

ỨNG DỤNG KỸ THUẬT CẢN CHỈNH THỜI GIAN ĐỘNG VÀO NHẬN DẠNG TỪ TIẾNG VIỆT ĐƠN LẺ

APPLIED DYNAMIC TIME WARPING TECHNIQUE TO RECOGNIZE VIETNAMESE ISOLATED WORDS

Bạch Hưng Nguyễn, Lương Chi Mai

Department of Pattern Recognition & Knowledge Engineering, IOIT

nguyenbh@netnam.org.vn

lcmai@ioit.ncst.ac.vn

Tóm tắt: Bài báo này giới thiệu kỹ thuật Cản chỉnh thời gian động (DTW) áp dụng nhận dạng các từ đơn lẻ tiếng Việt. Hệ thống sử dụng Linear Prediction Coefficients (LPC) để trích chọn đặc trưng. DTW được dùng để đo khoảng cách giữa mẫu cần nhận dạng và mẫu đã học. Luật Nearest Neighbor (NN) được sử dụng để đưa ra quyết định nhận dạng. Chúng tôi đã tiến hành thử nghiệm và đánh giá hiệu quả của kỹ thuật này trên các mẫu tiếng nói rời rạc của các chữ số từ 0 đến 9. Dựa trên các kết quả thí nghiệm, độ chính xác đạt được là 89%. Chúng tôi đề xuất một số hướng nâng cao độ chính xác nhận dạng trong bài toán nhận dạng từ tiếng Việt đơn lẻ và áp dụng kết quả nghiên cứu vào thực tế.

Abstract: This paper demonstrates Dynamic Time Warping technique (DTW) applied in the Vietnamese language. Isolated digits 0-9 and their concatenation patterns are used for data set. Linear Prediction Coding Coefficients (LPCCs) are extracted and formed feature vectors represented each speech signal. DTW is used to measure distances between referenced and evaluated vectors. These distances, indicating nearest of unknown vectors to references, incorporated with Nearest Neighbor (NN) decision technique are used to decide what number was spoken. Based on 1200 experiment results, our accuracy rate is 89%. We propose some directions to uplift the accuracy of Vietnamese isolated words recognition problem, and apply those results to the reality.

I. GIỚI THIỆU

Do sự khác nhau về ngữ điệu, về cách phát âm của từng người nói nên dạng sóng của một từ trong các lần nói khác nhau có thể có rất nhiều sự khác biệt. Thậm chí khi một từ được nói bởi một người thì sự khác nhau vẫn tồn tại bởi tốc độ nói, nói to/nhỏ, tâm lí khi nói. Một số mẫu có thể có thời gian nói dài hơn và nói to hơn, một số khác có thể nói nhỏ hơn và ngắn hơn. Bài toán nhận dạng từ đơn lẻ ở đây chính là bài toán so sánh hai tín hiệu và khó khăn ở chỗ độ dài các tín hiệu không giống nhau.

Mẫu tín hiệu tiếng sau khi thu vào máy tính là một dãy tín hiệu số. Chúng sẽ được chia thành các frame liên tục gối lên nhau. Giả sử mỗi frame được chọn với độ dài là $t_f=512$ mẫu tín hiệu và phân gối lên nhau giữa hai frame liên tiếp là $t_0=100$ mẫu tín hiệu. Mỗi frame này qua phép phân tích LPC ở pha trích chọn đặc trưng sẽ được biểu diễn bằng một véc tơ p chiều và ta gọi véc tơ này là véc tơ đặc trưng. Ý tưởng cơ bản của phép phân tích LPC được đề cập trong [Lawrence93]. Mục tiêu của phương pháp phân tích LPC là tìm các giá trị hệ số dự báo để cực tiểu hóa bình phương giá trị lỗi.

Giá trị p là một giá trị không cố định thường được lựa chọn theo kinh nghiệm và mục đích của việc lựa chọn p là để cố gắng giảm thiểu các bước tính toán trong phép phân tích LPC.

Điều đó có nghĩa là từ việc lưu giữ một frame với t_f mẫu ta chỉ cần lưu một véc tơ với p chiều. Như vậy một mẫu nhận dạng sẽ là một chuỗi các véc tơ đặc trưng. Chuỗi véc tơ đặc trưng này sẽ được dùng để so khớp với chuỗi véc tơ đặc trưng chuẩn đã được lưu trong chương trình để tìm khoảng cách, khoảng cách này là cơ sở để chúng ta quyết định kết quả nhận dạng.

Gọi $a = \{a_1, a_2, \dots, a_I\}$ và $r = \{r_1, r_2, \dots, r_J\}$ lần lượt là chuỗi véc tơ đặc trưng của tín hiệu 1 và tín hiệu 2. Ở đây r_j và a_i là các véc tơ đặc trưng (hay còn gọi là các frame) p chiều với $i=1, \dots, I, j=1, \dots, J$. Vì độ dài của hai chuỗi véc tơ a và r khác nhau nên $I \neq J$. Sự chuẩn hóa về thời gian giữa a và r là sự mở rộng một cách tuyến tính mẫu tín hiệu ngắn hơn đạt tới độ dài của mẫu tín hiệu dài hơn. Giả sử $J < I$. Kí hiệu $D(r, a)$ là khoảng cách giữa hai mẫu tín hiệu r và a và được tính như sau:

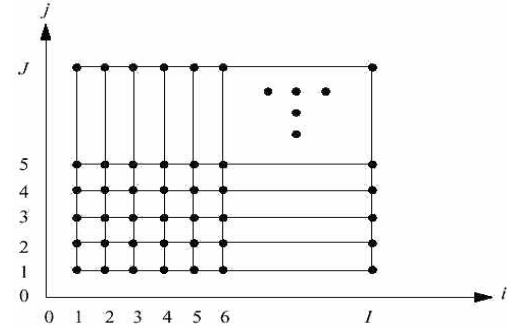
$$D(r, a) = \sum_{i=1}^I d_f(r_{j^*}, a_i) \quad ; \quad j^* = \left\lceil \frac{J}{I} * i \right\rceil \quad (1)$$

với $d_f(r_{j^*}, a_i)$ là khoảng cách giữa hai frame r_{j^*} và a_i .

Tuy vậy nhược điểm của phương pháp này là cho rằng tốc độ phát âm phụ thuộc hoàn toàn vào khoảng thời gian tồn tại của từ và độc lập với âm thanh đang được nói.

II. DYNAMIC TIME WARPING

Sự so khớp phi tuyến giữa các mẫu a và r như đã nói ở phần trước được biểu diễn trên lưới điểm hình 1. Trên lưới này tín hiệu a nằm dọc theo trục i , tín hiệu r nằm dọc theo trục j . Mỗi giao điểm trên lưới được định nghĩa như là một nút, nút (i, j) có nghĩa là so khớp frame i của tín hiệu a với frame j của tín hiệu r . Nút $(0, 0)$ được gọi là nút gốc nơi mà tất cả các đường đều xuất phát từ đây. $d_f(i, j)$ là khoảng cách giữa véc tơ đặc trưng a_i với r_j , khoảng cách này còn được gọi là *khoảng cách cục bộ*. Ta có thể



Hình 1: Lưới minh họa quy hoạch động

viết đầy đủ $d_f(i, j)$ là $d_f(a_i, r_j)$ với các véc tơ a_i và r_j nhận được qua phép phân tích LPC có dạng sau: $a_i = [1, a_{i2}, a_{i3}, \dots, a_{ip}]^T$ và $r_j = [1, r_{j2}, r_{j3}, \dots, r_{jp}]^T$

Trong báo cáo này chúng tôi áp dụng metric tổng quát Minkowski trong trường hợp $s=1$ và $s=2$, tương ứng với City block metric và Euclidian metric

$$\text{City block metric: } d(a_i, r_j) = \sum_{k=1}^p |a_{i_k} - r_{j_k}| \quad (2)$$

$$\text{Euclidian metric: } d_2(a_i, r_j) = \sqrt{\sum_{k=1}^p |a_{i_k} - r_{j_k}|^2} \quad (3)$$

$$\text{Ta định nghĩa giá tại nút } (0, 0) \text{ là: } d_f(0, 0) = 0 \quad (4)$$

Một đường đi ở đây được định nghĩa bởi các cặp nút $(i_{k-1}, j_{k-1}) \rightarrow (i_k, j_k)$. Một đường đi xuất phát từ nút $(0, 0)$ đến nút (i_k, j_k) có một giá toàn cục là giá tích lũy từ điểm bắt đầu của đường đi cho đến khi gặp nút (i_k, j_k) . Ta gọi giá này là $D(i_k, j_k)$ và được định nghĩa:

$$D(i_k, j_k) = D(i_{k-1}, j_{k-1}) + d_f(i_k, j_k) \quad (5)$$

Vì nút (0,0) là nút khởi đầu của tất cả các đường đi nên ta có: $D(0,0) = 0$ (6)

Ở biểu thức truy hồi (5) khi cho k chạy về tới k=1 ta có

$$D(i_k, j_k) = \sum_{m=0}^k d_f(i_m, j_m) \quad (7)$$

Rõ ràng là trên lưới quy hoạch động có nhiều đường đi xuất phát từ nút (0,0) đến nút (i_k, j_k) với các giá toàn cục khác nhau. Như vậy bài toán có thể được hiểu như là việc tìm kiếm một đường đi đầy đủ từ nút (0,0) tới nút đích (I,J) sao cho *giá toàn cục* là nhỏ nhất. Kí hiệu $D^*(i_k, j_k)$ là giá toàn cục nhỏ nhất. Khi đó từ (5) và (7) ta có:

$$\begin{aligned} D^*(i_k, j_k) &= \min \{D(i_{k-1}, j_{k-1})\} + d_f(i_k, j_k) \\ &= \min \left\{ \sum_{m=0}^k d_f(i_m, j_m) \right\} \end{aligned} \quad (8)$$

trong biểu thức này $(i_k, j_k) = (I, J)$, min lấy tại mọi đường đi có thể từ nút (0,0) đến nút (i_k, j_k) . Để tiện cho việc kí hiệu, sau này ta sẽ dùng kí hiệu D thay cho D^* .

III. CÀI ĐẶT DTW

Trong [Stuart99] đề xuất hai giải thuật cho DTW cho các từ đơn lẻ tiếng Anh là giải thuật DTW đối xứng và DTW phản đối xứng. Trong bài báo này chúng tôi cài đặt cả hai giải thuật này và áp dụng cho các chữ số đơn lẻ tiếng Việt.

Thuật toán DTW đối xứng (Symmetric DTW - SDTW)

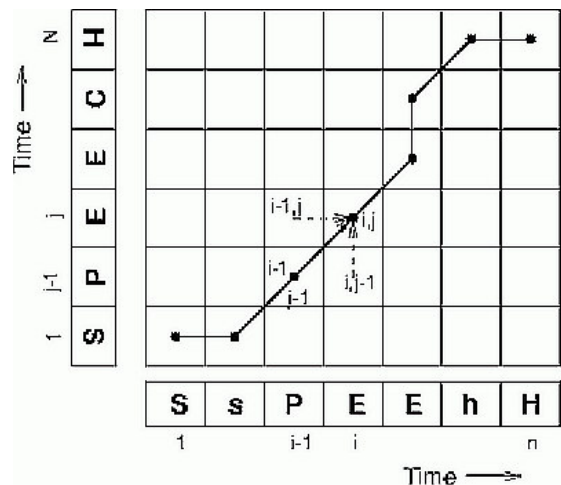
Trong thuật toán này ta cần phải tuân thủ các ràng buộc sau:

- Các đường so khớp không thể quay ngược chiều thời gian (nghĩa là chỉ có tiến lên)
- Tất cả các frame phải được sử dụng trong đường so khớp.
- Khoảng cách toàn cục được tính bằng cách cộng dồn tất cả các khoảng cách cục bộ.

Thuật toán được minh họa bằng ví dụ ở hình 2, ta sử dụng một ma trận thời gian-thời gian để mô tả trực quan cách căn chỉnh. Theo quy ước thì trục tung sẽ biểu diễn mẫu chuẩn "SPEECH", trục hoành biểu diễn mẫu nhận dạng "SsPEEhH". Trong ví dụ này mẫu nhận dạng "SsPEEhH" là mẫu có nhiều. ý tưởng ở đây là mẫu nhận dạng sẽ được so với tất cả các mẫu chuẩn thuộc cùng một lớp.

Mẫu nhận dạng phù hợp nhất là mẫu có khoảng cách đường căn chỉnh với mẫu chuẩn ngắn nhất. Một ý quan trọng ở thuật toán này là ta chỉ có thể đi đến ô (i,j) từ các ô (i-1,j-1), (i-1,j), (i,j-1) trong đó i là cột, j là hàng. Biểu thức quy hoạch động của DTW đối xứng là :

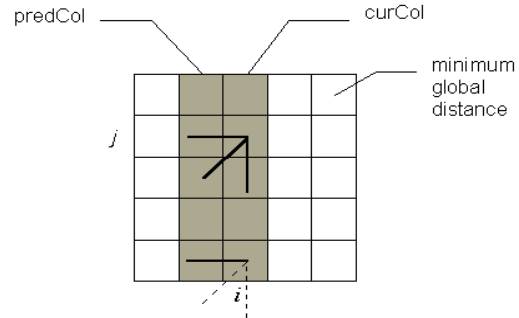
$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + d(i, j) \quad (9)$$



Hình 2: Minh họa về căn chỉnh thời gian động giữa mẫu chuẩn "SPEECH" và mẫu nhận dạng có nhiều "SsPEEhH"

với $d(i,j)$ là khoảng cách cục bộ tại (i,j) , $D(i,j)$ là khoảng cách toàn cục tại (i,j) .

Ta có thể nêu một cách cài đặt hiệu quả về bộ nhớ và khá nhanh trong tính toán của DTW đối xứng. Cải tiến ở đây là ta dùng hai mảng để lưu các cột liên tiếp. Ta quy ước các hàng, cột bắt đầu đánh số từ 0. Ô (i,j) và $(i,0)$ có sự khác nhau về các ô xuất phát. Đường đến ô $(i,0)$ chỉ có thể từ $(i-1,0)$, còn đường đến (i,j) có thể có 3 đường. Với $predCol$ và $curCol$ là hai ma trận một chiều có kích thước là Số vector đặc trưng của mẫu chuẩn. Hàm tính giá cục bộ giữa 2 vector x và y là hàm $d(x,y)$. Thuật toán có thể được diễn tả bằng ngôn ngữ tựa Pascal như sau:



Hình 3: Các hướng đi có thể của SDTW

1. Tính cột 0 ($predCol$)
2. FOR $i:=1$ TO Số vector đặc trưng của mẫu nhận dạng DO
3. Begin
4. $curCol[0] :=$ Giá cục bộ tại $(i,0)$ + Giá toàn cục tại $(i-1,0)$
5. FOR $j:=1$ TO Số vector đặc trưng của mẫu chuẩn DO
6. $curCol[j] :=$ Giá cục bộ tại (i,j) + MIN Giá toàn cục tại $((i-1,j-1), (i-1,j), (i,j-1))$
7. $predCol := curCol$
8. End
9. Giá toàn cục nhỏ nhất là $curCol[\text{Số vector đặc trưng của mẫu chuẩn}]$

Thuật toán DTW phản đối xứng (Asymmetric DTW - ASDTW)

Thuật toán DTW phản đối xứng là một cải tiến của DTW đối xứng. Biểu thức qui hoạch động như sau:

$$D(i, j) = \min\{D(i-1, j-1) + 2d(i, j), D(i-1, j) + d(i, j) + d_h, D(i, j-1) + d(i, j) + d_v\} \quad (10)$$

trong đó các giá trị d_h và d_v phù hợp có thể tìm được qua thực nghiệm. Ta có thể hiểu d_h và d_v là những trọng số ưu tiên, d_h là trọng số ưu tiên đường đi ngang (horizontal), d_v là trọng số ưu tiên đường đi dọc (vertical), các trọng số này có thể nhận giá trị 0.

Nếu chúng ta hạn chế các đường đi cho phép là:

- $(i-1, j-2)$ tới (i, j) hướng đi chéo mở rộng (bỏ qua một frame mẫu chuẩn, độ nghiêng chéo là 2)
- $(i-1, j-1)$ tới (i, j) hướng đi chéo chuẩn
- $(i-1, j)$ tới (i, j) hướng đi ngang (lặp lại một frame ở mẫu chuẩn, độ nghiêng 0)

thì chúng ta giả sử rằng mỗi frame của mẫu nhận dạng được sử dụng một và chỉ một lần. Điều đó có nghĩa rằng chúng ta có thể bỏ qua không cần đến sự chuẩn hóa độ dài mẫu và nó không đòi hỏi cộng hai lần khoảng cách cục bộ vào các hướng đi chéo chuẩn (độ nghiêng 1). Cách tiếp cận này được gọi là quy hoạch động phản đối xứng (asymmetric dynamic programming).

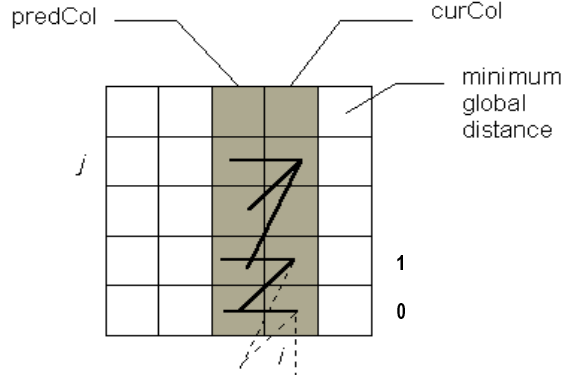
Theo hình 4 ta thấy có thể tới các ô (i,j) , $(i,0)$ và $(i,1)$ bằng các hướng khác nhau. Đường đi tới $(i,0)$ chỉ có thể xuất phát từ ô $(i-1,0)$. Đường đi tới $(i,1)$ chỉ có thể xuất phát từ ô $(i-1,0)$ và

$(i-1,1)$. Đường đi tới (i,j) có thể xuất phát từ 3 ô $(i-1,j)$, $(i-1,j-1)$ và $(i-1,j-2)$. Mặc dù vậy chúng ta cần lưu ý là sẽ có những trường hợp đặc biệt xảy ra. Hãy xem xét đường đi xuất phát từ ô $(0,0)$. Do đường đi sẽ phải tới cột 1 nên các hàng 1,2,... ở cột 0 sẽ trở nên vô nghĩa bởi vì chúng chắc chắn sẽ không được đi qua. Đường so khớp chỉ có thể đến được những ô đặc biệt từ một số vị trí giới hạn. Các trường hợp đặc biệt có $j=2i-1$ hoặc $2i$. Tổng giá cho mỗi trường hợp là:

- $2i-1$: Giá cục bộ + tổng khoảng cách nhỏ nhất tại $\text{predCol}[j-1]$ và $\text{predCol}[j-2]$
- $2i$: Giá cục bộ + tổng giá tại $\text{predCol}[j-2]$

Thuật giải tìm giá toàn cục ít nhất bây giờ sẽ phức tạp hơn đôi chút so với trường hợp đối xứng. Với predCol và curCol là hai ma trận

một chiều có kích thước là Số vector đặc trưng của mẫu chuẩn. Hàm tính giá cục bộ giữa 2 vector x và y là hàm $d(x,y)$. Hàm $\text{isSpecial}(\text{col},\text{row})$ kiểm tra một ô có là ô đặc biệt hay không. Ta trình bày thuật giải tựa Pascal như sau:



Hình 4: Các hướng đi có thể của DTW phản đối xứng

1. $\text{predCol}[0] =$ Giá cục bộ tại $(0,0)$
2. For $i:=1$ to số vector đặc trưng của mẫu nhận dạng DO
3. Begin
4. $\text{curCol}[0] =$ Giá cục bộ tại $(0,0) +$ Giá toàn cục tại $(i-1,0)$
5. For $j=1$ to (MIN của Số vector đặc trưng của mẫu chuẩn và $2i+1$)
6. begin
7. $\text{highestJ} = j$
8. If ô (i,j) là trường hợp đặc biệt Then
9. If (i,j) có hàng là 1 Then
10. $\text{curCol}[j] =$ Giá cục bộ tại $(i,j) +$ Giá toàn cục tại $(i-1,j-1)$
11. Else
12. If $(j \bmod 2 = 1)$ then
13. $\text{curCol}[j] =$ Giá cục bộ tại $(i,j) +$ MIN Giá toàn cục tại $((i-1,j-1), (i-1,j-2))$
14. Else
15. $\text{curCol}[j] =$ Giá cục bộ tại $(i,j) +$ Giá toàn cục tại $(i-1,j-2)$
16. Else
17. If (i,j) có hàng là 1 Then
18. $\text{curCol}[j] =$ Giá cục bộ tại $(i,j) +$ MIN Giá toàn cục tại $((i-1,j), (i-1,j-1))$
19. Else
20. $\text{curCol}[j] :=$ Giá cục bộ tại $(i,j) +$ MIN Giá toàn cục tại $((i-1,j-1), (i-1,j), (i-1,j-2))$
21. end
22. $\text{predCol} = \text{curCol}$
23. End
24. Giá toàn cục nhỏ nhất là giá trị ở $\text{curCol}[\text{highestJ}]$

IV. THÍ NGHIỆM

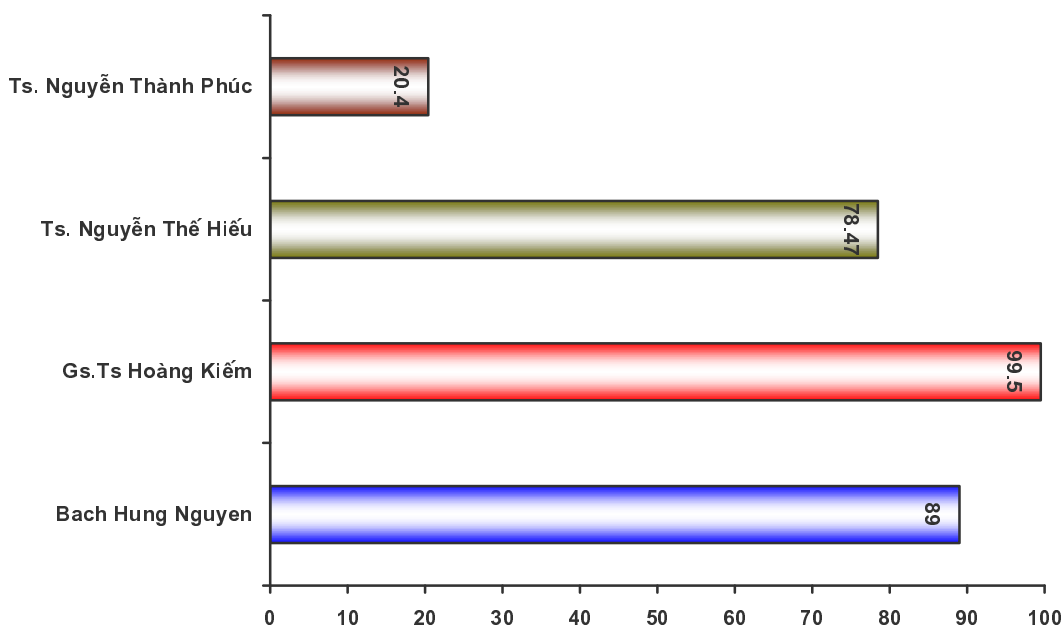
Hai thuật toán trên được cài đặt bằng ngôn ngữ C và chạy thử nghiệm trên DOS và Windows98. Tập từ được kiểm tra nhận dạng là {Không, Một, Hai, Ba, Bốn, Năm, Sáu, Bảy, Tám, Chín}. Dữ liệu mẫu cơ sở gồm 300 file dữ liệu của 3 người nói khác nhau, 1 nam, 2 nữ. Mỗi người có 100 file, mỗi từ được nói 10 lần. Dữ liệu kiểm tra gồm 300 file dữ liệu cũng của 3 người đó, mỗi người có 100 file, mỗi từ được nói 10 lần. Dữ liệu mẫu và dữ liệu kiểm tra

được thu vào những ngày khác nhau và tại những thời điểm khác nhau để đảm bảo tính khách quan của giọng đọc. Người nói sẽ đọc lần lượt các từ Không, Một, ..., Chín, sau đó lặp lại như vậy 10 lần. Ngoài ra còn thu thêm 100 file dữ liệu test của người thứ 4, để kiểm tra khả năng nhận dạng từ hoàn toàn độc lập.

Có hai kiểu kiểm tra chính là nhận dạng từ phụ thuộc vào người nói và nhận dạng từ độc lập với người nói. Ngoài ra còn có kiểu kiểm tra là nhận dạng phụ thuộc/độc lập người nói nhưng không có tính năng học mẫu mới và nhận dạng phụ thuộc/độc lập người nói nhưng có tính năng học mẫu mới. Các file dữ liệu chứa các véc tơ đặc trưng của những mẫu âm thanh đã được thu. Các véc tơ này được sinh ra nhờ phép phân tích LPC. Trong mỗi thí nghiệm, với 100 file dữ liệu test, số lần nhận dạng đúng sẽ là số % chính xác của chương trình. Kết quả kiểm tra được biểu diễn bằng ma trận kết quả theo mô tả trong [Hendrik98]. Các điều kiện khi tiến hành thí nghiệm:

Máy tính	CPU PenIII500Hz, 128MbRAM, Soudcard Yamaha 724, Microphone Headmounted OS Window98 hoặc DOS
Định dạng tiếng	Wave file, 16Bit Mono, 16KHz, Số hệ số LPC 15, Độ dài mỗi frame 20m.s, Phần gối lên nhau 10m.s
Môi trường thu tiếng	Phòng sinh hoạt bình thường

Trên cơ sở 1200 lần tiến hành thí nghiệm nhận dạng, chúng tôi đưa ra một số kết luận về việc áp dụng kĩ thuật DTW trong nhận dạng các chữ số đơn lẻ. Thực hiện việc so sánh kết quả về độ chính xác với các đề tài đã được thực hiện như [Hoang96] chúng tôi rút ra kết luận là để đạt tới độ chính xác cao (95% - 99,9%) thì nhất thiết DTW phải áp dụng các tri thức về ngữ âm học. Độ chính xác cao nhất cho toàn bộ 10 chữ số là 89%.



Qua các kết quả thí nghiệm thấy rằng các chữ số 9,0,7 thường cho kết quả nhận dạng chính xác cao, đặc biệt là số 9. DTW thường nhận sai các từ có nguyên âm giống nhau hoặc gần giống nhau. Trong tiếng Việt số 2 và 7 có cùng nguyên âm /ai/, số 5 và 8 có cùng nguyên âm /a/. Vấn đề nhiều ảnh hưởng rất nhiều đến độ chính xác của kết quả nhận dạng. Do vậy để thu được kết quả nhận dạng tốt cần phải đầu tư công sức nghiên cứu vấn đề xử lý nhiễu. Thuật toán DTW bất đối xứng trong một số trường hợp cho kết quả nhận dạng tốt hơn DTW đối

xúng, tuy nhiên thuật toán DTW đối xứng qua các thí nghiệm chạy đường như ổn định và cho kết quả nhận dạng hợp lí hơn.

V. KẾT LUẬN

Trong bài báo này chúng tôi đã nghiên cứu và áp dụng kĩ thuật DTW vào nhận dạng các chữ số đơn lẻ tiếng Việt. Mặc dù kết quả đạt được còn khiêm tốn nhưng chúng tôi thiết nghĩ đây là những bước nghiên cứu khởi đầu cho một hướng nghiên cứu còn khá mới ở Việt nam. Trong thời gian tới để nâng cao chất lượng nhận dạng khi áp dụng kĩ thuật này chúng tôi sẽ tập trung vào việc thử nghiệm một số giá trị p (là số chiều của véc tơ đặc trưng) để tìm ra giá trị p thích hợp với ngôn ngữ tiếng Việt; nghiên cứu bản chất của chuỗi véc tơ đặc trưng nhằm tìm thêm các đặc trưng mới, từ đó sử dụng các đặc trưng mới này nâng cao chất lượng nhận dạng; nghiên cứu một số phương pháp tạo mẫu đặc trưng, cải tiến thuật toán để rút ngắn thời gian đưa ra kết quả nhận dạng; cài đặt thử một số loại khoảng cách khác như: Itakura, Mahalanobis, ... để so sánh và lựa chọn loại tối ưu nhất; thực hiện khắc phục nhiễu bằng các thiết bị phần cứng chuyên dụng và các bộ lọc số phần mềm; nghiên cứu thuật toán học mẫu bằng cách dựa vào các đặc trưng của tiếng nói và tiếng Việt.

TÀI LIỆU THAM KHẢO

[Yu00] Yu, Z.: *Dynamic Time Warping Digit Recognizer*, Project Report for Degree of Master of Science, Mississippi State University, 2000.

[Serrgios99] Serrgios, T., Konstantinos, K.: *Pattern Recognition*, Academic Press, 1999, Chapter 7-9.

[Stuart99] Stuart, N.W.: *Speech Recognition by DTW*, Sheffield University, UK, 1999.

[Hendrik98] Hendrik, W.: *Speech Recognition*, Imperial College, 1998.

[Lawrence93] Lawrence, R., Bing, H., J.: *Fundamentals of Speech Recognition*, Prentice Hall Inc, 1993, Chapter 1-6.

[Hoang96] Hoang, K., Nguyen, M., T.: *Some Developed Algorithms in Vietnamese Isolated Word Recognition*, Hội nghị Khoa học Kỷ niệm 20 năm thành lập Viện CNTT, 1996.