第四章　實驗結果與分析

雖然中文斷詞的整體效能評估具有許多不確定性，如詞彙混淆或對「詞」認定的不同，都將左右斷詞效能的評估，然後為了較客觀地比較斷詞、構詞的各項影響差異，我們仍針對本文所提語音合成用的中文斷詞器各單元，設計如下幾個實驗，並嘗試分析，期能提供未來努力的方向。

4.1 實驗資料庫及效能定義

本文用來測試斷詞效能的資料庫，為由民國九十二年七月二十日起，至九十三年六月五日止，共 4,884 篇中時電子報焦點新聞中，以隨機方式抽取而出的五十篇新聞，總計共 42,007 個字，並委請新竹清華大學中國文學系同學，依其所受的專業語言學訓練，及對文字的敏銳感以人工方式斷詞，作為我們斷詞效能評估的依據。而在效能的評估方面，我們定義辨識率如下：

$$\text{召回率(Recall)} = \frac{\text{斷詞系統正確斷出詞的個數}}{\text{標準答案的中文詞個數}} \times 100\ \%$$

$$\text{精確率(Precision)} = \frac{\text{斷詞系統正確斷出詞的個數}}{\text{斷詞系統斷出詞的個數}} \times 100\ \%$$

4.2 長詞優先與動態規劃演算法的斷詞單元

在整個斷詞系統中，斷詞單元為主要的核心，其後的構詞單元是以規則的加入，來彌補斷詞單元的不足。故此，我們暫將構詞單元完全移除，藉以觀察斷詞單元中的長詞優先與動態規劃演算法的單獨表現。

斷詞方法 / 資料庫與效能		Longest Word First		Dynamic Programming	
		Recall	Precision	Recall	Precision
MIR DB	效　　能	81.95%	71.97%	80.06%	69.32%
	斷詞時間	118.4370 sec.		30.3910 sec.	
Sinica DB	效　　能	84.49%	76.21%	81.10%	74.81%
	斷詞時間	154.0780 sec.		36.4530 sec.	

表 4-1 長詞優先與動態規劃演算法效能實驗分析

在表 4-1 的實驗中，由於敝實驗室的語音資料庫，即 MIR DB 中未含有詞頻的資訊，因此在對 Dynamic Programming 的實驗中，我們採以自訂分數的方式給予權重。實驗中我們可發現，動態規劃演算法的計算時間遠低於長詞優先法，然而平均召回率，不論是對自訂分數的 MIR 語音資料庫，抑或含有詞頻訊息的中研院平衡語料庫，均掉落約 2~3%。

而在長詞優先的斷詞演算法中，我們也發現詞庫的豐富性，雖延長了辨識時間，卻也相對提高了斷詞的正確性。詞庫的收集、整理對斷詞系統之重要性可見一斑。

4.3 定量複合詞的構詞單元

在 3.3.1 章節定量複合詞的構詞介紹中曾提及，為了增加斷詞系統的整體效能、減低詞庫比對的時間，我們將原有的實驗室語音資料庫，及中研院的平衡語料庫，依定量複合詞的構詞規則，分別整理、刪減為 86,779 及 124,640 個。以下的實驗是繼 4.2 章節的實驗後，於斷詞單元後再分別加上定量複合詞的構詞單元，藉以比較、分析定量複合詞的構詞效能。

斷詞方法 / 資料庫與效能		Longest Word First		Dynamic Programming	
		Recall	Precision	Recall	Precision
MIR DB	效　　能	84.99%	78.65%	83.26%	76.21%
	斷詞時間	123.7810 sec.		33.3440 sec.	
Sinica DB	效　　能	87.28%	82.67%	83.96%	81.66%
	斷詞時間	163.7350 sec.		39.8280 sec.	

表 4-2 加上定量複合詞構詞的斷詞系統效能實驗

比較表 4-1 與 4-2 的實驗，我們可以發現不論是在 MIR 或 Sinica 的詞庫中，抑或 Longest Word First 與 Dynamic Programming 兩種不同的斷詞演算法，其召回率皆有 3%的成長。其次，在斷詞的精確率方面，也提高了 6%的準確度，其中又以動態規劃演算法搭配 MIR 語音資料庫的使用進步最多，增加了 6.89%，顯示加入定量複合詞構詞規則，對斷詞系統的確有顯著地幫助。

4.4 疊詞的構詞單元

在 3.3.2 疊詞構詞的章節中，曾介紹了{AABB}、{ABAB}、{AAAA}或{AA}、{ABB}、{AAAA 地}或{AA 地}，以及{{C}又{C}}，共六型的疊詞構詞規則，於是我們在定量複合詞的構詞單元後，又加上疊詞構詞的實驗，結果如表 4-3。

斷詞方法 / 資料庫與效能		Longest Word First		Dynamic Programming	
		Recall	Precision	Recall	Precision
MIR DB	效　　能	84.99%	78.67%	83.25%	76.27%
	斷詞時間	128.8910 sec.		39.3280 sec.	
Sinica DB	效　　能	87.27%	82.67%	83.96%	81.71%
	斷詞時間	166.7180 sec.		45.2820 sec.	

表 4-3 加上定量複合詞、疊詞構詞的斷詞系統效能實驗

對照 4.3 定量複合詞的構詞單元實驗結果，我們發覺無論是 Longest Word First 抑或 Dynamic Programming 的斷詞方法，加入疊詞構詞單元後的召回率並無改變，細究原因如下：

（一）評估效能的測試資料為新聞資料庫

在新聞採訪、寫作的報導文學中，因事件陳述的獨特表現手法，一般均少見疊詞用語，因此在以新聞資料測試疊詞構詞的實驗中，並無法顯現其效能。

（二）詞庫中已含有許多常見的疊詞

除新聞測試資料庫的原因外，我們斷詞單元中所用到的 MIR 與 Sinica 語料庫原本亦含有許多疊詞，因此，大部分的疊詞在斷詞單元中都將被處理。

4.5 姓名的構詞單元

如 3.3.3 節所述，單靠姓名資料庫的收集整理，勢必無法有效應用於斷詞系統。因此，我們在斷詞系統中加入最後的姓名構詞單元，實驗如下。

斷詞方法 / 資料庫與效能		Longest Word First		Dynamic Programming	
		Recall	Precision	Recall	Precision
MIR DB	效　　能	85.97%	82.69%	84.13%	79.92%
	斷詞時間	134.2190 sec.		41.4850 sec.	
Sinica DB	效　　能	87.51%	84.76%	84.07%	83.34%
	斷詞時間	171.9060 sec.		46.7340 sec.	

表 4-4 加上定量複合詞、疊詞與姓名構詞的斷詞系統效能實驗

相較 4.4 節，未加入姓名構詞的實驗中，可以明顯發現姓名構詞使斷詞系統的整體精確率提升 3~4%，顯示簡易的姓名構詞於斷詞系統中，確有其效果。

4.6 語音合成用之斷詞系統整體效能分析

綜合以上幾個實驗，我們整理、分析中文斷詞系統效能如下：

（一）Dynamic Programming 的運算時間較 Longest Word First 快

由於動態規劃演算法獨特的演算方式，使我們可省略許多不必要的計算，因此就運算量來說，Dynamic Programming 的確可大幅減少計算時間。以 124,640 字的中研院平衡語料庫為主要斷詞詞庫，平均 1,000 字的新聞只需 1.1125 秒，相較長詞優先法的 4.0923 秒，似乎較適合 Real-time 的系統使用。

（二）Longest Word First 的斷詞表現較 Dynamic Programming 高些

由下面四個表的實驗整理，我們可以發現長詞優先法不論在 MIR 實驗室語音資料庫，抑或 Sinica 平衡語料庫，其斷詞表現均較好，其中在 Sinica Corpus 3.0 含有詞頻訊息的語料中，差距尚達 3.44%。

（三）疊詞構詞在本次測試資料中並無顯著成效

疊詞構詞在這次的新聞測試資料庫中並無顯著的效果，然而因加入疊詞構詞後，造成的搶詞情況並不嚴重，召回率最多僅只下滑 0.01%。因此考量系統的通用性，及詞庫收集、整理的不完整性，疊詞構詞仍屬必要。

斷詞方法 / 效　能	LWF Baseline	LWF 加入定量複合詞構詞	LWF 加入定量複合詞、疊詞構詞	LWF 加入定量複合詞、疊詞與姓名構詞
Recall	81.95%	84.99%	84.99%	85.97%
Precision	71.97%	78.65%	78.67%	82.69%
斷 詞 時 間	118.4370 sec.	123.7810 sec.	128.8910 sec.	134.2190 sec.

表 4-5 MIR 實驗室語音資料庫在 Longest Word First 斷詞的各階段表現

斷詞方法 / 效　能	LWF Baseline	LWF 加入定量複合詞構詞	LWF 加入定量複合詞、疊詞構詞	LWF 加入定量複合詞、疊詞與姓名構詞
Recall	84.49%	87.28%	87.27%	87.51%
Precision	76.21%	82.67%	82.67%	84.76%
斷 詞 時 間	154.0780 sec.	163.7350 sec.	166.7180 sec.	171.9060 sec.

表 4-6 Sinica 平衡語料庫在 Longest Word First 斷詞的各階段表現

斷詞方法 / 效　能	DP Baseline	DP 加入定量複合詞構詞	DP 加入定量複合詞、疊詞構詞	DP 加入定量複合詞、疊詞與姓名構詞
Recall	80.06%	83.26%	83.25%	84.13%
Precision	69.32%	76.21%	76.27%	79.92%
斷 詞 時 間	30.3910 sec.	33.3440 sec.	39.3280 sec.	41.4850 sec.

表 4-7 MIR 實驗室語音資料庫在 Dynamic Programming 斷詞的各階段表現

斷詞方法 / 效　能	DP Baseline	DP 加入定量複合詞構詞	DP 加入定量複合詞、疊詞構詞	DP 加入定量複合詞、疊詞與姓名構詞
Recall	81.10%	83.96%	83.96%	84.07%
Precision	74.81%	81.66%	81.71%	83.34%
斷 詞 時 間	36.4530 sec.	39.8280 sec.	45.2820 sec.	46.7340 sec.

表 4-8 Sinica 平衡語料庫在 Dynamic Programming 斷詞的各階段表現

4.7 實驗錯誤分析

在上述中文語音合成斷詞系統的諸多實驗中，我們發現幾個常見的錯誤，分析如下：

（一）搶詞

搶詞為中文斷詞系統常見的錯誤，如原本應為

原本　可　靠　打工　生活　的　學生

將因搶詞現象而成

原本　可靠　打工　生活　的　學生

（二）專有名詞辨識

「約定成俗」一直是語言的可愛之處，然而卻使斷詞系統難以應付，如新興「連宋馬立強」一詞分別代指連戰、宋楚瑜、馬英九、朱立倫與胡自強五人。又如近幾年才有的「公投」一詞，或屬不同領域的專有名詞「隨選視訊」、「串流」等，都無法有效且及時的收錄詞庫中。

（三）詞庫整理的困難

實驗過程中也發現，詞庫的整理不完全對斷詞系統的表現亦有影響。如「台北一０一」一詞可能誤植為「台北一〇一」，或「若望若望保祿二世」等，都是詞庫整理不全。

（四）對「詞」的認定不同

每人對「詞」認定的不一致，也將影響斷詞系統整體表現。如是「火力發電廠」或「火力　發電廠」，是「地方首長」還是「地方　首長」，為「核四公投」、「核四　公投」抑或「核　四　公投」，這些都是可能使斷詞系統表現不佳的原因。