陳林志、葉國暉、陳大仁、陳冠瑜（2017），『基於時間參數提昇谷歌部落格搜尋引擎效能』，*中華民國資訊管理學報*，第二十四卷，第二期，頁 155-184。

基於時間參數提昇谷歌部落格搜尋引擎效能

陳林志*
國立東華大學資訊管理學系

葉國暉
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

陳冠瑜
國立東華大學資訊管理學系

摘要

部落格搜尋引擎是一種類似於谷歌的搜尋引擎，因為它們會自動收集來自網路上大量的資訊，並利用免費的介面讓一般人能搜索它們的資料庫。兩者之間的差異在於，部落格搜尋引擎主要是針對部落格進行索引並篩選掉一般的網頁，這個功能讓部落格搜尋引擎增加了一些特殊和獨特性。首先，每個部落格都有一個發佈日期，而部落格搜尋引擎可以顯示文章的發佈日期，相比一般搜尋引擎只能顯示最後更新日期，有時這些日期卻是不可靠的。其次，部落格搜尋引擎能抓取部落格文章發佈日期，相較於一般的搜尋引擎雖然有進階的搜索選項可以顯示日期，但這些都僅限於網頁的最後修改日期。

本論文中，我們使用四種語意模型分析谷歌部落格引尋引擎：潛在語意分析（LSA）、機率潛在語意分析（PLSA）、潛在狄利克里分配（LDA）、關係主題模型（RTM）。另外，我們提出一個利用時間參數來改良 RTM 的變形模型。根據實驗的結果，改良的 RTM 模型結合時間參數能提高谷歌部落格引擎效能。

關鍵詞：潛在語意分析、機率潛在語意模型、潛在狄利克里分配、關係主題模型、谷歌部落格搜尋

* 本文通訊作者。電子郵件信箱：lcchen@mail.ndhu.edu.tw
2015/05/13 投稿；2015/10/19 修訂；2016/04/19 接受

Chen, L.C., Yeh, K.H., Chen, D.R. and Chen, G.Y. (2017), 'Improving the performance of Google blog search based on the time parameter', *Journal of Information Management*, Vol. 24, No. 2, pp. 155-184.

Improving the Performance of Google Blog Search Based on the Time Parameter

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

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

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

Guan-Yu Chen
Department of Information Management, National Dong Hwa University

Abstract

Purpose－Blog search engines are similar to web search engines like Google in that they automatically gather large quantities of information from the web and give a free interface to allow the public to search their databases.

Design/methodology/approach－In this paper, we use four kinds of semantic models to analyze Google blog search engine: Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), and Relational Topic Model (RTM).

Findings－According to the result of experiment, our modified RTM's model can effectively combine the time parameter to Google blog search engine.

Research limitations/implications－The main difference between the two is that

* Corresponding author. Email: lcchen@mail.ndhu.edu.tw
2015/05/13 received; 2015/10/19 revised; 2016/04/19 accepted

blog search engines mainly index blogs and ignore the rest of the web. The special features of blogs give blog search engines some specific and unique attributes.

Practical implications – First, since each blog posting is dated, blog search engines can reported the date at which the posting was created. For normal web pages, search engines can only report the last updated date, and this is often not very reliable. Second, many blog search engines have a date-specific search capability. Again, some general search engines have this as an advanced search option, but only for the last modified date of pages.

Originality/value – In this paper, we propose a variant of RTM, which mainly focuses on the time parameter.

Keywords: latent semantic analysis, probabilistic latent semantic analysis, latent dirichlet allocation, relational topic model, Google blog search

壹、緒論

根據創市際發佈的分析數據（創世紀 2014），2014 年 10 月台灣有 1,353 萬位使用者透過桌上型電腦與筆電上網，總共花費 266 億 1000 萬分鐘在使用網路上，並且一共瀏覽 452 億 700 萬個網頁，平均每位使用者上網時間約為 1,966 分鐘、瀏覽 3,340 個網頁。所以許多私人公司以新聞的方式把部落格當作廣告增加能見度，而一般民眾則利用部落格來記錄自己的生活點滴，並通過寫作表達自己的心情（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；亦即部落格文件與傳統網頁文件最主要的差異在於呈現的方式為自由格式（Free format）（Jeong & Oh 2012）。

隨著科技進步，部落客們越來越常利用智慧型手機上網發表新的文章，根據痞客邦（台灣最大的部落格平臺）統計每天產出的文章數高達 35 萬篇，累計至今超過 3 億篇的文章和 4 億多的照片（余至浩 2014）。所以透過部落格搜尋引擎找出想要的文章就顯得相當重要，目前幾個知名的部落格搜尋引擎，例如 Google Blog Search、Bloglines 和 Technorati 等，其中 Google Blog Search 是目前最流行的部落格搜尋引擎。

部落格搜尋引擎主要索引的文件為部落格文件，由於部落格文件型態與一般網頁文件性質存在一定的差異，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。

由於部落格文章中的字詞（Term）和文件（Document）存在一層潛在的語意關係，我們稱之為主題（Topic），部落客在撰寫部落格文章時，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；另一方面，部落格文章的發佈時間也可能跟當時大家較常討論的主題極其相關。

潛在主題語意模型（文後以語意模型表示之）常被應用於尋找文件與字詞間所出現之潛在主題。近幾十年來，比較常被應用的語意模型包含：潛在語意主題（Latent Semantic Analysis, LSA）、機率潛在語意分析（Probabilistic Latent Semantic Analysis, PLSA）、潛在狄利克里分配（Latent Dirichlet Allocation, LDA)、關係主題模型（Relational Topic Model, RTM）。這些方法可以有效解決語

意分析中常見的同義詞及一詞多義問題，同時能透過多文件所存在之文件鏈結關係尋找適合的文件主題，然而這些語意模型並無針對文件內容進行相關文件主題之時間分類。由於現今網際網路的盛行，造成每日產生數目極其龐大的網頁文件，而這些文件內容的撰寫往往會與當下時間點發生之人、事、地、物有關，因此時間因素強烈的影響到文件主題分類效能的好壞。

為了解決上述(1)文件及字詞所共同出現的主題問題及(2)主題發佈時間所產生的時間問題。本研究利用多種語意模型，分析谷歌部落格搜尋引擎，並觀察不同語意模形是否能有效的發掘文件及字詞所存在之隱含主題關係，進而提升部落格搜尋引擎之查詢效能。同時，我們針對部落格文件與傳統網頁文件最主要的差別，自由格式之文件，我們選取最適合之語意模型，稱之為 RTM，進行後續處理。對於原始 RTM 語意模型，我們增加部落格文件之時間因素，以便符合部落格文件所存在之時間特性，並進而觀察增加時間因素是否能有效的增進部落格搜尋引擎之檢索效能。

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，最後對全文進行總結。

貳、文獻探討

一、部落格搜尋引擎

部落格的內容相較於一般網頁，文件類型比較偏向娛樂部分和個人意見陳述之自由格式文件（Fujimura et al. 2006; Jeong & Oh 2012），而近年來部落格文章數量急遽增加，網路使用者常利用部落格搜尋引擎找到所想要找的部落格文章（Kim & Yun 2014）。但是最近越來越多的垃圾部落格（Spam Blogs），企圖透過購買關鍵字的方法提升自身之排名，進而降低部落格搜尋引擎之精準度（Zhu et al. 2011）。

部落格搜尋引擎的設計不同於一般的搜尋引擎，他能搜尋部落格獨特的 HTML 結構，為了避免儲存過多的不必要資訊，部落格搜尋引擎會嘗試剖析部落格的結構，並儲存個別的部落格文章。雖然部落格搜尋引擎是一種有用的技術，但是依據選擇參數和發佈日期，是會影響到搜索精確度和效率（Thelwall & Hasler 2007）。

目前線上有幾個知名的部落格搜尋引擎 Technorati 和 Google Blog Search 等。Technorati 為 2002 年建立，其為全球第一個部落格搜尋引擎，可將部落客彼此之間的關係找出來，並建立特殊的部落格得分排名（Hearst et al. 2008）。從此而後，部落格應用領域日益發展，同時各個部落格搜尋引擎也被相繼開發出來，如表 1 所示；其中目前最受歡迎的平台為 Google Blog Search，其使用特殊的演算法

提高部落格文件之搜索精準度（Precision）及查全率（Recall）至 87%左右（Qureshi et al. 2011）。所以本研究以 Google Blog Search 為研究對象。

表 1：部落格搜尋引擎（Thelwall & Hasler 2007）

搜尋引擎名稱	內容	特色
Technorati	文章、標籤、部落格目錄	可查詢 top 100 熱門網誌
Bloglines	文章、摘要	可以添加額外搜索選項
Icerocket	部落格	簡潔、只有搜索框
Google Blog Search	部落格	使用特殊演算法提高準度

二、語意模型

目前最廣為人知的語意模型有 LSA、PLSA、LDA 以及 RTM。其中 LSA 在 90 年代後期被提出並應用於心理語言學上（Landauer et al. 1998）。LSA 是基於奇異值分解（Singular Value Decomposition, SVD）為基礎，利用 SVD 找出字詞對應文件中語意結構，從文件中發現隱含的語意，並在向量空間模型計算文件之間的語意關連度並降低樣本空間之維度，通過統計方法降低同義詞和一詞多義的影響，提高準確度。LSA 的優點在於把文件降維到一個低維度語意空間，減低一詞多義和同義詞的問題，但缺點在於 SVD 對於一詞多義的處理結果不甚理想。針對大量使用者文件以彙總形式呈現等問題，相關學者（Ozsoy et al. 2011; Yeh et al. 2005）利用 LSA 進行文件整理，達成文件彙總的目的；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；在音樂領域，有學者（Kuo et al. 2013; Logan et al. 2004）使用 LSA 分析歌詞及背景音樂所存在之雜訊問題；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；Cosma 與 Joy（2012）針對程式碼原始碼之抄襲偵測，其針對在所有收集資料集之中，使用 LSA 偵測某一特定程式碼在該資料集所存在之相同比例，抄襲的判斷主要是依據該比例與預設門檻值進行比較所計算而來。

PLSA 是由 Hofmann（1999）所提出。PLSA 定義了機率模型，使用兩層的機率對整個樣本空間建立模型，並使用期望值最大化（Expectation-Maximization, EM）演算法訓練潛在類別。EM 演算法包含了期望及最大化步驟，期望步驟主要透過已知參數計算潛在未知主題參數之期望值，最大化步驟則是嘗試將參數最大化，以求得模型的最佳解。PLSA 的優點在於：與 LSA 進行比較，其具有明確的統計學基礎，並有效解決同義詞和一詞多義的問題；然而，其缺點在於隨著文件和字詞的個數增加，矩陣變得越來越龐大，訓練參數值所花的 EM 演算法迭代時

間，隨著文件數增加而呈指數成長。PLSA 模型不只在語言分析上可應用，也應用於聲音解析方面（Mesaros et al. 2011）；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；透過 PLSA 分析重疊出現的聲音事件，並且也有應用於網頁探勘（Jin et al. 2004）；針對影像中人類的動作所可能產生的不同語意事件之偵測，Xu 等（2008; 2009）使用 PLSA 進行分群及事件之偵測。

LDA 是由 Blei、Ng 與 Jordan（2003）所提出。LDA 將每篇文件主題採用 Dirichlet 機率分佈形式給出，每份文件代表主題構成的機率分布，同時主題也是被所有文件所共享，一個主題可能包含很多的字詞，同時一份文件可能也是由不同主題所組合而成。LDA 生成模型中，M 篇文件會對應到 M 個獨立的 Dirichlet 機率的共軛分布（Conjugate distribution）；K 個潛在主題會對應 K 個共軛分布（Liu et al. 2011）。LDA 的優點在於其為一種非監督式生成模型，因此每一個主題均可找出相對應的詞語進行描述；但缺點在於資料集屬於非常態分佈或資料量小的情況下，其產生的效果會較差。Krestel、Fankhauser 與 Nejdl（2009）使用 LDA 尋找網路資源所存在之建議標籤，進而改善搜尋效能；針對軟體開發時所產生之問題回報及分類，Somasundaram 與 Murphy（2012）使用 LDA 進行潛在問題之建議；針對影像的自動分析，Liénou、Maître 與 Datcu（2010）依據不同影像間所存在之樣式關係，採用 LDA 進行樣式判別並進行分類；針對程式撰寫錯誤處理，Lukins、Kraft 與 Etzkorn（2008）使用 LDA 尋找錯誤可能出現地方，並提供可能錯誤程式碼之建議。

RTM 是由 Chang 與 Blei（2010）所提出。RTM 是 LDA 變型模型的一種，並且屬於階層式模型。在 RTM 模型之中，每份文件像 LDA 一樣從主題產生，推測文件和文件之間具有連接關係，即對一個文件網絡建立模型，透過此文件網絡的表達方式，使用者可以自行新增文件結點，並自行嘗試建立彼此間鏈結關係，因此其特別適合處理自由格式之文件（Chang & Blei 2010）。同時，其利用近似後驗推論參數估計和預測，並使用一個變數（二元變數）來表示文件之間的鏈結關係。RTM 優點在於相較 LDA，能增加考慮兩個文件之間的隱含關係，並透過文件間所存在之鏈結關係進行字詞之預測，所以針對自由格式之文件而言，RTM 效能能優於其它語意模型；然而其缺點在於因為考慮了不同文件之間所存在之鏈結因素，因此計算時間將拉長。目前有學者（Gethers & Poshyvanyk 2010）運用 RTM 模型分析物件導向軟體系統類別之間的關係；Moritz 等（2013）使用 RTM 尋找相近之應用程式介面（Application Programming Interface, API）；針對維基百科中消歧義連結（Disambiguation link），學者 Skaggs 與 Getoo（2014）使用 RTM 分析維基百科中具有歧義之文件，並依據使用者定義之主題尋找適合之文件。

參、研究流程

一、研究流程之概述

本研究流程如圖 1 所示。此流程圖包含下列幾個步驟：第一步，稱之為預處理（Pre-procedure）階段，其主要為抓取本研究所需之來源資料集，本研究主要資料來源為 Google Blog Search，本研究採用多引線（Multi-thread）的方式抓取資料來源，透過多引線的方式我們可以對多個網頁文件同時進行抓取，這代表理論上，多個網頁同時抓取與一個網頁抓取所需的時間是一樣的，這樣的做法可以方便我們同時處理大量的資料；第二步，稱之為自然語言處理（Natural Language Processing）階段，其主要將所擷取之文件進行一系列之自然語言處理，透過自然語言處理，我們可以系統化的將非結構化之 HTML 文件（陳林志 & 林育任 2013）轉成相對應之結構化文件，這種結構化文件的方法可以方便使用者快速的從非結構化文件之中尋找所需的結構化資料；第三步，稱之為矩陣階段（Matrix Step），其主要針對自然語言處理後所產生之所有字詞－文件配對進行文件矩陣之建立，矩陣階段主要是將文件資料轉換為數值資料，透過數值資料使用者可以運用不同的模型進行後續處理；第四步，稱之為語意模型（Semantic Models）階段，我們針對文件矩陣進行不同型態之語意模型處理，語意模型可以透過語意的推估過程，尋找文件所隱含之各種潛在語意關係；最後一步，稱之為效能評估（Performance Appraisal）階段，主要是針對不同語意模型進行效能分析。本研究流程之第一至三步請參考本節後續小節之相關說明；第四步請參考第肆節（語意模型之分析與比較)之相關說明；最後一步請參考第伍節（研究結果、分析與討論）之相關說明。

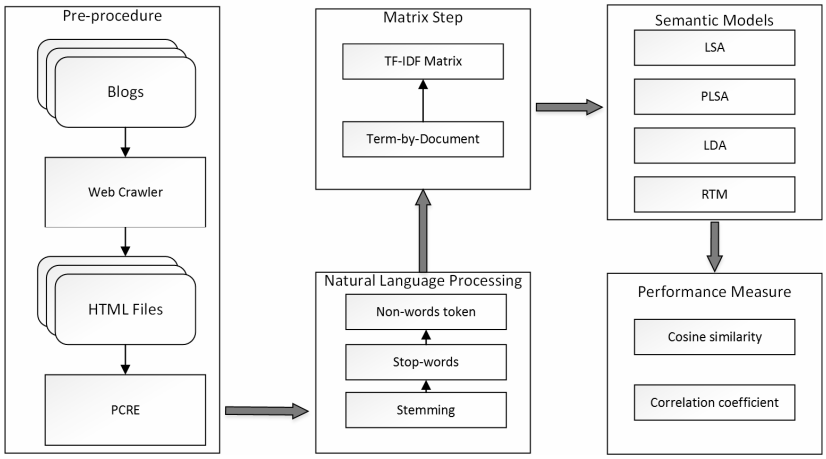


圖 1：研究流程圖

二、預處理階段

（一）網路爬蟲（Web Crawler）

首先抓取所需部落格搜尋引擎的網頁資料，我們採用的方式是透過網路爬蟲技術。使用網路爬蟲可以透過代理伺服器（Proxy Server）的方式，迅速的取回 Google Blog Search 所回傳之搜尋結果 HTML 頁面；由於 HTML 本身屬於非結構化文件，因此我們需要一種機制可以從非結構化文件中取出研究分析所需之資訊，此機制主要透過 PCRE（Perl Compatible Regular Expressions）（Hazel 2015）完成。

（二）PCRE

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）（Kiezun et al. 2012）。

三、自然語言處理階段

（一）字根還原處理（Stemming）

此步驟主要是讓英文的單字還原至其字根，主要將詞性（如 bad、worse）、時態（如 go、went）、複數型態（如 apples）去除，只留下其字根，方便搜尋引擎處理詞彙，並將大寫轉為小寫。本研究採用的字根處理為 Porter 字根演算法（Patil & Patil 2013）。

（二）停用字處理（Stop-words）

英文句子中有許多沒有意義的詞，文章中常常會頻繁出現，但是卻跟文章完全沒有關係，如 a、at、in、of 等。本研究採用 Fox（1989）所定義之停用字列表，該列表包含 421 個常見之停用字。

（三）非字標籤處理（Non-words token）

常見的非字標籤包含：HTML 標籤（如<b>、<table>等）、數字（0～9）、標點符號（如逗點符號、分號等）和特殊符號（如、等）。這些非字標籤會影響字詞擷取辨識上的精確度，因此我們將這些標籤進行去除。

四、矩陣階段

我們將文件和字詞出現的頻率利用矩陣來顯示，將所蒐集之部落格文件轉換

為字詞-文件矩陣（Term-by-document-matrix）後，並運用 TF-IDF（Term Frequency-Inverse Document Frequency）評估字詞出現在文件中的重要性。TF-IDF 評估中，TF 用來計算字詞在文件中出現的頻率，如公式(1)，其中 $n_{(i,j)}$ 表示字詞 i 在文件 j 中出現的頻率，n_j 表示文件 j 中所有字詞的數量；而 IDF 主要是依據所有文件中出現字詞頻率的倒數，如公式(2)，其中 N 為所有文件總數，df_i 為字詞 i 出現的文件總數。

$$TF_{(i,j)} = \log \frac{n_{(i,j)}}{n_j} \tag{1}$$

$$IDF = \log \frac{N}{df_i} \tag{2}$$

TF-IDF 公式的精神在於：如果字詞在一篇文章中出現的頻率高，並且在其它文章中較少出現，則認為此字詞有良好的區別能力。最終的 TF-IDF 就是考量到 TF 及 IDF 值，亦即 TF-IDF 等於 TF×IDF。

肆、語意模型之分析與比較

本研究總共採用了四種語意模型進行分析，分別為潛在語意分析（LSA）、機率潛在語意分析（PLSA）、潛在狄利克里分配（LDA）以及關係主題模型（RTM），茲分述如下。

一、潛在語意分析（LSA）

LSA 是以 SVD 為基礎的一個語意模型，其作法是對文件所表示之文件矩陣進行分析，再將文件投射到潛在語意空間。LSA 的 SVD 步驟如公式(3)所示，其中 A 為字詞-文件矩陣，U 為字詞的向量矩陣，S 為奇異值矩陣，V^T 為文件向量矩陣。

$$A = U \times S \times V^T \tag{3}$$

經過 SVD 處理後，我們將產生三個二維矩陣；U 保留了字詞的資訊，S 為 A 的特徵值所組成的對角矩陣，V^T 則保留文件的資訊。接下來，LSA 進行維度約化（Rank reduction）步驟，這個步驟必須設定所要保留之語意空間之維度 k，我們保留 S 矩陣中前 k 個特徵向量，其他值則當作雜訊過濾，以此得到一個去除雜訊之語意空間 S'。最後將 U、S'、V^T 重新進行矩陣乘積，以此獲得去除雜訊之字詞-

文件矩陣。

二、機率潛在語意分析（PLSA）

LSA 並非以統計的觀點出發，而 PLSA 定義了完整的機率模型，而且每個變數及相對應之機率分佈和條件機率分佈都有了明確解釋，並應用了最大相似度估測法和 EM 演算法。此方法是一種混合模型，需要使用兩層機率概念進行模型空間之建立（如圖 2）。

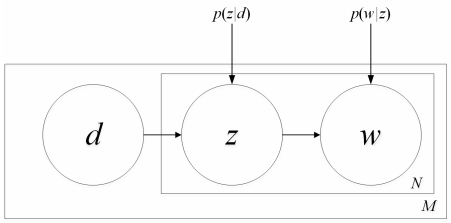


圖 2：PLSA 示意圖

PLSA 給定一個文件 d 之後需要以一定機率選擇與文件相對應的主題 z，從 z 中選擇字詞 w。(d_i, w_j)表示字詞-文件矩陣，d_i 表示的是文件，$d_i \in \{d_1,\ldots,d_M\}$；$w_j$ 則用來表示字詞，$w_j \in \{w_1,\ldots,w_N\}$；$z$ 則為一個潛在變數用來呈現共生矩陣資料關聯性，$z_k \in \{z_1,\ldots,z_K\}$。藉由 PLSA，我們可以將相關機率參數最大化，計算出相關機率分配 $p(d_i,w_j)$，此即為相似度矩陣。

要計算相似度矩陣必需先定義以下的機率符號：$p(d_i)$表示特定文件 d_i 發生的機率、$p(z_k|d_i)$表示給定一個已知文件 d_i，潛在主題為 z_k 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。基於以上定義，PLSA 將 $p(d_i,w_j)$共同出現的聯合機率以公式(4)表示：

$$p(d_i,w_j) = p(d_i)p(w_j \mid d_i), \text{ where } \quad p(w_j \mid d_i) = \sum\nolimits_K p(w_j \mid z_k)p(z_k \mid d_j) \qquad (4)$$

接著 PLSA 使用貝氏定理（Bayes' theorem）轉換公式(4)，並以公式(5)的形式表示之：

$$p(d_i,w_j) = \sum\nolimits_K p(w_j \mid z_k)p(z_k)p(d_i \mid z_k) \qquad (5)$$

公式(5)中 $p(w_j|z_k)$為字詞的機率，$p(d_i|z_k)$為文件的機率，$p(z_k)$為潛在主題的機率。下一步則是定義相似度函數，如公式(6)，藉由迭代來計算，其中 n 為迭代的

代數，$vsm(d_i, w_j)$為文件 d_i 中字詞 w_j 發生的機率權重。

$$L_n(d_i, w_j) = \sum_N \sum_M vsm(d_i, w_j) \log\{p(d_i, w_j)\} \tag{6}$$

接續為 EM 演算法的部分，EM 演算法主要重複執行期望步驟（E-Step）及最大化步驟（M-Step），附錄一顯示 EM 演算法之詳細過程。E-Step 及 M-Step 會不斷的迭代執行，直到完成一個停止準則為止（Chen 2011）。最後 M-Step 所獲得之參數即為 EM 演算法所求得的最佳解，並以此計算最終之結果。

三、潛在狄利克里分配（LDA）

LDA 主要是為了表示文件和主題的分佈機率，其示意圖如圖 3 所示。在 LDA 之中，α表示每一份文件之主題分佈對應到 Dirichlet 機率分佈之參數，β表示每一個主題之字詞分佈對應到 Dirichlet 機率分佈之參數，T 為主題個數，M 為文件的個數，N 為文件中字詞的數量。

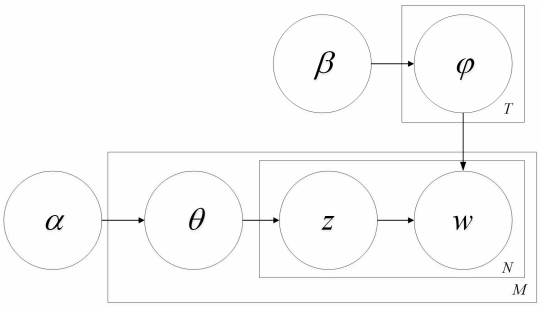


圖 3：潛在狄利克里分配示意圖

其餘 LDA 模型的參數使用公式(7)表示之；其中θ_d 為文件 d 中主題的分佈，其結果為 Dirichlet 分佈下參數α時之共軛分佈值；φ_t 為主題 t 中字詞的分佈，其結果為 Dirichlet 分佈下參數β時之共軛分佈值；$z_{i,d}$ 為文件 d 中字詞 i 的分佈，其結果為多項式分佈下參數θ_d 時之共軛分佈值，$w_{i,d}$ 為特定的字詞，其結果為多項式分佈下參數$\varphi_{zi,d}$時之共軛分佈值。

$$\begin{aligned}
&\theta_d \mid \alpha \sim Dirchlet(\alpha), d \in M \\
&\varphi_t \mid \beta \sim Dirchlet(\beta), t \in T \\
&z_{i,d} \mid \theta_d \sim Multinomial(\theta_d), i \in \mid w_d \mid \\
&w_{i,d} \mid \varphi_{z_{i,d}} \sim Multinomial(\varphi_{z_{i,d}}), i \in \mid w_d \mid
\end{aligned} \tag{7}$$

LDA 使用 Dirichlet 分佈設定文件及字詞之潛在機率分佈α及β，同時使用 EM 演算法估計α及β最終可能之估計值，LDA 演譯的過程請參照附錄二之說明。

四、關係主題模型（RTM）

在 RTM 模型之中，每份文件像 LDA 一樣從主題產生，推測文件和文件之間具有鏈接關係，即對一個文件網絡建立模型，透過此文件網絡的表達方式，使用者可以自行新增文件結點，並自行嘗試建立彼此間鏈結關係，因此其特別適合處理自由格式之文件（Chang & Blei 2010）。部落格文件間鏈結引用關係是由使用者自行建立的，透過 RTM 模型，使用者可以輕易的定義部落格文件之間的主題類別，並經由已發現之主題類別，尋找潛在相關之部落格文件。圖 4 為 RTM 的概念圖。假設文件 d 與 d'存在一個鏈結關係η，並可產生二元引用鏈結變數 c，除此之外 RTM 模型就跟 LDA 模型一樣。RTM 所定義的參數如下所示：C 代表鏈結索引矩陣，$1 \le c \le C$，而鏈結關係我們以η_c 的方式表示之；θ_d 為文件d 中主題的分佈，其代表每個主題在文件中出現的機率；φ_k 為主題中字詞的分佈。

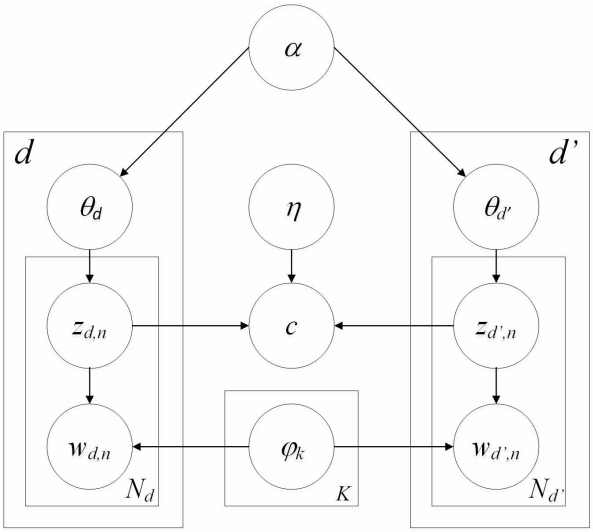


圖 4：RTM 概念圖

RTM 主要考慮到文件與文件間所存在之鏈結關係η，透過鏈結關係計算出文件間對不同主題之影響。RTM 在推估每一份文件與其主題之間出現的潛在機率使用 LDA 進行推估，針對不同文件中字詞及主題之間鏈結關係則使用加權總合的方式進行推估，RTM 演譯的過程請參照附錄三之說明。

每份部落格文件都有一個發佈時間，而部落格搜尋引擎可以剖析部落格文件之結構，以便顯示文件的發佈日期。相比一般搜尋引擎雖能顯示文件日期，但往往該日期只是網站的最後更新日期。一個熱門主題產生，往往同時間可能有許多相關之部落格文件進行討論；當使用者輸入某一主題，部落格搜尋引擎回傳之部落格文件彼此之間的發佈日期愈相近，則可能這些文件討論著相同的熱門主題。雖然 RTM 模型可以透過鏈結關係，呈現部落格文件之間的隱含關係，但是無法呈現時間對文件主題的影響，為了解決部落格文件中的時間問題，我們在原始 RTM 模型上增加一個時間參數 t，其模型如圖 5 所示。新的改良模型稱之為 RTM'，在此模型下文件 d 與 d'存在一個時間鏈結關係ν，並可產生二元引用鏈結變數 t 及時間鏈結索引ν_t。我們藉由 RTM'模型的結果，來觀察時間參數對文件主題的影響。

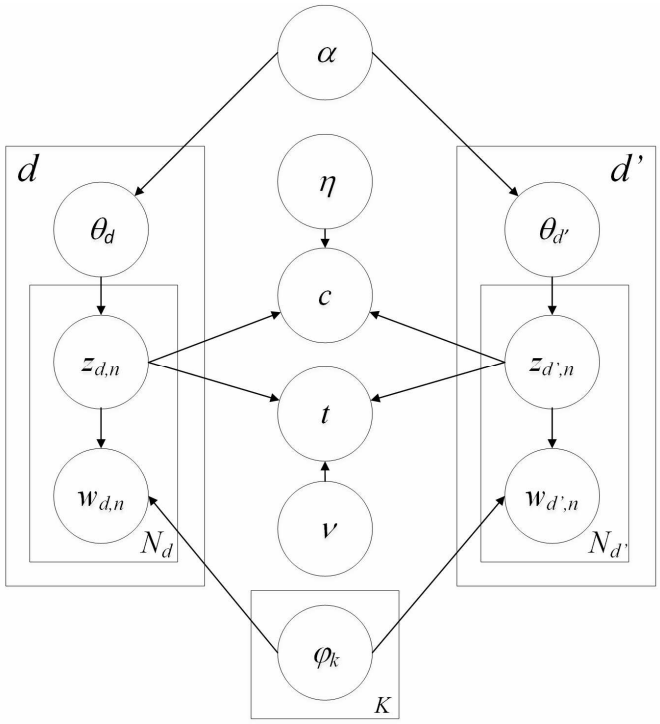


圖 5：關係主題改良模型（RTM'）

RTM'除了考量到 RTM 本身所擁有的文件間字詞及主題之間的文件鏈結關係，同時透過時間參數 t 的引入，以便發掘文件鏈結關係中所存在之時間特性，這個特性能夠額外的處理不同時間點下之文件分類。這樣的好處在於，我們不單單只是能夠依據主題進行文件分類；同時，我們也能確保單一主題也能夠按照不

同時間區間進行分類。這樣做法的精神在於同一時間區間的相同主題文件大致上所呈現的文件意義大抵上是一致的；相對地，不同時間區間下，雖然主題文件是一致，但由於時間區間不同，很容易出現主題的時間盲點問題，例如：當使用者想要瞭解 iPhone 主題下不同文件之分類，按照 RTM 的方法，雖然我們可能找出不同文件中具有 iPhone 主題之文件，但由於每年 Apple 公司都會發表新型的 iPhone 手機，如果不加上時間特性，則無法明確的區別如 iPhone 6 及 iPhone 6s 的差異。

伍、研究結果、分析與討論

一、實驗環境及參數設定

本研究之模擬程式皆建置於個人電腦上。前置處理、自然語言處理、矩陣處理皆使用 PHP 做為開發，語意模型分析實驗則使用 MATLAB R2012b 進行模擬。資料來源是使用 Google Blog Search 輸入查詢關鍵字所回傳的部落格文件為主。

本研究之資料經過語意模型分析後，須使用評估指標評比各分析模型之間的差異。效能分析的來源以矩陣為主，並評估矩陣向量之間的相似度。在評估效能時許多專家學者（Hofmann 2003; Nguyen & Bai 2011; Tan et al. 2005）較常採用餘弦相似度（Cosine similarity）和相關係數（Correlation coefficient）進行評比，故本研究也採用這兩種評估指標。

在語意模型中，主題個數的設置會影響到語意模型的分析結果，過多的主題個數會讓模型的執行效能降低，過少的主題個數會讓模型的鑑別度不足，因此如何設置適當的主題個數將是非常重要。本研究依據相關學者（Chen 2011）的建議，將潛在主題的個數（K）設置在 5 到 50 之間，而後續實驗皆採用此範圍的潛在主題個數。

二、實驗結果

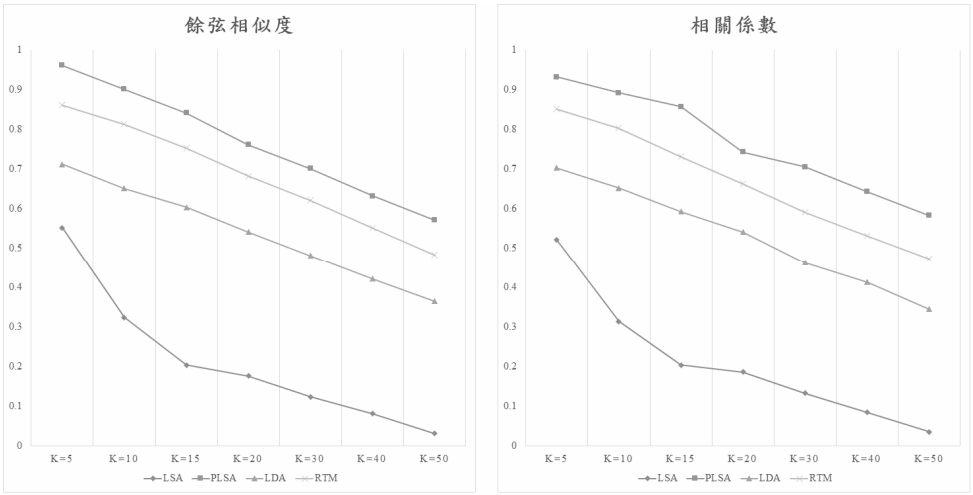
本實驗所選擇之關鍵字來源為 Google 及 Yahoo 於 2012 及 2013 年前十個熱門搜尋關鍵字，其分別選自下列資料集：Google 2012 年度熱門關鍵字（2012）、Google 2013 年度熱門關鍵字（2013b）、Yahoo 2012 年度熱門關鍵字（2012）、Yahoo 2013 年度熱門關鍵字（2013b）。這些關鍵字分別為：“Amanda Todd”、“Amanda Bynes”、“BBB12”、“Boston Marathon”、“Cory Monteith”、“Diablo 3”、“Election”、“Gangnam Style”、“Hurricane Sandy”、“Harlem 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複選取。所謂不重複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複出現時，進行關鍵字選取時，我們只選取一筆。下列關鍵字具有重複出現的特性：“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 Google Blog Search 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。

針對上述關鍵字，我們使用不同語意模型進行分析，得到了潛在主題個數範圍從 5 到 50 之間，文件筆數從 10 到 350 之不同語意模型的餘弦相似度，我們將不同文件數所獲得之餘弦相似度進行平均，即可獲得圖 6(a)的結果。相同地，我們利用類似的方法產生平均相關係數，結果如圖 6(b)所示。

根據圖 6(a)和 6(b)的結果，我們可以觀察到不同語意模型皆會隨著潛在主題個數增大，效能隨之降低。LSA 語意模型數值在 K 大於 5 之後有明顯的下降趨勢，亦即當潛在主題增加後，餘弦相似度及相關係數跟著下降。PLSA 的餘弦相似度和相關係數較無明顯的下降，呈現穩定的狀態，但是其執行時間（詳如後述），是和文件個數及字詞個數增加而呈現指數時間成長，在現今部落格搜尋文件個數及字詞個數都是非常巨大下，PLSA 將要花費相當多的時間才能完成。分析 LDA 及 RTM 模型的差異，觀察圖 6 的結果，我們發現 RTM 模型的效能明顯的優於 LDA 模型，這是因為 RTM 增加文件鏈結的特性，這個特性與部落格文件中經常出現引用鏈結相關，亦即使用者可以自行新增文件結點，並自行嘗試建立彼此間鏈結關係，因此 RTM 特別適合處理自由格式之文件。

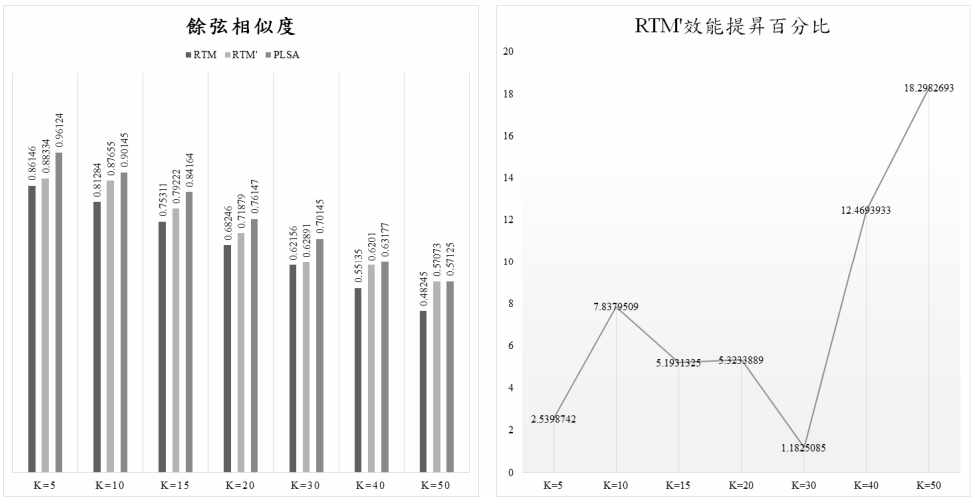


(a) 語意模型餘弦相似度比較 (b) 語意模型相關係數比較

圖 6：語意模型之效能比較

為了觀察時間參數對部落格文件的影響，我們針對 RTM 模型增加時間參數，稱之為 RTM'，並利用餘弦相似度進行觀察比較，結果如圖 7(a)所示。

觀察圖 7(a)後，我們發現 RTM'模型確實能提昇部落格文件之檢索效能，這個原因在於部落格文件主題間的分類，往往會與發佈時間、更改時間等時間參數習習相關。圖 7(b)顯示不同 K 下，RTM'能夠提昇 RTM 模型效能的百分比，其值範圍介於 1.18%至 18%左右，這更證明增加時間參數後，部落格文件之間的主題能夠分類清楚。



(a) RTM與RTM'模型之效能比較 (b) RTM'模型效能提昇百分比

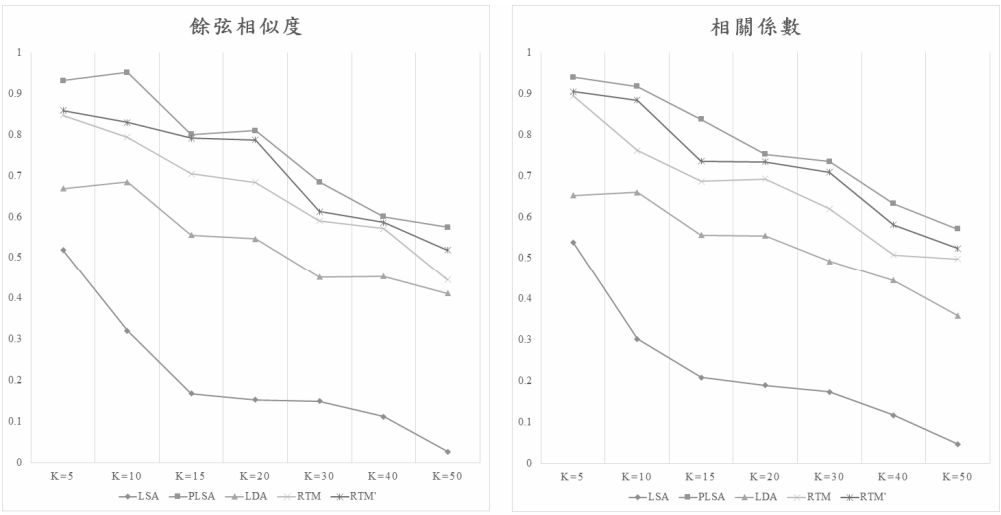
圖 7：RTM 與 RTM'模型效能差異之分析

接下來，我們增加一個實驗，主要分析不同的語言之關鍵字及網頁文件，所造成之效能分析。在這個實驗之中，我們選取 Google 2013（Google 2013a）、Google 2014（Google 2014）、Yahoo 2013（Yahoo 2013a）、Yahoo 2014（Yahoo 2014）年度熱門關鍵字，並經由上述選取方式選取下列熱門關鍵字：104、2048、Facebook、Gmail、Google、PChome、Yahoo、YouTube、iPhone 6、仁川亞運、天氣、太陽花、伊莉論壇、自拍棒、來自星星的你、毒澱粉、神魔之塔、高雄氣爆、郭雪芙、棒球、進擊的巨人、黃色小鴨、圓仔、路跑、網路 ATM、颱風停班課、熱氣球、霜淇淋、翻譯、餿水。相關的文件處理如下：

- 中文斷詞處理：我們使用用中研院的中文斷詞系統（中央研究院 2015）進行中文文件的斷詞。
- 中文字根處理：我們的研究之中，針對中文字根我們並無需要特別處理。
- 非字標籤處理：這部份的處理是與英文文件相似的。

停用字處理：我們首先選取相關之中文停用字字庫（Shijiebei2009 2015），該字庫包含 1893 中文簡體停用字。為了將簡體停用字轉成繁體停用字，我們首先將所有簡體字轉成繁體字，並檢查所有停用字之用法，將其中不恰當之簡體停用字改為繁體停用字。

圖 8 顯示不同語意模型下之餘弦相似度及相關係數之結果，圖中所呈現的趨勢與前面的實驗相似，亦即 RTM 模型特別適合處理自由格式之文件，而 RTM'模型除了基於 RTM 所具有之自由格式文件，還能夠特別處理時間參數，經由時間參數的強化，整體效能將能夠進一步的提昇。



(a) 中文處理之餘弦相似度　　(b) 中文處理之相關係數

圖 8：中文處理下之語意模型分析

圖 9 為語意模型的執行時間，我們可以發現雖然 PLSA 執行效能最好，但其執行時間確是其它模型的數以百倍，這是因為 PLSA 中，EM 演算法所估計參數的時間，是與文件個數及字詞個數增加而呈指數時間成長，迭代執行時間隨著文件個數增加而成長迅速。LSA 因為沒有經過迭代運算的過程，所以其執行的時間速度最快，但其本身的執行效能最差。LDA、RTM 和 RTM'雖然有迭代過程，但是並不直接估計所有文件及字詞，所以不會跟著文件及字詞大小有所劇烈影響。

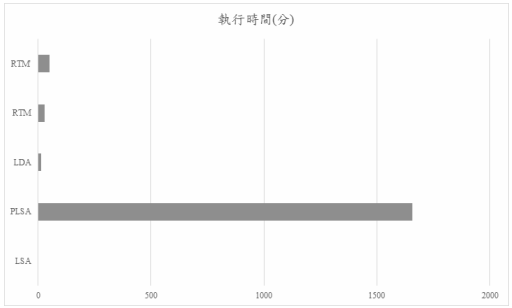


圖 9：語意模型執行時間

學者 Hofmann（2004）已經證明 PLSA 的時間複雜度為 O（M×N×L），其中 O（M×N）為 EM 演算法每次疊代所需的時間複雜度。然而，在現今網際網路環境之中，網頁文件總數（M）以及字詞總數（N）幾乎都是天文數字般的龐大（Kunder 2008）；與此同時，潛在主題總數（L）將會隨著 M 及 N 增加而增加（Inoue 2005）。學者（Chen 2012）已經證明，PLSA 只適合處理極小的資料集，然而其並無法在有限的時間下（依據文獻統計，網際網路回應時間必須在 0.5 秒以下才算有限時間（Menascé 2002），處理網際網路環境下，如此龐大 M、N 及 L 的資料集。即使透過硬體的改善，將現今電腦的計算能力提昇至每秒處理 10 億個指令，當 M、N 及 L 等於 1 百萬筆資料下（這在網際網路環境下是屬於資料少的情形），在 O（M×N×L）時間複雜度下，所需的執行時間是 32 年（Horowitz et al. 2007）之久，這代表在這樣的時間複雜度及大量資料處理環境下，PLSA 在有限時間下，不可能透過硬體的改善達成最佳解。是故，PLSA 後續的研究 LDA、RTM 以及本研究之 RTM'都是著重在有限時間下，求得合理解。

由於 RTM'是 RTM 的時間參數模型，因此 RTM'不單單是具有如 RTM 般處理多文件鏈結的能力，同時也能夠針對文件所發佈或修改時間進行妥善的處理。在現今網路發展上，常常出現以時間為討論主軸的文件來源。例如：Facebook 貼文、Line 訊息、部落格文件等，這類文件都需要以時間進行主題的分類；因此透過我們的 RTM'模型，我們可以針對這類文件進行有效且適當的文件分類。

陸、結論及未來方向

本研究嘗試在部落格搜尋引擎回傳之文件中，尋找適合的語意模型。依據實驗結果所示，使用不同類型之語意模型皆能提升谷歌部落格搜尋引擎之效能。在這些模型之中，PLSA 雖能得到最高的效能，但是所花費的計算時間相當可觀，是故不適合應用於部落格搜尋引擎；LDA 及 RTM 模型則可大幅降低語意模型分析所花的時間，其中 RTM 為 LDA 的改良模型，其增加考慮兩個文件之間相互鏈結關係，因此特別適用於自由格式之文件，依據實驗結果分析，RTM 效能明顯的優於 LDA。最後，由於部落格文件主題之分類，常常會與時間議題有關，是故我們針對 RTM 模型新增時間參數，透過這個時間參數的引用，部落格文件之間的主題能夠更加分類清楚。未來研究有幾個方向：(1)其它類型文件處理：我們預計將 RTM'引用到其它類型的文件，如 Facebook 貼文及 Line 訊息，以便驗證 RTM'是否可以運用在不同的文件類型之上。(2)提出 PLSA 修改模型：PLSA 的優點在於效能相當優良，但其缺點在於計算時間相當耗時，因此如何在時間及效能上求得一個平衡也是我們未來的一個研究方向。

誌謝

本文接受行政院科技部專題研究計畫（MOST 105-2221-E-259-030，104-2221-E-259-038，103-2221-E-259-023）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

中央研究院（2015），中文斷詞系統，http://ckipsvr.iis.sinica.edu.tw（存取日期 2015/09/21）。

余至浩（2014），痞客邦百億 Log 上雲端－挖掘社群行為尋找新服務，http://www.ithome.com.tw/news/90977（存取日期 2015/09/21）。

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 20 卷，第 1 期，頁 97-130。

創世紀（2014）。comScore 與創市際依據 comScore MMX™ 數據公佈 2014 年 10 月

台灣網路活動分析報告，http://www.insightxplorer.com/news/news_12_22_14.html（存取日期 2015/09/21）。

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003), 'Latent dirichlet allocation', *Journal of*

Machine Learning Research, Vol. 3, No. 1, pp. 993-1022.

Chang, J. and Blei, D.M. (2010), 'Hierarchical relational models for document networks', *The Annals of Applied Statistics*, Vol. 4, No. 1, pp. 124-150.

Chen, L.C. (2011), 'Term suggestion with similarity measure based on semantic analysis techniques in query logs', *Online Information Review*, Vol. 35, No. 1, pp. 9-33.

Chen, L.C. (2012), 'Building a term suggestion and ranking system based on a probabilistic analysis model and a semantic analysis graph', *Decision Support Systems*, Vol. 53, No. 1, pp. 257-266.

Cosma, G. and Joy, M. (2012), 'An approach to source-code plagiarism detection and investigation using latent semantic analysis', *IEEE Transactions on Computers*, Vol. 61, No. 3, pp. 379-394.

Fox, C. (1989), 'A stop list for general text', *ACM SIGIR Forum*, Vol. 24, No. 1-2, pp. 19-35.

Fujimura, K., Toda, H., Inoue, T., Hiroshima, N., Kataoka, R. and Sugizaki, M. (2006), 'Blogranger-a multi-faceted blog search engine', *Proceedings of the WWW 2006 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, Edinburgh, UK, May 22-26.

Gethers, M. and Poshyvanyk, D. (2010), 'Using relational topic models to capture coupling among classes in object-oriented software systems', *Proceedings of the 2010 IEEE International Conference on Software Maintenance*, Timișoara, Romania, September 12-18, pp. 1-10.

Google (2012), 'Google zeitgeist 2012', available at http://tinyurl.com/mc2f9nf (accessed 21 September 2015).

Google (2013a), '2013 hot keywords for Google', available at http://tinyurl.com/puj9brg (accessed 21 September 2015).

Google (2013b), 'Google zeitgeist 2013', available at http://tinyurl.com/kubnvvg (accessed 21 September 2015).

Google (2014), '2014 hot keywords for Google', available at http://tinyurl.com/pnqkld9 (accessed 21 September 2015).

Hazel, P. (2015), 'Pcre-perl compatible regular expressions', available at http://www.pcre.org/pcre.txt (accessed 21 September 2015).

Hearst, M.A., Hurst, M. and Dumais, S.T. (2008), 'What should blog search look like? ', *Proceedings of the 2008 ACM Workshop on Search in Social Media*, Napa Valley, California, USA, October 30, pp. 95-98.

Hofmann, T. (1999), 'Probabilistic latent semantic indexing', *Proceedings of the 22th Annual International SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, California, USA, August 15-19, pp. 50-57.

Hofmann, T. (2003), 'Collaborative filtering via gaussian probabilistic latent semantic analysis', *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, Toronto, Canada, July 28 - August 01, pp. 259-266.

Hofmann, T. (2004), 'Latent semantic models for collaborative filtering', *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 89-115.

Horowitz, E., Sahni, S. and Anderson-Freed, S. (2007), *Fundamentals of Data Structures in C*, Silicon Press, Summit, New Jersey.

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*, Michigan, USA, June 29-30, pp. 61-64.

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.

Jin, X., Zhou, Y. and Mobasher, B. (2004), 'Web usage mining based on probabilistic latent semantic analysis', *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, August 22-25, pp. 197-205.

Kiezun, A., Ganesh, V., Artzi, S., Guo, P. J., Hooimeijer, P. and Ernst, M.D. (2012), 'Hampi: A solver for word equations over strings, regular expressions, and context-free grammars', *ACM Transactions on Software Engineering and Methodology*, Vol. 21, No. 4, pp. 25:1-25:28.

Kim, J. and Yun, U. (2014), 'The blog ranking algorithm using analysis of both blog influence and characteristics of blog posts', *Lecture Notes in Electrical Engineering*, Vol. 274, No. 2014, pp. 13-17.

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 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, June 27- 29, pp. 158-162.

Krestel, R., Fankhauser, P. and Nejdl, W. (2009), 'Latent dirichlet allocation for tag recommendation' , *Proceedings of the third ACM conference on Recommender Systems*, New York, USA, October 23-25, pp. 61-68.

Kunder, M.d. (2008), 'The size of the world wide web', available at

http://worldwidewebsize.com/ (accessed 21 September 2015).

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 2013 IEEE International Conference on Multimedia and Expo*, San Jose, California, 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. 2-3, pp. 259-284.

Liénou, M., Maître, H. and Datcu, M. (2010), 'Semantic annotation of satellite images using latent dirichlet allocation', *IEEE Geoscience and Remote Sensing Letters*, Vol. 7, No. 1, pp. 28-32.

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 23th International Florida Artificial Intelligence Research Society Conference*, Florida, USA, May 19-21, pp. 235-240.

Liu, Z., Zhang, Y., Chang, E.Y. and Sun, M. (2011), 'Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing', *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 26:1-26:18.

Logan, B., Kositsky, A. and Moreno, P. (2004), 'Semantic analysis of song lyrics', *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, June 27-30, pp. 827-830.

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 2012 3rd International Conference on E-Business and E-Government*, Shanghai, China, May 11-13, pp. 439-442.

Lukins, S.K., Kraft, N.A. and Etzkorn, L.H. (2008), 'Source code retrieval for bug localization using latent dirichlet allocation', *Proceedings of the 15th Working Conference on Reverse Engineering*, Antwerp, Belgium, October 15-18, pp. 155-164.

McInerney, J., Rogers, A. and Jennings, N.R. (2012), 'Improving location prediction services for new users with probabilistic latent semantic analysis', *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, Pittsburgh, Pennsylvania, USA, September 05-08, pp. 906-910.

Menascé, D.A. (2002), 'Qos issues in web services', *IEEE Internet Computing*, Vol. 6, No. 6, pp. 72-75.

Mesaros, A., Heittola, T. and Klapuri, A. (2011), 'Latent semantic analysis in sound

event detection', *Procedding of the 19th European Signal Processing Conference*, Barcelona, Spain, August 29- September 2, pp. 1307-1311.

Moritz, E., Linares-Vásquez, M., Poshyvanyk, D. and Grechanik, M. (2013), 'Export: Detecting and visualizing api usages in large source code repositories', *Proceedings of the 2013 IEEE/ACM 28th International Conference on Automated Software Engineering*, TBD, CA, USA, November 11-15, pp. 646-651.

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, H.V. and Bai, L. (2011), 'Cosine similarity metric learning for face verification', *Lecture Notes in Computer Science*, Vol. 6493, No. 2011, pp. 709-720.

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.

Patil, C.G. and Patil, S.S. (2013), 'Use of porter stemming algorithm and svm for emotion extraction from news headlines', *International Journal of Electronics, Communication and Soft Computing Science and Engineering*, Vol. 2, No. 7, pp. 9-13.

Qureshi, M.A., Younus, A., Touheed, N., Qureshi, M.S. and Saeed, M. (2011), 'Discovering irrelevance in the blogosphere through blog search', *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, Kaohsiung, Taiwan, July 25-27, pp. 457-460.

Shijiebei2009 (2015), '1893 stop words for Chinese', available at http://blog.csdn.net/shijiebei2009/article/details/39696571 (accessed 21 September 2015).

Skaggs, B. and Getoor, L. (2014), 'Topic modeling for wikipedia link disambiguation', *ACM Transactions on Information Systems*, Vol. 32, No. 3, pp. 10:1-10:24.

Somasundaram, K. and Murphy, G.C. (2012), 'Automatic categorization of bug reports using latent dirichlet allocation', *Proceedings of the fifth India Software Engineering Conference*, Kanpur, India, February 22-25, pp. 125-130.

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

Thelwall, M. and Hasler, L. (2007), 'Blog search engines', *Online Information Review*, Vol. 31, No. 4, pp. 467-479.

Xu, C., Zhang, Y.F., Zhu, G., Rui, Y., Lu, H. and Huang, Q. (2008), 'Using webcast text for semantic event detection in broadcast sports video', *IEEE Transactions on*

Multimedia, Vol. 10, No. 7, pp. 1342-1355.

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 6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 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 21 September 2015).

Yahoo (2013a), '2013 hot keywords for Yahoo', available at http://tinyurl.com/ qcoxybv (accessed 21 September 2015).

Yahoo (2013b), '2013 year in review', available at http://tinyurl.com/q3zlabr (accessed 21 September 2015).

Yahoo (2014), '2014 hot keywords for Yahoo', available at http://tinyurl.com/ovkch3u (accessed 21 September 2015).

Yeh, J.Y., Keb, H.R., Yang, W.P. and Meng, I.H. (2005), 'Text summarization using a trainable summarizer and latent semantic analysis', *Information Processing & Management*, Vol. 41, No. 1, pp. 75-95.

Zeng, J., Cheung, W.K. and Liu, J. (2013), 'Learning topic models by belief propagation', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 35, No. 5, pp. 1121-1134.

Zhu, L., Sun, A. and Choi, B. (2011), 'Detecting spam blogs from blog search results', *Information Processing and Management*, Vol. 47, No. 2, pp. 246-262.

附錄一：EM 演算法過程

其中 E-step 的公式如下所示：

$$P(z_k \mid d_i, w_j) = \frac{p(d_i \mid z_k)p(z_k)p(w_j \mid z_k)}{\sum_K p(d_i \mid z_k)p(z_k)p(w_j \mid z_k)} \tag{8}$$

在 E-Step 之中，我們利用目前的估計參數計算潛在變數 z_k 的事後機率。M-Step 利用潛在變數在 E-Step 的估計值，使得觀察的聯合對數相似度期望值最大化，將所有參數使用公式(9)(10)(11)更新。

$$P(d_i \mid z_k) = \frac{\sum_M vsm(d_i, w_j)p(z_k \mid d_i, w_j)}{\sum_N \sum_M vsm(d_i, w_j)p(z_k \mid d_i, w_j)} \tag{9}$$

$$P(z_k) = \frac{\sum_N \sum_M vsm(d_i, w_j)p(z_k \mid d_i, w_j)}{\sum_N \sum_M vsm(d_i, w_j)} \tag{10}$$

$$P(w_j \mid z_k) = \frac{\sum_N vsm(d_i, w_j)p(z_k \mid d_i, w_j)}{\sum_N \sum_M vsm(d_i, w_j)p(z_k \mid d_i, w_j)} \tag{11}$$

M-Step 所得出的 $p(d_i|z_k)$、$p(z_k)$、$p(w_j|z_k)$再帶回公式(5)，將導致相似度函數，如公式(6)，持續的增加。

附錄二：LDA 演算過程

LDA 第一步就是同時考慮 T 維的 Dirichlet 分佈，如公式(12)所示。$P(\theta\,|\alpha)$為給定α時，θ的 Dirichlet 機率分佈。

$$P(\theta\,|\,\alpha)=\frac{\Gamma(\sum_T \alpha_i)}{\prod_T \Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\ldots\theta_T^{\alpha_T-1} \tag{12}$$

針對 N 個 z 及觀察到的特徵值 w，其聯合機率分佈如公式(13)所示；同時，我們整合了 z 的總和得到邊際分佈，如公式(14)所示。

$$P(\theta,z,w)=P(\theta\,|\,\alpha)\prod_N P(z_n\,|\,\theta)P(w_n,\beta) \tag{13}$$

$$P(w\,|\,\alpha,\beta)=\int P(\theta\,|\,\alpha)\left[\prod_N \sum_{z_n} P(z_n\,|\,\theta)P(w_n\,|\,z_n,\beta)\right]d\theta \tag{14}$$

在 LDA 模型中，文件中字詞的生成模型是假定每個字詞之間是獨立存在的，所以文件 D 是由下列聯合機率所定義的，如公式(15)所示：

$$P(D\,|\,\alpha,\beta)=\prod_N \int P(\theta\,|\,\alpha)\left[\prod_N \sum_{z_n} P(z_n\,|\,\theta)P(w_n\,|\,z_n,\beta)\right]d\theta \tag{15}$$

在 LDA 中有兩個主要的問題，推論和參數估計，推論問題的目的在於計算出從文件中觀察到隱藏變數的條件機率分佈，我們的目的是給定一份文件後，求出每種主題分佈的機率，以及文件中每個字詞來自某個主題的機率，這部分使用公式(16)來表示。公式(16)中的分母部分可由公式(14)求出。

$$P(\theta,z\,|\,w,\alpha,\beta)=\frac{P(\theta,z,w\,|\,\alpha,\beta)}{P(w\,|\,\alpha,\beta)} \tag{16}$$

而估計這些參數必須計算條件機率 $P(\theta,z\,|w)$，本研究使用 variational 近似法(Blei et al. 2003)來解決這一推論。利用這個新模型，可以最大化得到 $P(\theta,z\,|w,\alpha,\beta)$的近似值，如公式(17)所示。

$$(r^*,\varphi^*)=\arg\min_{\gamma,\varphi} D(q(\theta,z\,|\,\gamma,\varphi)\,\|\,P(\theta,z\,|\,w,\alpha,\beta)),\ \text{where}$$

$$q(\theta,z\,|\,\gamma,\varphi)=q(\theta\,|\,\gamma)\prod_N q(z_n,\varphi_n) \tag{17}$$

再利用 EM 演算法進行迭代，使得 variational 推論中的下界最大化，並求出此時的α與β。

附錄三：RTM 演算過程

θ_d 及 φ_k 的結果可以經由公式(18)和(19)計算得出

$$\mu_{\theta_d \to z^k_{w,d}}(k) = f_{\theta_d} \prod_{-w} \mu_{-w,d}(k)\alpha \tag{18}$$

$$\mu_{\varphi_k \to z^k_{w,d}}(k) = f_{\varphi_w} \prod_{-d} \mu_{w,-d}(k)\beta \tag{19}$$

然後我們只需要導出 η_c 對主題的影響，其表示如公式(20)所示。

$$f_{\eta_c}(k \mid k') = \frac{\sum_{(d,d')} \mu_w, d(k) \cdot \mu_w, d(k')}{\sum_{(d,d'),k'} \mu_w, d(k) \cdot \mu_w, d(k')}, \text{ where } \mu_w(k) = \sum_d x_w, d\mu_w, d(k) \tag{20}$$

RTM 原始使用廣義線性模型，但是這會造成推導過程越趨於複雜，公式(20)同樣能捕捉兩個鏈結文件的主題相互影響（Zeng et al. 2013），並代入公式(21)：

$$\mu_{\eta_c \to z^k_{w,d}}(k) = \sum_{d'} \sum_{k'} f_{\eta_c}(k \mid k')\mu_{d'}(k') \tag{21}$$

我們針對 η_c、θ_d、φ_k 使用加權總合的方式進行參數的估計，以避免算數溢位，其公式如下所示。

$$\mu(z_{w,d} = k) \propto \left[(1-\varepsilon)\mu_{\theta_d \to z_{w,d}}(k) + \varepsilon_{\eta_c \to z_{w,d}}(k)\right] \times \mu_{\varphi_k \to z_{w,d}}(k), \text{ where } \varepsilon \in [0,1] \tag{22}$$

當 ε 為 0 時，RTM 還原為原始 LDA 模型，並代入公式(23)和(24)得到 θ_d 和 φ_k。

$$\theta_d(k) = \frac{\mu_d(k) + \alpha}{\sum_k \left[\mu_d(k) + \alpha\right]} \tag{23}$$

$$\varphi_k(k) = \frac{\mu_k(k) + \beta}{\sum_k \left[\mu_k(k) + \beta\right]} \tag{24}$$

附錄四：RTM'演算過程

公式(25)能捕捉到兩個時間鏈結文件的主題相互作用關係，並代入公式(26)。最後將公式(26)的結果代入 LDA 中的公式(23)和(24)，即可得到θ_d和φ_k。

$$f_{v_t}(k \mid k') = \frac{\sum_{(d,d')} \mu_w, d(k) \cdot \mu_w, d(k')}{\sum_{(d,d'),k'} \mu_w, d(k) \cdot \mu_w, d(k')} \tag{25}$$

$$\mu_{v_t \to z^k_{w,d}}(k) = \sum_{d'} \sum_{k'} f_{\eta_c}(k \mid k') f_{v_t}(k \mid k') \mu_{d'}(k') \tag{26}$$