混合式會議主題分類法

林柏安
國立成功大學資訊管理研究所
r76004069@mail.ncku.edu.tw

王惠嘉
國立成功大學資訊管理研究所
hcwang@mail.ncku.edu.tw

摘要

網路上的會議相關資訊很多，為了讓研究學者快速找到適合的會議資訊，本研究利用文字探勘技術將會議資訊分類，讓使用者可以容易找到適合自己的會議。因過去文獻的傳統分類演算法並非專門針對學術會議資訊做處理，故可能會造成分類錯誤的情形發生。本研究的目的為設計一為以學術會議網頁資訊的分類演算法。

考量學術會議資訊中，常會出現該領域的專有名詞，而這類的專有名詞又以複合字詞居多，因此在分析字詞重要程度時，會把這類情形納入考慮。另外，不同的分類演算法有各自的優缺點，因此本研究採用混合式分類，期望能整合傳統分類演算法，達到更好的分類效益。

關鍵詞：文字探勘、特徵選取、SVM、混合式分類

壹、緒論(Introduction)

隨著資訊科技的快速成長，越來越多的訊息，已經走向數位化的趨勢，並且由於網際網路的普及，也使得全世界的數位化資源得以快速的交流。這些眾多的數位化資源，其中一項便是學術會議的訊息。參加學術會議能幫助研究學者多瞭解整體研究領域的動態趨勢，更能在會議中透過與其他研究學者的接觸，激發出更多創新的想法(Gonzalez-Albo & Bordons, 2011)。因此許多想要深入瞭解該研究領域，卻又沒有特定會議想參加的研究學者們，通常會利用網際網路搜索相關的學術會議(Conference、Workshop 和 Symposium)，並從中挑選出若干適合其個人的學術會議參加。

然而在眾多數位化的學術會議資訊當中，研究學者們要搜索出一個真正適合自己參與的學術會議，大多是上網搜尋或是註冊某些定期提供學術會議資訊的網站，如 All Conference、Conference Alert 和 DB world 等，但並非所有學術會議都會將資訊註冊到上述的網站中，因此這些經由學術會議資訊網站所搜索出來的資訊，往往還需要再透過人工方式進一步檢索，例如以「羽毛球」作為關鍵字在搜尋引擎上尋找羽毛球的廠牌及價格，但搜尋出來的結果可能會含有羽毛球運動的規則或是技巧，而使用者則必須再進一步過濾，這無疑是一件費時又費力的工作。因此想要快速且準確搜索出研究學者們適合參加的會議資訊，其中一個方式就是加入文字探勘的技術。

文字探勘的領域包含了各式各樣與文字處理相關的技術，例如：自然語言處理(Natural Language Processing)(Moisl, 2011)、資訊擷取(Information Retrieval)(Moisl, 2011)、機器學習(Machine Learning)(Pham & Afify, 2005)和知識管理等(Knowledge Management)(Harding, Shahbaz, Srinivas, & Kusiak, 2006)。若想要幫助研究學者們提升搜索適合學術會議的速度，則需要先將學術會議的相關資料進行處理，並使用部分上述所提及的文字探勘技術，不過現今文字探勘的演算法中，尚未有相關研究提出能套用在此類型資料的演算法，因此大部分的研究學者還是只能透過人工的方式來獲得合適的學術會議資訊。

然而，從網路上所搜尋出來的資訊，通常是五花八門，內含許多不相關的資料(Middleton, Shadbolt, & De Roure, 2004)，往往需要再透過人工的方式加以過濾，才得以找出真正所需的資訊。而當研究學者們在搜索想參加的合適學術會議時，也面臨了同樣的困境。

目前已有一些定期提供學術會議資料的網站，像是 All Conference、Conference Alert 等。雖然這些網站可以免去研究學者自行上網輸入關鍵字主動查詢的麻煩，但這些網站所搜尋出的學術會議內容常常與研究學者欲得到的資訊有所差距，因為當中可能含有許多雜訊，例如在人文類別中卻包含資訊類別的會議資訊，因此同樣必須再用人工的方式去檢索這些資料，才能夠從中找出真正的需求圖 1，點選 Information Technology 類別，卻出現了關於 Industrial 的會議；另外，這些網站的資料來源，大多都是會議舉辦人或是使用者自行輸入，無法做到自動擷取之效，如圖 2 所示，故本研究的目的是為學術會議網站建置一適合其自動分類的演算法。

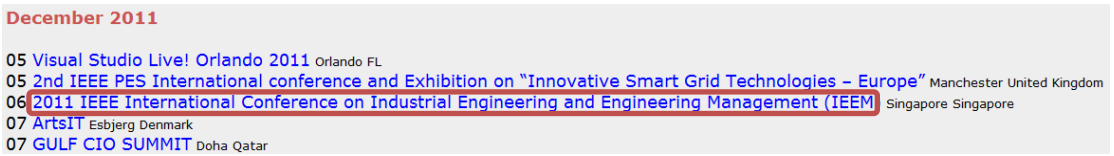

圖 1 分類錯誤示意圖

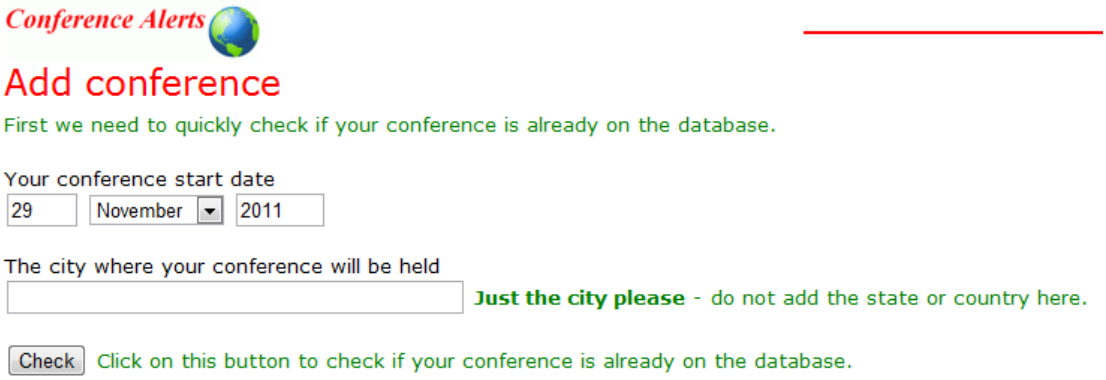

圖 2 新增會議事件功能圖

有鑑於上述狀況，文字探勘的技術正好可以處理這類需要特定需求且快速搜尋資料的問題，像是文件分群(Document Clustering)或是文件分類(Document Classification)等，而著名的演算法包括貝氏分類器(Naïve Bayes Classifier)、k 個最近鄰居法(k-Nearest

Neighbor, kNN)、決策樹(Decision Tree)以及支援向量機(Support Vector Machines, SVM)等(Joachims, 1998; Lewis & Ringuette, 1994; Maron, 1961)。現今這些傳統的演算法也經過後續其他學者的努力，有許多的不同改變或用途，像 Sun, Lim, and Liu (2009)應用 SVM 對比例失衡的資料做分類；Lu, Chiang, Keh, and Huang (2010)則是使用貝氏分類器與規則式資料庫替中文文字進行分類，但若將這些方法運用在學術會議類型的資料上，可能會產生一些問題：

（一） 傳統的演算法主要被套用在文章的分類處理，然而學術會議類型的資料跟文章的結構明顯不同，文章通常包含摘要、標題和本文內容，而學術會議類型的資料則有會議徵求的論文類型等，因此傳統的演算法可能不適合被使用於學術會議類型的資料上。

（二） 傳統的分類演算法主要是透過文章中重要的字詞來判斷，而字詞的重要性則是依據該字在文章或是標題中的出現頻率，但就學術會議類型的資料而言，較重要的字詞常常會是由兩個字所組成的專有名詞，可能會與傳統方法的考量不同，因此若採用傳統的分類方式來分類，極有可能造成分類成效不彰。

（三） 傳統分類演算法通常只採用單個分類器，但每個分類器都有各自適用的文章類型與優缺，若只使用一個分類器進行分類，可能會導致擁有多類型學術會議的資料分類錯誤。

針對上述問題，本研究將提出適用在會議資料的特徵選取方法以及結合不同分類法讓其分類效果更加彰顯。

本研究所蒐集的資料主要參考來源為 WikiCFP 網站上所記錄的學術會議連結，由於 WikiCFP 網站上所擁有的會議資料較為完整，故本研究將以 WikiCFP 網站做為主要資料來源。雖然網路上的學術會議資料包含了許多語系，但為了降低本研究處理會議資料的複雜度，再加上目前大部分的會議資訊以英文居多，故本研究方法將專注於英文的學術會議資料。

貳、文獻探討

一、 文件分類方法

文件分類是將所有收集的文件根據其標題和內容以進行分門別類，在之前這些文件分類的工作主要是用人力來完成的，但是隨著資訊科技的發達以及大量的文件資料湧現，分類問題逐漸由電腦來進行作業，目前常見的分類方式有：決策樹（Decision Tree）(Weiss, 1999)、k 個最近鄰居法（k-Nearest Neighbor, kNN）(Yang, 1994)、類神經網路（Neural Network, NN）(Ng, 1997)、簡單貝氏分類器（Naïve Bayes Classifier）和支援向量機（Support Vector Machine, SVM）(Dumais, 1998; Yang, & Liu, X., 1999)。以下將針對常見的分類方法進行簡單的介紹。

(1) 簡單貝氏分類器（Naïve Bayes Classifier）

貝氏分類器是以機率的方式來推薦分類的結果，採用 Bayes（貝氏）機率理論為基礎(Robertson, 1976)。貝氏分類器是屬於監督式的文件分類器，因其採用了監督式的學習

方式，故此分類器在進行分類前，必須先定義分類的類別，接著在訓練樣本的過程中，有效地處理測試資料分類。貝氏分類器利用貝氏決策準則(Bayesian decision rule)以及屬性間條件獨立的假設作為分類依據。根據貝氏定理，假設目前有 n 個屬性 $X_1, X_2, \cdots, X_n$，其中一筆資料 $x=(x_1, x_2, \cdots, x_n)$屬於第 j 類別值 C_j 的機率為：

$$P(C_j|x) = \frac{p(C_j, x)}{p(x)} = \frac{p(x|C_j)}{p(x)} \times p(C_j) \quad (1)$$

式子(1)的 表示某筆資料分類到類別值 C_j 的機率，也被稱為事後機率(posterior probability)，$p(x|C_j)$為概似機率(likelihood probability)，p(x)則為資料 x 出現的機率，比較不同類別的事後機率時，分母的部份同樣是此筆資料的出現機率，那麼 p(x)可以省略掉，因此式子(2)可簡化為：

$$p(C_j|x) \propto p(x|C_j) \times p(C_j) \quad (2)$$

根據貝氏定理的屬性條件獨立假設，將式子(7)展開：

$$p(C_j|x) \propto p(x_1|C_j) \times p(x_2|C_j) \times \ldots \times p(x_n|C_j) \times p(C_j) = \prod_{i=1}^{n} p(x_i|C_j) \times p(C_j) \quad (3)$$

若某一類別 C_j 的事後機率最大，則貝氏分類器中會預測該筆資料 x 的類別為 C_j。

(2) 支援向量機(SVM)

SVM 是由 Vapnik 在 1979 年提出的一種以統計學習理論(Statistical Learning Theory)為基礎的機器學習演算法，屬於監督式學習，須事先定義出所有類別，才能進行統計分類與回歸分析等，屬於一般化線性分類器(Vapnik, 1995)。SVM 是一個基於結構風險最小化定理所發展而成的一種分類方法，其主要是用來解決二元分類的問題，利用區分超平面（Separating Hyperplane）來分隔兩個或多個不同類別的資料，SVM 的概念是從特徵空間中找一個最佳的超平面，將之轉換成切割函數，並將訓練集的資料分成兩種類別，例如類別 A 和類別 B，類別 A 的資料會分布在超平面的同側，而類別 B 的資料會分布在另一側，當新資料的特徵向量輸入特徵空間後，就可藉此來判斷此新資料使屬於哪一個類別(Vapnik, 1995)。如圖 3 SVM 概念說明圖所示:

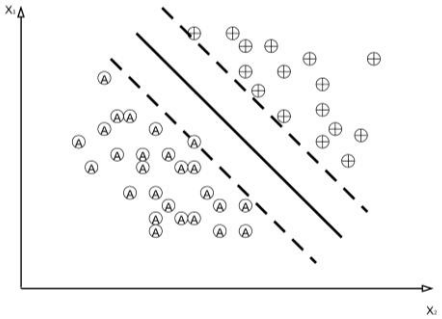

圖 3 SVM 概念說明圖

從上圖可以發現圖中的 Hyperplane(實線)，介於兩條 Hyperplane(虛線)之間，因為 SVM 的主要目的是找出能將兩群資料分的最遠的超平面。當 SVM 在一般線性可分割的狀況時，可直接使用超平面來進行分類；但是大部分實際問題通常是非線性可分割的狀況，因此針對這些狀況，Boser 等學者研究後採用核心函數來轉換型態，也就是將原始資料從低維空間轉換至高維空間，即可能有效達到線性分類的目的(Boser, Guyon, & Vapnik, 1992)。

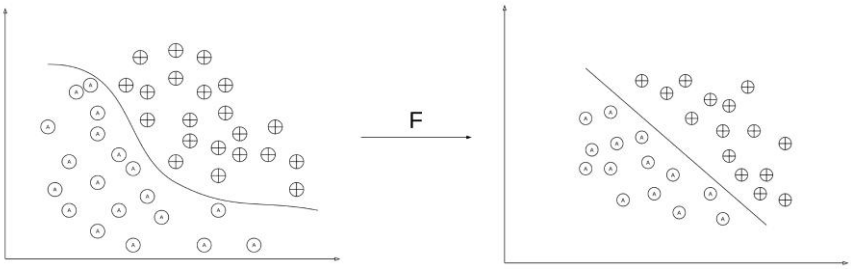

圖 4 核心函數轉換型態圖

如圖 4 核心函數轉換型態圖所示，圖左的原始資料是非線性的，因此要使用 F 函數(核心函數)將資料轉換到更高維的空間（圖右），使之能符合 SVM 的線性需求。

(3) 決策樹(Decision Tree)

決策樹是透過條列規則的方式，將不同文件分門別類，其最大的優點在於能提供視覺化的階層架構，清楚說明分類的準則(Debska & Guzowska-Swider, 2011)。決策樹中包含三種節點：根節點、內部節點和葉部節點，根節點位於整棵樹最上層的地方，葉部節點則位於整棵樹最底層的位置，界在根節點與葉部節點間的為內部節點。根節點與內部節點依照各個節點上的條件不同，可依序再往下細分，直到最後會碰到葉部節點，代表其最終分類結果。

二、 混合式分類法

混合式方法是指結合不同已存在的分類方法或模組，重新編排成一個新的演算法。(Govindarajan & Chandrasekaran, 2011)現今已有許多針對分類議題提出的混合式分類方法，如 Wu, Ken, and Huang (2010)曾提出以基因演算法為基底並結合 SVM 的混合式分類方法。混合式方法又可分成三種，順序式、嵌入式，以及平行式。順序式是將不同的方法或模組依序套用，彼此之間的方法沒有任何交集，單純套用方法而已；嵌入式則是將不同的方法加以融合，使得兩者合而為一，形成一個更為複雜的演算法；平行式則是將不同的方法同時套用，再透過其他方法將結果加以整合(Govindarajan & Chandrasekaran, 2011)。以 Govindarajan and Chandrasekaran (2011)提出的研究為例，該方法利用抽取不同樣本來建立決策樹，再將測試資料放入不同的決策樹模型中取得許多分類結果，接著用多數決的概念，統計出其最終分類結果。類似的作法還有 Zaghloul, Lee, and Trimi (2009)為擬定好的九個訓練資料類別，分別訓練出九個類神經網路，最後再透過這九個類神經網路做分類，結果顯示該方法大大減少類神經網路在訓練大量資料時所花費的成本。

參、研究方法

一、 研究架構

本研究整體架構如圖 5 所示，分為三大區塊：

(一)Conference Data 字詞處理模組：此模組將採用 WikiCFP 網站所搜尋出的 Conference Data 進行去除停用字前處理動作，接著再對處理過後的資料應用特徵選取的方法進行過濾。接著再進行剩餘的前處理動作，詞性標註以及字根還原等。此模組的目的為：得到經過處理後的會議資料。

(二)訓練資料特徵學習模組：此模組與上一階段處理的過程類似，先將訓練資料進行去除停用字、詞性標註和字根還原等動作，接著再對處理過後的資料進行本研究所提出的特徵選取方法，從中獲得特徵值。此模組的目的為：從訓練資料中得到各類別的特徵值。

(三)混合分類模組：此模組將上兩個模組產生的資料，放入不同的分類演算法中，得到由不同分類器所得出的分類結果，再將這些結果透過權重的方式，得出最終的分類結果。此模組的目的為：進行分類並得到最終結果。

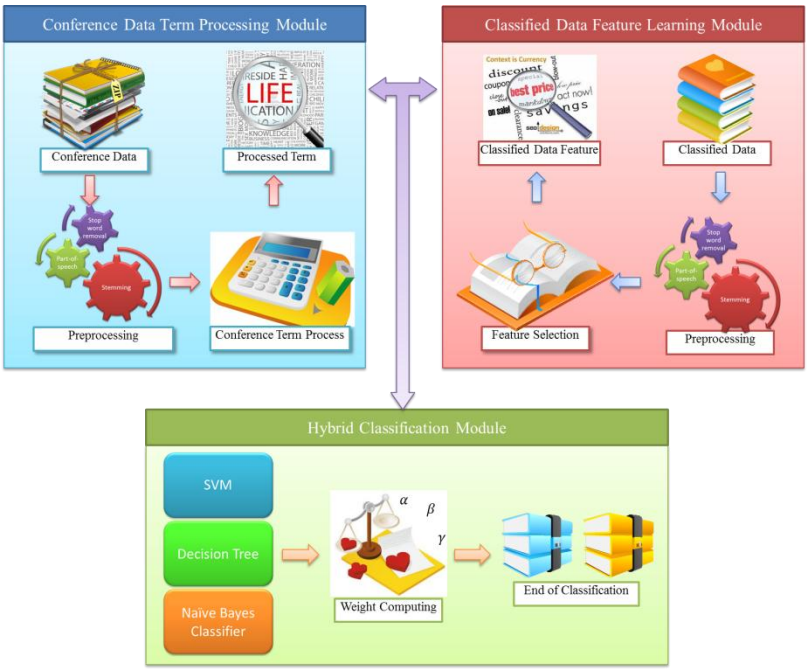

圖 5 研究架構流程圖

二、 資料蒐集與前處理

會議網站的資料龐大，在蒐集過程中很難將所有的網站資料全數收錄，又因 WikiCFP 網站上所擁有的會議資料較為完整，故本研究採用 WikiCFP 網站上的會議資料，作為資料收集的對象。本研究將蒐集會議網站相關的資料儲存於資料庫中，包含該會議欲召集的論文類型。本研究將會議資料集定義為 *CD*(Conference Data Set)，每個會議(Conference, *Conf*)中包含了許多欲召集的文章類型(Call For Paper, *CFP*)，故 $Conf_p \in CD$、$CFP_{pq} \in Conf_p$，其中 $Conf_p$ 為第 p 個 Conference，CFP_{pq} 為第 p 個 Conference 的第 q 個 Call For Paper。

資料蒐集完後，接著要先進行去除停用字動作，並得到與該會議相關的潛在特徵。另外訓練資料方面，本研究將採用兩種不同的訓練集來嘗試，一是從會議論文資料中分 fold，將一部分的資料作為訓練資料集；二是從目前學術文章收錄資訊較為完整的 THOMSON REUTERS (ISI) WEB OF KNOWLEDGE 網站中，蒐集大量文章的標題和摘要做為本研究訓練資料集。整份 THOMSON REUTERS (ISI) WEB OF KNOWLEDGE 資料(Data Set, *D*)中包含了 C 個類別(Class, *C*)，每個類別中又包含了 A 篇文章(Article, *A*)，文章中含有兩個項目，分別為標題(Title, *T*)以及摘要(Abstract, *Abs*)，因此本研究將每篇文章定義為 $A_{ij} = \{T_{ij}, Abs_{ij}\}$，其中 i 為所有資料集 D 中的第 i 個類別，j 為第 i 個類別中的第 j 份文章，故 $A_{ij} \in C_i$、$C_j \in D$。資料蒐集完後，接著要先進行前處理，包括去除停

用字、詞性標註和字根還原等動作，並得到與該會議或該類別相關的潛在特徵集。前處理過程分為三個階段，分別為去除停用字、詞性標註以及字根還原。

第一部分為去除停用字，由於文章中常常包含一些無助於分類的字詞，例如：定冠詞 a、an、the 等，所以必須將這些字詞一一去除，避免後續的龐大運算集造成分類準確率下降。

第二部分為詞性標註，由於在學術文件中的特徵，大多數以名詞為主，故經過停用字去除後的 CFP_{pq} 和 A_{ij} 會利用 POS 演算法進行詞性標註，並從經過標註後的結果中擷取出 $CFPpq$、T_{ij} 和 Abs_{ij} 屬於名詞的字詞(Word, W)。這裡先將 $CFPpq$、T_{ij} 和 Abs_{ij} 中所有的字詞分別定義為 CFP_W_{pqr}、T_W_{ijk} 和 $Abs\_W_{ijl} \in Abs_{ij}$，故 $CFP\_W_{pqr} \in CFP_{pq}$、$T\_W_{ijk} \in T_{ijk}$ 且 $Abs\_W_{ijl} \in Abs_{ij}$，其中 r 為 Call For Paper CFP_{pq} 中的第 r 個字詞，k 為標題 T_{ij} 的第 k 個字詞，l 為摘要 Abs_{ij} 的第 l 個字詞。透過 POS 演算法運算之後所得到的名詞結果如下所示：

$$CFPNoun_{pq} = \left\{CFP_{W_{pq1}}, CFP_{W_{pq2}}, \dots, CFP_{W_{pqr}} \middle| where\ CFP\_W_{pqr} \in noun\ and\ CFP\_W_{pqr} \in CFP_{pq}\right\} \quad (4)$$

$$TNoun_{ij} = \{T_{W_{ij1}}, T_{W_{ij2}}, \dots, T_{W_{ijk}} | where\ T_{W_{ijk}} \in noun\ and\ T\_W_{ijk} \in T_{ij}\} \quad (0)$$

$$AbsNoun_{ij} = \{Abs_{W_{ij1}}, Abs_{W_{ij2}}, \dots, Abs_{W_{ijl}} | where\ Abs_{W_{ijl}} \in noun\ and\ Abs\_W_{ijl} \in Abs_{ij}\} \quad (0)$$

第三部分為字根還原，主要工作是將所有字詞還原成字根。

三、 特徵選取

(一) 會議資料過濾

在本研究中，這些經過前處理後的潛在特徵不能直接當作分類的依據，因為有些潛在特徵本身可能只是碰巧出現在該類別中，例如「政治類」的新聞中，可能會有「政治人物」提到「棒球」，但「棒球」並不能成為「政治」類別的分類依據，然而這類碰巧出現的字詞可能也會出現於潛在特徵當中。

為避免上述情形發生，故本研究將使用字詞頻率作為會議資料的篩選工具，避免此類字詞出現於會議資料中，影響分類效果。底下將描述本研究的作法：

將每一個會議資料中所有的字詞 $Conf_p$ 建立會議字詞頻率陣列：

$$Conf\_TF_p = \{Conf_{pt}, Conf\_TF_{pt} | where\ Conf_{pt} \in Conf_p\} \quad (0)$$

其中 $Conf_{pt}$ 為會議 p 所擁有的 t 種字詞，$Conf\_TF_{pt}$ 為這 t 種字詞的字詞頻率。本研究將設定一門檻值 P，將 tf 值低於 P 的字詞予以刪除，剩下的字詞作為該會議的測試資料 $Conf\_TestData_p = \{W_{p1}, W_{p2}, \dots, W_{pn}\}$，$w_{pn}$ 為會議 p 所擁有的 n 個字詞。

(二) 訓練資料特徵選取

訓練資料集的部分，由於 THOMSON REUTERS (ISI) WEB OF KNOWLEDGE 所蒐集的資料相當龐大，但並非所有的文章都需要被列入訓練資料集中，若能將訓練資料作

篩選，則可在訓練過程更有效率。因此本研究將提出一個專門針對學術文件特徵選取的方法：

先針對每一個類別 C_i 計算出該類別中所有潛在特徵 $TNoun_{ij}$ 和 $AbsNoun_{ij}$ 的字詞頻率陣列，分別稱為 $CTNoun\_TF_i$ 和 $CAbsNoun\_TF_i$。

$$CTNoun\_TF_i = \{CTNoun_{im}, CTNoun\_TF_{im}\} \quad (0)$$

$$CAbsNoun\_TF_i = \{CAbsNoun_{in}, CTNoun\_TF_{in}\} \quad (0)$$

其中 $CTNoun_{im}$ 為類別 i 所擁有的 m 種潛在特徵，$CTNoun\_TF_{im}$ 為這 m 種潛在特徵的字詞頻率；$CAbsNoun_{in}$ 為類別 i 所擁有的 n 種潛在特徵，$CAbsNoun\_TF_{in}$ 為這 n 種潛在特徵的字詞頻率。在學術文件中，若只用一個字詞的字詞頻率作為特徵選取的工具，可能會使得部分冗字變成特徵值，影響分類準確率。學術文件上許多特定領域的字詞，常常是兩個甚至三個以上為一組，例如 Data Mining、Support Vector Machine 和 Decision Tree 等，故本研究認為有必要針對兩兩單字(Pairwise Words, PW)進行字詞頻率計算，才能使得這些字詞易於被辨識為特徵值。本研究以「text」作為關鍵字，在 THOMSON REUTERS (ISI) WEB OF KNOWLEDGE 網站搜索，並以前 1000 篇文章的關鍵字作為分析依據。如表 1 所示，在學術文章中的專有名詞，大多是以一個字或是兩個字為一組，其中又以兩個字的組合最多，故本研究將針對兩個字詞為一組的 PW 做字詞頻率。

表 1 學術文章專有名詞字數統計表

字數	個數
1	1270
2	1729
3	447
4	111
5	21
6	9
7	6
8	1

$$CTNoun2\_TF_i = \{CTNoun2_{im'}, CTNoun2\_TF_{im'}\} \quad (0)$$

$$CAbsNoun2\_TF_i = \{CAbsNoun2_{in'}, CAbsNoun2\_TF_{in'}\} \quad (0)$$

其中 $CTNoun2_{im'}$ 為類別 所擁有的 m' 種 PW 特徵，$CTNoun2\_TF_{im'}$ 為這 m'種 PW 的字詞頻率；$CAbsNoun2_{in'}$ 為類別 i 所擁有的 n' 種 PW 特徵，$CAbsNoub2\_TF_{in'}$ 為這 n' 種 PW 的字詞頻率。

有了以兩個單字為一組的字詞頻率後，接著要將字詞分開，同樣字詞的字詞頻率相加，例如 Data Mining 的字詞頻率為 9，Data Structure 的字詞頻率為 5，則將拆成 Data 的字詞頻率為 5+9=14，Mining 和 Structure 的字詞頻率分別為 9 和 5。本研究將之命名為 Double TF 方法(DTF)：

$$CTNoun2\_part\_TF_i = \{CTNoun2\_part_{im}, CTNoun2\_part\_TF_{im}\} \quad (0)$$

$$CAbsNoun2\_part\_TF_i = \{CAbsNoun2\_part_{in}, CAbsNoun2\_part\_TF_{in}\} \quad (0)$$

其中 $CTNoun2\_part_{im}$ 為類別 i 所擁有的 m 種潛在特徵，$CTNoun2\_TF_{im}$ 為這 m 種潛在特徵透過 DTF 所得到的字詞頻率；$CAbsNoun2_{in}$ 為類別 i 所擁有的 n 種潛在特徵，$CAbsNoun2\_TF_{in}$ 為這 n 種潛在特徵透過 DTF 所得到的字詞頻率。

接著將 $CTNoun\_TF_i$ 與 $CTNoun2\_part\_TF_i$ 合併；$CAbsNoun\_TF_i$ 與 $CAbsNoun2\_part\_TF_i$ 合併，得到此類別標題和摘要最終潛在特徵的分數：

$$CT\_TF_{im} = CTNoun\_TF_{im} + CTNoun2\_part\_TF_{im} \forall i = 1 \dots C, m = 1 \dots M \quad (0)$$

$$CAbs\_TF_{in} = CAbsNoun\_TF_{in} + CAbsNoun2\_part\_TF_{in} \forall i = 1 \dots C, n = 1 \dots N \quad (0)$$

接著計算出所有潛在特徵值的分數：將標題和摘要的最終潛在特徵分數相加總，得到 $C\_TermScore_{io}$，其中 o 代表第 i 類別中的第 o 個字詞。

若每一個類別都重複出現某一個字詞，即使這類的字詞在各類別中的 $C\_TermScore_{io}$ 很高，也不代表此一字詞足以作為特徵值，因此本研究將採用 idf 的概念，將這類字詞的分數降低。

$C\_idf_{io} = C\_TermScore_{io} \times \log_2 \frac{C}{C_{io}}$，其中 C 為所有類別數量，$C_{io}$ 為在類別 i 中第 o 個潛在特徵在所有類別中出現的次數。本研究將設定一門檻值 P'，讓 C_idf_{io} 高於 P'的潛在特徵作為類別 i 的特徵值 $Feature\_Set_i$。

四、 混合式分類模組

取得訓練資料和會議測試資料後，接著要將訓練資料放入分類器中訓練，才能為測試資料做分類，而本研究用來做測試的類別則是以 Science Direct 上的類別作依據。Govindarajan 等學者曾在 2011 年提出，混合式模組已慢慢開始發展用以合併現有方法，並增進其效能(Govindarajan & Chandrasekaran, 2011)。因此，考量不同演算法之間各有其優缺點，不同於過去的研究只單純使用一個分類器，在本研究中將採用混合式分類模組進行分類。

模組中包含了以下三種分類方法：決策樹(Decision Tree)、SVM 和貝氏分類器(Naïve Bayes Classifier)。K 個最近鄰居法也是許多研究中常採用的分類方法之一，但由於此方法在資料量較大時，會變得沒有效率，所以本研究並沒有採用 K 個最近鄰居法。

確定所需分類方法後，接著將訓練資料集和會議測試資料利用這三種分類方法進行分類，由於此三種方法是屬於各自獨立運作，故在混合式模組中是屬於平行式，其中 SVM、決策樹和貝氏分類器皆會產生一組分類結果，分別為 SVM_result_x、DT_result_x 和 NB_result_x。由於會議資料中共有 p 個會議，因此分類器會分別為這 p 個會議產生出 p 個分類結果，其中 x 為第 x 個會議。

在混合模組中，除了可以將分類器視為一模組外，也可同時加入其他的方法作為模組，例如多數決(Voting)、詞袋法(Bag of Word)等。以 Govindarajan 的方法為例，該方法將完整的訓練資料集隨機抽樣，分為 11 個子資料集，接著利用決策樹分別訓練，藉此

產生 11 顆決策樹模型，最後再將測試資料集放入 11 個決策樹中，產生出不同的分類結果，並以權重的方式，決定其最終分類結果。

本研究認為若單使用一種分類方法，可能會有部分資料因所選擇的演算法不同，讓最終分類結果產生錯誤，於是便修改 Govindarajan 在 2011 年提出的方法，採用不同分類器，並以仿照 Govindarajan 以權重的方式，為這三種分類方法分別給予不同權重，α、β和γ，其中$\alpha+\beta+\gamma=1$，而最終分類的結果，則會依照權重比例的不同而有變化接著為每一個分類計算出權重，公式如下：

$$weight_i = \alpha \times B_i^{svm} + \beta \times B_i^{DT} + \gamma \times B_i^{NB}$$

其中 $weight_i$ 代表的第 i 個類別的權重，而變數 B 則是代表該分類方法的結果是否屬於類別 i，如果是則為 1，如果不是便為 0，假設 SVM 得出的分類結果為類別 A，則B_A^{svm}=1。最終的分類結果，則會依照這三個分類結果組合，取出所佔權重最高的答案當作最後分類結果：

$$C_{final} = \arg Max[weight_i]$$

其中C_{final}為最後分類的結果。

最終的分類結果，則會依照這三個分類結果組合，取出所佔權重最高的答案當作最後分類結果，如表 2 所示：

表 2 混合式分類範例圖

分類方法	權重	分類結果	最終結果
SVM	0.6	A	A：0.6
Decision Tree	0.2	B	B：0.2
Naïve Bayes Classifier	0.2	C	C：0.2 **結果：A**

但若最後分類的結果，擁有最高權重的答案不只一種，則將該筆資料的答案定為多分類，意即最終分類結果會有兩個以上。

肆、結論與討論

本研究利用三個模組「Conference Data 字詞處理模組」、「訓練資料特徵學習模組」以及「混合分類模組」為學術會議網站進行分類。基於學術文件的專有名詞通常為兩兩字詞為一組，故本研究提出一適用於學術文件的特徵選取方法。最後再輔以混合分類模組，使用不同分類演算法進行分類。

為了解本研究的特徵選取方法是否優於過去傳統的特徵選取方法，故本研究將以傳統的 TF、DF 與 TF-IDF 特徵選取方法與本研究所提出的 Double TF(DTF)與 DTF-IDF 做比較。

在此實驗中，本研究以會議資料當作自身的訓練資料，採用 K 次交叉驗證法，其中 K=10，取出各特徵選取方法前 5%、10%、15%及 20%的字詞作為特徵值，並且使用三種分類方法—SVM、貝氏分類器和決策樹進行分類，而本研究所使用的決策樹演算法為 C4.5(J48)，SVM 的 Kernel Function 為 Radial Basis Function(RBF)，並預期 DTF 與 DTF-IDF 的特徵選取方法可以優於傳統的 TF、DF 以及 TF-IDF 等方法。

參考文獻

1. Boser, B. E., Guyon, I. M., & Vapnik, V. N. *A training algorithm for optimal margin classifiers*. Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, United States, 1992.
2. Debska, B., & Guzowska-Swider, B. Decision trees in selection of featured determined food quality. *Analytica Chimica Acta, 705*(1-2), 2011, pp. 261-271.
3. Dumais, S., Platt, J., Heckerman, D., & Sahami, M. *Inductive learning algorithms and representations for text categorization.* . Paper presented at the Paper presented at the Proceedings of the seventh international conference on Information and knowledge management, Bethesda, Maryland, United States, 1998.
4. Gonzalez-Albo, B., & Bordons, M. Articles vs. proceedings papers: Do they differ in research relevance and impact? A case study in the Library and Information Science field. *Journal of Informetrics, 5*(3), 2011, pp. 369-381.
5. Govindarajan, M., & Chandrasekaran, R. M. Intrusion detection using neural based hybrid classification methods. *Computer Networks, 55*(8), 2011, pp. 1662-1671.
6. Harding, J. A., Shahbaz, M., Srinivas, & Kusiak, A. Data mining in manufacturing: a review American Society of Mechanical Engineers (ASME). *Journal of Manufacturing Science and Engineering 128*(4), 2006, pp. 969–976.
7. Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. *Lecture Notes in Computer Science, 1398*,1998, pp. 137-142.
8. Lewis, D. D., & Ringuette, M. A Comparison of Two Learning Algorithms for Text Categorization. *In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval* , 1994, pp. 81-93.
9. Lu, S.-H., Chiang, D.-A., Keh, H.-C., & Huang, H.-H. Chinese text classification by the Naive Bayes Classifier and the associative classifier with multiple confidence threshold values. Knowledge-Based Systems, 23(6), 2010, pp. 598-604.
10. Maron, M. E. Automatic Indexing: An Experimental Inquiry. *Journal of the ACM (JACM), 8*(3), 1961, pp. 404 - 417.
11. Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. Ontological user profiling in recommender systems. *Acm Transactions on Information Systems, 22*(1), 2004, pp. 54-88.
12. Moisl, H. Finding the Minimum Document Length for Reliable Clustering of Multi-Document Natural Language Corpora. *Journal of Quantitative Linguistics, 18*(1), 2011, pp. 23-52.
13. Ng, H. T., Goh, W. B., & Low, K. L. Feature selection, perceptron learning, and a usability case study for text categorization. *SIGIR Forum, 31*(SI), 1997, pp. 67-73.

14. Pham, D. T., & Afify, A. A. Machine learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Journal of Engineering Manufacture: Part B 219*, 2005, pp. 395–412.
15. Robertson, S. E., & Jones, K. S. Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*(3), 1976, pp. 129-146.
16. Sun, A., Lim, E.-P., & Liu, Y. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems, 48*(1), 2009, pp. 191-201.
17. Vapnik, V. N. The nature of statistical learning theory: Springer-Verlag New York, Inc, 1995.
18. Weiss, S. M., Apte, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T., & Hampp, T. Maximizing Text-Mining Performance. IEEE Intelligent Systems Retrieved 4, 14, 1999.
19. Yang, Y., & Liu, X. *A re-examination of text categorization methods.* Paper presented at the Paper presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval,, Berkeley, California, United States, 1999.
20. Zaghloul, W., Lee, S. M., & Trimi, S. Text classification: neural networks vs support vector machines. *Industrial Management & Data Systems, 109*(5-6), 2009, pp. 708-717.

Abstract

To find out the suited conference information efficiently for researcher, this study will classify the conference by text mining. The previous references of traditional classification algorithm did not classify documents of conference information, so when we classify these academic documents, we may get some incorrect answers. The goal of this study is designing a classification algorithm for conference information.

There are many terminology nouns or phrases in the conference, and most of them consist of two words. Therefore, when we analyze the importance of the terms, we should take this situation into consideration. Moreover, there are pros and cons in different existing classification algorithms, so the hybrid classification is adopted to integrate the traditional algorithm.

Keywords: text mining, feature selection, SVM, hybrid classification