基於統計與佚代的中英雙語詞及小句對應演算法

黃子桓　　高照明
臺灣大學資訊工程學系　臺灣大學外國語文學系
tzhuan@csie.org　　zmgao@ntu.edu.tw

Abstract

本文提出一基於段落對應的雙語語料中迭代進行詞對應及小句對應 (subsentential alignment) 之模型， 並提出可行的實作方式。和基於句對應的詞對應演算法相比， 本文提出的演算法不須經過句對應，在現實應用有更大的彈性。 而和基於字典的句對應演算法相比，本文提出的演算法不須額外的字典支援， 完全藉由本身的統計資訊進行詞條的蒐集和利用。 實驗結果顯示詞對應和 K-vec 相比有較佳的 precison 和 recall 值， 而小句對應結果顯示約有 77.74% 的小句對應是完全或部份正確。

1　導言

語言翻譯在資訊的傳遞上扮演十分重要的角色，在過去，語言翻譯的工作皆以人工翻譯爲主。 由於電腦科學的進步，運算能力大幅提高，各類相關的理論、演算法也相繼被提出， 如何利用電腦來進行自動翻譯工作成了重要的研究課題。 在許多自動翻譯的研究中，詞對應 (word alignment) 是不可或缺的重要步驟， 其正確率往往對翻譯的結果有關鍵性的影響。 傳統詞對應乃是由人工所建立，如雙語詞典即是人工建立的詞對應資料庫。 但人工建立不但費時費力，難以跟上新詞增加的速度， 且詞典有其極限，再完善的詞典皆不可能包含所有雙語詞彙對應。 加以現今網路上已有大量的雙語機讀資料，在研究資料充沛的情形下， 由電腦自動建立詞對應亦爲一重要的研究方向。

現有的電腦自動建立詞對應研究中，有許多是基於已正確句對應的研究， 並且取得不錯的研究成果。 然而要達到正確句對應並不容易，以人工標示費時費力， 且在現實環境中，並不保證有正確句對應。而以機器自動句對應的演算法， 基於語言的特性，不同性質的文章，其正確率的變動非常大 (McEnery and Oakes, 1996)， 與廣泛應用的水準尚有差距。 以詞對應的角度來看，正確的詞對應有助於句對應；而從句對應來看， 正確的句對應對詞對應也是十分正面的助益。 句對應和詞對應可說是雞生蛋、蛋生雞的問題。 本研究主要探討如何在同一語料中同時進行句對應及詞對應，並藉由彼此提高正確率。 我們選擇使用已正確段落對應的中、英語料庫，理由如下：

1. 基於翻譯的習慣，以句爲單位來看，往往會有增減的情形，但若以段落爲單位來看， 則較少有增減的情形。因此一般的翻譯文章或者已正確句對應， 或者經過極少的工作即可達到正確段落對應，對於現實的應用有很大的幫助。
2. 由於語言的結構關係，段落與段落間往往有利於機器處理的分隔符號存在， 因此機器自動分段可達 100% 正確。因此使用正確段落對應的語料庫， 在分段上幾乎不會有失敗的情形發生。

2 相關研究

2.1 詞對應演算法

Fung 與 Church (1994) 的 K-vec 演算法將雙語語料庫各切分爲相等的 K 區塊， 每一詞皆記錄該詞在 K 個區塊中出現與否， 組成一 K 維 vector $(v_1, v_2, \ldots, v_K)$，$v_i \in \{0, 1\}$。 對雙語的兩兩詞彙，皆透過彼此 vector 計算各自頻率及共同出現於相同區塊的頻率， 並以 MI (Mutual Information) 來計算兩詞彙的相依程度。 由於 MI 對於頻率甚少的詞會計算出極大值，嚴重影響可信度， 因此藉由 t-score 值修正，透過給定的常數值，忽略 t-score 值小於該常數值的結果，將大大提昇 MI 的可信度。

由於 K-vec 演算法需要切割雙語語料成 K 個區塊，錯誤的切割將使結果不如預期。 因此 Fung 與 McKeown (1994) 再提出 DK-vec 以解決此問題。 在 DK-vec 中每一詞彙皆記錄兩 vector， position vector 記錄該詞彙出現於雙語語料的所有位置， recency vector 則記錄兩兩位置的距離。 以 position vector 的資料爲橫座標值，recency vector 的資料爲縱座標值， 並連接相鄰之點，則可得一分佈於 2-D 座標系的函式分佈取樣圖。 利用 pattern matching 的 Dynamic Time Warping 的技術， 可計算兩兩詞彙函式分佈取樣的相似程度， 從相似程度的高低可得兩詞彙的相依值。

Melamed (1998) 的 Competitive linking algorithm 是基於已正確句對應的詞對應演算法。 對於雙語的兩兩詞彙，competitive linking algorithm 使用 LLR (Log-Likelihood-Ratio) 來評估其相依程度。 當一雙語對應句中兩兩詞彙的相依權值皆計算完畢， 將所有雙語詞彙對由相依權值大至小排序，依序取出雙語詞彙對，若兩詞彙皆未與其它詞彙連結， 則連結此兩詞彙，否則忽略並處理下一詞彙對，直到所有的詞彙對皆已處理完畢。

2.2 句對應演算法

雙語句對應的研究開始於 90 年代初期。 Gale 與 Church (1991) 及 Brown 等 (1991) 觀察到長句的翻譯對應句一般而言較長， 而短句的翻譯句通常較短。他們利用句長的關連性配合動態規劃或 EM 演算法得到 96% 以上的正確率。 Gale 與 Church (1991) 及 Brown 等 (1991) 兩者最大的差別是前者透過人工先得到先驗機率 (prior probability) 而後者利用 EM 演算法得到相關的參數。Wu (1994) 及 Xu and Tan (1996) 以句長爲主結合一個包含日期及數字等訊息小的辭典得到 96% 的正確率。 以句長爲基礎的統計方法的優點是不需要語言知識及辭典就可以運作。 缺點是如果語料中含有豐富的多對多的句對應關係， 或是翻譯的語料中有增添或刪減的現象發生就會造成正確率大幅下降。 前述幾項研究由於大都採用議會的紀錄， 例如 Gale 與 Church (1991) 及 Brown 等 (1991) 用加拿大國會 Hansard 英法平行語料， Wu (1994) 則利用香港立法局議會質詢與答詢的中英平行語料， 由於是口語紀錄所以句子較短，且不少是一對一對應。 Gale 與 Church (1991) 統計 Hansard 語料 80% 以上是一對一的對應關係， 罕有多對多的對應關係或增添或刪減的情形發生， 所以以句長爲主的統計方法得到很好的效果。 但 McEnery and Oakes (1996) 以 Gale 與 Church (1991) 的方法做實驗卻顯示此種演算法的正確率對不同的文類與語言會產生很大的差異。 例如波蘭文英文平行語料的正確率因文類不同介於於 100% 與 64.4%， 而他們所實驗的中英新聞平行語料更低於 55%， 這證明單純以句長關連性顯然無法得到高正確率。

另一個不需要辭典的方法是 Kay and Röscheisen (1993) 以詞彙的頻率 (去除低頻的詞及高頻

的詞) 及在文章中出現的分佈， 建立可能的詞對應表及句對應表並不斷的修正， 以 relaxation 方法達到收斂。 與 Gale 與Church (1991) 及 Brown 等 (1991) 方法一樣， Kay and Röscheisen (1993) 的方法只有在一對一的情形佔絕大多數時才會有好的效果。 此外此種方法過度重視詞頻，文章的長度太短會造成正確率的大幅下降。 這個演算法另一個實做上的問題是處理十分耗時，無法快速處理大量語料。

子句對應 (clause alignment) 則如 Kit et al. (2004) 以雙語法律條文的 glossary 和雙語辭典，再加上適當的標點符號轉換、 數字轉換 (如阿拉伯數字與羅馬數字)，再設計一估計函數來結合全部資訊而得相似程度， 以其評估字句對應，可達 94.6% 的正確率。 Kit et al. (2004) 以詞彙訊息得到非常高的小句對應正確率的主要原因是所用的語料爲法律雙語文件且使用法律術語的辭典， 且此類文件中代表法律條文的數字一再出現。 在我們之前的實驗 (林與高 (2004)) 顯示在一般的中英雙語文章使用雙語辭典、數字、 及標點訊息在大句的對應正確率尚且不到 90%， 小句的正確率必定無法達到 Kit et al. (2004) 的水準。

Wu et al. (2004) 則提出利用句長和標點符號進行小句對應 (subsentential alignment)， 加上雙語中的同源資訊 (如雙語中相同的數字部份)， 以香港立法局會議記錄爲實驗資料，可達 98% 的正確率。 Wu et al. (2004）所用的英漢對譯語料爲香港立法局的議會紀錄， 內容全是議員與官員之間一問一答的紀錄。此類議會語料多屬逐句翻譯， 且少有意譯的情形，由於採一問一答及逐句翻譯在句對應及小句對應比較容易。 如用文章之類的對譯語料該演算法勢必無法得到如此高的正確率。

3 段落對齊平行語料的詞對應暨小句對應演算法

3.1 段落對齊平行語料的詞對應演算法

在 Association-based binlingual word alignment 中， 詞彙的出現頻率扮演著關鍵的角色。 不論使用 MI、t-score 或者 LLR 來評估兩詞彙的相依程度， 皆利用頻率的資訊來計算。 而另一關鍵的角色則是文章的切分。 將文章切分成若干區塊，提供了一個強烈的假設及限制， 即該詞彙若有詞對應，必然出現於同一區塊中； 正確的切分方式，能使詞彙的相依程度提高，反之則會降低。 基於上述說明，我們設計一演算法， 在現有的區塊中，尋找一切分方式，可使總體相依值提高最多， 不斷重覆此一過程直到所有切分方式都無法再使總體相依值提高。

令 $E = B_1^e B_2^e \ldots$、$C = B_1^c B_2^c \ldots$， 其中 $B_i^{e|c}$ 表示一英文 (中文) 區塊， 稱此時的切分狀態爲 Ω。 令 $B_i^e = e_{i,1} e_{i,2} \ldots$，$B_j^c = c_{j,1} c_{j,2} \ldots$， 其中 $e_{i,k}$ ($c_{j,l}$) 表示一英文 (中文) 詞彙。 令 $asso(e, c)$ 表示詞彙 e 和詞彙 c 的相依權值大小 (此相依權值可視需要選用如 MI、t-score、LLR 等。 在本實驗中我們以 MI 爲主，搭配 t-score 以過濾詞頻低的對應)， 則 $ASSO(\Omega) = \sum_i \sum_j asso(e_i, c_j)$ 即爲在 Ω 切分狀態下的總體相依值。 令 $new(\Omega, i, start_e, end_e, start_c, end_c)$ 表示一種新的切分狀態， 其意義爲在 Ω 切分狀態中，第 i 區塊被切分了， 切分方式爲 $e_{i,start_e} e_{i,start_e+1} \cdots e_{i,end_e}$ 和 $c_{i,start_c} c_{i,start_c+1} \cdots c_{i,end_c}$ 爲一組對應區塊， 而 $e_{i,1} e_{i,2} \ldots e_{i,start_e-1} e_{i,end_e+1} \cdots e_{i,|E_i|}$ 和 $c_{i,1} c_{i,2} \cdots c_{i,start_c-1} c_{i,end_c+1} \cdots c_{i,|C_i|}$ 爲另一組對應區塊。 因此，對 Ω 狀態而言，計算

$$value = \max_{\substack{1 \le start_e \le |E_i| \\ 1 \le start_c \le |C_i| \\ start_e \le end_e \le |E_i| \\ start_c \le end_c \le |C_i|}} ASSO(new(\Omega, i, start_e, end_e, start_c, end_c)) \quad \forall i = 1, 2, \ldots, |\Omega|$$

若 $value > ASSO(\Omega)$，即表示該切分方式能夠提高總體相依值， 依此時之 $start_e$、 end_e、 $start_c$、 end_c 進行切分， 可得一新的切分狀態 Ω'。若 $value \le ASSO(\Omega)$， 表示所有的切分方

式都無法再提高總體相依權值，因此該區塊沒有再被切分的必要。 重覆此一步驟，則切分之區塊數將會不斷增加， 直到所有的區塊都無法再被切分。

演算法如下：

1. 以雙語語料的段落對應作爲初始切分狀態。
2. 在目前切分狀態 Ω 中，對每一區塊進行切分嘗試，並記錄新的切分方式於 Ω'。
3. 如果 $|\Omega| = |\Omega'|$ 則結束，否則回到 2.。
4. 利用目前切分狀態求出詞彙間的相依權值並輸出結果。

3.2 段落對齊平行語料的詞對應暨小句對應演算法

在上述演算法中，如果加上特殊的限制條件，則可使切分區塊的自由度降低，形成特定的區塊。 例如限制 $start_{e|c}$ 的前一個詞必須是分句符號 (如句號、問號、驚嘆號等)， $end_{e|c}$ 後一詞也必須是分句符號，則所得的區塊對將成爲句對應或多句對應形式。 亦即此演算法爲一詞對應暨句對應之演算法。

3.3 加速與實作

在上述演算法中，由於要對所有可能的切分方式計算 $ASSO$ 值， 亦即對於所有可能的切分方式都要執行一次類似 K-vec 的演算過程， 則此演算法的計算複雜度將會十分地高， 在實作上雖然並不困難，但計算時間將會十分地久。 而若是加上對 $start_{e|c}$ 及 $end_{e|c}$ 的限制，將會有效減少可能切分方式的總數。 然而計算時間仍然相當長，因此難以取得廣泛應用。 在此我們提出一個加速的作法。

考慮上述的理論架構，對於每個可能的切分方式都要重新計算 $ASSO$ 值， 顯然付出太大的代價。 重新計算 $ASSO$ 值的理由在於這是一個足夠好、可信賴的評估方式， 可有效評估現行分割方式的優劣。 因此加速的關鍵即在於使用新的評估方式， 新的評估方式需滿足下列條件：

1. 和 $ASSO$ 相比同樣可被信賴。
2. 計算複雜度要低。

我們所提出的新評估方式說明如下：

在一個區塊中，我們可以指定的標點符號 (例如句號、問號等) 將區塊再切分爲較小的區塊 (可能包含一或多個句子)，稱爲子區塊。令 $B^e = S_1^e S_2^e \dots S_m^e$ 和 $B^c = S_1^c S_2^c \dots S_n^c$，其中 B^e 和 B^c 爲雙語語料中對應的其中一區塊； 而 $S_i^{c|e}$ 表示子區塊。令 $W_i^{e|c} = \{w_{i,1}^{e|c}, w_{i,2}^{e|c}, \dots\}$ 表示在 $S_i^{c|e}$ 子區塊中，所有相異詞彙所成的集合。 定義

$$score(S_i^e, S_j^c) = \sum_{e \in W_i^e} \sum_{c \in W_j^c} asso(e, c)$$

其中 $asso(e, c)$ 的定義如前所述。 則藉由求出

$$(max_e, max_c) = arg \max_{\substack{1 \le i \le |B^e| \\ 1 \le j \le |B^c|}} score(S_i^e, S_j^c)$$

可知，在現行條件下，$S^e_{max_e}$ 對應 $S^c_{max_c}$ 是最可信賴的。 因此，將 $S^e_{max_e}$ 與 $S^c_{max_c}$ 取出使成新的區塊。 對於每一區塊，重覆這個步驟，直到區塊的總數不再變動。

$asso(e, c)$ 乃根據目前爲止的詞對應相依權值來計算； 而接著詞對應乃根據新的切分狀態來求其相依權值。交互迭代後將會收斂， 亦即兩者皆不再變動。由於我們的切分是以標點符號切分的子區塊爲單位， 因此若目前處理區塊只包含一個子區塊，則不可再被區分， 因此該演算法保證會收斂。 而初始的段落對應則用於提供最初的相依權值計算， 此外也保證初始的區塊對應是完全正確的。

4 實驗材料

本研究使用的中英對譯文章取自光華雜誌 (http://www.sinorama.com.tw/ch/)，統計資料如下：

	段落數	總詞數	相異詞數
中文	59	3291	1192
英文	59	3908	1082

英文分詞以空白和標點符號爲主，搭配常見縮寫詞以減少分詞錯誤。 計算英文相異詞時則以一般變化規則 (-s -ing -ed 等) 加上不規則動詞變化表來還原各詞類原形。 中文分詞則以中央研究院中文斷詞系統 (http://ckipsvr.iis.sinica.edu.tw/) 來進行分詞。 該系統提供線上使用，爲現階段中文分詞正確率最高的系統之一。

5 實驗結果與討論

我們實作了我們所提出的演算法，並實作 K-vec 演算法以進行比較。 在 K-vec 演算法中，由於作者建議切分區塊數 $K = \sqrt{\text{total word number}}$ 會有較好的結果， 而在我們的實驗資料中，段落數的平方 ($59 \times 59 = 3481$) 恰約等於總詞數 (中文 3291、英文 3908)， 因此我們以段落作爲 K-vec 的區塊。 相依權值使用 MI 及 t-score 判別， MI 及 t-score 的參數比照原作者的建議：以 t-score 值爲篩選器， 只考慮 t-score ≥ 1.65 的詞對應。MI 則做爲主要相依權值的依據， 輸出時以 MI 的值由大到小排序，並捨棄 MI < 1.0 的結果。 在這個條件下的輸出如下表所示：

演算法	t-score 篩選值	MI 最小值	詞對應數	正確數	precision
K-vec	1.65	1.0	28	12	0.42
ours	1.65	1.0	79	32	0.40

雖然在 MI ≥ 1.0 的輸出條件下，我們的演算法 precision 較低， 但由 Figure1 和 Figure 2 可看出在同樣個數的輸出 (輸出皆以 MI 的權值大小爲序)下，我們的演算法有較好的表現。 在輸出前 10 條詞對應時，我們的演算法和 K-vec 差異不大， 但從第 10 條詞對應之後的輸出結果，明顯我們的演算法有更高的正確率， 到前 30 項輸出仍維持 0.6 以上的 precision， 而在前 30 項輸出時 K-vec 的 precision 僅約 0.45。

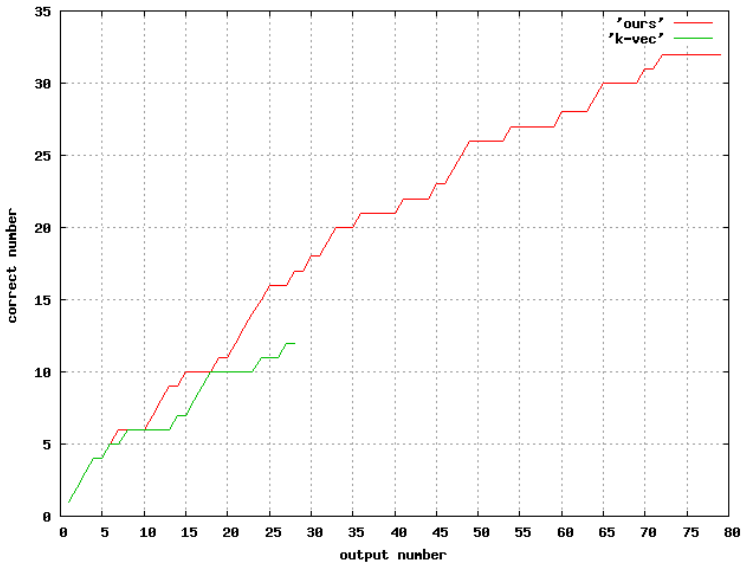

Figure 1: 輸出結果數與正確數關係圖

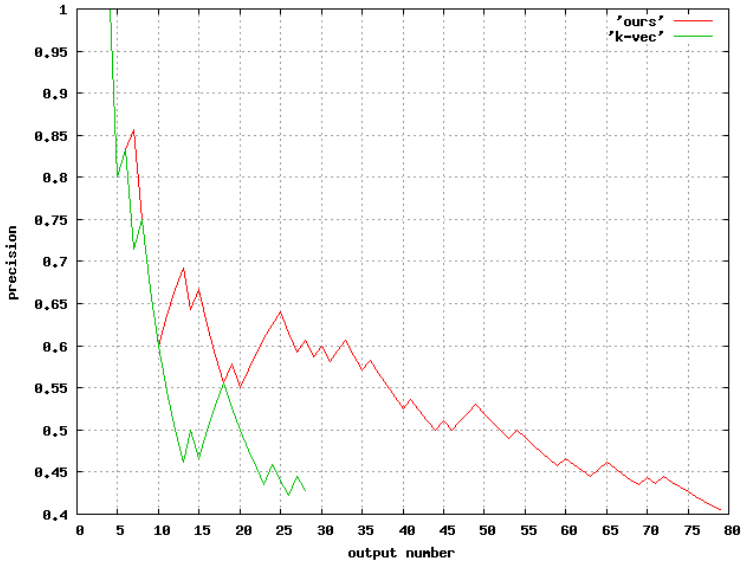

Figure 2: 輸出結果數與 precision 關係圖

由下面列舉的詞對應結果可以看出，部份正確的詞對應佔了相當的比例， 如果加上這些複合詞的擷取演算法，則可望大幅提升 precision。 此外，由於基於統計的相依權值計算相當依賴詞頻，過多或過少都會影響其信心。 以本實驗爲例，詞頻甚低的正確詞對應有可能無法通過 *t*-score 的篩選門檻； 而詞頻不夠高的功能詞對應也可能仍有相當高的 MI 值以致於未被過濾。 詞頻太低是所有基於統計的詞對應都會面臨的困難問題， 因爲過低的詞頻並沒有辦法分辨其爲偶然或是正確； 而功能詞的部份可用預先建立的功能詞列表來解決。

在 recall 方面，由於 recall 的計算需要以人工找出雙語語料中所有正確的詞對應， 在此實驗資料中，共有 1192 個相異中文詞、1082 個相異英文詞， 基於時間及人力的關係無法以人工標記此實驗資料的所有正確詞對應， 然仍可知在分母相同的條件下，我們的演算法有較高的 recall 值。

由於翻譯的關係，雙語對譯的用詞可能相當靈活， 例如同一個動詞卻在不同位置用不同的詞彙翻譯， 以致於所有的對譯詞頻都不高，而無法找出正確詞對應。 由於此原因，許多詞彙無法找出正確詞對應， 而最利於找出的則是有固定翻譯及適當詞頻的專有名詞。

以下試列舉出前 18 條由我們提出的演算法所得到的詞對應結果， 其正確對應與否於附註中說明，若部份正確則在附註中顯示正確之詞對應。

asso	英文詞	中文詞	附註
6.363	table	桌	正確
5.948	Kuo	郭慧明	正確
5.626	hope	希望	正確
5.141	Jen-an	人安	正確
4.948	each	天	each day 每天
4.877	volunteers	義工	正確
4.778	day	天	正確
4.778	goal	努力	錯誤
4.725	fund	經費	正確
4.626	Jen-an	基金會	The Jen-an Foundation 人安基金會
4.626	each	每	正確
4.626	month	月	正確
4.533	even	甚至	正確
4.488	but	長期	錯誤
4.404	welfare	社福	social welfare 社福
4.247	social	社福	social welfare 社福
4.141	elderly	失	elderly people 三失老人
4.041	day	每	each day 每天

在分句演算法方面，我們的原始語料共有 59 段落， 在實作中，我們指定這些符號 . , ; ! ? 。 ， ； ！ ？ 作爲切分區塊的標點符號限制。 經過我們的演算法，最後收斂時共輸出 265 個對應區塊。 下表是經由我們的演算法的分句結果與原始文章以 . ; ! ? 。 ； ！ ？ 分句的比較結果， 中文部份我們用兩組標點符號來分句，其中一組包含逗號，另一組不包含：

		句數	句平均詞數	標準差
原始文章以標點分句	中文 (使用逗號)	396	8.300	4.166
	中文	104	31.615	18.074
	英文	165	23.666	12.524
以本演算法分句	中文	265	12.384	9.683
	英文	265	14.709	9.172

在利用 regular expression 或其它方法解決英文縮寫點 (如：Mr. 或 I.B.M.) 的問題之後， 英文基本上可以靠句號，問號，驚嘆號，分號當作分隔句子的界限。 中文的句子無法像英文一樣靠標點符號來判斷。 原因是逗點在中文使用的非常的鬆散， 逗號和句號的使用是作者風格的問題而非文法的問題。 如果用句號、問號、驚嘆號、分號來分的話， 很多是比句子更大的言談單位 (discourse) ， 如果加上逗號的話又會造成許多只是詞組而不是句子。 這就是爲什麼當我們用 。 ！ ？ ；來分割句子時， 中文句數比英文句子少很多， 而中文加逗點作爲分隔句子界限之後又比英文句子多很多的原因。 從以上的討論，我們可以看出經過我們的分句演算法所得到的是比句子還要小的區塊， 可視爲一種小句對應的結果。

由於我們的演算法並不保證按順序對應， 因此輸出結果並不按原始文章的順序。另外基於我們演算法的特性， 不相鄰的區塊有被合而爲一的可能。因爲上述原因， 要對輸出的對應區塊分析其正確分句程度極爲困難。 因此我們採用較簡單的估計方式， 以人工標記 265 句中完全正確、部份正確及完全錯誤的小句對應， 完全正確表示該對應是最小可能的切分方式，例如「 E_iE_{i+1} 正確對應 $C_jC_{j+1}C_{j+2}$ 」即表示不論是 E_iE_{i+1} 或 $C_jC_{j+1}C_{j+2}$ 皆無法再切割以得到更小的正確對應。部份正確對應以上述正確對應爲例， 任意 $\{E_i, E_{i+1}\}$ 的子集合對應任意 $\{C_j, C_{j+1}, C_{j+2}\}$ 的子集合都可視爲部份對應。若非上述兩種情況，則稱爲錯誤對應。 實驗的結果統計如下：

	對應數	所佔全體比例
總對應數	265	-
完全正確	59	22.26%
部份正確	147	55.47%
完全錯誤	59	22.26%

由於我們的小句對應是基於「完全對應」，即任一小區塊皆必對應於某區塊， 且僅對應於該區塊。因此任一區塊若爲部份正確， 則必然會影響另一區塊爲部份正確或完全錯誤， 因此部份正確數佔了極大比例是可預期的結果。

以下試舉部份小句對應結果：

英文區塊	中文區塊
完全正確	
even into his old age	甚至 在 遲暮 之 年
whenever cswf has needed them	在 創世 有 需要 時
the service hua-shan offer the elderly are of two variety	華山 照顧 老人 的 方式 有 兩 種

部份正確	
it is also renowned as a “master fundraiser” and admire by other social welfare organization for operate at a surplus year after year	還被喻爲「募款高手」
not only is cswf well known for its service how do they do it	他們是怎麼做到的
we finally reach the pvs hospice	來到植物人安養中心：創世的發源地
完全錯誤	
at the start	幫幾個家庭喘口氣而已
cswf has open branch hospice around the country	創世的目標是全省２３個縣市都有植物人安養院
thus far they have complete 13	籌備中的有４個

6 結論與未來研究方向

我們的研究展示了一個不必依賴正確句對應，也不必依賴字典的迭代詞與小句對應演算法。 相較於依賴句對應的詞對應演算法，我們的演算法不必經過人工或機器的句對應， 可有效減少工作量，並且避免了由錯誤句對應所引發的錯誤。而相較於依賴字典的句對應演算法， 我們的演算法如同一邊分句一邊建立小型字典，除了不需額外資料庫外， 對於字典沒有的新字我們的演算法仍能透過統計的方式得到相依關係， 因此擁有較大的彈性可適應不同種類的文章。

在實驗結果裡，和同樣不需已句對應的 K-vec 演算法相比，我們提出的方法有較佳的 precision 值， 且在同樣的條件下能找出更多正確的詞對應，亦即有較佳的 recall 值。 而在句對應中，結果顯示有許多輸出是詞組和子句的對應， 換言之我們的演算法能得到小句對應， 這是目前大部分基於統計演算法不容易做到的。

未來的研究方向擇要列舉如下：

1. 目前我們的模型僅標示出一對一的詞對應， 實際上詞對應有很大的機會是多對多，尤其是具有特定翻譯的專業詞彙。 對於這些複合詞，若能在迭代過程中取出，則可增加詞對應的信心， 進而對句對應有正面的助益。 因此如何利用此模型來運用複合詞資訊，將是未來研究的方向之一。
2. 由於演算法的特性，在切分的情形下，會將原本不連續的區塊合併。 對詞對應而言，這個合併的動作並不會造成太大的影響， 但對句對應而言此動作並不恰當。而這個問題可透過修正的切法方法來解決，例如， 當找到最有信心的子區塊對應時，將該區塊切分成三組新對應而不是兩組， 可避免合併的動作。
3. 本研究基於兩前提：正確段落對應及正確分詞。 由於現實語料的支援，正確段落對應可視爲合理的假設，分詞對英文而言也有很高的正確率， 然而分詞對中文而言遠較英文困難，正確率也遠不及英文。 錯誤的分詞結果將改變詞頻，對詞對應的結果有很大的影響。 如何降低對分詞正確性的依賴是我們未來研究的課題。

4. 本研究的理論模型乃「不可迴溯式」，如果在過程中發生錯誤的切分， 則該錯誤會永久保留，甚至可能會擴散。雖然過程中每個步驟都儘可能選取最有信心的切分方式， 但不可避免一定有發生錯誤的可能。如果能在現有模型上加入可事後補救的機制， 將可使穩定度更爲提升。
5. 除了經統計所得的訊息外，在一般的雙語語料中常常還有其它的訊息可供利用， 例如數字、未翻譯的人名、地名、專業詞彙等，這些訊息比統計所的的詞對應更爲可靠， 因此在我們提出的演算法中結合這類訊息的使用，我們預期能得到更好的結果。

致謝

本研究得到國科會 NSC93-2815-C-002-063H 「從中英平行語料庫自動擷取雙語詞組知識」及 93-2411-H-002-013 「 詞彙語意關係之自動標注—以中英平行語料庫爲基礎(3/3)」經費補助，特此致謝。

參考資料

[1] Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In *Meeting of the Association for Computational Linguistics*, pages 169–176, 1991.

[2] R. Catizone, G. Russell, and S. Warwick. Deriving translation data from bilingual texts. *Proceedings of the First Lexical Acquisition Workshop*, 1989.

[3] B. Chang, P. Danielsson, and W. Teubert. Extraction of translation unit from chinese-english parallel corpora. *COLING-02: The First SIGHAN Workshop on Chinese*, 2002.

[4] P. Fung and K. Church. K-vec: A new approach for aligning parallel texts. *COLING-94: 15th International Conference on Computational Linguistics*, pages 1096–1102, Aug 1994.

[5] P. Fung and K. McKeown. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. *AMTA-94, Association for Machine Translation in the Americas*, pages 81–88, 1994.

[6] W. Gale and K. Church. A program for aligning sentences in bilingual corpora. *Proceedings of the Annual Conference of the Association for Computational Linguistics*, pages 177–184, 1991.

[7] M. Kay and M. Röscheisen. Text-translation alignment. *Computational linguistics*, (1):121–142, 1993.

[8] C. Kit, J. J. Webster, K-K. Sin, H. Pan, and H. Li. Clause alignment for hong kong legal texts: A lexical-based approach. *International Journal of Corpus Linguistics*, (1):29–51, 2004.

[9] I. D. Melamed. Models of co-occurrence. *IRCS Technical Report*, 1998.

[10] I. D. Melamed. Models of translational equivalence. *Computational Linguistics*, pages 221–249, 2000.

[11] Robert C. Moore. Association-based bilingual word alignment. *Proceedings, Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond, Ann Arbor, Michigan*, pages 1–8, 2005.

[12] D. Wu. Aligning a parallel english-chinese corpus statistically with lexical criteria. *Proceedings of the 32nd Annual meeting of the Association for Computational Linguistics*, 1994.

[13] Jian-Cheng Wu, Thomas C. Chuang, Wen-Chi Shei, and Jason S. Chang. Subsentential translation memory for computer assisted writing and translation. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 106–109, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[14] D. Xu and C. L. Tan. Automatic alignment of english-chinese bilingual texts of cns news. *Computational Linguistic Archive*, 1996.

[15] 林語君 and 高照明. 結合統計與語言訊息的混合式中英雙語句對應演算法. *ROCLING*, 2004.