繁體中文文本中對於日文人名及異體字的處理策略

林川傑[+]、詹嘉丞[*]、陳彥亨[+]、鮑建威[+]
國立臺灣海洋大學資訊工程學系
Department of Computer Science and Engineering
National Taiwan Ocean University
{cjlin, M98570019, M98570020}@mail.ntou.edu.tw[+], jjt@cyber.cs.ntou.edu.tw[*]

摘要

本論文提出一個可於進行繁體中文文章斷詞時，處理非繁體中文詞彙的方法。包括以日文漢字或中文書寫的日文人名，或是以異體字書寫的同義詞等。處理人名時，我們提出了姓名組合機率模型。處理日文人名時，我們也提出一個異體字對應的方法，可將日文姓氏及名用字對應至繁體中文用字。這方法甚至可以處理同一句子中同時出現日文及繁簡中文書寫方式的情形。在加入各種特殊類別以及中日人名處理方法後，斷詞效能 F-measure 由 94.16%提昇至 96.06%。另外對 109 篇標有日文人名的中文新聞文章進行斷詞實驗，測試集裡 862 個日文人名被成功斷成詞的比例爲 83.18%。論文中亦針對以異體字書寫的中文詞提出了一套可行的處理方式。

關鍵詞：中文斷詞、日文人名判斷、異體字

Keywords: Chinese word segmentation, Japanese name identification, variant form

一、緒論

中文斷詞在中文語言處理中，是一項重要而且必須的技術。然而中文文章裡日文人名及異體字的處理卻鮮少被研究。以往繁體中文是採 BIG5 編碼的時候，要寫出一個日文人名，書寫者常以其對應的繁體中文字元來改寫。像是原本日文漢字寫做「滝沢秀明」，在繁體中文文章中就會寫做「瀧澤秀明」。Unicode 編碼計畫出現後，就可出現多種語言字元出現在同一篇文章中的情形。各地的漢字字型多有不同，像「圖」這個字，大陸簡體字做「图」，而日文則是寫做「図」。這樣書寫習慣在中文斷詞處理中有什麼影響，在過去的研究中很少被提及。本論文便想探討這些漢字對應情形的處理。

中文斷詞的研究由來已久，現有的斷詞系統多爲規則式或是機率模型的系統。常用斷詞規則像是長詞優先規則，或是少詞優先規則。機率模型則常用馬可夫模型的 unigram 模型或是 bigram 模型等等，例如[1]。斷詞候選詞的集合多爲字典詞彙，或是以大型語料庫中蒐集詞彙。有的系統會使用構詞規則來產生部份的合法詞彙 [2]，像是一個名詞後面加上 “們” 也是合法的詞 (例如 “學生們”、“家長們”)。Wu and Jiang [3] 甚至結合文法剖析來進行斷詞。

除了斷詞歧義性之外，未知詞的處理也是一個重要課題。除了罕見詞彙 (像是 “薹售”)、專門術語 (像是化學名詞 “三聚氰胺”) 以及新發明詞彙 [4] (像是 “新流感”) 之外，具名實體 (named enitiy) 如人名、地名、組織名等的辨識技術也是研究的重點之一，例如 [5]。近來斷詞研究也探討了自動機器學習的方法，支持向量機 (SVM) [6]、條件隨機域 (CRF) [7][8] 都是曾應用在斷詞研究的機器學習方法。

較少有研究提及在繁體中文文章中處理非繁體中文詞彙的議題。比較相關的研究是探討不同區域中文詞彙使用上的差異對斷詞帶來的影響，例如使用台灣地區文章做爲訓練資料，拿來對大陸或是香港地區文章做斷詞的可能性，或是擴充字典來涵蓋各地域及各領域的詞彙等等 [9]。

本論文的問題定義爲，當中文文章中出現非繁體中文詞彙時，例如 “滝沢秀明” 這類以日文漢字書寫的日文人名、以 “瀧澤秀明” 對應中文漢字書寫的日文人名，或是 “裡面”、“裏面” 這類異體同義字，甚至是繁簡中文夾雜的文本，斷詞時能將正確候選詞判斷出來，以利斷詞成功。除第二節介紹斷詞系統基本架構外，第三節爲中日文人名處理模組，第四節說明中日漢字及異體字對應方式，第五節爲實驗結果以及討論，第六節爲結論以及未來展望。

二、斷詞策略

本文重點在探討斷詞時非繁體中文詞彙的處理策略，因此斷詞系統僅採用基本的 bigram 機率模型，旨在驗證各種策略對於斷詞效能的改善情形。候選詞除查詢已知詞列表外，另設計各種特殊類別 (如日期、數字等) 判斷規則來處理較具格式的詞彙。注意同一候選詞可能同時隸屬於多種特殊類別或是已知詞列表中的普通詞。如果在某個位置沒有找到任何長度的候選詞，系統會將該位置的字元視爲單字詞做爲候選詞。接著對每一種斷詞組合計算生成機率值，最高者做爲最後的輸出結果。

(一) 特殊類別候選詞

許多類別例如數詞、時間、日期、人名等等，其可能詞彙非常多樣，甚至可能是無限大的集合，字典不可能收錄所有詞彙，所以我們爲這些類別撰寫了判斷規則來發掘輸入句中這類候選詞。本系統所處理之特殊類別包含了地址 (可依不同國家擴增)、日期、時間、金錢、百分比、分數、網路 (IP、網址與 e-mail 地址等)、數字、外文字串及中日人名。字串中出現之英數字可爲全形、半形字以及漢數字 (一二…壹貳…)。外文字串依 Unicode 碼區可任意加入其他非拉丁字母之外文字集，如韓文、希臘文、阿拉伯文等等。因爲各外文多以空白爲斷詞符號，便可將連續出現之同一外語字串合而爲一詞。因爲本論文重點不在此，各判斷規則不做說明，僅人名的判斷規則會在第三章中介紹。

(二) 二元機率模型

產生斷詞組合之後，下一步要計算各組合的生成機率值 $P(S)$。機率值的計算方法很多，本論文使用了馬可夫的 bigram 機率模型，公式爲：

$$P(S = w_1 w_2 ... w_N) = P(w_1) \times \prod_{i=2}^{N} P(w_i \mid w_{i-1}) \qquad \text{(公式 1)}$$

其中 $P(w_i)$ 爲詞 w_i 的 unigram 機率，而 $P(w_i \mid w_{i\text{-}1})$ 爲詞 w_i 出現在詞 $w_{i\text{-}1}$ 後面的機率。爲了避免機率值連乘會過小而產生 underflow 現象，習慣上以等號兩邊取其 log 值來計算：

$$\log P(S = w_1 w_2 ... w_N) = \log P(w_1) + \sum_{i=2}^{N} \log P(w_i \mid w_{i-1}) \qquad \text{(公式 2)}$$

Bigram 模型訓練時因爲需要兩個中文詞緊鄰出現，容易有資料稀疏 (data sparseness) 的問題，也就是說大部份的中文詞 bigram 都無法訓練得到機率值。我們採用的解決方法是 backoff 至 unigram 模型，也就是當 $<w_{i\text{-}1}, w_i>$ 這個 bigram 不曾出現在訓練語料中時，$P(w_i \mid w_{i\text{-}1})$ 的值改由 $\alpha P(w_i)$ 來估算。

若是 bigram 中有特殊類別詞彙時，其 bigram 機率改以類別機率來計算。假設 w_i 屬特殊類別 S，則機率計算方式改爲：

$$P(w_i \mid w_{i-1})P(w_{i+1} \mid w_i) = P(S \mid w_{i-1}) \times P(w_{i+1} \mid S) \times P_G(w_i \mid S) \qquad \text{(公式 3)}$$

其中 $P(S \mid w_{i-1})$ 與 $P(w_{i+1} \mid S)$表示 S 類別與其他詞彙之 bigram 機率，$P_G(w_i \mid S)$ 表示類別 S 中出現 w_i 的機率，除人名外 (請見第三節)，各特殊類別值均設爲 1。

至於類別 bigram 機率模型，地址、金錢、編號、百分比、分數、網路和外文這幾類候選詞的邊界非常明確，而且不常有歧義性出現，因此我們完全信任以這些規則所判斷出來的候選詞。這幾類詞的類別 bigram 機率值均設定爲 1，表示出現這幾類候選詞的時候就會優先採用該候選詞斷法。

數字字元倒是常出現在非數詞彙中，像是 “一切”、“萬一”，其類別機率需由訓練語料統計而得。邊界明確的日期時間，如 “中華民國九十八年六月二十一日”，類別機率可設爲 1。若是有歧義現象者，像 “三十年” 可能是指 “民國三十年” 或是 “三十個年度”，它們的類別機率也採用訓練語料統計所得者。

在訓練機率模型之前，先以特殊類別判斷規則至訓練集中找尋特殊類別詞彙的出現，將之取代爲所屬類別標籤，再用來訓練 bigram 機率模型。找尋時會佐以詞性資訊，例如數字詞性一定要是 Neu，日期時間一定是 Nd 等。地址、金錢這類在原訓練集中會被斷成好幾個詞的情形，則是採多詞合併的比對策略。

類別機率的作法和 Gao *et al.* [2] 的做法很類似，但不同的是，他們將所有字典詞視爲一個類別，由各種構詞規則所衍生出來的詞也算同一種類別。與我們各種特殊類別各有其類別機率的做法十分不同。

(三) 降低計算量之演算法

當句子長度太長、或是候選詞數目太多時，會產生太多的斷詞組合，有時會高達十萬組以上，機率計算上相當耗時甚至不可行。爲了減少計算時間，我們使用 beam search 演算法來簡化計算的步驟，演算法的精神描述如下。

令原句中有 N 個字，則建立 N 個 priority queues，表示爲 record[i]，用來記錄到目前爲止，系統所找到涵蓋輸入句前 i 個字元的斷詞組合中分數最高的前 k 名。對於每個由位置 i+1 開始的候選詞 w (令其長度爲 b)，分別與 record[i] 中 k 種斷詞方式結合，並計算 $c_1 \ldots c_{i+1} \ldots c_{i+b}$ 的斷詞機率值，再與 record[$i+b$] 佇列中各斷詞組合機率值比較。如能排進前 k 名，就將最小機率者擠出佇列 record[$i+b$]。

斷詞開始時各 priority queue 均清空，i 由 0 開始反覆進行上述步驟，直至 N 個位置候選詞均被考慮完爲止。最後儲存在 record[N] 的第一名即爲機率最高的組合，做爲輸入句的斷詞輸出結果。本論文中 k 值設爲 20。

三、中日文人名處理

本節先討論如何在繁體中文文章中找出以日文漢字書寫的日文人名候選詞，至於找出以繁體中文對應字書寫日文人名的方法則留待第四、(二)節再來討論。日文人名的斷詞策略來自於中文人名的處理經驗，因此本節會先介紹產生中文人名候選詞的方法，再說明日文人名的處理方式。

(一) 中文人名處理

產生中文人名候選詞時，任何可能的中文姓名組合都可以當作中文人名的候選詞。在計算機率時，除了估算中文字能做爲姓或名的機率外，我們還估算了各種姓名組合出現的機率值。在中文文章中可能出現的中文人名組合型式如表一所列：

表一、中文人名組合型式

組合型式	可能組合	範例	組合型式	可能組合	範例
只有姓	單姓	林　老師	姓+名	單姓+單名	陳登
	複姓	諸葛　先生		單姓+雙名	王小明
只有名	單名	慧		雙姓+單名	張李娥
	雙名	國雄		雙姓+雙名	張陳素珠
				複姓+單名	諸葛亮
				複姓+雙名	司馬中原

辨識中文人名首先要有中文姓氏列表，本論文引用中文維基百科的兩個詞條「中國姓氏列表」[1]與「複姓」[2]，還有內政部戶政司[3]、中華百家姓[4]、千家姓[5]等網站共蒐集得 2,471 個姓氏。至於中文人名的名字部份，因爲人名可以隨便取，所以我們將所有的漢字都當成名用字的可能集合。

實際在提出可能的中文人名的候選詞時，並不考慮僅有單名而沒有姓的組合，原因是避免把每個中文單字都判斷成單字人名而大幅降低斷詞的效能。姓氏的雙姓部份也只考慮單姓+單姓的組合，不考慮複姓+單姓或是複姓+複姓這兩種組合，因爲並不曾見過。

$$P_G(w \mid S_{CHname}) = \max_{\sigma,\pi} P_\sigma(w \mid \pi) P_G(\pi \mid S_{CHname}) \qquad \text{(公式 4)}$$

一個中文字串w成爲中文姓名的機率定義如公式 4 所示，其中σ 說明和性別相關的機率模型，分別有男子名和女子名兩種。π 是w可以符合的一種姓名組合，用π = ‘xxxx’ 的格式來表示，以 ‘s’ 表示單姓，‘dd’ 表示複姓，‘n’ 表示人名裡單個字，例如雙姓+雙名的組合就表示爲π = ‘ssnn’，複姓+單名的組合就表示爲π = ‘ddn’。**姓名生成機率**$P_\sigma(w|\pi)$就是在性別σ 的姓名組合π 情形下生成中文人名w的機率。**姓名組合機率**$P_G(\pi \mid S_{CHname})$則是中文人名 (類別標爲S_{CHname}) 在文章中以π 這種姓名組合出現的機率。表二列出了各種姓名組合情形下，計算中文字串成爲中文姓名機率的算法。底下分別說明這兩種機率模型的建立方法。

計算姓名生成機率$P_\sigma(w|\pi)$ 時，我們採用Chen *et al*. [10] 的想法：假設姓名各字元之間無關，亦即姓氏與名字的選用無關，名用字間亦無相關。我們也假設姓氏出現機率與性別無關。表二中「姓名生成機率」欄定義了各種姓名組合的機率公式，其中LN_{CH}爲中文姓氏集合，FN_{CH}爲中文名用字的集合。

[1] http://zh.wikipedia.org/wiki/中國姓氏列表
[2] http://zh.wikipedia.org/wiki/複姓
[3] http://www.ris.gov.tw/ch4/0940531-2.doc
[4] http://www.greatchinese.com/surname/surname.htm
[5] http://pjoke.com/showxing.php

表二、各種中文姓名組合機率算法

姓名組合	姓名生成機率 $P_\sigma(w\|\pi)$	姓名組合機率
單姓	$P(c_1\|LN_{CH})$	$P(\pi=\text{'s'}\|S_{CHname})$
複姓	$P(c_1c_2\|LN_{CH})$	$P(\pi=\text{'dd'}\|S_{CHname})$
單姓+單名	$P(c_1\|LN_{CH})\times P_\sigma(c_2\|FN_{CH})$	$P(\pi=\text{'sn'}\|S_{CHname})$
雙名	$P_\sigma(c_1\|FN_{CH})\times P_\sigma(c_2\|FN_{CH})$	$P(\pi=\text{'nn'}\|S_{CHname})$
複姓+單名	$P(c_1c_2\|LN_{CH})\times P_\sigma(c_3\|FN_{CH})$	$P(\pi=\text{'ddn'}\|S_{CHname})$
單姓+雙名	$P(c_1\|LN_{CH})\times P_\sigma(c_2\|FN_{CH})\times P_\sigma(c_3\|FN_{CH})$	$P(\pi=\text{'snn'}\|S_{CHname})$
雙姓+單名	$P(c_1\|LN_{CH})\times P(c_2\|LN_{CH})\times P_\sigma(c_3\|FN_{CH})$	$P(\pi=\text{'ssn'}\|S_{CHname})$
複姓+雙名	$P(c_1c_2\|LN_{CH})\times P_\sigma(c_3\|FN_{CH})\times P_\sigma(c_4\|FN_{CH})$	$P(\pi=\text{'ddnn'}\|S_{CHname})$
雙姓+雙名	$P(c_1\|LN_{CH})\times P(c_2\|LN_{CH})\times P_\sigma(c_3\|FN_{CH})\times P_\sigma(c_4\|FN_{CH})$	$P(\pi=\text{'ssnn'}\|S_{CHname})$

建立單姓、複姓與每個名用字的出現機率，也就是 $P(c_i|LN_{CH})$、$P(c_ic_{i+1}|LN_{CH})$ 以及 $P_\sigma(c_j|FN_{CH})$ 是由一個大量的語料以 maximum likelihood 的方式統計而得，即：

$P(c_i|LN_{CH})$ ＝ 單姓 c_i 出現次數 / 所有人名個數
$P(c_ic_{i+1}|LN_{CH})$ ＝ 複姓 c_ic_{i+1} 出現次數 / 所有人名個數
$P_\sigma(c_j|FN_{CH})$ ＝ 名用字 c_j 出現次數 / 性別σ 所有人名名字字數總和

我們採用了收錄約一百萬個台灣地區的百萬人名表來統計姓氏與名的機率，其中男性姓名有 476,269 個，女性姓名有 503,679 個。由於百萬人名表中只有 953 個姓氏和四千多個名用字曾經出現，其他沒有統計資料的姓氏或名用字，我們也給予一個極小的機率值，以免造成姓名生成機率爲 0 的情形。多組實驗後經驗值建議爲 10^{-1000}。

接著估算姓名組合機率 $P_G(\pi \mid S_{CHname})$。由於我們希望得到的是各種姓名組合在中文文章中出現的機率，因此與百萬人名表中姓名組合分佈情形不盡相同。文章中常出現姓氏加上職稱的情形，例如 “林 老師”、“諸葛 先生”。在小說、書信或是話語中，也常出現僅有名字沒有姓氏、較親密的稱呼。這些現象並無法由僅是人名列表的百萬人名表觀察而得，所以需要準備一份真實文章中人名出現情形的大量訓練語料。

人名在中研院平衡語料庫中屬於專有名詞 (詞性 Nb)。我們於是以平衡語料庫中所有符合前述中文人名組合規則的專有名詞視爲中文人名。這些人名是在真實文章中出現的，符合我們的需求。但是因爲中文姓氏太多，容易將四個字以內的專有名詞都判斷爲人名，像 “中” 也是一個姓氏，“中興號” 這個客運名稱就會被誤判爲人名。爲了避免誤判，又希望能找出大部份的人名，我們於是只採用常見的姓氏和名用字來比對。這裡採用的是最常見單姓中出現機率 $P(c_i|LN_{CH})$ 合計 90%的 64 個姓氏 (陳林…程)、最常見男性名用字出現機率 $P_M(c_j|FN_{CH})$ 合計 90%的 467 個字 (文明…瀛)、最常見女性名用字出現機率 $P_F(c_j|FN_{CH})$ 合計 90%的 293 個字 (美淑…吉)，再加上所有已知複姓來判斷。判斷規則與優先順序如下，每個姓名只會被判斷成一種組合：

單字詞：單姓 > 單名 > 非中文人名
雙字詞：複姓 > 單姓+單名 > 雙名 > 非中文人名
三字詞：複姓+單名 > 單姓+雙名 > 雙姓+單名 > 非中文人名
四字詞：複姓+雙名 > 雙姓+雙名 > 非中文人名
五字詞：非中文人名

此外還再加入了“公孫氏”、“張姓”這類姓名組合，即姓氏加上“姓”或“氏”的組合，以π =‘p’來表示。以依照上列規則，平衡語料庫中 92,314 個專有名詞，有 39,612 個被判斷爲人名，各種姓名組合的出現頻率如表三所示。這些詞雖然有誤判或漏判爲人名的可能性，但期待由大量資料統計所得之數值仍有其準確性。本資料除了用以產生姓名組合機率外，也會用來計算中文人名類別S_{CHname}的類別bigram機率。

表三、中文姓名組合機率表

姓名組合機率	數量	機率值	姓名組合機率	數量	機率值
$P(\pi$=‘s’$\|S_{CHname})$	5,431	13.71%	$P(\pi$=‘ddn’$\|S_{CHname})$	126	0.32%
$P(\pi$=‘n’$\|S_{CHname})$	815	2.06%	$P(\pi$=‘snn’$\|S_{CHname})$	19,454	49.11%
$P(\pi$=‘p’$\|S_{CHname})$	487	1.23%	$P(\pi$=‘ssn’$\|S_{CHname})$	58	0.15%
$P(\pi$=‘dd’$\|S_{CHname})$	46	0.12%	$P(\pi$=‘ddnn’$\|S_{CHname})$	24	0.06%
$P(\pi$=‘sn’$\|S_{CHname})$	2,845	7.18%	$P(\pi$=‘ssnn’$\|S_{CHname})$	61	0.15%
$P(\pi$=‘nn’$\|S_{CHname})$	10,265	25.91%	總共	39,612	

舉個例子說明，計算“張德培”這個字串是否爲中文人名時，因爲“張”和“德”都是中文姓氏，所以會考慮兩種姓名組合$\boldsymbol{\pi}$={‘snn’, ‘ssn’}以及兩種性別$\boldsymbol{\sigma}$={M 男性, F 女性}四種情形的機率值，取最大的值做爲“張德培”這個字串成爲中文人名的機率值。計算結果發現，以男性的單姓+雙名這種情形分數最高。

姓名：張德培		
π	σ	機率計算
snn	男	log (P(張$\|LN_{CH}$)×P_M(德$\|FN_{CH}$)×P_M(培$\|FN_{CH}$)×$P(\pi$=‘snn’$\|S_{CHname}$)) = (-1.26) + (-1.87) + (-2.74) + (-0.31) = -6.18
snn	女	log (P(張$\|LN_{CH}$)×P_F(德$\|FN_{CH}$)×P_F(培$\|FN_{CH}$)×$P(\pi$=‘snn’$\|S_{CHname}$)) = (-1.26) + (-2.89) + (-3.27) + (-0.31) = -7.73
ssn	男	log (P(張$\|LN_{CH}$)×P(德$\|LN_{CH}$)×P_M(培$\|FN_{CH}$)×$P(\pi$=‘ssn’$\|S_{CHname}$)) = (-1.26) + (-6.02) + (-2.74) + (-2.82) = -12.84
ssn	女	log (P(張$\|LN_{CH}$)×P(德$\|LN_{CH}$)×P_F(培$\|FN_{CH}$)×$P(\pi$=‘ssn’$\|S_{CHname}$)) = (-1.26) + (-6.02) + (-3.27) + (-2.82) = -13.37

(二) 日文人名處理

中文文章書寫日文人名時，會有兩種情形。以往在 BIG5 編碼的環境下，要書寫一個日本人名，都會將人名中的漢字對應回中文漢字。舉例來說，要在中文文章中提到日本藝人“滝沢秀明”，就會把他的名字改寫爲“瀧澤秀明”。然而現在已有不少文件採用 Unicode 編碼，這使得日文漢字可以和繁體中文字同時並存在一篇文章中。本斷詞系統就希望兩種書寫方式的日文人名都能被找到成爲候選詞。

日文姓名與中文姓名的組成類似，都是使用姓與名的組合，不同之處爲日文姓氏長度可爲一到三個漢字，名的部份也是一到三個漢字，甚至可以是長度不定的平假或片假名。由於本論文著重在漢字寫法的日文人名處理，含有假名的人名就暫不考慮。

因爲日文姓名的名字部份長度不固定，而且與實際讀音的音節數較有關係。在缺乏日文人名相關資料的情形下，名的部份就不分漢字個數。已知日本人名中不會有雙姓的情形，因此姓名組合只有三種情形：只有姓、只有名、姓+名，如表四所列。

表四、日文人名姓名組合

姓名組合	只有姓	只有名	姓+名
範例	木村 長谷川	理惠 新一	伊藤由奈 高橋留美子

借用中文人名判斷的經驗，要判斷日文人名時，也需要一個日文姓氏列表，還需要一個大量的人名列表，來統計各姓氏及名用字的出現機率。最後要再統計中文文章中，各日文姓名組合的機率，以及日文人名類別 S_{JPname} 的類別機率。同樣地，一個中文字串是日文人名的機率定義爲：

$$P_G(w \mid S_{JPname}) = \max_{\pi} P_G(w \mid \pi) P_G(\pi \mid S_{JPname}) \qquad \text{(公式 5)}$$

公式中各符號的定義，請參見公式 4。但不同的是，因爲我們無法得到夠大量、已知性別的日文人名訓練語料，因此日文姓名生成機率暫不考慮性別。表五列出了各種姓名組合機率及其姓名生成機率的定義，其中m和n都是 1 到 3 之間的整數。而姓名組合中，'S' 表示姓氏出現，'N' 表示名字出現。這裡同樣地假設取名時姓氏與名用字無關，名的部份各字之間也獨立，也請注意名用字機率不分性別。

表五、各種日文姓名組合機率算法

姓名組合	姓名生成機率 $P(w\|\pi)$	姓名組合機率
只有姓	$P(c_{1\ldots}c_m \mid LN_{JP})$	$P(\pi='S' \mid S_{JPname})$
只有名	$P(c_1 \mid FN_{JP}) \times \ldots \times P(c_n \mid FN_{JP})$	$P(\pi='N' \mid S_{JPname})$
姓+名	$P(c_{1\ldots}c_m \mid LN_{JP}) \times P(c_{m+1} \mid FN_{JP}) \times \ldots \times P(c_{m+n} \mid FN_{JP})$	$P(\pi='SN' \mid S_{JPname})$

圖一、日文維基百科人名詞條範例「高橋留美子」

爲了蒐集日本姓氏，我們拜訪了一個日文網站「日本の苗字七千傑」[6]。此網站收集了 8,603 個日文姓氏，並且附有各姓氏約略人口統計，統計來源是日本全國的NTT電話簿漢字記載，總共包含了約 1.17 億人口的統計資料。雖然「日本の苗字七千傑」提供了人口統計資料，正好給我們計算各姓氏的機率值。然而中文維基百科的「日文姓名」詞條[7]中提到，日文姓氏數量高達 14 萬個之多，而「日本の苗字七千傑」只提供了 8,603 個姓氏的資料，數量明顯不足。此外，「日本の苗字七千傑」裡只有姓氏統計，也沒有名用字的資訊。我們還得另覓資料才行。

我們於是決定蒐集日文維基百科裡所有的人名詞條。在日文維基百科中，成爲詞條的日文人名，都會在本文中以粗體呈現，並且會以空格斷開姓和名。圖一的「高橋留美子」詞條便是一個例子。本文中“高橋留美子”第一次出現時，是粗體字、姓名間以空格斷開的。利用這種明確的格式，我們可以很快地蒐集日文維基百科中出現的日文人名。

不過日文維基百科中也會出現台灣或是中國的名人，像是王建民、曾國藩等。爲過濾掉中文人名，凡是名在兩個字以內，姓氏是已知中文姓氏的，全部刪去不用。我們下載了 2009 年 1 月 24 日的日文維基百科完整版[8]，利用前述格式，擷取了 65,778 筆不重複並且斷好姓名的日文人名組合，包含 12,907 個姓氏以及 2,320 個不同的名用字。表六列出 2,302 個名用字的統計數據，第三欄就是姓名生成機率中名用字機率$P(c_j|FN_{JP})$。

表六、日文人名名用字統計

名用字	次數	$P(c_j\|FN_{JP})$	累積比	名用字	次數	$P(c_j\|FN_{JP})$	累積比
子	4,821	3.60%	3.60%	亨	46	0.03%	89.99%
一	3,358	2.50%	6.10%	瑞	46	0.03%	90.03%
郎	3,237	2.41%	8.52%	…	…	…	…
美	2,230	1.66%	10.18%	褒	1	0.00%	99.99%
正	1,741	1.30%	11.48%	焔	1	0.00%	100.00%
…	…	…	…	共 2,320 種，共 134,055 次			

表七、日本姓氏統計

姓氏	出現次數	所佔比例 $P(c_{1...}c_m\|LN_{JP})$	姓氏	出現次數	所佔比例 $P(c_{1...}c_m\|LN_{JP})$
佐藤	1928000	1.65%	高井良	760	6.49×10^{-6}
鈴木	1707000	1.46%	斉藤	111	9.47×10^{-7}
高橋	1416000	1.21%	三遊亭	106	9.05×10^{-7}
田中	1336000	1.14%	…	…	…
渡辺	1135000	0.97%	城土	1	8.54×10^{-9}
伊藤	1080000	0.92%	駒尾	1	8.54×10^{-9}
…	…	…	總共 15,702 種姓氏，117,156,792 次		

[6] http://www.myj7000.jp-biz.net
[7] http://zh.wikipedia.org/wiki/日文姓名
[8] http://download.wikimedia.org/jawiki/20090124 中的 Articles, templates, image descriptions, and primary meta-pages

蒐集自日文維基百科的 12,907 個姓氏中，有不少並未收錄在「日本の苗字七千傑」網站中。我們將這兩者合併，做爲日文人名判斷的姓氏列表，合併後得 15,702 個姓氏。計算機率時，採用「日本の苗字七千傑」所提供的人口數。未出現在「日本の苗字七千傑」中的姓氏，頻率則是採用其在日文維基百科詞條中出現的次數。最後日文姓氏統計值如表七所列，其中第三欄就是姓名生成機率中的姓氏機率。請注意在這個列表中，“佐藤”到“高井良”這幾個姓氏來自網站，“斉藤”之後的姓氏則是蒐集自日文維基百科。

接著要估算日文人名各種姓名組合的機率值$P_G(\pi \mid S_{JPname})$。同樣地，我們用規則去比對中研院平衡語料庫中所有未被判斷爲中文人名的專有名詞。因爲中研院平衡語料庫裡日文人名會以繁體中文漢字來書寫，判斷之前姓氏列表和常用名用字列表都必須先轉換成中文漢字。第四、(二)節會介紹轉換成中文漢字的做法。

比對時，採用姓氏列表裡所有的姓氏，以及表六中累積比例在 90%以內的 437 個名用字 (子一…瑞)來判斷。判斷時，組合優先順序爲 姓+名 > 只有姓 > 只有名，每個姓名只會被判斷成一種組合。依照上述規則，平衡語料庫中 92,314 個專有名詞中有 4,849 個被判斷爲日本人名。這些詞會用來計算各種姓名組合機率 (如表八所列)，也會用來計算日文人名類別S_{JPname}的類別機率。然而在實作經驗上，“只有名”的這個組合會提出很多奇奇怪怪的候選詞，大大影響斷詞效果。因此我們不再採用只有名的姓名組合。

表八、日文姓名組合機率表

姓名組合機率	數量	機率值
$P(\pi=\text{'S'}\mid S_{JPname})$	718	14.90%
$P(\pi=\text{'N'}\mid S_{JPname})$	1,120	23.24%
$P(\pi=\text{'SN'}\mid S_{JPname})$	3,011	62.48%
總共	4,849	

最後以“滝沢光”的例子來說明姓名機率計算方式。“滝沢光”有兩種可能的姓名組合方式，一種以“滝沢”當姓，“光”當名，另一種是以“滝”當姓，“沢光”當名。兩個機率值以“滝沢”當姓氏的機率最高。

姓名：滝沢光	
組合	機率計算
SN	log (P(滝沢$\mid LN_{JP}$)×P(光$\mid FN_{JP}$)×$P(\pi=\text{'SN'}\mid S_{JPname})$) = (-7.35) + (-5.15) + (-0.076) = -12.576
SN	log (P(滝$\mid LN_{JP}$)×P(沢$\mid FN_{JP}$)×P(光$\mid FN_{JP}$)×$P(\pi=\text{'SN'}\mid S_{JPname})$) = (-10.70) + (-9.40) + (-5.15) + (-0.076) = -25.326

四、異體字處理

這個章節想要討論的主題，主要是在三種情形中出現。一是以中文漢字書寫的日文人名(如“滝沢秀明”寫做“瀧澤秀明”)，二是異體字寫法的同義詞 (如“裡面”和“裏面”)，三是繁體中文文章出現的簡體字 (像是“体育馆”)。後兩者雖然在文章中出現機率較少，尤其第三種情形要在 UTF-8 編碼的文件中才有可能出現，但爲了因應未來多語並存的可能性，此方向的研究仍有其必要。

(一) 異體漢字對應

由前面列出的三種情形來看，首先我們需要準備各種情形下異體漢字之間的對應列表。日文人名部份需要的是日文漢字和中文漢字的對應，異體字詞彙需要的是異體字對應表，而簡體詞彙部份則是要簡繁中文字元對應表。簡繁字元對應表比較容易取得，有不少軟體都提供簡繁轉換的功能。然而機率設定上要注意，這部份會在第(三)節中討論。

日中漢字及異體字對應表就沒有公定版本了。爲了產生這樣的對應表，我們使用了京都大學人文科學研究所安岡孝一 (Koichi Yasuoka) 與安岡素子 (Motoko Yasuoka) 所製作的異體字列表[9]，總共有 8,196 組異體漢字列表。每一組漢字都是在某些情形下的同義異體字，以下列範例中第一組爲例，“豊” 是日文裡的 “豐” 字，“丰” 則是簡體中文的 “豐” 字，而這兩個字本身又是合法的繁體中文字。異體字列表範例如下：

丰 豊 豐 靈 靈
秣 葖 蕾 藝
乾 乾 乾 干 澖

接下來，我們在每一組異體漢字中，找一個繁體中文字來做爲代表。若是該組中有多個繁體中文漢字，則選擇最常用的。繁體中文漢字頻率採用教育部八十七年常用語詞字頻表[10]。以第一組爲例，“丰”、“豊”、“豐” 都是繁體中文的漢字，三者中以 “豐” 字頻率最高，因此選它做爲這組代表字。如此一來，日文漢字 “豊” 可以對應至這個代表的中文漢字，異體字 “靈” 也可以對應到這個較常見的中文漢字了。

其實這個做法有不少要考慮的問題存在。首先，所謂的 “繁體中文” 字元其實不僅僅只有 BIG5 字集。Unicode 在定義字碼表時，就收錄了不少不在 BIG5 字集裡的繁體中文罕見字，像異體漢字第一組範例裡的 “靈” 字便是一例。由於我們的實驗資料集是以 BIG5 字集書寫的文章，本論文就先以 BIG5 字集做爲繁體中文字集。未來若有完整的繁體中文字元集，本章節所提做法就可再依據而調整。

另一問題是來自異體漢字的對應。在許多時候，兩個漢字會是同義的異體字，其實是基於某種特定的情形，並非百分之百同義。再以 “豊” 和 “豐” 爲例，“豊” 在繁體中文中是古代祭祀用的禮器 (參見教育部重編國語辭典[11])，與 “豐” 字完全不同義。只有在日文中 “豊” 才和中文 “豐” 同義。這可做爲未來研究主題。

(二) 日文人名用字之中文漢字對應

在第三、(二)節中曾經提到，如果要能判斷以繁體中文書寫的日文人名，需要將蒐集到的日文姓名用字轉換成繁體中文寫法才行。需要轉換的資料有兩個，一是日文的姓氏列表裡的姓氏，一是日文名用字列表裡面的漢字。

轉換日文姓氏的時候，不論姓氏字數多少，每個字都會以第(一)節介紹的方法將之轉換成對應的繁體中文字。例如 “滝沢” 就會轉換成 “瀧澤”，而 “中曽根” 就會轉換成 “中曾根”。轉換所得的中文寫法會合併至原本的日文姓氏列表中，機率就沿用原日文寫法姓氏的機率值。若是姓氏中有至少一個日文漢字沒有中文漢字對應的話，就不產

[9] http://kanji.zinbun.kyoto-u.ac.jp/~yasuoka/ftp/CJKtable/UniVariants.Z
[10] http://www.edu.tw/files/site_content/download/mandr/primary/shrest21.exe
[11] http://dict.revised.moe.edu.tw/cgi-bin/newDict/dict.sh?idx=dict.idx&cond=%E0T&pieceLen=50&fld=1&cat=&imgFont=1

生中文對應寫法。像是“古畑”的“畑”就沒有中文漢字對應。名用字列表的做法一樣，每個名用字都找到其對應的漢字，將之併入名用字列表中，機率也沿用原字機率值。

將對應漢字寫法合併至原來的列表，使得列表中同時有日文寫法和中文寫法。如此一來，即使句子中同時出現日文及中文寫法的人名，本系統皆可處理。舉例來說，若輸入句是「**滝沢聡就是瀧澤聰**」，因爲“滝沢”和“瀧澤”都在日文姓氏列表之中，“聡”和“聰”都出現在日文名用字列表中，“滝沢聡”和“瀧澤聰”都會被提出爲日本人名候選詞，而且兩者的機率值相同。

以同樣的概念，將日文姓氏名用字的繁體中文寫法再轉爲簡體中文並併入列表中的話，就可以處理「滝沢聡和泷泽聪都是瀧澤聰」這種日文、簡繁中文並存的文本了。這部份未以實驗進行驗證，僅以概念說明可行性。

(三) 對應異體詞的生成

爲了處理繁體中文文章中出現的簡字寫法以及異體字寫法，我們採用同一種策略處理：將所有已知繁體中文詞彙，轉換成各種可能的異體字寫法，包含簡體字寫法，再將這些詞彙合併回已知詞列表中。將已知詞轉換成各種異體寫法的方法，是先以第(一)節的方法找出詞中每一個字可能的異體字寫法，再去產生所有的組合。例如ABC爲一已知詞，A’、A”、B’ 與C’ 爲各別的異體字，我們會產生A’BC、AB’C、ABC’、A’B’C、AB’C’、A’BC’、A’B’C’、A”BC、A”B’C、A”BC’、A”B’C’ 這麼多種寫法。

將各異體字展開所得的異體詞，其機率值就承接原繁體詞的機率值。本論文之斷詞機率模型是二元模型，爲免機率列表過大，同一群異體詞就以同一個詞群編號來表示。斷詞時，找尋候選詞用異體展開後的已知詞列表，估算斷詞機率時則以各詞群編號之二元機率值來計算。

然而合併過程中，會遇到兩個不同的繁體中文詞彙對應至同一個異體詞的情形。這大部份是由於簡繁中文對應所造成，因爲簡繁對應是多個繁體字會對應到同一個簡體字，容易產生混淆。像是“白面”與“白麵”的簡體寫法都是“白面”，“改制”與“改製”的簡體寫法也都是“改制”。決定這情形異體字寫法機率值的設計法有三種想法，分別是取各原詞中機率最高、最低者，以及取其總和。第五、(四)節會討論這部份的實驗，最後系統採用了最高的機率值來做爲它的機率。

五、實驗

(一) 實驗資料與評估公式

中文斷詞實驗資料用的語料庫是中研院平衡語料庫 3.0 版[12]。平衡語料庫是專門針對語言分析所設計的，每個句子中各詞都以空白符號斷開，並且加註其詞性。文字爲現代漢語，涵蓋各種不同領域、不同主題的詞彙。平衡語料庫中共分成 316 個檔案，共有 743,718 個句子。

實驗評估方法是採 5-fold cross-validation。將 316 個檔案分成 5 組，當使用其中一組做爲測試集時，其他四組則作訓練集，用來產生已知詞列表、bigram機率，以及各類別bigram機率，因此五組實驗會有不同的機率值表。各組檔案與句子個數如表九所示：

[12] http://godel.iis.sinica.edu.tw/CKIP/20corpus.htm

表九、測試集句子資料

檔案編號	測試集	內含檔案	句子總數	已知詞	未知詞
000~065	ASBCset0	66	148,575	146,477	15675
066~129	ASBCset1	64	149,713	146,275	15877
130~183	ASBCset2	54	148,870	146,634	15518
184~244	ASBCset3	61	148,012	146,024	16128
245~315	ASBCset4	71	148,548	146,004	16148

在斷詞實驗中使用 precision、recall、F-measure，以及 BI-score 來評估系統的效能：

$$precision = \frac{\text{標準答案與系統斷出完全相同的詞彙數量個數}}{\text{系統斷出來的詞彙總數}} \quad \text{(公式 6)}$$

$$recall = \frac{\text{標準答案與系統斷出完全相同的詞彙數量個數}}{\text{標準答案的詞彙總數}} \quad \text{(公式 7)}$$

$$\text{F} - measure = \frac{2 \times recall \times precision}{recall + precision} \quad \text{(公式 8)}$$

BI-score= (正確 BI 標記個數) / 總字數 (公式 9)

這裡介紹一下 BI-score 的算法。對於一個輸入句，給定一種斷詞方式之後，句中的每一個字元都標上 B 或 I 的標籤。B 表示這字元在一個詞開頭的位置，I 表示這字元在詞的中間任何位置。比對系統提出的斷詞方式的 BI 標籤序列與標準答案斷詞方式的 BI 標籤序列，就可評估有多少比例的字元的斷詞情形是正確的。

5-fold cross-validation 五組實驗在算平均的時候，我們採用 micro-average 的概念。也就是說，precision 和 recall 的分母爲平衡語料庫中所有句子的所有斷詞詞數總和，分子則爲所有正確斷詞的詞數總和。BI-score 的分母則爲所有句子裡字元數總和，分子則爲所有字元中 BI 標記正確之總數。

(二) 斷詞系統基本效能

這一節先呈現本系統在基本架構下所達到的效能。Sys1a僅採用已知詞列表及bigram機率模型，Sys1b加上了各特殊類別候選詞產生規則，包括地址、日期、時間、金錢、百分比、分數、外文、網路 (IP、網址與信箱地址等)。如第二、(二)節所談，各特殊類別所得候選詞將直接採用 (亦即機率值設爲 1)。Sys2 加入了數詞類別，包含所有以漢字或全半形阿拉伯數字所表達之數字字串。爲免冒然將所有連續數字斷成一個詞造成錯誤，這裡設計兩組實驗來探討數詞的類別機率可能估算方式。如表十所示，加入特殊類別的系統Sys1b斷詞效能大幅提昇，而考慮真正數詞類別機率的系統Sys2b效果較好。

- **Sys2a：數詞類別機率值設爲 1**
- **Sys2b：以 maximum likelihood 方式計算數詞類別機率**

表十、斷詞系統基本架構加入特殊類別實驗結果

實驗	R	P	F	BI
Sys1a	95.66	92.72	94.16	96.96
Sys1b	95.87	93.31	94.57	97.20
Sys2a	95.97	93.57	94.76	97.30
Sys2b	**96.16**	**93.68**	**94.90**	**97.38**

(三) 中日文人名處理實驗

加入中文人名判斷規則後，機率計算時採用中文人名類別機率。本論文和 Chen *et al*.[10] 不同的地方是，我們多加入了姓名組合機率的概念，也允許未提及姓的雙名組合出現。以下分別設計實驗來驗證各方法提昇的效能如何：

- **Sys3a：中文人名類別機率、無雙名組合、無姓名組合機率**
- **Sys3b：中文人名類別機率、有雙名組合、無姓名組合機率**
- **Sys3c：中文人名類別機率、有雙名組合、有姓名組合機率**

各Sys3 實驗均以系統Sys2b爲基礎，加入三種中文人名處理策略，實驗結果如表十一所示。結果發現，加入中文人名判斷、雙名組合，以及加入姓名組合機率，都能提昇效能。可見姓名組合機率幫助不小，能成功地多辨識人名，對斷詞幫助也很大。

表十一、加入中文人名處理實驗結果

實驗	R	P	F	BI
Sys3a	96.39	94.97	95.68	97.90
Sys3b	96.42	95.49	95.95	98.05
Sys3c	**96.57**	**95.53**	**96.04**	**98.10**

接著驗證加入日文人名判斷規則與其類別機率，並且引入日文姓名組合機率後，對於系統效能的影響如何。由於實驗資料所用中文字皆限定在 BIG5 繁體中文字集範圍內，這裡日文人名處理用的是進行中日漢字對應之後的姓名列表及其機率值。以系統 Sys3c 爲基礎，實驗設計如下：

- **Sys4a：日文人名類別機率、無姓名組合機率**
- **Sys4b：日文人名類別機率、有姓名組合機率**

表十二、日文人名類別實驗結果

實驗	R	P	F	BI
Sys3c	**96.57**	95.53	96.04	98.10
Sys4a	96.54	95.54	96.04	98.10
Sys4b	96.56	**95.56**	**96.06**	98.10

表十二爲加入日文人名處理的系統效能比較。可以發現僅提出日文人名候選詞而不使用姓名組合機率的方法，反而讓系統效能下降。加入了姓名組合機率後，效能與系統Sys3c相比略有提昇，但改進並不明顯。這可能是因爲測試集裡日文人名很少的關係。這可由第三、(二)節在統計日文人名類別機率時窺知一二，因爲整個語料庫 74 萬多詞中，只有 4,849 個被判斷爲日文人名。

爲了了解加入日文人名類別真正的效能，我們設計了另外一組實驗來觀察。我們準備了出現有日文人名的 109 篇新聞文章，並用人工方式標記日文人名的位置，用以觀察這些日文人名是否能被成功地斷成詞。這 109 篇文章中，出現日文人名之處共有 862 個，屬於 216 個不同的日文人名。

觀察分爲兩部份，第一部份驗證加入日文人名類別前後，日文人名能夠被正確斷成詞的比例。觀察結果在表十三。分別以系統Sys3c以及Sys4b對這 109 篇新聞文章斷詞，統計正解的 862 個日文人名，有多少比例能正確成詞。實驗結果顯示，加入日文人名判斷模組能大幅提昇正確率。

表十三、日文人名正確成詞實驗結果

實驗	日文人名正確成詞數量	正確率
Sys3c	154	17.87%
Sys4b	717	83.18%
總數量	862	

第二部份觀察則是將斷詞系統對每個候選詞所猜測的類別輸出，看看系統認爲是日文人名的詞中有多少比例確實是日文人名 (precision)，也看看正解日文人名裡，有多少比例是因爲日文人名模組而成功結合成詞 (recall)。結果在表十四。

表十四、日文人名詞性標記實驗結果

實驗	P	R
Sys4b	74.31% (648/872)	75.17% (648/862)

實驗結果顯示，recall 和 precision 都有七成五左右，已有不錯的成功率。然而比起其他類別的高準確率，仍有相當大的進步空間。這也表示日文人名辨識不是很容易的問題。

底下來觀察幾個例子，看看加入日文人名類別前後斷詞正確與錯誤的情形。比較的是系統 Sys3 與系統 Sys4b。斷詞正確的範例：

系統 Sys3c	系統 Sys4b	系統 Sys3c	系統 Sys4b
小 林恭二	小林恭二	大 前 研一	大前研一
石原慎 太郎	石原慎太郎	藥師 丸 博子	藥師丸博子

大部分的日文姓名都能夠被正確辨識出來，不過也有少部份是斷詞錯誤的情形：

系統 Sys3c	系統 Sys4b	系統 Sys3c	系統 Sys4b
麻布 和 木材	麻布和 木材	瓦斯井 原有	瓦斯 井原有
國小 林佩萱 老師	國 小林 佩萱 老師	廣島 亞運 時	廣島亞運時

(四) 異體字處理實驗

異體字轉換希望能處理的是各種不同異體字寫法的情形，然而我們找不到適合於評估的實驗資料，因爲平衡語料庫都是以繁體中文寫成，其中的異體字出現頻率又很少。

這節實驗分兩個部份，第一部份是將平衡語料庫以軟體轉換爲簡體中文，看看系統對簡體斷詞的能力，順便探討多個不同的繁體中文詞彙對應至同一個簡體中文詞彙時，詞彙機率值的選擇方式。第二個部份則以真實的簡體中文文章做爲測試集。

第四、(三)節曾提到，因爲繁體中文和簡體中文字是多對一的關係，會有多個不同的繁體中文詞彙對應至同一個簡體中文詞彙的情形。這時，簡體中文詞彙的機率值的設定方法有三種：Sys5a採用各繁體中文詞彙中機率值最大者、Sys5b採用最小機率值、Sys5c採用機率總和。因爲中文以及日文人名判斷規則也會牽涉到機率合併的問題，這裡各 Sys5 系統改以未加入人名類別前的Sys2b當作基本系統。實驗結果如表十五所示，得知不管使用什麼方法對系統效能影響都不大，而把詞彙頻率加總與取最大的方式對系統效能影響較好。這也表示繁體文章中混用簡體詞彙，本系統仍會有不錯的斷詞效果，因爲三種策略效能差不多。最後選用系統Sys5a，即採用最大機率值。

表十五、多對一簡體詞機率設定實驗結果

實驗	R	P	F	BI
Sys5a	96.11	93.53	94.80	97.33
Sys5b	95.95	93.16	94.54	97.21
Sys5c	96.11	93.53	94.80	97.33

第二個實驗的測試集爲真實的簡體文章，我們使用 SIGHAN 1st Peking University Test Set 做斷詞實驗，共 380 行句子。我們沒有用它的 training set 來訓練系統，而直接使用系統 Sys5a，以及平衡語料庫的已知詞列表來進行斷詞實驗。實驗結果，precision 只有 86.56%，recall 也只有 81.47%，F-measure 值爲 83.94%，遠低於其他系統。由於 Peking University Test Set 的文章來自大陸地區，兩岸用語不同，文章中會有大陸地區才有的詞彙出現。另外，該語料的斷詞標準與中研院不同，像是人名的姓與名會被斷開 (如 "孙玉波")，所以斷詞正確率不高是可以預期的。本實驗僅在表達處理異體字詞的可能性，而不在比較效能。

六、結論

在本論文中我們提出了一個方法能夠處理在繁體中文文章中出現的日文人名及異體寫法中文詞的方法。文章爲 UTF-8 編碼，以支援各國文字同時出現。本論文建立之中文斷詞系統爲 bigram 機率模型，搭配各種特殊類別判斷規則及其機率模型。

中日文人名處理部份，我們提出姓名組合機率模型，並討論姓名機率值的訓練方法。也提出中日漢字轉換方式，因此不論以何種漢字書寫日文人名均可被判斷出來。由實驗數據可知，加入姓名組合機率確實可提昇系統效能，中日漢字對應的方式也能成功地偵測到大部份的日文人名。

實驗中所使用的日文姓氏列表僅包含了 15,702 個日本姓，與維基百科所提及之 14 萬個日本姓有相當大數量上的差距。如果有更完整的姓氏列表，可馬上合併至本系統，只要將罕見的姓氏機率值設爲極小值即可。此外，目前日文人名的判斷知識仍太粗略，未來可再試著加入讀音音節的組成機率，尋找大量日文人名訓練語料，尤其是姓名組合在文章中出現的機率訓練。

運用異體字對應表，各種異體寫法的中文詞也可成爲斷詞候選詞。對於異體字的處理方式，是以繁體轉簡體的中文測試集驗證了方法的可行性。僅管已知詞數量倍增，但在搭配雜湊表以及詞群編號的技巧下，對於處理速度影響不大。惟有多對一對應情形的機率模型需要再更仔細地研究。

參考文獻

[1] 彭載衍 and 張俊盛, "中文辭彙歧義之研究－斷詞與詞性標示", 第六屆中華民國計算語言學研討會論文集 (ROCLING-6), 1993, pp. 173-194.

[2] J. Gao, M. Li, and C.N. Huang, "Improved Source-Channel Models for Chinese Word Segmentation," In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (*ACL 2003*), 2003, pp. 272-279.

[3] A. Wu, and Z. Jiang, "Word segmentation in sentence analysis," In *Proceedings of the 1998 International Conference on Chinese Information Processing*, 1998 (pp. 169-180).

[4] L.F. Chien, “PAT-tree-based keyword extraction for Chinese information retrieval,” In *Proceedings of SIGIR97*, 1997, pp. 27-31.

[5] J. Sun, M. Zhou, and J.F. Gao, “A Class-based Language Model Approach to Chinese Named Entity Identification,” *International Journal of Computational Linguistics and Chinese Language Processing*, vol 8, no 2, pp. 1-28, 2003.

[6] X. Lu, “Combining machine learning with linguistic heuristics for Chinese word segmentation,” In *Proceedings of the FLAIRS Conference*, 2007, pp. 241-246.

[7] H. Zhao, C.N. Huang, and M. Li, “An improved chinese word segmentation system with conditional random field,” In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 162-165.

[8] Y. Shi, and M. Wang, “A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks,” In *Proceedings of International Joint Conference on Artificial Intelligence* (*IJCAI ’07*), 2007, pp. 1707-1712.

[9] 羅永聖, 結合多類型字典與條件隨機域之中文斷詞與詞性標記系統研究, 碩士論文, 台灣大學, 2008.

[10] H.H. Chen, Y.W. Ding, S.C. Tsai and G.W. Bian, “Description of the NTU System Used for MET2,” In *Proceedings of 7th Message Understanding Conference* (*MUC-7*), 1998. Available: http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html.