資訊檢索與知識探勘

曾元顯
輔仁大學圖書資訊學系
tseng@lins.fju.edu.tw

- 簡介
- 主題檢索
- 關聯分析
- 自動分類
- 自動歸類
- 自動摘要
- 時間事件分析
- 系統展示
- 結語

文件資訊探勘

- （text mining, knowledge discovery in text）意義：
  - 擷取隱晦、有用、未被發掘、有潛在價值的資訊或知識
  - 互動、反覆的過程來探索文件庫以發現新的、有趣的訊息或規律
  - 依賴人工解讀結果，使發現的訊息變成有用的資訊或知識
- 具體項目（工具）：
  - 資訊檢索、擷取、關聯、摘要、歸類、分類、時間事件分析
- 應用：
  - 資訊搜尋、知識萃取、知識管理、犯罪分析、案例追蹤
- 使用的技術：
  - 資料庫管理技術、統計、機器學習、人工智慧、資訊視覺化、資訊科學、圖書館學、簡單的文字處理工具、處理流程的彈性串連
- 考量的因素（面臨的挑戰）：
  - 要能處理大量資料
  - 要能快速回應、提供互動性
  - 多面向、多維度的分析
  - 高階、視覺化的使用介面

主題檢索

- 意義：
  - 根據使用者的資訊需求，找出符合需求之文件或文字
- 應用：
  - 前案檢索、相似案例檢索（技術專利、法院判例）
  - 案例比對
  - 案例關聯
  - 案例分類
  - 案例歸類
  - 案例時間事件分析
- 使用技術：
  - information retrieval、NLP、machine learning

自動索引

- 意義：
  - 對文件、詞彙進行分析、轉換、組織
  - 便於有效率的高階運用
- 應用：
  - 檢索、關聯、分類、歸類、摘要、趨勢分析等工作的核心運算與結構
- 使用的技術：
  - Hash, trie, B-tree, …
  - fast sorting, data compression, …
  - Stemming, stopwords, ngrams, …
  - Authority control, thesaurus, topic map, ontology, …
  - Natural language processing, machine learning, …
  - File format parsing, language identification, …
  - Security control, user control, access control, robot, …
  - Support for different OSs, DBMS, platforms, …

資訊檢索的問題

- 字串不匹配（vocabulary mismatch）：查詢詞與文件記載(或索引詞)不同
  - 同義：「筆記型電腦」vs「筆記本電腦」(形似)，「閣揆」vs「行政院長」
  - 廣狹義：「攜帶型」vs「掌上型」，
- 使用者需求差異大：同樣的檢索詞，但相關的文件會因人而異
  - Known item search
    - 已知「作者」、「人名」；已知文件內的字串：「嘿嘿嘿」、「這我不聽他的」
  - Unknown item search：
    - 無法精確表達查詢字串：人名、地名、機構名、專有名詞、特定領域名稱
    - 不知如何表達查詢字串：「晶圓代工的發展前景」、「電視廣告對兒童的影響」
- 領域需求差異大：斷詞需求、查詢功能
  - 「中醫工會」：「治虛寒，五香、加八角、加薑，加味米酒…」
  - 「社文中心」：「D'eng Xiaoping's legacy 」
- 資料本身不一致、不乾淨，檔案格式差異大
  - 民83年 vs 1994、年代日期格式不同
  - 異常標點符號、字碼、dash 、single quote
  - 資料誤植、OCR 雜訊文字
  - Data cleaning is required
- 文件格式、資訊架構、作業環境
  - 需要解各種檔案格式：HTML、XML、Office、PDF、ZIP、EMAIL、BBS …
  - 資訊來源與權限控管：File systems、DBMS、Web、Notes …

檢索系統的五個面向

可從這五點瞭解及預測核心檢索系統的表現
- （未考慮文件格式、權限控管、資訊架構）

- 索引詞模式
- 檢索模式
- 權重模式
- 索引檔結構
- 查詢模式

索引詞模式

- 檢索系統建構索引詞所依據的方法
- 關係系統比對查詢字串的能力
- 「以詞彙爲主」（word-based）
  - 前組合
  - 詞庫更新不及、或涵蓋範圍不足，會有找不到資料的情形
- 「以字元爲主」（character-based）
  - 後組合
  - 「中國」會索引成「中」及「國」
  - 比對到含「中國」、「國中」或「開發中的國家」等文件
- 「N-gram」索引法
  - N-gram爲文件中任意N個連續字元
  - 「中國社會」N=2時產生「中國」、「國社」、「社會」三個索引詞
  - 可排除或降低「字元法」中類似「中國」與「國中」的字串順序問題
  - 可省去「詞彙法」中維護詞庫的煩惱

檢索模式

- 系統比對檢索條件與相關文件的依據
- 「布林模式」
  - 優點：速度快、檢索者可完全控制檢索過程，並預測檢索結果
    - 對需求明確的檢索（如明確的作者名、題名）非常有效
  - 缺點：結果沒照符合程度排序、一般使用者較難表達複雜查詢條件
- 「向量模式」
  - 轉換文件及查詢語句到向量空間後比對相似度，常用餘弦夾角（cosine）
  - 例：「李遠哲院長」、「李院長遠哲」兩詞，以（李，遠，哲，院，長，李遠，遠哲，哲院，院長，李院，長遠）爲維度，得（1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0）與（1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1）
  - 兩向量餘弦夾角爲7/9 = 0.78，在最高值爲1的度量中，相似度爲0.78
  - 可允許使用者輸入任意字串，查詢時不必受資料誤植、錯字、冗字的限制
  - 可概略稱爲「近似字串查詢」、「容錯查詢」、或是「模糊搜尋」（fuzzy search）、「近似自然語言查詢」或「自然語言查詢」
- 「機率模式」
  - 將查詢詞彙與相關文件的不確定性，以機率描述並加以運算
  - 亦可做到向量模式的查詢效果，兩者不同處在基本假設與運算模式

向量權重模式

- 指定索引詞與查詢詞權重的方式
- 權重因素：
  - term frequency (TF)
  - inverse document frequency (IDF)
  - document length (normalization)
  - positional information
  - Number of hyperlinks (in-wards or out-wards)
- 常用乘法原則將這些因素組合（不是很精確的作法）
- 查詢詞：詞長、詞頻
  - tf * (3w-1), where w is the length of the term
- 文件詞：詞頻、文長、文件篇數
  - TF*IDF = log(1+ tf) * log(N/df)
  - Document length normalization:
    - byte size for document terms vs Cosine

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

$$Sim(D_i, Q_j) = \frac{\sum_{k=1}^{t} d_{i,k} q_{j,k}}{\sqrt{\sum_{k=1}^{t} d_{i,k}^2} \sqrt{\sum_{k=1}^{t} q_{j,k}^2}}$$

索引檔結構

- 加快檢索的速度、影響檢索的成效
- 「反向索引檔」(inverted file)
  - 記錄每個索引詞及其出現文件的編號，可直接取得包含某索引詞的所有文件
- 「特徵檔」（signature file）
  - 將文件中編碼成0與1組成的特徵向量，檢索時，第一階段經特徵檔運算，過濾掉不可能的文件，第二階段把誤引（false drop）的文件剔除
  - 特色：可快速大量非相關文件的過濾
  - 索引建構速度快，「漸進式索引」（incremental indexing）製作容易
- 「隱含語意索引法」（latent semantic indexing）
  - 運用向量空間運算縮減索引詞維度，並關連相關文件的方法
  - 文件、檢索條件都以此轉換矩陣轉換到縮減的向量空間，再運算相似度
  - 特色：轉換後，相關的詞彙會經由文件所包含的內容而產生關連
- 特殊的「搜尋樹」
  - B 樹：精確比對、後切截檢索、範圍查詢
  - PAT樹
    - 後切截檢索、鄰近字串檢索、範圍查詢、最常出現的字串檢索，以及常規式檢索(regular expression search)等功能
    - 適合字典或辭典等較少更新的靜態資料庫

使用者查詢模式的進展

- Boolean model / 布林邏輯
- Ranking / 重要性排序
- Fuzzy search / 容錯式、 近似字串、 近似自然語言
- Relevance feedback / 相關回饋、 漸進式查詢、 範例查詢
- Information filtering / 資訊過濾
- Query by dialog / 個別化、 對話式查詢
- Query by voice / 語音檢索
- Query by natural language / 自然語言檢索
- Intelligent search agent / 時空無礙、 虛擬實境的檢索精靈

檢索的其他策略

- 相關詞提示(Term suggestion)
- 相關詞回饋 (Term relevance feedback)
- 查詢詞擴展 (Query expansion, relevance feedback)

相關詞提示與相關詞回饋

- 檢索成效，非常倚賴檢索詞的品質
- 從文件資料庫中擷取統計上重要的詞彙，作爲
  - 相關詞提示（term suggestion）：由互動方式挑取檢索詞
  - 相關回饋(relevance feedback)：檢出文件中挑取重要特徵回饋系統
    - 相關文件回饋（document relevance feedback）
    - 相關詞回饋（term relevance feedback）
- 相關回饋的優點 ：
  - 免除使用者選擇檢索語彙與設計查詢條件的細節，允許建構有用的檢索條件而不用對檢索環境及資料庫有深入瞭解；
  - 拆解檢索過程成一步步較小的步驟，可以逐漸逼近所要檢索的主題；
  - 提供一個控制的查詢修改過程，終端使用者僅需最少的訓練就可有效而合理的進行檢索
- 相關詞提示：
  - Altavista (LiveTopic, 1996, Java-based Interface)英文單字詞回饋
  - Excite : about 1997, keyword selling

關聯分析

- 詞彙關聯：索引典、標題表
- 文件關聯：歸類
- 概念關聯：分類

前言

- 檢索失敗的主要因素之一：「字彙不匹配問題」
  - 「查詢詞」與「索引詞」不相同的情況
  - 例:「筆記型電腦」與「筆記本電腦」,「行政院長」與「閣揆」
  - 改進方法：「查詢擴展」、「權威檔」、「索引典」
- 「查詢擴展」(query expansion)
  - 加入更多與查詢主題相關的詞彙，或更改查詢詞的權重
- 「權威檔」 (authority file)
  - 記錄及解決同義異名詞的工具
  - 索引或檢索時，將各種同義異名詞對應起來，視爲相同的詞彙處理

前言

- 「索引典」 (thesaurus)
  - 除同義詞外，還有紀錄廣義詞、狹義詞、反義詞、、相關詞等
  - 列舉主題詞彙，將詞彙間的語意或主題關係標示出來的知識庫
  - 查詢時，可互相推薦，以擴展或縮小查詢範圍，或提示相關概念的不同查詢用語
  - 例「攜帶型電腦」：「筆記型電腦」、「掌上型電腦」
  - 使檢索從「字串比對層次」，提升到「語意比對層次」
  - 人工製作索引典，準確度高，但召回率低、成本大、建構速度慢、事先選用的詞彙可能與後續或其他新進的文件無關
  - 一般目的索引典運用在特定領域的文件檢索上，無法提升檢索效能
  - 針對每一種文獻領域製作索引典，耗時費力

前言

- 「共現索引典」(co-occurrence thesaurus)
  - 利用詞彙的「共現性」，自動建構「詞彙關聯」（term association）
  - 或稱「關聯詞庫」
  - 成本低、建構速度快、召回率高、與館藏文件用詞一致，但準確率低
  - 詞彙關係：主題相關，不一定語意相關
    - 例：「李登輝」與「康乃爾」、「中華電訊」與「ADSL」

研究方法

- 文獻探討、技術瞭解、優缺點分析、適用範圍分析
- 歸納重點
- 提出改進方法
- 實驗測試
- 成效比較
  - 不同研究之間的比較
  - 同一研究內，對照組之比較
- 提出適用情況與應用方向
- 持續評估與改進

相關研究：Salton ’89

- Salton 曾提出建構共現索引典的架構：
  - 算出各個詞彙間的相似度
    - 「相似度」：詞彙在各文件之間，共同出現的情形（或主題相似度）
    - 重要的索引詞彙，任兩詞彙皆拿來比對相似度
    - 計算量至少 M^2，M ：所有重要詞彙的個數
  - 依此相似度將詞彙**歸類**成「索引典類別」(thesaurus classes)（或「主題類別」）

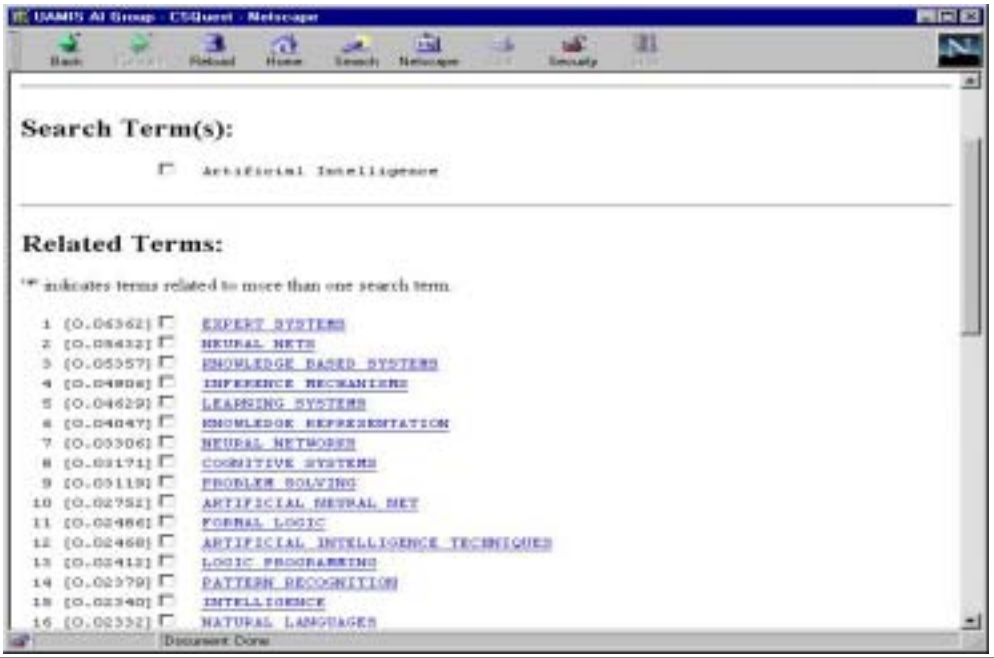
$T_j=(d_{1j}, d_{2j}, \ldots, d_{nj})$, n：所有文件的個數

$$sim(T_j, T_k) = \frac{\sum_{i=1}^{n} d_{ij} d_{ik}}{\sqrt{\sum_{i=1}^{n} d_{ij}^{2} \sum_{i=1}^{n} d_{ik}^{2}}}$$

相關研究：Salton ’89

- 歸類方式，主要有：
- Complete-link:
  - 一開始，每個詞彙（元素），都單獨視爲一類
  - 兩個類別之間的相似度，若超過某個門檻值，就結合並歸成同一類，如此重複歸類
  - 兩個類別之間的相似度，定義爲跨類別元素之間**相似度最低者**
  - 易產生多數個索引典類別（thesaurus class），但每類僅有少數個詞彙
- Single-link:
  - 同上述作法，但兩個類別之間的相似度，定義爲跨類別元素之間**相似度最高者**
  - 易產生少數個類別，但每類都有大量的詞彙
- 透過共現索引典的查詢擴展，檢索成效的召回率，通常可提升 10% 至 20%
- 小結：
  - 歸類運算量太大，運用在大量文件上，耗時長久

相關研究： Chen ’96



相關研究： Chen (JASIS ’95)

- 定義**非對稱**的詞彙相似度
- 詞彙 T_j 在文件 i 中的權重:

$$d_{ij} = tf_{ij} \times \log\left(\frac{N}{df_j} \times w_j\right)$$

- 詞彙 T_j 及 T_k 在文件 i 中的權重:

$$d_{ijk} = tf_{ijk} \times \log\left(\frac{N}{df_{jk}} \times w_j\right)$$

- Cluster_weight(T_j, T_k)

$$= \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \times \log\left(\frac{N}{df_k}\right) \Big/ \log(N)$$

- Cluster_weight(T_k, T_j)

$$= \frac{\sum_{i=1}^{n} d_{ikj}}{\sum_{i=1}^{n} d_{ik}} \times \log\left(\frac{N}{df_j}\right) \Big/ \log(N)$$

- 若T_j =「Artificial Intelligence」, w_j =2

相關研究： Chen (JASIS ’95)

- 從 4714 文件中 (共 8 MB), 產生了 1,708,551 個**詞對**（co-occurrence pairs ）
- 由於關聯**詞對**太多，每個詞，限制其關聯詞數最多100 個，如此刪除了 60% 的**詞對**，剩下 709,659 個**詞對**（由 7829 個不同的詞組成）
- 產生上述的詞對，在 Sun Sparc 工作站上要花 9.2 CPU 小時、磁碟空間 12.3 MB
- 成效評估：
  - 6個受試者，16 個預選的詞，請每個受試者先就每個詞，聯想出相關的詞彙；再從系統提示的關聯詞，判斷哪些是相關或不相關
  - 兩種結果比較，召回率分別爲 28.60% 與 61.89% ；精確率爲 77.08% 及 24.17%
- 小結：
  - 人工聯想精確率高、召回率低；機器產生關聯詞較多、準確度較低

相關研究： Sanderson and Croft (SIGIR’99)

- 概念階層的範例：[from Sanderson and Crofts’ paper]

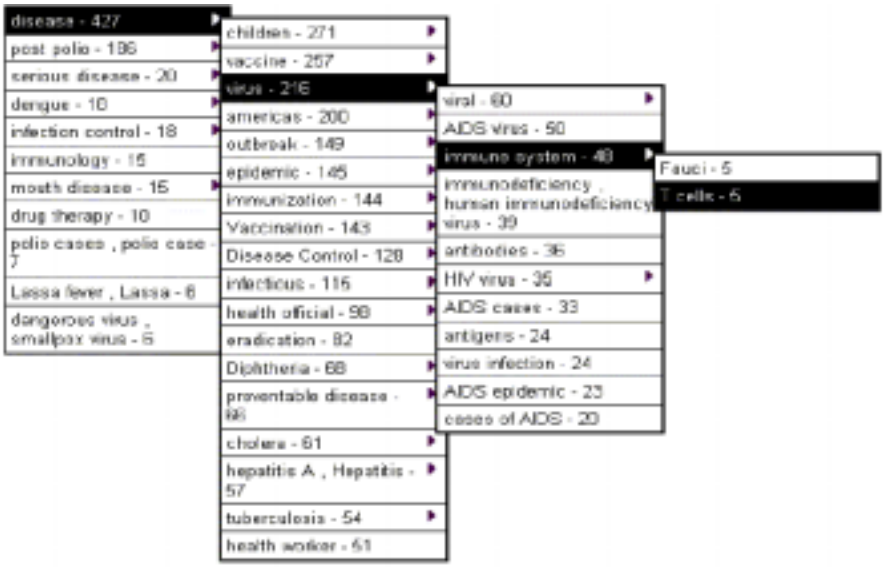


Figure 4: Second fragment of concept hierarchy from TREC topic 302

相關研究： Sanderson and Croft (SIGIR'99)

- 目的：從檢出的文件中自動產生**概念階層**（concept hierarchies），便利使用者瞭解檢出文件的大致內容
- 第一步：詞彙選擇（決定哪些詞彙要列在概念階層中）:
  - 來源 1: 檢索結果的前幾篇中比對程度較佳的段落裡，找出常常一起出現的詞彙
  - 來源 2: 每一篇檢出文件的最相關段落裡，取符合下列條件的詞彙：**(df_in_retrieved_set / df_in_collection) >= 0.1** 者
  - 平均從 TREC 的每個查詢結果的前 500 篇文件中，擷取出 2430 個詞
- 第二步：詞彙關聯分析:
  - 任意兩個詞都拿來做 **包含** 關係（subsumption relationship）比較：
  - **$P(T_j \mid T_k) = 1$ and $P(T_k \mid T_j) < 1$**, if T_j (較廣義的詞) **包含** T_k (較特定的詞)
  - 由於上述條件太嚴苛，放寬成： **$P(T_j \mid T_k) >= 0.8$ and $P(T_k \mid T_j) < 1$**, if T_j 包含 T_k
  - 平均每個查詢擷取出 200 **包含對**（subsumption pairs）
  - 由這些 **包含對** 產生 **概念階層**，即**包含者**爲**父節點**，**被包含者**爲其**子節點**

相關研究： Sanderson and Croft (SIGIR'99)

- 成效評估：測試**包含者**與**被包含者**的關聯程度（relatedness）
  - 由 8 個受試者判斷，67% **包含對**被判斷爲相關（interesting for further exploring）
  - 比較：51% **詞彙對**（隨意配對，而非用包含關係配對者）被判斷爲相關
- 小結：
  - 此方法在查詢時才進行，查詢反應時間會受影響
  - 提示的詞彙只限於檢索結果的前N篇，不是一個 全域索引典（global thesaurus）
  - 隨機配對，關聯度高，顯示詞彙選擇的重要性

關聯詞分析

- 先前的作法
  - 「共現性的單位」爲「文件」
  - 兩個詞彙在文件中距離越大，關係密切的可能性越低
  - 需要分析的**詞對個數多**，許多詞對的關聯**分析徒勞無功**
  - 計算量：M^2n，M:所有詞彙個數, n :所有文件個數
  - 例：n=10,000, M=10,000 (M=1000), 計算量：10^{12} (10^{10})
- 新的作法
  - 「共現性的單位」縮小到「段落」或「句子」
  - 需要分析的**詞對個數少**
  - 計算量：K^2Sn，K:文件關鍵詞數, S:文件句子數, n:同上
  - 例：n=10,000, K=30, S=20, 計算量：$6x10^6$

關聯詞分析：新的方法：[Tseng 2002]

- 主要分二個步驟：
  - 擷取個別文件的關鍵詞
  - 關鍵詞的關聯分析與累積
- 關鍵詞擷取
  - 關鍵詞：文件內有意義且具代表性的詞彙
  - 關鍵詞：呈現文件主題意義的最小單位
  - 各種文獻自動化處理的必要步驟。
  - 關鍵詞的認定是主觀的判斷，不利於電腦的自動處理
  - 「重複性」假設：
    - 如果文件探討某個主題，那麼應該會提到某些特定的字串好幾次
    - 具有客觀性、可自動處理
    - 假設簡單，可適用於不同領域

關聯詞分析：新的方法：[Tseng 2002]

- 第一步：詞彙選擇：
  - 每篇文件先用 **詞庫**（長詞優先法）斷詞
  - 再由**關鍵詞擷取演算法** 擷取關鍵詞（至少出現2次者）（包含新詞）
  - 以 **停用詞** 過濾擷取出的關鍵詞，並依詞頻（term frequency）高低排序
  - 選 詞頻最高的 N 個詞作關聯分析
- 第二步：詞彙關聯分析:
  - 每篇文件選出來的詞，以 DICE公式計算兩個詞彙的 權重wgt
  - **關聯詞** 的權重超過門檻值（1.0）者，才依下面公式累積其權重
  - **關聯詞** 的最後相似度定義爲：
    - 原方法：僅單純累加每對關聯詞的權重 $sim(T_j, T_k) = \sum_{i=1}^{n} wgt(T_{ij}, T_{ik})$
    - 新方法：加入 IDF (inverse document frequency) 及 **詞彙長度**

$$sim(T_j, T_k) = \frac{\log(w_k \times n / df_k)}{\log(n)} \times \sum_{i=1}^{n} wgt(T_{ij}, T_{ik})$$

關鍵詞自動擷取方法

比較：

- 詞庫比對法：詞庫需持續維護更新
- 統計分析法：容易遺漏統計特徵不足者
- 文法剖析法：需詞庫、詞性標記等資源與運算
  - 適合作爲關鍵詞的名詞片語少於 50% [Arppe 1995]

關鍵詞自動擷取方法 [Tseng 97, 98, 99, 2000]

- 找出最大重複出現字串（maximally repeated pattern）的演算法

- **token** : 一個中文字（character）或英文字（word）
- **n-token**: 輸入文字中，任意連續的 n tokens （與 n-gram 類似）

- 演算法三步驟：

  步驟一 : 轉換輸入文字成 2-token 串列

  步驟二 : 依合併規則重複合併 n-tokens 成 (n+1)-tokens，直到無法合併

  步驟三 : 依過濾規則，過濾不合法的詞彙

關鍵詞自動擷取過程範例

- 輸入文字: "**BACD**B**CD**A**BACD**", 假設 門檻值 = 1
- **步驟一** : 產生
  L = (BA:2 AC:2 CD:3 DB:1 BC:1 CD:3 DA:1 AB:1 BA:2 AC:2 CD:3)
- **步驟二**: token 合併 :
  第一次 :合併 L 成 L1= (BAC:2 ACD:2 BAC:2 ACD:2)
  丟掉: (BA:2 AC:2 CD:3 DB:1 BC:1 DA:1 AB:1 BA:2 AC:2 CD:3)
  留住 : (CD:3)
  第二次 : 合併 L1 成 L2 = (BACD:2 BACD:2)
  丟掉 : (BAC:2 ACD:2 BAC:2 ACD:2)
  留住 : (CD:3)
  第三次 : 合併 L2 成 L3 = ( )
  丟掉 : ( )
  留住 : (CD:3 BACD:2)
- **步驟三**: 無須過濾

關鍵詞自動擷取範例 [Tseng 2000]：英文範例

Web Document Clustering: A Feasibility Demonstration

Users of Web search engines are often forced to sift through the long ordered list of document returned by the engines. The IR community has explored document clustering as an alternative method of organizing retrieval results, but clustering has yet to be deployed on the major search engines.

The paper articulates the unique requirements of Web document clustering and reports on the first evaluation of clustering methods in this domain. A key requirement is that the methods create their clusters based on the short snippets returned by Web search engines. Surprisingly, we find that clusters based on snippets are almost as good as clusters created using the full text of Web documents.

To satisfy the stringent requirements of the Web domain, we introduce an incremental, linear time (in the document collection size) algorithm called Suffix Tree Clustering (STC), which creates clusters based on phrases shared between documents. We show that STC is faster than standard clustering methods in this domain, and argue that Web document clustering via STC is both feasible and potentially beneficial.?

Terms extracted before filtering

1. clusters based on : 3
2. document clustering : 3
3. of Web : 3
4. on the : 3
5. search engines : 3
6. STC is : 2
7. Web document clustering : 2
8. Web search engines : 2
9. clustering methods in this domain : 2
10. requirements of : 2
11. returned by : 2

Terms extracted after filtering

1. clusters based : 3
2. document clustering : 3
3. Web : 3
4.
5. search engines : 3
6. STC : 2
7. Web document clustering : 2
8. Web search engines : 2
9. clustering methods in this domain : 2
10. requirements : 2
11. returned : 2

關鍵詞自動擷取範例 [Tseng 2000]：中文範例

Comparison of Three Metadata Related Standards

在本文中，我們介紹了三個跟 metadata 相關的標準，它們分別是 FGDC 的 Digital Geospatial Metadata、Dublin Core、和 URC。雖然它們各有自己的設計目標和特質，但都是假設其操作環境為類似網際網路的環境。FGDC 的 Digital Geospatial Metadata 是設計來專門處理地理性資料，由於它有聯邦行政命令的支持，可說是已成為美國在地理方面的資料著錄國家標準。Dublin Core 則比較像是 USMARC 的網路節縮版，使非專業人士也能在短時間內熟悉和使用此格式來著錄收藏資料，但在現階段祇針對類似傳統印刷品的電子文件。由 IETE 的 URI 工作小組所負責的 URC，其原始的設計目的雖是用來連結 URL 和 URN，但為因應電子圖書館時代的要求，其內含逐漸擴大，雖然尚在發展中，但由於有 IETE 的支持，未來成為網際網路上通用標準的可能性極大。在此文中，我們也從幾個不同角度，分析和比較這三個 metadata 格式的異同和優缺點。

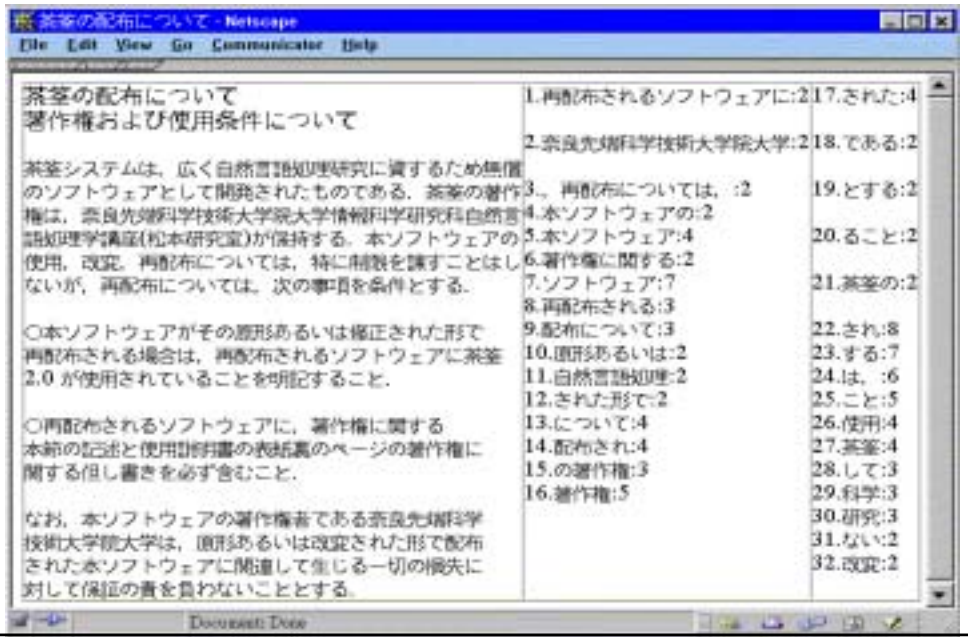
Terms before filtering

1. 設計 : 3
2. 資料 : 3
3. 網路 : 3
4. 標準 : 3
5. Dublin Core : 2
6. FGDC 的 Digital Geospatial Metadata : 2
7. IETE 的 : 2
8. 三個 : 2
9. 文中 : 2
10. 比較 : 2
11. 它們 : 2
12. 由於 : 2
13. 地理 : 2
14. 成為 : 2
15. 我們 : 2
16. 的支持 : 2
17. 的設計目 : 2
18. 格式 : 2
19. 著錄 : 2
20. 電子 : 2
21. 網際網路 : 2
22. 環境 : 2
23. 雖然 : 2
24. 類似 : 2

Terms after filtering

1. 設計 : 3 (design)
2. 資料 : 3 (data)
3. 網路 : 3 (network)
4. 標準 : 3 (standard)
5. Dublin Core : 2
6. FGDC 的 Digital Geospatial Metadata : 2
7. IETE : 2
8. 三個 : 2 (three)
9. 文中 : 2 (in the article)
10. 比較 : 2 (comparison)
11. 它們 : 2 (they)
12. 由於 : 2 (owing to)
13. 地理 : 2 (geography)
14. 成為 : 2 (become)
15. 我們 : 2 (we)
16. 支持 : 2 (support)
17. 設計目 : 2 (*incorrect term*)
18. 格式 : 2 (format)
19. 著錄 : 2 (record)
20. 電子 : 2 (electronics)
21. 網際網路 : 2 (Internet)
22. 環境 : 2 (environment)
23. 雖然 : 2 (although)
24. 類似 : 2 (similar)

關鍵詞自動擷取範例 [Tseng 2000]:直接運用於日文



關鍵詞擷取成效評估

- 評估資料：
  - 100篇台灣新聞（抓自2000年6月3日中國時報網站）
- 結果：
  - 平均每篇文件有 33 個關鍵詞
  - 平均每篇文件有 11 (33%) 個關鍵詞不在詞庫中（含 123, 226 個詞 ）
  - 相異的關鍵詞總共 2197 個
  - 其中有 954 個詞（954/2197 = 43%）不在詞庫中
  - 954 個詞中有 79 個是錯誤不合法的詞（人工檢視結果），錯誤率 8.3%
  - 整體錯誤率則爲 3.6% (=79/2197)

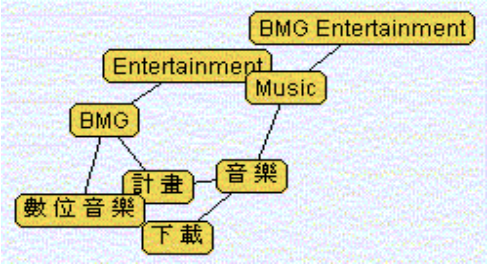
單篇文件關鍵詞擷取範例

BMG Entertainment與Sony Music計畫在Internet 上銷售數位音樂。（美國矽谷/陳美滿）

根據San Jose Mercury News報導指出，BMG Entertainment計畫在6月上旬或中旬開始在Internet 上銷售數位音樂。消費者將可直接將音樂下載至PC，而無需購買CD或錄音帶。該公司爲執行上述計畫已與多家高科技廠商合作，包括 IBM、Liquid Audio 與 Microsoft。BMG隸屬於Bertelsmann公司。

另外，Sony Music也將於下週一宣佈該公司計畫於本月底開 始提供數位音樂下載。消費者將可在手提裝置上聆聽下載來的數位音樂。此項數位音樂下載將是市場上首項具有防止盜錄功能的產品。網路音樂市場在過去幾年已顯現市場潛力，主要拜MP3規格之賜。

1：音樂 (7)
2：數位音樂 (5)
3：下載 (4)
4：計畫 (4)
5：BMG (3)
6：Music (2)
7：Sony Music (2)
8：Entertainment (2)
9：BMG Entertainment (2)



關聯詞分析：新的方法：[Tseng 2002]

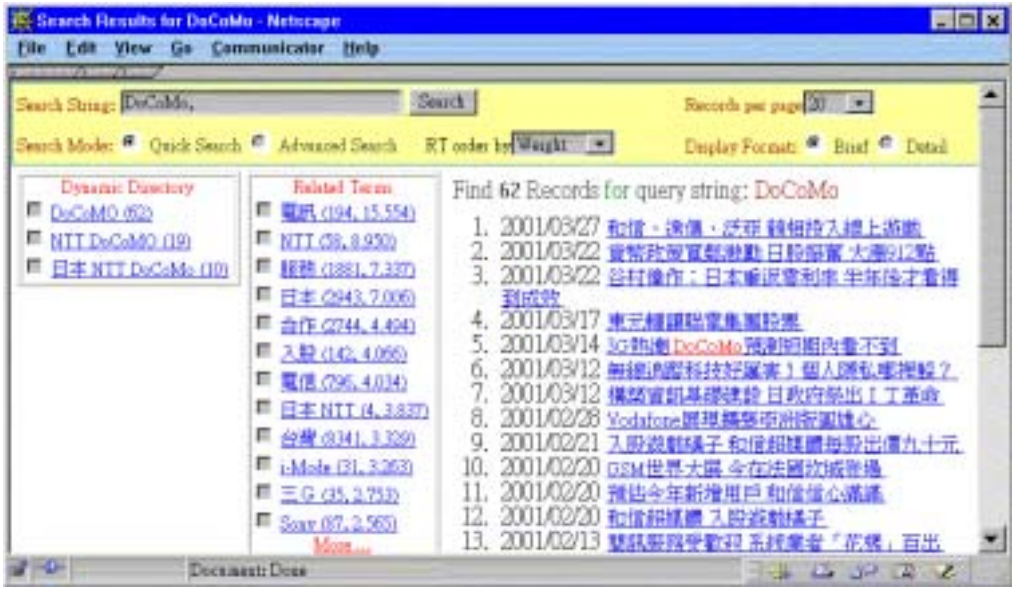
- 第一步：詞彙選擇：
  - 每篇文件先用 **詞庫**（長詞優先法）斷詞
  - 再由**關鍵詞擷取演算法** 擷取關鍵詞（至少出現2次者）（包含新詞）
  - 以 **停用詞** 過濾擷取出的關鍵詞，並依詞頻（term frequency）高低排序
  - 選 詞頻最高的 N 個詞作關聯分析
- 第二步：詞彙關聯分析:
  - 每篇文件選出來的詞，以 DICE公式計算兩個詞彙的 權重 wgt：
  - **關聯詞** 的權重超過門檻值（1.0）者，才依下面公式累積其權重
  - **關聯詞** 的最後相似度定義爲：
    - 原方法：僅單純累加每對關聯詞的權重 $sim(T_j, T_k) = \sum_{i=1}^{n} wgt(T_{ij}, T_{ik})$
    - 新方法：加入 IDF (inverse document frequency) 及 **詞彙長度**

$$sim(T_j, T_k) = \frac{\log(w_k \times n / df_k)}{\log(n)} \times \sum_{i=1}^{n} wgt(T_{ij}, T_{ik})$$

關聯詞擷取效率比較

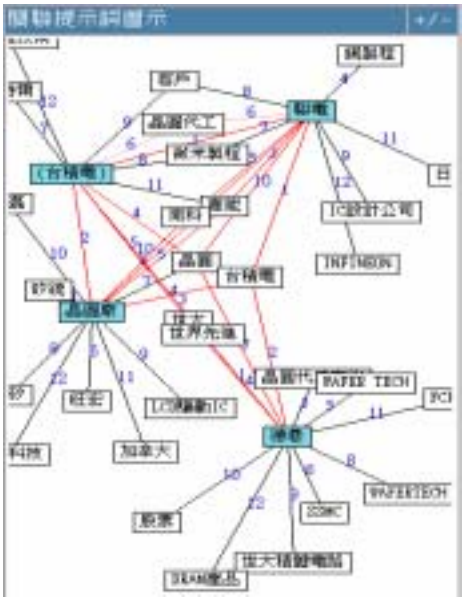
- Chen ’95 ’96 的方法：
  - 4714 文件, 8 MB, 費時9.2小時取出 1,708,551 個關聯**詞對**
  - 限制每個詞的關聯詞數最多100 個，共刪除了 60% 的**詞對**
  - 2GB文件，費時 24.5 CPU小時，產生4,000,000個關聯詞對
- Tseng的方法：
  - 336,067 新聞文件, 323 MB
  - 費時約 5.5 小時，擷取出11,490,822 個關鍵詞
  - 全部關聯詞數: 248,613, 平均每個詞有9個關聯詞
  - 2004: NTCIR 38萬篇中文新聞文件，51分鐘
    - 斷詞、索引詞擷取、關鍵詞擷取、關聯詞分析、反向索引檔建立

關聯詞應用範例（1/3）



關聯詞應用範例（2/3）

關聯詞應用範例（3/3）



關聯詞排序

- 關聯詞可按三種方式排序
  - 強度：
    - 即關聯詞共現性的強度
  - 詞頻：
    - 按關聯詞出現的文件篇數（df）排序，df 越高者，排在越前面
  - 時間：
    - 按關聯詞出現在最近文件的次序排序
    - 目的：讓最近才出現的關聯詞不必累積到足夠大的強度，即可排序在前面
    - 如：「李登輝」的關聯詞中，出現「康乃爾」，因爲李登輝最近又重訪康乃爾
    - 對具有時間事件的文件集可能很重要
- 關聯詞提示的順序不同，使用者感覺的關聯度不同

關聯詞排序

查詢詞「古蹟」的關聯詞，依「詞頻」,「時間」,「強度」排序

關聯詞成效評估

- 目的
  - 瞭解查詢詞與其提示的關聯詞之間的關聯(relatedness)情況
- 以兩種方式評估：
  - 直接計數前N（50）個被受試者判定爲有關聯的關聯詞數
    - 優點：簡單，可回溯比較
    - 缺點：不能細微區分排序的差異
  - 以精確率與召回率評估哪一種排序方式較好
    - 計算平均精確率的程式爲 TREC及NTCIR用的 `trec_eval` 程式
- 評估方式：
  - 邀請5位研究所同學，就30個查詢詞（每人6個），從系統提示出來的前50個關聯詞中，判斷是否跟查詢詞相關

trec_eval 的部分輸出

Queryid (Num): 4（即 查詢詞 ：「古蹟」）
Total number of documents (terms) (for「古蹟」)
Retrieved: 50
Relevant: 43
Rel_ret: 35(即 找到且被判斷爲相關者）
Interpolated Recall - Precision Averages:
at 0.00 1.0000
at 0.10 1.0000
at 0.20 1.0000
at 0.30 0.9412
at 0.40 0.9130
at 0.50 0.8800
at 0.60 0.8438
at 0.70 0.7949
at 0.80 0.7447
at 0.90 0.0000
at 1.00 0.0000
Average precision (non-interpolated) for all rel. terms
0.7315（單一查詢的平均精確率）

Precision:
At 5 terms: 1.0000
At 10 terms: 1.0000
At 15 terms: 0.9333
At 20 terms: 0.9000
At 30 terms: 0.8333

R-Precision (precision after R
(= num_rel for a query) docs retrieved):
Exact: 0.7442

關聯詞成效評估

- 從25233篇新聞文件中擷取關聯詞
- 結果：
  - 排序　詞頻　時間　強度
  - 關聯比例　48%　59%　69%
  - 平均精確率　0.302　0.403　0.528
  - 「詞頻」最差，因爲高頻詞，代表的主題較範圍較大，以致於跟任何查詢詞的關係都不大
- 結論：
  - 依「強度」排序的效果最好
- 比較：
  - (Sanderson & Croft SIGIR99) 關聯比例：67 %

結語

- 共現索引典（關聯詞庫）的優點
  - 快速呈現館藏文獻內容，具備主題摘要效果
  - 提供館藏內容的有效瀏覽
  - 即時反應館藏文件索引、查詢用詞，降低「字彙不匹配」問題
  - 提供非專業使用者專業的導引
- 共現索引典（關聯詞庫）的缺點
  - 館藏文獻沒記載、或統計不足的關聯詞無法擷取
    - 如：「紅樓夢」與「石頭記」
  - 關聯屬性沒有標示

計劃成果

- 相關論文
  - Yuen-Hsien Tseng, "Automatic Thesaurus Generation for Chinese Documents", **Journal of the American Society for Information Science and Technology**, Vol. 53, No. 13, Nov. 2002, pp. 1130-1138.
  - Yuen-Hsien Tseng, "Fast Co-occurrence Thesaurus Construction for Chinese News," Proceedings of the 2001 **IEEE Systems, Man, and Cybernetics Conference**, Tucson, Arizona, USA, October 7-10, 2001, pp.853-858.
- 相關專利
  - 曾元顯, *數位文件關鍵特徵之自動擷取方法*, **中華民國發明專利第 153789**
  - 曾元顯, *索引典自動建構方法*, **申請中**.
- 後續論文
  - Yuen-Hsien Tseng, Da-Wei Juang and, Shiu-Han Chen "Global and Local Term Expansion for Text Retrieval," Proceedings of the Fourth NTCIR Workshop on Evaluation of Information Retrieval, Automatic Text Summarization and Question Answering, June 2-4, 2004, Tokyo, Japan.
  - 中文互動式檢索輔助功能之效益評估 -以關聯提示詞爲例, 2004年
- 後續應用
  - 促進國內檢索技術提昇

中文互動式檢索輔助功能之效益評估
以關聯提示詞爲例 -- 葉佳昀

- 2004年以相同文件、相同查詢詞、不同受試者重複實驗
  - 小文件集25233篇
  - 中文件集15,4720篇
  - 小文件集的相關比例為 69.87%
  - 中文件集的相關比例為 78.33%
  - **文件越多，效果越好**
- 30個查詢詞

中東	地震	亞洲國家	通航	債券	環保署
中油	有線電視	邱義仁	博物館	奧運	職棒
主計處	朱鎔基	國安基金	晶圓代工	會計師	醫生
古蹟	李安	國科會	鄉鎮	調查局	顏慶張
生物科技	那斯達克	被害人	雅虎	選民	黨員

NTCIR 中文主題檢索成效

- 012::導演，黑澤明
- 012::查詢日本導演黑澤明的生平大事

RunID	Rigid		Relax	
	MAP	% imp	MAP	% imp
C-C-T+AT	0.2119	-	0.3217	-
C-C-T+MT	0.4094	93.20	0.5442	69.16
C-C-T+BRF	0.2881	35.96	0.3912	21.60
C-C-T+MT+BRF	0.4795	126.29	0.5962	85.33
C-C-T+AT(p)	0.2472	16.66	0.3892	20.98
C-C-T+MT(p)	0.4174	96.98	0.5918	83.96
C-C-T+BRF(p)	0.3602	69.99	0.5576	73.33
C-C-T+MT+BRF(p)	0.6707	216.52	0.6779	110.72
Max of C-C-T	0.7145		0.7492	
Avg of C-C-T	0.5083		0.5954	
Min of C-C-T	0.2119		0.3217	

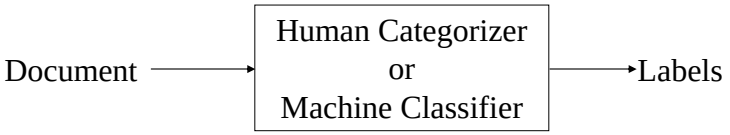
國內系統類似功能

自動分類 (Classification)

- 意義：
  - 將文件按主題自動分配到已定義的類別
  - 自動對文件給定某種標籤（labeling）
  - 排序、組織文件
  - 將非結構化文字訊息，轉成結構化便於利用的資訊
- 相關的處理：
  - 自動資訊過濾（information filtering）
    - 將新到文件按使用者需求，分送給使用者
  - 自動歸類（clustering）
    - 將主題相同、或描述同一事件的文件自動歸成一類，而沒有事先定義類別的需要、或限制

Automatic Text Categorization

- Text Categorization (document classification)
  - Given a text document, generate some **predefined** categories **based on its content** (or topics)
  - The categories may just be some labels that are irrelevant to the meanings of the document
    - Ex: a high-tech report without talking about confidential materials may be labeled as “confidential”
  - Thus it can also be considered as a “**document labeling**” task
- Automatic Text Categorization
  - Generate the categories by a machine



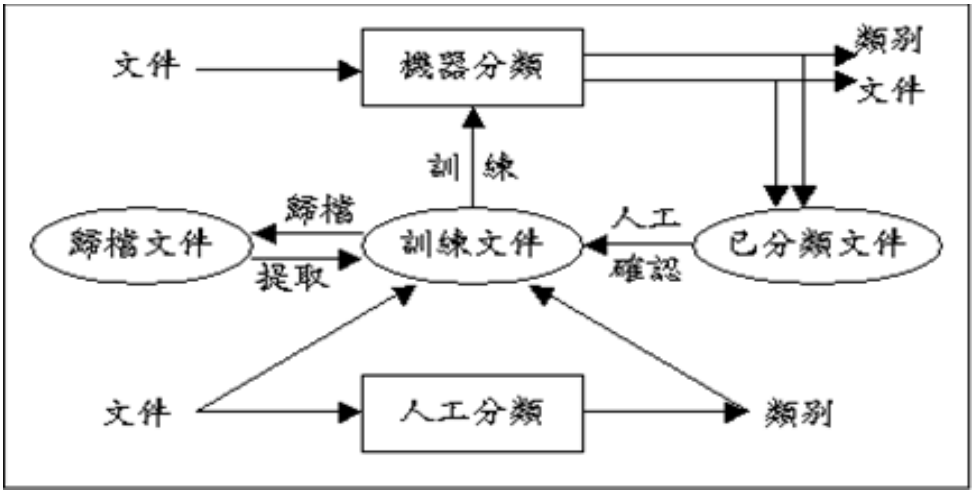
Applications

- Collection sorting or browsing by topics
  - Web directories of Yahoo, Kimo, Openfind
- Topic-based retrieval
  - keyword-based searches under some topics
- Document management
  - storage, access, filing, especially for physical paper documents
- Email, web page filtering
  - Spam filtering, confidential message filtering
  - Sexual, violent, filthy web page filtering
- Routing, information dissemination
- Text mining, knowledge discovery

Approaches

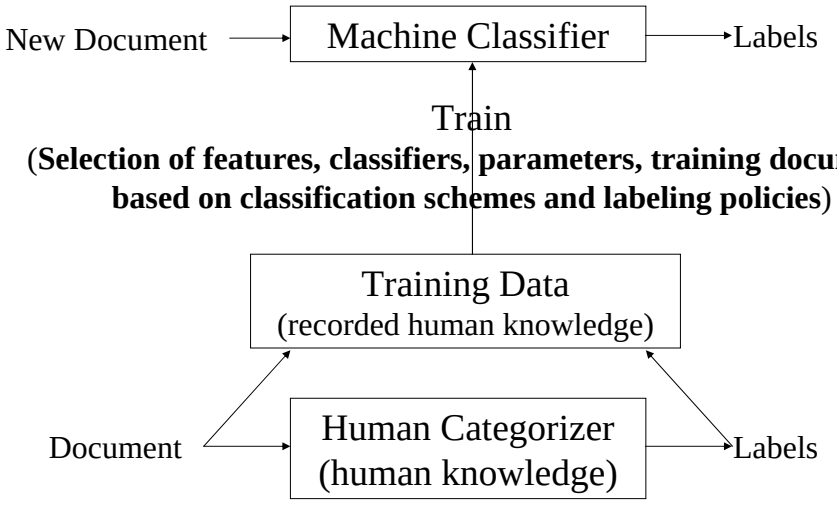
- Rule-based: semi-automatic rule induction
  - based on some human-crafted rules or knowledge
    - Ex: Area classification (Taiwan, China, US, UK, …)
    - "A report about software industry in Silicon Valley" belongs to "US"
- Learning from training data: fully automatic rule induction
  - When the rules are hard to induce and when there are documents that have been labeled by experts,
  - the knowledge of how to classify documents is all in the training data;
  - machine classifiers can be trained to learn the implicit rules

分類流程圖



主題分類範例

Automatic Rule Induction



Factors That Affect Effectiveness

- Feature selection
- Feature reduction
- Preprocessing: summarization
- Choice of Classifiers
- Classification scheme (Taxonomy)
- Labeling policy
- Data Inconsistency
- Number of training documents (Selection of training documents)
- Selection of Categories for effectiveness demonstration
- Performance evaluation
- Parameter tuning

Feature Selection

- Words, n-grams, noun phrases, word pairs (need not adjacent), …
  - Most works use bags of words for English documents
  - N-grams are suitable for multilingual documents [Damashek '95, Science]
  - Word pairs are used in [Al-Kofahi, 2001] for classifying 13,779 categories
- A feature set optimal for training documents is not necessary optimal for test documents
  - Ex: long n-gram (n=15 or 20)
- A feature should be predictive for the documents to be classified.
  - A short n-gram is predictive for documents
  - A long n-gram is predictive for categories
- Feature selection may best depend on applications (taxonomy)
- Features
  - Maybe too many => reduce efficiency
  - Maybe noisy => reduce effectivness

Feature Reduction (1/3)

- Many approaches to reduce the number of features:
  - DF threshold, TDxIDF, CFxIDF, stop words, from positive texts only [Ng et al SIGIR'97], …
  - Mutual information, information gain, chi-square test, correlation [Ng et al SIGIR'97], odds ratio [Mladenic ICML-99], …
- How many features should be used?
  - Table from [Ng SIGIR'97], No. of features for each classifier (for each category) in Reuters (93 categories using perceptrons)

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

- Yang and Pedersen [1997] only observe improved performance for Reuters (removing noisy terms), and did not see any improvement from feature reduction on OHSUMED.

Feature Reduction (2/3)

- [Bekkerman SIGIR 2001]
  - The BEP of Reuters almost approaches its maximum with only 50 words (chosen by MI), but the graph of 20 NG constantly goes up while its slope constantly lowers.
  - Only 3 words can achieve 79.1% micro-average for the largest 10 categories in Reuters

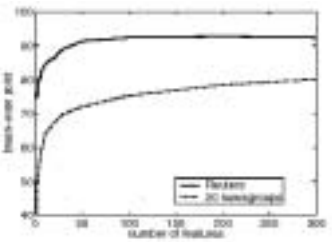


Figure 3: Learning curves (BEP vs. number of words) for the Reuters-21578 and the 20NG over the Mutual Information top 10 (above) and the top 500 (below) words using BOW-based representation and SVM

Category	1st word	2nd word	3rd word	BEP
earn	vs+	cts+	loss+	93.5%
acq	shares+	vs−	Inc+	76.3%
money-fx	dollar+	vs−	exchange+	53.8%
grain	wheat+	tonnes+	grain+	77.8%
crude	oil+	bpd+	OPEC+	73.2%
trade	trade+	vs−	cts−	67.1%
interest	rates+	rate+	vs−	57.0%
ship	ships+	vs−	strike+	64.1%
wheat	wheat+	tonnes+	WHEAT+	87.8%
corn	corn+	tonnes+	vs−	70.3%

Table 3: Three best words in terms of *MI* and their categorization BEP rate of the 10 largest categories of Reuters. The micro-average over these categories is 79.1%. '+' means that the word contributes by its appearance, '−' means that the word contributes by its disappearance

Feature Reduction (3/3)

- Joachims' Fig 1 [1998] for Reuters "acq" category:
  - All features are ranked by (binary) information gain
  - Feature sets: 1-200, 201-500, 501-1000, 1001-2000, …
  - Worse feature sets still perform much better than random (using naïve Bayse as the classifier)

- Number of features required depends on
  - classifiers
  - collections
- FR is ?able for many categories

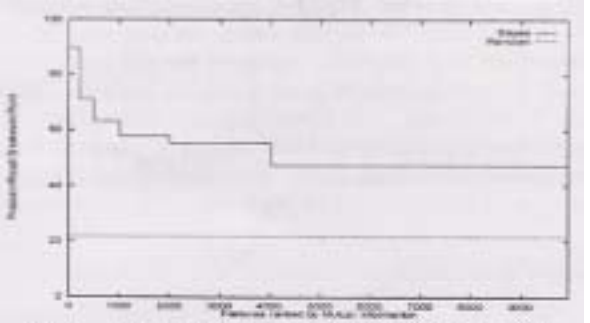


Fig. 1. Learning without using the "best" features.

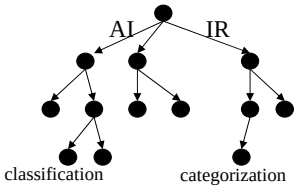
Preprocessing: Summarization

- Will summarization help classification?
- Lee, 2001, with Reuters collection
  - Precision of full-text classification: 65%
  - Classification after summarization (max. 3 sentences): 71.9%
- Lin and Tseng, 2001, with Reuters collection
  - micF of full-text classification by Vector Space Model: 0.689
  - micF of abstract (30%) classification by Vector Space Model: 0.715
  - micF of full-text classification by KNN: 0.753
  - micF of abstract (30%) classification by KNN: 0.728
- Better classifiers may be better at choosing and weighting terms
- Summarization may improve efficiency, but may not improve effectiveness for better classifiers for short documents
  - in Reuters, 7 sentences per documents in average
- For long documents, summarization effect is unclear (due to lack of test collections)

Choice of Classifiers

- SVM, KNN, Naïve Bayes, Neural net, LLSF, LMS, …
- [Yang and Liu, 1999] with Reuters 90 categories:
  - { SVM, KNN} > LLSF > multilayer perceptrons >> multinomial Naïve Bayes
- Another collection may show a different ordering
- Multiple classifiers can be better
  - A better classifier may not perform better on every category
  - If classification errors of different classifiers can be predicted, effectiveness can be improved by combining different classifiers
  - See [Lam 2001], [Al-Kofahi 2001] [Yang 2000], [Larkey 1996]

Classification Scheme (Taxonomy)



- Flat structure: usually for small schemes
- Hierarchical structure
  - Mostly for large taxonomies
  - Categories are more similar in topics at lower hierarchy
  - It's not always the case that categories in nearby positions in the hierarchy are more similar in topics than those categories that are far apart in the hierarchy. (see the diagram)
  - It's not always the case that if 'B.F' is a child of 'B', and 'B.F' is assigned to a document, then 'B' is assigned to the document too. [Lewis 2002]
  - Documents may be classified to internal nodes, instead of leaf nodes, due to
    - The union of leaf nodes != their internal node
      - Does not expect there will be other siblings
      - There is really no proper leaf node to label the document

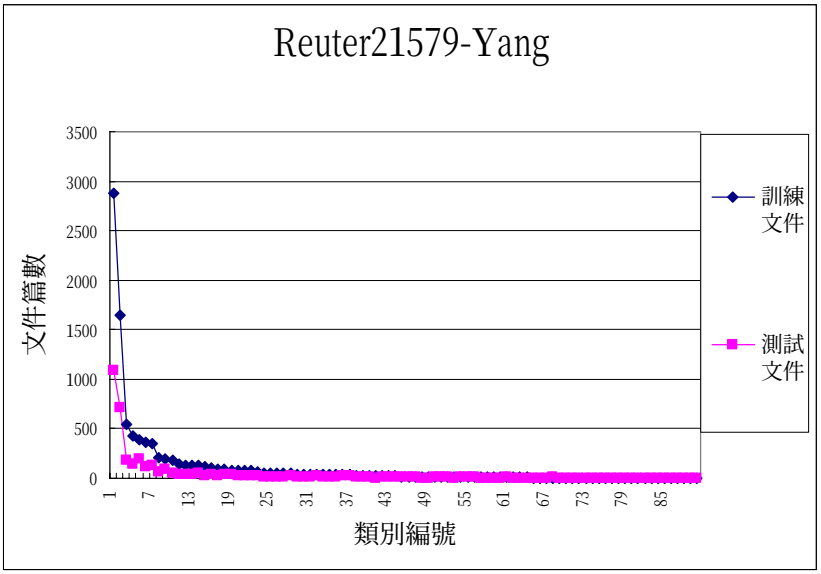
Classification Scheme (2/2)

- How to measure the effectiveness of a classifier in a hierarchical structure?
  - If 2 classifiers give two labels: 'B.E' and 'B.E.G.R', respect., which one is better if the document belongs to 'B.E.G'?
    - in multiple binary classifiers, both are incorrect
  - Is edge distance reliable?
- Will hierarchical multiple classifiers perform better than flat classifiers?

Labeling policy

- Single assignment
  - A document is assigned to one and only category
  - Suitable for physical document (paper) management
  - Disadvantage: categories assigned by categorizers may not match the expected categories of users
- multiple assignment
  - A document is assigned to (none), one, two, or more categories
  - Allow users to find the same document from different subjects
- Multiple binary classifiers are usually adopted for multi-class/multi-label problems, but may have problems:
  - Assume that each category is independent from each other
  - Need extra efforts to assure at least one category or at most k categories for practical applications.

Skewed Distribution of Categories



Skewed Distribution of Categories: 10 largest Categories

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

Skewed Distribution of Categories: 20 smallest Categories

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

Category selection

- Reuters' category distribution is skewed
- What can we learn about a category from a ***single*** training example? Are they a matter of research? [Bekkerman 2002]
- Different category selections make comparison difficult
- Can we train and test on only large categories?
  - Yes, if we only want to compare the effectiveness of different classifiers
- Or should we use all the categories?
  - Skewed category distribution is the nature of text data
  - Real-world problems are almost all skewedly distributed
  - Yes, if we want to deal with real-world problems

Data Inconsistency

- From trec_filtering mailing list:
  - Reuters' classification is found to be inconsistent [Kreines, 2001]
  - If you give 100 people the same document to classify, they will come up with an average of about 30 different classifications. [Herb 2001]
  - The same person will classify an object differently depending on the person's needs, desires, and recent experience.[Herb 2001]
- [卜小蝶 2001] :
  - 根據本研究以二名不同分析人員，針對 1,000個檢索詞彙作分類測試，在大類上有10%、小類有20%的不一致性。
  - 因此，在人工分類的步驟中，每個特徵詞彙都需經過二名分析人員分類，當詞彙被分到不同類別時，則由第三名分析人員加以判斷決定。

Data Inconsistency [Kao 2001]

- Here is a real-life example from Kao's work experience posted in ddlbeta@scils.rutgers.edu :
  - A **domain expert** (with about 30 years' experience in the domain) is specifically hired back from retirement to do this task on contract, so he also does not have other extractions at work.
  - But we found in 3 months apart, he classified the same messages, under two different occasions (for 2 different sub-tasks) very differently.
  - In our case, each message belongs to multiple classes. There are over 500 classes to choose from (which he's very familiar with), and each message on average belongs to 5 classes. Some belong to over 30 classes.
  - In these examples that he classified them differently, there are more differences (in what classes are assigned) than not. I.e. about half agreed, but the other half of classes assigned to each message are different.
  - You can imagine how much diversity you'd get when different experts are involved.
  - We actually use this as an argument for computer aided classification. I.e. we use classifier to aid, not to replace human experts to do their job.

Data Inconsistency [Kao 2001]

- This hurts a classifier in two different ways.
  - One, since the training data is not accurate, a classifier can't be trained as well as it could be.
  - Two, since the ideal world answer is less than perfect, it hurts the evaluation of the classifier.
- Even knowing the training data is imperfect, a classifier can be quite successful.
- In another task, domain experts are only allowed to assigning one class to each message, while in reality each message belongs to multiple classes.
- We built a classifier which was trained on this knowingly imperfect sample set. It turned out to work very well, i.e. manual examination of sampled results shows the additional classes assigned are by and large correct.
- As you can guess, the traditional recall and precision measures do not help here.

Effectiveness Reliability [Tseng 2001]

- In a hierarchical classification task with multiple assignments allowed, a better classification method (verified by a number of papers) yielded slightly inferior precision and recall levels than a naive one.
- After inspecting the testing data, the labels assigned by human indexers were found to be coarser and less consistent than the labels given by the better classifier for quite a large number of documents.
- Although the inaccuracy also occurred in the training data, a good automatic classifier can still learn well in a noisy environment. (As long as the noise does not become the signal.)
- Given the precision and recall values of different classifiers, is it reliable to tell that one is better than the others by use of an "imperfect" collection, without knowing how inconsistency is in the collection?
- Is single training document reliable?

Number of Training Documents

- How many training documents are needed for sufficient effectiveness ?
- [楊允言 1993]

所用訓練資料月份	關鍵詞數量	訓練資料召回率			測試資料召回率		
		第一名	第二名	第三名	第一名	第二名	第三名
7~12 月	5,579	94.86%	99.56%	99.95%	67.14%	76.67%	82.86%
8~12 月	5,085	94.67%	99.41%	99.85%	61.98%	74.14%	79.09%
9~12 月	4,344	96.09%	99.74%	99.87%	61.69%	77.39%	82.76%
10~12 月	3,379	97.09%	99.90%	100%	59.77%	71.26%	78.93%
11~12 月	2,379	98.16%	99.54%	100%	53.82%	67.18%	74.81%
12 月	1,297	99.62%	100%	100%	52.64%	67.43%	71.65%

- The more, the better
  - But one may use categorization zone (sub-sampling) to reduce the number

Performance Evaluation

- Accuracy: (a+d)/(a+b+c+d), insensitive to small classes
- Micro-average

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

- Macro-average

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

- Micro-average tends to be dominated by large classes
- Macro-average tends to be dominated by the large number of small classes

	Retrieved	Not Retrieved
Relevant	a	b
Non-relevant	c	d

Performance Evaluation

- From William Hersh at ddlbeta@scils.rutgers.edu [2002]
  - This field needs to move beyond simulated data sets and recall/precision type metrics
  - We have all seen the decent results that good systems can generate, but now we need to figure out to make these tools truly useful to researchers and analysts who are going to use them to solve real-world problems.
  - That is, we need to show that we can help a researcher extract knowledge from mining the literature that results in scientific discovery previously not possible (or excessively time-consuming).

Performance Evaluation

- 人工檢驗結果：
  - 測試114篇新聞，51篇有問題，正確率：55%。
- 分類器報告的數據：
  - Total file Num. N=114
  - Average F(=2*P*R/(P+R)): 0.831
- 使用者的標準是：
  - 如果該文件應選5類，分類器選了5類，但其中一類差異極大，整篇文件的分類就算錯誤。
  - 對我門而言，沒有所謂答對4/5，而是分類錯誤。因爲編輯人員就要動動滑鼠，在分類樹展開、關閉、搜尋、點選，以取消不適當的類別。
  - 編輯人員每天要確認 300 則新聞的類別，而這只是其每日1/3的工作量。

Parameter tuning

- What are the users' applications of text categorization.
  - Collection browsing, document management, searching, filtering, …
- Recall-oriented
  - To allow users to access the same documents from different viewpoints
  - May give many labels to a document
- Precision-oriented
  - To allow efficient document management
- Validation set
  - Should use cross-validation techniques to avoid the risk of optimization by an unreliable subset.

Threshold Tuning

- [Yang 2001 SIGIR] applied kNN to 5 collections and showed that:
  - Threshold policy makes significant differences in effectiveness
  - Optimal strategy may vary by applications
  - Rcut: For each document, sort classes by score and assign Yes to each of the R top-ranking classes
  - Pcut: For each class, sort the test documents by score and assign Yes to each of the top $P(C_i)$ x M x K documents
    - Performs well on rare classes
  - Scut: Tune the threshold for each classes individually
    - Tends to overfit (best for Reuters, worse for OHSUMED)
  - RTcut: uses both the rank and score information of classes

How effective can a machine classifier be?

- Two human categorizers can not have 100% consistency, let alone the consistency between a machine classifier and a human categorizer.

Conclusions

- Effectiveness measuring should consult the users to reflect their needs
- Effectiveness reports may best include both all categories and top N largest categories, if precision and recall are used
- Combining different classifiers may improve effectiveness
- Number of training documents is the major factor that affect effectiveness
- There are inherent difficulties in automatic classification
- How effective can a machine classifier be?
  - Agree with human categorizers to the same degree as human categorizers agree with one another, provided that the set of categories does match the subject matter of incoming documents [Dorre 1999].

Difficulties in Automatic Classification

- Classification scheme may change
  - Categories may be deleted, merged, split, moved
    - Split “baseball” into “professional baseball” and “amateur baseball”
  - Things may change over time
    - Lee Deng-Hue may no longer be “President”, nor “KMT”
- Classification inconsistency in training data
  - Due to different categorizers
    - Different background knowledge
    - Different familiarity with the classification scheme
  - Due to different criteria even for the same categorizer
    - the same document may be classified differently at different time
    - The larger the classification scheme, the more inconsistency may occur
- Inconsistency between categorizers (classifiers) and users
- All of these may be faced by a human categorizer, as well.

自動摘要 (Automatic Summarization)

- 摘要的類型：
  - 資訊性摘要（informative summary）
    - 提供精簡的內容資訊，使讀者瞭解文件的重要內容
    - 30% 的重要原文，常可提供 80% 到 90% 的原文重點
  - 指示性摘要（indicative summary）
    - 提示此篇文章的存在，並提供足夠的資訊，使讀者能決定是否應該閱讀原始文件
    - 一、兩句的原文，如標題，常可視爲指示性摘要
  - 評述性摘要（critical summary）
    - 以簡要的方式，對原文作一評論，使讀者大致瞭解原文的內容與目的
- 摘要產生方式：
  - 自動產生主題簡要文句：目前技術還不成熟
  - 自動選擇重要句子：目前大部分系統的作法
- 自動選句主要方法：
  - 評量每個句子的重要性，依此選取使用者需要的句子數量（如：30%）
  - 關鍵詞的多寡：越重要的句子包含越多或越重要的關鍵詞
  - 句子位置：標題、最前後段落、每一段落的第一句與最後一句, …
  - 特殊句型：「… 不過，…」、「結論是…」、「認爲 …」, …

自動摘要

- 選句範圍：
  - 單篇文件摘要
  - 多篇文件摘要：需具備文件歸類（clustering）能力
- 句子的重要性：
  - 關鍵詞的多寡：越重要的句子包含越多或越重要的關鍵詞
  - 句子位置：標題、最前後段落、每一段落的第一句與最後一句, …
    - 與文件型態有關
  - 特殊句型：「… 不過，…」、「結論是…」、「認爲 …」, …
- 呈現方式：
  - 重點式摘要：將選到的句子依原文中順序列出
  - 瀏覽式摘要：將選到的重要句子 highlight 出來
- 應用
  - 應用於文件分類、歸類，提升成效
  - 檢索結果或大批文件的瀏覽呈現
  - 協助人工分析閱讀全文

自動摘要範例（1/3）

什麼是Napster?

Napster 目前是網際網路中用戶成長最快的 軟體，因爲使用這套資源共享軟體，幾乎可搜 尋任何你想得到的音樂和歌曲，只要同好的硬 碟中找到該音樂和歌曲，然後立即且免費下載 。下載的音樂和歌曲不但可在電腦和MP3隨身聽 播放，甚至可「燒錄」成CD唱片。

Napster 的用戶估計已達數百萬之多，每天 使用此軟體下載的歌曲多達三百萬首。由於軟 體炙手可熱，Napster 公司已取得一千五百萬 美元創投資金，但同時卻成爲多起訴訟的對象 ，被控以侵犯著作權及詐財，其中包括知名重 金屬樂「Metallica」所提控訴。

不過，Napster 並未將MP3檔案（作用是將C D唱片上的音樂轉換爲壓縮的電腦檔案）儲存在 自己的電腦伺服器內，而是用戶本身將音樂和 歌曲儲存在自己的電腦內，並彼此間直接交換 這些音樂和歌曲。因此Napster堅稱自己只是 提供軟體共享的導管，無力約束用戶行爲。

除著作權法之外，Napster 所引起的爭議還 要牽涉更廣泛，包括科技創新對唱片業者未來 的衝擊、廿一世紀 網路使用倫理以及歌手與其聽眾之間的關係等等。

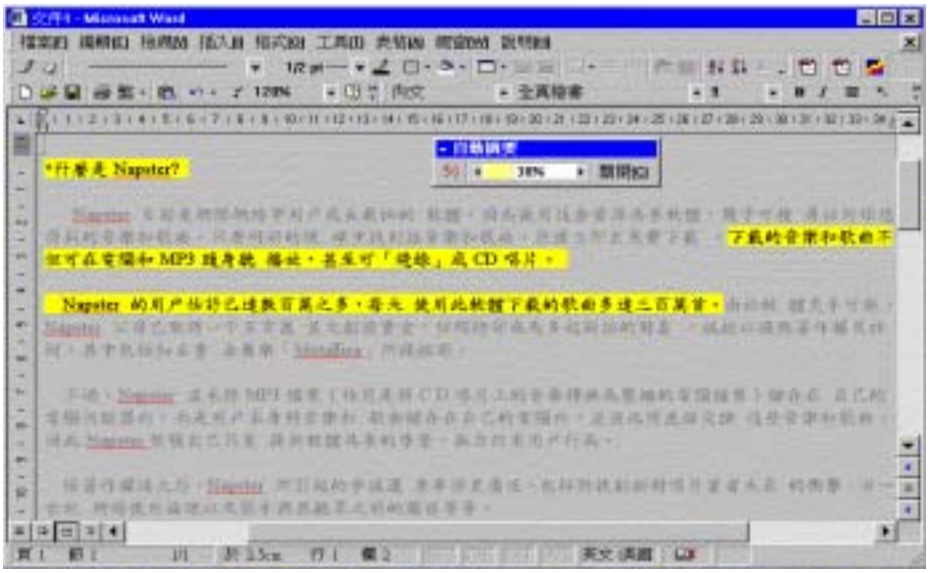
（資料來源：中國時報網站，2000.05.30）

自動摘要範例（2/3）

- 摘要產生軟體：[Tseng 2000]（軟體著作）

自動摘要範例（3/3）

- 摘要產生軟體：Word 2000



自動歸類 (Clustering)

- 前述之關聯詞可視爲對詞彙做歸類
- 文件歸類方式：
  - 依內文主題做歸類
  - 依規則做歸類：網頁搜尋引擎常用之方法
- 依內文歸類之主要方法：
  - 兩兩文件比較其相似度，相似度高者歸成一類
- 歸類結果呈現方式：
  - 單層平面式歸類
  - 階層式歸類
- 歸類的應用
  - 多篇文件摘要
  - 相關文件提示（瀏覽某篇文件，提示更多篇相關文件，惟此功能也可由動態搜尋取代）、相似文件比對、協助發現文件之間的關聯
  - 主題與事件偵測
- 歸類的運算量高，至少爲$O(n^2)$，需限制所需歸類的文件數量，才能作有效歸類。

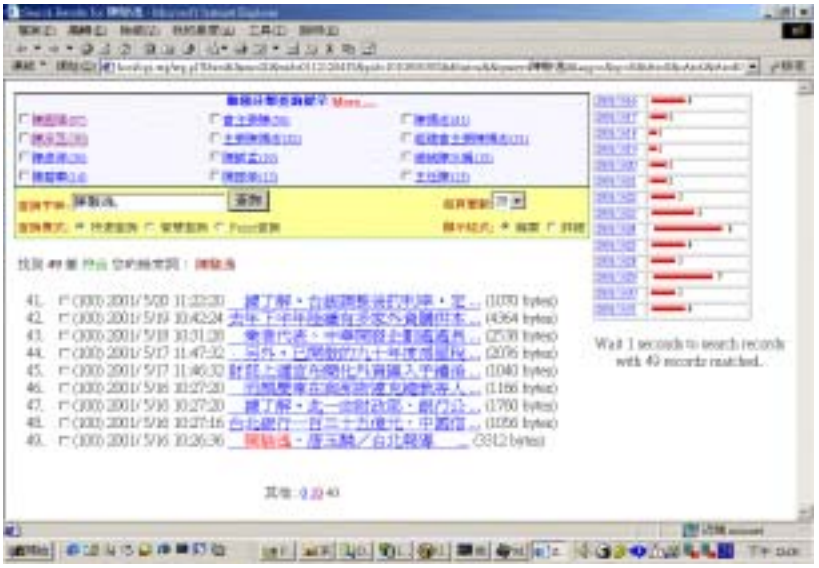
自動歸類範例

- 527:0.093189 (**病例**:0.7167, **香港**:0.6491, **感染**:0.4912, ＳＡＲＳ:0.2205)
  - 425:0.102990 (**死亡**:0.7924, **香港**:0.7815, **北京**:0.7688, **病人**:0.7192)
    - 40:0.281695 (**北京**:1, **累計**:0.8628, **疑似病例**:0.8135, **診斷**:0.8135)
      - 13:0.345352 (**診斷**:1, **新增**:1, **疑似病例**:1)
        - » 34 : 香港僅增一例WHO：擬撤旅遊警告
        - » 92 : 大陸稱疫情平緩WHO：勿遽下結論
      - 147 : 大陸SARS通報新低
    - 47:0.269720 (**全球**:0.5687, **急性**:0.5245, **呼吸道**:0.4888, **症候**:0.4888)
      - 90 : SARS敲警鐘WHA催生全球防疫網
      - 116 : ＷＨＯ公布臺灣煞死率全球第三
  - 120:0.191171 (**世衛組織**:0.5687, **病例**:0.3759, **感染**:0.2576, ＳＡＲＳ:0.1156)
    - 35 : 星新病例追查感染源
    - 152 : ＷＨＯ：菲切斷傳染鏈除煞
- 675:0.081792 (ＷＨＯ:0.5783, **臺灣**:0.3431,)
  - 161:0.164859 (**連戰**:0.4514, ＷＨＯ:0.4322, **臺灣**:0.2564,)
    - 41:0.281161 (**政治**:0.6592, **連戰**:0.5332, **主席**:0.5028, ＷＨＯ:0.3729)
      - 9:0.356199 (**連戰**:0.4338, ＷＨＯ:0.3033, **疫情**:0.2078, **臺灣**:0.1800)
        - » 47 : 臺灣入WHO連戰促北京去政治化
        - » 52 : CNN專訪連戰嚴譴北京打壓
      - 108 : 連戰促扁疫情危急勿政治操弄
    - 48 : 吳伯雄：參與世衛五重點動員海內外推動
  - 224:0.142773 (**陳建仁**:0.6592, ＷＨＯ:0.3729, **臺灣**:0.2212, ＳＡＲＳ:0.1422)
    - 16:0.342666 (**陳建仁**:0.8135, ＷＨＯ:0.3033, **臺灣**:0.1800, ＳＡＲＳ:0.1156)
      - 113 : WHA未邀陳建仁報告疫情
      - 161 : 湯姆森：WHO專家赴臺滅火 陳建仁：WHA拒我遺憾激動
    - 127 : 世衛宣布全臺旅遊示警朝野同促醫護抗疫補網

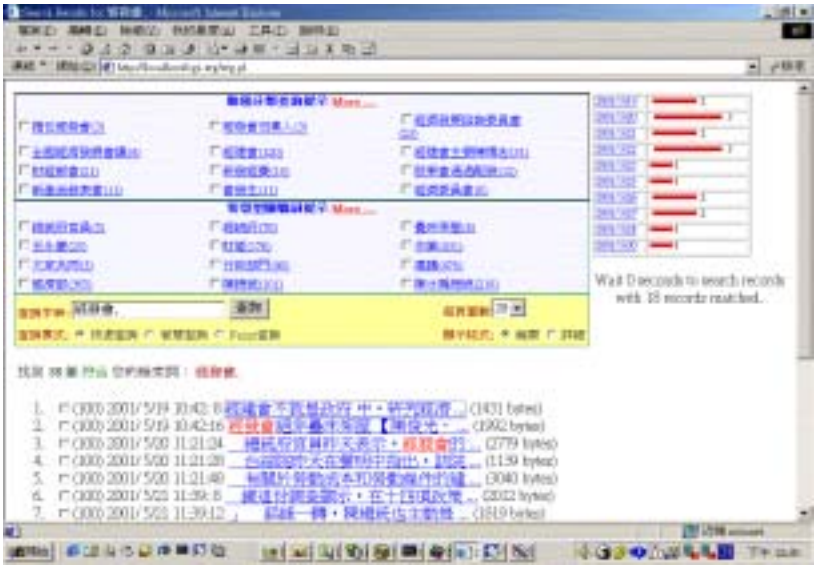
時間事件分析

- 非結構化文字資訊結構化後，可利用 DBMS 的 工具做分析（data mining）
  - 結構化處理：關鍵詞擷取、關聯詞分析、分類、歸類等
- 將文件按時間、類別或其他屬性，進行交叉分析
  - 時間事件分析（範例一、範例二）
- 對詞彙分佈進行分析
  - 權威詞彙控制、犯罪術語歸類、人物關聯分析
- 對主題分佈進行分析
  - 犯罪情形、偵辦手法、地域差別
- 結合其他結構化資訊一起分析
  - 整合的資訊分析系統

時間事件分析（記者5月份發稿量）



時間事件分析（經發會起始日期）



結語

- 文件資訊探勘
  - 互動、反覆的探索與分析過程
  - 解決新問題需人工導引分析過程、人工解讀分析結果
  - 需要檢索系統的輔助，索引爲其他自動化探勘工具的核心
- 成效
  - 對該領域的新進人員、生手，提供專家級的知識輔助，使其更快進入狀況
  - 加強現有工具之功能
    - 加強資訊的存取、利用、分享、分析
    - 強化 DBMS 分析功能，整合結構化與非結構化資訊