國立政治大學資訊管理研究所

碩士學位論文

指導教授：楊建民博士

文件距離為基礎 kNN 分群技術與新聞事件偵測追蹤之研究

A Study of Relative Text-Distance-based kNN Clustering Technique and News Events Detection and Tracking

研究生：陳柏均

中華民國一百年七月

致謝

這篇論文能夠完成，首先要感謝林我聰老師、邱光輝老師與季延平老師的悉心指導，有了三位老師的指教，讓本論文的內容可以更加的嚴謹與完整。除此之外，更要感謝指導教授楊建民老師兩年來的教導，楊老師除了授予我們在學業上的知識精華外，更給予了許多寶貴的經驗與人生智慧的傳承；從楊老師的談吐與待人接物之間，無不可以看到一個智者的身影與自我學習的典範，在這裡致上最崇高的謝意與最誠摯的祝福給楊老師。

短短兩年的研究所生涯中，很有幸的能夠認識到許多優秀的朋友們。感謝月純學姊與敏珠學姊在論文上的指教與建議；感謝春美、孝文、承翰等學長姐們在學業上的啟發與鼓勵；感謝鴻仁、婉婷、康維、國傑等學弟妹所帶來的歡笑與活力；感謝又誠、士揚、取向、彥璋和祺堯等碩班夥伴們豐富了我的研究所生活；感謝非常適時出現的 Jason，從你身上我得到了不同的視野與經歷，但我們仍要繼續努力；感謝 APa、賢能、哲雯、宗達、國言與宗軒等前輩，是你們教導了我在實務上的經驗與知識；當然，最特別的感謝要獻給最特別的一群小小戰友們，謝謝智民、漢瑞、振和與章威，我們共同學習奮鬥的精神從學業、論文一路延續到了虛擬世界中，相信這份精神會一直的凝聚下去。要感謝的人太多了，總之，謝謝所有幫助過我，陪我體驗人生中喜怒哀樂的朋友們，祝福你們永遠健康快樂。

總結這兩年研究所的時光，獲得的知識與經歷遠遠超出原先的想像，也因此要感恩的對象與心情一時間難以用三言兩語描繪表達，但肯定的是這兩年所得到的滿滿收穫將會是未來不斷成長茁壯的養分。最後，感謝一路陪伴我的衍姍與我最敬愛的父母，是你們帶給我不斷前進的動力，我會繼續加油的！

摘要

新聞事件可描述為「一個時間區間內、同一主題的相似新聞之集合」，而新聞大多僅是一完整事件的零碎片段，其內容也易受到媒體立場或撰寫角度不同有所差異；除此之外，龐大的新聞量亦使得想要瞭解事件全貌的困難度大增。因此，本研究將利用文字探勘技術群聚相關新聞為事件，以增進新聞所帶來的價值。

分類分群為文字探勘中很常見的步驟，亦是本研究將新聞群聚成事件所運用到的主要方法。最近鄰 (k-nearest neighbor, kNN)搜尋法可視為分類法中最常見的演算法之一，但由於 kNN 在分類上必須要每篇新聞兩兩比較並排序才得以選出最近鄰，這也產生了 kNN 在實作上的效能瓶頸。本研究提出了一個「建立距離參考基準點」的方法 RTD-based kNN (RelativeText-Distance-based kNN)，透過在向量空間中建立一個基準點，讓所有文件利用與基準點的相對距離建立起遠近的關係，使得在選取前 k 個最近鄰之前，直接以相對關係篩選出較可能的候選文件，進而選出前 k 個最近鄰，透過相對距離的概念減少比較次數以改善效率。

本研究於 Google News 中抽取 62 個事件(共 742 篇新聞)，並依其分群結果作為測試與評估依據，以比較 RTD-based kNN 與 kNN 新聞事件分群時的績效。實驗結果呈現出 RTD-based kNN 的基準點以常用字字彙建立較佳，分群後的再合併則有助於改善結果，而在 RTD-based kNN 與 kNN 的 F-measure 並無顯著差距($\alpha = 0.05$)的情況下，RTD-based kNN 的運算時間低於 kNN 達 28.13%。顯示 RTD-based kNN 能提供新聞事件分群時一個更好的方法。最後，本研究提供一些未來研究之方向。

關鍵字：文字探勘、kNN、事件偵測與追蹤、分類分群

ABSTRACT

News Events can be described as "the aggregation of many similar news that describe the particular incident within a specific timeframe".Most of news article portraits only a part of a passage, and many of the content are bias because of different media standpoint or different viewpoint of reporters; in addition, the massive news source increases complexity of the incident. Therefore, this research paper employs Text Mining Technique to cluster similar news to a events that can value added a news contributed.

Classification and Clusteringtechnique is a frequently usedin Text Mining, and K-nearest neighbor(kNN) is one of most common algorithms apply in classification. However, kNN requires massive comparison on each individual article, and it becomes the performance bottlenecks of kNN. This research proposed Relative Text-Distance-based kNN(RTD-based kNN), the core concept of this method is establish a Base, a distance reference point, through a Vector Space, all documents can create the distance relationship through the relative distance between itself and base. Through the concept of relative distance, it can decrease the number of comparison and improve the efficiency.

This research chooses a sample of 62 events (with total of 742 news articles) from Google News for the test and evaluation. Under the condition of RTD-based kNN and kNN with a no significant difference in F-measure (α=0.05), RTD-based kNN out perform kNN in time decreased by 28.13%. This confirms RTD-based kNN is a better method in clustering news event.At last, this research provides some of the research aspect for the future.

Keyword: Text Mining, kNN, Events Detection and Tracking, Classification and Clustering

目錄

圖目錄

表目錄

第一章　　緒論

第一節　研究背景

在這個變動日益快速的時代，資訊的數量呈爆炸性的成長，新聞可以說是一般人最普遍容易接受到的資訊之一，亦是政府機關或企業透過媒體監測來了解社會大眾反映的重要來源。由於新聞大量與即時的特性，使得網際網路逐漸成為新聞的重要傳播途徑。以台灣地區的線上新聞內容為例，一天之內可發生數百條不同的主題事件，同一個主題來自於媒體的相關報導少則十來篇，多則上百篇皆有。面對如此大量且來源不同的即時資訊，加上各家媒體對於新聞事件的角度與立場不同，使得閱聽人一時間難以整理消化。因此，如何過濾這些資料，並且從這些大量的資料中挖掘出有價值的資訊變成一項很重要的課題。

隨著資料量不斷的成長，人們開始發現，從這些看似雜亂無章的紀錄中似乎可以找出一些規則或模式；再加上快速成長的資通訊科技輔助，才得以讓我們能忠實的記錄下足夠的資料來觀察與發現隱含的事實—在這些條件的匯集之下，加速了資料探勘(Data Mining)這門學問的產生與運用。

資料探勘為知識發掘(Knowledge Discovery)的重要步驟之一，其嘗試透過統計、數學、電腦科學等方式挖掘出各種可用的資訊，不過資料探勘的方法僅適合處理結構化程度較高的資料，對於半結構化或是非結構化的資料則較無用武之地。但平常人類所使用的語言、文字等皆屬於結構化程度較低的資訊來源，其中卻往往存在著比結構化資料更高的知識含量與利用價值，也因此嘗試去觀察分析低結構化資料的文字探勘(Text Mining)逐漸受到重視。文字探勘的目的與資料探勘類似，兩者皆是希望透過觀察大量的資料來發現隱藏於其中的事實，並結合了資料

探勘、資訊擷取、機器學習、統計學等領域的知識。雖然文字探勘技術可以運用的範圍日益廣泛，但隨著資料量的暴增，文字探勘應用往往需要龐大的運算能力與運算時間，這也使得文字探勘較難被採用在時間急迫性較高的應用上。

第二節　研究動機

k-最近鄰(k-Nearest Neighbor, kNN) 為文字探勘中很常被運用的分類方法之一。kNN 運用了「相似的事物容易群聚在一起」的概念，也就是找出「前 k 個最近」的鄰居，再觀察這些最近鄰大多屬於哪種類別作為判斷類別的依據。雖然 kNN 一般被視為分類的方法，但若將其整合於分群流程，同樣可以達到分群效果。儘管有研究指出 kNN 的分類結果與效率不遜於目前其他常見方法 (Yang, Yiming,Lin,Xin, 1999; Joachims, T, 1998)，但由於在文字探勘的向量空間中，文件之間的遠近(相似度的高低)關係必須要所有文件比較後才得以產生，因此 kNN 在尋找一資料的前 k 個最近鄰時，必須要與所有文件進行比較才得選出，這也形成了分群效率的瓶頸所在。因此，本研究試圖提出一個在文字探勘的環境下，以 kNN 為依據改良而成的方法 RTD-based kNN (RelativeText-Distance-based kNN)，利用在文字向量空間中建立出虛擬文件作為基準點，進而建立出距離索引的概念來預先排序文件的相似度關係，並透過減少相似度的比較次數降低運算時間，將其應用於新聞的事件分群中。

新聞分群的動機在於發掘新聞真實的面貌，以改善閱聽人在觀看新聞或是監測媒體時的效益─因為新聞傳達的內容往往影響大眾對於事件的觀感與判斷。儘管新聞所呈現的內容均是取材於真實世界，但報導的內容容易受到各種內外部因素的影響而呈現出許多偏向(News Bias)。這些偏向包含了記者的主觀意圖、媒體組織和意識形態等，使得新聞以不同的面貌被形構出來。亦有學者從社會學的觀點來看新聞報導與新聞事件的關係，認為記者在報導政治、社會等事件時，通常

已預設了某種政治立場或主觀意識，而利用新聞報導作為工具來達成其目的，並合理化記者的新聞選擇政策或意識形態，因此新聞的客觀與社會的真實之間亦難畫上等號。此外，新聞事件的發生也有其生命週期，被報導的重點往往隨著事件的發生過程有許多差異，如 2011 年 3 月發生的日本大地震可視為許多新聞的一個事件集合，但媒體報導的重點從最原始的地震、傷亡，到後來的核子危機、環境污染等差異極大。過多且不同來源的新聞資訊除了造成閱聽人在閱讀上的困擾外，也由於新聞的零碎鬆散，無法讓閱聽人能清楚的看到整個事件的全貌。若能透過文字探勘技術對於新聞的群聚與處理，勢必將有助於改善。

第三節　研究目的

總結前述之背景與動機，本研究所要達成的目的如下：

1. **提出 RTD-basedkNN 演算法運用於新聞事件分群。**
2. **應用 RTD-based kNN 於新聞事件偵測追蹤以改善其效率。**
3. **比較 RTD-based kNN 與 kNN 新聞偵測追蹤之績效。**

第二章　　文獻探討

第一節　資料探勘

2.1.1　　資料探勘定義

隨著日常生活中各種資料量不斷擴張，資料探勘(Data Mining)技術已經在許多領域受到重視並廣為應用。其目的在於從大量的資料中找出隱藏於其中的資訊，以便進一步加以解釋或運用。

在定義方面，謝邦昌(1996)認為資料探勘是尋找隱藏在資料中的訊息，如趨勢(Trend)、特徵(Pattern)及相關性(Relationship)的過程，也就是從資料中發掘資訊或知識(KDD)。Fayyad(1990) 認為資料探勘就是一個萃取出資料中有效的、嶄新的，可具有效益且最終能被理解的重要過程，最終目的是了解資料的形樣。Roiger, R., Geatz, M.(2003) 則表示資料探勘是一種從整個資料庫裡的資料，利用一種或多種電腦技術自動分析或去擷取知識的過程。

Fayyad(1996) 與 Han(2005) 皆認為資料探勘是知識發現(Knowledge Discovery from Data, KDD)的重要步驟，但資料探勘並非同義於知識發現(KDD)。綜合前述定義可以發現，資料探勘僅是從大量資料中發現知識的程序之一，但在知識發現的過程中與資料探勘卻是很重要的一個步驟。而 Fayyad et al. (1996) 提出了一連串反覆式的 KDD 步驟，各步驟彼此交互影響，如圖 2-1，分別為：

1. 資料選擇(Selection)

   確認知識發現的操作對象，即目標資料(Target Data)作為整個程序中

的探勘目標。

2. 前置處理(Preprocessing)

   適當的處理不完整、遺失或錯誤的資料來消除雜訊，決定目標資料的型態、欄位、資料綱要等。

3. 資料轉換(Transformation)

   對目標資料進行簡化、轉換，以減少資料的處理量。通常透過選取特徵值來降低維度(Dimension Reduction)、轉換或編碼等方式。

4. 資料探勘(Data Mining)

   為KDD中最重要的步驟。透過分群、分類、關聯規則、決策樹、迴歸分析和時間序列分析等演算法找出資料的特徵或規則。

5. 解釋或評估(Interpretation/Evaluation)

   將資料探勘產出的特徵或模式轉換為圖形、圖表等成較為容易理解的表達方式，以供決策參考。同時也必須評估探勘結果是否合理或適用，並進一步決定是否對各步驟進行必要之調整。

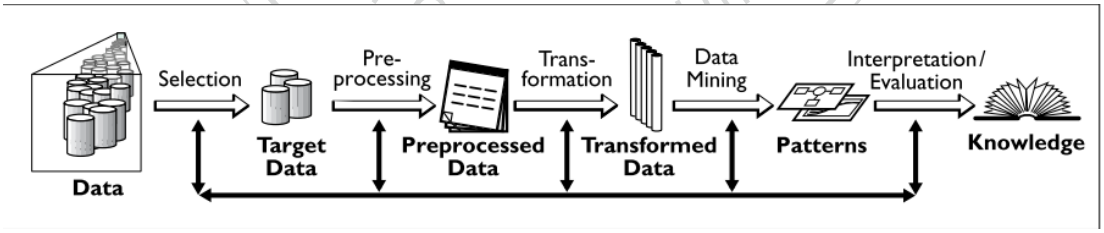

圖 2-1 KDD 步驟資料來源：Fayyad(1996)

2.1.2 常用資料探勘方法

在整個知識發現的過程中，資料探勘可以視為最重要的步驟之一，因此必須依照資料的特性與目的來決定採用何種資料探勘方法。在常見的資料探勘方法中，依據需求、分析方式或產生的知識型態可分類為下列幾項(羅閩隆，2004)：關聯

法則(Association Rules)、推估(Estimation)、預測(Prediction)、分類(Classification)與群集偵測(Cluster Detection)，說明如下：

1. 關聯法則

   主要用於尋找資料集中資料項目或屬性間的關聯，以分析及了解資料中隱藏的含意或是找出未知的關聯性。如透過交易資料瞭解顧客購買產品的順序及喜好，作為商品排列或是擺放位置的參考。

2. 推估

   適合用於處理連續或有順序性數值，可用來推估一些未知的連續性變數。如利用信用卡申請者之教育程度、收入、職業等因素，推估其信用卡消費額度與適合哪一種促銷專案。

3. 預測

   預測分析與推估分析相當接近，差異點在於預測是用於推估未來的數值與趨勢。預測通常採用歷史資料作為已知的變數值訓練資料，並建立起模型描述過去至現在觀察值之變化，再利用最近的資料輸入至模型中，藉以獲得對於未來觀察值變化的預測。

4. 分類

   最基本的分類是從已知特定類別的資料集合中，依據資料的屬性或特徵建立出一個分類模式，用來描述資料與類別間的關係，再依據此分類模式對其他未經分類或是新的資料做預測，決定其所屬的類別。

5. 分群

   叢聚資料及探勘分析方法，主要是計算每筆資料間的相似程度、影響關係，並將擁有相似屬性或特徵的資料群聚為同一個叢集(Jain, 1999)，叢集內資料的描述將會以叢集的特性來取代個別資料的屬

性。叢集內資料的屬性越相似越好，而叢集間彼此的差異性則是越大越好。目前常用的分群方法有k-means、LSH，或是利用模糊理論(Fuzzy Theory)來進行叢聚探勘的分析(Krishnapuram et al.,2001 ; Rousseeuwet al, 1996)等。

第二節　文字探勘

2.2.1　文字探勘定義

有別於傳統資料探勘，文字探勘(Text Mining)所處理的通常為半結構化或者非結構化等以自然語言撰寫出來的文件；資料探勘(Data Mining)技術則主要針對於結構化的表格資料，卻難以處理半結構化與非結構化的文件(Feldman, 1995; Singh, 1997)。文字探勘試圖從文件中找出重要的字詞(Term)或片語(Phase)、字詞間的關聯強度(Association Degree)、分類或推論規則等(Classification or Prediction Rule)(巫啟台，2002)，結合數學、統計、機率、人工智慧、資料檢索及資料庫等相關知識，用於從大量的資料中萃取出有用的資訊。為了增加結果的有效性與準確性，文字探勘必須嘗試讓機器瞭解文件的本意，因此要透過字詞處理技術來分析與表達文件以便做進一步的運用。目前較常被使用的字詞處理技術含斷詞處理、字詞權重計算、向量空間模型表示等。

2.2.2　斷詞處理與權重計算

斷詞處理的目的在於將文件斷成各個有意義字詞(Term)的集合，而中文斷詞的斷詞過程有別於印歐語系，印歐語系文件在詞與詞間以空白隔開，因此斷詞僅需以空白相隔即可斷出獨立詞彙(Nie,1996)；相較之下，中文文件中詞與詞間並無明顯區隔可用於斷詞。目前在中文斷詞領域大致有三種方法，分別是：詞庫式斷詞法 (Chen,1992)、統計式斷詞法 (Fan,1988; Sproat,1990)及混合式斷詞法

(Nie,1996)，說明如下：

1. 詞庫式斷詞法

   為目前普遍使用的斷詞方式，其演算法相當直覺且實作容易。然而斷詞的品質和詞庫的大小有相當的關係，因此必須時常對詞庫的內容加以維護。有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，以提昇斷詞的品質(陳克健，1986)。

2. 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階馬可夫模式(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)，因此統計式斷詞法大多只針對兩字詞進行處理，超過兩字詞以上的詞語就無法有效擷取。

3. 混合式斷詞法

   混合式斷詞法整合了詞庫斷詞法及統計斷詞法。(Nie,1996)利用詞庫斷出不同組合的詞彙，然後以字詞的統計資訊，找出最佳的斷詞組合。此法仍需要大型的語料庫提供統計資訊。

由於每篇文件中各個字詞的重要程度並不相同，因此在經過斷詞處理後，各個字詞可透過權重(Weight)來表達其在文件中的重要性。而權重又可分為在文件中的重要性(Local Weight)(表 2-1)及在整個文件集中的重要性(Global Weight)(表 2-2)。

表 2-1 常見 Local Weight 計算方式

公式名稱	Local Weight 公式
Within-document frequency (term frequency, tf)	$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}$
Binary	$1\ if\ \mathrm{t}f_{i,j} > 0\ ;\ 0\ if\ \mathrm{t}f_{i,j} = 0$
Log	$1 + \log tf_{i,j}\ if tf_{i,j} > 0$ $0\ if tf_{i,j} = 0$
Normalized Log	$\left(1 + \log tf_{i,j}\right)/\left(1 + \log a_j\right)\ if tf_{i,j} > 0$ $0\ if tf_{i,j} > 0$
Augmented normalized term frequency	$0.5 + 0.5\left(tf_{i,j}/x_j\right) if tf_{i,j} > 0$ $0\ if tf_{i,j} = 0$

資料來源： Popescu(2001)整理

表 2-2 常見 Global Weight 計算方式

公式名稱	Global Weight 公式
Inverse document frequency	$\log(N/n)$
Probabilistic inverse	$\log[(N - n_i)/n_i]$
Entropy	$1 - \sum_{j=1}^{N} \frac{\frac{tf_{i,j}}{tF_i} \log \frac{tf_{i,j}}{tF_i}}{\log N}$
Global frequency IDF	tF_i/n_i
No global weight	1

資料來源： Popescu(2001)整理

k 為文件 j 中的字詞數，$f_{i,j}$為字詞 i 於文件 j 中出現的次數，$tf_{i,j}$為字詞 i 在文件 j 出現的頻率(Term Frequency，詞頻)。a_j為文件 j 中所有字詞詞頻的平均數，

x_j為文件 j 中出現次數最多的字詞數，N 為整個文件集中的文件總數，n_j為字詞 i 在文件集中所出現頻率(Document Frequency，文件頻率)，F_i為字詞 i 在整個文件集中所出現的總次數。

欲表達字詞在一文件中的重要程度，最常用的字詞權重計算方式為 TF-IDF(Term Frequency – Inverse Document Frequency)，計算方式為取 local weight 中的詞頻 (tf_{ij}) 乘上 global weight 中的逆向文件頻率 (Inverse Document frequency)，即：

$$w_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \log(N/df_i) = tf_{i,j} \times idf_i \quad \text{(公式 1)}$$

其中$w_{i,j}$為字詞 i 在文件 j 中的權重，$tf_{i,j}$為字詞 i 在文件 j 中的詞頻，df為字詞 i 出現在整個文件集的文件數，N 為整個文件集的文件數。TF-IDF 的涵義為字詞在文件中的重要性是與其在文件中出現的次數成正比，但與其在所有文件集中出現的文件數成反比，原因在於若字詞出現於其他文件的頻率越高，則對於能代表本文件的識別力就越低。

為了避免因文件長度差異而影響文件集中各字詞之權重比較，可將 TF-IDF 所算出的字詞權重做正規化處理，方法為將權重除以文件向量中所有元素(權重)平方和再開根號，即文件長度$\|\vec{d_j}\|$，正規化權重如公式 2。

$$w_{i,j} = \frac{tf_{i,j} \times idf_i}{\|\vec{d_j}\|} \quad \text{(公式 2)}$$

2.2.3 向量空間模型(Vector Space Model, VSM)的運用

在文字探勘中，向量空間模型是最簡單也最具有生產力的模型(Salton, 1983)，因此是目前最被廣為使用的資訊檢索模式，最早由 Gerard Salton 所提出(Salton, 1975)。其目的是在文字檢索的過程中，將文件轉化成字詞索引的集合，同時針對各個字詞索引給予權重(Weight)，來表達每個字詞在文件中的重要程度與價值，而最常用的權重計算方式為前述 TF-IDF 計算。建立索引的方式為在文件集集合 D 中，找出一組屬性，使得 D 中某一文件能有一組屬性值具有足夠的資訊來代表文件，該組屬性值即稱為文件的索引向量，而此文件向量即代表在向量空間模型中的一篇文件。

在一文件集中，每個索引字詞即代表空間中一個維度，每個維度上的值則代表該文件在這個維度上的重要程度，通常以權重表示。以圖 2-2 為例，三維空間中文件皆由三個不同字詞(T_1, T_2, T_3)所組成，依照每個文件中索引字詞的權重不同，在空間中的位置亦然不同。若將此例子延伸到多維度，可以數學矩陣的方式表達及運算，如圖 2-3 所示，其權重為字詞 i 在文件 k 中的權重。

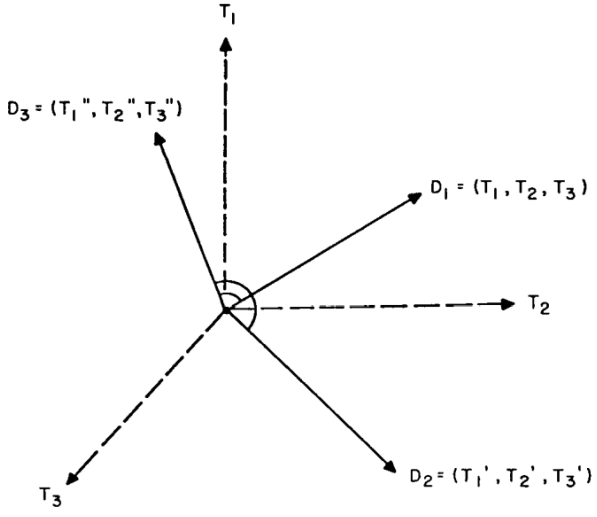

圖 2-2 向量空間模型資料來源：Salton, Gerard, Wong , A. &Yang , C.S. (1975)

$$\begin{bmatrix} & Term_1 & Term_2 & \dots & \dots & \dots & Term_i \\ Doc_1 & W_{11} & W_{12} & \dots & \dots & \dots & W_{1i} \\ Doc_2 & W_{21} & W_{22} & \dots & \dots & \dots & W_{2i} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ Doc_k & W_{k1} & W_{k2} & \dots & \dots & \dots & W_{ik} \end{bmatrix}$$

圖 2-3 (1983)

2.2.4 相似度計算

在文字探勘的向量空間模型中，計算兩文件的相似程度最常用的方法即計算兩文件的餘弦相似度(Cosine Similarity)，主要以兩組相同基底(Base)與維度(Dimension)向量間的角度(Angle)差距來度量兩向量間的距離(Jia-Ming, You., Keh-Jiann, Chen, 2006 ; Teng, W.-G., & Lee, H.-H., 2007)。其計算結果會介於 0 至 1，當兩個向量間的角度差距越小時，表示該向量的餘弦角度越小(兩篇文章越相似)，結果越接近 1；反之，則越接近 0(陳崇正，2009)。餘弦相似度於二維空間如圖 2-4 所示。在 n 維空間的夾角公式則為：

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad \text{(公式 3)}$$

$A = (X_1, Y_1)$, $B = (X_2, Y_2)$

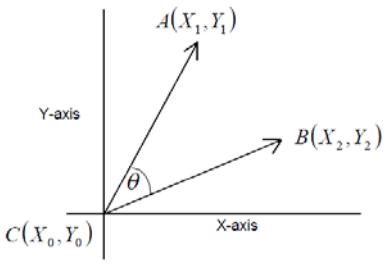

圖 2-4 二維空間中的餘弦相似度資料來源：陳崇正(2009)

在向量空間模型中，若所有文件中的字詞權重皆經過正規化處理，兩文件的相似度則亦可運用歐幾里得距離來判斷。距離越近則代表兩文件越相似，計算的公式為：

$$\text{Dist}(A, B) := \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \quad \text{(公式 4)}$$

2.2.5 分類技術

在文字探勘中，分類主要是利用文件的特徵或是屬性將其歸類到事先定義好的類別中，因此必須透過已知類別的訓練資料建立模型，藉此預測新資料的所屬類別，屬於監督式學習法(Supervised Learning)。常見的分類技術包含傳統的「類神經網路」(Artificial Neural Network, ANN)、「最小平方誤差法」(Linear Least Square Fit, LLSF)、以距離(相似度)為基礎的「K 個最鄰近法」(k-Nearest Neighbor Algorithm, KNN)、以統計方法中貝氏定理為基礎的「簡單貝氏分類法」(Naïve Bayes, NB)以及從空間中找出超平面(Hyper-plan)做為分隔基礎的「支援向量機」(Support Vector Machine, SVM)。Yang et al. (1999)與 Joachims (1998)曾以統計的方法比較上述幾種分類法的效率與分類結果，綜合評比後發現優異程度為 {KNN, SVM} > LLSF > ANN >> NB。

另外在 Sebastiani (2002)的整理中，「機率模型」(Probabilistic Model)、「決策樹」(Decision Tree)、「決策規則模式」(Decision Rule Model)以及「例舉式學習」(Example-Based Learing)等方法也都曾被利用在文件分類模式的建構。

2.2.6 分群技術

分群是依照文件的相異性或相似性，將相異性較低或相似性較高的文件群聚起來，目標是使得群集內每個文件彼此擁有極高的相似度，但每個群集間的相似程度則是越低越好。分群不像分類需要利用已知的資料訓練並指定類別，事先也並不知道分出來的群集數，屬於非監督式(Unsupervised Learning)學習。

Jiawei Han and Micheline Kamber(2006)將分群法依其性質分成五大類，分別是：分隔式分群(Partitioned)、階層式分群(Hierarchical)、密度基礎分群(Density-based)、網格式分群(Grid-based)與類神經網路分群(Neural network)，其中又以分割式演算法中的 k-means 最廣為人知。k-mean 由 J. B. MacQueen 於 1967 年所提出，分群前必須先設定群集數量 K，利用反覆式的計算叢集重心來使各群集重心趨於穩定。但 k-means 缺點在於重心的概念容易受到資料的離散程度影響，且事先設定的群集數量亦未必正確，若資料量龐大易造成整體效率低落。

值得一提的是，kNN 雖然被歸類於分類演算法中，但在實作上亦可不用事先設定類別與給予訓練資料，如 Yang et al.(1999)利用 KNN 於「類別數未知」的新聞事件的偵測追蹤，即可視為於分群的運用。

第三節 k-最鄰近演算法 (k-Nearest Neighbor, kNN)

2.3.1 kNN分類演算法於文字探勘

T.M. Cover and P.E. Hart 於 1967 年提出 k-最近鄰演算法，至今仍為常用的分類方法之一。理論上，在文字探勘中，資料就是因為擁有某些共同的相似特徵而

被歸類在同一類別。所以 kNN 的概念為：未知類別的資料與「同類型資料的相似度」應該要比「不同類型資料的相似度」高。kNN 分類法採用向量空間模型來分類，在對文件分類前必須將文件轉換為向量空間模型，再藉由計算與已知類別內文件的相似度，來評估未知類別文件的可能類別。換言之，即是透過未知類別資料與各類別內的文件比較相似度，來判斷所屬的類別，其中 k 為取樣文件數，代表了要取與未知類別文件前 k 個最相似的已知類別文件，藉以判斷未知類別文件應該被歸類至何處。而文件的相似度在文字探勘中一般採用 cosine 相似度計算。kNN 分類步驟如下：

1. 將新進文件轉換為向量表示。
2. 將新進文件與文件集內所有文件比較相似度，取出前k份最相似的文件。
3. 將這k份文件所屬的事件當成候選的事件類別。
4. 將這 k 份文件與新文件的相似度依照所屬的事件個別加總，相加結果數值最高的類別即為新文件所屬類別 (但相加結果亦須大於所訂的門檻值)。

2.3.2 kNN運用於新聞事件的偵測與追蹤

新聞代表了讀者與新聞界共同感興趣的新事件或新觀念，而新聞事件(Event)可以視為描述著同一個「主題」的新聞群集，通常都會有數篇不同來源或角度的新聞集合而成，並且僅存在一個特定的時間區間中，也因此可被定義成「在特定的時間及地點所發生的相關事物之集合」；而新聞事件的追蹤則可被定義為「發現包含在連續的新聞串流中有關新的或之前未發現的事件」(Allan et al, 1998)。

在美國國防部高等研究計畫局所主導的「主題偵測與追蹤(Topic Detection and Tracking, TDT)」計畫中，「新聞事件的追蹤與偵測」即為其中的一個子項目，

該計畫的研究目的為「從各種管道的新聞串流中找出或追蹤事件」。參與 TDT 的先導性計畫含卡內基美隆大學(Carnegie Mellon University, CMU)與麻州大學(University of Massachusetts, UMass) (Yang et al., 1999)兩校。在 CMU 的「新聞事件的偵測與追蹤」研究中，將已存在的事件皆透過事件內所有新聞文件計算出質心(Centroid)作為代表。新進的文件則先透過時間篩選出候選事件，並找出新進文件與候選事件中最相似事件的相似度，若結果小於一門檻值(Threshold)(此門檻值通常介於 0.15 至 0.23 間)(戴尚學，2003 ; Yang et al., 2000 ; Yang et al., 1999)，則判斷為不屬於已存在的事件，反之，則再繼續透過事件追蹤來判斷其所屬事件。

經事件偵測判定為「非新事件」的新聞將交由事件追蹤處理，事件追蹤的目的在於將新進新聞文件正確的歸類至已存在的事件(新聞群集)中，歸類的方式採用 Single-Pass Clustering 流程，即對於現有的群集中，透過分類的方法判斷是否被歸類在這些群集內。在分類的部分，CMU 使用 kNN 進行群集的指派，其評估了 TDT 的需求(每個事件都要能獨立的追蹤，而事件中不含其他事件的分類知識)，將 kNN 改為 2-way kNN(戴尚學，2003 ; Yang et al., 2000 ; Yang et al., 1999)。最大的差異在於原本的 kNN 僅會被加入於相似度最高的事件群集，不符合 TDT 每個事件都要能夠獨立的被追蹤，因此 2-way kNN 針對每個候選事件獨立判斷是否應該被歸類在其中。在 2-way kNN 中，比較的對象可分為兩組：「目標群集」(要判斷新文件是否屬於此群集，內含文件稱為 Positive Document)以及「其他群集」(目標事件群集以外的文件，稱為 Negative Document)。對於新進文件與候選事件群集計算兩者的相關分數，若相關分數大於一設定的門檻值(如 0.15)，則判斷新進文件屬於此候選群集。相關分數為在新進文件與前 k 個最近鄰中的相似度中，屬於 Positive D ocument 的總和減去屬於 Negative Document 的總和。

在上述的方法中，由於 Positive Document 通常遠低於 Negative Document，若 k 數太大，可能造成太多 Negative Document 被選到，因此即使每篇 Negative Document 與新進文件的相似度都很低，相加起來仍可能比新進文件與 Positive Document 的相似度加總還高；相反的，若 k 取太小，則容易取到都是 Negative Document，造成了 k 值的大小很容易影響判斷的結果(戴尚學，2003) 。為了避免上述情況對於判斷結果所造成誤差，CMU 也提出了兩個改良公式，一個為在原本的 2-way kNN 加入了平均的概念，將新進文件與 Positive Document (Negative Document)的相似度加總除以在 Positive Document (Negative Document)取到的文件數(一共取 k 個)。另一個則是 Positive Document 與 Negative Document 各取 k 個最近鄰，可以保證當 k 值設小時兩個群集都仍會被抽樣到。

第三章　研究方法與設計

經過文獻探討對於文字探勘、分群分類及新聞事件的偵測追蹤有了大致上的介紹之後，本研究提出利用「文件相對距離」改善 kNN 的方法- RTD-based kNN(RelativeText-Distance-based kNN)，透過驗證的方式找出此方法中最佳的參考基準點與各項參數，並將 RTD-based kNN 與 kNN 應用於新聞事件的分群來比較並評估效果。由於在眾多常見的線上新聞媒體中，僅 Google News 匯集大量不同來源的新聞內容，並依照相似性聚集成相關新聞事件，因此本研究的資料來源將採用 Google News 所提供的新聞作為實驗樣本，依照其對於新聞的分群作為 kNN 與 RTD-based kNN 分群的評估基準。

第一節　研究設計

本研究分為兩大階段進行，第一階段為提出並評估 RTD-based kNN 應用於新聞事件分群時的效果，由於 RTD-based kNN 目的在於利用建立距離參考基準的概念預先針對相似度排序，因此考慮的參數除了事件偵測時與文件相似的門檻值外，還必須找出最佳的基準點建立策略。第二部分則將 RTD-based kNN 與 KNN 比較新聞偵測追蹤的時間與結果，並透過 Google News 的相關新聞做評估標準，亦即對於新聞資料集進行分群，使得報導同一事件的相關新聞能自動群聚。研究流程如圖 3-1，整體研究架構如圖 3-2。

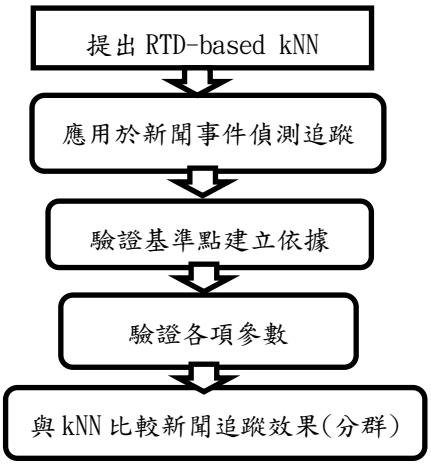

圖 3-1 研究流程圖資料來源：本研究整理

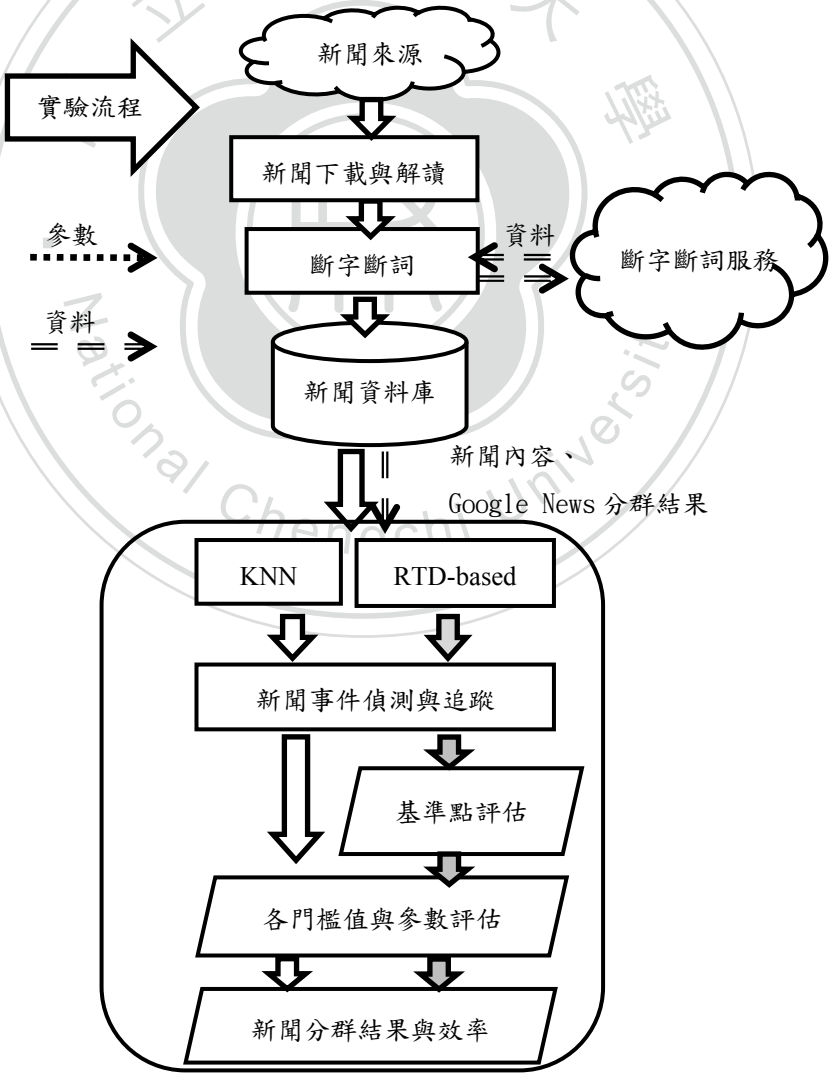

圖 3-2 研究架構圖資料來源：本研究整理

第二節　RTD-based kNN 演算法

3.2.1　kNN分類法描述

本研究嘗試提出一個改良於 kNN 分類的方法- RTD-based kNN。原本的 kNN 在分類時必須找出 k 個最近鄰作為判斷標準，以圖 3-3 為例，若欲判斷資料 Data 屬於黑點(A,B,C)或白點(D,E,F)，則取與資料(Data)「前 3 相鄰」(設 k = 3)的點(即 A,B,E)判斷。由於這前 3 相鄰點中，屬於黑點的距離平均((1+0.5)/2=0.75)大於屬於白點的距離平均(0.2/1=0.2)，因此將資料歸類於白點中。

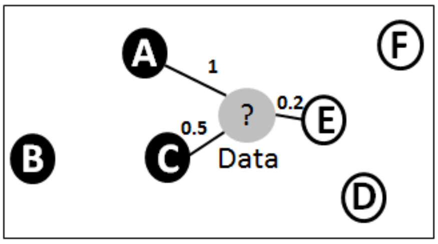

圖 3-3 kNN 分類圖例資料來源：本研究整理

將 kNN 運用於新聞事件的分群時，亦有 2-way kNN 的用法(戴尚學，2003；Yang et al., 2000；Yang et al., 1999)。最大的差異在於原本的 kNN 僅會被加入於相似度最高的事件群集，不符合新聞事件分群中每個事件都要能夠獨立的被追蹤，因此 2-way kNN 針對每個候選事件獨立判斷是否應該被歸類在其中。在 2-way kNN 的方法裡，比較的對象可分為兩組：目標事件群集 (要判斷新文件是否屬於此群集，內含文件稱為 Positive Document)以及其他群集 (目標事件群集以外的文件，稱為 Negative Document)。對於新進文件與候選事件群集來說，計算的結果為兩者的相關分數(Relevance Score)，公式如下：

$$r(\vec{x},k,D) = \sum_{\vec{y}\in p_k} \cos(\vec{x},\vec{y}) - \sum_{\vec{z}\in q_k} \cos(\vec{x},\vec{z}) \quad \text{(公式 5)}$$

其中$\vec{x}$為新進文件的文字向量，$\vec{y}$ ($\vec{z}$)為Positive (Negative) Document的向量，D為整個文件集，k為與新進文件最近鄰(相似)的文件數，$p_k(q_k)$為k個最相似的Positive (Negative) Document之集合。若Relevance Score大於一門檻值，則表示此文件屬於這個群集。

在上述的方法中，由於Positive Document通常遠低於Negative Document，若k數太大，可能造成太多Negative Document被選到，因此即使每篇Negative Document與新進文件的相似度都很低，相加起來還是很有可能比新進文件與Positive Document的相似度加總還高；相反的，若k取太小，則容易取到都是Negative Document，造成了k值的大小很容易影響判斷的結果 (戴尚學，2003) 。為了避免上述情況對於判斷結果所造成誤差，Yang et al. (1999)提出了兩個改良公式：

$$r(\vec{x},k,D)=\frac{1}{|P_k|}\sum_{\vec{y}\in p_k}\cos(\vec{x},\vec{y})-\frac{1}{|Q_k|}\sum_{\vec{z}\in q_k}\cos(\vec{x},\vec{z}) \quad \text{(公式 6)}$$

$$r(\vec{x},kp,kn,D)=\frac{1}{|U_{kp}|}\sum_{\vec{y}\in U_{kp}}\cos(\vec{x},\vec{y})-\frac{1}{|V_{kn}|}\sum_{\vec{z}\in V_{kn}}\cos(\vec{x},\vec{z}) \quad \text{(公式 7)}$$

其中kp為Positive Document中對於新進文件x的k個最近鄰，kn為Negative Document中對於新進文件x的k個最近鄰，U_{kp}為kp之集合，V_{kn}為kn之集合。公式6在原本的2-way kNN加入了平均的概念，將新進文件x與Positive Document (Negative Document)的相似度加總除以在Positive Document (Negative Document)取到的文件數(一共取k個)。公式7則是Positive Document與Negative Document各取k個最近鄰，可以保證當k值設小時兩個群集都仍會被抽樣到。

3.2.2 kNN問題

前述之kNN中，影響效率最大的關鍵在於找出「k個最近鄰」。在多維的向量空間裡，目標文件必須與所有文件計算兩兩相似度，並經過排序之後才能找出k個最近鄰，儘管比較結果可以儲存再利用，但由於文件間彼此的相似度僅相互有意義，因此可再用性極低。

以分類 n 篇新聞為例，對於每篇新聞來說，在不儲存比較結果的情況下，每篇需皆須與其他篇新聞比較以取得相似度排序，共要比較$n \times (n-1)$次；若儲存比較結果，則必須比較$\frac{n\times(n-1)}{2}$(如表3-1)，當資料數量越趨龐大時，即使儲存運算結果，也會因為數量太多而增加結果的存取時間。因此無論是否儲存相似度比較結果，整體而言對每篇文件取前k個最近鄰著實造成很大的運算負擔，這也是本研究欲改善的問題所在。

表3-1 kNN分類相似度比較次數

	儲存比較結果	不儲存比較結果
相似度比較次數	$\frac{n\times(n-1)}{2}$	$n\times(n-1)$

資料來源：本研究整理

3.2.3 參考距離的概念

本研究提出一個修改前述kNN問題的觀念，核心概念是「利用相對的參考距離來建立與其他文件的遠近關係」。kNN在比較時找的k個「最近鄰」目的僅是建立出遠近的概念，在所有向量權重皆正規化的前提下，若能在向量空間中建立一個標的做為參考的基準點(Base)，並讓每篇文件都與這個參考點比較距離(參考距

離)並紀錄結果(距離參考資料集)。當一文件需要找出與本身相鄰的k個目標時，可先從距離參考資料集中找參考距離與自身的參考距離相近的文件開始比較相似度，若相似度大與一門檻值，則判斷為最近鄰之一，重複直到找出k個最近鄰為止。

如圖3-4，假設於二維的空間中，欲找出資料Data的k個最近鄰，原本kNN的作法是將Data與所有文件比較距離，進而求出前k個相近點。若能先將所有點與基準點比較距離，排序儲存於如圖中的「距離參考資料集」，(假設)圖中的Data與Base的距離為1，可先利用此距離取出前後最相近的n筆(如圖例中取出前後各兩筆)，篩選出A,B,C,D四點進行後，再進行相似度比較。若大於設定之門檻值則判斷為k個最近鄰點之一；若比較完這四點仍無法取到k個最近鄰，則可加大所選取的範圍。透過基準點的建立，可以讓在選取k個最近鄰時與先篩選掉許多差距過遠的點，減少判斷時相似度比較的次數。

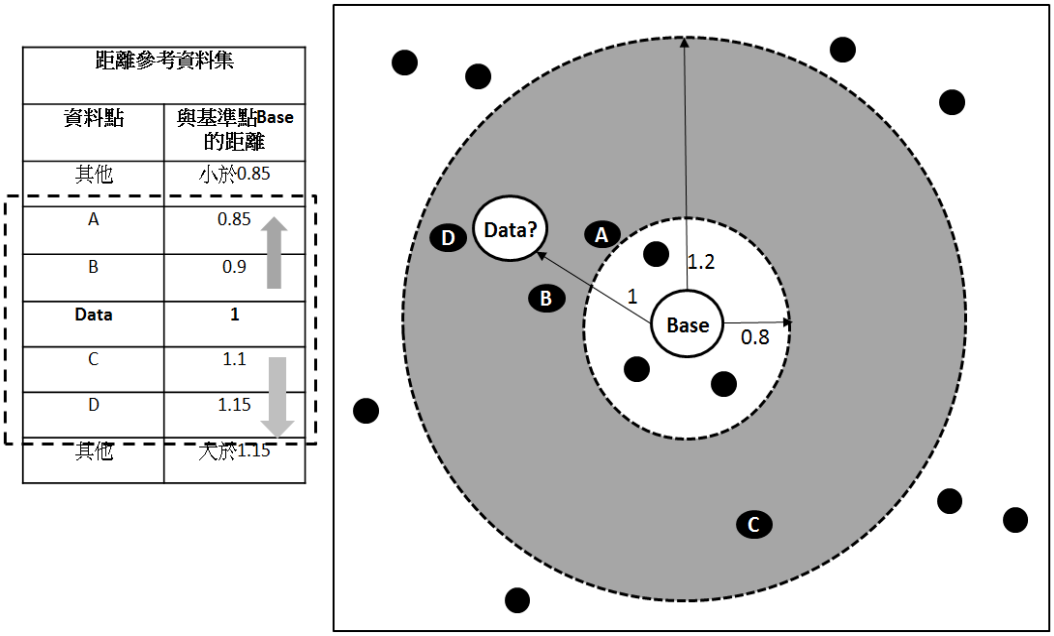

圖3-4基準點的概念示意圖資料來源：本研究整理

第三節　分群結果的合併

RTD-based kNN利用了與基準點的距離作初步的排序，讓相似度比較接近的文件與基準點的距離(參考距離)盡可能相近，減少kNN分群時為了取前k個最相似文件所需的相似度比較次數與時間。但考量基準點文件的詞彙數大小、運算效能與文件內容的差距，參考距離仍無法完全精確地把相似度前k高文件篩選出來，造成RTD-based kNN分群的結果與kNN有著些許落差。在經過初步的實驗後，發現由於RTD-kNN不像kNN比較完所有文件後再排序出前k個最近鄰，造成RTD-based kNN在選擇最近鄰時選到的不一定是所有文件中的最近鄰，而是與參考距離相近中的最近鄰，造成其有時會出現單一事件分成不同子事件的情形，可利用在分群後透過群集質心的相似度計算進行合併來改善。由於新聞文件在分群後，共有的特徵值在經過質心的合併過程中會更加的明顯，可以很容易地透過比較各事件的質心發現應屬同一事件的不同群集，進而將其合併。因此本研究提出在經過RTD-based kNN分群後再比較各事件群集的質心，若Cosine相似度大於一門檻值則將其合併為同一事件。

第四節　新聞的偵測與追蹤

在文獻探討曾提到，卡內基美隆大學(Carnegie Mellon University, CMU)為參與「新聞事件的偵測與追蹤」先導型計畫的學術單位之一，本研究將會以RTD-based kNN 應用 CMU 對於新聞事件偵測與追蹤的方法(Yang et al., 1999)來處理新聞事件的群聚。在偵測事件前必須先將已存在的事件透過其內的所有新聞文件計算出質心(Centroid)作為代表(公式 8)：

$$\overrightarrow{c_m} = \frac{\sum m_i \overrightarrow{c_i}}{\sum m_i} \quad \text{(公式 8)}$$

新進的文件則透過下列公式的計算，用以判斷是否屬於新的事件。若結果大於一門檻值(Threshold)，則判斷為不屬於已存在的事件，反之，則再繼續透過事件追蹤來判斷其所屬事件。

score(x) = 1 $-max_{c_i \in window}\{\ \mathrm{sim}(\vec{x}, \vec{c_i})\ \}$ ……(公式 9)

score(x) = 1 $-max_{c_i \in window}\{\left(1-\frac{k}{m}\right)\mathrm{sim}(\vec{x}, \vec{c_i})\ \}$ ……(公式 10)

其中 x 為新進文件，$\vec{c_i}$為在時間區間內($c_i \in window$)(即與新進文件發生日期較相近的事件)所有事件的質心。而公式 10 則基於公式 9 加入了時間衰退的概念。m 為時間區間內所含的文件數，k 則為群集中，文件 x 的時間至最新文件間的文件數。可以發現 x 文件若在時間區間內越舊，對新事件的影響力較弱。

經事件偵測判定為「非新事件」的新聞則交由事件追蹤處理，其目的在於將新進新聞文件正確的歸類至已存在的事件(新聞群集)中，歸類的方式採用 Single-Pass Clustering 流程，步驟如下：

1. 取出第一份文件當作第一個事件。
2. 取出另一份文件，比較相似度。
3. 利用時間篩選出適當的候選事件群集(文件發生時間是在事件的開始與結束間，或是開始與結束的 n 天內)。
4. 將文件指派到候選事件中的適當事件中，並重新計算事件的群集質心。若新進文件的發生時間早於事件的發生時間，則將事件的開始時間更新為文件發生時間；若新進文件的發生時間晚於事件的發生時間，則將事件的結束時間更新為文件發生時間。

5. 若文件沒有加入任一事件，則自成一新事件。
6. 重複步驟2~5，直到所有文件皆處理完畢

在步驟4「指派到適當群集」中，CMU採用的即是2-way kNN進行群集的指派，本研究將改用RTD-based kNN判斷。

第五節　實驗流程與內容

本研究所實驗的 RTD-based kNN 各項參數如下：

1. 基準點的建立策略：透過計算基準點與各文件的參考距離來讓較相似的文件參考距離相近，減少在取 k 個最近鄰時的相似度比較數量。此實驗的目的在於在各門檻值固定的情況下，測試各種基準點文件建立策略的結果。
2. 事件偵測門檻值的影響：進行事件偵測時，若欲分群的目標文件相關分數(公式 10)小於此門檻值，則繼續進行事件追蹤(開始分群)，否則即成立為新事件。此實驗的目的在於實驗各個事件偵測門檻值的新聞事件偵測追蹤結果。
3. 文件相似門檻值的影響：RTD-based kNN 中，欲分群的目標文件在利用與基準點的距離找出距離相近的文件後，若目標文件與距離相近的文件 Cosine 相似度大於此門檻值，則此距離相近的文件可作為目標文件的最近鄰之一。此實驗的目的在於實驗各個文件偵測門檻值的新聞事件偵測追蹤結果。
4. k 值的影響：RTD-based kNN 在分群時必須透過參考距離取出前 k 個最近鄰，用以判斷是否屬於候選事件。此實驗透過 k 值的改變來探討 k 值的增加是否對於新聞事件偵測追蹤的結果造成影響。
5. 事件合併的差異：在進行新聞事件偵測追蹤後，透過計算各個質心的相似度來判斷是否合併事件。此實驗探討進行合併前後的差異。

第六節　評估方法

本研究透過處理新聞事件的偵測與追蹤來比較kNN與RTD-based kNN兩者之結果與效率。效率方面為比較kNN與RTD-kNN在處理完所有新聞後的所需時間；分群結果採用的基準則是Google News對於新聞所作的相關分群，由於新聞事件已有相同的基準可以比較，因此可視為分類並採用分類評估標準。評估的指標含精確率(Precision)、招回率(Recall)以及F-measure，精確率代表正確預測的百分比，招回率則代表捕捉到正確分類的百分比，F-measure則是針對前述兩指標綜合而成的評估指標。在系統現有的效能下，兩者通常容易呈現負相關的成長，故透過F-measure可以同時衡量精確率與招回率之平衡，精確率與招回率之關係如圖3-5，相關公式與定義如下。

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{(公式11)}$$

$$\text{Recall} = \frac{TP}{TP+TN} \quad \text{(公式12)}$$

$$\text{F} - \text{measure} = \frac{2\times Precision\times Recall}{Precision+Recall} \quad \text{(公式13)}$$

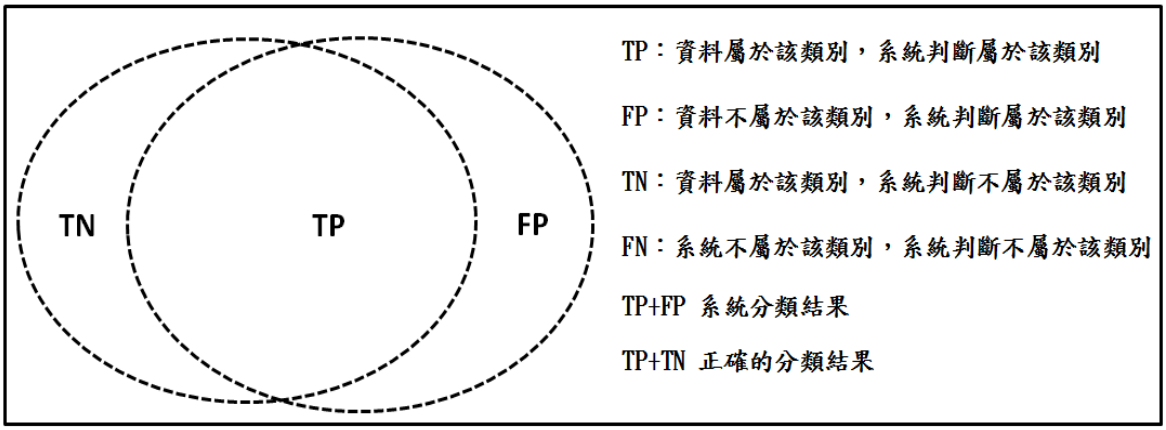

圖3-5評估標準示意圖資料來源：本研究整理

第七節　新聞來源與特性

本研究以 Google News 台灣版為新聞資料來源，其內容不但整合了台灣四大報系電子報(中時、自由、聯合、蘋果)，更廣納了許多其他華文新聞來源。同時，亦利用 Google News 中「相關新聞」的分群結果作為評估分群效果的依據，其「相關新聞」的內容是對於各新聞列出有關報導，涵蓋時間則依據事件長短有所不同，符合本研究對於新聞事件的定義。儘管，無法得知 Google News 所做的分群是否完全正確，但可以知道的是 Google News 所提供的內容是較廣為大眾所接受的。新聞的取樣則從 Google News 中九大類別隨機抽取 62 事件，過濾掉重複的新聞後共含 742 篇有效新聞，各類別事件與新聞分布如表 3-2。

表 3-2Google News 各類別事件與新聞數

類別	台灣	科技	社會	娛樂	財經	國際	運動	兩岸	健康
事件數	9	9	9	8	7	7	7	3	3
新聞數	118	79	140	67	61	96	99	30	52

資料來源：本研究整理

在這 742 篇新聞中，平均每篇新聞含 361.74 個詞彙(Term)，各新聞與同事件內(群內)其他新聞的平均 cosine 相似度為 0.2396，與不同事件(群外)其他新聞的平均 cosine 相似度為 0.013719，由此可知以新聞內容來說，在經過字詞權重處理後，同一事件的新聞彼此相關程度遠大於不同事件的新聞，可作為新聞事件偵測追蹤時相似度門檻值的參考。

第四章 實驗結果

第一節　基準點建立

RTD-based kNN 的主要概念在於先計算並儲存各文件與基準點的文件距離，進而在取前 k 個最相近文件時透過這些參考距離來減少運算量。文件基準點的概念是在空間中建立出一份虛擬的文件，而如何建立出這份文件才能使參考距離有最好的效果成為 RTD-based kNN 很重要的議題。在文字向量空間中，文件距離的範圍由 0 至$\sqrt{2}$，分別代表完全相同與完全不同。為了讓距離的參考有意義，基準點與各文件間必須要有共同出現的詞彙距離才會小於$\sqrt{2}$，因此建立基準點這份文件的詞彙必須由整個文件集所擁有的詞彙所構成。為了比較不同基準點對於 RTD-based kNN 的分群結果所造成的影響，本研究實驗了幾種基準點建立策略，除了隨機挑選外，更利用各種計算詞彙權重的指標做考量，建立策略如下：

1. 取文件集內 df (Document Frequency)前 n 高的詞彙
2. 取文件集內 tfc (tfidf 正規化)前 n 高的詞彙
3. 隨機抽取文件作為基準點

在 k 值為 15，事件偵測門檻值為 0.2，文件相似門檻值為 0.15 的設定下，經過事件合併的處理後，表 4-1 與表 4-2 分別為以最高 df 的詞彙與以 tfc 最高的詞彙建立基準點的新聞偵測追蹤結果，表 4-3 則代表隨機抽取之文件作為基準點的分群結果。由三種策略的結果看來，三種策略的結果差距並不大，以 df 前 n 高的字彙建立之基準點平均 F-measure 為 85.37%，tfc 前 n 高的字彙建立之基準點平均 F-measure 為 84.12%，隨機文件建立之基準點平均 F-measure 為 84.30%，其中結果較為突出的，分別為 tfc 前 250 高的詞彙(87.41%)與 df 前 1000 高的詞彙(86.68%)。

再觀察這兩種詞彙標準的分布情形，df 前 2000 高的詞彙分布如圖 4-1，其中前 1000 高的詞彙 df 值大致在 0.86 以上，df 前 700 高的詞彙 df 值甚至接近 1。而 tfc 前 2000 高的詞彙分布如圖 4-2，可以發現前 250 高的詞彙 tfc 值大致介於 0.4 以上。比較以 df 建立之基準點與以 tfc 建立之基準點可以發現，雖然新聞事件偵測追蹤結果的最佳結果在以 tfc 為基準點建立策略之中，但以 df 為基準點建立策略的 F-measure 的表現則較為平均(圖 4-3)，因此在後續幾節的實驗中將以 df 前 1000 高為基準點建立策略。

表 4-1 以最高 df 的詞彙建立之基準點資料來源：本研究整理

編號	基準點策略	Precision	Recall	F-measure	Time(second)
1	df 前 250 高	83.56%	83.56%	83.56%	8498.97
2	df 前 500 高	89.94%	81.94%	85.75%	8366.41
3	df 前 750 高	89.94%	81.94%	85.75%	11503.20
4	df 前 1000 高	89.29%	84.23%	86.68%	7881.91
5	df 前 1500 高	88.86%	81.67%	85.11%	8678.17
	平均	**88.32%**	**82.68%**	**85.37%**	**8985.73**

資料來源：本研究整理

表 4-2 以最高 tfc 的詞彙建立之基準點

編號	基準點策略	Precision	Recall	F-measure	Time(second)
1	tfc 前 250 高	86.84%	88.01%	87.41%	6080.71
2	tfc 前 500 高	83.49%	83.15%	83.32%	7138.38
3	tfc 前 750 高	88.13%	85.04%	86.56%	6456.48
4	tfc 前 1000 高	78.29%	82.61%	80.39%	12011.79
5	tfc 前 1500 高	80.64%	85.31%	82.91%	11384.06
	平均	**83.48%**	**84.82%**	**84.12%**	**8614.29**

資料來源：本研究整理

表 4-3 以隨機文件建立之基準點

編號	基準點策略	Precision	Recall	F-measure	Time(second)
1	隨機文件	79.92%	85.31%	82.42%	8293.22
2	隨機文件	81.81%	85.44%	83.59%	9236.14
3	隨機文件	90.06%	81.81%	85.73%	7633.80
4	隨機文件	85.75%	85.18%	85.46%	9625.11
5	隨機文件	77.17%	83.83%	80.36%	7811.53
	平均	**82.94%**	**84.31%**	**84.51%**	**8519.96**

資料來源：本研究整理

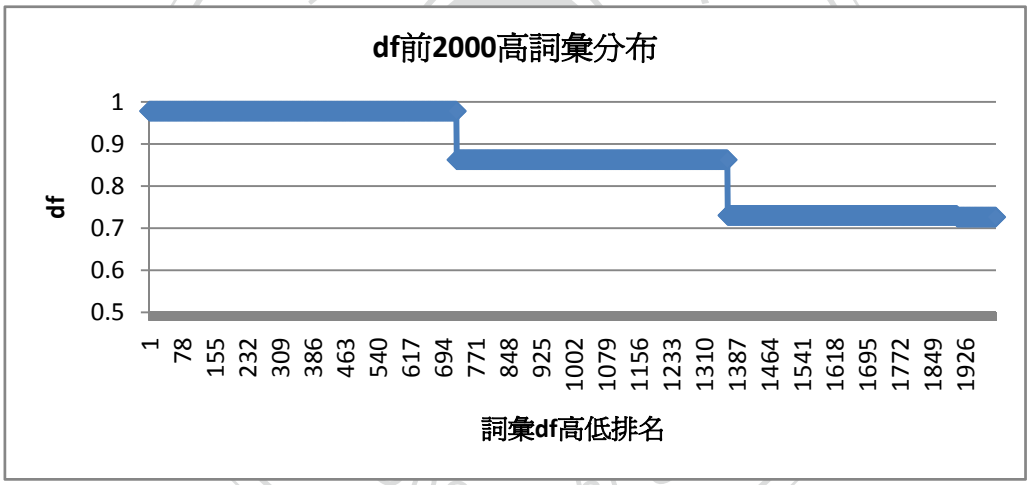

圖4-1df前2000高詞彙分布資料來源：本研究整理

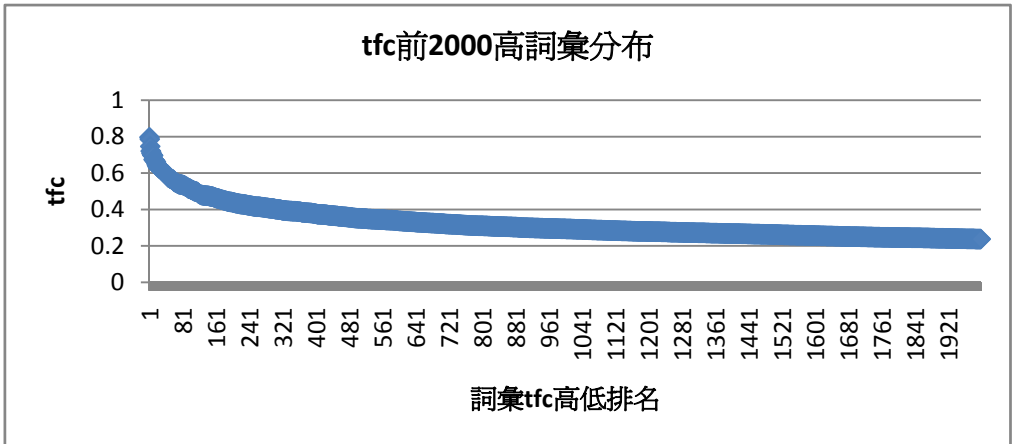

圖4-2 tfc前2000高詞彙分布資料來源：本研究整理

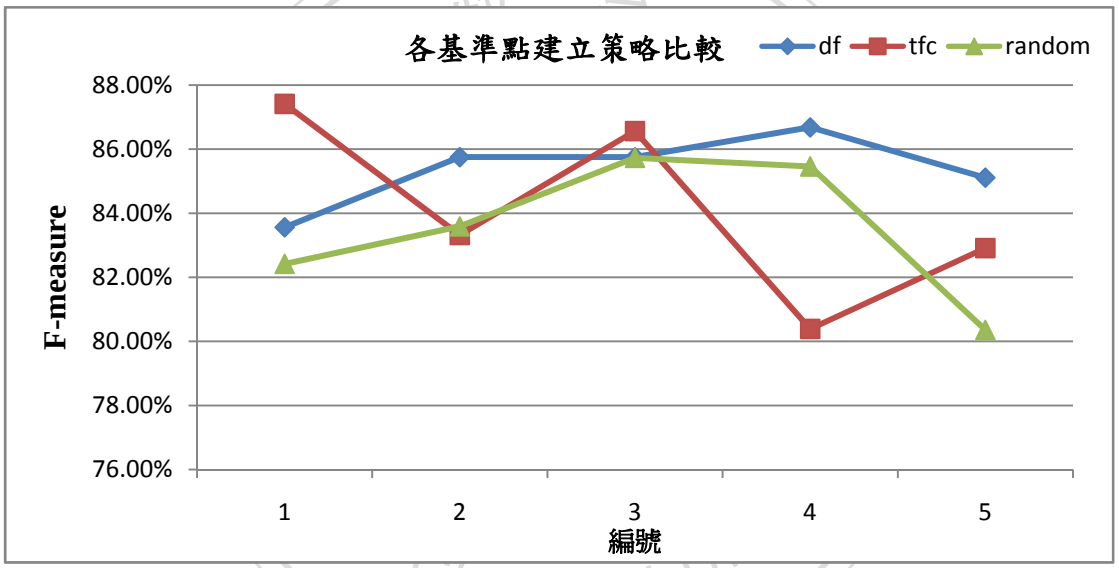

圖4-3 各基準點建立策略比較資料來源：本研究整理

第二節　事件偵測門檻值

新聞事件偵測的目的在於判斷新進新聞是否可能屬於目前已存在的事件中，亦或是自行成立為新事件。在新聞事件的偵測過程中，公式10所計算出的相關分數若小於事件偵測門檻值，則再繼續透過事件追蹤來判斷所屬事件。文獻探討曾提到相關的研究事件偵測門檻值大約都設定在0.15~0.23間，因此本研究事件偵測門檻值範圍界定於0.15至0.225，從中實驗四個區間的不同組合，分別為0.15、0.175、0.2、0.225，k為15時事件合併前結果如表4-4，事件合併後結果如表4-5；k為30時事件合併前結果如表4-6，事件合併後結果如表4-7。

從k為15合併前時各事件偵測門檻值的結果來看(見表4-4)，事件偵測門檻值為0.175與0.15時平均F-measure較高(分別為71.50%與71.46%)，事件門檻值為0.25時平均F-measure(65.12%)則較遠低於其他三者(差距達5%以上)；進一步觀察可發現，在事件門檻值為0.225、文件相似門檻值也是0.225時分群結果最差(65.12%)，所耗費時間幾乎高於其他門檻的平均值兩倍之多。在經過合併之後(見表4-5)，各事件偵測門檻值的分群結果有明顯改善，平均F-measure由69.75%提升到85.92%，所需時間則從平均5147.5秒增加到8579.86秒，但事件偵測門檻值為0.225的分群結果與效率同樣受到文件相似門檻值為0.225時的影響(79.64%)，降低了整體平均。

k值提高到30時，在事件合併前，分群結果較好的事件門檻值為0.175與0.15時，較差的則在事件門檻值為0.225時，但整體來說各事件偵測門檻值的F-measure差異並不大(皆在約2%之內)，所耗費的時間也較為平均。在經過事件合併後，平均F-measure由78.88%提升到87.34%，所需時間則從平均5921.88秒增加到8556.45秒，各事件門檻值間的F-measure差距亦不大，皆位於86%至88%之間。

表 4-4k=15 各事件偵測門檻合併前結果

k=15 (合併前)							
事件偵測門檻	文件相似門檻	Precision	Recall	F-measure	Time (second)	Average F-measure	Average Time(second)
0.225	0.225	87.35%	35.44%	50.43%	9469	**65.12%**	5785.5
	0.2	91.54%	56.87%	70.16%	4555		
	0.175	91.52%	56.74%	70.05%	4561		
	0.15	89.83%	57.14%	69.85%	4557		
0.2	0.225	90.26%	56.20%	69.27%	4824	**70.95%**	4722.25
	0.2	91.72%	58.22%	71.23%	4730		
	0.175	91.75%	58.50%	71.44%	4640		
	0.15	90.22%	59.70%	71.86%	4695		
0.175	0.225	90.26%	56.20%	69.27%	5199	**71.50%**	4878
	0.2	91.72%	58.22%	71.23%	4995		
	0.175	88.33%	61.19%	72.29%	4702		
	0.15	88.21%	62.53%	73.19%	4616		
0.15	0.225	90.26%	56.20%	69.27%	5658	**71.46%**	5204.25
	0.2	91.72%	58.22%	71.23%	5391		
	0.175	87.98%	61.19%	72.18%	4942		
	0.15	87.90%	62.67%	73.17%	4826		
Average F-measure : 69.75% ; Average Time : 5147.5 seconds							

資料來源：本研究整理

表4-5k=15各事件偵測門檻合併後結果

k=15 (合併後)							
事件偵測門檻	文件相似門檻	Precision	Recall	F-measure	Time (second)	Average F-measure	Average Time(second)
0.225	0.225	88.74%	72.24%	79.64%	16095.11	**84.38%**	10116.45
	0.2	89.59%	82.35%	85.81%	8232.96		
	0.175	89.70%	83.29%	86.37%	8109.06		
	0.15	90.80%	81.13%	85.69%	8028.65		
0.2	0.225	88.52%	82.08%	85.17%	8546.62	**86.31%**	8100.39
	0.2	89.74%	83.69%	86.61%	8095.06		
	0.175	89.77%	83.96%	86.77%	7877.95		
	0.15	89.29%	84.23%	86.69%	7881.91		
0.175	0.225	88.52%	82.08%	85.17%	8925.86	**86.66%**	7922.08
	0.2	89.77%	83.96%	86.77%	8314.05		
	0.175	89.65%	85.18%	87.35%	7603.27		
	0.15	89.52%	85.18%	87.29%	6845.13		
0.15	0.225	88.52%	82.08%	85.17%	9335.49	**86.32%**	8180.51
	0.2	89.77%	83.96%	86.77%	8649.52		
	0.175	88.48%	84.90%	86.66%	7815.55		
	0.15	88.38%	85.04%	86.68%	6921.48		
Average F-measure : 85.92% ; Average Time : 8579.86 seconds							

資料來源：本研究整理

表4-6k=30各事件偵測門檻合併前結果

k=30 (合併前)							
事件偵測門檻	文件相似門檻	Precision	Recall	F-measure	Time (second)	Average F-measure	Average Time(second)
0.225	0.225	91.30%	66.44%	76.91%	5532	**77.16%**	5528.5
	0.2	91.95%	66.17%	76.96%	5471		
	0.175	91.37%	67.12%	77.39%	5545		
	0.15	90.60%	67.52%	77.37%	5566		
0.2	0.225	91.33%	66.71%	77.10%	5910	**78.94%**	5665.25
	0.2	92.36%	68.46%	78.64%	5695		
	0.175	90.39%	71.02%	79.55%	5556		
	0.15	89.18%	73.32%	80.47%	5500		
0.175	0.225	91.36%	66.98%	77.29%	6378	**79.83%**	5953
	0.2	92.36%	68.46%	78.64%	6189		
	0.175	87.77%	75.47%	81.59%	5609		
	0.15	88.92%	75.74%	81.80%	5636		
0.15	0.225	91.36%	66.98%	77.29%	7236	**79.59%**	6540.75
	0.2	86.69%	75.47%	80.69%	6801		
	0.175	92.36%	68.46%	78.64%	6102		
	0.15	87.65%	76.55%	81.73%	6024		
Average F-measure :78.88%; Average Time : 5921.88 seconds							

資料來源：本研究整理

表4-7k=30各事件偵測門檻合併後結果

k=30 (合併後)							
事件偵測門檻	文件相似門檻	Precision	Recall	F-measure	Time	Average F-measure	Average Time(second)
0.225	0.225	90.19%	84.23%	87.11%	9005.20	**87.50%**	8671.69
	0.2	89.93%	85.44%	87.63%	8744.09		
	0.175	90.14%	85.04%	87.52%	8197.46		
	0.15	91.51%	84.23%	87.72%	8740.02		
0.2	0.225	90.26%	84.91%	87.50%	9292.40	**87.96%**	8321.37
	0.2	90.06%	86.66%	88.32%	8160.29		
	0.175	87.97%	87.74%	87.85%	7998.00		
	0.15	89.01%	87.33%	88.16%	7834.77		
0.175	0.225	88.89%	86.25%	87.55%	9777.72	**86.36%**	8322.90
	0.2	90.08%	86.93%	88.48%	8160.29		
	0.175	80.96%	88.81%	84.70%	7703.89		
	0.15	81.55%	88.14%	84.72%	7649.71		
0.15	0.225	88.89%	86.25%	87.55%	10643.77	**87.54%**	8909.84
	0.2	90.08%	86.93%	88.48%	9246.62		
	0.175	85.14%	88.81%	86.94%	8244.15		
	0.15	85.88%	88.54%	87.19%	7504.82		
Average F-measure :87.34%; Average Time : 8556.45 seconds							

資料來源：本研究整理

第三節　文件相似門檻值

RTD-based kNN利用相對文件距離預先排序的概念選擇前k個最近鄰，但由於文件距離不具有方向性，有可能選到的文件雖然與目標文件的參考距離相近，但實際上相似度卻很低。如圖4-4中，假設雖然文件A與文件B與基準點(Base)的距離同樣為1，但兩者與目標文件Data的相似度卻是A > B ($\theta a < \theta b$)，因此在篩選出與目標文件參考距離相近的文件後，仍必須將其與目標文件比較Cosine相似度，同時利用文件相似門檻過濾掉距離相近但相似度差距大的文件(如圖4-4中的文件B)。

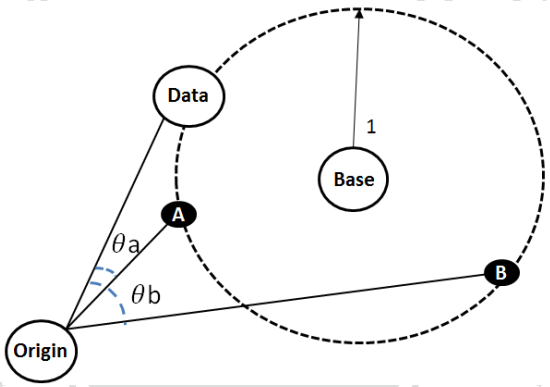

圖4-4文件相似門檻值示意圖資料來源：本研究整理

從實驗的結果來看，在k為15事件合併前，整體F-measure為69.76%，文件相似門檻值為0.15時F-measure最高(72.02%)，但與門檻值為0.175的時(71.49%)差距不到1%，最低的F-measure與最長的運算時間皆在門檻值0.225時(64.56%)。在經過事件合併之後，整體F-measure為85.92%，文件相似門檻值0.2(86.49%)、0.175(86.79%)、0.15(86.59)的結果皆差距不到1%，合併前後運算時間則都是0.15時最低。當k提高到30後，未經事件合併之前整體F-measure為78.88%，F-measure最高是在門檻值0.15之時(80.34%)，此時運算時間亦最短。合併後的整體F-measure提高到87.34%，最高F-measure則在文件相似門檻值為0.2時(88.23%)。

表4-8k=15各文件偵測門檻合併前結果

k=15 (合併前)							
文件偵測門檻	事件相似門檻	Precision	Recall	F-measure	Time	Average F-measure	Average Time(second)
0.225	0.225	87.35%	35.44%	50.43%	9469	**64.56%**	6287.5
	0.2	90.26%	56.20%	69.27%	4824		
	0.175	90.26%	56.20%	69.27%	5199		
	0.15	90.26%	56.20%	69.27%	5658		
0.2	0.225	91.54%	56.87%	70.16%	4555	**70.96%**	4917.75
	0.2	91.72%	58.22%	71.23%	4730		
	0.175	91.72%	58.22%	71.23%	4995		
	0.15	91.72%	58.22%	71.23%	5391		
0.175	0.225	91.52%	56.74%	70.05%	4561	**71.49%**	4711.25
	0.2	91.75%	58.50%	71.44%	4640		
	0.175	88.33%	61.19%	72.29%	4702		
	0.15	87.98%	61.19%	72.18%	4942		
0.15	0.225	89.83%	57.14%	69.85%	4557	**72.02%**	4673.5
	0.2	90.22%	59.70%	71.86%	4695		
	0.175	88.21%	62.53%	73.19%	4616		
	0.15	87.90%	62.67%	73.17%	4826		
Average F-measure :69.76; Average Time : 5147.5 seconds							

資料來源：本研究整理

表4-9k=15各文件偵測門檻合併後結果

k=15 (合併後)							
文件偵測門檻	事件相似門檻	Precision	Recall	F-measure	Time	Average F-measure	Average Time(second)
0.225	0.225	88.74%	72.24%	79.64%	16095.11	**83.79%**	10725.77
	0.2	88.52%	82.08%	85.17%	8546.62		
	0.175	88.52%	82.08%	85.17%	8925.86		
	0.15	88.52%	82.08%	85.17%	9335.49		
0.2	0.225	89.59%	82.35%	85.81%	8232.96	**86.49%**	8322.90
	0.2	89.74%	83.69%	86.61%	8095.06		
	0.175	89.77%	83.96%	86.77%	8314.05		
	0.15	89.77%	83.96%	86.77%	8649.52		
0.175	0.225	89.70%	83.29%	86.37%	8109.06	**86.79%**	7851.46
	0.2	89.77%	89.96%	86.77%	7877.95		
	0.175	89.65%	85.18%	87.35%	7603.27		
	0.15	88.48%	84.91%	86.67%	7815.55		
0.15	0.225	90.80%	88.13%	85.70%	8028.65	**86.59%**	7419.29
	0.2	89.29%	84.23%	86.68%	7881.91		
	0.175	89.52%	85.18%	87.29%	6845.13		
	0.15	88.38%	85.04%	86.68%	6921.48		
Average F-measure :85.92%; Average Time : 8579.86 seconds							

資料來源：本研究整理

表4-10k=30各文件偵測門檻合併前結果

k=30 (合併前)							
文件偵測門檻	事件相似門檻	Precision	Recall	F-measure	Time	Average F-measure	Average Time(second)
0.225	0.225	91.30%	66.44%	76.91%	5532	**77.15%**	6264
	0.2	91.33%	66.71%	77.10%	5910		
	0.175	91.36%	66.98%	77.29%	6378		
	0.15	91.36%	66.98%	77.29%	7236		
0.2	0.225	91.95%	66.17%	76.96%	5471	**78.73%**	6039
	0.2	92.36%	68.46%	78.64%	5695		
	0.175	92.36%	68.46%	78.64%	6189		
	0.15	86.69%	75.47%	80.69%	6801		
0.175	0.225	91.37%	67.12%	77.39%	5545	**79.29%**	5703
	0.2	90.39%	71.02%	79.55%	5556		
	0.175	87.77%	75.47%	81.59%	5609		
	0.15	92.36%	68.46%	78.64%	6102		
0.15	0.225	90.60%	67.52%	77.37%	5566	**80.34%**	5681.5
	0.2	89.18%	73.32%	80.47%	5500		
	0.175	88.92%	75.74%	81.80%	5636		
	0.15	87.65%	76.55%	81.73%	6024		
Average F-measure :78.88%; Average Time : 5921.88							

資料來源：本研究整理

表4-11k=30各文件偵測門檻合併後結果

k=30 (合併後)							
文件偵測門檻	事件相似門檻	Precision	Recall	F-measure	Time	Average F-measure	Average Time(second)
0.225	0.225	90.19%	84.23%	87.11%	9005.20	**87.43%**	9679.77
	0.2	90.26%	84.91%	87.50%	9292.40		
	0.175	88.89%	86.25%	87.55%	9777.72		
	0.15	88.89%	86.25%	87.55%	10643.77		
0.2	0.225	89.93%	85.44%	87.63%	8744.09	**88.23%**	8577.82
	0.2	90.06%	86.66%	88.32%	8160.29		
	0.175	90.08%	86.93%	88.48%	8160.29		
	0.15	90.08%	86.93%	88.48%	9246.62		
0.175	0.225	90.14%	85.04%	87.52%	8197.46	**86.75%**	8035.88
	0.2	87.97%	87.74%	87.85%	7998.00		
	0.175	80.96%	88.81%	84.70%	7703.89		
	0.15	85.14%	88.81%	86.94%	8244.15		
0.15	0.225	91.51%	84.23%	87.72%	8740.02	**86.95%**	7932.33
	0.2	89.01%	87.33%	88.16%	7834.77		
	0.175	81.55%	88.14%	84.72%	7649.71		
	0.15	85.88%	88.54%	87.19%	7504.82		
Average F-measure :87.34%; Average Time : 8556.45							

資料來源：本研究整理

第四節　k值的提升

在前兩節的實驗中，除了針對事件偵測門檻值與文件相似門檻值做測試之外，也分別實驗了k值為15與k值為30時對於各個門檻值的影響，在各種不同的新聞事件偵測門檻值與文件相似度門檻值的情況下，事件合併前k值增加所造成的影響整理如表4-12，在事件合併後k值增加所造成的影響整理如表4-13。在表4-12與表4-13中，各事件偵測門檻值的平均F-measure表示的是在該事件偵測門檻值之下，各文件相似門檻值的平均結果。比較兩表的結果可以發現，若不考慮將事件合併，k值的提升有助於F-measure的提高，當k由15提升30時，各門檻值的平均F-measure由69.76%提高到78.87%(各門檻值平均增加13.2%)，運算時間則由平均5147.5秒增加到5921.88秒(各門檻值平均增加15.81%)，但無論k為15或30，平均的F-measure皆不到80%。若運算過後將事件合併，則k值的提升對於平均F-measure影響程度不高，k取15時為85.92%，k取30時為87.34%，各門檻值平均僅提高了1.67%。其運算時間亦差別不大，平均只增加了不到1個百分點(0.61%)。

表4-12事件合併前k值增加的影響

事件合併前						
事件偵測門檻	k為15時平均F-measure	k為30時平均F-measure	F-measure成長率	k為15時平均運算時間	k為30時平均運算時間	時間增加率
0.225	65.12%	77.16%	18.49%	5785.50	5528.50	-4.44%
0.2	70.95%	78.94%	11.26%	4722.25	5665.25	19.97%
0.175	71.50%	79.83%	11.65%	4878.00	5953.00	22.04%
0.15	71.46%	79.59%	11.38%	5204.25	6540.75	25.68%
平均	**69.76%**	**78.88%**	**13.20%**	**5147.5**	**5921.88**	**15.81%**

資料來源：本研究整理

表4-13事件合併後k值增加的影響

事件合併後						
事件偵測門檻	k為15時平均F-measure	k為30時平均F-measure	F-measure成長率	k為15時平均運算時間	k為30時平均運算時間	時間增加率
0.225	84.38%	87.50%	3.70%	10116.45	8671.69	-14.28%
0.2	86.31%	87.96%	1.91%	8100.39	8321.37	2.73%
0.175	86.66%	86.36%	-0.35%	7922.08	8322.90	5.06%
0.15	86.32%	87.54%	1.41%	8180.51	8909.84	8.92%
平均	**85.92%**	**87.34%**	**1.67%**	**8579.86**	**8556.45**	**0.61%**

資料來源：本研究整理

第五節　合併前後的差別

經過初步的實驗結果發現，由於新聞群聚成事件後，其事件的特徵會更加明顯，因此相似程度門檻值與前述兩個門檻值相比相對較高，本研究將事件合併的門檻值定為0.25，代表兩個事件的質心若相似度大於0.25時，則將其視為同一事件作合併。合併的策略則採反覆式合併，即合併完一輪後繼續檢查是否還可以合併，直到所有事件的相似度皆小於門檻值為止。如同前兩節之實驗的結果，k為15時事件合併的差異如表4-14，k為30時事件合併的差異如表4-15。從結果可以看出RTD-based kNN在合併後的F-measure明顯增加，k為15時各門檻值平均增加23.31%，k為30時平均增加10.75%；但效果的提升相對的合併也要花上不少時間，k為15時增加了66.51%的運算時間，k為30時增加了44.94%的運算時間。

表4-14k=15事件合併前後的影響

k=15						
距離偵測門檻	合併前平均 F-measure	合併後平均 F-measure	F-measure 成長率	合併前平均運算時間	合併後平均運算時間	時間增加率
0.225	65.12%	84.38%	29.58%	5785.50	10116.45	74.89%
0.2	70.95%	86.31%	21.65%	4722.25	8100.39	71.54%
0.175	71.50%	86.66%	21.21%	4878.00	7922.08	62.40%
0.15	71.46%	86.32%	20.79%	5204.25	8180.51	57.19%
平均	**69.76%**	**85.92%**	**23.31%**	**5147.5**	**8579.86**	**66.51%**

資料來源：本研究整理

表4-15k=30事件合併前後的影響

k=30						
距離偵測門檻	合併前平均 F-measure	合併後平均 F-measure	F-measure 成長率	合併前平均運算時間	合併後平均運算時間	時間增加率
0.225	77.16%	87.50%	13.40%	5528.50	8671.69	56.85%
0.2	78.94%	87.96%	11.43%	5665.25	8321.37	46.88%
0.175	79.83%	86.36%	8.18%	5953.00	8322.90	39.81%
0.15	79.59%	87.54%	9.99%	6540.75	8909.84	36.22%
平均	**78.88%**	**87.34%**	**10.75%**	**5921.88**	**8556.45**	**44.94%**

資料來源：本研究整理

第六節　與kNN的比較

為了與kNN進行比較，本研究透過調整kNN的事件偵測門檻值並衡量其於各門檻值時的表現，再分別就新聞偵測追蹤結果與運算時間進行比較。下表中列出以kNN進行新聞事件偵測追蹤的結果，表4-16為k設15，表4-17為k設30。

表4-16k為15時kNN新聞事件偵測追蹤結果

kNN (k=15)				
事件偵測門檻	Precision	Recall	F-measure	Time(seconds)
0.225	80.14%	78.84%	84.11%	11337
0.2	87.52%	84.10%	85.77%	11516
0.175	87.38%	85.85%	86.61%	11483
0.15	87.29%	89.76%	88.50%	11381
平均	**85.58%**	**84.64%**	**86.25%**	**11429.25**

資料來源：本研究整理

表4-17k為30時kNN新聞事件偵測追蹤結果

kNN (k=30)				
事件偵測門檻	Precision	Recall	F-measure	Time(seconds)
0.225	93.47%	77.22%	84.58%	11387
0.2	89.79%	81.81%	85.61%	11382
0.175	89.60%	83.56%	86.47%	12440
0.15	87.34%	89.22%	88.27%	11272
平均	**90.05%**	**82.95%**	**86.23%**	**11620.25**

資料來源：本研究整理

圖4-5為k取15時kNN與RTD-based kNN於各個事件偵測門檻值的F-measure比較，參考圖4-7的平均差距來看，kNN與未經事件合併的RTD-kNN平均差距達16.49%，而kNN與經過事件合併的RTD-kNN則是在各個事件偵測門檻值表現各有優劣，但整體而言差異皆不大(最大差距2.18%)。當k為30時，kNN與未經過事件合併的RTD-based kNN差距小於k為15時，兩者平均差為7.35%(圖4-7)。再以經過事件合併的RTD-based kNN比較(圖4-6)，其與kNN最大差距為門檻值2.25時(2.92%)，其他門檻值的結果也相差不遠。總結比較的結果，未經過事件合併的RTD-based kNN F-measure在k為15或30皆低於kNN；經過事件合併的RTD-based kNN則是在k為30時F-measure較kNN高，兩者平均起來，經過事件合併的RTD-based kNN則高於kNN 0.39%。

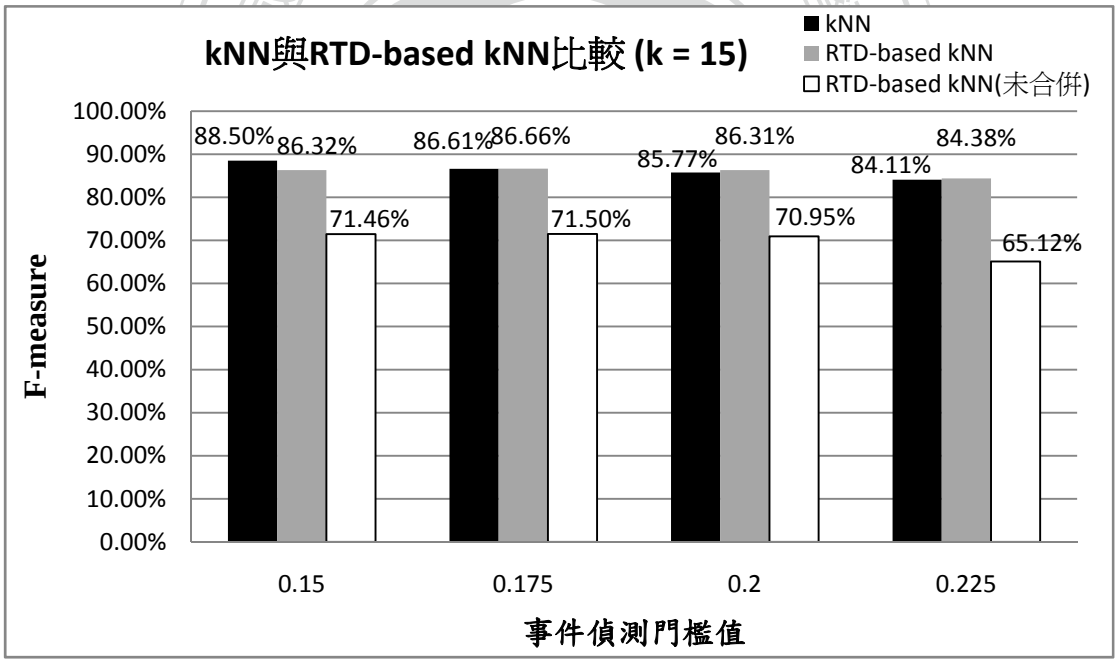

圖4-5kNN與RTD-based kNN於k為15時F-measure比較資料來源：本研究整理

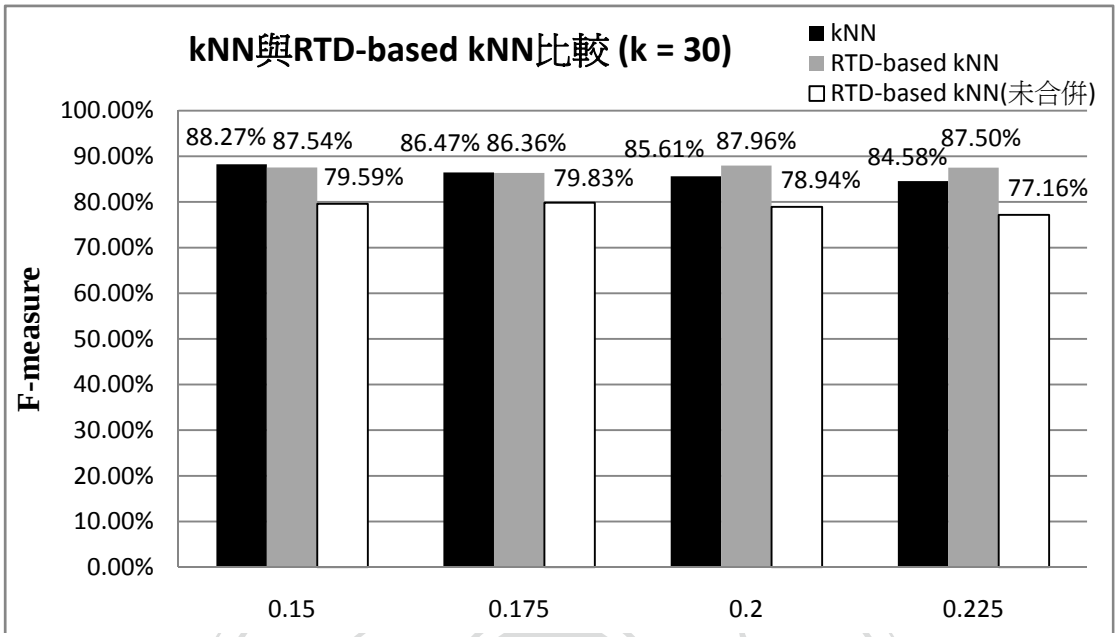

圖4-6kNN與RTD-based kNN於k為30時F-measure比較資料來源：本研究整理

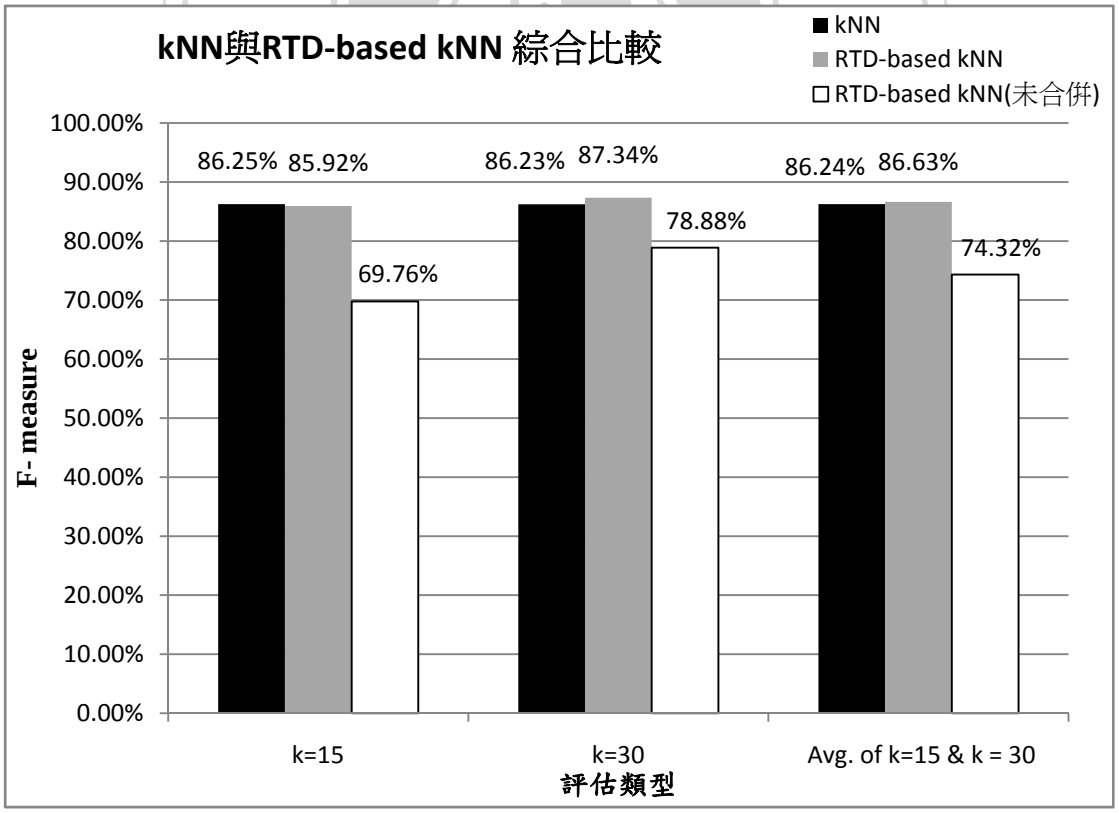

圖4-7 kNN與合併前RTD-based kNN的平均F-measure比較資料來源：本研究整理

最後，將經過事件合併後的 RTD-based kNN 與 kNN 的 F-measure 作統計檢定，在 95%的信心水準下，檢驗兩者結果是否有顯著性的差異。比較的內容是以兩者在各個參數下的 F-measure 做綜合比較。以 RTD-based kNN 來說，本研究實驗了 k 為 15 或 k 為 30 時的事件偵測門檻值(0.225、0.2、0.175、0.15)下，各個文件相似門檻值(0.25、0.2、0.175、0.15)的結果；由於 kNN 只有事件偵測門檻值，為了跟 RTD-based kNN 的結果作成對抽樣的比較，在相同事件偵測門檻值下，與 RTD-based kNN 不同文件相似門檻值比較的 kNN F-measure 視為相同表現，檢定內容共 32 筆資料整理如表 4-18。

虛無假設為 RTD-based kNN 與 kNN 的平均 F-measure 相等，對立假設為 RTD-based kNN 與 kNN 的平均 F-measure 不相等。經過計算之後，檢定統計量 Z^0 不在拒絕域中，因此不拒絕虛無假設，即 RTD-based kNN 與 kNN 的 F-measure 並沒有顯著性的差別，檢定過程如下：

$$H_0: \mu_{RTD-based\ kNN} = \mu_{kNN}$$

$$H_1: \mu_{RTD-based\ kNN} \neq \mu_{kNN}$$

$\bar{d}_i$ =0.003859, δ_d=0.020004, n=32, $\alpha = 0.05$

$$Z^0 = \frac{\bar{d}_i}{\delta_d/\sqrt{n}} = \frac{0.003859}{0.020004/\sqrt{32}} = 1.09367$$

R.R. : $Z^0 < Z_{0.025} = -1.96\ or\ Z^0 > Z_{0.975} = 1.96$

$Z_{0.025} < Z^0 < Z_{0.975}$，不拒絕 H_0

表 4-18 RTD-based kNN 與 kNN F-measure 檢定內容

	kNN		RTD-based kNN			RTD 與 kNN F-measure 差距 (d_i)
	事件偵測門檻值	F-measure	事件偵測門檻值	文件相似門檻值	F-measure	
k=15	0.225	0.8411	0.225	0.225	0.7964	-0.0447
		0.8411		0.2	0.8581	0.017
		0.8411		0.175	0.8637	0.0226
		0.8411		0.15	0.8569	0.0158
	0.2	0.8577	0.2	0.225	0.8517	-0.006
		0.8577		0.2	0.8661	0.0084
		0.8577		0.175	0.8677	0.01
		0.8577		0.15	0.8669	0.0092
	0.175	0.8661	0.175	0.225	0.8517	-0.0144
		0.8661		0.2	0.8677	0.0016
		0.8661		0.175	0.8735	0.0074
		0.8661		0.15	0.8729	0.0068
	0.15	0.885	0.15	0.225	0.8517	-0.0333
		0.885		0.2	0.8677	-0.0173
		0.885		0.175	0.8666	-0.0184
		0.885		0.15	0.8668	-0.0182
k=30	0.225	0.8458	0.225	0.225	0.8711	0.0253
		0.8458		0.2	0.8763	0.0305
		0.8458		0.175	0.8752	0.0294
		0.8458		0.15	0.8772	0.0314
	0.2	0.8561	0.2	0.225	0.875	0.0189
		0.8561		0.2	0.8832	0.0271
		0.8561		0.175	0.8785	0.0224
		0.8561		0.15	0.8816	0.0255
	0.175	0.8647	0.175	0.225	0.8755	0.0108
		0.8647		0.2	0.8848	0.0201
		0.8647		0.175	0.847	-0.0177
		0.8647		0.15	0.8472	-0.0175
	0.15	0.8827	0.15	0.225	0.8755	-0.0072
		0.8827		0.2	0.8848	0.0021
		0.8827		0.175	0.8694	-0.0133
		0.8827		0.15	0.8719	-0.0108

資料來源：本研究整理

再以 kNN 與 RTD-based kNN 的運算時間進行比較，在 k 為 15(圖 4-8)或是 k 為 30(圖 4-9)的條件下，RTD-based kNN 在事件合併的前後運算時間皆少於 kNN，由圖 4-10 可以看出當 k 為 15 與 k 為 30 時，事件合併前運算時間分別降低了 54.94%與 48.95%，事件合併後運算時間分別降低了 29.56%與 26.70%，因此無論是否有經過事件的合併，RTD-based kNN 所需的運算時間至少少於 kNN25%以上。

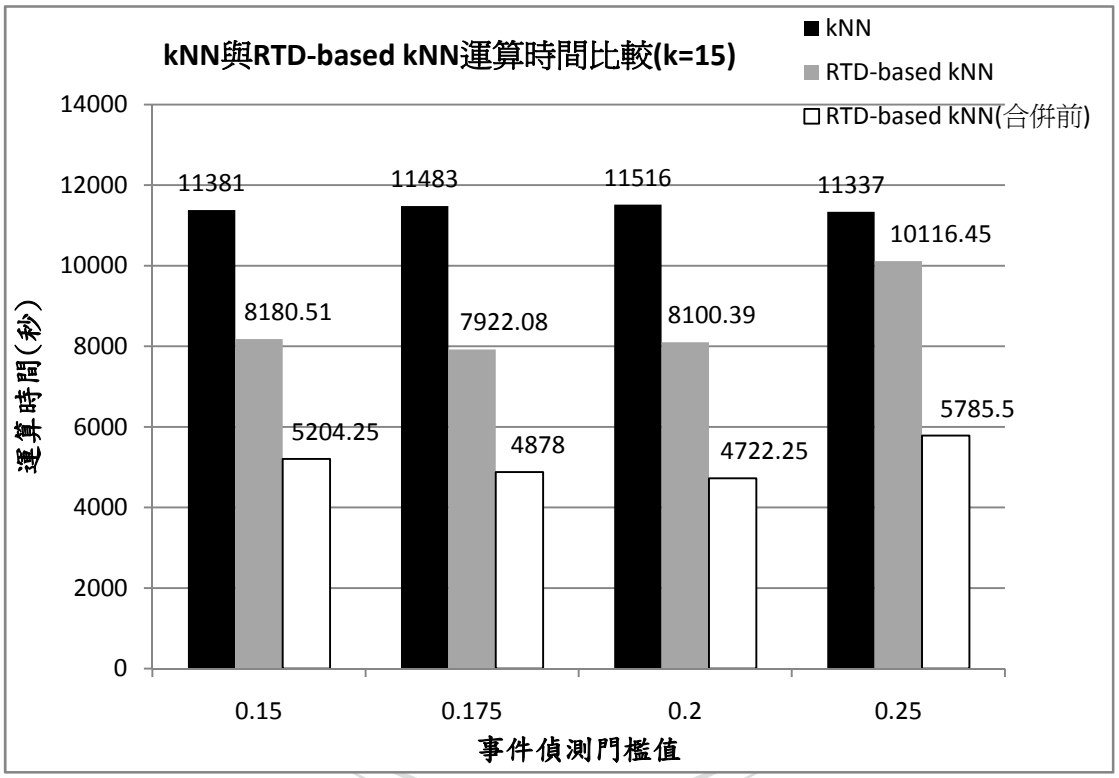

圖 4-8 k 為 15 時 kNN 與合併前 RTD-based kNN 運算時間比較資料來源：本研究整理

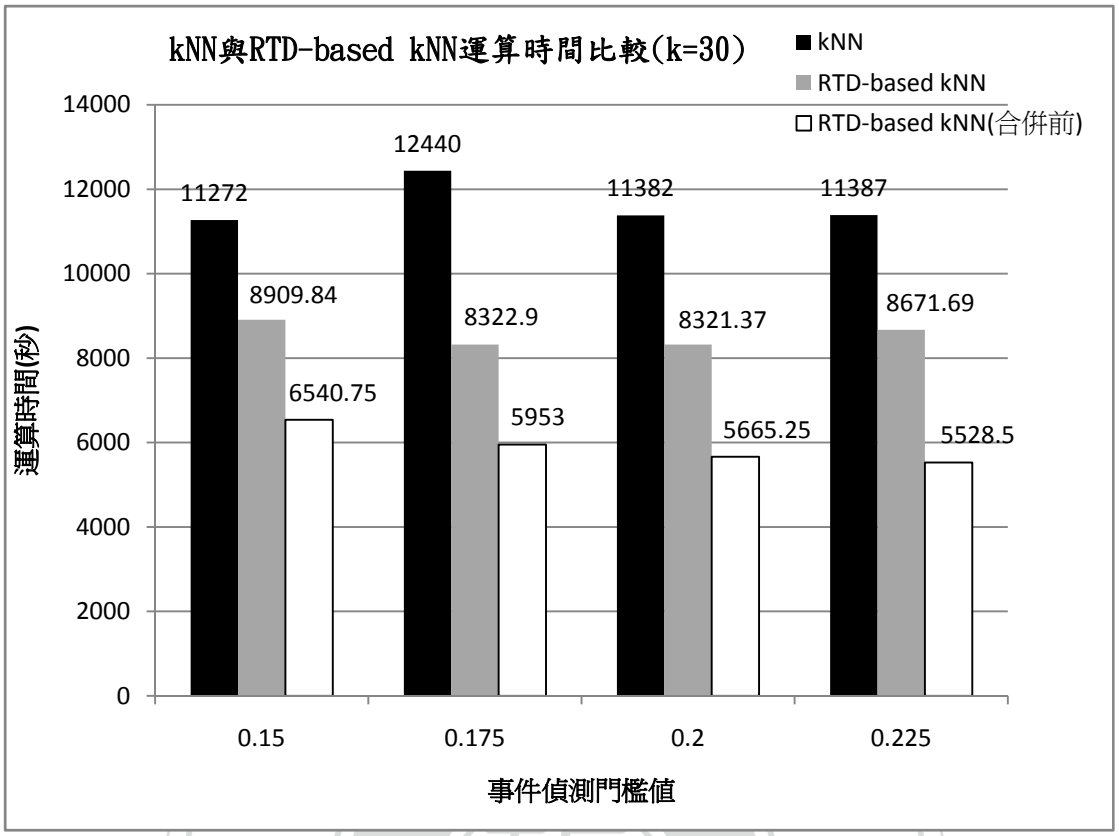

圖 4-9 k 為 30 時 kNN 與合併前 RTD-based kNN 運算時間比較資料來源：本研究整理

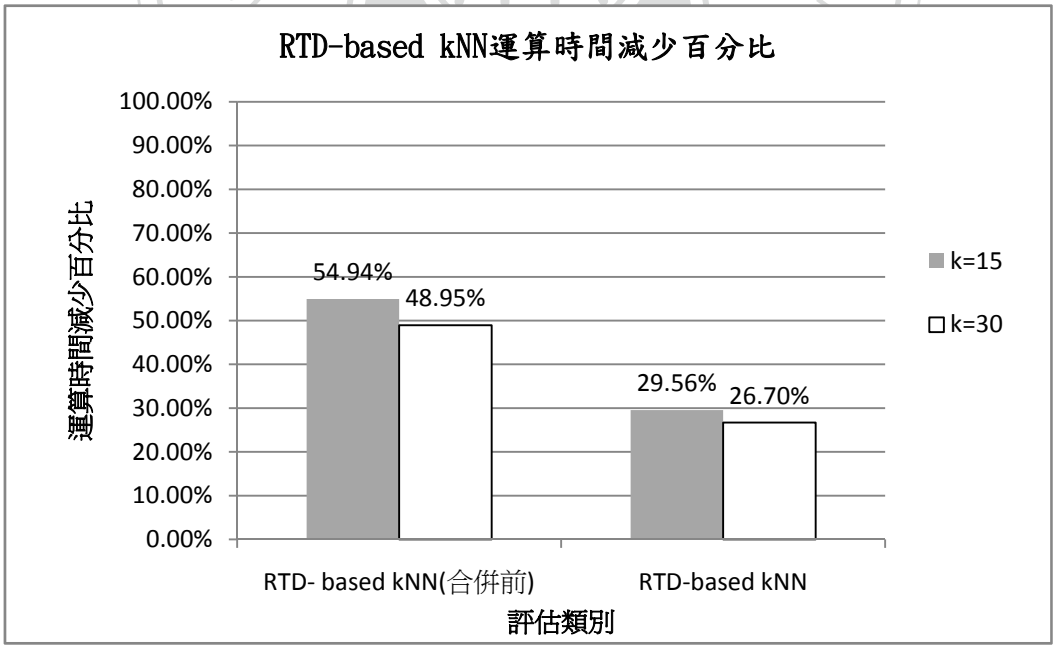

圖 4-10RTD-based kNN 運算時間減少百分比資料來源：本研究整理

表 4-19RTD-based kNN 與 kNN 之事件偵測追蹤綜合比較

	F-measure 平均增加	運算時間平均減少
事件合併前	-16.04%	51.95%
事件合併後	0.45%	28.13%

資料來源：本研究整理

綜合比較 RTD-based kNN 與 kNN 在新聞事件偵測追蹤的效果，在各項參數結果的平均下，若未經過事件的合併，RTD-based kNN 的 F-measure 平均降低了 16.4%，但運算時間減少了 51.95%之多；若經過事件的合併，在 F-measure 的表現上 RTD-based kNN 則較 kNN 高出 0.45%，運算時間減少了 28.13%。由此可歸納出 RTD-based kNN 最佳的分群表現並不遜色於 kNN，運算時間上更大大少於 kNN 的所需時間。

第五章結論與未來展望

第一節結論與建議

本研究提出利用在向量空間建立一文件距離基準點的方式，增加新聞分群時的效率，並整合於新聞事件追蹤的應用。由於文件參考距離僅需在初始時計算並可重複被參考，使得在新聞事件追蹤時相似度的比較次數大幅減少，並藉此降低整體運算時間。總結整體實驗過程可分為以下結果：

1. 基準點的建立：雖然以 df、tfc 或隨機文件為基礎建立的基準點平均 F-measure 差距並不大，但由於基準點是作為所有文件距離參考之標準，因此必須一開始選定後即確立，否則所有參考距離皆必須重新運算，建立時仍建議以 F-measure 變動程度最小的 df 作為基準點建立策略。再者，由於 df 較高的詞彙代表其出現在整個文件集中的次數較頻繁，表示其為新聞文章寫作時較常出現的詞彙，因此當新文件進來時這些字彙再次出現的機率較高，容易跟以 df 為建立基礎的基準點比較出距離上的差距，使得參考距離的遠近關係較為明顯。

2. 事件偵測與文件相似門檻值：在事件合併前，k 取 15 時 F-measure 的最佳表現在事件偵測門檻值為 0.175 與文件相似門檻值 0.15 時(73.19%)，k 取 30 時亦同(81.80%)。若將事件進行合併，k 取 15 時 F-measure 最佳表現在事件偵測門檻值為 0.175 與文件相似門檻值為 0.15 時(87.29%)，k 取 30 則在事件偵測門檻值為 0.175 與文件相似門檻值為 0.2 時(88.48%)，或是事件偵測門檻值為 0.15 與文件相似門檻值為 0.2 時(88.48%)。總結起來，除了事件偵測

門檻值為 0.225 與文件相似門檻值為 0.225 的設定外，其餘組合的 F-measure 結果與運算時間差距皆不算大。因此以 RTD-based kNN 處理新聞事件群聚時，可以先由事件偵測門檻值 0.175 作為初始進而調整。

3. k 值的提高：k 值的提高無論是在事件合併前後皆有助於分群效果的提升，但同樣的也會增加運算時間。在未經過事件合併的情況下，k 值的提高對於運算時間與 F-measure 值的影響程度皆較高；若運算後有進行事件合併的處理，k 值的提升影響則程度不大。

4. 事件的合併：由於 RTD-based kNN 並非比較完所有文件後才選出前 k 個最近鄰，使得在最近鄰的選取上，選到的可能是相對距離較近的最近鄰，而非所有文件中的最近鄰，造成容易將事件分成各個小事件的結果，因此本研究也提出了持續將事件合併的概念。經過 RTD-based kNN 對於新聞事件分群後，事件的合併雖然會增加整體運算時間，但對於分群的效果有大幅度的提升，使得 RTD-based kNN 分群結果可以與 kNN 的 F-measure 無顯著性差異。

總體而言，RTD-based kNN 在不進行事件合併的情況下，新聞偵測追蹤的效果是較 kNN 為差的，但所需要的計算時間遠低於 kNN 達五成以上(本研究實驗平均減少 51.95%的時間)。若在事件偵測追蹤後將事件合併，RTD-based kNN 的新聞偵測追蹤效果無顯著上的差距，運算時間仍是領先於 kNN 有不小的差距(本研究實驗平均減少 28.13%的時間)。因此，若對於新聞事件偵測追蹤的需求較為急切或是運算資源有限，在結果仍有一定的可信程度下，未將事件合併的 RTD-based kNN 可以提供一個快速篩選的機制，提供較即時的服務或是時間較為急迫的決策參考。若對於事件偵測追蹤的結果有較為嚴謹的需求，經過事件合併後的 RTD-based kNN 可以在較短的時間內提供與 kNN 相近的結果。

第二節未來展望

對資料進行分類分群是人類在做決策時的一個很基本的思考依據，而分類法中的 kNN 利用了自然界中物以類聚的特性處理分類分群，其幾乎不需訓練的特性使得 kNN 很容易應用在不同的領域中，若將其整合於分群流程，亦可作為分群的方法之用。由於本研究所評估的標準為 Google News 所分群的結果，並建立在「分群結果越接近 Google News 則越佳」的假設下做評估，未來可利用其他評估方法作為比較與改善的依據，如專家審查、統計分析等

本研究證實了利用 RTD-based kNN 處理新聞群聚為事件可以得到良好的效果並大幅降低處理時間，對於其他領域需要同類群聚的資料理論上依然適用，因此可思考應用於對於時間需求較為急迫的文字探勘運用，如即時性的媒體監測、整合於對股票市場的當日預測和交易資料的即時反應處理等，但必須對於所分析的資料特性有深入的瞭解才能得到較有價值的效果。

此外，將新聞群聚為事件的作法有助於幫助使用者瞭解完整事件的發生經過，並且透過不同媒體的報導彙整出事件的真實發展，若能結合其他文字探勘技術，透過將新聞事件的內容自動彙整出重點摘要，勢必能大大的減少閱聽人的負擔並增進其資訊獲取的效率。

最後，資料探勘與文字探勘的目的同樣是在大量的資料中彙整出資訊，但也因為其處理的資料量龐大，往往需要很長的運算時間。改進方法通常可以分為兩大部份，一部分是透過運算資源的改善，如利用多執行緒、多核心，甚至是透過分散式運算、導入於雲端運算等等；另一部份則是透過方法論上的改善，本研究所提出的即是後者，因此可思考透過與前者的結合來減少更多 RTD-based kNN 所需要的處理時間。

參考文獻

中文部分

1. 巫啟台(2002)。文件之關聯資訊萃取及其概念圖自動建構 (碩士論文)，國立成功大學資訊工程學系碩士論文。

2. 陳克健、陳正佳、林隆基(1986)。中文語句的研究－斷詞與構詞。中央研究院技術報告，TR-86-006。

3. 陳昱絃 (2007)。以螞蟻演算法探勘推薦系統上之分類規則，國立成功大學工程科學系碩士論文。

4. 陳崇正 (2009)。應用網路書籤與VSM相似度演算法於強化實踐社群的形成，國立中正大學資訊工程研究所碩士論文。

5. 黃孝文 (2010)。雲端運算服務環境下運用文字探勘於語意註解網頁文件分析之研究，國立政治大學資訊管理研究所碩士論文。

6. 戴尚學 (2003)。運用事件偵測與追蹤技術於中文多文件摘要之研究，國立雲林科技大學資訊管理系碩士論文。

7. 謝邦昌 (2006)。資料採礦與商業智慧，台北市：鼎茂圖書出版股份有限公司。

英文部分

1. Allan ,J. , Papka, R. & Lavrenko , V. (1998). On-line New Event Detection and Tracking. In Proceedings of ACM SIGIR, pp37-45.

2. Chen, K. J., Kiu, S. H. (1992). Word Identification for Mandarin Chinese Sentences. Fifth International Conference on Computational Linguistics, pp.101-107.

3. Cover, T.M., Hart, P.E. (1967). Nearest Neighbor Pattern Classification, IEEE Transaction on Information Theory. v.IT-13 n.1, pp.21-27.

4. Fayyed, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). The KDD Process of Extracting Useful Knowledge from Volumes of Data. , Communication of the ACM, v.39, pp. 27-34.

5. Fan, C.K., Tsai, W.H. (1998). Automatic Word Identification in Chinese Sentences by the Relaxation Technique. Computer Proceeding of Chinese and Oriental Languages, pp.33-56.

6. Feldman, R., Dagan, I. (1995). Knowledge Discovery in Textual Database(KDT). Proceedings of the first ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.112-117.

7. Han , Jiawei, Kamber, Micheline (2006). Data Mining: Concepts and Techniques

8. Jain, A.K., Murty, M.N. & Flynn, P.J.(1999). Data Clustering,A Review. ACM Computing Surveys, v.31 n.3, pp.264-323.

9. Joachims , T.(1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the European Conference on Machine Learning Springer, pp. 137–142.

10. Krishnapuram, Raghu,Joshi, Anupam,Yi, Liyu (2001). Low-Complexity Fuzzy Relational Clustering Algorithm for Web Mining. IEEE Transactions on Fuzzy System, v.9 n.4, pp.595-607.

11. Li, B.Y., Lin, S., Sun, C.F. & Sun, M.S. (1991).A Maximal Matching Automatic Chinese Word Segmentation Algorithm using Corpus Tagging for Ambiguity Resolution. R.O.C. Computational Linguistics Conference, Taiwan, pp.135-146.

12. MacQueen, J. B.(1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, pp.281-297.

13. Berry, M., Linoff, G. (2000). Mastering Data Mining, The Art & Science of Customer Relationship Management, Wiley Publishing.

14. Nie, Jian-Yun, Brisebois, Martin & Ren, Xiaobo (1996). On Chinese Text Retrieval. Conference Proceedings of SIGIR, pp.225-233.

15. Popescu, A.(2001). Implementation of Term Weighting in a Simple IR System. Personal course project, University of Helsinki.

16. Roiger, Richard, Geatz, Michael (2003). Data Mining: A Tutorial Based Primer. Addison Wesley Higher Education.

17. Rousseeuw, P.J., Kaufman, L., Trauwaert, E.(1996). Fuzzy Clustering using Scatter Matrices. Computational Statistics and Data Analysis, v 23, pp.135-151.

18. Salton, G., McGill, M. (1983). Introduction to Modern Information Retrieval, New York: McGraw-Hill.

19. Salton, G., Wong, A., Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM, v.18 n.11, pp.613-620.

20. Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, v.34 n.1, pp.1-47.

21. Singh, L., Scheuermann , P. & Chen , B. (1997). Generating Association Rules from Semi-Structured Documents Using an Extended Concept Hierarchy. ACM IKM, pp.193-200.

22. Sproat, R, Shih , C., 1990. A Statistical Method for Finding Word Boundaries in Chinese Text. Computer Processing of Chinese and Oriental Languages, pp. 336-351.

23. Teng, W.-G., Lee, H.-H.(2007). Collaborative Recommendation with Multi-Criteria Ratings. Journal of Computers (Special Issue on Data Mining), v.17 n.4, pp.69-78.

24. Yang, Yiming (1997), An Evaluation of Statistical Approaches to Text Categorization. Technical Report CMU-CS-97-127, Carnegie Mellon University.

25. Yang, Y., Pierce, T. & Carbonell, J.(1998). A Study on Retrospective And On-Line Event Detection. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.28-36.

26. Yang , Yiming, Lin, Xin (1999). A Re-examination of Text Categorization Methods. Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp.12-29.

27. Yang, Y., Carbonell, J.G., Brown, R., Pierce, T., Archibald, B. T. & Liu, X. (1999). Learning Approaches for Detecting and Tracking News Events. IEEE Intelligent Systems, v.14 n.4, pp.32-43.

28. Yang, Y., Ault, T., & Pierce, T. (2000). Improving Text Categorization Methods for Event Tracking. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.65-72.

29. You , Jia-Ming, Chen, Keh-Jiann (2006). Improving Context Vector Modelsby Feature Clustering for Automatic Thesaurus Construction , Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.

附錄A：Google News新聞來源與事件

本附錄列出從資料來源 Google News 所取得的其中兩篇事件(Event)，包含新聞編號(依下載順序排序)與新聞標題。

Google Event 1

1 : 共諜滲台不減國安局：查辦絕不手軟
2 : ・週刊點名異動蔡得勝…
4 : 國安局長將異動？蔡得勝斥為創意小說
6 : 蔡得勝任期屆滿下？綠委點名前憲兵司令李翔宙上
8 : 週刊爆_國安局長將換__蔡得勝：創意性小說
9 : 蔡得勝：國安統合運作良好
10 : 周刊說國防部鬥國安局軍方否認
12 : 國安局長蔡得勝將調整職務？國防部否認部長高華柱介入

Google Event 2

13 : 免簽遭矮化立委：馬衝「百國免簽」出賣主權
14 : 免簽疑遭矮化外長：持續交涉
15 : 百國免簽綠委怒:用矮化換的
16 : 綠委再爆馬政府出賣主權換免簽？
17 : 綠委疑：主權換免簽__外部：絕對沒有_
18 : 綠委疑：主權換免簽外部：絕對沒有
19 : 綠委批給我免簽國稱台灣屬於中國
20 : 馬政府笑納？綠委：克羅埃西亞給免簽，把台灣列中國一省
21 : 免簽疑遭矮化外交部積極交涉

附錄B：RTD-based kNN群聚事件結果

本附錄列出當k為30，事件偵測門檻0.175，文件偵測門檻值0.2，經事件合併後的RTD-based kNN分群結果其中兩事件，包含新聞編號(依下載順序排序)與新聞標題。

RTD-based kNN Event 1

1：共諜滲台不減國安局：查辦絕不手軟
2：・週刊點名異動蔡得勝…
4：國安局長將異動？蔡得勝斥為創意小說
6：蔡得勝任期屆滿下？綠委點名前憲兵司令李翔宙上
9：蔡得勝：國安統合運作良好
10：周刊說國防部鬥國安局軍方否認
12：國安局長蔡得勝將調整職務？國防部否認部長高華柱介入

RTD-based kNN Event 2

13：免簽遭矮化立委：馬衝「百國免簽」出賣主權
14：免簽疑遭矮化外長：持續交涉
15：百國免簽綠委怒:用矮化換的
16：綠委再爆馬政府出賣主權換免簽？
17：綠委疑：主權換免簽__外部：絕對沒有_
18：綠委疑：主權換免簽外部：絕對沒有
19：綠委批給我免簽國稱台灣屬於中國
20：馬政府笑納？綠委：克羅埃西亞給免簽，把台灣列中國一省
21：免簽疑遭矮化外交部積極交涉