第１章 緒論

１.１ 研究動機

這是一個資訊爆炸的時代，隨著網路的蓬勃發展，越來越多的資訊在網際網路上流通，該如何在如此龐大的資訊洪流之中找到我們所需要的資料，將會是一個益發重要的課題。

隨著網際網路的發達，人們的閱讀習慣正逐漸改變中，根據調查，雖然報紙仍是大學生最常閱讀的文字媒介，但是網路的閱讀率已經和書籍並駕齊驅了[15]。隨著電腦與網路的普及，相信未來的網路閱讀率終將超過報紙，成為人們最常接觸的資訊來源。而在網際網路中，又以全球資訊網（World Wide Web）最為人所熟悉，其便利的使用介面以及豐富多元的呈現方式，擄獲了越來越多的使用人口，這也使得全球資訊網逐漸躍升為網際網路中的主角，因此越來越多的網站如雨後春筍般成立，當中也包含了許多新聞網站，例如：中時電子報[16]、聯合新聞網[20]、明日報[19]…等。由於新聞網站眾多，為了讓使用者閱讀時更為方便，因此也出現了一些新聞整合性質的網站，例如：太一信通網羅新聞[18]、奇摩新聞[17]，其功能是收錄多家新聞網站的新聞報導，加以整合並提供多樣化的新聞服務，但是部份的新聞服務還是需要人力的介入，因此若我們可以讓電腦在這方面提供更大的助力，就能夠有效地減少處理新聞文件所花的時間，也更能節省花在這方面的人力資源。

１.２ 研究目的

太一信通網羅新聞所提供的「新聞群聚」就是一項需要人力介入的新聞服務，而為了使其更為自動化，我們將文件分群（document clustering）的技術應用在新聞文件上，達到新聞分群（news clustering）的目的。當網羅新聞系統到其他提供線上新聞報導服務的網站上抓取新聞標題（news title）之後，會先以新

聞分類（news classification）[13]的技術將所有的新聞報導分門別類。而新聞分群的工作就是針對每一個新聞類別，找出其中描述同一事件的新聞報導，將其合為一個群聚（cluster），以便於提供「新聞群聚」服務，這樣的服務除了方便使用者在線上同時瀏覽與比較多家新聞網站的報導之外，也可以再利用多文件摘要（multi-document summarization）[14]將同一群聚當中的新聞報導內容融合，藉以提供新聞事件的摘要服務。

本論文之目的在於利用字串相似度的度量方法求得新聞標題之相似度，並藉著與分群演算法的結合來提高電腦對於新聞標題自動分群的準確度。以太一信通網羅新聞為例，該網站在每天收錄各家新聞報導的標題之後，會先將收錄到的所有標題依網羅新聞所訂定的類別定義做新聞分類，隨後再針對各類別產生新聞群聚。但是目前完成新聞群聚的過程仍然需要人力的介入，使得整個過程較為耗時，因此若能讓提高電腦辨識新聞群聚的準確度，並協助完成分群的動作，整個系統即可有效地節省人力，處理時間也將更為快速。

1.3 研究方法簡述

在完成新聞群聚之前首先要針對同一類別當中所有的新聞標題做兩兩比對，找出兩者之間的相似度，往後才能根據相似度找出哪些新聞該視為同一個群聚。目前系統只採用新聞標題來計算相似度的原因是整個系統需要很快速地處理完所有的新聞，並馬上提供服務，若加上處理新聞內文，雖然對分群的正確率會有助益，但是對整體的速度也有極大的影響。因此我們透過只處理新聞標題來節省時間，而面對分群準確率會受到影響的情形，我們期望能夠利用論文當中提出的方法彌補這個缺陷。

在計算新聞標題之間的相似度時，除了利用基本的距離度量（distance measure）之外，我們也提出一個函數，藉以結合先前所求得的各項字串距離，函數中包含了多個參數來調整各項字串距離所佔的比重，我們利用不需導式的最

佳化方法（derivative free optimization method）[10]來對這些參數做最佳化，希望代入最佳化參數之後，經由此函數運算所得到的相似度能夠提升分群準確率。

取得相似度後即可利用分群法來將新聞標題做分群的動作，本論文使用的是階層式分群法（hierarchical clustering）[4,12]當中的階層式聚合演算法（hierarchical agglomerative algorithm），這個方法在執行過程中會將所有的新聞逐一合為群聚，直到所有的新聞連結成為一個二元樹。在連結的過程中將各階層的連結條件記錄下來，而系統的訓練階段就是試著從二元樹的任何階層切入，並把切入點以下仍然連結在一起的文件視為群聚，之後將此結果與資料庫中利用人工輔助完成的標準答案互相比較，反覆執行切入與比較的動作，最後就能夠找出表現最好的切入點，做為分群門檻值（clustering threshold）。而在測試階段裡，將各個類別的新聞各自形成二元樹之後，就可以找出樹中連結條件與訓練所得的切入點之值最為相近的一個階層，進而從該階層切入，即為分群結果。

1.4 章節簡介

本論文的安排如下：

第一章 說明進行此項研究的動機，以及希望達到的研究目標，並對於整個研究使用的方法與流程作簡單的說明。

第二章 介紹一些相關研究的論文，包括文件分群的相關工作，除了利用「階層式分群法」，也有利用「非階層式分群法」來完成分群工作，另外也有應用在多文件自動摘要系統上的新聞分群相關作法。此外我們也會簡單介紹一下本論文使用的新聞資料庫，對於這些新聞標題資料的特性做些簡單的描述。

第三章 說明在分群的過程中所使用到的方法，包括我們如何計算新聞標題之間的相似度，如何從同一類別的新聞中找出群聚，如何在分群過程中調整我們設定的一些參數以及分群門檻值。另外則介紹另一種以統計數據為

基礎的訓練方法。

第四章 簡介整個實驗的流程，針對每一個步驟作更為詳細的說明，最後說明實驗所使用的評估方式。

第五章 列出實驗結果並解釋實驗結果中數據所代表的意義。

第六章 針對實驗結果作一些相關討論，最後並提出未來的研究方向。

第 2 章 相關研究與資料庫概述

2.1 相關研究

文件群聚的技術可以將數量龐大的文件依其性質區分為許多群聚，讓使用者能夠快速地找出需要的文件，也更易於瀏覽。除了應用階層式分群法[4,12]來達到分群目的之外，也有非階層式的分群方法[5]，例如：k-means[8]和 Scatter/Gather[6]兩種分群法，運用此類方法通常需要先找出 k 個隨機的中心點，再根據各文件與這一些中心點的距離來決定哪些文件適於合為同一群聚，隨後再根據這些群聚得到新的中心點，並重新利用新的中心點來修正分群的結果，反覆操作這些步驟，最後就能順利得到分群結果。但是此類方法需要先設定分群的群聚數目，而我們在新聞分群的過程中並未能預先得知要形成多少個新聞群聚，或許當天重要的新聞事件較多，系統就需要形成較多的新聞群聚，反之，若當天新聞事件不多，可能只需形成少量的新聞群聚即可。由於新聞群聚數目有隨著日期不同而變動的特性，因此我們的系統並未採用這兩種非階層性分群法。

在應用階層式分群法所做的文件分群研究當中，最早是由 Jardine 和 van Rijsbergen[1]利用單一連結法(single linkage method)作為分群的工具。Griffiths[3]等人則是除了利用單一連結法之外，也加入了完整連結法（complete linkage method)、平均連結法（average linkage method）與沃德法（Ward’s method）來做比較，由比較的結果來看，發現單一連結法的表現最差，而表現較好的是沃德法與完整連結法，這是由於這兩種方法在分群過程中會產生較為平衡的二元樹，這使得在同一群聚當中的文件通常會有較高的相關性；反之，表現較差的單一連結法與平均連結法就容易產生不平衡的二元樹，亦即很容易產生包含大量文件的群聚，群聚當中的文件數目過度擴張的結果，就會造成不相干的文件很容易被歸為同一類別，因而影響了整體的表現。而在 Oren Zamir[11]等人利用階層式分群法所做的分群研究中，也提出了完整連結法雖是計算複雜度較高的方法，其複雜

度為 $O(n^3)$，不過表現是最好的。

除了純粹利用文件中的字數頻率來協助分群之外，也可以將文件內容加以斷詞再行分群，在 Oren Zamir 等人的論文中即比較了只用單字和加入斷詞所做的分群工作，其結論是加入斷詞之後的分群工作會有較好的結果。在黃聖傑與陳信希[14]的研究中，除了斷詞之外，還加以分析其詞性，利用詞性分析的結果，抓出名詞與動詞來計算文件之間的相似度，從其實驗結果來看，此方式所完成的新聞分群準確率可達九成。但由於一般斷詞都需要語料庫來輔助，而查詢語料庫通常較為耗時，若我們加上斷詞的工作，可能會拉長系統的處理時間，因此我們在短期內未有嘗試使用語料庫與斷詞方法的想法，我們期望能在不影響處理時間的前提之下，用其它的方法來拉高正確率。

2.2 新聞資料庫概述

本篇論文所使用的是太一信通的新聞資料庫，資料庫中所儲存的新聞標題是太一信通每天固定到新聞網站中所收錄的，而我們所使用到的新聞是從 50 個網站當中抓取回來的，平均一天會收到 1530 篇新聞，由於各家媒體都有自己的分類定義，所以在收錄完新聞之後，還需將這一些新聞重新對映到網羅新聞本身所定義好的 86 種類別當中，其中有一些類別是每天固定都會收錄到，例如：政治、娛樂、體育…等；有一些則是會根據新聞事件的出現而動態新增，例如：總統大選、921 大地震、世大運…等。

而網羅新聞在個別的類別之中，除了將描述相同事件的新聞合為一個群聚之外，並提供相關新聞的超連結以便於到各網站瀏覽新聞，而使用者除了可以在短時間內得知所有的新聞事件，也可以針對自己有興趣的事件，很方便地瀏覽與比較多家新聞的報導，在這講求時間與效率的資訊時代當中，可說是一項十分方便的工具。

若針對資料庫的內容來作分析，我們採用的新聞是從 1999/09/01 到 2000/04/14 之間的新聞事件，在這期間的新聞中，總共產生了 94248 個新聞群聚，而群聚當中的新聞數量波動也很大，從 2 則新聞到 50 則新聞都有，因此我們很難去預估說大概一個群聚當中需要幾則新聞才是恰當的，群聚中新聞數量的分佈如圖 2.1 所示，可以看出以 2 則新聞就組成一個新聞群聚佔是最多的一種，共有 14064 筆，佔了總數當中的 14.92%，另外我們也發現有部分的群聚當中只有包含 1 則新聞，共有 150 筆，會產生此類群聚是由於網羅新聞本身在顯示新聞時是以群為單位，而此類新聞雖然沒有其他相關報導，但卻又十分地重要，為了要將這一類新聞顯示出來，因此就須將新聞本身歸為一個群聚。

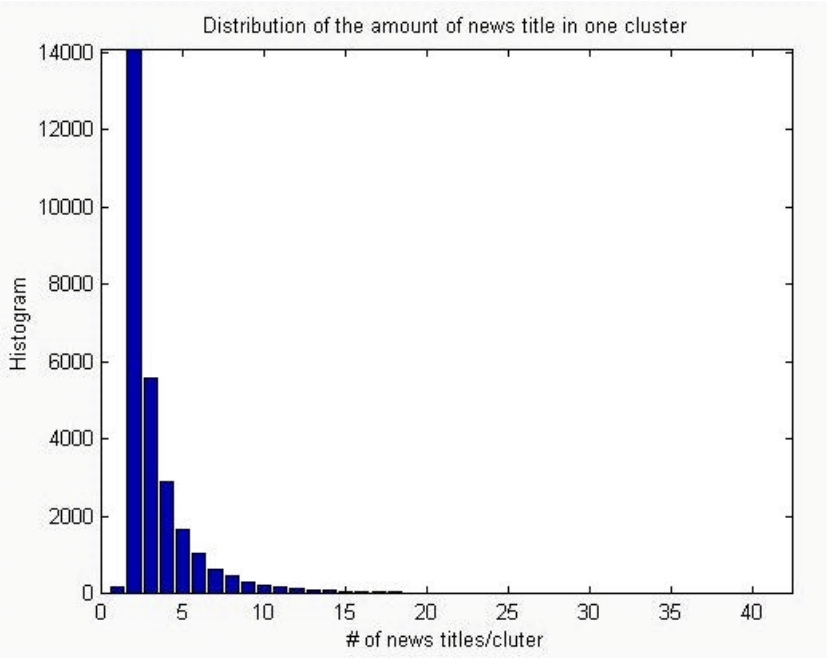


圖 2.1　各群聚中的新聞數目統計圖

由於新聞的類別眾多，所以我們很難找出一個適合於所有類別的最佳分群標準，因此我們採取的方法是在訓練階段中針對每一個不同的類別找出不同的分群標準。

第３章 研究方法

３.１ 字串相似度

由於我們是根據新聞的標題來分群，直觀來看就等同於多個字串的分群，因此我們利用比對字串之間距離的方法來取得相似度，而我們使用的方法有最長相同子字串（longest common subsequence）[9]、最長相同連續子字串（longest common consecutive subsequence）以及歐幾里德距離（Euclidean distance），最後我們提出一個函數，將上述的幾個方法得到的字串距離結合起來，希望能夠藉以提升新聞分群準確度。底下我們會對這一些方法做詳細的說明。

３.１.１ 最長相同子字串

最長相同子字串（longest common subsequence）簡稱 *LCS*，這是在比對字串相似度時常使用的方式，在解釋此方法之前，我們要先瞭解何謂子字串（subsequence），假設目前有一字串 **x** = [$x_1, x_2, \ldots, x_m$]，另有一字串 **y** = [$y_1, y_2, \ldots, y_n$]為 *X* 的子字串，則必定會找到一組遞增的索引[$i_1, i_2, \ldots, i_n$]，而有下列的結果：

$$x_{i_j} = y_j, \quad j = 1, 2, \ldots, k$$

舉例說明，假若 **x** = [C, A, D, A, B, C, A]，**y** = [A, D, B, A]，則對應的索引就是[2, 4, 5, 7]。

而假若目前有 **x** 與 **y** 兩個字串，且存在一個字串 **z** 恰同時為 **x** 與 **y** 的子字串，則我們就可以稱 **z** 為 **x** 與 **y** 的相同子字串（common subsequence），而在這一些相同子字串當中長度最長者，即為最長相同子字串（*LCS*）。例如有兩字串 **x** = [C, A, D, A, B, C, A] 與 **y** = [A, B, D, C, A, C]，則[A, D, C]，[C, A, C]與[A, D, A, C]均是 **x** 與 **y** 的相同子字串，而其中[A, D, A, C]擁有最長的長度，所以 *LCS*(**x**, **y**) = [A, D, A, C]。

LCS 在本論文中的應用如下，我們將每天收錄的新聞當中屬於相同類別的新聞標題兩兩代入前面敘述的 **x** 與 **y** 當中，所得到的最長相同子字串就是兩新聞標題之間的相似度。

3.1.2 最長相同連續子字串

由於我們發現在新聞標題當中，有時候一些特別長的相同子字串可能是一些對於分群更有幫助的詞彙，譬如底下的兩則新聞標題：

Title1：國政顧問團昨提前解散

Title2：前晚 國政顧問討論提前退場

則 *LCS*(Title1, Title2) = 國政顧問提前，而其中「國政顧問」算是句子當中很重要的訊息，因此我們想要加重這一類相同字句的重要性，於是我們除了使用 *LCS* 之外，還另外加上了另一種計算字串之間相似度的方式：最長相同連續子字串。

最長相同連續子字串（longest common consecutive subsequence）簡稱 *LCCS*，由字面上來看，這個方法所要尋找的也算是一種最長的相同子字串，但是與 *LCS* 最大的不同在於 *LCCS* 所尋找的必須是相同「連續」子字串，也就是說這個相同子字串在原字串當中的索引需為連續的值，而 *LCS* 的相同子字串就不存在這一項限制。

假設目前有一字串 **x** = [$x_1, x_2, \ldots, x_m$]，另有一字串 **y** = [$y_1, y_2, \ldots, y_n$]為 **x** 的連續子字串，則必定會找到一組連續且遞增的索引[$i, i+1, i+2, \ldots, i+n-1$]，使得：

$$x_{i+j-1} = y_j, \quad j = 1, 2, \ldots, k$$

舉例說明，**x** = [C, A, D, A, B, C, A]，**y** = [D, A, B, C]，則對應的索引就是[3, 4, 5, 6]。

倘若目前有 x 和 y 兩個字串，且存在一個字串 z 恰同時為 x 與 y 的連續子字串，則我們就可以稱 z 為 x 與 y 的相同連續子字串（common consecutive subsequence），在這一些相同連續子字串當中長度最長者就是「最長相同連續子字串」（*LCCS*）。例如有兩字串 **x** = [C, A, D, A, B, C, A]與 **y** = [A, B, D, C, A, D]，則[A, B]與[C, A, D]均是 x 與 y 的相同連續子字串，而其中[C, A, D]擁有最長的長度，所以 *LCCS*(**x**, **y**) = [C, A, D]。

為了使得新聞標題的相似度更為周全，所以我們在實驗中所使用的新聞相似度是將 *LCS* 與 *LCCS* 求得的距離結合在一起，希望能夠藉此提高新聞分群的準確度。

3.1.3 歐幾里德距離

但是有時候運用 *LCS* 與 *LCCS* 時，我們會發現如果新聞當中名詞出現的順序不相同，則上述兩者都無法完全掌握住重要的訊息，例如底下兩則新聞標題：

Title1：教育部長人選曾志朗長考

Title2：曾志朗教育部掌舵行事低調形象好

我們察覺兩種方法都無法將「曾志朗」與「教育部」同時找出來，但是這兩個詞彙在標題中都是很重要的，所以我們使用字串之間的「歐幾里德距離」來求其字串距離，這個度量方法的優點是只管字出現的頻率，不會因為字出現的順序不同而受到影響。

在求空間中兩點之間距離的時候，使用歐幾里德距離（Euclidean distance）是最基本的一種方式，假設 k 維的空間中有兩點存在，**x** = [$x_1, x_2, \ldots, x_k$]，**y** = [$y_1, y_2, \ldots, y_k$]，則兩者之間的歐幾里德距離就可以表示為：

$$d(x,y)=\sqrt{\sum_{i=1}^{k}(y_i-x_i)^2}$$

而在我們的應用上，由於我們處理的是中文的新聞標題，因此我們將大五碼（BIG-5）當中被歸類為常用字的 5401 個字做為參考的維度，各新聞標題就可視為是一個在 5401 維空間中的向量，其中每一個維度的值則是該維度所對應的中文字出現在標題中出現的頻率。我們藉此方式將新聞重新表示成多維空間中的向量之後，為了避免標題長度對字串距離有所影響，因此我們只取用這些新聞標題的單位向量（unit vector），將其代入上述的公式中即可求出新聞標題兩兩之間的距離。

3.1.4 權重平均法

由於前述三項計算字串距離的方式各有優劣，因此我們以權重平均法（weighted-average method）來將這些使用不同的度量方法所求得之字串距離結合在一起，我們的想法是針對每一項字串距離都給予它一個參數，隨後再找出方法來調整這一些參數，希望能夠調整出一個對分群最為有利的參數值，而我們提出的函數如下：

$$S_{i,j} = \frac{k_1 \times LCS_{i,j} + k_2 \times LCCS_{i,j} + k_3 \times \left(\frac{k_4}{Euc_{i,j} + k_4} \right)}{k_1 + k_2 + k_3}$$

Where $k_1, k_2, k_3, k_4 > 0$

$S_{i,j}$ 就是字串 i 與 j 的相似度，在我們的系統中字串 i, j 就是代表同一天中屬於相同類別的兩則新聞標題，而 k_1, k_2, k_3, k_4 即為我們所給予的參數值，$LCS_{i,j}$ 就是兩則新聞的「最長相同子字串」，同樣地，$LCCS_{i,j}$ 是兩字串的「最長相同連續子字串」，而 $Euc_{i,j}$ 則是兩字串的「歐幾里德距離」。

根據這一些度量方法的定義，我們會發現 $LCS_{i,j}$、$LCCS_{i,j}$ 兩者的度量值越大表示兩字串越相似，但是 $Euc_{i,j}$ 的度量值卻是越接近 0 表示兩字串越相似，為了能夠使這三者的涵義一致，所以我們對 $Euc_{i,j}$ 的值做若干修改，首先我們將 $Euc_{i,j}$

的值置於分母，變為 $\frac{1}{Euc_{i,j}}$，使其值能夠隨著字串距離越相近而變大。

但是若有兩則新聞標題完全相同的狀況，則 $Euc_{i,j} = 0$，使得 $\frac{1}{Euc_{i,j}}$ 無法計算，為了避免此狀況發生，我們在分母的部份加上一個值，但是又希望加入該值之後不會對整個評量方式造成太大的影響，所以就在分子與分母部份加入相同的值，在此我們也將加入的值定為參數之一，由系統在訓練階段當中來調整參數的大小，因此 $Euc_{i,j}$ 被重新表示為：$\frac{k_4}{Euc_{i,j}+k_4}$。

為了避免調整參數的過程當中，我們的目標函數 $S_{i,j}$ 會有過度擴張的現象，所以我們對參數值設定了兩個條件，第一個條件是我們在整個函數的分母部份加入 k_1, k_2, k_3 三者之和，第二個條件是 k_1, k_2, k_3, k_4 皆屬於正數，希望藉由這兩個條件，使得 $S_{i,j}$ 不會毫無限制地擴張。

決定了如何將字串之間的 LCS、$LCCS$ 與歐幾里德距離結合的方式之後，我們即可將這些距離度量的值代入公式當中，再利用程式反覆訓練並調整參數，得到最理想之參數值後，在往後測試階段的實驗中即可代入這些參數值。

3.2 分群法

分群法（clustering method）[12]主要就是要找出資料當中包含的群聚（cluster）數目，並可藉由找出各群聚的代表點，達到降低資料量的目的。而分群法主要分為兩類，一是「階層式分群法」（hierarchical clustering）[4]，另一類是「分割式分群法」（partitional clustering）。

一般而言，階層式分群法可劃分為「階層式聚合演算法」（hierarchical agglomerative algorithm）與「階層式分裂演算法」（hierarchical divisive algorithm）兩類，其主要結構都可以樹狀結構來表示，若是採用聚合演算法，則資料是由樹狀結構的底部向上方聚合；而若採用分裂演算法，則是由樹狀結構的頂端向下方

層層分裂。而本論文當中使用的是「階層式聚合演算法」，底下我們會對此方法做詳細的解說。

3.2.1 階層式聚合演算法

如前所述，階層式聚合演算法是由樹狀結構的底部開始向上層層聚合。起初我們將每一筆資料都視為是一個群聚，而在每一階層的運算過程之中，我們會將距離最相近的兩個群聚合為一個新的群聚，整個演算法的步驟如下：

1. 將 n 筆資料視為 n 個群聚，每一個群聚當中最少包含了一筆資料。
2. 在所有的群聚當中找出距離最為相近的兩個群聚 Ci、Cj。
3. 合併 Ci、Cj 成為一個新的群聚。
4. 若群聚的數目未達到我們要求的終止狀態，則重複步驟 2。

整個聚合的過程會如同圖 3.1 所示，最後會形成一株二元樹，一開始各筆資料皆是一個群聚，所以共有 10 個群聚，最初合併的群聚是編號 7 和 10 的資料。隨後是 6 和 9 的資料，以此類推，到最後整個過程收斂之後，所有的資料就完全聚合為一個群聚。在圖上 X 軸的值表示的是各筆資料的編號，Y 軸則是兩個群聚合併時的距離。

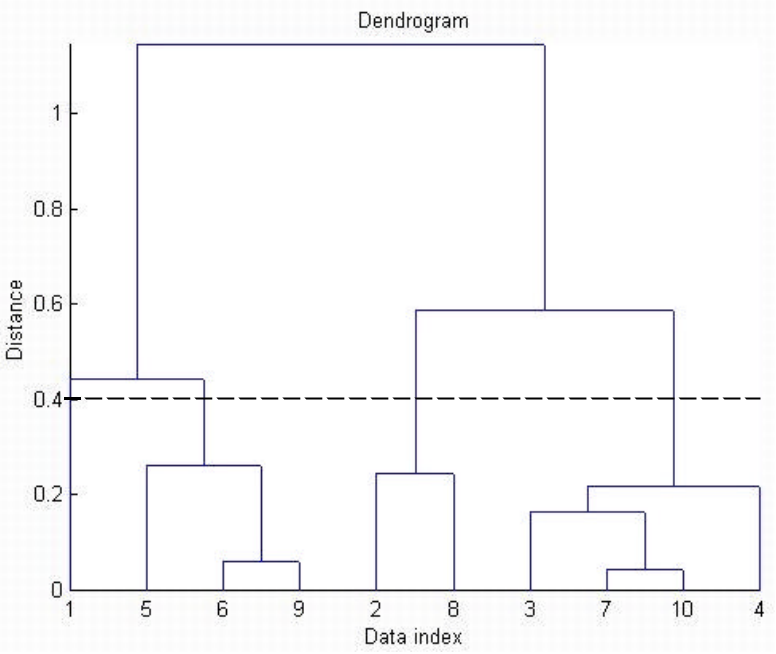


圖 3.1　**階層式分群法樹狀結構表示圖**

在訓練階段中，我們就是要找出對於分群最有助益的距離，在圖 3.1 **之中，若合併距離為** 0.4 **以內所形成的群聚最合於標準答案，那以距離** 0.4 **為分群門檻值（**clustering threshold**），也就是圖** 3.1 **當中的虛線，我們就能夠得到** 4 **個群聚，分別為**{1}、{5, 6, 9}、{2, 8}**和**{3, 7,10, 4}**。**

而在演算法的步驟 2 **當中，我們提到距離最為相近的兩個群聚，在計算距離之前，我們需要瞭解群聚之間距離的定義，對這些不同的定義我們說明如下：**

1. **「單一連結法」（**single-linkage method**），根據其定義，群聚** C_i **與** C_j **的距離為兩者之中最相近兩點間的距離，其表示法如下：**

$$D_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} d(a,b)$$

2. **「完整連結法」（**complete-linkage method**），根據其定義，群聚** C_i **與** C_j **的距離為兩者之中相距最遠的兩點間的距離，其表示法如下：**

$$D_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a,b)$$

3. 「平均連結法」(average-linkage method)，根據其定義，群聚 C_i 與 C_j 的距離為兩者當中各點與各點之間距離總和的平均，其表示法如下：

$$D_{\text{ave}}(C_{\text{i}},C_{\text{j}}) = \frac{1}{|C_{\text{i}}||C_{\text{j}}|} \sum_{a\in C_{\text{i}},b\in C_{\text{j}}} d(a,b)$$

4. 「沃德法」(Ward' s method)，根據其定義，群聚 C_i 與 Cj 的距離為兩者中各維度的變異數平方和，也就是說這個方法需要先針對先找出其兩個群聚的平均值，表示法如下：

$$D_{\text{ward}}(C_{\text{i}},C_{\text{j}}) = \sum_{a\in C_{\text{i}}\cup C_{\text{j}}} \left\| a - m_{\text{i,j}} \right\|^2$$

其中 $m_{\text{i,j}}$ 是 C_{i} 與 C_{j} 的平均值。

在這些距離的算法之中，本論文所採用的是「單一連結法」與「完整連結法」。

3.3 不需導式的最佳化方法

常見的不需導式之最佳化方法（derivative free optimization method）[10]有下列四種：基因演算法（genetic algorithm)、模擬退火法（simulated annealing)、隨機搜尋法 (random search algorithm) 和下坡式 simplex 搜尋法 (downhill simplex search)。這些方法都是利用反覆疊代的方式，逐漸找出函數中最佳化的值，雖然這樣的方法很有彈性，不論函數是連續性或是離散性，皆可利用此類方法來處理，不過這些方法的缺點就是疊代過程的運算量十分龐大，因此通常要花比較多的時間才能找出最佳化的值。

我們使用權重平均法計算新聞標題之間的相似度時，設定了一些可供調整的參數，由於我們沒有辦法根據目標函數的導數來對這些參數作最佳化，所以我們使用下坡式 simplex 搜尋法來調整參數，底下我們會詳細說明這個方法。

3.3.1 下坡式 simplex 搜尋法

在說明下坡式 simplex 搜尋法之前，我們要先對 simplex 做一些解釋，所謂的 simplex 就是在 n 維空間當中的 n+1 個點，例如在 2D 空間中的 simplex 是一個三角形，在 3D 空間中的 simplex 則是四面體。

因此在下坡式 simplex 搜尋法運作之前，必須先要找出 n+1 個點當成起始的 simplex。一般找起始點最簡單的方式為先隨機找出一點 P_0，則其他 n 點則可隨之設定為

$$P_i = P_0 + l_i e_i, i = 1, \ldots, n$$

其中 e_i 為 n 維空間基底（basis）中第 i 個單位向量（unit vector），而 l_i 為常數，此常數可以根據問題的特性訂定其大小。

對於點 P_i 上的函數值我們將其表示為 y_i，因此我們可以找出函數對應於 simplex 中 n+1 個點之值，並將其中的最大值與最小值表示為 y_h 與 y_l：

$$y_l = \min_i (y_i)$$
$$y_h = \max_i (y_i)$$

而這兩個點我們表示為 P_l 和 P_h。

我們定義 simplex 中這 n+1 個點的中心點（centroid）為 $\overline{P}$，在搜尋法執行的過程中，每一次的循環都是先求得 P_h 相對於 $\overline{P}$ 的反射點（reflection point）P^*。根據 P^* 的值，就可以採取下列四種步驟的其中之一來調整 simplex，使 simplex 可以更迅速地找到函數的最小值。這些步驟包括：

A. 由 P_h 反射

B. 由 P_h 反射且再行擴張

C. 沿著連結 P_h 與 $\overline{P}$ 的維度收縮

D. 將所有的維度往 P_l 的方向縮減

若是一個二維空間的函數，則我們可以將上述四種步驟表示如圖 3.2 所示：

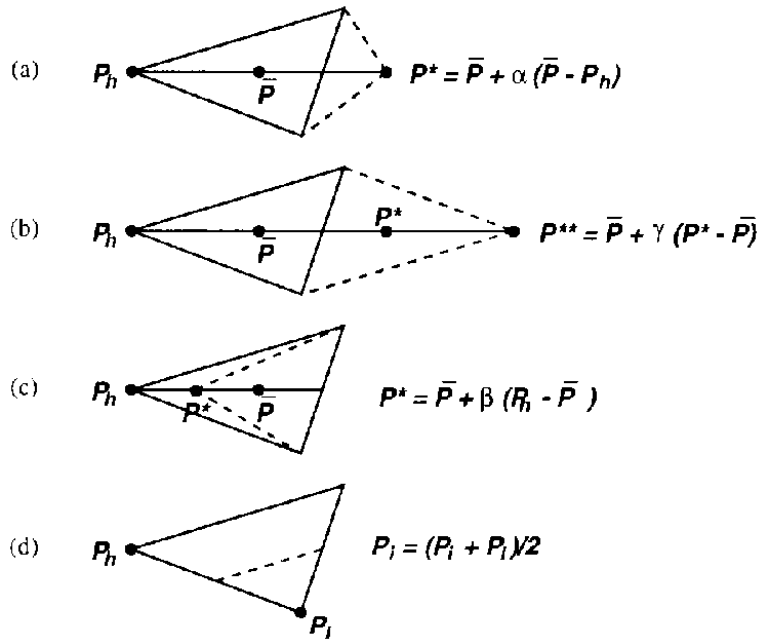


圖 3.2　下坡式 simplex 搜尋法任一循環執行後之結果

決定採取何種步驟之前，我們需要定義四種區間的範圍，以便於算出 P^* 之後，可以根據 P^* 所在的區間來決定使用的方法，而這些區間的定義如下：

- 區間 1：$\{y | y \le y_l\}$
- 區間 2：$\{y | y_l < y \le \max_{i,i\neq h}\{y_i\}\}$
- 區間 3：$\{y | \max_{i,i\neq h}\{y_i\} < y \le y_h\}$
- 區間 4：$\{y | y_h < y\}$

清楚區間的定義之後，就可以更深入地了解前述的四種步驟運作方式。

反射（Reflection）：我們定義反射點 P^* 及其值 y^* 為：

$$P^* = \overline{P} + \alpha(\overline{P} - P_h)$$
$$y^* = f(P^*)$$

α 為反射係數（reflection coefficient**），我們定義它為正數。因此可以得知 P^* 是位於 P_h 與 $\overline{P}$ 的連線上，且隔著 $\overline{P}$ 與 P^* 遙望。而隨後的步驟就可以根據根據 P^* 的函數值 y^* 來決定：**

1. **若 y^* 位於區間** 1**，進行擴張（**expansion**）步驟**
2. **若 y^* 位於區間** 2**，將 P_h 替換成 P^*，結束此次循環**
3. **若 y^* 位於區間** 3**，將 P_h 替換成 P^*，進行收縮（**contraction**）步驟**
4. **若 y^* 位於區間** 4**，進行收縮步驟**

擴張（Expansion）：我們定義擴張點 P^{**} 及其值 y^{**} 為：

$$P^{**} = \overline{P} + \gamma(P^* - P_h)$$
$$y^{**} = f(P^{**})$$

γ 為一大於單位向量的擴張係數（expansion coefficient**），若 y^{**} 位於區間** 1**，將 P_h 替換成 P^{**}，並結束此次循環；若 y^{**} 不是位於區間** 1**，將 P_h 替換為先前求得的 P^*，再結束此次循環。**

收縮（contraction）：我們定義收縮點 P^{**} 及其值 y^{**} 為：

$$P^{**} = \overline{P} + \beta(P_h - \overline{P})$$
$$y^{**} = f(P^{**})$$

β 為一介於 0 **與** 1 **之間的收縮係數（**contraction coefficient**），若 y^{**} 位於區間** 1**、**2 **或** 3**，將 P_h 替換成 P^{**}，並結束此次循環；若 y^{**} 位於區間**

4，則進行縮減（shrinkage）步驟。

縮減（shrinkage）：將所有的點 P_i 替換為 $(P_i + P_1)/2$ ，並結束此次循環。

前述中的係數都是在開始搜尋之前就定義好的，而整個搜尋過程就是上述的四個步驟反覆進行，直到搜尋的結果與我們所訂定的終止條件符合為止。

底下我們將整個搜尋的過程以圖像化的方式呈現出來，圖 3.3 **就是我們要搜尋最小值的函數平面，這是一個“** peaks **”函數。圖** 3.4 **為其等高線圖，我們嘗試從不同的起始點來找出最小值，搜尋的過程以圖中的不規則線段來表示，我們發現搜尋的結果與起始點有很大的關係，不過如果從同樣的起始點出發的話，得到的結果是不會改變的。因此我們瞭解下坡式** simplex **搜尋法很有可能只找尋到區域極小值（**local minimum**），要克服這個問題的話，可以試著將** a **、** g **和** b **這些係數改為隨機值，如此就有可能擴大整個搜尋的範圍，也較為容易跳脫區域極小值的範疇；除此之外，也可以多嘗試從不同的起始點來執行搜尋，如此也較易於找出何者才是真正的全域最小值（**global minimum**）。**

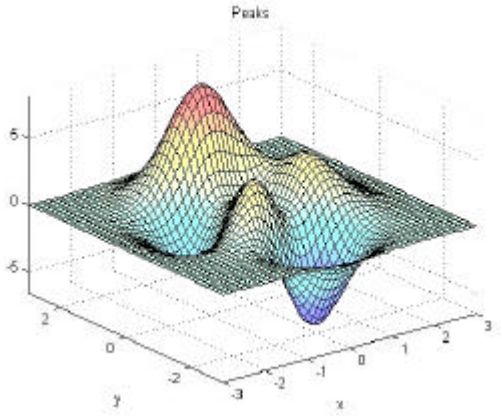


圖 3.3 “peaks”**函數平面**

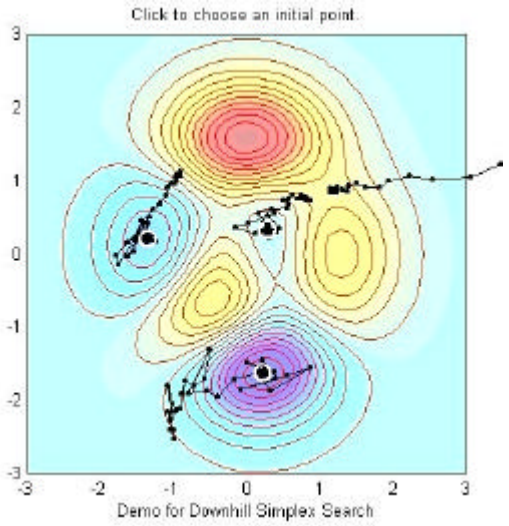


圖 3.4 **下坡式** simplex **搜尋法搜尋過程示意圖**

我們使用下坡式 simplex 搜尋法來為我們調整權重平均法的參數以及分群門檻值，目的是希望調整出對於分群最有幫助的參數。在參數訓練階段中，每當訓練出一組參數時，就會使用階層式聚合演算法來分群，隨後再以不同的分群門檻值（clustering threshold）來計算分群的結果，最後即可找出最恰當的參數與門檻值。

3.4 統計法

這個方法的精神是要利用大量資料統計的結果，找出能夠使得分群的錯誤率最低的門檻值。

實際運用上我們是針對各類別做統計，首先將類別當中的新聞標題兩兩比對，計算並記錄每一個配對的新聞標題相似度。由於我們可以從標準答案中得知哪些配對是屬於同一群聚之中的新聞標題，因此我們可將這些求得的相似度分為兩個集合：同一群聚與不同群聚。最後我們希望能夠找出兩個函數分別可以符合這兩個集合的分佈狀態，再根據這兩個函數就能夠求得門檻值，利用此門檻值就能夠取出最少量的錯誤配對。

相似度高於門檻值的配對我們都將視為屬於同一群聚，如果標準答案顯示此配對為不同群聚，這就是錯誤的配對；同樣地，相似度低於門檻值，但是在標準答案中卻屬於同一群聚的，也是錯誤的配對。

以圖 3.5 為例，假設這兩個集合當中配對的相似度分佈情形如圖中的曲線(a)與(b)所示，(a)是屬於同一群聚中配對的相似度形成之直方圖，而(b)則是不屬於同一群聚中的配對相似度形成之直方圖。我們可以發現以兩曲線的交點（圖上虛線）為其門檻值時，得到的錯誤配對數量將是最少的。

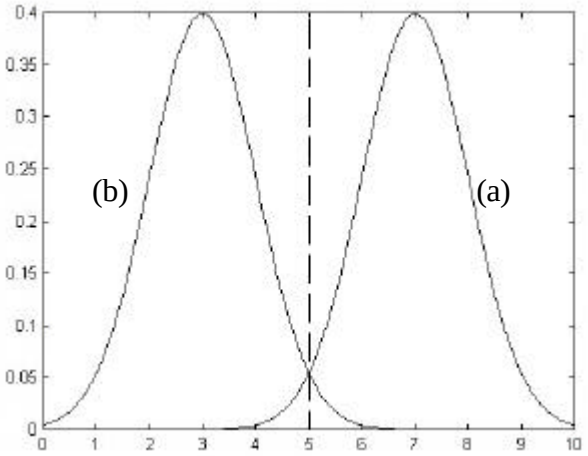


圖 3.5　新聞標題配對的相似度集合分佈模擬圖

這個方法與前述方法最大的不同點，在於訓練階段中統計法無須經由反覆調整，而能夠直接求得門檻值。且系統可以根據需要來調整門檻值的位置，如果希望能夠盡量減少原本不屬於同一群聚，但是卻被分為同一群聚的情形，則可以在計算這一類配對的分數時，提高其扣分的比重，則門檻值就會往右移；反之，若是覺得這樣反而使得原本正確的分群結果也被裁掉，就可以針對原本屬於同一群聚，但是卻被分為不同群聚的配對，提高其扣分比重，則門檻值將會往左移動。

這個優點可以讓系統根據其本身需求來調整各種配對所佔的比重，希望分群結果能夠讓往後的人工處理更為方便迅速。

第 4 章 實驗方法

4.1 實驗流程簡介

在前述的各項方法當中，我們也無法得知哪些方法結合起來可以對新聞標題分群最有效果，因此需要進行多次的實驗以便於測試與比較，而我們實驗的流程如圖 4.1 所示，流程中各步驟的詳細做法將會在下面各節說明。

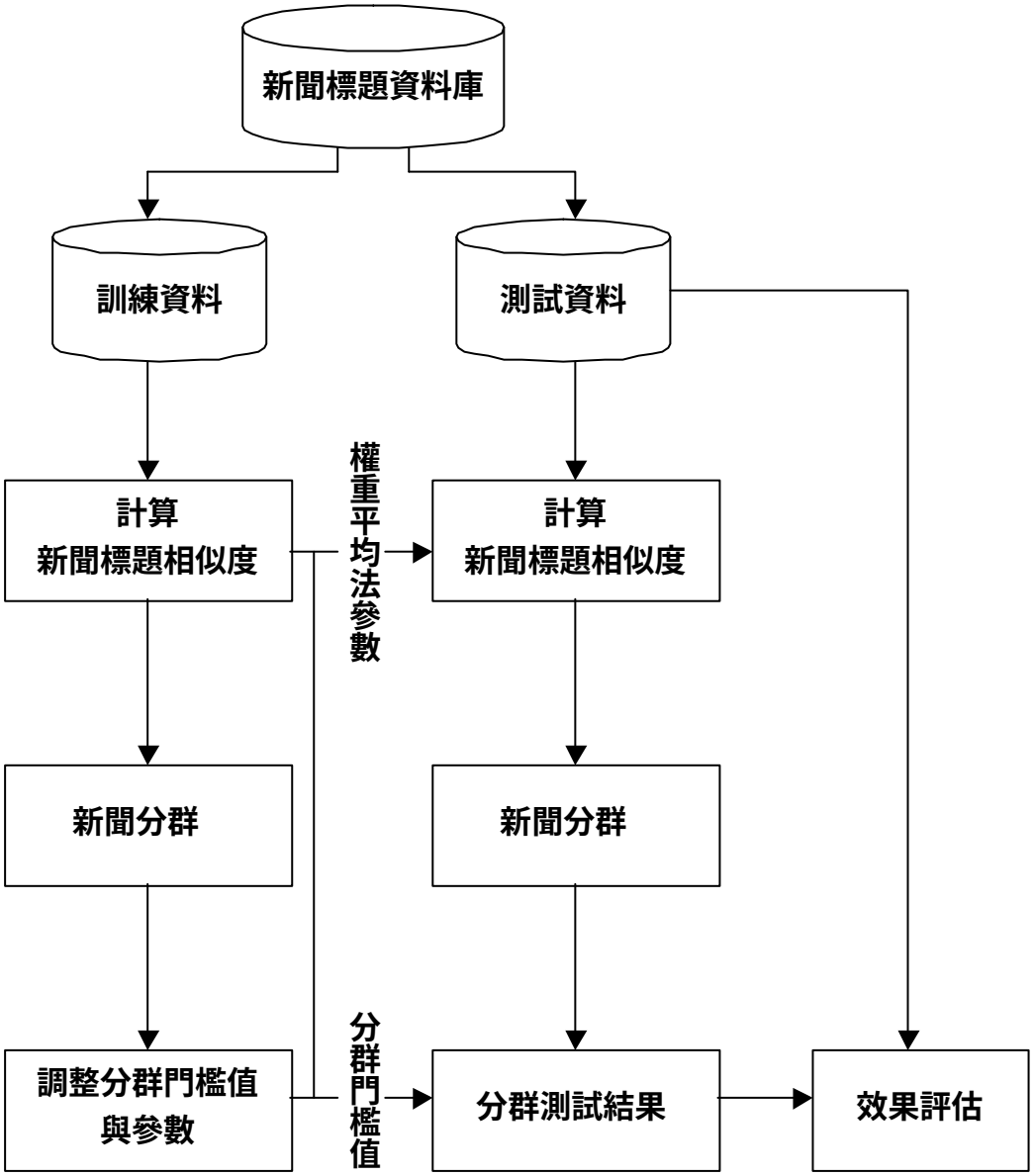


圖 4.1　實驗流程圖

4.2 訓練資料之擷取

網羅新聞中的新聞群聚是每天處理當天的新聞成為新聞群聚，為了要模擬實際的流程，所以實驗中我們也是取出一天的新聞標題作為測試資料，再利用其前幾天新聞群聚的資訊作為訓練資料。

而我們首先遇到的課題就是該以多少天的分群資料做為實驗的訓練資料，因此我們簡單的做了一項統計，我們以資料庫中標準答案的分群資料為基準，再使用不同的新聞標題相似度計算方法與不同的新聞類別來統計，得到每一天該類別最恰當的分群門檻值，我們隨機取了兩個類別：「政治」與「財經」。採用的相似度計算法則是「*LCS* 與 *LCCS* 的平均值」和「歐幾里德距離」。所得到的結果如圖 4.2 所示，圖中的 X 軸是天數，Y 軸則是對應的分群門檻值，我們發現不管相似度的計算方式與類別為何，所得到的數據都是隨著時間而有劇烈變化。這樣的數據顯示出新聞具有很強的時效性，除非新聞事件很重要，否則該事件在新聞報導中的重要性都會隨著時間而遞減，逐漸被別的新聞事件取代掉，因此我們在取用訓練資料的部份就以短期的新聞分群資訊為主。

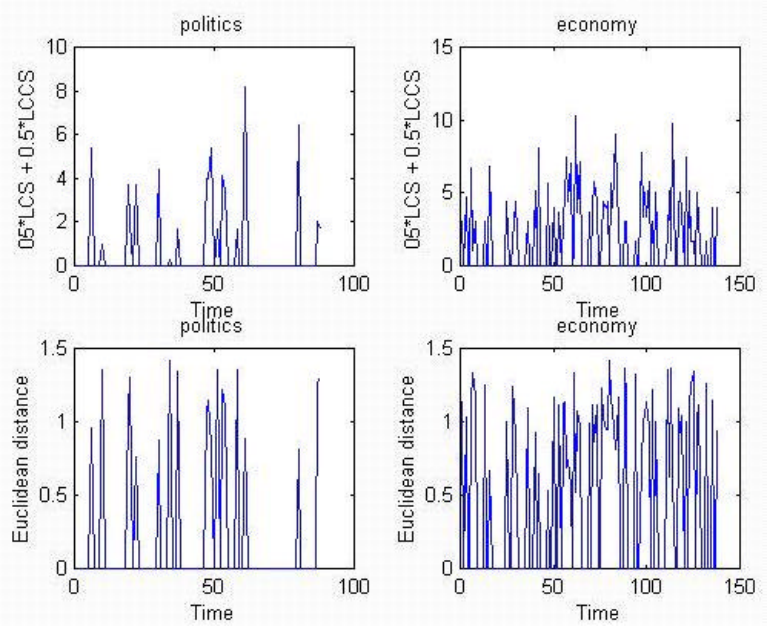


圖 4.2　分群門檻值與時間之關係圖

接著我們對這兩種類別做簡單的實驗，我們根據不同的訓練天數來求得其適當的門檻值，再將求得的門檻值與實際答案比對，我們想要找出門檻值與標準答案最為相近的訓練天數，如此就更能有效提升新聞分群的準確度。由於我們的目標是短期的訓練資料，所以在實驗當中使用的訓練天數上限訂為 10 天，圖 4.3 即是針對不同的訓練天數，所得到的門檻值與正確答案的誤差值。

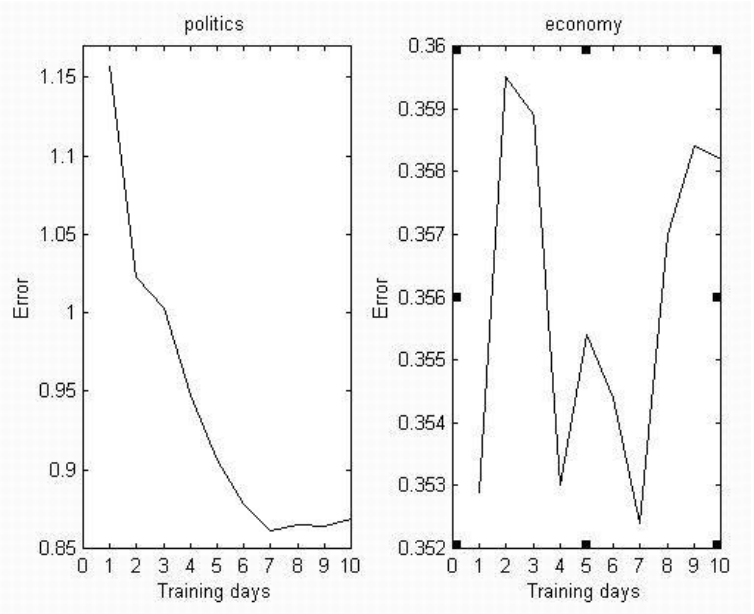


圖 4.3　訓練天數與分群門檻誤差值之關係圖

我們會發現兩者都是在訓練時間為七天的時候，表現得最好，因此我們就使用七天做為我們擷取訓練資料的天數。

4.3 計算新聞標題相似度

在第三章我們提出一些關於字串相似度的度量方法，而在實際的運用上，我們也在實驗中直接利用了「歐幾里德距離」與「權重平均法」來計算新聞標題之間的相似度，但是我們在實驗中並未有單獨使用「*LCS*」與「*LCCS*」，而是將這兩者以某種比例來結合在一起，成為一種相似度的度量。因為我們無法得知在分群過程當中，「*LCS*」與「*LCCS*」 何者對於分群的幫助較高，因此我們在實驗

一與實驗二當中利用兩者平均值作為相似度。

而在實驗三與實驗四則是使用「歐幾里德距離」作為新聞標題相似度的度量方法時，就如同 3.1.3 所提到的，我們先統計 5401 個中文字在每一則新聞標題當中出現的頻率，再將新聞標題轉換為一個 5401 個維度的向量，每一個維度的值即為對應該維度的中文字出現的頻率。而兩個向量之間的「歐幾里德距離」就是兩則新聞標題的相似度。

「權重平均法」則是在實驗五與實驗六做為相似度的度量方法，此方法在訓練階段和測試階段中的操作略微不同。在訓練階段中，我們還需要輔以「下坡式 simplex 搜尋法」來調整公式中的參數，以求得最能符合資料中的分群結果的參數。而在測試資料當中，就省略了調整參數的這個部份，而是直接利用訓練階段所求得的參數代入公式中，求得新聞標題相似度。

4.4 新聞分群

在實驗中使用的分群法是 3.2.1 所提到的「階層式聚合演算法」，也就是一開始將各新聞標題視為單一群聚，隨後逐漸將最接近的兩個群聚合而為一，直到全部的新聞標題聚合成為一個群聚為止。

而針對於不同的群聚之間距離的計算方式，我們採用的是「單一連結法」與「完整連結法」。在實驗中，針對每一種相似度的度量方法，我們會分別利用這兩種演算法來做分群，藉以比較何者較適用於系統之中。至於「沃德法」，由於在計算群聚之間的距離時，需要先算出兩群聚的平均值，但是我們並無法有效地定義新聞群聚之間平均值的計算方式，因此目前我們未採用這個方法。

4.5 計算新聞群聚相似度

為了在調整參數時以及往後比較實驗結果能夠有一個量化的值可以參考，於是我們希望找出分群結果與標準答案之間的相似度。但是在計算之前我們要先解決的是如何將將程式產生的群聚與人工的標準答案配對的問題，由於每天的新聞群聚並沒有固定群數，所以很可能程式產生的群聚數有 N_1 個，但是在標準答案中卻有 N_2 個群聚，而 N_1 大於或小於 N_2 都有可能，因此計算相似度之前要先將程式產生的群聚與標準答案的群聚配對，之後才能計算群聚之間的相似度。

在配對的問題上我們所採用的是分派問題（assignment problem）[2]的解法，舉例來說，假設程式所得到的群聚數目有 2 群，而標準答案有 3 群，則我們就先產生一個 2x3 的矩陣，矩陣中的值就是相對應的兩個群聚之相似度，結果如圖 4.4(a)所示，矩陣中的行代表標準答案中的群聚內所包含的文章編號，列則是代表程式產生的群聚內包含的文章編號。接著如圖 4.4(b)所示，我們找出矩陣中的最大值，則處於該值所在的行與列上的群聚就是互相對應的兩個群聚，找出對應的群聚後，在往後的對應過程中就不再考慮處於該列與該行之中的所有群聚，如同圖中虛線表示，這些數值在往後的步驟中將被忽略掉。重複以上所述步驟，如果 N_1 小於 N_2，則 N_1 當中的所有群聚皆可在標準答案中找到對應的群聚；反之，若 N_1 大於 N_2，則程式產生的群聚會有找不到對應的群聚之情形，在此我們就將這一些找不到對應的群聚忽略掉。圖 4.4(d)即為對應完之後的結果。

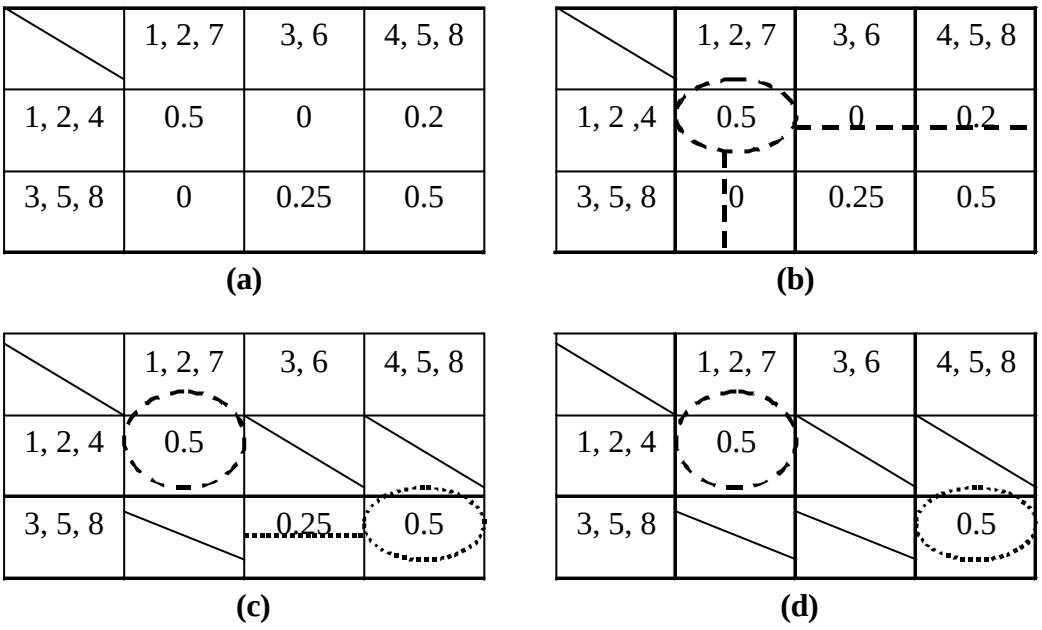


圖 4.4 **群聚相似度計算過程** (a)**初始矩陣，存放各群聚兩兩之間的相似度** (b)**對應程式產生的群聚一** (c)**對應程式產生的群聚二** (d)**對應結果**

而關於計算對應群聚之間的相似度，我們比較了兩種方法，分別是：

1. **集合相似度（**set similarity**）[7]：此方法常被使用在比較兩個集合之間的相似度，假設有兩集合** A **與** B**，則兩者之間的相似度為** $\frac{|A \cap B|}{|A \cup B|}$ **，例如** A = {1, 3, 4}**，** B = {1, 2, 4, 5}**，則** $A \cap B = \{1,4\}$ **，** $A \cup B = \{1,2,3,4,5\}$ **，因此** A **與** B **的相似度就是** $\frac{2}{5} = 0.4$**。**
2. F-Measure**：使用這一項計算公式之前，我們必須要先求出兩個值，精確率（**precision**）跟召回率（**recall**），我們以圖** 4.5 **來說明這兩個值的計算方式。**

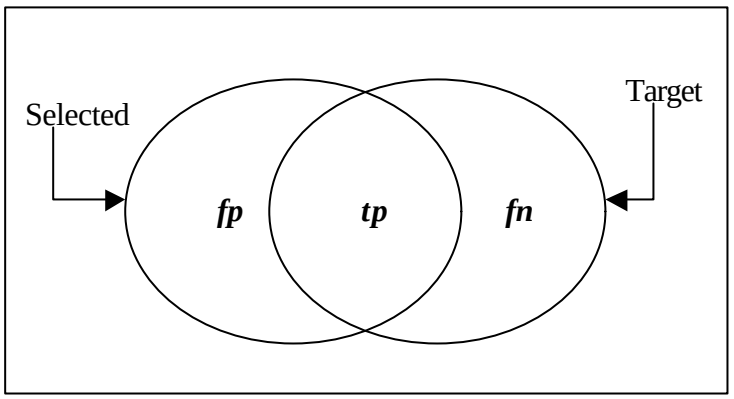


圖 4.5　**精確率與召回率之關係圖**

圖中的 Selected **與** Target **是互相對應的兩個新聞群聚，**Selected **部分是程式產生的新聞群聚所包含的文章，**Target **部分是正確答案中的新聞群聚包含的文章。*tp* 則是程式產生的群聚中與正確答案完全相同的文章數，*fp* 是程式產生的群聚中不存在於正確答案的文章數，*fn* 則是不包含在執行結果中的正確答案之文章數。**

底下我們以 P **代表精確率，以** R **代表召回率，因此** $P = \frac{tp}{fp + tp}$，$R = \frac{tp}{tp + fn}$，**而** $F - Measure = \frac{2PR}{P + R}$。

從公式中可以發現在 F-Measure **中，需要精確率與召回率都很高才能得到較高的值，任何一者都不能偏廢，所以我們利用這樣的一個方法來取得精確率與召回率的平衡。**

有了群聚對應的方法以及相似度的計算方式，我們就可以算出對應群聚之間的相似度，我們可以將這一些相似度做一個平均，就可以代表程式所產生的所有群聚與正確答案之間的相似度，公式表示如下：

$$SIM(L) = \frac{1}{N_1} \sum_{i=1}^{N_2} \max_{0 < j \le N_2} \{similarity(i, j)\}$$

SIM(*L*)：二元樹中階層 *L* 的分群結果與標準答案的相似度
N_1：程式產生的分群結果之群聚數目
N_2：標準答案的群聚數目
similarity(*i*, *j*)：程式產生的群聚 *i* 與正確答案之群聚 *j* 的相似度

SIM(*L*)是一個較為整體的相似度，在取得門檻值以及調整參數的時候就可以用來當成我們參考的標準。在最後評估結果時，我們也是用這個方法求得系統產生的群聚與正確答案的相似度。

在調整參數的過程中，有「集合相似度」或「F-Measure」兩種相似度的計算方式可供使用，為了比較何者能夠取得較好的結果，因此我們隨機找了連續十天的新聞標題做評估，分別使用這兩種相似度的計算方式來調整參數，再利用這些參數來取得分群結果，最後比對兩種結果，希望藉此可以找出表現較好的方法。

在這兩種分群結果中，雖然是以不同的方法來調整參數，但是還是該以相同的計算方法來比較結果時，所以我們最後是以一致的相似度計算方法來求得 *SIM*(*L*)。例如在調整參數方面不論是使用「F-Measure」或「集合相似度」，但是最後在比對分群結果的相似度時，我們是統一採取「集合相似度」的計算方式，並不以其調整參數的方式為依歸。我們將比較的結果列於表格 4.1 中，表格中的每一列是以不同的相似度計算方式來調整參數所得到的數據，而每行則是以不同的評估方法所得到的結果。

	F-Measure	精確率	召回率	集合相似度
以「F-Measure」調整參數	0.6901	0.7354	0.6501	0.6383
以「集合相似度」調整參數	0.6895	0.7340	0.6501	0.6380

表格 4.1　比較以「集合相似度」與「F-measure」分別調整參數之結果

從比較中我們可以發現兩者所得的結果其實相差不大，但是「F-measure」略優於「集合相似度」，推究其原因，若根據圖 4.4 來看，使用「集合相似度」時，所考慮到的只是 *tp* 與 *tp+fp+fn* 之間的比例大小，但是使用「F-measure」就考慮得比較全面，因此使用「F-Measure」會有比較好的表現，所以我們在實驗當中即以「F-Measure」為調整參數的依據。

4.6 調整參數

調整參數這個步驟只在訓練階段當中使用，而此處我們要調整的參數有兩個：

- 分群門檻值：我們希望以這個數為指標，在測試階段的分群過程中，當相似度小於該值的時候，就不再進行分群的動作，而當時的分群結果即為我們想要的新聞群聚。在訓練階段中調整門檻值的過程就像圖 4.3 所示，首先我們要將所有的新聞逐漸聚合為一個大群聚，之後則以圖中 Y 軸上相似度的上下限為範圍，試著調整門檻值，門檻值一經調整，則藉由此門檻值所得的分群結果就會與訓練資料的標準答案相互比對，而在整個調整過程當中，能夠取得與標準答案最為接近的分群結果的門檻值，就是在測試階段當中需要的一項參數。

  例如在圖 4.6 當中，若是以 0.4 為門檻（下方的虛線），則我們會得到四個新聞群聚，分別是{1}、{5, 6, 9}、{2, 8}和{3, 7, 10, 4}，若是以 0.7 為門檻（上方的虛線），則分群的結果就是{1, 5, 6, 9}和{2, 8, 3, 7, 10, 4}，假設經過比較之後，我們發現門檻值取為 0.4 與標準答案比較相近，則門檻值將被調整為 0.4。

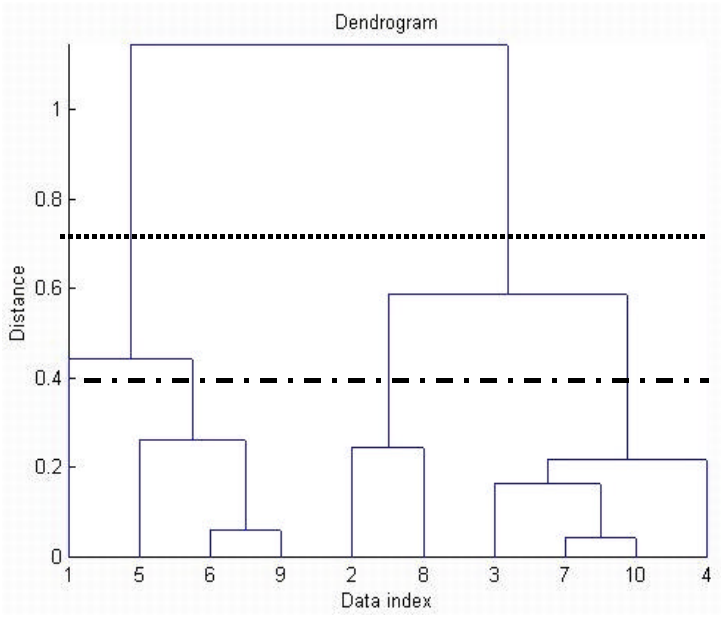


圖 4.6　取得分群門檻值

- **權重平均法參數：在訓練階段當中，若我們在計算新聞標題相似度的方式上是採用「權重平均法」，則我們除了要調整分群門檻值之外，還需要調整函數當中的參數（k_1、k_2、k_3 和 k_4），我們利用 3.3.1 當中提到的「下坡式 simplex 搜尋法」來調整這一些參數值。在調整過程中，每當權重平均法的參數作了些微調整，我們就需要重新計算新聞標題相似度，重新分群，重新找出相對應於此分群結果的最佳門檻值，最後算出分群結果與標準答案的相似度，而我們調整的目標函數就是分群之後的結果與標準答案的相似度，希望能夠調整出最接近標準答案的結果。因此我們要保留的是表現最好的參數以及對應的分群門檻值。**

4.7 統計法之實作

當我們統計了大量的資料之後，發現要找出符合整個分佈情形的函數不是一件容易的事，因為整個分佈並非是平滑曲線的狀態，例如我們以「2000 年總統大選」這個類別為例，其統計結果如圖 4.7 所示，上圖記錄的是在同一個群聚之中的配對，而下圖則是不屬於同一群聚的配對，X 軸代表的是這一些配對的相似度，而 Y 軸則是配對的數量。

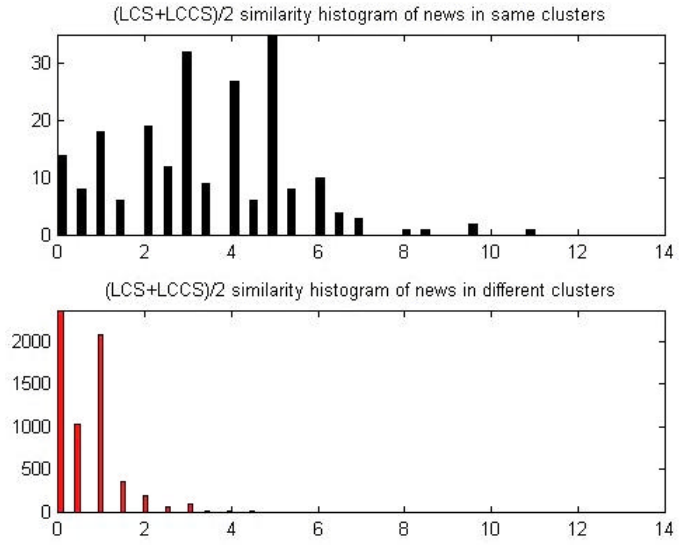


圖 4.7　「2000 年總統大選」類別中新聞標題相似度分佈情形

既然無法馬上找出符合分佈狀態的曲線，我們就改以「窮舉法」來找出最佳的參數以及分群門檻值，我們列出各種可能的參數組合，分別代入公式之中，求得兩個集合的分佈之後，再以 X 軸上所有的數值做為門檻值來測試，希望找出能夠產生最少錯誤配對的門檻值。

上述的參數是用來結合不同的距離度量方法，例如我們可以計算當 *LCS* 與 *LCCS* 混合使用時，兩者的比重該如何調整才能有效地降低錯誤的分群結果；同樣地，我們也可以用相同的方法將 *LCS*、*LCCS* 與歐幾里德距離三者結合使用。

第 5 章 實驗結果

由於我們是分別針對 86 個新聞類別來作新聞群聚，數量繁多，因此在往後列出各類別最後所得到的相似度時，我們只列出 10 個類別作為參照，這 10 個類別是測試結果與標準答案比較之後，F-Measure 最高的前 5 名與最後 5 名。

另外我們列出的還有所有類別的 *SIM*(*L*)之總平均值，由於有一些類別的出現天數較少，因此在平均值的計算上，我們也會根據各類別所出現的天數作為加權的標準。

• 實驗一

在實驗一中，我們利用 *LCS* 與 *LCCS* 的平均值作為新聞標題之間的相似度，而採用的分群法是階層式聚合演算法中的「完整連結法」。所得結果如表格 5.1 與表格 5.2。

總平均值：

F-Measure	精確率	召回率
0.5008	0.5239	0.4797

表格 5.1　實驗一分群結果之總平均值

個別類別結果：

類別名稱	F-Measure	精確率	召回率
大專運動會	1	1	1
評論	0.9155	0.9155	0.9155
評論（大陸）	0.8792	0.8792	0.8792
保齡球	0.8774	0.8824	0.8725
財經	0.8393	0.8377	0.8409
…	…	…	…
大陸	0.2575	0.2624	0.2528
排球	0.25	0.25	0.25
電玩	0.2454	0.2405	0.2505
旅遊	0.2410	0.2506	0.2321
科技投資	0.0768	0.0838	0.0708

表格 5.2　實驗一各類別之分群結果

- **實驗二**

在實驗二中，我們利用 *LCS* 與 *LCCS* 的平均值作為新聞標題之間的相似度，而採用的分群法是階層式聚合演算法中的「單一連結法」。所得結果如表格 5.3 **與表格** 5.4**。**

總平均值：

F-Measure	**精確率**	**召回率**
0.5371	0.5327	0.5416

表格 5.3　**實驗二分群結果之總平均值**

個別類別結果：

類別名稱	F-Measure	**精確率**	**召回率**
大專運動會	1	1	1
評論	0.9061	0.9061	0.9061
評論（大陸）	0.8792	0.8792	0.8792
保齡球	0.8774	0.8824	0.8725
娛樂	0.8083	0.8054	0.8122
…	**…**	**…**	**…**
大陸	0.3113	0.2795	0.3513
電玩	0.2948	0.2601	0.3401
旅遊	0.2745	0.2554	0.2968
排球	0.25	0.25	0.25
科技投資	0.1709	0.1173	0.3194

表格 5.4　**實驗二各類別之分群結果**

將此結果與「實驗一」比較，我們會發現「實驗二」整體的表現優於「實驗一」的數據，但是從各類別的表現來觀察，會看到在「實驗一」中表現很好的類別數據有下降的趨勢，但是表現比較差的類別在數據上卻有提升，兩相抵銷之下，提升的趨勢較大，因此也改善了整體的表現。

- **實驗三**

在實驗三中，我們利用「歐幾里德距離」作為新聞標題之間的相似度，而採用的分群法是階層式聚合演算法中的「完整連結法」。所得結果如表格 5.5 **與表格** 5.6**。**

總平均值：

F-Measure	**精確率**	**召回率**
0.6427	0.6951	0.5977

表格 5.5　**實驗三分群結果之總平均值**

個別類別結果：

類別名稱	F-Measure	**精確率**	**召回率**
評論	0.9765	0.9765	0.9765
評論（大陸）	0.9758	0.9758	0.9758
財經	0.9318	0.9318	0.9318
保齡球	0.8774	0.8824	0.8725
房地產	0.8410	0.8410	0.8410
…	…	…	…
企業	0.4688	0.5703	0.3979
硬體	0.4557	0.4909	0.4252
生活	0.4399	0.5451	0.3688
旅遊	0.4375	0.4952	0.3918
排球	0.25	0.25	0.25

表格 5.6　**實驗三各類別之分群結果**

我們發現「實驗三」的數據較前兩個實驗都有明顯的提升，雖然有一些類別，例如「大專運動會」的召回率下降，但其精確度仍然是 1**，與前兩個實驗的數據相同，不過也造成了** F-Measure **值的下降，導致排名往後退。**

- **實驗四**

在實驗四中，我們利用「歐幾里德距離」作為新聞標題之間的相似度，而採用的分群法是階層式聚合演算法中的「單一連結法」。所得結果如表格 5.7 **與表格** 5.8**。**

總平均值：

F-Measure	**精確率**	**召回率**
0.6630	0.7080	0.6233

表格 5.7　**實驗四分群結果之總平均值**

個別類別結果：

類別名稱	F-Measure	**精確率**	**召回率**
評論	0.9765	0.9765	0.9765
評論（大陸）	0.9758	0.9758	0.9758
財經	0.9422	0.9432	0.9413
保齡球	0.8774	0.8824	0.8725
房地產	0.8410	0.8410	0.8410
…	…	…	…
企業	0.4943	0.5924	0.4241
硬體	0.4741	0.5097	0.4431
生活	0.4614	0.5442	0.4004
旅遊	0.4424	0.5004	0.3964
排球	0.25	0.25	0.25

表格 5.8　**實驗四各類別之分群結果**

將「實驗四」與「實驗三」互相比較之後，我們發現「實驗四」的數據略高於後者，尤其是在「實驗三」當中表現較差者，在這次的實驗中數據都有提升，反而是原先排名在前的類別，表現都是持平。

- **實驗五**

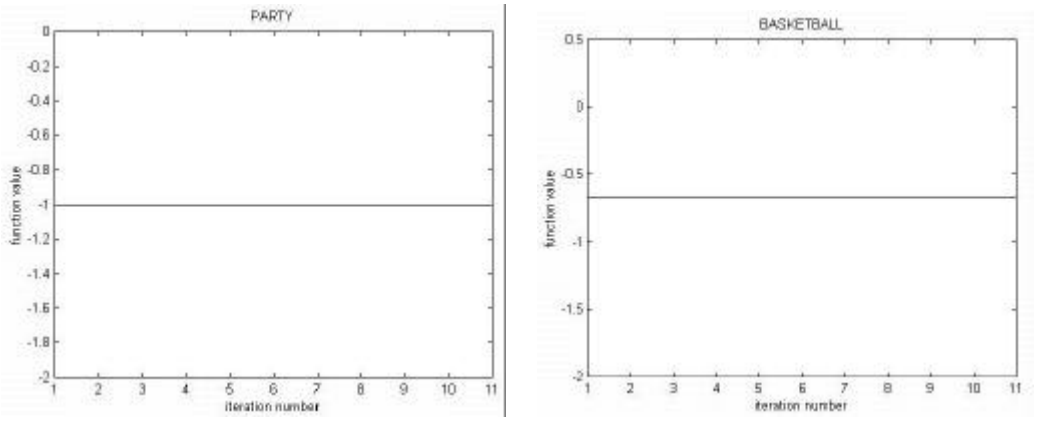
實驗五的新聞標題相似度是將 *LCS*、*LCCS* **與「歐幾里德距離」合併使用，合併的公式就是** 3.1.4 **所提出的「權重平均法」，再利用「下坡式** simplex **搜尋法」來調整公式中的參數值，採用的分群法是階層式聚合演算法中的「完整連結法」。但是由於使用「下坡式** simplex **搜尋法」調整參數時，整個過程會十分地耗時，而我們又沒有足夠的時間來完成所有的類別，因此我們從所有的類別當中挑出兩個類別來做實驗。若這兩個類別能夠有令人滿意的結果，就表示本實驗所嘗試的方法應是可行的。**

我們所使用的兩個類別是「黨派」與「籃球」，這兩個類別在前四個實驗當中的表現屬於中等，數據如表格 5.9 **所示。**

	黨派			**籃球**		
	F-Measure	**精確率**	**召回率**	F-Measure	**精確率**	**召回率**
實驗一	0.4503	0.4733	0.4294	0.5673	0.6350	0.5126
實驗二	0.5314	0.4950	0.5735	0.6158	0.6545	0.5814
實驗三	0.5381	0.6547	0.4567	0.6077	0.7074	0.5327
實驗四	0.5605	0.6611	0.4864	0.6613	0.7375	0.5994

表格 5.9 **「黨派」與「籃球」兩類別在實驗一至四之分群結果**

在調整參數的過程中，我們試著畫出整個函數值的變化幅度，若函數值有越來越小的趨勢，就表示整個調整的過程發揮了作用，反之，就表示我們需要重新從另一個起始點來找尋函數的最小值了。函數的變化趨勢如圖 5.1 **所示：**



(a) **(b)**

圖 5.1　**利用「下坡式** simplex **搜尋法」調整參數過程中目標函數值變化圖**
(a)**新聞類別：黨派**　(b)**新聞類別：籃球**

從圖中可以看出不論搜尋法如何改變參數大小，最後的函數值都不受到影響。而以調整過的參數以及門檻值來做測試之後，得到的結果如表格 5.10**。**

	F-Measure	**精確率**	**召回率**
黨派	0.4486	0.4420	0.4553
籃球	0.5584	0.6163	0.5105

表格 5.10　**實驗五之分群結果**

「實驗五」所得到的結果較之前四個實驗都差，而從先前的函數變化圖來看，我們可以得之主因是在於「下坡式 simplex **搜尋法」沒有發揮功效。結果就造成整個分群的結果與我們起始設定的參數值極為相關，也就是說若初始的參數值設得好，則分群的表現也會很優秀；反之，若初始的參數設得差，則不論如何調整參數，效果都無法改善。**

- **實驗六**

實驗六與實驗五的條件大致相同，新聞標題相似度也是以 3.1.4 **所提出的「權重平均法」來計算，輔以「下坡式** simplex **搜尋法」來調整公式中的參數，不同的是採用的分群法是階層式聚合演算法中的「單一連結法」。而用來測試的類別也是「黨派」與「籃球」。測試後的分群結果如表格** 5.11**。**

	F-Measure	**精確率**	**召回率**
黨派	0.5209	0.4623	0.5964
籃球	0.5996	0.6215	0.5792

表格 5.11　**實驗六之分群結果**

將「實驗六」的結果與「實驗五」比較，會發現「實驗六」的表現較好，但是若觀察調整參數的過程中，函數值的變化情形仍然與圖 5.1 **的結果相同，因此本實驗的結果也是與初始的參數值極為相關。而從「實驗五」與「實驗六」來看，「下坡式** simplex **搜尋法」完全沒有發揮我們預想的功能，使得這兩個實驗並未由於將** *LCS*、*LCCS* **與「歐幾里德距離」結合而提升效果。**

- **實驗七**

在實驗七當中，訓練階段採用的是「統計法」，我們分別統計屬於同一群聚與不同群聚間新聞標題兩兩的相似度，藉由其分佈情形來找出能夠得到最少錯誤配對的分群門檻值，再將此門檻值代入測試階段中使用。

除了找出分群門檻值外，在計算新聞標題相似度方面我們結合了 *LCS* **與** *LCCS* **做為度量方法，並且加入參數來調整這些字串距離在整個度量中所佔的比例，以下為本實驗所使用的公式：**

$$S_{i,j} = k_1 * LCS + k_2 * LCCS \text{ ， where } k_1 + k_2 = 1$$
$$k_1, k_2 > 0$$

而實驗中用來測試的類別是「黨派」與「籃球」。測試後的分群結果如表格 5.12**。**

類別	分群法	F-measure	精確率	召回率
黨派	完整	0.4520	0.5142	0.4032
	單一	0.5356	0.5403	0.5309
籃球	完整	0.5473	0.6072	0.4982
	單一	0.5930	0.6157	0.5719

表格 5.12　**實驗七之分群結果**

若將此結果與條件相似的「實驗一」與「實驗二」相比較，會發現在表現上互有優劣。在「黨派」類別部分表現方面，「實驗七」的表現比較好，但是「籃球」類別方面是「實驗二」的表現較好，所以很難得到何種方式在分群工作上一定能夠有較佳表現的結論。

- **實驗八**

本實驗的訓練階段採用的也是「統計法」，我們分別統計屬於同一群聚與不同群聚間新聞標題兩兩的相似度，藉由其分佈情形來找出能夠得到最少錯誤配對的分群門檻值，再將此門檻值代入測試階段中使用。

除了找出分群門檻值外，在計算新聞標題相似度方面結合了 *LCS*、*LCCS* **與「歐幾里德距離」做為度量方法，並且加入參數來調整這些字串距離在整個度量中所佔的比例，以下為本實驗所使用的公式：**

$$S_{i,j} = k_1 * LCS + k_2 * LCCS + \frac{k_3}{Euc_{i,j} + 1},\ \text{where } k_1 + k_2 + k_3 = 1$$

$$k_1, k_2, k_3 > 0$$

而實驗中用來測試的類別是「黨派」與「籃球」。測試後的分群結果如表格 5.13**。**

類別	分群法	F-measure	精確率	召回率
黨派	完整	0.5008	0.5620	0.4516
	單一	0.5684	0.5762	0.5609
籃球	完整	0.5344	0.6164	0.4716
	單一	0.5549	0.6041	0.5131

表格 5.13　**實驗八之分群結果**

將「實驗八」與「實驗七」比較，在「黨派」類別部分，本實驗有較佳的表現，但是換到「籃球」類別，則「實驗七」表現較好。雖然在實驗一到四中，使用「歐幾里德距離」做為度量方法有最佳的結果，但是並不保證將其與 *LCS*、*LCCS* **結合使用會有更好的效果，可能還是需要是個別類別而定。**

第 6 章 討論與未來研究方向

6.1 討論

在我們所做的實驗當中最好的分群結果，是以新聞標題之間的「歐幾里德距離」為其相似度，若只有使用 *LCS* 和 *LCCS* 的話，由於部分關鍵詞的出現次序不同，就會失去一部份的資訊，例如我們在 3.1.3 所舉的例子，或是「美中關係」與「中美關係」，利用 *LCS* 或 *LCCS* 都只能找到三個相同的字，無法如同使用「歐幾里德距離」一般能夠完全涵蓋相同的字。當然這個方法也會有缺點產生，例如「國中」與「中國」兩個不相關的詞彙，若使用「歐幾里德距離」來計算的話，兩者會被視為相同的詞彙。雖然有此缺陷，但是這對於分群的結果影響卻極為有限，因此在計算新聞標題相似度上使用「歐幾里德距離」會具有比較好的結果。

比較了不同的分群演算法的分群結果之後，我們發現「單一連結法」比「完整連結法」更適用於我們的系統之中。在討論分群法的論文中，多數都提及「單一連結法」容易造成各群聚之間「大者恆大，小者恆小」的現象，因此所得到的分群結果往往不是很好。但是在我們的實驗中結果卻與這一個說法大相逕庭，這是由於新聞群聚的特性使然，因為在每個新聞類別之中的新聞群聚並非是平均分佈的，其中只有部分新聞需要形成新聞群聚，這與「單一連結法」所產生的群聚分佈類似，若是使用「完整連結法」，雖然會使得所有的文章會很平均地分佈在各群聚之中，但是卻與真實的情況不能相容，所以在系統中使用「單一連結法」會得到比較好的分群結果。

針對於「下坡式 simplex 搜尋法」並未有效地調整參數的原因，我們認為是由於這個方法調整參數的幅度過小，以致於雖然有調整參數，但是卻不會影響到分群法所造出的二元樹結構。而我們的目標函數就是在二元樹當中找出一個恰當的門檻值，使得分群結果的相似度為最高，而由於二元樹結構未曾改變，因此造成目標函數的值維持不變。對於這項缺失，我們嘗試增加整個參數調整的幅度，

但是卻發現這樣的動作使得整個程式難以收斂，因而在訓練階段就會耗費了更多的時間，因此我們以「統計法」來試著完成調整參數的動作。

但是利用「統計法」得到的結果也不如預期地好，我們發覺這是由於「統計法」與實驗一到六當中訓練階段所使用的方法兩者目標不同所致。之前訓練階段的方法是找出能夠得到最高 $SIM(L)$的分群門檻值，而「統計法」的目標則是找出最少錯誤配對的分群門檻值，但是計算最後的結果都是利用 $SIM(L)$來評比，所以「統計法」在評比之後並未能夠完全表現出其優勢。而使用「統計法」做為訓練方法時，使用的相似度完全是新聞標題之間的相似度，但在「測試階段」中使用「階層式分群法」時，其相似度卻有可能是群聚之間的距離，或許是我們先前忽略了此項因素的影響，所以「統計法」的表現才不如預期中的好。

另外，我們也發現有一些新聞標題當中最重要的關鍵字是英文，例如這兩則新聞標題：

- AMD 推出 750-MHzAthlon 與 533-MHzK6-2 處理器
- AMD 讓 CPU 的速度達到 750MHz

這兩則新聞標題中，AMD 與 750MHz 是很重要的資訊，但是我們的作法是處理中文，這樣就會忽略了這兩個很重要的訊息了。因此我們覺得未來要加強分群結果的準確度，處理英文的關鍵字是很必要的一件事。

在實驗中採用的新聞標題相似度計算方式都是在觀察標題之間相同的字，我們是假設描述相同事件的新聞標題都會有若干相同的字出現在標題中，而系統就可以根據這一些字來算出標題之間的相似度。但是我們發現有一些在標準答案中被歸類於同一群聚的新聞標題，卻完全沒有相同的字存在。例如：

- 國政顧問團宣布解散
- 人事名單推移 出現骨牌效應

我們推論這兩則新聞當初會被歸為同一事件，可能是由於人事名單的問題而導致國政顧問團的解散。

- 陳水扁新政府的憲政互動關係

- 唐飛公布四閣員名單

這可能是由唐飛公布閣員名單這一件事情，來看唐飛與陳水扁之間的互動關係。

- 犬籍登記管理今起實施北市有七十三處登記站
- 帶小狗植晶片藝人帶頭示範

這兩則應該都是講述犬籍登記的事情，但第二則新聞所使用的描述文字不同，因此兩則新聞當中找不到相同的字。

如果我們單純只以相同的字作為評判新聞是否需要為同一群聚的依據，則分群的結果將會一直受到影響，因此我們未來要提升分群的準確度，克服這類的情形也是很重要的課題。

6.2 未來研究方向

首先我們要找到別的方式來調整「權重平均法」當中的參數，由於目前「下坡式 simplex 搜尋法」不可行，因此這個相似度的計算公式並未發揮預期中的效果，如果可以找到其他調整參數的方式，則或許這個計算公式會有更好的表現。

針對於「統計法」表現不佳之處，由於我們在實驗中只取用了兩個類別，在樣本數不夠多的狀況之下，我們也很難斷定這個方法的優劣，因此未來還需要將多個類別都代入運作，並與先前的實驗作比較，方能得知這樣的方法是否可行。

在前面一節提到的英文關鍵字的部份，也是我們將來需要在取用特徵值時需要加入的一項工作，我們相信加入英文關鍵字做為分群的特徵之一，將使得分群結果更為周延，也更為正確。

而對於同屬一個群聚中的新聞標題，卻完全沒有一個字相同的情形，我們認為可以透過建立字詞的關連性方面來著手，我們可以將一些關鍵的字定為同義的關係，如此雖然新聞標題使用的是不同的描述字眼，但是系統仍然可以視其為相同的事件，透過這樣的方式，就能夠使得部份本屬於同一群聚的新聞標題，更易於凝聚在同一群聚之中。

如果能夠克服處理時間過久的問題，加入語料庫的支援也能夠提升分群的準確度，系統可以透過語料庫中詞性的標注，從新聞標題中取得更多的資訊，對於分群的工作應該是更有幫助。

參考文獻

[1] N. Jardine, C. J. van Rijsbergen, "The Use of Hierarchical Clustering in Information Retrieval", *Information Storage and Retrieval*, 7, pages 217-240, 1971.

[2] D. T. Phillips, A. Ravindran, J. J. Solberg, *Operations Research: Principles and Practice*, pages 79-84, John Wiley & sons, Inc., New York, 1976.

[3] A. Griffith, H. C. Luckhurst, P. Willet, "Using Inter-Document Similarity Information in Document Retrieval Systems", *Journal of the American Society for Information Science*, 37, pages 3-11, 1986.

[4] P. Willet, "Recent trends in hierarchical document clustering: a critical review ", *Information Processing ad Management*, 24, pages 557-597, 1988.

[5] E. Rasmussen, "Clustering algorithms", In W.B. Frakes and R. Baeza-Yates, editors, Information Retrieval, pages 419-442. Prentice Hall, Eaglewood Cliffs, N.J., 1992.

[6] R. Cutting, D. R. Karger, J. O. Pederson, J. W. Tukey, "Scatter/gather: a cluster-based approach to browsing large document collections", In *15th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318-329, 1992.

[7] R. Schalkoff, "Similarity Measures, Matching Techniques, and Scal-Space Approaches", In Pattern Recognition: Statistical, Structural and Neural Approaches, pages 329-330, John Wiley & sons, Inc., New York, 1992.

[8] V. Faber, "Clustering and the Continuous k-Means Algorithm", *Los Alamos Science*, November 22, 1994.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, pages 314-320, MIT Press, Cambridge, Massachusetts, 1996.

[10] J. -S. Roger Jang, “Derivative-Free Optimization”, In J. -S. Roger Jang , C. –T. Sun, E. Mizutani editors, *Neural-Fuzzy and Soft Computing*, pages 173-193. Prentice Hall, Inc., Upper Saddle River, New Jersey, 1997.

[11] O. Zamir, O. Etzioni, O. Madani, R. M. Karp, “Fast and Intuitive Clustering of Web Documents”, In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 287-290, 1997.

[12] H. -C. Lin, J. -S. Roger Jang, "Survey and Implementation of Clustering Algorithm", *MS Thesis*, Tsing Hua University, Taiwan, R.O.C., 1999.

[13] **林頌華，張智星，“新聞標題自動分類”，碩士論文，國立清華大學資訊工程所，新竹，**1999.

[14] **黃聖傑，陳信希，“多文件自動摘要方法研究”，碩士論文，國立台灣大學資訊工程所，台北**, 1999.

[15] “**網路成學子閱讀新寵 閱讀率達二成八 直逼報紙 側重娛樂影視休閒資訊”，大學報，政治大學新聞系**, June 5, 2000.

[16] **中時電子報** http://www.chinatime.com.tw

[17] **奇摩新聞** http://news.kimo.com.tw

[18] **網羅新聞** http://www.taiyi.com

[19] **明日報** http://www.ttimes.com.tw

[20] **聯合新聞網** http://www.udnnews.com.tw