視覺化故事書產生引擎

黃正宇 游家祥 朱威達
資訊工程學系 資訊工程學系 資訊工程學系
國立中正大學 國立中正大學 國立中正大學
lairoeaes@hotmail.com xneonvisionx@gmail.com wtchu@cs.ccu.edu.tw

摘要

此論文整合各領域技術,開發一個替文章添加插圖的系統。我們結合自然語言處理、影像搜尋與影像處理，讓系統能判斷文章中的關鍵字、從網路中搜尋影像、並適當地在文章中插入圖片，產生圖文並茂的故事書。系統介面有兩種模式，第一是自動模式，輸入一段英文故事，系統自動幫使用者找出圖片貼在各文字段落下。第二個模式是手動模式，使用者自行尋找符合文章的圖片，系統幫忙產生出 html 程式碼，幫助使用者產生圖文並茂的部落格文章或另外產生 html 網頁。

關鍵詞：視覺化故事、自然語言處理、影像比對、mutual reinforcement

1. 前言

1.1 動機

圖片對人的吸引力遠比文字對人的吸引力高。我們在讀一篇純文字的文章時，若是能用圖片加以輔佐，一定多少會增加文章的可讀性。因此我們做出一個可以自動判斷文章中的關鍵字，並挑出適當的圖供讀者觀看文章的系統。

而為了讓系統有所彈性以及讓使用者有所互動，我們做出自動與手動模式，也設計了靈活與直覺的介面，讓使用者更有意願對我們系統作互動。

1.2 概念

為了使系統能處理使用者輸入的故事，了解句子中的意思，我們需要使用自然語言處理的技術，包括分割句子、詞性分析、語意推測。系統再根據語意設定適當的關鍵字，使用影像搜尋引擎尋找符合圖像，再根據適當的組合，產生與語意相近的畫面，讓故事更加多采多姿。

對於靈活與直覺的介面,我們從近年來流行的智慧型手機得到靈感。他們運用大量的拖曳效果與幾顆簡單的按鈕，讓使用者運用他們的軟體。我們依照這種概念，使用了 HTML5 的新技術——Drag and Drop API，我們使用了這個在去年才釋出的 API，讓此系統更能簡單地運用。

1.3 文獻探討

Joshi 等人發表一篇名為 The Story Picturing Engine 的論文(以下簡稱 SPE) [1]，與我們的研究題目十分相似。其目的也是找出故事的關鍵字，再搜尋相對應的圖片插在文字旁邊。他們的方法如下：

(1)將一個故事段落中的重要名詞擷取出來，再以這個關鍵字去搜尋圖像資料庫，找出候選圖片。
(2)建立一個有向圖(directed graph)，每個點(node)代表一張圖片，每個邊(edge)的權重代表圖片與其他圖片的相似度。
(3)利用Mutual reinforcement概念找出最合適的圖片。Mutual reinforcement (圖1)是把各個項目視為圖(graph)的點，項目之間的某種關係為邊(edge)，根據與該點連接的邊的權重總和，判斷那個點最為重要。SPE是以相似度作為依據，認為某張圖與其他圖相似度越高，就越有可能是要尋找的目標。

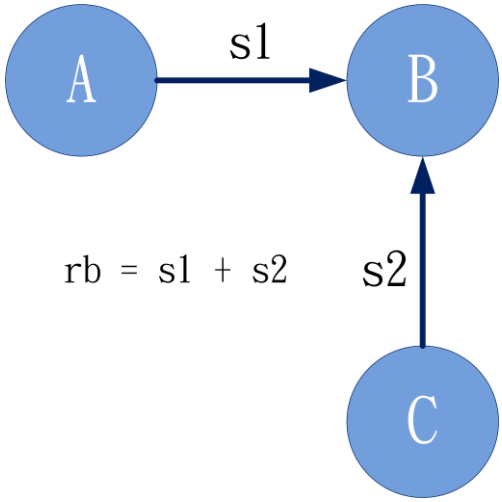

圖 1 Mutual reinforcement 概念圖

SPE 這個系統是最早自動化產生圖文並茂文章的系統，但他們搜尋出來的圖片並沒有精確地呈現出故事的細節，因為 SPE 只針對名詞做處理，忽略動詞與副詞，當然就無法表現故事中所要表達狀況。除此之外，整個流程的處理時間、圖像資料庫的大小，都影響著 SPE 的搜尋時間與搜尋出來的圖是否符合故事時間、環境等等。

整體而言他們所提出來的想法十分有創意，帶給我很多的啟發與相關技術的研究，有助於這項研究。

1.4 開發環境

1.4.1 OpenNLP

OpenNLP 為一個研究機構開發出來的自然語言處理 API [2]，提供許多自然語言處理的基本功能，包括句子偵測、詞性分析、詞性分析、消除指代詞等，為許多研究人員提供基本的語言處理。我們利用其分析語法結構來找出語句中的主詞，進而判斷出故事中的主角與場景。

1.4.2 AlchemyAPI

AlchemyAPI 是更有智慧的自然語言處理 API [3]，不單只是分析語句詞性，更找出許多在文章中的各項特徵，例如文章分類（Text Categorization）、關鍵字萃取（Keyword Extraction）、情緒分析（Sentiment Analysis）等等，對於分析語言中隱藏的訊息有莫大的幫助。

1.4.3 OpenCV

OpenCV 是一套開放的電腦視覺函式庫，提供電腦視覺的各項演算法實作，包含基本的影像處理、各種過濾器、邊緣偵測、特徵分析、機器學習等，在分析圖片與處理提供完善的支援。

1.4.4 HTML5 簡介

HTML 5 是 HTML 下一個的主要修訂版本，現在仍處於發展階段。目標是取代 1999 年所定訂的 HTML 4.01 和 XHTML 1.0 標準，以期能在網際網路應用迅速發展的時候，使網路標準達到符合當代的網路需求。

廣義論及 HTML5 時，實際指的是包括 HTML、CSS 和 JavaScript 在內的一套技術組合。HTML5 為多媒體和網頁應用程式立下標準並提供運作機制，它希望能夠減少瀏覽器對於需要外掛程式的豐富性網路應用服務（plug-in-based rich internet application，RIA)，如 Adobe Flash、Microsoft Silverlight，與 Oracle JavaFX 的需求，並提供更精細的方式來呈現網頁元素，如透明度、圖文旋轉、文字分欄、內嵌字體、點陣繪圖、向量繪圖等，也可運用許多新的 API，讓網頁應用程式可以輕鬆地達成更多功能。

1.4.5 SIFT

SIFT [9] (scale-invariant feature transform)是在電腦視覺(computer vision)領域中偵測區域性特徵點的演算法，具有角度、尺度、旋轉不變性，也就是即使物體有大小或旋轉，甚至是微幅的視角改變，對同一個物體，計算出來的特徵點還是一樣，是一個很好的物體辨識演算法。

目前有許多實作 SIFT 的函式庫可供使

用，我們將 Rob Hess 先生的 SIFT Library [10] 加入到我們系統中，作為圖片分析的工具。此函示庫還提供兩張圖片最佳配對的演算法，來判斷兩張圖片是否有相同的物體。

2. 系統說明

2.1 系統架構

2.1.1 自動模式

整個系統分成兩種模式，第一個是自動模式－系統根據使用者的輸入，自動分析內容並產生結果；另一種為手動模式，提供編輯介面與影像搜尋，讓使用者能更輕鬆編輯圖文並茂的文章。

自動模式，顧名思義就是讓系統能夠很聰明地找出符合故事的影像，想要達到如此功能，可以分成幾個步驟，如圖 2 所示：

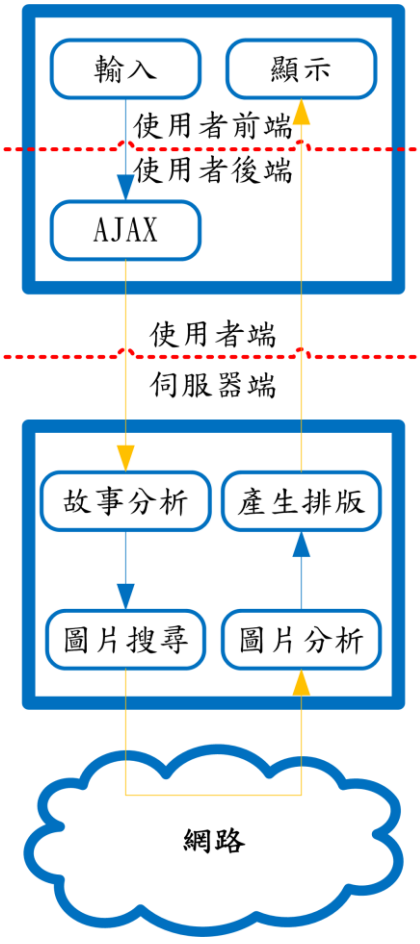

圖 2 自動模式流程圖

1. 使用者輸入故事標題與內文，發送給伺服器。
2. 系統對故事做基本的自然語言處理後，判斷故事的關鍵字。
3. 系統以關鍵字自 Google 影像搜尋引擎(Google Image Search)尋找可能圖片。在此，系統可利用 Google image search API 搜尋到不同顏色屬性或風格的圖片。
4. 系統使用 SIFT 比對，從搜尋而得的圖片中找出最佳的圖片。
5. 系統最後將圖片與文字以預設的介面顯示給使用者觀看。

首先，提供簡單的網頁表單讓使用者輸入故事的標題與內文，並讓使用者自行使用系統預設的分段符號將故事分成多個段落，系統就能以此作為最終成品的分頁依據。除此之外，使用者可以調整系統的圖片搜尋選項，針對偏向某一種色系的圖片進行搜尋，或是包含特殊繪畫風格，例如臉部特寫、相片、插畫等圖片進行搜尋。此功能主要由系統呼叫 Google image search API 完成，可以讓使用者限制要找出的影像風格。

完成輸入後，文章傳送給系統進行處理。系統在文章分析之前先將分段符號去除，以獲得原文與各個段落，接著以一個段落作為分析單位，去尋找一張符合此段落的圖片。

要處理段落中的敘述，系統使用自然語言處理(Natural Language Processing)的技術，目的是為了能讓電腦能理解人類的語言，並能在多個語言間互相轉換。目前處理英文的技術比較成熟，並有許多現成的函式庫可以使用，因此我們就以英文作為分析的語言。

要分析段落，可從各個句子下手，接著分析句子的結構，解讀句子的意思。想要達到上述目標，可以使用 OpenNLP 裡面的函式達成。首先，使用句子偵測器(sentence detector)將敘述切割出多個段落，再使用切割器(Tokenizer)，獲得句子中的各個單詞(word)，最後交由詞性分析工具

(Part-of-Speech tagger)，將各個單詞標註上詞性，完成基本的段落結構分析。

接著為了能找出符合段落的圖片，就需要能讓系統理解段落中的敘述。但就目前的技術來講，還需要克服許多的難題。我們以「如果段落中出現頻率高的名詞，很有可能是段落中的很重要的人事物。」這個觀點去作為搜尋圖片的依據。如果輸入的故事是有名的話，那麼一定會有很多圖片，且會出現關鍵人物。即使故事是使用者自行創作的，也能自動尋找關鍵的人物所組成的圖片回來給使用者參考。

但如果純粹以單詞頻率作為關鍵字判斷的標準，效果還不夠好，因為如果敘述過短而出現次數甚低的時候，就沒辦法準確判斷。我們需要一個資料庫來協助，然而這需要大量的資料來分析。所幸AlchemyAPI 提供關鍵字萃取的功能，原本它被使用於讓網頁的文章能自動產生關鍵字，方便搜尋引擎能快速分類。我們則利用它產生關鍵字來搜尋可能符合文章的圖片。AlchemyAPI 的分析結果包含數個關鍵字與該關鍵字的相關程度，數值範圍是 1 到 0 之間，本系統以相關程度 0.6 的字詞為關鍵字。

獲得由單詞頻率判斷與 AlchemyAPI 產生的關鍵字群後，就可以此搜尋符合故事敘述的圖片。目前有許多的網路搜尋引擎都具有影像搜尋的功能，我們選擇一個有提供影像搜尋 API 的搜尋引擎，直接從網路中尋找。Google 提供一個完整的影像搜尋 API，而且允許多個關鍵字同時搜索，系統將所有的關鍵字串起來，向 Google 請求搜尋，找到的影像回傳到系統進行處理，最多可以達到 32 張。

再來就是讓系統能夠判斷甚麼是重要的圖片。想要達到這樣的功能，我們有許多想法，一種是建立大型圖片資料庫，並給適當的索引，這樣的做法其實在使用 Google 影像搜尋時就已經包含在裡面了，甚至還用到許多影像處理的技術加以分類，提高搜尋的準確度。在本系統中，我們修改 SPE 中提到 Mutual reinforcement 概念來增加搜尋的精確度。方法是將使用同一個關鍵字而得圖片，兩兩比對相似程度，再以下公式計算重要程度：

$$r_i = \sum_{j=1, j\neq i}^{L} s_{i,j}, \quad (1)$$

r_i 表示第 i 張圖片的重要程度，L 表示候選圖片的個數，$s_{i,j}$ 表示第 i 張與第 j 張的相似度。計算兩兩圖片的相似程度，我們以 SIFT 找出特徵點，再以 SIFT library 的配對函式計算兩個圖片有多少特徵點符合，作為相似度的依據。最後再以最高數值作為最可能的圖片。實作上為了增快處理速度，以 C 語言改寫 SIFT library 的函式，讓 PHP 能呼叫多個程式同時分析各個段落所找到的圖片群。最後，將找到的圖與故事傳到使用者端，經由系統預設的介面顯示出來。

2.1.2 手動模式

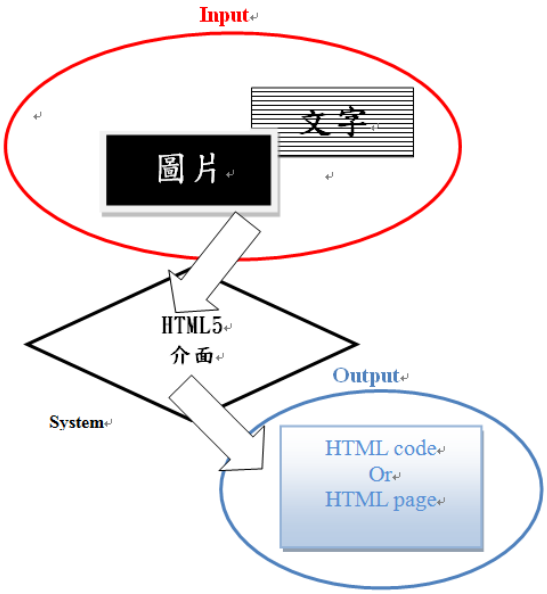

圖 3 手動模式系統架構圖

手動模式主要有三個部分(圖 3)：輸入、輸出和系統介面，輸入包含圖片以及文字段落，因為是手動模式，所以圖片是由使用者自行由我們所建置的 Google image search API 的介面找尋，文字則在簡易編輯區編輯。系統介面則是用 HTML5 建成，使用者可自行拖拉圖片或是段落至指定區域，之後文章會依序使用者的拖拉順序依序排版完成。最後使用者可以選擇是否要產生 HTML code 或是 HTML page

作保存。

2.2 系統介面

2.2.1 自動模式(auto mode)

如圖 4 所示,使用者輸入故事標題與內容，選擇圖片的色系或特定類別，例如臉部特徵照、相片、線條畫等等。另外還可設定是否使用 SIFT 分析，因為使用 SIFT 會花比較長的執行時間，使用者可以取消以減少處理時間，直接從找到的影像中挑選相關程度最高且大張的圖片。

送出後，系統分析完產生結果，如圖 5 所示。可能有多個結果，使用者可以選擇箭頭觀看下一頁，如圖 6 所示。

最後，整合系統效能評估畫面，我們請使用者填寫對於成果的滿意度，做為系統回饋之用，如圖 7 所示。

2.2.2 手動模式(manual mode)

首先使用者開始編輯文章與尋找圖片，圖片搜尋有顏色、檔案類型以及風格樣式可以選擇，如圖 8 所示。找到圖片後，使用者可將選取到的段落或是喜歡的圖片拖曳至指定區域，如圖 9 所示。

最後使用者選擇產生 HTML code 或是 HTML page，將自己的圖文文章存取下來(圖 10，圖 11)。

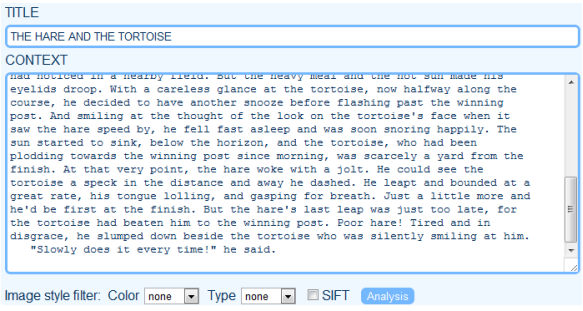

圖 4 故事輸入區

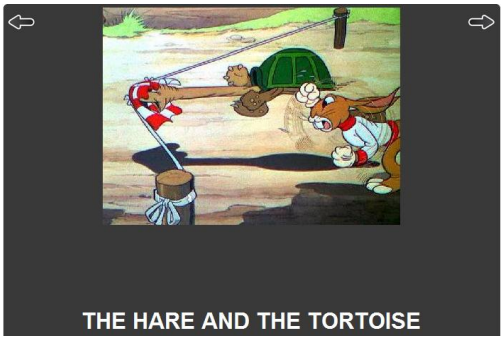

圖 5 顯示成果（頁一）

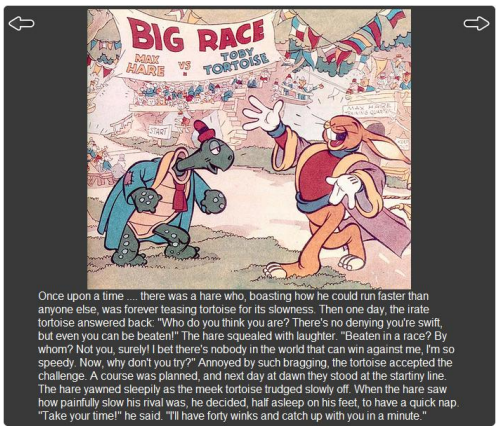

圖 6 顯示成果（頁二）

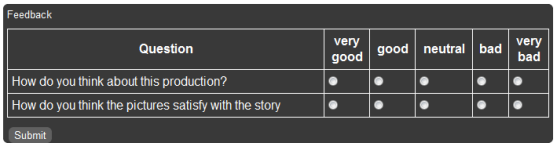

圖 7 系統回饋

圖 8 簡易編輯區以及圖片搜索區

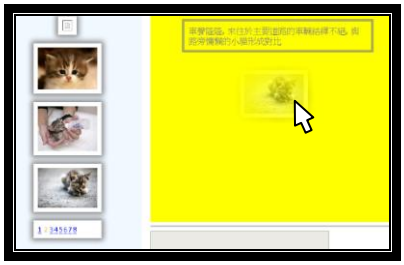

圖 9 簡易編輯區以及圖片搜索區

圖 10 HTML 頁面展示

圖 11 HTML code 產生區

3. 實驗

3.1 不同類型文章對自動模式的結果與討論

我們找了四種較常出現在部落格的文章。這些文章語言皆為英文，每個類型的文章數量為九或十篇，文長約 100~500 字。此實驗目的是為了找出那一種文章類型對我們的系統最適合。

表 1 不同文章類型之符合文義圖片比率表

文章類型/文章數量	符合文義圖片數/所有圖片	符合文義圖片比率	達到一定符合率的文章比例	
			50%	60%
Description /10	28/37	76%	80%	80%
Dialogue /9	11/27	41%	67%	22%
Narrative /10	8/51	16%	0%	0%
Theme /10	25/40	63%	90%	50%
Total /39	72/155	46%		

註：

1. **符合文義的圖片之定義是指該張圖片是否符合段落中的場景、人物或重點物件。不符合以上三點、圖片品質不佳（太小、不清晰等）或無法顯現等，皆視為不符合文義。**
2. **符合文義圖片比率計算方式：**
   (符合文義圖片數/所有圖片數)*100%
3. **系統測試之全部文章細節連結**
   http://www.cs.ccu.edu.tw/~hchy97u/NGSPR/doc/Expriment1_Detail.pdf

首先由上至下第一種為 Description，文長 300~500 字，這類的文章皆為遊記，主要是在記敘事物。就結果來看，系統產生出來的圖片大都為當地景觀，整體符合率也相當高，約 76%。符合率這麼高的原因，我們推測是因為這些文意明顯，作者在撰寫文章時大多都是一再地強調景點，所以用 AlchemyAPI 找出關鍵詞時，這些一再強調的景點字詞很容易就被挑到了。

第二種為 Dialogue，文長 100~200 字，這類的文章為兩人的日常生活對話，整體符合率約 41%。我們發現若是兩人持續在講同一事物，AlchemyAPI 可輕易找出重點詞，若兩人講的事情相當廣，像是對話之中有許多不同的名詞，則 AlchemyAPI 可能找出的是比較不重要的詞，造成圖文之間不是很搭配。

第三種為 Narrative，文長約 300~500 字，這類的文章有人物劇情，有時有對白，像是平常閱讀的小說。這類的文章整體符合率相當差，僅有 16%。因為文章裡有對

話更有許多分散的名詞，這類的文章因為著重於情節對話，而 AlchemyAPI 無法得知情節，無法挑出讀者覺得重要的詞。最後造成挑出的圖明顯地搭配不上文字，比上一類 Dialogue 程度更甚。

第四種文章類型 Theme，每篇文長約 300~500 字，這類的文章主要是在定義一種物品，與第一類文章類似，若是文章持續提到這種物品或是相關的詞，AlchemyAPI 找出的關鍵詞及有很大機會符合文意，整體符合率有 63%。

以整體符合率來排序，效果最好的是 Description，依序是 Theme，Dialogue，最差的是 Narrative 類的文章。

有著劇情的文章類型，充滿著複雜的詞句，有時候是對話，有時候又有作者的伏筆，這類的文章充滿著各式各樣的詞、各式各樣虛構的人物，AlchemyAPI 就算找出了關鍵詞，但是那個詞卻不一定是最符合劇情的詞，圖文合在一起時無法搭配。這個現象也顯示了這個系統無法精準地處理這一類型的文章。而反觀沒有劇情的文章類型，因為整段文章反覆說明著固定幾個名詞，所以 AlchemyAPI 者出的關鍵詞很容易符合文章大意，以至於成果相當不錯。

3.2 使用 Mutual reinforcement 是否有提高搜尋圖片的準確率

本實驗目的是為了測試以 Mutual reinforcement 概念是否真的能增進搜尋圖片的準確度。我們使用 SIFT 進行計算圖片的相似度與重要程度。進行此實驗前，我們有幾個假設：

1. 從段落中找到的關鍵字是非常符合文意的。
2. Google 影像搜尋所找到的圖片，大部分都符合關鍵字。

以上述假設為前提下，我們設計一組實驗來證明我們的想法是否有效，進行的方式為：

1. 選擇數篇世界知名的童話故事，之所以這樣選是因為知名的童話故事，通常會有許多的插圖，甚至是改編成電影、動畫等等，這樣去進行實驗才能符合假設一，而且也容易比較出好壞。
2. 對同一個故事，分別以不使用 SIFT 與使用 SIFT 分析。
3. 再比較每個段落，比較兩張圖中那一張比較符合文意，比較符合的圖片給予 1 分，無法判斷則各給予 0.5 分，接著計算各個方式的平均分數 (總分數/頁數 * 100)。

表 2 各個故事的在不同模式的平均分數

故事名稱	不使用 SIFT	使用 SIFT
龜兔賽跑	100	0
三隻小豬	40	60
灰姑娘	57	43
美女與野獸	42	58
國王的新衣	50	50
白雪公主	72	28
糖果屋	69	31
總平均	**61.43**	**38.57**

註：

1. 各個故事的搜尋結果之連結
http://www.cs.ccu.edu.tw/~hchy97u/NGSPR/doc/Expriment2_Detail.php

從表 2 中的平均分數來說，不使用 SIFT 分析所獲得的分數較高，也就說明了我們使用 Mutual reinforcement 的概念無法有效提高搜尋圖片的準確度，反而降低了。我們從實作的工具、分析的步驟、被分析的圖片這幾點為出發點，思索失敗的原因，總結之後，認為有以下幾點可能是原因所在：

第一，假設搜尋的關鍵字都是段落中的重點，足以代表段落描述的情境。在搜尋回來的圖片中，的確有幾張是符合意境，但每張圖片都非常不一樣，以至於用相似度來判斷重要性就顯得沒有意義，有時反而讓結果顯得更糟糕。這跟我們當初的想法大相逕庭，原本預期會是有相似的圖片出現，但出現的卻是許多不同人物、背景，

意境相似的影像。這似乎也是有道理的，網路上的圖片來源可能是某電影的截圖、故事書的插畫、個人的作品等等，每樣作品對於同個情境的圖片應該只會有一張最具代表性的，不同的作品當然畫面就會不一樣，所以找到的圖片才會十分迥異。如果搜尋的是某一項特定的東西，而且該物體有特定的形狀，那麼用此方式應該會很成功，只是我們搜尋的是情境，沒有足夠明確的形體，加上剛剛提到的因素，而讓這個想法不是那麼的有用。

第二，即使找到的圖片，有某幾張十分接近，但使用 SIFT 尋找區域特徵點(local feature)時，在彩色、光影變化甚大的圖片下，會找到許多非預期的特徵點，於是進行比對的時候，許多非預期的特徵點就會影響正確的特徵點配對，導致沒辦法正確計算出相似度，而讓結果顯得不好。除此之外，SIFT 在許多的研究中，都是用來辨識某個特定物體在某個影像中的位置，而我們這次用來比對的是多個物體比對多個物體的圖片，所以 SIFT 找到區域特徵點是多個物體的集合，這樣就不知道到底哪些的特徵點是屬於同個物體。我們應該要將圖片中的物體切開再進行比對，然而要將物體從影像中取出，就目前的技術還沒有合適的辦法，使得這項工作沒辦法達到更好的地步。

第三，系統只能從圖色系與特殊類別來限制圖片的風格，然而就一個有圖故事書而言，除了各段落圖片的色彩、畫風要一致，圖片間的意境也要連貫，否則就不能充分表達故事的進行。但要系統去理解圖片該如何擺放適當的順序是一件困難的是，先要能從故事中的文字建立主要人事物的互動流程，再辨識圖片的各個物件是否存在那些故事中的元素，並加以推論該擺放在哪個段落，這需要大量的資料建立起一套推理的邏輯才有可能完成。在現在的系統中，還未實現這樣的功能。

綜合以上三點，讓系統找出適當的圖片、合理的排列順序困難重重，產生的成品也就差強人意。

4. 結論與未來目標

4.1 結論

系統的目標是讓電腦能替文章插入圖片，產生圖文並茂的作品。為了這個目標，我們以「找出各個段落的重點文字作為搜尋圖片的依據，在網路上搜尋並篩選圖片，在插入到各個段落中。」這個構想出發，開發出一個不用讓人為了找尋插圖而煩惱的編輯系統，讓電腦能去尋找網路上的資源給使用者，說不定使用者還能因此得到許多啟發，寫出更具創造力的作品。

由 3.1 的實驗我們發現這個系統很適合專門寫英文旅遊文章的部落客，因為遊記型文章圖片文義符合的情況相當好。現在的人喜歡旅遊，若是旅遊後想要撰寫文章卻沒有適當的照片貼圖時，此系統可輕鬆的幫忙使用者找出想要的圖片。而且使用此系統的人不需要懂 HTML 的程式碼，使用者們只要在我們的文字編輯區編輯好，透過產生原始碼的功能，即可簡單地複製至他們想要的網站頁面發表。

4.2 未來目標

目前系統功能還很陽春，未來會朝著方便性與實用性發展。針對方便性，我們希望能自行調整成品的版面配置，且能自動產生結果到各大部落格，讓使用者可以分享給他們的朋友們欣賞。而在實用性上，我們希望可以讓系統能從網路上尋找相關文章給使用者，讓使用者可以參考其他人的作品，改善自己的文章，或獲得更多資訊。甚至可以讓系統從網路上的文章學習，產生出作品給使用者評分，或許人們可以從這樣的角度，對自己熟悉的語言有新的認識也說不定。

致謝

The work was partially supported by the National Science Council of Taiwan, Republic of China under research contract NSC 100-2221-E-194-061.

參考文獻

[1] D. Joshi, J.Z. Wang, and J. Li, “The Story Picturing Engine – A System for Automatic Text Illustration,” ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 2, no. 1, pp. 68-89, 2006.

[2] OpenNLP, http://incubator.apache.org/opennlp/

[3] AlchemyAPI, http://www.alchemyapi.com/

[4] HTML Living Standard, HTML Drag Drop API, http://www.whatwg.org/specs/web-apps/current-work/multipage/dnd.html#dnd

[5] HTML5ROCKS, Native HTML5 Drag and Drop, http://www.html5rocks.com/en/tutorials/dnd/basics/#toc-introduction

[6] HTML5ROCKS, Reading local files in JavaScript, http://www.html5rocks.com/en/tutorials/file/dndfiles/#toc-introduction

[7] Google, Image Search Developer's Guide, http://code.google.com/intl/zh-TW/apis/imagesearch/v1/devguide.html

[8] Google, JSON Developer's Guide, http://code.google.com/intl/zh-TW/apis/imagesearch/v1/jsondevguide.html

[9] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[10] Rob Hess, SIFT library, http://blogs.oregonstate.edu/hess/code/sift/