基於對照表以及語言模型之簡繁字體轉換

Chinese Characters Conversion System based on Lookup Table and Language Model

李民祥 Min-Hsiang Li, 吳世弘 Shih-Hung Wu
朝陽科技大學資訊工程系
Department of Computer Science and Information Engineering
Chaoyang University of Technology
{s9827608, shwu}@cyut.edu.tw

楊秉哲 Ping-che Yang, 谷圳 Tsun Ku
資訊工業策進會
Institute for information industry
{maciaclark, cujing}@iii.org.tw

摘要

中國大陸與台灣的文字同屬於華文字體，但字體上卻分爲簡體字與繁體字。中國大陸與台灣近年來在中文書籍及網路上皆有大量的資訊交流。基於閱讀習慣，文字勢必需要執行簡繁轉換後才利於雙方的讀者閱讀。傳統的簡繁轉換擁有簡體一字對繁體多字的歧異問題以及兩岸用語不同的問題。因此，本研究設計一個具有擴展性的簡繁轉換系統，透過擷取維基百科新增對照表內容來改善兩岸用語不同的問題，以及使用語言模型改善簡體字一個字對繁體字多個字的歧異問題。此系統可以降低各種中文電子書籍執行簡繁轉換後人工校正的成本。具有彈性的架構使得系統可以持續擴充改進。

關鍵詞：簡繁轉換，語言模型，維基百科，對照表

Abstract

The character sets used in China and Taiwan are both Chinese, but they are divided into simplified and traditional Chinese characters. There are large amount of information exchange between China and Taiwan through books and Internet. To provide readers a convenient reading environment, the character conversion between simplified and traditional Chinese is necessary. The conversion between simplified and traditional Chinese characters has two problems: one-to-many ambiguity and term usage problems. Since there are many traditional Chinese characters that have only one corresponding simplified character, when converting simplified Chinese into traditional Chinese, the system will face the one-to-many ambiguity. Also, there are many terms that have different usages between the two Chinese societies. This paper focus on designing an extensible conversion system, that can take the advantage of community knowledge by accumulating lookup tables through Wikipedia to tackle the term usage problem and can integrate language model to disambiguate the one-to-many ambiguity. The system can reduce the cost of proofreading of character conversion for books, e-books, or online publications. The extensible architecture makes it easy to improve the system with new training data.

Keywords: Chinese character conversion, Language model, Wikipedia, Lookup table.

Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing (ROCLING 2010), Pages 113-127, Puli, Nantou,Taiwan, September 2010.

一、緒論

由於中國大陸與台灣數位出版合作的開啓，中文書籍相互流通的機會增加，簡體字與繁體字的轉換技術開始變得重要。近年來，隨著台灣與中國大陸兩岸交流逐漸頻繁，以及網路資訊快速發展，文字書信已經成爲兩岸溝通的媒介之一。然而，由於中國大陸普遍使用簡體中文，台灣主要使用繁體中文，因此雙方在文字溝通上勢必要先經過簡繁轉換的程序後才利於閱讀。一般來說，簡繁轉換是依據簡繁字對照表來進行轉換，這個方法主要使用單字詞一對一的方式進行簡繁轉換。不過在許多情況下，簡體字對應繁體字經常是一對多的狀況，所以僅使用一對一的方式進行轉換常常會出現字詞不適用的狀況，稱此爲「一對多簡繁字」。

中國大陸以及台灣已著手研究簡繁轉換工具的有：中國大陸的中國科學院軟體所、四通利方資訊技術有限公司、新天地公司；台灣的 IBM 公司、倚天資訊股份公司，以及其它研發團隊等。目前也有許多文書處理軟體包含著內建的簡繁轉換系統，例如：Microsoft Office、Sun 的 OpenOffice；以及網路上可查詢到的雙語字典，例如：Google Translate。然而，這些轉換的結果通常參差不齊，簡繁轉換後依然需要依靠人工來校正不精確轉換的錯誤[1]。根據文獻，王寧擷取了 150 萬字的小說簡體字語料，使用 Office Word2003 執行簡體轉換繁體的功能，發現許多簡體字對繁體字一對多的情況無法正確被轉換[2]。簡而言之，簡繁轉換的困難在於簡體字存在著一對多簡繁字的情況，使得在不同名詞或是動詞搭配時，無法正確轉換出應該對應到的字，例如：簡體字的「下面」在敘述位置時，轉換爲繁體字後爲「下面」；而簡體字的「下面」使用在動詞時，轉換爲繁體字後則是「下麵」[3]。並且，簡繁用詞問題需要依靠蒐集大量的簡體以及繁體用詞的對照表來提供轉換，例如「坐公車」互相對應「坐公交車」。

本論文提供以繁體字語料庫建構的語言模型以及收集維基百科簡繁詞彙對照表，並計算語言模型的分數來達到提升簡體字轉換繁體字正確性的方法。例如：繁體的「坐公車」轉換爲簡體的「坐公交車」，簡體的「吃面」轉換爲繁體的「吃麵」而非「吃面」。實驗部分，由於繁體字轉換簡體字爲多對一的問題，僅需查表即可完成轉換。所以我們著重於簡體字轉換繁體字時一對多簡繁字的選擇辦法。我們以包含多種一對多簡繁字轉換的常用字的句子進行簡體字轉換繁體字的測試，並且與目前幾種知名的翻譯工具進行比較，接著分析本系統轉換錯誤的問題。接著，我們引入斷詞系統[11]，改善原本系統無法轉換正確的幾種情況，並且分析無法正確轉換的一對多簡繁字。最後，透過調整語言模型以及對照表的大小來驗證語言模型以及對照表大小對於簡繁轉換正確率是否有相對的影響。

二、系統設計與方法

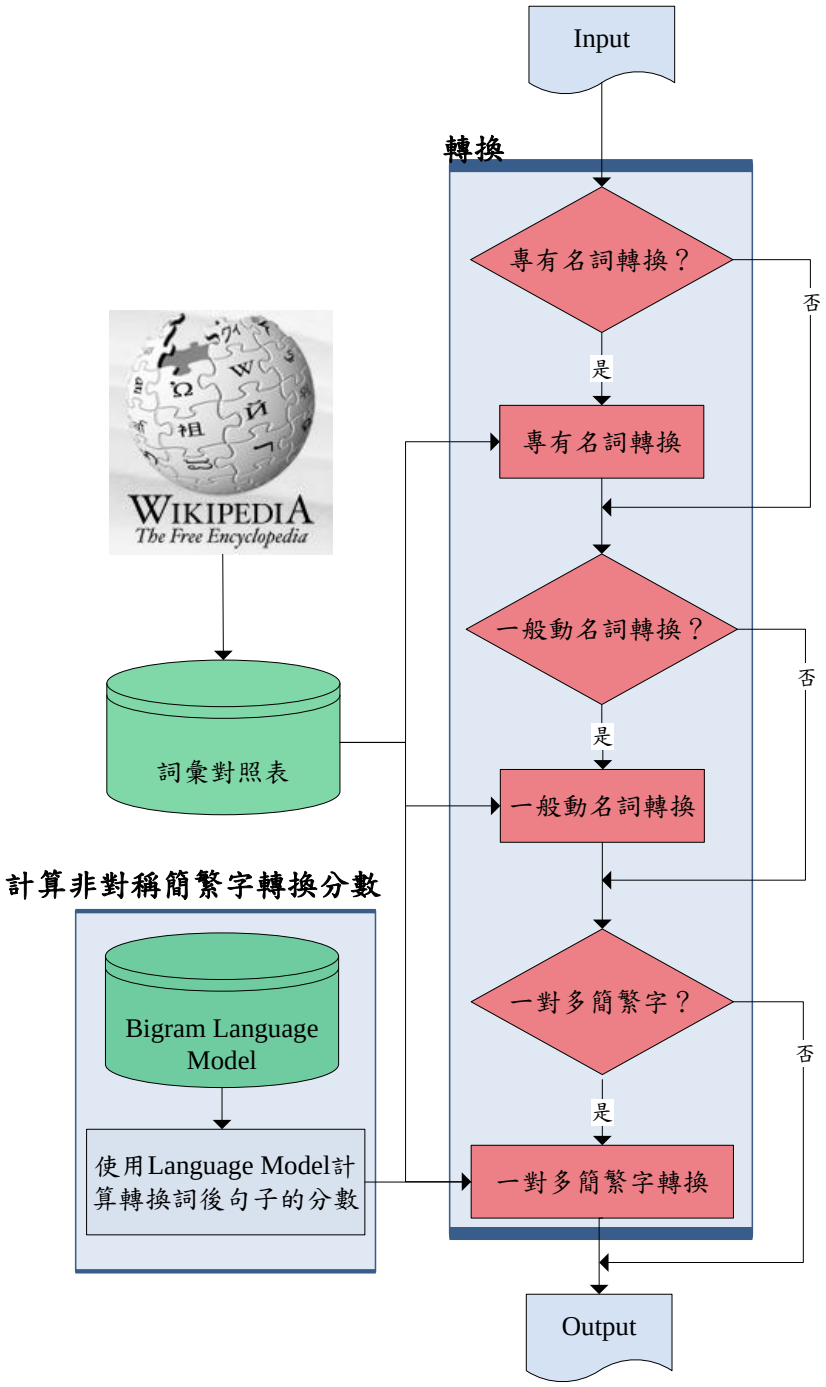
（一）、系統流程設計

本篇論文的系統中，簡繁轉換的文字編碼皆使用 Unicode 的編碼方式，因爲 Unicode 爲國際編碼，它給予每一個字符唯一的編碼表示，並且包含了現有規範中所有簡體字與繁體字的日常用字。所以使用 Unicode 可以省去繁體字編碼(BIG5)與簡體字編碼(GB)的轉換步驟。

轉換過程分爲三個步驟：一、首先使用對照表判斷是否需要進行專有名詞以及一般

名詞轉換。二、判斷是否含有一對多簡繁字。三、使用語言模型計算分數。

對照表內容以維基百科簡體字與繁體字條目名稱做爲專有名詞以及一般名詞的對照表、以及維基百科提供的簡繁轉換一對多簡繁字。一對多簡繁字爲簡體字轉繁體字一對多的狀況，例如：簡體字的「皇后」轉換爲繁體字有兩種可能，分別爲「皇后」以及「皇後」。因爲簡體字的「后」對應的繁體字爲「后」以及「後」。最後，我們蒐集 1998 年至 2005 年的新聞語料庫做爲我們的建構語言模型的語料庫，並使用語言模型計算簡體字轉換繁體字時出現一對多簡繁字的分數。系統流程圖如圖一所示：



圖一、系統流程圖

（二）、對照表收集

中國大陸以及台灣同屬華文市場，但書籍內容用字遣詞仍有很多差異。智慧型的文體轉換，必須要解決編碼、詞彙以及簡體字對繁體字轉換時一對多及多對一的問題。兩岸不同詞彙的比對和轉換，包括人名地名組織名以及領域專有名詞[4]。目前的技術多只著重編碼的轉換，以及專有名詞的轉換。因此，對照表的內容需要豐富的轉換對照，包括成語、中外人名、地名、組織名。

對照表部份，由於維基百科擁有大量的條目資料，並且提供了對於一般名詞以及專業名詞的準確度，Martin Hepp [5]提到，維基百科中有 92.67%的條目名稱即使過了一段時間後，條目名稱依然沒有改變，有 6.67%是改變了名稱，但語義上保持不變，僅剩下的0.67%爲可被刪除的條目。因此，維基百科中所有條目名稱中有 99.34%的條目名稱是可以被信任的。基於這個理由，專有名詞轉換的部份我們主要依靠維基百科做爲轉換的輸出。

維基百科提供一對多簡繁字以及一對一單字詞的轉換對應字。因此我們以人工方式蒐集維基百科的簡體字與繁體字的一對多簡繁字以及一對一單字詞，用來做爲簡體字與繁體字相互轉換時所依據的來源。繁體字與簡體字用詞對照部份，一共收集 7180 筆對照詞彙；一對多簡繁字以及一對一單字詞一共收集 6619 筆，其中一對多簡繁字一共有 475 筆，一對一單字詞爲 6144 筆。表一、表二以及表三分別爲部分對照表內容、部分一對多簡繁字以及部分簡繁轉換一對一單字詞的範例。

表一、部分對照表內容

	繁體字用詞	簡體字用詞
簡繁用詞對照表	快閃記憶體	闪存
	網際網路	因特网
	解碼	译码
	印表機	打印机
	埠	端口
	蟻后	蚁后

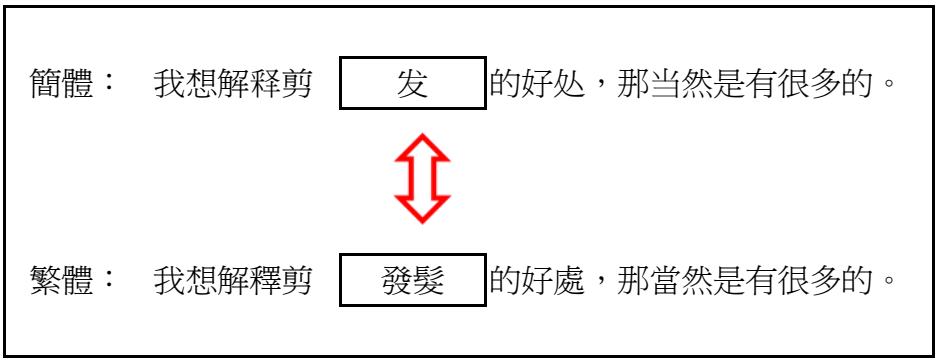
表二、部分一對多簡繁字

	繁體字單字詞	簡體字單字詞
一對多簡繁字	板闆	板
	辟闢	辟
	表錶	表
	發髮	发
	并並併竝	并
	乾干幹榦	干
	面麵麪麺	面

表三、部分簡繁轉換一對一單字詞

	繁體字單字詞	簡體字單字詞
簡繁轉換一對一單字詞	獃	呆
	僱	雇
	韓	韩
	號	号
	輓	挽
	兩	两
	嚴	严

如圖二所示，上面的句子爲簡體中文，下面的句子爲將轉換的繁體中文。透過對照表的簡繁用詞轉換以及一對一轉換，可以精確的轉換出正確用詞。但是，一對多簡繁字的對照表僅提供可能轉換的字詞，並沒有提供如何正確轉換一對多簡繁字。



圖二、一對多簡繁字的轉換問題

因此，我們使用語言模型計算一對多簡繁字 bigram 以及 unigram 的機率值，取得圖二例子中「剪發」、「發的」以及「剪髮」、「髮的」出現機率較高的 bigram 機率值，藉由較高 bigram 機率值來做爲選擇字的轉換方式。

（三）、語言模型

我們使用統計式語言模型的方法（Statistical language model）[6]，篩選出正確性較高的翻譯方式[8][12]。系統使用N-gram 語言模型計算一個句子中字詞組合的機率，機率越高代表越可能符合正確文法，反之則代表可能越不符合正確文法。首先建立語言模型，我們使用Maximum Likelihood Estimation (MLE) [7]，計算出語料庫中每個字出現的相對頻率並且藉此計算機率值，如公式(1)所示：

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})} \quad (1)$$

其中，C代表某個字W出現的頻率。由於一個句子是由n個字組成，因此一個句子的機率可以計算爲如公式(2)所示：

$$P(w_1^n) \equiv P(w_1, w_2, ..., w_n) \quad (2)$$

其中 w_n 表示句子中第 n 個字。$P(w_1^n)$ 表示 1 到 n 個字出現的機率值。

假設字詞的機率爲獨立事件，一個句子條件機率可由連乘得到，如公式(3)所示：

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)\times\ldots\times P(w_n \mid w_1^{n-1}) \\ &= P(w_1)\prod_{k=2}^{n} P(w_k \mid w_1^{k-1)}) \end{aligned} \quad (3)$$

然而，組成一個句子的字詞是有限的，無法由過去歷史出現的無限字來做預測，因此我們將公式(3)改寫爲如公式(4)所示：

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1}) \quad (4)$$

代表依據前(n-1)個字出現的機率來預測目前第 n 個字所出現的機率，而所謂的 N-gram 就是當 N=2 時，稱爲 bigram，如公式(4)所示：

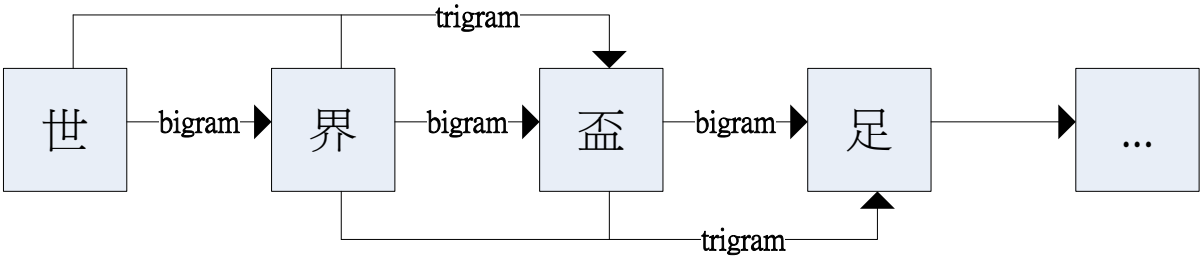
$$P(w_n \mid w_{n-1}) \quad (5)$$

當 N=3 時，稱爲 trigram，如公式(5)所示：

$$P(w_n \mid w_{n-1}w_{n-2}) \quad (6)$$

依此類推至 N-gram。

如圖三舉例，利用前兩個字出現的情況下，預測下一個字出現的機率。如圖三所示：此圖舉例說明以「世」 與「界」爲例：由「世」出現的情況下來推測「界」出現的機率，稱爲 bigram。同理，「界」與「盃」也是 bigram；若是由「世」與「界」出現的情況下來推測「盃」出現的機率，則稱爲 trigram。然而，建構 trigram 的語言模型會造成語言模型的內容龐大，造成系統速度降低。因此，我們利用中文字出現二字詞的比例很高的特性，本研究使用的語言模型爲計算 bigram 的出現頻率。



圖三、舉例說明 bigram 以及 trigram

（四）、Smoothing

然而，MLE在字詞出現頻率正常的情況下可以運作良好，但由於訓練資料稀疏，有些字詞出現的頻率會很低甚至是零，因此語言模型計算分數時可能會發生找不到要計算的字詞，導致無法正確預測下一個字的錯誤狀況，使得正確率降低。頻率是零的情況有兩種，一種是代表兩個字之間無意義的結合，也就是真正的零；另一種是假的零，意

思就是雖然這個字在文集中沒出現過，但是卻是真實世界中存在的字詞，只是訓練語料裡沒有出現。爲了避免出現機率相乘後爲零的狀況，我們使用Smoothing的方法，分配bigram以及unigram出現機率的權重值，以bigram的機率爲主要的機率分配，並給予較高的權重值；而unigram則給予較小的權重值，因爲unigram提供的線索比起bigram提供的線索所含有的資訊來得較低。

Smoothing的方法可分成折扣的方法和模型結合的方法，折扣的方法就是調整機率，將機率較高者分配其值給機率爲零者；而模型結合的方式就是利用內插法和補插法，當trigram無效時，使用bigram，bigram無效時則使用unigram。本系統的Smoothing爲模型結合的方法。我們會使用Interpolated Kneser-Ney smoothing的演算法[9]。Interpolated Kneser-Ney smoothing公式如公式(7)所示：

$$\begin{aligned} &P_{\text{interpolate}}(w \mid w_{i-1}w_{i-2}) = \\ &\lambda P_{\text{trigram}}(w \mid w_{i-1}w_{i-2}) + (1-\lambda)[\mu P_{\text{bigram}}(w \mid w_{i-1}) + (1-\mu)P_{\text{unigram}}(w)] \end{aligned} \tag{7}$$

其中 λ 以及 μ 表示分配的權重值。分別計算 trigram、bigram 以及 unigram 的機率值，並給予權重分配的 λ 和 μ ，藉以避免發生 trigram 或是 bigram 的機率值爲零的狀況。而我們主要是使用 bigram 語言模型，因此本系統使用 Interpolated Kneser-Ney smoothing 的公式時需要稍作修改，我們將公式(7)改寫爲：

$$P_{\text{interpolate}}(w \mid w_{i-1}) = \lambda P_{\text{bigram}}(w \mid w_{i-1}) + (1-\lambda)P_{\text{unigram}}(w) \tag{8}$$

我們刪除沒有使用到的 trigram，只計算 bigram 以及 unigram 的機率值。其中，我們認爲 bigram 的頻率的資訊強度大於 unigram 的資訊強度，因此 λ 設定爲 0.9。

（五）、Entropy 以及 Modified Entropy

評估的內容中有一項很重要的標準─Entropy，它被廣泛的使用在測量資訊上面[10]。其定義爲下列公式(9)：

$$H(X) = -\sum_{x \in T} P(x) \log_2 P(x) \tag{9}$$

其中隨機變數 X 涵蓋的範圍包含可預測的 T 集合(例如字母，字詞或部分的語音)，這裡表示一對多簡繁字合併一對多簡繁字前後單字詞的字串。$P(x)$ 爲 MLE 所計算出來的機率值，x 表示 X 的 bigram。因爲只需要取得機率連乘後的最大值，所以我們減少公式(9)的計算量，加快計算時間。我們實際計算時使用公式(9)的 Entropy 改寫後的公式(10) Modified Entropy。

$$H'(X) = -\sum_{x \in T} \log_{10} P(x) \tag{10}$$

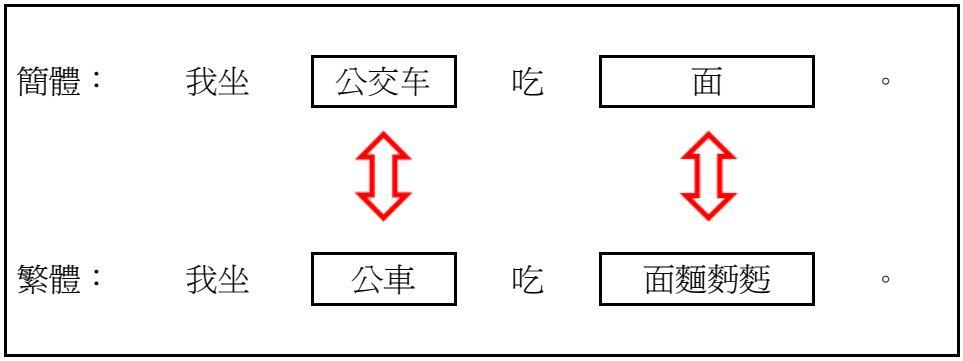
三、實驗結果與分析

由於統計模板頻率需要大量的語料資料，因此我們蒐集新聞語料庫做爲我們的語料庫。語料庫的整理如表四所示：

表四、語料庫資料整理

資料來源	年份	新聞社	文件數	檔案大小
新聞語料庫	1998-1999	China times	38,163	209MB
		China times Commercial	25,812	
		China times Express	5,747	
		Central Daily News	27,770	
		China Daily News	34,728	
	1998-1999	United Daily News	249,508	320MB
	2000-2001	United Daily News	172,421	1.03GB
		United Express	91,958	
		Ming Hseng News	168,807	
		Economic Daily News	463,873	

對照表部份，我們使用維基百科提供的一對多簡繁字對照表，其中一對多簡繁字的數量爲 475 筆，當系統判斷要轉換的字爲一對多簡繁字時使用語言模型進行 bigram 的計算，選擇出分數最佳的對應字；如果系統判斷要轉換的字爲一對一單字詞時，則使用維基百科提供的 6144 筆一對一單字詞對照表直接進行轉換。系統的輸入以及輸出皆使用 Unicode 編碼的文字。我們也從維基百科中抽取了 7180 筆簡繁用詞對照表，使用方式如前述。如圖四所示：先判斷句中是否含有專有名詞以及一般動名詞，如果有則先轉換，否則進行一對一單字詞轉換以及一對多簡繁字轉換。如圖四中出現可以對應的一般名詞的轉換，「公車」相互對應「公交车」；接著，判斷句中是否含有一對多簡繁字，如果有則使用語言模型計算出最佳的對應字，否則直接進行轉換。如圖四中出現可以直接轉換的「時」與「时」，以及需要使用語言模型進行計算分數的「麵」與「面、麵、麪、麪」。



圖四、系統轉換的例子

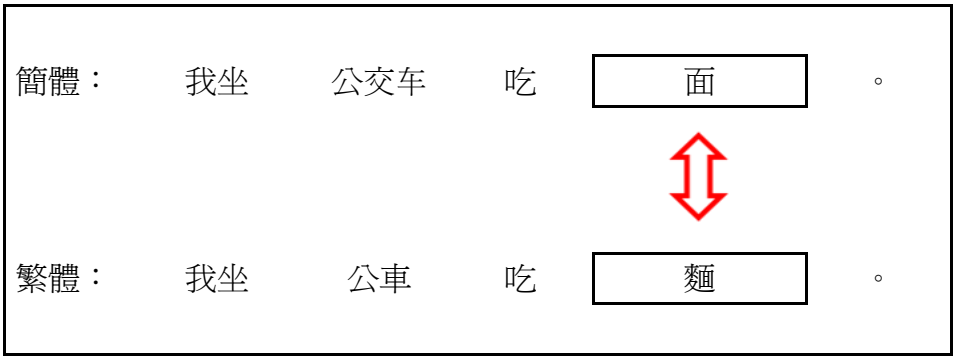
語言模型計算分數部分，我們計算當簡體字轉換爲繁體字需要選擇一對多簡繁字時候的分數，使用前述的 Modified Entropy 做爲最後計算的分數。Modified Entropy 越低代表該字的組合機率越高。因此，我們選擇計算 Modified Entropy 最低的組合。表五爲

圖四中一對多簡繁字的語言模型分數計算例子。我們計算出「吃面」與「面。」、「吃麵」與「麵。」、「吃麪」與「麪。」以及「吃麪」與「麪。」等四種一對多簡繁字組合在語言模型中出現的機率相乘的 Modified Entropy 分數。其中，「麪」以及「麪」在訓練的語料庫當中出現次數皆爲 0，所以只會計算「吃」以及「。」的 unigram 機率，才會造成「吃麪。」以及「吃麪。」Modified Entropy 分數相同的狀況。

表五、一對多簡繁字的 Modified Entropy 計算分數

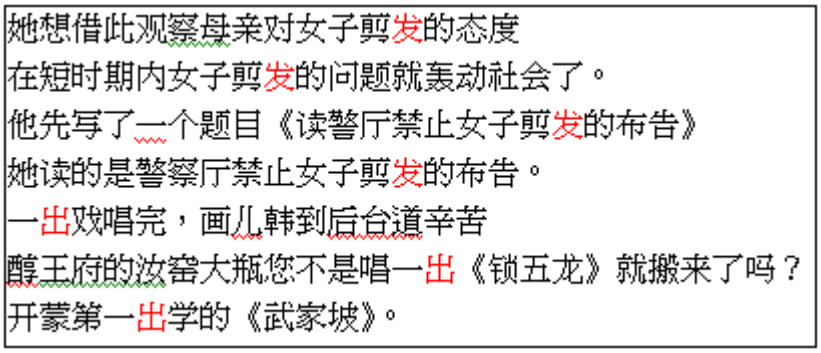
簡體=>繁體	Modified Entropy
吃面。=>吃面。	18.164046939
吃面。=>吃麵。	12.016282836
吃面。=>吃麪。	62.00000001
吃面。=>吃麪。	62.00000001

圖四最後轉換的結果如圖五所示：



圖五、轉換結果

簡繁轉換大部分的問題是出在一對多簡繁字的問題上，因此我們的測試集主要針對句中包含一對多簡繁字的簡體句子進行簡體轉換繁體的測試。然而，各個領域皆有其適用的簡繁轉換對照，這部份透過收集大量的對照表即可正確轉換。所以，我們使用王寧[2]提供的 271 句包含一對多簡繁字的簡體中文小說句子進行測試。因爲小說使用的文字多爲一般讀者較常接觸的一般動名詞，因此可以較準確的評估我們系統的正確性。圖六爲我們進行測試的部分資料，表六爲測試集的資料整理。其中，一對一單字詞爲僅有一種可能的轉換結果，因此我們主要評估一對多簡繁字轉換的正確與否。評估部分，我們使用 Accuracy，如公式(11)所定義。



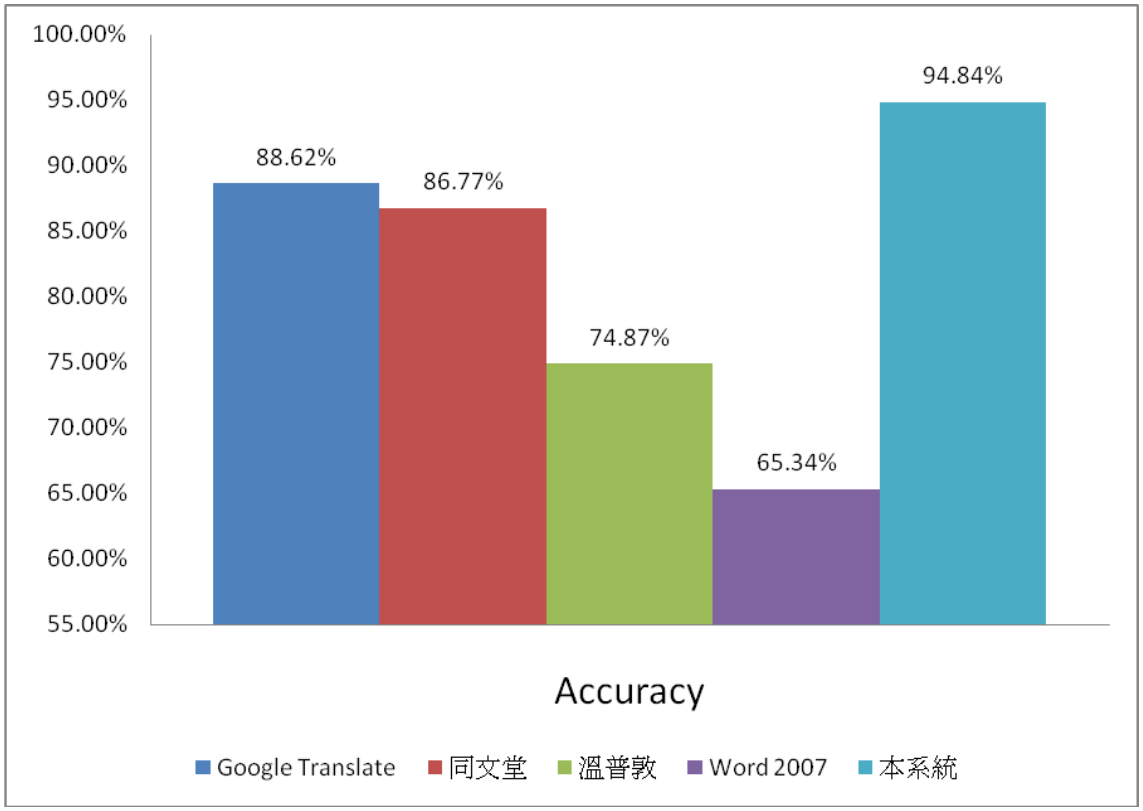
圖六、部分測試集

表六、測試集資料整理

一對多簡繁字字數	一對一單字詞字數
756	4418

$$\text{Accuracy} = \frac{tc}{W} \times 100\% \qquad (11)$$

tc 表示正確轉換一對多簡繁字的字數，*W* 表示一對多簡繁字的字數。我們搜尋過去文獻以及目前市面上的簡繁轉換方式，並未發現同樣使用語言模型的轉換方式。目前轉換品質較佳的系統如 Google Translate[13]、Microsoft Word 2007[14]、溫普敦[15]、同文堂[16]等四種知名的翻譯軟體。圖七爲這四種系統與我們系統的比較。



圖七、與其它系統比較的結果

圖七的實驗結果顯示，我們的系統對於簡繁轉換的效果比其它幾種效果來得好。因此，我們找出未被成功轉換的一對多簡繁字，如表七所示。其中帶有底線的爲轉換錯誤的字。

表七、部份轉換錯誤的句子

轉換錯誤的句子
已經幹了的道路
這是從前麵茶棚裡留聲機上放出來的
外麵糊了紙
現在他剛從六百裡外的煤礦回來
她摸出表來看
但是她依舊昂然自得地畫動槳
好一出大悲劇

接著，我們針對錯誤的部分，以手動方式蒐尋可能造成錯誤的對照表內容，發現對照表中含有容易因爲斷詞不佳時會造成轉換錯誤的對照詞彙，如表八所示：

表八、斷詞不佳時容易造成轉換錯誤的對照詞彙

繁體用詞	幹了	麵茶	麵糊	裡外
簡體用詞	干了	面茶	面糊	里外

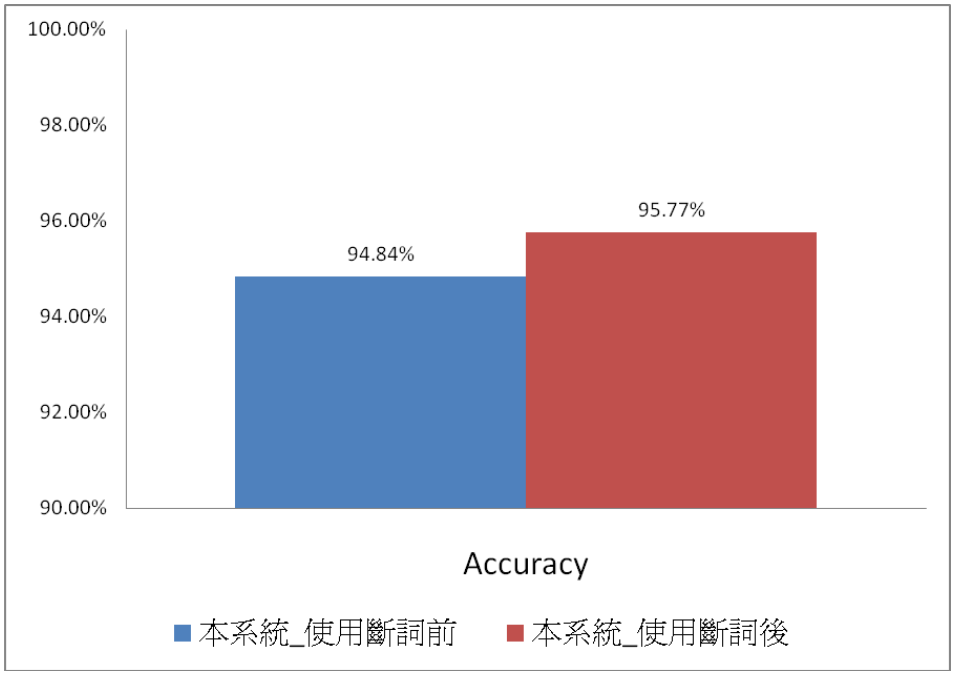
因爲它們包含了一對多簡繁字的單字詞，因此簡體字的「干了」並非只能轉換爲如對照表中繁體字的「幹了」，而是可以轉換爲「乾了」或是「幹了」。；簡體字的「面茶」並非只能轉換爲繁體字的「麵茶」。這是因爲簡體字的「面」在繁體字時經常使用在「裡面」、「外面」、「上面」等詞彙。但是，簡體字的「面」與後一個字成詞時則成爲「麵茶」、「麵糊」等詞彙；同理，其它容易因爲斷詞不佳而造成轉換錯誤的對照表內容也是一樣的狀況。因此，斷詞的正確與否，對於簡繁轉換有著絕對的影響力。所以我們進一步引入中國科學院開發的簡體字斷詞系統[11]，嘗試改善上述的問題。我們將斷詞後不成詞的連續單字詞合併，避免文字資訊遺失。如圖八所示，經由合併連續單字詞的步驟可以保留原有的文字資訊，使得圖八例子中簡體字的「看表」可以找到對照表中繁體字的「看錶」。

圖八、合併斷詞後的連續單字詞

引入斷詞系統後，表七所示的幾種狀況可獲得改善。例如簡體字的「前面茶棚」應該爲繁體字的「前面」以及「茶棚」，但未引用斷詞的系統會因爲對照表中含有簡體字「面茶」對應至繁體字「麵茶」的關係，造成簡體字的「前面茶棚」轉換爲繁體字的「前麵茶棚」的錯誤結果；引入斷詞系統後可以正確斷出「前面」以及「茶棚」，使得轉換結果正確。再次執行實驗後，其結果如圖九所示。

實驗結果顯示，引入斷詞系統雖然可以改善系統效能，但成效不大。因此，我們關心剩下沒被成功轉換的一對多簡繁字的類型。我們發現主要的錯誤轉換爲要轉換爲「錶」卻轉換爲「表」、要轉換爲「划」卻轉換爲「劃」、要轉換爲「齣」卻轉換爲「出」。因此，我們找出這些一對多簡繁字被錯誤轉換時，使用語言模型計算分數的 bigram 組合字以及其句子。表九爲錯誤轉換的類型以及被轉換錯誤的 bigram 組合字的 Modified Entropy 分數，其中帶有底線的爲轉換錯誤的字。表九中錯誤轉換的一對多簡繁字是因爲進行計算的 bigram 在語言模型的機率低於被轉換的 bigram 的機率，因而轉換爲不正確的字。然而，語言模型中所有的 bigram 皆由 unigram 組合起來。因此，unigram 頻率較高的字，自然會擁有較多的 bigram。基於這個理由，我們找出「表」、「出」、「劃」、「錶」、「齣」、「划」等六個主要被錯誤轉換的 unigram 頻率。我們發現，由於「表」、「出」、「劃」在語言模型中 unigram 的頻率皆爲「錶」、「齣」、「划」的一百倍以上，造成大多數的 bigram 皆由 unigram 頻率高的那方組成，使得「錶」、「齣」、「划」擁有較少可以依據的 bigram

頻率資訊來做爲能夠被正確轉換的 bigram 頻率。至此，我們透過對照表、語言模型以及加入斷詞後的系統，仍有無法解決的一對多簡繁字類型，這些類型是我們認爲困難的問題。過度頻率如表十所示。



圖九、第二次實驗的結果

表九、部分錯誤轉換的類型

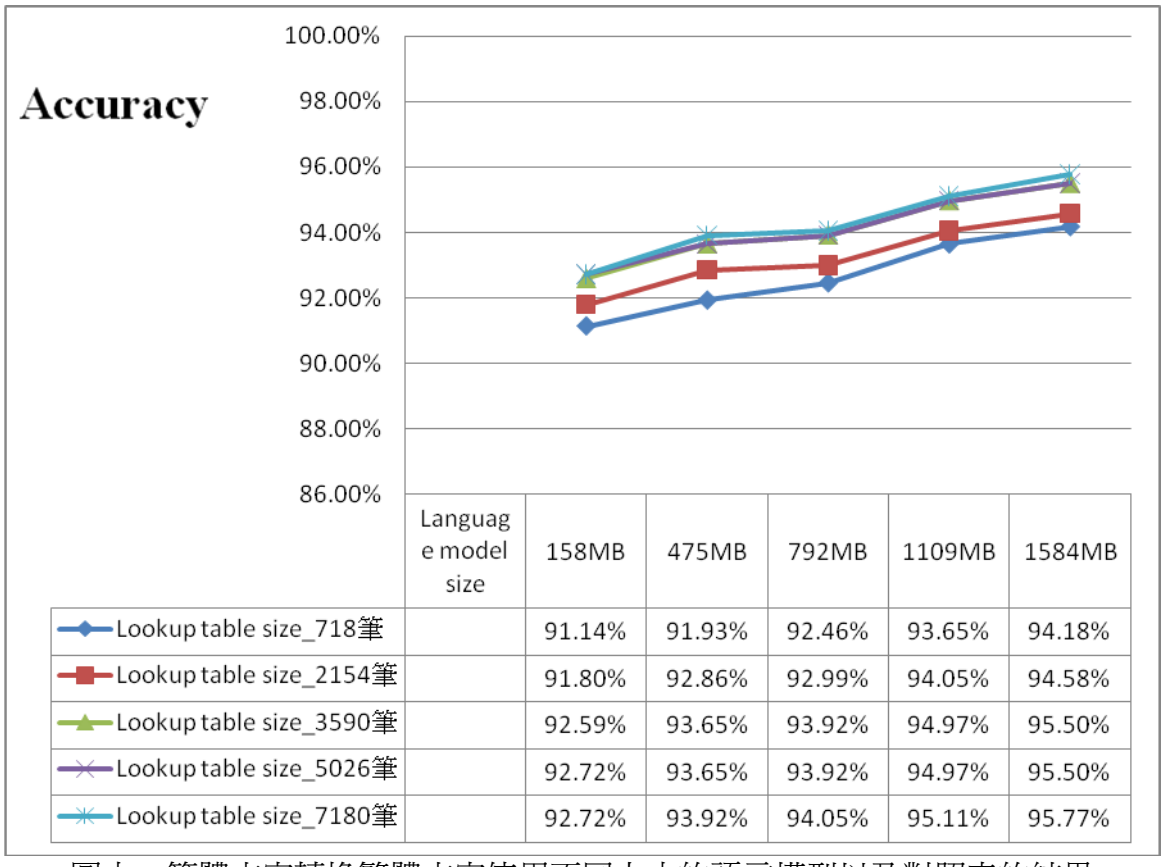
轉換錯誤的句子	計算的組合字	Modified Entropy 分數
醇王府的汝窯大瓶您不是唱一出《鎖五龍》就搬來了嗎？	一出《	9.949089426
	一齣《	12.5808102
開蒙第一出學的《武家坡》。	一出學	10.672644324
	一齣學	14.336685057
佩珠打算回去，她摸出表來看，快到拾二點鐘了	摸出表來	18.568075432
	摸出錶來	23.575641152
	摸齣表來	45.15278956
	摸齣錶來	69.91470532
然後自己坐到船尾，把住槳慢慢地劃起來。	地划起	14.946993213
	地劃起	13.737994911

表十、過度的頻率比較

頻率比較	出	齣
	0.004183773	0.000012075
	表	錶
	0.002347138	0.000006012
	劃	划
	0.000268397	0.000015572

最後，我們使用語料庫的 10%(158MB)、30%(475MB)、50%(792MB)、70%(1109MB)以及 100%(1584MB)大小來建構語言模型，以及將對照表的大小分爲 10%(718 筆)、30%(2154 筆)、50%(3590 筆)、70%(5026 筆)以及 100%(7180 筆)。利用不同的語言模型大小以及對照表大小來判斷是否會影響簡繁轉換的準確性。(請注意，本次實驗語言模型以及對照表的大小，僅是本研究使用新聞語料庫以及維基百科對照表所建構的資料。其它研究人員可以使用自行建構的資料。) 評估方式如公式(11)定義的 Accuracy。如圖七所示：橫軸爲使用不同的語言模型大小，每一條線分別代表使用不同的對照表大小，縱軸爲 Accuracy。

從圖十中我們可以看出，簡繁轉換的 Accuracy 隨著語言模型以及對照表使用的數量越來越大時，Accuracy 也越來越高。使用我們系統建構的 1584MB 的語言模型大小以及 7180 筆的對照表大小時 Accuracy 可達到 95.77%。其中我們注意到，當對照表大小從 50%開始，對照表對於 Accuracy 的提升較無語言模型大小 158MB 以及 475MB 時來得顯著。這是因爲我們的測試集主要針對一對多簡繁字進行轉換的測試，大部分句子沒有包含需要簡繁用詞轉換的專有名詞以及一般名詞。劉匯丹[4]提到，一個好的簡繁轉換系統必須要有足夠的知識，方能轉換出正確的詞彙。意思是說，因爲中國大陸與台灣因爲文化關係，許多專有名詞以及一般動名詞使用不同名詞但是意思相同的詞彙，例如先前提到的「公車」互相對應「公交車」。因此需要大量的對照表來提供應該正確轉換的詞彙。語言模型大小部分，因爲測試集主要針對一對多簡繁字進行轉換的測試，因此當語言模型越大時，我們可以看出 Accuracy 有顯著的提升。



Languag e model size	158MB	475MB	792MB	1109MB	1584MB
Lookup table size_718筆	91.14%	91.93%	92.46%	93.65%	94.18%
Lookup table size_2154筆	91.80%	92.86%	92.99%	94.05%	94.58%
Lookup table size_3590筆	92.59%	93.65%	93.92%	94.97%	95.50%
Lookup table size_5026筆	92.72%	93.65%	93.92%	94.97%	95.50%
Lookup table size_7180筆	92.72%	93.92%	94.05%	95.11%	95.77%

圖十、簡體文字轉換繁體文字使用不同大小的語言模型以及對照表的結果

四、結論

本篇論文的研究主要是改善傳統簡繁轉換僅執行一對一編碼轉換，而沒有考慮一對多簡繁字的問題，造成一對多簡繁字一直無法有效的被正確轉換。因此，以實驗的測試集爲例，測試集包含的一對多簡繁字爲日常用字佔測試集中所有字數約 15%，我們的系統可以將這 15%的一對多簡繁字執行 94.84%的正確轉換。第二次實驗加入了斷詞後僅能夠提高約 1%的正確率，並且餘下轉換錯誤的類型是我們認爲困難的轉換字，需要倚靠其它方法來解決。

實驗部分我們調整語言模型以及對照表的大小來測試是否對於簡繁轉換的效能有影響。從實驗結果來看，語言模型數量越大對於轉換結果有正向幫助，但是如果語言模型數量過大，卻會影響系統轉換的速度。這也是本系統建構語言模型時只考慮 bigram 分數的原因，因爲建構 trigram 會使得語言模型數量過大，造成轉換速度下降。對照表部分，由於中國大陸與台灣有許多用詞不同的狀況，因此需要大量的對照表提供正確轉換的詞彙。但是對照表數量過於龐大，也會造成系統轉換速度下降。因此，對照表的建構可以針對特定領域蒐集對照的詞彙，例如醫學領域的對照詞彙、資訊科學的對照詞彙…等，如此一來，針對需要轉換的用詞領域來蒐集對照表，減去不必要的資訊儲存於對照表中，避免對照表數量過大的情況。

本研究提供的方式可以讓其它研究人員以及使用者自行選擇建構語言模型的大小以及語料庫，對照表也能夠讓各人員自行選擇想要使用的對照表。因此本研究具有彈性的架構使得系統可以持續擴充改進。在未來，我們將著手建構簡體中文以及繁體中文的平行語料庫，利用簡體中文以及繁體中文的文法幾乎相同的特性，使用一些找尋新詞的方式，嘗試找出繁體中文內被判斷爲新詞的詞彙，但是簡體中文對列句子的相同位置卻沒有發現可能是名詞的詞彙，接著利用繁體中文句子中新詞的上下字爲線索，找尋出簡體中文對列句子中可能爲對應詞彙的新詞。最後，將發現的新詞加入系統的對照表中，藉以自動擴展對照表的內容。

致謝

This study is conducted under the "Digital Convergence Service Open Platform Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

參考文獻

[1] 王曉明, 魏林梅, "談簡繁轉換的幾個關鍵問題", 5TH CDF 研討會數位社群雙效（CD2E）, 2008 年 12 月 24 日.

[2] 王寧, 王曉明, "兩岸四地漢字的轉換與溝通", 第三屆兩岸四地中文數位化合作論壇, 2005 年 10 月.

[3] 李樹德, "Word“中文簡繁轉換”存在的問題與解決對策", http://www.yywzw.com/show.aspx?id=1570&cid=142.

[4] 劉匯丹,吳健, "基於詞語消歧的分層次漢字簡繁轉換系統", 5TH CDF 研討會數位社群雙效（CD2E）, 2008 年 12 月 24 日.

[5] Martin Hepp, Katharina Siorpaes, Daniel Bachlechner, "Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management," IEEE Internet Computing, vol. 11, no. 5, pp. 54-65, Sep./Oct. 2007.

[6] Ronald Rosenfeld, “Adaptive Statistical Language Modeling: a Maximum Entropy Approach” Ph.D. Thesis Proposal, Carnegie Mellon University, September 1992.

[7] Slavam. Katz, “Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer ”, IEEE Transactions on ACOUSTICS, SPEECH, and SIGNAL PROCESSING, VOL. ASSP-35, NO. 3, MARCH 1987, pp 400-401.

[8] 陳勇志, 吳世弘, 盧家慶, 谷圳, "中文混淆字集應用於別字偵錯模板自動產生", 第二十一屆自然語言與語音處理研討會, 2009 年 9 月.

[9] J. Goodman, "A Bit of Progress in Language Modeling, Extended Version," Microsoft Research, Technical Report MSR-TR-2001-72, 2001.

[10] Adam L. Berger, Vincent J. Della Pietra, Stephen A. Della Pietra, “A maximum entropy approach to natural language processing”, Computational Linguistics, Volume 22, Issue 1, March 1996, pp: 39-71.

[11] 中科院計算所 ICTCLA2009, http://ictclas.org/index.html

[12] 洪大弘, “基於語言模型及正反面語料知識庫之中文錯別字自動偵錯系統”, 私立朝陽科技大學資訊工程系碩士論文, 2008 年 1 月 5 日.

[13] Google translate. http://translate.google.com.tw/#zh-CN|zh-CN|

[14] Microsoft Office, http://office.microsoft.com/zh-tw/

[15] 溫普敦, http://www.winperturn.com.tw/

[16] 同文堂, http://tongwen.openfoundry.org/