文件模型建構的方法，LSA 提供直接的分析方式，較精確地建構文件的知識模型，且避免使用輔助知識可能發生的語意混淆的問題。

5. LSA 與傳統 IR 的不同在於 LSA 可以涵蓋字詞間關聯程度，更可藉由維度約化將原文件內容中潛在的語意表現出來。
6. LSA 具有知識推演的能力，如果將原始矩陣中的任一個數值改變後，其結果會影響到最後重建的矩陣，且影響的範圍不只是原先經過改變的數值，更可能影響到矩陣中的其他數值。

由於 LSA 採用 SVD 及維度約化進行潛在語意內容的擷取，容易因為 SVD 及維度約化等因素，產生某些先天限制，這些限制如下（Landauer et al. 2013）：

1. 沒有明顯表達字詞出現次數的機率模型。
2. 無法解決一詞多義的問題。
3. SVD 的優化目標是基於 Frobenius 正規化，這相當於隱含了對資料的高斯噪聲假設。然而字詞出現的次數為非負數的，這明顯不符合高斯假設，其更接近多重正規化分佈。
4. 對於計次向量而言，由於矩陣重建時產生負數矩陣，因此歐幾里德距離表達是不合適的。
5. 特徵向量的方向沒有對應的物理解釋。
6. SVD 的計算複雜度很高，而且當產生新的文件時，整個模型需要重新計算。
7. 維度約化所選擇的維度是隨機的。

二、機率潛在語意分析

為了解決 LSA 模型在文件和字詞上的呈現，無法刻劃出字詞出現的機率問題，因此 Hofmann（1999）提出了機率潛在語意分析（probabilistic latent semantic analysis, PLSA）進行解決。PLSA 是一個基於統計上潛在類別模式的一種自動文件檢索方法，可以對計數資料做個別的因子分析（Hofmann et al. 2008）。

不同於 LSA 是將文件和字詞向量投射至潛在語意空間的作法，PLSA 以生成模型（Aspect Model）作為主要的架構，可用於分析字詞和文件共同出現（co-occurrence）的現象，使用機率密度函數作為觀察到的文件和字詞間潛在語意關聯性的呈現方式，並利用期望值最大化（expectation maximization, EM）演算法進行參數最大化估計，進而推估潛在參數（latent variables）結果之機率模型。PLSA 對於 LSA 是一個重要的進階觀點（Hofmann 1999）。

PLSA 透過參數估計的方法，推估字詞與文件在主題空間的潛在機率。其推估的過程主要採行 EM 演算法不斷的重複修正所估計的潛在機率參數，並經由參數最大概似估計法則進行演算法終止判斷的依據。EM 演算法主要是針對無法觀察之潛在參數進行參數最大化估計之演算法。EM 演算法經常用在機器學習或計算機視覺的資料分群領域。EM 演算法經由兩個步驟交替進行計算，第一步是期望步驟

（expectation step），利用已觀察到之隱藏變數現有估計值，計算未知潛在參數之概似估計值；第二步是最大化步驟（maximization step），最大化在期望步驟上求得的潛在參數概似估計值。最大化步驟上找到的估計值被用於下一個期望步驟計算中，這個過程不斷交替進行。

PLSA 模型目前被廣泛應用於許多領域，Hennig（2009）使用 PLSA 進行以主題為基礎之多個文件彙總工作；Zhanga 與 Gong（2010）針對不同影像間所出現的動作採用 PLSA 進行分類；McInerney、Rogers 與 Jennings（2012）針對使用者日常生活的移動路徑，使用 PLSA 進行位置預測；針對每日聲音事件（例如：辦公室、街道、商店等），Mesaros、Heittola 與 Klapuri（2011）使用 PLSA 進行事件之偵測；針對音樂的不同風格分類，Zeng 等（2009）使用 PLSA 進行分類；針對影像中人類的動作所可能產生的不同情境，Xu 等（2009）使用 PLSA 進行動作識別。PLSA 的主要的特徵，是針對字詞和文件所產生之共同事件尋求一個生成模型。PLSA 的特色如下（Hofmann et al. 2008）：

1. 定義了機率模型，而且每個變數以及相對應之的機率分佈和條件機率分佈都有明確的物理解釋。
2. 可以妥善處理一詞多義問題。
3. PLSA 隱含的多項式分配（multi-nominal）假設符合文件特性。
4. 可以利用各種模型選擇和複雜度控制準則來確定潛在主題的維度。

三、EM 演算法終止條件

PLSA 模型主要使用 EM 演算法進行潛在變數的推估，雖然 EM 演算法可以收斂到區域最佳解，但它可能花費很多時間才能達到此解。為了節省計算時間，許多學者嘗試使用不同機制去決定 EM 演算法是否該收斂與否。根據相關文獻，這些機制包含下列兩個終止機制：(1)設定一個固定的執行世代當成 EM 演算法的最大允許世代（Gibson et al. 2005; Metaxoglou & Smith 2007; Nguyen et al. 2007）；以及(2)設定一個預設的門檻值（threshold）以便決定 EM 演算法是否終止與否。

現在讓我們來討論第二個終止條件。Ristad 與 Yianilos（1998）定義 EM 演算法的終止條件為訓練語料庫中兩個連續世代的累進總和機率小於一個固定的門檻率；Zhang 與 Goldman（2001）將 EM 演算法推廣至分散密度概念以便解決多實例之學習問題，其在期望步驟選擇一些實例集合，在最大化步驟，他們從所有選擇的實例集合中計算分散密度的機率，當兩個連續世代的累進機率小於或等於一個固定的機率值的話，其判斷 EM 演算法應該終止；許多學者（Gibson et al. 2005; Pernkopf & Bouchaffra 2005; Wen et al. 2008）使用對數概似函數當 EM 演算法的效能函數，EM 演算法終止的條件如下：假設兩個連續世代的效能函數相對改變值小

於一個相對小的機率值。

雖然上述兩個終止條件與區域最佳解比較起來都能夠得到快速的回應時間，但是這兩個終止條件可能造成下列兩個主要問題：首先，設定一個小的執行世代數可能造成最後產生的解與區域最佳解差異很大；其次，設定一個大的執行世代數可能造成最後產生的解與區域最佳解差異只有一點點，但卻花費非常多的時間才完成。

為了避免上述兩個潛在問題，本研究採用改善歷史進度與變動歷史進度的概念來動態決定 EM 演算法是否該終止與否。這樣做法的隱含精神在於：由於改善歷史及變動歷史進度在 EM 演算法的每個世代都不會固定，所以實際 EM 演算法的執行世代數是變動的。如此一來，我們可以動態決定 EM 演算法的執行世代，以避免 EM 演算法花費大幅時間執行，然而改善值卻很少的情況發生。

四、部落格搜尋

早期的網際網路稱為 Web 1.0，使用者可經由入口網站獲取世界各地之廣泛資訊，然而其資訊的傳播僅實現了單向的網站對使用者，缺乏與使用者做積極之互動。近年來，強化使用者參與概念的要求下，Web 2.0 才真正實現，現今許多社群網站即是基於 Web 2.0 的概念所建置。Web 2.0 是網路運用的新時代，網際網路成為了新的平台，內容因為每位使用者的參與而產出的個人化內容，經由人和人之間的分享，形成了現在的 Web 2.0 世界。Web 2.0 最廣為人知的應用便是部落格，部落格又稱為網路日誌，是一種通常由個人管理，不定期發表新的文章、圖片、或影片的網頁，亦即記錄使用者的生活。部落格發展至今影響了全世界，儘管近年來其它類型之社群網站崛起，部落格的服務依然有它的優勢，例如：記錄生活點滴、傳達個人思想、抒發個人情緒、交流專業知識、企業宣傳文化等等。在多家部落格業者提供的服務之下，世界上已經有太多的部落格，於是衍生出部落格搜尋引擎之服務，例如 Google Blog Search（2014）、Sysomos（2014）、Technorati（2014）等等。

近年來，部落格搜尋的相關研究已如雨後春筍般的出現；接下來，我們整理近年相關之研究。Jeong 與 Oh（2012）提出一個部落格搜尋架構，此架構能夠在給定一個主題下進行更深入的搜尋，經由在社群網站收集某些具有智慧的特徵值以及使用這些特徵值所衍生出來的後續查詢。Keikha 等（2013）從部落格搜尋尋找適合的產品預覽（product review），在他們的方法裡，需要決定部落格文章中是否包含任何產品預覽的句子；其決定方法包含兩個步驟：(1)針對所有產品建立詞彙集合，針對每一項產品，建立產品觀點的特性詞彙；(2)針對每一個詞彙集合進行相關部落格搜尋，並整理所有的產品預覽。Wyner 與 Engers（2010）提出基於

部落格搜尋之 e 化政府架構，基於此架構，其提出了豐富的線上討論論壇架構，在架構中部落格貼文之正反意見使用結構化的方式記錄，並且所有的表達意見都可經由可評估的方法進行分類；該架構主要是透過整合模式（integrating mode）、自然語言處理（natural language processing, NLP）、本體論（ontology）等方式建構，並以多線程討論列表（multi-threaded discussion list）的方式呈現。Zhu、Sun 與 Choi（2008）提出一個基於部落格搜尋引擎回傳結果所進行的垃圾部落格貼文偵測技術，該技術主要分析部落格搜尋引擎中以時間序列方式回傳之結果，並且對於這些部落格貼文中經常出現在前面排名的搜尋結果，建立及維護其部落格輪廓（Blog Profile），最後所有可能的垃圾部落格貼文都會呈現在此輪廓之中。Takama 等（Takama et al. 2005）以關鍵字地圖（keyword map）為基礎所建立之互動式部落格搜尋，其提出一個演算法可以考慮在多個主題下，產生不同主題下對該關鍵字地圖有興趣之使用。Kim 等（2008）發現部落格頁面之中，往往存在許多非傳統網頁的特色，例如：引用連結（trackback link）、部落客之名聲、標籤以及相關使用者回應；其提出一個新的引用連結排名演算法，以便進行部落格貼文排名，該演算法透過部落客聲譽積分、引用積分，以及內容積分等多樣式評比重新排名。

參、研究方法

接下來，我們針對本研究所使用的研究方法和整體架構進行描述。首先，在第一部份，我們簡略討論本論文之研究流程，如圖 1 所示；接下來，在第二部份，我們說明本研究所採用之部落格文件來源；第三部份說明文件前置處理的方法和流程；第四部份說明本研究的機率矩陣處理；第五部份說明 LSA 的詳細過程；第六部份為 PLSA 的詳細流程；最後，我們討論 EM 演算法的終止條件。

一、研究流程

根據圖 1 所示，首先我們利用網路爬蟲（web crawler）擷取線上部落格文件，接著透過 PCRE（perl compatible regular expressions）正規文法（Hazel 1997）將擷取的原始部落格文件做規則語法的處理，擷取出我們想要的文件內容並統一相關文件格式。自然語言處理則是使用一連串的文字處理，嘗試將非結構化之網頁文件進行結構化處理，同時去除不必要之停用字及非文數字標籤（如標點符號或特殊符號）。矩陣處理則是將自然語言處理過的部落格文件形成可以輸入語意模型的三種類型矩陣，最後運行 LSA 及 PLSA 語意模型。

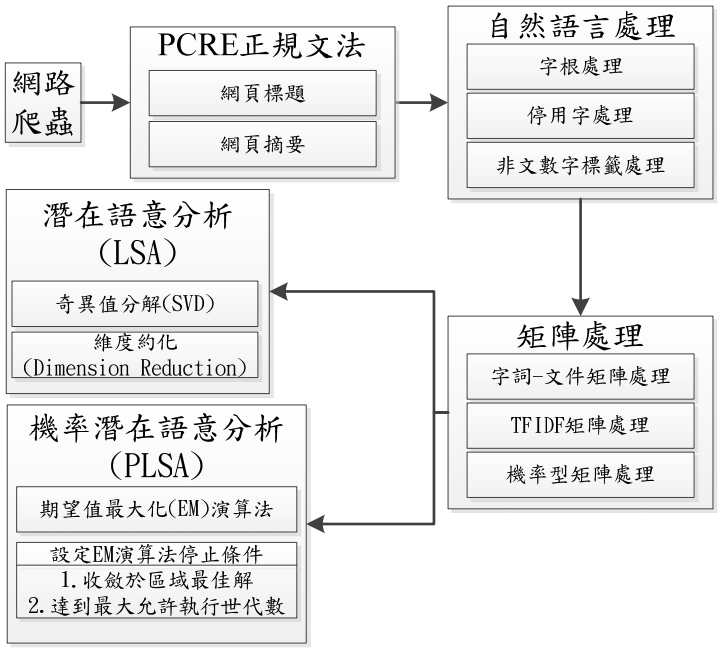


圖 1：本論文之研究流程圖

二、部落格文件資料來源

首先，我們必須取得線上部落格文件，本研究使用 Google Blog Search（2014）為我們主要的部落格文件來源。Goolge Blog Search 是 Google 搜尋引擎內的一個附加功能，其將回傳經關鍵字查詢後之部落格文件搜尋頁面，此搜尋頁面多為知名部落格服務平台上部落客所發表的文章，如 WordPress（2014）、Blogger（Google 2014）、MySpace（2014）等等，Google Blog Search 可以透過 site 參數的方式擷取特定部落格服務平台之文章。由於 Google Blog Search 是一個應用最為廣泛且索引頁面也最多的搜尋平台（Wikipedia 2014），經由對 Google Blog Search 使用元搜尋（metasearch）技術（Selberg & Etzioni 1997）後，我們將可取得最為廣泛的文件來源，因此我們決定選擇它的搜尋結果當作本研究部落格文件的資料來源。

三、文件前置處理

在文件前置處理的過程之中，我們首先透過網路爬蟲輸入欲搜尋的關鍵字，並將取得的網頁文件轉換成可以後續處理的資料格式。由於網頁文件屬於非結構化文件(陳林志&林育任 2013)，為了取得可以處理之結構化內容，我們利用 PCRE 正規文法（它是由一組函數執行自定義之規則文法）取得適當之結構化資料，本研究經由自行分析之 Google Blog Search 規則文法，並透過 PCRE 正規文法擷取所

需的資料內容，這部份的資料內容包含網頁標題（web title）及網頁摘要（web snippet），其相關文件格式資訊如表 2 所示。

表 2：經 PCRE 正規文法後所擷取的部落格文件資訊

擷取資訊	內容
網頁標題	More Confirmation of Sharp Supplying iPad 3 Displays - Mac Rumors</h3>
網頁摘要	The Wall Street Jornal offers their own sources that confirm that Apple will be using Sharp as a source of display parts for the upcoming iPad 3. ...

觀察表 2 後，我們發現擷取下來的網頁標題和網頁摘要皆存在符號、停用字、網頁標籤等無意義的資訊，對於往後的研究分析容易造成誤判；因此本研究採用一系列的自然語言處理，如圖 1 的自然語言處理階段，其中包含字根處理（stemming）、停用字處理（stop words）、非文數字標籤處理（non-word tokens），相關描述如下：

- 字根處理：第一個自然語言處理為字根處理，在英文的文法結構中，由於名詞的單複數（如 cat 和 cats）、詞性的變化（如 good 和 goodness）、動詞的時態（如 walk 和 walked），導致語意大致相同的詞卻有多種不同方式呈現。若是將這些語意大致相同的字詞各自視為不同的字詞，則相對稀釋了其重要性；故我們必須先經過字根還原的處理，將具有相同字根的字詞視為同一個字詞。本研究之字根處理方法以 Porter 字根演算法（Porter 1980）為依據。接下來，我們以 Landauer、Foltz 與 Laham（1998）的例子進行說明（如下表所示），其中共有 9 句文章標題，其中 c1~c5 的文章屬於人機介面領域；m1~m4 則是關於數學圖論的文章。

表 3：Landauer 範例之文章標題表格

文章	標題
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
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

表 3 是未經過任何自然語言處理過的原始資料，在我們自然語言處理的第一個步驟，我們將所有的單字還原成字根，透過 Porter 字根演算法得到結果如表 4 中之第 2 行所示。

表 4：經過自然語言處理（含字根、停用字、非文字標籤）所產生的結果

文章	經過字根處理後之標題	移除停用字處理後之標題	經非文字標籤處理後之標題
c1	human machin interfac for abc comput applic	human machin interfac abc comput applic	human machin interfac abc comput applic
c2	a survey of user opinion of comput system respons time	survey user opinion comput system respon time	survey user opinion comput system respon time
c3	the eps user interfac manag system	eps user interfac manag system	eps user interfac manag system
c4	system and human system engin test of eps	system human system engin test eps	system human system engin test eps
c5	relat of user perceiv respons time to error measure	relat user perceiv respon error measur	relat user perceiv respon error measur
m1	the generat of random, binary, order tree	generat random, binary, tree	generat random binary tree
m2	the intersect graph of path in tree	intersect graph path tree	intersect graph path tree
m3	graph minor iv: width of tree and well-quasi-ord	graph minor iv: width tree well-quasi-ord	graph minor iv width tree well quasi ord
m4	graph minors: a survey	graph minors: survey	graph minors survey

觀察表 4 之第 2 行後，我們可以發現，所有的單字已經還原成字根的形態，並且都已經轉換小寫，但是還存在大量的停用字。在下個步驟我們使用停用字處理去除所有的停用字。

- 停用字處理：在一般的英文文章中，往往出現大量的停用字（如 a、is、the、

or、of 等）。然而這些停用字單獨存在時大部分是無意義的，而且會影響字詞的擷取及辨識上的準確度和錯誤引導，因此這些停用字是必須去除。本研究所使用的停用字字典為 Fox（1989）所建議之 421 個停用字字典，該字典經由 Fox 統計大量英文文章後所產生。觀察表 4 中之第 3 行後，我們可以發現，所有停用字皆已經移除，但還存在標點符號或特殊符號等非文數字標籤符號等相關問題，因此還需要做一些處理。在下個步驟我們使用非文數字標籤處理去除所有非文數字標籤符號。

- 非文數字標籤處理：在一般的英文網頁文件中，可能含有標點符號（如：-,";' 等等）、特殊符號（如：@#%&等等）及網頁標籤（如：</p>、
等）等相關之非文數字符號。這些符號可能會影響到字詞的擷取和辨識上的準確度，因此我們必須將這些非文數字符號移除。表 4 之第 4 行為經過非文數字標籤處理後的結果。

觀察表 4 的最後結果，我們可以發現，所有非文數字標籤符號都已經被移除，文件前置處理的流程已經完成，可以進入下一個階段，矩陣處理步驟。

四、矩陣處理

（一）VSM 矩陣及 TFIDF 矩陣處理

經過文件前置處理後的部落格文件會是以表 4 之第 4 行的形式呈現，但是要將這些部落格文件進行 LSA 或 PLSA 的語意分析處理，則需要將相關文件轉換為 VSM 矩陣的形式。

表 5：VSM 矩陣的呈現形式

文件 / 字詞	I like apple	I hate hate apple
I	1	1
like	1	0
hate	0	2
apple	1	1

表 5 為一個簡單的 VSM 矩陣之範例。假設文件 1 的內容為"I like apple"；文件 2 的內容為"I hate hateapple"。觀察此兩文件，我們發現字詞"I"在文件 1 出現 1 次、文件 2 也出現 1 次；字詞"like"在文件 1 出現 1 次、文件 2 出現 0 次。

本研究將所蒐集之部落格文件轉換為 VSM 矩陣形式，主要是將 Google Blog Search 所回傳之每一筆網頁標題及網頁摘要視為 1 份文件，並且使用停用字將此

文件進行字詞之切割。表 6 假設為本研究經過處理的網頁標題和網頁摘要範例，其中輸入的查詢為 ipad，文件總數為 10，切割之字詞總數為 110。

表 6：本研究經過處理的字詞和文件範例

Documents		Terms			
d1	ion piano apprentic ipad piano teacher alway learn play piano…	t1	ion piano apprentic	t11	internet equival
d2	instapap ipad refresh friends read engadgetinstapaper intern…	t2	ipad	t12	engadgetinstapaper
d3	ipad live tonight pm edt there! tipbipad live dam ipad pod…	t3	piano teacher always	t13	bookmark
d4	featur ipad appl gazettei earli start talk ipad dont so appl an…	t4	learn	t14	top to to makeov
d5	tim cook yeah ipad eat mac sales probabl greg kumparak m…	t5	play	t15	outing
d6	ipad mini rumor reviv claim inch screen mac taiwan unit da…	t6	piano	t16	ipad fear not
d7	blurb launch ipad ebook publish serviceonlin book publish…	t7	instapap	t17	reading ipad live
d8	chart day ipad bigger hit iphonethat becaus ipad benefit app…	t8	ipad refresh	t18	pm edit
d9	top essenti iphon ipad blog app real blog iphon ipad apps w…	t9	friends	t19	there tipbipad live
d10	powerpoint ipad slideshark happen semless for unfamiliar…	t10	read	more	…

表 6 的字詞和文件皆已經過自然語言處理和去除冗長、重複字詞處理，接下來將之形成 VSM 矩陣。表 7-(a)為本研究形成的 VSM 矩陣的一個範例，此表格中的第二列，可以看到此字詞在每個文件都有出現，代表這個字詞對於這個查詢有很大的相關性，參照表 6 我們可以發現第二個字詞是 ipad 與輸入查詢 ipad 是相同的字詞。

表 7：本研究所產生之 VSM 及 TFIDF 矩陣

VSM 矩陣						TFIDF 矩陣					
	d_1	d_2	d_3	d_4	d_5		d_1	d_2	d_3	d_4	d_5

t_1	1	0	0	0	0	t_1	0.33219	0	0	0	0
t_2	1	2	3	4	1	t_2	0	0	0	0	0
t_3	1	0	0	0	0	t_3	0.33219	0	0	0	0
t_4	1	0	0	0	1	t_4	0.23219	0	0	0	0.23219
t_5	1	0	0	0	0	t_5	0.33219	0	0	0	0
t_6	3	0	0	0	0	t_6	0.99658	0	0	0	0
t_7	1	0	0	0	0	t_7	0.33219	0	0	0	0
t_8	0	2	0	0	0	t_8	0	0.66439	0	0	0
t_9	0	1	0	0	0	t_9	0	0.33219	0	0	0
t_{10}	0	1	0	0	0	t_{10}	0	0.33219	0	0	0

VSM 矩陣形成後，我們可以將這個矩陣再進行 TFIDF（term frequency-inverse document frequency）的處理，表 7-(b)為本研究形成的 TFIDF 矩陣。

PLSA 為我們主要的研究方法之一，因此針對機率這個課題，本研究將 VSM 矩陣轉換成機率型式之 VSM 矩陣。接下來我們說明這個轉換工作如何完成。

（二）機率型矩陣處理

這個工作主要將原始的 VSM 矩陣轉換成機率形式的矩陣，本研究定義了一個簡單的機率方法，如下列公式所示；其中 TP(i,j)為文件 j 中、字詞 i 出現的機率、TF(i,j)為字詞 i 在文件 j 中出現的次數、TW(j)為文件 j 的總字數。

$$TP(i,j) = TF(i,j) / TW(j) \tag{1}$$

下列為一個簡單的計算範例。我們套用表 5 的例子，“I”這個字詞在文件 1（“I like apple”）中出現的機率為 1/3=0.3333，在文件 2（“I hate hate apple”）中出現的機率為 1/4＝0.25。表 8 為依據表 7-(a)所產生之機率型矩陣。

表 8：本研究所產生之機率型矩陣

	d_1	d_2	d_3	d_4	d_5
t_1	0.07143	0	0	0	0
t_2	0.07143	0.11111	0.15	0.19048	0.03571
t_3	0.07143	0	0	0	0
t_4	0.07143	0	0	0	0.03571
t_5	0.07143	0	0	0	0
t_6	0.21429	0	0	0	0

t_7	0.07143	0	0	0	0
t_8	0	0.11111	0	0	0
t_9	0	0.05556	0	0	0
t_{10}	0	0.05556	0	0	0

經過處理後之 VSM 矩陣，TFIDF 矩陣，和機率型矩陣，將進行 PLSA 處理。PLSA 的詳細流程，將在後續進行詳述。

五、潛在語意分析（LSA）

LSA 是以數學統計為基礎的知識模型，以奇異值分解和維度約化為核心作為邏輯推演的方式。LSA 除了可以作為文件的知識呈現之外，並可用來推演潛在語意和知識關聯，LSA 的知識模型和知識推演過程，接近於人腦用來理解文件知識的推演與認知機制模型（Dumais 2005）。

目前 LSA 常被應用在資訊擷取、字詞與文件相關性的判斷、同義詞的建構、文件品質優劣的判別、文件摘要的評量等研究（Kanejiya et al. 2003）。LSA 不僅僅只是依文章關鍵詞出現的頻率及位置，計算出兩篇文章的相似度，在 LSA 的運算過程中，原始的二維矩陣會利用 SVD 技術分解成三個二維矩陣，其中兩組為奇異向量（singular vector），另一組對角矩陣則用來保存奇異值（singular value）。在對角矩陣中保留適當個數的奇異值（又稱維度約化），並過濾雜訊後，再將三個矩陣相乘，就可以得到具有潛在語意的新矩陣，此矩陣能夠正確地推理更深層次的關係。因此在以人類意識為主的判斷上，LSA 比那些長久以來就被人們所遺棄的表象方法，更能提供較佳的預測效能（Dumais 2005）。由於 LSA 為向量空間模型的延伸方法，基本上權重值給定方法和向量空間模型極為相似，都可以用不同方法來取得文件和字詞間的統計量，如 TFIDF 和熵（entropy）等等。

（一）奇異值分解及維度約化

奇異值分解主要是解決一個最小平方問題。給定一個實數矩陣 $A_{m\times n}$，可以假設 $m\geq n$ 且 $rank(A)=r$，其中 r 表示奇異值的數量，也代表該矩陣的 rank 大小。即使維度降低為 r，但對於較稀疏的資料而言，r 的值仍可能過大。而在 LSA 的應用上，通常將奇異值的維度 r 降低到約 100-300 左右（Dumais 2005），對 $A_{m\times n}$ 操作 SVD 可以表示成以下公式；其中 A 為給定的實數矩陣、U 為字詞向量矩陣、S 為保存奇異值的矩陣、V^T 為文件向量矩陣。

$$A = USV^T \tag{2}$$

保留原始矩陣 A 的前 K 個維度所形成之近似矩陣 A_K，列向量投影到以右奇異

矩陣 V 之行向量為基底的空間，在這個空間中可以將字詞用一個新的向量空間來呈現，也就是說在這個空間中投影的向量就是 US 的列向量。亦即，u_iS 為 w_j 在 r 維空間的位置，其中 $1 \le i \le m$。

另一方面，在 A_K 矩陣之中，行向量投影到左奇異矩陣 U 之行向量為基底的空間，也採用類似上述之做法，在此空間中可以將文件視為一個新的向量空間。換言之，在這個空間中投影的向量也就是 SV^T 的行向量；進一步來看，SV^T_j 為 d_i 在 r 維空間的位置，其中 $1 \le j \le n$。

根據上述的說明，SVD 最基本的概念是近似矩陣 A_K 保留了原始矩陣 A 的結構。潛在語意分析空間中若兩個字詞相近，可能代表此兩字詞，往往出現在相同類別的文章之中，但並不代表這些字詞必然出現在這些的文章裡。同理，矩陣裡兩個文章相近，代表可能這兩篇文章裡擁有相同的語意，但並不代表這兩篇文章擁有相同的字詞。我們以表 9 的例子說明 LSA 的實行步驟，其中文件來源 c1~c5 及 m1~m4 的來源為表 3 經由自然語言處理後所產生之表 4 結果。

表 9：依據表 4 所形成之 VSM 矩陣

[X] =									
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1
r（human,user）= -0.38 r（human,minors）= -0.29									

在表 9 之中，我們使用斯皮爾曼等級相關係數（Spearman rank correlation coefficient）計算字詞 human 和 user 的語意關係為-0.38；而 human 和 minors 的語

意關係為-0.29。我們可以由表 9 中看出 human 和 user 是出自同一類型的文件，human 和 minors 是出自不同類型的文件，反而 human 和 minors 的語意關係卻比 human 和 user 之間來得高。接下來，我們經由 SVD 將表 9 的矩陣分解成表 10 的三個二維矩陣。

表 10：經由 SVD 分解後的矩陣

[X] = [W]×[S]×[P]’								
[W] =								
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	0.02	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
[S] =								
3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36
[P]’								
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	-0.02
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	-0.05	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

假設表 9 的 VSM 矩陣表示成[X]，經過 SVD 分解後的三個矩陣分別表示成表 10 的[W]、[S]、[P]’，其中[W]保留有原始字詞的資訊、[S]為對角矩陣、[P]’為[P]的轉置矩陣並保留有原始文件的資訊。經過 SVD 分解後必須進行維度約化的動作，我們假設只保留對角矩陣[S]的 2 個最大奇異值（亦即 K=2），其他較小且不重要的奇異值視為雜訊，並將之去除（如表 10 之[S]矩陣之中我們只保留 3.34 及 2.54，其餘設定為 0）。經維度約化之後得到的新對角矩陣，再將原始之[W]及[P]’矩陣進行矩陣乘積，得到一個去除雜訊後之新 VSM 矩陣[$\overline{X}$]，結果如表 11 所示。

表 11：經由去除雜訊後產生之新 VSM 矩陣

[X] =									
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.46	0.40	0.38	0.47	0.18	0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62
r（human,user）= 0.94 r（human,minors）= -0.83									

參考表 11，再將 human 和 user 這兩個字詞的列向量直接拿來計算其相關係數時，其值為 0.94。此外在表 9 中字詞 tree 在 m4 文件中的出現值原本為 0，但在表 11 中的值卻增為 0.66；在表 9 中字詞 survey 在 m4 文件中的出現值原本為 1，但在表 11 中的值卻降為 0.42。由此可以看出，經過 SVD 的轉換後，所產生的語意空間可以將許多原本在字面看不見的資訊顯現出來。類似於 LSA 的思想，在 PLSA 中也引入了一個潛在階層（Latent Class）的概念，但這次要用機率模型的方式來表達 LSA。接下來我們要描述 PLSA 的細節。

六、機率潛在語意分析（PLSA）

（一）PLSA 模型參數

SA 偵測共同字詞的結構和解決同義字問題的效果非常的好，但無法刻劃出字詞出現的機率，另外 LSA 無法識別在不同主題中相同字詞之一詞多義現象，而且 LSA 中 SVD 所產生的數值可能為負。為了解決上述這些問題，Hofmann（1999）提出了 PLSA 模型，此方法可以從所蒐集之字詞中，尋找字詞間可能存在之一詞多義關係。

LSA 模型不同於 LSA 將文件和字詞向量投射至潛在語意空間的作法，其方法是以生成模型作為主要架構，使用機率密度函數作為觀察到的文件和字詞間潛在語意關聯性的呈現方式，並利用 EM 演算法進行參數最大化估計，進而推估潛在參數結果之機率模型（Dempster et al. 1977）。

LSA 模型主要的特徵，是針對字詞和文件共同事件尋求一個生成模型，本文資料集是由文件-字詞對（d_i,w_j）所組成，文件以 $d_i \in \{d_1,\ldots,d_N\}$ 表示，其文件總數為 N；另外，字詞以 $w_j \in \{w_1,\ldots,w_M\}$ 表示，其字詞總數為 M。PLSA 存在一個潛在主題變數 $z_k \in \{z_1,\ldots,z_K\}$，其潛在主題總數為 K，此潛在主題呈現字詞及文件可能共同出現之關聯性。現在，讓我們正式的描述如何計算本研究所需要的訓練參數，本研究經由 PLSA 模型最終會形成如下表之 PDW 矩陣。

表 12：本研究之 PLSA 訓練參數

[PDW] =				
$p(1,1)$	…	$p(d_i,1)$	…	$p(N,1)$
⋮	⋮	⋮	⋮	⋮
$p(1,w_j)$	…	$p(d_i,w_j)$	…	$p(N,w_j)$
⋮	⋮	⋮	⋮	⋮
$p(1,M)$	…	$p(d_i,M)$	…	$p(N,M)$

為了要計算 PDW 矩陣內的項目，$p(d_i,w_j)$，我們使用下列公式進行計算；其中 $p(d_i)$ 表示特定文件 d_i 發生的機率、$p(w_j|z_k)$ 表示給定一個潛在主題 z_k，出現字詞 w_j 的機率、$p(z_k|d_i)$ 表示給定一個已知文件 d_i，出現潛在主題 z_k 的機率。

$$p(d_i,w_j)=p(d_i)p(w_j\mid d_i)=p(d_i)\sum_{k=1}^{K}p(w_j\mid z_k)p(z_k\mid d_i) \tag{3}$$

接著 PLSA 應用貝氏定理（Bayes Rule）轉換公式(3)為下列公式：

$$p(d_i,w_j)=\sum_{k=1}^{K}p(w_j\mid z_k)p(z_k)p(d_i\mid z_k) \tag{4}$$

為了要計算 $p(d_i,w_j)$，PLSA 使用最大概似估計法則計算出 PLSA 模型的參數，經由概似估計函數（likelihood estimation function）計算最大概似估計值。下列函數是我們在世代 n（Iteration=n）下，所定義之概似估計函數 $L_n(d_i,w_j)$，經由最大化此函數值後，我們可以決定最終之 $p(d_i,w_j)$參數：$p(w_j|z_k)$、$p(z_k)$、$p(d_i|z_k)$；其中 $td(d_i,w_j)$ 表示文件 d_i 中字詞 w_j 所發生的權重，其內容可以是 VSM 矩陣（如表 7-(a)形式），也可以是 TFIDF 矩陣（如表 7-(b)形式）或機率形式矩陣（如表 8 形式）。

$$L_n(d_i,w_j)=\sum_{i=1}^{N}\sum_{j=1}^{M}td(d_i,w_j)\log\{p(d_i,w_j)\} \tag{5}$$

此函數也可以解釋為求得 $p(w_j|d_i)$與 $td(d_i,w_j)/td(d_i)$兩個分配之間的 K-L 散度（kullback–leibler divergence）為最小；其中 $td(d_i)$代表 $td(d_i,w_j)$矩陣中文件 d_i 下之所有字詞分量值之加總，即更好的 $p(d_i,w_j)$，刻劃共生矩陣的實際分配。

然而在 PLSA 模型中的最大化概似估計函數的標準程序為 EM 演算法，EM 演算法為 PLSA 模型的核心方法；一般來說，EM 演算法需要迭代執行下列兩個步驟：(1)期望步驟（expectation step）、(2)最大化步驟（maximization step）。其中條件機率的更新和最大化都是基於公式(5)的概似估計函數去計算的，對於一個給定的初始值，在 EM 演算法的迭代過程中，概似估計函數的數值會隨著執行世代（在本論文之中，每完成一次迭代，我們將之稱為一個世代）增加而遞增。接下來，我們詳述 EM 演算法。

（二）EM 演算法

EM 演算法在統計中被用於尋找，依賴於不可觀察的潛在變數（latent variable）的機率模型中，參數的最大概似估計。在統計計算中，EM 演算法是在機率模型中尋找參數最大概似估計或者最大後驗估計的演算法，其中機率模型依賴於無法觀測的潛在變數。EM 演算法經常用在機器學習和計算機視覺的資料分群領域。EM

演算法經過兩個步驟交替進行計算，第一步是期望步驟（expectation step），利用已觀察到之隱藏變數現有估計值，計算未知潛在參數之概似估計值；第二步是最大化步驟（maximization step），最大化在期望步驟上求得的潛在參數概似估計值。最大化步驟上找到的估計值被用於下一個世代的期望步驟計算中，這個過程不斷交替進行。圖 2 為本研究 PLSA 步驟詳細的流程圖，PLSA 核心的 EM 演算法的兩個步驟詳細陳述如下所示：

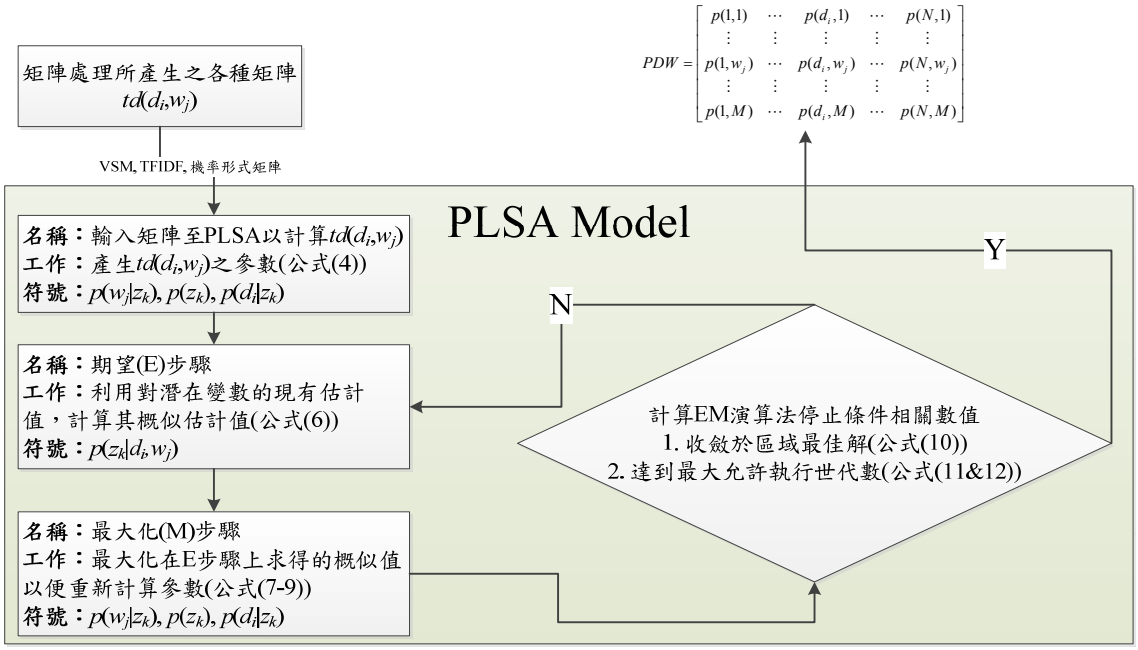


圖 2：本研究 PLSA 步驟流程圖

1. 期望步驟（expectation step）：在期望步驟中，主要的目的是要利用變數 d_i 及 w_j 的現有估計值，以便估算潛在變數 z_k 的概似估計值，其公式如下所示：

$$p(z_k \mid d_i, w_j) = \frac{p(d_i \mid z_k)p(z_k)p(w_j \mid z_k)}{\sum_{k=1}^{k} p(d_i \mid z_k)p(z_k)p(w_j \mid z_k)} \tag{6}$$

2. 最大化步驟（maximization step）：在最大化步驟中，主要是針對期望步驟所求得的潛在變數 z_k 之最大概似估計，並在最大化的過程中，重新計算 $p(d_i,w_j)$ 之相關參數：$p(w_j|z_k)$、$p(z_k)$、$p(d_i|z_k)$，其參數更新如下：

$$p(w_j \mid z_k) = \sum_{t-1}^{N} td(d_i, w_j)p(z_k \mid d_i, w_j) / \sum_{t=1}^{N}\sum_{j=1}^{M} td(d_i, w_j)p(z_k \mid d_i, w_j) \tag{7}$$

$$p(z_k) = \sum_{t=1}^{N} \sum_{j=1}^{M} td(d_i, w_j) p(z_k \mid d_i, w_j) / \sum_{t=1}^{N} \sum_{j=1}^{M} td(d_i, w_j) \quad (8)$$

$$p(d_i \mid z_k) = \sum_{j-1}^{N} td(d_i, w_j) p(z_k \mid d_i, w_j) / \sum_{t=1}^{N} \sum_{j=1}^{M} td(d_i, w_j) p(z_k \mid d_i, w_j) \quad (9)$$

最大化過程主要是透過 Lagrange 最佳化（Hofmann et al. 2008）達成潛在變數最大概似估計，透過最大化過程我們可以將 $p(w_j|z_k)$、$p(z_k)$、$p(d_i|z_k)$ 這些 PLSA 的參數轉變成公式(7-9)的表達形式，再將重新獲得之 PLSA 參數代入公式(4)。這將會導致概似估計函數（公式(5)）持續的增加；上述期望步驟和最大化步驟將會不斷交替執行，直到達成我們所有設定之 EM 演算法停止條件。接下來，我們討論 EM 演算法之停止條件。

七、EM 演算法停止條件討論

PLSA 的時間複雜度與 EM 演算法的收斂速度息息相關，Hofmann、Schölkopf 與 Smola（2008）已經證明出 PLSA 的時間複雜度為 O(M×N×K)，其中 O(M×N) 為 EM 演算法中每個世代的執行時間。然而，在現今 Internet 環境，字詞的總數（M）以及文件的總數（N）都是非常巨大的（Kunder 2008）。同時，潛在主題的個數（K）也會跟隨著 M 及 N 成長（Inoue 2005）；當 PLSA 要應用在大規模的 IR 問題時，在如此巨大的 M、N 及 K 下，這很容易造成效能退化的問題。

根據文獻探討中，關於 EM 演算法的終止條件的討論中，我們使用下列兩種情形進行判斷：(1)收斂於區域最佳解；以及(2)達到最大允許執行的世代數。接下來，我們討論這兩種形。

- 情形一：收斂於區域最佳解

  在這種情形下，我們定義當兩個連續世代的改善值小於一個預設之門檻值時，則求得區域最佳解。我們使用下列公式定義收斂於區域最佳解；其中 I_n 表示兩個連續世代的改善值、λ 表示預定的門檻值。

$$I_n \le \lambda$$

其中

$$I_n = L_n(d_i, w_j) - L_{n-1}(d_i, w_j) \quad (10)$$

- 情形二：達到最大允許執行世代數

  當最大允許世代數設定很大時，很容易浪費計算資源，但卻只能得到些許

的改善。相對的來說，世代數設定很小時，可能得到一個假性收斂的結果。如果最大允許世代數能經由改善歷史進度的狀態來決定，則所求的解將是一個成本-效能解。

現在讓我們正式定義 EM 演算法所需執行世代數為：”連續未改善的世代數”大於”最大允許沒有明顯改善的世代數”。如果上述條件存在的話，代表進一步的計算只能得到微小的效能改善。在我們這個判斷機制下，EM 演算法需要考慮下列兩條曲線：

1. 成本曲線：即連續未改善的世代數
2. 效能曲線：最大允許沒有明顯改善的世代數

第 n 個世代下，成本曲線 CN_n 的定義如下；其中 $\overline{I_n}$ 代表所有 I_γ 的平均值($1\le\gamma\le n$)。

$$CN_n = \begin{cases} CN_{n-1}+1 & if\ I_n < \overline{I_n} \\ 0, & otherwise \end{cases} \tag{11}$$

接下來，我們針對第 n 個世代下，效能曲線 MAN_n 的定義如下；其中 $\sigma(I_n)$ 代表所有 I_γ 的標準差；K 代表 PLSA 中潛在主題的總數；$\overline{\sigma(I_n)}$ 代表所 $\sigma(I_\gamma)$ 的平均值。

$$MAN_n = \left[I_n / \overline{I_n} \times K \times \sigma(I_n) / \overline{\sigma(I_n)} \right] \tag{12}$$

根據公式(12)的定義，效能曲線是基於下列兩個參數來動態決定：”改善歷史進度”（$I_n / \overline{I_n}$）與”變動歷史進度”（$\sigma(I_n) / \overline{\sigma(I_n)}$）。假設 $I_n / \overline{I_n}$ 或 $\sigma(I_n) / \overline{\sigma(I_n)}$ 的數字大的話，我們假設在概似估計函數（公式(5)）有很大的改善空間或很大的變動幅度。這隱含了兩個意義：(1)往後改善或不穩定的機會大；(2)當改善機會大或每個世代改善變動幅度大時，則距離區域最佳解的機率也會降低。

在我們的 EM 演算法終止條件的判斷上，我們使用 CN_n 及 MAN_n 曲線進行判斷。假設存在一個世代 n，使得 CN_n 的值大於 MAN_n，此時即終止 EM 演算法的運算。輔助說明請參照圖 3。

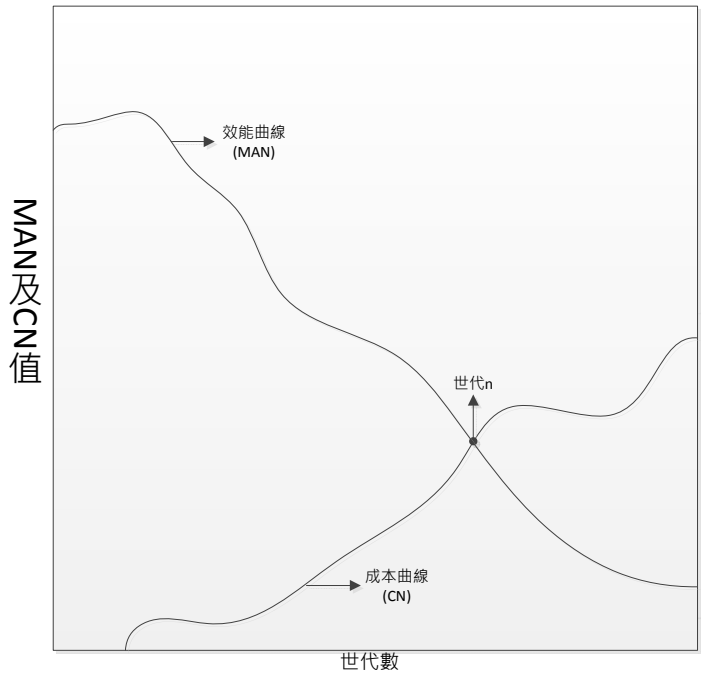


圖 3：成本及效能曲線分佈圖

肆、實驗分析

一、實驗資料

本研究針對使用者使用 Google Blog Search 時，經由輸入關鍵字後，所回傳之頁面結果，經過文件前置處理，作為部落格文件的資料來源。

本實驗所選擇之關鍵字來源為 Google 及 Yahoo 於 2012 及 2013 年前十個熱門搜尋關鍵字，其分別選自下列資料集：Google 2012 年度熱門關鍵字（2012）、Google 2013 年度熱門關鍵字（2013）、Yahoo 2012 年度熱門關鍵字（2012）、Yahoo 2013 年度熱門關鍵字（2013）。這些關鍵字分別為：“Amanda Todd”、“Amanda Bynes”、“BBB12”、“Boston Marathon”、“Cory Monteith”、“Diablo 3”、“Election”、“Gangnam Style”、“Hurricane Sandy”、“Harlem Shake”、“iPad 3”、“iPhone 5”、 “iPhone 5s”、“Jennifer Lopez”、“Jodi Arias”、“Justin Bieber”、“Kate Middleton”、“Kate Upton”、“Kim Kardashian”、“Lindsay Lohan”、“Michael Clarke Duncan”、“Miley Cyrus”、“Minecraft”、“Nelson Mandela”、“North Korea”、“Obamacare”、“Olympics 2012”、“Paul Walker”、“PlayStation 4”、“Political Polls”、“Royal Baby”、“Samsung Galaxy S4”、“Selena Gomez”、“Whitney Houston”。我們經由篩選的方式，將所有關鍵字

進行不重覆選取。所謂不重覆選取是指我們去除某些關鍵字在不同搜尋引擎或不同年度有重覆出現時，進行關鍵字選取時，我們只選取一筆。下列關鍵字具有重覆出現的特性：“iPhone 5”（出現在 Yahoo 2012(#2)及 Yahoo 2013(#9)）、“Kim Kardashian”（出現在Yahoo 2012(#3)及Yahoo 2013(#2)）、“Kate Upton”（出現在Yahoo 2012(#4)及 Yahoo 2013(#3)）、“Kate Middleton”（出現在 Google 2012(#6)及 Yahoo 2012(#5)）、“Whitney Houston”（出現在 Google 2012(#1)及 Yahoo 2012(#6)）、“Olympics 2012” （出現在 Google 2012(#7）及 Yahoo 2012(#7)）。

我們將上述之關鍵字經由 Google Blog Search 進行搜尋，並將回傳文件中每一筆網頁標題和網頁摘要當做一篇部落格文件。由於 GoogleBlogSearch 在進行實際搜尋時，其能擷取的文件筆數大約在 400 筆以下（不同關鍵字所回傳的文件數量會有所些許差別），但至少都有 350 筆文件結果。因此對於每個關鍵字我們分別選取 10、20、40、80、160、240、350 筆文件當作實驗分析之資料來源。

二、評估指標

由於本研究將來源資料轉換為矩陣的格式，因此必須採用機器學習中的相似性度量來進行評估；主要是評估矩陣中向量之間的相似度，若特定語意模型相似度越高，則代表該語意模型效能愈好。本研究採用餘弦相似度（cosine similarity）及相關係數（correlation coefficient）作為實驗結果的評估指標。

（一）餘弦相似度

餘弦相似度的度量方式是相似度評估之研究領域始祖，餘弦相似度主要是以兩組相同基底（base）與維度（dimension）向量間的角度（angle）差距來度量兩向量之間的距離（distance），所計算的結果會介於 0 至 1 之間。當兩個向量間的角度差距越小時，計算結果就趨近於 1，即代表兩向量間的相似度越高；反之越趨近於 0，表示兩向量相似度越低。例如，在二維空間中有兩個向量 A（$X_1, \ldots, X_n$）和 B（$Y_1, \ldots, Y_n$），其餘弦相似度計算公式如下所示。

$$COS(A,B) = \frac{\sum(X_i \times Y_i)}{\sqrt{\sum X_i^2} \times \sqrt{\sum Y_i^2}} \tag{13}$$

Tan、Steinbach 與 Kumar（2005）指出餘弦相似度為文件分類中，最常被用於度量文件間距離的方法；在本研究中，矩陣中的每個向量間餘弦相似度越高，則代表隱含的潛在主題效能越好。

（二）相關係數

在機率論與統計學中，相關係數可以顯示兩個隨機變數之間線性關係的強度和方向，在這個廣義的定義下，有許多根據資料特性的定義，以用來衡量資料相關的係數，最常用的是皮爾遜積差相關係數（Hofmann 2003）。其定義是兩個變數的共變異數除以兩個變數的標準差。

而在機器學習中的相似性度量方法，也有使用相關係數來測量兩向量之間的距離，公式如下所示：

$$\rho(A,B)=\frac{\sum(X_i-\bar{X})\times(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2\times\sum(Y_i-\bar{Y})^2}} \tag{14}$$

相關係數是衡量向量 A 與 B 相關程度的一種方法，相關係數的取值範圍是[-1,1]。相關係數的絕對值越大，則表明 A 與 B 相關度越高。當 A 與 B 線性相關時，相關係數取值為 1（正線性相關）或-1（負線性相關）。

接下來，我們以一個例子說明如何將語意模型所產生之矩陣，經由餘弦相似度及相關係數計算相似度。假設下表是經由語意模型所產生之矩陣內容。

表 13：語意模型之矩陣內容

	d_1	d_2	d_3
t_1	0.56	0.12	0.33
t_2	0.34	0.25	0.25
t_3	0.74	0.33	0.15

我們首先計算，字詞 t_1 向量（<0.56, 0.12, 0.33>）與 t_2 向量（<0.34, 0.25, 0.25>）之間之餘弦相似度，計算過程如下所示：

$$Cos(t_1,t_2)=\frac{0.56\times0.34+0.12\times0.25+0.33\times0.25}{\sqrt{0.56^2+0.12^2+0.33^2}\times\sqrt{0.34^2+0.25^2+0.25^2}}=0.93424$$

同樣地，我們可以計算 $Cos(t_1,t_3)=0.92443$ 及 $Cos(t_2,t_3)=0.91938$，最後我們將這三個數值進行平均，得到平均餘弦相似度為 0.92602。接下來，我們計算，字詞 t_1 向量與 t_2 向量之間之相關係數（其中 t_1 的平均值 $\bar{t_1}=0.34$、t_2 的平均值 $\bar{t_2}=0.28$），計算過程如下所示：

$$\rho = (t_1, t_2) = \frac{(0.56-0.34)\times(0.34-0.28)+(0.12-0.34)\times(0.25-0.28)+(0.33-0.34)\times(0.25-0.28)}{\sqrt{\left[(0.56-0.34)^2+(0.12-0.34)^2+(0.33-0.34)^2\right]\times\left[(0.34-0.28)^2\times(0.25-0.28)^2+(0.25-0.28)^2\right]}} = 0.87884$$

同樣地，我們可以計算$\rho(t_1,t_3)=0.69701$及$\rho(t_2,t_3)=0.95468$，最後我們將這三個數值進行平均，得到平均相關係數為0.84351。

三、潛在主題個數之設置

為了決定 LSA 和 PLSA 潛在主題個數的範圍，我們設計了一個實驗，藉由不同 K 值（潛在主題個數）的設定來觀察各個矩陣的效果。首先我們選取的 K 值範圍是由 Hofmann、Schölkopf 與 Smola（2008）建議，其範圍為 2、10、20、30、40、50、60、70、80、90、100，並採用文件數 160 的部落格文件矩陣，經過 LSA、PLSA 的運算後，求得各種矩陣在不同 K 值之下的餘弦相似度，以便決定本研究之潛在主題個數範圍。不過由於 PLSA 是經由 EM 演算法求得，它的機率值不會是一個固定的數值，因此在這個實驗中，我們針對各個矩陣反覆跑了五次的 PLSA 運算，藉此得到五組不同的 PLSA 結果，最後的餘弦相似度，則是這五組 PLSA 的平均值。

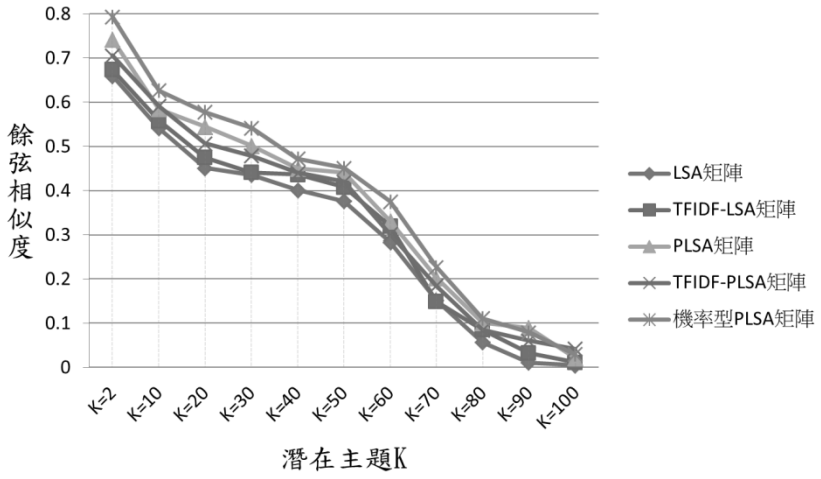


圖 4：不同 K 值下的各種矩陣餘弦相似度

圖 4 為經由代入不同潛在主題個數 K，計算矩陣的餘弦相似度的結果。由該圖可以看出，各類型矩陣皆在 K 值越小的情況下得到愈好的餘弦相似度，這是必然的結果，因為潛在主題的個數越少，LSA 和 PLSA 所去除的不明確、模糊的語意資訊程度越高，故矩陣餘弦相似度越高，但由於去除過多語意資訊可能會導致語意資訊不完全。當 K>60 時，各個矩陣的餘弦相似度大幅度下降，因為保留過多

的不明確、模糊的語意資訊，因此矩陣的餘弦相似度隨著 K 值越大，下降越多。參考上述的實驗，我們決定將潛在主題個數 K 的實驗範圍設置為餘弦相似度較穩定的 K=2～50；在接下來的實驗，潛在主題個數均設定在這個範圍之中。

四、實驗結果

在本實驗之中，我們討論兩種實驗數據：(1)使用語意模型能提昇多少效能以及使用不同語意模型之效能比較；(2) EM 演算法終止條件之成本效能解驗證。

（一）不同語意模型之驗證

為了詳細說明同義詞及一詞多義對不同語意模型的影響，我們以輸入查詢"mobile phone jb"進行比較。表 14 顯示不同模型之餘弦相似度及相關係數。觀察這個表格的結果，語意模型（LSA 及 PLSA 矩陣）的餘弦相似度及相關係數明顯優於非語意模型（VSM 及 TFIDF 矩陣）。

表 14：當輸入查詢為"mobile phone jb"時，不同模型下之餘弦相似度及相關係數

(a) VSM 矩陣						(b) TFIDF 矩陣					
	d_1	d_2	d_3	d_4	d_5		d_1	d_2	d_3	d_4	d_5
mobile phone	2	2	3	2	0	mobile phone	0	0	0	0	0
jb	0	3	0	2	3	jb	0	0.67	0	0.44	0.66
jailbreaking	3	0	3	0	2	jailbreaking	0.67	0	0.67	0	0.44
cellular phone	0	5	2	1	3	cellular phone	0	0.48	0.19	0.10	0.29
jelly bean	0	2	0	4	0	jelly bean	0	0.80	0	1.59	0
餘弦相似度=0.51163 相關係數=0.44062						餘弦相似度=0.48257 相關係數=0.40253					
LSA 矩陣						PLSA 矩陣					
	d_1	d_2	d_3	d_4	d_5		d_1	d_2	d_3	d_4	d_5
mobile phone	0.53	0.47	0.28	0.38	0.61	mobile phone	0.38	0.22	0.35	0.37	0.27
jb	0.25	0.31	0.42	0.36	0.19	jb	0.32	0.35	0.27	0.29	0.31
jailbreaking	0.28	0.52	0.33	0.28	0.22	jailbreaking	0.45	0.37	0.40	0.43	0.34
cellular phone	0.46	0.43	0.24	0.37	0.57	cellular phone	0.43	0.36	0.44	0.40	0.35
jelly bean	0.32	0.37	0.52	0.54	0.25	jelly bean	0.38	0.29	0.37	0.41	0.34
餘弦相似度 =0.91671 相關係數 =0.66927						餘弦相似度 =0.98961 相關係數 =0.70190					

接下來，我們分析其中原因。觀察表格中的資料，我們發現字詞"mobile phone"與"cellular phone"具有同義詞特性；同樣地，字詞"jb"與"jailbreaking"或"jelly bean"同樣也具有同義詞特性（即"jailbreaking"或"jelly bean"都可以縮寫成"jb"）。在這個實驗之中，LSA 的餘弦相似度及相關係數明顯提昇至 0.91671 及 0.66927。亦即經由 LSA 的去除雜訊處理，我們可以將文件中具有相同主題之同義字詞提昇其重要性，進而提昇檢索效能。

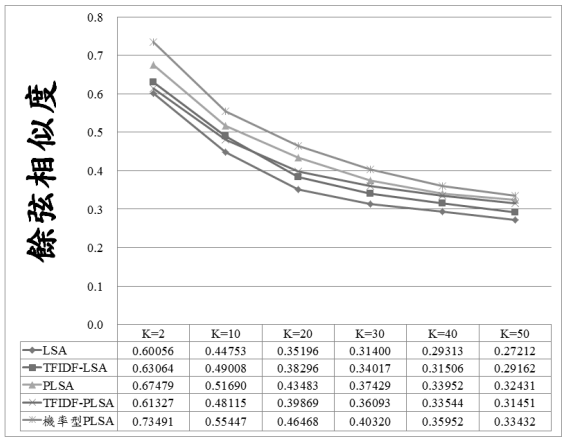
雖然字詞"jb"與"jailbreaking"或"jelly bean"具有同義詞特性，然而"jb"同時也包含"jailbreaking"及"jelly bean"兩種不同意思，即一詞多義特性。觀察 PLSA 的結果，我們發現 PLSA 比 LSA 來得優良，其中的餘弦相似度及相關係數再進一步的提昇至 0.98961 及 0.70190。這個效能的提昇主要是因為 PLSA 可以進一步的處理一詞多義，這個結果也呼應了 Ishida 與 Ohta（2002）的觀察，即 LSA 及 PLSA 對於同義詞處理能力都極為良好，然而 LSA 中的 SVD 分解後，每個列向量只能表示一個字詞，因此其欠缺處理一詞多義的能力。然而 PLSA 使用生成模型計算字詞與文件間共同出現之潛在機率，透過潛在機率的呈現方式，我們可以清楚的區別字詞間不同意義及型態，進而達成一詞多義的處理。

接下來，我們使用較大資料集進行實驗。針對上述所選取之 34 個關鍵字，使用 Google Blog Search 進行搜尋，並將回傳之文件經由前置處理後所產生之 VSM 矩陣進行不同語意模型之比較。我們比較的語意模型分別為原始 LSA、原始 PLSA、TFIDF-LSA、TFIDF-PLSA、機率型 PLSA，並以餘弦相似度及相關係數進行比較。

首先，我們以不同潛在主題的角度來觀察餘弦相似度及相關係數，圖 5-(a)及(b)為這兩個指標的結果，圖中的數值為不同文件數（10、20、40、80、160、240、350）之平均值；由圖 5-(a)及(b)可得知不同的語意模型都呈現類似的趨勢，隨著 K 值（潛在主題個數）越來越大，相似度越低。LSA 的餘弦相似度和相關係數是最低的，而機率型 PLSA 則為最好的，機率型 PLSA 在 K=2 時和 LSA 的差距最大，餘弦相似度差距為 0.13435，相關係數差距為 0.28096。各種語意模型的差距會隨著 K 值的增加，語意模型相似度的差距越來越小。然而，當 TFIDF-PLSA 在 K=2、K=10 時，其效能沒有比 TFIDF-LSA 還優良，但在 K>10 之後效能都略優於 TFIDF-LSA，這說明潛在主題個數 K 愈大時，PLSA 方法有更好的效能。觀察圖 5-(a)及(b)後，我們發現 TFIDF-PLSA 整體效能並未比 PLSA 來得好，其原因在於熱門關鍵字在 Google Blog Search 內的回傳筆數都相當龐大，每個關鍵字都超過百萬筆回傳結果，這將使得 TFIDF 數值中的 IDF（逆文件頻率）數值偏高，產生的 TFIDF 矩陣內的數值也較大，經過 PLSA 的運算後提昇的相似度並未比使用原始 VSM 矩陣的效能還要好。

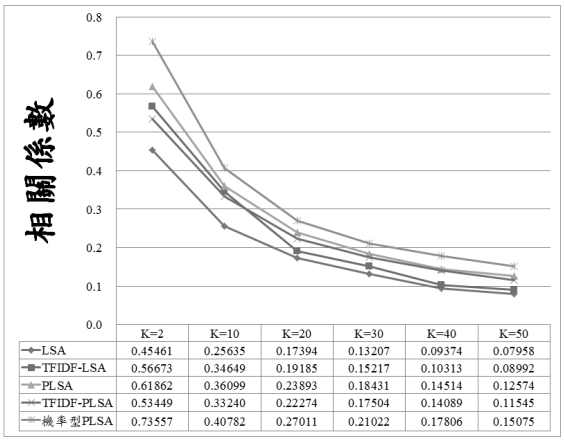
接下來，我們以不同文件數的角度來觀察兩種評估指標的結果，圖 5-(c)及(d)

分別為餘弦相似度及相關係數的結果，圖中數值為所有 K 值（2、10、20、30、40、50）的平均值。觀察圖 5-(c)及(d)後，我們發現各種語意模型的餘弦相似度、相關係數會隨著文件數的增加而上升，證實了語意模型在部落格文件數越大的情況下，相似度提昇越明顯。根據圖 5 的所有結果，PLSA 及各種加權過的 PLSA 語意模型應用於部落格文件分析上會比 LSA 及 TFIDF-LSA 有更好的餘弦相似度，這代表經由 PLSA 處理後，文件能夠獲得較佳之相似性，這代表 PLSA 具有較佳的相似語意處理能力。



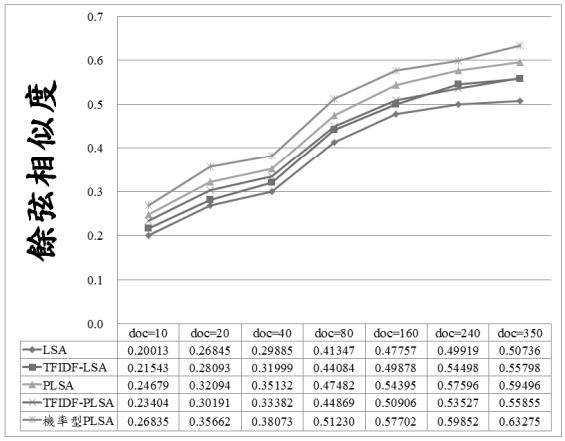
	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.60056	0.44753	0.35196	0.31400	0.29313	0.27212
TFIDF-LSA	0.63064	0.49008	0.38296	0.34017	0.31506	0.29162
PLSA	0.67479	0.51690	0.43483	0.37429	0.33952	0.32431
TFIDF-PLSA	0.61327	0.48115	0.39869	0.36093	0.33544	0.31451
機率型PLSA	0.73491	0.55447	0.46468	0.40320	0.35952	0.33432

（a）不同潛在主題個數下之餘弦相似度



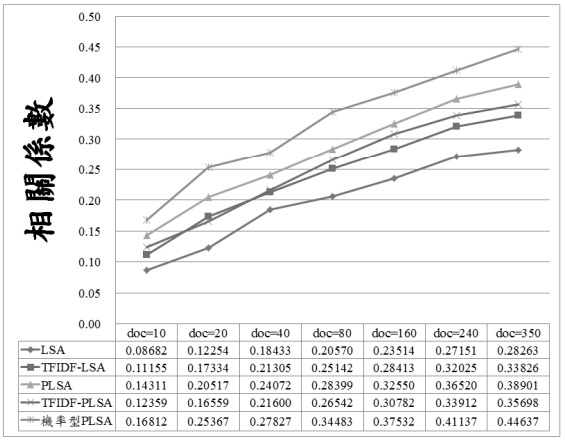
	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.45461	0.25635	0.17394	0.13207	0.09374	0.07958
TFIDF-LSA	0.56673	0.34649	0.19185	0.15217	0.10313	0.08992
PLSA	0.61862	0.36099	0.23893	0.18431	0.14514	0.12574
TFIDF-PLSA	0.53449	0.33240	0.22274	0.17504	0.14089	0.11545
機率型PLSA	0.73557	0.40782	0.27011	0.21022	0.17806	0.15075

（b）不同潛在主題個數下之相關係數



	doc=10	doc=20	doc=40	doc=80	doc=160	doc=240	doc=350
LSA	0.20013	0.26845	0.29885	0.41347	0.47757	0.49919	0.50736
TFIDF-LSA	0.21543	0.28093	0.31999	0.44084	0.49878	0.54498	0.55798
PLSA	0.24679	0.32094	0.35132	0.47482	0.54395	0.57596	0.59496
TFIDF-PLSA	0.23404	0.30191	0.33382	0.44869	0.50906	0.53527	0.55855
機率型PLSA	0.26835	0.35662	0.38073	0.51230	0.57702	0.59852	0.63275

（c）不同文件數下之餘弦相似度



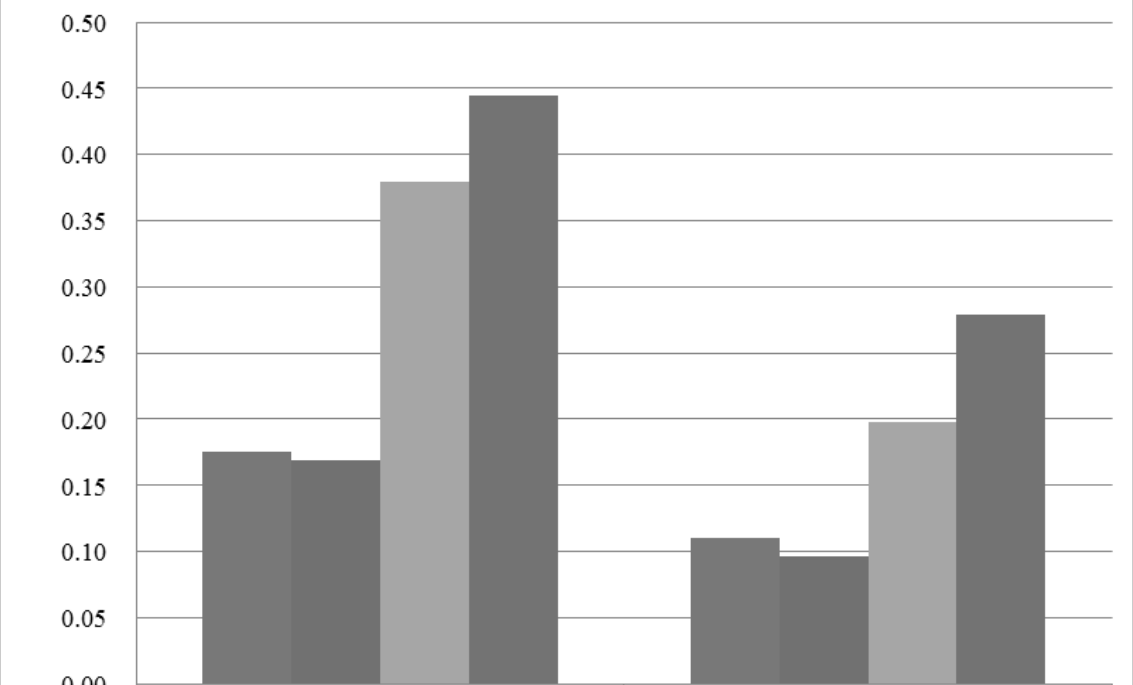
	doc=10	doc=20	doc=40	doc=80	doc=160	doc=240	doc=350
LSA	0.08682	0.12254	0.18433	0.20570	0.23514	0.27151	0.28263
TFIDF-LSA	0.11155	0.17334	0.21305	0.25142	0.28413	0.32025	0.33826
PLSA	0.14311	0.20517	0.24072	0.28399	0.32550	0.36520	0.38901
TFIDF-PLSA	0.12359	0.16559	0.21600	0.26542	0.30782	0.33912	0.35698
機率型PLSA	0.16812	0.25367	0.27827	0.34483	0.37532	0.41137	0.44637

（d）不同文件數下之相關係數

圖 5：不同模型下之效能評比

最後，我們驗證語意模型提昇傳統部落格搜尋的檢索效能。在這個實驗之中，我們採用之語意模型為原始 LSA 及 PLSA。圖 6 的結果為不同模型之餘弦相似度及相關係數評比：文件前置處理所得到的 VSM 矩陣、TFIDF 矩陣、LSA 矩陣以及 PLSA 矩陣；其中 VSM 及 TFIDF 矩陣的結果為原始部落格搜尋（非語意模型）之

結果。由圖中得知，經過 LSA 或 PLSA 語意模型實行之部落格文件，不論餘弦相似度及相關係數，皆能有一定程度之提昇。這代表著，當我們針對部落格搜尋引擎回傳之部落格文件使用語意模型進行後置處理，可以提昇整體部落格檢索的效能；亦即能夠針對部落格搜尋引擎所回傳之部落格文件進行重新排序，而重新排序的結果會加上同義詞及一詞多義的特性。這樣的好處可以將具有語意相關之潛在文件（這部份文件的原始排名可能非常後面）顯示於較前面的搜尋結果，以便方便使用者閱讀。



	餘弦相似度	相關係數
VSM	0.17595	0.11004
TFIDF	0.16859	0.09666
LSA	0.37988	0.19838
PLSA	0.44411	0.27896

圖 6：語意模型與非語意模型之比較

（二）EM 演算法終止條件之驗證

在這個實驗之中，我們想要驗證我們所提出之 EM 演算法終止條件是否可以達到成本效能解。我們隨機產生下列參數：M（字詞的總數）、N（文件的總數）、K（潛在主題個數）以及 VSM 矩陣。有興趣的讀者可以至下列網址進行模擬：http://pwsc.sytes.net/simulation/stopping_em.php。其餘參數包括：n（世代數）、I_n（兩

個連續世代的改善值）、$\overline{I_n}$（所有 I_γ的平均值（$1\le\gamma\le n$））、$\sigma(I_n)$（所有 I_γ的標準差）、$\overline{\sigma(I_n)}$（所有$\sigma(I_\gamma)$的平均值）、MAN_n（效能曲線）、CN_n（成本曲線）以及 $L_n(d_i,w_j)$（最大概似估計函數值），可以參照前面之描述。

圖 7 顯示一個模擬結果的例子，為了方便說明 EM 演算法的終止條件，我們定義圖中的 Y 軸為 MAN_n、CN_n 及 $L_n(d_i,w_j)$。在第一個終止條件（即收斂於區域最佳解），我們設定公式 10 的λ參數為 0.01，這代表當兩個世代的改善值小於等於一個極小的數值λ時，則此終止條件即達成。在圖 7 這個模擬之中，區域最佳解出現在第 77 個世代(n=77)，在這個世代我們所求得的最大概似估計函數值為 537.459（$L_n(d_i,w_j)$=537.459）。在第二個終止條件（即達到最大允許執行世代數），我們判斷最大允許執行世代數為 29（n=29），在這個世代之中，我們發現連續未改善的世代數（CN_n=21）大於最大允許沒有明顯改善的世代數（MAN_n=17），在這個世代我們所求得的最大概似估計函數值為 515.47（$L_n(d_i,w_j)$=515.47）。在圖 7 這個模擬之中，EM 演算法應該終止於第 29 個世代，而不是第 77 個世代，其原因在於我們額外增加 48（77-29）個世代，$L_n(d_i,w_j)$只能增加 21.889（537.459–515.47）。這代表當 EM 演算法執行世代數超過第二個終止條件時，我們將要花費許多的計算資源，才能得到些許的改進。

為了驗證我們的第二個終止條件可以達成成本效能解，我們針對第二個終止條件所產生之動態世代數與某些預設的世代數進行模擬，並以成本-效能比（cost performance ration, CPR）進行比較，我們採用的模擬次數為 1000 次，預設的世代數分別為“n=20”、“n=40”、“n=60”、“n=80”以及“n=100”。

在 CPR 指標裡，我們首先需要定義第 n 個世代所達成的效能率（performance rate, PR）指標 PR_n，其公式如下所示；其中 $L_n(d_i,w_j)$為第 n 個世代所實際達到的最大概似估計函數值、$L_{optimal}(d_i,w_j)$為區域最佳解所達到的最大概似估計函數值。

$$PR_n = \frac{L_n(d_i, w_j)}{L_{optimal}(d_i, w_j)} \tag{15}$$

圖 8-(a)是 1000 次模擬所產生的 PR 分布，為了簡化圖形的複雜度，圖 8-(a)及(b)中的每個點是將 50 次模擬結果進行平均。不同 n 的平均 PR 值分別為 0.896464(n=20)、0.986531(n=40)、0.998265(n=60)、0.999788(n=80)、0.999984(n=100)以及 0.959145（n=第二個終止條件）。

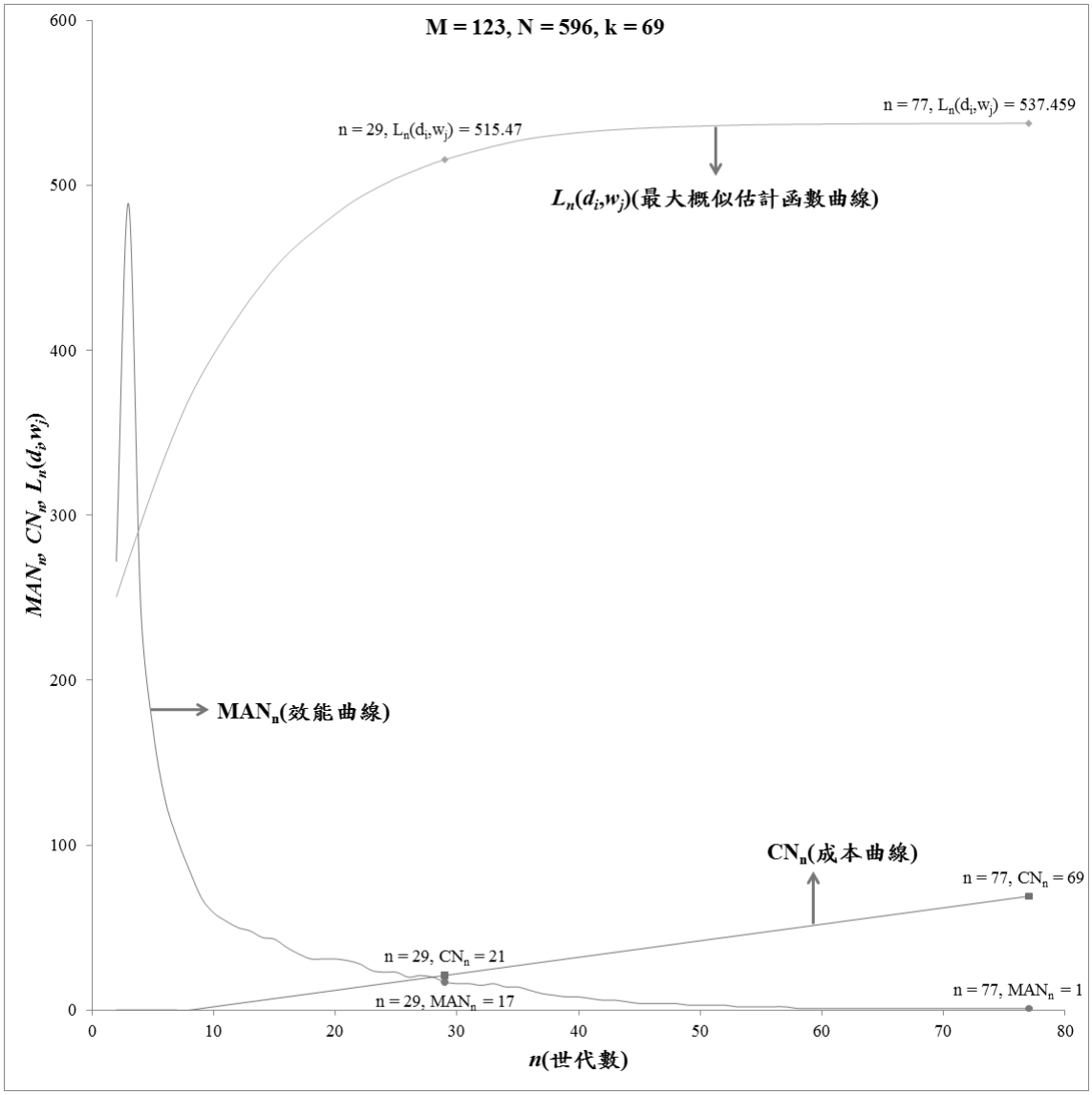


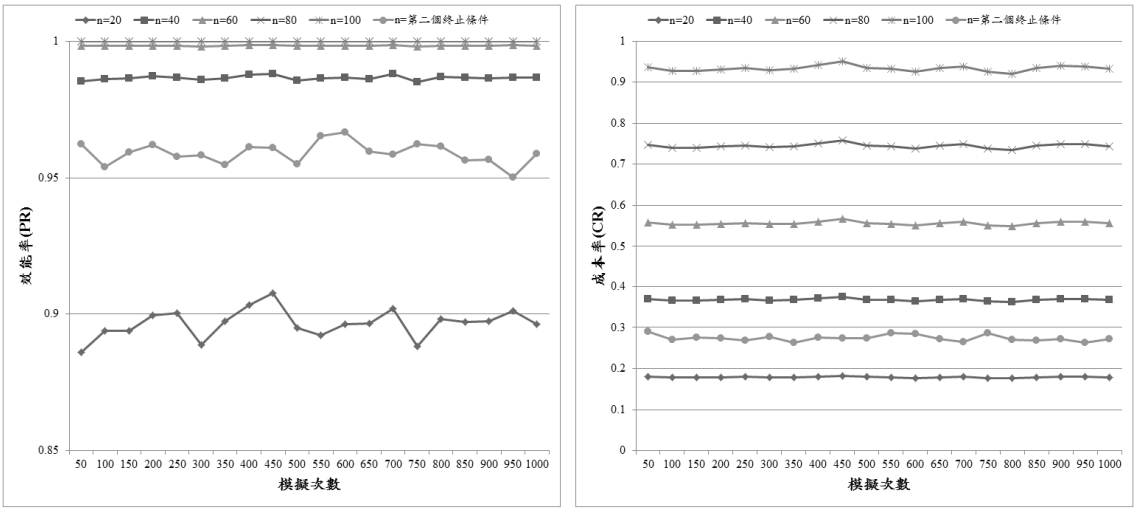
圖 7：EM 演算法終止條件模擬結果

在 CPR 指標裡，我們接下來需要定義第 n 個世代所達成的成本率（cost rate, CR）指標 CR_n，其公式如下所示；其中 RQN_n 為第 n 個世代所實際需要的世代數、$RQN_{optimal}$ 為區域最佳解所實際需要的世代數。

$$CR_n = \frac{RQN_n}{RQN_{optimal}} \tag{16}$$

較低的 CR 值代表使用較少的計算資源，然而較低的 CR 值也會造成較低的 PR 值。圖 8-(b)是 1000 次模擬所產生的 CR 分布。不同 n 的平均 CR 值分別為

0.179098(n=20)、0.367623(n=40)、0.556147(n=60)、0.744672(n=80)、0.933196(n=100)以及 0.274129（n=第二個終止條件）。



（a）1000次模擬所產生的PR分布　　（b）1000次模擬所產生的CR分布

圖 8：1000 次模擬所產生的 PR 及 CR 分布

對於兩個可比較的世代 n1 及 n2，其中 n1>n2，我們定義 $CPR_{n1,n2}$ 為”效能率增加值”（PR_{n1}-PR_{n2}）與”成本率增加值”（CR_{n1}-CR_{n2}）的比值，公式如下所示：

$$CPR_{n1,n2} = \frac{PR_{n1} - PR_{n2}}{CR_{n1} - CR_{n2}} \tag{17}$$

我們以 n2=20 當成評比的基準，表 15 顯示不同 n1 下之 $CPR_{n1,20}$ 的結果。根據表 15 的結果顯示，我們發現”n1=第二個終止條件”明顯優於其它的 n1，其結果至少優於 18.1843%（0.659590-0.477747）。當超過”n1=第二個終止條件”時，CPR 下降的幅度非常顯著。因此，我們所提出之 EM 演算法終止條件所產生的結果具有成本效能解之特性。

表 15：使用 n2=20 下，不同 n1 之成本-效能比

	PR	CR	CPR
n2=20	0.896464	0.179098	
n1=40	0.986531	0.367623	0.477747
n1=60	0.998265	0.556147	0.269995

n1=80	0.999788	0.744672	0.182689
n1=100	0.999984	0.933196	0.137277
n1=system	0.959145	0.274129	0.659590

伍、結論與未來研究方向

根據實驗結果顯示，我們所提出的機率型 PLSA 矩陣，在所有的語意模型中取得最好的效能，亦即解決本論文的第一個研究目的（提出適合部落格文件檢索系統的語意模型）。將原始的 VSM 和 TFIDF 模型增加 LSA 和 PLSA 語意模式後，亦能增加實驗效能，正如 LSA、PLSA 文獻所提，其可以解決同義詞及一詞多義問題，亦即解決本論文第二個研究目的（解決部落格文件的同義詞和一詞多義問題）。實驗結果顯示，經過語意模型分析過的部落格文件皆比未經語意模型來得優良，呼應了本論文第三個研究目的（使用語意模型後是否能提昇部落格文件檢索效能）。為了提昇 PLSA 的執行效能，我們使用成本效能的概念改進 PLSA 的執行時間，亦即完成本論文第四個研究目的。

由於 PLSA 使用 EM 演算法進行參數估計，然而 EM 演算法所需的執行時間相當耗時，為了解決這個問題，我們提出一套 EM 演算法終止條件判斷的機制，按照我們的機制所產生的結果具有成本效能解之特性。我們的 EM 演算法終止條件可以應用在以 EM 演算法為基礎的任何模型，如 PLSA、人工智慧領域中的分群、貝氏網路的學習、以及隱藏馬可夫等等模型。

未來研究方向主要著重在使用更完整的文件，考慮到 Google Blog Search 能擷取的網頁數量有一定的限制，所以本研究所使用的文件集略小，另外網頁摘要並非完整的文件，未來可針對更完整的文件進行分析。

誌謝

我們感謝二位匿名審稿委員提供寶貴意見，以便改善我們論文的品質。本文接受行政院科技部專題研究計畫（MOST 103-2221-E-259-023 & NSC 102-2410-H-259 -068）之補助研究經費，順利完成此篇著作之研究工作，謹此致謝。

參考文獻

陳林志、林育任（2013），『個人化的網頁摘要文件分群系統』，*資訊管理學報*，第二十卷，第一期，頁 97-130。

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society for*

Information Science, Vol. 41, No. 6, pp. 391-407.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977),'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, pp. 1-38.

DiNucci, D. (1999), 'Design & new media: fragmented future—Web development faces a process of mitosis, mutation, and natural selection', *Print*, Vol. 53, No. 4, pp. 32-35.

Dumais, S.T. (2005),'Latent semantic analysis', *Annual Review of Information Science and Technology*, Vol. 38, No. 1, pp. 188-230.

Evangelopoulos, N.E. (2013), 'Latent semantic analysis', *Wiley Interdisciplinary Reviews: Cognitive Science*, Vol. 4, No. 6, pp. 683-692.

Fox, C. (1989),'A stop list for general text', *SIGIR Forum*, Vol. 24, No. 1-2, pp. 19-21.

Gibson, S., Wills, A. and Ninness, B. (2005), 'Maximum-likelihood parameter estimation of bilinear systems', *IEEE Transactions on Automatic Control*, Vol. 50, No. 10, pp. 1581-1596.

Golub, G.H. and Reinsch, C. (1970), 'Singular value decomposition and least squares solutions', *Numerische Mathematik*, Vol. 14, No. 5, pp. 403-420.

Google (2012), 'Google Zeitgeist 2012', available at http://tinyurl.com/mc2f9nf (accessed 8 May 2014).

Google (2013), 'Google Zeitgeist 2013', available at http://tinyurl.com/kubnvvg (accessed 8 May 2014).

Google (2014), 'Blogger: Blogger Dashboard', available at https://www.blogger.com/home (accessed 8 May 2014).

Hazel, P. (1997), 'PCRE - Perl compatible regular expressions', available at http://www.pcre.org/pcre.txt (accessed 8 May 2014).

Hennig, L. (2009), 'Topic-based multi-document summarization with probabilistic latent semantic analysis', *Proceedings of the Seventh International Conference on Recent Advances in Natural Language Processing(RANLP 2009)*, Borovets, Bulgaria, September 14-16, pp. 144-149.

Hofmann, T. (1999), 'Probabilistic latent semantic indexing', *Proceedings of the Twenty-second Annual International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval (SIGIR 1999)*, Berkeley, CA, USA, August 15-19, pp. 50-57.

Hofmann, T. (2003), 'Collaborative filtering via gaussian probabilistic latent semantic analysis', *Proceedings of the Twenty-sixth Annual International ACM Special*

Interest Group on Information Retrieval Conference on Research and Development in Informaion Retrieval (SIGIR 2003), Toronto, Canada, July 28-August 1, pp. 259-266.

Hofmann, T., Schölkopf, B. and Smola, A.J. (2008), 'Kernel methods in machine learning', *The Annals of Statistics*, Vol. 36, No. 3, pp. 1171-1220.

Inoue, M. (2005), 'The remarkable search topic-finding task to share success stories of cross-language information retrieval', *Proceedings of the Fifth Workshop on Important Unresolved Matters*, Ann Arbor, Michigan, USA, June 29-30, pp. 61-64.

Ishida, K. and Ohta, T. (2002), 'An approach for organizing knowledge according to terminology and representing it visually', *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, Vol. 32, No. 4, pp. 366-373.

Jeong, O.-R. and Oh, J. (2012), 'Social community based blog search framework', *Lecture Notes in Computer Science*, Vol. 7240, No. 2012, pp. 130-141.

Judicibus, D.D. (2008), 'World 2.0', available at http://tinyurl.com/lfl4l4a (accessed 8 May 2014).

Kanejiya, D., Kumar, A. and Prasad, S. (2003), 'Automatic evaluation of students' answers using syntactically enhanced LSA', *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications using Natural Language Processing-Volume 2(HLT-NAACL 2003)*, Edmonton, Canada, May 27-June 1, pp. 53-60.

Keikha, M., Crestani, F. and Carman, M.J. (2013),'Searching blog sites with product reviews', *Lecture Notes in Computer Science*, Vol. 8018, No. 2013, pp. 495-500.

Kim, J.-H., Yoon, T.-B., Kim, K.-S. and Lee, J.-H. (2008), 'Trackback-rank: an effective ranking algorithm for the blog search', *Proceedings of the Second International Symposium on Intelligent Information Technology Application - Volume 03 (IITA 2008)*, Shanghai, China, December 20-22, pp. 503-507.

Klein, R., Kyrilov, A. and Tokman, M. (2011), 'Automated assessment of short free-text responses in computer science using latent semantic analysis', *Proceedings of the Sixteenth Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCES 2011)*, Darmstadt, Germany, June 27-29, pp. 158-162.

Kunder, M.d. (2008), 'The size of the world wide web', available at http://worldwidewebsize.com/ (accessed 8 May 2014).

Kuo, F.-F., Shan, M.-K. and Lee, S.-Y. (2013), 'Background music recommendation for video based on multimodal latent semantic analysis', *Proceedings of the IEEE International Conference on Multimedia and Expo (IEEE-ICME 2013)*, San Jose,

CA, USA, July 15-19, pp. 1-6.

Landauer, T.K., Foltz, P.W. and Laham, D. (1998), 'An Introduction to Latent Semantic Analysis', *Discourse Processes*, Vol. 25, No., pp. 259-284.

Landauer, T.K., McNamara, D.S., Dennis, S. and Kintsch, W. (2013), *Handbook of Latent Semantic Analysis*, Psychology Press, London, UK.

Lintean, M., Moldovan, C., Rus, V. and McNamara, D. (2010), 'The role of local and global weighting in assessing the semantic similarity of texts using latent semantic analysis', *Proceedings of the Twenty-third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)*, Daytona Beach, Florida, May 19-21, pp. 235-240

Luh, C.-J., Yang, S.-A. and Huang, D.T.-L. (2012), 'Estimating search engine ranking function with latent semantic analysis and a genetic algorithm', *Proceedings of the Third International Conference on E-Business and E-Government (ICEE 2012)*, Shanghai, China, May 11-13, pp. 439-442.

McInerney, J., Rogers, A. and Jennings, N.R. (2012), 'Improving location prediction services for new users with probabilistic latent semantic analysis', *Proceedings of the Fourteenth ACM Conference on Ubiquitous Computing (Ubicomp 2012)*, Pittsburgh, PA, USA, September 5-8, pp. 906-910.

Merholz., P. (1999), 'Peterme.com', available at http://tinyurl.com/ya77on (accessed 8 May 2014).

Mesaros, A., Heittola, T. and Klapuri, A. (2011), 'Latent semantic analysis in sound event detection', *Procedding of the Nineteenth European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, August 29-September 2, pp. 1307-1311.

Metaxoglou, K. and Smith, A. (2007), 'Maximum likelihood estimation of VARMA models using a stage-space EM algorithm', *Journal of Time Series Analysis*, Vol. 28, No. 5, pp. 666-685.

Mishne, G. and Rijke, M.d. (2006), 'A study of blog search', *Lecture Notes in Computer Science*, Vol. 3936, No. 1, pp. 289-301.

Myspace (2014), 'Featured Content on Myspace', available at https://myspace.com/ (accessed 8 May 2014).

Nardi, B.A., Schiano, D.J., Gumbrecht, M. and Swartz, L. (2004), 'Why we blog', *Communications of the ACM*, Vol. 47, No. 12, pp. 41-46.

Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N. and Siegwart, R. (2007), 'A comparison of line extraction algorithms using 2D range data for indoor mobile

robotics', *Autonomous Robots*, Vol. 23, No. 2, pp. 97-111.

O'Reilly, T. (2005), 'What is web 2.0: design patterns and business models for the next generation of software', available at http://tinyurl.com/nx36fj (accessed 8 May 2014).

Ozsoy, M.G., Alpaslan, F.N. and Cicekli, I. (2011), 'Text summarization using latent semantic analysis', *Journal of Information Science*, Vol. 37, No. 4, pp. 405-417.

Pernkopf, F. and Bouchaffra, D. (2005), 'Genetic-Based EM Algorithm for Learning Gaussian Mixture Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp. 1344-1348.

Porter, M.F. (1980), 'An algorithm for suffix stripping', *Program*, Vol. 14, No. 3, pp. 130-137.

Ristad, E.S. and Yianilos, P.N. (1998), 'Learning string-edit distance', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 522-532.

Search, G.B. (2014), 'Google Blog Search', available at http://www.google.com/blogsearch (accessed 8 May 2014).

Selberg, E. and Etzioni, O. (1997), 'The metacrawler architecture for resource aggregation on the web', *IEEE Expert*, Vol. 12, No. 1, pp. 11-14.

Sysomos (2014), 'Sysomos: business intelligence for social media', available at http://www.sysomos.com/ (accessed 8 May 2014).

Takama, Y., Kajinami, T. and Matsumura, A. (2005),'Blog search with keyword map-based relevance feedback', *Lecture Notes in Computer Science*, Vol. 3614, No. 2005, pp. 1208-1215.

Tan, P.-N., Steinbach, M. and Kumar, V. (2005), *Introduction to Data Mining*, Addison-Wesley Press, Boston, Massachusetts, USA.

Technorati (2014), 'Technorati', available at http://technorati.com/ (accessed 8 May 2014).

Wen, X.-B., Zhang, H. and Jiang, Z.-T. (2008), 'Multiscale unsupervised segmentation of SAR imagery using the genetic algorithm', *Sensors*, Vol. 8, No. 3, pp. 1704-1711.

Wikipedia (2013), 'History of blogging', available at http://tinyurl.com/792l8ve (accessed 8 May 2014).

Wikipedia (2014), 'Google blog search', available at http://tinyurl.com/gujaa (accessed 8 May 2014).

WordPress (2014), 'WordPress.com-get a free blog here', available at http://wordpress.

com/ (accessed 8 May 2014).

Wyner, A. and Engers, T.v. (2010), 'A framework for enriched, controlled on-line discussion forums for E-government Policy-Making', *Proceedings of Ongoing Resarch and Pojects of IFIP eGOV and ePart 2010*, pp. 357-366.

Xu, J., Ye, G., Wang, Y., Herman, G., Zhang, B. and Yang, J. (2009), 'Incremental EM for probabilistic latent semantic analysis on human action recognition', *Proceddings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2009)*, Genova, Italy, September 2-4, pp. 55-60.

Yahoo (2012), 'Yahoo!'s year in review reveals the daily search habits of 2012', available at http://tinyurl.com/kw47q8r (accessed 8 May 2014).

Yahoo (2013), '2013 year in review', available at http://tinyurl.com/q3zlabr (accessed 8 May 2014).

Zeng, Z., Zhang, S., Li, H., Liang, W. and Zheng, H. (2009), 'A novel approach to musical genre classification using probabilistic latent semantic analysis model', *Proceedings of the IEEE international conference on Multimedia and Expo (ICME 2009)*, New York, NY, USA, June 28-July 03, pp. 486-489.

Zhang, Q. and Goldman, S.A. (2001), 'EM-DD: An improved multiple-instance learning technique', *Neural Information Processing Systems*, Vol. 14, No., pp. 1073-1080.

Zhanga, J. and Gong, S. (2010), 'Action categorization by structural probabilistic latent semantic analysis', *Computer Vision and Image Understanding*, Vol. 114, No. 8, pp. 857-864.

Zhu, L., Sun, A. and Choi, B. (2008), 'Online spam-blog detection through blog search', *Proceedings of the Seveteenth ACM Conference on Information and Knowledge Management (CIKM 2008)*, Napa Vally, CA, USA, October 26-30, pp. 1347-1348.