國 立 交 通 大 學

資訊科學與工程研究所

碩 士 論 文

部落格文章寫作輔助系統

A Computer-Aided System for Blog Article Writing

研 究 生：紀孝承

指導教授：李嘉晃 教授

中 華 民 國 九 十 九 年 六 月

部 落 格 文 章 寫 作 輔 助 系 統

A Computer-Aided System for Blog Article Writing

研 究 生：紀孝承　　Student：Hsiao-Cheng Chi

指導教授：李嘉晃　　Advisor：Chia-Hoang Lee

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

Jun 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

部 落 格 文 章 寫 作 輔 助 系 統

學生：紀孝承　　指導教授：李嘉晃 教授

國立交通大學 資訊科學與工程研究所碩士班

摘要

由於網路資訊的發達，越來越多的人們習慣於部落格上發表各類文章，分享生活上的點點滴滴，其中使用者的年齡層範圍，不僅涵蓋國中與國小的學生，甚至包括部份的銀髮族。於撰寫部落格的過程中，有時候靈感源源不絕，文章揮灑自如，卻也不乏腸枯思竭的狀況發生。本篇論文結合人機互動與自然語言兩門領域，提出並實作一套部落格文章輔助寫作系統，目的是幫助使用者在撰寫部落格文章的過程中，若發生不知從何構思詞句下筆寫作時，本系統能夠適時適度的提供文章寫作的概念和方向，激發使用者創作的泉源。

由於部落格文章的主題相當廣泛，為提供使用者寫作的相關提示，我們使用網路上大量的部落格文本作為系統資料集，使參考提示的廣度足夠完善。另外，我們亦提供人性化的操作介面，方便使用者撰寫文章。本系統以手寫板作為輸入介面，提供手寫辨識的功能，並給予常用詞彙的建議選項。系統會自動分析使用者撰寫的句子，並萃取出句中的重點詞彙串列，組成查詢字串搜尋網路上相關的範例句子，最後產生文章接續可以撰寫的主題方向。我們初步的實驗結果顯示使用者在操作本系統時，平均節省約 20%的寫作時間，並認為此套系統確實可以幫助人們於寫作時靈感的激發與詞彙的學習。

A Computer-Aided System for Blog Article Writing

Student：Hsiao-Cheng Chi　　Advisor：Prof. Chia-Hoang Lee

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

Abstract

As the popularity of Internet, more and more people get used to sharing the episodes of their lives on their own blogs. During the course of writing, people may stop at some points and spend a lot of time thinking the next sentence. In addition, people tend to use some text segments that have been appeared in other articles. Based on the above observations, we propose and implement a computer-aided system for blog article writing. The system will assist users to compose articles, especially when they struggle to find something to write.

Currently, the Web can be regarded as a big database and it contains a variety of blog articles. We collected a huge amount of blog articles and these articles will be employed as system corpus. The system adopts a handwriting pad as input interface. When the user is writing, the system will automatically analyze current input sentence and extract important words to generate a keyword list. Then, it will use the keyword list to compose a query string to obtain blog artilces from the Internet. Finally, our system will give some advices about sentences or outlines that users can depict in the next sentence. Our experimental results show the system can assist people to compose blog articles efficiently and they can save about 20% of time during the course of writing.

誌謝

首先，感謝指導教授李嘉晃老師對我的悉心指導，才能有今日的成果。老師就像我的良師益友，時而嚴厲，時而慈祥，不論是研究討論或課堂授課時，所教導我的專業知識和處世道理，都著實讓我獲益良多。這些過程與經驗，都將成為我一生受用無窮的寶庫。

同時，我亦感謝這兩年來陪伴在我身邊的實驗室同學們、學長以及學弟。尤其是我的同學們，佑州、喻安、瑞敏，總是不斷的鼓勵我，對我的幫助更是多不勝數。兩年的時間，雖然不是很長，但是曾經有過的歡笑淚水，這些回憶會一輩子永存在我的心中。

最後，我要感謝我的家人，感謝你們對我的愛護和包容。謝謝你們在背後默默的支持，使我能夠順利的完成碩士學位。

心中有太多的感謝不知道如何表達，在此僅以本篇論文表示我對你們最誠摯的感謝，並祝福你們身體健康、萬事如意，謝謝。

紀孝承　謹誌

資訊科學與工程研究所

智慧型系統實驗室

中華民國九十九年七月

目錄

圖目錄

表目錄

第一章、緒論

1.1 研究動機

近年來，由於網路資訊的日益普及，Web 2.0 的蓬勃發展，使得人們開始習慣於網路上發表個人文章，如：樂多 網誌、痞客邦、無名小站、天空部落格…等。Nardi 等人提出詳細的實驗調查[1]，說明人們於網路上發表部落格的主要動機有以下五個原因：

1. 紀錄生活中發生的事件與活動，分享給周圍的親朋好友閱讀。
2. 發表對於有興趣事物的看法，例如：政治立場的評論。
3. 抒發個人心情的管道，將無法說出的心底話以文字方式呈現。
4. 練習如何將想法適當的表達出來，增強自我寫作能力
5. 藉由部落格組成社群論壇，針對某個主題發表各自的文章。

部落格平台提供了一個與全世界交流的機會，每個人均可以透過部落格與網路另一端的使用者分享自己的心情，因此，個人網誌的寫作，已漸漸成為很多人生活中的一部分；雖然部落格網站皆有提供簡易的輸入和排版介面，讓人們可以將個人網誌發表於網路，但卻缺乏許多便利性的輔助，如：常用詞彙的提示、建議的替換詞等。除此之 外，多數的使用者於撰寫的過程中，時常會遇到需要構思下一句的情況，此時閱讀量豐富的人們會快速搜尋腦中的記憶找出合適的句子段落作為參考對象，並加以修飾套用。然而，有些人苦思良久卻仍然遲遲無法下筆，此類的情況更普遍發生在年齡層較低的人身上，其中一項原因是他們所看過的書籍資料和學習過的文學知識較少的緣故。

近年來，作文寫作已經成為國中基測的標準項目，然而學生的寫作能力卻是漸漸的不如往昔。雖然部落格的寫作不若一般作文正式，但學生仍舊可以藉由撰寫網誌的過程，達到訓練文章組織與表達能力的目的。有鑑於此，我們期望能夠在使用者撰寫部落格文章時，利用電腦整理網路上大量的資訊文本，即時的提供輔助資訊，如：文章寫作的方向提示、常用詞彙的建議，藉此減少使用者於構思文章和輸入文字時所花費的時間成本。如此不僅能有效加速整個寫作的過程，同時也讓使用者在瀏覽相關資料的過程中，達到間接學習如何寫作的目的。

1.2 研究目的

我們期望發展一個可以幫助所有部落格寫作者的輔助系統，特別是正處於學習年齡的國中和國小學生，在碰到寫作瓶頸的時候能夠藉由系統的文章提示啟發創作的靈感，進而從操作的經驗中提升寫作的能力和技巧。本系統考量到不常使用電腦鍵盤的銀髮族與小朋友，選擇使用手寫板作為輸入介面，讓使用者感覺宛若書寫於傳統紙本上，可將主要的精力集中在撰寫文章上。

由於部落格文章本身無主題上的設限，文章寫作的範圍相當廣泛，有限的資料集並無法提供足夠的資訊做為輔助提示。因此，系統使用 Google Blog Search Engine[2]搜尋相關的部落格文章，並且以網路上大量的文本作為系統的資料集，解決提供資料的問題。系統將根據使用者所輸入的句字，作為 最基本的資訊來源，並分析字句中的重點所在，以此關鍵詞串列組成查詢字串尋找網路上相關的範例句子，並且自動分門別類，提供不同種類文章的概念方向。利用系統產生的提示句子，我們希望輔助使用者於陷入思考階段時，能夠透過參考資料的啟發，激發其創作的潛力，最後順利完成內心所期望的文章。

1.3 論文架構

第一章：緒論，快速簡單的介紹論文研究的動機，以及探討期望的目的。

第二章：相關研究，說明自然語言領域中關於文本資料處理的研究，以及概述本論文中所使用的技術背景知識。

第三章：系統設計，將整個論文研究的每一部分和環節作完整詳細的解說，包含詞句重點的萃取、資料文本的蒐集、提示句子的產生與分類。

第四章：實驗結果與討論，根據論文研究所提出的理論實作系統，實際尋找人員操作系統並紀錄寫作時間和使用者評價，最後探討實驗數據所呈現的分布結果和原因。

第五章：結論與未來展望，對整個論文研究作總結統整，並且提出結論與未來系統研究的方向。

第二章、相關研究

2.1 文章斷詞與詞性標記

對於電腦而言，文章僅是一堆代碼與符號的組成，實際上電腦並不懂得其中所代表的真正涵意，更遑論分析文章中的重點為何。因此，在自然語言處理的相關應用領域中，文章斷詞與詞性標記的工作是一項重要且不可或缺的部分。舉凡是機器翻譯、語言分析、摘要製作和資訊擷取等研究，都需要先將文章經過斷詞處理，萃取出較小且有意義的單位詞，而後才能利用數學統計的模型，進而找出文章重點詞彙的所在。

由於中文語言的特性，句子中所組成的各詞之間是相互連續直到結束或遇到標點符號為止，並沒有如英語文章般詞與詞之間會透過空格單元分別開來，所以在斷詞處理的比較上，中文遠比英語要來的困難許多。下列將舉一簡單例子說明中文斷詞處理的工作。

欲處理的原始句：今天天氣真好。

可能的斷詞結果：[今][天][天][氣][真][好]。

[今天][天][氣][真][好]。

[今天][天氣][真好]。

[今天天氣][真好]。

…

正確的斷詞結果：[今天][天氣][真][好]。

由上述例子中，可以觀察到中文斷詞的組合千變萬化，若斷詞的結果造成錯誤，將使得語意上的解釋迥然不同，並且嚴重的影響後續研究的成果。因此，我們將藉由使用中央資訊科學研究所詞庫小組中文斷詞系統[3]，完成斷詞與詞性標記的工作，經過分析統計其斷詞結果的正確率可高達95%。

2.2 人機介面

工程師在開發軟體的同時，時常會忽略使用者介面設計的重要性。其中人機介面的核心概念即為Usability，簡而言之，就是探討使用者運用系統提供的功能達到目的的難易程度。一個堪稱為好的系統，除了能提供完善的服務外，人機介面的設計同樣需要經過謹慎妥當的考量。因此，於本篇研究中，使用者的操作介面亦是我們探討的重點之一。

自古以來，人們就習慣於使用筆書寫文章，若系統能提供以手寫筆為輸入媒介，使用者便可以最熟悉的方式來完成文章的寫作，簡單快速的輸入文字。因此，平日不熟悉使用電腦鍵盤的族群，就不需花費大量時間於尋找文字按鍵的位置，可以專注於文章的創作；對於長期使用鍵盤為輸入的人們，亦是一個可以練習文字書寫的好機會。

目前市面上最為廣泛使用的手寫輸入裝置為平板電腦[4][5]，使用者可以透過觸控筆來進行系統的操作。原理共分為三種：電阻式、電容式和電磁式，其中電磁式是現今手寫裝置主要的運用理論，方法為電腦中的數位板會於螢幕的表面產生微弱的磁場，而這種磁場僅會對觸控筆中的裝置發生作用。所以，只有當觸控筆於螢幕表面上移動或點擊時，數位板才會進行快速的定位，並且通知作業系統完成滑鼠相對應的動作。

近幾年，由任天堂株式會社所開放的電玩手把，Wii Remote，利用紅外線定位與藍芽傳輸技術，當玩家握住手把操作時，使用者的動作將會同步對應到遊戲中角色的動作，其主機暢銷成功的原因即是充分發揮人機互動的中心理念。然而，考慮該手把背後欲傳達的互動方式，若將手把對應成筆來寫字，亦不失為一種低成本的手寫裝置。

由於Wii Remote所提供的硬體資源與易取得性，網路上有許多開發教學的文章[6]和應用於電腦互動的實際案例，如：將手把模擬成電腦滑鼠或簡報筆使用。實際上使用者所握的手把是紅外線接受器，當玩家揮舞動作時，手把從前方紅外線發射器所偵測到的紅外線光源訊號就會改變位置，此時手把再透過藍芽傳輸將資訊傳達給後方的遊戲主機或電腦處理，並產生相對應的畫面動作。

Johnny Chung Lee[7]針對手把的互動原理提出新穎的思維和改良方法，並且成功的以電子白板系統展示說明改變後的好處。如圖2-1所示，原始的互動模式為使用者握著手把揮舞動作，接收紅外線訊號，但長時間的握住手把懸空，容易令人感覺到疲勞。如果手中握的是體積小且較輕的筆，使用者便可以長時間且舒適的操作系統。因此，Johnny Chung Lee反其道而行，將發射器與接收器兩者的動作互換過來，並且將紅外線發射器製作成筆的外型方便操作，其實驗結果於網路上有大量的點閱率，甚至白板系統也被運用在學校課堂教學時使用。

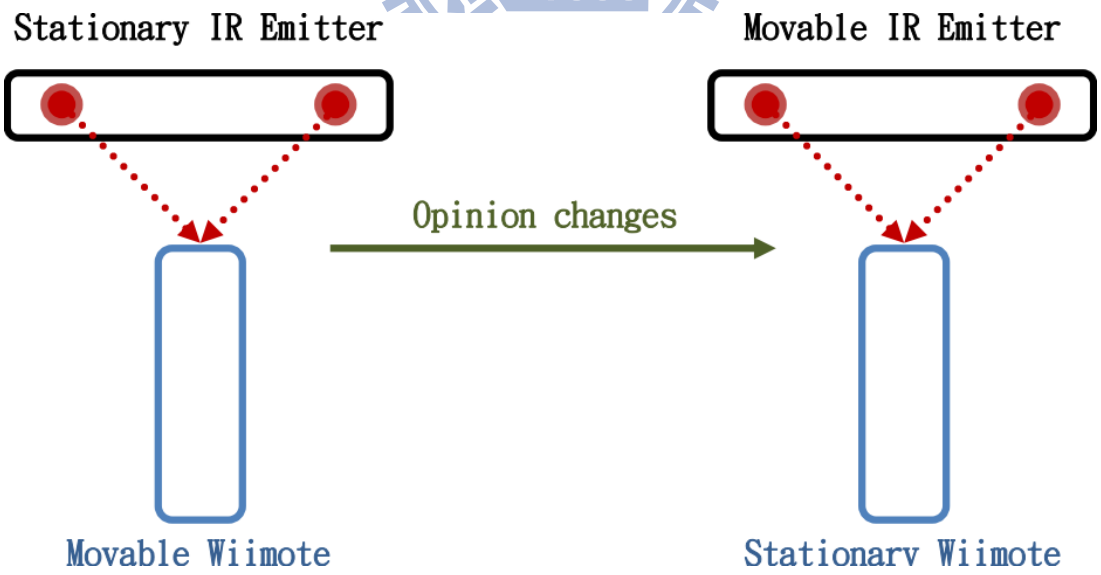

圖2-1：Johnny Chung Lee的設計思維。

在文章寫作中，手寫文字是一項繁瑣卻無可避免的過程，如何能減少中間所耗費的時間成本，可以藉由許多不同的手段達成。最簡單的方式即是提供手寫文字辨識的功能，讓使用者不需要完整的輸入文字的所有筆畫。更為有效的方法則是讓系統進一步的預測使用者接續欲輸入的文字，並且提供良好的介面讓使用者能夠容易且快速的選取，Kurihara等人提出Speech Pen[8]，即是以上述目的為出發點的相關研究；以演講者的報告內容、電腦上的手寫文字作為基本資料，經過文字和語音辨識的分析，系統會即時將有用的資訊建立至資料庫中，當演講者在電腦上撰寫筆記和重點時，系統經過演算法的判斷處理尋找使用者可能要寫下的句子，並提供手寫提示，節省使用者單純手寫文字所花費的時間，如下圖2-2所示，當使用者僅寫下字串「Rec」時，系統經過即時判斷辨別出是欲寫的字為「Recognition」或「Recommended」，以及後續分別可能使用的詞彙，「technology」、「logic」等詞，方便使用者快速選取。

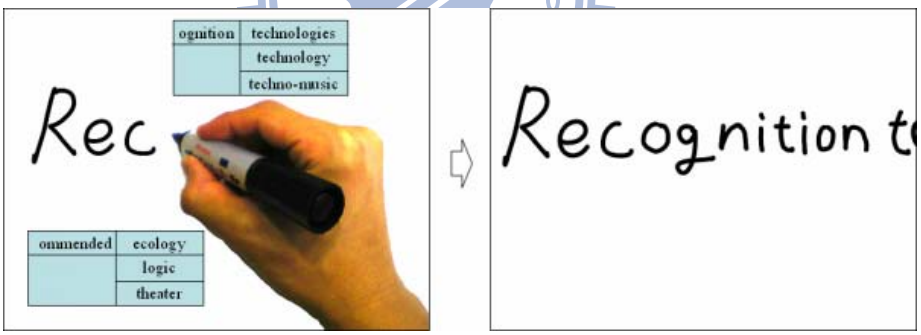

圖2-2：手寫辨識輔助提示。

2.3 文章輔助寫作系統

各式各樣的寫作輔助的工具陸續問世，針對不同的主題或領域，所提供的服務功能或系統的呈現方式也許有所差異，但其本質和目的殊途同歸，皆是為撰寫文章時給予使用者適當的寫作提示，幫助人們觸動寫作的靈感，讓文章的內容不僅兼具深度和廣度，用字遣詞亦多變豐富。

WriteAhead[9]是一個論文摘要輔助寫作的系統，支援撰寫多達 11 種不同領域的英文摘要，系統會依據使用者寫下文章和句子作為查詢依據，選擇適當的字詞、片語、搭配詞和轉折語提供使用者撰寫摘要時多元的參考。另外，對於大部分以英語為第二語言的人，往往在英語寫作上碰到不少問題，如：文章上下語意不連貫、詞義不明、選詞不當、冠詞和時態的使用方法。因此，Ting Liu等人[10]，針對中文語言的使用者開發一套輔助系統，幫助人們學習英文句子正確的寫作方式。微軟研究院所開發的線上英語寫作軟體[1][11]，如圖 2-3 展示，不僅包含拼字與文法上的檢查服務，並且即時提供合適的寫作範例。

圖2-3：Microsoft Research ESL Assistant。

[1] http://www.eslassistant.com

目前盛行的部落格網誌，亦有相關的輔助寫作工具。由 Tien-Chi Huang 等人[12]發表的輔助系統，其功能為參照使用者曾經閱讀的文章和撰寫的網誌，進行內容的分析，並且萃取出關鍵字串作為利用 Google Blog Search Engine[2]搜尋資料的依據，最後將取得的相關文章顯示給使用者參考。與本研究最主要的差異在於此輔助系統僅幫助使用者匯整關聯的網誌，並將整篇文章完整的顯示給使用者參考，只提供整個段落的寫作範例。因此，使用者須在自行閱讀文章範例後，才能找出對文章主題寫作有用的句子。而本系統則會於蒐集相關網誌後，對文章做更深層的分析，並萃取出資料中可用的提示句予人參考，如：使用者寫下「今天天氣很好」，則本系統會即時提供類似「我和家人決定一起到郊外踏青」、「藍天白雲好漂亮」、「心情特別好」等寫作方向，而非描述有關「天氣好」的整篇部落格文章。因此，使用者可以減少閱讀相關資料和融會貫通所需花費的大量時間。另外，本系統以手寫辨識為輸入介面，並提供常用詞彙等功能，可以給予使用者在撰寫文章時適當的輔助。

線上智慧型文章生成輔助系統[13]，則是提供另一種輔助寫作的型式；透過事先收集網路上大量的部落格文章，經過程式訓練產生文章生成的模型，而後使用者僅需要選擇文章的主題概念，系統便會自動生成相關的文本提供使用者參考。國外網站亦有類似功能的輔助寫作工具[14]，藉由讓使用者手動填入主題和論點，再以此資訊自動產生相關文章的內容，並提供文章細節修改。Kiyotaka Uchimoto 等[15]的研究論文，想法原理與線上智慧型文章生成輔助系統[13]類似，方法是運用關鍵字為基礎資訊，並利用語料庫對每個關鍵字進行擴展，最後再根據詞彙語意和相依關係組合擴展後的項目成為生成句子。

2.4 Support Vector Machine

支援向量機（Support Vector Machine），由Vapnik等[16]於1995年根據統計學習理論提出的方法，是目前表現較好的一種分類演算法，其概念為事先給予一群分類好的資料集，如圖2-4所示，利用這些已知的資料訓練產生預測模型。爾後，若有尚未分類的資料時，都可以直接使用該模型預測該資料的結果。簡而言之，我們可以把模型想像成是一個黑箱，當任意資料通過模型後都會被對應至符合條件的區域，且作出分類結果，如圖2-5所示。

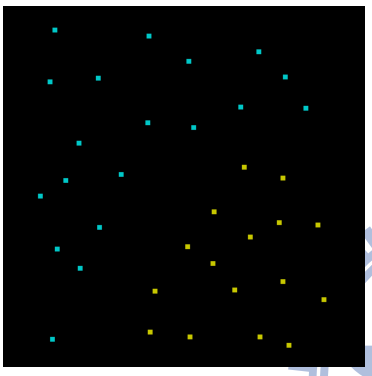

圖2-4：原始資料集的分布圖。

圖2-5：經過SVM分類後的結果。

然而，有時候原始空間中的資料分布，並非線性可分割（Non-linearly Separable）。因此，我們必須將資料映射至較高的維度中，才有機會以超平面將資料分割開來，如下圖 2-6 所示，在原本二維空間中無法線性分割的資料，再將資料點轉換至更高維度的三維空間後，可找到一超平面將資料點線性分割。

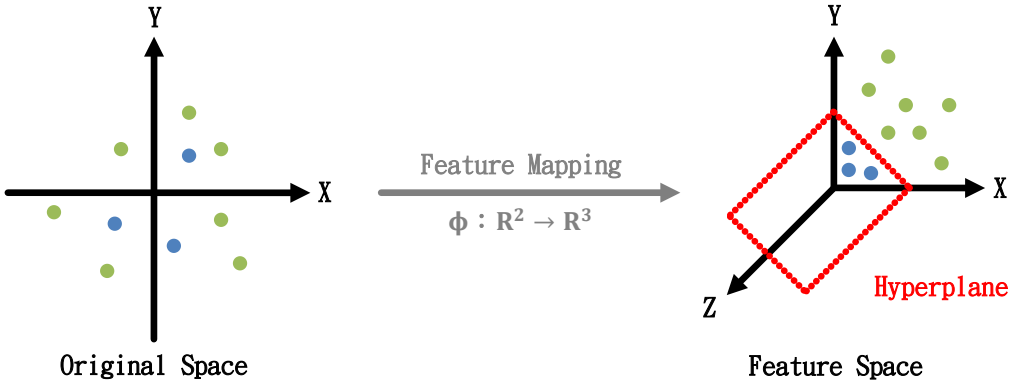

圖2-6：將原始資料轉換至高維度中進行切割。

為達成上述的目的，我們將利用訓練資料來尋找空間中的超平面，透過該超平面將資料順利的切開，如圖 2-7 所示的實線，並且期望該平面將兩側類別的距離分開的越遠越好，讓該超平面可以達到最一般化的效果（Generalization），否則容易使預測結果偏向某一類別，而過於迎合（Overfitting）訓練資料，造成未來使用該模型預測測試資料時，分類的結果不盡理想。下列為支援向量機的各項基本定義。

訓練資料集：$D = \{(X_i, Y_i) \mid X_i \in R^d, Y_i \in \{+1, -1\}\}$，$i = 1, \cdots, n$

X_i：第 i 個資料的特徵屬性，表示為 d 維度的向量。

Y_i：第 i 個資料的類別，於此表示為兩種類別的其中一種，+1 或−1。

分隔的超平面表示式：$w \bullet x - b = 0$

w：代表為平面的法向量（Normal Vector），$w \in R^d$。

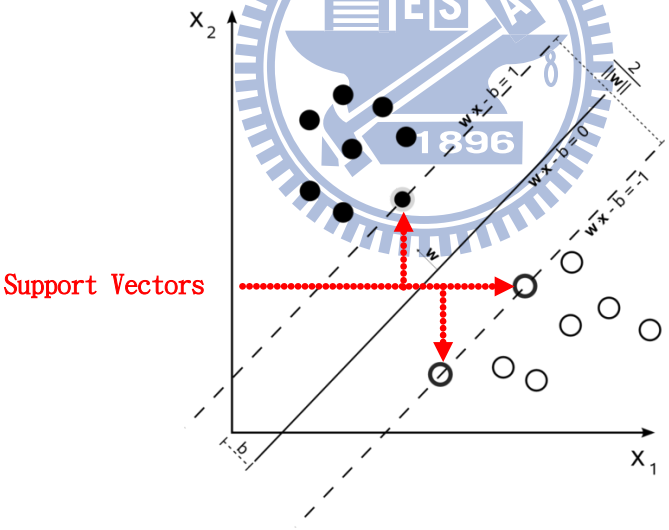

圖2-7：超平面示意圖。

如圖 2-7 所示，假設 $P: w \bullet x - b = 0$ 為一可將兩種類別資料分隔之超平面，藉由適當的重新調整（Rescaling），我們可以定義兩個平行於 P 的輔助超平面，並且這兩個輔助超平面會分別通過兩種類別距離 P 最近的所有資料點（Support Vectors），圖 2-6 中所示之虛線，其定義如下：

$$w \bullet x - b = +1 \quad (2.1)$$

$$w \bullet x - b = -1 \quad (2.2)$$

考量到分類器的一般化情況，支援向量機的目標為使得兩個輔助超平面的距離越大越好，利用幾何學的原理，發現兩個輔助的超平面，(2.1)和(2.2)，中間的距離為 $\frac{2}{\|w\|}$。因此，欲讓兩者間距有最大值，必須使得$\|w\|$的數值越小越好。除此之外，為避免資料落於兩平面之間，將加入以下限制式：

$$w \bullet x_i - b \geq +1，\text{if } y_i = +1，i = 1,\cdots,n \quad (2.3)$$

$$w \bullet x_i - b \leq -1，\text{if } y_i = -1，i = 1,\cdots,n \quad (2.4)$$

組合限制式(2.3)和(2.4)：

$$y_i \times (w \bullet x_i - b) \geq 1，i = 1 \cdots n$$

整理上述結果，最佳化問題可以由下列數學式表達解釋之：

Minimize $\quad \frac{1}{2} w^T w$

Subject to $\quad y_i \times (w \bullet x_i - b) \geq 1，i = 1 \cdots n$

欲解求極值的問題，可將式子轉換成 Lagrange Multipliers Function：

ϕ：Feature Mapping，$x_i \rightarrow \phi(x_i)$

$$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i [y_i (w^T \phi(x_i) + b) - 1] \quad (2.5)$$

Subject to $\quad \alpha_i \geq 0，i = 1 \cdots n$

分別對 w 與 b 作偏微分：

$$\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i) \tag{2.6}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.7}$$

將偏微分後取得結果代回式子(2.5)，可轉換為求最大值的問題，並解出 α ：

k(Kernel Function)

$$W(\alpha) = L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i [y_i (w^T \phi(x_i) + b) - 1]$$

$$= \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i y_i \, w^T \phi(x_i) + \sum_{i=1}^{n} \alpha_i$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, k(x_i, x_j)$$

欲將資料點進行升維的處理，必須尋找合適的映射函式，使得所有於低維空間中交錯難分的資料點，在更高維度的空間中能夠區隔開來。在數學泛函分析中，學者發現利用 Kernel Function 可以解決尋找適當映射函式計算成本過大的問題，此類函式僅需處理原始空間中的函數，其值便會是高維空間中的內積，實際範例如下列所示：$x = (x_1, x_2) \rightarrow \phi(x) = (z_1, z_2, z_3) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$

$$\langle \phi(x), \phi(x') \rangle$$

$$= \langle \phi(x_1, x_2), \phi(x_1', x_2') \rangle$$

$$= \langle (z_1, z_2, z_3), (z_1', z_2', z_3') \rangle$$

$$= \langle (x_1^2, \sqrt{2} x_1 x_2, x_2^2), ({x_1'}^2, \sqrt{2} x_1' x_2', {x_2'}^2) \rangle$$

$$= x_1^2 {x_1'}^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 {x_2'}^2$$

$$= (x_1 x_1' + x_2 x_2')^2$$

$$= (\langle x, x' \rangle)^2$$

$$= k(x, x')$$

將α代回式子(2.6)，即可取得w。然而，於式子(2.7)中並沒有存在α與b之間的關係式，故利用Karush-Kuhn-Tucker Theorem，計算出b值，其詳細的條件式如下列所示：

$$y_i(w^T\phi(x_i)+b)-1\geq 0，i=1\cdots n \tag{2.8}$$

$$\alpha_i\geq 0，i=1\cdots n$$

$$\alpha_i[y_i(w^T\phi(x_i)+b)-1]=0，i=1\cdots n \tag{2.9}$$

由式子(2.9)可知，若式子(2.8)不等於零時，則α_i必為零。反之，如果式子(2.8)等於零時，α_i會大於零，此些資料點即為Support Vectors。整理結果後，可以導出下列關係式：

$$\begin{cases}\alpha_i[y_i(w^T\phi(x_i)+b)-1]=0\\ \alpha_i>0\end{cases}\rightarrow\ y_i(w^T\phi(x_i)+b)-1=0$$

$$\rightarrow\ y_i(w^T\phi(x_i)+b)=1$$

$$\rightarrow\ w^T\phi(x_i)+b=\pm 1=y_i$$

$$\rightarrow b=y_i-w^T\phi(x_i)$$

最後，預測模型如下列所示：

$$w^T\phi(x_{new})+b=\sum_{i=1}^{n}\alpha_i y_i\,k(x_i,x_{new})+b$$

$$\rightarrow y_{new}=\text{sign}(\textstyle\sum_{i=1}^{n}\alpha_i y_i\,k(x_i,x_{new})+b)，y_{new}=\begin{cases}+1，x_{new}\in\text{First Class}\\ -1，x_{new}\in\text{Second Class}\end{cases}$$

當新的資料點要進行分類時，經過此模型判斷，我們就可以快速的取得分類的結果。目前支援向量機的現成工具方面，LIBSVM[17]為目前最熱門及方便的支援向量機工具軟體之一，本研究亦採用此軟體來完成主題分類的預測模型，用以判斷文章所屬的類別。

第三章、系統設計

3.1 系統架構

本研究的主要目的，是於人們在創作文章遇到瓶頸時，提供適當的提示服務幫助使用者思考寫作的方向，同時減少整體寫作所花費的時間。當使用者正煩惱文章後續要撰寫的內容時，若系統能夠顯示參考提示，便可以激發使用者思考的潛力，創造出豐富的文章內容。舉例說明如下所示:

輸入句子：可惜天公不作美，天氣陰陰的，並非期待的暖陽。

參考提示：1. 雲層厚的跟甚麼一樣

2. 還算是可以接受

3. 前兩天的陽光普照到今天變成陰雨綿綿

圖 3-1 為系統對應的流程圖，依照功能與目的可以劃分為下列兩個部份：

1. 輸入介面模組（Input Model）

   主要負責處理使用者的輸入資料，包含手寫辨識（Handwriting Recognizer）和擷取資料庫（Term Database）提供常用詞彙列表（Bigram Selection）。

2. 文章提示模組（Suggestion Model）

   此部份將根據輸入資料的類型產生不同的查詢字串（Sentence Query or Clause Query），並至網路搜尋相關資料（Search Google），將資料進行主題分類（Topic Classifier），最後產生文章寫作提示（Show Reference）。

在使用者以手寫輸入筆劃資料時，系統會即時進行文字辨識，並提供可能使用的常用詞彙串列。對於每一筆輸入資料，系統都判斷是否完成子句或是句子。若結果為是，則產生對應的查詢字串，進行資料的蒐集和處理，並產生對應的參考提示；如果皆否，則等待使用者輸入下一筆資料，再進行分析。詳細的介紹和說明，我們將在後續的章節逐一闡述之。

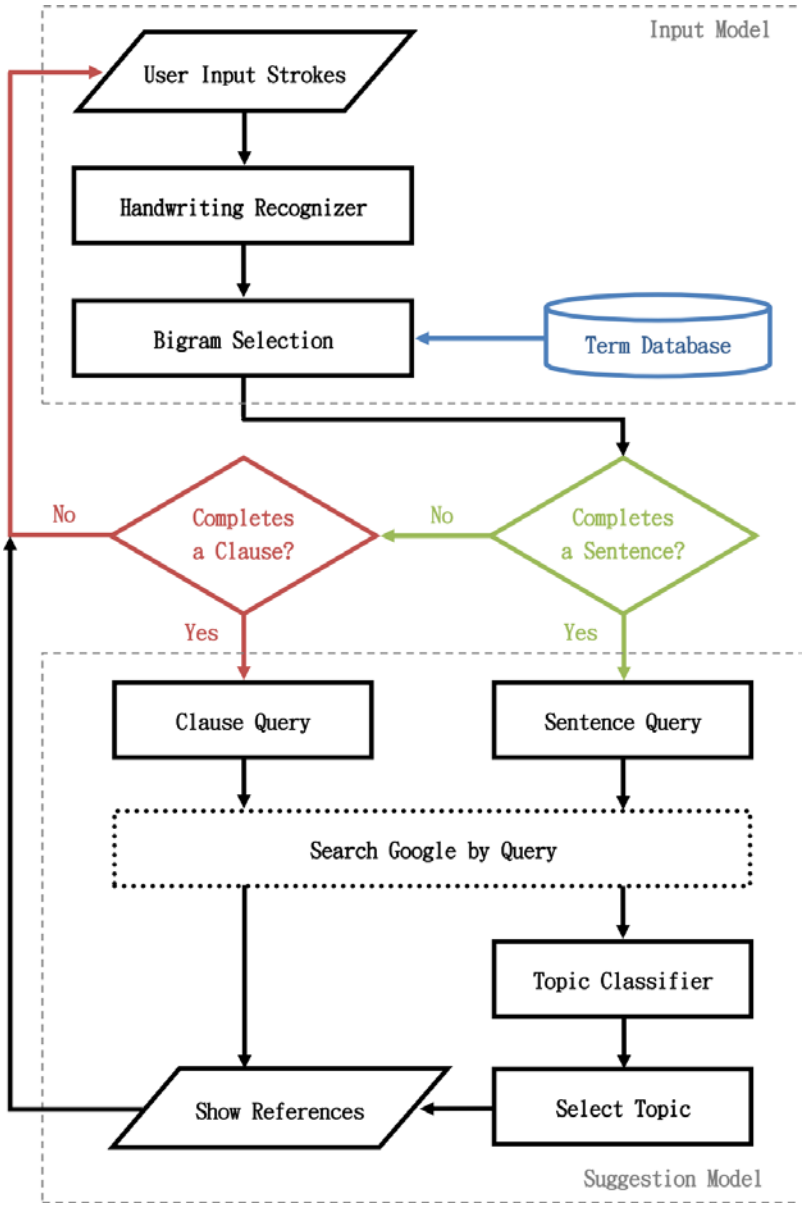

圖3-1：系統流程圖。

3.2 前置處理

為達到提供使用者常用字詞與提示句子的目的，我們分別收集 Libtabe[18] 與新浪部落[19]中的資訊作為建構資料庫的基本數據。Libtabe，為 Linux Xcin 軟體輸入法底下的中文處理函式庫，主要提供軟體開發者處理中文字、詞、句等資料處理功能的整合性開發函式庫。目前的常用詞彙共有約 13 萬筆，其檔案中的每一行為一筆紀錄資料，格式如下：

(Pharse)　(Frequency)　(Notional Phonetic Alphabet)

Pharse：中文詞彙。

Frequency：經統計文章後所得到的詞頻數目。

Notional Phonetic Alphabet：詞彙的注音符號組成。

下列為兩筆範例資料：

地底　100　ㄉㄧˋ ㄉㄧˇ

天氣　1700　ㄊㄧㄢ ㄑㄧˋ

上述例子中，第一筆資料的中文詞彙為「地底」，詞彙的頻率為 100。詞頻大小與詞彙統計上的機率成正比關係；例如，上述第二筆資料「天氣」於一般性的中文文章中，出現的頻率約是第一筆資料的 17 倍。於本系統中，將利用該詞庫所提供的資訊，查詢常用辭彙；我們將根據每筆中文詞彙與詞頻，在資料庫中建立一筆對應資料，詞彙的注音符號序列則捨棄不用。

當使用者寫下文字時，系統將自動以此文字為詞的起始字查詢資料庫，取得相關聯的候選詞，並按照詞頻的高低作排序，意思即為擁有越高詞頻的中文詞彙將會被排序至越前方，提供使用者快速選取。在選取候選詞完畢後，系統會自動更新該詞於資料庫中的詞頻，因此，經過一段時間的操作後，越常被使用者利用的詞彙越容易被排在越前面，亦達到選詞個人化的效果。

在使用者寫完整句後，系統必須提供不同類別的撰寫方向，讓使用者有更多彈性的參考選擇。然而，為降低花費於文章歸類處理的時間，我們利用支援向量機 (Support Vector Machine) 作為分類的演算法；透過事先訓練好類別預測的模型，提供系統即時分類的需求。

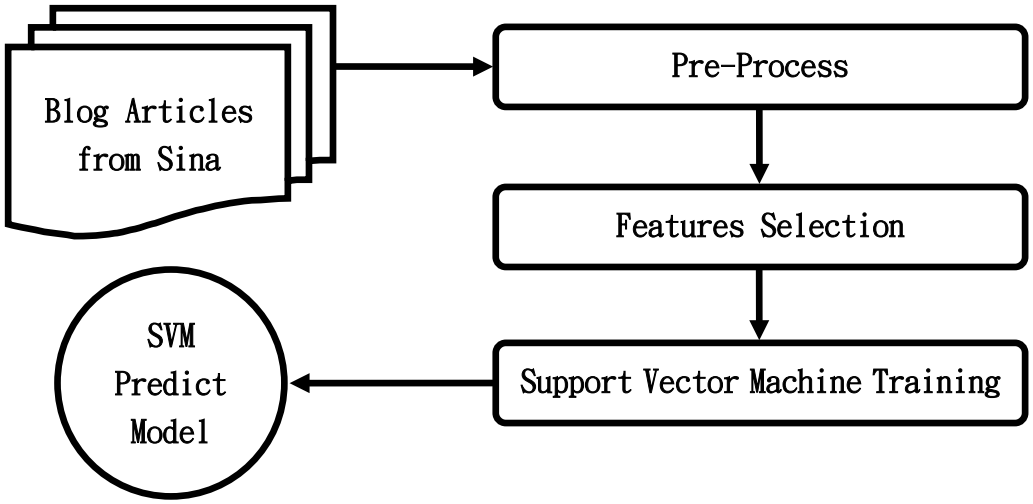

圖3-2：Training SVM Model。

上圖 3-2 為訓練模型的流程，首先，我們透過程式至新浪部落蒐集文章，全部有 2,262 篇涵蓋五個不同類別的文章，分別是運動類、心情類、旅遊類、電影類以及美食類；在前處理的部分（Pre-Process），我們針對每個類別的文章統計出現的詞彙和次數，並於特徵萃取的階段（Features Selection），根據詞頻的高低萃取出代表詞彙串列，再將所有類別的代表詞串列取聯集作為訓練預測模型時所使用的特徵集合，其演算法如下列所示：

Input： Percent = 1.5%

CategorySet = { Sport, Mind, Travel, Movie, Food }

Output： Set of FeatureList for Support Vector Machine Training

Extract-FeatureList（Percent, CategorySet）

1. **foreach** $Category_i \in CategorySet$ **do**
2. Count the frequency of each $Word_i$ in $Category_i$
3. Sort word list by frequency
4. Extract the top Percent frequency words to $KeywordList_i$
5. **end**
6. FeatureList = Union set of each KeywordList

上述參數中，Percent 為限制變數，表示對每類別取前幾百分比高的詞頻當作該類別的代表詞彙串列，根據需求可作數值的微調，在此我們取 1.5%為系統計算時使用的參數。當取得特徵集合後，將每篇文章以特章向量表示之，並且根據類別給予對應的標籤，舉例說明：

$$\text{FeatureList} = \{\text{天氣、美景、感動、旅遊、攝影}\}$$

$$\text{Article} = \{\text{今天天氣真好，非常適合全家出門旅遊。}\}$$

$$\text{Vector} = [1_{\text{天氣}}\ 0_{\text{美景}}\ 0_{\text{感動}}\ 1_{\text{旅遊}}\ 0_{\text{攝影}}]$$

支援向量機的資料格式：

$$\text{Label}_i \quad 1:1_{\text{天氣}} \quad 2:0_{\text{美景}} \quad 3:0_{\text{感動}} \quad 4:1_{\text{旅遊}} \quad 5:0_{\text{攝影}}$$

$$\text{Label} = \{1_{\text{運動類}}, 2_{\text{心情類}}, 3_{\text{旅遊類}}, 4_{\text{電影類}}, 5_{\text{美食類}}\}$$

3.3 系統概述

根據使用者輸入的筆跡資料，系統會利用手寫辨識的模組，判斷可能欲寫的文字結果（圖 3-3.2），並同時提供以該字為首的常用詞串列，以「今」字為首的常用詞有「今天」、「今日」、「今年」等（圖 3-3.3）。使用者曾經寫過的字詞亦會被記載於系統的歷史訊息中，並按照使用的次數作排序，提供使用者快速選取，來節省手寫文字的時間。當系統經判斷發現已完成子句或句子的輸入，即啟動文章提示模組對該句子進行重點分析(圖 3-3.4)。子句與句子間的差異，可以藉由下例解釋說明之：

完整的句子：今天天氣真好，適合出門玩。

其中的子句：1. 今天天氣真好

2. 適合出門玩

由此可知，子句為句子的一部分，完整的句子可由數個子句所構成。當使用者輸入完一個子句或句子，系統會尋找出子句或句子中的關鍵字詞彙串列，作為至 Google Blog Search Engine[2]蒐集資料的依據。如果分析的是子句，系統將給予接續該子句可以描述的相關內容作為寫作參考；若為句子，則提供使用者後續可以撰寫的文章主題或方向。如圖 3-3.5 所示，提示會被顯示於畫面上，使用者不僅可以參考資料做進一步的聯想，亦可以搭配使用提示來組成新的句子，產生更多不同的變化；如圖 3-3.6 所示，使用者分別選取第一條與第二條提示句中部分詞彙組成「天氣好拍照都好美」之結果。

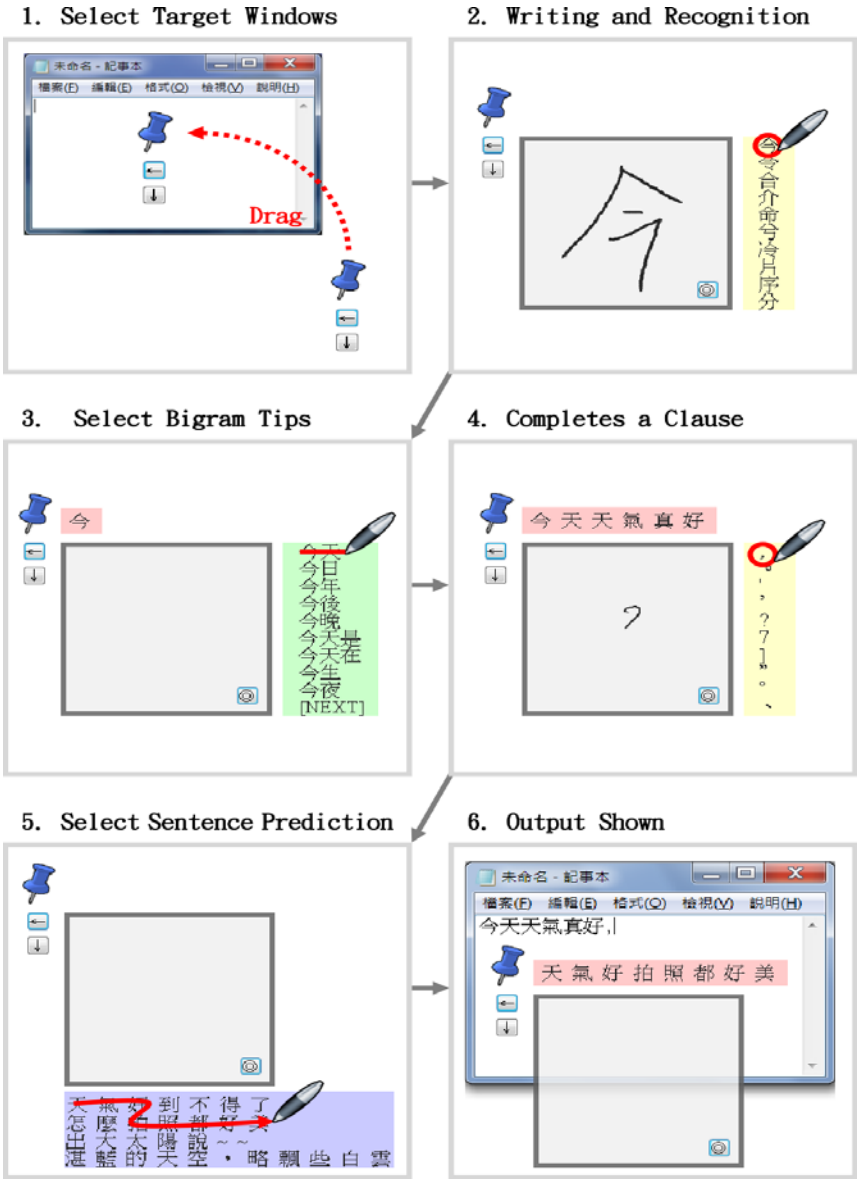

圖3-3：系統簡易展示圖。

3.4　輸入介面

圖 3-4 為系統輸入介面，當使用者於主視窗（灰色視窗）輸入筆跡時，系統會即時進行文字辨識，並且將可能的結果按照相似程度作高低排序，顯示於右方區塊（黃色視窗）方便使用者選取。目前我們的系統中，採用 Microsoft Tablet PC Platform SDK[20]完成手寫辨識的功能。位於主視窗上方的區塊為暫存區（紅色視窗），表示所有經選取的字詞都會先儲存於此，使用者可於暫存區中內進行小幅度的修改，藉由劃過欲丟棄的文字來進行刪除的目的，當完成一個子句之後，系統才將資料輸入至文字編輯器內。

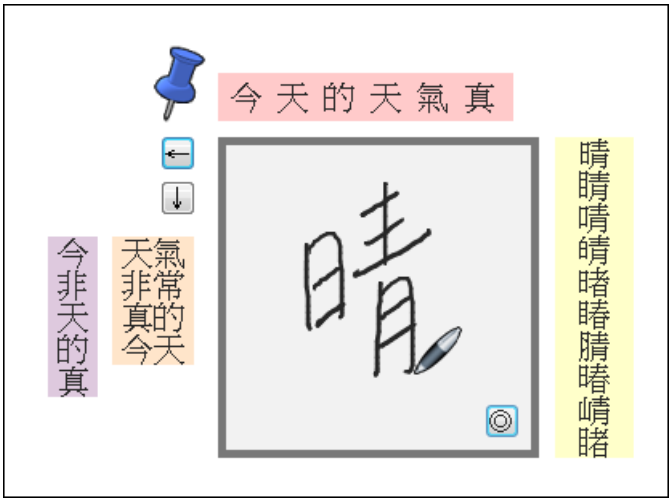

圖3-4：輸入介面展示圖。

在人們於寫作文章時，會因主題內容和用詞習慣，頻繁的使用某些字藻詞彙，為節省反覆輸入相同文字所花費的時間，系統分別提供使用者過去曾經寫過的字（紫色視窗）與詞（橙色視窗），並依照使用頻率排序，只要簡單的點選於該視窗上的字詞，便會自動加入至暫存區內保存。當使用者圈選辨識結果後（圖 3-3.2），系統將透過 Bigram Model 提供相關的常用詞串列，如圖 3-3.3 中綠色視窗所示，其選取的方式同上所述。

3.5 分析子句核心關鍵詞

欲提供使用者下一句撰寫的方向，須依據已輸入的文章內容作為參考基準點，萃取文章中所論述的重點詞彙，並利用分析出來的結果至網路上尋找相關的網誌作為寫作提示。詳細的流程與步驟，如下圖 3-5 所示。每當使用者選取文字辨識的結果時（Select a Character），系統會於判斷使用者輸入的文字，來決定是否已達到完成子句的要求，判定的方式為檢查選取的文字是不是為:句號、逗號、分號、問號以及驚嘆號（One of Target Puncuation），其意義為當使用者寫下上述標點符號的其中一種時，通常代表暫存區中的資料已經包含完整的主詞與動詞，且足夠表達獨自的意義而成為一個子句。

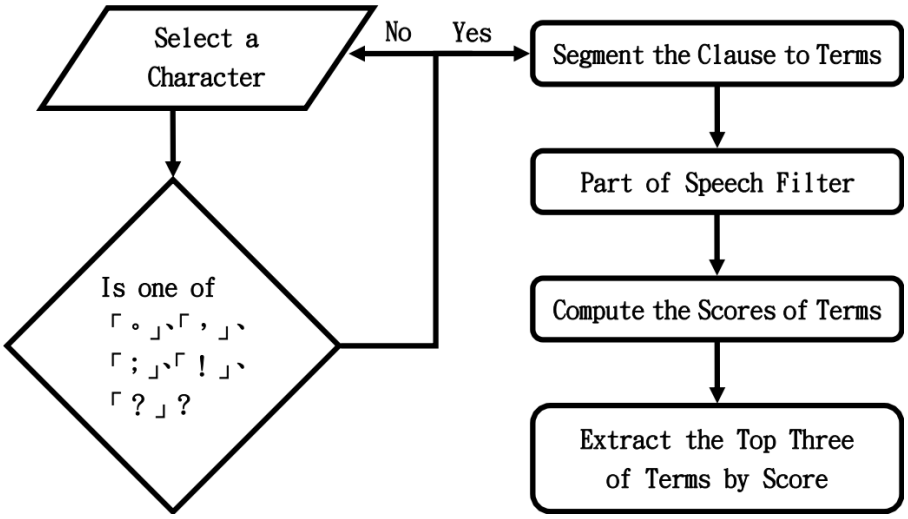

圖3-5：分析子句核心關鍵詞的流程圖。

由於子句是數個字和詞所組成的集合，分析重點詞彙時，必須先將子句拆解成較小的單位。在文章斷詞的部分（Segment the Clause to Terms），我們藉由中央資訊科學研究所詞庫小組中文斷詞系統[3]，完成子句斷詞處理的工作。下列為中文斷詞系統處理之過程與結果，回傳的資訊會將子句切割成最基本的語言單位，並包含每個單位對應的中文詞性。

輸入的子句：今天天氣真好

斷詞的結果：今天(Nd)　天氣(Na)　真(D)　好(VH)

根據中研院平衡語料庫詞類標記集顯示中文詞性的種類共多達 46 種，其中含有許多種類的詞性不會直接影響到句意的表達，如：副詞（D）、語助詞（T）、感嘆詞（I）等。在中文新聞自動摘要系統[21]中，指出名詞與動詞為較能表達意義的詞性，並歸納出其中 10 種詞性作為關鍵詞的詞性，如表 3-1 所示。於過濾詞性的階段（POS Filter），我們將斷詞結果進行詞類的簡化，僅保留表中所列出的詞性種類，減少其他無意義的詞類以免影響分析子句關鍵詞的結果。

表 3-1：保留的詞性列表。

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

對於所有子句中的詞彙，我們都分別計算權重分數（Compute Score），最後保留權重分數最高的三個詞彙為該子句的核心關鍵詞(Extract the Top Three of Terms by Score）。所以，越重要的詞彙得到的分數將會越高，若詞性不存在表中，直接將其權重分數設為零，表示直接忽略該詞彙於子句中的影響力。本系統採用兩種不同面向計算句中其餘詞彙的權重分數，分別為位置分數和權重分數，最後詞彙所得權重即為兩者分數的總合，以下我們將針對兩種分數的計算方式作詳細的說明：

1. 位置分數（Position Score）

考慮中文的語言結構，往往越靠近結尾的詞彙越是表達子句意義的所在。我們以詞彙出現在子句中的位置來決定其基本分數，下列為計算範例。

輸入子句：今天天氣真好

斷詞處理：[今天] [天氣] [真] [好]

分數計算：詞彙於子句中的位置 ÷ 子句中詞彙的數量

[天氣]= 2 ÷ 4 = 0.5

2. 詞性分數（Part of Speech Score）

統計訓練資料集 2,262 篇部落格文章中詞性的使用比例，當作各種詞類的基本權重分數。舉例說明，假若普通名詞（Na）在一般文章中出現的機率是百分之二十五，則系統會自動將子句中所有詞性為普通名詞的詞彙都給予 0.25 的詞性分數。另外，子句的重點詞彙，通常可由動詞和名詞搭配組成，例如：「我和同學去騎單車」，經過人工判斷的結果，並捨去單一高頻字（Stop Word）的項目，最後重點應為「騎」和「單車」。為盡量讓取得的關鍵詞彙能夠涵蓋動詞與名詞，我們在計算詞性分數時，讓重複的詞性依照出現次數做分數折扣。舉例說明：

輸入子句：我和同學去騎單車

詞性標記：我(Nh) 和(Caa) 同學(Na) 去(D) 騎(VC) 單車(Na)

分數計算：$[\text{單車}] = \text{統計普通名詞於一般性文章出現之機率} = 0.25$

$[\text{同學}] = \text{普通名詞再次出現時所得分數須折扣}$

$= 0.25 \times (1 - \text{DiscountRate})$

$0 \leq \text{DiscountRate} \leq 1$

計算權重的演算法如下列所示：

Input： Clause

DiscountRate = 25%

Output： Score of each word in Clause

Compute-Words-Weight（Clause, DiscountRate）

1. Number = the Amount of Terms in Clause
2. CountSet = $\begin{Bmatrix} \text{"Na"}:0, \text{"Nb"}:0, \text{"Nc"}:0, \text{"VH"}:0, \text{"VHC"}:0, \\ \text{"VA"}:0, \text{"VB"}:0, \text{"VC"}:0, \text{"VE"}:0, \text{"VK"}:0 \end{Bmatrix}$
3. **foreach** $\text{Word}_i \in$ Clause **do**
4. Times = CountSet [POS of Word_i]
5. PositionScore = $\text{Word}_i \rightarrow$ Index ÷ Number
6. POSScore = (1 − DiscountRate × Times) × Probability [POS of Word_i]
7. $\text{Word}_i \rightarrow$ Score = PositionScore + POSScore
8. CountSet [POS of Word_i] = Times + 1
9. **end**

在上述參數中，DiscountRate 為折扣變數，表示有關每次詞性重複發生時，分數減少的量值（Penalty）。隨著同詞性出現次數的增加，該詞彙權重中詞性分數的部分會越來越低，亦代表該詞彙在整個子句中的影響力越來越小，根據系統需求作微調，於此我們取 25%作為程式計算時使用的參數。當子句中的所有詞彙都被計算出各自權重後，我們選取擁有最高分數的三個詞彙作為該子句的核心關鍵詞，下一步將以此結果組成查詢字串至網路上收尋相關的網誌文章，提供使用者寫作的題示。

3.6 產生查詢字串

在系統中，我們以逗號作為子句和子句之間的區隔，而以句號、分號、問號或驚嘆號當作句子與句子彼此的分界。當系統發生完成子句或句子兩種情況時，都會進行核心關鍵詞的分析，並且根據子句結束時所使用的標點符號種類，最後組成不同查詢目的的字串。詳細的流程，如下圖 3-6 所示。

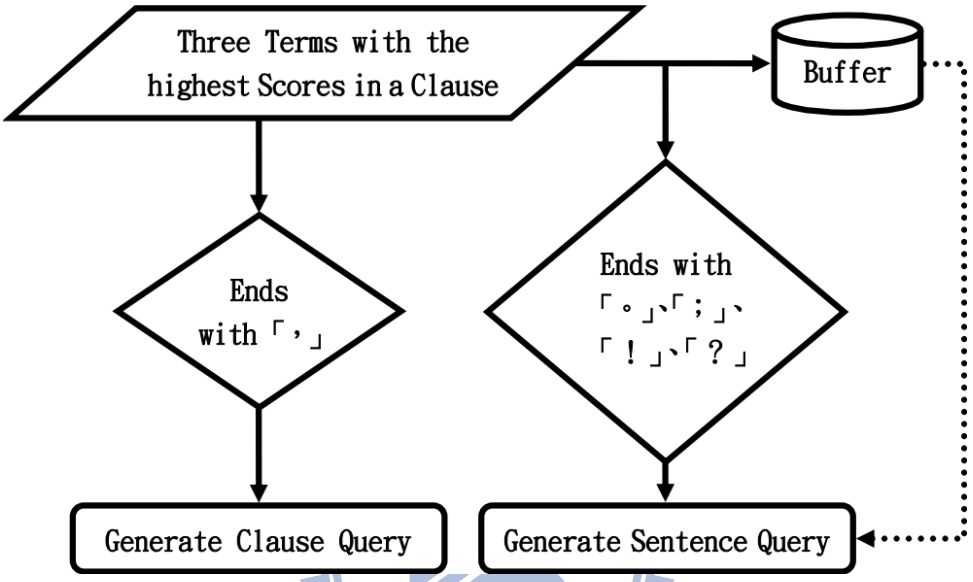

圖3-6：產生查詢字串的流程圖。

下列為判斷的條件以及如何組成不同查詢字串的細部解釋：

1. 子句查詢（Clause Query）

   假如所分析的子句結尾為逗點(Ends with Comma)，代表整個句子尚未結束，在後續子句中，使用者會繼續針對同一個主題作敘述，如：「今天的天氣相當適合踏青，」，表示接下來使用者可能會對郊遊地點與活動過程做進一步的闡述。此時系統的提示會以目前子句為主要的參考對象，並且將取出的核心關鍵詞按照原始的詞彙順序作排列。對於任意兩個核心關鍵詞中若存在其他詞彙，我們以萬用字元「＊」代表之，該符號的意義為可以被取代成為任意字詞。最後的查詢字串會以一對雙引號包起來，目的是限制查詢字串必須出現在同一句中，期望尋找於網路上擁有類似子句結構的部落格文章作為系統提示。圖 3-7 為組成子句查詢字串的範例。

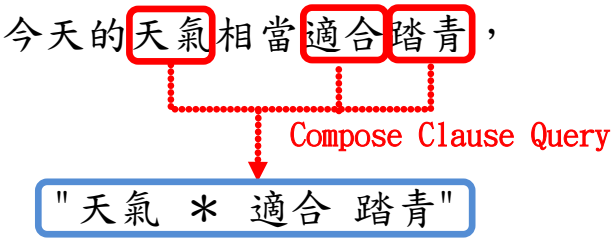

圖3-7：子句查詢字串示意圖。

2. 句子查詢（Sentence Query）

若最後使用的標點符號是句號、分號、問號或驚嘆號其中一種（Ends with Closing Punctuation），則表示目前敘述的事物已經暫時結束，使用者欲在接下來的文章中撰寫新的主題與內容，如：「今天的天氣相當適合踏青，我和家人去陽明山郊遊。」，使用者可能會接續描寫關於陽明山的風景和郊遊的心情。我們考慮完整句子的內文，以找出富含該句子核心關鍵詞的資料，並分析相關文章的句子提供寫作提示。在此查詢字串不會加入雙引號，原因是為取得網路上談論相似主題內容的網誌，僅需找出使用同樣詞彙的文章即可，過多的限制反而容易造成沒有符合的資訊。為快速組成句子查詢，系統在每次分析子句之後，都會將該子句的核心關鍵詞紀錄餘暫存區內(Buffer)。因此，原本系統需要產生與句子關聯的查詢字串時，須對組成該句子的所有子句內容再進行關鍵詞分析，此時便可藉由直接抓出暫存區的紀錄資料以取得整個句子的重點詞彙，減少系統重新分析句子所花費的時間。最後系統會重新清空暫存區內的資料，表示接續紀錄的資料是新的句子。圖 3-8 為組成句子查詢字串的範例。

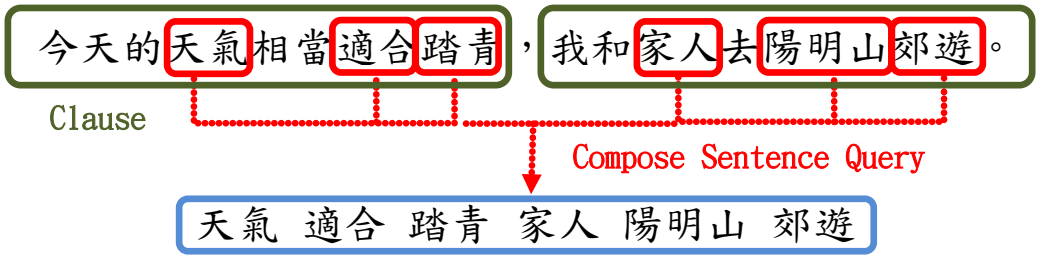

圖3-8：句子查詢字串示意圖。

3.7　萃取與使用提示資料

產生文章提示的流程示意圖，如下圖 3-9 所示。當系統完成分析核心關鍵詞彙的動作，會將結果組成查詢字串（Clause Query or Sentence Query），並至 Google Blog Search Engine[2]尋找符合的部落格文章作為給予使用者的參考。依照查詢字串的不同，所取得的參考資訊亦有所差別。若提供的查詢字串是來自子句的核心關鍵詞，則系統會收集到一系列相關聯的句子（Set of Sentences），結果如圖 3-10 黃色方框，並計算每個候選句的分數，最後再將高分數的句子當作提示訊息。如果查詢字串的來源是由句子生成，則會取得用字遣詞類似的段落文章（Set of Paragraphs），如圖 3-10 紅色區域，表示所描述的主題即有可能相同。系統將這些段落經過事前訓練好的預測模型（SVM Predict Model），並且為每個段落標上對應的文章類別（Label Category），如圖 3-10 棕色線條，用意是提供主題分類，讓使用者能夠針對目前寫作的方向選取相關提示訊息。

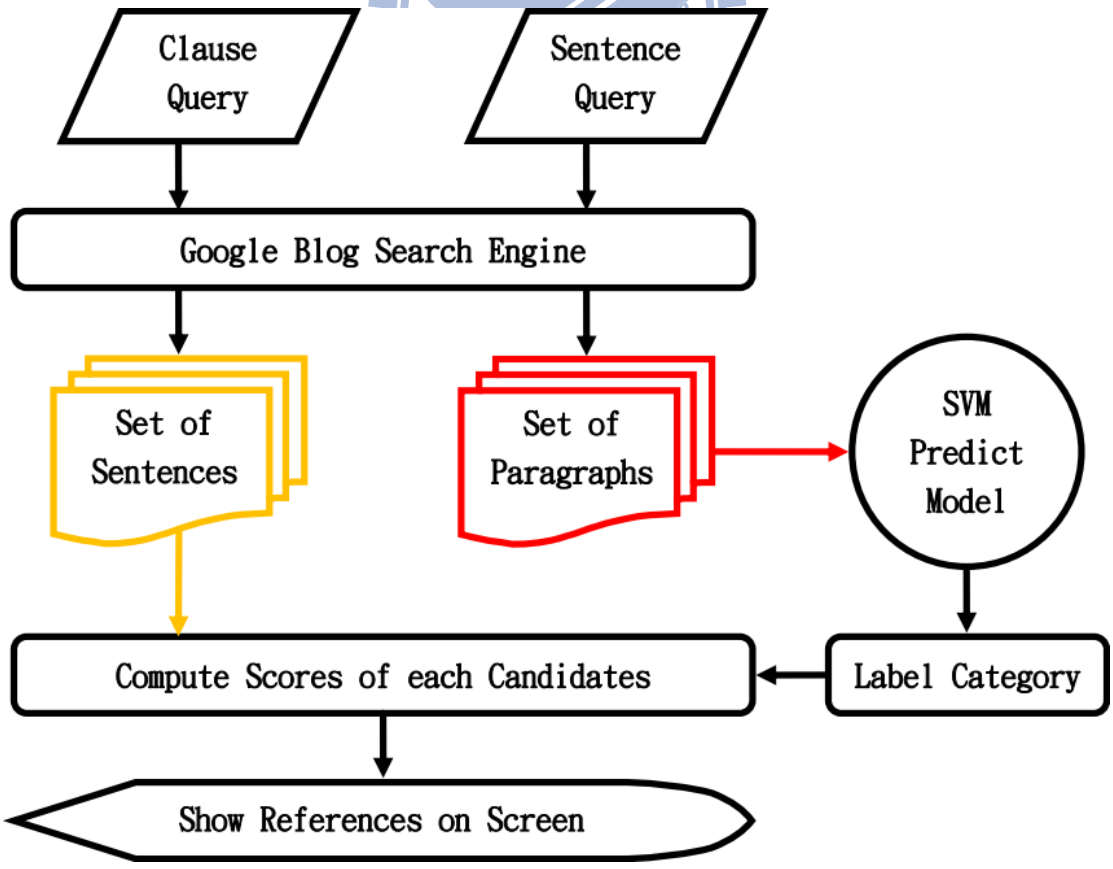

圖3-9：參考提示產生的過程。

圖3-10：Google Blog Search Result Example。

我們以查詢字串的種類作為提示分類之依據，將系統所產生的參考資訊劃分為兩種型別，子句與句子提示。下列為各類提示計算分數的詳細方式，並舉範例與圖解說明之：

1. 子句提示（Clause Reference）

藉由此類查詢字串所收尋的結果，會從網路上獲得相關的句子。系統將根據查詢字串中的核心關鍵詞彙作為評分的標準。下圖 3-11，為子句類型查詢字串計算分數的實際範例：

查詢字串："天氣 ＊ 適合 踏青"

取得資料：今天天氣非常的適合踏青，是晴朗無雲的好天氣。

計分方式：① 查詢字串完整出現於資料中，「天氣…適合踏青」，代表與使用者所撰寫的子句有非常高的相似程度，系統會給予該句極高的分數(Score A)，如下圖紅色區域。

② 只有查詢字串的部分詞彙出現於資料中，「天氣」、「適合」或「踏青」，系統會給予基本分數(Score B)，如下圖綠色方塊。

③ (Score A) ≫ (Score B)

最後該句子的分數為所有獲得分數的加總，(Score A) + (Score B)，當所有的句子都計算出各自總分，系統會自動挑選較高的項目成為提供使用者參考寫作的提示。

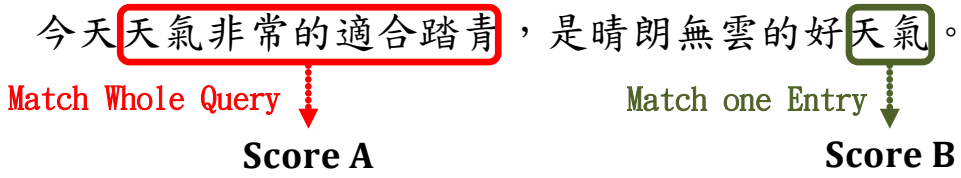

圖3-11：查詢子串為「"天氣 * 適合 踏青"」的提示句分數。

2. 句子提示（Sentence Reference）

若以完整句子的核心關鍵詞作為查詢字串，可以由部落格搜尋引擎取得相關聯的段落文章。首先，系統會利用支援向量機模型預測所有段落，並標上對應的主題類別。而後，將擁有相同類別的段落收集起來，再以句點切割成提示句，分別計算每個句子的分數，並按照分數高低排序。下圖 3-12，為句子類型查詢字串的計算範例：

查詢字串：晴朗 郊遊

訓練特徵：｛今天，天氣，家人｝

取得資料：今天是個晴朗無雲的好天氣，適合全家人出門郊遊。

計分方式：① 當系統於句中尋找到「晴朗」、「郊遊」，會給予較高的分數(Score A)，表示該資料與目前寫作的方向有某種程度的相似，如下圖藍色方框。

② 假如找到的項目屬於特徵集合內的元素，「今天」、「天氣」或「家人」，統 僅給予基本分數(Score B)，如下圖紅色方框。

③ (Score A) > (Score B)

最後結果是所有取得分數的加總，範例中計算出來的總分為 2 × (Score A) + 3 × (Score B)。對於每個文章類別，系統都會挑選其中分數最高的幾個句子成為該類別的提示。在使用者根據文章撰寫的方向選擇主題類別後，系統會自動將該主題底下整理好的資訊提供給使用者作為參考。

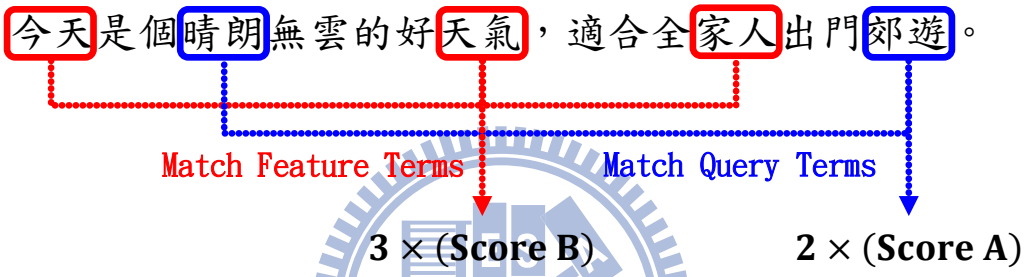

圖3-12：查詢字串「晴朗 適合 郊遊」的計算範例。

提示的使用方式，如下圖 3-13 所示。首先，使用者根據目前文章選取寫作的類型，系統會列出該主題底下的參考句（藍色視窗）。我們以 Ogata 等人提出的 Speech Repair[22]作為提示選取介面的設計雛型，將所有句子中的組成單元切割成字元單位，並按照原始句中的順序排列，如：「今天天氣真好」經過處理後會成為「今」、「天」、「天」、「氣」、「真」、「好」。使用者可以自由選擇提示句內的各字元搭配(紅色軌跡)，亦允許加入自己的手寫文字（藍色圓圈)。如範例中，使用者依序利用提示「我們」和「去八斗子潮境公園情人湖」，再手寫輸入「玩」，組成新的句子。提示訊息的主要目的是期望使用者可以藉由觀察參考句進行腦力激盪，如：範例文章寫「天氣好」、「到基隆玩」，則使用者參考提示句「情人湖」、「基隆八斗子潮境公園」，聯想到可以描述去了哪些基隆的景點遊玩。

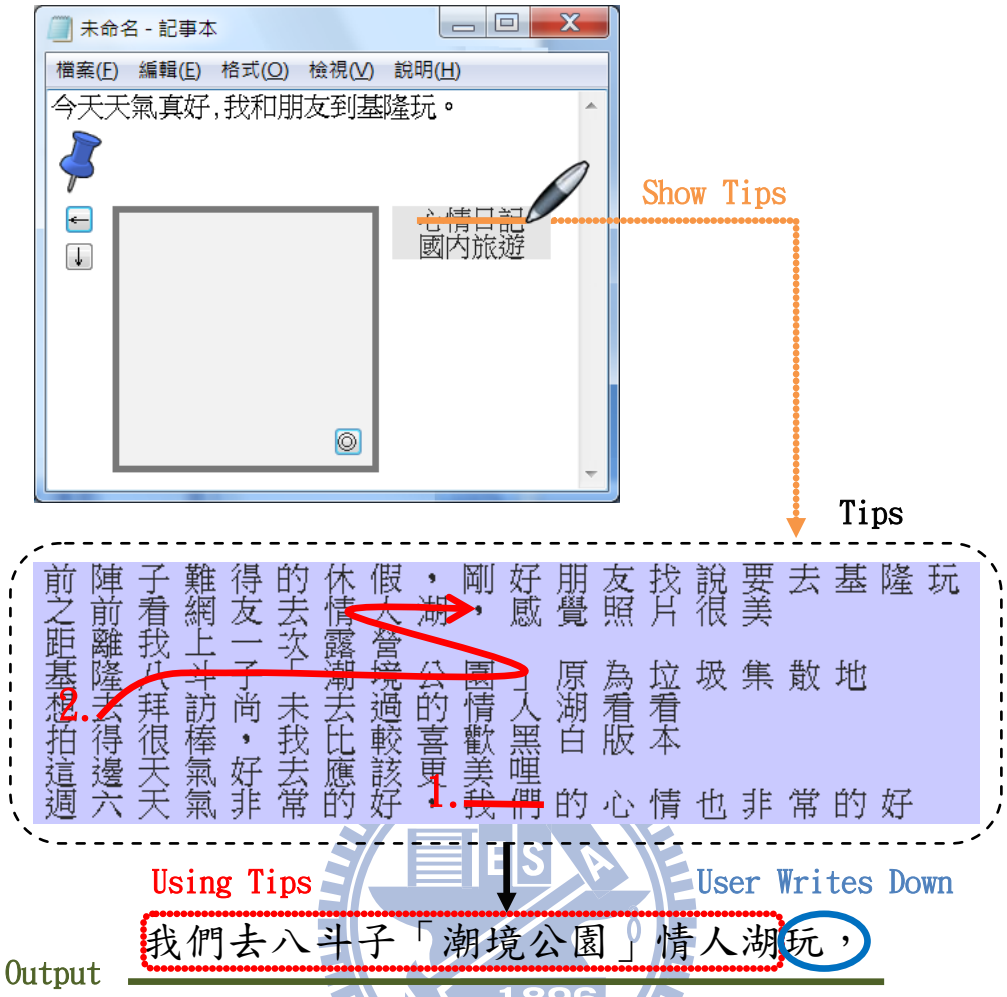

圖3-13：系統提示的使用方式。

第四章、實驗過程與結果

4.1 實驗設計

本研究的系統實作於 Microsoft Windows 7 作業系統，軟體開發的環境平台為 Visual Studio 2008 C#.NET，程式語言是 C#。系統利用 Microsoft Tablet PC Platform SDK[20]提供使用者手寫辨識的功能，並以網路中大量的部落格文章作為程式提示的資料來源。實驗測試時，我們所使用的電腦配備為 Intel DualCore 2.93-GHz 中央處理器、4096 MB 記憶體、640 GB 硬碟，搭配使用 Wacom Intuos 2 Graphics Tablet 6x8 XD-0608-U 數位繪圖板，讓受測人員能夠以手寫方式快速輸入文字。

我們邀請 12 位測試人員，10 位男性與 2 位女性，年齡層的範圍為 12-43 歲，進行系統實驗與效能評估，其中包含 1 位國小生、1 位國中生和 1 位大學生，其餘人員的學歷皆為研究所含以上。首先，我們請受測人員練習系統的操作方式，等待使用者熟悉所有細節後，每位人員會被分配文章主題進行實際寫作的測試，其種類包含旅遊類、運動類、心情類、美食類和電影類，我們將從中亂數挑選兩種類型的主題讓同一位測試者撰寫，最後紀錄使用者在兩種模式下所花費的時間總和與使用評價。

對於每一類型的文章主題，我們都要求測試人員分別使用無提示訊息的程式和有參考提示的版本，以手寫輸入的方式進行文章寫作，並限制文字的總數至少超過 200 個字元。為避免測試人員撰寫同類型文章的時間過近，因腦海中記憶猶新的緣故，而造成第二次花費時間的縮減，我們特別規定使用者第一次和第二次寫作時間的間隔須差距至少一個禮拜以上。另外，亦要求測試人員文章的內容不得過度相似，意即若使用者須撰寫電影類的主題時，利用無提示和有提示程式所寫的文章內容，其描述對象不得是關於同一部電影之心得。

4.2 實驗結果

由於每個人的思維模式與表達能力有所差異的緣故，使得每位測試人員的寫作時間與文章篇幅皆不盡相同。因此，為讓統計數據可以公平且清楚的表示實驗結果，對於各主題下的每篇文章，我們都將該篇文章的總字數除以寫作時間總長，取得文章平均每個字元所花費的秒數（Seconds / Character），其中寫作時間為手寫與思考的時間總和。詳細的統計數據，如下圖 4-1 所示，各類別以顏色深淺分別標示使用無提示和有提示版本時，平均每個字元所花費的時間。

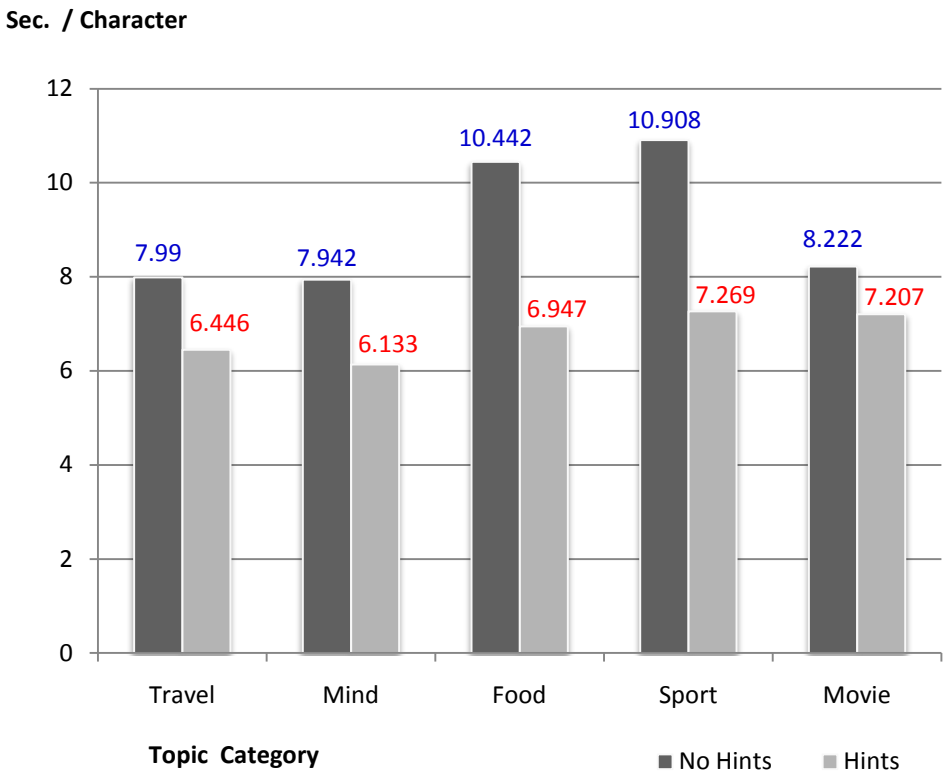

圖4-1：各類文章寫作時間的統計資料。

根據觀察數據長條圖顯示，當測試人員使用有提示版本的程式時，於每一個字元上所花費的時間，皆較使用提示前還少，甚至於部份主題下，寫作的效率更有顯著的提升。下圖 4-2 為各類別使用提示前後減少時間的百分比，我們將針對每個類別的統計資料作討論說明。

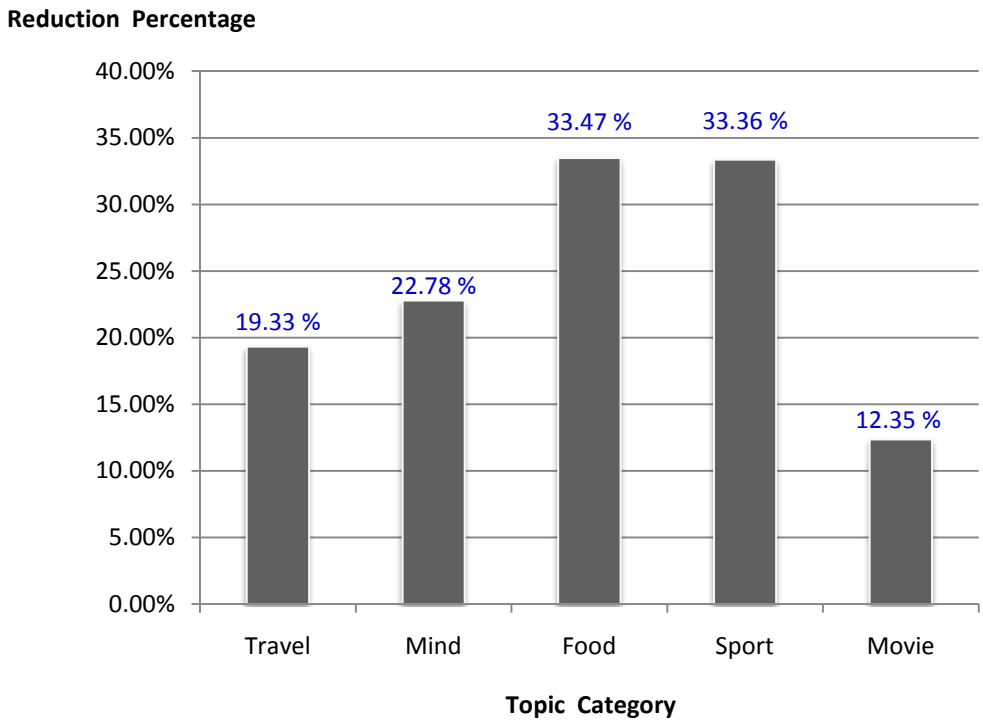

圖4-2：提示前後寫作時間減少的比例。

在整理統計數據的資料表現後，我們發現美食類與運動類的文章，時間減少的比例最為顯著，均可達 30%以上，上述原因可能是由於談論到美食類的網誌，其內容多半為講述食物如何美味可口的修飾詞與比喻句；而運動類的文章則大部份在討論各種運動的方式與對身體的益處。因此，當描述的事物已被收斂於某個範圍時，給予的提示的可用性就會大大提升，亦容易切合使用者所寫的主題。然而，對於旅遊類與心情類，每個人在旅遊或一天中所碰到的人事物的組合千變萬化，容易有主題發散的狀況產生，所以提示輔助的效果不如運動類與美食類文章好，節省約 20%的寫作時間。電影類的文章，則是效果最不顯著的類別，僅減少約10%的時間，我們檢視有可能是因為關於心得類的文章，使用者往往過於主觀，擁有自己獨特的觀點與看法，造成參考提示不易被採納使用。經由以上結果及分析，我們初步歸納文章類型不同，會直接影響到系統幫助使用者寫作文章的成效。

4.3 使用者評比

由於本研究的主要目是提供適當的參考提示，幫助使用者撰寫部落格文章。因此，除了從數據統計的角度分析系統是否能夠有效的輔助寫作外，我們亦將測試人員使用系統後的評價作為系統整體性重要的評估依據。評比的方式，主要是請使用者針對下列五個部份給予適當的分數：

1. 實用性（Practicality）
   ① 在寫作的過程中，當我不知如何下筆時，提示句能夠幫助我思考，進而想到可以撰寫的內容。
   ② 因為使用此程式，讓我從寫作文章上得到不少便利。
2. 準確性（Accuracy）
   ① 提供的句子中，會出現類似於我想要或是預計要描述的主題。
   ② 我在寫作的過程中，曾經使用過提示句當作我文章內容的一部分。
3. 互動性（Interaction）
   ① 操作介面讓我能輕易自然的使用。
   ② 我從第一次使用到熟悉系統的操作只花了少許的時間。
4. 學習性（Learnability）
   ① 提示句含有我未曾看過的詞彙用法，讓我順便學習新的用詞。
5. 滿意度（Satisfaction）
   ① 我覺得整體使用完後的感覺還不錯。
   ② 我會推薦國中、國小學生或在學年齡的孩子，當作學習寫作文章的輔助工具。

我們要求使用者依照各項符合程度給予 1－5 分不等的分數，其中評比分數的表示意義，如下表 4-1 所示：

表 4-1：評比分數與對應的意義。

意義	極不同意	不同意	普通	同意	極為同意
分數	1	2	3	4	5

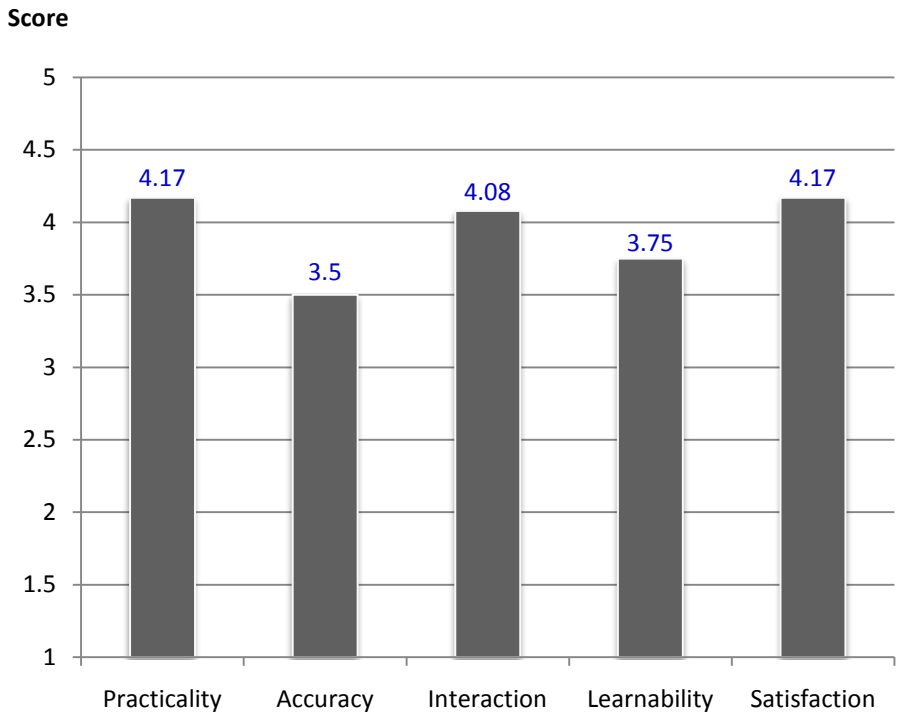

圖4-3：各項標準的平均值。

測試人員各項評價的平均值，如上圖 4-3 所示，數值越高的部份表示系統的表現越符合使用者期望。觀察結果顯示，系統的實用性與滿意度都高達 4.17 分，代表大部份的測試人員都曾經在陷入創作苦思的時候，透過參考提示聯想到接下來的撰寫內容，解決無法下筆的困境，並且認為此套輔助寫作的系統能夠真正幫助人們撰寫部落格文章。另外，在互動性方面的結果亦取得 4.08 的高分，表示介面操作相當人性化，普遍的使用者都能夠在短時間內學會如何操作系統的各項功能，駕輕就熟的使用系統完成文章的寫作。

然而，於學習性的部分卻稍嫌不足，只得到 3.75 分。我們推測原因有可能是受測人員的學歷多數為研究所以上，由於使用者本身在語言程度上已有一定的水準，使得透過參考提示學習新的用語和詞彙的可能性會比較少。然而，我們分析受測人員中屬於學齡中的孩子，發現他們對於大部分的事物仍處於學習階段，許多名言錦句皆尚未聽過，因此，容易藉由使用輔助提示從中學習新知。

在所有評比項目中，得分最低的部分是準確性，主要的原因是系統僅能依據當前撰寫的內容收集網路上的相關資料，再整理成提示給予使用者參考，假如測試人員本身已經想好整個文章的寫作架構與描述方式，可能會發生離題的狀況，例如：當使用者描述的內容是「天氣好」，並在內心中決定接續描寫有關於「出外郊遊」的事情，若此時系統查詢的結果為「打籃球」和「去游泳」，則使用者會直覺的認為提示不符合預期。

另外，我們亦請使用者在對五個部份作完評比後，留下本系統的使用心得。在匯整資料後，我們發現國中和國小的孩子，對系統產生的寫作提示特別感興趣，喜歡利用這些提示句拼湊文章的內容。正所謂，一切偉大的創作都是由臨摹開始，這表示在寫作的過程中，不僅能訓練孩子們組織文章的能力，並同時激發創作的潛能和靈感。較為令人困擾的部分，是許多測試人員在使用手寫輸入的時候，會忘記某些國字的寫法，因此耗費比較多的時間於輸入上，但換個角度思考，這個問題本身是由於人們長期習慣使用鍵盤輸入法，而越來越少動筆手寫文字所造成的結果，甚至使用者可以藉此機會將遺忘已久的國字重新熟悉記起。

本篇論文研究的主要動機是希望幫助平時寫作能力善待加強的使用者，在寫作陷入苦思時，藉由參考提示聯想到接續的寫作方向。在測試的人員中，就有實際例子，該人員因平時寫作的經驗較少，因此使用提示前所花費的時間總長為 1 小時 7 分鐘，而使用提示後僅寫 31 分鐘，減少的時間幾乎是原來的一半，亦即寫作的速度整整快上兩倍。然而，我們亦注意到某些測試人員在寫作的時間上，使用提示前後的差距並不大，經過調查後，發現當使用者本身很有主見，清楚的明白每一個段落寫作的方向，過程中就不再需要參考系統給予的提示，而可以順暢的完成文章。

第五章、結論與未來展望

5-1 研究總結

本篇論文主要介紹部落格輔助寫作系統，目的是幫助陷入寫作困境的人們，藉由給予使用者撰寫方向的提示，激發腦力思考，最後順利完成文章。最後，實驗的數據結果顯示，大部份的測試人員，在使用系統時確實能夠從參考提示得到寫作的幫助，花費的時間平均減少約原來的 20%。在使用評價的部分，使用者對系統的表現多數感到滿意，我們亦發現學齡中的孩童，喜愛使用系統提示句組成文章內容，這樣的結果顯示系統提示句對於孩童在寫作文章時具有啟發性的功能，如此不僅可以激發孩童的潛能創造出更多不同的文章內容，同時也可以透過輔助提示學習新的詞彙語用法。

5-2 未來展望

在自然語言的領域中，解決問題的方法都可以藉由數學統計與機率模型達成，但卻缺少語意解析的部分，因此容易造成文章不通順的狀況產生。在本篇研究論文的實驗評價中，使用者普遍認為系統所提供的提示準確率仍不夠，會有牛頭不對馬嘴的狀況產生。因此，若是未來能夠針對上下文結構的語意關係進行研究，並且建立相關的知識庫，輔助文章提示的產生，或許可以增加參考提示的準確性，亦更加貼近使用者創作文章的想法。另外，為解決使用者忘記文字寫法的困擾，我們亦考慮加入注音符號的方式，讓系統的輸入方式有更多種選擇。

參考文獻

[1] Bonnie A. Nardi, Diane J. Schiano, Michelle Gumbrecht, Luke Swartz, Why we blog. Communications of the ACM, Volume 47 Issue 12, pp.41-46, 2004.

[2] Google Blog Search Engine, http://blogsearch.google.com.tw

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

[4] Tablet Personal Computer, http://en.wikipedia.org/wiki/Tablet_PC/

[5] Rob Jarrett, Philip Su, Building Tablet PC Applications. Microsoft Press, ISBN：0-7356-1723-6, 2002.

[6] Managed Library for Nintendo's Wiimote, http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx

[7] Johnny Chung Lee, Hacking the Nintendo Wii Remote. IEEE Pervasive Computing, Volume 7 Issue 3, pp.39-45, 2008.

[8] Kazutaka Kurihara, Masataka Goto, Jun Ogata, Takeo Igarashi, Speech pen: predictive handwriting based on ambient multimodal recognition. CHI'06, pp.851-860, 2006.

[9] Ping-Che Yang, WriteAhead: An Abstracts Writing Assistant System for Academic Writing. 國立清華大學, 碩士論文, 2009.

[10] Ting Liu, Ming Zhou, Jianfeng Gao, Endong Xun, Changning Huang, PENS: a machine-aided english writing system for Chinese users. ACL'00, pp.529-536, 2000.

[11] Microsoft Research ESL Assistant, http://research.microsoft.com/en-us/projects/msreslassistant/

[12] Tien-Chi Huang, Shu-Chen Cheng, Yueh-Min Huang, A blog article recommendation generating mechanism using an SBACPSO algorithm. Expert Systems with Applications: An International Journal, Volume 36 Issue 7, pp.10388-10396, 2009.

[13] Bo-Yuan Ding, Intelligent Computer-Aided Article Writing. 國立交通大學, 2009.

[14] Article Writing Assistant, http://articlewritingassistant.com

[15] Kiyotaka Uchimoto, Hitoshi Isahara, Satoshi Sekine, Text generation from keywords. Proceedings of the 19th international conference on Computational linguistics, Volume 1, pp.1-7, 2002.

[16] Corinna Cortes, Vladimir Vapnik, Support-vector networks. Machine Learning, Volume 20 Number 3, pp.273-297, 1995.

[17] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: a Library for Support Vector Machines, 2001. Software Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[18] Pai-Hsiang Hsiao, Libtabe：A Chinese character and word handling library, http://cle.linux.org.tw/xcin/libtabe/index.html

[19] 新浪部落, http://blog.sina.com.tw/article_categ.shtml

[20] Microsoft Tablet PC Developer Center, http://msdn.microsoft.com/tabletpc

[21] Zan-Wei Liao, Automatic Text Summarization System for Chinese News. 國立交通大學, 2009.

[22] Ogata Jun, Goto Masatakao, Speech Repair: Quick Error Correction just by Using Selection Operation for Speech Input Interfaces. Eurospeech '05, pp.133-136, 2005.