基於反轉檔查找與最佳片段選取演算法的中文語音合成系統

林政源 謝明峰 陳冠廷 張智星

國立清華大學資訊工程學系

{gavins, pacific, marco, jang}@cs.nthu.edu.tw

摘要

本論文主要是解決以大量語料庫為基礎的語音合成的兩個問題，其一是搜尋比對大量語料庫非常費時，其二是從不同語句所取出的片段語音檔來加以接合，因為韻律參數的不一致，會使聽者明顯感覺不自然。因此，我們提出了反轉檔查找技巧來解決搜尋時間的問題，為求整體句子的自然韻律表現，我們提出了最佳片段選取演算法來達成這個目標，而對於PSOLA在調整音長表現可能不佳的情形，我們改以WSOLA方式實作。在搜尋比對時間與MOS評分的實驗中，我們均獲得到了不錯的成果。

1 系統簡介

近年來，隨著電腦科技不斷的蓬勃發展，中文文字轉語音 (TTS, Text-To-Speech) 的合成系統也慢慢朝向由單音節為主的合成單元架構轉變成以大量語料庫 (large corpus-based) 為主的合成單元架構。這方面的研究目前有 Heo-Jin Byeon 的 Event-Driven f0 Weighting[5], 大陸學者 Min Chu 等人的 Domain Adaptation[1]的方法, Ivan Bulyko提出的BMM models[6]以及台大周福強博士的decision trees方法[10]等。

一般而言， 採用大量語料庫的系統，其合成品質較單音節為主的系統來的好。因為它的方法是直接從語料庫擷取所需要的片段進行接合，所以在韻律表現上會較自然，也因為如此，在聲音方面所需調整的地方就會不太多，這也避免了聲音經過調整後而造成音質破壞的疑慮。然而，採用大量語料庫的做法也有其缺點，以下列出二個常見的問題:

1. 輸入文句需要和大量語料庫作比對：

   文句經過斷詞以後，再去語料庫找尋可能的詞句片段，並取出後加以接合，然而若演算法設計不當則會讓比對時間相對費時，所以發展一個有效率的演算法來縮短比對時間對系統的效能是非常重要的。
2. 詞句片段之間的韻律參數差異性問題：

   從不同語句所取出的片段語音檔來加以接合，因為韻律參數的不協調，會使聽者明顯感覺不自然。

有鑑於兩種缺點的考量，本論文採用反轉檔查找技巧來降低比對時間，而以動態規劃演算法來尋找最佳的接合片段使其合成自然度提升。這兩種方法將在第三以及第四節中論述。

2 系統架構說明

本論文所建立的中文語音合成系統架構將如下圖表示：

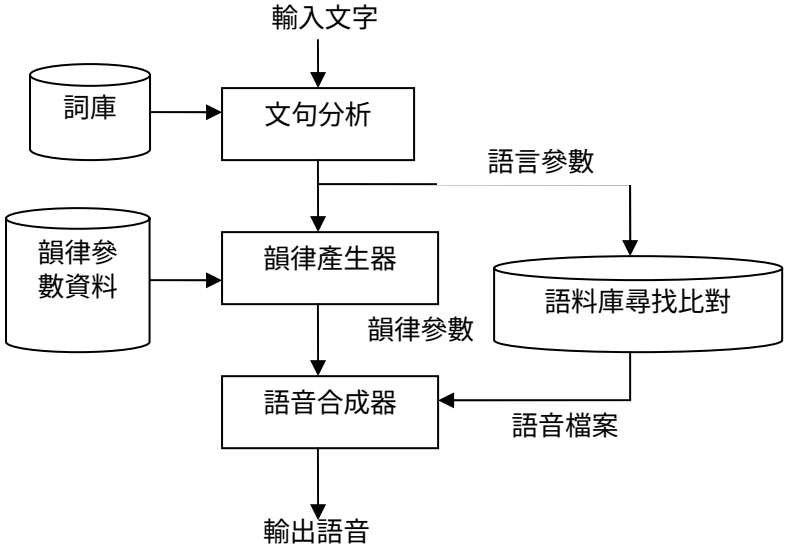

圖1. 中文文字轉語音系統流程圖

此系統主要分為四大類：

1. 文句分析：將所輸入的文字加以分析，得到音節以及詞的語言參數。
2. 韻律產生器：將語言參數轉換成語音合成所需要的韻律參數，而韻律產生器所需要的參數資料，是以類神經網路來獲得。
3. 語音合成器：根據韻律參數，將語料庫中所得到的語音檔案加以調整。
4. 語料庫搜尋比對：這是本論文最重要的一環，主要是將分句分析的結果和語料庫作比較查詢，並找出最適當的語音檔案當作輸出。

2.1 文句分析

當文句輸入時，第一步驟就是針對此文句做分析，以得到其語言參數，如此才可進一步的得到韻律參數，合成出所需要的語音。而所謂語言參數，又可以分為音節的語言參數和詞層的語言參數。文句分析的系統如下圖所表示。

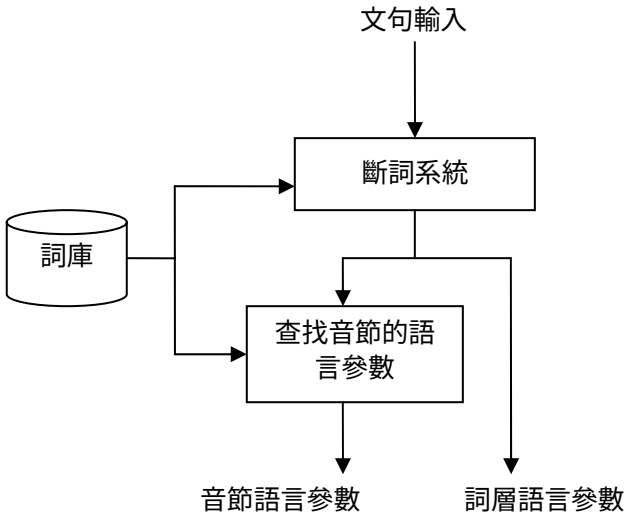

圖2．文句分析系統流程圖

在文句分析中，斷詞的處理為最重要的部分。而本篇論文的斷詞方法是根據一個大詞庫 (中研院漢語平衡語料庫，Sinica Corpus 3.0 [18]，共有130,757個詞。)，然後輸入語句再比對此大詞庫來進行查找。在斷詞的研究上，也有相當多的方法[12][11][16]，我們為了系統的效率則採用長詞優先法，再以各種構詞的方法補足其詞庫不足的缺點。

2.2 韻律產生器

語音合成系統的關鍵技術就在於韻律的變化是否平順自然。而韻律的變化包括音調高低起伏、音量的大小變化、每個音節的長短及停頓這三個部分。而韻律產生器大致上有幾種方法：規則法、統計法、類神經網路法[8][17]。目前大多數的實驗結果以類神經網路為較佳，故本論文採用其方式來製作韻律產生器。在類神經訓練的實作方面，輸入是音節和詞層的語言參數，輸出是韻律參數。而音節語言參數包括本音節的聲母、韻母、音調，下個音節的聲母、音調等。詞層語言參數包括本詞詞長、下一個詞的詞長和本音節在本詞的位置、本詞和下詞之間的標點符號。輸出的韻律參數為音節間的停頓時間、聲母長度、韻母長度、音節的韻母平均能量、基頻軌跡。其中基頻軌跡參數是以正交化展開的前四階係數[13]表示。

而基本的單層類神經網路函式運算時，輸入參數有 m 個輸入參數 $i_{1\sim m}$ 和 n 個輸出參數 $o_{1\sim n}$，而輸出參數和輸入參數的關係為：

$$o_j = f(\sum_{k=1}^{n} i_k \cdot w_{k,j} + b_j)$$

$w_{k,j}$ 為第 k 個輸入神經元到第 j 個輸出神經元的加權值，b_j 則是偏差值(bias)，其關係圖如下圖：

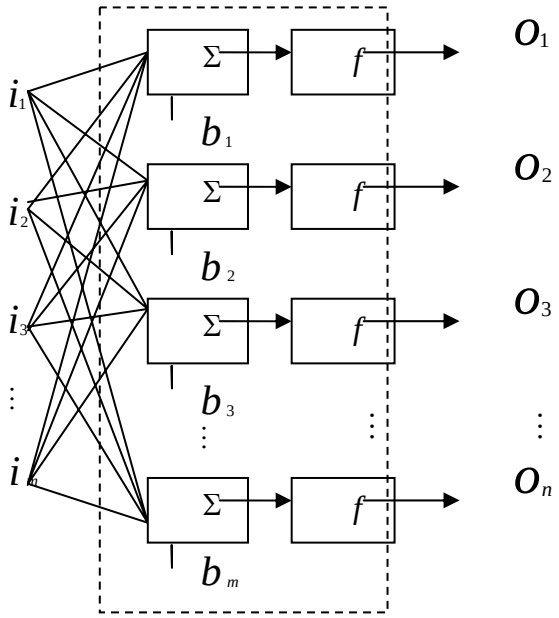

圖3 單層類神經網路結構圖

而 $f(x)$ 可以自行定義，本論文使用最簡單的線性轉換函數(Linear Transfer Function)，即：

$$f(x) = x$$

2.3 語音合成器

語音韻律參數主要包括音調高低軌跡、語音長度、音量大小三部分。而在調整音調方面，常用的方法有大致上分為兩種：Sinusoidal Modeling [4]與PSOLA (Pitch Synchronous Overlap and Add) [2]，雖然Sinusoidal Modeling方法的合成音質與PSOLA 相當，但是基於執行效率的考量以及音高調整幅度通常不大，我們採用後者(PSOLA)作為基本校正音調的方法。而調整音長的方法，本論文所採用的是WSOLA (Overlap-add Technique Based on Waveform Similarity)方法[9]。採用此方法的理由是：我們的語音合成單元大部分來自大量語料庫，所以無法對每一個語音檔作基週標位的修正，然而使用PSOLA方法調整音長時，基週標位須十分準確，否則音長調整後的音質會有雜音的現象，但是使用WSOLA方式調整時，並不需要基週標位的資訊，WSOLA是用AMDF (Average Magnitude Difference Function)[3] 進行音框比對，藉以找出最適合的音框作波形疊加。所以WSOLA單就調整音長而言，其效果較PSOLA為佳。

2.4 語料庫搜尋比對

大量語料庫的語音合成系統必須能在大量的資料中，找到需要的片段加以接合。而在實作上，會遇到以下的問題：

1. 需要有較大的儲存空間：

   目前大部分的Embedded的系統較不適合採用此方法實做TTS系統，所以目前以大量語料庫為基礎的TTS都是在PC的硬體上執行居多。由於硬碟容量日漸增大、語音壓縮的技術也不斷地改良，所以用PC實作的TTS系統，也會更加的普及與實用。
2. 搜尋語料庫中所需的片段：

   如何從大量的語料庫中，快速地去找到最適合的片段來接合，是這方面設計最需要解決的問題。我們將採用反轉檔的查找技巧來克服這個問題，即使語料庫的大小增為兩倍，我們的搜尋時間也不會線性成長兩倍，甚至兩者的時間差異極小。然而，除了搜尋時間的問題之外，我們希望找到的片段長度越長越好，這樣韻律的表現最為自然，所以我們提出建立最長詞數表的方法並搭配之前的反轉檔可以很快的找到所需要的最佳片段。這些方法將在第三節中加以闡述。
3. 片段與片段之間差異過大：

   從大量語料庫中所取出來的片段，會受到前後音、句子節奏韻律和個人情緒的影響，造成片段與片段之間韻律參數有極大的差異，會讓聽者明顯感受出是由不同片段組合而成的語音。例如，片段的平均音高或者音量差異過大、片段間的接合不連續等。在本論文中，我們提供了另一個基於動態規劃演算法為基礎的最佳片段選取法來解決這方面的問題，這將在第四節會加以說明。

3 反轉檔與最長連續詞數表

3.1 反轉檔查找

反轉檔的目的是在改變原本語料庫的資料結構以減少搜尋時間。若原來的資料是經常變動的，就不適合採用反轉檔 (因為每次的變動都要再重建反轉檔)。而在本系統中，大量語料庫的資料是固定的，因此可以使用反轉檔的技巧來進行查找的動作。首先我們必須先將文句編號，再進行斷詞的動作。下面是語料庫中有的文句：

表1　語料庫中的句子

語料庫中的句子	1.母親 真 偉大 2.我 母親 是 老師 3.當 老師 是 他 一生 的 夢想 4. ….

根據以上的句子，我們可以建立以下的反轉檔：

表2 反轉檔範例

母親 老師 夢想 是 ⋮	<1>,<2> <2>,<3> <3> <2>,<3> ⋮

然而，反轉檔只存入出現的句子編號，並不能讓我們快速的找到本詞和下一個詞的關係，因此在反轉檔中我們要存入這個詞在句子中，所連接的下一個詞。因此反轉檔變成以下格式：

表3 反轉檔範例二

母親 老師 夢想 是 ⋮	<1,真>,<2,是> <2, Φ >,<3,是> <3, Φ > <2,老師>,<3,他> ⋮

不過，由於語料庫的資料量龐大，一個詞往往會重覆出現多次，所以必須再加入一個數值，就是下一個詞出現在該詞反轉檔的哪一個位置，而範例反轉檔如下：

表4 反轉檔範例三

母親 老師 夢想 是 ⋮	<1,真,1>,<2,是,1> <2, Φ ,0>,<3,是,2> <3, Φ ,0> <2,老師,1>,<3,他,1> ⋮

上例的第一列第二行：<2,是,1> 表示 『母親』這個詞是在出現在大量語料庫中的第2句，它的下一個詞是『是』，而『是』這個詞是出現在它反轉檔的第 1 位置（『是』在大量語料庫中可能有好幾個），所以我們根據上例來看，『是』的第 1 個位置所擺放的是 <2,老師,1>，如此我們又可以繼續追蹤下去。建立這個反轉檔的資料結構之後，就可以很快的找到在語料庫中，每個詞的下一個詞的反轉檔位置，系統也就可以快速的找到輸入文句中任何一個詞開始，接下來連續最長的文句了。

3.2 建立最長連續詞數表

當語音合成系統輸入文句時，需要立刻從大量語料庫中找到相同的文字片段，並加以取出。而找到片段的原則是每個片段的字數越多越好。在同個片段中，是由人在同一時間所錄下的連續語音，所以一定是最自然的，在語音合成的觀點上，當然是越自然越好。因此在搜尋時，以找到最長的片段為優先。然而，當語料庫十分龐大時，在尋找比對的所花的時間甚巨。又因為希望能有單一最長的片段，無法使用由左往右找出最長連續片段的方式來進行。而當找到最長片段並取出時，還要從剩下的文句繼續比對語料庫，再找出次長的片段，如此反覆進行，計算量將會非常大。在這種情況之下，我們可以先建立以下表格：若輸入的文句可以被斷詞系統斷出 N 個詞，而我們要找到從每個詞 S_n 開始，和資料庫中最長的連續詞數。其中 $1 \le n \le N$ 。例如欲輸入下列文句：

母親 明年 將 離開 台南 前往 嘉義

而在語料庫中有下列相關語句，括號部分是與輸入文句相同的部分：

表5 與輸入文句有關的語料庫範例

語料庫中和輸入文句相關的句子	1. (母親) 真 偉大 2. 他 (明年 將 離開) 台北 3. (離開 台南) 後 的 生活

	4. (前往 嘉義) 的 路 很 遙遠

可以得到以下的表格：

表6 每個詞開始最長連續詞數表格

輸入文句的斷詞	母親 S_1	明年 S_2	將 S_3	離開 S_4	台南 S_5	前往 S_6	嘉義 S_7
從本詞起最長的連續詞數	1	3	2	2	1	2	1

根據此一表格，就可以找到最長連續片段。再將最長連續片段從輸入文句中取出來，而被取出來的詞在其表格中的數字補上0，以上表為例，會得到以下表格：

表7 取出片段後的連續詞數表格

輸入文句的斷詞	母親 S_1	明年 S_2	將 S_3	離開 S_4	台南 S_5	前往 S_6	嘉義 S_7
從本詞起最長的連續詞數	1	0	0	0	1	2	1

而表格內還未成為0的詞，就是剩下來仍需從大量語料庫取出的句子。在所有的數字還未變成0之前，仍需要重覆取出數字最大的部分做處理。本例到最後會斷成以下句子:

（母親）（明年 將 離開）（台南）（前往 嘉義）

4 最佳片段選取演算法

在大量語料庫中尋找所需的片段之後，我們並非直接拿來作語音檔的接合，因為這樣的接合會造成不自然的韻律，這裡可能的問題大致上有兩種：

1. 最佳片段選取問題：
   因為片段是從句子中所取出來的，所以同一種詞在不同句子中所表現的韻律就會有所不同，例如聲音的音量、音高或音長等。所以我們必須在這個詞所有出現的句子中，找到最適合的片段來合成。
2. 片段與片段之間的接合問題：
   即使找到最適合的片段來合成，但還是存在前後片段的接合不協調的問題。所以我們也必須在每一句的所有片段，找到它們最佳的組合方式來克服這個問題。

事實上，關於以上的兩個問題，可以同時以我們所提出的動態規劃演算法來解決。以下就是我們針對每一個句子的最佳選取片段演算法:

1. 我們制定狀態機率 (State Probability) 來定義 較佳的可能片段，通常保留前三名可能的片段，即每一個詞皆有三個候選片段。
2. 我們制定狀態轉移機率 (Transition Probability) 來定義片段之間可能組合的選擇。
3. 最後，根據前兩者累積的機率值，由最大機率值的片段回溯找出最佳可能的組合路徑。

4.1 狀態機率

首先，關於第一項算出狀態機率，我們考量到即使選出最佳的片段後，仍需要參考韻律參數而加以改變才作接合的處理，所以我們定義其機率的計算應該要根據音高和音長的差異來制定，也就是希望需要調整的韻律不要與原先的韻律差別太大，使得經由改變音高音長而破壞音質的情況降低。所以我們制定了以下的公式:

$$dist_i(j) = \sum_{j=1}^{M}\sum_{k=1}^{N} \frac{\left|Pitch(j,k) - TrainPitch(k)\right|}{Pitch(j,k)} + \frac{\left|Duration(j,k) - TrainDuration(k)\right|}{Duration(j,k)}$$

有 M 個候選詞，一個詞有 N 個音節，所以 $1 \le j \le M$ ， $1 \le k \le N$ ， $Pitch(j,k)$ 為第 j 個候選詞的第 k 個音節的平均基頻軌跡，單位為 Hz ， $TrainPitch(k)$ 是第 k 個音節經過韻律產生器所產生出來的平均基頻軌跡。而 $Duration(j,k)$ 為第 j 個候選詞的第 k 個音節的音長， $TrainDuration(k)$ 是經過韻律產生器的音長， $Duration$ 為聲母和韻母音長的加總，單位為秒。而結果 $dist_i(j)$ 表示句子之中第 i 個詞語的第 j 個候選詞的距離值。

計算出每個候選詞的dist值之後，只保留前3個最小距離的候選詞，而狀態機率就以其距離的倒數成正比，它的定義如下:

$$Sprob_i(j) = \left(\frac{\frac{1}{dist_i(j)}}{\sum_{k=1}^{3} \frac{1}{dist_i(k)}} \right)$$

上例公式說明：第i個詞語的第j個候選詞片段它的狀態機率值公式。

4.2 狀態轉移機率

至於狀態轉移機率，我們考慮的因素並不再以候選片段的音高或音長當作特徵了，因為前後片段的音高或音長本來就會不一樣。我們這邊所考量的方向是希望所選出來的候選片段組合，有最通順的接合品質，不會讓聽者感到整句話是由幾個詞分開唸出來的效果。而能達成最好的接合品質，就是去觀察每個候選片段的前一個音或者它的後一個音，然後跟其他候選片段來作比較。

舉例來說，假設某一個句子有個5個詞片段：〔今天〕〔去〕〔台北〕〔買〕〔衣服〕，句子中的每個片段皆有3個候選詞。如何決定〔去〕這個片段的第一個候選詞與〔今天〕這個片段的哪一個候選詞有最佳的接合效果，我們所採用的是相近音查表法來決定。例如〔去〕這個片段的第一個候選詞，它在原句子中其前面所接的字是〔點〕（例如原句是：她點去了這個痣），此時我們便可利用此資訊去查詢相近音表的〔點〕與〔天〕的相似程度，這裡的相似程度是以音節中的韻母來比較，而上例的韻母則是‘ㄧㄢ’。

另外，我們也需要判斷這三個〔今天〕的候選片段它們後面所接的音，然後與〔去〕作相似音的比對。例如，共有3個候選詞〔今天〕，第一個〔今天〕後面所接的字為〔我〕，第二個〔今天〕後面所接的字為〔天〕，而第三個〔今天〕後面所接的字為〔氣〕，如此我們將會給予第三個候選詞〔今天〕有較高的狀態移轉機率。而這裡的相似程度是以音節中的聲母來比較，而上例有較高的狀態移轉機率的聲母則是‘ㄑ’（因為〔去〕的聲母也是‘ㄑ’。）

至於如何決定相似程度的高低，則可採用兩種方式來計算，一種就是直接採用rule based的國語聲韻母分類表[15]，另一種則是利用語音辨識的技巧去統計那些聲韻母的發音最為相近，由於rule based所定義的分類表較難以具體描述其相似度的高低，所以我們採用後者的方式去統計聲韻母的發音，建立了一個相近發音查詢表，包含兩種統計模式，一是聲母相近發音的統計，另一個則是韻母相近發音的統計。所以狀態移轉機率的公式定義如下:

$$Tprob_i(j_1, j_2) = \left(\frac{similarTable(j_1\_nextword,\ j_2\_prevword)}{\sum_{k=1}^{3} similarTable(j_1\_nextword,\ k\_prevword)} \right)$$

上例公式說明：第 i 個詞語中的第 j_1 個候選片段到第 $i+1$ 個詞語中的第 j_2 個候選片段的狀態轉移機率值，而similar Table表示相近發音查詢表，可查詢聲韻母之間的距離。這裡的聲母距離指的是第 j_1 個候選片段的下一個字的聲母和第 j_2 個候選片段首字的聲母作比較，而韻母距離則是指第 j_2 個候選片段的上一個字的韻母和第 j_1 個候選片段尾字的韻母作比較。

4.3 累積機率以及回溯最佳路徑

累積機率的方式本質上是採用乘法，但是電腦的精確度是有限位數，所以我們將累積機率的方式改為加法，因此最後機率值會再取ln。所以第 i 個詞語的第 j 個候選片段的累積機率值其定義為 $P(i,j)$ ：

$$P(i,j)=\max_{k}\left(P(i-1,k)+\ln(Tprob_i(k,j))\right)+\ln(Sprob_i(j))$$

初始值 $P(1,j)=\ln(Sprob_1(j)),\quad for\quad j=1\ \ to\ \ n$ (在本論文,n = 3)

$B(i,j)$ 紀錄了第 i 個詞語中的第 j 個候選片段，它的前一個詞語與它有最佳的接合效果的候選片段位置：

$$B(i,j)=\arg\max_{k}\left(P(i-1,k)+\ln(Tprob_i(k,j))\right)$$

所以經由最高累積機率的 $P(lastword,j)$ 值回溯到 $P(1,j)$，參考相對應的 $B(i,j)$ 紀錄，即可以找出最佳的接合片段組合。

5 實驗結果

本論文所採用的大量語料庫為台北科技大學黃紹華教授[19]所提供的語料庫，共655個語音檔（句子），總共包含35085音節，錄音時間為9300秒，取樣頻率為20 KHz，16位元編碼。在採用這些語料檔案之前，我們有人工修正過的每個音節的子音和母音的位置標示，以配合後面的斷詞分析與韻律訓練。

首先，在類神經訓練方面，我們所使用的演算法為倒傳遞演算法(Back Propagation Algorithm)，使用訓練語料中的455個語音檔，測試語料為200個語音檔，所得到的實驗數據如表8所示。而計算誤差的公式為均方根誤差(Root Mean Square Error, RMSE)，算式如下：

$$RMSE=\sqrt{\frac{\sum_{t=1}^{N}(T(t)-S(t))^2}{N}}$$

N 為總共的個數，$1\le t\le N$ ，$T(t)$ 為訓練出來的結果，而 $S(t)$ 為正確的結果。

表 8 類神經訓練韻律參數的內外部測試結果

	整體語料庫資料	內部測試	外部測試
基頻軌跡平均	平均145.87H,標準差23.13Hz	RMSE 19.1Hz	RMSE 20.1Hz
聲母長度	平均56.3ms,標準差44.5ms	RMSE 16.5ms	RMSE 17.6ms
韻母長度	平均141.7ms,標準差52.2ms	RMSE 33.7ms	RMSE 38.2ms
停頓長度	平均16.8ms,標準差50.2ms	RMSE 38.4ms	RMSE 38.7ms
聲音能量	平均65.18dB,標準差6.23dB	RMSE 4.46dB	RMSE 5.04dB

我們從測試語料挑選某一個語句，並觀察其韻律參數的結果：

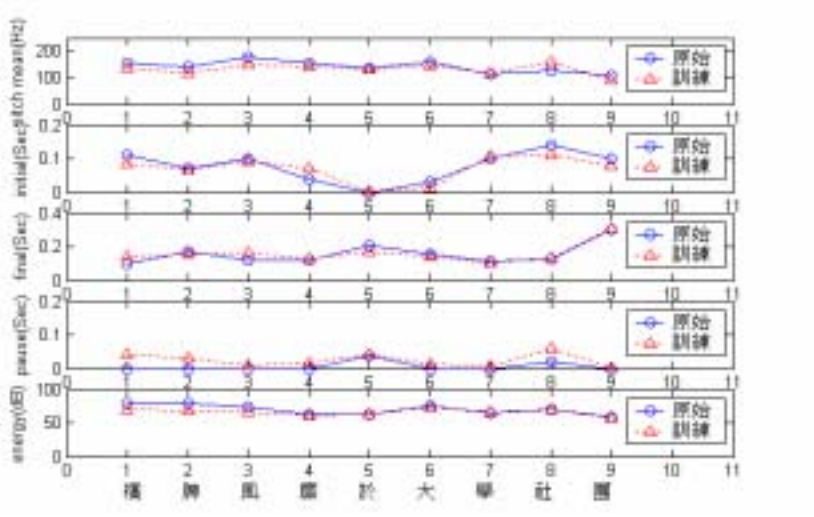

圖4 外部測試所得的韻律參數和原本韻律參數比較圖

實驗結果顯示，此類神經訓練法所得的韻律參數確實可以給予後面的語音合成器採用。此外，為了證實我們採用的反轉檔查找法能有效的找尋所需要的候選片段，我們以互相關比對法(Cross-correlation)[14] 來作為比較。測試語料為75任意句子，2005個音節，共1291個詞，平均每句26.7個音節、17.2個詞，執行時間如下：

表 9 演算法時間比較表

使用演算法	執行時間
互相關比對法	5955.47秒
反轉檔	65.96秒
互相關比較法＋最長連續詞數表	173.53秒
反轉檔＋最長連續詞數表	3.82秒

由此可知，使用反轉檔和最長連續詞數表，可以加速找到輸入文句和大量語料庫中對應的片段，和最慢的單純使用互相關比較法快了1500倍之多。

除了加速找尋所需的片段後，我們對每個詞保留前三名候選片段，然後再使用動態規劃演算法配合相近音查表，便可求出最佳的候選片段組合。為了證實此方法的確可行，我們採用MOS (Mean Opinion Score) [7]的評分方式，針對我們所提的方法以及隨意取出任一候選片段來實驗。另外，在大量語料庫的前提下，為了證實WSOLA調整音長方面的能力會優於PSOLA，我們在實驗過程中也加入這方面的比較。對於合成語句所花的時間，我們也作了統計，實驗平台為Pentium IV 1.6 GHz，執行環境為WINDOWS XP + MATLAB。測試語料為任意20句，參與合成語句的聽力測驗總共為10人，以下為其實驗的結果：

表 10 合成語句的MOS值與其計算時間統計

使用演算法	平均MOS	平均花費時間
任意取出候選片段組合＋PSOLA	2.6	6.3 秒
任意取出候選片段組合＋WSOLA	2.8	8.1 秒
最佳片段選取演算法＋PSOLA	3.3	7.8 秒
最佳片段選取演算法＋WSOLA	3.5	9.7秒

WSOLA在音長調整方面的合成品質確實勝過PSOLA，這是因為PSOLA音質的好壞取決於基週標位(Pitch Mark)的正確性，而大量語料的資料量通常很大，在實作上較難掌握每個音節的正確基週標位，而WSOLA並不需要基週標位的資訊。但是，若以系統效率而言，WSOLA所花費的時間則會較久，較不適合real time的系統設計。而最佳片段選取演算法的確大大的改善了原先使用隨意片段選取方法的音質。

6 實驗結果

在本論文中，我們已經實作了一個完整的TTS系統，此系統是以大量語料庫為基礎，並且配合我們所提出的反轉檔查找法與最佳片段選取演算法，使得此系統提升了在大量語料庫的搜尋速度之外，也保有較貼近自然的人聲，另外，也實驗證實了WSOLA對此系統的在合成音質方面的貢獻。

7 References

[1] Chu Min, Li Chun, Peng Hu, Chang Eric, "DOMAIN ADAPTATION FOR TTS SYSTEMS", *ICASSP* 2002

[2] F. Charpentier and Moulines, "Pitch-synchronous Waveform Processing Technique for Text-to-Speech Synthesis Using Diphones," European Conf. On Speech Communication and Technology, pp.13-19, Paris, 1989

[3] G.S. Ying and L.H. Jamieson and C.D. Michell, "A probabilistic approach to AMDF pitch detection", Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on Volume: 2 , 1996 , Page(s): 1201-1204 vol.2

[4] George E.B, Smith M.J.T., "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model", IEEE Transactions on Speech and Audio Processing, 1997

[5] Heo-Jin Byeon, Yung-Hwan Oh, "An event-driven f0 weighting for prosody control in a large corpus-based TTS system", Signal Processing Letters, IEEE 2004.

[6] I. Bulyko, M. Ostendorf and J. Bilmes. "Robust Splicing Costs and Efficient Search with BMM Models for Concatenative Speech Synthesis", in *Proceedings of ICASSP*, 1:461-464, 2002.

[7] ITU-T, Methods for Subjective Determination of Transmission Quality, 1996, Int. Telecommunication Unit.

[8] S. Haykin,"Neural Networks – A Comprehensive Foundation," Macmillan College Publishing Company, 1994

[9] Werner Verhelst and Mark Roelands"An Overlap-Add Technique Based on Waveform Similarity For High Quality Time-Scale Modification of Speech" In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 554--557, Minneapolis, USA, apr #"27--30 1993

[10] 周福強, "以語料庫為基礎之新一代中文文句翻語音合成技術", 國立臺灣大學電機工程學研究所博士論文, 1998.

[11] 唐大任,"中文斷詞器之研究", 國立交通大學電信工程系碩士論文, 2001.

[12] 朱怡霖, "中文斷詞與專有名詞辨識之研究", 國立臺灣大學資訊工程學研究所碩士論文, 2001.

[13] 王逸如,"對基週軌跡做向量量化之線性預估語音編碼 ", 國立交通大學電信研究所碩士論文, 1886.

[14] 謝明峰, "使用大量語料庫的中文語音合成系統實作", 國立清華大學資訊工程所碩士論文, 2004.

[15] 郭智超,"以音節為基礎之中文語音文件檢索系統的研究", 國立清華大學資訊應用所碩士論文, 2003.

[16] 鍾綸,"用於語音合成的中文斷詞分析 ", 國立清華大學資訊應用所碩士論文, 2004.

[17] 黃紹華, "中文文句翻語音系統中韻律訊息產生器之研究", 國立交通大學電子研究所博士論文,1995.

[18] http://rocling.iis.sinica.edu.tw/ROCLING/corpus98/wordlist.htm

[19] http://214lab.ee.ntut.edu.tw/