國 立 交 通 大 學

資訊科學與工程研究所

碩 士 論 文

自動摘要系統基於AdaBoost

Automatic Summarization System based on AdaBoost

研究生：鍾喻安

指導教授：李嘉晃 教授

中 華 民 國 九 十 九 年 六 月

多文章自動摘要系統基於 AdaBoost

Automatic Summarization System based on AdaBoost

研 究 生：鍾喻安　　Student：Yu-An Chung

指導教授：李嘉晃　　Advisor：Chia-Hoang Lee

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

Jun 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

自動摘要系統基於AdaBoost

學生 ： 鍾喻安　　指導教授：李嘉晃 教授

國立交通大學 資訊科學與工程研究所碩士班

摘要

隨著科技和網路的發展，網路上的資訊以指數倍數的速度成長，搜尋引擎雖然可以幫我們找到相關的資訊，但是往往符合查詢條件的結果還是數以千計;藉由自動摘要的發展，可以自動從大量的文件和資料中，取得讀者所想要得到的資訊，以方便讀者閱覽。

一般使用機器學習方法於摘要問題上，大多為單一機器學習的摘要方法，這些方法主要是使用標題、相似度、或者詞彙的重要程度…等特徵，去選取文章中的重點句子;不同於傳統機器學習方法，整體學習法則可被視為 meta-algorithm，可以同時合併各種不同之演算法，以產生更好的分類結果。

本論文應用了 AdaBoost 演算法於摘要問題上。AdaBoost 是一種群體學習演算法，它提供了一個設計架構，允許系統設計多個弱分類法，其中這些弱分類法必須有 50%以上的正確率。目的是藉由分類問題和已知的分類結果訓練出一組較好的弱分類法集合與此集合裡弱分類法各自的權重，最後再予以合併，形成一強分類器。在本系統裡，我們根據文件中重要特徵資訊設計弱分類法，應用 AdaBoost 於中文之文件摘要，本實驗結果在壓縮率 15%、20%、30%摘要，準確率分別為 50.0182264%、48.1455086%、49.5370552%，已有將近五成的正確率。英文部分，ROUGE-1、ROUGE-2、ROUGE-SU4 準確率分別為 41.201% 、10.003%、14.845%。

Automatic Summarization System based on AdaBoost

Student ： Yu-An Chung Advisor：Prof. Chia-Hoang Lee

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

Abstract

Due to the rapid advancement of digital technology in the last two decades, there has been an increasingly large amount of digital content available on the Web. The enormous and continuously growing volume of data necessitates the development of efficient and effective text summarization systems. In this paper, we propose to employ AdaBoost to perform news summarization task. One of the features of AdaBoost is that it allows the system to incorporate many rules of thumb into the system and it can adaptively change the weightings of these rules. When the training process is completed, the system can employ the linear combination of weak classifiers with weightings to construct a strong classifier. We take into account several features to design weak classifiers.

In system performance evaluation, we collected 200 news from 4 different categories as the data set and performed the experiments under different compression rates. The experiment results show that our system works stably and the average F-values are 0.5002, 0.4815 and 0.4954 under 15%, 20%, and 30% compression rates. The experiment results show that AdaBoost with weak classifiers can outperform the systems using SVMs and SVR in news summarization application. Meanwhile, we also apply our system to English text summarization corpus, which is DUC2002, and the recall of ROUGE-1、ROUGE-2、ROUGE-SU4 are 41.201% 、10.003%、 14.845%.

致謝

本篇論文的完成，首先要感謝我的指導教授李嘉晃老師，在這兩年的研究及學習過程中，老師熱心的指導與諄諄教誨，讓我受益良多，在學術的研究和論文的寫作上給我很大的幫助，在此向老師致上最高的謝意。

另外，也要感謝實驗室同學，佑州、孝承、瑞敏，在這兩年的研究生活中，提供我在研究課題以及生活上的支持與幫助，有你們的陪伴讓我受益良多。

最後，我要感謝我的家人，爸、媽、老哥、老姐和玫茵，有你們的支持與照顧，讓我在求學期間可以專心於課業之上，沒有後顧之憂，才可以順利的取得碩士學位。僅以本篇論文，獻給我的家人，以及曾給予我幫助的所有人，謝謝你們。

目錄

表目錄

圖目錄

第一章、緒論

1.1 研究背景和動機

在這資訊和科技發達的時代，伴隨著我們的是諸多的文字和資訊，如何在現今社會緊湊的步伐中，取得閱讀和時間的平衡點，是值得我們研究的議題;本論文以此作為出發點，希望透過機械學習方法，從冗長的文章中擷取重點或使用者想閱讀的資訊。

一般使用機器學習方法於摘要問題上，大多為單一機器學習的摘要方法。AdaBoost 是一種群體學習演算法，它提供了一個設計架構，允許系統設計多個弱分類法，其中這些弱分類法必須有 50%以上的正確率。目的是藉由分類問題和已知的分類結果訓練出一組較好的弱分類法集合與此集合裡弱分類法各自的權重，最後再予以合併，形成一強分類器。在本系統裡，我們根據文件中重要特徵資訊設計弱分類法，應用 AdaBoost 於中英文之文件摘要。

1.2 研究目的與方法

本論文的研究目的為讓使用者可以將一篇文章或者多篇的文章作為系統的輸入，經過系統摘要後呈現給使用者文章的重點和使用者想取得的資訊。許多使用者在瀏覽網頁或搜尋主題的資訊內容時，往往希望先瀏覽網站的重點內容以了解該網站資訊是否符合自己的需求;例如，目前大多數搜尋引擎，均在每個收尋結果上，列出該結果頁面的摘要，讓使用者能夠透過該摘要來判斷是否該連接符合需求，藉此來達到節省時間以及迅速掌握文章重點之目的。另外，摘要也可以整合在螢幕較小的電子產品中，例如，手持式電子產品，此類產品因局限於螢幕的大小，不適合閱讀胞蘭大量文字的資料;摘要技術可將資訊重點呈現於手持式螢幕上，因此，摘要技術對於手持式產品的數位內容呈現，可扮演舉足輕重之角

色。

本論文方法部分是利用了 AdaBoost[1][2][3]做為一個框架，結合多個針對文件特徵設計的簡單分類法則，形成一個更準確的摘要方法。會使用 AdaBoost 主要原因為，一般使用機器學習方法於摘要問題上，大多為單一機器學習的摘要方法，這些方法主要是使用標題、相似度、或者詞彙的重要程度…等特徵，去選取文章中的重點句子;不同於傳統機器學習方法，整體學習法則可被視為 meta-algorithm，可以同時合併各種不同之演算法，以產生更好的分類結果。

AdaBoost 是一種群體學習演算法[2]，它可結合其他分類法，如:決策樹、類神經網路等方法，甚至於簡單的 decision rule，每個分類法可被視為一個弱分類器，而 AdaBoost 扮演的角色是一個框架的角色，將這些弱分類器列為它框架的一部份，只要弱分類法具備 50%以上的正確率，AdaBoost 可讓這些弱分類法藉由分類問題和已知的分類結果訓練出一組弱分類法集合和此集合裡每個弱分類法對於所有分類問題的重要權重，最後再予以合併，形成一強分類器。

在實驗比較部分，其它較著名的機器學習的演算法，例如，Support Vector Machine(SVM)[4]、以及 Support Vector Regression(SVR)[5];也已被使用在摘要的技術上，本論文也用了相同的文集去實做了這些方法當作我們比較的數據。我們分別應用 AdaBoost 於中文以及英文資料集上之摘要，中文部分，我們使用了奇摩新聞的兩百篇文章以及三份人工所做的摘要，英文則是使用英文摘要常使用的 DUC02[6][7]文集，以及 DUC02 所提供的人工摘要。

1.3 論文架構

第一章：前言，敘述本研究之動機與目的。
第二章：相關研究，敘述本研究之相關研究與背景。
第三章：系統設計，將本研究之系統整體架構與概念方法做一個完整介紹。
第四章：實驗結果與討論，將本研究之系統產生的結果與人工實驗之結果做一比對與分析。

第五章：結論與未來展望，將本研究之系統成果做一總結，並提出結論與探討未來研究之方向。

第二章、相關研究

2.1 常見的摘要分類

文章自動摘要可以根據不同的目的，簡單的分成下面五類:

1.單文件摘要與多文件摘要(single-document summarization and multi-document summarization):此類別根據系統所要摘要文件的多寡來分類，單文件摘要代表系統做摘要時每次只處理單篇文章，而多文件摘要則代表處理摘要時可以處理相同主題下的多篇文章。

2.單一語言摘要與多語言摘要(monolingual summarization and multilingual summarization):此類別是根據系統所要摘要的語言來分類，單一語言摘要輸入的文件只有一種語言，輸出的文件也只有一種語言，而多語言摘要則是輸入文件可以有多種語言，而輸出的結果為單一語言的摘要。

3.選取式摘要與抽象式摘要(extractive summarization and abstract summarization):此類別根據系統所要摘要的形式來做分類，選取式摘要是找出文章裡面的句子或者段落來做摘要，抽象式摘要摘是瀏覽過整篇文章後，再依照自己的想法重寫出一篇摘要。

4.資訊性摘要與指示性摘要(informative summarization and indicative summarization):此類別根據文章資訊與閱讀目的來做分類，資訊性摘要由文章中找出較重要的使用者資訊,指示性摘要則是提供足夠的且不同面向的資訊給使用者，讓使用者選擇他所想要繼續閱讀的摘要內容。

5.一般性摘要與使用者導向摘要(generic summarization and query-based summarization):此類別根據文章重點與使用者需求來做分類，一般性摘要由文章中找出較富有資訊與重點的摘要，使用者導向摘要則是根據使用者需求，或者使用者的回覆來做文章摘取，產生讀者想閱讀的重點資訊。

2.2 文章自動摘要常使用的方法和特徵

2.2.1 自動摘要的常使用方法:

1. 統計式(statistic)方式:

利用統計的方式去計算詞彙的重要程度，因為高頻的詞彙往往和主題的相關性較大。例如:TF*IDF 方法計算句子權重[8][9][10]。

2.語意式(semantic)方式:

利用語言的結構與語意方式，來分析文章中的重要層級(hierarchy)，產生一個階層概念圖，藉此來產生一篇摘要[11]。

3.潛藏語意分析(latent semantic analysis)方式:

利用線性代數方法為核心模型，其中包括奇異值分解與維度約化兩個過程。利用維度約化找出潛藏之語意，用來判斷詞彙與字句之間的關係[12]。

4.相似度(similarity)的方式:

藉由相似度來計算句子和已知重要的詞彙的相似程度來決定選取句子的與否[13]。

5.機器學習(machine learning)方式

機器學習是利用大量的資料集以及對應於此資料集問題的答案,作為電腦學習時,所需要的知識基礎,當電腦學習結束,就可以利用此學習基礎,做為往後運算所需要的背景知識。例如: AdaBoost[1][2][3] ,SVM[4] ,SVR[5]

2.2.2 自動摘要的常使用特徵:

我們根據字彙和句子將常見的特徵值分成兩類:

根據字彙:

1. 使用TF-IDF計算:

   由於高頻的詞彙(TF)往往和主題的相關性較大，以及反文件頻(IDF，即詞彙出現在各篇文章的次數之 inverse)用以判斷該詞彙是否具有鑑別力。所以，字彙重要程度=$tf_{word} \times idf_{word}$

2. 標題字[14][15]:

   由於標題字往往跟文章的內容有重要的關聯性存在，所以，假如在本文中的字彙有出現在標題裡，代表應該加重該此字彙的權重。

3. 提示片語[14][16]:

   提示片語，例如:”In this paper”,”Conclusion”,”Result”等，這些片語出現的句子通常具有涵蓋整篇或者整段的內容，在此情況下，該加重其後續字彙的權重。

4. 偏見字(biased word)[14]

   在文章中，某個字彙的語意上想要更深層，或是想要更清楚的表示此字彙的意思，就可以使用偏見字。Ex:”All men(people) are created equal.”，因為強調人人平等，所以在men之後加上偏見字people。

5. 大寫字

出現大寫字加重其權重，只適用於英文。例如:McDonald[16]認為在句子中專有名詞具備相當之重要性，但是目前專有名詞抽取的技術未成熟，暫時以大寫字代替。

6. 跟主題相關的字[14]:

跟主題字彙共同出現的字，由於主題通常是動詞或者名詞，而相關字就代表著修飾這些動詞或者名詞的字，例如: 跟主題字一起出現的形容詞或者副詞。

7. 地方詞:

在旅遊類文章中地方詞往往代表著作者想要表達的過程之一，因此在句子中假如出現地方詞須加重其權重。

根據句子:

1. 句子出現位置(location)[14][16]:

由於文章的前幾句往往會概略的描述文章整體的內首，因此文章前幾句通常是較為重要的。

2. 句子所在段落的位置[14]:

由於文章的前幾句往往會概略的描述文章整體的內容，因此首段是較為重要的段落，末段通常皆是結論，所以亦為重要段落。

3. 句子長度[14][16]:

摘要中的句子，其長度不宜過長或者過短，大約在 5~15 字間較為恰當。

4. 動名詞個數:

往往文章中的句子裡所包含的動名詞個數，代表著句子的重要性。

2.3 Ensemble learning 演算法

Ensemble learning [17]演算法是透過多次執行所預設好的多個基礎學習演算法，並且針對每個基礎學習演算法產生的學習結果進行投票，最後整合投票的結果構成一致同意的學習結果。Ensemble learning 演算法的有兩種主要的方法：

第一種方法是用不同的學習演算法去建造不同的學習結果，這些學習結果形成的組合具有多樣化的特性，而且每一學習結果的準確度也是完全不同。雖然每個學習結果對於新資料點的預測的錯誤率是合理的值，但是學習結果和學習結果之間，在大多數預測裡常常是彼此不一致的。如果能夠對這些單獨的學習演算法做某種程度上的整合，並且產生一個具有比較一致性的學習結果，是接下來第二種方法所想要改善的。

第二種整體學習方式是用採用組合多個基礎學習演算法的來訓練學習結果。此方式的意思是說，把權重高的票投給和資料誤差小的基礎學習演算法，然後把權重低的票投給和實際資料誤差大的基礎學習演算法，藉由不同權重的投票方式結合所有的基礎學習演算法，並產生一個比任何單獨基礎學習演算法都好的整體學習演算法。這種方法所生成的整體學習演算法，可以用來代表所有的基礎學習演算法。

Freund 和 Schapire (1996，1997) [1] 所提出的 AdaBoost 演算法就是一個整體學習演算法的代表。透過學習演算法，極盡可能地將整體的分類錯誤降低到最小，每次增加一個合理的分類器到整體學習演算法之中，分類錯誤可相對的降低。每次重複這個步驟並逐次累積合理的分類器，最後產生一個經由加權總數所得到的整體分類器，此整體分類器可以使得分類錯誤減少到最小。

2.4 AdaBoost

2.4.1 AdaBoost 演算法

AdaBoost 演算法是由 Freund 和 Shapire 在 1995 年所發表的演算法[1][2][3]，詳細演算法如 Algorithm 2-1 所示;此演算法將$(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m)$，此 m 個 pair 當作演算法的輸入，$x_1, x_2, \cdots, x_m$，是已知 training sample，他們的 label 分別是 $y_1\ y_2, \cdots, y_m$，而Y_i屬於{+1, -1}，並且假設這 m 個點的權重一開始皆是 $D_1(i) = 1/m$。

Algorithm 2-1. AdaBoost Algorithm

Given: $(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m)$

Where $x_1 \epsilon X$, $y_m \epsilon Y = \{+1, -1\}$

Initialize $D_1 = 1/M$.

For t=1,…,T

{

1. Train weak learner using distribution D_t
2. Get weak hypothesis $h_t: X \to \{+1, -1\}$ with error $\sum_{i:h_t(X_i)!=Y_i} \mathbf{D_t(i)}$
3. choose $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)/2.$
4. Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} e^{-\alpha_t}, \text{if } Y_i = h_t(X_i) \\ e^{\alpha_t}, \text{if } Y_i \neq h_t(X_i) \end{cases}.$$

$$= \frac{D_t(i) * \exp(-\alpha_t * Y_i * h_t(X_i))}{Z_t}$$

Where Z_t is a normalization factor(chosen so that D_{t+1} will be a distribution).

}

Output the final hypothesis:

$$H(X_i) = \text{sign}\left(\sum_{t=1}^{T} (HT_t(X_i) * \alpha_t)\right)$$

2.4.2 AdaBoost 演算法分析

首先已知 training sample 有 m 個點，X_1 X_2,…, X_m，他們的 label 分別是 Y_1 Y_2,…, Y_m，而Y_i屬於{+1, -1}，並且假設這 m 個點的權重一開始皆是 $D_1(i) = 1/m$。並且預設有 n 個基本分類器。

接下來我們會跑 T 個回合的迴圈，每個迴圈主要目的是調整此 m 個點權重，並挑選一個錯誤率最低的基本分類器。

以下為演算法的迴圈:

{

1. 計算每一基本分類器的錯誤率，錯誤率計算$\sum_{i:h_t(X_i)!=Y_i} D_t(i)$，$h_t$:基本分類器，$D_t(i)$:第 t 回合第 i 點的錯誤率。

   Ex:在第一回合，要決定初始基本分類器，我們會從已經預設好的基本分類器集裡選擇一個最好一個基本分類器。選擇方法如下:

   利用每個基本分類器分別去測試此 m 個點的分類結果，將預測出來的分類結果，分別和人工摘要結果做比較，看預測出來的結果有無跟 Y_1 Y_2,…, Y_m的 label 是一樣的，如果沒有，增加此點的權重給此分器當作錯誤率，初始錯誤率計算$\sum_{i:h_t(X_i)!=Y_i} D_1(i)$，$h_t$:基本分類器。

   並找出錯誤率最低的基本分類器當作我們的初始基本分類器

2. 找出錯誤率最低的基本分類器當作我們此第 t 回合的基本分類器HT_t。

3. 接下來要決定 α_t 的值，目的在於在下一回合中使整體的錯誤率會最低。

   我們經由證明(證明在後面)可以得知 $\alpha_t = \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)/2$ 時，整體的 error 會最低。

4. 一開始此 m 個點的權重都是一樣的，在每一回合中，我們要提升這回合分錯的點的權重，以及降低此回合被分對的點的權重，此舉的目的是讓下一回合所挑選的基本分類器能夠將此回合被分錯的點分對，以此類推，當經過 T 個回合後，達到最後所有點都可能被分對過，所以，我們利用改變每個點權重來達成：

(1)For i=1:m

begin

$$Temp(i)= D_t(i) * \begin{cases} e^{-\alpha_t}, if\ Y_i = h_t(X_i) \\ e^{\alpha_t}, if\ Y_i \neq h_t(X_i) \end{cases}$$

End

(2)$Z_t=\sum_{i=1}^{m} Temp(i)$;

(3)For i=1:m

begin

$$D_{t+1}(i) = \frac{Temp(i)}{Z_t} * \begin{cases} e^{-\alpha_t}, if\ Y_i = h_t(X_i) \\ e^{\alpha_t}, if\ Y_i \neq h_t(X_i) \end{cases}$$

End

5. 當此回合所挑出的基本分類器錯誤率 > 50%，則跳出此迴圈。

}

1~5:說明跑 T 個回合的迴圈，並在每回合中更新 training sample X_1 X_2 ,…, X_m 的權重$D_t(i)$,在下一回合中,利用更新過的權重選擇一個錯誤率最小基本分類器,提升這回合分錯的點的權重，以及降低此回合被分對的點的權重，此舉的目的是讓下一回合所挑選的基本分類器能夠將此回合被分錯的點分對。此舉的目的是讓不同的基本分類器可以互相填補各自在分類方法上不足的地方,也就是結合多個基本分類器，讓這些基本分類器變成一個比較強大的基本分類器。

所以最終的強分類器就是由一堆小的基本分類器組合起來的：

$H(X_i)=\text{sign}(\sum_{t=1}^{T}(HT_t(X_i)*\alpha_t)$ ，H: 代表強分類器.

2.4.3 AdaBoost 演算法證明

接下來證明，當 $\alpha_t=\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)/2$ 時，整體錯誤率將會降至最小。

$$\text{Training error}(H)=\frac{1}{m}\sum_i^m\begin{cases}1,\text{if } y_i\neq H(x_i)\\0,\text{if } y_i=H(x_i)\end{cases}\quad(1)$$

$$=\frac{1}{m}\sum_i^m\begin{cases}1,\text{if } y_i f(x_i)\leq 0\\0,\text{if } y_i f(x_i)>0\end{cases}\quad(2)$$

$$\leq\frac{1}{m}\sum_i^m\exp(-y_i f(x_i))\quad(3)$$

$$=\sum_i^m D_{t+1}(i)\prod_t Z_t\quad(4)$$

$$=\prod_t Z_t\quad(5)$$

(1) training error 就是把最終的強分類器套用在所有的點上，如果 y_i 不等於 $H(x_i)$，則代表分出來的結果和實際結果不一樣，就代表預測錯誤；相反則代表沒有錯誤，預測正確，最後加起來後再除以總數量 N 就是 training error。

(2) 我們將 $\sum_{t=1}^{T}(HT_t(X_i)*\alpha_t$ 視為 f(x)，所以(1)式也可以改成(2)式，

(3) 已知exp(?) >= 0，又如果K >= 0 則exp(K) >= 1，所以我們可以由(2)式推到(3)式

(4) 在證明這個式子前，我們先看看以下的推導

$$D_{t+1}(i)=\frac{D_t(i)}{Z_t}*\begin{cases}e^{-\alpha_t},\text{if } Y_i=h_t(X_i)\\e^{\alpha_t},\text{if } Y_i\neq h_t(X_i)\end{cases}$$

由AdaBoost 的流程，我們可以得

到上述的式子，原因無他就在於我們這個是設 binary case:{-1, +1}。

所以我們把每一個 t 都寫出來：

$D_2(i) = D_1(i) \exp(\ldots)/ Z1$

$D_3(i) = D_2(i) \exp(\ldots)/ Z2$

$D_4(i) = D_3(i) \exp(\ldots) / Z3$

…

$D_{T+1}(i) = D_T(i) \exp(\ldots) / ZT$

等號左邊相乘 = 等號右邊相乘，再消掉相同的的部分就可以得到

計算過程:

$$D_{T+1}(i) = \frac{1}{N} * \frac{\exp\left(\sum_{t=1}^{T} -y_i * HT_t * \alpha_t\right)}{\prod_t Z_t}.$$

$$= \frac{1}{N} * \frac{\exp\left((-y_i)\sum_{t=1}^{T} HT_t * \alpha_t\right)}{\prod_t Z_t}.$$

又$f(x_i) = \sum_{t=1}^{T} HT_t * \alpha_t$

$$=> D_{T+1}(i) = \frac{1}{N} * \frac{\exp\left(-y_i f(x_i)\right)}{\prod_t Z_t} \quad (6)$$

$$=> \exp\left(-y_i f(x_i)\right) = D_{T+1}(i) * N * \prod_t Z_t \ .$$

由**(6)**的推導就可以得到(4)的結果

(5) 再把 Distribution 的部分消掉(和等於 1)，就可以得到其實 training error 就是把 normalization 的部分都乘起來。

最後我們再來看 Zt 的部分，利用 minimize Zt 來找到αt，最終使得 training error 會最小。

$$Z_t = \sum_i^m Dt(i) * \begin{cases} e^{-\alpha t}, \text{if } Y_i = h_t(X_i) \\ e^{\alpha t}, \text{if } Y_i \neq h_t(X_i) \end{cases} \quad (7)$$

$= \sum_{i:h_t(X_i)=Y_i} D_t(i)e^{-\alpha_t} + \sum_{i:h_t(X_i)!=Y_i} D_t(i)e^{\alpha_t}$. **(8)**

$= e^{-\alpha_t} \sum_{i:h_t(X_i)=Y_i} D_t(i) + e^{\alpha_t} \sum_{i:h_t(X_i)!=Y_i} D_t(i)$.**(9)**

$= e^{-\alpha_t}(1-\varepsilon_t) + e^{\alpha_t}\varepsilon_t$ **(10)**

α_t = ln (1 –ε_t /ε_t) / 2 時，Z_t值會最小

$= 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$.**(11)**

$= \sqrt{1-4\gamma_t^2}$.**(12)**

$\leq e^{-2r_t^2}$.**(13)**

證明過程:

6. **(7)**為 normalization 的定義

7. **(8)**和**(7)**為相同的意思

8. **(9)**是把**(8)**裡的係數取到 sigma 外面來

9. **(10)**根據定義，所有基本分類器和以知 label 分出來不同的D_t和是ε_t，基本分類器和以知 label 分出來相同的D_t 和是$1-\varepsilon_t$。

10.此時要讓**(10)**的值最小，方法很容易，取微分為 0 的點，我們就可以知道當 αt = ln (1 － ε_t / ε_t) / 2 時，值會最小，再代回**(10)**的式子就可以得到**(11)**

11.最後因為我們假設 ε_t 是小於等於 1/2，所以我們令 ε_t = 1/2 - γ_t，γ_t代表此分類器比亂猜好多少值，帶入得到**(12)**

12.最後利用 1 + x <= exp(x)的概念就可以讓**(12)**式寫成**(13)**，以上證明完成了。

所以，整體錯誤率:

$\text{Training error}(H) \leq \prod_{t=1}^{T} Z_t$

$= \prod_{t=1}^{T} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$

$= \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2}$

$\leq \exp(-2\sum_{t=1}^{T}\gamma_t^2)$.

$\rightarrow$Training error(H) $\leq \exp(-2\sum_{t=1}^{T}\gamma_t^2)$

代表每多增加一個回合，整體的錯誤率是成指數的方式下降的。所以，強分類器的預測錯誤率必定小於任何單一的基本分類器。

2.5 SVM(Support Vector Machine)摘要方法

SVM[4]是一種監督式的學習方法，主要用來解決分類的問題(for 2-class problem)。Tsutomu HIRAO 提出應用 SVM 於摘要模型上[4]。本論文第四章的實驗部分也有應用 SVM 方法作為本論文比較的方法。我們先將訓練文章裡每個句子轉換成 SVM 空間上的點。

轉換方式為:1.先將句子利用特徵集(features)轉成向量。

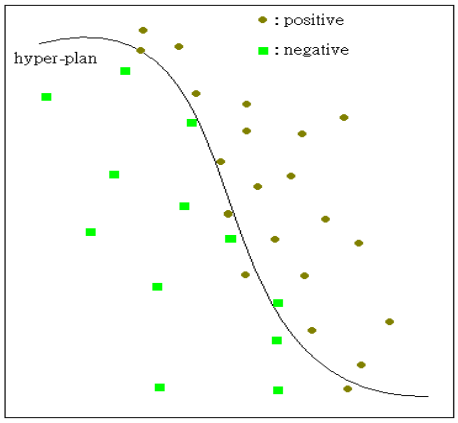
2.向量再轉換成空間上的點。

接下來，每個向量會對應到一個值，Answer$\in\{+1,-1\}$。Answer是決定於原本的句子是否有在人工摘要裡，如果有，則回傳+1，如果沒有，則回傳-1，$\{+1,-1\}$代表SVM會將這些向量分成兩類。

有了空間上的點與已知的分類答案，就可以將空間上的每一點表示成，

F：句子向量 → 此句是否在人工摘要出現 $\in\{+1,-1\}$，接下來就可以開始進行空間上的訓練，當訓練結束，會有一超平面將此空間裡分部的點分成兩群，有了此模組，我們也將測試句子轉換成向量，將此向量丟進空間裡，它將座落於hyper-plan的其中一側,如果是+1群,它則是SVM系統要選取的摘要句。

下圖為一簡單SVM分類結果示意圖。



2.6 SVR(Support Vector Regression)摘要方法

本論文也實作了 S. Li、Y. Ouyang、W. Wang 與 B. Sun 等人使用 SVR 於摘要之方法[5]，作為我們論文比較的對象。接下來是實作的過程，Support Vector Machine 除了分類問題之外，也可以用來處理迴歸的問題。所以處理迴歸問題的 Support Vector Machine 又稱做 Support Vector Regression(SVR)，所謂的迴歸指的是每個實體所對應的標籤是一個連續的實數，F(i)：Vector → R。

一開始，我們也是將每個句子轉換成向量，向量裡的元素是由已訂定的特徵集(features)所回傳的值，在找出向量對應的分數F(i): Vector → Score，

有了空間上的點以及所對應的分數，就可以開始尋找空間中的平面。如同SVM，SVR也是找尋空間中的最合適的平面，而它和SVM不同的是，SVM訓練出來的是一分為二的超平面，而SVR找的是能夠準確預測資料分佈的平面，所以向量對應的是一個分數而不是極端值。

下式為每個句子所需要回歸的連續實數，Vector→Score 裡的 Score 計算方式[4]。

$$Score(s) = \frac{1}{|s|} \sum_{t_j \epsilon s} \sum_{t_i \epsilon S} Same(t_i, t_j)$$

$\sum_{t_j \epsilon s} \sum_{t_i \epsilon S} Same(t_i, t_j)$:在此篇文章摘要 S 裡和在句子 s 裡詞彙相同的

個數

|s|:此句的詞彙個數

下面為我所取的特徵值集和

(1)Word-based Feature

$V_1(s) = \frac{1}{|s|}\sum_{t_j \epsilon s}\sum_{t_i \epsilon title} Same(t_i, t_j)$

|s|:句子詞彙個數

$t_j \epsilon s$:表示屬於此句子的詞彙

$t_i \epsilon title$:表示屬於標題的詞彙

$\sum_{t_j \epsilon s}\sum_{t_i \epsilon title} Same(t_i, t_j)$:在標題裡和在此句有多少詞彙是一樣的。

(2) Phrase-based Name Entity Feature

$V_2(s) = \frac{|entity(s) \cap entity(title)|}{|s|}$

|entity(s) ∩ entity(title)|:句子裡的專有名詞和 title 裡專有名詞相同的個數

(3) Centroid Feature

$V_3(s) = \sum_{t_j \epsilon s} tfidf(t_j)$

$tfidf(t_j)$: t_j的 TFIDF 值

(4)Named Entity Number Feature

$V_4(s) = \frac{|entity(s)|}{|s|}$

|entity(s)|:句子裡專有名詞的個數

(5)Sentence Position Feature

$V_5(s) = 1 - \frac{i-1}{n}$

i:是句子在這篇文章的位子(第幾句)

此向量的含義是說越靠近第一句的句子越重要，權重越重

有了 Vector(s)={ V1 ,V2 ,V3 ,V4 ,V5}以及 Score(s)，

就可以將每個句子轉換成 S:Vector→Score 的型態, 作為 training 和 testing 使用。

2.6 中央研究院斷詞系統

由於中文句子裡不具有像英文句子裡由空格隔開每一個詞彙，所以必須有一個好的斷詞處理方式，方能讓中文句子被斷詞成我們所認知的詞彙，所以在本論文中所使用的斷詞處理是中央研究院[18]的斷詞系統，經過處理後，再利用中研院所提供的詞性對照表來幫助我們篩選我們所需要的詞性再做處理，如表 2.4.1。

表 2.4.1

簡化標記	對應的 CKIP 詞類標記	
Na	Naa Nab Nac Nad Naea Naeb	普通名詞
Nb	Nba Nbc	專有名詞
Nc	Nca Ncb Ncc Nce	地方名詞
VA	VA11, 12, 13 VA3 VA4	動作不及物動詞
VB	VB11, 12 VB2	動作類及物動詞
VC	VC2 VC31, 32, 33	動作及物動詞
VE	VE11 VE12 VE2	動作句賓動詞
VH	VH11, 12, 13, 14, 15, 17 VH21	狀態不及物動詞
VHC	VH16 VH22	狀態使動動詞
VK	VK1, 2	狀態句賓動詞

2.7 DUC 介紹

在自然語言 Document Understanding Conference (DUC)[6][7]是由由美國國家標準與技術研究所(National Institute of Standards and Technology 簡稱 NIST)與美國情報局先進研發活動 (Advanced Research and Development Activity center of the U.S. Department of Defense，簡稱 ARDA) 所共同推動的標準資料集，從西元 2001 年開始推動，到目前為止，已提供了 DUC 2001 到 DUC 2007 等七個不同的標準資料集，它的目的是讓做摘要的學者們有一個共同參考的文集和評估方式，讓大家的系統可以更進一步的互相評估與比較。在每個 DUC 文集中包含了多個 Topic，每個 Topic 裡有 Title、Question 和這個 Topic 裡所包含文章的 Head、Article，以及相對於每個 Topic 的人工摘要。

第三章、系統設計

3.1 概念

一般使用機器學習方法於摘要問題上，大多為單一機器學習的摘要方法，這些方法主要是使用標題、相似度、或者詞彙的重要程度等特徵，去選取文章中的重點句子;不同於傳統機器學習方法，整體學習法則可被視為 meta-algorithm，可以同時合併各種不同之演算法，以產生更好的分類結果。

本論文方法部分是利用了 AdaBoost[1][2][3]做為一個設計架構，以結合多個針對每個特徵設計的簡單分類法則，形成一個具有多個特徵的摘要方法。AdaBoost 是一種群體學習演算法[2]，它可將簡單的 decision rule 當成弱分類器，AdaBoost 扮演的角色是一個框架的角色，將這些弱分類器列為它框架的一部份，只要弱分類法具備 50%以上的正確率，AdaBoost 可讓這些弱分類法藉由分類問題和已知的分類結果訓練出一組弱分類法集合和此集合裡每個弱分類法對於所有分類問題的重要權重，最後再予以合併，形成一強分類器。

在本系統裡，我們將基本摘要方法作為 AdaBoost 裡的弱分類法，分別對重要句子，以及非重點句作分類。另外，分類問題和已知的分類結果，我們使用了新聞文集以及 3 位學者所做的摘要，當作我們的訓練文集。圖 3-1 為一簡單的 AdaBoost 演算法的概念示意圖:

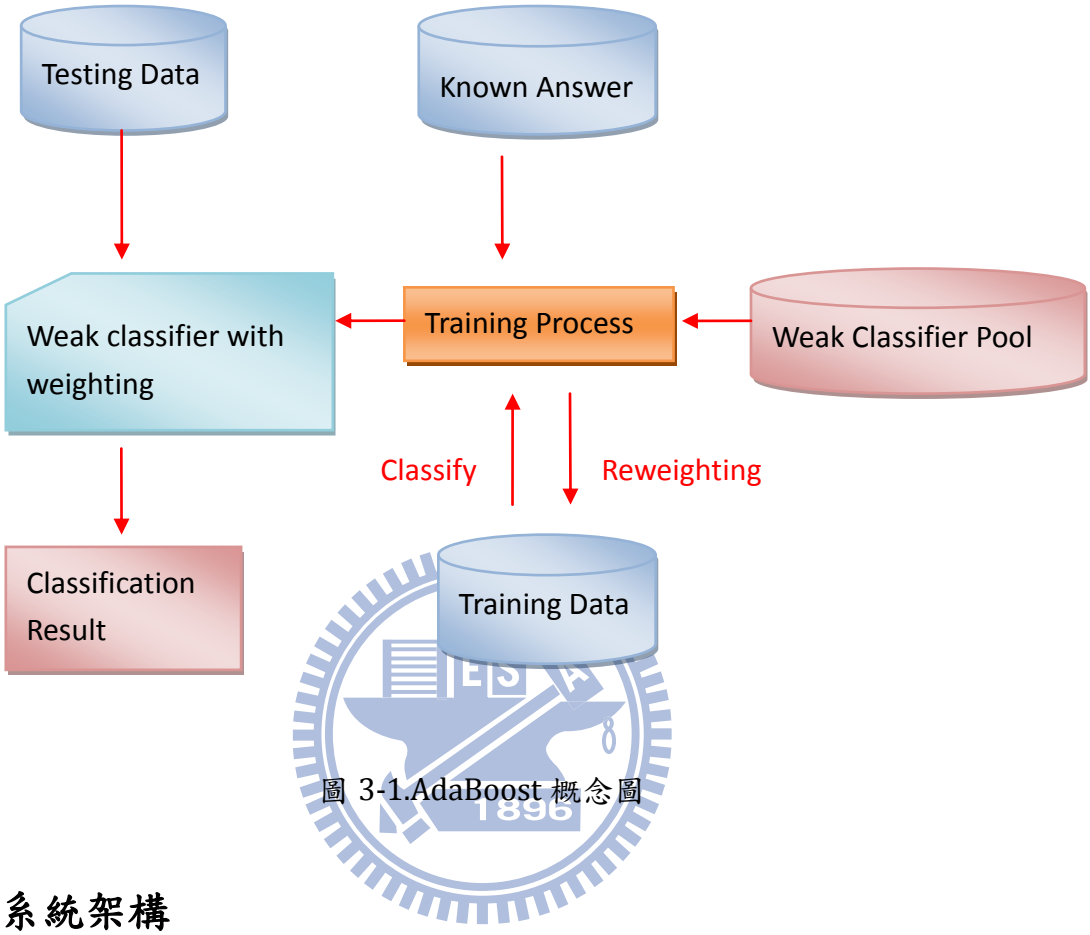


圖 3-1.AdaBoost 概念圖

3.2 系統架構

本系統共分為三個子系統架構，第一部分為文章前處理，在此會將 Yahoo 新聞的 200 篇文章做斷句、斷詞以及詞性標記的處理，並且計算各類別的詞彙重要程度(term frequency 的計算和 document frequency 的計算)，第二部分為系統核心部分，訓練系統模組，利用 AdaBoost 演算法將多個基本且簡單的分類方法利用訓練文集和人工摘要訓練出一強摘要方法，此強摘要方法是由多個基本分類器和其基本分類器各自的權重所組成。第三部分系統會利用此強分器對每個句子做評分，再將所有句子以分數做高低的排序，分數高低代表句子在文章中的重要程度，並找出符合壓縮率的摘要句數，做為系統的摘要。

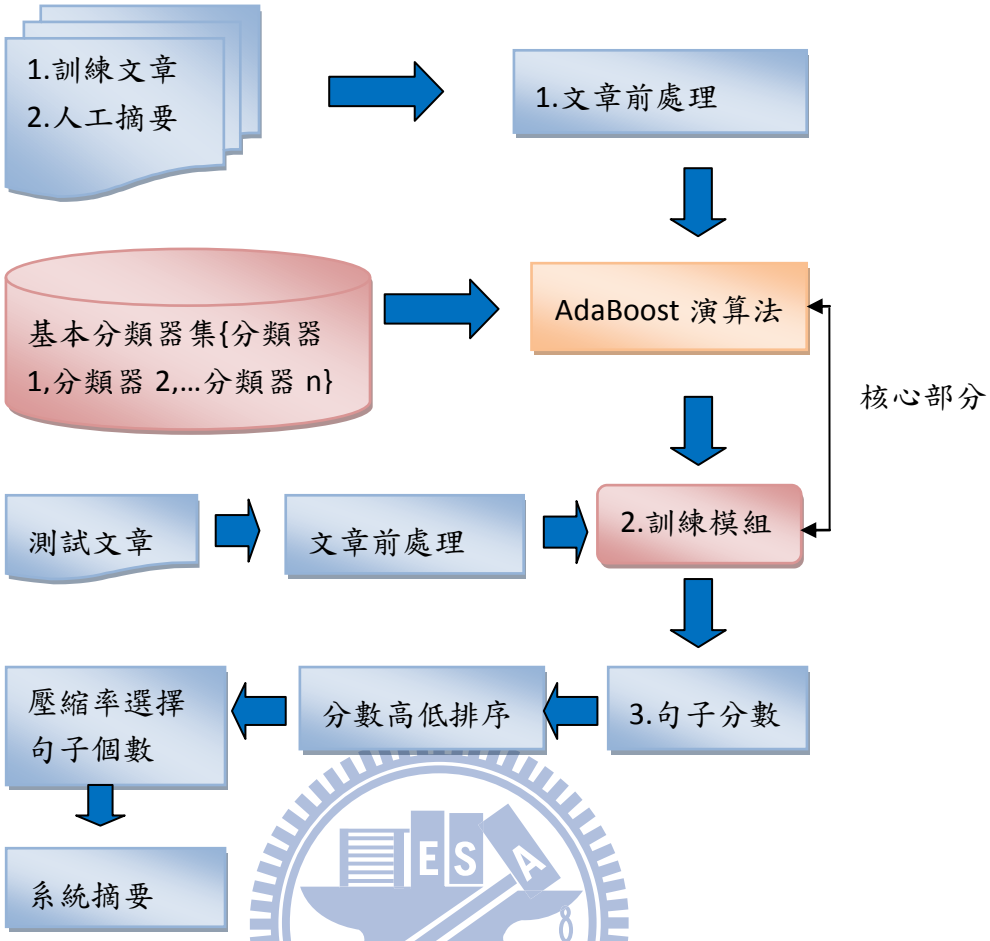


圖 3-2:系統流程架構

3.2.1 中文文章前置處理

本系統的資料集是由 200 篇 Yahoo 新聞類文章所提供，Yahoo 200 篇文章又細分為政治，社會， 旅遊，健康類各 50 篇。而前處理部將分成三個部分，各別為測試文章，訓練文章，以及人工摘要做前處理。另外，前處理整體流程分成三個階段，如圖 3-3。

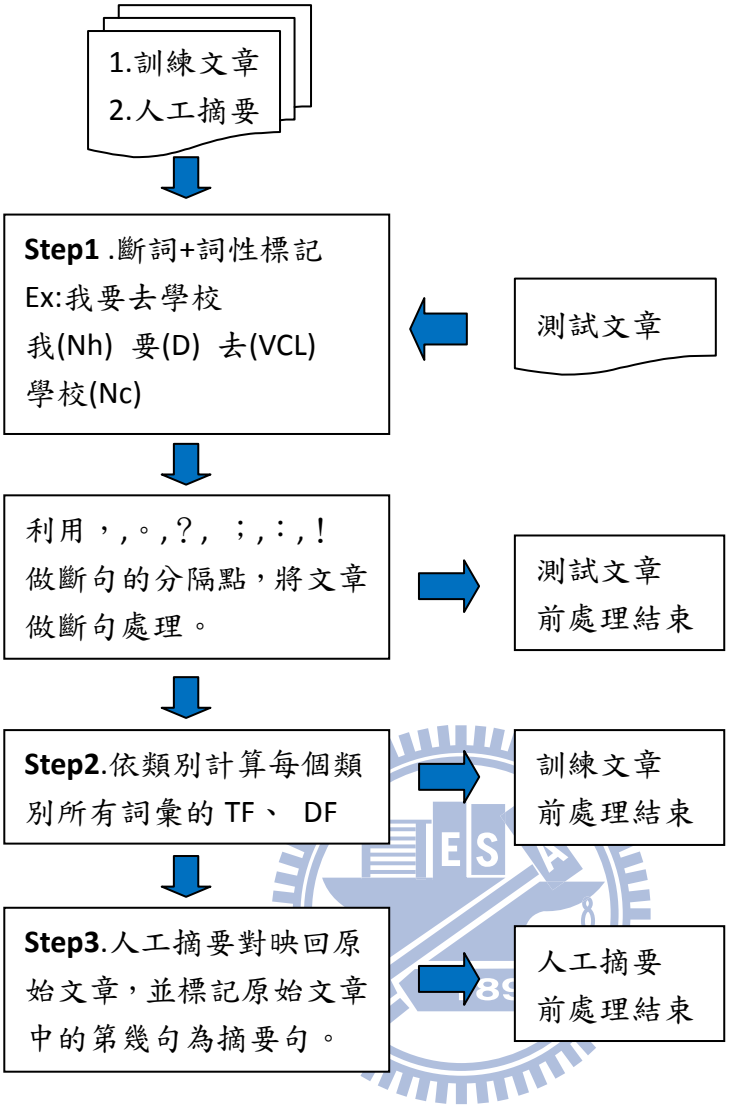


圖 3-3 :前置處理流程架構

Step1

我們使用中央研究院的斷詞系統[18]做斷詞和詞性的標記，將文章做斷詞、詞性標記以及斷句。

Step2

先計算所屬每個類別所有的詞彙重要程度(term frequency 值和 document frequency 值)[10]。

Step3

將人工摘要對映回原始文章，並標記原始文章中的第幾句為摘要句，第幾句為非摘要句。

3.2.2 英文文章前置處理

英文文章我們是收集自 DUC02 的文集，前處理部分主要分成兩個部分。

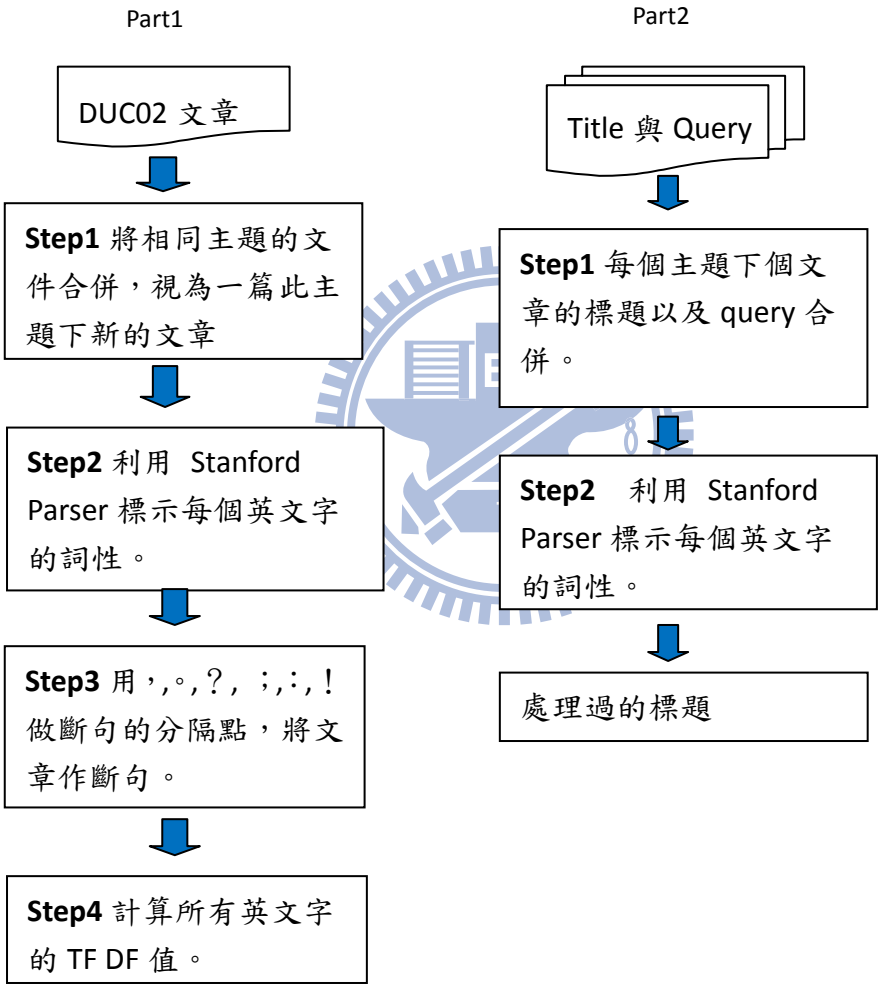


圖 3-3:英文文章前置處理流程架構

Part1

Step1

為將相同主題下的文章作合併，合併的方式為，DUC02 文集種共有 58 個主題，每個主題下分別有 7~15 篇文章，且每個主題下有相對應的人工摘要，合併結束

將有 58 篇不同主題的文章，以及相對應的主題摘要。

Step2

利用 Stanford Parser[19]，來處理詞性的標記。

Step3

利用，,。,?, ；,：,！做為斷句的分隔點，將文章斷句。

Step4

計算英文字彙的 term frequency 值和 document frequency 值[10]。

Part2

Step1

將每個主題下 7~15 篇文章標題以及主題 Query 合併成整個主題的標題。

Step2

利用 Stanford Parser 對標題上的每個英文字彙做詞性的標記[19]。

3.3 系統所使用的基本摘要方法

在本論文裡，我們使用 10 個基本且重要的摘要方法作為 AdaBoost 裡的基本分類器。

(1)Term Frequency(TF)分類方法[9][10]

首先，必須針對收集來的不同類別 Yahoo 新聞文章{政治、社會、旅遊、健康}，分別計算每個類別所有詞彙的出現頻率。

Ex: 在政治類文章中 立法院在政治類 50 篇文章中總共出現 10 次，監察院出現 20 次，則 TF(立法院)=10，TF(監察院)=20。

表 3-1 政治類、健康類 TF 表

立委(N) 45	買到(Vt) 2
質疑(Vt) 12	健康(Vi) 35
人力(N) 9	購買(Vt) 3
計畫(N) 14	是(Vt) 200
教部(N) 2	時候(N) 7
嚴格(Vi) 6	縣政府(N) 1
標題(N) 50	農業(N) 1
教育部(N) 12	處長(N) 3
推出(Vt) 3	林景和(N) 3
培育(Vt) 1	指出(Vt) 40
今天(N) 19	提到(Vt) 1
立法院(N) 23	聯想到(Vt) 1
引發(Vt) 7	辛辣(Vi) 1
朝野(N) 10	刺鼻(Vi) 1

表 3-1 為政治類和健康類部分詞彙的 Term-frequency，有了各類別詞彙的出現頻率，接下來將要計算各個類別，所有詞彙的出現頻率平均值，來當作此基本摘要法的門檻值，TF-threshold。一個 Term Frequency (TF)分類方法的回傳值，取決於這個句子裡面所有詞彙出現頻率的加總除以句子詞彙的個數跟 TF-threshold 做比較，如果大於，則回傳 1，小於，則回傳-1。

$$\text{TF-weak-classifier(S)}=\begin{cases} 1, \text{if } \left(\sum_{\text{word}\in S}\text{TF(word)}\Big/|S| > \text{TF}-\text{threshold}\right) \\ -1, \text{if } \left(\sum_{\text{word}\in S}\text{TF(word)}\Big/|S| < \text{TF}-\text{threshold}\right) \end{cases}$$

word ∈ S ： 代表所有屬於此句子裡的所有詞彙，此詞彙皆屬於{ Na、Nb、Nc、VA、VB、VC、VE、VH、VHC、VK }

TF(word):代表這個詞彙的出現頻率值。

|S|:句子裡詞彙個數此詞彙皆屬於{ Na、Nb、Nc、VA、VB、VC、VE、VH、VHC、VK }

(2)Document-Frequency(DF) 分類方法[9][10]

將收集來的不同類別 Yahoo 新聞文章{政治、社會、旅遊、健康}，分別計算每個類別所有詞彙，在多少篇文章中出現過，將這些在某些文章中的出現次數視為每個類別詞彙的 Document-Frequency。

Ex: 在政治類文章中，在所有政治類 50 篇文章中，6 篇文章有出現 "立法院" ，5 篇文章有出現"監察院"，則 DF(立法院)=6 ，DF(監察院)=5。

表 3-2 政治類、健康類 DF 表

立委(N) 20	買到(Vt) 1
質疑(Vt) 8	健康(Vi) 15
人力(N) 2	購買(Vt) 3
計畫(N) 5	是(Vt) 45
教部(N) 2	時候(N) 7
允(Vt) 1	縣政府(N) 1
嚴格(Vi) 4	農業(N) 1
標題(N) 50	處長(N) 1
教育部(N) 2	林景和(N) 1
推出(Vt) 2	指出(Vt) 29
培育(Vt) 1	提到(Vt) 1
今天(N) 12	聯想到(Vt) 1
立法院(N) 12	辛辣(Vi) 1
引發(Vt) 6	刺鼻(Vi) 1
朝野(N) 5	味道(N) 2

表 3-2 為政治類和健康類部分詞彙的 Document-frequency，當計算完各類別詞彙的文章出現頻率後，就可分別計算各個類別的，Document-Frequency(DF) 分類方法門檻值，DF-threshold。我們將各類別每個詞彙的 DF 值加總取平均值，當作我們的 DF-threshold。

所以，一個 Document-Frequency(DF) 分類方法的回傳值取決於此句子裡每個詞

彙的文章出現頻率值加總，再除以句子詞彙的個數，是否大於 DF-threshold ，大於，回傳 1，小於，回傳-1。

$$\text{DF-weak-classifier(S)}=\begin{cases} 1, \text{if } \left(\sum_{\text{word}\in S} \text{DF(word)} \Big/ |S| > \text{DF} - \text{threshold}\right) \\ -1, \text{if } \left(\sum_{\text{word}\in S} \text{DF(word)} \Big/ |S| < \text{DF} - \text{threshold}\right) \end{cases}$$

DF(word):代表word這個詞彙的在多少篇文章出現過。

word ∈ S ： 代表所有屬於此句子裡的所有詞彙，此詞彙皆屬於{ Na、Nb、Nc、VA、VB、VC、VE、VH、VHC、VK }

|S|:句子裡詞彙個數，此詞彙皆屬於{ Na、Nb、Nc、VA、VB、VC、VE、VH、VHC、VK }

(3)標題詞彙(Keyword)和查詢字(Query)

如果句子裡有出現標題詞彙和查詢字，回傳 1，否則，回傳-1。

(4)地方詞

如果句子裡有出現地方詞，回傳 1，否則，回傳-1。

(5)年代詞

如果句子裡有出現年代詞，回傳 1，否則，回傳-1。

(6)專有名詞

如果句子裡有出現專有名詞，回傳 1，否則，回傳-1。

(7)連接詞

如果句子裡有出現連接詞，回傳 1，否則，回傳-1。

(8)動名詞出現次數

如果句子裡有出現動名詞次數超過 5 個，回傳 1，否則，回傳-1。

(9)形容詞出現次數

如果句子裡有出現形容詞次數超過 3 個，回傳 1，否則，回傳-1。

(10)句子位子

在文章中開始的前五個句子，回傳 1，否則，回傳-1。

3.4 系統概念流程

以下為詳細的系統概念例子，主要分為兩個部分 Step1，Step2

Step1 是說明本系統使用一個簡單的例子步驟來描述系統訓練文章的過程。

Step2 則是使用一個簡單的流程圖說明文章訓練結束後，輸入文章，如何使用 Step1 所訓練的模組產生系統摘要。

Step1

訓練句子	S1	S2	S3	S4	S5	S6	
人工 label	**+1**	**+1**	**+1**	**-1**	**-1**	**-1**	//+1 代表在人工摘要有出現
句子權重	**1/6**	**1/6**	**1/6**	**1/6**	**1/6**	**1/6**	
							Error rate
w1 predict	**-1**	**+1**	**-1**	**-1**	**-1**	**-1**	**2/6**
w2 predict	**+1**	**-1**	**+1**	**-1**	**-1**	**-1**	**1/6**→w2 此回合的分類器
w3 predict	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**3/6**

此回合所挑選的分類器為:w2

Error rate ε_1=1/6

$$\alpha_1 = \ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right)/2$$

Update weight $D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} e^{-\alpha_t}, \text{if } Y_i = h_t(X_i) \\ e^{\alpha_t}, \text{if } Y_i \neq h_t(X_i) \end{cases}$

訓練句子	S1	S2	S3	S4	S5	S6	
人工 label	**+1**	**+1**	**+1**	**-1**	**-1**	**-1**	//+1 代表在人工摘要有出現
句子權重	**1/8**	**3/8**	**1/8**	**1/8**	**1/8**	**1/8**	
							Error rate
w1 predict	**-1**	**+1**	**-1**	**-1**	**-1**	**-1**	**2/8**→w1 此回合的分類器
w2 predict	**+1**	**-1**	**+1**	**-1**	**-1**	**-1**	**3/8**
w3 predict	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**3/8**

此回合所挑選的分類器為 w1

Error rate $\varepsilon_2 = 2/8$

$$\alpha_2 = \ln\left(\frac{1-\varepsilon_2}{\varepsilon_2}\right)/2$$

Update weight $D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} e^{-\alpha_t}, \text{if } Y_i = h_t(X_i) \\ e^{\alpha_t}, \text{if } Y_i \neq h_t(X_i) \end{cases}$

訓練句子	S1	S2	S3	S4	S5	S6	
人工 label	**+1**	**+1**	**+1**	**-1**	**-1**	**-1**	//+1 代表在人工摘要有出現
句子權重	**5/17**	**4/17**	**5/17**	**1/17**	**1/17**	**1/17**	
							Error rate
w1 predict	**-1**	**+1**	**-1**	**-1**	**-1**	**-1**	**10/17**
w2 predict	**+1**	**-1**	**+1**	**-1**	**-1**	**-1**	**4/17**
w3 predict	**+1**	**+1**	**+1**	**+1**	**+1**	**+1**	**3/17**→w3 此回合的分類器

此回合所挑選的分類器為 w3

Error rate $\varepsilon_3 = 3/17$

$$\alpha_3 = \ln\left(\frac{1-\varepsilon_3}{\varepsilon_3}\right)/2$$

圖 3-4 系統概念流程 Step1

Step2

Key={w2,w1,w3} 和α={$\alpha_1, \alpha_2, \alpha_3$} 當作訓練模組

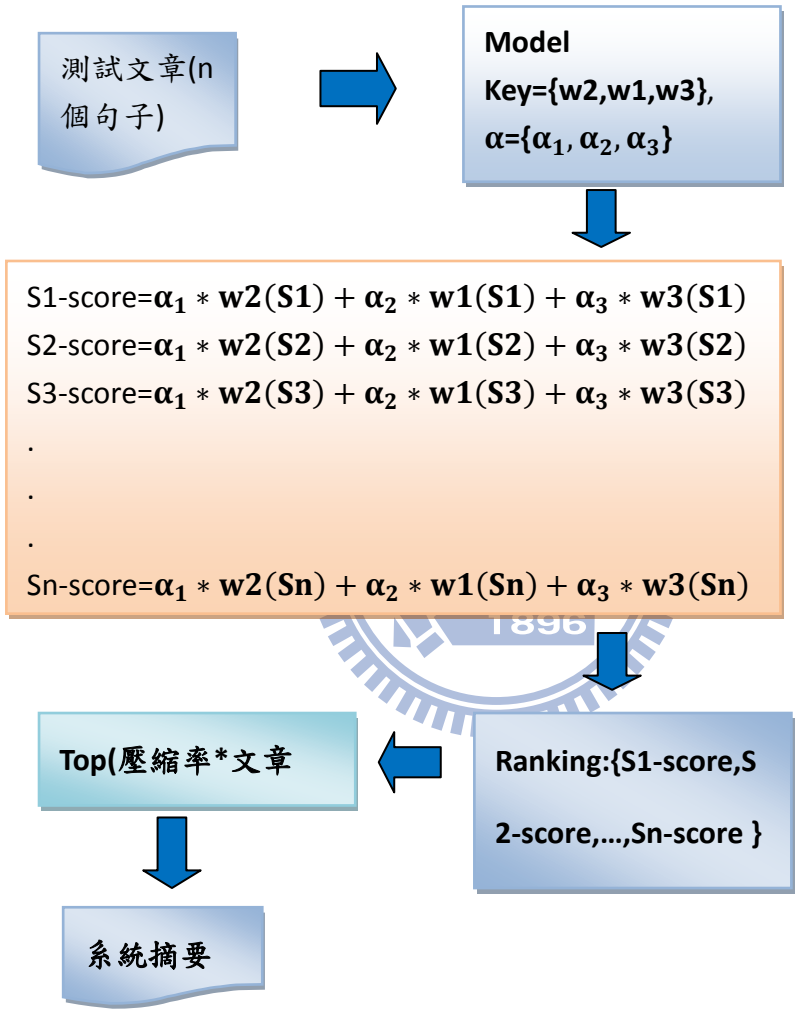


圖 3-5 系統概念流程 Step2

第四章、實驗過程與結果

4.1 實驗資料集

本實驗資料集是利用 Yahoo 新聞所收集的四個類別文章，分別為政治類、旅遊類、健康類、社會類文章各 50 篇，總共 200 篇文章當作中文的資料集。

另外本系統還使用 DUC2002 [5][6]的當作本論文的英文摘要的資料集，作為本論文更進一步的評估與測試，使用 DUC2002 而非採用其他年份主要因素為，本系統在訓練模組時，所需的人工摘要格式為 extract 而非 abstract，所以此年份的資料集較符合本系統所需。

4.2 實驗方法

4.2.1 中文方面

分別請了三位中文系的學者，來為我們做人工摘要的摘要者，

他們分別對 Yahoo 新聞的 200 篇文章做了 15%，20%，30%壓縮率的人工摘要。

而實驗方法我們使用 Precision、Recall、F-value 當作我們的摘要評估的方法。

Precision: $\frac{|S \cap H|}{|H|}$

Recall: $\frac{|S \cap H|}{|S|}$

F-value: $\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$

|S|:本系統所產生摘要句數

|H|:人工摘要句數

|S∩H|:人工摘要和系統摘要交集句數

4.2.2 英文方面

評估方面本論則是利用 ROUGE 幫我們計算 ROUGE-1，ROUGE-2，ROUGE-SU4。

ROUGE[20]是由 ISI 的 Lin 和 Hovy 所提出的評估準則。

下列算式為 ROUGE-N 的基本算法:

$$\text{ROUGE-N}=\frac{\sum_{S\epsilon HSummaries}\sum_{n-gram\epsilon S}\text{Count}_{match}(n-gram)}{\sum_{S\epsilon HSummaries}\sum_{n-gram\epsilon S}\text{Count}(n-gram)}$$

HSummaries: 表示人工摘要

$\text{Count}_{match}(n-gram)$: 表示系統摘要和人工摘要同時出現 n-gram 的個數

Count(n-gram): 表示人工摘要所出現 n-gram 的個數。

以下為 N-gram 的計算方式:

N-gram Language Models:

$P(W)= P(W_1, W_2, W_3, \dots, W_n)$

$= P(W_1)P(W_2|W_1)P(P(W_3|W_2, W_1)\dots P(W_n|W_1, W_2, W_3, \dots, W_{n-1})$

$=\prod_{i=1}^{n} P(W_i|W_1, W_2, W_3, \dots, W_{i-1})$

Ex:

(1)Unigram model：

$P(W)= P(W_1, W_2, W_3, \dots, W_n)$

$\approx P(W_1)P(W_2)P(W_3)\dots P(W_n)$

(2)Bigram model：

$P(W)= P(W_1, W_2, W_3, \dots, W_n)$

$\approx P(W_1)P(W_2|W_1)\ P(W_3|W_2)\dots P(W_n|W_{n-1})$

4.3 實驗結果

4.3.1 本系統各類別不同壓縮率實驗結果

表 4-1. 三學者取最大交集 F-value 結果

maximum	健康類	社會類	政治類	旅遊類	Average
壓縮率 15%	0.403217714	0.409789481	0.722542597	0.465179265	**0.500182264**
壓縮率 20%	0.430865875	0.451039275	0.624679255	0.419235941	**0.481455086**
壓縮率 30%	0.467568888	0.479611168	0.548401229	0.485900922	**0.495370552**
Average	**0.433884159**	**0.446813308**	**0.63187436**	**0.456772043**	

上表的結果是利用三人工摘要和本系統摘要做比較，算出系統和三人分別比對計算出的 F-value 值，取最大值做為我們的結果。

表 4-2. 三學者取平均交集 F-value 結果

average	健康類	社會類	政治類	旅遊類	Average
壓縮率 15%	0.254722644	0.289347451	0.627853674	0.330860251	**0.375696005**
壓縮率 20%	0.295539202	0.30323755	0.545859853	0.341880128	**0.371629183**
壓縮率 30%	0.360153354	0.376645013	0.481896376	0.386716033	**0.401352694**
Average	**0.303471733**	**0.323076671**	**0.551869968**	**0.353152137**	

上表的結果是利用三學者的人工摘要和本系統摘要做比較，算出系統和三人分別比對計算出的 F-value 值，取最大值做為我們的結果。

4.3.2 本系統與其他系統比較評估結果

表 4-3，表 4-4，表 4-5 分別為不同壓縮率(壓縮率 15%、壓縮率 20%、壓縮率 30%)，與其它摘要系統的比較結果。

表 4-3. 本系統與其它摘要系統的比較結果

F-value	壓縮率 15%	壓縮率 20%	壓縮率 30%
本系統	**0.50018**	**0.48145**	**0.495370552**
關鍵詞擴展+LSA (C.H Lee, Z.W. Liao,2009)[25]	0.4831925	0.4937185	0.5004
SVMs	0.1547	0.2508	0.3691
SVR(S. Li, Y. Ouyang, W. Wang, B. Sun,2007)[5]	0.3646	0.3896	0.456

4.3.3 DUC02 評估結果

表 4-6，表 4-7，表 4-8 分別為本系統以 DUC02 為文集，ROUGE-1，ROUGE-2，ROUGE-SU4 Recall，Precision ，F-value 的結果。

表 4-5. ROUGE-1 Recall,Precision,F-value 結果

DUC 2002	ROUGE-1
X ROUGE-1 Average_R	0.41201 (95%-conf.int. 0.39991 - 0.42456)
X ROUGE-1 Average_P	0.39288 (95%-conf.int. 0.38245 - 0.40313)
X ROUGE-1 Average_F	0.40139 (95%-conf.int. 0.39084 - 0.41159)

表 4-6.　ROUGE-2　Recall,Precision,F-value 結果

DUC 2002	ROUGE-2
X ROUGE-2 Average_R	0.10003 (95%-conf.int. 0.09149 - 0.10891)
X ROUGE-2 Average_P	0.09525 (95%-conf.int. 0.08771 - 0.10363)
X ROUGE-2 Average_F	0.09738 (95%-conf.int. 0.08927 - 0.10602)

表 4-7.　ROUGE-SU4　Recall,Precision,F-value 結果

DUC 2002	ROUGE-SU4
X ROUGE-SU4 Average_R	0.14845 (95%-conf.int. 0.14082 - 0.15656)
X ROUGE-SU4 Average_P	0.14131 (95%-conf.int. 0.13442 - 0.14878)
X ROUGE-SU4 Average_F	0.14449 (95%-conf.int. 0.13756 - 0.15216)

表 4-8. 本系統與其它摘要系統的比較結果

	ROUGE-1	ROUGE-2	ROUGE-SU4
CQPSum(2010)[22]	0.42241	0.17177	0.19320
Our System	**0.412**	**0.100**	**0.148**
Rel+Bigram(2009)[24]	0.403±0.076	0.180±0.076	
Rel+NoBigr(2009)[24]	0.403±0.080	0.180±0.082	
DGM(2008)[23]	0.390	0.008	
S28	0.427	0.217	0.173
S21	0.414	0.171	0.193
DUC baseline	0.411	0.210	0.166
S19	0.408	0.208	0.163
Lead	0.384	0.177	
Rel	0.389	0.178	
MMR	0.392	0.178	
GM	0.375	0.083	

S28 , S21 , S19 : the top 3 performing DUC 2002 systems From the 13 participating systems

Lead : Extract the leading sentences

Rel : learned via the SVMRank Algorithm

MMR : Maximal Marginal Relevance

DGM : document-based graph model

GM : graph model

4.3.4 結果分析:

由表 4-1 可得知,健康類、社會類、政治類、旅遊類四個類別在系統摘要 15%、20%、30%壓縮率與人工摘要結果做比較，三個壓縮率的平均正確率各別為 0.500182264、0.481455086、0.495370552，大概在五成左右，其效果是不錯的。另外，比較健康類、社會類、政治類、旅遊類四個類別在 15%、20%、30%壓縮率下的平均正確率，各別為 0.433884159　0.446813308　0.63187436　0.456772043 ;我們發現政治類新聞摘要正確率有六成多，較其他類別高出一些，根據我們的觀察，此現象跟本系統所使用的弱分類器有關;在政治類新聞的人工摘要中,專有名詞、動名詞數,相較於其他類別,在摘要出現的頻率是比較高的，所以本系統對政治類文章的分類是較為準確的。

由表 4-8 可得知，以 ROUGE-1 來看，本系統與近幾年的摘要系統作比較，是具有競爭力的，以 ROUGE-2 來看，ROUGE-2 是取決於人工摘要與系統摘要中 2-gram match 個數，由於本系統所取的特徵值較注重於句子中的資訊含量，缺乏句子間連貫性的特徵，因此在連貫性方面略顯不足。另外 ROUGE-SU4 方面，由於 ROUGE-SU4 的準確率是取決於，在人工摘要或系統摘要中，共同出現的兩個字,其距離只要不超過四個字,則可視為在人工摘要與系統摘要中 match 的單位，而非連續的兩個字，所以其效果，相較於 ROUGE-2 是略為提升的。

第五章、結論與展望

5.1 研究總結

經過多次的實驗與修正，本系統摘要結果和人工摘要比對有將近 5 成的正確率。本系統使用 AdaBoost 觀念，其摘要方法是由多個且簡單且不同面向的摘要方法所組成，除了在摘要方法的加入與移除是較為方便外，也可以利用此觀點作為簡易摘要的評估，因此，本系統的方法是附有彈性的。另外，本系統可以適用於單文件或者多文件摘要，也因為如此，實驗部分，本系統使用了雅虎新聞文章以及對應的單文件人工摘要，也使用了 DUC 文集以及相對應多文件摘要作為我們的資料文集，並且以 ROUGE 做評估，效果也有在水平之上。

5.2 未來展望

由於本系統所使用的多面向的摘要方法是容許使用者加入或者移除新的弱分類器方法，未來如果有好的摘要方法可以加入本系統，其效果是值得觀察的。目前比較缺乏的面向，便是語意分析和文章可讀性的部分，未來也許可以朝這兩個方向去琢磨。

參考文獻

[1] Yoav Freund[1] and Robert E. Schapire" A desicion-theoretic generalization of on-line learning and an application to boosting", Journal of Computer and System Sciences, August 1997.

[2] Y. Freund , R. Schapire, "A Short Introduction to Boosting" Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.

[3] L. Mason, J. Baxter, P. Bartlett, and M. Frean," A decision-theoretic generalization of on-line learning and an application to boosting", Advances in Large Margin Classifers, pages 221-246. MIT Press, Cambridge, MA, USA, 2000.

[4] T Hirao, H Isozaki, E Maeda," Extracting Important Sentences with Support Vector Machines" ,Proceedings of the 19th, 2002.

[5] S. Li, Y. Ouyang, W. Wang, B. Sun "Multi-document Summarization Using Support Vector Regression" Proceedings of DUC, 2007

[6] Document Understanding Conference (DUC) URL：http://duc.nist.gov/

[7] P. Over ,W. Liggett "Introduction to DUC-2002 an Intrinsic Evaluation of Generic News Text Summarization Systems" National Institute of Standards and Technology

[8] M. Hu, A. Sun, and E. Lim, " Comments-Oriented Document Summarization:Understanding Documents with Readers' Feedback" ,SIGIR Conference 20-24 July 2008, Singapore

[9] J. Ramos " Using TF-IDF to Determine Word Relevance in Document Queries", 2001.

[10] S. Yohei ," Sentence Extraction by tf/idf and Position Weighting from Newspaper", Proceedings of the Third NTCIR Workshop, 2003.

[11] A.h Bawakid and M. Oussalah" A Semantic Summarization System", University of Birmingham at TAC 2008.

[12]Y. Gong and X. Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis" (2001)

[13] D. K. Evans, K. McKeown, J. L. Klavans” Similarity-based Multilingual Multi-Document Summarization”, IEEE Transactions on Information(2005)

[14] Y.T. Chen, H.S. Chiu, H.M. Wang, B. Chen,” A Unified Probabilistic Generative. Framework for Extractive Spoken Document. Summarization,Interspeech, 2007.

[15] D McDonald, H Chen”Using sentence-selection heuristics to rank text segments in TXTRACTOR”,Proceedings of the 2nd ACM/IEEE-CS, 2002

[16] T Minka, J Lafferty,” Expectation-Propagation for the Generative Aspect Model” ,Proceedings of the 18th Conference on Uncertaint, 2002.

[17] R. Polikar, “Ensemble Based Systems in Decision Making”, IEEE Circuits and Systems Magazine, vol.6, no.3, pp. 21-45, 2006.

[18] 中央研究院資訊科學研究所詞庫小組中文斷詞系統
URL：http://ckipsvr.iis.sinica.edu.tw/

[19] Stanford Parser
URL：http://nlp.stanford.edu:8080/parser/

[20] C.Y. Lin” ROUGE: a Package for Automatic Evaluation of Summaries”, In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004.

[21] Penn Treebank II Tags
URL：http://bulba.sdsu.edu/jeanette/thesis/PennTags.html

[22] E. Lloret and M. Palomar,” Challenging Issues of Automatic Summarization: Relevance Detection andQuality-based Evaluation”,2010.

[23] X. Wan,” An Exploration of Document Impact on Graph-Based Multi-Document Summarization”,2008.

[24] A. F. T. Martins and N. A. Smith ” Summarization with a joint model for sentence extraction and compression”, from ACL Workshops ,2009

[25] C.H Lee, Z.W. Liao ” Automatic Text Summarization System for Chinese News,NCTU Institutional Repository(2009)