使用不同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
lcchen@mail.ndhu.edu.tw

吳忠澄
國立東華大學資訊管理碩士學位學程
tiredapple@gmail.com

摘要

近年來社群網路的應用-部落格成長迅速，通常我們使用部落格搜尋引擎，找尋所需的部落格文件。但部落格文件在部落格搜尋引擎中更新相當頻繁，部落客常面臨資訊過載的問題。且當我們使用部落格搜尋引擎查找部落格文件時，常出現同義詞和一詞多義等問題。本論文中，我們使用兩種語意分析模型，潛在語意分析(LSA)和機率潛在語意分析(PLSA)解決以上問題。LSA 利用奇異值分解技術擷取字詞間的語意關係，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，機率潛在語意分析，奇異值分解

使用不同語意模型分析線上部落格文件

壹、緒論(Introduction)

自美國國家科學基金會(NSF)在 1995 年開放網際網路供商業使用後，大大的改變人類的生活。早期的網路稱為 Web1.0，僅實現單項信息的傳播，直到 Web 2.0 的概念逐漸實現。Web 2.0 是一個架構在知識上的環境，人與人之間互動而產出的內容，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(Judicibus 2008)。

Web 2.0 在近幾年來已經發展成熟，典型的網路站點有：網路社群、部落格、Wiki 等等，而最廣為人知的應用便是部落格(Blog)，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、影音的網頁，亦即記錄使用者的生活(Nardi et al. 2004)，由於世界上已有太多的部落格，因此近年一些部落格搜尋引擎服務開始出現，如 GoogleBlogSearch(Google 2010)、BlogScope(BlogScope 2007)、Technorati(Technorati 2010)，但技術尚未成熟，難免會有許多隱含的資訊無法被搜尋引擎擷取到，也有可能產生資訊過載(Information overload)的情形。

在我們搜尋部落格文件時，常會碰到同義詞(Synonymy)(兩個不同字詞分別有不同語意)及一詞多義(Polysemy)(一個字詞有多種語意)之問題。然而字詞(Term)不一定為文件(Document)最基本的組成元素，在字詞和文件之間有一層隱含的語意關係，我們稱之為主題(Topic)。針對語意關係的課題，學者們提出以下的語意模型：潛在語意分析(Latent Semantic Analysis, LSA)(Deerwester et al. 1990)和機率潛在語意分析(Probabilistic Latent Semantic Analysis, PLSA)(Hofmann 1999)。

本研究考慮到部落格文件和字詞間隱含的潛在語意關係，利用 LSA、PLSA 及加權過後的 LSA、PLSA 等語意模型去分析線上部落格文件，並觀察是否能夠提昇部落格文件檢索的效能。本研究之研究目的有如以下三點：

(一) 應用於線上部落格文件檢索的語意模型建議
(二) 解決部落格文件的同義詞和一詞多義問題
(三) 使用語意模型後是否能提昇部落格文件檢索效能

貳、相關文獻探討

在許多的應用上，資訊檢索和機器學習可以說密不可分。本節，我們將討論與本論文相關的文獻，其中包含：向量空間模型、TFIDF、潛在語意關係、機率潛在語意關係。

一、向量空間模型

在資訊檢索系統中，文件通常由向量表示，意指是為特徵的字詞出現在每篇文件的現象，稱為向量空間模型(Vector Space Model)(Salton et al. 1975)。

在 VSM 模型的實現上，觀察圖 1，其中 $w_{i,j}$ 表示字詞 i 在文件 j 中出現的頻率值，整個文件集可以透過圖 1 呈現的字詞文件矩陣來表示，即為向量空間模型。

$$VSM = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1j} \\ w_{21} & w_{22} & \cdots & w_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i1} & w_{i2} & \cdots & w_{ij} \end{bmatrix}$$

圖 1 向量空間模型

二、TFIDF

詞語的選取是很重要的，因為其精確性會影響到檢索的結果。因此，適當的權重值對於表現文件特徵有正面的幫助，也有助於提昇檢索的正確率。其中一種簡單而廣泛被採用的法則為 TFIDF(Term Frequency and Inverse Document Frequency)(Madsen et al. 2005)。

在一份給定的文件中，詞頻(Term Frequency, TF)指的是某一給定的字詞在該文件出現的次數。它的重要性可表示為：

$$tf(i,j) = \frac{w_{i,j}}{\sum_k w_{k,j}} \tag{1}$$

上式中，$w_{i,j}$ 表該字詞在文件 $d_{i,j}$ 中出現的次數，而分母則為文件 d_j 中所有字詞的出現次數之和。

逆向文件頻率(Inverse Document Frequency, IDF)是一個字詞普遍重要性的度量。某一特定詞語的 IDF，可以由總文件數目除以包含該字詞之文件的數目，再將得到的商取對數得到，得到如下表示：

$$df(i) = \log \frac{|D|}{|\{d : t_i \in d\}|} \tag{2}$$

其中 $|D|$ 表示語料庫中的文件總數，$|\{d : t_i \in d\}|$ 表示包含詞語 t_i 的文件數目。最後，我們同時考慮 TF 及 IDF 的特性，得到 TFIDF 公式如下：

$$tfidf(i,j) = tf(i,j) \times idf(i) \tag{3}$$

根據上述公式，某一特定文件中高頻率的詞語，以及該詞語在整個文件集合中低的文件頻率，即可產生出較高的 TFIDF 值。因此，TFIDF 傾向於過濾掉常見的詞語，保留重要的詞語。

三、潛在語意分析(LSA)

LSA 是以數學統計為基礎的知識模型，為向量空間模型的一種延伸，以奇異值分解(Singular Value Decomposition, SVD)與維度約化(Dimension Reduction)為核心作為邏輯推演的方式，從一篇文章中具有某些概念的詞，擷取出來並重新呈現的一種理論與方法(Landauer et al. 1997)。

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。在其運算過程中，原始的二維矩陣會利用奇異值分解(SVD)技術分解成三個二維矩陣，其中兩組為奇異向量(Singular Vector)，另一組為保存奇異值的對角矩陣(Simgular Value)。對角矩陣中保留適當個數的奇異值，並過濾雜訊後，在將三個矩陣相乘即可得到具有潛在語

意的新矩陣。因此能夠正確推理更深層次的語意關係(所以稱為 Latent semantic)(Landauer et al. 1998)。

LSA 廣泛的應用於 IR 領域(Hofmann 2004)，也被 Landauer 採用在心理語言學的分析上(Landauer et al. 1998)、自動化摘要寫作評量的計分標準(Kanejiya et al. 2003)。LSA 詳細的工作原理請參照第三節說明。

四、機率潛在語意分析(PLSA)

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 1999)。不同於 LSA 是將文件和字詞向量投射至潛在語義空間的作法，PLSA 以 Aspect Model 作為主要的架構，可用於分析詞彙和句子的共同出現(co-occurrence)的現象。使用機率密度函式作為以觀察到的文件和字詞之間潛在語意關聯性的呈現方式，並利用最大相似度估計法則，結合了 EM 演算法不斷的訓練文件參數推估出隱含的模型參數。PLSA 對於 LSA 是一個重要的進階觀點(Hofmann 1999)。

PLSA 模型目前被廣泛應用於許多領域，包括文件分類(Cai et al. 2003)、文件分段(Brants et al. 2002)、網頁探勘、語音辨識、圖像辨識及語言模型調適等。其主要的特徵，是針對字詞和文件共同事件尋求一個生成模型。PLSA 詳細工作原理請參照第三節說明。

參、研究架構

本研究系統流程如圖 2 所示，首先我們取得部落格文件來源，透過 PCRE(Hazel 2011)和 NLP 等文件前置處理，接著產生字詞-文件、TFIDF 矩陣、機率矩陣，最後將矩陣實施語意模型，並評估其效能。

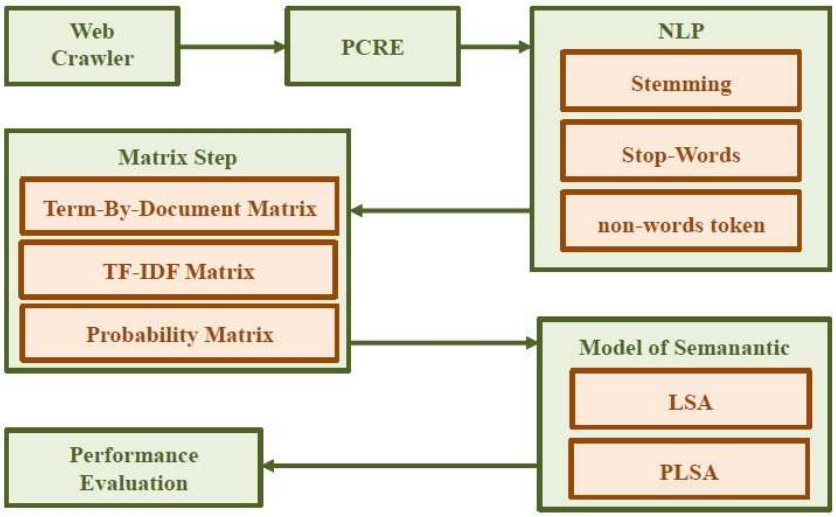


圖 2 研究流程圖

一、部落格文件資料來源

首先，我們必須取得線上部落格文件，在本研究中，我們使用 GoogleBlogSearch(US)(Google 2010)，其為 Google 搜尋引擎的一個附加功能，經查詢關

鍵字後所擷取的部落格文件結果多為部落格服務平台內部落客所發表的文章，如 Twitter、Blogger、Myspace 等等。線上部落格文章示經由電腦的選取及排序，通常依部落格的知名度和文章的重要性排序，然而 GoogleBlogSearch 已經將這類工作做好，因此我們選擇它的搜尋結果作為本研究的資料來源。

二、文件前置處理

前置處理階段，本研究利用 PerlCompatible RegularExpressions(PCRE)擷取部落格文件，接著採用一系列的 NLP 處理如下：字幹處理(Stemming)、停用字(Stop words)、非字記號(Non-words token)（例如：標點符號，Html 標籤，數字等），描述如下：

(一) 字幹處理

在英文的文法結構中，由於名詞的單複數(如 cat 和 cats)、動詞的時態(如 walk 和 walked)、詞性變化(如 good 和 goodness)等，導致相同的詞有不同的方式呈現，但其代表的意義大致是相同的。若將其各自視為不同的詞語，則相對稀釋了重要性。故需進行字幹處理。本研究隻字幹處理方法以 Porter 所提出的演算法(Porter 2011)為基準。

(二) 停用字

在英文使用中，會出現大量如 a、in、the、of 等停用字。然而這些字大部分單獨存在時是無意義的，且會影響字詞的擷取和便是上的準確，甚至造成錯誤引導。本研究採用的停用字字典為 Fox 所提出的 421 個停用字(Fox 1989)。

(三) 非字記號

在文章中可能含有一些標點符號(如.,-‘等)、特殊符號(如%#$等)以及 Html 標籤(如
等)。這些符號可能會影響詞語判斷，所以我們將之移除。

三、矩陣處理

由於語意模型的實施必須採用矩陣形式，因此我們將部落格文件形成以下三種不同類型的字詞-文件矩陣：(1)VSM 矩陣(2)TFIDF 矩陣(3)機率矩陣。

VSM 矩陣、TFIDF 矩陣產生方法如第二節描述，機率矩陣我們定義一個簡單的機率方法如下：

$$TOP(i,j) = \frac{TermOccurrd_{(i,j)}}{TotalDocumentWords_j} \tag{4}$$

上式中，*TOP(i,j)*為在文件 *j* 中，字詞 *i* 出現的機率，分子部分為字詞 *i* 在文件 *j* 中出現的次數，分母部分為文件 *j* 的總字數。圖 3 為本研究所產生的三種矩陣，左至右為 VSM、TFIDF、機率矩陣。

1	0	0	0	1
1	0	0	0	0
3	0	0	0	0
1	0	0	0	0
0	2	0	0	0

0.232193	0	0	0	0.232193
0.332193	0	0	0	0
0.996578	0	0	0	0
0.332193	0	0	0	0
0	0.664386	0	0	0

0.071429	0	0	0	0.035714
0.071429	0	0	0	0
0.214286	0	0	0	0
0.071429	0	0	0	0
0	0.111111	0	0	0

圖 3 三種字詞文件矩陣

四、語意模型

本研究使用兩種語意模型方法，我們將產生的三種矩陣進行 LSA 和 PLSA 流程，工作原理如下：

(一) 潛在語意分析(LSA)

圖 4 為 LSA 的示意圖：利用奇異值分解(SVD)和維度約化步驟，可將潛在的語意顯現出來，更使原本的知識模型提升到較高層次的語意層面。

圖 4 潛在語意關係(LSA)工作原理

在 LSA 的運算過程中，原始的二維矩陣會利用 SVD 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，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，如圖 5 所示。

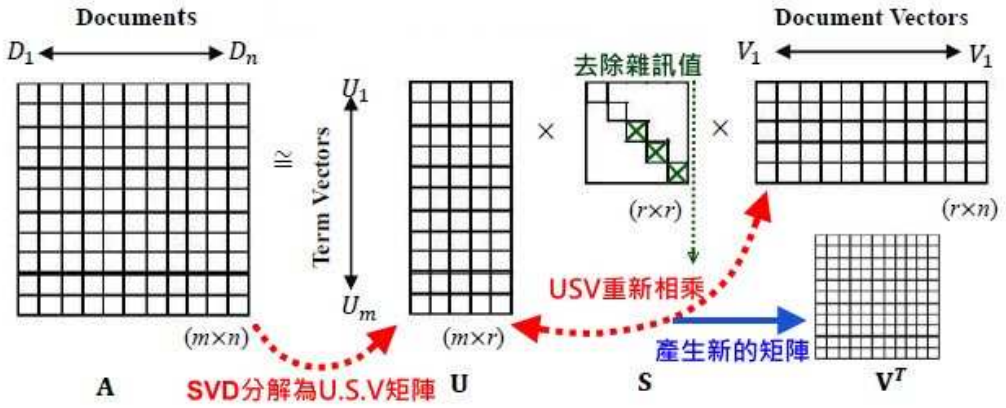


圖 5 三種字詞文件矩陣

(二) 機率潛在語意分析(PLSA)：

LSA 模型在文件和字詞上的呈現，並非以統計觀點出發，因此 Hofmann 提出 PLSA，模型如圖 6 所示，PLSA 核心為 EM 演算法(Hofmann 1999)。

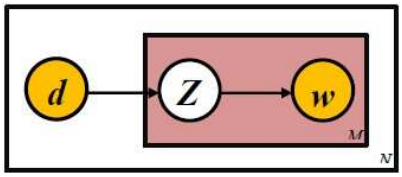


圖 6 PLSA 概念圖

PLSA 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(Hofmann 1999)。資料及由字詞-文件對 (d_i, w_j)，文件以 $d_i \in \{d_1, \cdots, d_N\}$ 表示，其個數為 N；另外，字詞以 $w_j \in \{w_1, \cdots, w_M\}$ 表示，字典相當於 M 個字詞所形成之集合。假設每一字詞在給定的文件中潛在主題 $Z_k \in \{Z_1, \cdots, Z_k\}$ 下產生。將字詞-文件對 (d_i, w_j) 共同出現(co-occurrence)

的聯合機率如下式：

$$p(d_i, w_j) = p(d_i)p(w_j|d_i) = \sum_{k=1}^{k} p(w_j|Z_k)p(Z_k)p(d_i|Z_k) \tag{5}$$

在 PLSA 模型中，文件經由 $p(w_j|Z_k)$ 的因子混合描繪其特性。將 Z 視為潛在變數，可以更容易地對 PLSA 模型利用 EM 演算法學習參數。最大化對數相似度可表示為：

$$\begin{aligned} L_n &= \sum_{i=1}^{N}\sum_{j=1}^{M} vsm(d_i, w_j)\log\{p(d_i, w_j)\} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} vsm(d_i, w_j)p(w_j|Z_k)p(Z_k)p(d_i|Z_k) \end{aligned} \tag{6}$$

其中 $vsm(d_i, w_j)$ 為本研究所產生的矩陣。在 EM 演算法的 E-step 中，利用目前估計的參數來計算潛在變數的事後機率，公式如下：

$$p(Z_k|d_i, w_j) = \frac{p(d_i|Z_k)p(Z_k)p(w_j|Z_k)}{\sum_{k=1}^{k} p(d_i|Z_k)p(Z_k)p(w_j|Z_k)} \tag{7}$$

在 M-step 中，利用潛在變數在 E-step 時的估測，使得連核對數相似度的期望值最大化。參數會更新如下：

$$p(d_i|Z_k) = \frac{\sum_{j=1}^{M} vsm(d_i, w_j)p(Z_k|d_i, w_j)}{\sum_{i=1}^{N}\sum_{j=1}^{M} vsm(d_i, w_j)p(Z_k|d_i, w_j)} \tag{8}$$

$$p(Z_k) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} vsm(d_i, w_j)p(Z_k|d_i, w_j)}{\sum_{i=1}^{N}\sum_{j=1}^{M} vsm(d_i, w_j)} \tag{9}$$

$$p(w_j|Z_k) = \frac{\sum_{i=1}^{N} vsm(d_i, w_j)p(Z_k|d_i, w_j)}{\sum_{j=1}^{M}\sum_{j=1}^{N} vsm(d_i, w_j)p(Z_k|d_i, w_j)} \tag{10}$$

經過 EM 演算法不斷的迭代後直到一個停止準則為止，最後的參數即為 EM 演算法所求得的區域最佳解(Chen 2011)。本研究將矩陣處理所產生的三種矩陣進行 PLSA 步驟，最終將得到 PDW 矩陣，形成如圖 7。

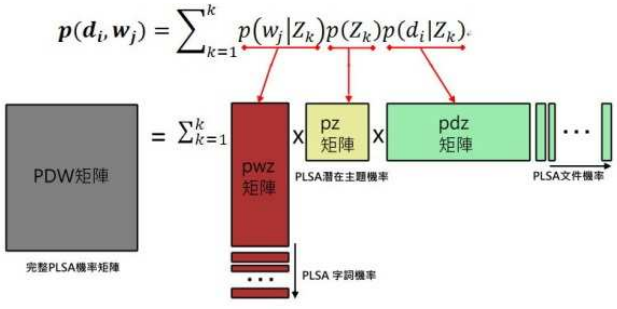


圖 7 PDW 矩陣形成示意圖

肆、研究結果

我們以程式來模擬實際應用的結果，並且與未實施語意模型的 VSM 矩陣、TFIDF 矩陣作比較本節首先會說明實驗所使用的軟硬體環境，接下來描述實驗資料，然後敘述採用的評估指標，最後則是實驗內容和結果。

一、實驗環境和資料

我們的模擬程式建置於個人電腦(PC)上，前置處理使用 PHP 撰寫程式碼、語意模型分析實驗利用 MATLAB 模擬。

實驗資料部分為 GoogleBlogSearch(US)查詢關鍵字的部落格文件結果，查詢關鍵字採用 40 個來自 GoogleZitgeist(Google 2011)、YahooBuzz(Yahoo 2011)、Bing Top2011 Search(Bing 2011)的熱門關鍵字以及選取 40 個 Dogpile 搜尋引擎(Dogpile 2011)提供的隨機關鍵字作為我們分析的資訊，每一個回傳頁面結果包含 10 篇部落格文件。每個查詢的關鍵字分別取 10、20、40、80、160、240、350 篇文件作實驗分析。

二、評估指標

本研究採用機器學習中的相似性度量來評估矩陣中向量之間的相似度，若相似度越高，則代表語意模型的效能越好，我們採用餘弦相似度(Cosine Similarity)、相關係數(Correlation coefficient)作為實驗結果的相似性度量。

(一) 餘弦相似度

餘弦相似度(Cosine Similarity)的度量方式是相似度研究領域的始祖，為文件分類中，最常被度量文件間距離的基本方法(Tan et al. 2005)主要以兩個相同基底與維度向量間的角度差距來度量兩向量之間的距離，計算結果會介於 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 和 B，其計算式如下：

$$Sim(A,B)=\cos\theta=\frac{A\times B}{|A|\times|B|}=\frac{X_1\times X_2+Y_1\times Y_2}{\sqrt{X_1{}^2+Y_1{}^2}\times\sqrt{X_2{}^2+Y_2{}^2}} \tag{11}$$

(二) 相關係數

在機率論與統計學中，相關係數(Correlation Coefficient)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，有許多根據數據特點而定義的用來衡量數據相關的係數，最常用的是皮爾遜積差相關係數。其定義是兩個變數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：

$$\rho_{XY}=\frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}=\frac{E((X-EX)(Y-EY))}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{12}$$

二、實驗設置

本研究有一個重要的門檻值，潛在主題 K，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 值的設定來觀察各個矩陣的結果。首先預設

K 值為 2、10、20、30、40、50、60、70、80、90、100，並採用文件數 160 的 VSM 矩陣，經過 LSA、PLSA 實行後，求得不同 K 值之下的平均餘弦相似度，圖 8 為實驗的結果，由於 PLSA 每次的運算都會產生不同的數值，因此我們反覆的進行五次 PLSA 求得平均值，最後我們決定採用相似度下降幅度較小的 K=2～K=50 作為本研究潛在主題個數的範圍。

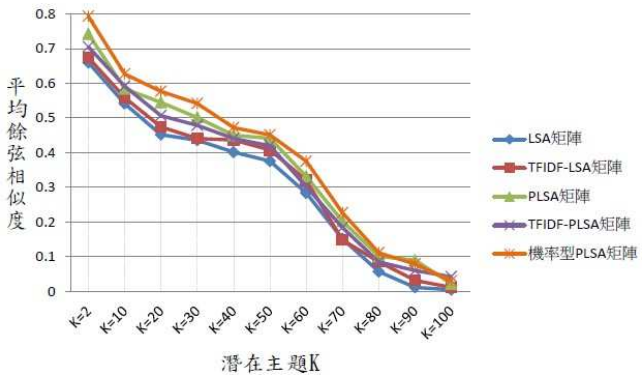


圖 8 決定門檻值範圍之實驗結果

三、實驗結果

首先我們比較 LSA、PLSA、TFIDF-LSA、TFIDF-PLSA、機率 PLSA 等矩陣進行差異比較，最後再將上述矩陣和未實施語意模型的 VSM 矩陣、TFIDF 矩陣進行比較。實驗分為熱門關鍵字、隨機關鍵字兩部分評比。

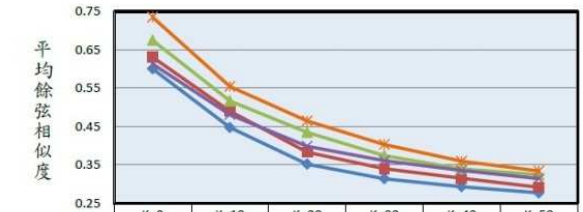
(一) 熱門關鍵字

圖 9 為熱門關鍵字的評比，可以觀察到不同的語意模型都呈現類似的趨勢，隨著 K 值越來越大，相似度越低，其中 LSA 的平均餘弦相似度、平均相關係數是最低的，而機率 PLSA 為最好的。

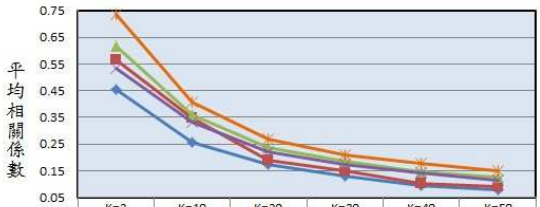
TFIDF-PLSA 在 K=2、K=10 時效能未比 TFIDF-LSA 好，但在 K>10 之後效能都略優於 TFIDF-LSA，這說明了 PLSA 的方法在潛在主題個數越多的情況下相較於 LSA 會有更好的效能。

我們可觀察到 TFIDF-PLSA 整體並未比 PLSA 好，影響因素為熱門關鍵字在 Google Blog Search 內的總回傳結果筆數極大，每個關鍵字都有上百萬筆，會使 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原始 PLSA 的效能還要好。

我們以不同文件數的角度來觀察相似度的結果，圖 10 為不同文件數下的餘弦相似度、相關係數，可看出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。

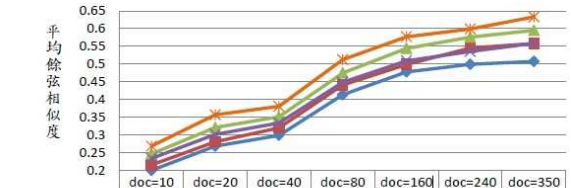


	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.60056	0.44753	0.35196	0.31400	0.29313	0.27712
TFIDF-LSA	0.63064	0.49008	0.38296	0.34017	0.31506	0.29162
PLSA	0.67479	0.51690	0.43483	0.37429	0.33952	0.32431
TFIDF-PLSA	0.61327	0.48115	0.39869	0.36093	0.33544	0.31451
機率型PLSA	0.73491	0.55447	0.46468	0.40320	0.35952	0.33432

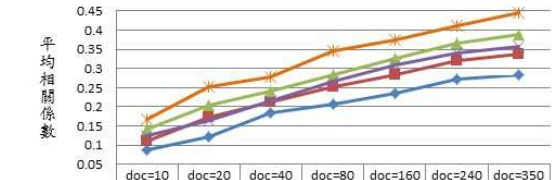


	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.45461	0.25635	0.17394	0.13207	0.09374	0.07958
TFIDF-LSA	0.56673	0.34649	0.19185	0.15217	0.10313	0.08992
PLSA	0.61862	0.36099	0.23893	0.18431	0.14514	0.12574
TFIDF-PLSA	0.53449	0.33240	0.22274	0.17504	0.14089	0.11545
機率型PLSA	0.73557	0.40782	0.27011	0.21022	0.17806	0.15075

圖 9 熱門關鍵字評比



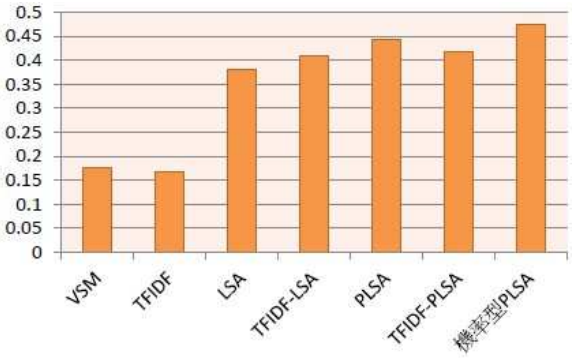
	doc=10	doc=20	doc=40	doc=80	doc=160	doc=240	doc=350
LSA	0.20013	0.26845	0.29885	0.41347	0.47757	0.49919	0.50736
TFIDF-LSA	0.21543	0.28093	0.31999	0.44084	0.49878	0.54498	0.55798
PLSA	0.24679	0.32094	0.35132	0.47482	0.54395	0.57596	0.59496
TFIDF-PLSA	0.23404	0.30191	0.33382	0.44869	0.50906	0.53527	0.55855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



	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

圖 10 不同文件數下的評比(熱門)

由圖 11 得到結果，PLSA 和各種加權過的 PLSA 語意模型應用於部落格文件搜尋上會比 LSA 及 TFIDF-LSA 有更好的效能。



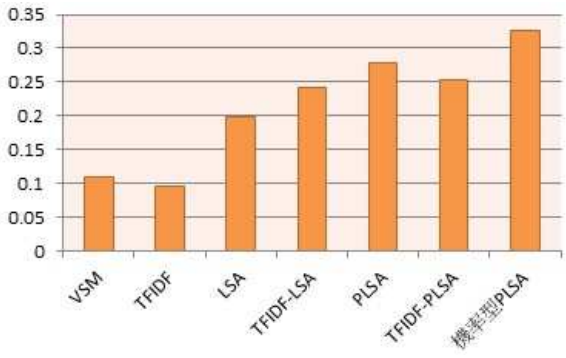


圖 11 熱門關鍵字的整體平均餘弦相似度(左)、相關係數(右)

本研究部落格文件中效能最低的語意模型 LSA，在圖 12 與文件前置處理得到的 VSM 原始矩陣、TFIDF 矩陣做比較，由圖中得知，搜尋熱門關鍵字時，經過語意模型實行的部落格文件，VSM 部分餘弦相似度可提昇 0.20477、相關係數可提昇 0.08834；TFIDF 部分餘弦相似度可提昇 0.21213、相關係數可提昇 0.10172，證實了經過語意模型處理的部落格文件會提昇不少的熱門關鍵字搜尋上的效能。

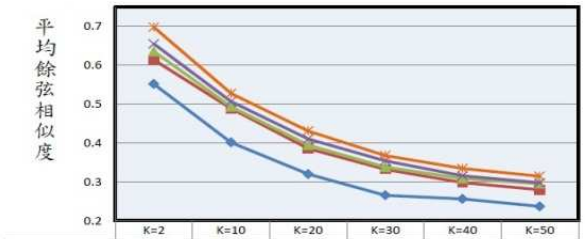
	餘弦相似度	相關係數
VSM	0.17595	0.11004
TFIDF	0.16859	0.09666
LSA	0.38072	0.19838

圖 12　LSA 與未實施語意模型比較(熱門)

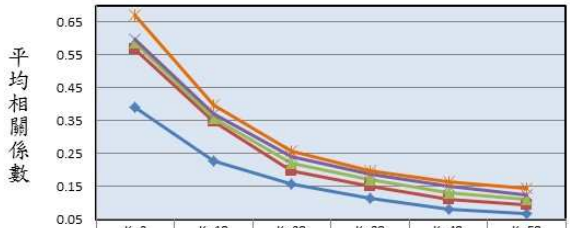
(二) 隨機關鍵字

圖 12 為隨機關鍵字的相似度評比，隨機關鍵字與熱門關鍵字呈現類似的趨勢，但整體平均餘弦相似度略微下降，我們認為影響因素為隨機關鍵字相較於熱門關鍵字，大部分為 2 字詞以上的多字詞，且 Google Blog Search 所得到的回傳頁面結果也較少，使得整體效能略低於熱門關鍵字。

但在 TFIDF-LSA、TFIDF-PLSA 部分，成長幅度相較於熱門關鍵字大，而 TFIDF-LSA 效能逼近 PLSA，TFIDF-PLSA 在相似度也超越了原始的 PLSA，因為隨機關鍵字不同於熱門關鍵字，總回傳結果只有數萬甚至數千，將使得 TFIDF 的 IDF(逆文件頻率)值偏小，經過 LSA 和 PLSA 的運算之後相似度有較顯著的成長。

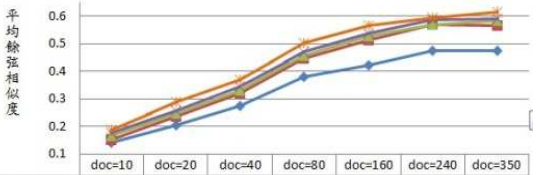


	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.55137	0.40120	0.31989	0.26527	0.25594	0.23684
TFIDF-LSA	0.61368	0.48843	0.38552	0.33200	0.29785	0.27953
PLSA	0.63517	0.49341	0.39474	0.33779	0.30847	0.29523
TFIDF-PLSA	0.65461	0.50654	0.41025	0.35393	0.31557	0.29755
機率型PLSA	0.69758	0.52719	0.43073	0.36760	0.33426	0.31420

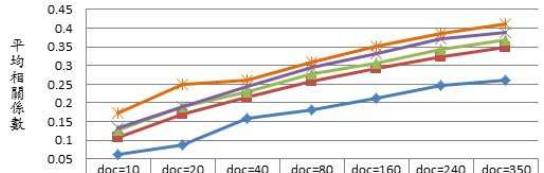


	K=2	K=10	K=20	K=30	K=40	K=50
LSA	0.39002	0.22944	0.15640	0.11597	0.08215	0.06691
TFIDF-LSA	0.56921	0.34754	0.19827	0.14979	0.11140	0.09622
PLSA	0.58659	0.35886	0.22003	0.17033	0.13138	0.11238
TFIDF-PLSA	0.59743	0.36953	0.24127	0.18940	0.15106	0.12558
機率型PLSA	0.67142	0.39697	0.25779	0.19868	0.16519	0.14549

圖 13 隨機關鍵字的評比



	doc=10	doc=20	doc=40	doc=80	doc=160	doc=240	doc=350
LSA	0.14013	0.20506	0.27485	0.37887	0.42029	0.47420	0.47554
TFIDF-LSA	0.15158	0.23499	0.31870	0.44664	0.51355	0.56726	0.56378
PLSA	0.16385	0.24557	0.33190	0.45711	0.52539	0.56867	0.58312
TFIDF-PLSA	0.17577	0.25811	0.34548	0.47042	0.53698	0.58573	0.58902
機率型PLSA	0.18749	0.28722	0.36803	0.50051	0.56478	0.59327	0.61552



	doc=10	doc=20	doc=40	doc=80	doc=160	doc=240	doc=350
LSA	0.06383	0.08901	0.15940	0.18261	0.21184	0.24690	0.26078
TFIDF-LSA	0.10810	0.16965	0.21671	0.25945	0.29141	0.32311	0.34940
PLSA	0.12904	0.18980	0.22868	0.27775	0.30590	0.34289	0.36877
TFIDF-PLSA	0.13388	0.18902	0.24521	0.29540	0.33146	0.37094	0.38741
機率型PLSA	0.17258	0.24916	0.26149	0.31051	0.35044	0.38637	0.41091

圖 14 不同文件數下的評比(隨機)

圖 14 轉成文件數的維度來看和熱門關鍵字有同樣的結果，文件數越大效能越好。由圖 15 可得到隨機關鍵字的整體平均餘弦相似度、相關係數結果和熱門關鍵字相同，另隨機關鍵字也進行了 LSA 與未實施語意模型的矩陣 VSM 和 TFIDF 進行效能比較，

圖 16 呈現出，經過語意模型實行的部落格文件，VSM 部分餘弦相似度可提昇 0.18952、相關係數可提昇 0.07681；TFIDF 部分餘弦相似度可提昇 0.15318、相關係數可提昇 0.04788，隨機關鍵字經過語意模型實行後效能也有顯著的提升。

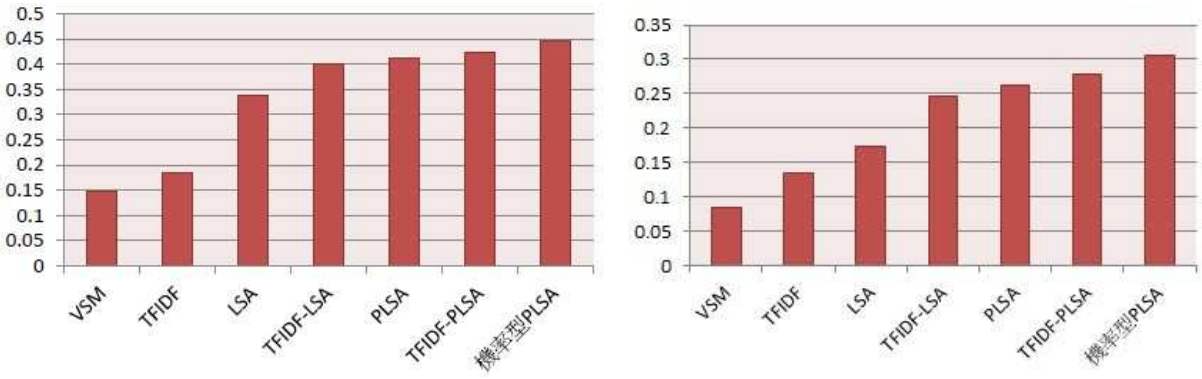


圖 15 隨機關鍵字的整體平均餘弦相似度(左)、相關係數(右)

圖 16 LSA 與未實施語意模型比較(隨機)

(三) 所有關鍵字比較

接續前面結果，我們將熱門關鍵字及隨機關鍵字合併比較，呈現如圖 17，我們可以觀察到除了隨機關鍵字的 TFIDF-PLSA 餘弦相似度略優於熱門關鍵字，其它的語意模型皆為熱門關鍵字較佳。而熱門關鍵字的 TFIDF-LSA 餘弦相似度和 LSA 約相差 0.0277 的效能，隨機關鍵字則相差 0.06108 的效能，隨機關鍵字的成長幅度較大；熱門關鍵字使用 TFIDF-PLSA 加權過後反而呈現下降的趨勢，相似度下降 0.02678，反之隨機關鍵字則提昇 0.01227 的效能，相關係數部分如圖 16 所表示。由此實驗結果可得知，經過 TFIDF 加權的 LSA、PLSA 可在隨機關鍵字部分表現的較好。

圖 16 熱門與隨機關鍵字的比較

伍、結論與建議

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(應用於線上部落格文件檢索的語意模型建議)。將原始的模型和 TFIDF 模型增加 LSA 和 PLSA 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處理的 VSM 矩陣和 TFIDF 矩陣效能優良，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)。

由於本研究是以 stop words 作為詞語切割的基準，因此可能會因為 Term 的切割使其缺乏完整的語意，因而影響到實驗的餘弦相似度和相關係數。未來可能在詞語切割採用 N 字詞(N-Gram)的方法解決。

研究限制部分，考慮到 GoogleBlogSearch 能擷取的網頁數量有一定的限制，所以本研究使用的文件集略小，然而網頁摘要並非完整的文件，未來可針對完整的部落格文件進行研究。另外因實驗時間的限制。本研究僅使用 LSA、PLSA 兩種語意模型作為基準分析，未來可加入更多樣化的語意模型進行模擬實驗。

參考文獻

1. Bing "Bing Top2011 Search", March 30,2012 (available online at http://www.bing.com/community/site_blogs/b/search/archive/2011/11/28/2011trends.aspx).
2. BlogScope "BlogScope", March 30,2012 (available online at http://www.blogscope.net/).
3. Brants, T., Chen, F., and Tsochantaridis, I. "Topic-based document segmentation with probabilistic latent semantic analysis," in: *Proceedings of the eleventh international conference on Information and knowledge management*, ACM, McLean, Virginia, USA, 2002, pp. 211-218.
4. Cai, L., and Hofmann, T. "Text categorization by boosting automatically extracted concepts," in: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, ACM, Toronto, Canada, 2003, pp. 182-189.
5. Chen, L. C. "Term suggestion with similarity measure based on semantic analysis techniques in query logs," *Online Information Review* (35:1) 2011, pp 9-33.
6. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. "Indexing by latent semantic analysis," *Journal of the American Society for Information Science* (41) 1990, pp 391-407.
7. Dogpile "The Project of Term Suggestion: Listing of Testing Queries", March 30,2012 (available online at http://cayley.sytes.net/li/listing_all_testing_keywords.php).
8. Fox, C. "A stop list for general text," *SIGIR Forum* (24:1-2) 1989, pp 19-21.

9. Google "Google Blog Search", (available online at https://www.google.com/blogsearch?hl=en).
10. Google "GoogleZitgeist", March 30,2012 (available online at http://www.google.com/zeitgeist/).
11. Hazel "PCRE - Perl Compatible Regular Expressions", March 30,2012 (available online at http://www.pcre.org/).
12. Hofmann, T. "Probabilistic latent semantic indexing," in: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, Berkeley, California, United States, 1999, pp. 50-57.
13. Hofmann, T. "Latent semantic models for collaborative filtering," *Acm Transactions on Information Systems* (22:1), Jan 2004, pp 89-115.
14. Judicibus, D. d. "World 2.0", March 28, 2012 (available online at http://www.lindipendente.eu/wp/it/2008/01/02/world-2-0/).
15. Kanejiya, D., Kumar, A., and Prasad, S. "Automatic evaluation of students' answers using syntactically enhanced LSA," in: *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing - Volume 2*, Association for Computational Linguistics, 2003, pp. 53-60.
16. Landauer, T. K., and Dumais, S. T. "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological Review* (104:2), Apr 1997, pp 211-240.
17. Landauer, T. K., Foltz, P. W., and Laham, D. "An introduction to latent semantic analysis," *Discourse Processes* (25:2-3) 1998, pp 259-284.
18. Madsen, R. E., Kauchak, D., and Elkan, C. "Modeling word burstiness using the Dirichlet distribution," in: *Proceedings of the 22nd international conference on Machine learning*, ACM, Bonn, Germany, 2005, pp. 545-552.
19. Nardi, B. A., Schiano, D. J., Gumbrecht, M., and Swartz, L. "Why we blog," *Commun. ACM* (47:12) 2004, pp 41-46.
20. Porter, M., and Boulton, R "Snowball:ALanguage for Stemming Algorithms", March 30,2012 (available online at http://snowball.tartarus.org/).
21. Salton, G., Wong, A., and Yang, C. S. "A vector space model for automatic indexing," *Commun. ACM* (18:11) 1975, pp 613-620.
22. Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to Data Mining* Addison Wesley, 2005.
23. Technorati "Technorati", March 30,2012 (available online at http://technorati.com/).
24. Yahoo "YahooBuzz", March 30,2012 (available online at http://yearinreview.yahoo.com/2011/us_top_10_searches#Top 10 Searches).

Using different semantic models to analysis online blog post

Lin-Chin Chen
Department of Information Management, National Dong Hwa University
lcchen@mail.ndhu.edu.tw

Chung-Cheng Wu
Master of Information Management Program, National Dong Hwa University
tiredapple@gmail.com

Abstract

In recent years, the online blogging community is growing bigger as a community network. 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. However, the blogger often suffer an information overload problem because the blog posts are updated frequently from different blog search engines. We have encountered synonym (two terms are syntactically different but semantically interchangeable expressions) and polysemy (a term has different meanings) problems when we search from the blog search engine. 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. LSA uses a truncated 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. According to the results of simulation analysis, we conclude that the semantic models can effectively be applied to the blog search engine.
Keywords: Semantic Analysis, Probabilistic Latent Semantic Analysis, Blog Search, Singular Value Decomposition.