大學圖書館
9 卷 1 期（民國 94 年 3 月） 頁 2-19

分類不一致對文件自動分類效果的影響

The Effect of Inconsistency in Training Data on Automatic Text Categorization

曾元顯 **Yuen-Hsien Tseng**
輔仁大學圖書資訊學系教授
Professor, Dept. of Library & Information Science,
Fu Jen Catholic University
Email: Tseng@lins.fju.edu.tw

【摘 要】

本文探討分類不一致對自動分類成效的影響。經由近似文件的自動偵測，以及兩種分類方法針對兩個測試文件集做的比較實驗，本文發現：訓練資料的分類不一致性，即便高達 34%，幾乎也不會影響分類器的成效。此項發現，其重要的意涵是，即便過去的研究使用了一致性不高的測試集做實驗，其結論仍舊是有效的。當然，分類不一致性高的資料，拿來訓練後，不管分類器好壞，其得到的分類成效都是比較低的。除了以上發現外，本文也介紹了一套中文分類測試集，免費提供各界研究使用。另外，作者也提出了一套偵測複本或相似文件的可靠方法，與過去的方法比較，此方法可以偵測過去方法所無法偵測到的相似文件。

【Abstract】

This article discusses the effect of inconsistency in training data on the performance of text classifiers. Our experiments show that the inconsistency, even reaching a level as high as 34%, hardly affects the effectiveness of the classifiers. Better classifiers perform better independent of duplicates and label inconsistency. The implication is that past experiments (especially on the Reuters-21578 collection) remain valid. In the experiment process, the author proposes a duplicate detection technique that is far more effective than previous ones. A new Chinese test collection for text categorization is also introduced for general free download.

關 鍵 詞：文件分類；一致性；分類測試集；主題分析；複本偵測
Keywords：Document classification; Consistency; Test collection for categorization; Subject analysis; Duplicate detection

臺灣大學學術期刊資料庫

壹、研究問題陳述

文件分類是根據文件內容的主題給定類別，對文件進行分門別類的加值處理，以便利後續文件利用的工作。（註1）（註2）這些類別常爲事先定義、管理者用以組織文獻的知識架構。由於牽涉到自然語言的詞彙辨識與文意的分析與理解，傳統上文件分類都由人工進行。隨著數位文件的快速增加，自動化的文件分類，有時變得不可或缺。例如專題新聞派送、垃圾郵件過濾等需要即時性分類的工作，都必須利用電腦進行自動化處理。

在利用電腦進行文件自動分類時，需要事先提供電腦訓練資料。所謂訓練資料，通常是指人工事先完成的分類文件及其對應的類別，亦即一組（文件，類別）的對應資料。然而受分類架構設計、類別定義範圍，以及分類者學識背景、主題判斷認知差異等影響，主題內容類似的文件，不見得都會被人工分類對應到相同的類別，因而造成分類不一致的情況。

例如：Kreines （註3）曾報導他們參加2001年的TREC filtering 評比，發現Reuters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（註4）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（註5）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。從上述實例看來，分類不一致，幾乎是文件分類一定會碰到的情況。

電腦在此種訓練資料不一致的情況下學習人工的分類經驗，以進行自動分類，其影響有兩個：一、分類器無法在資料完全一致的環境下獲得訓練，可能造成分類器混淆、難以收斂、過度訓練（overfitting 或參數調整過度）或訓練不足（underfitting 或參數調整不足）等問題，影響分類器的效果。二、人工分類的結果，通常也是分類器用來評估成效的標準答案，當標準答案都不一致時，可能會影響評分，造成結果難以預測或數據不可靠。

有鑑於此，本文特別想探討分類不一致對自動分類的影響。然而第一個問題是，如何知道分類不一致？以主題分類而言，所謂不一致是指相同主題的不同（文件編號不同）文件，被分到不同類別。因此，必須先偵測「相同主題」的文件出來，然後比對它們的類別，才可知道是否分類一致。在本文中，只要類別有不一樣的情形，就視爲不一致。例如文件X的

臺灣大學學術期刊資料庫

類別爲 A 與 B，而與其相同主題的文件Y的類別爲A與C，雖然有一個類別A是一致的，但類別B與C不同，則還是視文件 X 與 Y 的分類不一致。然而所謂「主題相同」是主觀的判斷，不適合電腦的自動處理。本文提出以文件的「相似性」來近似主題的相同性，亦即大膽假設相似的文件，即屬於相同的主題，如此就可自動偵測分類的一致性了。此點假設，對相似度極高的文件，例如複本、抄襲稿等，相當適用，但對於不相似但卻屬於相同主題的文件，就不適用了。不過其可視爲是一種比較嚴格的假設：有複本關係的文件，卻分在不同的類別，顯然是一種分類不一致的情況，而沒有爭議。相對的，不相似的文件而屬同一主題的情況，是比較容易引起爭議的。

有關相似文件的偵測將於下一節介紹。本研究面臨的第二個問題是，如何控制或獲得不同程度的不一致性，以便探討其影響。對同一份分類測試集而言，可以區分不同時期、不同分類者、不同類別的文件，甚至調整文件相似度門檻來獲得可能的不一致性程度。但若該分類測試集是在嚴格的流程中製作出來的，則其整體的分類不一致性，應當相當一致。因此，最好的方法是同時採用不同分類測試集，而且越多越好，以便獲得可靠的不同程度的不一致性。爲此，除了採用過去研究常用的分類測試集外，我們特別尋找到一份橫跨17年歷史的分類資料，製作成新的測試集，供研究使用。第三節將介紹本研究使用的測試集，一份爲中文的測試集，稱爲 FJU CTC測試集，經由本實驗驗證，其具有相當低的分類一致性，另一份爲英文的 Reuters-21578，其爲文件分類研究常用的測試集。有此兩測試集，就可在比較嚴謹的環境下，探討不一致性對自動分類效果的影響了。此部份將於第四、五節討論，最後一節則爲本文的結論。

貳、相似文件自動偵測

在一份文件集（a collection of documents）中，偵測文件相似性的技術可運用於複本偵測（重複文件剔除）、文件歸類、同案追蹤、抄襲偵測等，因此是相當有應用價值的技術。然目前這方面的探討不是很多，本節介紹三種自動偵測的方法，其各有優缺點，分述如下：

一、R-measure

Khmelev 和 Teahan 根據文件子字串的重複程度，提出 R-measure 的方法來計算文件集內各文件的相似度。（註 6）所謂字串（string）可視爲一連串的字元（character）。以 T 代表一字串，則以 T[i..j]代表其子字串（substring），其中 i 與 j 爲正整數，且 i<=j。若以|T|代表 T 的長度，則 T=T[1..|T|]，T[i..i]爲 T 的第 i 個字元，T[1..j]稱爲 T 的字頭字串（prefix），

臺灣大學學術期刊資料庫

T[i..|T|]稱爲T的字尾字串（suffix）。

假設文件集中有 m 篇文件，由 $T_1, T_2, ..., T_m$ 表示，根據 Khmelev 和 Teahan 的定義，某篇文件 T 在此文件集中的R值爲：

$$R(T|T_1, T_2, \cdots T_m)$$

$$= \sqrt{\frac{2}{l(l+1)}\sum_{i=1}^{l} Q(T[i..l]|T_1, T_2, ... T_m)}$$

其中l代表文件T的字串長度（亦即l=|T|），T[i..l]代表T的第i個字尾字串，$Q(S|T_1, T_2, ..., T_m)$代表 S 的某個字頭字串的長度，此字頭字串在 $T_1, T_2, ..., T_m$其中一份文件出現，而且是 S 的字頭字串中最長的。例如，假設T="cat sat on", T_1="the cat on a mat", T_2="the cat sat"，則

$R(T|T_1, T_2)$

$$= \sqrt{\frac{2}{10 \times 11}((7+6+5+4+3)+(5+4+3+2+1))}$$

$$= \sqrt{0.7272} = 0.8528$$

其中7是來自於"cat sat"這個字串（其爲 T[1..10]出現在T_2中的最長的字頭字串），其長度爲 7；而 6 來自於"at sat"，其爲T[2..10]出現在T_2中最長的字頭字串；第二個 5 來自於"at on"，爲 T[6..10]出現在T_1中最長的字頭字串，其餘依此類推。亦即，T 在此文件集中，跟某些文件相似的程度爲0.8528。

爲了同時計算每一份文件Ti在文件集中的相似度，Khmelev 和 Teahan 建議將所有文件串接起來變成單一個長字串，即$S_c = T_1\$T_2\$...\$T_m\$$，其中$爲文件分隔符號，然後利用 suffix array 的技術（註7）來計算。其所需計算時間（計算量）約$O(|S_c|)$，亦即跟所有文件長度的總和成正比，或者相當於O(n*m)，其中n是文件的平均長度。而其所需記憶體爲$O(|S_c|) + O(m)+O(M)$，其中$M = \max_{j=0,...,m}|T_j|$。

二、S-measure

直接運用 R-measure 於相似文件的偵測並不適合，因爲很多資訊檢索的工作，經常要對文件進行字幹處理（stemming）、停用詞（stopword）捨棄、標點符號刪除等動作。因此，相似文件的個數（或相似文件的定義）會因索引詞策略的不同，而有所不同。雖然文件可經上述前置處理後，再計算 R-measure，但相似的文件，詞彙順序可能會有稍爲更動的情況，這對依賴於字串順序必須一致才能偵測相似度的 R-measure 而言，就不適用了。

有鑑於此，筆者提出利用資訊檢索的技巧來偵測相似文件的方法，稱爲 S-measure。此方法應用檢索系統對文件集建立索引，然後將每一篇文件視爲檢索字串依次送入檢索系統中做相似度比對的查詢，蒐集每一個查詢結果即可獲得每一篇文件對其他文件的相似程度。如此檢索系統可對文

臺灣大學學術期刊資料庫

件的標點符號、雜訊、詞彙等做前置的正規化（normalization）處理，查詢時可選用不同的相似度計算方法，只要其最後的數值可以正規化到0與1之間代表其相似度即可。此方法最重要的優點是：一、不受詞彙順序的影響；二、可以藉由控制、調整比對的查詢字串，而定義出不同的相似文件。例如，我們可以刪除文件所有的停用詞（通常爲虛詞、連接詞等無主題意義的詞彙），或只留下出現頻次超過一定門檻的關鍵詞，進行相似度比對，如此會容許較多的文件有較高的相似度，即便這些文件在停用詞上有很多差異，只要其關鍵詞相似即可。相對的，我們也可以拿文件的全部字串送入檢索系統做相似比對，如此只有複本或近似複本才可能有較高的相似度出現，因而定義出比較嚴格的相似文件。

此方法需事先對文件集建立索引，其效率爲檢索系統建索引的效率。一般設計良好的檢索系統，建索引所花的時間大約跟文件總長度成正比。在比對單一篇文件的相似度上，由於透過檢索系統的反向索引檔運作，其需要的計算時間最長約O(n*m)，所需的記憶體最高也是約O(n*m)，其中 n 爲該文件的詞彙數，m 爲所有的文件篇數。因此計算每一篇文件對其他文件的相似度，全部需要$O(n*m^2)$的計算量，而記憶體需求爲O(n*m)。這個估計結果，是假設最糟糕的情況發生，亦即每一篇文件的每個詞彙都出現在所有文件裡。在自然語言的文字分佈上，這幾乎不可能，只會有少數的詞彙出現在很多文件，而絕大多數的詞彙只出現在少數的文件。因此，實際運用 S-measure 時，其全部的計算量應該是遠低於$O(n*m^2)$（約O(n*m)，甚至爲O(n)），所需的記憶體，則約O(m)。

三、D-measure

S-measure 雖然完全改善 R-measure 過度依賴子字串順序的缺點，其完全無視於詞彙出現順序的特性，反而可能會造成寬鬆認定相似文件的缺點。尤其當我們刪除很多停用詞，進行相似性比對時，相似度高的文件，可能還需進一步的驗證，才能確定其是否爲相似文件。爲此，我們尋找出第三種相似文件偵測的方法，即動態規劃比對法（dynamic programming），在此簡稱爲 D-measure。它可以比對字串的順序，但也容許字串中有插入、刪除、轉換的情形，因此其嚴格程度，介於R-measure與S-measure之間。

在動態規劃裡，所謂編輯距離是指利用「插入」、「刪除」、與「代換」的動作，將一個字串轉成另一個字串「所需最少的步驟」（或是「所需最少的計算成本」）。在此我們運用的動態規劃法可表達如下：(註8)
假設有兩文件字串A與B，長度各爲n與 m。將兩字串從頭比對起，則比對

臺灣大學學術期刊資料庫

到A的第i個字（以A[i]表示）與B的第 j 個字（以 B[j]表示）的編輯距離爲：

d[i, j] = min(d[i-1, j] + w(A[i], 0),
d[i-1, j-1] + w(A[i], B[j]),
d[i, j-1] + w(0, B[j]))

其中min(X, Y, Z)表示取X, Y, Z 三個數中最小的值，而初始值爲：

d[0, 0] = 0
d[i, 0] = d[i-1, 0] + w(A[i], 0) , 1<= i <= n
d[0, j] = 0 , 1<= j <= m

另外，函數w(X, Y)的意義爲

w(A[i], B[j]) : 表示將A[i]代換成B[j]的計算成本
w(A[i], 0) : 表示插入 A[i]的計算成本
w(0, B[j]) : 表示刪除 B[j]的計算成本

以A = adc， B = adecdecf爲例，且假設代換、插入與刪除的計算成本都爲1，則d[i, j]可以表示成矩陣的第i列的第j行，如下：

A \ B	d	e	c	a	d	e	c	F
a	1	1	1	0	1	1	1	1
d	1	2	2	2	0	1	2	2
c	2	2	2	3	1	1	1	2

當上述動態規劃比對完成，從最後一列的後面掃瞄，找到數值最低者，即爲其編輯距離。爲了便於比較不同長度文件的相似度，Lopresti 等人（註9）以公式：

$$sim = \exp(\frac{edit}{edit - m})$$

將編輯距離轉換成相似度，其中 exp 爲自然指數（natural exponent），edit 爲編輯距離，m 爲短文件的長度。此相似度介於0到1之間，數值越高，相似程度越高。以上面的A與B字串爲例，其相似度爲exp(1/-2)=0.6065。

從上述矩陣的運算可知，計算任意兩文件的D-measure需要$O(n^2)$的計算量與記憶體，且假設兩文件長度都爲 n。計算文件集中所有文件的D-measure，則需$O(m^2*n^2)$的計算量。由於沒有快速的演算法可用，當文件量大時，D-measure 並不適合計算所有文件的相似度，而只適合少量文件的計算，例如用來驗證具備高 S-measure 的文件，是否也有高的D-measure。

參、分類測試集介紹

本文使用兩種測試集，一爲筆者根據輔仁大學中國社會文化研究中心（Socio-Cultural Research Center，簡稱「社文中心」）累積多年的資料製作而成的中文分類測試集，稱爲FJU CTC （註10），另一個爲文件自動分類研究常用的英文測試集Reuters-21578（Yang 版本）。分述如下：

一、FJU CTC

FJU CTC 測試集的來源爲1966年

臺灣大學學術期刊資料庫

到1982年之間中國大陸各電台的廣播內容，由當時在香港的中國消息分析（China News Analysis, CNA）天主教機構，爲蒐集當時難以獲得的大陸消息請人每天收聽大陸廣播轉寫文字而成，此資料集簡稱「廣播抄稿」。西元1994年，CNA的資料（以及人員）由香港移轉到輔仁大學社文中心。2000-2001 年間，由國科會計畫補助，將廣播抄稿請人工打字成電子檔案，共計 42371 篇廣播稿，其中只有 30710 篇文件有人工標示的類別。筆者根據下列自行發展的原則來篩選文件，以製作成此分類測試集：

㈠盡可能使用所有可用的文件，以充分運用既有的資源，包括人工的分類與電子化的檔案。

㈡爲了訓練並測試分類器的成效，文件集會分成兩部分：訓練組（training set）與測試組（testing set）。所有分在訓練組裡的文件其日期必須早於所有分在測試組裡的文件，如此才能反應一般分類器的實際使用情況。

㈢爲了有效訓練與測試分類器，每一個類別必須在訓練組與測試組中都有文件。若不符合此要求的文件與類別，則不納入分類測試集中。

㈣分類不一致的同一主題文件，最好能夠被指出來或隔離出來，以降低成效評估的不可靠性。

根據上述原則，筆者以約 70%文件作爲訓練、30%文件作爲測試的比例，將全部文件分爲訓練組與測試組。最後符合前三個原則的文件在訓練組有 19901 篇，日期範圍爲 1966/01/01 到 1976/12/31，而測試組有 8110 篇，日期範圍爲 1977/01/01 到 1982/12/26，共計 28011 篇。由於文件來自於廣播手抄稿以及人工打字，因此錯字、漏字，甚至句子或段落錯漏的情況，不算少見。但這並非缺點，任何文件都有這些情況發生，只是程度不同。此項特色，甚至可以考驗分類器容忍這些雜訊的能力。至於第四點原則，則是筆者的臆測，也是本文的動機之一，將在後面的內容中討論。

香港天主教CNA機構（以及之後的輔大社文中心）根據自行發展的分類表，對其館藏（包含七八十萬篇新聞剪報以及此廣播抄稿）進行人工分類。符合上述原則的類別共計82類，類別分佈呈現高度的偏向，亦即少數類別有很多文件，而大多數類別只有少數文件。表一列出最大與最小的十個類別的文件個數。

在類別標示方面，大部分文件只給一類，少數給超過一類，最多到四類，但多重分類的比例很少，因此其每篇文件的平均類別數，在訓練組爲 1.035 類，在測試組爲 1.007 類。由於此文件集橫跨17年之久，類別標示的一致性恐難保持。（註11）但任何分類測試集都有此問題，只是程度不同。這也是本文要偵測分類一致性，以及探討其對分類影響的主要原因之一。

臺灣大學學術期刊資料庫

表一　FJU CTC 最大與最小的十類文件統計表

編號	類名	訓練	測試	編號	類名	訓練	測試
1	P52	3985	1073	73	E39	9	2
2	P10	1903	788	74	E83	7	18
3	Cu4	1115	236	75	E1a	7	7
4	E34	1085	88	76	Cu73	5	2
5	E3	950	248	77	HK	4	10
6	P5	854	536	78	E1-3	4	2
7	E6	797	234	79	E31	4	1
8	P4	736	501	80	E1-4	3	74
9	E5	569	149	81	P9	3	1
10	E36	548	310	82	P53	2	9

二、Reuters-21578（Yang 版本）

根據 Lewis 的描述 （註12），此測試集來自於1987年由 Reuters Ltd.與 Carnegie Group, Inc.的專家進行整理與分類的路透社新聞稿。1990 年此份資料提供麻州大學（University of Massachusetts）電腦與資訊科學系資訊檢索實驗室使用。David D. Lewis 和 Stephen Harding 對其進行資料格式與相關資料檔的建置工作。1991 與1992 年間，在芝加哥大學資訊與語言研究中心David D. Lewis與Peter Shoemaker 進一步的整理其格式與檔案。此時的版本稱之爲「Reuters-22173, Distribution 1.0」，並於1993到1996年間置於麻州大學的anonymous FTP Server上供人下載。

1996年8月在 ACM SIGIR 的會議中，一組研究團體談論到 Reuters-22173 在不同研究中的結果難以比較的問題後，大家決定應該要制定新版的測試集，希望降低其格式上的模糊問題，提供文件清楚說明其標準的使用方法，並降低許多打字、分類及格式上的錯誤。

Steve Finch 和 David D. Lewis隨後在1996 年的9 月至11 月中完成了該項工作，並大量引用了SGML 標籤，同時除去了 595 篇重複的文件（根據文件的時間標籤，去除連秒數都相同的重複文件），最後形成了有 21578 篇文件的新版測試集「Reuters-21578, Distribution 1.0」。

在此測試集中，有五棵「分類樹」（或稱「分類架構」或「類別集合」（category set））， 分別是 EX-

臺灣大學學術期刊資料庫

CHANGES, ORGS, PEOPLE, PLACES 與 TOPICS。每篇文件都根據其內容給定每個分類樹內適當的類別，而且可以給定多個類別。例如 "nasdaq"（EXCHANGES）, "gatt"（ORGS）, "perez-de-cuellar"（PEOPLE）, "canada" , "japan"（PLACES), "gold"（TOPICS）。基本上，除了 TOPICS 外，其他分類樹的分類法則大體上可以根據文件內是否有提到其類別名稱即可分類，如文章中提到 australia, nasdaq 等，當然這些名稱最好是該文章的主題焦點。至於 TOPICS 的分類法則就沒那麼明顯，因此是文件自動分類最常用的分類樹。（註13）

針對 TOPICS，此測試集將其分成訓練組與測試組的分割作法（split）共有三種，稱爲：ModLewis、ModApte、ModHayes。其中最常用的是ModApte 分割，其訓練組有9603篇文件，測試組有3299篇文件，類別總數約120類。然而 Yang 在其研究中（註14），根據 ModApte 分割，將沒有類別的文件刪除，同時也限制只有在訓練組跟測試組都有文件的類別才留下來。如此類別總數剩下 90 類，訓練組文件則有7769篇，測試組有 3019 篇，平均每篇文件有 1.3 個類別。本文使用的 Reuters-21578 測試集，不管有沒有特別提到，都是根據 Yang的修改原則而來。

同樣的，此測試集的類別分佈也很不平均，表二列出最大與最小的十個類別的文件個數。

肆、近似文件自動偵測結果

有了上述兩個測試集，筆者利用 S-measure 來偵測兩測試集的相似文

表二　Reuters-21578（Yang 版本）最大與最小的十類文件統計表

編號	類名	訓練	測試	編號	類名	訓練	測試
1	earn	2877	1087	81	palladium	2	1
2	acq	1650	719	82	palmkernel	2	1
3	money-fx	538	179	83	rand	2	1
4	grain	433	149	84	castor-oil	1	1
5	crude	389	189	85	cotton-oil	1	2
6	trade	369	118	86	groundnut-oil	1	1
7	interest	347	131	87	lin-oil	1	1
8	wheat	212	71	88	nkr	1	2
9	ship	197	89	89	rye	1	1
10	corn	182	56	90	sun-meal	1	1

臺灣大學學術期刊資料庫

件，同時也請 Teahan 先生運用其 R-measure 技術來偵測兩測試集的相似文件，以作爲比對。運用 S-measure 時，需要一套檢索系統，筆者以 WebGenie 系統（註15）對文件建索引，並進行整篇文件的相似比對。WebGenie 目前建索引的策略是文件中任何單一字符（token，即一個「中文字」或一個「英文字（word）」）、雙連字符（2-token，即文件中所有相連的兩個字符），以及多字關鍵詞都建爲索引詞，而且沒有停用詞（亦即所有的詞彙包含 the、of、我、你等字或詞都建入索引，沒有因爲該字或詞爲停用詞而沒有鍵入索引）。除了提供布林（Boolean）查詢外，WebGenie 也提供相似度比對的查詢，其計算公式乃參考 Singhal 等人用以改良 Cosine 相似度的計算方法（註16），稱做 bytesize 正規化（bytesize normalization）公式，如下所示：

$$Sim(d_i, q_j) = \frac{\sum_{k=1}^{T} d_{i,k}\, q_{j,k}}{(bytesize_{d_i})^{0.375}\sqrt{\sum_{k=1}^{T} q_{j,k}^2}}$$

其中d_i代表文件i，q_j代表查詢j，$d_{i,k}$代表詞彙k在文件i中的權重，常爲其詞頻的函數（如$\log(1+tf_{i,k})$），$q_{j,k}$代表詞彙k在查詢j中的權重，常爲詞彙出現篇數的函數（如$\log(1+n/df_k)$），n 爲所有文件的個數），而 bytesizedi 表示文件 i 的長度（以 byte 爲單位）。此公式乃 Cosine 公式的近似，比 Cosine 容易計算，且比較不受文件雜訊（如打錯字或OCR錯字，造成詞頻統計錯誤）的影響。其值最小爲 0，最大超過 1。當超過 1 時，視需要可以切截成1，使其落在0與1之間，以表示相似度。在 Singhal 以及我們過去的成效評估中（註17），此相似度公式可以得到比 Cosine更好的檢索成效。

筆者採用此商用系統最主要的理由，是此系統以 C 及 C++寫成，花費數年調整效能，因此其建索引的速度極快，同時提供 150 多個功能完備的 API（Application Programming Interface）可供呼叫與運用，以程式設計的角度而言，在整合不同平台與程式系統方面，極爲便利。當然，其他免費或商用的檢索系統，只要具備主題查詢功能而可輸出相似度者，均可利用。

爲了凸顯 S-measure 的彈性，對 Reuters-21578 的測試集，筆者沒有做任何處理，就將全文送入查詢，跟其他文件一起做相似性比對。平均而言，每篇文件有 133 個英文字（佔原文件詞彙數的 100%）被送入查詢。而對於FJU CTC測試集，在建完索引後，筆者將WebGenie裡全部的索引詞倒出來，作爲斷詞詞庫，對每一篇文件進行長詞優先的斷詞後，刪除功能詞、連接詞等無主題詞彙，再送入查詢。平均而言，每篇文件約只有88個詞彙送入查詢，約佔原文件詞彙數的38.42%。

臺灣大學學術期刊資料庫

依照上述細節進行的相似文件自動偵測，其結果如表三所示。其中R-measure 方法偵測出的相似文件數量，在 Reuters-21578 中有 177 篇，在 FJU CTC 中有 113 篇。這些文件跟文件集中某些文件，有最高的 R-measure 相似度（>=1.0），亦即這些文件是其他文件的複本。當我們使用 S-measure 偵測時，則複本數分別提高到 269 與 1956 篇。之所以會有如此的差別，一如前述，上述三種相似度計算方式，都有其認可的範圍。當索引詞定義不同時，相似文件數也會不同。舉例而言，表四是 Reuters-21578 中的兩份文件，編號分別是 2782 與 2880，它們只有少數幾個地方不同，如底線所示。此兩份文件的相似度，依照R-measure爲0.17，依照D-measure爲0.83，而依照S-measure爲1.0。顯示 R-measure 非常不能容許少數文字的差異，而 S-measure 容許度最高，D-measure 次之。這兩篇文件，是從 S-measure 偵測出的269篇複本中，以 D-measure 做確認，找出 D-measure 最低的文件而來，因此是這 269 篇複本中最不相似的兩篇。就人的判斷而言，這兩篇可被認爲是複本關係，而少有爭議。以此例而言，S-measure 成功偵測出複本，其他則否。運用 R-measure 遺漏偵測的情況，算蠻嚴重的，在 Reuters-21578 中遺漏 269-177=92 篇，在FJU CTC中遺漏1965-113=1852 篇。FJU CTC 的情況似乎很嚴重，其原因在於我們連停用詞都拿掉不比較，只比較具有主題意思的詞彙。因此，像表五中的兩篇文章，可算是相同主題，但其字串有很多不同。以R-measure 一定無法偵測，但 S-measure 的相似度爲 1.0。同樣的此兩篇文件也是從 1965 篇複本中，透過 D-measure確認而來（其D-measure相似度只有0.0839）。

當然S-measure的偵測結果，也會有引起爭議之處。第一種情況是文件A是文件B的一部份時（即A可能是B的節錄，而 B 是完整版），則可能會出現A到B（即用A做查詢）的相似度有最高值 1.0（但 B 到 A 的相似度低），此時會被認爲文件 A 與文件集中的某一篇文件爲複本關係。此缺點可將上述相似度計算公式改回 Cosine 公式來改善。第二種情況，則來自於

表三　複本偵測結果

	R-measure At R>=1.0		S-measure At S>=1.0	
Reuters-21578	177	1.64%	269	2.49%
FJU CTC	113	0.40%	1965	7.02%

臺灣大學學術期刊資料庫

表四　S-measure 偵測出來的兩份相似文件：文件編號 2782 與 2880

2782: TECHNIGEN PLATINUM CORP> IN METALS FIND. Technigen Platinum corp said it initial results of a 13-hole drilling program on its R.M. Nicel platinum property in Rouyn-Noranda, Quebec, indicate "extensive" near-surface zones "highly" enriched in gold, platinum and palladium were found in rocks on the periphery of a sulphide deposit.
2880: TECHNIGEN PLATINUM CORP IN METALS FIND. Technigen Platinum corp said initial results of a 13-hole drilling program on its R.M. Nicel platinum property in Rouyn-Noranda, Quebec, indicate extensive near-surface zones highly enriched in gold, platinum and palladium. The metals were found in rocks on the periphery of a sulphide deposit.

表五　S-measure 偵測出來的兩份相似文件：
文件編號 00631 與 03671

00631: 大量印行「寶書」 ….，我省從去年九月到今年七月，十個月內，就印製發行了「毛澤東選集」三百八十三萬部。這個數字等於文化大革命以前十五年出版發行數的十倍以上。同時，還印製發行了「毛主席語錄」、「毛澤東著作選讀」和毛主席著作單行本一千多萬册。 目前，「毛澤選集」、「毛主席錄」、毛主席著作單行本，在我省城鄉各地已遍大行。到在爲止，武漢地區及全省廣大工農兵群眾，都普遍購買了「毛澤東選集」、「毛主席語錄」等寶書。 全省農村，幾乎家家户户都有了「毛主席語錄」。….。
03671: 三年來毛著大量出版發行 文化大革命近三年來，我省大量出版發行毛主席著作。據統計：從一九六六年到今年三月，全省已出版發行「毛澤東選集」（….），相當於文化大革命以前十五年出版發行總合的十七倍。同時，還出版發行了「毛澤東著作選讀」二百三十多萬册，「毛主席語錄」四千七百多萬册，毛主席的「老三篇」、「老五篇」等….著作彙綿本、單行本三千多萬册，毛主席的光輝形相、照片和張貼的毛主席語錄八千多萬張。

非主題詞彙不拿來比較，因此會有字串差異很大的（短）文件，會被偵測爲複本的情況。如表六所示，其 S = 1.0，但 D = 0.0779。就複本而言，這兩種情況都是錯誤的偵測，但就相同主題的文件而言，這些還是正確的偵

臺灣大學學術期刊資料庫

表六　S-measure 偵測出來的兩份相似文件：
文件編號 12174 與 16719

12174: 本台新聞節目時間表 從五月十四日起，本台實行一九七三年夏秋季節目時間表播音。這個節目時間表實行後，新聞節目每天四次： 第一次：九、三O——九、五O。 第二次：十三、三O——十三、四五。 第三次：二十一點到二十一點二十分。 第四次：二十三點五十分到零點。
16719: 新疆電台新聞節目播送時間表 本台從五月十二號起實行新節目時間表播音。 新聞節目播送的時間是： 北京時間：每天九點十分、十四點、十八點十分、二十三點（每星期一至星期五二十一點）。

表七　不同 S-measure 相似程度的近似文件數與分類一致性統計表

Collections	label	S>=1.0		S>=0.8		S>=0.6	
		# Doc	%	# Doc	%	# Doc	%
FJU CTC	Inconsistent	1922	6.86	7894	28.18	9618	34.34
	Consistent	43	0.15	351	1.25	2438	8.70
	Total	1965	7.02	8245	29.43	12056	43.04
Reuters-21578	Inconsistent	62	0.57	121	1.12	246	2.28
	Consistent	207	1.92	628	5.82	1165	10.80
	Total	269	2.49	749	6.94	1411	13.08

測。

如果我們把S-measure的相似程度放寬，則我們可以得到不同數量的重複或相似文件。若把這些相似文件的分類狀況拿來比對，可以大致瞭解其分類的一致性。如表七所示，當 S-measure 放寬到0.6時，可以看出 FJU CTC 的近似文件有12056篇，其中有9618篇的分類不一致，佔全部文件集數量的34.34%。相對的，Reuters-21578則僅有246篇（佔2.28%）的文件分類不一致。

臺灣大學學術期刊資料庫

伍、分類不一致的影響：實驗與結果

如同前述，分類的不一致性，可能混淆了分類器的學習，造成其效果降低，同時也可能讓成效評估的結果變得不可靠或無法預期。爲了驗證此效應，筆者設計如下的實驗：把近似文件（不管其分類是否一致）從原文件集中刪除，消除明顯的分類不一致性或一致性後，將剩下的文件集拿來訓練與測試分類器，此成效跟原文件集的成效做比較，看看分類成效有何變化。

在文件自動分類的技術中，SVM（Support Vector Machine）與 KNN（K-Nearest Neighbor）是被驗證爲效果最好的兩種分類方法。（註18）由於 SVM 的訓練所需時間比較不容易預測（有時很長有時很短），在此實驗中，筆者選用 KNN 分類器。KNN 的分類原理算蠻簡單的：將輸入文件與所有的訓練文件做比對，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。以公式表示如下：

$$y(q, c_j) = \sum_{d_i \in KNN} Sim(d_i, q) y(d_i, c_j)$$

其中 y(d,c)爲1或0的值，代表訓練文件d是否爲類別c，而KNN表示跟文件 q 最相近的 K 篇訓練文件的集合。在後面的實驗中，K 都取20，且在FJU CTC的實驗中，由於大部分文件都只分一類，因此以 y 值中最大者的類別 c，做爲文件 q 的類別，而在 Reuters-21578 的實驗中，由於平均一篇文件分到 1.3 個類別，因此除了指定y值最大的類別給文件q外，若排序第二大的 y 值，其爲第一大的一半或以上，也將其類別指定給該文件。

由於影響KNN分類成效的最大因素是相似性的準確度，筆者於是試驗了兩種相似度計算方式，一種是前述的 bytesize 正規化公式，另一種是稱爲 BM11的運算模式。BM11是機率模式（Probability Model）的相似度計算公式（註 19），在近年的檢索實驗中，被驗證爲比bytesize或Cosine這種向量模式（Vector Space Model, VSM）的效果還好，其公式如下：

$$BM11(d_i, q_j) = \sum_{k=1}^{T} tf_{j,k} \log\left(\frac{n - df_k + 0.5}{df_k + 0.5}\right)\left(\frac{tf_{i,k}}{tf_{i,k} + \frac{dl_i}{Avgdl}}\right)$$

其中$tf_{j,k}$爲詞彙k在文查詢j中的出現次數，df_k爲詞彙 k 在文件集中的出現篇數，dl_i是文件 i 的長度（可爲任何單位），Avgdl 是所有文件的平均長度，其他符號同bytesize公式。

在成效評估方面，不同的度量方式有各自不同的強調對象，因而容易導致偏向（bias）的結論。本文以MicroF以及MacroF值同時呈現分類的效果，其計算方式如下：

$$MicroF = \frac{2 \times \sum_{i=1}^{c} TP_i}{2 \times \sum_{i=1}^{c} TP_i + \sum_{i=1}^{c} FP_i + \sum_{i=1}^{c} FN_i}$$

臺灣大學學術期刊資料庫

$$\text{MacroF} = \frac{1}{c}\sum_{i=1}^{c}\frac{2 \times TP_i}{2 \times TP_i + FP_i + FN_i}$$

其中C是類別總數，i代表某一類別，而TP_i（True Positive）、FP_i（False Positive）、FN_i（False Negative）分別代表：是類別i而且也正確分爲類別i的篇數、不是i類卻分爲i類的篇數、是i類卻沒有分爲i類的篇數。此F值乃從精確率（P）與召回率（R）的常見公式：F = 2PR/(P + R) = 2TP/(2TP + FP + FN)演化而來，其中P = TP/(TP + FP)、R = TP/(TP + FN)。

由於MicroF是全部文件一起累加統計，不分類別，因此容易受到少量的大類別（佔大多數文件）表現好壞的影響。相對的，MacroF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。將兩種平均數據都報告出來，可以瞭解大多數文件的分類效果（MicroF），以及大多數類別的分類效果（MacroF）。

表八顯示刪除近似文件後的成效，其中第三欄是用測試集的全部訓練文件與測試文件得到的結果；第四欄則刪除所有 S-measure 大於等於1.0的文件（包括訓練與測試文件）；第五欄刪除所有 S-measure 大於等於0.8的文件；第六欄則刪除所有 S-measure 大於等於0.6的文件。歸納結果如下：

一、所有的情況都顯示，機率模式（BM11）的相似度計算比向量模式（VSM）效果還要好。

二、在FJU CTC測試集中，雖然不一致的近似文件遠多於分類一致的近似文件，但是分類成效在近似文件移除後，並沒有顯著的變好。

三、在Reuters-21578測試集中，雖然一致的近似文件遠多於分類不一致的近似文件，但是分類成效在近似文件移除後，並沒有顯著的變差。

上述結果顯示，在我們偵測到的近似文件比例以及不一致比例的程度上，分類器的效果都不至於大受影響。這一點跟我們原先的預測與常識判斷大相逕庭。筆者猜測這種不一致不僅出現在相似文件，也出現在同主題但不相似的文件，因此即便刪除了高達43%的文件，剩下的文件的不一致性（或一致性）還是差不多，因此分類效果沒有差別太大。

陸、結語

本文探究分類的（不）一致性對分類效果的影響。透過近似文件的自動偵測，從其類別資訊得知分類不一致性的情況，在刪除明顯的分類（不）一致文件後，進行成效評估，結果發現分類器的效果沒有受到明顯的影響。這個探究過程與結果有幾個重要的意涵：

一、重複文件對影響分類器的學習成效並不大，至少在文件集夠大時是如此。過去的研究，例如 Teahan 等人（註 20），對重複文件

臺灣大學學術期刊資料庫

表八　MicroF（上半部）與 MacroF（下半部）

Collection	Model	All docs.	S<1.0	S<0.8	S<0.6
FJU CTC	VSM	0.4412	0.4504	0.4605	0.4787
	0.4383	0.4494	BM11	0.4679	0.4788
Reuters-21578	VSM	0.8170	0.8156	0.8240	0.8247
	0.8192	0.8161	BM11	0.8253	0.8232
FJU CTC	VSM	0.2916	0.2844	0.3177	0.3182
	0.2881	0.2844	BM11	0.3236	0.3080
Reuters-21578	VSM	0.3676	0.3428	0.4211	0.4072
	0.3681	0.3603	BM11	0.4206	0.3823

影響分類器的擔憂應該不存在。當然，對小文件集而言，重複文件仍有影響。（註 21）

二、Reuters-21578 是被廣泛運用的分類測試集，本文偵測出其複本情況還蠻嚴重的，高達 2.49%，但是本實驗顯示這些複本不會影響分類效果。因此過去以 Reuters-21578 爲實驗對象的研究，其結果都仍舊有效（valid）。（否則以往的研究，其結果都不可信，必須排除重複或近似的文件，再重做所有的實驗。）

三、本實驗顯示不一致性高的測試集，其分類成效也比較差。當然分類成效受到很多因素影響，不一致性高可能跟分類架構的設計或類別的定義與區隔等因素有連帶的關係。只要調整類別的定義與架構，也許就能改善不一致性，同時也改善訓練與分類的效果了。

本文得到的結論，必須小心解讀。例如，筆者在做一份小文件集的分類實驗時，就碰到這種現象：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，「體育」類下面還有「職業棒球」類，像報導郭泰源的文件，在訓練資料中有時候被分爲「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，顯示訓練與測試文件的分類相當不一致。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，但是因爲答案是「體育」，因此較好的分類器反而得出較差的數據。

上述結果，可能是文件集小的關係，也可能是訓練文件與測試文件的分類一致性分佈不平均或分佈不一致的關係。

臺灣大學學術期刊資料庫

最後，本文的貢獻包括：一、提出有效的近似文件自動偵測方法；二、發展出一套中文的分類測試集；三、探討分類一致性對分類成效的影響，本文說明其重要意涵，並提醒讀者小心解讀其結果。

誌謝：

本研究由國科會專題研究計畫補助：NSC 92-2213-E-030-017-。另外，特別感謝輔仁大學中國社會文化研究中心累積十數年的廣播手抄稿及其人工分類資訊，才得以發展出此中文分類測試集。

註釋：

註 1：曾元顯，「文件主題自動分類成效因素探討」，中國圖書館學會會報 68 期（2002 年 6 月）：頁 62-83。

註 2：Fabrizio Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys 34:1(2002):1-47.

註 3：M. Kreines, "Reuters Corpus Problems," 2 Oct. 2001, <http://trecfiltering@list.research.microsoft. com>

註 4：卜小蝶，「網路使用者檢索詞彙主題分類探析」，在台灣大學圖書資訊學系四十週年系慶研討會論文集，2001 年 11 月 16 日，頁 113。

註 5：Anne Kao, "Re: Reuters Corpus Problems," 2 Oct. 2001, <http://trecfiltering@list.research.microsoft.com>.

註 6：Dmitry V. Khmelev and William J. Teahan, "A Repetition Based Measure for Verification of Text Collections and for Text Categorization," ACM SIGIR (2003): 104-110

註 7：U. Manber and G. Myers, "Suffix Arrays: a New Method for On-line String Searches," SIAM Journal of Computations 22:5 (1993): 935-948.

註 8：Daniel Lopresti and Jiangying Zhou, "Retrieval Strategies for Noisy Text," in Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval, April 15-17, 1996, 255-269.

註 9：同前註。

註 10：Yuen-Hsien Tseng, "FJU Test Collection for Evaluation of Chinese Text Categorization," <http://www.lins.fju.edu.tw/seng/Collections/Chinese_TC.html>

註 11：Yuen-Hsien Tseng and William John Teahan, "Verifying a Chinese Collection for Text Categorization," in Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '04, July 25 - 29 Sheffield, U.K., 2004, 556-557.

註 12：David D. Lewis, "Reuters-21578 Text Categorization Test Collection, Distribution 1.0" README file (v 1.2), 1997, <http://www.research.att.com/ewis/>

臺灣大學學術期刊資料庫

註 13：Franca Debole and Fabrizio Sebastiani, "An Analysis of the Relative Hardness of Reuters-21578 Subsets," Journal of the American Society for Information Science and Technology.

註 14：Yiming Yang and Xin Liu, "A Re-Examination of Text Categorization Methods," in Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1999, 42 - 49.

註 15：WebGenie 3.23, <http://www.webgenie.com.tw/>

註 16：Amit Singhal, Gerard Salton, and Chris Buckley, "Length Normalization in Degraded Text Collections," in Proceedings of Fifth Annual Symposium on Document Analysis and Information Retrieval, April 15-17, 1996, 149-162.

註 17: Yuen-Hsien Tseng and Douglas W. Oard, "Document Image Retrieval Techniques for Chinese," in Proceedings of the Fourth Symposium on Document Image Understanding Technology, Columbia Maryland, April 23-25th, 2001, 151-158.

註 18：同註 14。

註 19：S. E. Robertson and S. Walker, "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval," in Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1994, 232-241.

註 20：同註 6。

註 21：Yuen-Hsien Tseng and Da-Wei Juang, "Document-Self Expansion for Text Categorization," in Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '03, July 28 - Aug. 1, Toronto, Canada, 2003, 399-400.

臺灣大學學術期刊資料庫