# 以 Viterbi 中文名片地址欄位的後處理

王元凱
輔仁大學電子工程學系
ykwang@mails.fju.edu.tw

范國清
中央大學資訊工程學系
kcfan@csie.ncu.edu.tw

莊堯棠
中央大學電機工程學系
ytjuang@mbox.ee.ncu.edu.tw

陳泰宏
中央大學電機工程學系
s3472051@cc.ncu.edu.tw

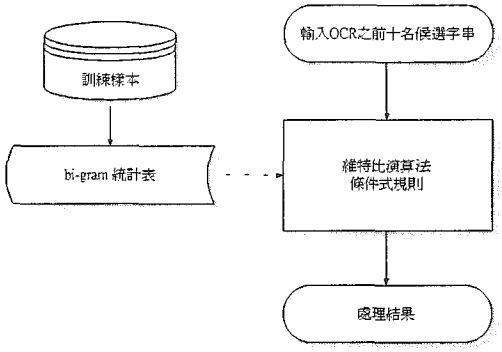
摘要

*現今市面上所銷售的 OCR 系統，在針對印刷體文字影像轉換成文字方面已經有很高的辨識率，但是在辨識名片影像時，由於受限於名片的大小，使得文字多半過小，以致於 OCR 系統常無法有良好的辨識率。*
*換句話說，名片地址欄位中的文字，經過OCR 處理後，正確字元並不會全部都位在第一候選字，甚至有些名片在以 OCR 處理後，會有遺漏字的出現，也就是說在前十名的候選字中依然不包含正確字元。如何運用統計方法來做中文名片地址欄位的後處理，以將正確字元找出來，是本文所要討論的問題。*
*本論文使用維特比演算法（viterbi），並藉著事先從地址資料庫計算得到的前後相關字統計資訊，從前十名候選字中找到正確字元，或在遺漏字的位置插入機率較高的字元，去補正 OCR 的結果。實驗以 100 張名片做測試，正確率可以由原 OCR 的 78. 44%提高到 93. 53%。*

*關鍵詞：viterbi, OCR, 相關字, 後處理, 候選字.*

## 1 簡介

在現今商業蓬勃發展下，我們會接觸更多我們從未碰面的人，名片成爲了人與人之間快速的溝通橋樑。名片傳達了許多資訊，例如姓名、住址和電話號碼等。如果能自動抽取這些資訊並加以儲存管理，將能更有效的利用這些資訊。中文名片處理系統主要是將整理名片的工作做自動化處理。
一般而言，文件處理系統包含了以下幾個步驟：前處理，文件平面的分析，字元的切割和群組，字元辨識，以及後處理[3]。目前已有許多名片處理的技術可以對名片的各個欄位作正確的分類，但是大部分的系統在作完欄位的分類以及字元的辨識後，仍然只能提供 OCR 處理後的前十名候選字供使用者自行更正。文獻[4]提到以資料庫比對的方法，將候選字中屬於該類別的字，排列於較高的名次，方便使用者更正。但是由於名片內的文字細小，因此 OCR 的正確率皆相當低，造成在實際應用上常需要作大量人爲的介入選字，因此會失去自動化的目的。
在中文文書自動校稿系統中，文獻[6]提出的中文自動錯字偵測法主要包和兩個步驟：(1) 假斷詞步驟，參考一詞庫，找出無法和相鄰字形成複字詞的單字詞取出；(2) 判斷步驟，根據各落單單字詞的詞頻及和前一字、後一字的接續強度來判斷是否爲正確字。
文獻[2]的中文錯別字自動訂正方法主要是先把原文書中之中文字，以事先整理各中文字之字形、字音、字義或輸入碼相近字的「綜合近似字集」代換，產生候選字串，在利用語言模型評分各候選字串，找出評分最高者，再與原文書對照比照，即可偵測出原文書的錯別字並提供相對應的正確字。
雖然有少數論文已提出針對一般文件之 OCR 後處理的方法，卻未針對名片的特性來進行研究。本論文提出一有效的方法來做中文商業名片地址欄的自動選字。在使用大量的地址資料庫以及維特比演算法，將分散在前十名候選字中的正確字找出後，偵測出 OCR 後的遺漏字元，藉著地址資料庫的前後相關字去補正 OCR 所遺漏的字。
我們假設名片的欄位已經正確的抽取出來，針對名片資料中最複雜的欄位—地址欄，提出一統計方法，來做名片地址欄位 OCR 後的自動選字。其中主要是以維特比演算法爲理論基礎，由三個主要的步驟組成，系統流程圖如圖一。
首先將一大量的地址資料作統計分析，統計在地址資料欄中所有曾出現的字元，根據這些字元建立字元之間的 bi-gram 統計表。以 bi-gram 表中的統計值作爲維特比演算法的機率統計值，最後作遺漏字元的偵測，補正，以及數字的糾正。




圖一 系統流程圖

## 2 維特比演算法

維特比演算法目前已經被廣泛的應用到許多的領域中，譬如語音辨識[5]等方面。該演算法可針對一已知的觀測序列（Observation sequence），找出產生此序列最可能的狀態（state）。換言之，根據兩個狀態間的關聯性，採用 Viterbi 式動態規劃搜尋找出一條最佳的狀態序列 (Optimal states sequence)，即爲評分最高的候選字串。
以下簡單介紹維特比演算法，假設我們有 N 個字，每個字有十個候選字，這$10\times N$ 個字構成我們已知的觀測序列，首先定義符號如下：

(1) $C_{t,i}$ 爲第 t 行，第 i 列的輸入候選字串，$t=1,2,\cdots,N$ ，$i=1,2,\cdots,10$ ，$C_t$ 爲第 t 行的所有候選字，

(2) $P_{t,i}=\max_i\left[P'_{t,i}\right]$，其中 $P'_{t,i}$ 爲 $C_t$ 與 $C_{t+1,i}$ 在統計上的相關度，$t=1,2,\cdots,N-1$ ，$i=1,2,\cdots,10$ ，

(3) $\Psi_t=\arg_i[P_t(i)]$ ，

(4) $Q^*=\max_j[Q_j]$ 爲維特比演算法最佳路徑下的統計值，其中 $Q_j$ 爲第 j 條路徑下 $P_t, t=1,2,\cdots,N$ 之和，i.e., $Q_j=\sum_{t=1}^{N-1}P_t$

(5) $S^*=C_1^*C_2^*C_3^*\cdots C_N^*$ 表示得到 $Q^*$ 時構成路徑的最佳狀態序列，其中 $C_t^*$ 代表第 t 個字的最佳狀態（即最佳候選字）。

根據這些定義的符號，我們將維特比演算法以虛擬碼（psudocode）表示如下：

Function Viterbi( $P'$ ) // $P'$ 代表所有的 $P'_{t,i}$ 所構成的陣列

Load( $P'_{t,i}$ )

//回溯路徑(Path Backtracking)

$Q_j=0$

For j=1 to 10

  For t=N-1 to 1

    $Q_j=Q_j+P_{t,j}(\Psi_t)$

  next t

next j

$Q^*=\max_j[Q_j]$

$S^*=C_1^*C_2^*C_3^*\cdots C_N^*$

return $S^*$

## 3 提出之方法

### 3.1 建立 bi-gram

中文文句是由一連串的字元所構成，詞和詞之間並沒有像是英文文書一般有空白字元做區隔。中文最小有意義的單位是詞，而中文詞彙的構成有一個字元的詞，二個字元的詞到數個字元所構成的詞，其中大多爲一字詞和二字詞。因此，在做在語言模型評分方面，常用到的統計的方法爲一階馬可夫模型（first-order Markov model）[1]，即是以文書中相鄰兩字的接續強度（bi-gram）來做語言模型評分。

我們以大量的一般常用名片地址欄作爲資料庫，統計常用的地址字元數，並統計兩兩字元之間的相關性，$S^*=C_1^*C_2^*C_3^*\cdots C_N^*$ ，並建立常用的地址字元。

首先將地址資料作統計分析，統計在地址資料欄中所出現的字元，根據兩兩字元間出現在地址資料欄中次數，作 bi-gram 的統計。以 bi-gram 的統計值作一 bi-gram 表，其中 $a(i,j)$ 代表第 i 列所代表的字與第 j 行所代表之字兩者相鄰的機率。圖二是一個例子，假設進行統計的文字只有「台北市塔城街 6 號 1 樓金門」這十二個字，而計算這十二個字相鄰出現的機率表。

以「台北市」三個字爲例，由查表得知「台北」及「北市」的相鄰出現機率爲 $a(1,2)=12662$ ，$a(2,3)=12655$ ，都非常高，因此「台北市」出現的機率也相對提高。

| \ | j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| i | \ | 台 | 北 | 市 | 塔 | 城 | 街 | 6 | 號 | 1 | 樓 | 金 | 門 |
| 1 | 台 | 0 | 12662 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 北 | 0 | 0 | 12655 | 0 | 0 | 15 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 市 | 0 | 443 | 0 | 17 | 0 | 3 | 0 | 0 | 0 | 0 | 72 | 0 |
| 4 | 塔 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 城 | 0 | 0 | 0 | 0 | 0 | 59 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 街 | 0 | 0 | 1 | 0 | 0 | 0 | 115 | 0 | 839 | 0 | 0 | 0 |
| 7 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 146 | 1351 | 137 | 568 | 0 | 0 |
| 8 | 號 | 4 | 1 | 0 | 1 | 0 | 0 | 510 | 0 | 4804 | 12 | 0 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 676 | 1484 | 1065 | 4562 | 0 | 0 |
| 10 | 樓 | 4 | 1 | 0 | 0 | 0 | 0 | 46 | 0 | 137 | 4 | 0 | 0 |
| 11 | 金 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 |
| 12 | 門 | 0 | 0 | 4 | 0 | 0 | 37 | 0 | 0 | 0 | 0 | 0 | 0 |

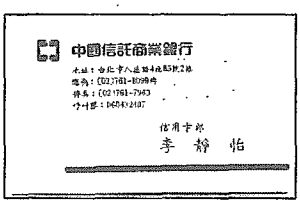
圖二 一個 bi-gram 的例子

### 3.2 文字識別

文字識別對名片地址欄位中的每一個字元影像進行辨識，得到其前十名的候選字串。假設在地址欄位中有 N 個字元，令這 N 個字元分別爲 $C_1,C_2,\cdots,C_N$ ，其中每個字元的前 10 個候選字爲 $C_t(1),C_t(2),\cdots,C_t(10)$ ，$t\in(1\cdots N)$。因此名片地址欄 S 在文字識別後可定義如下：

$$S=C_t(i)\quad t=1,2,\ldots,N,\quad i=1,2,\ldots,10$$

S 代表一個陣列。圖三是一個例子，圖三(a)是一張名片，其地址欄有 13 個字元，而構成如圖三(b)的候選陣列表。




圖三(a) 一張名片影像

| \ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 杳 | 北 | 市 | 八 | 越 | 鑄 | 4 | 粳 | 8 | 5 | 麗 | 2 | 樸 |
| 2 | 杳 | 此 | 申 | 入 | 蕊 | 鉻 | 礫 | 玫 | 靄 | 蠢 | 虢 | z | 鏤 |
| 3 | 含 | 光 | 卞 | 几 | 虺 | 庵 | ' | 段 | 疆 | 晷 | 競 | 茗 | 濮 |
| 4 | 合 | 儿 | 中 | 儿 | 慮 | 蔣 | 樓 | 崁 | 囂 | 憑 | 號 | 忿 | 棲 |
| 5 | 古 | 化 | 命 | A | 崧 | 躆 | 墦 | 役 | 贗 | 石 | 藏 | Z | 樓 |
| 6 | 台 | 兆 | 車 | 凡 | 應 | 鎔 | 碓 | 履 | 霉 | 愚 | 聽 | 翌 | 摟 |
| 7 | 舍 | 止 | 牢 | 𠂉 | 痣 | 珞 | 祺 | 展 | 鬣 | 毒 | 就 | 岔 | 樓 |
| 8 | 會 | 先 | 聿 | 人 | 德 | 躊 | 躆 | 裁 | 蠶 | 蓉 | 籤 | 磬 | 棣 |
| 9 | 倉 | 死 | 帘 | 六 | 泌 | 路 | 禱 | 屐 | 習 | 聶 | 蕊 | 壑 | 縷 |
| 10 | 審 | 兇 | 守 | 爪 | 蠢 | 瘩 | 播 | 渡 | 攀 |  | 蘊 | 蠶 | 擴 |

圖三(b) 名片中地址欄位 13 個字元「台北市八德路 4 段 85 號 2 樓」的所有候選字

由圖三(b)中我們可以看出有許多的候選字並非會出現在地址欄位的字，因此我們藉由在建立 bigram 時得到的常用地址字元來過濾圖三(b)，得到圖四，其中被改以標註「X」的位置即代表原候選字不爲常用地址字元。將不屬於地址文字之字元刪除，結果如下圖四：

| i | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | X | 北 | 市 | 八 | 越 | X | 4 | X | 8 | 5 | X | 2 | X |
| 2 | 杏 | X | X | X | X | X | X | X | X | X | X | X | X |
| 3 | X | 光 | X | X | X | X | X | 段 | X | X | X | X | X |
| 4 | 合 | X | 中 | X | X | X | X | X | X | X | 號 | X | X |
| 5 | X | 化 | X | A | X | X | X | X | X | 石 | 藏 | X | 樓 |
| 6 | 台 | X | 車 | 凡 | X | X | X | X | X | X | X | X | X |
| 7 | 舍 | X | X | X | X | X | X | 展 | X | X | X | X | X |
| 8 | 會 | X | X | 人 | 德 | X | X | X | X | X | X | X | |
| 9 | 倉 | X | X | 六 | 泌 | 路 | X | X | X | X | X | X | X |
| 10 | X | X | 守 | X | X | X | X | X | X | X | X | X | X |

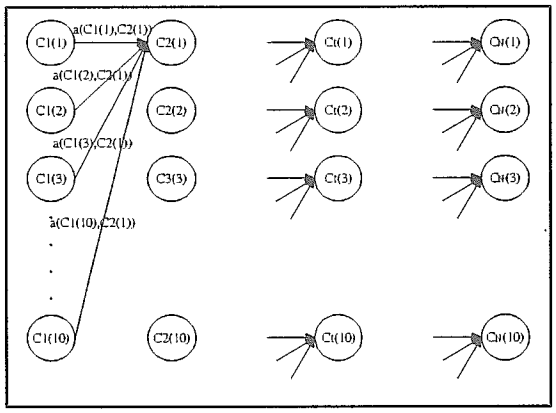
圖四 將不屬於地址文字之字元刪除

3.3 維特比演算法

根據 bi-gram 統計表，作維特比動態搜尋規劃，計算兩字間 $C_t$ 與 $C_{t+1}$ 的相關性（統計值）$P_t^{'}$。然而由於有 10 個候選字，因此我們得針對 $C_{t+1}$ 的每個候選字都計算其與 $C_t$ 的相關性，因此定義

$$P_t(i) \qquad 1 \le t \le N-1,\ 1 \le i \le 10$$

演算法的說明見圖五。在圖五中，$a(C_1, C_2)$ 爲 $C_1$，$C_2$ 在 bi-gram 表中的值，$P_t^{'}(j) = a(C_t(i), C_{t+1}(j))$ $j = 1 \cdots 10,\quad t = 1,2,\cdots,N-1$ 而 $P_t(i) = \max_{i=1\cdots10}\left[P_t^{'}(i)\right]$。最後根據回溯路徑，找出評分最高的候選字串，即爲糾正後的結果。




圖五 維特比演算法的路徑計算

3.4 遺漏字的處理

目前的 OCR 系統在處理後的前十名候選字中，並沒有百分之百的提供正確的字元讓使用者以人爲方式更正，OCR 處理後的前十名候選字中若沒有任何的一個字是正確字，是爲被 OCR 所遺漏的字元，本文將之稱爲遺漏字元。由於名片上的字體多半偏小，名片上的地址欄位在 OCR 處理過後會有一些字元沒有出現在前十名的候選字串中，這時就要做遺失字元的補救。

當有遺漏字出現而進行維特比演算時，會發現有些字元 $C_t$ 與 $C_{t+1}$ 之間沒辦法藉由 bi-gram 計算其相關性，即 $P_t(i) = 0,\ i = 1 \cdots 10$，導致維特比演算會有回溯路徑中斷的情況發生。最主要的原因是 bi-gram 其中一個字元的前十名候選字中並不包含正確字元，因此我們以維特比演算法所記錄的統計值 $P_t(i)$ 來偵測遺失的字元，並做補正。

3.4.1 遺失文字的偵測

在計算相鄰兩字間的統計值後，若發現 $P_t$ 值爲零，表示此兩字在我們的地址資料庫中沒有相關性，則這兩個字其中必有一字爲遺漏字元。我們以第一字的前相關性和第二字的後相關性來找出哪一個字才是遺漏字元。假設 $C_t, C_{t+1}$ 之間無相關性(即 $P_t = 0$)，則比較 $C_{t-1}, C_t$ 的統計值 $\sum_{i=1}^{10} P_{t-1}(i)$，和 $C_{t+1}, C_{t+2}$ 的統計值 $\sum_{i=1}^{10} P_{t+1}(i)$，若 $\sum_{i=1}^{10} P_{t-1}(i) > \sum_{i=1}^{10} P_{t+1}(i)$，則 $C_{t+1}$ 爲遺漏字，反之則 $C_t$ 爲遺漏字。

3.4.2 遺失文字的補救

在偵測出遺漏字 $C_t$ 後，根據 bi-gram 表，以 $C_{t-1}$ 後相關字或 $C_{t+1}$ 前相關字填補。我們集合 $C_{t-1}$ 的後相關字中在 bi-gram 表中有最大值的字來以及 $C_{t+1}$ 的前相關字中在 bi-gram 表中有最大值的字，選取前十名來填補遺漏字行 $C_t$，之後繼續維特比演算法的路徑回溯步驟，找出最佳路徑解。

3.5 數字的糾正

在完成維特比回溯路徑後，我們以得分最高路徑做爲我們所要更正後的結果，但是由於數字欄位在 OCR 處理後通常會有數個相似的數字爲候選字。經觀察分析在數字中 Top10 的候選字裏，通常排名越前面的爲正確字的機會越大。因此，利用此一特性，搜尋最佳路徑字串中的數字字元（1,2,…10 及一,二,…,十），針對此字元尋找 Top10 排名最前面的數字字元更新原數字，如此完成數字的糾正。

3.6 範例

茲以一範例說明所提方法的作用過程：

圖六(a) 爲 OCR 處理後之 TOP10 候選字，首先將不屬於地址欄位的字元刪除，得到的結果如圖六(b)。

| i | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 台 | 北 | 而 | 南 | 港 | 路 | 二 | 段 | S | 7 | 號 | 7 | 樓 |
| 2 | 合 | 儿 | 甫 | 商 | 湛 | 咯 | 工 | 股 | s | ? | 魷 | ? | 機 |
| 3 | 舍 | 叱 | 南 | 甫 | 灌 | 略 | 三 | 設 | 9 | 丁 | 銑 | y | 棋 |
| 4 | 谷 | 尤 | 百 | 薦 | 潜 | 賂 | 一 | 毆 | e | 了 | 鱸 | 又 | 櫝 |
| 5 | 昔 | 此 | 衝 | 籣 | 泡 | 胳 | 王 | 殷 | g | y | 鱔 | 一 | 棲 |
| 6 | 俗 | 九 | 佰 | 篇 | 濬 | 駱 | 2 | R | 5 | T | 鱔 | 丁 | 檐 |
| 7 | 古 | 尤 | 街 | 筒 | 澹 | 酪 | 乙 | 賤 | a | ▽ | 魅 | 了 | 櫻 |
| 8 | 怡 | 允 | 甭 | 冑 | 洵 | 烙 | 口 | 霞 | 8 | 下 | 懿 | 下 | 樵 |
| 9 | 含 | 光 | 偭 | 籥 | 漕 | 踡 | Z | 陵 | G | J | 鯖 | T | 禳 |
| 10 | 峉 | 沘 | 筒 | 隋 | 渚 | 鉻 | 玉 | 陂 | 3 | | 臘 | | 樸 |

圖六(a) OCR 後之 Top10 候選字

| i | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 台 | 北 | X | 南 | 港 | 路 | 二 | 段 | S | 7 | 號 | 7 | 樓 |
| 2 | 合 | X | X | 商 | X | X | 工 | X | X | X | X | X | 機 |
| 3 | 舍 | X | 南 | X | X | 略 | 三 | 設 | 9 | 丁 | X | X | 棋 |
| 4 | 谷 | X | 百 | X | X | X | X | X | X | X | X | X | X |
| 5 | X | X | X | X | X | X | X | X | X | X | X | X | X |
| 6 | X | 九 | X | X | X | X | 2 | R | 5 | T | X | 丁 | X |
| 7 | X | X | 街 | X | X | X | 乙 | X | a | X | X | X | X |
| 8 | 怡 | X | X | X | X | X | 口 | X | 8 | 下 | X | 下 | X |
| 9 | X | 光 | X | X | X | X | X | 陵 | G | J | X | T | X |
| 10 | X | X | X | X | X | X | 玉 | X | 3 | | X | | X |

圖六(b) 刪除不屬於地址文字之字元

| i | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12762 | 0 | 0 | 402 | 86 | 795 | 1489 | 0 | 255 | 1176 | 495 | 539 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 573 | 0 | 277 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 2101 | 0 | 571 | 0 | 0 | 2 | 0 |
| 7 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 303 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 824 | 0 | 0 | 0 | 0 |

圖六(c) 維特比動態搜尋規劃

因爲圖六(c)中 $P_3$ 行全爲 0，而發現有遺漏字的存在因此要偵測 $C_2$ 和 $C_3$ 何者爲遺漏。所以將 $P_2$ 行和 $P_4$ 行的總和作比較，$\sum_{i=1}^{10} P_2(i)=15$，$\sum_{i=1}^{10} P_4(i)=402$，因爲 $\sum_{i=1}^{10} P_2 < \sum_{i=1}^{10} P_4$，所以 $C_3$ 爲遺漏字。

由 $C_2$ 行上的字的後相關字和 $C_4$ 行上的字的前相關字，做相關字遞補，根據 bi-gram 統計表，找出統計值最大的字作遞補。更新 $P_2$ 和 $P_3$，繼續維特比演算法，找出最佳路徑解，結果如圖七。

| i | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12762 | 0 | 1090 | 402 | 86 | 795 | 1489 | 0 | 255 | 1176 | 495 | 539 |
| 2 | 0 | 195 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 573 | 0 | 277 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1798 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 12754 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 2101 | 0 | 571 | 0 | 0 | 2 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 303 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 824 | 0 | 0 | 0 | 0 |

圖七 最佳路徑解

經過維特比演算法得到的結果爲：

台北市南港路２段３７號７樓

搜尋屬於數字的字元，該數字字元所對應的 Top10 候選字中，找出排名最前面的數字作更正，而得到最後的正確結果：

台北市南港路二段９７號７樓

## 4 實驗結果

本實驗以 12656 筆地址資料作爲訓練樣本，該樣本取自公司電話號碼簿的台北市部分，總字數共有 194905 個字，經過統計後，經常用在台北市地址欄位的字元數共有 660 個，佔 13053 個中文字的 5.06%。Bi-gram 的統計即先建立的空白出現次數表，並從 194905 個字的訓練樣本得到該表的內容。圖八是訓練樣本的部分內容。

台北市塔城街６６號１樓
台北市金門街４號之５，１樓
台北市西園路１段２０３巷４號之１，２樓
台北市民權西路１０４號３樓
台北市光復南路４１７巷１６３號１樓
台北市信義路５段５號５Ｂ－１３
台北市昆明街１３６巷１弄１０號地下室１樓
台北市吳興街６００巷８１號２樓
台北市永吉路３２號９樓之１
台北市民生東路５段１３７巷４弄１０號６樓
台北市復興北路３４２號３樓之１
台北市士林區天玉街６３號１樓
台北市忠孝東路４段１０７號４樓
台北市和平東路１段２１６號１２樓之２
台北市仁愛路３段１３６號１２樓１２０５Ａ室
台北市忠孝東路４段１０７號４樓
台北市士林區中山北路６段２４７號３樓

圖八 部分訓練樣本

測試樣本取自 100 張位於台北市的名片地址欄，共有 1484 個字，在做完 OCR 後的 TOP1，正確選出的字有 1164 個字，不正確選出的字有 320 個，其中包含遺漏字元 99 個字，也就是說有 99 個字其候選字中都沒有正確字元。以此計算其正確率爲

$$R_{top1}=\frac{\text{TOP1為正確字字數}}{\text{總字數數}}=\frac{1164}{1484}=78.44\%$$

以本文之方法作糾正，正確選出的字元共有 1388 個字，仍有 96 個字無法糾正爲正確字元，由此算得經提出的維特比演算法處理後，正確率爲

$$R_{viterbi}=\frac{\text{糾正後為正確字數}}{\text{總字數}}=\frac{1388}{1484}=93.53\%$$

其中 OCR 前十名候選字中包含了正確字卻沒被選出的有 40 個字。偵測爲遺漏字總數有 70 個，成功補正了遺漏字元 27 個，偵出爲遺漏字元卻沒被更正正確的有 43 字。

$$\text{偵出率}=\frac{\text{偵測為遺漏字元總數}}{\text{OCR遺漏字元總字數}}=\frac{70}{99}=70.70\%$$

檢視實驗結果發現未能更正的主要原因有二：

(1) 訓練資料不夠完整

由於名片地址欄的格式很不固定，如圖八中的兩個例子：

| |
|---|
| 原地址：台北市１００復興北路３６３號１２樓 |
| 更正後：台北市號０區復興北路３６３號１２樓 |

圖八(a) 錯誤範例一

| |
|---|
| 原地址：台北市南京東路一段２４號（天津街口） |
| 更正後：台北市南京東路一段24號（天津街陽） |

圖八(b) 錯誤範例二

是由於資料庫並不包含郵遞區號以及住址後的註解所造成。

(2) 遺漏字太多

由於有的名片字形太小，或是名片影像掃瞄不良，會導致 OCR 的辨識率太低，造成太多的遺漏字，這也是無法更正的原因之一。

| |
|---|
| 台北市北投文林北路１００號３Ｆ |

圖九(a) 原地址

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 片 | 北 | 小 | 北 | 片 | 文 | 林 | 北 | 沁 | I | O | O | 沈 | 3 | 一 |
| 八 | 儿 | 山 | 扎 | 汶 | 爻 | 科 | 扎 | 泌 | I | O | O | 沁 | 召 | F |
| 六 | 七 | 十 | 儿 | 沁 | 丈 | 朴 | 儿 | 治 | 工 | Q | Q | 此 | 8 | 卜 |
| 心 | 扎 | 巾 | 化 | 注 | 之 | 杯 | 札 | 路 | l | o | o | 叱 | 三 | 尸 |
| 七 | 几 | 爪 | 札 | 泣 | 交 | 仆 | 牝 | 砝 | 1 | 。 | 。 | 化 | 另 | 『 |
| 之 | 止 | 沁 | 牝 | 汝 | 艾 | 杵 | 七 | 戊 | 一 | ○ | ○ | 汍 | S | 已 |
| 斗 | 廾 | 卡 | 几 | 咕 | 久 | 杆 | 几 | 將 | 一 | 臼 | 口 | 汁 | 沼 | 阡 |
| 入 | 化 | 心 | 仆 | 戍 | 父 | 桿 | 廾 | 咯 | | C | 日 | 汰 | 己 | 曰 |
| 汁 | 凡 | 千 | 七 | 戌 | 叉 | 杜 | 化 | 峪 | J | c | 曰 | 止 | 弓 | E |
| 打 | | + | 凡 | 女 | 史 | 什 | 扑 | 浴 | i | g | 目 | 北 | 呂 | 日 |

圖九(b) OCR 後 TOP10 候選字

| |
|---|
| 片沖山北市文林北路１００號３Ｆ |

圖九(c) 更正後地址

過多的遺漏字使得資訊不足，無法有效的利用地址資料庫做更正，所以造成系統的誤判。

## 5 結論

本文以統計的方法以及維特比演算法來做中文名片 OCR 之後處理，由實驗結果可知經由本文所提之方法能確實提高 OCR 的 TOP1 正確率，而所採用的資料庫爲不需特別處理的大量地址資料，並不需要再做額外的分類處理，若有新的地址資料，也可直接加入資料庫中，而且以統計和查表的方法作缺字的偵測和補正也確實有效。本文所提出之方法確實可以作爲名片地址欄位 OCR 後的字動選字。

檢視實驗結果可發現尚有列於 TOP10 候選字中的正確字沒有被偵測出來，主要原因爲地址資料庫的不完整，以及受到遺漏字的影響。不全的地址資料庫在系統判斷的過程中會無法找出 TOP10 中的正確字，而過多的遺漏字亦會造成系統的誤判。

未來應增加地址資料庫完備性，並在遺漏字的補正上希望可以藉由圖形的比對來增高補正率。除此之外，也可考慮以 OCR 遺漏字元的前後相關字作爲候選字，將對應的遺漏字元重做 OCR，將再辨識的結果與前後相關字作相似性比對，找出最相似字，如此應可以再提高更正後的正確率。

## 參考文獻


[1] B.-I. Li and e. al., "A maximal matching automatic Chinese word segmentation algorithm using corpus tagging for ambiguity resolution", R.O.C. Computational Linguistics Conference, Taiwan, pp. 135-146

[2] Chao-Huang Chang, "A Pilot Study on Automati Chinese Spelling Error Correction", Communications of COLIPS,Vol.4, No.2, Dec 199 4, pp. 143-149.

[3] C. H. Wu, "Chinese hand -written character Segmentation in Form Document", Master thesis, Institute of Computer Science and Information Engineering, National Chiao Tung University, Taiwan, R.O.C., 1997.

[4] Ming-Yuan Chen, "Item Identification From Business Cards, Master thesis", Institute of Computer Science and Information Engineering, National Chiao Tun University,Taiwan, R.O.C., 1999.

[5] Jay G.Wilpon et al., "Autoatic Recognition of Keywords in Unvonstrained Speech Using Hidden Markov Models", IEEE Trans on Assp., Vol.38, No.11, Nov 1990, pp. 1970-1878.

[6] 施得勝、王良志、陳志達、聶素芬 "基於統計的中文錯字偵測法"，電腦與通訊，1992 年 8 月號，19-26 頁.