

# 語料庫統計值與全球資訊網統計值之比較：以中文斷詞應用為例

林筱晴 陳信希

國立台灣大學資訊工程學系

[hclin@nlg.csie.ntu.edu.tw](mailto:hclin@nlg.csie.ntu.edu.tw); [hhchen@csie.ntu.edu.tw](mailto:hhchen@csie.ntu.edu.tw)

**摘要.** 近年來全球資訊網(World Wide Web, 簡稱 Web)快速成長, 不同來源、不同領域、不同媒體的資訊透過網路傳遞到使用者手上。Web 除了扮演資訊傳播的角色外, 也可以被視為是一個超大的資料集, 提供語料庫為基礎—統計導向方法(Corpus-Based Statistics-Oriented Approach)所需要的統計值。本文以中文斷詞應用為例, 由傳統語料庫和全球資訊網中, 取得運用 word-based n-gram model 解斷詞歧義時所需要的統計值, 藉以比較傳統語料庫和全球資訊網的差異。在第一組實驗, 我們假設完全沒有未知詞, 運用傳統語料庫的統計值最佳, 其次依序為 Google 為基礎、AltaVista 為基礎、和 Openfind 為基礎。在第二組實驗, 我們針對指定實體辨識, 地名和組織名這兩類有不錯的效能。在第三組實驗, 我們整合斷詞系統與指定實體辨識模組, 全球資訊網統計值比傳統語料庫的統計值好。在最後一組實驗, 我們將傳統語料庫和全球資訊網混合在一起, 以全球資訊網統計值解決未知詞問題, 再以語料庫統計值解斷詞歧義性, 實驗顯示具有最佳的斷詞效能。

## 1. 緒論

在統計式自然語言處理(statistical natural language processing), 語言模型的設計、和統計值的來源是兩個實驗成功不可缺少的要件。因為語言是“活的”(live), 在日常生活中不斷有新的辭彙、和新的用法產生, 系統必須能及時反應新的語言現象, 因此使用具有時效性的資源非常重要。對於傳統語料庫而言, 資料量規模固定、內容領域變動小、時效性較弱是其缺點, 但優點是可以先加上標記(tagging), 增加附加價值, 同時可以直接透過程式, 精確掌握所需要的統計資訊。相對地, 全球資訊網(World Wide Web, 簡稱 Web)擁有十分龐大的資訊量、收集各種不同種類的文件、動態性等優點, 但缺點是沒有加上語言標記, 通常需要透過搜尋引擎(search engine)取得統計資訊, 容易受到搜尋引擎本身設計上的限制(例如, 文件索引的方式、查詢詞彙處理等)。本文將 Web 視為一個資料量龐大、且具時效性的語料庫, 研究如何利用網路上的資訊來訓練統計式語言模型, 並與傳統語料庫比較。近年來, 運用 Web 於自然語言處理, 有些相關論文發表。Zhu 和 Rosenfeld (2001)運用 Web 改善 trigram model, Computational Linguistics 期刊(2003) 也發行專刊探討這個課題, Resnik 和 Smith (2003) 將 Web 視為平行語料庫, 提供翻譯模型所需要的雙語句子。Keller 和 Lapata (2003) 說明由語料庫和 Web 上所擷取的英文 bigram 統計值是有關聯性, 但顯而易見這項理論在中文上, 由於斷詞的問題, 不見得就成立。

中文斷詞在中文自然語言處理上是個基本的工作, 許多自然語言處理應用都以斷詞作為前置處理, 例如機器翻譯、問答系統、自動摘要等。歧義性是中文斷詞系統第一個必須解決的問題, 由於中文字串可能有多種不同的斷詞組合, 斷詞系統必須選出其中最好的一種斷詞方式。另外, 受限於辭典覆蓋度的問題, 未知詞處理也是必要的工作。而指定實體(named entities, 簡稱 NE)是常見的未知詞, 一般斷詞系統都會輔以指定實體辨識模組, 提出策略自動辨識出人名、地名、和組織名等的存在(Chen, Ding 和 Tsai, 1998; Chen, Yang 和 Lin, 2003)。本文以中文斷詞系統為例, 以統計式語言模型作為基礎, 將 Web 統計資訊應用在中文斷詞上。透過對搜尋引擎查詢, 所傳回的網頁數(page count), 模擬統計式模型所用到的詞頻, 藉此比較以傳統語料庫和以全球資訊網為基礎的差異。

本文第 2 節首先說明中文斷詞所用到的語言模型, 以及所需的統計值, 傳統語料庫和全球資訊網如何提供, 在這裡我們假設完全沒有未知詞問題的存在。第 3 節討論指定實體辨識, 我們運用可能比例測試(likelihood ratio test)方法, 判斷某一字串是否為指定實體。第 4 節嘗試將斷詞和

指定實體辨識整合，以彰顯傳統語料庫和全球資訊網的特點。第 5 節提出結論與未來可能的研究。

## 2. 中文斷詞

### 2.1. 實驗材料

中文斷詞實驗所用到的辭典，是從中研院平衡語料庫(ASBC, 1995)擷取。我們將語料庫中的詞，直接收錄在辭典內，共 145,929 個詞。此外，我們將平衡語料庫切分成兩部分，作為訓練語料庫(5,386,820 個詞)、以及測試語料庫(586,698 個詞)。前者用以訓練實驗的語言模型，後者作為實驗測試的標準答案。

詞綴為具有衍生性的附著語素，可根據構詞律組合成詞。前綴為「附加於別的成分之前構成詞」，例如：大卡車、大騙子中的「大」屬於前贅詞。後綴為「附加於別的成分之後構成詞」，例如：犧牲者、造物者中的「者」屬於後贅詞。在前綴/後綴詞表中，我們共收錄 1,135 個前綴詞，以及 1,419 個後綴詞。

### 2.2. 演算法

#### 2.2.1. Word-based N-gram Model

本文採用 word-based bigram model 作為斷詞系統的語言模型，對中文字串  $C_1C_2C_3\dots C_n$ ，查完辭典後可能會得到一種以上的組合，歧義性分析就是從這些不同的組合中，挑選出最可能的詞串

$\hat{W} = w_1w_2\dots w_m$ ，使得機率值  $\prod_{i=1}^m P_r(w_i | w_{i-1})$  為最大，也就是說：

$$\hat{W} = \underset{w}{\operatorname{argmax}} P_r(W|C) \approx \underset{w}{\operatorname{argmax}} \prod_{i=1}^m P_r(w_i | w_{i-1})$$

$P_r(w_i | w_{i-1})$  代表在上一個字是  $w_{i-1}$  的情況下， $w_i$  出現的機率。這個數值可以轉換成頻率，以最大可能估計(maximum likelihood estimation, MLE)計算如下：

$$P_r(w_i | w_{i-1}) \approx P_r(w_{i-1}w_i) / P_r(w_{i-1}) = \frac{\operatorname{Count}(w_{i-1}w_i) / N}{\operatorname{Count}(w_{i-1}) / N} = \frac{\operatorname{Count}(w_{i-1}w_i)}{\operatorname{Count}(w_{i-1})}$$

$\operatorname{Count}(w_{i-1}w_i)$  是詞  $w_{i-1}$  和  $w_i$  相鄰出現的次數， $\operatorname{Count}(w_{i-1})$  是詞  $w_{i-1}$  出現的頻率。採用傳統語料庫策略，詞頻從中研院平衡語料庫訓練取得。採用全球資訊網策略，詞頻是以搜尋引擎(如:Google)所傳回的網頁數來模擬。檢索時查詢關鍵字(search term)前後會加上雙引號(“ ”)，代表必須是“exactly match”。

由於統計資料的稀疏性(sparseness)問題，不論是透過訓練語料庫、或是全球資訊網，所得到的  $\operatorname{Count}(w_{i-1}w_i)$  數值，會出現某些相鄰詞組沒有共同出現過的情形，因而發生「零機率」，即  $P_r(w_i | w_{i-1}) = 0$ ，在實驗中我們將其機率值設定為一個極小的數值( $10^{-12}$ )。

除了 word-based bigram model 外，我們也同時規劃 word-based tri-gram model 的實驗。我們想知道詞串的掃描方向，對於整個實驗結果的影響，所以除了由左到右掃描字串外，也由右而左掃描詞串，作了 reverse bigram model 的實驗。

#### 2.2.2. Prefix/Suffix Rule

對於某個詞串 AB，若 A、B、AB 皆收錄在辭典內，則會有「A B」和「AB」兩種不同的斷詞組合，在解歧義性階段會選擇其中之一，作為斷詞結果。例如：「使用者」，因為辭典包含「使用、者、使用者」等詞彙，所以會產生「使用 者」和「使用者」兩種不同的組合。我們觀察中

研院平衡語料庫，這種包含前綴/後綴的詞彙，大部分都被斷成單一詞，而非兩個詞，所以增加 prefix/suffix rule 的策略：給予詞串 AB，查前綴/後綴詞表，發現 A 是前綴(或 B 是後綴)，且 AB 收錄於辭典中，則我們只建議「AB」這單一詞為候選者。例如「使用者」的「者」，查表發現屬於後綴詞，則系統就將之斷成「使用者」，而不會有拆開成兩個字的情形發生。

## 2.3. 結果與討論

### 2.3.1. 實驗結果

在第一組實驗，有三種語言模型：bigram model、reverse bigram model、和 tri-gram model；兩種策略：Dict-only 策略和 Dict+prefix/suffix rule 策略；全球資訊網：Google、Openfind、和 AltaVista，共有 11 個實驗，結果如表 1 所示。

首先我們固定語言模型為 bi-gram model，以及字串掃描方向為由左到右，以觀察傳統語料庫和全球資訊網之差異，不管在 Dictionary only 或 Dictionary and prefix/suffix rule 策略，bi-corpus 的 F-measure 都最好，其次依序為 bi-google、bi-altavista、和 bi-openfind。當字串掃描方向改為由右到左，這四種方法的順序不變，由右到左策略比由左到右的效能好。比較 bi-gram 和 tri-gram models，使用傳統語料庫和全球資訊網，bi-gram model 都比 tri-gram model 好。

表 1. 歧義性分析結果

	Dictionary Only			Dictionary and Prefix/Suffix Rules		
	precision	recall	F-measure	precision	recall	F-measure
bi-corpus	96.30%	95.39%	95.84%	97.09%	95.17%	96.12%
bi-google	94.76%	94.42%	94.59%	95.83%	94.24%	95.03%
bi-openfind	94.67%	93.42%	94.04%	95.39%	93.75%	94.56%
bi-altavista	94.70%	93.71%	94.20%	95.70%	94.16%	94.92%
rev-corpus	96.75%	95.08%	95.91%	97.00%	94.92%	95.95%
rev-google	94.87%	93.86%	94.36%	95.92%	94.30%	95.10%
rev-openfind	94.84%	93.56%	94.19%	95.52%	93.87%	94.69%
rev-altavista	94.82%	93.80%	94.31%	95.80%	94.23%	95.01%
tri-corpus	96.59%	94.64%	95.61%	96.66%	94.63%	95.63%
tri-google	93.88%	93.24%	93.56%	95.20%	93.83%	94.51%
tri-altavista	93.82%	93.13%	93.47%	95.04%	93.67%	94.35%

### 2.3.2. 實驗討論

我們將斷詞錯誤分成四種類型。

1. 合併：表示在標準答案中應該為分開的幾個詞，但系統卻將這些字合併為一個詞彙。例如標準答案應為「有一些」(例：外頭有一些矮房子)，系統答案卻斷成「有一些」。
2. 拆開：表示在標準答案中應該為一個詞彙，但系統卻將此詞彙拆開成數個詞。例如標準答案應為「商學所」，系統答案卻斷成「商學 所」。
3. 搶字：表示兩個相鄰的辭彙，其中一個詞彙的部分中文字元被另一個詞彙搶走形成它的一部分。例如標準答案應為「法務 部門」，系統答案卻斷成「法務部 門」。
4. 其他：不屬於以上三種型態的錯誤，通通歸於其他。例如標準答案應是「及 第二組」，系統答案卻斷成「及第 二組」；另外，又如標準答案應是「原 裝設 於」，系統答案卻斷成「原裝 設於」。

根據這四種錯誤型態分類，我們計算不同模型的錯誤個數，如表 2 所列。

表 2. 錯誤型態之分佈

	Dictionary Only				Dictionary and Prefix/Suffix Rules			
	合併	拆開	搶字	其它	合併	拆開	搶字	其它
bi-corpus	12,136	3,805	1,244	402	14,931	942	835	84
bi-google	12,133	6,811	2,496	508	14,894	2,225	2,518	518
bi-openfind	14,828	3,936	3,915	718	14,967	1,702	3,896	743
bi-altavista	14,675	5,280	2,541	742	14,895	2,162	2,518	783
rev-corpus	14,852	1,330	1,438	109	15,066	744	1,482	112
rev-google	14,584	5,363	2,409	507	14,917	2,051	2,402	529
rev-opefind	14,834	3,798	3,572	710	14,975	1,674	3,557	735
rev-altavista	14,669	5,138	2,378	723	14,896	2,117	2,357	747
tri-corpus	14,961	787	2,318	153	15,125	527	2,319	150
tri-google	14,547	6,556	2,835	1,329	14,848	2,507	2,843	1,307
tri-altavista	14,578	6,125	3,230	1,447	14,848	2,326	3,239	1,447

由表 1 所列的實驗結果可知，無論採用何種語言模型，最後的表現都是傳統語料庫的方法最佳。這是因為傳統語料庫方法，所用的訓練語料庫是斷過詞的語料，在訓練階段掌握真正的詞頻。相對地，全球資訊網的統計值則是用搜尋引擎傳回的網頁數來估算，而網頁數直觀上僅代表「有多少個網頁包含此特定的詞」，是資訊檢索中所稱的「文件頻率」，並不是真正的詞頻。此外，全球資訊網的網頁，沒有像傳統語料庫已經人工斷詞標記，頻率會有誤差。例如，我們對 Google 下「門聯」這個查詢，會發現包含「澳門聯網」這個字串的網頁，也被計算在網頁數中，但事實上「澳門聯網」正確斷詞為「澳門 聯網」，所以不能計算在「門聯」的網頁數內，因此以網頁數來替代詞頻會有誤差。

由表 2 可觀察到傳統語料庫方法發生搶字類型的錯誤情況，明顯比全球資訊網方法來得少。字串 ABC 若可以斷成 A/BC 或是 AB/C，則會有相鄰兩詞彙互相搶字的情況，根據 word-based bigram model，這兩種斷詞組合的機率值為  $count(ABC)/count(A)$ ，及  $count(ABC)/count(AB)$ 。在全球資訊網方法，這兩個數值的分子部分相同，都是字串 ABC 的網頁數，所以分母部分決定了這兩個數值之間的大小關係。如果 A 屬於高頻字，則  $count(A)$  數值很大，使得  $count(ABC)/count(A) < count(ABC)/count(AB)$ ，則斷詞系統會選擇 AB/C 作為斷詞結果，反之亦然。這種斷詞模式往往造成錯誤的結果，例如：將「後來/的」誤斷成「後/來的」、將「是/故意」誤斷成「是故/意」。相對於傳統語料庫方法，因為統計值皆來自於經過斷詞的訓練語料庫，因此分子部分的  $count(ABC)$  在這兩種斷詞組合下，實際上為  $count(A BC)$ 和  $count(AB C)$ 此二個不同的數值，所以這種錯誤情況的發生較少。

統計值來自 Google 的實驗，拆開類型的錯誤最多，這是因為 Google 採用查辭典方式建立索引，所以發生「substring 的網頁數比 superstring 的網頁數來得少」這種情形，造成  $P_i(w_{i-1} | w_i) = \frac{Count(w_{i-1}w_i)}{Count(w_{i-1})}$  的數值大於 1。當發生這種情況時，斷詞結果會傾向於拆開成多個詞彙，

例如含「原住民」的網頁數為 170,000，而「原住」的網頁數只有 9,990，因此在例子「南島語族系的原住民」中，會被誤斷詞為「原住 民」。

實驗採用 prefix/suffix rule 策略可以減少拆開類型的錯誤，因為我們會把所有包含前綴/後綴的詞彙斷成單一詞彙，而不會拆成兩個詞彙。雖然這個策略同時也可能增加合併類型的錯誤，將標準答案中應該分開的兩個相鄰詞彙，合成單一詞彙。例如：標準答案為「性/騷擾」，系統誤斷為「性騷擾」，但整體斷詞系統的效能是提升了。我們從錯誤答案中也發現，有些字串在中研院平衡語料庫中，並沒有一致性的斷詞，例如「小河」在「隨著屋後小河的潮漲潮落」和「我和弟弟溜到附近的小河玩水」，這兩個句子就有不同的斷詞標記。

我們仔細比較三個搜尋引擎的差異，Google 在文件層次、和查詢層次皆參考內建辭典，會有 substring 沒有收錄到辭典中，所以網頁數會較 superstring 少的情形發生。而 AltaVista 因為沒有查詞的動作，所以與檢索詞彙字串比對，成功的網頁就會被傳回。以「巴拿馬」和「巴拿」的例子來說明：對 Google 查詢「巴拿馬」，傳回的網頁數為 89,900；但如果查詢「巴拿」，則會發現它的網頁數為 11,200，比「巴拿馬」的數值來得少。對 AltaVista 查詢「巴拿馬」，得到的網頁數為 393，查詢「巴拿」所得到的網頁數為 9,560，大於「巴拿馬」的數值，與「substring

的限制性比 superstring 弱，應有較多的網頁包含」這個假設吻合。

Openfind 顯示的網頁數和實際傳回的網頁數不符合，例如查詢「中文詞知識庫計畫」，會出現「Openfind 找到 10 篇相關網頁」的訊息，但實際上只顯示了 7 個網頁連結。另外，Openfind 會偵測使用者的查詢流量，若是查詢動作過於頻繁，會限制此使用者的查詢，網頁會出現「很抱歉，系統偵測您的查詢狀況異常，已限制查詢。」的訊息，因此不適合對 Openfind 作大量的查詢。

Google 所顯示的網頁數只是個估算的數值，例如我們對 Google 搜尋「電腦」會傳回數值 4,110,000，搜尋「網際網路」則會傳回數值 322,000。這是由於 Google 為分散式查詢，為了爭取時間效率，當一台機器搜尋結束之後，即立刻回傳資訊給使用者，因此造成網頁數不精確的結果。

總結，傳統語料庫方法因為可以精確計算出詞頻，相較於全球資訊網方法受限於搜尋引擎的限制，造成網頁數並不準確，因此在斷詞解歧義實驗有較佳的結果，不過差距約在 1%-1.5% 間，全球資訊網方法仍可媲美於傳統語料庫方法。

### 3. 指定實體辨識

#### 3.1. 實驗資源

由於中研院平衡語料庫，沒有對指定實體作特別的標記，而是將其切分成連續的辭彙，例如「塞凡尼克國際公司」被標記為「塞凡尼克 國際 公司」，因此這個語料庫不適合用來做為評估指定實體辨識效能的素材。本階段的實驗採用 MET-2 測試語料(MUC, 1998)，作為指定實體辨識的測試文件。

這階段實驗所用到的辭典，仍然是由中研院平衡語料庫所擷取的辭彙組成，但做了小部分的更改。我們檢查 MET-2 測試語料所有答案(即指定實體)，如果指定實體已被收錄在辭典中，則從辭典中刪除這些詞彙。主要的目的是「使辭典與答案形成互斥(mutual exclusive)的關係」，以精確地評估系統效能。

指定實體關鍵詞集，收錄中文人名、地名、與組織名等三種類型的指定實體的關鍵詞。其中，中文人名部分共收錄了 387 個中文姓氏，例如：陳、李等等。地名部分收錄了 32 個關鍵詞，例如：市、港等等。組織名則收錄了 851 個關鍵詞，例如：黨、工會等等。我們根據指定實體關鍵詞集，由查詞典處理後的句子中，找出可能的指定實體候選詞，再由後續步驟判斷是否為指定實體，並確認其邊界。

有一些詞語出現頻率極高，例如「的、了」，這類的詞稱為停用字(stopword)。實驗中採用的停用字集，共收錄了 1,332 個停用字。當我們掃描詞串時，查詢此停用字集，就可得知哪些詞屬於停用字。

#### 3.2. 可能比例測試(Likelihood Ratio Test)

##### 3.2.1. 基本演算法

假設指定實體的組成成分間形成 collocation，要判斷  $w^1$  和  $w^2$  之間的關聯性，我們採用可能比例測試(Manning and Schutze, 1999)如下：

$$\text{假設一. } P(w^2 | w^1) = p = P(w^2 | \neg w^1)$$

$$\text{假設二. } P(w^2 | w^1) = p_1 \neq p_2 = P(w^2 | \neg w^1)$$

$$p = \frac{c_2}{N} \quad p_1 = \frac{c_{12}}{c_1} \quad p_2 = \frac{c_2 - c_{12}}{N - c_1} \quad (\text{假設 } p_1 > p_2)$$

其中， $c_1$  表示  $w^1$  出現的頻率， $c_2$  表示  $w^2$  出現的頻率， $c_{12}$  表示  $w^1$  和  $w^2$  同時出現的頻率， $N$  則為語料庫內的辭彙總數。假設一代表  $w^1$  和  $w^2$  是獨立的，假設二則代表  $w^1$  和  $w^2$  有關聯性。

假設機率分布是 binomial distribution，計算可能比例(likelihood ratio)  $\lambda$  的  $\log$  值。而  $-2\log \lambda$  的數值是呈現  $\chi^2$  機率分配，當自由度等於 1 時，我們將信賴指標設定為 99%，其臨界值為 2.71。如果我們將所有參數值都代入公式，計算之後得到  $-2\log \lambda$  值大於 2.71，則代表“接受假設二”，意即是  $w^1$  和  $w^2$  是有關聯的。

假設統計值來自檢索全球資訊網回傳的網頁數，令中文字串  $w = c_1c_2\dots c_k$  為等待檢驗的指定實體候選字串，為了增加檢索字串的限制性，我們所取的子字串為原來字串頭尾各去除一個字元，亦即  $w_{left} = c_1c_2\dots c_{k-1}$ ，和  $w_{right} = c_2c_3\dots c_k$ ，則參數  $c_1 = pc(w_{left})$ 、 $c_2 = pc(w_{right})$ 、 $c_{12} = pc(w)$ 、 $N$ =搜尋引擎所收錄網頁總數，其中  $pc(word)$  代表  $word$  的網頁數。有了這些參數值，套進上述公式計算  $-2\log \lambda$  的值。如果大於 2.71，則代表通過公式檢驗，我們認定字串  $w$  為一指定實體。如果小於 2.71，則表示非指定實體。舉例來說，令  $w$  為「艾特納保險公司」，則  $w_{left}$  為「艾特納保險公」， $w_{right}$  為「特納保險公司」。我們藉由搜尋引擎得到此三個字串的網頁數，並計算  $-2\log \lambda$  值。

基本演算法如後：先運用辭典把句子切分出各種可能的詞組合，然後掃描詞串。若是掃描到的某個詞，屬於指定實體的關鍵字，則表示此詞串聯其前/後相鄰的幾個詞可能組成一個指定實體，因此驅動指定實體辨識的檢驗。如果屬於中文人名姓氏，則我們往後檢查最多兩個字，計算字串  $w_i w_{i+1} w_{i+2}$  是否通過公式檢驗。若是某個詞  $w_i$  屬於地名、或是組織名的關鍵詞，則我們最多往前檢查五個詞，計算此串聯字串是否通過公式檢驗。我們對於通過公式檢驗的字串，再根據關鍵詞的類別，給予不同的語意標記。

例如，字串「酒泉衛星發射中心」，經過查辭典之後，得到「酒泉衛星發射中心」。掃描詞串發現「中心」是地名的關鍵詞，我們串聯前面五個詞(即「酒泉衛星發射中心」)，將此字串的各相關參數值，即  $w$ 「酒泉衛星發射中心」、 $w_{left}$ 「酒泉衛星發射中」、與  $w_{right}$ 「泉衛星發射中心」，在搜尋引擎所得到的網頁數套進公式檢驗。

對於外國人名部分，是以連續出現  $n$  個以上的單字詞，所串聯的字串來做判斷。如果此連續單字詞所組成的字串，能夠通過公式檢驗，我們就視之為外國人名，將字串加上人名的標示。

### 3.2.2. 修改演算法

根據原來公式的定義，當  $pc(w)$ 、 $pc(w_{left})$ 、和  $pc(w_{right})$  有一值為零時，則就不會通過檢驗。但是我們發現可能有種情況：檢索某字串  $w$ ，回傳的網頁數大於零( $pc(w) > 0$ )，但是其子字串的網頁數  $pc(w_{left})$ 、或  $pc(w_{right})$  卻等於零。如果依照原始公式，這種情形的字串就會被遺漏，所以將原來檢驗過程，做了點小小的修改：當  $pc(w) > 0$  時，若是  $pc(w_{left})$ 、或  $pc(w_{right})$  的值等於零時，則直接通過檢驗，不需要計算  $-2\log \lambda$  的數值。

## 3.3. 實驗與結果

### 3.3.1. 實驗一：字典與答案為互斥

表 3 列出第一組指定實體辨識實驗的結果，系統效能以 F-measure 表示，altavista\_1 和 google\_1 是以原始未修改的公式檢驗，altavista\_2 和 google\_2 是以修改後的公式檢驗。本實驗所用的辭典，與 MET-2 測試語料中的指定實體集為互斥關係，表示測試語料中的每個答案，都不會收錄於辭典內。由於一些指定實體  $w$  的網頁數大於零，但  $w_{left}$  或  $w_{right}$  卻等於零，因此無法被辨識出。實驗驗證 altavista\_2 和 google\_2 的效能，都比對應的 altavista\_1 和 google\_1 高。另外，一些停用字出現頻率太高，因而導致系統辨識出的指定實體邊界錯誤，例如：「在太原衛星發射中心」，“在”是多餘的。因此，當我們掃描詞串時，碰到停用字就停止。altavista\_3 和 google\_3 是以修改後的

公式，加上判斷是否包含停用字的結果。修正後的公式，加上停用字的判斷，有助於提升指定實體的辨識，所以之後的實驗以 altavista\_3 和 google\_3 為基礎。本方法可以找出，如「李塵風」、「西昌衛星發射中心」、和「國際衛星通信組織」等指定實體。

表 3. 指定實體辨識實驗一之結果

	PER	LOC	ORG	Total
altavista_1	34.60%	12.98%	51.41%	32.12%
altavista_2	34.60%	12.54%	52.81%	32.50%
altavista_3	43.80%	16.95%	62.58%	39.44%
google_1	47.03%	10.22%	21.37%	22.06%
google_2	48.70%	10.02%	40.42%	29.18%
google_3	55.84%	16.72%	54.65%	37.84%

### 3.3.2. 實驗二：辭典內增加收錄國名和省名

實驗一地名辨識不好的主要原因之一，是許多地名並沒有包含常見的關鍵字，例如國名：印度、巴西等等，所以沒辦法根據關鍵字來驅動這些地名的辨識。修改的方式是：在辭典內增加收錄世界各國的國名，以及大陸各省的省名，這部分在地名來說是比較屬於 closed set。有了這方面的資訊之後，我們可以對文件中的國名與省名作標記，以增加地名的辨識效能。文件中也常常出現地名縮寫，例如：英、法、德等等，我們在辭典內增加收錄一些國名縮寫，因此部分國名縮寫可以被辨識出來，但像是「中」或「以」這類國名縮寫，本身屬於常常出現的停用字，則沒被收錄在辭典內。

表 4 中的 altavista\_4 和 google\_4 表示以修改後的公式，和增加收錄地名的辭典所實驗的結果，altavista\_5 和 google\_5 表示增加國名縮寫的標記，altavista\_5\_n 和 google\_5\_n 表示：我們檢查連續出現 n 個以上的單字詞是否通過檢驗，通過視之為外國人名，藉此觀察對人名辨識的影響。表 4 顯示增加收錄國名和省名，以及標記部分國名縮寫後，能提高辨識效能。增加外國人名的判斷，在連續 5 個單字詞判斷時分數為最高。AltaVista 在組織名辨識效能較 Google 佳，而 Google 辨識人名和地名的效能則較 AltaVista 佳。

表 4. 指定實體辨識實驗二結果

	PER	LOC	ORG	Total
altavista_4	48.30%	74.29%	62.71%	67.45%
altavista_5	48.30%	77.45%	62.71%	69.54%
altavista_5_2	23.16%	76.40%	62.71%	59.39%
altavista_5_3	40.27%	77.48%	62.71%	67.34%
altavista_5_4	47.93%	77.65%	62.71%	69.28%
altavista_5_5	50.91%	77.45%	62.71%	69.81%
altavista_5_6	48.45%	77.45%	62.71%	69.57%
google_4	61.40%	75.61%	55.15%	67.77%
google_5	61.40%	78.94%	55.15%	70.02%
google_5_2	30.07%	77.95%	55.15%	60.12%
google_5_3	51.25%	78.91%	55.15%	67.92%
google_5_4	58.12%	78.99%	55.15%	69.38%
google_5_5	63.91%	78.94%	55.15%	70.32%
google_5_6	61.21%	78.94%	55.15%	70.00%

### 3.3.3. 實驗三：混合使用 Google 和 AltaVista 的統計值

本實驗結合此兩搜尋引擎的優點：當我們要辨識一詞串是否為人名、或地名時，採用 Google 搜尋回來的網頁數代入公式檢驗。如果要辨識是否為組織名，所套入公式的參數值，則是採用 AltaVista 所傳回的網頁數。表 5 中的 mix\_1 表示以修改後的公式，以及混合使用 Google 和 AltaVista 的統計值後的實驗結果，mix\_2 表示增加國名縮寫的標記後的實驗結果。mix\_2\_n 表示我們檢查連續出現 n 個以上的單字詞是否通過檢驗。

表 5. 指定實體辨識實驗三結果

	PER	LOC	ORG	Total
mix_1	60.30%	75.47%	62.69%	69.56%
mix_2	60.30%	78.80%	62.69%	71.69%
mix_2_3	50.00%	78.76%	62.69%	69.47%
mix_2_4	56.49%	78.85%	62.69%	70.90%
mix_2_5	62.43%	78.80%	62.69%	71.92%

### 3.3.4. 實驗四：刪除測試資料內網頁數等於 0 的答案

以全球資訊網為基礎的指定實體辨識，效能的好壞取決於網路上是否有收錄此詞彙的資訊。如果沒有，搜尋結果的網頁數就會是零，不能通過檢驗，因此遺漏此答案。我們想知道在最佳情況下，此演算法的效能，即每個測試文件中的指定實體都可以找到對應的統計值，所以將原來測試文件根據 Google 和 AltaVista 做了兩份不同的修改，把網頁數等於 0 的指定實體從原始文件中刪除，以此修改後的文件當做系統的輸入資料。表 6 中的 altavista\_6 和 google\_6 是以修改後的演算法，配合增加收錄地名的辭典所得到的結果，altavista\_7 和 google\_7 表示增加國名縮寫的標記，altavista\_7\_n 和 google\_7\_n 表示我們檢查連續出現 n 個以上的單字詞是否通過檢驗。

表 6. 指定實體辨識實驗四結果

	PER	LOC	ORG	Total
altavista_6	67.53%	80.96%	78.16%	78.74%
altavista_7	67.53%	84.75%	78.16%	81.18%
altavista_7_4	66.67%	84.81%	78.16%	80.94%
altavista_7_5	70.29%	84.75%	78.16%	81.42%
google_6	64.74%	77.55%	58.93%	70.58%
google_7	64.74%	81.06%	58.93%	72.93%
google_7_4	61.16%	81.12%	58.93%	72.26%
google_7_5	67.50%	81.06%	58.93%	73.27%

表 6 顯示 AltaVista 的實驗結果比 Google 好，原因之一是在 AltaVista 的實驗，我們從測試文件刪除較多的指定實體，所以要辨識的指定實體個數 AltaVista 實驗比 Google 實驗少。另外，某些詞彙在 Google 的三個統計值 ( $pc(w)$ ,  $pc(w_{left})$ ,  $pc(w_{right})$ )，雖然都不等於零，但套進公式計算之後的結果，卻不能通過檢驗。例如「新華社」等詞彙，因而造成 Google 無法辨識出這些指定實體。對應之下，AltaVista 中幾乎不發生這種情形，唯一例外的例子是「羅俏」這個詞。



### 3.4. 分析與討論

#### 3.4.1. 測試資料之分析

在 MET-2 測試語料中，全部共有 1,301 個指定實體，單字詞因為只包含一個字元，所以無法套用公式。不包含重複，且可供檢測的指定實體共有 384 個，我們分析這 384 個指定實體，結果如表 7 所示。

表 7. 測試資料分析

	通過檢測		未通過檢測		網頁數等於 0		substring page count 較 superstring 少	
	AltaVista	Google	AltaVista	Google	AltaVista	Google	AltaVista	Google
PER	70	97	36	9	35	9	1	27
LOC	108	112	16	12	16	6	3	84
ORG	92	122	62	32	62	22	9	121
Total	270	331	114	53	113	37	13	236

就 AltaVista 而言，未通過的指定實體總共有 114 個，而網頁數等於零的有 113 個，猜測檢驗不通過的原因，可能是來自於搜尋引擎所收錄的網頁中沒有包含此詞彙的資訊。如果搜尋結果的網頁數大於零，幾乎大部分都通過檢驗。唯一的例外就是「羅俏」這個人名，雖然 AltaVista 可以搜尋到有關這個詞的網頁，可是這些數值代入公式後，並不通過檢驗。就 Google 而言，未通過的指定實體總共有 53 個，而網頁數等於零的只有 37 個，這表示有 16 個指定實體雖然在 Google 上可以搜尋到相關網頁，可是這些數值代入公式，並不會通過通過檢驗，例如「新華社」。

進一步分析網頁數等於零，以及通不通過檢定，和搜尋引擎內部的設計間的關係。考慮原始檢驗的三個重要參數： $w = c_1c_2...c_k$ 、 $w_{left} = c_1c_2...c_{k-1}$ 、及  $w_{right} = c_2c_3...c_k$ ，我們故意將組合成指定實體的辭彙頭尾各去除一個字元，例如「艾特納保險公司」，則  $w_{left}$  為「艾特納保險公」， $w_{right}$  為「特納保險公司」。在 substring 的結合性比 superstring 弱的假設下， $w_{left}$  和  $w_{right}$  的網頁數應該大於  $w$  的網頁數，可能比例測試就建立在這個假設之下。AltaVista 完全用字串比對，上述假設是成立的，其主要的問題是網頁數等於零的情況太多。猜測的原因是，地名和組織名都比較長。相對的，Google 經查詞處理，前述的假設可能就不成立，造成比例測試不通過，但其網頁數等於零的情況較少。

#### 3.4.2. 錯誤分析

因為某些指定實體本身包含停用字，例如「美國航空航天局」和「楊天」中的「天」是停用字，在這種情況下系統就無法正確地辨識出。某些外國人名的錯誤：「塞萬提斯」會變辨識成「萬提斯」，因為「萬」是中文人名姓氏。部分錯誤是因為指定實體太長，如「中國衛星發射測控系統部」會被斷詞成「中國 衛星 發射 測 控 系統 部」，「部」是組織名的關鍵字，我們最多只往前找五個詞，因此「衛星發射測控系統部」，就會被辨識為組織名，屬於左邊界錯誤。

有些錯誤是因為網頁數等於零所引起的，所以會造成 MIS 或 INC 的錯誤。像是把「美國艾科斯達衛星公司」，辨識為「斯達衛星公司」。有時候網頁數等於零不只會造成 INC 錯誤，同時也會增加 SPU 錯誤，例如「香港亞太通信衛星公司」，會被辨識為「香港(LOC) 亞太通信衛星公司(ORG)」；「中國空間技術研究所」，會被辨識為「中國(LOC) 空間技術研究所(ORG)」。有些指定實體的網頁數雖然大於零，可是卻不會通過公式檢驗，會造成 MIS 錯誤，如「新華社」和「歐洲宇航局」等。

我們進一步分析各種錯誤情形。首先是 INC 錯誤，這類型錯誤就是指系統辨識出的指定實體邊界錯誤，例如「西昌衛星發射中心」，辨識為「抵運西昌衛星發射中心」。造成這種錯誤的原因，包括指定實體本身包含停用字、網頁數等於零、指定實體長度太長等因素。

總結，造成遺漏錯誤的原因有：

1. 指定實體本身包含停用字。
2. 網頁數等於 0。
3. 指定實體本身沒有關鍵字，像是『法塔赫』武裝、五角大樓等，這類型的指定實體因為缺乏關鍵字，所以無法驅動指定實體辨識。
4. 有些國名的縮寫沒辦法辨識出，例如「中」、「以」、「日」等。
5. 有些指定實體的關鍵字，並沒有收錄在關鍵字集中，因此不會驅動指定實體的辨識工作，例如「日本科技廳」中的「廳」。
6. 外國人名不一定會被斷詞為連續的單字詞，因此無法藉由連續單字詞的策略來辨識。

多餘錯誤發生的原因有：

1. 網頁數等於零。
2. 通過公式檢驗，但實際上並不是專有名詞，例如「私人公司」。
3. 人造衛星的名稱「亞洲二號」，系統將「亞洲」標記為地名。
4. 連續單字詞的標記，常常會標記錯誤為人名。

由本實驗可以說明：全球資訊網的查詢結果，和搜尋引擎的內部設計有關。藉由關鍵字驅動辨識工作，透過搜尋引擎得到網頁數，套入可能比例測試中，可以判斷一個字串是否為指定實體。

## 4. 斷詞與指定實體辨識的整合

### 4.1. 實驗資源

CTS 語料是由華視新聞內容所收集而成，文本與中研院平衡語料庫採用相同的斷詞標準與詞性標記。指定實體，會被標記為連續詞彙，例如「美商奇異公司」，被斷為「美商/奇異/公司」，但是在自然語言處理應用，我們希望斷詞的結果是將所有具有「完整語意」的詞彙切分出邊界，即我們希望得到的是「美商奇異公司」，因此我們對於人名、地名、和組織名這三種型態的指定實體，以人工標記的方式，將這些被拆開的專有名詞，組合起來成為單一的詞彙。用此修改過的語料作為測試語料，觀察增加未知詞偵測，對斷詞結果所帶來的影響。

### 4.2. 實驗步驟

首先經過查字典的步驟，找出各種候選詞彙，然後掃描詞串，進行指定實體辨識。如果某些相鄰的詞串經辨識出為屬於人名、地名、或是組織名，表示我們偵測到未知詞，則將這些詞彙組合起來成為單一詞。最後進行解歧義性步驟，找出此句子的最佳斷詞組合。

### 4.3. 結果與討論

表 8 為實驗結果，1\_corpus、1\_google、1\_altavista 表示僅以辭典斷詞，未加上指定實體辨識模組的實驗。2\_google\_uw、2\_altavista\_uw，則是辨識出未知詞之後，才進行解歧義的實驗。2\_google\_corpus\_uw 則是以 Google 統計值辨識出未知詞，這些辨識出來的指定實體，將輸入句子切分成幾個小句子，即「S1 UW1 S2 UW 2 S3 ...」，我們再分別以語料庫統計值，對這些小句子 S1、S2 等進行歧義性分析。

完全使用辭典，亦即未加指定實體辨識，與第 2 節的實驗結果一致，傳統語料庫方法的效能 92.38%，仍然比全球資訊網方法 91.21% 和 90.31% 好。當考慮指定實體，整合進 Google 和 AltaVista 為基礎的斷詞中，效能分別增加 2.36% 和 2.11%，首先超越純傳統語料庫的方法 1.19% 和 0.04%。再考量傳統語料庫統計值，解決斷詞歧義性的效能，比全球資訊網統計值佳的現象，我們先以全球資訊網偵測未知詞的存在，再以傳統語料庫統計值來解歧義性，得到的結果為最佳

94.66%。

表 8. 增加未知詞偵測的斷詞結果

	Recall	Precision	F-Measure
1_corpus	94.03%	90.79%	92.38%
1_google	93.07%	89.43%	91.21%
1_altavista	92.45%	88.27%	90.31%
2_google_uw	94.11%	93.02%	93.57%
2_altavista_uw	93.42%	91.45%	92.42%
3_google_corpus_uw	95.02%	94.31%	94.66%

測試文件共包含 7,169 個詞，其中有 224 個未收錄在系統辭典內的詞彙。這些未知詞，包含普通名詞「等壓線」、「國際約」，或是動詞「拘提」、「收賄」，以及專有名詞「華視晚間新聞」、「魏政賢」等。我們利用 web 資訊，可以辨識出未知詞，例如「魏政賢」、「大傑旅行社」、「中山足球場」等。但是對於其他型態的未知詞，則無法切分出正確邊界，因此「等壓線」會被斷為「等壓線」，造成斷詞錯誤。由於中文人名辨識模組，只能找到長度為 2 或是 3 的人名，因此「張趙惠朱」無法正確的辨識出來，造成錯誤。另外由於某些指定實體中含有停用字，所以無法辨識成功，例如「彭南雄」斷為「彭南雄」。

## 5. 結論

本論文應用全球資訊網的統計值於自然語言處理上，並以中文斷詞為例。解歧義性實驗顯示傳統語料庫方法，得到的結果優於全球資訊網方法，但兩者差距不大。針對人名、地名、和組織名三種類型設計出一套指定實體辨識，實驗顯示辨識成功與否，取決於搜尋引擎的內部結構，和收錄的網頁內容。如果指定實體曾出現在網路上，則它的網頁數大於零，有很大機會通過公式檢驗，被成功地辨識出來。由於全球資訊網資訊量龐大、且具及時性，對指定實體辨識有很大潛力。

最後將斷詞和指定實體辨識系統整合，運用指定實體辨識模組偵測未知詞，予以正確的邊界切分。實驗顯示在原本的斷詞系統中加入未知詞偵測，對於斷詞效能是有助益的。由於傳統語料庫統計值解決歧義性問題的效能，略勝於全球資訊網統計值，而全球資訊網統計值可以用於指定實體辨識，因此結合雙方面的優點，先利用全球資訊網統計值偵測未知詞，再利用傳統語料庫解歧義性，使斷詞系統得到最佳的表能。

由於傳統語料庫資訊量不夠龐大、不具即時性、且需要大量人工標記、耗時，相較之下全球資訊網擁有資訊量龐大、即時性、且取得容易等優勢，本文提出全球資訊網方法解決中文斷詞問題，不需要太多語言知識，只需要透過搜尋引擎介面得到網頁數，視之為詞頻應用於統計模型上，實作容易。實驗顯示全球資訊網資訊，在自然語言處理上是有用的。

## 註謝

本文部分成果為國科會計畫 NSC 93-2752-E-001-001-PAE 補助。

## 參考文獻

- [1] Hsin-Hsi Chen, Yung-Wei Ding and Shih-Chung Tsai (1998). "Named Entity Extraction for Information Retrieval." *Computer Processing of Oriental Languages, Special Issue on Information Retrieval on Oriental Languages* 12(1), 1998, 75-85.

- [2] Hsin-Hsi Chen, Changhua Yang and Ying Lin (2003). "Learning Formulation and Transformation Rules for Multilingual Named Entities." *Proceedings of ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models*, July 12, Sapporo, Japan, 2003, 1-8.
- [3] CKIP (1995). A Description to the Sinica Corpus. *Technical Report 95-02, Academia Sinica, Taipei*.
- [4] F. Keller and M. Lapata (2003). "Using the Web to Obtain Frequencies for Unseen Bigrams." *Computational Linguistics* 29(3), 459-484.
- [5] Christopher D. Manning and Hinrich Schutze (1999). Foundations of Statistical Natural Language Processing, *MIT Press*, 1999.
- [6] MUC (1998). *Proceedings of 7<sup>th</sup> Message Understanding Conference*, Fairfax, VA, 29 April - 1 May, 1998, [http://www.itl.nist.gov/iaui/894.02/related\\_projects/muc/index.html](http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html).
- [7] P. Resnik and N. A. Smith (2003). "The Web as a Parallel Corpus." *Computational Linguistics*, 29(3), 349-380.
- [8] X. Zhu and Ronald Rosenfeld (2001). "Improving Trigram Language Modelling with the World Wide Web." *Proceedings of the International Conference on Acoustics Speech and Signal Processing*.