文件主題自動分類成效因素探討

Effectiveness Issues in Automatic Text Categorization

曾元顯 輔仁大學 圖書資訊學系

「中國圖書館學會會報」, 2002 年 6 月, 第 68 期, 頁 62-83

摘要：

文件主題分類乃是根據文件內容給定類別的工作。文件分類不僅是圖書館學中資訊組織與主題分析的重要課題，也是現今知識管理主要的應用工具。透過文獻的回顧、比較，與筆者的經驗，本文整理了 12 項跟分類成效有關的因素進行探討。有些因素在不同的學者間有不同的看法或結論，本文一一評論並舉適當事例加以佐證。部分重點列舉如下：我們的實驗發現，前置摘要處理，對較差的分類器有幫助，但對較好的分類器則會減低其成效。其次，本文舉出分類不一致為普遍現象，來說明類別選擇的原則，並從筆者的經驗中提醒讀者如何解讀不同數據的比較結果。此外，成效評估方式不應只依賴傳統的精確率與召回率，應當去瞭解使用者真正的需求與目的，才能設計、調整出最佳的分類系統。最後，機器分類似乎比人工分類有較高的一致性，因此自動分類系統可以輔助人工分類，一起達到低成本、高精確的要求。文章中也將簡要提到一些自動分類系統實際應用的情況及其效益。

Abstract:

Text categorization or document classification is a task to assign predefined labels to documents based on their content. This article discusses twelve issues that are related to the effectiveness of automatic text categorization based on a literature review and the author experiences in this task. Parts of the conclusions include: (1) Automatic summarization, based on our experiments, does not improve classification effectiveness, as opposite to some previous studies. (2) Inconsistency (in assigning labels) seems inevitable in this task and thus may adversely affect the effectiveness of better classifiers, especially when the test documents are highly inconsistent (3) Performance measure should go beyond traditional precision/recall metrics. Some cost functions should be considered to better meet users' needs. (4) Machine classifiers seem to be more consistent than humans in assigning labels to documents. Thus classifiers can aid human experts to do the job in a cost-effective way, as is manifested in 2 reported real-world cases.

關鍵詞：文件分類、成效分析、知識管理

keywords: document classification, effectiveness analysis, knowledge management

壹、前言

「文件主題分類」或簡稱「文件分類」（document classification or text categorization）是指依文件「內容主旨」給定「類別」（class or category）的意思。例如，新聞文件可按其報導的內容，給予「政治」、「外交」、「娛樂」、「運動」等類別。通常，這些類別都是事先定義或選定，以符合管理者的需求與期望。而給定類別的工作，傳統上都由人工閱覽文件，根據其主題大意，給予適當的類別標示。文件分類是圖書館學中資訊組織與主題分析領域裡一項非常重要的課題[1]，也是現今知識管理主要的應用工具之一[2]。

文件分類的目的，在對文件進行分門別類的加值處理，使得文件易於管理、利用。例如圖書館中的書籍按學科領域分類，不僅提供館員便於上架排列，也方便使用者瀏覽借閱。分類後的文件，並可提供使用者依主題查找文件而不受文件用詞的限制。例如：討論「文化傳承」主題的文件，不見得都會使用「文化」或「傳承」這些詞彙，像古蹟維護的議題或傳統習俗的活動報導都跟「文化傳承」的主題有關，但用詞卻不相同。另外，文件分類後，還可顯示館藏文件的主題分佈與範圍，對館藏文件的後續徵集，提供重要的決策參考。

文件分類不僅在圖書館中有大量的應用，在全球資訊網出現後，對使用者尋找網路資訊的協助，也扮演非常重要的角色。例如，大部分的入口網站，像「奇摩」、「雅虎」都聘請大量的人員進行文件分類，以提供網站或網頁分類目錄的服務。以「雅虎」而言，其搜尋系統甚至委外建置，過去數年來從 Infoseek 換成 Altavista 再換成 Google，但是其分類目錄則由本身持續不斷的維護，足見分類對其企業本身的價值與重要性。

近年來，拜資訊技術普及運用之賜，各個企業與機構的數位文件不斷累積，數量大到難以有效的管理與利用，文件分類的需求也就因應而生。為此，如何利用自動化的技術，快速有效的協助人工分類，來應付大量暴增的分類需求，是現今資訊服務與知識管理的重要課題。

文件分類自動化後，會帶出更新、更便利的應用方式，除了提供館藏瀏覽（collection browsing）、主題檢索（topic-based retrieval）、文件管理（歸檔、調閱、分享）外，還可應用在網頁過濾、電子郵件過濾、資訊選萃（SDI, Selected Dissemination of Information）、資訊配送（information filter or routing）、甚至是文字探勘（text mining）、新知發掘（knowledge discovery）、知識管理（knowledge management）等領域。例如，企業內情郵件外流阻絕、公共領域（政府單位、圖書館）瀏覽器的色情、暴力網頁過濾等，可透過文件分類，將內容不適當的文件標示、阻絕起來。又如，使用者長期穩定的資訊需求，可表達成某些主題類別，

由系統自動分析每日新進文件的內容，自動發佈配送，免除使用者需要常常主動查詢的困擾。再者，把各個學校、機構、企業求才的網頁、公告蒐集分類後，可以協助發現最近人才需求的趨勢，做為個人或企業專才養成的參考，甚至競爭產業的前景與景氣的榮枯，都可從中獲得參考訊息。另外，將文件分門別類，輔以一些統計功能，可讓使用者容易發現某些趨勢、提供下達決策的判斷依據。像客戶投訴的電子郵件，依產品與投訴類型分類後，可以得知哪些產品的投訴郵件最多、甚至投訴的問題類型。企業依此訊息可在短時間內發現問題產品以及問題型態，繼而找出原因快速改善。知識管理的一項主要課題是知識分享，將不同專業人員的著作文件集中後按主題分類存放，將有助於不同專業的人發現彼此的文件與專精知識，從而提升知識分享的效率。學術界的學刊、會議論文集便是最好的例證，這些我們習以為常的期刊論文發表與傳播方式，造就了百年來學術成就的突飛猛進。類似的機制若能深入到各個機構企業，將有助於提升整體的競爭力。

上述這些應用，需要具備高效率的自動分類機制，在短時間內蒐集、分類完成大量文件，才比較有決策參考與應用的價值。過去數年，國內外有關文件自動分類的研究相當豐富 [3-23]。雖然新的技術、方法、報告常常出現，但筆者仍觀察到幾個現象：

一、 剛接觸自動分類問題的研究者，如研究生、工程師，對某些觀念不是很清楚，在製作或應用自動分類系統時，常事倍功半；
二、 即便是經驗豐富的學者，對於影響自動分類成效的因素，某些觀點彼此間的認知與看法並不一致，甚至同一學者對相同的問題，也曾有前後看法不同的情形發生；
三、 影響分類成效的因素眾多，一一檢討、實驗驗證，非常耗時費力，很少研究能夠全面照顧到每個因素，但有很多研究已嚴謹的討論到部分的個別因素；
四、 自動分類系統的使用者，對系統的運作不是很清楚，以致難以發揮系統最大的成效。

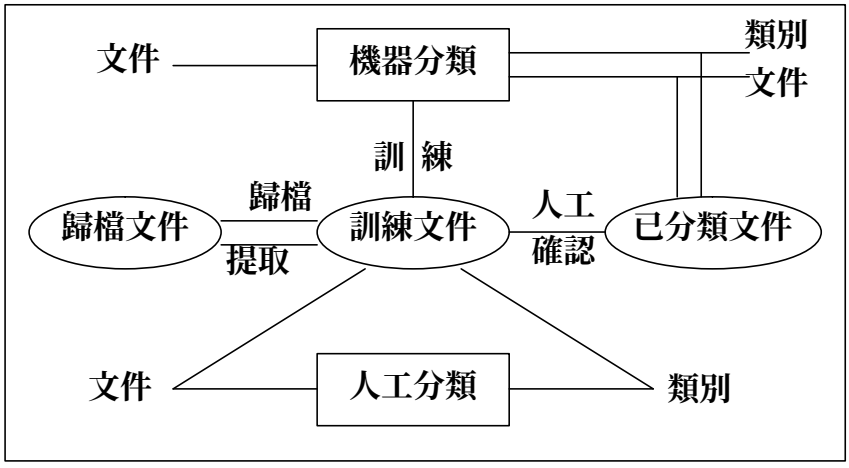
因此，綜合各方的研究、討論，加以整理、比較，以釐清問題，有其必要。

本文的目的，不在提出一套新的自動分類方法、報告其分類成效，而是基於上述的緣由，來探討自動分類成效的各項議題。文中除了以文獻回顧與評論的方式做說明、比較外，也加入筆者自行發展與應用自動分類系統的經驗，期使讀者能對文件自動分類的問題有深入的瞭解，進而能夠有效的製作與應用這些系統，提升資訊服務與知識管理的效能。

貳、 文件自動分類

文件分類，需要瞭解文件的主題大意，才能給定類別，因此是相當高階的知識處理工作。要將文件分類自動化，必須先整理出分類時的規則，電腦才能據以執行。然而，有效的分類規則通常難以用人工分析歸納獲得。因此，機器在做自動分類之前，還必須加以訓練，使其自動學習出人工分類的經驗與知識。

所謂訓練，就是讓機器去分析一堆「訓練文件」，如圖一所示。訓練文件記錄了人工做文件分類的知識，這種知識相當隱晦，只是一堆（文件=>類別）的對應記錄。機器在反覆的閱讀文件以及其標示的類別後，自動歸納出一些對應規則，使其下次看到類似的文件時，可以給出適當的類別。



圖一：自動分類流程圖。

機器做文件分類，需要測試其成效。學術界常用的分類測試集（test collection）包括：Reuters、OHSUMED、20NG 等人工已分類好的資料。Reuters 為路透社 1987 年的新聞資料，由 David Lewis 分別於 1990 及 1996 年整理而成，包含 21578 篇新聞，每篇文件有 1-5 個路透社人員選定的類別，全部共有 135 個類別。OHSUMED 則由 William Hersh 及其同事整理而成 [24]，共包含 1987 年到 1991 年間 233,445 篇的醫學文獻，這些文獻都含有篇名與摘要，以及平均 13 個取自 MeSH（Medical Subject Headline）分類表的類別，全部用到 14,321 個類別。20NG 為 20 newsgroups 的縮寫，包含將近 20,000 篇文章，均勻的分佈在 20 個 UseNet 的討論群，其中大部分的文件只有一個類別，只有 4.5%的文件有兩個或兩個以上的類別。這些分類測試集通常一部份（例如 70%）被選用來訓練分類器，稱為訓練文件，另外的一部份（30%）則用來測試分類器的成效，稱為測試文件。在實際運用時，不必拘泥於這種比例，在系統效能能夠忍受的情況下，訓練資料通常是用得越多越好。

現今自然語言理解的技術，還無法讓電腦瞭解任意的自由文句。因此機器在分析文件時，常將文件分解成一個個語意較小的單位，通常為文件的關鍵詞彙，或稱「特徵詞彙」，再從這些詞彙與類別中找出對應的關係。當類別名稱與特徵詞彙剛好一樣時，這種分類工作就比較容易，經常一個關鍵詞，就足以決定文件的類別。但主題分類通常不這們簡單，其類別的定義常常是較為抽象的主題。例如「天災」，一篇報導豪雨過後出現淹水、土石流的新聞，便需根據「豪雨」、「淹水」、「土石流」等詞彙，來判定其為「天災」類別，而不是只根據是否出現「天災」或「天然災害」來判定其類別。

有時候類別的定義會抽象到無法事先知道其意義。例如：討論高科技晶圓製程的文件，歸為「A」類，其他的文件歸為「B」類。至於 A 類是什麼意義，可視其後續的應用而定。有時候可以說 A 類就是「機密的」文件，B 類不是；或者說 A 類是「高價的」文件，而 B 類是屬於「低價的」文件。這時文件還是依照其內容大意作分類，但是卻標示為 A 或 B 這種沒有意義的標籤。所以，文件分類也可以說是替文件貼上標籤（labeling）的工作。

另一個跟「文件分類」（document classification or text categorization）很像的問題是「文件歸類」（document clustering）。文件歸類的意義是將文件按內容主題的相似度歸納分群，而不需依照某些事先給定的主題或類別來聚集文件。其目的之一是在發覺一堆文件中所包含的各種事件。例如，把一週的新聞作歸類，可以得知該週有哪些主要的新聞事件。文件歸類也可以自動學習做到，由於沒有人工事先設定類別，因此其學習方法稱為非監督式學習（unsupervised learning）。相對的，由於自動分類是按類分群，因此其學習方法稱為監督式學習（supervised learning）。Roussinov 與 Chen [25] 曾引述 Duda 與 Hart 書上 [26] 的說法，說「歸類」比「分類」難：

The distinction is that with supervised learning we know the state of nature (class label) for each sample, whereas with unsupervised learning we do not. As one would expect, the problem of unsupervised learning is the more difficult one.

但事實上，歸類是只要文件夠相似，就可以放在一起，但分類是根據類別的定義要求，將文件放在一起。有時候高科技晶圓製程的文章與軍事消息報導同屬於「機密類」，但這兩種文章，卻不見得會有任何相似的內容。也就是說，（文件=>類別）的對應可以是任意規定、任意複雜的。因此筆者認為，分類不見得是比歸類簡單的問題。

通常機器學習出來的分類規則分為兩種：一種是符號式的（symbolic），一種是數值式的（numeric）。符號式的規則長相大都像：「IF conditions, THEN categories」，即符合某些布林條件，就分為某些類別，此種規則人工大多能理解，甚至可以加以修改。數值式的規則，則只是一堆詞彙字串及其數值權重，再加上

某種結合這些數據的運算公式，因此人工大多難以理解、無從修改。符號式的分類器有 C4.5、RIPPER、sleeping experts [27]、Swap-1 [28]等，而數值式的分類器有 SVM（Support Vector Machine）、KNN（K-Nearest Neighbors）、LLSF（Linear Least Square Fit）、Perceptron、Neural Network、Naï ve Bayse 等 [29]。

很多研究嚐試提出不同的方法，讓自動分類達到最高的成效。然而影響機器學習與分類成效的因素很多，超過十數個，逐一檢討某個因素，以實驗驗證其對成效的影響，將是相當耗費時間與成本的作法。例如：Bekkerman 等人以 SVM 分類器運用在 20NG 的實驗驗證時間，需要電腦連續四日不停的計算 [30]，更何況每項因素都有數種變項。一一比較不同變項的組合狀況，將有上千種實驗待驗證比較，所需時間將超過數千甚至上萬個小時。既然過去已有相當多的研究，各自就某些影響成效的部分因素做過探討，本文便根據這些研究結果，加上筆者的經驗，來討論自動分類的成效問題。

參、 成效因素分析

根據過去的文獻，把跟自動分類成效有關的因素歸納後，將其整理成 12 項，分別討論如下：

一、特徵選擇（feature selection）

如前所述，自動分類系統會先將文件分解成一個個語意較小的單位，至於這個單位要如何選擇，就是「特徵選擇」的問題。英文的文件，只要依空格將文件分解成一個個詞（word），再進行學習、分類，就有不錯的成效。但 Al-Kofahi 等人 [31] 等人，為了對付 13,779 個類別的分類問題，選擇了 word pair 當特徵詞彙。這是因為他們要分類的法律文件用詞都很相似，但卻要非常精細的分成 13,779 類，光用英文的 word 不容易區別出不同的類別，因此把出現在鄰近的詞彙組成一對一對的 word pair 當作文件的特徵，來增加區別能力。例如：「離家少年遭竊」跟「離家少年偷竊」，兩者只有一詞之差，但前者可能要分在「偷竊報案」的類別，而後者可能要分在「少年犯罪」的類別。如果特徵詞只有選出（離家，少年，遭竊，偷竊）這四個詞，則兩者相似度達 2/3，卻必須分在不同類別。但如果特徵詞擴展到（離家，少年，遭竊，偷竊，”離家-少年”，” 離家-遭竊”，” 少年-遭竊”， “離家-偷竊”，”少年-偷竊”），兩者相似度則降到 3/6，有助於區別不同類別的文件。

對中文的文件而言，可以選擇詞彙作為特徵詞，但因為中文沒有空白斷開詞彙，需要做斷詞的工作。而斷詞的過程，常需要字典輔助，但字典無法收納所有的詞彙，因此會有未知詞的問題需要處理。為了避免這些問題，有時可以選擇

n-gram 作為特徵詞。N-gram 是文件中任意連續的 n 個字的字串，雖然大部分的 n-gram 沒有意義，但 n-gram 仍能抓住文件的用詞，可以有效的代表該文件。

有時候對訓練文件效果最好的特徵詞彙，對未來的待分類文件其效果並不一定最好。例如，當 n-gram 的 n 取非常長時，則幾乎每篇文件取一個 n-gram 就足以代表該篇文件。當標題被視為文件的一個特徵詞時，則一篇文件只要一個標題來代表該文件即可。在分類系統看過、學過這樣的訓練文件後，拿訓練文件本身做分類，很容易就可達到百分之百的正確率。但對沒有看過、學過的待分類文件，則幾乎沒有分類正確的可能。短的 n-gram，比較容易出現在未來的文件中，但不容易用來區分類別。相對的，長的 n-gram 不容易再出現，但一旦出現則比較容易用來預測其類別。因此，特徵詞的選擇方式視應用而定。如果一定要固定一種特徵選擇策略來應付各種不同的應用，那麼取變動長度的 n-gram（即 n 由短到長都有），或同時取短詞與長詞來作為特徵詞，是比較可靠的作法。

二、特徵詞彙刪減（feature reduction）

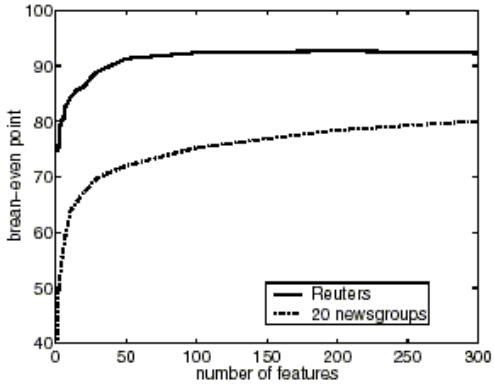
決定特徵選擇策略後，常會因全部文件的特徵詞太多，導致系統耗費過多的記憶空間或計算時間，而效率不佳。另外，不同的特徵詞對分類成效的貢獻應該都不同，擇優汰劣，可以降低特徵詞數，提升效率，而不致大幅降低分類效果，甚至還有可能因為過濾掉干擾分類的雜訊詞（noisy term），而提升分類效果。特徵詞刪減的方法很多，簡單的方法像依據停用詞（the、of、因為、所以等詞）、文件篇數（document frequency, DF）、詞頻與反相篇數（TF x IDF）等來過濾詞彙。像篇數過少的詞，只出現在一、兩篇文件，可能以後也不會再出現，但這類詞卻最多，經常佔 60%-70%以上。刪除這些詞，可以大幅降低特徵詞數。另外，把每篇文件的特徵詞以 TF x IDF 的值排序，取最高的前 N 個詞，可以刪除一些比較沒用的高頻詞。

如果把每個詞彙在每個類別的分佈資訊考慮進去的話，可以用更好的方法來刪減詞彙。這些方法有：mutual information [32], information gain, chi-square test, correlation [33], odds ratio [34]等等。然而要刪減多少詞彙，才不會大幅影響成效。Ng 等人 [33] 曾以 Perceptron 的學習與分類方式，就 Reuters 的測試集試驗三種特徵詞刪減的成效，如表一所示。可以看見，刪減的方式雖有好壞，但保留的詞彙越多，效果越好。Bekkerman 等人 [30] 以 SVM 的分類器試驗 Reuters 及 20NG 的文件發現，以 mutual information 刪減特徵詞時，對 Reuters 的文件刪減到只剩下 50 個詞時，分類成效就達到最大值，亦即保留更多的詞，分類的效果也不會再更好，如圖二所示。但是對於 20NG 的文件，留下越多的詞，效果越好，但隨著詞彙數的增多，成效的進步便緩慢下來。Yang and Pedersen [32]的實驗顯示，對 Reuters 的文件特徵詞刪減甚至可以提升分類成效，但對 OHSUMED 的文件，就沒有這種現象。Joachims [35] 曾以 SVM 分類器，把 Reuters 的其中

一個類別（acq 類）拿來做分析，他將屬於該類的詞彙按 information gain 排序，然後取其中的部分詞彙做分類，並與隨機取用的詞彙做分類的成效比較，結果如圖三所示。以最好的前 200 個詞做分類，成效接近 90%，以排序在第 201-500 的詞彙做分類，成效超過 70%，以排序在第 501-1000 的詞做分類，成效超過 60%，即便是光用排序在第 4000 位以後的所有詞做分類，成效都還超過 40%，比隨機取詞的分類效果還要好（只有 21%）。顯示，即便是評估起來最差的詞，都還含有相當多的資訊，而具備些許分類的價值。

表一：特徵詞刪減的成效比較。此表格取自 Ng 等人的論文[33]。

Feature Selection	20 features	50 features	100 features	200 features
Correlation	0.784	0.792	0.799	0.802
Chi-square	0.742	0.771	0.790	0.794
Frequency	0.717	0.763	0.778	0.785



圖二：特徵詞數的分類成效圖。此圖取自 Bekkerman 等人的論文 [30]。

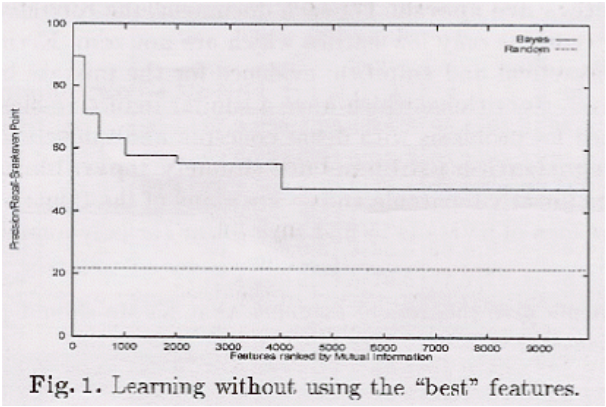


Fig. 1. Learning without using the "best" features.

圖三：不同特徵詞的分類成效圖。此圖取自 Joachims 的論文 [35]。

綜合這些實驗結果，似乎特徵詞刪減的效果，受分類器以及文獻特性影響。就相同的分類效果而言，越好的分類器，需要的特徵詞越少。而不同的文獻，需

要的特徵詞彙個數也不相同。刪減的詞數視不同的應用而定。

在做特徵詞刪減時，如果是運用到特徵詞在類別中的分佈資訊時，通常都針對每一個類別分別進行刪減的動作。這會造成每個類別都有自己獨特的特徵詞集，如果有 1000 個類別，就有 1000 個特徵詞集。當一篇待分類文件送進分類系統時，必須從文件中選取各個類別的特徵詞集進行個別分類，把符合的類別分出來。如果系統不做特徵詞刪減，系統只需一份特徵詞集，也許還是需要就個別類別分別進行分類。但比較起來，特徵詞刪減的動作，導致要為每個類別維護一份特徵詞集，當面對的分類問題有大量類別的情況（上百、上千甚至上萬個類別的問題），系統的負擔就很大，而且在分類效果不確定的情況下，有沒有必要做，是值得考慮的問題。

三、前置摘要處理

文件自動摘要的目的，是希望能自動去除文件冗餘的訊息，留下精華的部分，以節省使用者閱讀文件的力氣。此外，摘要後的文件，資料量較少，也可提升後續文件自動處理的效率。基於這樣的觀察，文件自動摘要的技術，可能有助於文件的自動分類。目前大部分的文件自動摘要技術，是從文件內文本身，自動選取最重要的數個片段，來代表該文件，作為該文件的摘要。研究顯示，這樣的摘要動作，取原文 30%的內容時，常可涵蓋原文 80%左右的訊息 [36]。

李祥賓 [37] 曾以 Reuters 的文件做實驗，取每篇文件最重要的三個句子做文件分類，並與全文分類做比較，結果摘要前的正確率為 65%，摘要後的正確率為 71.9%，顯示其自動摘要對文件分類有所幫助。然而筆者與指導的學生 [38] 做出來的實驗結果則有所不同。同樣以 Reuters 的文件做實驗，以向量分類法做分類時，摘要前（即全文）的成效為 68.9%，摘要後（取原文 30%的內容）的成效為 71.5%，分類成效有提升。但以 KNN 法做分類時，摘要前的成效為 75.3%，摘要後（取相同的 30%內容）的成效為 72.8%，分類成效降低。這顯示較佳的分類方法（成效較高者），在詞彙的選擇與權重的決定上，效果較好，自動摘要的過程中，丟掉了將近 70%的文字，反而會讓其損失一些分類的資訊。較差的分類方法，可能由於選詞能力不佳，靠著自動摘要助其丟掉一些冗詞，才得以提升成效。

因此，自動摘要雖然可以提升系統的效率（因為摘要後要處理的資料量變少），但對較好的分類方法而言，並不見得可以提升短文件的分類效果，尤其像 Reuters 這類文件，平均一篇文件只有 7 個句子。但是對較長的文件有沒有幫助，則還不清楚，文獻上幾乎沒發現這樣的研究。部分的原因可能是缺乏長的分類測試文件，以致於還沒有資源可以做驗證。

四、分類器選擇

過去數年的研究，提出過多種分類器，像 SVM（Support Vector Machine）、KNN（K-Nearest Neighbors）、LLSF（Linear Least Square Fit）、multilayered perceptrons、naï ve Bayse、Rocchio 方法等，不同的研究在不同的環境下報告出來的實驗數據，多少有些許的差距，對於哪一種分類器效果較好，就有了不太一致的結論。為此， Yang 與 Liu [29] 特別以統計考驗的方法，以 Reuters 文件的 90 個類別，比較了五種分類器的成效。其實驗結果發現：

{ SVM, KNN} > LLSF > multilayered perceptrons >> multinomial naive Bayes

也就是 SVM 與 KNN 一樣好，雖然 SVM 數據比 KNN 高一些，但是沒有統計上的顯著差異。而這兩種分類器都比 LLSF 好， LLSF 又比 multilayered perceptrons 好，naï ve Bayes 是這五種裡的分類效果較差的。Yang 只根據 Reuters 的文件得出這樣的結果，如果換成別種文件、別種應用，這個順序會不會一樣？Joachims [35] 取 Reuters 的前十大類做實驗，以 micro-average 計算成效，結果：

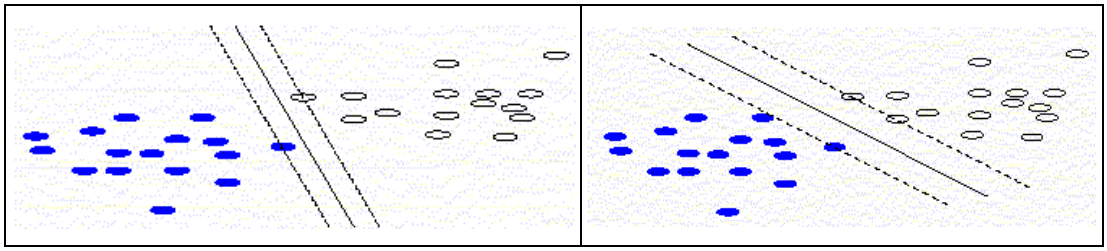
SVM (0.864)>KNN (0.823)>{Rocchio(0.799),C4.5(0.794)}>naï ve Bayes (0.72)

其中括號裡的數字為 micro-average。同樣的報告中，Joachims 對 OHSUMED 文件的 23 個類別做實驗，結果：

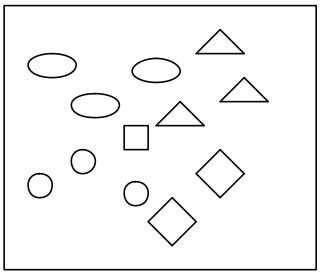
SVM(0.66) > KNN (0.591) > naï ve Bayes (0.57) > Rocchio (0.566) > C4.5(0.50)

而且在所有的 23 個類別中 SVM 都比 KNN 表現得好，顯示出有統計上的顯著性差異。

SVM 是 1990 代發展出來的方法，在機器學習的理論裡稱得上是突破性的進展。如圖四所示，假設要將文件分為正、負兩類，圖左為一般分類器學出的分割狀況，如圖中直線，虛線顯示正範例與負範例的邊界；圖右則為 SVM 學出的分割狀況，正範例與負範例之間有最大、最寬的邊界，因此對將來新近的文件，應該可以分得更好。SVM 在很多應用上幾乎都有最好的表現，但是它的一個明顯缺點是訓練時間很長。Platt [39] 曾提出一些作法來加快 SVM 的學習時間。另外它是一個二元分類器（binary classifier），亦即每個 SVM 只能將輸入分成兩個類別（Yes 或 No），應用在多個類別的分類問題時（multi-class problems），就要稍加變化。最簡單的應變方式就是每個類別就訓練一個 SVM 分類器，待分類文件進來時就由每個 SVM 進行個別分類，符合者就把文件標示為該類別。



圖四：圖左：一般分類器學出的超平面，如圖中直線，虛線顯示正範例與負範例的邊界。圖右：SVM 學出的超平面，正範例與負範例之間有最大、最寬的邊界。（此圖取自 Yang 1999）



圖五：KNN **分類方法示意圖。正方形代表待分類文件，其他形狀代表已分類文件。**

KNN **相對上是一個概念較簡單的方法：輸入文件與所有的訓練文件比對，找出相似度最高的前** K **篇文件，再從這** K **篇文件的類別，決定輸入文件的類別。決定類別的方式可以用簡單的票選（**voting**）或加權的票選（**weighted voting**）。如圖五所示，正方形代表「待分類文件」，其他形狀代表「已分類文件」，不同的「形狀」代表不同「類別」，取** K=4 **時，最靠近正方形的文件有橢圓形一篇、三角形一篇，圓形兩篇。如用票選，則可能會取圓形類別，如用加權的票選，且假設數據如下：**

Similarity_Distance(**正方形，橢圓形**)= 0.5

Similarity_Distance(**正方形，三角形**)=0.9

Similarity_Distance(**正方形，圓形**)=0.4+0.4=0.8

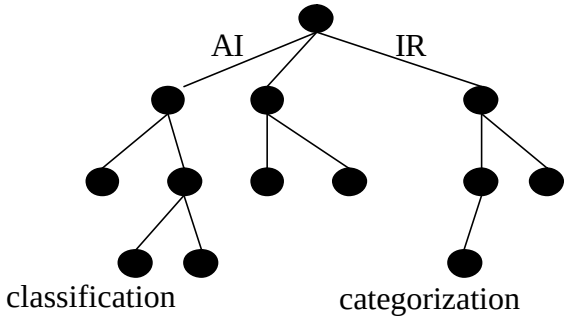
則可能會取「三角形」為其「類別」。從這個說明與例子可知，影響 KNN **分類成效最大的因素是相似性的準確度。一旦選錯了相似性計算公式，使得文件的相似度不能被正確的反應出來時，**KNN **的成效就大打折扣了。**

KNN **的計算量，主要是花在輸入文件與所有訓練文件的相似度比對上。如果可以為訓練文件事先建立索引，則這樣的比對，類似拿一篇文件的全文送進檢索系統進行查詢。只要檢索系統設計得當，**KNN **的分類速度其實夠快。**Lam **與** Ho **觀察到** KNN **還是會受到雜訊文件（即分錯類別的訓練文件）的干擾，因此提出一個改良** KNN **的方法，實驗結果顯示其分類成效及分類速度都有提升，但訓練時的計算量卻變大** [40]。

除了 SVM **與** KNN **外，還有沒有更好的分類器？即便是最好的分類器也並不一定每個類別都分得最好，如果這種現象成立，而且如果不同的分類器其分類錯誤的情形可以預測的話，那麼同時運用多種分類器，可能可以進一步提升分類的效果。例如：假設可以知道具備某種特性的文件，由某種分類器來分類，可以得到最好的分類效果時，則不同特性的文件以後就選擇由適合該特性的分類器進行分類，以克服單一分類器無法全部都分得最好的情形。**[41][31][42]**等論文都曾運用過多重分類器（**multiple classifiers**），而得到更好的效果。**

五、分類架構

「分類架構」（classification scheme）或稱「分類表」可大略分成單層式及階層式（hierarchical）兩種。單層的分類架構大都運用在類別數較少的情況下，例如新聞常分成：政治、社會、文藝、體育、娛樂等版面，方便讀者依類閱覽。階層的分類架構，則運用在類別數較多的情況，類別先依大類區分，大類下再逐層細分。因此越下層的類別、越互相鄰近，在主題上通常是越相近。但是例外的情形，往往也不在少數，特別是分類架構設計不良時尤然。例如將文件先按地區分類，再依主題分類，結果容易發生像報導美國中學校園槍擊事件，跟德國中學校園槍擊事件，文件內容相當但卻分在不同大類的情形。因此，在圖書館學上，還會根據類別的屬性先做分類，再對文件做分類。例如先設計主題分類、地區分類、時間分類，然後以主題分類當主分類，地區與時間當複分類。以剛剛的兩篇文件為例，就可分別標示成「校園槍擊—美國」與「校園槍擊—德國」，其主題一樣，但可用地區複分類加以區別。這樣的概念在數位文件中也常看到，只是比較沒有用到複分類的概念，而是以多個分類表的形式出現。例如 Reuters 的文件，除了常被用來做分類實驗的 Topic 分類表外，其實還包括 Place、People、Organization、Exchange 這四個分類表。每篇文件，都可以分別在這五個分類表中有數個類別。然而即便是做了這樣的區分，還是避免不了我們剛剛提過的情形：越鄰近的類別，通常越相近，但是也常有例外。人類知識的發展，常會讓原本看似沒有相關的概念，最後都有所關聯。圖六顯示了一個例子：「AI」（人工智慧）下可能有「機器學習」類（machine learning），其下有「資訊分類」（classification）這個應用類。而另一個領域「IR」（資訊檢索），其下可能有「文件組織」類（document organization），在其下有「文件分類」（categorization）這個類別。在眾多學術領域中，AI 跟 IR 原本差距剖大，圖中顯示 AI 下的 classification 跟 IR 下的 categorization 也差很遠（如果以節點跟節點之間的距離計算，則相差了 6 個節點的距離），但主題卻極端相近。



圖六：階層分類範例。

這裡特別強調這一點的目的是在指出：在階層架構的分類表中，文件分類的

成效不容易評估。假設有兩個分類器，對所有文件的分類結果都一樣，但對一篇新到的文件分別分出「B->E」以及「B->E->G->M」這兩類別，前者在第二層，後者在第四層，但假若該篇文件的正確類別是在第三層的「B->E->G」類，那麼我們該如何區別哪一個分類器比較好？同樣的，如果一個分類器分出「B->E->H」，另一個分出「A->D->I」，看起來「B->E->H」跟標準答案「B->E->G」比較相近，應該是比較好的分類結果，但是前一段的例子中可知，有時候不見得「A->D->I」與「B->E->G」相差比較遠，在主題上就有比較大的差異。

單層的分類架構，比較容易評估；多階層的分類架構，會有上述的問題。因此分類架構影響分類成效的地方在於：目前沒有更精緻的評估方式來區別不同分類器的表現。另外，目前看到的分類研究，對多階層分類的問題，大多沒有利用到類別之間的階層資訊，而是把每個類別視為一個個獨立的單一分類問題來處理。Ruiz 等人發現：同樣的分類器，以階層結構組織起來，比起只用單層結構做分類，有更好的效果 [43]。但是有些階層式的分類結構，對階層式分類器是不恰當的。例如，Ng 等人的論文中 [33]，曾提到一個實際在運用的分類架構，其第一層按國家地區分類，第二層再按政治、經濟、社會等主題分類。那麼關於跨國家地區的經濟問題，階層式分類器在第一層必須先選定國家類別時，便變成沒有多大意義。

在階層式分類架構中，常常會有文件分在中間階層類別的情形。也許有人會問：為什麼有了子類別還要把文件分到其父類別？難道所有子類別的聯集，仍然沒有辦法涵蓋父類別的主題嗎？這是分類表設計與運用的問題。會造成這種情況，可能是當初設計這些子類別時，還沒有想到還會有哪些子類別需要列進去，但又不願任意新增一個子類別出來，因此就把文件「暫時」放在父類別上。又或許是該篇文件內容真的很籠統，真的沒有適當的子類別，只好放在父類別上。通常，規定文件都必需分在最底層的類別，而不要分在中間層的類別，對後續的自動分類，會比較好處理。

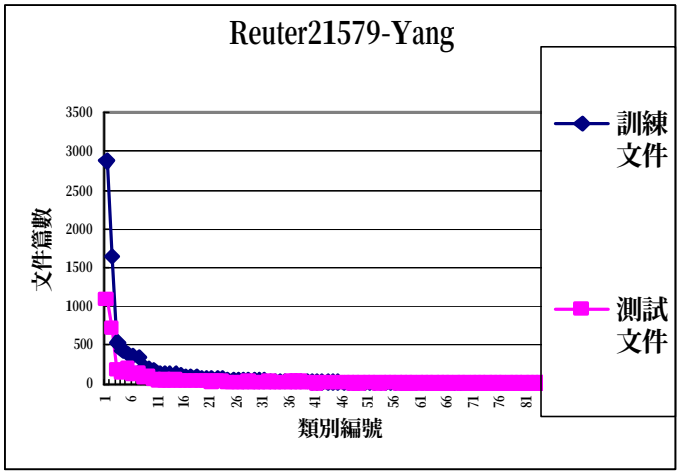
六、文件標示原則

每篇文件在標示類別時，可以只標示一個，或者標示多個，前者稱為「單一標示」（single labeling），後者稱為「多重標示」（multiple labeling）。單一標示適用於實體文件的管理，以方便實體文件按類存放、展現或歸檔。但其缺點是分類者的標示，不見得符合使用者預期的標示，這會讓使用者不易找到自己所需的資料。多重標示就是只要文件符合什麼主題就給出什麼類別，讓使用者可以從不同的角度找到相同的資料。數位文件比較沒有存放的問題，反而有取用查找的問題，因此適合多重標示。因此在圖書館的實體文件中，除了應用單一分類的「分類表」外，通常還會應用可多重分類的「主題表」（subject headings），以協助使用者找到資料。

如前所述，大部分的自動分類作法，都把每個類別視為一個獨立的分類問題，再運用二元分類器（binary classifier），就每個類別單獨分類。這當然符合多重標示的應用。但是實用上，通常還是會限制每篇文件的最大類別數，以及最少類別數。一旦分類器設計不佳，或訓練資料不良，導致分類的結果，不是太多類別（每個分類器都說 Yes），或是沒有類別時（每個分類器都說 No），便需要額外的作法來保證其符合限制條件。最簡單的作法，是按分類器計算出來的數值（數值超過門檻者輸出 Yes，低於門檻者輸出 No）排序，都沒有分類器輸出 Yes 時，取數值最大的那個類別，以保證至少分出一類；Yes 的個數超過限制時，數值較低的類別就捨棄，以保證不超過最大的類別數。這樣做並沒有把分類器的門檻資訊用得最好，是不是還有其他更好或最佳化（optimal）的作法，是值得考慮的方向。

七、類別選擇

不同的研究，實驗的細節不太相同，有的研究只取 Reuters 的前 10 大類做實驗，有的取 90 類，導致彼此間的實驗數據無法比較。為何有些學者只取部分類別做實驗？圖七是 Reuters 的 90 個類別所含文件篇數的分佈圖。表二、表三是最大的 10 類與最小的 20 類文件數量的分佈表。這些數據顯示，文件數多的類別只佔極少數，文件數少的類別佔大多數。這種現象在文字相關的統計中常常出現，不是 Reuters 的文件獨有。把訓練與測試文件數只有 1 篇的小類，拿來實驗，報告數據，會面臨一個問題，就是這樣的數據可不可靠。分對了一篇，就是 100%，分錯了，就是 0%。當測試資料本身有問題時，整個分類成效的數據就大受影響。因此，如果只是要比較不同分類器的成效，取大類做實驗，報告出來的數據比較可靠。當然，在解決實際的分類問題時，每個類別都要做分類，同樣會碰到文件分佈極不平均的挑戰。但真實世界中，經常可以自由加大訓練量，使得單篇訓練文件的情況不致發生。



圖七：Reuters 的 90 個類別的文件篇數分佈圖。

表二：Reuters 文件前十大類別的文件數量分佈表

	類別名稱		訓練文件			測試文件	
		篇數	百分比	累計百分比	篇數	百分比	累計百分比
1	earn	2877	0.30	0.30	1087	0.29	0.29
2	acq	1650	0.17	0.47	719	0.19	0.48
3	money-fx	538	0.06	0.53	179	0.05	0.53
4	grain	433	0.05	0.57	149	0.04	0.57
5	crude	389	0.04	0.61	189	0.05	0.62
6	trade	369	0.04	0.65	118	0.03	0.65
7	interest	347	0.04	0.69	131	0.03	0.69
8	wheat	212	0.02	0.71	71	0.02	0.71
9	ship	197	0.02	0.73	89	0.02	0.73
10	corn	182	0.02	0.75	56	0.01	0.74

表三：Reuters 文件後二十類別的文件數量分佈表

編號	類別名稱	訓練	測試	編號	類別名稱	訓練	測試
71	coconut	4	2	81	palladium	2	1
72	coconut-oil	4	3	82	palmkernel	2	1
73	jet	4	1	83	rand	2	1
74	cpu	3	1	84	castor-oil	1	1
75	potato	3	3	85	cotton-oil	1	2
76	propane	3	3	86	groundnut-oil	1	1
77	copra-cake	2	1	87	lin-oil	1	1
78	dfl	2	1	88	nkr	1	2
79	naphtha	2	4	89	rye	1	1
80	nzdlr	2	2	90	sun-meal	1	1

八、分類不一致

Kreines [44]曾報導他們參加 2001 年的 TREC filtering 評比，發現 Reuters 新的分類文件有分類不一致的現象：索引詞幾乎一樣的兩篇文件，卻被分在不同的類別。卜小蝶 [45]在做詞彙的分類研究時也發現，以二名不同分析人員，針對 1,000 個檢索詞彙作分類測試，在大類上有 10%，小類有 20%的不一致性。Kao [46]也報告類似的問題：他們聘請了一位有 30 年經驗的分類專家專心從事分類的工作，類別總共 500 多類，每篇文件平均分到 5 類，有的甚至 30 類。但相隔才三個月，卻發現這位專家對相同的文件做很不同的分類。Kao 提到，這種分類的不一致性對分類器的成效有兩種影響：一是分類器無法在資料完全正確的環境下獲得訓練；二是既然連人工分類的答案都不完美，分類器的成效當然會受影響。筆

者做分類實驗時，就發現這種現象：比較好的分類器，得出較差的成效分數。經檢視測試資料發現，像報導郭泰源的文件，在訓練資料中有時候被分為「職業棒球」、有時候只分在「體育」類，但剛好測試資料中，是分在「體育」類。比較好的分類器雖然將測試文件的類別分在「職業棒球」，但較差的分類器分不到這麼細的類別而只分在「體育」類，但是因為答案是「體育」，因此較好的分類器反而得出較差的數據。

從上面四個案例中可以瞭解：一、分類不一致是普遍且難以避免的現象；二、即便分類資料不完全正確，但較好的分類器還是能夠從中做有效的學習，而得出漂亮的分類；三、自動化的分類，可能比人工分類的一致性還高；四、在不清楚分類一致性的程度時，不同分類器做出來的成效數據，難以比較，數據較高者，不一定是較好的分類器。

這些觀察也可以支持前一小節中提到的概念：若要比較成效，只要選擇大類比較即可，單篇訓練文件的類別，分類一致性的問題較大，從中得出的數據比較不可靠。這是因為在只有訓練文件的情況下，類別的主題意義是以這些範例文件定義出來的，一旦範例文件不足，類別的主題意義便不足，對新進的文件，當然難以正確的做出分類。

這些觀察也支持自動分類系統的運用。分類系統不是要用來取代人工分類，而是降低人力負擔，讓寶貴的人力資源運用在機器分類結果的校正上，既可降低分類不一致的問題，也可同時提升整體的效率與正確度。

九、訓練資料量

對自動分類器而言，要多少訓練文件才能夠獲得足夠的成效？Dumais 等人 [47] 曾針對這個問題，以 SVM 分類器對 Reuters 的 10 大類做分類試驗，用全部訓練資料時，成效達 92%，只用 10%的正範例時成效 89.6%，5%時成效 86.2%，1%時成效 72.6%。當正範例文件少於 5 篇時，對某些類別而言其成效就不穩定。他們認為至少 20 篇正範例文件才能提供穩定的成效。

另一方面，楊允言 [48] 的實驗結果如表四所示，訓練資料只用 12 月份時，測試文件的成效為 52.64%，訓練資料用 7 到 12 月時，進步到 67.14%，即訓練文件越多，效果越好，而且進步的幅度，未見緩慢下來。相對的，表四中訓練文件越多，對訓練文件再做分類的效果反而降低，這可能是該分類器的特性，但也可能是文件越多，不一致性的情況越大所致。

如前所述，類別的定義，是由訓練文件給定，因此訓練文件越多越好。但越多的訓練文件，造成系統的負擔越重。這可由文件取樣（sampling）等技巧來有

效改善 [49]。

表四：訓練文件份量與成效比較表。此表資料取自楊允言論文。

所用訓練資料月份	關鍵詞數量	訓練資料召回率	測試資料召回率
7~12 月	5,579	94.86%	67.14%
8~12 月	5,085	94.67%	61.98%
9~12 月	4,344	96.09%	61.69%
10~12 月	3,379	97.09%	59.77%
11~12 月	2,379	98.16%	53.82%
12 月	1,297	99.62%	52.64%

十、成效評估方式

傳統的分類成效評估方式，類似檢索成效評估，對每一個類別，所有的文件將被劃分於如表五所示的四種狀況中，即屬於該類的文件，被系統正確分為該類的有 a 篇、沒被系統分為該類的有 b 篇；而不屬於該類的文件，被系統分為該類的有 c 篇、沒被系統分為該類的有 d 篇。對每個類別都做這樣的統計後，即可計算「正確率」(A, accuracy)、「精確率」(P, precision)、「召回率」(R, recall)，如下：

accuracy=(a+d)/(a+b+c+d), precision=a/(a+c), recall=a/(a+b)

其中正確率受 d 值影響很大，當 d 遠大於其他值時，不管有沒有正確分類，其「正確率」都接近 1。由於有這樣的不合理存在，這個評估方式盡可能不要用。而精確率與召回率不能單獨使用，理由是系統很容易做出高精確、低召回，或低精確、高召回的結果。為同時兼顧這兩個數據，經常再定義 F=2PR/(P+R)，來比較不同系統的成效。

如果同時有好幾個類別要一起考量，則有 micro-average 與 macro-average 兩種平均方法，定義如下：

$$micro\,\mathrm{Pr}ecision=\frac{\sum_i a_i}{\sum_i a_i+\sum_i c_i},\quad micro\,\mathrm{Re}call=\frac{\sum_i a_i}{\sum_i a_i+\sum_i b_i}$$

$$macro\,\mathrm{Pr}ecision=\frac{1}{m}\sum_{i=1}^{m}\frac{a_i}{a_i+c_i},\quad macro\,\mathrm{Re}call=\frac{1}{m}\sum_{i=1}^{m}\frac{a_i}{a_i+b_i}$$

其中註標 i 是指第 i 個類別，而 m 是類別的總數。同理，F 值也可依原公式，帶入相關的 P 與 R，分別得到 micro-F、或 macro-F。Micro-average 由於是全部文件一起累加統計，不分類別，因此容易受到大類別（佔大多數文件）表現好壞的影響。相對的，macro-average 考慮每個類別的成效後再做平均，因此容易受到

大量的小類別影響。在展示成效時，經常這兩種平均數據都報告出來，以便分析比較。從數值上來看，通常 micro-average 都遠高於 macro-average。

表五：文件數量分佈表

	系統分為該類	系統不分為該類
屬於該類別	a	b
不屬於該類別	c	d

然而，研究人員漸漸發現這種評估方式對實際的應用而言，並不恰當。Hersh [50] 曾提出這樣的看法：

> *This field needs to move beyond simulated data sets and recall/precision type metrics. …*
>
> *We have all seen the decent results that good systems can generate, but now we need to figure out to make these tools truly useful to researchers and analysts who are going to use them to solve real-world problems. …*
>
> *That is, we need to show that we can help a researcher extract knowledge from mining the literature that results in scientific discovery previously not possible (or excessively time-consuming).*

亦即，這領域需要的不僅是模擬的資料以及召回/精確型態的評估方式，因為在這樣的環境下，我們已見識到良好的系統所能產生的優秀結果，但對於實際的應用，我們需進一步展示自動分類真正的好處，看它是否能讓研究人員發現過去所未能發掘的新知，或加快獲得過去要耗費極大力氣才能獲得的結果。

筆者就曾碰過這樣的案例：在為使用者建置好分類系統後，人工使用評估結果為：測試 114 篇文件，51 篇有問題，正確率：55%（=(114-51)/114）。效果比預期的差！筆者很好奇，就察看了系統報告的數據，發現 F 值為 0.831，對上千個類別的分類問題而言，這是相當的高的數據。與使用者溝通後，才瞭解他們的標準是：如果該文件應選 5 類，分類器選了 5 類，但其中一類差異極大，整篇文件的分類就算錯誤。對使用者而言，沒有所謂答對 4/5，而是整篇文件分類錯誤。這是因為一旦有任何錯誤，使用者就必須動用滑鼠，在操作介面的分類樹上展開、關閉、搜尋、點選，以取消不適當的類別，或增加新的類別。使用者每天要確認 300 篇文件的類別，而這只是其每日 1/3 的工作量。比起精確率、召回率，以節省使用者工作負擔為分類成效的評估依據，在這個例子中是非常恰當的。

十一、 參數調整

在 F=2PR/（P＋R）的公式中，為得到最佳的數據，必須把分類器的表現調整到 P=R 的情況（精確率等於召回率），以求得最高的 F 值。然而實際的應用中，必須瞭解使用者的需求型態，有些是精確導向（precision-oriented），有些是召回

導向（recall-oriented）。像文件歸檔、過濾的應用，需要精確導向的分類機制，盡可能不要出錯；像文件瀏覽、檢索的應用，則需要召回導向，盡可能不要遺漏。例如，在前一段的例子中，我們把系統調整成精確導向，再增加訓練資料量，就達到使用者期望的效果了。

Yang 在 SIGIR 1999 年的論文中提到，應該要對每一個類別都訓練出其分類的門檻值出來，才是最佳的分類作法 [29]。但筆者認為，在實際的分類問題中，這樣的做法有其風險：若訓練資料與測試資料的性質差異過大時，不相干的類別可能超過門檻值，而被標示為文件的類別。結果在 SIGIR 2001 中，Yang 就發表了這樣做並非對每種分類問題都有最好的結果 [51]，推翻了自己先前的說法。其結論是：不同的分類問題（分類架構與分類文件），需要用到不同的門檻策略，才能得到最好的結果。

十二、 分類器的最大成效

假若所有的參數都選擇、調整到最好的情況下，自動分類到底能做到多好？最好的技術，可以做到完全正確的分類嗎？

在「分類不一致」那一小節裡，我們知道任兩個人都無法做到完全一致的分類，更何況機器如何能跟任何一個人做到完全一致的分類。因此自動分類器的最大成效，受限於類別之間的區別程度。不同的類別，越容易區別出來，人工分類越不會不一致，自動分類器做出來的成效數字就越高。反之，類別越不容易區別，人工分類都容易混淆、容易發生前後分類不一致時，自動分類器做出來的成效數字就越低。因此，不能單以絕對的成效數據來判斷自動分類器的表現，例如，F 值只達 0.48，就說自動分類效果不到五成。也許對於這個分類問題，不同專家的分類一致性僅達 55%，那麼 F 值為 0.48 的分類器，其實已達人工分類的 0.48/0.55=87.28% 效果了。所以，分類器與人工分類的不一致性，若達到不同人工分類之間的不一致性，即可視為最佳的分類狀況。

影響分類器達到最大成效的另一個原因，是訓練資料的代表性。以訓練資料做為分類器唯一知識來源的情況下，每個類別的意義、範圍、以及其代表的主題，都是由訓練資料列舉表示，因此這是一種將類別「以範例做定義」（defined by examples）的模式。另一種定義類別的模式是「以敘述做定義」（defined by descriptions），即類別的意義、範圍以及主題是以清楚、完整的文字加以敘述、說明。「以範例做定義」時，機器較容易接受，因為分類器只要依範例學習、分類即可，不必做太多文字理解的運算。「以敘述做定義」時，要機器理解文字敘述的內容，是較困難的工作。由於以範例定義類別時，若範例不夠、涵蓋的範圍不完整時，自然類別便定義的不完整，分類器對新進文件做分類時，便不容易分得好。這種情況特別容易發生在列舉式的類別，如「民間社團」類。如果訓練資

料中已經有一大堆「基金會」、「獅子會」，但若待分類文件是關於「童子軍團」的報導，分類器依然可能不知道要將其分在「民間社團」類。

肆、 自動分類系統的運用

瞭解自動分類系統的特性及影響其分類成效的各種原因後，可以幫助我們把自動分類系統運用得更好，充分發揮其優點，但仍有幾點事項需要注意：

一、分類架構的設計

一個設計不良的分類表（或分類架構），不僅人工分類困難，機器分類困難，將來要運用、修改也很麻煩。常見的不良設計有：

（一）類別定義不清、範圍重疊；

（二）事件與主題混淆；

（三）以關鍵詞當作類別。

設計分類表時，應當瞭解分類表的目的、以及將來的應用方向，到底是要提供館藏瀏覽、還是提供主題檢索、按類歸檔、還是通通都要。不管哪一種應用，基本的原則是盡可能讓每個類別單獨代表所有文件中的某個主題，且讓所有的類別涵蓋所有的主題。亦即同屬於同一父類別下的任意兩子類別，其主題的交集是空集合，且所有子類別的主題聯集盡可能為其父類別的主題。當文件內容橫跨多項主題時，則以多重標式的方式，來涵蓋文件提到的數項主題。另外，當發現類別在階層架構中容易重複使用時，則應區別類別的屬性，依其屬性歸納制訂出不同的分類表，如地區、時間複分表等，以便對所有文件做不同性質的切割。基本上，在制訂分類表時，應當清楚定義每個類別，以便分類者有所遵循。行政院主計處擬定的「行業分類表」，分大、中、小、細類四個階層，每個類別都有：編號、名稱、定義，以及細類內容或範例列舉，值得參考 [52]。

分類表的設計，是希望能夠持久有效，每個類別應盡可能包含有效的文件數量。這需要區分「主題」與「事件」的概念。相對於主題，事件是在時、空上具有侷限性的 [53]。例如「天災」是屬於「主題」，但「331 地震」是屬於事件。如果將「331 地震」當作類別，附屬於「天災」之下，那麼每次遇到一個地震案例就可能要新增類別，不是很好的設計。

上述的「事件」，其文件經常可用「關鍵詞」查詢得到。因此，若以「關鍵詞」即可蒐集到想要的文件，那麼這樣的「關鍵詞」並不需要單獨設類。例如把某一人名、機構名單獨設類，放在「風雲人物」、「著名企業」的類別下，在現今全文檢索技術發達的情況下，似無必要，甚至風雲人物、著名企業這些類別，也

不甚必要。因為，即便沒有這些類別，使用者仍然能夠檢索到想要的人名、機構文件。而如果連這些類別也要，那麼我們如何能夠預測使用者的所有需求，事先建構好所有的類別，例如：使用者可能在看過「風雲人物」的類別後，還要再區分「男性風雲人物」與「女性風雲人物」等等。當然，這裡不是說這些類別一定不要，而是要視使用者的應用情形，做縝密的考量。

二、與檢索系統整合

有時候我們需要將過去發生的一些重要事蹟整理成卷，例如：「921 大地震」、「911 恐怖攻擊事件」等。當初在發生這些事件時，可以就其往後的影響，決定要不要設類。若有設類，將來應用當然比較方便。但像「經發會」這樣的事件，在當初發生時吸引政、經、社會的極大關注，當然具有相當的份量，但其往後的影響則很難預估，是否單獨設類，見仁見智。根據上面的說明，這些都屬於「事件」，可以不設類，那麼以後要應用時如何補救？此種回溯性的需求，可以運用全文檢索系統，配合查詢技巧，動態的產生需要的類別或專卷，以降低任意新增類別的機會，並減輕需要事先分類的負擔。

另外，分類的目的之一在提供主題檢索，與檢索系統結合，可以運用關鍵詞，再加上類別主題的範圍限制，讓檢索文件更精準省力。不僅如此，由於分類的工作乃將非結構化的文件，做某種程度的結構化轉換，一些更高階的檢索系統，便可據以提供趨勢分析、綜合摘要、類別對比等文字資訊探勘 [54] 工具或資料庫管理系統才能提供的功能。

三、類別的修正、改分、細分

分類表用過一段時間後，不免會碰到時、空環境的變化，而需要修改。例如：香港不再是殖民地，李登輝也不再屬於國民黨等。常見的修改有：類別新增、刪除、更名、改分、細分等。「改分」是指將某個子類別從原來某個父類別下，改變到另一個父類別下。「細分」是指對某個類別的主題再予與區分，個別再獨立成適當的多個類別。這些變動，基本上只要改變訓練資料記載的分類資訊，分類系統即可隨之改變過來。要特別注意的是，分類好的歸檔文件，要不要修改其分類資訊？這是管理文件的政策問題。站在使用者的立場，舊的類別應當給予保留，使得習慣過去分類架構的使用者仍然可以繼續沿用。國際著名的學會 ACM，其論文的分類策略，便是讓使用者可以利用舊的類別，找到以往的文件。這些淘汰的舊類別，仍然保留，只是沒有用在新文件的分類上面 [55]。相對的，歸檔文件也要記載新的分類資訊，使得回溯性的文件，也能夠如同新文件一樣，被同一主題的新類別對應到。

伍、 結語

本文整理了 12 項跟分類成效有關的因素進行探討。有些因素在不同的學者間有不同的看法或結論。例如前置摘要處理對分類成效的幫助，我們的結論便跟別人不同。又如選擇哪些類別作為評估比較的對象，常是學者間爭議辯論的焦點。本文舉出分類不一致的普遍現象，來說明類別選擇的原則，並從筆者的經驗中提醒讀者小心解讀不同數據的比較結果，亦即：一致性不高的資料，其報告出來的數據比較不可靠。另外，成效評估方式不應只依賴傳統的精確率與召回率，應當去瞭解使用者真正的需求與目的，才能設計、調整出最佳的分類系統。最後，分類系統不需要百分之百正確才能應用，機器分類似乎比人工分類有較高的一致性，透過人工校正機器的分類結果，既可節省人力成本，又能維持正確度。筆者曾協助國內一家著名機構建立分類系統，使其從原本兩組分類人員變成一組，減少了超過一半的人力，而每天處理的文件數量則增加 40%。另一個運用相同分類系統的機構則在不增加任何人員的情況下，將全文文件分類歸檔、標示、分享，增進其知識管理的能力。顯見自動分類系統實具低成本、高效益的應用價值。

文件主題的自動分類，其成效主要是依據訓練文件中（文件=>類別）的對應得來。這種對應關係，若無法單獨從文件內容獲得，則自動分類的成效就很有限。例如：把最近一個星期收到的個人電子郵件自動標示為「待處理郵件」以及「不需處理郵件」。這牽涉到收件人的背景環境以及其與寄件者的互動關係等「外部知識」，難以單純從郵件內容獲得。同樣是關於知識研討會的兩份郵件通告，一份是產業界舉辦的，一份是學術界舉辦的，收件人到底會參加並回覆那個研討會，視其當時的興趣、動機，甚至工作時程表而定。若這種興趣或動機是（相對）持久而穩定的，以致可以反映在過去的郵件回覆上，再加上一份工作時程表的配合，那麼自動分類還有可能可以勝任。否則，與隨機自動標示無異，難以保障其成效。文件自動分類系統的應用，應當把握這個原則，有這樣的認知，才能發揮自動分類的成效。

主題分類的概念，傳統上都由圖書館學領域教授、傳承、研究、發展與應用。但近年來數位文件普及，再加上知識管理的需求逐日殷切，有研究者預測，五年以後各個公司、機構的資訊科技與資訊服務的相關人員，將像圖書館員一樣，普遍具備文件分類的概念。屆時，自動分類系統的應用將更普遍。但在那之前，文件分類的概念需要加強宣導、教育，相關的問題還需一一克服。未來還值得研究的課題，至少還包括：少量訓練文件的自動分類，分類架構自動建議或建構，分類錯誤的自動偵測與更正，外部知識結合分類，以及更多的相關應用與評估等等。

參考文獻：

[1] 胡述兆，吳祖善，*圖書館學導論*，漢美圖書有限公司，民 78 年。

[2] Hsinchun Chen, *Knowledge Management Systems: A Text Mining Perspective*, http://ai.bpa.arizona.edu/go/download/chenKMSi.pdf.

[3] 黃森原，*中文文件自動分類*，國立交通大學資訊科學學系，碩士論文，民 84 年。

[4] 陳俊凱，*利用類神經網路作文件自動分類之研究*，淡江大學資訊工程學系，碩士論文，民 84 年。

[5] 顧皓光，*網路文件自動分類*，國立台灣大學資訊管理學系，碩士論文，民 85 年。

[6] 陳智偉，*文件分類的方法與分析*，國立清華大學資訊工程學系，碩士論文，民 86 年。

[7] 林頌華，*新聞標題自動分類*，國立清華大學資訊工程學系，碩士論文，民 87 年。

[8] 劉自誠，*能有效變更之模組化階層式文件分類*，國立台灣科技大學電子工程系，碩士論文，民 88 年。

[9] Sue J. Ker and Jen Nan Chen, “A Text Categorization Based on Summarization Technique,” Proceedings of NLPIR Workshop of ACL 2000, pp. 79-83.

[10] David D. Lewis, “An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task,” Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and development in Information Retrieval, 1992, Pages 37 – 50.

[11] Elizabeth D. Liddy, Woojin Paik and Edmund S. Yu, “Text Categorization for Multiple Users based on Semantic Features from a Machine-Readable Dictionary,” ACM Transactions on Information Systems Vol. 12, No. 3 (Jul. 1994), Pages 278 – 295.

[12] Yiming Yang and Christopher G. Chute, “An Example-based Mapping Method for Text Categorization and Retrieval,” ACM Transactions on Information Systems Vol. 12, No. 3 (Jul. 1994), Pages 252 - 277

[13] Yiming Yang, “Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval,” Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, Pages 13 – 22.

[14] Yiming Yang, “Noise Reduction in a Statistical Approach to Text Categorization,” Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1995, Pages 256 – 263.

[15] Makoto Iwayama and Takenobu Tokunaga, “Cluster-based Text Categorization: a Comparison of Category Search Strategies,” Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1995, Pages 273 – 280.

[16] Marc Damashek, "Gauging Similarity with N-grams: Language-Independent Categorization of Text," Science, Vol. 267, 1995, pp.843-848.

[17] Leah S. Larkey, “Automatic Essay Grading Using Text Categorization Techniques,” Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1998, Pages 90 – 95.

[18] Raymond J. Mooney and Loriene Roy, “Content-Based Book Recommending using Learning for Text Categorization,” Proceedings of the fifth ACM Conference on ACM 2000 Digital Libraries, 2000, Pages 195 – 204.

[19] Fabrizio Sebastiani, Alessandro Sperduti and Nicola Valdambrini, “An Improved Boosting Algorithm and its Application to Text Categorization,” Proceedings of the 9th International Conference on Information and Knowledge Management CIKM 2000, Pages 78 – 85.

[20] Jhy-Jong Tsay and Jing-Doo Wang, “Improving Automatic Chinese Text Categorization by Error Correction,” Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages, 2000, Pages 1 – 8.

[21] C. K. P. Wong, R. W. P. Luk, K. F. Wong and K. L. Kwok; “Text Categorization using Hybrid (Mined) Terms,” Proceedings of the 5th International Workshop on Information Retrieval with Asian Languages, 2000, Pages 217- 218.

[22] Jason D. M. Rennie and Ryan Rifkin, "Improving Multiclass Text Classification with the Support Vector Machine," Massachusetts Institute of Technology. AI Memo AIM-2001-026. 2001. http://www.ai.mit.edu/~jrennie/papers/aimemo2001.ps.gz

[23] Thorsten Joachims, “A Statistical Learning Model of Text Classification for Support Vector Machines,” Proceedings of the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2001, pp. 128-136.

[24] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam, “OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research,” Proceedings of the 17th Annual International ACM SIGIR Conference on Research and development in Information Retrieval, 1994, Pages 192 – 201.

[25] Dmitri G. Roussinov and Hsinchun Chen, “A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation,“ Communication and Cognition in Artificial Intelligence

Journal (CC-AI), Vol. 15, N. 1-2, Pages 81-111, 1998

[26] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York, NY, 1973.

[27] William W. Cohen and Yoram Singer, "Context-Sensitive Learning Methods for Text Categorization," Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1996, Pages 307 – 315.

[28] Chidanand Apt, Fred Damerau and Sholom M. Weiss, "Towards Language Independent Automated Learning of Text Categorization Models," Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, Pages 23 – 30.

[29] Yiming Yang and Xin Liu, "A Re-Examination of Text Categorization Methods," Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1999, Pages 42 – 49.

[30] Ron Bekkerman, Ran El-Yaniv, Yoad Winter, Naftali Tishby, "On Feature Distributional Clustering for Text Categorization," Proceedings of the 24th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2001, pp.146-153.

[31] Khalid Al-Kofahi, Alex Tyrrell, Arun Vachher, Tim Travers, and Peter Jackson, "Combining Multiple Classifiers for Text Categorization," Proceedings of the Tenth International Conference on Information and Knowledge Management 2001, Atlanta, Georgia, USA, pp. 97-104.

[32] Yiming Yang and J. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," Proceedings of the International Conference on Machine Learning (ICML'97), 1997, pp. 412-420.

[33] Hwee Tou Ng, Wei Boon Goh and Kok Leong Low, "Feature Selection, Perception Learning, and a Usability Case Study for Text Categorization," Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1997, Pages 67 – 73.

[34] Dunja Mladenic, etc, "Feature selection for unbalanced class distribution and Naive Bayes," Proceedings of the International Conference on Machine Learning (ICML'98), 1998, http://www.cs.cmu.edu/~TextLearning/pww/yplanet.html.

[35] Thorsten Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proceedings of the European Conference on Machine Learning, 1998, Berlin, pp. 137-142.

[36] **黃聖傑，*多文件自動摘要方法研究*，國立台灣大學資訊工程學系，碩士論文，民 88 年。**

[37] **李祥賓，*新聞文件摘要之研究*，東吳大學資訊科學系碩士論文，2001 年。**

[38] 林政緯，*文件自動分類及其成效評估之研究*，輔仁大學圖書資訊學系碩士論文初稿，2001 年。

[39] Platt, J. “Fast Training of SVMs using Sequential Minimal Optimization,” in B. Scholkopf, C. Burges, and A. Smola (Eds.) Advances in Kernel Methods – Support Vector Learning, MIT Press, 1998.

[40] Wai Lam and Chao Yang Ho, “Using a Generalized Instance Set for Automatic Text Categorization,” Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1998, Pages 81 – 89.

[41] Wai Lam, Kwok-Yin Lai, “A Meta-Learning Approach for Text Categorization,” Proceedings of the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2001, pp.303-309.

[42] Leah S. Larkey and W. Bruce Croft, “Combining Classifiers in Text Categorization,” Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1996, Pages 289 – 297.

[43] Miguel E. Ruiz and Padmini Srinivasan; “Hierarchical Neural Networks for Text Categorization,” Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1999, Pages 281 – 282.

[44] M. Kreines, “Reuters Corpus problems,” trecfiltering@list.research.microsoft.com, Oct. 2, 2001.

[45] 卜小蝶，"網路使用者檢索詞彙主題分類探析"，台灣大學圖書資訊學系四十週年系慶研討會， 2001 年 11 月 16 日，頁 113。

[46] Anne Kao, “Re: Reuters Corpus problems,” trecfiltering@list.research.microsoft.com, Oct. 2, 2001.

[47] Susan Dumais, John Platt, David Heckerman and Mehran Sahami, “Inductive Learning Algorithms and Representations for Text Categorization,” Proceedings of the 1998 ACM 7th international Conference on Information and Knowledge Management, 1998, Pages 148 – 155.

[48] 楊允言，”中文文件自動分類之探討”，頁 241-256。

[49] David D. Lewis and William A. Gale “A Sequential Algorithm for Training Text Classifiers,” Proceedings of the 17th Annual International ACM SIGIR Conference on Research and development in Information Retrieval, 1994, Pages 3 – 12.

[50] William Hersh, “Re: medical classification tools?” ddlbeta@scils.rutgers.edu, Feb. 22, 2002.

[51] Yiming Yang, “A Study on Thresholding Strategies for Text Categorization,”

Proceedings of the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2001, pp. 137-145.

[52] 行 政 院 主 計 處 「 行 業 分 類 表 」 , http://www.dgbas.gov.tw/dgbas03/bs1/text/indu/indu.htm。

[53] Yiming Yang, Tom Ault, Thomas Pierce and Charles W. Lattimer, “Improving Text Categorization Methods for Event Tracking,” Proceedings of the 23rd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2000, Pages 65 – 72.

[54] Ronen Feldman, Ido Dagan, Haym Hirsh, "Mining Text Using Keyword Distributions," Journal of Intelligent Information Systems, Vol. 10, No. 3, May 1998, pp. 281-300.

[55] Introduction to the ACM Computing Classification System [1998 Version] http://www.acm.org/class/1998/ccs98-intro.html