國 立 交 通 大 學

多媒體工程研究所

碩 士 論 文

線上智慧型文章生成輔助系統

Intelligent Computer-Aided Article Writing

研 究 生：丁柏元

指導教授：李嘉晃 教授

中 華 民 國 九 十 八 年 六 月

線上智慧型文章生成輔助系統

Intelligent Computer-Aided Article Writing

研 究 生：丁柏元　　Student：Bo-Yuan Ding

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 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

線上智慧型文章生成輔助系統

學生 ： 丁柏元　　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

本文以近千篇部落格文章做為系統產生的基底，使用中研院斷詞系統執行中文斷詞工作，並加入 HowNet 輔助以及將 Google 搜尋當成語料庫，實做出一個中文文章自動產生輔助系統。首先將三萬多篇部落格文章進行資訊和情感類的判斷並加以分類，取出其中為情感類的文章，再加以做SPLR分析、整理，作為文章產生的資料庫。系統會依使用者所期望之概念做詞義的延伸，再利用文章骨架作關鍵字擴展，填充其內容而產生文章。產生的文章可利用Google即時擷取可替換句子並做相似度分析，以讓使用者可替換所期望之填充句，已達輔助的功能。

由於在擷取文章資料庫時做過分類，故產生的出文章偏向日記類、部落格式的文章，較貼近生活。而概念詞的輸入及擴展，以及相似替換句的擷取更能輔助提供者產生期望之文章。期望藉由半自動、半輔助之方式，產生供使用者參考的文章範本，以節省使用者在寫作時所花費的時間。

Intelligent Computer-Aided Article Writing

Student ： Bo-Yuan Ding Advisor：Prof. Chia-Hoang Lee

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

Abstract

In this paper, we proposed and implemented an intelligent computer-aided article writing system, which makes use of about one thousand of essays that gathered from web blogs, and aided by using HowNet and Google search. First, we classify affective articles from 30000 articles that gathered from web blogs. Next, we reparse and reform those articles as our corpus by SPLR processing. The System will extend the concept entered by the user, and use the corpus to extend keyword list. Finally, the system generates the article based on the keyword list and content in the corpus. Meanwhile, the system allows the users to replace the sentences and the system will adopt Google as a corpus to provide some candidate sentences for user references.

The system corpus was gathered from web blogs and that could add more variety to the content. The computer-aided writing system could help users organize content from concepts and reference the sentences written by other people. The first prototype system shows that it could assist the users to finish a blog article effectively.

目錄

圖目錄

表目錄

第一章、緒論

1.1 研究動機

自動文本生成在自然語言處理中一直是個有趣但又不容易有所突破的領域。一般人在寫作時，通常會先構思好文章的架構、主旨等文章內容的骨幹，從而下筆寫作。其中，在寫作的過程中，有時會有下一句不知道要寫什麼而進入深思的情況，此時就會在腦中搜尋曾閱讀過的文章作為參考，將可能的句子加以修改套用進來。

在此將上述的方法做為基礎，研發出一套輔助寫作的系統。從一開始的文章主旨，也就是文章主要概念的延伸，到生成文章架構，到填充句子並藉由Google[15]龐大的資料量做為替換句的參考。以期能迅速的產生出符合使用者期待的文章作為參考，加速寫作的進行。

1.2 研究目的與假設

由於 Web 2.0 的發展，造成了部落格的寫作盛行。本系統透過程式在網路上抓取了 3 萬多篇部落格文章，並擷取其中偏日記類的文章作為語料庫，並經由前處理取出文章較為特別的詞，做為此文章之骨架，再利用語料庫中的句子加以填充以產生參考用的文章。概念詞的輸入以及替換詞相似度的擷取及比對，可以輔助使用者去架構出所期望之文章。

1.3 論文架構

第一章：前言，描述本文的研究動機、目的，將本文系統的初衷和基本概念做一個介紹。

第二章：相關研究，說明文章生成輔助的概念和實行方法，以及已有哪些相關研究。

第三章：系統設計，將前置作業和系統的整體架構做一個完整介紹。

第四章：實驗過程與結果討論，將實作出來的系統做一些比較與討論，分為概念詞擴展、替換詞的相似度排序和生成文章評比。

第五章：系統的結論與展望，將系統做總結和探討系統未來方向。

第二章、相關研究

2.1 文本生成

文本生成一直是自然語言處理研究中一項有趣但又極富挑戰性的課題。因應用目的之不同、生成方法之不同，而有各式各樣的生成系統。MIT Media Laboratory 的 Hugo Liu 和 Push Singh(2002)[1]研發的 MAKEBELIEVE 就利用使用者給予的詞當種子，借由 OMCS 知識庫去產生簡短的文章。Rong Jin(2003)[2]利用統計式學習的標題生成系統，也是近年來較顯重要的研究成果。Anja Belz(2007)[3]利用統計機率分析的方式產生氣象預報類的文章。Fu Ren 和 Qingyun Du(2008)[4]則利用模組的方式，產生形容地理空間的文章，並加以利用在地圖系統上。由上述可看出在自動文本生成的發展上，大多數都是借由統計模型或是知識庫的輔助，而生成的目的也都較偏向某些特殊應用。而近年來由於部落格的發展盛行，造成了語料量迅速的增加。在此就是利用了此特性，發展出一套新的生成模式，利用網路上擷取龐大的語料庫，擷取關鍵字作擴展生成之動作，以彌補中文缺乏知識庫的不足。

2.2 文章分類

語料庫文章內容的品質很直接的影響了結構、文章產生的優劣與否。由於部落格文章較貼近生活，內容較自由、廣泛，且資料量龐大，適合作為語料庫當作生成的基底。但也因為內容廣泛，有些過多資訊量的文章，如：新聞報導、專業文章等，有較多專有名詞及不可替代的語意，故此部分的文章必須過濾掉。

關於中文部落格文章的相關研究，台大的楊昌樺等(2006)[5]就擷取國內各大知名的部落格文章，並自訂喜、怒、哀、樂四種情緒，利用機器學習方式去判斷文章的情感方向。而X Ni等(2007)[6]更發展出了可判斷中文部落格文章偏情感類或是資訊類的系統，並對兩類各整理出20個具代表性的詞。

在此也利用了X Ni等(2007)[6]研究結果中這各20個代表性的詞，透過統計的運算給予其權重，當成文章分類之依據。由於部落格文章本來就較偏向日記、情感類文章，故用此方法消去資訊量過多的文章，取得的情感類文章效果不俗。

2.3 同義詞擴展

為了讓生成文章能接近使用者的期待，故加入了概念生成的功能。其中會對使用者輸入的概念詞做同義詞擴展，並以此找出語料庫中還有此概念之文章做為骨架。而其中同義詞的生成更是利用了HowNet[15]作為一關鍵詞擴展的依據。車萬翔等(2004)[7]就曾使用HowNet作為字典，去對中文相似句子做檢索。L Dai等(2008)[8]更從HowNet中分析出主要語意和修飾語意的概念，進而去做同義詞的相似程度分析。本系統參考上述之研究，提出了HowNet分層比對的方法，進而去找出不同權重的相似度，以對使用者輸入的概念詞加以擴展。

2.4 Edit Distance

Edit Distance 在自然語言處理中，是常常被使用到的演算法。主要目的是去評量句子和句子之間的相似程度。曾元顯(2004)[9]就曾使用此方法去比對文章中句子和標題的相似度，應用在中文自動摘要上。在此也利用 Edit Distance 演算法的特性，對填充句和其候選句做相似度比對，讓輔助系統中替換句的結果能更貼近使用者期望。

2.5 斷詞與詞性標記

斷詞與詞性標記是自然語言處理中基礎且重要的一部份，機器翻譯、資訊擷

取、摘要製作及自動作文評分系統等研究都需利用斷詞及詞性標記處理後的結果來進行下一步動作，故斷詞的結果的正確率對研究成果有直接影響。

在中文的句子中，通常不存在有空白這個單元，所以不像英文的句子可以分得很清楚哪邊為一個字，哪邊為一個詞，所以在此藉助中央研究院的詞庫小組中文斷詞系統[10]來做斷詞與詞性標記的工作，其正確率可達到 95~96%之間，透過以上的方式，將部落格語料庫的文章分隔出字與詞，還有後續工作所需要的詞性。

2.6 中文作文寫作輔助系統

[余思翰, 中文作文寫作輔助系統, 2007[12]]，是第一代的文本產生系統。此系統的語料庫是採用國中生所寫的 689 篇作文來做為他的語料庫。此系統將每一篇的作文中的詞與平衡語料庫的詞做頻率的統計，將相同字詞，但排名相差 100 名以上的詞取出，且取前 300 個當做整個作文產生系統的關鍵字，再將這些關鍵字和所有的作文文章去比對，將每篇文章的所有關鍵字取出並組合成一串列，在此稱為關鍵字串列(Keyword List)，所以文章有 689 篇，則關鍵字串列就有 689 串。

如下圖 2-1 所示，使用者可在畫面上選取一串關鍵字串列，系統將關鍵字取出後就會自動把適合的填充詞給一一填入關鍵字之間，則產生了一篇全新的作文文章。

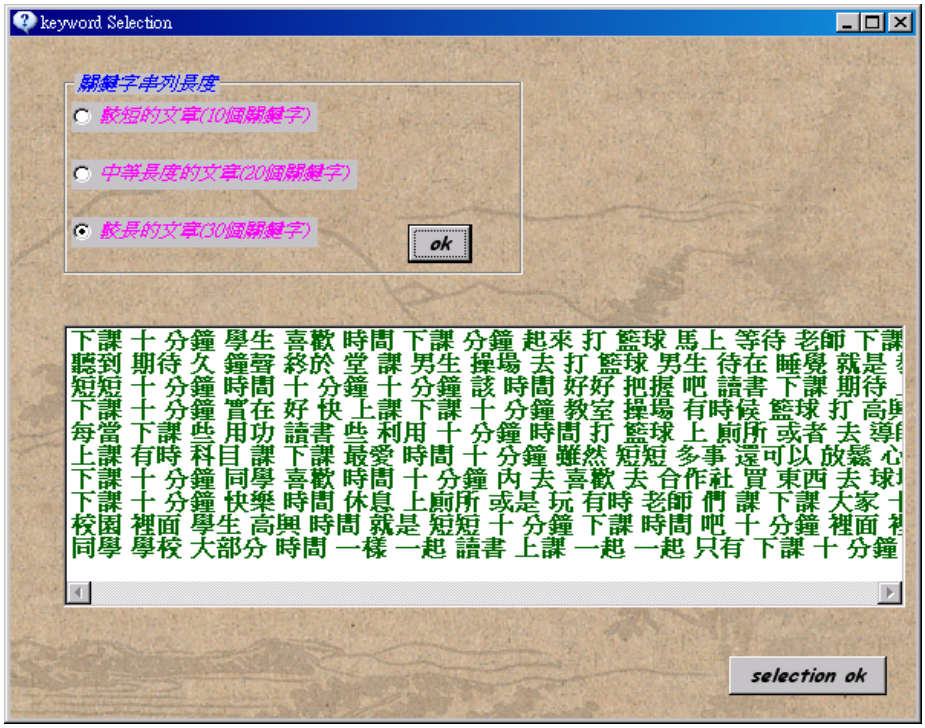


圖 2-1：中文作文寫作輔助系統，關鍵詞選取畫面

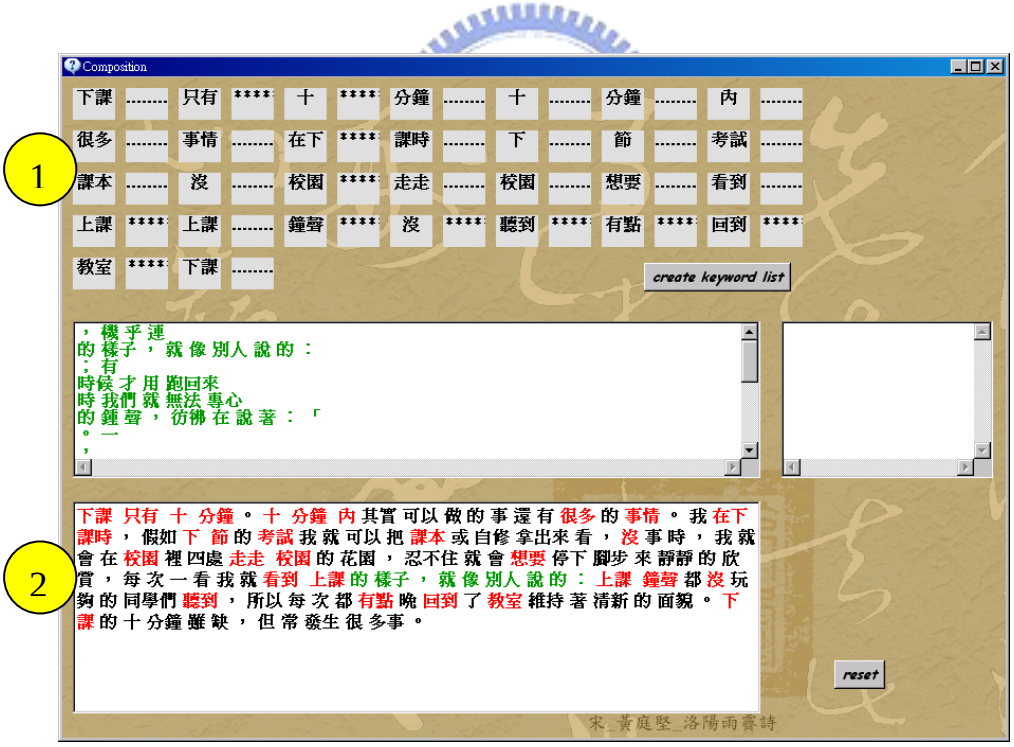


圖 2-2：中文作文寫作輔助系統，文章產生畫面

該中文作文寫作輔助系統尚有幾項缺點：

1. 關鍵字串列尚有贅詞：圖 2-1 所示，關鍵字串列，感覺並不真正像是文章中的關鍵字，串列中還是有贅詞存在，並無準確取出文章中的關鍵詞。

2. 關鍵字串列跟最後產生的文章關聯性低：圖 2-1 所示，關鍵字串列，使用者無法在觀看關鍵字時，就能預想之後文本產生的內容情況。
3. 關鍵字串列內的關鍵字太多：圖 2-2 中①所示，產生出來的關鍵字太多，使用者不易看清操作使用，30 個的關鍵字過多，使用者在操作更換填充詞，需要上下觀看，不夠 User Friendly。
4. 關鍵字串列內的關鍵字與文章長度無關：圖 2-2 中②所示，作文寫作要點，要掌握起、承、轉、合的寫法，雖然關鍵字很多，但產生出來的文章長度不夠，無法掌握寫作文的要點。

2.7 中文情書自動產生系統

[陳智維,中文情書自動產生系統,2008[13]]，是第二代的文本產生系統。此系統的語料庫是由人工蒐集 BBS 及網路上之 446 篇情書類文章。並將文章中的動詞、名詞做 SPLR 的計算，取出的較特殊文章作為代表性的關鍵字，並取 SPLR 值前 5 高的詞作為關鍵字串，做由此五個關鍵字前後各取兩個做關鍵字擴展，而產生文章。

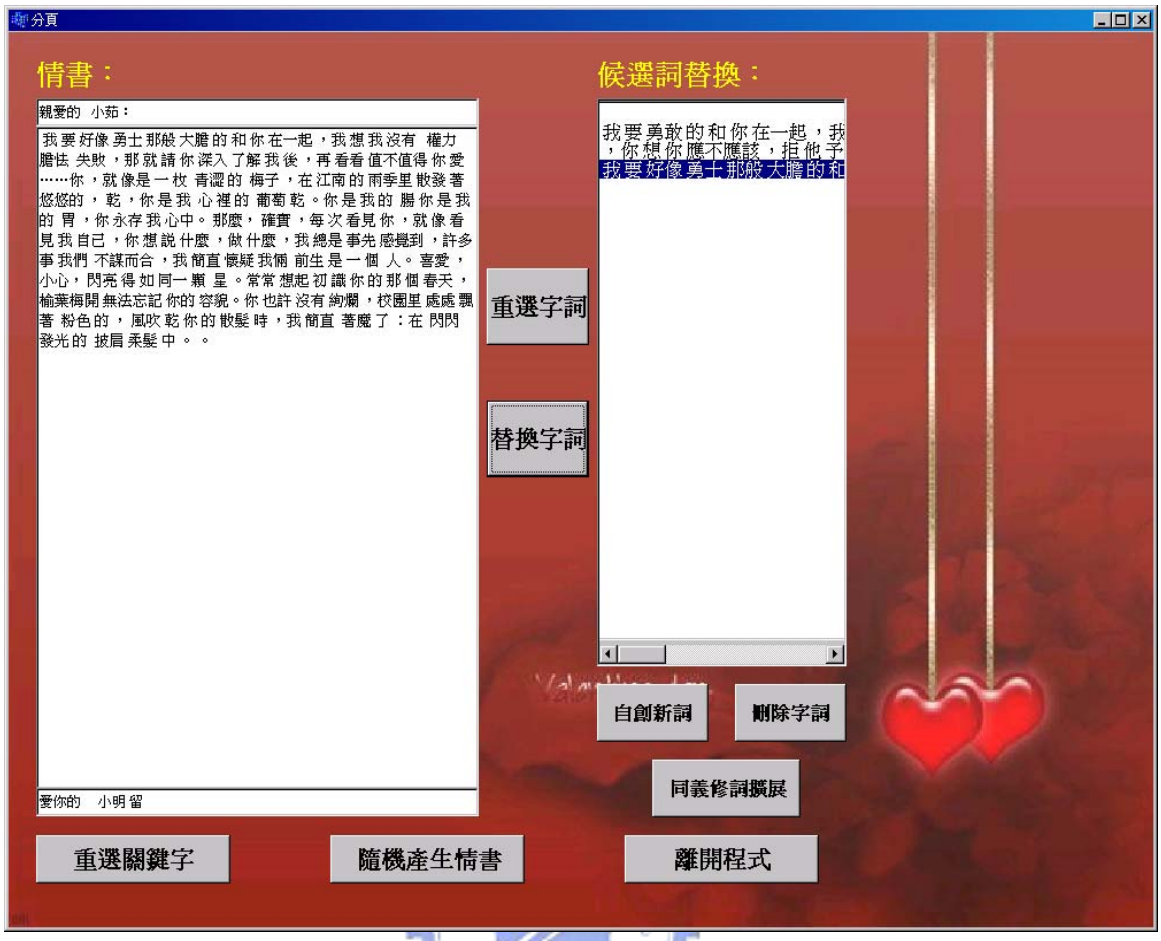


圖 2-3：中文情書自動產生系統，產生畫面

該中文作文寫作輔助系統尚有幾項缺點：

1. 語料庫品質不佳且過少：語料庫單純由人工從 BBS 及網路擷取 446 篇，並無對內容長度、雜訊做處理，導致生成的結果可讀性及品質尚不足，且因為資料量不多，故產生的文章常常出現相同片段的文章。
2. 關鍵字串的選取過於制式：關鍵字的選取單純取前五高做為文章架構，有時會產生文章頭尾互不對應的情形。
3. 填充詞的替換選擇少：由於填充詞選擇單純由原本的語料庫中搜尋，故在語料庫不大的情況下，選擇很少。
4. 僅提供隨機產生：由於文章的生成僅提供隨機的產生，故較不易產生出與使用者期望之內容，缺少輔助性。

後續會將上述的缺失改善，並強化輔助的特性，讓文章能更貼近使用者的期待，以達輔助生成之目的。

第三章、系統設計

3.1 概念

之前的系統只著墨在生成的部分，缺少輔助性，故使得生成之文章無法貼近使用者所期待。在此將系統使用流程分成以下部分：

1. 藉由使用者輸入的概念詞做擴展，找出架構相關度高的文章。
2. 由文章所取的關鍵詞去做擴展，作為欲生成文章的骨架。
3. 對產生好的架構，由語料庫中去加以填充以形成文章雛型。
4. 對於填充詞的部分，系統藉由和 Google 連結找出可替代句，並做相似度排序，提供使用者替換的輔助功能。

藉由以上敘述之步驟，已達到文本生成輔助的效果，以貼近使用者期待之文章內容。

3.2 系統架構

本系統共分為 4 個系統架構，第一部份為情感判斷和前處理，此處將 31869 篇部落格文章做過濾，擷取情感類文章，斷詞後計算 SPLR 數值，取出每篇文章較特殊的詞來代表此篇文章，並重編成方便系統運算之格式。第二部份為生成部份之架構，利用先前語料庫計算出的 SPLR，找出文章的主要關鍵字，並加以擴展，進而置入填充句來產生文章。第三和第四部份較偏向使用者輔助的部分。第三部分使用者輸入概念詞的擴展，其中包含 HowNet 前處理和分層比對法的部分，可以讓文章的內容架構可以更貼近使用者的期望。第四部分則是提供了替換填充句的部分，藉由和 Google 聯結，取得填充句資料並做相似度的分析，增加文章變化的彈性。圖 3-1 為系統的流程架構圖，其中的每一部份流程與架構將在後續章節中詳細介紹。

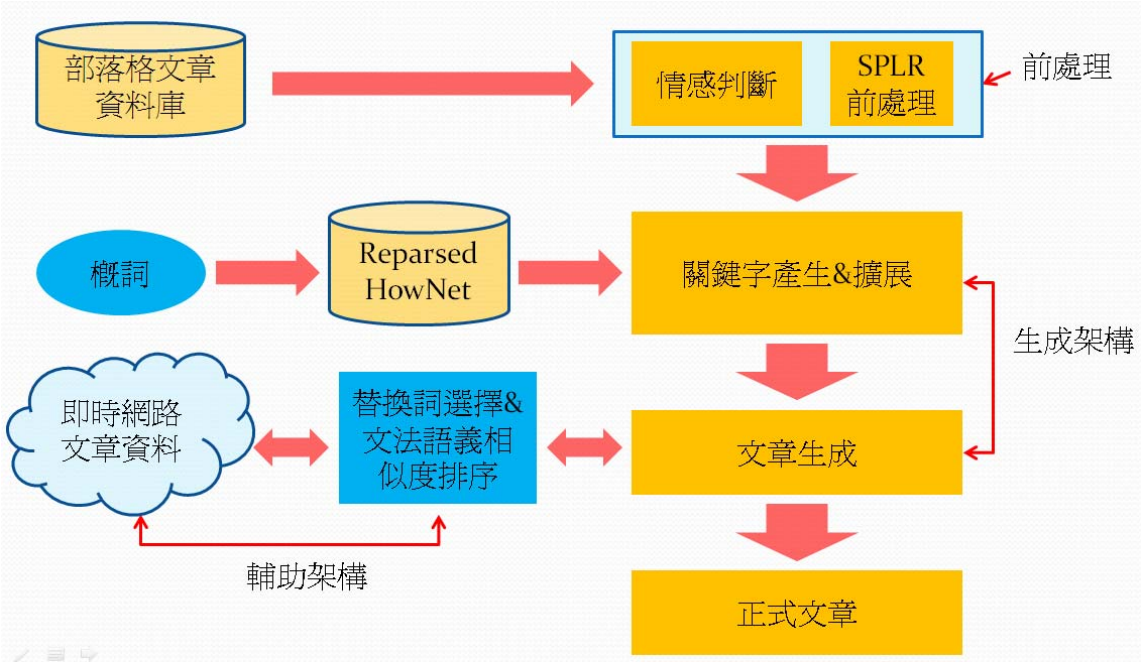


圖 3-1：系統流程架構

3.3 前置作業

本系統的語料庫主要藉由程式從台灣 Yahoo!奇摩搜尋引擎上擷取部落格文章，藉由搜尋引擎所提供的部落格搜尋，對台灣擁有最多使用者的無名小站網誌[16]以及 Yahoo!奇摩[17]本身提供的部落格服務做文章擷取。由於其中可能含有資訊量過多、文章長度過長或過短的問題，且為了加快系統處理時的效能，故前處理主要分成 3 部分。第一部分為收集語料庫的部份，主要在從網路擷取部落格文章。第二部分為情感類文章的擷取，藉此刪除不適合當作生成架構之文章。再經由 CKIP 做斷詞，得到具詞性標記的語料庫。第二部分則把情感類的語料庫做 SPLR 的運算，並在此將文章重新整理格式，以加快系統生成時的效能。

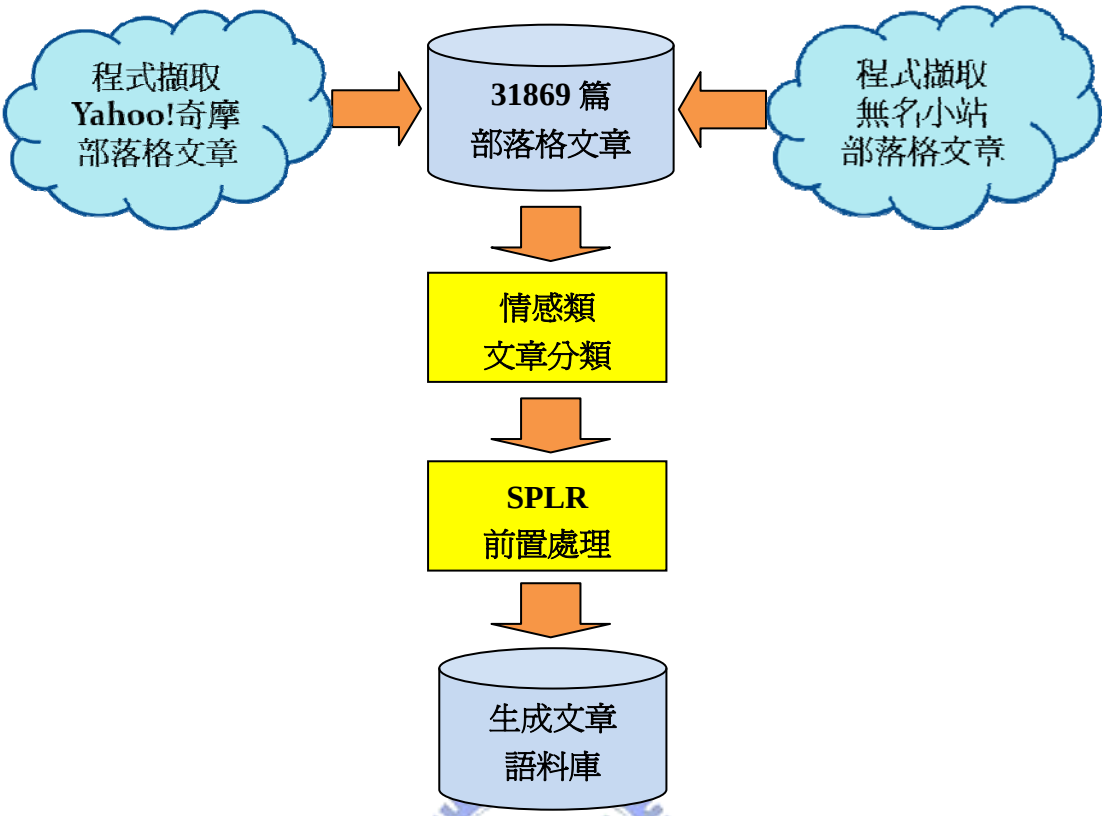


圖 3-2：前置處理流程架構

3.3.1 部落格文章的收集

本系統語料庫來源是從台灣最大入口網站 Yahoo!奇摩提供的部落格搜尋擷取而來，由於搜尋服務的範圍僅限無名小站網誌以及 Yahoo!奇摩本身提供的部落格服務，故可確保多數的使用者皆為台灣繁體中文的使用者。又因為無名小站網誌為台灣地區擁有最多使用者的網誌服務，故網誌撰寫者的年齡層、職業、性別等也較廣泛，減少網誌內容偏向某特一族群的變因。由於蒐集的目標是偏日記類或情感類的文章，而從 X Ni 等(2007)[6]研究結果中發現情感類文章使用頻率最高的詞為“今天”，故在此用此詞當作種子，從 2008 年 8 月開始隨機時間對 Yahoo!奇摩搜尋引擎做部落格文章搜尋，並藉由分析搜尋結果傳回的 HTML 碼，取得部落格文章超連結，進而取得文章內容。從 2008 年 8 月 12 日到 2008 年 9 月 28 日共收集了 31869 篇的部落格文章。

3.3.2 情感類文章的判斷

由於部落格文章之內容有時並非是作者自己撰寫，而是轉錄自新聞網站，或是一般的官方網站，轉錄方式可能是複製文字或是僅提供超連結網址。且撰寫部落格年齡層廣泛，文章內容的主題、語意用詞、文章長短也沒有限制。故所擷取的部落格文章中，會含有不適合作為生成基底的雜訊。

基於上述理由，前置處理一開始先刪去不達平均長度的文章，去除可能僅為轉貼網址、語意或訊息不多的文章。接著再刪去含有不雅字語等特定詞的文章。接著藉由 X. Ni 等人(2007)[6]研究結果中整理出的情感、資訊類各 20 個詞做為基底，對 31869 篇文章做統計，先計算出各個詞在語料庫出現的次數，再去求各個詞在其類別中的比例，當作此詞的權重，以下例為例：

*“**真的**”出現的次數為 22100*

而全部 20 個情感詞的出現次數為 168969

*故“**真的**”的分數為 22100/168969=0.13*

經上述統計後，情感、資訊類的每個詞都將會得到一個個別的權重分數，共計兩類各 20 個詞，分別有各 20 個分數，如表 3-1 所示。

表 3-1：資訊、情感類詞的權重分數

Informative Category		Affective Category	
攝影: 0.054	工具: 0.052	真的: 0.131	覺得: 0.059
圖文: 0.004	地圖: 0.029	朋友: 0.056	感覺: 0.05
報告: 0.133	動漫: 0.007	開心: 0.053	事情: 0.033
數位: 0.018	專家: 0.028	喜歡: 0.058	幸福: 0.012
建設: 0.018	文學: 0.04	一起: 0.049	晚上: 0.032
美術: 0.032	資訊: 0.119	今天: 0.258	快樂: 0.023
奧運: 0.209	房產: 0.001	媽媽: 0.02	心情: 0.025
經濟: 0.101	資源: 0.026	記得: 0.014	希望: 0.039
影視: 0.002	藝術: 0.087	明天: 0.031	日子: 0.013
軍事: 0.006	工程: 0.033	哭: 0.038	每天: 0.021

藉由這些所得的權重，在拿每一篇文章來做對應，對應到情感類的詞則加上所對應的權重分數；相反的，若是對應到資訊類的詞則減去所對應的權重分數。則每篇文章皆會有一分數代表其內容偏向資訊類或情感類的程度，如下列式子：

第 i 篇文章會有一 $Score_i$：

$$Score_i = affective\_score_i - imformative\_score_i$$

其中，$affective\_score_i$ 代表此篇文章對應到情感類文章所得分數；$imformative\_score_i$ 代表此篇文章對應到資訊類文章所得分數。經過此權重計算後，取分數大於平均值的文章做為情感類文章，也就是適合應用在本系統的語料庫。

3.3.3 SPLR 的計算

SPLR 主要在自然語言處理中的目的是在找文章中的未知詞(Unknown

word)，即較特別、較少見的用詞。由於本系統採用的文章語料庫龐大，為了能明確區別每篇文章主要的重點、差別，故在此用 SPLR 找出文章中較特別的詞當作關鍵詞，以讓使用者能明確地區分每篇文章的主要架構不同處，進而由關鍵詞去擴展生成。主要流程如下圖 3-3 所示，底下有更詳細之介紹。

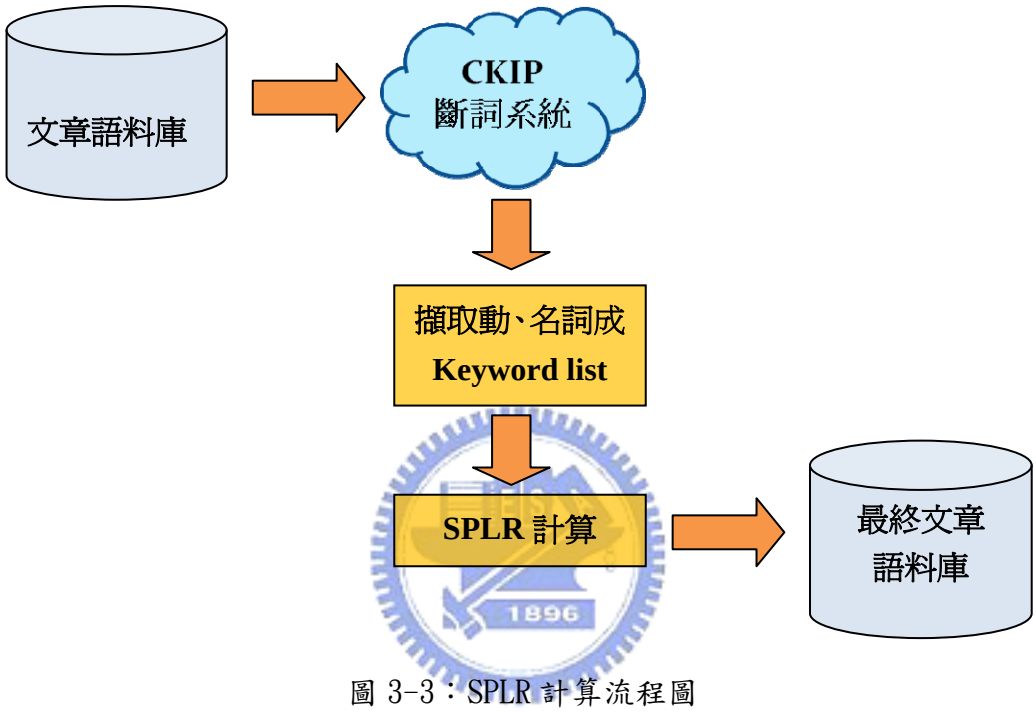


圖 3-3：SPLR 計算流程圖

首先將文章語料庫藉由中研院斷詞小組(CKIP)[10]做斷詞的處理，得到含有詞性符號的文章格式。由於名詞(Na)和動詞(VH)最可代表一句話的重點所在，故將這兩種詞性取出，當作文章最基本的關鍵字串列，即僅包含名詞、動詞的串列。但由某些名詞、動詞的字數少於 2 個字，讓使用者感受的意境不高，也較無法表達文章的含意，故在此取得之關鍵字串，會在除去字數少於 2 個字的詞。結果如下列所示：

家人 國家 行程 風光 馬路 人群 … …

以下，先定義一些後續公式會使用到的集合，以利後續公式的推導。

關鍵字：

$$KW_i \quad , \quad \forall i = 1,2,3,4 \ldots..$$

關鍵字串列集合：

$$Keyword\_List = \{ KW_i \ , i = 1,2,3,\cdots,n \},$$

n：表示該篇文章關鍵字總數

再來，將已取出的關鍵詞串列，使用 SPLR 方法去計算評分，給予每個關鍵詞一相對應的權重，以期在生成階段時能以此為依據，找出真正重要且能代表此篇文章的關鍵字，展現出來讓使用者選取。其計算方法如下：

$$SPLR = \frac{tf(KW_i)}{Max\,(tf(KW_i\_L), tf(KW_i\_R))} \quad , KW_i\_len > 1 \quad (1)$$

$$KW_i\_L = \left\{ KW_i\_L \left| \left\lceil \frac{KW_i\_len}{2} \right\rceil \right. \right\} \quad (2)$$

$$KW_i\_R = (KW_i - \{ KW_i\_L \}) \quad (3)$$

KW_i：關鍵詞

KW_i_len：關鍵詞字數

KW_i_L：擷取關鍵詞左半邊

KW_i_R：擷取關鍵詞右半邊

tf：在語料庫中出現次數的頻率

以下例子用來說明 SPLR 的運作方法，以“酸甜苦辣”和“學生會長”做例子來介紹。其中，Keyword 表示 tp，tf 表示出現在語料庫中的頻率次數。

例 1：

tp1 = 酸甜苦辣　　tL = 酸甜苦　　tR = 甜苦辣

tf(tp1) = 文章中共出現 10 次

tf(tL) = 文章中共出現 10 次

tf(tR) = 文章中共出現 10 次

$$\text{則 SPLR} = \frac{10}{10} = 1$$　（1 即為此關鍵字的得分）

例 2：

tp2 = 學生會長　　tL = 學生會　　tR = 生會長

tf(tp2) = 文章中共出現 20 次

tf(tL) = 文章中共出現 40 次

tf(tR) = 文章中共出現 20 次

$$\text{則 SPLR} = \frac{20}{40} = 0.5$$　（0.5 即為此關鍵字的得分）

利用以上方法可找出文章中較特殊、具代表性的詞，以此做為此篇文章的基本架構。並將文章格式化成圖 3-4 之形式，對每篇文章皆分隔出關鍵字串、SPLR 值以及填充句，用以加速之後生成部分的效能。

家人 國家 行程 風光 馬路 人群 怎麼樣 風景 相反 乾淨 人意 馬路 繁華 人家 ……

0.019 0.007 0.141 0.008 0.017 0.001 0.009 0.045 0.029 0.356 0.001 0.017 0.059 0.032 ……

整天待在家反正沒事，今天就和

到屏東縣內的雙流

風景區遊覽一番，

約九十分鐘，抵達目的地之前，腦海中幻想著園區的周遭

，有點髒的廁所、大

旁、不算少的

、不

的

，沒想到一切正好

，除了廁所

的令

外，入口雖然在台九線大

旁，但是這條路已遠離

，頂多每隔幾公里才會發現一戶

……

圖 3-4：最終語料庫中文章之形式

3.4 生成架構

在此將介紹生成部分的架構，主要分成兩部分講解。首先，先在第一部分介紹本系統的文章生成模型，及本系統生成架構的概念。第二部分主要藉由前置處理中 SPLR 計算所得的值，找出文章中的核心關鍵字串，加以擴展形成文章之主架構。第三部分進而利用填充句的置入而形成新的文章主體。

3.4.1 文章生成模型

在自然語言處理中，機器翻譯的主要工作是將一段語言的文字轉換成另一段語言的文字。統計式的機器翻譯是指藉由分析兩種語言對應的機率，進而統計出相對應的模組。此種方法在機械翻譯方面非常廣泛的應用，且能套用至任何語言之間的轉換。統計式機械翻譯的靈感主要來自於資訊檢索。給定一 f 代表法文原文，e 代表欲轉換成的語言，則主要做法即為 f 句子中，對應到 e 最大的機率，即利用貝式定理，如下式(1)。

$$\hat{e} = \underset{e}{\operatorname{argmax}} P(e|f) = \frac{P(e)P(f|e)}{P(f)} \qquad (1)$$

由於上式的分母和 e 是獨立的，而找最大值的 ê 又相當是在最大值的 e，故可在將上式簡化成公式(2)。

$$\hat{e} = \underset{e}{\operatorname{argmax}} \left(P(e)P(f|e)\right) \qquad (2)$$

經由觀察，人們在寫作時會先去思索關於文章概念的詞，從一些概念性的詞去延伸，試著加入相關連的語句，進而形成一篇文章。由此觀察現象和統計式機械翻譯，本系統生成模型建立在公式(3)。T和 K分別代表填充句和關鍵詞，而做法則是藉由所給予的關鍵字，去找出相對應機率最大的填充句，當作生成之結果。相對的，公式(3)也可減化成公式(4)。

$$\hat{T} = \underset{T}{\operatorname{argmax}} P(T|K) = \frac{P(T)P(K|T)}{P(K)} \tag{3}$$

$$\hat{T} = \underset{T}{\operatorname{argmax}} \left(P(T)P(K|T)\right) \tag{4}$$

本系統的生成部分即是利用上述公式做為基本架構，再由此架構分成下面將介紹的兩部分：關鍵字串列生成和文章產生。

3.4.2 關鍵字串列生成

在此介紹關鍵字串列擴展生成之部分，其流程架構如圖 3-5，更進一步說明如下。

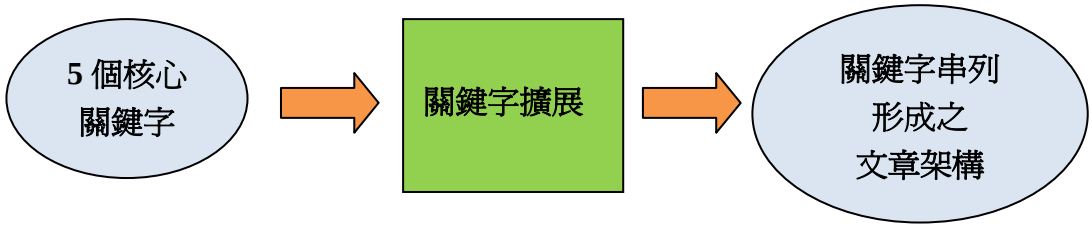


圖 3-5：關鍵字擴展流程架構

藉由之前 SPLR 前置處理，每個關鍵字皆會有一個 SPLR 值，而 5 個核心關鍵字即藉由此數值的高低取得。但為了顧及文章內容的完整性，保持文章頭尾有互相對應，第一個核心關鍵字和最後一個核心關鍵字乃是取位置在文章頭、尾位置，但為後續對應關鍵字擴展，故取第 3 個和倒數第 3 個關鍵詞做為第一個和最

後一個核心關鍵詞，如下式(1)所示：

$$KW\_SPLR = for\ i=1\ to\ n=3 \left\{ KW\_S \left\| \begin{array}{l} Max\left\{ Keyword\_List \right\} \& \\ POS(KW\_S) > 3 \& \\ POS(KW\_S) \\ < \left| Keyword\_List \right| - 3 \end{array} \right. \right\} Keyword\_List - \left\{ KW\_S \right\} \quad (1)$$

經由以上公式可得由五個詞組成之核心關鍵字串，一方面可藉由核心關鍵字串讓使用者瞭解文章的主要概念，一方面文章的架構也由此核心關鍵字串擴展生成。系統產生的核心關鍵字串如下圖所示：

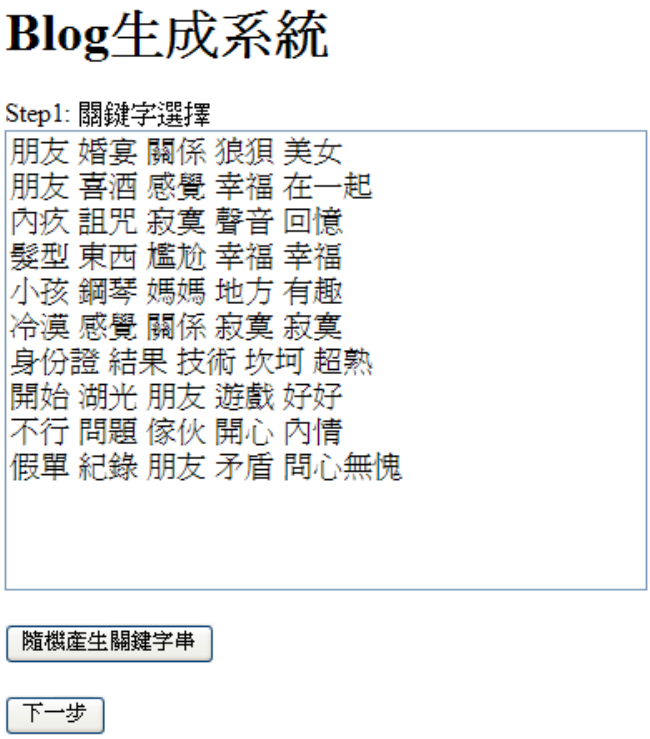


圖 3-6：提供使用者選擇之核心關鍵字串

生成的方法，由圖 3-7 使用者選取的核心關鍵字串，系統在對所選取文章的核心關鍵字串加以前後擴展，即用下面的公式(2)，將 KW_SPLR 關鍵詞與 Keyword List 中其它關鍵詞，一起做生成的動作。

$$Core\_Keyword = \left\{ KW_{i+j} \middle| \begin{array}{l} i = KW\_SPLR(u)\_Loc \\ |j| \le n \\ 1 \le i + j \le Keyword\_List\_Total \\ ,n \in Z^{+} \end{array} \right\} \quad (2)$$

KWi+j：表示關鍵詞在原文中的位置

Keyword_List_Total：關鍵字串列總數

n：取出 KW_SPLR 關鍵詞前與後各 n 個關鍵詞(例如：n=2)

KW_SPLR(u)_Loc：表示 KW_SPLR 內第 u 元素在原文中的位置

下圖 3-7 表示生成相互連結的情況。

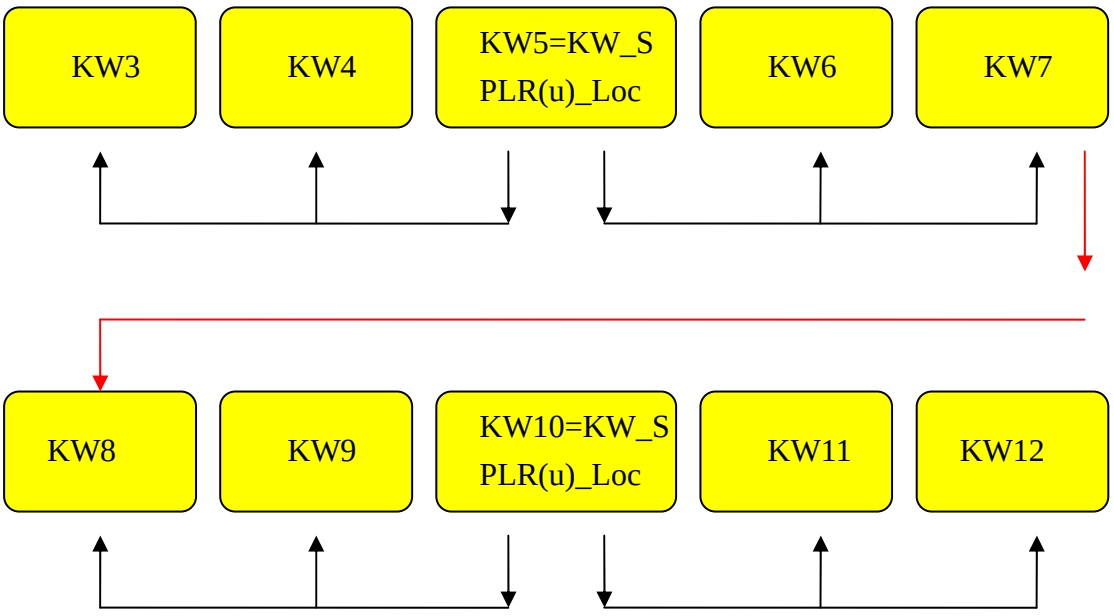


圖 3-7：KW_SPLR 關鍵字生成方式

上圖 3-7 假設 KW_SPLR 其中一個的位置計算出來等於 5，即 KW5。則將 KW5 相鄰位置的前 2 個關鍵詞(KW3，KW4)與後各 2 個關鍵詞(KW6，KW7)取出，形成一個由 5 個關鍵詞所組成的小集合，同樣的 KW10 前後抓出 KW8 KW9 KW11 KW12，4 個關鍵詞所組成的另一個小集合，最後的集合則由 5 組小集合所形成。其中，這樣的產生方式，萬一 KW_SPLR 彼此距離未大於 4 個關鍵詞，則採交集的方式產生，如下圖 3-8 所示。

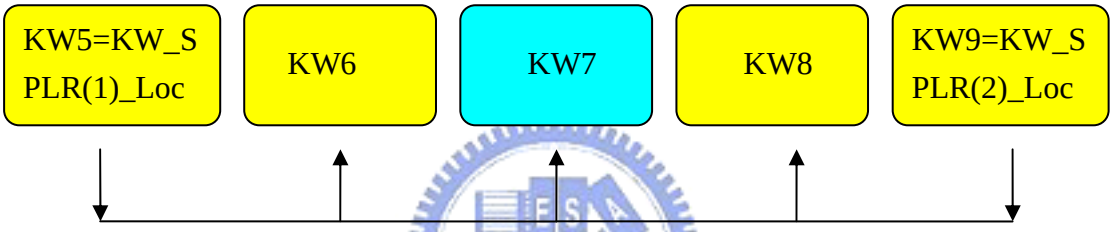


圖 3-8：KW_SPLR 關鍵詞產生重疊

上圖 3-8 藍色區塊，就是二個 KW_SPLR 在生成關鍵詞時，所產生的重疊現象，做法採用交集的方法避免同時抓取到相同的關鍵詞。以上敘述就是此節關於關鍵詞串列生成的方法。

3.4.3 文章之產生

將文章的主幹都架構好後，再來就是要將候選填充詞填入關鍵詞與關鍵詞中，延伸出一篇文章，有如骨幹上的血肉一般，如下圖 3-9 所示，為產生一篇隨機情書的流程架構圖，之後再將詳細介紹其運作方法。

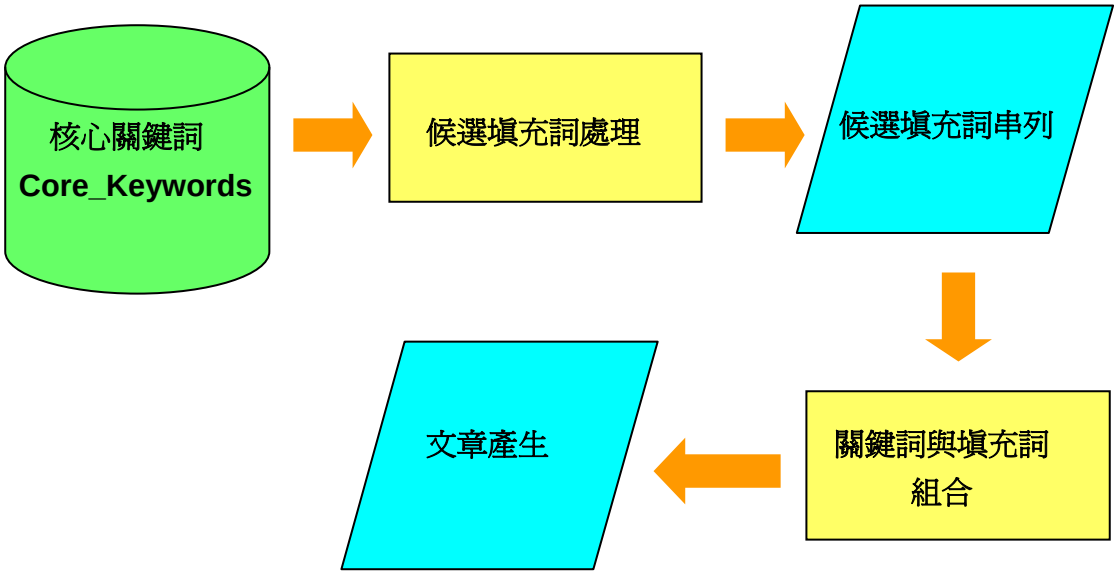


圖 3-9：文章產生流程架構

接下來，要先來介紹如何產生可填入在關鍵詞之間的填充詞，在此先定義接在關鍵詞與關鍵詞間的句子稱為填充詞，可允許接在關鍵詞與關鍵詞間的稱為候選填充詞，如果是多數，則稱候選填充詞串列。以下公式(1)為填充詞集合的收集方式，再將所有找到的填充詞收集起來，成為填充詞串列。

$$Candidate(u) = \left\{ C_u = Unit(i,j) \right.$$
$$\left. \left\| Pre\_KW \cap Word(i,j) \cap Pos\_KW \right| > 0, \forall i,j \right\} \quad (1)$$

Unit(i, j)：表示在第 i 篇 j 位置找到填充詞

Pre_kw：指前一個關鍵詞

Pos_kw：指 Pre_word 關鍵詞後 1 個關鍵詞

Word(i, j)：表示可接於 Pre_word 與 Pos_word 間的填充詞

舉例，Pre_KW 為**禮拜**，Pos_KW 為**開學**，則搜尋出來的 Word(i, j)有以下 3 種變化例子。

<1>……這+**禮拜**+**一就要**+**開學**+……

<2>……這+**禮拜**+**一要**+**開學**+……

<3>……這+**禮拜**+**才剛**+**開學**+……

會結合出以上 3 種變化的句子。

利用上述方式，將填充詞一一找出並且收集起來(2)，則成為候選填充詞串列，之後以此基底，則可利用關鍵詞與候選填充詞串列來組成一篇新的文章。

$$Candidate(u,n)=\left\{\begin{array}{l} Unit(i,j)Unit(i,j+1)Unit(i,j+2)\cdots \\ \quad \left| Unit(i,j) \right. \\ \quad \left| \in Candidate \right. \end{array}\right\} \quad (2)$$

n：為接在哪一個關鍵詞後的位置

舉例：

KW1=火車，KW2=車站，KW3=離別

Candidate(1,1)=「緩緩的駛進」

Candidate(1,2)=「開進了」

Candidate(2,1)=「，我們終於要」

Candidate(2,2)=「，難過的」

以此為組合排列的話，共會有 6 種變化，如下：

火車+緩緩的駛進+車站+ ，我們終於要+ 離別

火車+開進了+車站+ ，難過的+離別

火車+緩緩的駛進+車站+ ，難過的+離別

火車+開進了+車站+ ，我們終於要+離別

．．．．．．

以上的例子，可清楚得知文章是如何生成的，圖 3-10 與圖 3-11 為生成的例子。

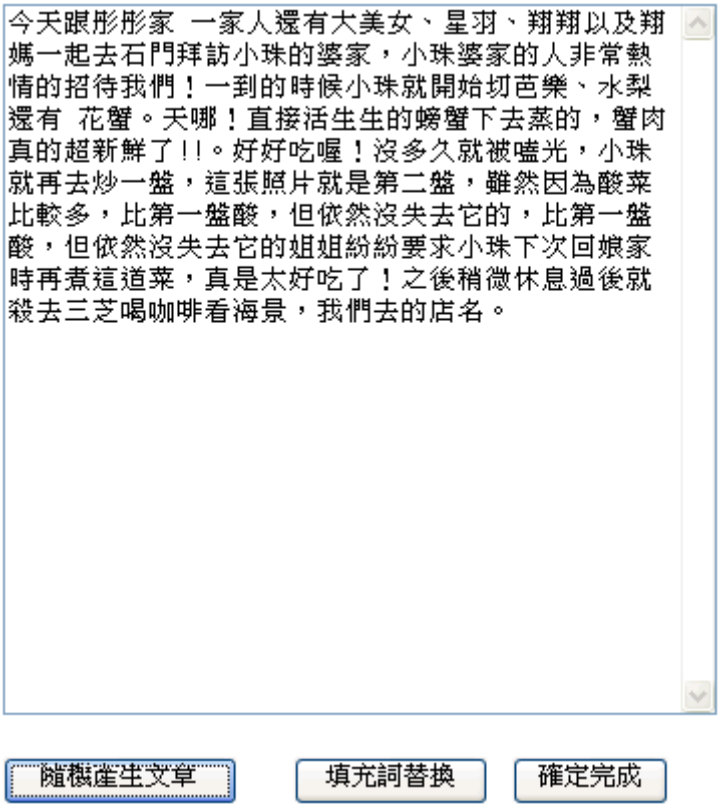


圖 3-10：產生之文章範例 1

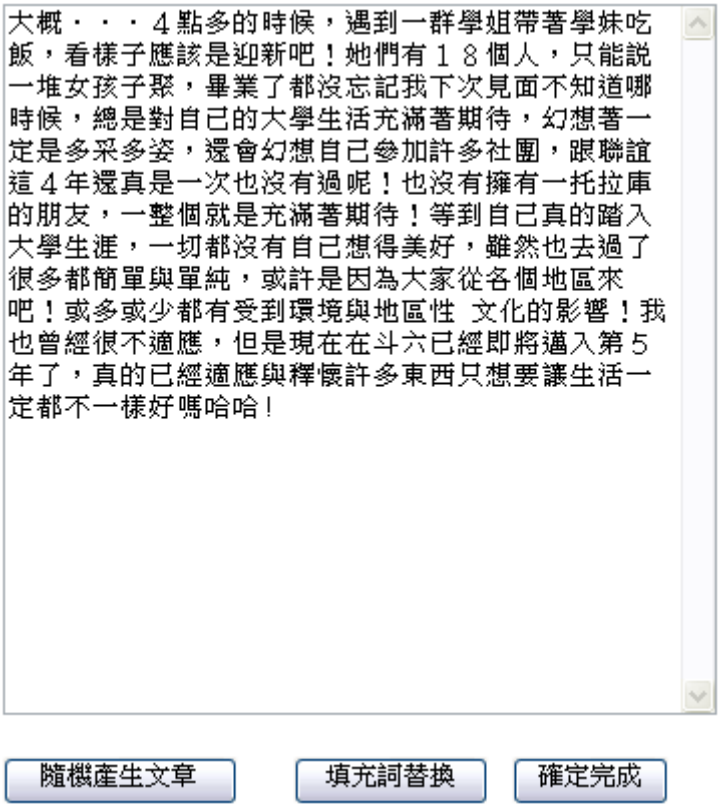


圖 3-11：產生之文章範例 2

3.5 文章概念的比對

為了增強輔助的功能，並且讓產生的文章能貼近使用者所期望之內容，在此加入了概念生成的功能。主要是藉由使用者輸入一特定概念詞，藉由先前處理好的 Reparsed HowNet 做概念的擴展，並利用分層比對法找出含有此概念的文章架構。在此分做兩部分，第一部分為 HowNet 的重新架構，相當於前置處理，以利加速概念生成的速度。第二部分為分層比對法，可利用先前 Reparsed HowNet 之資料，快速比對出含有相同概念的文章架構，詳細的介紹將在後續一一說明。

3.5.1Reparse HowNet

HowNet 是一個中文語料集的字典，內容包含了每個詞編號(NO.)、中文詞語(W_C)、中文詞性(G_C)、中文例句(E_C)、相對應的英文詞語(W_E)、英文詞性(G_E)、英文例句(E_E)以及最重要的概念定義(DEF)。在自然語言處理中，常應用在同義詞的找尋或是詞義的判斷，相當於英語文處理常使用的 WordNet。其 HowNet 原本的格式如下圖 3-12 所示。

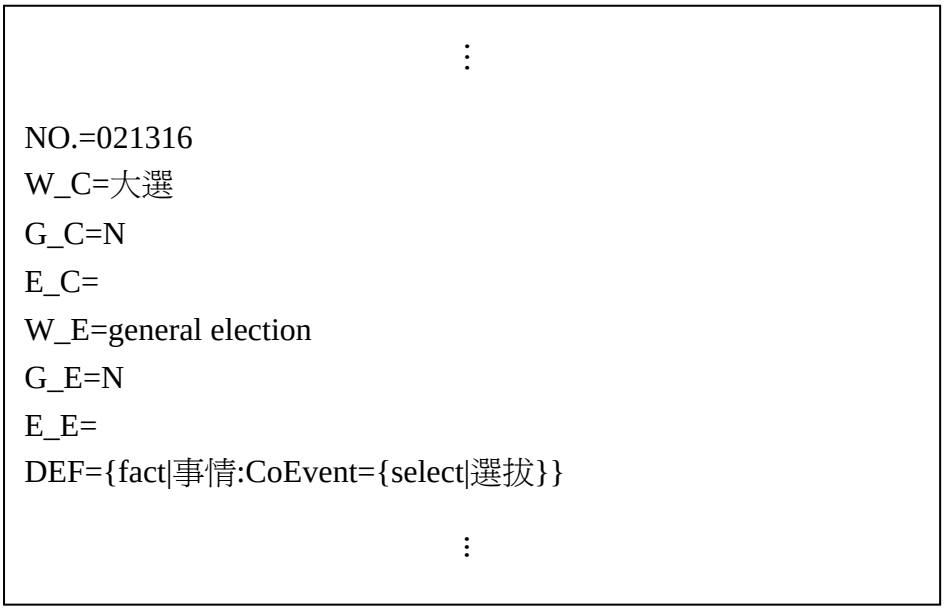


圖 3-12：HowNet 格式

由於在本系統中，只利用到中文詞語(W_C)和概念定義(DEF)，故為了加快系統處理速度，在此先將 HowNet 重新結構成系統所需格式，以方便後續處理。由於 HowNet 中，相同的詞可能因含有多種相對應的英譯，或是含有 2 種以上的概念定義，而造成在 HowNet 中有多個相同詞的不同資料。故在處理成新架構時有兩個原則：

1. 將中文詞語(W_C)和概念定義(DEF)相同，但其它屬性有不同的資料，依

中文詞語(W_C)合併。如圖 3-13 所示，由於有相同中文詞語(W_C)為“電影”和相同概念定義(DEF)為“DEF={shows|表演物}”，故在此除去其它屬性，並將資料合併成同一資料。

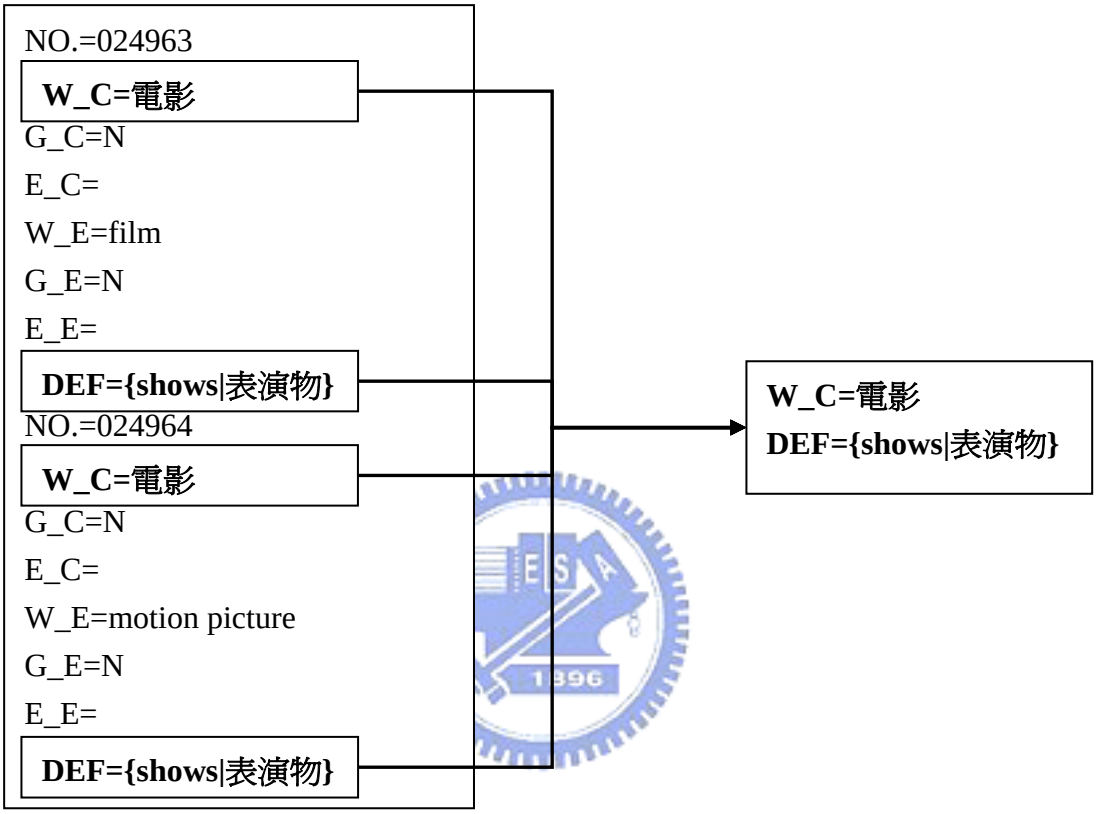


圖 3-13：HowNet 之合併-1

2. 將不同概念定義(DEF)但相同中文詞語(W_C)的資料，依中文詞語(W_C)合併。如圖 3-14 所示，由於中文詞語(W_C)“哼”包含了兩種不同的概念定義(DEF)“DEF={MakeSound|發聲}”和“DEF={sing|唱}”，故依中文詞語(W_C)而合併成一筆資料。

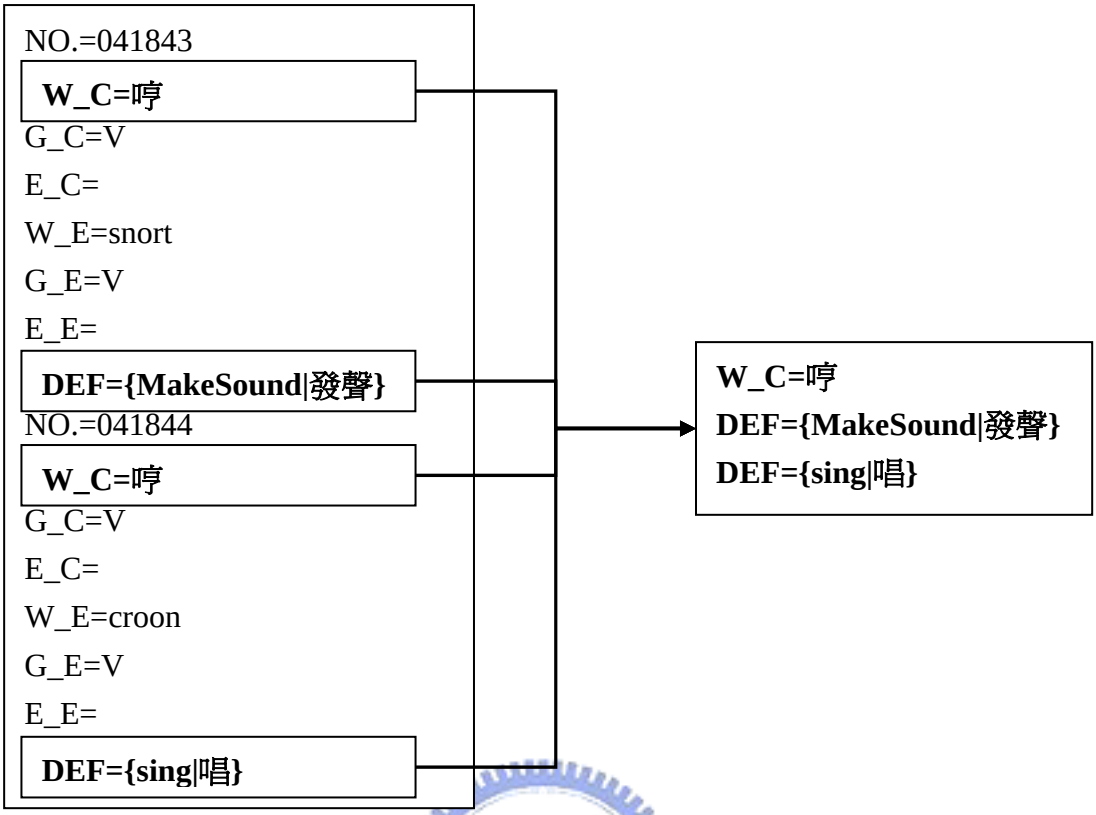


圖 3-14：HowNet 之合併-2

依上述兩種原則，對 HowNet 中所有資料做處理，即可得到只剩下中文詞語(W_C)和概念定義(DEF)兩種屬性的資料格式，並可清楚找出相同詞的多種詞義，如下圖 3-15 所示。

⋮

W_C=交款
DEF={pay|付:possession={money|貨幣}}
W_C=交流
DEF={PropertyValue|特性值:scope={electricity|電}}
DEF={communicate|交流}
DEF={exchange|交換}
W_C=交流電
DEF={electricity|電}

⋮

圖 3-15：重新架構後之 HowNet

3.5.2 分層比對法

由於單靠比對完全一樣的概念定義(DEF)來找尋同樣概念的詞，常常會漏失一些有部分概念相關，但未必完全相關的詞。且對於詞與詞之間的相關性，單靠完全比對概念定義(DEF)，很難在關連性之間在區隔出各自的關係強度。L Dai 等人(2008)[8]年提出主要語意和修飾語意的概念，以圖 3-16 為例，“醫生”的主要語意為“人”(human)，而其它則為形容此概念的修飾語意。參考此概念下，在此提出分層比對法。

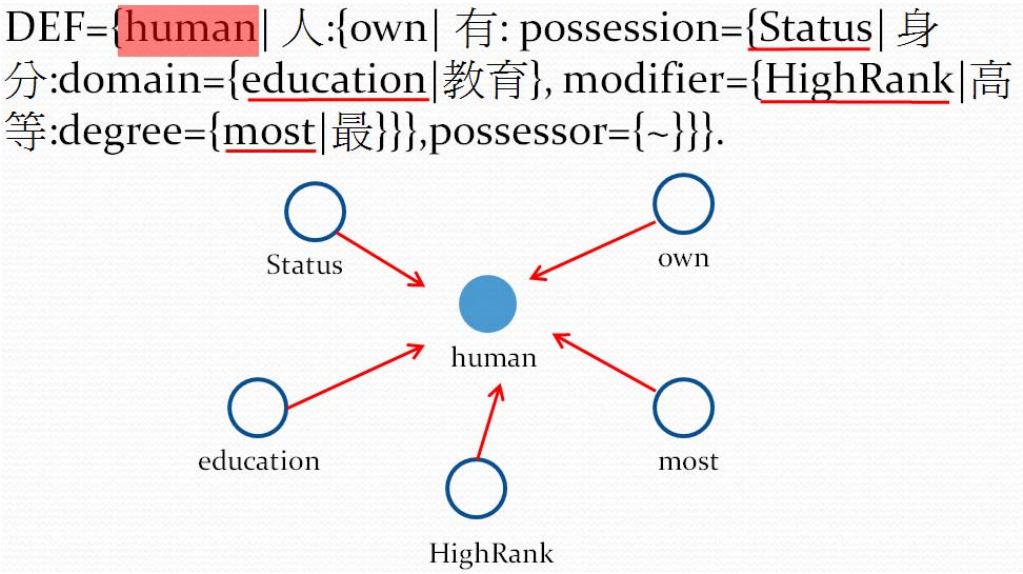


圖 3-16：主要語意和修飾語意之示意圖

為避免完全比對的方式，會造成某些詞和其它詞的關連性過低，故提出了分層比對法，以讓詞語之間的關連性更有彈性。經計算，HowNet 有 77%詞的語意定義(DEF)只有 3 層以下，故一方面考量這些 3 層以下的詞可能關係強度都不高，一方面考量若 3 層以下的詞語也都比對其概念定義(DEF)，可能會造成系統運算速度降低。由於以上考量，只對詞義定意(DEF)有 3 層之詞，往下 3 層做定意的比對。又由於最上層的修飾語意形容的特性是分支最細，也就是對主要語意做較細部的形容，故最上層比對相同則給予 3 分，相對的第 2 層 2 分，第 1 層比對相同 1 分。參考以下例子以“醫生”去找概念相同的詞：

W_C=醫生

DEF={human|人:HostOf={Occupation|職位},domain={medical|醫},{doctor|醫治:agent={~}}}

W_C=醫師

DEF={human|人:HostOf={Occupation|職位},domain={medical|醫},{doctor|醫治:agent={~}}}

W_C=御醫

DEF={human|人:HostOf={Occupation|職位},domain={medical|醫}{royal|皇},{doctor|醫治:agent={~}}}

若只單純採取完全比對法，則“醫生”只能找到“醫師”，而無法和“御醫”有任何關係。但若採用分層比對法，則“醫生”可找出和“御醫”的相關度為 1 分。

採用分層比對的方是再對之前處理過的 Reparsed HowNet 做處理，把每個關連的詞依關連程度的分數串連起來，如下圖 3-17 所示。

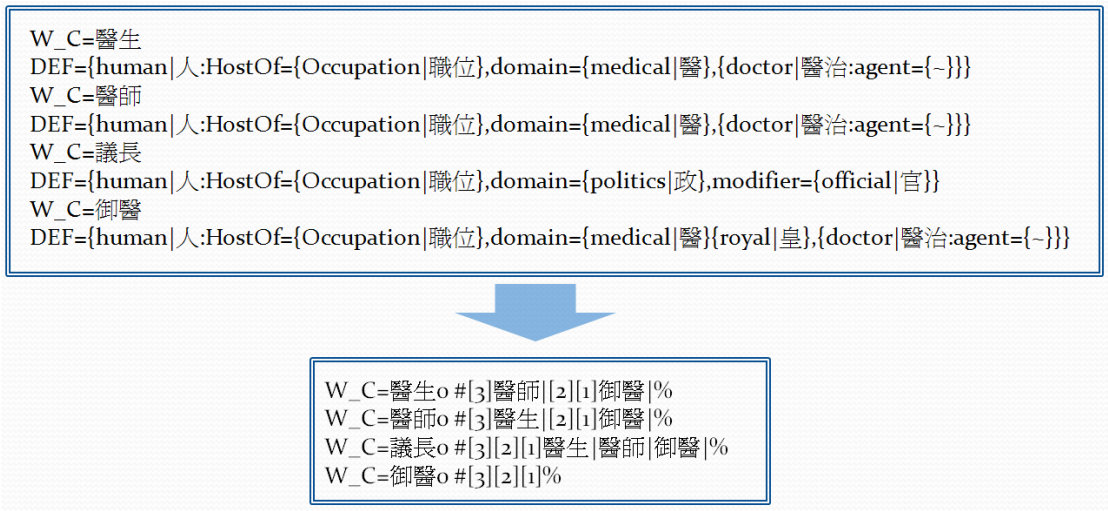


圖 3-17：分層比對法

則最終 Reparsed HowNet 中每一個詞後方皆會有不同相關程度的詞語，如圖 3-18。之後使用者利用概念生成時，則會藉由此 Reparsed HowNet 找出與使用者輸入概念相關的詞，並去文章語料庫中比對，比對到相同的詞，則加上相對應的分數(1-3 分)，比對後分數最高的文章即表示其內容做接近使用者所期望之概念。

```
⋮

W_C=旅遊區 #[1]景區|旅遊地|旅遊點|%
W_C=旅遊熱線 #[1]熱線|%
W_C=旅遊勝地 #[3]古跡|名勝|名勝古跡|勝地|勝跡|史跡|[2][1]名城|名山大川|%

⋮
```

圖 3-18：Reparsed HowNet 供系統使用的最終形式

3.6 填充句替換

由於生成文章結果中，某些填充句可能非使用者所期望，但使用者又想不出其它可替換的句子時，站在輔助使用者的立場，本系統提供填充句替換的功能，以讓使用者能參考更多的填充句例子。在此可分成兩部分，第一部分即為對 Google 做萬用字元的搜尋，並擴展其長度的變化。第二部分則是對 Google 傳回的填充句，做與原填充句的語意、文法相似度比對。其主要流程如下圖 3-19：

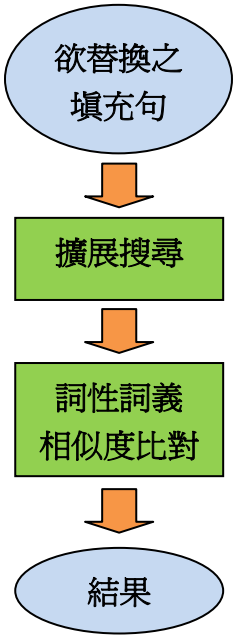


圖 3-19：填充句替換流程圖

3.6.1 擴展搜尋

在此期望填充句的資料來源能夠符合時代語言用法的變化，且豐富多元，故本系統在填充句替換時，以 Google 為資料來源，做萬用字元的搜尋。萬用字元的搜尋通常會用“*”來代表一單字元，“?”來代表一句子，而 Google 的萬用字搜尋只支援單字元“*”，故“*”的數量會影響搜尋所取得填充句資料的長度。

基於以上理由，本系統對 Google 做擴展搜尋時，會在萬用字元前、後加上原本替換句前、後的關鍵詞，並以原本填充句的長度為依據，適當的前、後增加一字元長度，以期能找到適當的填充句，方法主要如下：

若其原填充句長度為 N：

If 2<N<6：則萬用字元長度為 N-1,N+1,N

Else if N<=2：則萬用字元長度為 2,3,4

Else if N>=6：則萬用字元長度為 4,5,6

由以上方法，可將搜尋字串用邏輯符號“or”串連起來，一次對 Google 做萬用字員搜尋，例子如下所示：

若有一句子為“爸爸［不知不覺的］睡著”其中：

關鍵字為：“爸爸”、“睡著”

原填充句為：不知不覺的

因為此例子之填充句的長度為 5，2<5<6，故萬用字元分別前、後擴展為 5-1、5+1，故丟入 Google 之字串如下：

“爸爸*睡著” or “爸爸*****睡著” or “爸爸******睡著”***

其中搜尋之結果有可能包含廣告、新聞或技術類的專業文章，故將搜尋範圍鎖定在部落格文章。而由於 Google 所提供的部落格搜尋結果較少，有些部落格服務內的文章並不會涵蓋進來，故在搜尋字串最後加上“blog”，把搜尋的範圍鎖定在網誌類的文章，較符合本系統之需求。

3.6.2 詞性詞義相似度比對

上一節對 Google 做萬用字搜尋後，即可得到至多 100 筆的替換句，但若直接把這些替換句按照 Google 未經特別的排序回傳給使用者，也會造成使用者一定程度的負擔，故本系統對於傳回來的替換句，會即時自動傳給 CKIP 做斷詞，再和原填充句做詞性和詞義的相似度比對，下面將詳細敘述。

文章的組成是由一句又一句的句子串連起來而形成，每個句子最重要的 2 個屬性即是文法和其所表達之語意；我們將從 CKIP 斷好詞的填充句，取其中的 POS 詞性和原填充句做 Edit Distance 相似度演算法，當作文法的相似度；取原本個別的字和原填充句做 Edit Distance 相似度演算法，當作語義相似度。其 Edit Distance 相似度演算法如下圖 3-20 所示：

```
int EditDistance(char s[1..m], char t[1..n])
    declare int d[0..m, 0..n]

    for i from 0 to m
        d[i, 0] := i
    for j from 0 to n
        d[0, j] := j

    for i from 1 to m
        for j from 1 to n
       {
            if s[i] = t[j] then cost := 0
                           else cost := 1
            d[i, j] := minimum(
                d[i-1, j] + 1,        // deletion
                d[i, j-1] + 1,        // insertion
                d[i-1, j-1] + cost    // substitution
              )
        }

    return d[m, n]
```

圖 3-20：Edit Distance 演算法

由於單純的只參考詞性或詞義相似度，可能會造成期替換句不盡客觀，故在此詞性和詞義相似度的值分別加總，取值最小的為最相近。如下式：

S={Si|i=1,2,3,……,N}　Google 取得之第 i 候選填充句為 Si

G={Gi|i=1,2,3,……,N}　Si 和原句之詞性相似度為 Gi

W={Wi|i=1,2,3,……,N}　Si 和原句之詞義相似度為 Wi

則每句候選填充詞之相似分數：P={Pi|i=1,2,3,……,N}=G+W

第四章、實驗結果與討論

4.1 概論

在此分成三部分來討論，第一部分為概念生成下，HowNet 分層比對法對照其它方法的實驗。第二部分為填充句替換功能中，語意文法相似度比對法的效能比對。前兩部分都是關於輔助功能的部分，第三部分則是關於生成效果的實驗，對系統生成的文章做統計式的評估，並和之前的系統比對。

4.2 概念生成

在此介紹概念生成部分，HowNet 分層比對法的效能評估。實驗的比照對象為使用完全比對 HowNet 概念定義(DEF)方法找尋關連詞的效果。實驗目的為驗證使用 HowNet 分層比對法後，能改進只完全比對定義的效能，一方面能擴展得到相關概念詞的量，一方面能提升所得相關概念詞的質。實驗之範例如下例所示：

例 1：

對於概念為“急救中心”

完全比對 HowNet 概念定義之結果如圖 4-1：

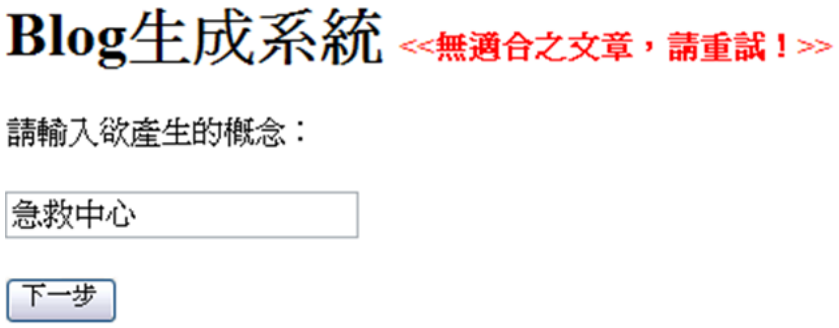


圖 4-1：“急救中心”完全比對法之結果

本系統發展出的 HowNet 分層比對法之傳回結果如圖 4-2：

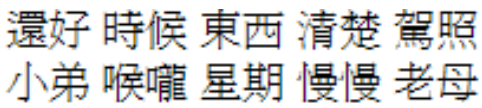


圖 4-2：“急救中心”分層比對法之結果

在此可發現，對於一些文章中較少出現的詞，採用對 HowNet 完全比對概念定義(DEF)的方式，很容易會無法在語料庫中找到相對應的詞，因為其定義完全一樣的同義詞，可能都沒出現在系統語料庫中；相對的採用分層比對法，由於可進一步擴展出不同層級的同義詞，故對於語料庫沒有的詞，可由分層出的同義詞中，找出相對應的文章，且維持著和概念一定關連度的效果。

例 2：

概念輸入為“蛀牙”

完全比對 HowNet 之結果如圖 4-3：

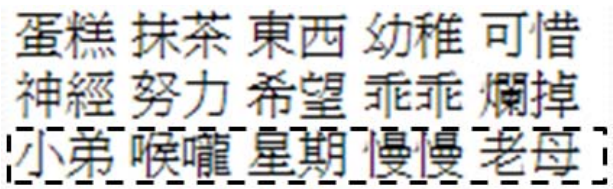


圖 4-3：“蛀牙”完全比對法之結果

本系統發展出的 HowNet 分層比對法之傳回結果如圖 4-4：

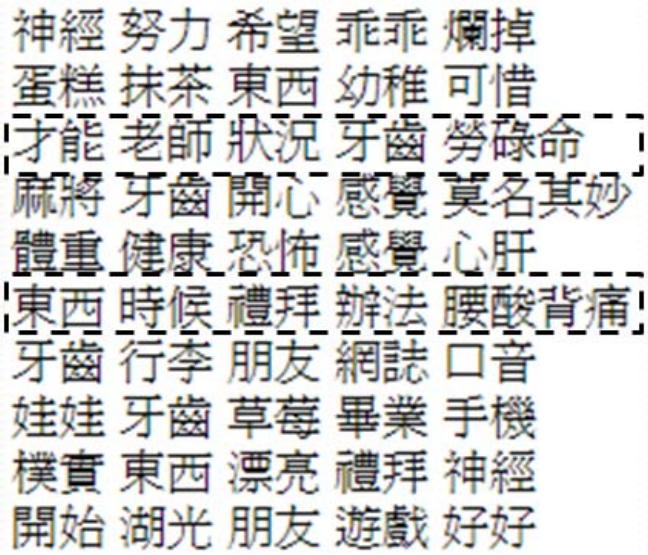


圖 4-4：“蛀牙”分層比對法之結果

在此例中，我們可發現完全比對法所得到的結果明顯較少，藉由分層比對法所找出的關鍵詞，則可得到較多的相關文章。如圖 4-5 所示，分層比對法除了和“蛀牙”相關聯的“蛀齒”、“齲齒”等可找出外，更可找出和“牙齒”這概念相關的詞，如“牙齒”、“臼齒”、“犬齒”等。在此我們只看比較幾篇回傳結果有差異的部份，即虛線框起的部分。完全比對法所得的文章範例，內文明顯和“蛀牙”並沒有明顯的關聯，如圖 4-6 所示；而藉由本系統發展的分層比對法的結果範例中，可發現皆是和蛀牙有關的內文，如圖 4-6 和圖 4-7 所示。

W_C=蛀牙 #[3]蟲牙|齲|齲齒|蛀齒|[2][1]板牙|槽牙|大牙|臼齒|門齒|門牙|犬齒|犬牙|上下牙|牙齒|智齒|獠牙|齙|齙牙|齦|%

圖 4-5：“蛀牙”分層比對法找出的相關詞

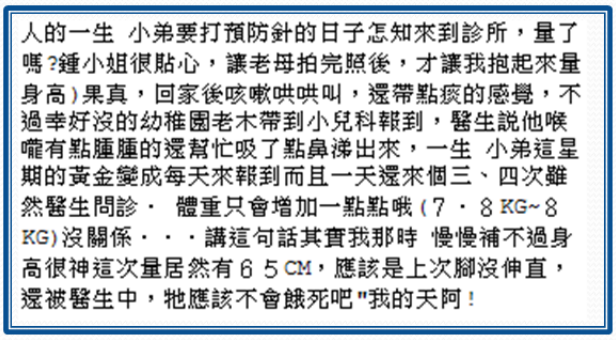
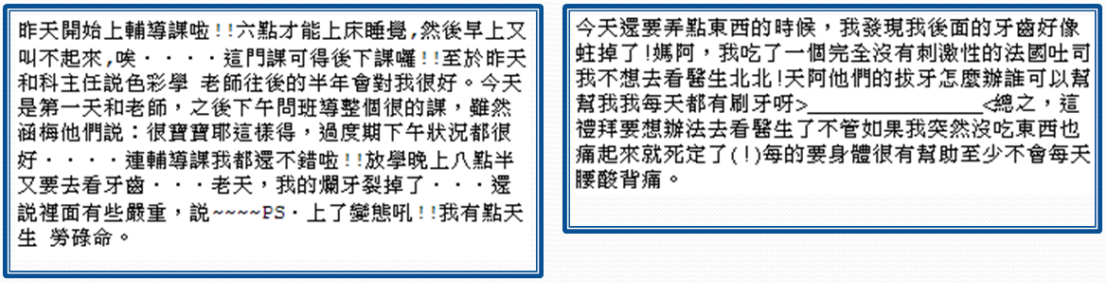

人的一生 小弟要打預防針的日子怎知來到診所，量了嗎?鍾小姐很貼心，讓老母拍完照後，才讓我抱起來量身高)果真，回家後咳嗽哄哄叫，還帶點痰的感覺，不過幸好沒的幼稚園老木帶到小兒科報到，醫生說他喉嚨有點腫腫的還幫忙吸了點鼻涕出來，一生 小弟這星期的黃金變成每天來報到而且一天還來個三、四次雖然醫生問診． 體重只會增加一點點哦(7．8KG~8KG)沒關係．．．講這句話其實我那時 慢慢補不過身高很神這次量居然有65CM，應該是上次腳沒伸直，還被醫生中，牠應該不會餓死吧"我的天阿!

圖 4-6：圖 4-3 中虛線所框起的文章內容


昨天開始上輔導課啦!!六點才能上床睡覺,然後早上又叫不起來,唉．．．．這門課可得後下課囉!!至於昨天和科主任說色彩學 老師往後的半年會對我很好。今天是第一天和老師，之後下午問班導整個很的課，雖然涵梅他們說：很寶寶耶這樣得，過度期下午狀況都很好．．．．連輔導課我都還不錯啦!!放學晚上八點半又要去看牙齒．．．老天，我的爛牙裂掉了．．．還說裡面有些嚴重，說~~~~PS．上了變態吼!!我有點天生 勞碌命。

今天還要弄點東西的時候，我發現我後面的牙齒好像蛀掉了!媽阿，我吃了一個完全沒有刺激性的法國吐司我不想去看醫生北北!天阿他們的拔牙怎麼辦誰可以幫幫我我每天都有刷牙呀>_____<總之，這禮拜要想辦法去看醫生了不管如果我突然沒吃東西也痛起來就死定了(!)每的要身體很有幫助至少不會每天腰酸背痛。

圖 4-7：圖 4-4 中虛線所框起的文章內容

4.3 填充句替換

在此介紹填充句替換系統中，詞性詞義相似度比對的效能實驗。比照對象為無任何相似度比對之結果、單純只比對語意之結果、單純只比對文法之結果。實驗目的為驗證對回傳之填充候選句做綜合詞性、語義比對的效果會優於其它之效果。以下分別對上述三個比照對象做實驗。

例 1：

比照對象：無相似度比對，完全依照 Google 回傳之順序

實驗句子：下午[忽然下起了一陣]大雨

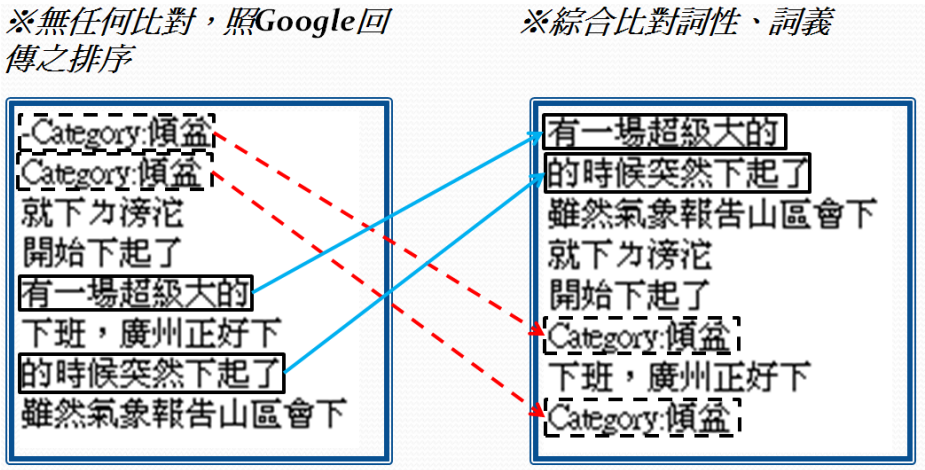


圖 4-8：無相似度演算法之對照

上例是尋找關鍵字“下午”和“大雨”之替換句“忽然下起了一陣”。圖 4-7 中左邊是完全無任何演算法處理過，即單純 Google 回傳之結果，右圖則是本系統詞性詞義相似度演算法之結果。可從圖中看出較不恰當的填充句“Category:傾盆”的排序明顯的被下降，而最恰當的填充句“有一場超級大的”和“的時候忽然下起了”則被排序到最上面，意即這兩個替換句和原替換句相似度最高。

例 2：

比照對象：單純只做 POS 詞性相似度之比對

實驗句子：禮拜[四要去染]頭髮

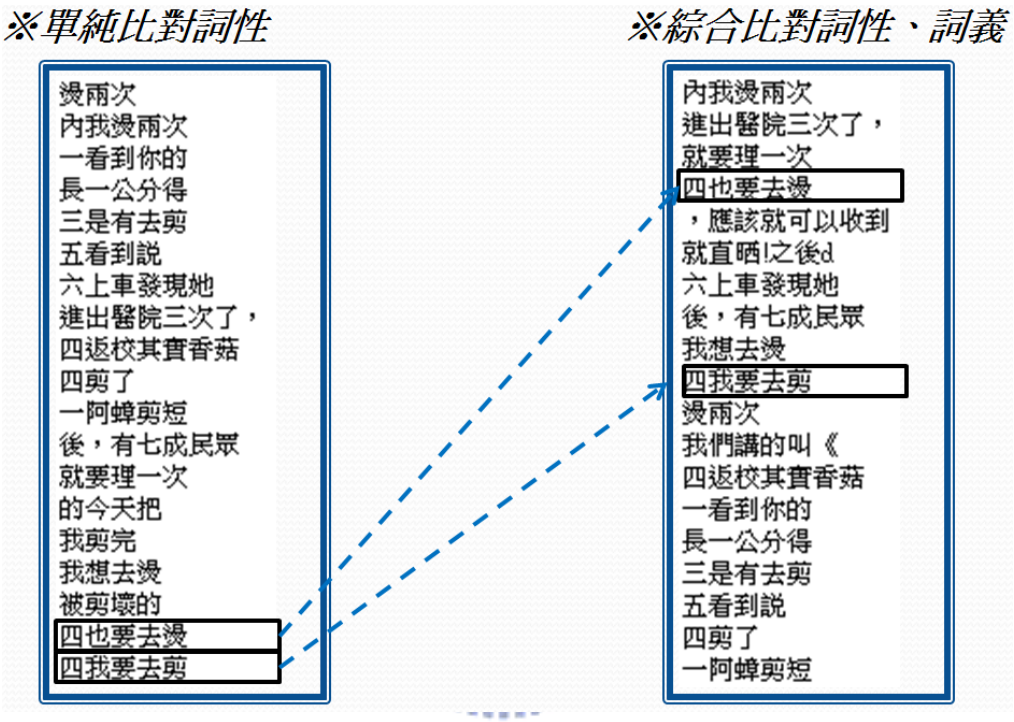


圖 4-9：單純只做詞性相似度之對照

上例是尋找關鍵字“禮拜”和“頭髮”之替換句“四要去染”。圖 4-8 中左邊是單純只比對詞性相似度，右圖則是本系統詞性詞義相似度演算法之結果。很適合當作此例的替換句“四也要去燙”和“四我要去剪”，在單純比對詞性的方法中，被排序到了最後，但在本系統的綜合比對下，成功的把此兩句的排序往上提升。

例 3：

比照對象：單純只做詞義的相似度比對

實驗句子：開心[的坐上]計程車

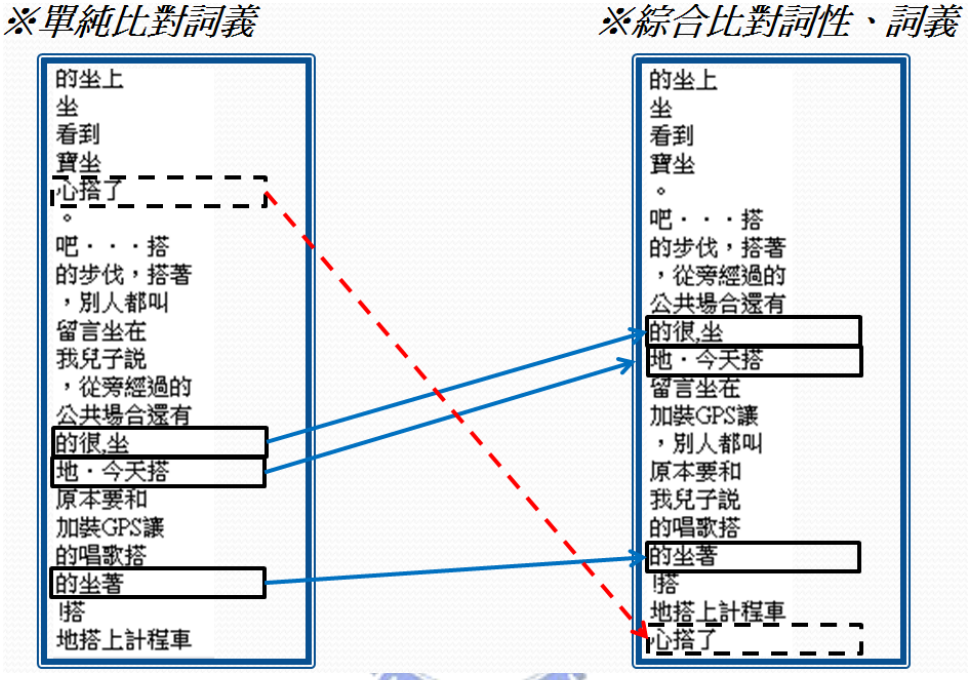


圖 4-10：單純只做詞義相似度之對照

上例中，除了套進去能較符合原意的填充句“的很，坐”、“地，今天搭”和“的坐著”有成功的提高其優先權外，左方很不是合的“心搭了”，也成功的往下移至最下方。

由以上三個例子可觀察出，由於出發點不同，詞性詞義相似度比對在此替換句系統下的效能很明顯的優於 Google 本身的排序。而單純只比對詞性或是單純只比對詞義，雖然也有一定的效果，但常常在某些情形下效果又不盡理想。故在語意文法相似度方法中，結合此兩種方法，在大多數情況下可成功的維持一定的效能。

4.4 使用者評比

由於本系統是基於生成輔助的角度，能產生貼近使用者期望之文章，並提供適當的輔助，故在此使用者評比的部分，乃是希望從使用者的角度，去對本系統做實用性方面的實驗。

在此的實驗對象包括下列 4 項：

1. 可讀性：包含文章順暢度、語意和邏輯等。
2. 切題：是否貼近部落格類或日記類之文章風格。
3. 概念符合：文章與所期待之概念是否相同。
4. 使用性：使用者對整個系統之設計、簡易使用，若使用者有需求是否會考慮使用本系統。

訂下上述的評分項目後，本系統取“颱風”、“生日”、“逛街”、“感冒”、“電影”、“腳踏車”、“火車”、“運動”、“晚餐”和“學校”10 種不同的概念，各取其概念生成的文章 2 篇，總計 20 篇文章作文欲評比文章。然後請 10 人分別對 20 篇文章分別做可讀性、切題、概念符合之評比，評比分數由 1-5 分，越高分代表文章越符合評比條件，反之亦然。再由同樣 10 人，分別使用本系統，對使用性給予一評比分數 1-5 分，分數越高代表系統越方便使用，反之亦然。結果如表 4-2 所示。

而此實驗的比照對象，中文情書生成系統也隨機產生 20 篇情書文章，由相同 10 人做可讀性、切題和使用性的評比，做為參考比較的數據。結果如表 4-1 所示。

	可讀性	切題	使用性
平均值	2.925	3.575	2.857
>=3	63.5%	87.0%	71.4%

表 4-1：中文情書自動產生系統之評比結果

	可讀性	切題	使用性	概念符合
平均值	3.445	4.07	4.571	3.83
>=3	86.5%	92.5%	99.9%	89.0%

表 4-2：本系統之評比結果

由表 4-1 及表 4-2 之結果可明顯看出，本系統在可讀性、切題兩方面的效果平分皆比中文情書產生系統要好，期可能之原因除了文章特性不同外，本系統的分類處理也過濾掉了架構、內容較不恰當的文章，進而在組成新文章時，能有較好的效果。而由於本系統式架構在網頁程式上，並盡量減少使用者過多的操控，優化使用者介面，已達到“防呆”的效果，故在使用性上也較中文情書自動產生系統要好。而在本系統的概念符合項目上，本系統的分層比對法也達到了近 90% 的效能。

第五章、結論與展望

5.1 研究總結

經過多次實驗與修正，本系統能藉由使用者輔助產生出與其期望相近的文章。其中輔助方面的改進，新加入的概念詞擴展和替換句的比對方法，除了在本系統能達到一定的效果外，也能應用在其它自然語言的問題。而網路擷取並分類處理過的大量語料庫，一方面能過濾掉不必要的雜訊，更能使系統產生出貼近生活且多變化的文章。本系統不但模擬了人們寫作時的狀況，更提供輔助的功能，以期更貼近使用者的需求。由實驗結果更可瞭解，本系統無論在生成的品質、輔助效能的幫助，都能達到一定程度的效果，更貼近使用者所需。

5.2 未來工作

由於中文在自然語言處理的領域上，一直缺少實用性大的知識庫，以利架構語意的解析。故在未來希望能建立出一套相對應的知識庫，進而改進生成的方法，利用知識庫的輔助，解析出語意來提高生成文章的效果。另一方面，也希望未來能擴大語料庫的資料量，並加以分類成更細部的分類，讓生成的方向能更細部，更貼近不同使用者的不同期望。

參考文獻

[1] Hugo Liu, Push Singh. (2002). "MAKEBELIEVE: Using Commonsense to Generate Stories" Proceedings of the Eighteenth National Conference on Artificial Intelligence, AAAI 2002, July 28 - August 1, 2002, Edmonton, Alberta, Canada. AAAI Press, 2002, pp. 957-958.

[2] R Jin. "STATISTICAL APPROACHES TOWARD TITLE GENERATION" 2003.

[3] A Belz. "Probabilistic Generation of Weather Forecast Texts" Proceedings of NAACL HLT 2007, pages 164–171, Rochester, NY, April 2007.

[4] Fu Ren, Qingyun Du, "Study on Natural Language Generation for Spatial Information Representation" Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on , vol.4, no., pp.213-216, 18-20 Oct. 2008.

[5] 楊昌樺, 高虹安, 陳信希 (2007)。"以部落格語料進行情緒趨勢分析." 第十九屆自然語言與語音處理研討會論文集，2007 年九月 6-7 日，台灣，台北，205-218。

[6] X. Ni, G. Xue, X. Ling, Y. Yu, and Q. Yang. "Exploring in the weblog space by detecting informative and affective articles." Proceedings of the 16th international conference on World Wide Web, pages 281 – 290, 2007.

[7] 車萬翔, 劉挺, 秦兵, 李生. "基于改进编辑距离的中文相似句子检索." 高技術通訊,2004(7):15-19.

[8] L Dai, B Liu, Y Xia, SK Wu, "Measuring Semantic Similarity between Words Using HowNet." Proceedings of the 2008 International Conference on Computer Science and Information Technology - Volume

00, Pages 601-605, 2008.

[9] 曾元顯, “中文手機新聞簡訊自動摘要”, 第十六屆自然語言與語音處理研討會, 台北, 2004 年 9 月 2-3 日, 頁 177-189.

[10] 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/

[11] 張道行,「Automatic Chinese Unknown Word Extraction Using Small-Corpus-Based Method」,國立交通大學,博士論文.(2003)

[12] 余思翰,「中文作文寫作輔助系統」,國立交通大學,碩士論文.(2007)

[13] 陳智維,「中文情書自動產生系統」,國立交通大學,碩士論文.(2008)

[14] Google URL：http://www.google.com.tw/

[15] HowNet 知網 URL：http://www.keenage.com/

[16] 無名小站網誌 URL：http：//www.wretch.cc/blog/

[17] Yahoo!奇摩 URL：http://tw.blog.yahoo.com/