第二章 文獻探討

有關於試題方面的研究大部分都著重在試題品質的管控及建題機制方面，如檢查試題重複性、一致性、完整性及關聯性方面作探討 (蕭經武 民 88)；藉由固定數量的題幹正確解及誘答項，來排列組合出各種測驗題的智慧型造題機制，來達到快速豐富題目並避免因過度使用，而造成猜題現象的問題 (侯妤青 民 89)。本研究主要在於建構一套試題分類系統，並朝以下三方面進行探討:

1. 中文斷詞
2. 文件分類
3. 相似度分析

第一節中文斷詞

針對中文的試題而言，進行分類之前，必須先要可以對中文進行斷詞的工作，以便於了解每一種類型的試題中所包含的關鍵字詞有哪些。然而在斷詞方面，中文比起印歐文字要困難許多。

中文字與印歐文字的差別，主要在於斷詞的方式(Nie 1996)。以印歐語系而言，主要的特色是:

1. 一個字 (Word) 是由字母組成, 字與字之間使用空白隔開。
2. 一個英文字所代表的意義如果翻譯成中文，通常是數個中

文字所構成的字詞。

以中文字而言，主要的特色在於:

1. 字與字之間緊密相連。
2. 有意義的詞中所包含的文字與介詞或副詞可能重複使用。

中文文件大致上有下列幾種斷詞法，分別是：詞庫式斷詞法(Chen.K.J and S.H Kiu, 1992)、統計式斷詞法 (Fan 1988, Sproat 1990)、混合式斷詞法 (Nie 1996)及基因演算斷詞法以下將分別對這些斷詞法作說明:

一、 詞庫式斷詞法

為目前普遍使用的斷詞方法，其演算法相當直覺且實作容易。然而，斷詞的品質與詞庫大小有相當的關係。所以，必須時常對詞庫的內容加以維護，另外其他學者將詞庫斷詞法，輔以一些詞性的結構，發展出規則式斷詞法(陳克健 民 85)，提昇斷詞的品質。

二、 統計式斷詞法

統計式斷詞法(Sproat 1990)乃參考一大型語料庫(corpus)上的統計資訊，單純以鄰近字元同時出現頻率高低作為斷詞的依據。由於語料庫屬於領域相關(domain dependent)，不同語料庫間的統計資訊不適合

互用(Nie 1996)。再者，統計式斷詞常受限於一接馬可夫模式(first-order Markov Models)(Li 1991)，進一步擴充此模式會提高演算法的時間複雜度(Nie 1996)，所以大多只針對二字詞進行處理，三字詞如:「大賣場」、四字詞如「小額投資」等就無法有效擷取。

三、 混合式斷詞法

將詞庫斷詞法及統計斷詞法整合。(Nie 1996)利用詞庫斷出不同組合的詞彙，然後利用詞彙的統計資訊，找出最佳的斷詞組合。此法乃需要大型的語料庫提供統計資訊。

四、 基因演算斷詞法

基因演算法(Genetic Algorithms)是在 1975 年由 John Holland 提出，其理論是基於達爾文物競天擇、適者生存，不適者淘汰的為基礎，所發展出來的最佳化搜尋演算法。在一個群體中，會因為環境的限制，適應力較好的個體會存活，而延續下一代，經由數代的演化，而逐步的得到許多可能解，甚至是最佳解。

使用基因演算法來作中文斷詞的處理上 (謝佳倫 民 88)，假設使用者輸入的文字有 n 個字，則染色體長度為 n-1，若基因之為 1，則該基因所對應到字詞以下將作間隔。反之為 0 則不做間隔，以「強調將

堅決支持」七個字為例:

C_1	C_2	C_3	C_4	C_5	C_6	C_7	
強	調	將	堅	決	支	持	
1	0	0	1	1	0	1	← 染色體

斷開的關鍵詞 W1 其組成為 C1，用 W1 (C1)代表，第二組斷開的關鍵詞 W2 其組成為 C1,C2,C3，用 W2(C1,C2,C3)代表，以此類推第三及第四組的關鍵詞分別為，W3(C5)、W4(C6,C7)，則適應函數 (Fitness function) F 為:

$$F = \sum_{i=1}^{y} \left[T(W_i) \times L(W_i)^2 \right] \qquad (2\text{-}1)$$

式中

$T(W_i)$：關鍵詞 W_i 在詞庫中出現的機率

$L(W_i)$：關鍵詞 W_i 的長度

$0<i<y$, i 為正整數

IF $L(W_i) > N$ *THEN* $T(W_i) = -1$

IF W_i *can"t find THEN* $T(W_i) = 1$

經由數代的複製、交配、突變之後，得到最佳的斷詞如下:

C1 C2 C3 C4 C5 C6 C7

強 調 將 堅 決 支 持

0 1 1 0 1 0 1 ← 染色體

(引用自張育銘、黃國禎 民 90)

第二節 文件分類 (Text Categorization)

試題分類的概念將嘗試由文件分類的概念移植過來，因此以下將針對文件分類作說明。

文件分類通常所指的是由一群專家，針對一類群的文件進行分類的工作，然而隨著文件的增加，這樣的工作將會變得十分的困難，而且無法持續的進行這類的工作，因此自動化文件分類在智慧型的資訊系統中，是很重要的技術(M. Iwayama, 1994)。

文件分類的方法很多，大致上分成決策規則 (Decision rule)、知識庫 (Knowledge base)及文件相識度 (Text similarity)等等。

大部分的文件分類研究，都將文件的類型鎖定新聞內容，因為新聞內容在產生之初，都已經根據報社預先規劃進行分類，因此新聞內容及其分類的結果，是最容易取得而且客觀的研究素材。在教育領域中，試題也都會依據課程大綱或能力項目來出題並且分類，因此這樣的素材也是具有客觀標準。以下將針對文件分類做介紹:

一、 最早的文件分類文獻

英文的文件分類， 是由 Maron 於 1961 年提出 (Maron 1961)，其文件分類的方法，是由文件中所萃取出一些關鍵詞當作線索。並假設電腦可以從文件中自動萃取出這些關鍵詞，那便可以自動分類。其實驗採用 405 篇文章的摘要當作描述樣本，並使用 206 篇當作訓練資料，145 篇當作測試資料，結果得到 3263 個詞，去掉出現頻率最高，及只出現過一兩次的詞，留下 1088 個詞。再藉由亂度(Entropy)公式：

$$M(C)=-P^{+}log_2P^{+}-P^{-}log_2P^{-} \qquad (2\text{-}2)$$

其中

C 代表所要分類的文件類別

P^{+} 代表文件被歸類為"+"類的機率

P^{-}代表文件被歸類為"-"類的機率

將分佈平均者去掉，保留分佈不平均者，因為不平均者才有分類的價值。最後只留下 90 個詞的關鍵字。其實驗結果，訓練資料有 84.6%的回歸率，而測試資料也有 51.8%的回歸率。

二、 路透社 Reuters-21578

在英文的文件分類領域中，由 CONSTRUE 及 Hayes 採用 Reuters-22173 為路透社建立的規則式 (rule based)的文件分類系統，這套系統在人工與機器的分類上所得結果的一致性相當高。而在 1996 年 ACM SIGIR 研討會上，與會專家為了讓 Reuters-22173 有更高的標準。因此 Steven Finch 與 David D. Lewis 更著手刪除重覆的文件 595 篇，並減少了拼字上的錯誤，而使文件總數降為 21578 篇，並訂為 Reuters-21578。此外他們也為每一份文件訂上了統一的文件格式，如分別為文章的起始與結尾標上<REUTERS> </REUTERS>的標籤 (Tag)，讓文件分類研究者有一套統一的實驗資料標準(蔡文憲 民 87)。

三、 字詞權重函數 (Term Weighting)

字詞權重在文件分類的領域十分重要，其目的是要藉由權重，來得知哪些字詞可以成為分類的特徵字詞 (feature word)，而 Term Frequency 及 Inverse Document Frequency 在字詞權重是最基礎的理論，說明如下:

(一) 字詞出現頻率(Term Frequency, TF):

關鍵詞出現率指的是某一關鍵詞在某類文件中的出現次數, 在文件 *d* 中關鍵詞 *t* 的權重可定義為:

$$W(d,t)=TF(d,t) \qquad (2\text{-}3)$$

其中

TF(d,t) ：文件 *d* 中出現關鍵詞 *t* 的權重

TF 值可以得到很高的回歸率(Recall Rate)，但並不精密。主要是因為關鍵詞，如果經常出現在各種文件類別的話，這些關鍵詞作為某種文件分類的特徵並不明顯。因此最好將出現各文件類別中頻率高的關鍵詞從關鍵詞集(Term Collection)之中移除，以提高回歸率。

(二) 逆文件頻率(Inverse Document Frequency, IDF):

出現單一關鍵詞的文件數量稱之為逆文件頻率。逆文件頻率所表達的概念是，關鍵詞是否普遍的出現在各個文件當中，如果普遍出現的比例越高，則越無法突顯分類的特徵；相對的越低，最好是集中出現在同一個分類之中，則越容易突顯。假設關鍵詞為 *t*，則 *IDF* 定義如下:

$$IDF(t) = N/df(t) \qquad (2\text{-}4)$$

其中

N：代表文件的總數

df(t)：代表含有關鍵詞 t 的文件總數

如果使用 *IDF* 來表達關鍵詞的特徵(term specificity)，則將可以提高回歸率(Salton and Buckley, 1988)。

(三) 同時強調字詞出現頻率及普遍性 *TF* ✗ *IDF*

而根據 Salton 建議，如果使用 *TF*✗ *IDF* 當作加權數的話，則將會有更好的執行效率。

$$W(d,t) = TF(d,t) \cdot IDF(t) \qquad (2\text{-}5)$$

其中

W(d,t)：關鍵詞 *t* 在 *d* 文件類別的權數

TF(d,t)：關鍵詞出現率(Term Frequency)

IDF(t)：逆文件頻率(Inverse Document Frequency)

藉由上述的公式，可以很明顯的提高回歸率及精密度。

(四) 加權式逆文件頻率 (Weighted Inverse Document Frequency, WIDF) (M. Iwayama, 1994):

依據 *IDF* 的定義，只論關鍵字出現的文件類別總數，不論各類文件出現該關鍵字次數，將會出現特徵分佈不合理問題。以行來代表文件分類 d_i，以列代表關鍵詞 t_j，其出現頻率如下表:

表 2-1 文件集合範例

	D_1	**D_2**	**D_3**	**D_4**
T_1	0	40	3	0
T_2	2	50	3	2
T_3	3	2	3	2
T_4	0	80	0	0
T_5	0	30	20	0

(參考 M. Iwayama 1994)

以 T_1、T_4 及 T_5 的關鍵字而言，其分佈在 D_1~D_4 的頻率並不平均，因此可以很容易的辨別其分類。T_2 的分佈也不平均，在 D_2 有 50 的權重，而 T_3 可以清楚的看出權重分佈平均，而計算得到:

$$1/df(T2)=1/(1+1+1+1)=1/4$$

$$1/df(T3)=1/(1+1+1+1)=1/4$$

所得到的結果都是 1/4，結果所得到到 *IDF* 值都是 1，顯然並不合理。應該將其在該類文件中的出現頻率表現出來，所以如果

可以改成 50/(2+50+3+2)，將會更加合理。因此使用 *WIDF*，在文件 *d* 中含有 *t* 關鍵詞定義如下:

$$WIDF(d,t) = \frac{TF(d,t)}{\sum_{i \in D} TF(i,t)} \qquad (2\text{-}6)$$

其中

TF(d,t)：代表關鍵字在 *d* 文件類別中出現的頻率

$\sum_{i \in D} TF(i,t)$ ：*i* 代表 *D* 的文件集合的範圍內的各類文件。

因為 *TF(d,t)*已經包含在分子了，所以不需要再另外乘上 *TF(d,t)*。因此可將 *W(d,t)*定義為；

$$WIDF(d,t) = \frac{1}{\sum_{i \in D} TF(i,t)} \qquad (2\text{-}7)$$

由 M. Iwayama 的實驗中, 可以得知, 其回歸率確實比 *TF*〆 *IDF* 提高了 4.4%，而最高的回歸率也達到 96% (該實驗以英文為主)。

(五) 採用取平方的 *IDF*

由上述的內容得之，使用$\log\frac{N}{df(t)}$會降低詞的特殊性，所以分類效果不甚理想，為了拉大特殊詞與常用詞之間的差距，可以採用$\left(\frac{N}{df(t)}\right)^2$作為 *IDF* (林頌華 民 88)。則權重定義可調整成：

$$W(d,t)=TF(d,t)\cdot\left(\frac{N}{df(t)}\right)^2 \qquad (2\text{-}8)$$

其中

TF(d,t)：關鍵詞出現率(Term Frequency)

df(t)：代表含有關鍵詞 *t* 的文件總數

N：文件類別總數

在林頌華的研究用於處理中文新聞分類，採用二、三字詞為基本單位來處理太一新聞資料庫，在 5376 筆測試資料中對 53767 筆資料作計算，則以原始類別來說，可得到 78.89%的正確率，而第三高分則可高達 93.43% (該實驗以中文為主)。

以上所探討的文件分類關鍵字權重函數，可以發現大部份都是在逆文件頻率 *IDF* 上作調整，以達到最佳的分類效果，主要是因為太過於普遍出現的關鍵詞，會干擾到分類的效果;相反的關鍵詞越集中在單一的類別之中，則其效果越好。以下就分別依各種字詞權重函數*(TWF)*之優劣分析於表 2-2:

表 2-2 各種字詞權重函數 (TWF) 比較表

TWF	公式	分析
TF	$W(d,t)=TF(d,t)$	1. 可以表現出一個關鍵字在某文件類別上重要的程度。 2. 無法辨識出關鍵字是否普遍出現在各文件類別之中
IDF	$IDF(t) = N/df(t)$	1. 可以表現是一個關鍵字普遍存在於各文件類別的程度 2. 無法突顯關鍵字在單一文件類別的重要性
TF× IDF	$W(d,t)=TF(d,t) \cdot IDF(t)$	1. 可以表現是一個關鍵字普遍存在於各文件類別的程度，也可以了解關鍵詞是否普遍的出現在各種文件類別的程度。 2. 會減弱普遍存在於各文件類別，但卻有高出現率的的關鍵詞特徵。

表 2-2 各種字詞權重函數 (TWF) 比較表 (續)

TWF	公式	分析
WIDF	$WIDF(d,t)=\frac{TF(d,t)}{\sum_{i\in D}TF(i,t)}$	1. 可以表現是一個關鍵字普遍存在於各文件類別的程度，也可以了解關鍵詞是否普遍的出現在各種文件類別的程度。 2. 可以表現出普遍存在於各文件類別，但卻有高出現率的關鍵詞特徵。
TF× IDF^2	$W(d,t)=TF(d,t)\cdot\left(\frac{N}{df(t)}\right)^2$	1. 可以表現是一個關鍵字普遍存在於各文件類別的程度，也可以了解關鍵詞是否普遍的出現在各種文件類別的程度。 2. 可以讓集中在某一文件類別的關鍵字，其特徵更加突顯。

第三節 文件相似度(Text Similarity)

相識度測量方法分為是向量模式 (Vector Model)及機率模式(Probabilistic Model)兩類，以下將介紹兩類的測量模式:

一、 向量模式 (Vector Model)

文件向量可以用每一個包含在文件中的關鍵詞權重來構成，以用於區別文件之間的差別定義如下:

$$V_d=(w_1,w_2,\ldots\ldots,w_n) \qquad (2\text{-}9)$$

其中

w_i：$1<=i<=n$，代表每一個關鍵字的權重

V_d：文件類別 d 的向量

衡量向量間差異的方法有很多，以 Jaccard 函數來說，其定義如下:

$$Sim(V_i,V_j)=\frac{\sum_{k=1}^{n}(w_{ik}\cdot w_{jk})}{\sum_{k=1}^{n}w_{ik}+\sum_{k=1}^{n}w_{jk}-\sum w_{ik}\cdot w_{jk}} \qquad (2\text{-}10)$$

二、 機率模式 (Probabilistic Model)

藉由機率的理論所計算出文件 d 在 C_i 類別出現機率為 $P(C_i|d)$，則:

$$P(C_i \mid d) = \sum_t P(C_i \mid t,d)P(t \mid d) \qquad (2\text{-}11)$$

關鍵詞 t 的範圍超過 C_i 及 d 的向量元素，假設 C_i 及 d 的條件彼此獨立，則 $P(Ci|t,d)=P(Ci|t)$，則可以求得公式如下:

$$P(C_i \mid d) = \sum_t P(C_i \mid t)P(t \mid d) \qquad (2\text{-}12)$$

使用 Bayes rule，則最後可以得到，公式如下:

$$P(C_i \mid d) = P(C_i)\sum_t \frac{P(t \mid C_i)P(t \mid d)}{P(t)} \qquad (2\text{-}13)$$

關鍵詞 t 隨機於 C_i 類別取出，則其機率為 $P(t|C_i)$ ；關鍵詞 t 若隨機於文件 d 中取出，則其機率為 $P(t|d)$ ，$P(t)$及 $P(Ci)$分別代表關鍵詞及文件類別的機率，而所有的機率都是由訓練資料集所評估出來的。當一個文件 d 被歸類到類別 C_i 之後，則會得到 $P(C_i|d)$的最高機率。