國立交通大學

資訊科學與工程研究所

碩 士 論 文

中 文 作 文 寫 作 輔 助 系 統

Chinese Essay Writing Auxiliary Systems

研 究 生：余思翰

指導教授：李嘉晃 教授

中 華 民 國 九 十 七 年 一 月

中文作文寫作輔助系統

Chinese Essay Writing Auxiliary Systems

研 究 生：余思翰　　Student：Szu-Han Yu

指導教授：李嘉晃　　Advisor：Chia-Hoang Li

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

Jan 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年一月

中文作文寫作輔助系統

學生 ： 余思翰　　指導教授：李嘉晃 博士

國立交通大學資訊學院　資訊科學與工程研究所碩士班

中文摘要

本文以數百篇指定題目的中學生文章為基底，實做出一個作文寫作輔助系統。首先將作文語料庫的文章進行分解、重新整理，再利用處理過後產生的文字片段拼湊出一篇新的、同為指定題目的作文。原型系統對語料庫文章的句子作切割，並以接龍的方式來產生新的文章，接著對原型系統在架構及介面上加以改良，改良後系統則是以關鍵詞串列作為新文章的內容架構，再提供填充字串讓使用者完成一篇可閱讀的作文。兩個系統皆充分利用語料庫文章取材類似，句法偏向簡短的特性，並且有機會產生一篇十分類似中學生所寫的作文，無論在題材、或是句型方面都和語料庫中的文章十分相像。本文提出的系統仍在發展初段，仍有改進空間，但已表達出一個作文寫作輔助系統可能的進行方式。

Chinese essay writing auxiliary systems

Student ： Shin-Hung Lin Advisor：Prof. Chia-Hoang Lee

Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

Abstract

In this paper, we propose and implement an auxiliary essay writing systems, which make use of hundreds of essays of some specific topic written by junior high school students. We decompose and reorganize the essay corpus into word segments, which will later be used to construct a new essay of the same topic. In the primitive system, each sentence in the corpus is partitioned into units. Users can select among these units to reforms a new essay by the way of solitaire. Then we continue to improve the system in framework and user interface. In the improved system, keyword lists are extracted from the corpus and stuffing strings are provided for the user to generate and modify a new human-readable essay. Both of two systems were made available by the features of similarity among essays in the corpus and the essays most constitute of shorter sentences. Each of the systems gets the chance to generate a new essay that highly resembles one written by a real high school student in materials and sentence pattern. Systems proposed here are at preliminary stage, having a lot of room to perform better. However, we give a glimpse of what an auxiliary essay writing system may look like and address a possible way that a text generation system can adopt.

目錄

圖目錄

表格目錄

第一章、緒論

1.1 研究動機

自動文本產生在自然語言處理中是一個奧妙且有趣的範疇，在開始下筆寫一篇作文之前必先費心構思整篇文章的主旨、論點、架構等文章內容的骨幹，在寫作之時，則必須注意用字遣詞是否優美恰當，並加入各種修辭方法如譬喻、排比、轉化等來加強讀者對文章的感受。寫作的學問是如此精深，致使各類寫作教學書籍應需求而生。然而，對於一些容易發揮的中學生作文題目，學生寫出來的文章立意取材皆十分相似，差別僅在於修辭的手法和文章的結構佈局。在一些實驗後發現，不同文章中，主題相同的兩個句子，彼此接合後極有機會變成一句語意語法皆通順，且在其他篇文章皆未曾出現過的新句子。因此本文便藉機探討以文章接龍與改良後的合成方式來實做一個作文輔助系統，及其實用性及後續研究。

1.2 研究目的

如前文所述，寫作是一門深奧的藝術，一篇文章從內容構思、語句修飾到最終的細節整理往往耗費不少時間。尤其取材必須兼顧靈活和不背離題目，若非箇中老手勢必要花上許多工夫。因此，本文期盼藉著以"下課十分鐘"為題共 689 篇由中學生所撰寫的文章所構成的語料庫為基底，用語料庫文字片段擷取的方法來做出一個半自動的作文寫作輔助系統，目的在輔助使用者能在短時間內產生一篇通順的文章，當作自己寫作的範本，縮短作文寫作所需的時間。

1.3 論文架構

第一章為前言，敘述本文的研究動機與目的，提到本文系統之所以可行的原因及基本概念。第二章為相關研究，說明自動文本產生的概念及實行的方法，包括內容選擇(content selection)及斷詞系統。第三章將說明原型系統－接龍系統所需的前置作業、基本架構以及系統運行的方式，並在最後一節探討系統在使用上的缺點，改良後的系統－合成系統各項細目則在第四章一一說明。在第五章我們將會檢討系統改良後的成效以及介紹程式的使用者介面。第六章我們探討系統的可能的後續工作。

第二章、相關研究

本章節敘述相關研究及構想，包括自動文本產生議題探討。

2.1 內容選擇

內容選擇(content selection)是概念-文本產生中一個重要的部份，一個內容選擇元件（content selection component）決定一個自然語言產生系統在文本產生時該包含哪些內容，這類系統通常都會利用大型的資料庫，其中包含系統輸出所可能涵蓋的內容。內容選擇最重要的一點是選出內容的共存性(coherence)，讓產生出來的文本能連成一氣["to produce a text that hangs together", McKeown, 1985]。內容選擇可視為一種分類的工作[Duboue and McKeown ,(2003)]，可利用一個文本集合以及其對應固定某個領域的資料庫（表格型態）來學習資料庫中哪些項目（entry）該成為輸出文本的內容。Regina Barzilay 和 Mirella Lapata（2003）為一個足球賽事報告產生系統提出一個集合式的內容選擇模型(collective content selection)，他們考慮資料庫中所有項目的子集合，並計算每個子集合的語意關聯性，以分數最佳的子集合當作選擇的內容。

本文系統所使用的語料庫為數百篇同一題目的中學生作文。由於題目相同，加上中學生的寫作能力差距不會太大，使得改良後系統的內容選擇工作較為輕鬆，我們萃取出語料庫文章的關鍵詞串列成為輸出文章的內容骨幹。

2.2 斷詞與詞性標記

英文中最小的語意單位為詞(word)，詞由字母(letter)所組成，在一個句子中，每個詞之間以空白隔開；而在中文句子中通常不存在任何空白，所有中文字

(Chinese character 相當於英文 letter)彼此相鄰，故須先進行斷詞工作，將句子分割為一串詞(相當於英文 word)，之後便可如同英文一般進行詞性標記的工作。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。本系統使用中央研究院資訊科學研究所詞庫小組中文斷詞系統 1.0 版作為斷詞的工具，其正確率在 95%-96%間[http://ckipsvr.iis.sinica.edu.tw/apply.htm]。

第三章、系統設計與實驗 － 接龍

3.1 概念

文字接龍是指從目前已經有的字串中猜測決定下一個該出現的字(詞)，如果每次只決定一個字(詞)，則字串長度每次增加一，如此接龍進行的速度將十分緩慢。每次決定的字(詞)數愈多，接龍的速度愈快。為了保持每次接龍語意上的完整，本文定義"常用單元"，即是在寫作時經常重複使用的文字片段，像是「最 快樂 的 時間」、「活動 一下 筋骨」等常用用語。每次接龍決定一至多個常用單元不僅可讓接龍速度加快，也較符合視覺上的感受。

3.2 前置作業

接龍系統所採用的作文語料庫為 689 篇中學生作文，作文的題目為"下課十分鐘"，每一篇皆經過人工批改，依文章的內容分為 1 ~ 6 級分，各級分的篇數依序為 45、128、210、208、91、7 篇，經觀察後發現 1、2 級分的文章離題情況嚴重，錯字繁多且較多不通順之語句，3、4 級分的文章語句雖較通順但內容較不豐富，故本系統僅採用 5 級分以上的文章作為資料庫，累計共 98 篇。所有文章皆經過斷詞處理及詞性標記，我們以 ES 表示經過斷詞及詞性標記處理過後的作文語料庫。

3.2.1 常用單元

本系統定義在作文語料庫中出現頻率大於等於 2 的文字片段為常用單元。我們定義在經過斷詞處理後的文章中連續 n 個詞的序列為以詞為底的 n-gram(在

本文中皆簡稱為 n-gram），其長度為 n；我們用 Gram(S,n)表示句子 S 中所有 n-gram 所成的集合； Gram(S,m,n) = $\bigcup_{i=m...n} Gram(S,i)$ 表示不同長度 n-gram 的集合。以下列句子為例：

原句：好不容易的熬過了這四十五分鐘。

S：好不容易 的 熬過 了 這 四十五 分鐘 。

Gram(S,2)：{ 好不容易 的, 的 熬過, 熬過 了, 了 這,
這 四十五, 四十五 分鐘 }

Gram(S,3)：{ 好不容易 的 熬過, 的 熬過 了, 熬過 了 這,
這 四十五 分鐘 }

Gram(S,2,3) = Gram(S,2) ∪ Gram(S,3)

將作文語料庫中每一篇文章中的每一個句子(以逗號、句號、分號及驚嘆號為準)的 n-gram set 聯集起來並紀錄頻率，形成一個作文語料庫的 total n-gram set，記為 Gram(ES,m,n) = $\bigcup_{S \in ES} Gram(S,m,n)$

由於作文資料庫總字數並不多，無法確實反應每個 n-gram 的使用情形，故本系統將中研院平衡語料庫 3.1 版納入參考，Gram(ES,m,n)中每個 n-gram 的總頻率為其分別在作文語料庫及平衡語料庫中出現的頻率的總和，我們以 freq(g)表示 n-gram g 的頻率。最後，保留 Gram(ES,m,n)中頻率大於 1 的 n-gram 成為常用單元集並記為 Gram_Dict。

3.2.2 句子分割

一個句子可視為數個常用單元的組合，如句子：「是 在 學校 最 快樂 的 一段 時間」可分割成：[是 在 學校][最 快樂 的][一段 時間]，或是：[是][在 學

校][最 快樂][的 一段 時間]，分割法並不唯一。長度愈長的單元所包含的語意愈完整，我們希望將句子切割成較少的單元（每個單元會較長），因此，本系統採用最長長度分割法來完成語料庫的句子分割。

最大長度分割法演算法如下：

$S = w_1 w_2 w_3 \cdots w_n$

Do{

$FU = Gram(S, 2, \min\{n, 6\}) \cap Gram\_Dict$

$Max\_Len\_U = \{U \,|\, U \in FU, length(U) = \max\_len(FU)\}$

$m_u =$ unit in Max_Len_U with max frequence

$S = u_h m_u u_t$

}

遞迴處理u_h及u_t直到$|u_h|$及$|u_t|$小於1

S 表示一個經過斷詞處理的句子，max_len(G)為 n-gram set G 中所有長度最長的 n-gram 的集合，max_freq(G)則代表 G 中頻率最高的 n-gram，以 S = "這 就是 我 下課 十 分鐘 所得 到 歷險記"為例，將 S 切割成 Gram(S,2,6)後，長度最長且存在於 Gram_Dict 的單元為[下課 十 分鐘]，以此對 S 作分割後可得u_h＝ "這 就是 我"，m_u＝ [下課 十 分鐘]，u_t＝ "所得 到 歷險記"，將視為另一個新的句子處理，此時 S = m_u，因此u_h與u_t皆為長度為零的空字串，遞迴處理結束。對作文語料庫中每一篇文章所有句子做分割後可得到分割好的作文語料庫 PES，一篇分割好的文章可表示成 $pes = u_1 u_2 \cdots u_n$，我們用 unit(i,j)表示第 i 篇分割好文章中的第 j 個單元。圖 1 為一篇語料庫作文的第一段經過分割後的樣子。

[每][走 一 步]
[就 好像][挖 到][了 一 塊][寶藏]
[好像][偵探][一步步][的 把][線索][找出來][一樣]
[有 一 種 莫名 的][興奮]
[像是][冒險][一樣]
[這 才 知道][校園][有 很多 的][奧秘][等 著 我們 去][解開]
[去 發掘]
[這 就是 我][下課 十 分鐘][所得][到][歷險記]

圖 1 以最大長度分割法分割過的段落

3.3 系統架構

接龍系統架構可分為初始單元集的生成和接龍所必須的候選單元集的產生，以及最後供使用者選擇的候選單元串列集。

3.3.1 初始單元集

接龍系統是由一個最初的單元開始，不斷往後接直到文章完成為止。我們知道一篇作文的各段落有其規範的寫法（如常見的起、承、轉、合），若將作文語料庫中後段的單元當作起始單元，產生的文章結構較不完整、自然。所以，我們將分割好的作文語料庫中每一篇第一句話的第一個單元蒐集起來成為初始單元集，記為 Ini_Units。初始單元集將提供使用者作文接龍的第一個單元，圖二顯示初始單元集中的第一到九項。

[下課 是]
[每當 下課 時]
[叮噹 叮噹]
[「]
[叮咚 一]
[噹]
[聽到 了]
[噹噹 噹噹]
[噹]

圖 2 初始單元集的一部分

3.3.2 候選單元集

接龍的概念是參考已存在的文字片段來決定下一個要接在其後的詞（在本系統中為單元），故系統必須提供一個機制根據前文決定一組可能適合當作接續的單元組，我們稱這組單元為候選單元集。給定一個作文單元 u，一個直覺的方法是在切割好的作文語料庫中找出所有最後一個詞跟 u 的最後一個詞相同的單元，將這些單元的下一個單元蒐集起來成為候選單元集，但中文的文法不像英文等語文那樣嚴謹，我們可以利用中文模稜兩可的特性放寬接龍的條件，如此可避免接出來的文章可能只是語料庫中的某篇作文而已，因此我們不限制最後一個詞要跟 u 的最後一個詞相同，只要一個單元包含的詞和 u 包含的詞交集非空，便把該單元的下一個單元視為候選單元。定義單元 u 及其對應的候選單元集為

$$Candidate(u) = \{\ c_u = unit(i,j)\ \ | |Words(\text{pre\_unit}(c_u)) \cap Words(u)| > 0,\ for\ all\ i,j\ \}$$

Words(u)為單元 u 所包含的詞，pre_unit（unit(i,j)）＝ unit(i,j-1)。也就是在分割好的作文語料庫中搜尋所有其 Words 集合和 Words(u)交集個數大於 0 的單元，將這些單元的下一個單元蒐集起來便是 Candidate(u)。假設 u = [每當 下課 時]，Words(u) = {每當,下課,時}，作文語料庫中所有與 Words(u)交集大於等於 1 的單元及其下一個單元包括：

unit(1,0) unit(1,1) = [每當 下課 時] [就是]

unit(47,21) unit(47,22) = [此 時 的] [心 都]

unit(26,5) unit(26,6) = [下課 時] [大家 都]

unit(24,13) unit(24,14) = [下課 時] [每 個 人 都]

unit(7,0) unit(7,1) = [下課 時] [同學 紛紛]

……

……

陰影部份蒐集起來便是 Candidate(u)。

3.3.3 候選單元串列集

如果直接使用候選單元集，那麼每接一次字串長度會增加 1 單元的長度（2 到 6 個詞），經實際操作後發現，要完成一篇文章，使用者選擇的次數將十分頻繁。要減少使用者挑選候選串列集次數的方法便是增加每個候選單元的長度，我們定義長度為 n 的候選單元串列集由候選單元集延伸而成，一個長度 n 的候選單元串列為連續 n 個在原文中相鄰的單元，記為

$$Candidate(u,n) = \{unit(i,j)unit(i,j+1)unit(i,j+2)\cdots unit(i,j+n-1) \mid unit(i,j) \in Candidate(u)\}$$

n 為 1 的候選單元串列集就是候選單元集。候選單元串列集的長度關係到接龍速度的快慢。根據對切割好的作文語料庫的觀察，一個句子約略由三個單元組成，因此我們將候選單元串列集長度設為 3，這樣接龍便會以大概一次接一句話的速度進行。同樣以 u = [每當 下課 時]為例：

Candidate(u,3) = {

unit(1,1)unit(1,2)unit(1,3) = [就是][我 最 期待 的][時候]

unit(47,22)unit(47,23)unit(47,24) = [心 都][有如][平原 上]

unit(26,6)unit(26,7)unit(26,8) = [大家 都][打成一片][、]

unit(24,14)unit(24,15)unit(24,16) = [每 個 人 都][好像 從][鬼門關]

unit(7,1)unit(7,2)unit(7,3) = [同學 紛紛][離開][自己 的 座位]

……

……

……

}

圖 3 顯示依據單元[每當 下課 時]所產生的候選單元串列集排序過後的前 9 項，下一節將提到排序的方法。

[每當 下課 時] [就是][我 最 期待 的][時候]
[同學 紛紛][離開][自己 的 座位]
[大家 都][打成一片][、]
[每 個 人 都][好像 從][鬼門關]
[總是][有 幾 分][喧鬧]
[熱鬧][在 上課][又 繼續]
[心 都][有如][平原 上]
[心靈 深處][最 真實 的][吶喊]
[大家 都][露出 了][大大]

圖 3 依據單元[每當 下課 時]所產生的候選單元串列集

3.3.4 候選單元串列集排序

不是每個候選單元串列都可以和已存在的單元串列完美結合，不良的接龍狀況包括：語句不通順或無意義以及和已存在的單元串列產生語意上的矛盾。而候選單元串列集往往包含數十個選項，使用者必須一一查看哪些選項適合接在已存在的單元串列之後，十分費時及傷神。若能對候選單元串列集事先作排序，將接龍效果較好的候選單元串列排在較前面，讓使用者只需觀看前幾項便可挑出適合的選項，將可提高系統的實用性與便利性。排序的方法有許多種，一個直覺的想法是：一個候選單元串列和已存在單元串列的契合度可由此候選單元串列在原文中的前文和已存在單元串列的相似度來略為判斷，簡單的實做方法就是計算已存在單元串列最後一個單元和候選單元在原文中前一個單元的交集。假設目前已存在的單元串列為 Cur_Unit_List：$u_1u_2u_3\cdots u_n$ ，

而 Candi_Unit_List = $cu_1cu_2\cdots cu_m$ ∈ Candidate(u_n,m) 為候選單元串列集中的一個串列，定義

$$S(Candi\_Unit\_List) = \left|Words(\text{pre\_unit}(cu_1)) \cap Words(u_n)\right|$$
$$for\ \ a\ \ given\ \ Cur\_Unit\_List$$

為候選單元串列 Candi_Unit_List 相對於已存在的單元串列 Cur_Unit_List 的語意關聯分數，也就是在蒐集 Candidate(un)的過程中，用以挑選符合條件的單元的公式所得到的值。依據語意關聯分數對候選單元串列集作排序，如此一來便可減輕使用者挑選的負擔。

假設 Cur_Unit_List = [每當 下課 時](un = [每當 下課 時])，

Candi_Unit_List = [大家 都][打成一片][、]

S([大家 都][打成一片][、])

= S(unit(26,6)unit(26,7)unit(26,8))

= | Words(pre_unit(unit(26,6))) ∩ Words(un) |

= | Words([下課 時])Words([每當 下課 時]) |

= 2

而 Candidate(u_n,3) 內各選項依照 S 分數排序後順序為:

[就是][我 最 期待 的][時候]

[同學 紛紛][離開][自己 的 座位]

[大家 都][打成一片][、]

[每 個 人 都][好像 從][鬼門關]

[心 都][有如][平原 上]

……

……

……

3.4 系統運行

作文接龍一開始讓使用者從 Ini_Units 挑選一個初始單元作為接龍的起始，設為 Cur_Unit_List。接來下便依據 Cur_Unit_List 最後一個單元產生一組候選單元串列集供使用者挑選，將使用者選定的候選單元串列附加到 Cur_Unit_List 之後，不斷重覆此過程直到接龍文章完成(由使用者自行判斷是否完成)。圖表 4、5、6 大致說明系統運作流程：系統一開始的畫面如圖 4 所示，接龍模式分為開頭、中段和結尾，分別經由標示為 1、2、3 的按鍵來切換；標示為"，"及"。"的按鍵則是讓使用者在文章中加入逗號或句號；Save 按鍵可將接龍的文章存成以按鍵左邊文字方塊內所顯示的文字檔；Start 鍵開始一篇新的接龍，而 Clear 鍵則清除所有顯示，按下 Start 鍵後左側文字方塊便會顯示初始單元集(如圖 5 所示)，點選其中一項，再按下 Add to Text 鍵後，所選之單元串列便會加入在下方的文字方塊內，同時重新顯示下組候選單元串列(依據所選之串列，如圖 6 所示)，使用者藉由循環來完成接龍文章。

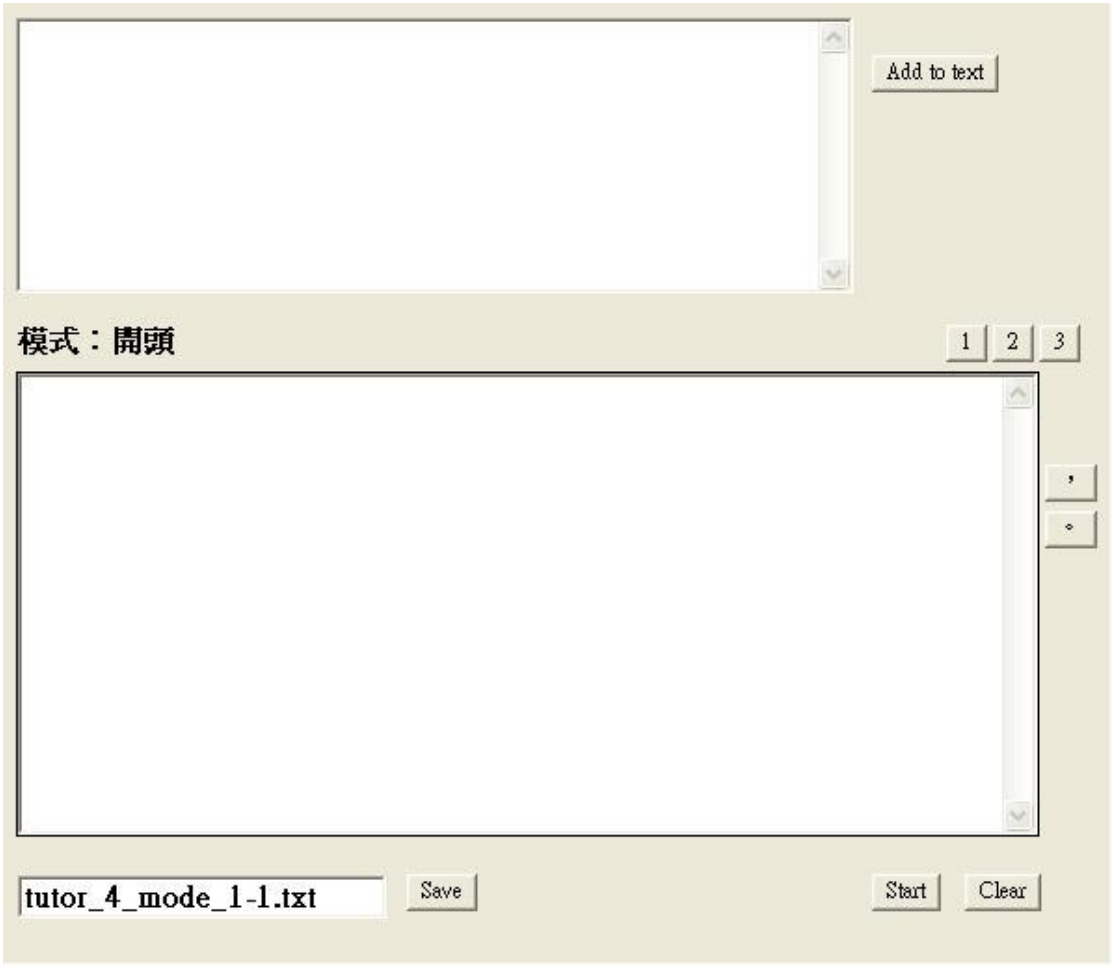

圖 4. 接龍系統的程式起始畫面

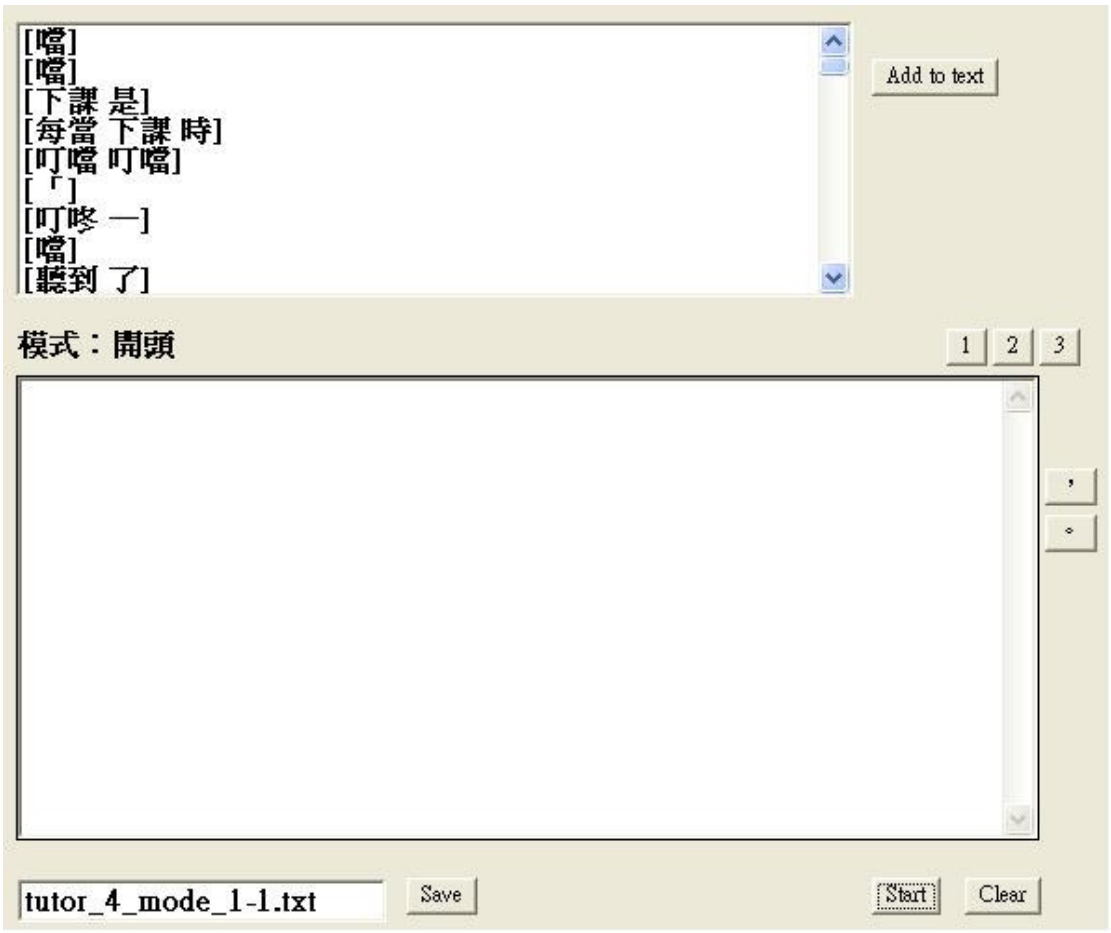

圖 5 按下 Start 鍵後便會顯示初始單元集

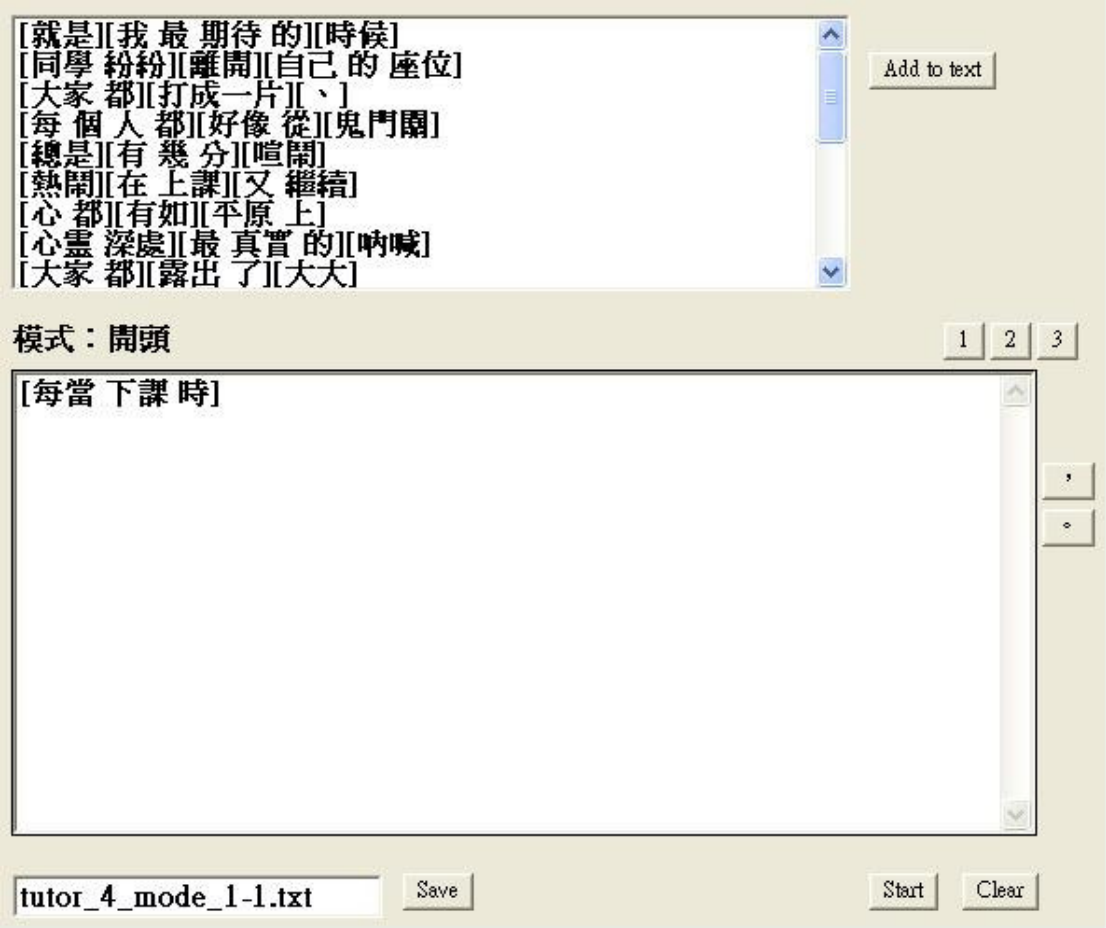

圖 6 選好初始單元，按下 Add to text 後的畫面

3.5 實際操作與檢討

唯有實際操作才能真實地判斷出一個系統的好壞，並找出系統在背景理論和使用界面上的優缺點，接龍系統為作文輔助寫作系統一個初步的實現，因此仍有許多需要改進的地方。在不斷反覆操作系統之後，可以歸納出數個必須改進的地方。首先，系統缺乏主題導向功能，在系統顯示出下一組候選單元串列集之前使用者完全無法預知接龍文章接下來究竟會往何處發展，不能事先瞭解作文內容最後會包含哪些題材，也不能隨心所欲地接出自己想要的文章。從寫作的觀點來看，一篇好的作文必須先擬定大綱、選擇題材，再有條理的將這些概念組織成文章段落，若系統能提供一個功能讓使用者事先瞭解欲產生文章的大綱，較符合一般寫作的流程。其次是候選單元串列集選項過多的問題，雖然經過排序後會將較

佳的選項往前移，但仍有可能唯一能使前後文通暢的選項還是排在較後面，如果限制串列集的元素個數可能會讓接龍無法順利進行，這導致使用者必須逐一仔細觀看每個選項並仔細比對檢查前後文，如此一來，要將文章完成必須花上不少時間，作文輔助系統的目的便是要讓使用者輕鬆的完成一篇文章，因此，這部份也必須做調整。此外，另一個需要調整的是：候選單元串列集內的所有選項皆無法與前文配合，就像走進死胡同一般，這種情況一旦發生，使用者只能選擇重新開始，並不是每次都能接出一篇自然的文章。圖 7 為一段不通順的接龍片段，文章前半還算流暢，但後半便不知所云，即是所有選項皆無法與前文搭配的後果。

[在 聽到][熟悉 的 聲音][就 知道 是][下課 了]，[這 是][每 位 同學][期盼][已 久 的][短暫][時光]，[是 辛苦][上 了][一 堂 課 的][每 個 人][如同][剛 從][福利社][跑 了][進去][腦 中][彷彿][就 會 呈現][銷售][一空][這 四 個 字]

圖 7 不通順的文章片段

第四章、系統功能改良

第三章說明了作文寫作輔助系統－接龍系統的概念、實作以及操作檢驗，在本章中，我們將針對 3.5 節提到的缺點進行改良。在相關研究中指出，文章的生成通常需要兩個階段，內容選擇以及連結所選的內容，此一概念亦可套用在作文寫作輔助系統上，設計出適合本系統內容選擇和連結的演算法，系統也將改以合成的方式進行。

4.1 概念

自動文本產生系統中通常分為兩個部份，第一部份是挑選文本的材料，也就是文章的主要內容概念，目的是產生文章的骨幹；第二部份則是句子完成，是指將概念延伸成為完整的句子，就像是為骨幹填上血肉。改良系統將作文語料庫中所包含的關鍵詞視為一篇作文的主要概念，定義了關鍵詞後，對於任何關鍵詞串列中的兩個相鄰關鍵詞，便可在整個作文語料庫中搜尋可填入其中的字串達到文章合成的功能。我們先對作文語料庫中的關鍵詞作定義，接著將語料庫中的每篇文章分離成關鍵詞串列和填充字串兩部份，成為系統可利用的格式。

4.2 前置作業

本系統同樣需利用經過斷詞處理和詞性標記的作文語料庫，並將 3、4 集分的文章加入，共 506 篇。

4.2.1 關鍵詞

我們發現有些詞在作文語料庫中出現的頻率相當的高，但在一般語料庫（平衡語料庫）中的使用卻並不頻繁，這些詞可視為作文語料庫的關鍵詞。由於平衡語料

庫和作文語料庫的字數比例懸殊，因此不能單憑在這兩個語料庫中出現的頻率來判定一個詞是否為關鍵詞，像是"同學"在平衡語料庫中的頻率 1640，高於作文語料庫中的頻率 948，但幾乎每篇作文都會寫到"同學"，而平衡語料庫中僅有一小部份比例的文章提到"同學"，因此"同學"應該視為關鍵詞。但若將頻率除以語料庫總字數，則會由於兩個語料庫懸殊的字數比例而導致幾乎每個詞都是關鍵詞。若以詞性來看，許多作文語料庫包含的名詞如：學校、操場、老師、教室、廁所；形容詞如：水洩不通、快樂、大排長龍等，均與題目密切相關，可當作關鍵詞，而部分功能詞如：雖然、就 出現的頻率也相對較高，同樣可視為關鍵詞，但一些名詞如：音樂、噪音僅在一兩篇文章中出現；大部分功能詞如：的、地、在等並不適合當作關鍵詞，因此，光是以詞性來判斷是否為關鍵詞似乎也並不準確。我們發現把作文語料庫和平衡語料庫中的所有詞彙分別蒐集起來並將其依照出現的頻率排序，觀察一個詞彙分別在兩個語料庫中的排名，較能準確顯示該詞彙在兩個語料庫中的使用傾向，因此我們利用這項特徵來挑出所有的關鍵詞，在經過一些觀察後，我們將邊界值設為 100 是可行的，只要在平衡語料庫中的排名減去在作文語料庫的排名差距大於 100 便可視為關鍵詞。但如果一個詞在兩個語料庫中出現的頻率都不高時，由於平衡語料庫的詞彙數數倍於作文語料庫，其在兩者中的排名差距必定懸殊，造成低頻詞都會成為關鍵詞，這並不合理，因此我們僅取前 300 個詞為關鍵詞。表格 1 為部份關鍵詞顯示。

表格 1 萃取出來的關鍵詞組的部份顯示

關鍵詞	詞性	排名 (平衡語料庫)	排名 (作文語料庫)
下課	VH	6359	2
，	COMMACATEGORY	695	4
分鐘	Nf	978	8
十	Neu	334	9
時間	Na	132	17
學生	Na	123	19

上課	VA	1347	25
同學	Na	351	26
老師	Na	194	27
教室	Nc	1479	28
大家	Nh	150	33
玩	VC	680	36
些	Nf	273	39
。	PERIODCATEGORY	94818	40
合作社	Nc	3031	41
就是	Cbb	630	48
有的	Neqa	570	49
校園	Nc	711	52
去	VCL	288	54
噹	D	18776	57

4.2.2 關鍵詞串列與填充字串

定義關鍵詞之後，便可對作文語料庫中的每一篇文章作關鍵詞與填充字串作分離的動作。作法是先標記一篇文章中的關鍵詞，然後將這些關鍵詞萃取出來形成一個關鍵詞串列；而兩個關鍵詞之間的其他詞則標記成這兩個關鍵詞所對應的填充字串。關鍵詞串列代表一篇文章的骨幹，每篇語料庫文章可對應到一個關鍵詞串列，但一串關鍵詞串列卻可能對應到多篇文章；而兩個關鍵詞所對應的填充字串則代表這兩個關鍵詞如何可銜接成一個可閱讀的句子。對所有作文語料庫的文章完成分離後，便可找出任意兩個關鍵詞間的所有填充字串。我們用 Keyword_list(i)表示從分離好的作文語料庫中第 i 篇文章萃取出來的關鍵詞串列; Keyword_list(i,j)表示第 i 篇文章關鍵詞串列的第 j 個關鍵詞,而 S_string(i)[k,k+1]表示第 i 篇文章中夾在第 k 個和第 k+1 個關鍵詞間的填充字串。以作文語料庫中第 3-27 篇的第一段為例，以下為經過斷詞及詞性標記後的結果，粗體部份為關鍵詞：

這(Nep) **短短(VH)** 的(DE) **十(Neu)** **分鐘(Nf)** ，對(P) 每(Nes) 個(Nf) 人(Na) 來(D) 說(VE) 或多或少(D) 都(D) 有(V_2) 一定(A) 用處(Na) 。不同(VH) 的(DE) 人(Na) 對(P) **時間(Na)** 快慢(Na) 有(V_2) 不同(VH) 感受(Na) ，在(P) 這(Nep) **十(Neu)** **分鐘(Nf)** 充實(VHC) 自己(Nh) 的(DE) ；在(P) 這(Nep) **十(Neu)** **分鐘(Nf)** 得過且過(VH) 的(DE) ，你(Nh) 問(VE) 我(Nh) **該(D)** 做(VC) 什麼(Nep) — **時間(Na)** 是(SHI) 自己(Nh) 的(DE) ，**好好(VH)** **把握(VC)** **吧(T)** ！

將關鍵詞和期間的填充字串分離後可得到關鍵詞串列：

S0 **短短(VH)** **S1** **十(Neu)** **S2** **分鐘(Nf)** **S3** **時間(Na)** **S4** **十(Neu)** **S5** **分鐘(Nf)** **S6** **十(Neu)** **S7** **分鐘(Nf)** **S8** **該(D)** **S9** **時間(Na)** **S10** **好好(VH)** **S11** **把握(VC)** **S12** **吧(T)** **S13**

以及期間所對應的填充字串，null string 為空填充字串：

S0：這(Nep)

S1：的(DE)

S2：null string

S3:，對(P) 每(Nes) 個(Nf) 人(Na) 來(D) 說(VE) 或多或少(D) 都(D) 有(V_2) 一定(A) 用處(Na) 。不同(VH) 的(DE) 人(Na) 對(P)

S4：快慢(Na) 有(V_2) 不同(VH) 感受(Na) ，在(P) 這(Nep)

S5：null string

S6：充實(VHC) 自己(Nh) 的(DE) ；在(P) 這(Nep)

S7：null string

S8：得過且過(VH) 的(DE) ，你(Nh) 問(VE) 我(Nh)

S9：做(VC) 什麼(Nep) —

S10：是(SHI) 自己(Nh) 的(DE) ，

S11：null string

S12：null string

S13：null string

處理過後的作文語料庫，每篇作文詞數平均為 298.6，標準差為 62；平均關鍵詞數為 74.6，標準差為 17.9；平均關鍵詞密度為 0.25，標準差為 0.04，可看出詞數和關鍵詞數約成比。

4.3 系統架構

4.3.1 候選關鍵詞串列集

如前文所述，一個關鍵詞串列可視為一篇文章的骨幹，由於每篇文章的關鍵詞密度相差不大，因此一篇文章愈長，其所萃取出的關鍵詞串列也愈長，所以我們可藉由關鍵詞串列的長短來控制合成文章的長度(總字數)。改良系統提供三種關鍵詞串列長度：較短（10 個關鍵詞）、中等(20 個關鍵詞)、較長(30 個關鍵詞)供使用者選擇。候選關鍵詞串列的產生的方式是：從已經分離好的作文語料庫中隨機挑選數個關鍵詞串列，並擷取每個串列的前 n 個詞。我們將選出來的關鍵詞串列記為 Sel_list，其中第 i 個關鍵詞記為 Sel_list(i)，直接從語料庫作文萃取關鍵詞串列的好處是可以確保一個關鍵詞和其左右鄰近其他關鍵詞在語意上的關聯性，這避開了兩個關鍵詞因關聯性不高而無法連結成一個句子的窘境。

4.3.2 候選填充字串集

有了關鍵詞串列，便可為這個串列填入填充字串，改良系統提供的方法是：只要使用者選擇串列中相鄰的兩個關鍵詞，便會啟動搜尋機制，從分離好的作文語料庫中找出所有這兩個關鍵詞間的填充字串作為候選填充字串集。假設使用者所選的關鍵詞串列為 Sel_list，而第 k 篇作文關鍵詞串列中的第 j 和 j+1 個關鍵詞分別與 Sel_list 中第 i 和 i+1 個關鍵詞相等，則其對應的填充字串 S_string(k)[j,j+1] 可填入 Sel_list 的第 i 和 i+1 個關鍵詞間，因此候選填充字串集可定義如下：

$$SCandi(Sel\_list, i, i+1) = \{$$
$$S\_string(k)[j, j+1] \mid$$
$$Keyword\_list(k, j) = Sel\_list(i) \;\&\&$$
$$Keyword\_list(k, j+1) = Sel\_list(i+1)$$
$$j = 1 \cdots \text{length}(Keyword\_list(k)) \;\; for \;\; all \;\; k$$
$$\}$$

舉例來說，假如 Sel_list 為：

同學(Na) 喜歡(VK) 短短(VH) 下課(VH) 時間(Na) 吧(T) 節(Nf) 課(Na) 十(Neu) 分鐘(Nf)

可填入第 0 個詞"同學"和第 1 個詞"喜歡"之間的填充字串集為

SCandi(Sel_list,0,1) ＝ {

S_string(3)[12,13] ＝ "便 帶 著 所"

S_string(94)[19,20] ＝ "都 很"

S_string(297)[2,3] ＝ "都 非常"

S_string(297)[87,88] ＝ "都"

S_string(470)[12,13] ＝ "所 期待 的 。 我"

……

…

}

圖 8 為可填在"同學"、"喜歡"兩個關鍵詞之間的候選填充字串集的一部份，圖中為排序過後的結果，no words 表示兩個關鍵詞間可不用再填入任何文字。

圖 8 "同學"及"喜歡"對應的候選填充字串集

4.3.3 候選填充字串集排序

如同候選單元串列集，我們也對候選填充字串集作排序的動作以減輕使用者的負擔。考慮 Sel_list(k)和 Sel_list(k+1)及其所對應，屬於 SCandi(i,i+1)的一個填充字串 S_string(i)[j,j+1]，在直覺上我們認為關鍵詞串列 Keyword_list(i)的第 j+2 個關鍵詞若與 Sel_list(k+2)相同，則此填充字串在語意上與現有串列搭配度應該與兩者不相同時來的好。因此定義語意關聯分數

$$NS(S\_string(i)[j,j+1]) = \begin{cases} 0 \ if \ \ Sel\_list(k+2) \neq Keyword\_list(i,j+2) \\ 1 \ if \ \ Sel\_list(k+2) = Keyword\_list(i,j+2) \end{cases}$$
$$S\_string(i)[j,j+1] \in SCandi(Sel\_list,k,k+1)$$

此外，一個填充字串在原文中的位置也會對將其加入合成文章後，合成文章的整體流暢度有所影響，舉例來說，在考慮 Sel_list 中的第 1、2 個關鍵詞時，單就位置而言，假設作文語料庫中文章的關鍵詞密度相差不大，$S\_string(i_1)[2,3]$ 可能會優於 $S\_string(i_2)[15,16]$，這是因為 $S\_string(i_1)[2,3]$ 通常會比較符合文章首段該有的內容(一篇語料庫文章不太可能只有 4、5 個關鍵詞)；而 $S\_string(i_2)[15,16]$ 的內容會比較適合放在文章的中後段，所以，我們再定義填字充串的位置分數：

$$PS(S\_string(k)[j, j+1]) = \begin{cases} 1/i-j & if \ \ i \neq j \\ 1 & if \ \ i = j \end{cases}$$

其中 S_string(k)[j,j+1] ∈ SCandi(i,i+1)，如此的算法至少可避免接近文章末段的文句出現在合成文章的一開始。而 Total_score(S_string(k)[j,j+1]) = NS + PS，我們依據Total_score對SCandi(i,i+1)裡的候選填充字串排序。圖 9 為填充字串"都 非常" (相對於關鍵詞"同學"、 "喜歡"所產生之候選填充字串集)的各項參數資料，next keyword 代表在原文中的下一個關鍵詞(在圖例中，"喜歡"的下個關鍵詞為"下課")；es id 為所屬文章編號，position 代表在關鍵詞串列中的位置，如"喜歡"是串列中第 87 個關鍵詞；score 為依照 4.3.3 節所述排序方法所得到的分數；between words 表示該填充字串。

圖 9 填充字串的各項參數

使用關鍵詞串列做文章合成，文章會在最後一個關鍵詞結束，此時有可能出現語意不完整的情況，看起來就像是文章的最後一句話被最後一個關鍵詞截斷(關鍵詞只是一種表達概念的方法)，為了解決這個問題，我們附加"。"到使用者所選出的關鍵詞串列最後，讓"。"成為所有串列的最後一個關鍵詞，這樣一來，我們就可以在原本串列的最後一個關鍵詞和新加入的"。"之間也填入填充字串，語意截斷的問題可以因此獲得改善。

4.4 系統運行

作文合成分成兩個步驟，第一個步驟為關鍵詞串列的選擇，本系統預設每

次顯示 10 個關鍵詞串列供使用者選擇，使用者可按更新鍵更新系統顯示的候選關鍵詞串列集。第二個步驟是預設合成文章顯示以及合成文章的修改。關鍵詞串列一旦選定便可決定兩兩相鄰關鍵詞間的所有的候選填充字串集，這些填充字串集經過排序後，由系統各自從排名前五項的填充字串中隨機挑選一項出來合成系統的預設文章。當然，系統預設文章可能不是整篇都很通順，因此，本系統提供機制，讓使用者可找出相鄰兩個關鍵詞間所對應的其他填充字串集，以便讓使用者修改預設文章，系統介面將在 5.2 節詳述。圖 10 為第一個步驟的選擇視窗，反白的部份為挑選的關鍵詞串列；依據此關鍵詞串列產生的預設合成文章則如圖 11 所示；而圖 12 則顯示圖 11 中可填於"下課"及"通常"間的其他候選填充字串，由於系統預設所選的填充字串未必適當，故系統提供可讓使用者做替換的其他選項。

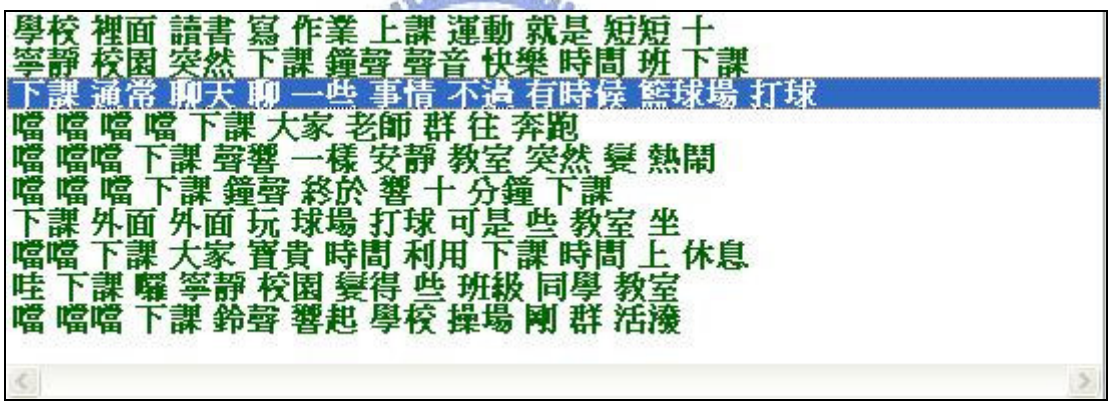

圖 10 第一個步驟 － 候選關鍵詞串列選擇

圖 11 第二個步驟 － 預設合成文章顯示

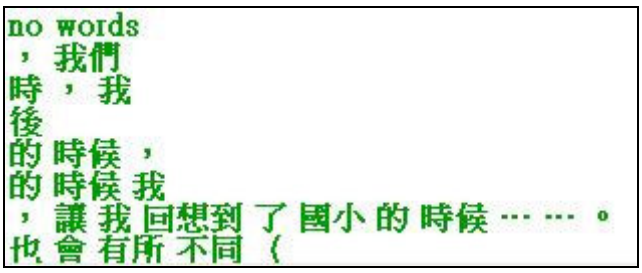

圖 12 "下課"及"通常"的候選填充字串

第五章、系統比較與討論

本章將對作文寫作輔助系統的原型系統以及進一步改良後的合成系統進行比較與分析，觀察改良後的系統是否確實改善原本系統的缺點，並且保留優點。

5.1 系統改良成效

候選關鍵詞串列存在的目的就是要讓使用者大致瞭解合成文章將會包含哪些概念，這個機制讓系統缺乏主題導向的問題確實獲得相當程度的解決，雖然目前所提供的方法稍微缺乏彈性，但至少使用者不會對文章的走向一無所知。另外，候選填充字串集的元素個數也較候選單元串列集來得少，加上預設合成文章的機制，使用者只需修改文章較不通順的地方，不必再逐一檢查所有的填充字串。而填充字串集元素個數較少的原因很顯然是因為同時被左右兩邊的關鍵詞限制住，相對的候選單元串列僅受前一單元最後一個詞制約，在雙重制約的情況下使得填充字串集雖然較小但與前後文的契合度卻更加可靠。系統經過改良後，使用者走入死胡同的情況也有改善，這似乎也可歸功於填充字串雙重制約的特性，但候選填充字串集所有選項都不合適的狀況還是存在。在系統的原始構想中，我們盡量不讓使用者自行輸入任何東西，而以提供選項的方式讓使用者作選擇，這是考量到一旦讓使用者輸入就必須同時提供一個機制判斷輸入是否適當並且與前後文契　合，但由於作文語料庫內的文章數不足，因此不論是單元串列集或是填充字串集都無法完全避開所有選項皆不適合的困境，因此在合成系統中，我們還是提供一個讓使用者自行輸入填充字串的功能，以便在上述的情況發生時依舊可以維持合成文章的通順度，同時也能讓文章更為靈活多變。

5.2 系統介面與操作

在整個系統實現與改良的過程中，介面的設計是很重要的一部分，合成系統的程式介面分成主視窗及關鍵詞串列選擇視窗，主視窗的控制元件說明如下：

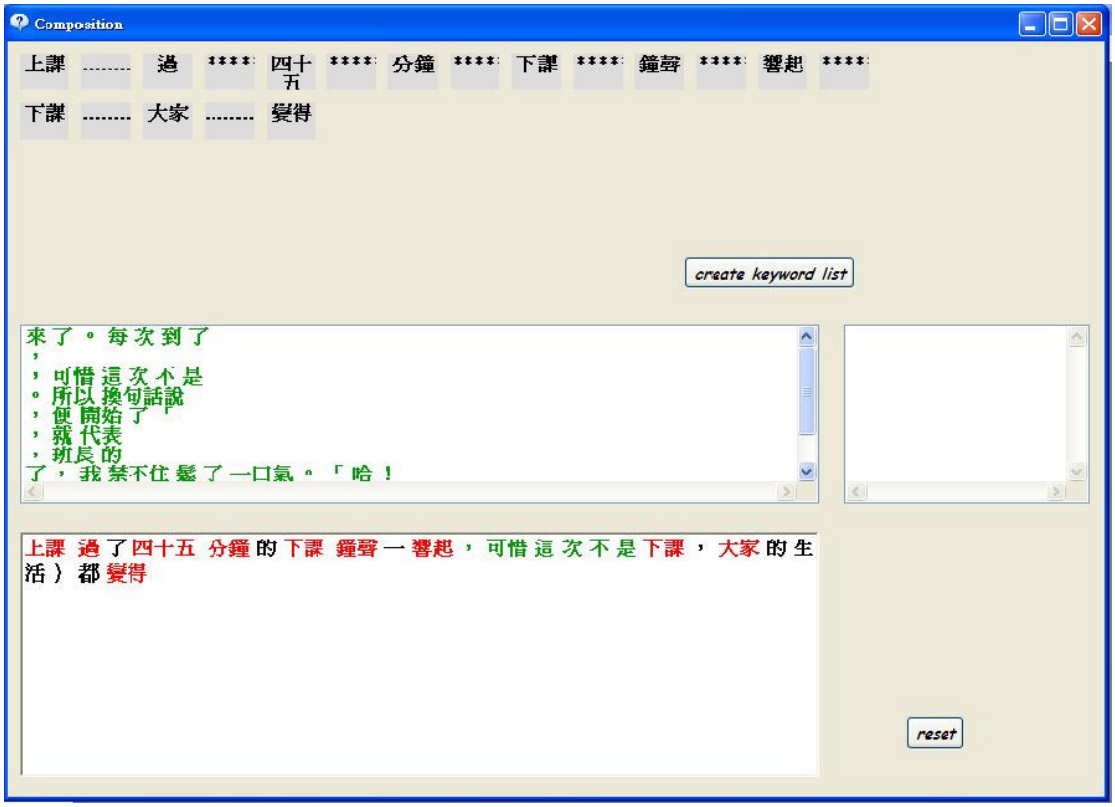

圖 13 程式主視窗介面操作。

在程式視窗上按下"create keyword list"後會跳出關鍵詞串列選擇視窗，選好串列後會回到主視窗，此時視窗畫面如圖 13 所顯示，在視窗最上方顯示關鍵詞串列和填充字串標籤，最下方的文字方塊顯示預設合成文章，我們以不同顏色的字體來顯示關鍵詞和填充字串，字串標籤一經點選文字顯示會從"....."變為"****"同時中間的 listbox 會顯示該字串標籤所對應的填充字串集，文字方塊中對應部份的字體也會變成與周遭不同的顏色，右方的 listbox 則是用來顯示被選填充字串如位置分數、語意關聯分數的各項參數。

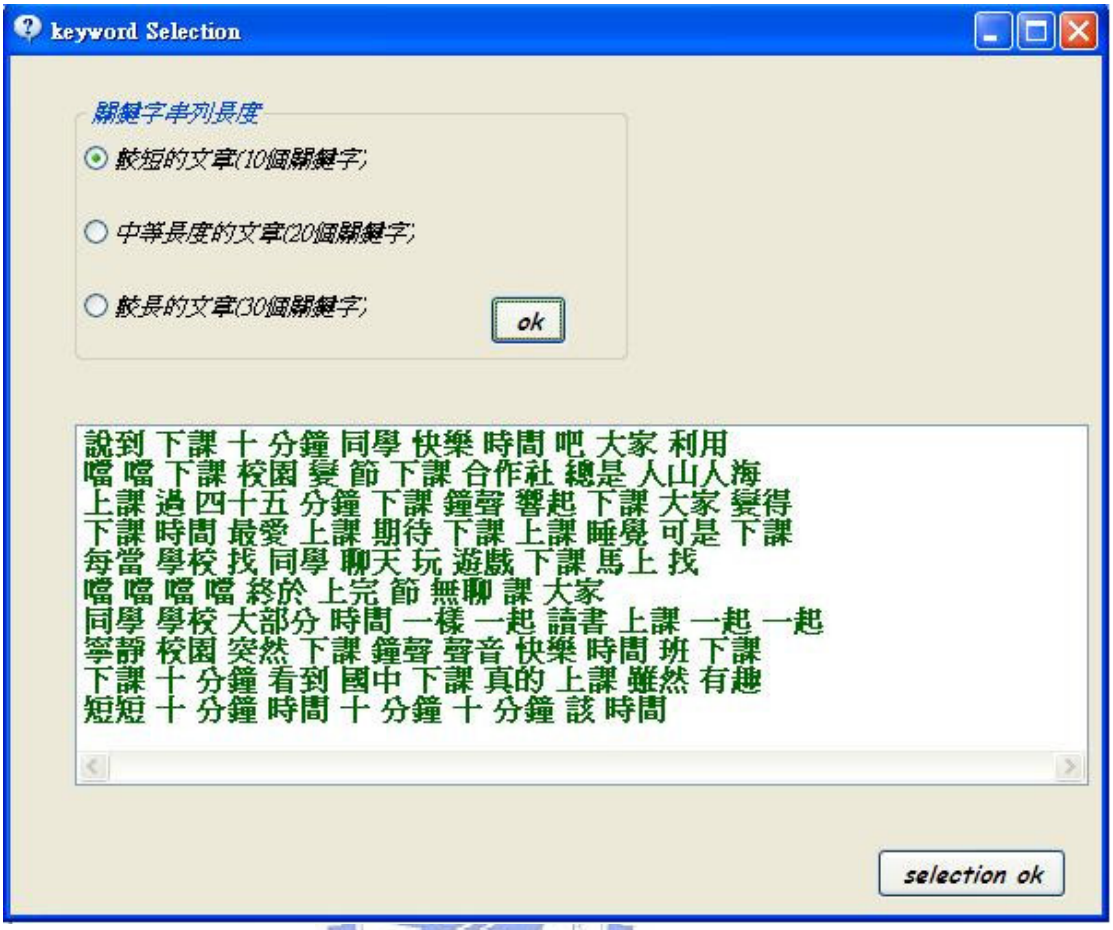

圖 14 關鍵詞串列選擇視窗介面操作

而關鍵詞串列選擇視窗如圖 14 所示，radio box 內有短、中、長三個選項供使用者選擇關鍵詞串列長度，按下 ok 鍵便會在下方 listbox 中顯示 10 組所選長度的串列，選好串列後按下"selection ok"就會進行處理並回到主視窗。

第六章、未來工作與展望

本系統經過實驗、檢討、修正後已具有雛型，但許多功能仍可再加強，並且加入更多功能以使系統更活潑、實用。我們將系統的改進與加強分成兩部份討論：現有基礎功能改進和附加功能的增加。

6.1 基礎功能改進

關鍵詞串列修改

目前關鍵詞串列的產生方式是隨機從分離好的作文語料庫中選取任一串列並且擷取前 n 個詞，這樣產生出來的串列未必每一個關鍵詞都能讓使用者滿意。可以讓使用者對已經選好的串列作修改，也就是對串列中的關鍵詞作替換、刪除甚至加入其他關鍵詞等動作直到使用者對整個串列感到滿意為止。但這部份同時必須考慮到串列經使用者變更後是否仍能從作文語料庫中找到相對應的填充串，未經修改的串列至少有其對應的填充字串可利用，一經修改此部份便無法保證。

以包含關鍵詞用語代替關鍵詞

關鍵詞串列代表合成文章的骨幹，目的是讓使用者瞭解合成文章的主要內容，而單憑關鍵詞串列有時各關鍵詞的關聯性並不是那麼明顯，如串列：上課、久、有時候、打鐘、下課、十、分鐘……，其中上課、久、有時候三個詞的關聯性並不明確，若改成關鍵用語串列：上課 太 久、有時候 會、下課 十 分鐘……概念會更清晰，更有助於使用者對文章內容的掌握。

文章長度控制

如前文所述，關鍵詞串列的長度愈長，合成文章的字數愈多，但實際操作系統後發現，即使選擇中等或較長的串列，所產生的文章字數仍感不足，據觀察，這是由於語料庫中作文關鍵詞密度大所致。有可能一句話中每個詞都是關鍵詞，一個 20 個關鍵詞的串列只是三、四句話的濃縮，即使替換填充字串字數也難以超過 200 字，改變關鍵詞串列的萃取方式或許能改善此種狀況，像是在原文中兩個相鄰的關鍵詞只取其中一個加入關鍵詞串列或是限定關鍵詞串列中任兩個關鍵詞至少相隔幾個詞。

6.2 附加功能增加

替換用語和同義詞

我們知道同一句話有不同的表達方式；我們也常用不同的詞彙來表達同一個概念，如"每當下課時"和"一到下課"在作文語料庫中便經常交替使用；"鈴聲"和"鐘聲"是一樣的概念，一篇好的作文不該一直重複使用某個詞或是用語，提供可替換用語和同義詞替換可使系統的彈性更大，使用者更能隨心所欲地創造出令自己滿意的文章。

其他進階功能

其他更進階的功能包括從平衡語料庫或是其他語料庫中尋找填充字串，這必須考慮到合成文章離題的可能性，另外加強合成文章的修辭也是可嘗試的進階功能，這部份或許可從較常使用的排比或譬喻法著手。此外，目前為止，合成的文章常會有語法怪異的情況發生，通常是發生在某個關鍵詞的前後文，雖然候選填充字串集的存在就是要讓使用者可自行修正此一情況，但創造出一個語法檢查機制減少預設文章怪異語法的出現可使系統更為強健，關於這方面最直接的想法是利用詞性標記的 n-gram 來給每個文字片段一個分數。

參考文獻

[1] Regina Barzilay & Mirella Lapata, Collective Content Selection for Concept-To-Text Generation, Proceedings of the conference on Human Language Technology and Empirical Methods.(2003)

[2] Besag, On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48:259–302.(1986)

[3] Boykov, O. Veksler, R. Zabih., Fast approximate energy minimization via graph cuts. In ICCV, 377–384.(1999).

[4] Kiyotaka Uchimoto & Satoshi Sekine & Hitoshi Isahara, Text Generation from Keywords.(2002)

[5] 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/