

國立交通大學

多媒體工程研究所

碩士論文

多文件摘要系統基於**Mutual Reinforcement**原理



Multi-Document Summarization System Based on  
Mutual Reinforcement Principle

研究生：楊瑞敏

指導教授：李嘉晃 教授

中華民國九十九年六月

# 多文件摘要系統基於 Mutual Reinforcement 原理

Multi-Document Summarization System Based on

Mutual Reinforcement Principle

研 究 生：楊瑞敏

Student : Ruin-Min Yang

指 導 教 授：李嘉晃

Advisor : Chia-Hoang Lee

國 立 交 通 大 學

多 媒 體 工 程 研 究 所



碩 士 論 文

A Thesis

Submitted to Institute of Multimedia 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

# 多文件摘要系統基於 Mutual Reinforcement 原理

學生：楊瑞敏

指導教授：李嘉晃 教授

國立交通大學資訊學院 多媒體工程研究所碩士班

## 摘要

根據研究報告指出，網際網路的蓬勃發展造成每年產生的數位化文件與影像等資料之總數皆呈倍數成長。為了有效率地了解這些電子文件的資訊，本論文發展自動摘要系統將這些大量的數位化文件去蕪存菁，在不流失其原本的資訊的條件下，讓使用者快速且有效地了解這些資訊的內容。

本論文所提出的自動摘要系統考慮了三個不同面向來對句子作評分以作為挑選摘要句子的依據：1. 字詞與句子之間的關係；2. 標題與句子之間的關係；3. 句子與句子之間的關係。在對句子評分之前，本系統利用 Alignment 演算法與 Mutual Reinforcement 原理移除資料集中資訊量較低的句子，以避免這些低資訊量的句子被選取成摘要句子。而上述所提及的三個不同面向則是分別利用 HITS 演算法、餘弦相似度計算方法與 PageRank 演算法來實現。


本論文使用的資料集為 DUC 資料集，其為英文資料集且組成文件為新聞類文章。根據 ROUGE 評估工具的評估結果顯示，本摘要系統所產生的系統摘要達到不錯的效能。

# Multi-Document Summarization System Based on Mutual Reinforcement Principle

Student : Ruin-Min Yang Advisor : Prof. Chia-Hoang Lee

Institute of Multimedia Engineering  
College of Computer Science  
National Chiao Tung University

## Abstract

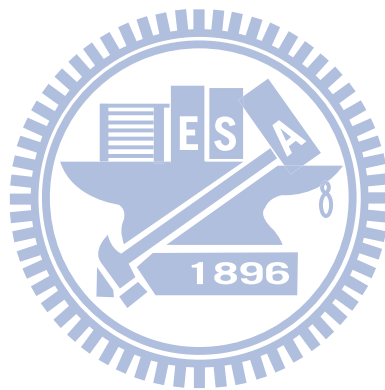


According to the research report, the rapid development of the Internet results in the amount of the digital document, video, or other data to grow in double rate per year. In order to find out the information of these electronic files efficiently, this thesis develops an automatic summarization system to sieve out the non-information data of digital documents. Therefore, users can find out the contents of information efficiently without losing the meaning of the original documents.

The automatic summarization system proposed in this thesis considers three different aspects for the sentence scoring: first, the relationship between words and sentences; second, the relationship between the titles and sentences; finally, the relationship between sentences and sentences. Before the sentences scoring, this summarization system uses Alignment algorithm and Mutual Reinforcement Principle to remove the sentences that have fewer

information on the original dataset to avoid these sentences with fewer information to be selected as a part of the summary. The HITS algorithm, the cosine similarity calculation methods and the PageRank algorithm are employed respectively to achieve the above three different aspects.

The dataset used in this thesis is the DUC dataset, and the constituent documents of the DUC dataset are the English news articles. The evaluation results of the evaluation tools ROUGE show the performance of the summary generate by this summarization system is good.



## 誌謝

能夠順利地完成這篇碩士畢業論文，最需要感謝的就是我的指導教授—李嘉晃教授。在我遇到瓶頸的時候，教授總是有耐心地教導我，給我一個正確的方向，這讓我的研究能順利地完成。教授對我的照顧不僅止於課業上，教授也時常關心我的生活，讓我能無憂無慮地進行研究。

再來我要感謝的是阿豪學長與建良學長，這兩位學長協助我完成了整篇論文的方向。對於我的問題，學長們也總是不厭其煩地回答我。我真的非常感謝學長們這兩年來的指導。要感謝的人很多，不管是最會解決問題的大洲，還是時常跟我一起討論問題的小紀、小鐘。還有在我做研究苦悶時，帶給我很大歡樂的學弟們，真的很謝謝大家在我研究所這段期間的幫忙。

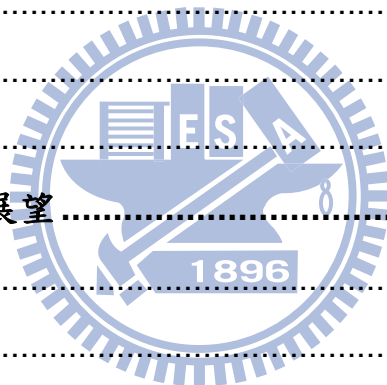
最後要感謝的是一直在背後默默支持著我的家人，尤其是我的父母親，他們總是無怨無悔地陪伴我這兩年來的求學生涯。不論是喜怒哀樂，我的家人們總是一起與我分享與承擔。

在念研究所的這兩年中，真的很謝謝所有曾經幫助過我的人，真的很謝謝你們的支持與關心，謝謝！

# 目錄

摘要 .....	iii
<b>Abstract</b> .....	<b>iv</b>
誌謝 .....	vi
目錄 .....	vii
圖目錄 .....	ix
表目錄 .....	x
<b>第一章、緒論</b> .....	<b>1</b>
1.1 研究背景與動機 .....	1
1.2 研究目的與方法 .....	1
1.3 論文架構與綱要 .....	2
<b>第二章、相關研究</b> .....	<b>3</b>
2.1 摘要系統分類 .....	3
2.2 摘要方法介紹 .....	5
2.3 DUC 資料集介紹 .....	7
2.4 向量空間模型介紹 .....	9
2.5 Alignment 演算法介紹 .....	11
2.6 Mutual Reinforcement 原理介紹 .....	12
2.7 HITS 演算法介紹 .....	14
2.8 PageRank 演算法介紹 .....	17
2.9 ROUGE 評估工具介紹 .....	19
<b>第三章、系統設計</b> .....	<b>22</b>
3.1 基本概念 .....	22
3.2 系統架構 .....	23
3.3 前置處理 .....	24

3.3.1 詞性標記處理 .....	24
3.3.2 字詞還原處理 .....	26
3.3.3 同義詞合併處理.....	27
3.3.4 常用字詞移除處理.....	29
3.4 低資訊量句子移除 .....	30
3.5 候選摘要句子評分 .....	32
3.5.1 特徵值 1: 字詞與句子之間的關係.....	33
3.5.2 特徵值 2: 標題與句子之間的關係.....	34
3.5.3 特徵值 3: 句子與句子之間的關係.....	36
3.6 後置處理.....	38
<b>第四章、實驗結果與討論 .....</b>	<b>39</b>
4.1 實驗資料集 .....	39
4.2 實驗方法.....	39
4.3 實驗結果.....	41
<b>第五章、結論與未來展望 .....</b>	<b>45</b>
5.1 研究總結 .....	45
5.2 未來展望 .....	45
<b>參考文獻 .....</b>	<b>47</b>





## 圖目錄

圖 2.3-1 DUC 資料集的基本架構 .....	7
圖 2.3-2 DUC 資料集的實例.....	8
圖 2.4-1 向量空間模型 .....	10
圖 2.6-1 二分圖 $G(T, S, W)$ .....	13
圖 2.7-1 Good Authority 和 Good Hub .....	14
圖 2.7-2 HITS 演算法的關係矩陣之實例 .....	16
圖 2.7-3 HITS 演算法的迭代過程 .....	16
圖 2.8-1 PageRank 公式的說明圖示 .....	17
圖 2.8-2 PageRank 演算法的超連結資訊實例 .....	19
圖 2.8-3 PageRank 演算法的迭代過程.....	19
圖 3.2-1 摘要系統的主要架構流程圖 .....	23
圖 3.5.1-1 Good Sentence 和 Good Term .....	33
圖 3.5.3-1 改寫 PageRank 公式的說明圖示 .....	37

## 表目錄

表 2.3-1 DUC 資料集的基本組成 .....	8
表 2.5-1 Alignment 演算法.....	12
表 2.7-1 HITS 演算法 .....	15
表 2.8-1 PageRank 演算法.....	18
表 3.3.1-1 Stanford Parser 的實例 .....	25
表 3.3.2-1 Porter Stemmer 的轉換規則 .....	26
表 3.3.2-2 Porter Stemmer 的實例 .....	27
表 3.3.3-1 Wordnet Synsets 的實例.....	28
表 3.3.4-1 常用字詞列表 .....	29
表 3.3.4-2 移除常用字和執行字詞還原過程的實例.....	29
表 3.4-1 本摘要系統的 Alignment 演算法.....	31
表 3.4-2 本摘要系統的 Mutual Reinforcement 演算法 .....	32
表 3.5.1-1 本摘要系統改寫的 HITS 演算法 .....	34
表 3.5.2-1 本摘要系統的 Title Similarity 演算法 .....	35
表 3.5.3-1 本摘要系統改寫的 PageRank 演算法 .....	38
表 4.1-1 DUC 資料集的基本組成 .....	39
表 4.2-1 候選摘要句子的評估結果 .....	41
表 4.2-2 DUC 2002 的評估結果 .....	42
表 4.2-3 DUC 2002 評估結果與其他系統之比較 .....	42
表 4.2-4 DUC 2006 的評估結果 .....	43
表 4.2-5 DUC 2006 評估結果與其他系統之比較 .....	43
表 4.2-6 DUC 2007 的評估結果 .....	44
表 4.2-7 DUC 2007 評估結果與其他系統之比較 .....	44

# 第一章、緒論

## 1.1 研究背景與動機

根據全球資訊儲存設備與資訊管理系統的領導廠商EMC公司於西元2009年委託知名研究機構IDC進行全球創造或複製的數位資訊總量與總類估算結果報告，其所做的「經濟緊縮，數位世界膨脹 (As the Economy Contracts, the Digital Universe Expands)」研究報告指出，西元2008年全球所創造的數位資訊量比原先預估的資訊量還要高出3%，即一千六百萬 GigaBytes，而此西元2008年全球所產生的資訊量竟然高達4865.22億GigaBytes。IDC估算此數位世界正以每18個月成長一倍的速度產生數位資訊，這是非常驚人的資訊量。

有鑑於此，如何有效地管理這些數位化資訊的自動化系統已成為重要的研究議題。在此，我們特別針對數位化文件的部分加以研究，並希望此研究能夠幫助人們在此資訊爆炸的時代中面對這麼大量的資訊時，能夠快速且有效地了解這些資訊的重點內容。而為了要達到快速且有效地了解大量資訊的內容，本論文的目的就是使用摘要系統的技術從這些數位化文件中擷取出重要的資訊，使得在這個時間就是金錢的時代，幫助人們在最短的時間獲取最多的資訊。

## 1.2 研究目的與方法

摘要系統的發展目的為從大量的原始文件中擷取其中較重要且較小量的資訊，以讓使用者在最短的時間即可了解該原始文件的重點所在。而一般來說，系統摘要必須要符合下列特性：可讀性、流暢性、簡潔性、概括性與客觀性等各種特性。

而本論文的研究方法可以分為下列四個部分：

第一個部分為對輸入文件作前置處理的動作，這個部分相當地重要。因為

若是沒有對輸入文件作前置處理動作的話，則可能會影響之後的低資訊量句子移除動作與評分演算法，所以在這個部份我們提出了六種不同的前置處理。

第二個部分為個別對每一篇文章作低資訊量句子移除的動作。在這個部分是希望將那些資訊量較低的句子移除後，使得之後的相似度運算、評分演算法等計算可以不受那些低資訊量句子的影響。在此，我們利用了兩種方法來移除低資訊量的句子。

第三個部分為本研究的重點，即對上個部分所留下的較有資訊量句子作評分的動作。在這個部份，我們將在資訊檢索中相當有名的網頁排序演算法轉換成對文件中的句子排序演算法，並在下個部分會利用此排序結果產生系統摘要。在此，我們利用了三種方法分別對句子作評分的動作。

第四個部分為後置處理動作，即為對預備作為摘要的句子再作一次的挑選動作，使得最後產生的系統摘要真正富含資訊的。

### 1.3 論文架構與綱要

本論文的章節分配如下：

第一章：敘述本研究之背景與動機、目的與方法。第二章：相關研究，詳細介紹本研究之相關研究與參考資料，如：DUC 資料集介紹、向量空間模型介紹、Alignment 演算法介紹、Mutual Reinforcement 原理介紹、HITS 演算法介紹、PageRank 演算法介紹、ROUGE 評估工具介紹等。第三章：系統設計，將本研究之摘要系統的整體架構、基本概念與摘要方法作完整介紹。第四章：實驗結果與討論，將本研究之摘要系統產生的系統摘要與人工摘要利用評估工具作效能的評估，除此之外，也會跟其他摘要系統作效能的評估。第五章：結論與未來展望，將本研究之摘要系統成果做總結，並提出結論與探討未來研究之方向。

## 第二章、相關研究

自動摘要系統的研究先驅為 IBM 公司的 Luhn [ 1 ]，其在西元 1950 年代開始研究自動摘要系統。此階段的自動摘要方法只是初步地利用字詞的各種不同之特徵值來作為挑選句子以作為摘要的依據。而之後才陸續有研究來分析推理文件中的結構、模板、語法、詞性或語義等特徵值。除此之外，隨著機器學習、機率統計模型、語言學、認知心理學等各種領域的研究發展，自動摘要系統也隨之進入了多元化方法的時代。

### 2.1 摘要系統分類

摘要系統根據其不同的輸入、目標與輸出，有下列各式各樣的分類，其分類型態如下[2][3]：

#### ■ 輸入文件的不同

根據輸入文件的大小不同，可分為單文件(Single-Document)摘要系統與多文件(Multi-Document)摘要系統。單文件摘要系統的輸入即為單篇文件；多文件摘要系統的輸入則為一群有相同主題的多篇文件。以近年的研究看來，多文件摘要系統是目前的趨勢，同時也較單文件摘要系統來的有挑戰性。而本摘要系統即為多文件的摘要系統。

根據輸入文件的語言不同，可分為單一語言(Monolingual)摘要系統、多語言(Multilingual)摘要系統與跨語言(Cross-Lingual)摘要系統。這三者的差別如下：單一語言摘要系統的輸入文件與輸出摘要為同一語言；多語言摘要系統的輸入文件與輸出摘要也為相同語言，只不過此輸入文件與輸出摘要都包含了多種語言，除此之外，此摘要系統必須能夠同時對多種語言的字詞和文法等進行分析；跨語言摘要系統則是輸入文件與輸出摘要為不同的語言，此摘要系統通常需要依靠翻譯系統來協助。而本摘要系統為單一語言的摘要系統。

根據輸入文件的形式不同，可分為文本(Text)摘要系統與多媒體(Multi-Media)摘要系統。文本摘要系統的輸入即為純文字的文件；而多媒體摘要系統的輸入則可能為圖像、語音、視訊等文件。而本摘要系統為文本的摘要系統。

根據是否需要額外的訓練資料，可分為非監督型(Unsupervised)摘要系統與監督型(Supervised)摘要系統。非監督型摘要系統並不額外需要訓練資料，也不需要對其訓練資料的機器學習過程；監督型摘要系統則是需要額外的訓練資料和其使用的機器學習方法。而本摘要系統為非監督型的摘要系統。

#### ■摘要的目標不同

根據摘要的功能不同，可分為資訊性(Informative)摘要系統與指示性(Indicative)摘要系統。資訊性摘要系統為貼切地傳達輸入文件的各種訊息並意圖可以用此摘要來取代整個輸入文件；指示性摘要系統只是判斷輸入文件是關於何種主題的，並不像資訊性摘要系統會傳達輸入文件的訊息，其目的只是讓使用者決定要不要仔細閱讀此輸入文件。而本摘要系統即為資訊性的摘要系統。

根據摘要的內容不同，可分為一般性(Generic)摘要系統與使用者導向(User-Oriented)摘要系統。一般性摘要系統為將輸入文件中提及的所有資訊作為摘要；使用者導向摘要系統則是需要使用者額外輸入Query，並希望其系統摘要依據此Query所提供的使用者喜好等資訊產生的。而本摘要系統可以依照是否有使用者所下的Query而為一般性的摘要系統或使用者導向的摘要系統。

#### ■摘要的輸出不同

根據摘要的形式不同，可分為提取式(Extractive)摘要系統與合成式(Abstractive)摘要系統。提取式摘要系統為從輸入文件中挑選出其中的句子作為摘要；合成式摘要系統的一般架構為分析輸入文件後，利用生成文章的各種技術而產生摘要。而本摘要系統即為提取式的摘要系統。

## 2.2 摘要方法介紹

上個小節介紹的是摘要系統的分類，上述各式各樣的分類都可以應用在各種不同的摘要方法上。在這個小節裡，我們將介紹常見的摘要方法，其可以粗略地分成下列幾種方法[4][5]：

■ 統計學(Statistical)方法：利用統計下列各種特徵：字詞頻率、Cue Phrase、專有名詞、標題字詞、TF-IDF、Entropy、Mutual Information、句子位置、句子長度、句子結構、段落位置等各種特徵來作為摘取式摘要句子的評分依據。

■ 語意式(Semantic)方法：利用文件中的段落結構與其字詞語意來建立語彙鏈結(Lexical Chain)；建立此語彙鏈結時通常需要 Parser、Tagger 與 Wordnet 中的同義詞、上下位詞之幫助。接著利用此語彙鏈結來建立階層式的主題概念，並依據此階層式主題概念來挑選出該篇文章的主要之主題。

■ 分群(Clustering-Based)方法：利用分群的各種演算法將文件中所有的句子依照其語意的相關程度來分群，一個群別即代表了一個主題。常見的分群方法有 K-mean Clustering、Hierarchical Clustering、Spectral Clustering 等方法。接著挑選每一群的代表句子來作為其摘要，在挑選代表句子時通常會使用 MMR 的方法來將摘要句子之間的重覆性達到最小。另外還有跟分群方法類似的分類(Classification-Based)方法，而分類方法是只將文件中的所有句子分成兩類，即取其作為摘要或不取其作為摘要這兩類。

■ 圖形理論(Graph Theoretic)方法：通常將文件中的句子視為圖形中的節點，而節點與節點之間的無向邊或有向邊則通常由句子與句子之間共同出現字詞數量或相似度所決定。若所建立的圖形明顯地被分割成幾個區塊，則此時就像上述所介紹的分群方法一樣，分別取出每一個區塊的代表句子作為摘要。而區塊中的代表句子取法如下：若其中一個節點的分支度很高，則此代表此節點所

代表的句子跟許多句子具有一定的相似度，即此句子涵括了這些句子的資訊，故可將此句子視為重要的句子。除了可以建立句子與句子之間的圖形模型之外，還可以建立文章與句子之間或文章與文章之間的圖形模型。在下面的相關研究章節中，所介紹的 HITS 演算法與 PageRank 演算法即皆是一種圖形理論的方法。

■ 機器學習(Machine Learning)方法：首先必須取得預備作為訓練資料的資料集與其相對應的摘取式人工摘要，接著可利用此摘取式的人工摘要來為資料集中的每一個句子標記為 Summary Sentence 或 Non-Summary Sentence，並藉此訓練各種所定義的特徵進而得到一個訓練模型。而之後就可利用此訓練模型來判斷測試資料集中的每一個句子是否可挑選為摘要句子。常見的機器學習方法有 SVM(Support Vector Machine)、SVR(Support Vector Regression)等。

而機率模型(probabilistic model)方法也是一種機器學習的方法，其方法為利用各種數學機率模型的假設來訓練、學習各種參數或其機率，例如：字詞與文章共同出現的機率、句子的生成機率等。而常見的機率模型方法有 HMM (Hidden Markov Model)、TMM (Topical Mixture Model)、LDA (Latent Dirichlet Allocation) 等方法。

■ 潛藏語意分析(Latent Semantic Analysis)方法：利用線性代數中的奇異值分解將文章(或句子)-字詞關係矩陣分解成三個矩陣，並可依據此分解矩陣判斷出重要的文章(或句子)與字詞。此被命名成LSA的原因是即使文章之間沒有共同出現的字詞，此方法亦可以將那些具有語意相關的文章分成一群，而通常出現在相關領域中的字詞也會被分成一群。而LSA利用找出主要且相互正交的奇異向量空間中的代表句子作為其摘要，其中主要的奇異向量空間能確保其代表句子必定會包含文章所提及主題的資訊，相互正交的奇異向量空間能確保其代表句子之間沒有冗餘度。除了上述的基本LSA，LSA還有很多變形，其中最著名的即為pLSA(Probability Latent Semantic Analysis)。



## 2.3 DUC 資料集介紹

本論文所使用的資料集為 DUC(Document Understanding Conference)資料集，是由美國國家標準與技術研究所(National Institute of Standards and Technology，簡稱 NIST)及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。NIST 除了推動此標準資料集之外，也推動了一系列的研討會議來幫助使用此標準資料集的參加者發表其研究方法並同時評估這些方法的效能。

他們從西元 2001 年開始推動，到目前為止，已提供了 DUC 2001 到 DUC 2010 等十個不同的標準資料集。這些標準資料集的主要目標為讓自然語言處理相關的研究人員能夠利用這些較大規模的實驗來發展自動摘要系統及評估其系統效能。

本論文所使用的資料集為 DUC 2002、DUC 2006 及 DUC 2007，其中 DUC 2006 及 DUC 2007 中的文章是由 AQUAINT 語料庫所提供。而 AQUAINT 語料庫的組成文章為新聞類文章，分別來自於美聯社、紐約時報及新華社。

下圖為 DUC 資料集的基本架構：

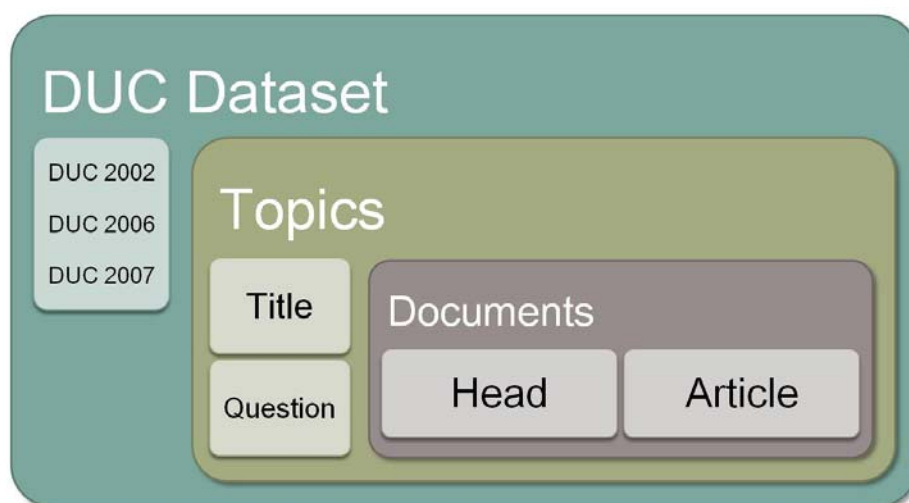


圖 2.3-1 DUC 資料集的基本架構

由上圖可知，對每一個不同的主題來說，本多文件摘要系統所能使用的資源有 Title、Question 和這個主題所包含文章的 Head、Article。

下圖為 DUC 2006 中的其中一篇文章，其相對架構名稱請對照圖 2.3-1。

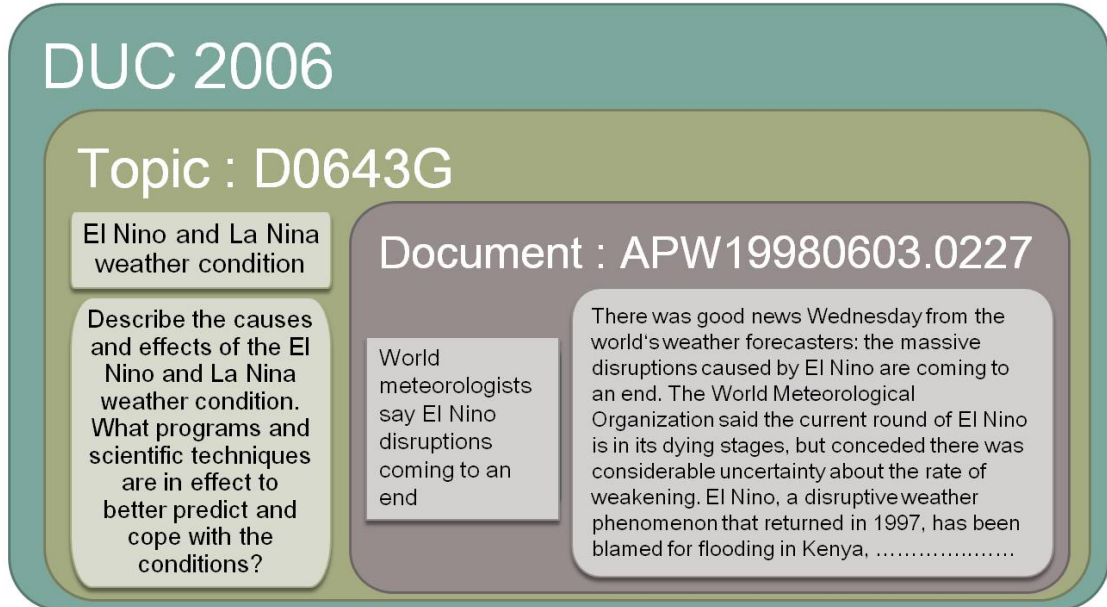


圖 2.3-2 DUC 資料集的實例

下圖展示了 DUC 2002、DUC 2006 和 DUC 2007 資料集的基本組成元件：

	DUC 2002	DUC 2006	DUC 2007
# of Topics	59	50	45
# of Documents in Each Topic	5~15	25	25
Head	○	○	○
Title	×	○	○
Question	×	○	○
Words of the Manual Summary	200	250	250
# of the Manual Summary	2	4	4

表 2.3-1 DUC 資料集的基本組成

由上圖可知，本多文件摘要系統的目標為根據每一個不同的主題產生 200 或 250 個字左右的系統摘要。在 DUC 2006 和 DUC 2007 中對於每一個不同的主題額外提供了 Title 及 Question，其目的是希望我們能夠針對這個 Question

產生相對應的答案，而此答案即可視為系統摘要。而之後此系統摘要跟所提供的人工摘要互相做評估的實驗，其中 DUC 所提供的人工摘要為人工看完所有文章並針對 Question 所撰寫的合成式摘要。

## 2.4 向量空間模型介紹

向量空間模型( Vector Space Model )是由 Gerard Salton 等人[ 6 ]所提出並將之應用於 SMART 資訊檢索系統。向量空間模型的概念其實很簡單，就是將文章投影至向量空間中，而每一篇文章都視為一個向量，其中向量的維度則是由所有文章相異字詞的數量所決定。至於文章投影向量中的元素則是可以簡單地用 0 或 1 來表示，即若此篇文章有出現字詞  $i$ ，則此文章投影向量的元素  $i$  為 1；反之，則為 0。

除了上述簡單的向量表示法之外，也可以使用加權的向量表示法使得重要的字詞得到較大的值，最常用的加權向量表示法非 TF - IDF( Term Frequency - Inverse Document Frequency)莫屬，其中 TF 為字詞頻率，IDF 為逆文件頻率。若是某個字詞在某篇文章中出現多次，代表此字詞在此文章被提及多次，則我們可以將此字詞視為重要的字詞，即給其較大的權重值，此部分即為 TF。然而若是某個字詞在資料集中的每一篇文章都出現且出現多次，代表此字詞可能是常用字詞或此字詞較不重要且不具識別能力，則我們希望給其較小的權重值。

TF-IDF 的計算公式如下：

$$\text{TF-IDF}_i = \text{tf}_i \times \log\left(\frac{|D|}{|i \in D|}\right)$$

其中  $\text{tf}_i$  為字詞  $i$  在此篇文章出現的次數， $|D|$  為資料集中所包含的文章數， $|i \in D|$  為字詞  $i$  在資料集的文章中出現的篇數。由上述公式可得知，若是字詞  $i$

在此篇文章出現的次數愈多，則 TF-IDF 公式中的前項值會愈大，即給予較高的加權值；若是字詞  $i$  出現在愈多的文章中，即表示字詞  $i$  愈沒有識別效果，則 TF-IDF 公式中的後項值會愈小，即給予較低的加權值。

此時，我們已經可以將文章投影到向量空間模型。假設資料集中有  $N$  篇文章，總共有  $M$  個相異字詞，而其投影向量空間模型的圖示如下：

$$\begin{array}{l}
 D_1 = \begin{array}{|c|c|c|c|c|} \hline W_{1,1} & W_{1,2} & W_{1,3} & \dots & W_{1,M} \\ \hline \end{array} \\
 D_2 = \begin{array}{|c|c|c|c|c|} \hline W_{2,1} & W_{2,2} & W_{2,3} & \dots & W_{2,M} \\ \hline \end{array} \\
 \dots \\
 D_N = \begin{array}{|c|c|c|c|c|} \hline W_{N,1} & W_{N,2} & W_{N,3} & \dots & W_{N,M} \\ \hline \end{array}
 \end{array}$$

圖 2.4-1 向量空間模型

其中  $D_i$  為第  $i$  篇文章， $W_{ij}$  為其字詞  $j$  在文章  $i$  中的 TF-IDF 或任何 TF-IDF 的變形。

上面敘述都是將整篇文章投影到向量空間模型上，而我們也可以將同主題中每篇文章的每一個句子照上述方法投影到此向量空間模型上。本論文採用的向量空間模型即為後者。

有了此向量空間模型後，我們就可以在此向量空間上做相關的向量運算。對於自然語言處理及資料檢索來說，最常使用的向量運算為相似度計算，其中最常見的計算方法有 Dot、Cosine、Dice、Jaccard 相似度計算等。在本論文中，我們使用的是 Cosine Similarity 來計算句子之間的相似度，故在此僅介紹 Cosine 相似度，其計算公式如下：

$$\text{Cos\_Sim} ( S_i, S_j ) = \frac{S_i \cdot S_j}{\|S_i\| \times \|S_j\|}$$

其中  $S_i$  和  $S_j$  分別為第  $i$  個和第  $j$  個句子所投影的向量，“ $\cdot$ ”為內積運算，

$\|\cdot\|$  為 2-norm 運算。此公式即是在計算  $S_i$  和  $S_j$  向量之間的餘弦夾角，因為我們設定的向量元素皆大於等於 0，所以算出來的相似度之值會介於 0 到 1 之間。若  $\text{Cos\_Sim}(S_i, S_j) = 0$ ，則代表  $S_i$  和  $S_j$  之間的向量互為垂直，即他們之間的敘述相似度為零；若  $\text{Cos\_Sim}(S_i, S_j) = 1$ ，則代表  $S_i$  和  $S_j$  之間的向量重合，即代表他們幾乎為同一個敘述。

## 2.5 Alignment演算法介紹

Alignment 演算法是由 Massih Amini 和 Nicolas Usunier [ 7 ] 所提出的並應用在文件摘要系統中，其目標是想要移除文章中沒有資訊量的句子。因為在 DUC 2006 和 DUC 2007 中對於每一個主題都有提供 Question 並希望系統摘要是指對此 Question 所產生的，所以此篇作者提出了下列假說：

呼應 Question 的句子必定會與 Question 有最大的語義相似度。

接著他們利用 Marcu's Alignment Algorithm [ 8 ] 來實現上面的假說，即此演算法會使得剩餘的句子統統向 Question 對齊。其演算法簡述如下：

Input : Topic Question and a Document

Output : The set of candidate sentences

Algorithm :

do{

Remove a sentence from the document set such that the similarity between the Question and the remaining sentence of the document set is higher

} until the similarity between the Question and the document set  
is lower

表 2.5-1 Alignment 演算法

此篇論文的相似度計算方法跟我們前面介紹的不太一樣，其相似度算法為

$$\text{Sim}(S, Q) = \frac{\sum_{w \in S \cap Q} c(w, S) \times c(w, Q)}{\sum_{w \in S} c^2(w, S) \sum_{w \in Q} c^2(w, Q)}$$

其中S為此文件中的Sentence，Q為Topic Question， $c(w, Z) = \text{tf}(w, Z) \times \log(\text{df}(w))$ ，Z可為Sentence或Question， $\text{tf}(w, Z)$ 為word w在Z集合中的頻率， $\text{df}(w)$ 為word w的文件頻率。

此演算法的輸出為摘要候選句子集合，根據此論文的實驗，此摘要候選句子集合與原先的句子集合相比的結果為摘要候選句子集合只遺失了一小部分的資訊量，但是摘要候選句子集合所包含的句子數量卻僅為原先的句子集合的四分之一。



## 2.6 Mutual Reinforcement 原理介紹

Mutual Reinforcement Principle是由Hongyuan Zha [ 9 ]所提出，其原理可簡述如下：

重要的字詞會出現在多個重要的句子之中；

而重要的句子必定會包含多個重要的字詞。

對於資料集中的每一篇文章來說，有其組成的句子和字詞。我們定義其字詞集合為 $T = \{t_1, t_2, \dots, t_M\}$ ，其句子集合為 $S = \{s_1, s_2, \dots, s_N\}$ 。接著我們可利

用這兩個集合來建立一個二分圖(Bipartite Graph)， $G(T, S, W)$ 來描述字詞與句子之間的關係，其中 $W$ 為此二分圖的邊集合， $W_{M \times N} = [w_{ij}]$ ，而 $w_{ij}$ 為字詞 $i$ 在句子 $j$ 出現的頻率或加權頻率。其圖示如下：

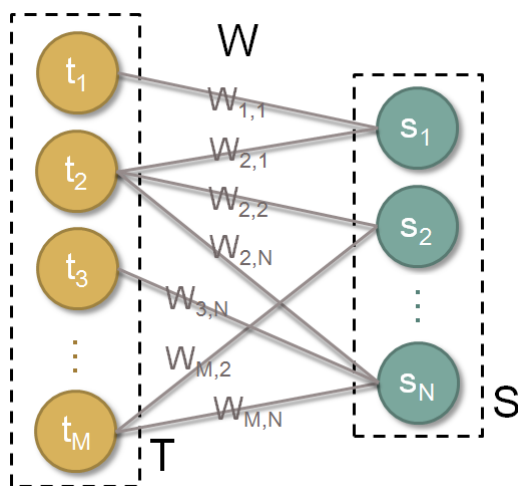


圖 2.6-1 二分圖  $G(T, S, W)$

根據Mutual Reinforcement原理，字詞與句子的重要程度是相互關聯的。我們將此重要程度轉換成分數大小，意即我們要去計算字詞和句子的分數並用此分數來代表其重要程度。為了方便起見，我們定義 $u(t_i)$ 與 $v(s_j)$ 分別代表字詞與句子的分數數值。然而根據Mutual Reinforcement原理，字詞和句子的分數是相互貢獻的，我們可以將此敘述轉換成下列數學式子：

$$u(t_i) \propto \sum_{t_i \sim s_j} w_{ij} \times v(s_j)$$

$$v(s_j) \propto \sum_{s_j \sim t_i} w_{ij} \times u(t_i)$$

其中 $t_i \sim s_j$ 代表 $t_i$ 與 $s_j$ 之間是有邊相連的。我們也可以將所有的 $u$ 、 $v$ 分別集成向量 $U$ 、 $V$ ，則上列式子就可以轉換成下列矩陣的運算式子：

$$U = \frac{1}{\sigma} W V$$

$$V = \frac{1}{\sigma} W^T U$$

其中 $1/\sigma$ 為比例常數。我們很容易可以看出 $U$ 為 $W$ 的奇異值 $\sigma$ 所對應的左奇異向量， $V$ 為 $W$ 的奇異值 $\sigma$ 所對應的右奇異向量。其中若是我們取 $\sigma$ 為 $W$ 中的最大奇異值，則可以保證 $U$ 、 $V$ 中所有的元素皆為非負。

## 2.7 HITS演算法介紹

HITS( Hypertext Induced Topic Selection )演算法是由J.M. Kleinberg[ 12 ]所提出，其主要概念是希望依據網頁之間的超連結資訊來將網頁排序，其技術應用在CLEVER搜尋引擎上。

首先，我們先定義兩個名詞：Hub與Authority。對於某個主題而言，Authority為提供重要且可靠的主題資訊之網頁，Hub為包含超連結至與主題相關的Authority之網頁。Hub與Authority之間的關係符合上一節所介紹的Mutual Reinforcement 原理：

A Good Hub指向許多Good Authorities；

A Good Authority被許多Good Hubs所指向。

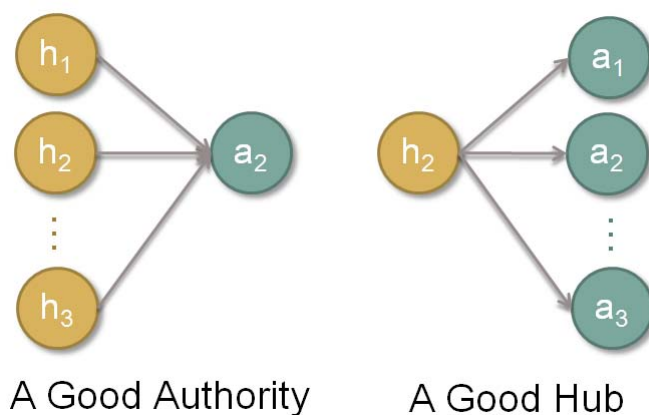


圖 2.7-1 Good Authority 和 Good Hub

因為網頁不能像字詞和句子一樣清楚地劃分成兩個集合，所以也就不能像上一節一樣建立一個二分圖來描述網頁跟網頁之間的關係。故我們必須對每一個



網頁分別去計算其Authority和Hub的分數，以辨明此網頁是否為Authority或Hub，此即為HITS演算法的目標。

為了說明HITS演算法，我們建立一個集合  $P = \{ p_1, p_2, \dots, p_N \}$  為網頁集合，其集合個數為  $N$  個。接著建立一個關係矩陣  $M_{N \times N} = [ m_{ij} ]$ ，其中若  $p_i$  指向  $p_j$ ，則  $m_{ij} = 1$ ；反之， $m_{ij} = 0$ 。而網頁的Authority和Hub分數分別用  $A_{N \times 1} = [ a_1, a_2, \dots, a_N ]^T$  和  $H_{N \times 1} = [ h_1, h_2, \dots, h_N ]^T$  表示。

其演算法如下：

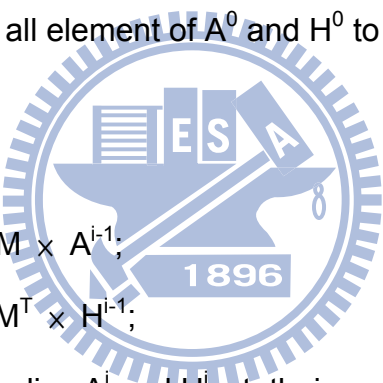
Input : P、M

Output : A、H

Algorithm :

Step 1. Initialize all element of  $A^0$  and  $H^0$  to 1,  $i = 0$

Step 2. do{



$i++;$

$H^i = M \times A^{i-1};$

$A^i = M^T \times H^{i-1};$

Normalize  $A^i$  and  $H^i$  s.t. their squares sum to 1;

} until A and H converge

表 2.7-1 HITS 演算法

其中  $H^i = M \times A^{i-1}$  可看成每個  $p_i$  的Hub分數都是由  $p_i$  所指向的  $p_j$  之Authority分數所貢獻；而  $A^i = M^T \times H^{i-1}$  則可看成每個  $p_i$  的Authority分數都是由  $p_i$  所被反指向的  $p_j$  之Hub分數所貢獻。

下面我們舉個例子說明此演算法：考慮網頁集合  $P = \{ p_1, p_2, p_3 \}$ ，關係矩

陣  $M_{N \times N} = [ m_{ij} ]$ ，其中  $M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ ，我們以下列圖示表示此關係矩陣：

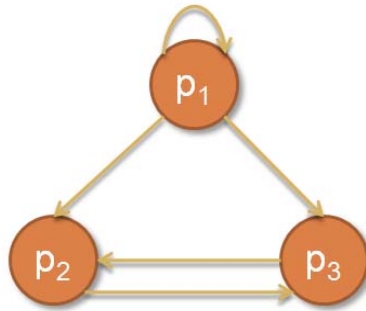


圖 2.7-2 HITS 演算法的關係矩陣之實例

以上是輸入的資料，接著我們開始迭代此演算法：

步驟一：初始化A與H為 $A = [1\ 1\ 1]^T$ ， $H = [1\ 1\ 1]^T$ 。步驟二：此迭代過程以下列圖示表示：

$$\begin{array}{l}
 A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0.333 \\ 0.667 \\ 0.667 \end{bmatrix} \rightarrow \begin{bmatrix} 0.469 \\ 0.625 \\ 0.625 \end{bmatrix} \rightarrow \begin{bmatrix} 0.451 \\ 0.631 \\ 0.631 \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} 0.459 \\ 0.628 \\ 0.628 \end{bmatrix} \\
 H = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0.905 \\ 0.302 \\ 0.302 \end{bmatrix} \rightarrow \begin{bmatrix} 0.870 \\ 0.348 \\ 0.348 \end{bmatrix} \rightarrow \begin{bmatrix} 0.889 \\ 0.323 \\ 0.323 \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} 0.888 \\ 0.325 \\ 0.325 \end{bmatrix}
 \end{array}$$

圖 2.7-3 HITS 演算法的迭代過程

由上面迭代過程可知，A與H最終將會收斂於一個固定的值，而此A與H為輸出資料，即根據A與H所相對應的分數高低而完成對網頁集合的排序順序。接著，搜尋引擎即會依照此排序順序回傳分數較高的網頁至客戶端。

根據此篇論文的定理3.1闡述，若此演算法可以無限次地迭代，則此演算法必定可以收斂於一個固定的值。而定理3.2則說明其Authority的分數A會收斂至 $M^T M$ 的最大特徵值所對應的特徵向量；Hub的分數H會收斂至 $M M^T$ 的最大特徵值所對應的特徵向量。根據實驗，大概只需要20次的迭代過程就可以達到其收斂值。

## 2.8 PageRank演算法介紹

PageRank 演算法是由 Sergey Brin 和 Lawrence Page [ 16 ]所提出，其主要目標是希望依據網頁之間的超連結資訊來將網頁排序，其技術應用在 GOOGLE 搜尋引擎上。

PageRank 演算法的基本概念為

Good web page 必定被許多 Good web page 所指向。

Sergey Brin 和 Lawrence Page 依據上述的基本概念與隨機衝浪模型(Random Surfer Model)建立了下列 PageRank 的計算公式：

$$PR(A) = (1 - d) + d \times \left( \frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \dots + \frac{PR(T_n)}{C(T_n)} \right)$$

其中 PR(A)為網頁 A 的 PageRank 計算值，d 為 damping factor，其 d 值必須介於 0 到 1 之間， $T_i$ 為有超連結指向網頁 A 的網頁，PR( $T_i$ )即為網頁  $T_i$  的 PageRank 計算值， $C(T_i)$ 為網頁  $T_i$  所對外超連結至其他網頁的網頁總數量。此公式可用下列圖示來說明：

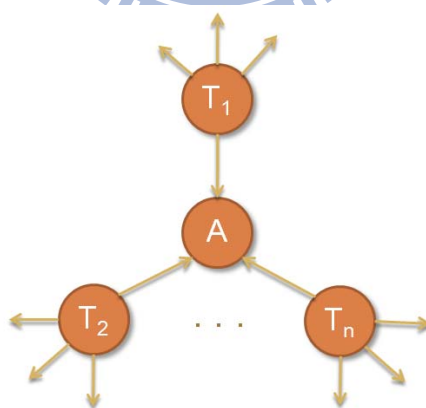


圖 2.8-1 PageRank 公式的說明圖示

由上列公式得知，某個網頁的 PageRank 值都是由其他超連結至自己的網頁所貢獻的，即若是有較多的網頁超連結至此網頁，則此網頁必定會得到較高的 PageRank 值。除此之外，若是  $C(T_i)$  的值愈大，則  $PR(T_i)/C(T_i)$  的值愈小，其

意思為若是  $T_i$  超連結到愈多的網頁，則  $T_i$  貢獻給網頁 A 的分數愈少，但是此貢獻度主要還是依據  $T_i$  的 PageRank 值來決定。至於前項  $(1-d)$  則為 damping factor，可以把它想成這是一個隨機瀏覽至此網頁的機率，意為使用者在瀏覽網頁時不可能一直隨著網頁的超連結資訊來瀏覽網頁，必定會有中斷目前網頁並跳至跟目前網頁毫無相關網頁的情形，故才有這個隨機瀏覽的機率值。根據此篇論文闡述，此  $d$  值為 0.85 時可達到不錯的效果，即隨機瀏覽機率值為 0.15。

每個網頁的 PageRank 值並非如上面所述的其為一次計算值，而是一個迭代過程。跟 HITS 演算法一樣，每個網頁的初始值皆為 1，並透過不斷地迭代而得到最後的 PageRank 值，此迭代演算法如下：

一樣考慮網頁集合  $P = \{p_1, p_2, \dots, p_N\}$ ，其集合個數為  $N$  個，還有這些網頁之間的超連結資訊，如所需的對外超連結的網頁統計值。

Input : P、The Link Information that Need

Output : PR of all web pages

Algorithm :

Step 1. Initialize  $PR(p_1)^0, PR(p_2)^0, \dots, PR(p_N)^0$  to 1、 $k=0$

Step 2. do{

$k++$ ;

for  $i = 1$  to  $N$

$$PR(p_i)^k = (1-d) + d \times \sum_{\forall T_j \rightarrow p_i} \frac{PR(T_j)^{k-1}}{C(T_j)^{k-1}}$$

end

} until PR of all web pages converge

**表 2.8-1 PageRank 演算法**

下面我們舉個例子說明此演算法：考慮網頁集合  $P = \{p_1, p_2, \dots, p_4\}$ ，其網頁的超連結資訊如下圖表示：

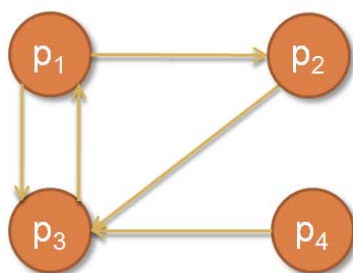


圖 2.8-2 PageRank 演算法的超連結資訊實例

步驟一：為了方便，我們將每個網頁的PageRank值以矩陣形式表示， $PR_{4 \times 1} = [1, 1, 1, 1]^T$ ；每個網頁的對外超連結統計值也以矩陣形式表示， $C_{4 \times 1} = [2, 1, 1, 1$

$]^T$ ；而網頁與網頁之間的超連結關係也以矩陣形式表示， $M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ 。

步驟二：此迭代過程以下列圖示表示：

$$PR = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0.575 \\ 2.275 \\ 0.15 \end{bmatrix} \rightarrow \begin{bmatrix} 2.084 \\ 0.575 \\ 1.191 \\ 0.15 \end{bmatrix} \rightarrow \begin{bmatrix} 1.163 \\ 1.036 \\ 1.652 \\ 0.15 \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} 1.49 \\ 0.783 \\ 1.577 \\ 0.15 \end{bmatrix}$$

圖 2.8-3 PageRank 演算法的迭代過程

由上面迭代過程可知，PR最終將會收斂於一個固定的值，而此PR為輸出資料，即根據PR所相對應的分數高低而完成對網頁集合的排序順序。接著，搜尋引擎即會依照此排序順序回傳分數較高的網頁至客戶端。這邊需要注意的是，PageRank值只是目前GOOGLE排序網頁的其中一個依據。

## 2.9 ROUGE評估工具介紹

ROUGE(Recall-Oriented Understudy for Gisting Evaluation)評估工具是由Chin-Yew Lin[ 18 ]所發展的，其主要目標是要幫助摘要系統能自動地評估其效能

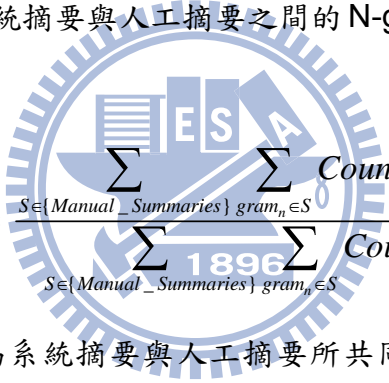
。其自動評估的方法有下列五種：

- ROUGE-N: N-gram based co-occurrence statistics
- ROUGE-L: LCS based statistics
- ROUGE-W: Weighted LCS based statistics that favors consecutive LCSes
- ROUGE-S: Skip-bigram based co-occurrence statistics
- ROUGE-SU: Skip-bigram plus unigram-based co-occurrence statistics

因為 NIST 在評估不同摘要系統的效能時，只評估 ROUGE-N 與 ROUGE-SU 這兩個自動評估方法的值，所以我們下面只介紹這兩個自動評估方法的計算公式。

。註：NIST 除了會評估摘要系統的自動評估值之外，也會用人工去評估各個摘要系統所產生的系統摘要之連貫性、簡潔性、語法性、可讀性和內容品質等。

ROUGE-N 為計算系統摘要與人工摘要之間的 N-gram 字詞交集數量之評估值，其計算公式如下：


$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Manual\_Summaries}\}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in \{\text{Manual\_Summaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

其中  $\text{Count}_{\text{match}}(gram_n)$  為系統摘要與人工摘要所共同出現的 N-gram 字詞， $\text{Count}(gram_n)$  為人工摘要中 N-gram 字詞的總數量。上述公式為 Recall 版本，另外 ROUGE-N 也會計算其 Precision 與 F-measure 版本的評估值。NIST 所規定的 N 值為 1 與 2，代表我們需計算其系統摘要之 1-gram 與 2-gram 字詞的 Recall、Precision 與 F-measure 評估值。

因為 ROUGE-SU 是 ROUGE-S 的延伸版本，所以我們先介紹 ROUGE-S。ROUGE-S 跟 ROUGE-2 的計算方式差不多，只不過 ROUGE-2 所計算的 2-gram 字詞必須要是連續的，而 ROUGE-S 則允許所計算的 2-gram 字詞可以跳過其中間的字詞，即此 2-gram 字詞不需要是連續的。其計算公式如下：

$$\text{ROUGE-S} = \frac{\text{Skip}_2(X, Y)}{C(m, 2)}$$

其中  $X$  為人工摘要， $Y$  為系統摘要， $Skip_2(X,Y)$  為系統摘要與人工摘要所共同出現的 2-gram 字詞，而此 2-gram 字詞不需要是連續出現的， $m$  則為人工摘要所包含的字詞總數， $C(m,2)$  則為組合函數，即為對  $m$  個相異的字詞中不重複地取出 2 個字詞的組合總數。

此 ROUGE-S 會有一個潛藏的問題，就是 Skip-bigram 中的 2-gram 字詞是有前後順序的，我們考慮下面的例子：

$S_1$ : police killed the gunman

$S_2$ : gunman the killed police

若根據 ROUGE-S 的公式，則  $S_1$  與  $S_2$  之間的 ROUGE-S 評估值會為 0。這似乎與我們對於此兩個句子之間的相似度認知有一段差距。為了解決此問題，作者將 ROUGE-S 延伸為 ROUGE-SU，其延伸為在 ROUGE-S 中額外對 1-gram 字詞作考量。



## 第三章、系統設計

### 3.1 基本概念

本摘要系統為多文件摘要系統，意即我們要對多篇文章的內容進行研究與分析，再依照此研究與分析的結果產生壓縮率極高的系統摘要。本摘要系統使用的資料集為 DUC 資料集，根據相關研究的 DUC 介紹得知，我們可使用的資源有針對主題的 Title 與 Question，針對各篇文章的 Head 與 Article。本摘要系統將會充分利用這些資源來產生出效能不錯的系統摘要。

上段敘述有提到本摘要系統必須產生壓縮率極高的系統摘要，這代表著我們需要研究且分析的文章句數遠比要產生的系統摘要的句數還要多很多。然而，在文章中，並非每一個句子都蘊含了關於主題的資訊，亦或有一些句子可能在描述較為細節的部分。對於僅能為 200 或 250 個字詞的系統摘要來說，上述那些句子肯定不該被選取為系統摘要。因此，為了避免這個情況以及避免那些無資訊量的句子影響句子評分演算法的結果，本摘要系統利用 Alignment 演算法來將這些無資訊量的句子移除，而剩餘的句子集合則作為候選摘要句子。

接著，本摘要系統要對候選摘要句子作評分的動作，以作為我們挑選摘要句子的依據。我們考慮了三個不同的面向，分別為 1. 字詞與句子之間的關係、2. 標題與句子之間的關係和 3. 句子與句子之間的關係。我們針對這三個不同的面向分別利用 HITS 演算法、餘弦相似度計算方法和 PageRank 演算法來實現對句子的評分動作，這個部分我們留到下面的章節再作詳細的介紹。我們將這三個不同面向的評分結果作線性組合，而得到最終的句子評分之分數，再依此分數的高低來選取系統摘要的句子。



## 3.2 系統架構

下列流程圖為本摘要系統的主要架構流程圖：

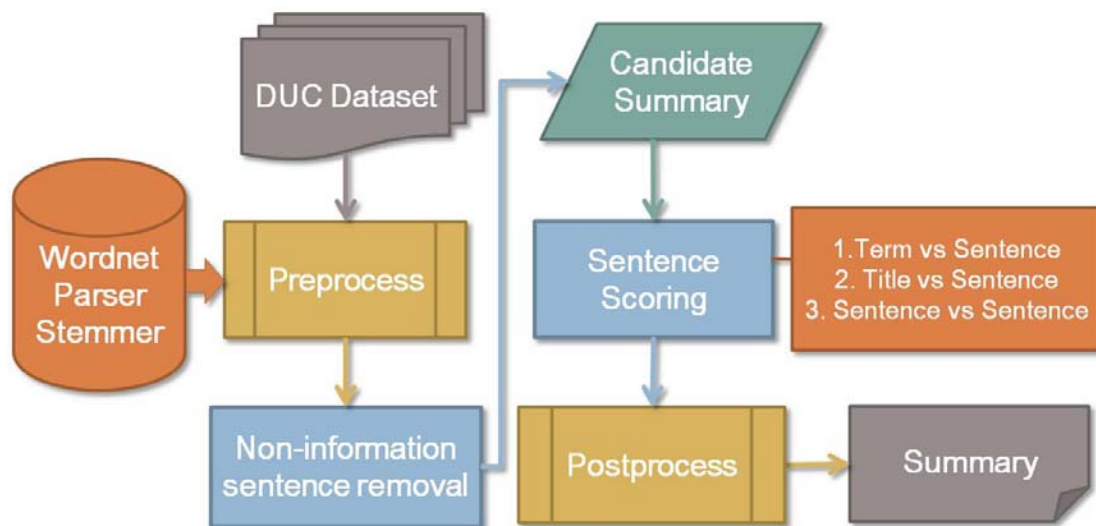


圖 3.2-1 摘要系統的主要架構流程圖

由上面的架構流程圖得知，本摘要系統主要分為四個部分：第一個部分是對資料集作前置處理的動作；第二個部分是利用 Alignment 演算法作無資訊量句子的移除動作使得剩餘二分之一的句子集合成為候選摘要句子；第三個部份是本摘要系統的重點所在，我們完整地考慮三種不同的面向來對句子作評分動作，接著再將這三個不同的分數線性組合以成為我們最後選取摘要的分數依據；第四個部分則是根據第三部分的分數高低來挑選句子且依照一些後置處理的規則來過濾這些候選摘要句子。最後，產生本摘要系統的系統摘要。

這四個部分的詳細過程與概念，將於下面的章節介紹。

### 3.3 前置處理

我們必須先對 DUC 資料集中的每一篇文章做一些前置處理的動作，才可更準確地將文章投影到向量空間中。我們分別對文章做了斷句處理、雜訊移除處理、詞性標記處理、字詞還原處理、同義詞合併處理和常用字詞移除處理。

對文章做斷句處理是以句點、驚嘆號、問號、分號等標點符號做為斷句的依據。我們不以逗點作為斷句的依據是因為希望產生的系統摘要能具有流暢性。

對文章做雜訊移除處理是因為 DUC 資料集的文章中包含一些無意義但頻繁出現的字詞，例如：&QL、&UR、&LR 等。為了避免這些雜訊影響其向量空間的建立，故我們收集並統計此類的雜訊，並在此前置處理時就將之移除。

剩下的前置處理部分我們將以下列四個小節做詳細的介紹。

#### 3.3.1 詞性標記處理

我們希望將文章中的每個字詞都標記其詞性，以方便我們做後續挑選字詞為動詞、名詞的處理。我們使用的剖析工具為由 Stanford Nature Language Processing Group 所研發的 Stanford Parser [ 31 ]，其為一種機率統計模型的詞性剖析程式。

而 Stanford Parser 的輸出結果包含詞性標記、語法結構和語法依存關係。因為我們不需要使用語法依存關係，所以在此省略不介紹。其中詞性標記的輸出結果是我們較感興趣的部分。

下面為 Stanford Parser 的實例：

#### Input

**There was good news Wednesday from the world's weather forecasters: the massive disruptions caused by El Nino are coming to an end.**

### Output (詞性標記)

There/EX was/VBD good/JJ news/NN Wednesday/NNP from/IN the/DT world/NN ` ` s/JJ weather/NN forecasters/NNS :/: the/DT massive/JJ disruptions/NNS caused/VBN by/IN El/NNP Nino/NNP are/VBP coming/VBG to/TO an/DT end/NN ./.

### Output (語法結構)

```
(ROOT
  (S
    (S
      (NP (EX There))
      (VP (VBD was)
        (NP (JJ good) (NN news))
        (NP (NNP Wednesday))
        (PP (IN from)
          (NP
            (NP (DT the) (NN world))
            (`` `)
            (NP (JJ s) (NN weather) (NNS forecasters))))))
      (: :)
      (S
        (NP
          (NP (DT the) (JJ massive) (NNS disruptions))
          (VP (VBN caused)
            (PP (IN by)
              (NP (NNP El) (NNP Nino))))))
        (VP (VBP are)
          (VP (VBG coming)
            (PP (TO to)
              (NP (DT an) (NN end))))))
        (. .)))
```

表 3.3.1-1 Stanford Parser 的實例

因為名詞與動詞是在所有詞性中較為有資訊量的詞性，所以我們可利用此詞性標記結果來擷取出名詞與動詞的字詞，並只利用這些字詞來建立文件的向量空間模型，以達到降低向量空間模型維度的效果。

### 3.3.2 字詞還原處理

我們希望將文章中的每一個字詞作正規化處理，意即將意義相同但其單複數、型態、詞性不相同的字詞全部還原成其原型，使得在作字詞統計的處理時能夠更加地正確。例如：將 `connected`、`connecting`、`connection`、`connections` 全部還原成 `connect`。

我們使用的工具為由 Martin Porter 所提出的 Porter Stemming 演算法[ 32 ]，其為對英文字詞移除常見的字詞型態及詞尾變化的演算法。

下面介紹此 Porter Stemming 演算法的轉換規則：

Step 1a	Example
<b>SSES</b> → <b>SS</b>	caresses → caress
<b>IES</b> → <b>I</b>	ponies → poni
<b>SS</b> → <b>SS</b>	caress → caress
<b>S</b> →	cats → cat
Step 1b	Example
<b>(m&gt;0) EED</b> → <b>EE</b>	agreed → agree
	feed → feed
<b>(*v*) ED</b> →	plastered → plaster
	bled → bled
<b>(*v*) ING</b> →	motoring → motor
	sing → sing
<b>AT</b> → <b>ATE</b>	conflat(ed) → conflate
<b>BL</b> → <b>BLE</b>	troubl(ed) → trouble
<b>IZ</b> → <b>IZE</b>	siz(ed) → size
...	...

表 3.3.2-1 Porter Stemmer 的轉換規則

上面轉換規則中， $(m>0)$ 代表其(CV)組合至少要出現一次，其中 C 為子音字母、V 為母音字母， $(*v^*)$ 代表此字詞至少要包含一個母音字母。因為篇幅有限，所以我們在此只簡略介紹前幾個轉換規則。

下面為 Porter Stemmer 的實例：

Input
There was good news Wednesday from the world's weather forecasters: the massive disruptions caused by El Nino are coming to an end.
Output
There <b>wa</b> good <b>new</b> <b>Wednesdai</b> from the world weather <b>forecast</b> the <b>massiv disrupt caus</b> by El Nino <b>ar come</b> to an end

表 3.3.2-2 Porter Stemmer 的實例

### 3.3.3 同義詞合併處理

因為我們使用向量空間模型來表示每個句子，所以若能將同意義的字詞合併成同一個維度，意即將這些同義的字詞視為同一個字詞，則可使得我們在相似度計算及評分演算法上的結果更加地正確。

我們使用的工具為由普林斯頓大學 George A. Miller 所領導研發的 Wordnet [33]，其為一種詞彙資料庫。他們分別為名詞、動詞、副詞和形容詞建立一個認知性的同義詞集合(Synsets)，其字詞跟字詞之間的鏈結為他們的語義或詞彙之間的關係，其關係有同義關係、反義關係、上下位關係、整體與部分關係、繼承關係等。

下面為使用 Wordnet 中 Synsets 的實例：

Input		
There was good news Wednesday from the world's weather forecasters: the massive disruptions caused by El Nino are coming to an end.		
Output		
字詞	詞性	詞性所相對應的同義詞集合
There	Adverb	thither

good	Noun	commodity trade_good goodness
	Adjective	undecomposed unspoiled unspoilt honest salutary sound serious effective in_effect in_force well right ripe dependable safe secure dear near adept expert practiced proficient skillful skilful just upright beneficial estimable honorable respectable full
	Adverb	thoroughly soundly well
news	Noun	newsworthiness intelligence tidings word
Wednesday	Noun	Midweek Wed
world	Noun	human_race humanity humankind humans mankind man worldly_concern earthly_concern populace public globe reality domain universe existence creation cosmos macrocosm
	Adjective	global planetary worldwide world-wide
weather	Noun	weather_condition conditions atmospheric_condition
	Verb	endure brave brave_out
	Adjective	upwind
massive	Adjective	monolithic monumental
by	Adverb	aside away past
EI	Noun	elevated_railway elevated_railroad elevated overhead_railway elevation altitude ALT
coming	Noun	orgasm climax sexual_climax approach approaching advent
	Adjective	approaching forthcoming upcoming
end	Noun	remainder remnant oddment conclusion close closing ending destruction death goal last final_stage terminal
	Verb	terminate stop finish cease

表 3.3.3-1 Wordnet Synsets 的實例

此時，我們將這些同義詞合併成同一維度，以降低向量空間模型的維度。

### 3.3.4 常用字詞移除處理

在閱讀文章時，會常常看到某些常用字詞，這些常用字詞通常都不帶有任何重要的意義。因此我們希望可以將這些常用字詞移除，以避免這些常用字詞影響相似度的計算及減少向量空間的維度。

我們使用的工具為 DUC 資料集所提供的常用字詞列表，其列表如下：

Common Words List						
a	about	above	across	after	again	all
almost	alone	along	also	always	am	among
an	and	another	any	anybody	anyhow	anyone
apart	are	around	as	aside	at	away
b	be	become	been	before	behind	below
beside	best	better	between	beyond	both	brief
but	by	c	can	certain	could	.....

表 3.3.4-1 常用字詞列表

下面為移除常用字詞且執行字詞還原過程的實例：

Input
<b>There was good news Wednesday from the world's weather forecasters: the massive disruptions caused by El Nino are coming to an end.</b>
Output
<b>good wednesdai world weather forecast massiv disrupt el nino come end</b>

表 3.3.4-2 移除常用字和執行字詞還原過程的實例

以上就是我們對 DUC 資料集中的所有文件所作的前置處理動作。此時，我們可以将文件投影至向量空間模型中，而且跟沒作前置處理動作時所建立的向量空間模型相比，必定會更加符合這些文件的語意資訊。

### 3.4 低資訊量句子移除

上面 3.1 小節有提到本摘要系統必須產生壓縮率極高的系統摘要，這代表著我們需要研究且分析的文章句數遠比要產生的系統摘要的句數還要多很多。然而，在文章中，並非每一個句子都蘊含了關於主題的資訊，亦或有一些句子可能在描述較為細節的部分。對於僅能為 200 或 250 個字詞的系統摘要來說，上述那些句子肯定不該被選取為系統摘要。因此，為了避免這個情況以及避免那些無資訊量的句子影響句子評分演算法的結果，本摘要系統利用 Alignment 演算法來將這些低資訊量的句子移除。

除此之外，我們相信句子中所包含的字詞數量或動詞名詞數量過少的話，其資訊量必定也不高。故我們也會在此對符合上述兩種情況的這些低資訊量的句子作移除處理的動作。

我們在上面 2.5 小節中介紹了 Alignment 演算法。在此複習一下他們所提出的假說：呼應 Question 的句子必定會與 Question 有最大的語義相似度。對於本摘要系統而言，我們希望其候選摘要句子集合能夠保留其原始句子集合一半的數量，以確保之後在計算評分演算法時有足夠的資訊。故我們將表 2.5-1 的演算法改寫如下：

**Input :** Topic Question and Topic Title 、 all Article of this Topic

**Output :** The set of candidate sentences of this Topic

**Algorithm :**

For all Article of this Topic do{

Step1. Remove the sentence contains the number of the words, nouns and verbs in this sentence which are fewer than 5, 2 and 1.

Step2. Remove on sentence from the Article set such that the



```
similarity between Title, Question and the remaining
sentences in the Article set is higher than before until
the remaining sentences in the Article set is half of the
original Article set.
```

```
}
```

表 3.4-1 本摘要系統的 Alignment 演算法

此演算法中的相似度計算方法使用的是2.4節中所介紹的餘弦相似度計算公式。

這邊需要注意的是我們在此演算法中是對同一主題內的各篇文章分別作 Alignment 的動作，並依照相似度的高低移除低資訊量的句子後，其所產生的候選摘要句子集合為原先文章的句子集合之一半。最後在輸出的部分，我們是將這同一主題內所有文章的候選摘要句子全部集成此主題的候選摘要句子集合，以方便作下一節的候選摘要句子評分的動作。

除了可利用餘弦相似度計算公式之外，我們也可以用上面2.6小節所介紹的 Mutual Reinforcement 原理對文章中的每一個句子評分，以判斷這些句子的資訊量多寡。故我們將表3.4-1的演算法改寫如下：

**Input :** Topic Question and Topic Title 、 all Article of this Topic

**Output :** The set of candidate sentences of this Topic

**Algorithm :**

For all Article of this Topic do{

Step1. Remove the sentence contains the number of the words, nouns and verbs in this sentence which are fewer than 5, 2 and 1.

Step2. Use the following formulas to calculate the sentence score matrix V, and then remove half of all sentences

with the lower score according to the sentence score matrix V.

$$U = \frac{1}{\sigma} WV$$
$$V = \frac{1}{\sigma} W^T U$$

}

表 3.4-2 本摘要系統的 Mutual Reinforcement 演算法

其中U為字詞的分數矩陣，其初始值為其TF-IDF值，但若是此字詞在Topic Question或Topic Title中出現的話，則此字詞的分數會得到額外的加權值。我們加權的原因是因為要呼應其所提出的假說。而句子分數矩陣V的初始值則皆為1。然而根據Mutual Reinforcement原理，字詞和句子的分數是相互貢獻的。

這邊也一樣需要注意的是我們在此演算法中是對同一主題內的各篇文章分別作Alignment的動作，並依照資訊量的多寡移除低資訊量的句子後，其所產生的候選摘要句子集合為原先文章的句子集合之一半。最後在輸出的部分，我們是將這同一主題內所有文章的候選摘要句子全部集成此主題的候選摘要句子集合，以方便作下一節的候選摘要句子評分的動作。

### 3.5 候選摘要句子評分

在此部分，本摘要系統要對候選摘要句子作評分的動作，以作為我們挑選摘要句子的依據。我們考慮了三個不同的面向，分別為 1. 字詞與句子之間的關係、 2. 標題與句子之間的關係和 3. 句子與句子之間的關係。我們針對這三個不同的面向分別利用 HITS 演算法、餘弦相似度計算方法和 PageRank 演算法來實現對句子的評分動作，至於詳細的評分演算法將於下面的三個小節作介紹。

### 3.5.1 特徵值 1: 字詞與句子之間的關係

我們在上面2.7小節中介紹了HITS( Hypertext Induced Topic Selection )演算法，其主要概念是希望依據網頁之間的超連結資訊來將網頁排序。在此，我們想要將此概念轉換成字詞與句子之間的關係並用之將句子排序。

在此複習一下Hub與Authority之間的關係：

A Good Hub指向許多Good Authorities。

A Good Authority被許多Good Hubs所指向。

我們將上面概念轉換成字詞與句子之間的關係，即為Mutual Reinforcement原理：

重要的字詞會出現在多個重要的句子之中；

而重要的句子必定會包含多個重要的字詞。

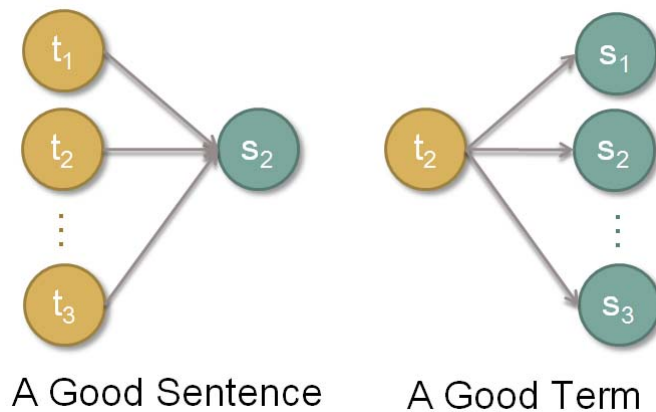


圖 3.5.1-1 Good Sentence 和 Good Term

其中  $t \rightarrow s$  代表為字詞  $t$  被句子  $s$  所包含，意即為在原本HITS演算法的超連結關係在此轉換成包含關係。

HITS演算法的目標為分別對每一個網頁去計算其Authority和Hub的分數，以辨明此網頁是否為Authority或Hub。而在此我們的目標則是去計算每一個字詞與每一個句子的分數，而兩者之間的分數是相互貢獻的。

為了說明我們轉換過後的演算法，我們建立一個集合  $S = \{s_1, s_2, \dots, s_N\}$  為句子集合，其集合個數為  $N$  個，集合  $T = \{t_1, t_2, \dots, t_M\}$  為字詞集合，其集合個數

為M個。接著建立一個包含關係矩陣 $M_{N \times N} = [m_{ij}]$ ，其中若 $s_i$ 包含 $t_j$ ，則 $m_{ij} = 1$ ；反之， $m_{ij} = 0$ 。而字詞與句子的分數分別用 $U_{M \times 1} = [u_1, u_2, \dots, u_M]^T$ 和 $V_{N \times 1} = [v_1, v_2, \dots, v_N]^T$ 表示。

其演算法改寫如下：

**Input :** The word set T, the sentence set S, the relationship matrix M

**Output :** The score of word vector U, the score of sentence vector V

**Algorithm :**

Step 1. Initialize all elements of  $U^0$  to its TF-IDF ,

Initialize all elements of  $V^0$  to 1 ,  $i = 0$

Step 2. do{

$i++$ ;

$U^i = M \times V^{i-1}$ ;

$V^i = M^T \times U^{i-1}$ ;

Normalize  $U^i$  and  $V^i$  s.t. their squares sum to 1;

} until U and V converge;

表 3.5.1-1 本摘要系統改寫的 HITS 演算法

其中 $U^i = M \times V^{i-1}$ 可看成每個 $t_i$ 的字詞分數都是由 $t_i$ 所指向的 $s_j$ 之句子分數所貢獻；而 $V^i = M^T \times U^{i-1}$ 則可看成每個 $s_j$ 的句子分數都是由 $s_j$ 所包含的 $t_i$ 之字詞分數所貢獻。這邊另外值得注意的是，我們不是將字詞的分數初始化成1，而是使用在2.4小節所介紹其相對應的TF-IDF。使用TF-IDF值更能突顯出包含多個重要字詞的句子之分數會較高，更加符合我們所定義的字詞與句子之間的關係。

由上述的演算法可知，句子的分數矩陣 V 為輸出結果，意即此為特徵值 1 所對句子的排序順序。

### 3.5.2 特徵值 2: 標題與句子之間的關係

我們在上面 2.4 小節中介紹了 Cosine Similarity 來計算句子之間的相似度，其計算公式如下：

$$\text{Cos\_Sim} ( S_i, S_j ) = \frac{S_i \cdot S_j}{\|S_i\| \times \|S_j\|}$$

在此，我們將利用上述的相似度公式來計算標題與句子之間的相似度，並用此來將句子排序。

因為文章的標題可以看成是此篇文章所討論的主題與重點所在，所以標題對於摘要系統來說是非常有資訊量的。除了標題之外，我們在 2.3 小節有介紹過 DUC 資料集還有另外提供了 Topic Title 和 Topic Question，這兩個資訊也對於摘要系統來說是富含資訊量的。

故我們將上述討論的三個富含資訊量的 Article Head、Topic Title、Topic Question 集合起來成為 New Title，接著一一去比對 New Title 與句子之間的相似度，並用此相似度的高低將句子排序，其演算法如下：

考慮一個集合  $S = \{ s_1, s_2, \dots, s_N \}$  為句子集合，其集合個數為  $N$  個。

**Input :** New Title、S

**Output :** Cos\_Sim of all sentences

**Algorithm :** for all sentence  $s_i$  in S{

$$\text{Cos\_Sim} ( \text{New Title}, S_i ) = \frac{\text{New Title} \cdot S_i}{\|\text{New Title}\| \times \|S_i\|}$$

}

**表 3.5.2-1 本摘要系統的 Title Similarity 演算法**

其中 New Title 為 Article Head、Topic Title、Topic Question 所投影至向量空間模型的向量，“·”為內積運算， $\|\cdot\|$ 為 2-norm 運算。此公式即是在計算 New Title 和  $S_i$  向量之間的餘弦夾角，因為我們設定的向量元素皆大於等於 0，所以算出來的相似度之值會介於 0 到 1 之間。而其 Cos\_Sim 相似度的值愈高，即

代表 New Title 與  $S_i$  之間的相似度愈高。

由上述的演算法可知，所有句子的 Cos\_Sim 之分數為輸出結果，意即此為特徵值 2 所對句子的排序順序。

### 3.5.3 特徵值 3: 句子與句子之間的關係

我們在上面 2.8 小節中介紹了 PageRank 演算法，其主要目標是希望依據網頁之間的超連結資訊來將網頁排序，在此，我們想要將此概念轉換成句子與句子之間的關係並用之將句子排序。

在此複習一下 PageRank 演算法的基本概念為

Good web page 必定被許多 Good web page 所指向。

我們將上面概念轉換成句子與句子之間的關係為

Good sentence 必定被許多 Good sentence 所指向。

其中上述的“所指向”代表為其兩個句子之間有一定的相似度

根據上面概念，我們改寫了 PageRank 的計算公式如下：

$$PR(A) = (1 - d) + d \times \left( \frac{PR(s_1)}{C(s_1)} + \frac{PR(s_2)}{C(s_2)} + \dots + \frac{PR(s_n)}{C(s_n)} \right)$$

其中  $PR(A)$  為句子 A 的 PageRank 計算值， $d$  為 damping factor，其  $d$  值必須介於 0 到 1 之間， $s_i$  為與句子 A 有一定的相似度的句子，即在此有  $n$  個句子與句子 A 相似， $PR(s_i)$  即為句子  $s_i$  的 PageRank 計算值， $C(s_i)$  為句子  $s_i$  與其他句子的相似度總和。此公式可用下列圖示來說明：

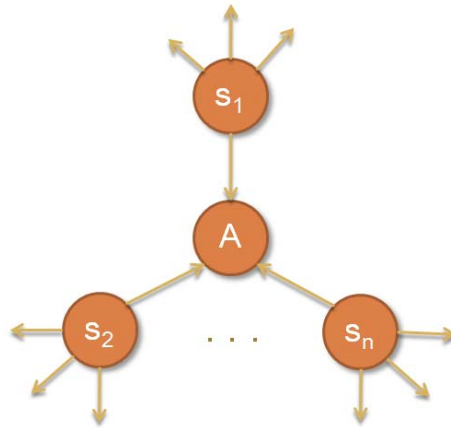


圖 3.5.3-1 改寫 PageRank 公式的說明圖示

由上列公式得知，某個句子的 PageRank 值都是根據其他句子與自己的相似度多寡所貢獻的，即若是有較多的句子與此句子相似，則此句子必定會得到較高的 PageRank 值。除此之外，若是  $C(s_i)$  的值愈大，則  $PR(s_i)/C(s_i)$  的值愈小，其意思為若是  $s_i$  與愈多的句子相似，則  $s_i$  貢獻給句子 A 的分數愈少，但是此貢獻度主要還是依據  $s_i$  的 PageRank 值來決定。

考慮一個集合  $S = \{s_1, s_2, \dots, s_N\}$  為句子集合，其集合個數為  $N$  個。還有這些句子之間的相似度資訊。此迭代演算法改寫如下：

**Input :** S、The Similarity Information that Need

**Output :** PR of all web page

**Algorithm :**

Step 1. Initialize  $PR(s_1)^0$ 、 $PR(s_2)^0$ 、 $\dots$ 、 $PR(s_N)^0$  to 1、 $k=0$

Step 2. do{

$k++$ ;

    for  $i = 1$  to  $N$

$$PR(s_i)^k = (1-d) + d \times \sum_{\forall s_j \text{ is similar with } s_i} \frac{PR(s_j)^{k-1}}{C(s_j)^{k-1}}$$

end

} until PR of all sentences converge

表 3.5.3-1 本摘要系統改寫的 PageRank 演算法

其中  $s_j$  is similar with  $s_i$  代表  $s_j$  與  $s_i$  有一定的相似度，改用更精確地方式說明，即為  $s_j$  與  $s_i$  之間的相似度高過一個門檻值。而此相似度計算方法使用的是 2.4 節中所介紹的餘弦相似度計算公式。

由上述的演算法可知，所有句子的 PageRank 之分數為輸出結果，意即此為特徵值 3 所對句子的排序順序。

### 3.6 後置處理

將 3.5 小節的三個分數經過線性組合後，再依照其所得到的加權排序順序來挑選句子並且依照下列規則來過濾這些候選摘要句子。

而此線性組合中的各個權重值則是由各個分數的評估效能所決定，意思是說我們將 3.5 小節中的三個分數各別利用 ROUGE 評估工具得到其效能結果後，再依照此效能高低決定此分數在線性組合中所對應的權重值。最後，產生 200 或 250 個字詞的系統摘要。其挑選候選摘要句子的規則如下：

1. 若是挑選到的句子長度超過 20 個字詞的話，則將此句刪除而不成為系統摘要的一部分。這是因為我們希望產生的系統摘要能具有簡潔性。
2. 對於 DUC 資料集中的同一個主題來說，其組成文章大約有 5~25 篇。而此規則即為從每一篇的組成文章中摘取一到二句的句子作為系統摘要。這是因為我們希望產生的系統摘要能具有概括性。
3. 根據上述所定義的規則 1 與規則 2 來挑選摘要句子，直到所挑選到的摘要句子集合之字詞總和達到 200 或 250 個為止



## 第四章、實驗結果與討論

### 4.1 實驗資料集

我們所使用的 DUC 資料集在上面 2.3 小節有詳細介紹過，本論文所使用的資料集為 DUC 2002、DUC 2006 及 DUC 2007，其中 DUC 2006 及 DUC 2007 中的文章是由 AQUAINT 語料庫所提供。而 AQUAINT 語料庫的組成文章為新聞類文章，分別來自於美聯社、紐約時報及新華社。

下圖展示了 DUC 2002、DUC 2006 和 DUC 2007 資料集的基本組成元件：

	DUC 2002	DUC 2006	DUC 2007
# of Topics	59	50	45
# of Documents in Each Topic	5~15	25	25
Head	○	○	○
Title	×	○	○
Question	×	○	○
Words of the Manual Summary	200	250	250
# of the Manual Summary	2	4	4

表 4.1-1 DUC 資料集的基本組成

### 4.2 實驗方法

我們在上面小節中曾提及了 Recall、Precision、F-measure，在此我們將介紹此類評估準則，其公式如下：

$$\text{Recall} = \frac{|\text{System Summary} \cap \text{Manual Summary}|}{|\text{Manual Summary}|}$$

$$\text{Precision} = \frac{|\text{System Summary} \cap \text{Manual Summary}|}{|\text{System Summary}|}$$

$$\text{F-value} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

其中 System Summary 為自動摘要系統產生的系統摘要；Manual Summary 為 DUC 資料集所提供的人工摘要。對上面這三個公式而言，其值皆是愈大愈好。對 Recall 與 Precision 來說，分子皆為系統摘要與人工摘要的交集，分母則分別為人工摘要與系統摘要。至於系統摘要與人工摘要交集的單位如何制定，則由下面詳細介紹。

我們在上面 2.9 小節中介紹了 ROUGE 評估工具，其主要目標是要幫助摘要系統能自動地評估其效能，並延續說明上面所介紹的 Recall 與 Precision。

而 NIST 在評估不同摘要系統的效能時，只評估 ROUGE-N 與 ROUGE-SU 這兩個自動評估方法，所以我們下面只複習這兩個自動評估方法的計算公式。

ROUGE-N 為計算系統摘要與人工摘要之間的 N-gram 字詞交集數量之評估值，其計算公式如下：

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Manual\_Summaries}\}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in \{\text{Manual\_Summaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

其中  $\text{Count}_{\text{match}}(gram_n)$  為系統摘要與人工摘要所共同出現的 N-gram 字詞， $\text{Count}(gram_n)$  為人工摘要中 N-gram 字詞的總數量。上述公式為 Recall 版本，另外 ROUGE-N 也會計算其 Precision 與 F-measure 版本的評估值。NIST 所規定的 N 值為 1 與 2，代表我們需計算其系統摘要之 1-gram 與 2-gram 字詞的 Recall、Precision 與 F-measure 評估值，而 Recall 與 Precision 公式中分子的交集單位即為 1-gram 與 2-gram 字詞。

因為 ROUGE-SU 是 ROUGE-S 的延伸版本，所以我們先介紹 ROUGE-S

。ROUGE-S 跟 ROUGE-2 的計算方式差不多，只不過 ROUGE-2 所計算的 2-gram 字詞必須要是連續的，而 ROUGE-S 則允許所計算的 2-gram 字詞可以跳過其中間的字詞，即此 2-gram 字詞不需要是連續的。其計算公式如下：

$$\text{ROUGE-S} = \frac{\text{Skip}_2(X, Y)}{C(m, 2)}$$

其中 X 為人工摘要，Y 為系統摘要， $\text{Skip}_2(X, Y)$  為系統摘要與人工摘要所共同出現的 2-gram 字詞，而此 2-gram 字詞不需要是連續出現的，m 則為人工摘要所包含的字詞總數， $C(m, 2)$  則為組合函數，即為對 m 個相異的字詞中不重複地取出 2 個字詞的組合總數。至於 ROUGE-SU，其為在 ROUGE-S 中額外對 1-gram 字詞作考量，而 ROUGE-SU 後面會接一個數字，此數字代表在計算不連續的兩個字詞中間可以跳過多少個字詞，例：ROUGE-SU4 為計算系統摘要與人工摘要中不連續的兩個字詞之重複數量，而這兩個字詞中間最多可以跳過四個字詞。

### 4.3 實驗結果

我們對系統架構中的第二個部分產生的候選摘要句子集合利用 ROUGE 自動評估工具作效能評估，其評估結果如下表，其中使用的資料集為 DUC 2006：

Size of the Candidate	Non-Information Sentence Removal Algorithm	ROUGE-1 Recall of the Candidate
1/2	Alignment	0.919
	Mutual Reinforcement	0.924
1/4	Alignment	0.828
	Mutual Reinforcement	0.852

表 4.2-1 候選摘要句子的評估結果

其中 Size of the Candidate 代表候選摘要句子集合的大小為原先句子集合的二分之一或四分之一；而在第三章中我們利用了兩種不同的演算法來移除無資訊量的句子，其分別為 Alignment 演算法與 Mutual Reinforcement 演算法。

由上表得知此效能評估的結果顯示，當候選摘要句子集合大小為原先句子集合的四分之一時，其ROUGE-1的Recall評估值達到八成；而當候選摘要句子的大小為原先的二分之一時，其ROUGE-1的Recall評估值更高達九成。這代表了上述的兩個演算法皆有達到移除無資訊量句子並大量地保留原先資訊的效能，而這其中又以Mutual Reinforcement演算法得到較佳的資訊保留效果。

我們將系統架構中的第四個部分產生的摘要句子集合作為最後的系統摘要，並利用ROUGE自動評估工具作效能的評估，其評估效果如下列各表，其中使用的資料集分別為DUC 2002、DUC 2006、DUC 2007：

	Recall	Precision	F-value
ROUGE-1	0.41652	0.38696	0.40085
ROUGE-2	0.09982	0.09267	0.09603
ROUGE-SU4	0.15401	0.14296	0.14815

表 4.2-2 DUC 2002 的評估結果

Summarization System	ROUGE-1	ROUGR-2
NetSum	0.44963	0.11167
CRF	0.44006	0.10924
SVM	0.43235	0.10867
Manifold-Ranking	0.42325	0.10677
CQPSum	0.42241	0.10177
<b>Our system</b>	<b>0.41652</b>	<b>0.09982</b>
S21	0.41488	0.09075
S19	0.40823	0.08878
CLusterCMRW	0.38546	0.08652
ClusterHITS	0.37872	0.08133

表 4.2-3 DUC 2002 評估結果與其他系統之比較

因為 ROUGE 自動評估工具在西元 2004 年時才正式發表，所以 NIST 於

DUC 2005 之後的資料集才開始規定使用 ROUGE 自動評估工具做為評估的依據。故我們查閱一樣是使用 DUC 2002 資料集與 ROUGE 評估工具的摘要系統之論文，共有四篇[26][27][28][29]，其中年份分別為西元 2009、2009、2006 和 2008 年。而根據表 4.2-3 顯示，跟上面所述的摘要系統比較後，我們的摘要系統獲得還不錯的排名。

	Recall	Precision	F-value
ROUGE-1	0.40462	0.36708	0.38466
ROUGE-2	0.08617	0.07814	0.08190
ROUGE-SU4	0.14371	0.13022	0.13654

表 4.2-4 DUC 2006 的評估結果

Summarization System	ROUGE-2	ROUGR-SU4
System 24	0.09505	0.15464
System 12	0.08987	0.14755
System 23	0.08792	0.14486
System 8	0.08707	0.14134
System 28	0.08700	0.14522
System 15	0.08679	0.14170
<b>Our System</b>	<b>0.08617</b>	<b>0.14371</b>
System 31	0.08576	0.14381
System 33	0.08444	0.14483
System 2	0.08408	0.13912

表 4.2-5 DUC 2006 評估結果與其他系統之比較

NIST 在 DUC 2006 與 2007 評估準則中只評估 ROUGE-2 與 ROUGE-SU4，而不評估 ROUGE-1，這是因為 NIST 為了避免所提出的摘要方法會針對 1-gram 字詞這個方面去做加強而達到 ROUGE-1 評估值較高的效果。

	Recall	Precision	F-value
ROUGE-1	0.42480	0.38270	0.40240
ROUGE-2	0.10264	0.09252	0.09726
ROUGE-SU4	0.15959	0.14372	0.15114

表 4.2-6 DUC 2007 的評估結果

Summarization System	ROUGE-2	ROUGR-SU4
System 15	0.12448	0.17711
System 29	0.12028	0.17074
System 4	0.11887	0.16999
System 24	0.11793	0.17593
System 13	0.11172	0.16446
System 20	0.10879	0.15844
<b>Our System</b>	<b>0.10836</b>	<b>0.16296</b>
System 23	0.10810	0.16280
System 7	0.10795	0.15990
System 3	0.10660	0.15991

表 4.2-7 DUC 2007 評估結果與其他系統之比較

在針對DUC 2006與DUC 2007資料集所舉行的Document Understanding Conference會議中大約都有三十個左右的參加者，其所提出的摘要系統分別為System 2到System 32。根據表4.2-5與表4.2-7顯示，跟這三十個參加的摘要系統比較後，我們的摘要系統排名皆在第七名。這是一個還不錯的成績，但是可能還是有進步的空間。

## 第五章、結論與未來展望

### 5.1 研究總結

本論文所完成的多文件摘要系統是基於Mutual Reinforcement原理所設計出來的，其原理相當地直觀：重要的字詞會出現在多個重要的句子之中；而重要的句子必定會包含多個重要的字詞。除此之外，更利用了Alignment演算法與Mutual Reinforcement演算法保留文章中較富含資訊量的句子，這個部分幫助了下一個階段的句子評分演算法。

而在句子評分演算法中，本摘要系統完整地考慮了三個不同的面向：字詞對句子、標題對句子、句子對句子等三種不同的關係，並分別利用HITS網頁排序演算法、餘弦相似度計算方法、PageRank網頁排序演算法來實現上列三個不同的面向的評分值，此部分為本摘要系統的重點所在。

接著，在上列三個不同的句子分數經過線性組合後，得到最終的句子分數排序順序，摘要系統最後則依此加權排序順序來挑選摘要句子，並再依照規定的摘要字數來組成系統摘要。

根據評估實驗結果，我們所提出的摘要系統跟其他摘要系統比較後，獲得還不錯的排名。這是因為我們的摘要系統不僅僅只是考慮字詞或句子等單方面的因素，而是完整地考慮了字詞對句子、標題對句子、句子對句子等三種不同的面向來做為挑選摘要句子的依據，使得最後的系統摘要能夠具有可讀性、流暢性、簡潔性、概括性與客觀性等各種特性。

### 5.2 未來展望

在未來的研究中，希望能針對本摘要系統不足的部分，進而研究一些技術或方法使得之後所提出的摘要系統能夠獲得更佳的效能。而下列的議題則可能成

為之後我們所更深入探討的研究方向：

1. 在文章中必定會包含許多代名詞(Pronoun)，而此代名詞為代替前面或後面句子中的其中一個名詞。若是我們可以將此代名詞成功地還原成其所代替的名詞，則此對於所建立的投影向量空間模型有極高的幫助。除此之外，在最後產生的系統摘要句子中就不會出現這些代名詞，而造成語意混淆不清甚至是錯誤的情形。故若我們能將每個代名詞都成功地還原成其名詞，則對於系統摘要的效能肯定有一定程度地提升。

2. 在本摘要系統中，較缺乏的為對於文章中字詞或句子的語意分析，故希望可以將此類的語意分析式方法加入我們的摘要系統中。在此類的方法中，我們可以藉由字詞或句子的語意(包含結構、詞性、同義、反義、上下位等)所建立的語彙鏈結來架構其文章中階層式的主題類別，進而分析這些主題類別是否為文章中的主要討論的主題，最後我們則可以根據此類的主要主題來做為挑選其系統摘要的主要依據。





## 參考文獻

- [ 1 ] Hans Peter Luhn, Keyword-in-context index for technical literature. *American Documentation*, 11(4):288–295. ISSN: 0002-8231.
- [ 2 ] Stergos Afantenosa, Vangelis Karkaletsis, Panagiotis Stamatopoulos, Summarization from medical documents: a survey, *Artificial Intelligence in Medicine*, 33(2), 157-177.
- [ 3 ] Sparck-Jones K, Automatic summarizing: factors and directions. In: Mani I, Maybury MT, editors. *Advances in automatic text summarization*. 1999. p. 10—12 [chapter 1].
- [ 4 ] Alice H. Oh, *Generating Multiple Summaries Based on Computational Model of Perspective*, A PhD Thesis of Massachusetts Institute of Technology, September 2008.
- [ 5 ] Jade Goldstein Stewart, *Genre Oriented Summarization*, A PhD Thesis of Carnegie Mellon University, December 2008.
- [ 6 ] Gerard Salton, Andrew Wong, and Chung Shu Yang, A vector space model for Information Retrieval, In *Proceedings of Journal of the American Society for Information Science*, 18(11):613-620, November 1975.
- [ 7 ] Massih R. Amini, Nicolas Usunier, A Contextual Query Expansion Approach by Term Clustering for Robust Text Summarization, In *Proceedings of Document Understanding Conference 2007*, April 2007. Presented at NAACL-HLT 2007.
- [ 8 ] Daniel Marcu, The Automatic Construction of Large-Scale Corpora for Summarization, In *Proceedings of the 22<sup>nd</sup> ACM SIGIR Conference*, 1999.
- [ 9 ] Hongyuan Zha, Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering, In *Proceedings of*

SIGIR 2002, January 2002.

[ 10 ] Ya Zhang, Xiang Ji, ChaoHsien Chu, Hongyuan Zhang, Correlating Summarization of Multisource News with K-Way Graph Bi-clustering, In Proceedings of ACM SIGKDD Explorations Newsletter, December 2004.

[ 11 ] Hany Hassan, Ahmed Hassan, Ossama Emam, Unsupervised Information Extraction Approach Using Graph Mutual Reinforcement, In Proceedings of Empirical Methods for Natural Language Processing (EMNLP), 2006.

[ 12 ] J.M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, In Proceedings of 9th ACM–SIAM Symp. on Discrete Algorithms, 1998.

[ 13 ] Soumen Chakrabarti, Data mining for hypertext: A tutorial survey, In Proceedings of ACM SIGKDD, Jan 2000.

[ 14 ] Jidong Wang, Huajun Zeng, Zheng Chen, Hongjun Lu, Li Tao, Wei-Ying Ma, ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects, In Proceedings of SIGIR 2003, July 2003.

[ 15 ] Furu Wei, Wenjie Li, Qin Lu, Yanxiang He, Query-Sensitive Mutual Reinforcement Chain and Its Application in Query-Oriented Multi-Document Summarization, In Proceedings of SIGIR 2008, July 2008.

[ 16 ] Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, In Proceedings of Seventh International World-Wide Web Conference, April 1998.

[ 17 ] Meishan Hu, Aixin Sun, and Ee-Peng Lim, Comments-Oriented Document Summarization: Understanding Documents with Readers' Feedback, In Proceedings of SIGIR 2008, July 2008.

[ 18 ] Chin-Yew Lin, ROUGE: A Package for Automatic Evaluation of

Summaries, In Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, 2004.

[ 19 ] Sujian Li, You Ouyang, Wei Wang, Bin Sun, Multi-document Summarization Using Support Vector Regression, In Proceedings of Document Understanding Conference 2007, April 2007. Presented at NAACL-HLT 2007.

[ 20 ] Xiaojun Wan and Jianwu Yang, Multi-Document Summarization Using Cluster-Based Link Analysis, In Proceedings of SIGIR 2008, July 2008.

[ 21 ] Xiao-Chen Ma, Gui-Bin Yu, Liang Ma, Multi-document Summarization Using Clustering Algorithm, In Proceedings of IEEE, 2009.

[ 22 ] Dani Yogatama, Kumiko Tanaka-Ishii, Multilingual Spectral Clustering Using Document Similarity Propagation, In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, August 2009.

[ 23 ] M.F. Porter, An algorithm for suffix stripping, In Proceedings of Program, July 1980.

[ 24 ] Jen-Yuan Yeh, A Study on Extraction-based Multidocument Summarization, A PhD Thesis of National Chiao Tung University, March 2008.

[ 25 ] Zan-Wei Liao, Automatic Text Summarization System for Chinese News, A Master Thesis of National Chiao Tung University, June 2009.

[ 26 ] Elena Lloret and Manuel Palomar, Challenging Issues of Automatic Summarization: Relevance Detection and Quality-based Evaluation, In Proceedings of Informatica, April 2009, 29-35.

[ 27 ] Ramiz M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, In Proceedings

of Expert Systems with Applications, May 2009, 7764 – 7772.

[ 28 ] Xiaojun Wan and Jianwu Yang, Improved Affinity Graph Based Multi-Document Summarization, In Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, June 2006, pages 181–184.

[ 29 ] Xiaojun Wan and Jianwu Yang, Multi-Document Summarization Using Cluster-Based Link Analysis, In Proceedings of SIGIR 2008, July 2008.

[ 30 ] The DUC Dataset's URL: <http://duc.nist.gov/>

[ 31 ] The Stanford Parser's URL:

<http://nlp.stanford.edu/software/lex-parser.shtml>

[ 32 ] The Porter Stemmer's URL: <http://tartarus.org/~martin/PorterStemmer/>

[ 33 ] The Wordnet's URL: <http://wordnet.princeton.edu/>

