# LỜI CẢM ƠN

Sau thời gian thực tập tại Viện nghiên cứu MICA và thực hiện đồ án tốt nghiệp , em đã hoàn thành đề tài : " Xây dựng hệ thống nhận dạng online trên nền tảng Sphinx4 ". Đầu tiên , em xin gửi lời cảm ơn chân thành tới Tiến sĩ Nguyễn Việt Sơn, giảng viên của trường Đại Học Bách Khoa Hà Nội, cán bộ nghiên cứu thuộc Viện nghiên cứu quốc tế MICA. Cảm ơn Thầy đã nhiệt tình giúp đỡ và chỉ bảo trong suốt quá trình thực hiện đề tài này.

Em cũng xin cảm ơn ban lãnh đạo MICA, TS Nguyễn Quốc Cường , TS Trần Đỗ Đạt , anh Đăng Khoa ... đã tạo mọi điều kiện thuận lợi cho em được thực hiện đồ án tại Viện. Cảm ơn các thầy cô, các anh chị trên Viện đã cho em những lời khuyên bổ ích, thiết thực cho đề tài của mình.

Cuối cùng, xin gửi lời cảm ơn sâu sắc tới bạn bè, gia đình, những người đã luôn động viên, giúp đỡ em trong suốt quá trình thực hiện đề tài này.

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc357925338)

[MỤC LỤC 2](#_Toc357925339)

[DANH MỤC HÌNH VẼ 4](#_Toc357925340)

[CHƯƠNG 1 : GIỚI THIỆU 6](#_Toc357925341)

[1.1. Một số ứng dụng của hệ thống nhận dạng tiếng nói. 6](#_Toc357925342)

[1.2. Các khó khăn trong ứng dụng nhận dạng tiếng nói 7](#_Toc357925343)

[1.3. Mục tiêu đề tài 8](#_Toc357925344)

[CHƯƠNG 2 : TỔNG QUAN VỀ NHẬN DẠNG TIẾNG NÓI VÀ CÁC GIẢI THUẬT SỬ DỤNG 9](#_Toc357925345)

[2.1. Tổng quan về hệ thống ARS. 9](#_Toc357925346)

[2.1.1. Sơ đồ khối hệ thống nhận dạng tiếng nói 9](#_Toc357925347)

[2.1.2. Trích chọn đặc trưng. 10](#_Toc357925348)

[2.1.3. Huấn luyện và nhận dạng các đặc trưng. 11](#_Toc357925349)

[2.2. Các giải thuật sử dụng 12](#_Toc357925350)

[2.2.1. Thuật toán MFCC ( Mel – Frequency Cepstrum Coefficients ) 12](#_Toc357925351)

[2.2.1.1. Tín hiệu âm thanh đầu vào ( Speech signal ) 13](#_Toc357925352)

[2.2.1.2. Pre - Emphasis 13](#_Toc357925353)

[2.2.1.3. Phân khung và cửa sổ hóa tín hiệu 14](#_Toc357925354)

[2.2.1.4. DFT (Discrete Fourier Transform) 17](#_Toc357925355)

[2.2.1.5. Thang tần số Mel 18](#_Toc357925356)

[2.2.1.6. Logarit bình phương biên độ 19](#_Toc357925357)

[2.2.1.7. IDFT 20](#_Toc357925358)

[2.2.1.8. Các hệ số đặc trưng trong MFCC 20](#_Toc357925359)

[2.2.2. Huấn luyện và nhận dạng 21](#_Toc357925360)

[2.2.2.1. Mô hình Markov ẩn (Hidden Markov Model - HMM) 21](#_Toc357925361)

[2.2.2.2. Mô hình Markov liên tục (Continuous density HMM) 27](#_Toc357925362)

[2.2.2.3. Mô hình ngôn ngữ 28](#_Toc357925363)

[CHƯƠNG 3 : XÂY DỰNG HỆ THỐNG NHẬN DẠNG TRÊN SPHINX4 30](#_Toc357925364)

[3.1. Giới thiệu về Sphinx4 30](#_Toc357925365)

[3.1.1. Tổng quan về Sphinx4 30](#_Toc357925366)

[3.1.2.Những công cụ cần trang bị 30](#_Toc357925367)

[3.2. Xây dựng mô hình nhận dạng trên Sphinx4 31](#_Toc357925368)

[3.2.1. Chuẩn bị cơ sở dữ liệu 31](#_Toc357925369)

[3.2.2. Xây dựng thư viện cơ sở dữ liệu và các mô hình 34](#_Toc357925370)

[3.3.Xây dựng chương trình nhận dạng 37](#_Toc357925371)

[3.3.1. Sơ đồ khối chương trình 37](#_Toc357925372)

[3.3.2. Mô tả hoạt động 38](#_Toc357925373)

[3.4. Kết quả: 39](#_Toc357925374)

[3.4.1.Đánh giá tốc độ gửi nhận client-server 39](#_Toc357925375)

[3.4.2. Tốc độ nhận dạng 40](#_Toc357925376)

[3.4.3. Tỉ lệ nhận dạng 40](#_Toc357925377)

[CHƯƠNG 4 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 44](#_Toc357925378)

[Tài liệu tham khảo 45](#_Toc357925379)

# DANH MỤC HÌNH VẼ

Hình 2.1. Sơ đồ khối của hệ thống nhận dạng tiếng nói

Hình 2.2. Sơ đồ khối phương pháp LPC

Hình 2.3.Sơ đồ thuật toán MFCC

Hình 2.4. Rời rạc hóa tín hiệu tiếng nói

Hình 2.5. Phổ của âm [aa] trước và sau khi thực hiện Pre - emphasis

Hình 2.6. Chia tín hiệu thành các frame nhỏ

Hình 2.7.Cửa sổ Hamming 128 điểm.

Hình 2.8. Ảnh hưởng của cửa sổ Hamming lên tín hiệu.

Hình 2.9. Biên độ phổ sau khi FFT.

Hình 2.10. Thang tần số Mel

Hình 2.11. Tính toán sử dụng thang tần số Mel

Hình 2.12. Tính log bình phương biên độ của tín hiệu đầu ra bộ lọc tần số Mel

Hình 2.13. Hệ số đặc trưng delta và doubledelta.

Hình 2.14. Mô hình Markov ẩn liên tục

Hình 3.1. Sơ đồ khối chương trình nhận dạng

**BẢNG CÁC THUẬT NGỮ VIẾT TẮT**

|  |  |
| --- | --- |
| THUẬT NGỮ  VIẾT TẮT | THUẬT NGỮ ĐẦY ĐỦ |
| *ASR* | *Automatic Speech Recognition* |
| *HMI* | *Human Machine Interface* |
| *WER* | *Word Error Rate* |
| *LPC* | *Linear Prediction Coding* |
| *MFCC* | *Mel – Frequency Cepstrum Coefficients* |
| *PCA* | *Principle Components Analysis* |
| *DTW* | *Dynamic Time Warping* |
| *VQ* | *Vector Quantization* |
| *HMM* | *Hidden Markov Modeling* |
| *ANN* | *Artifitial Neural Network* |
| *DFT* | *Discrete Fourier Transform* |
| *IDFT* | [Inverse Discrete Fourier Transform](https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&sqi=2&ved=0CEEQFjAC&url=http%3A%2F%2Fwww.mathworks.com%2Fhelp%2Fmatlab%2Fref%2Fifft.html&ei=2YulUbPHBLGujALugoHwBQ&usg=AFQjCNFwNcZQirnqABnx3vCGFgoiTPzsuw&sig2=6sdckD5nJ3GJHTF1_OWe2Q&bvm=bv.47008514,d.cGE) |
| *IFFT* | [Inverse Fast Fourier Transform](https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&sqi=2&ved=0CEEQFjAC&url=http%3A%2F%2Fwww.mathworks.com%2Fhelp%2Fmatlab%2Fref%2Fifft.html&ei=2YulUbPHBLGujALugoHwBQ&usg=AFQjCNFwNcZQirnqABnx3vCGFgoiTPzsuw&sig2=6sdckD5nJ3GJHTF1_OWe2Q&bvm=bv.47008514,d.cGE) |
| *DCT* | *Discrite Cosin Transform* |

# CHƯƠNG 1 : GIỚI THIỆU

## 1.1. Một số ứng dụng của hệ thống nhận dạng tiếng nói.

Hệ thống nhận dạng tiếng nói ASR ( Automatic Speech Recognition) có những ứng dụng tuyệt vời trong tất cả các lĩnh vực của đời sống, nếu được áp dụng thành công sẽ là một cuộc cách mạng trong giao tiếp người - máy HMI (Human Machine Interface ), các ứng dụng của nó bao trùm trên nhiều lĩnh vực như công nghiệp, an ninh và giải trí ...

**➢ Trong lĩnh vực điều khiển, tìm kiếm :**

Các hệ thống với bộ từ vựng nhỏ, nhập từ rời rạc có thể áp dụng trong những ứng dụng tương đối đơn giản để cải thiện hiệu quả nhập thông tin vào máy (nhập tiếng nói nhanh gấp hai lần nhập thông tin bằng cách gõ chữ). Nó còn có thể ứng dụng trong môi trường sản xuất (công việc phân loại) , trong những trường hợp mà đôi tay không thể thực hiện (chẳng hạn như trong phòng tối, trong buồng lái…), trong các ứng dụng điều khiển từ xa với thiết bị, điều khiển robot, đồ chơi trẻ em, hay trong các thiết bị yêu cầu thu nhỏ phải loại bỏ hệ thống phím nhấn. Ngoài ra, nhận dạng giọng nói có thể sẽ là một phương pháp hữu hiệu giúp cho người khiếm thị dễ dàng giao tiếp điều khiển với thiết bị. Nói chung là trong những nhiệm vụ đặc biệt có xu hướng giới hạn bộ từ vựng và nội dung thông điệp.

Những ứng dụng thực tiễn mà hệ thống này sẽ mang lại là vô cùng to lớn như các máy tính của chúng ta sẽ không cần các bàn phím, các hệ thống điều khiển sẽ không cần bảng điều khiển phức tạp, máy điện thoại sẽ không còn cần đến các bàn quay số...

Các ứng dụng như thế này có thể tìm thấy ở những hê thống hiện đại như : ứng dụng SIRI của Apple hay Google Voice Search của Google ...

**➢ Trong lĩnh vực chuyển đổi tín hiệu:**

Trong các cuộc nói chuyện do bất đồng ngôn ngữ, hay do những vấn đề tế nhị về văn hóa mỗi dân tộc, để chuyển đổi qua lại giữa hai ngôn ngữ, cùng với hệ thống dịch thuật trên văn bản kết hợp với hai hệ thống nhận dạng tiếng nói sẽ cho phép cuộc nói chuyện diễn ra bình thường và tự nhiên. Hệ thống chuyển đổi ngôn ngữ trực tiếp này rất hữu ích trong các cuộc hội thảo lớn có nhiều quốc gia, dân tộc tham dự.

Tuy nhiên, hệ thống kiểu như thế này đòi hỏi khả năng nhận dạng rất lớn, cơ sở dữ liệu khổng lồ. Cho tới thời điểm hiện tại, mức độ ứng dụng còn rất hạn chế.

**➢ Trong lĩnh vực nhận diện:**

Hệ thống nhận dạng tiếng nói kết hợp với xử lý tổng hợp giọng nói còn được ứng dụng trong lĩnh vực nhận diện tiếng nói. Hệ thống mật mã giọng nói cho phép nhận dạng người thông qua tiếng nói, chẳng hạn rút tiền ra khỏi ngân hàng hay các tác vụ khác mà không cần kiểm tra chữ ký hay các giấy tờ khác vì có yêu cầu bí mật về nhân thân. Hoặc ứng dụng trong các hệ thống khoá tự động mà chìa khoá là tiếng nói. Hệ ASR như vậy đã có những áp dụng trong thực tế. Có thể tham khảo thêm tại địa chỉ <http://en.wikipedia.org/wiki/Speech_recognition> cho các ứng dụng của hệ thống ASR trong thực tế.

## 1.2. Các khó khăn trong ứng dụng nhận dạng tiếng nói

Việc ứng dụng hệ thống ASR cho đến thời điểm hiện nay vẫn còn nhiều hạn chế là vì một số khó khăn do bản thân đối tượng mà nó nhắm đến, trong trường hợp này là tiếng nói hay âm thanh, là một đối tượng không ổn định. Các khó khăn có thể kể đến như:

– Sự biến động của người nói trong việc phát âm:

+ Tiếng nói thay đổi theo thời gian, theo độ tuổi.

+ Tình trạng sức khỏe. Một người khi khỏe mạnh sẽ phát âm khác hẳn so với khi gặp ốm đau, ví dụ như cảm cúm chẳng hạn.

+ Tốc độ nói.

+ Với một người, trong một khoảng thời gian ngắn, việc phát âm một từ trong nhiều lần khác nhau có thể khác nhau.

– Ảnh hưởng của ngoại cảnh:

+ Nhiễu, tiếng ồn của môi trường xung quanh.

Ví dụ một người nói trong không gian yên tĩnh sẽ dễ nghe hơn là ở ngoài đường phố.

+ Microphone để thu âm có thể khác nhau trong những tình huống khác nhau.

+ Khoảng cách từ miệng người nói đến Microphone .

Điều kiện lý tưởng cho việc thực hiện nhận dạng tiếng nói nói chung và âm thanh nói riêng là tiếng nói ổn định kể cả trong lúc huấn luyện và lúc nhận dạng. Tiếng nói của mỗi người là duy nhất, không trùng lẫn với những người khác. Do đó, cho đến thời điểm hiện tại, việc nhận dạng âm thanh, tiếng nói là một công việc rất khó khăn.

## 1.3. Mục tiêu đề tài

**-** Xây dựng hệ thống nhận dạng tiếng Việt online đặt trên server ( có thể sử dụng server MICA hoặc laptop) trên nền tảng Sphinx4. Trong phạm vi đề tài , em tập trung xây dựng cơ sở dữ liệu xoay quanh một số chủ đề chính : tin nhắn (soạn tin nhắn bằng tiếng nói, ra lệnh gửi, hủy... ), truy vấn thông tin (thời tiết, giá vàng,ngoại tệ, chứng khoán...) trên Internet. Mô hình nhận dạng dựa trên bộ từ vựng nhỏ (từ điển khoảng 250 từ), các kịch bản huấn luyện được xây dựng từ trước tập trung vào các chủ đề trên.

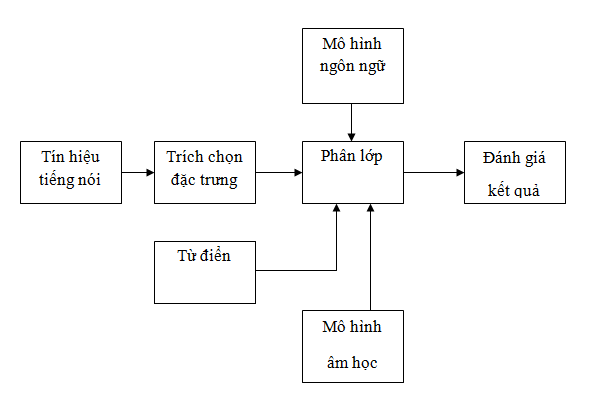
- Server nhận dạng và Client tạo ra trên nền tảng JAVA, kết nối với nhau theo phương thức Socket TCP-IP. Khi client hoat động, chương trình ghi lại tiếng nói của con người , tự động ngắt sau 1,5 giây không có lời nói, rồi gửi lên server. Server nhận file, thực hiện quá trình nhận dạng rồi trả về kết quả (file text) cho client đã yêu cầu.

- Đánh giá các kết quả đã thực hiện, cách cải thiện chất lượng nhận dạng và hướng phát triển .

# CHƯƠNG 2 : TỔNG QUAN VỀ NHẬN DẠNG TIẾNG NÓI VÀ CÁC GIẢI THUẬT SỬ DỤNG

## 2.1. Tổng quan về hệ thống ARS.

### 2.1.1. Sơ đồ khối hệ thống nhận dạng tiếng nói

****

Hình 2.1. Sơ đồ khối của hệ thống nhận dạng tiếng nói

Tín hiệu tiếng nói sau khi được lấy mẫu được đưa qua khối xử lí tín hiệu để trích chọn đặc trưng phục vụ quá trình nhận dạng. Sau khi trích chọn đặc trưng, các vectơ đặc trưng được đưa vào khối nhận dạng, kết hợp với các hiểu biết về đặc trưng ngôn ngữ và đặc trưng âm học của tiếng nói, khối nhận dạng sẽ đưa ra kết quả nhận dạng hợp lí nhất. Bộ tham số của mô hình nhận dạng thu được trong quá trình huấn luyện mô hình với một tập dữ liệu đào tạo đã có trước. Kích thước của tập đào tạo càng lớn thì khả năng nhận dạng của mô hình càng cao.

Để đánh giá các mô hình nhận dạng tiếng nói, người ta thường sử dụng sai số nhận dạng cấp từ WER (word error rate ) được tính theo công thức :

 (2.1)

trong đó : Nword  là tổng số từ trong văn bản

Ins là số từ bị thêm vào do sai số

Del là số từ bị mất khi nhận dạng

Sub là số từ nhận dạng sai

Như sơ đồ trên, tất cả các hệ thống ASR bao gồm 2 module chính: trích chọn đặc trưng (feature extraction ) và nhận dạng đặc trưng ( feature matching ).

– Feature extraction: tách các dữ liệu từ tiếng nói đặc trưng nhất của người nói vào.

– Feature matching: là quá trình nhận dạng câu nói đang giao tiếp với hệ thống bằng cách so sánh các đặc trưng của câu nói này với những câu đã được huấn luyện.

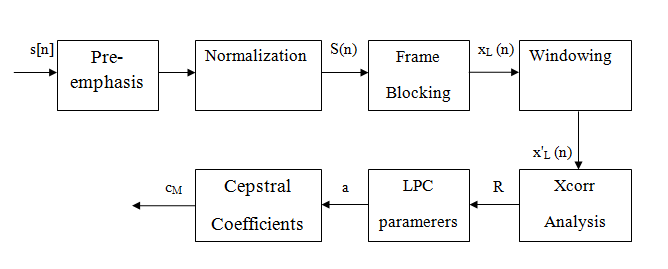
Hai phần này sẽ được nói rõ hơn trong phần 2.2 và phần 2.3.

### 2.1.2. Trích chọn đặc trưng.

Trích chọn đặc trưng của mẫu là một phần quan trọng của bất cứ hệ thống nhận dạng nào. Một cách lý tưởng, một đối tượng khác nhau sẽ có một hoặc nhiều đặc trưng. Các đặc trưng càng khác nhau giữa các đối tượng thì việc nhận dạng càng chính xác.

Việc nhận dạng sẽ dựa trên các đặc trưng này, có thể sử dụng 1 đặc trưng hoặc kết hợp nhiều đặc trưng lại với nhau. Với các hệ thống ASR hiện nay, thường chỉ sử dụng một đặc trưng của tín hiệu âm thanh. Cho tới thời điểm hiện nay, các phương pháp chủ yếu để tách đặc trưng có thể kể đến như: Linear Prediction Coding (LPC), Mel – Frequency Cepstrum Coefficients ( MFCC ), Principle Components Analysis (PCA) và các phương pháp khác...

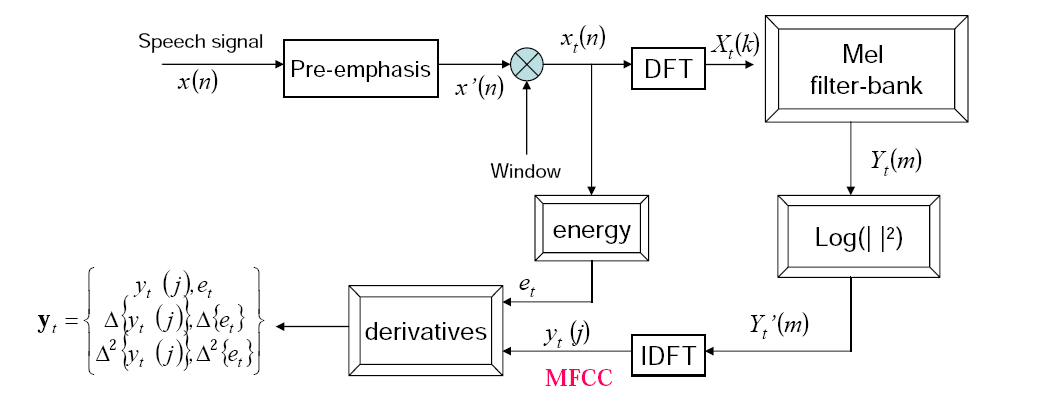
– Linear Prediction Coding.



Hình 2.2. Sơ đồ khối phương pháp LPC

Toàn bộ một hệ thống ASR dùng phương pháp LPC được miêu tả đầy đủ tại địa chỉ : <http://www.clear.rice.edu/elec532/PROJECTS98/speech/cepstrum/cepstrum.html>

– Mel – Frequency Cepstrum Coefficients.



Hình 2.3.Sơ đồ thuật toán MFCC

Toàn bộ hệ thống ASR dùng phương pháp MFCC được miêu tả chi tiết tại :

LSA 352 Speech Recognition and Synthesis - Dan Jurafsky

### 2.1.3. Huấn luyện và nhận dạng các đặc trưng.

Các đặc trưng sau khi được tạo thành, dù bằng bất cứ phương pháp nào cũng sẽ được dùng để huấn luyện tạo cơ sở dữ liệu và nhận dạng về sau.

Các kĩ thuật chính được dùng trong việc nhận dạng âm thanh có thể kể đến là:

– **Dynamic Time Warping – DTW**.

Phương pháp này phổ biến trong thập niên 1970 & 1980.Hệ thống ASR dùng DTW được miêu tả chi tiết với MATLAB tại : <http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/>

– **Vector Quantization – VQ**.

Phương pháp này cho kết quả khả quan hơn so với phương pháp DTW, trong phạm vi từ vựng nhỏ ( khoảng 20 từ). Phương pháp này được mô tả chi tiết trong :

LSA 352 Speech Recognition and Synthesis - Dan Jurafsky.

– **Hidden Markov Modeling/Artifitial Neural Network – HMM/ANN**.

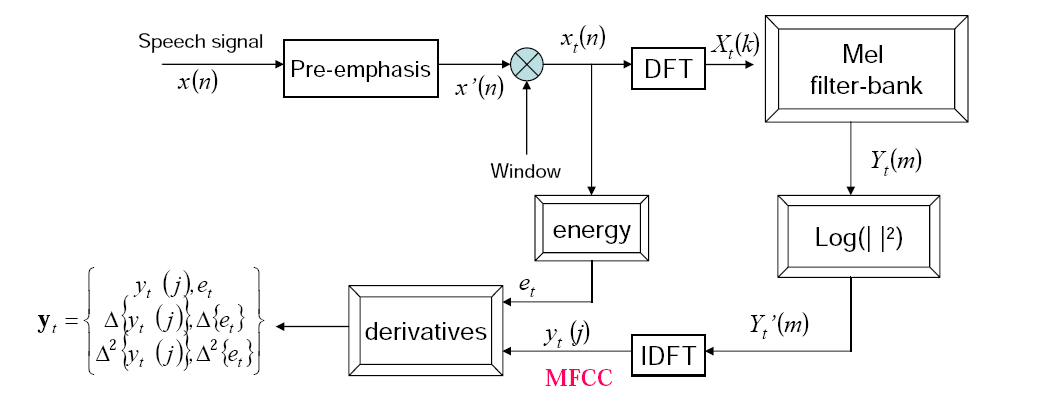
HMM là kĩ thuật mới nhất, được dùng trong các hệ thống đòi hỏi số từ vựng rất lớn, lên đến hàng ngàn từ. Mô hình này sẽ được mô tả chi tiết ở phần sau.

## 2.2. Các giải thuật sử dụng

Trong phạm vi đề tài, ứng dụng chỉ tập trung vào một số chủ đề cụ thể (tin nhắn, truy vấn thông tin ...) với cơ sở dữ liệu trung bình (khoảng 250 từ). Vì vậy, em đã chọn công cụ sphinx4 để phát triển hệ thống nhận dạng. Thuật toán thực hiện quá trình trích chọn đặc trưng là MFCC (Mel – Frequency Cepstrum Coefficients ), và kĩ thuật HMM (Hidden Markov Model ) được dùng cho việc huấn luyện và nhận dạng. Hai kĩ thuật này phù hợp với yêu cầu đặt ra của ứng dụng.

### 2.2.1. Thuật toán MFCC ( Mel – Frequency Cepstrum Coefficients )

Với quy trình được thể hiện ở mô hình sau:

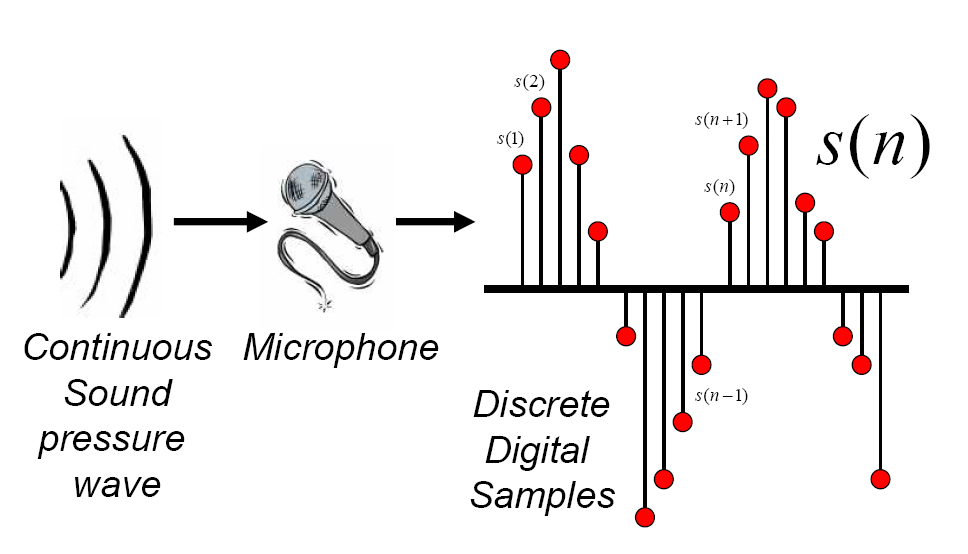


Hình 2.3. Sơ đồ thuật toán MFCC

#### 2.2.1.1. Tín hiệu âm thanh đầu vào ( Speech signal )

- Quá trình thu âm lời nói thực chất là quá trình biến đổi một tín hiệu liên tục thành một tín hiệu rời rạc (qua microphone).

Tần số lấy mẫu của Microphone thường chọn 16 KHz , còn tần số lấy mẫu của Telephone ( điện thoại) thường là 8 KHz. Lí do là bởi tai người có thể nghe được các âm thanh có tần số từ 20 Hz đến 10 KHz ( thính nhất trong khoảng từ 100Hz đến 5 KHz). Theo định lý lấy mẫu Shannon, tần số lấy mẫu phải lớn hơn 2 lần tần số lớn nhất đo được. Vì vậy, việc chọn tần số lấy mẫu của Microphone 16 KHz là hợp lí.

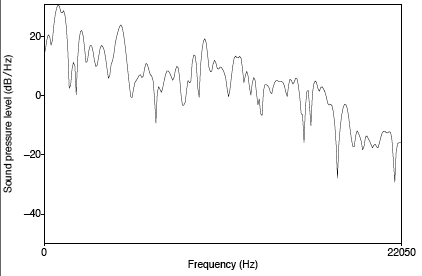
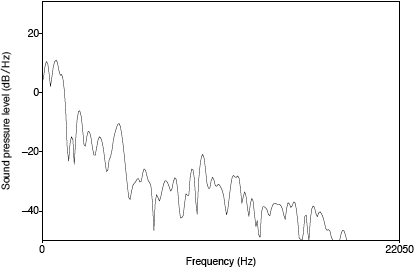


Hình 2.4. Rời rạc hóa tín hiệu tiếng nói

#### 2.2.1.2. Pre - Emphasis

- Mục đích : tăng cường năng lượng của các âm thanh ở tần số cao.

Lí do là bởi phổ năng lượng của âm thanh ở tần số thấp lớn hơn phổ năng lượng âm thanh ở tần số cao. Vì vậy , việc mất mát thông tin hoặc sai lệch thông tin với các âm ở tần số cao thường hay xảy ra. Thực hiện Pre - emphasis sẽ đem lại nhiều thông tin hơn cho mô hình âm học ( Acoustic Model) , qua đó cải thiện chất lượng nhận dạng.

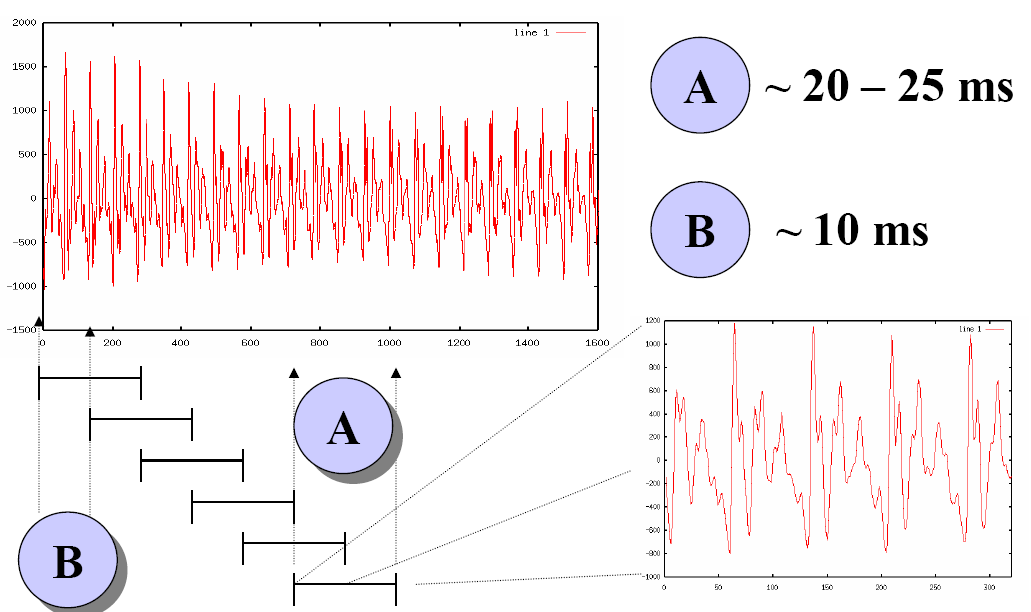


Hình 2.5. Phổ của âm [aa] trước và sau khi thực hiện Pre - emphasis

#### 2.2.1.3. Phân khung và cửa sổ hóa tín hiệu

Đặc điểm của tín hiệu tiếng nói là tín hiệu chậm biến đổi theo thời gian, hay còn được gọi là “quasi – stationary”. Khi xem xét một khoảng thời gian đủ ngắn, khoảng từ 5 – 100ms, tính chất của nó gần như là ổn định. Tuy nhiên, trong một khoảng thời gian dài, tính chất của nó bị thay đổi, phản ánh những âm thanh khác nhau được nói.

Vì vậy phân tích khoảng thời gian ngắn (short time spectral analysis) thường được sử dụng trong phân tích tính chất của tín hiệu tiếng nói. Trong khi một từ được phát ra có thể dài vài giây, nên cần thiết phải chia các tín hiệu thu được thành các frame nhỏ, các frame này có độ dài tương ứng 20 – 25 ms, hai frame liên tiếp cách nhau khoảng 10 ms.



Hình 2.6. Chia tín hiệu thành các frame nhỏ

Để tránh sự thay đổi đột ngột giữa các frame, sẽ có sự lặp lại giữa các frame liên tiếp. Ví dụ, mỗi frame đầu tiên có N mẫu; frame kế tiếp cũng sẽ có N mẫu, nhưng chỉ có M ( M < N ) mẫu là “mới”, còn lại N – M mẫu đầu tiên là N - M mẫu cuối cùng của frame đầu tiên, quá trình như vậy tiếp diễn cho những frame sau.

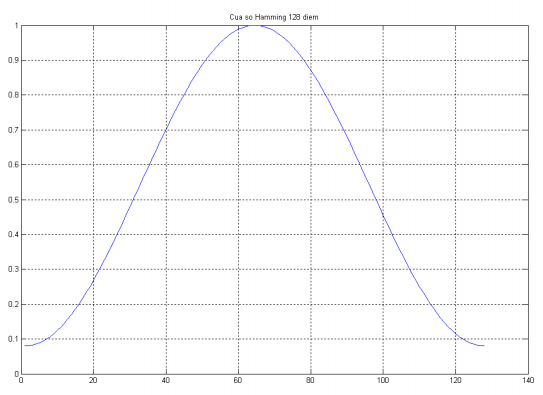
Bước tiếp theo là cửa sổ hóa mỗi frame để giảm sự không liên tục giữa điểm đầu và điểm cuối của mỗi frame. Nguyên tắc của việc cửa sổ hóa là giảm thiểu méo dạng phổ bằng cách làm hẹp dần ở phía đầu và cuối của mỗi frame.

Gọi cửa sổ là w(n), 0 ≤ n ≤ N – 1, với N là số phần tử trong mỗi frame. Tín hiệu sau khi cửa số hóa như sau:

y1(n) = x1(n) \* w(n) 0 ≤ n ≤ N – 1. (2.2)

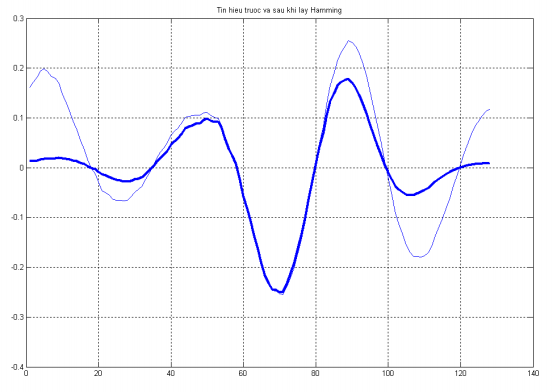
 Thông thường cửa sổ Hamming được dùng, có dạng như sau:

  (2.3)



Hình 2.7.Cửa sổ Hamming 128 điểm.

Tín hiệu sau khi Hamming hóa sẽ bị “thu nhỏ” lại ở hai đầu :



Hình 2.8. Ảnh hưởng của cửa sổ Hamming lên tín hiệu.

#### 2.2.1.4. DFT (Discrete Fourier Transform)

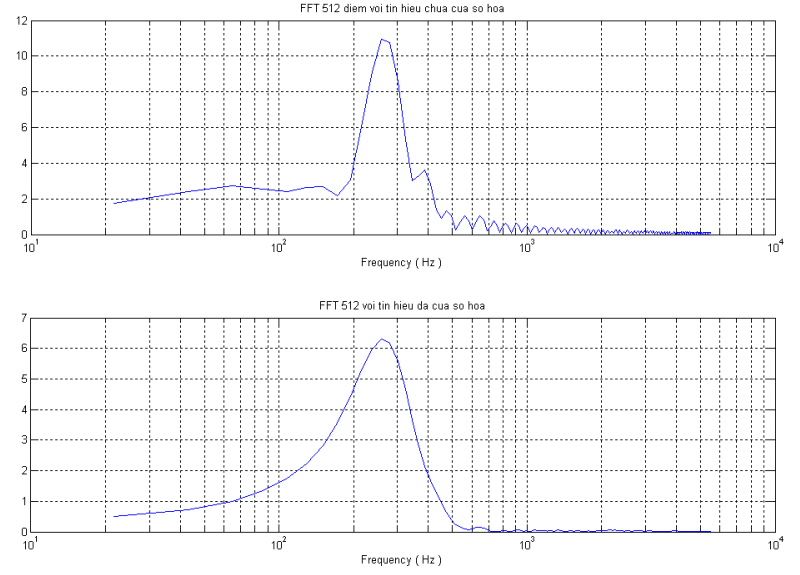
Mục đích của quá trình này là chuyển đổi tín hiệu thời gian thành tín hiệu tần số.

**-** Đầu vào : tín hiệu sau khi nhân với hàm cửa sổ x[n], x[m].

- Đầu ra : Số phức X[k] đại diện cho thành phần tần số của tín hiệu ban đầu

Thông thường, ta hay sử dụng thuật toán FFT ( Fast Fourier Transform ) để thực hiện nhanh DFT. Công thức tính :  (2.4)

Thông thường , chọn N = 512 hoặc 1024.



Hình 2.9. Biên độ phổ sau khi FFT.

Hình trên tín hiệu không có cửa sổ hóa.

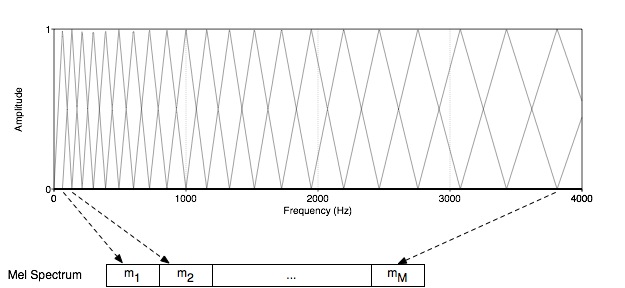
Hình dưới, tín hiệu có bị cửa sổ hóa bằng Hamming.

#### 2.2.1.5. Thang tần số Mel

Một số nghiên cứu vật lý về tai người cho thấy phản ứng của tai người với tín hiệu tiếng nói không tuân theo quy luật tuyến tính về tần số. Vậy một cách tiếp cận chủ quan, mỗi tín hiệu âm thanh được phát ra sẽ đươc chuyển đổi lại cho phù hợp, lúc này tần số mel ( mel frequency ) được sử dụng. Mel – frequency tuyến tính ở tần số dưới 1Khz và logarithmic ở tần số trên 1Khz. Công thức mel – frequency được tính như sau:

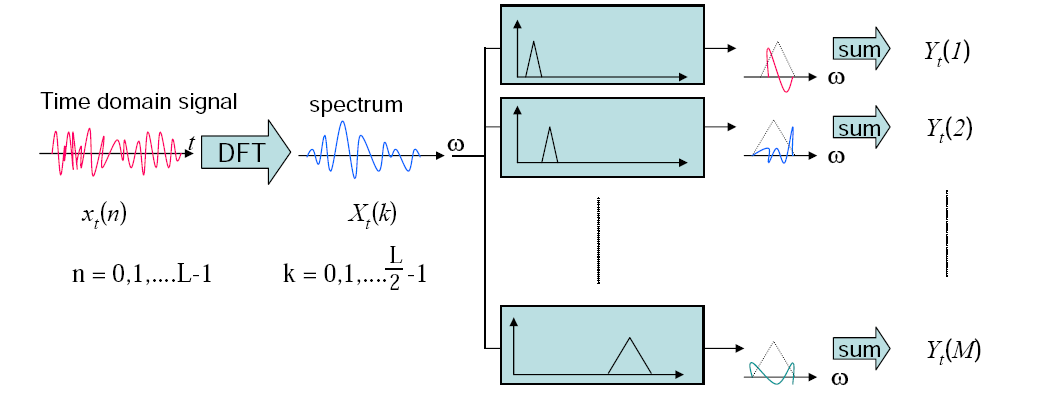
 (2.5)

Trên thực tế, để tính toán mel – spectrum thì người ta dùng những cửa sổ lọc – filterbank, được sắp xếp một cách đồng đều như Hình 2.10 bên dưới.



Hình 2.10. Thang tần số Mel

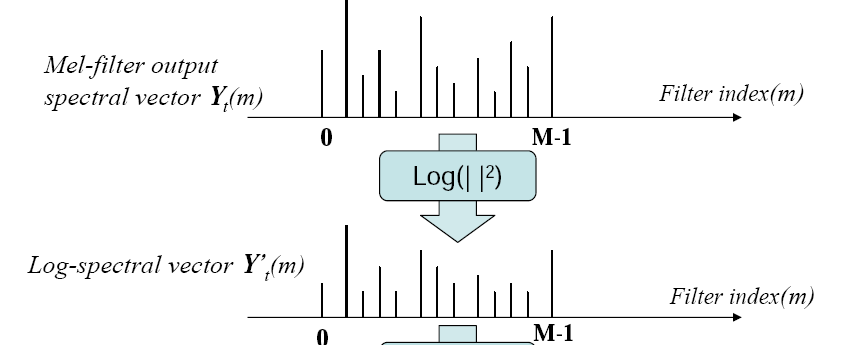
Đầu ra của mỗi bộ lọc là tổng của các phổ thành phần sau khi qua bộ lọc :



Hình 2.11. Tính toán sử dụng thang tần số Mel

#### 2.2.1.6. Logarit bình phương biên độ

Thao tác này thực chất là tính logarit của bình phương biên độ tín hiệu đầu ra sau khi qua bộ lọc tần số Mel. Nguyên nhân cần thực hiện thao tác này bởi tai người ít nhạy cảm với những sự thay đổi biên độ nhỏ tại biện độ cao so với tại biên độ thấp. Việc tính log bình phương biên độ sẽ làm cho tần số ước tính ít phụ thuộc hơn vào sự thay đổi đầu vào ( ví dụ như biến đổi năng lượng khi người nói tiến gần đến Micro hơn).



Hình 2.12. Tính log bình phương biên độ của tín hiệu đầu ra bộ lọc tần số Mel

#### 2.2.1.7. IDFT

Bước này chuyển đổi logarit của mel spectrum về trục thời gian, kết quả được gọi là "mel- frequency cepstrum coefficients (MFCC) ". MFCC của phổ tiếng nói là đặc trưng cho đặc tính phổ biên độ của tín hiệu từ các frame.

Phương pháp biến đổi ở đây dùng DCT ( discrite cosin transform ) thay vì IFFT, vì những đặc điểm sau:

– Tín hiệu là thực

– IFFT áp dụng cho tín hiệu là số phức, trong khi DCT là số thực.

– DCT có chức năng tương tự như IFFT, nhưng hiệu quả hơn do sử dụng số thực.

Có nhiều công thức tính DCT, có thể tham khảo thêm tại địa chỉ : <http://en.wikipedia.org/wiki/Discrete_cosine_transform>

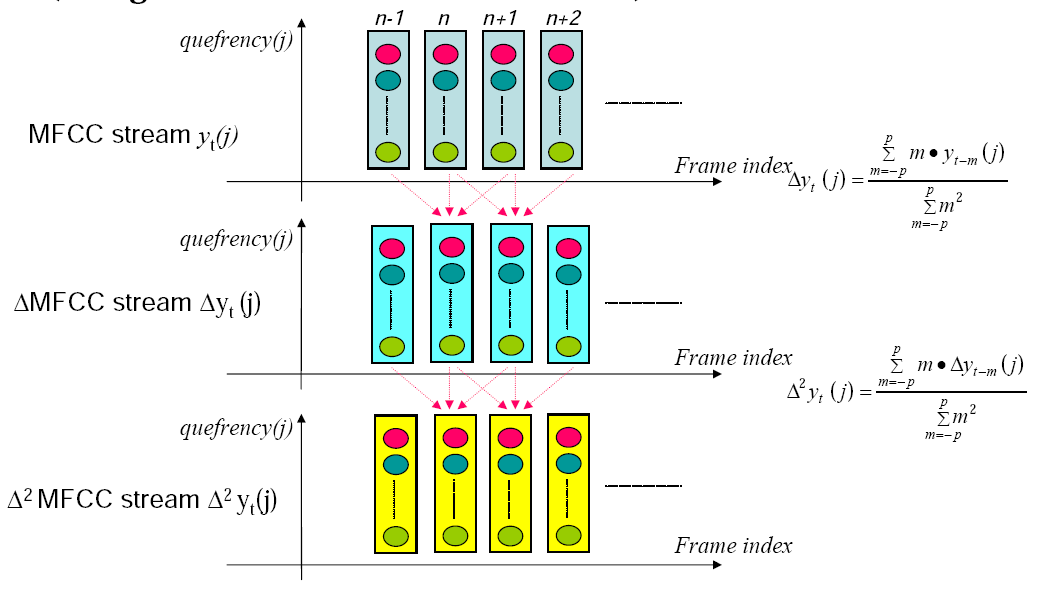
Công thức thường dùng nhất ở đây là công thức 2.

 n = 0,1,...,K - 1 (2.6)

#### 2.2.1.8. Các hệ số đặc trưng trong MFCC

Các hệ số Cepstral chứa các thông tin về tần số, không có thông tin về năng lượng. Ta cần phải thêm một hệ số đặc trưng của năng lượng :  .

Và chúng ta cũng biết tiếng nói là tín hiệu thay đổi theo thời gian. Vì thế, chúng ta cũng phải thêm hệ số để mô tả sự thay đổi của các hệ số đặc trưng theo thời gian. Các hệ số đó gọi là đặc trưng delta và đặc trưng double-delta .



Hình 2.13. Hệ số đặc trưng delta và doubledelta.

Như vậy, một vectơ đặc trưng MFCC điển hình có định dạng như sau :

- Độ dài mỗi khung (frame) : 25 ms

- Khoảng cách các frame : 10 ms

- Hệ số pre-emphasis : 0.97

- Số lượng các hệ số đặc trưng :

+ 12 MFCC ( hệ số thang tần số Mel)

+ 12 hệ số delta MFCC .

+ 12 hệ số delta bình phương MFCC.

+ 1 hệ số đặc trưng năng lượng.

+ 1 hệ số delta năng lượng.

+ 1 hệ số delta bình phương của năng lượng.

Như vậy, tổng cộng có 39 hệ số đặc trưng trong MFCC.

### 2.2.2. Huấn luyện và nhận dạng

#### 2.2.2.1. Mô hình Markov ẩn (Hidden Markov Model - HMM)

Như đề cập đến ở phần I. Các kĩ thuật chính cho đến nay dùng để nhận dạng giọng nói có thể kể đến như DTW, HMM và VQ. Trong phần này thì mô hình HMM được chọn vì nó phù hợp với quy mô của mô hình huấn luyện và nhận dạng (từ điển trung bình - khoảng 100 đến vài nghìn từ).

Mô hình Markov ẩn là [mô hình thống kê](http://vi.wikipedia.org/w/index.php?title=M%C3%B4_h%C3%ACnh_th%E1%BB%91ng_k%C3%AA&action=edit&redlink=1) trong đó hệ thống được mô hình hóa được cho là một [quá trình Markov](http://vi.wikipedia.org/wiki/Qu%C3%A1_tr%C3%ACnh_Markov) với các tham số không biết trước và nhiệm vụ là xác định các tham số ẩn từ các tham số [quan sát](http://vi.wikipedia.org/w/index.php?title=Quan_s%C3%A1t&action=edit&redlink=1) được, dựa trên sự thừa nhận này. Các tham số của mô hình được rút ra sau đó có thể sử dụng để thực hiện các phân tích kế tiếp.

Trong một mô hình Markov điển hình, trạng thái được quan sát trực tiếp bởi người quan sát, và vì vậy các xác suất chuyển tiếp trạng thái là các tham số duy nhất. Mô hình Markov ẩn thêm vào các đầu ra: mỗi trạng thái có xác suất phân bổ trên các biểu hiện đầu ra có thể. Vì vậy, nhìn vào dãy của các biểu hiện được sinh ra bởi HMM không trực tiếp chỉ ra dãy các trạng thái.

Mô hình Markov ẩn đặc trưng bởi các tham số sau :

- O = {o1, o2,..., oM} : Một bảng kí tự quan sát ở đầu ra. Các kí tự này đại diện cho đầu ra các đại lượng vật lí của hệ thống.

- Ω {1,2,..., N} : Tập hợp các trạng thái để biểu diễn một không gian trạng thái. st biểu diễn trạng thái tại thời điểm t.

- Aij = {aij} : Ma trận xác suất chuyển trạng thái. aij  là xác suất chuyển trạng thái từ i sang j.

- B = {bi(k)} : Ma trận xác suất đầu ra. bi(k) là xác suất đầu ra của kí tự ok khi trạng thái i được gia nhập.

- π = {πi} là phân bố trạng thái khởi tạo. πi = P (s0 = i)

- Chuỗi vecto quan sát X (X1, X2,..., XT).

Áp dụng mô hình HMM vào nhận dạng tiếng nói, ta phải giải quyết 3 vấn đề chính sau :

- Vấn đề đánh giá : Cung cấp  cho mô hình các tham số ( chuỗi quan sát O = (o1,o2, ..., oT) và một mô hình HMM (Φ = (A, B,π)) , tính xác suất của dãy đầu ra cụ thể (P (X | Φ). Giải bằng [thuật toán tiến trước](http://vi.wikipedia.org/w/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_ti%E1%BA%BFn_tr%C6%B0%E1%BB%9Bc&action=edit&redlink=1).

- Vấn đề giải mã : Cung cấp cho mô hình các tham số Φ và X , tìm dãy các trạng thái (ẩn) S (s0,s1,...,sT) có khả năng lớn nhất mà có thể sinh ra dãy đầu ra đã cung cấp. Giải bằng [thuật toán Viterbi](http://vi.wikipedia.org/w/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_Viterbi&action=edit&redlink=1).

- Vấn đề học : Cung cấp dãy đầu ra X và mô hình Φ , tìm tập hợp có khả năng nhất của chuyển tiếp trạng thái và các xác suất đầu ra. Giải bằng [thuật toán Baum-Welch](http://vi.wikipedia.org/w/index.php?title=Thu%E1%BA%ADt_to%C3%A1n_Baum-Welch&action=edit&redlink=1).

**Vấn đề đánh giá - Thuật toán tiến trước**

Để tính toán xác suất P(X| Φ) của chuỗi quan sát X=(X1,X2,...,XT) được cho bởi mô hình HMM Φ, cách thông thường là tính tổng xác suất của toàn bộ chuỗi trạng thái có thể :

   (2.7)

Ngoài ra để tính P(X| Φ), ta sẽ liệt kê toàn bộ các chuỗi trạng thái S có độ dài T, được phát sinh bởi chuỗi quan sát X, sau đó tính tổng toàn bộ các xác suất đó. Xác suất mỗi tập S chính là xác suất chuỗi trạng thái và xác suất đầu ra dọc theo tập S đó.

 (2.8)

trong đó  là  . Đối với chuỗi trạng thái tương tự S thì xác suất đầu ra dọc theo tập hợp có thể viết lại bằng cách áp dụng giả thiết đầu ra độc lập :

 (2.9)

Thay công thức 2.8 và 2.9 vào biểu thức 2.7 ta được :

 (2.10)

Tổng quát, ta thấy thuật toán tiến trước là thuật toán đệ qui gồm 3 bước chính :

- Khởi tạo :  (2.11)

- Đệ qui :   (2.12)

**-** Kết thúc :  (2.13)

**Vấn đề giải mã - Thuật toán Viterbi**

Việc tìm kiếm tập hợp tốt nhất liên quan rất lớn đến việc tìm kiếm trong hệ thống nhận dạng tiếng nói . Khi chuỗi trạng thái là ẩn trong mô hình HMM, tiêu chuẩn được sử dụng rộng rãi nhất chính là tìm chuỗi trạng thái có xác suất xảy ra cao nhất trong khi phát sinh ra chuỗi quan sát. Ngoài ra, chúng ta có thể tìm kiếm chuỗi trạng thái S (s0,s1,...,sT) có P(S, X| Φ) là lớn nhất. Vấn đề này tương tự với vấn đề đường đi tối ưu trong chương trình động. Một kĩ thuật cơ bản dựa trên một chương trình động được gọi là giải thuật Viterbi, có thể được sử dụng để tìm kiếm chuỗi trạng thái tốt nhất cho mô hình HMM. Giải thuật Viterbi được xem xét như là một dạng sửa đổi của giải thuật tiến trước. Giải thuật sẽ lưu trữ tập hợp tốt nhất. Trước tiên ta định nghĩa xác suất của tập hợp tốt nhất :  (2.14)

Trong đó : Vt(i) là xác suất tối ưu của chuỗi trạng thái tại thời điểm t, cái mà phát sinh là chuỗi quan sát  (đến thời điểm t) và kết thúc tại trạng thái i.

Tổng quát, thuật toán Viterbi cũng là một thuật toán hồi qui gồm 4 bước :

+ Khởi tạo :  1 ≤ n ≤ N (2.15)



+ Đệ qui :  2 ≤ t ≤ T ; 1 ≤ j ≤ N (2.16)

 2 ≤ t ≤ T ; 1 ≤ j ≤ N (2.17)

+ Kết thúc : The best score =  (2.18)

 (2.19)

+ Theo dõi trở lại :  t = T-1, T-2,..., 1 (2.20)

 is the best sequence. (2.21)

**Vấn đề ước lượng thông số mô hình HMM - Giải thuật Baum-Welch**

Việc ước lượng thông số mô hình Φ = (A, B,π) để mô tả chuỗi quan sát là rất quan trọng. Điều này là khâu khó nhất trong 3 vấn đề bởi ta không biết phương pháp phân tích để cho xác suất joint probability của dữ liệu là lớn nhất . Thay vì tính toán như vậy, vấn đề có thể giải quyết bằng giải thuật bước lặp Baum-Welch.

Trước tiên, ta cần định nghĩa một số khái niệm sau :

- Xác suất lùi  trong đó  là xác suất phát sinh vectơ quan sát  , khi mô hình HMM đang ở trạng thái i tại thời điểm t,  được tính bằng biểu thức qui nạp :

+ Khởi tạo :   (2.22)

+ Qui nạp :  (2.23)

 được tính đệ qui từ thời điểm t-1 sang t, còn  được tính đệ qui từ thời điểm t+1 sang t.

- là xác suất xảy ra sự chuyển dịch trạng thái từ i sang j tại thời điểm t :

  (2.24)



Ta có thể lặp lại các bước tính toán trên để nâng cao chất lượng của các thông số HMM Φ = (A, B,π) bằng cách tính toán để tỉ lệ likelihood  lớn nhất cho mỗi bước lặp. Gọi  là thông số mới được tạo ra từ thông số trong bước lặp trước. Dựa theo giải thuật EM (tạo ra các thông số mô hình bằng việc maximize hàm log-likelihood của dữ liệu chưa chỉnh và lặp lại việc maximize kì vọng toán học của hàm log-likelihood của dữ liệu hoàn chỉnh) thì quá trình maximize tương đương với việc maximize hàm Q sau :

 (2.25)

trong đó :  (2.26)

 (2.27)

Biểu thức được viết lại như sau :

 (2.28)

trong đó :   (2.29)

 (2.30)

Khi chia Q-function thành 3 phần độc lập,  lớn nhất có thể sẽ được tính toán bằng cách maximize từng thành phần riêng biệt, dựa vào các ràng buộc sau :

 i ;  j

Toàn bộ các phần trong biểu thức có thể tính như sau :

 trong đó  (2.31)

Bằng phép nhân Lagrange thì hàm trên đạt max tại 

Với biểu thức này ta sẽ có mô hình ước lượng như sau :

 (2.32)

 (2.33)

Xem xét 2 biểu thức trên, dễ thấy công thức (2.32) là tỉ lệ giữa số kì vọng của việc dịch chuyển trạng thái từ i sang j với số kì vọng của sự dịch chuyển từ trạng thái i. Đối với xác suất đầu ra ước lượng lại ở biểu thức (2.33) , thì tử số chính là số kì vọng của thời gian mà dữ liệu quan sát phát ra từ trạng thái j cùng với kí tự quan sát ok, mẫu số là giá trị kì vọng của thời gian mà dữ liệu quan sát phát ra từ trạng thái j.

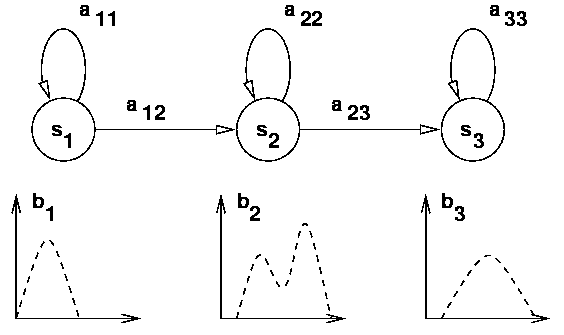
Tùy theo giải thuật , giải thuật Baum -Welch bảo đảm việc nâng cao likelihood ở mỗi bước lặp, và cuối cùng là likelihood sẽ hội tụ tại 1 giá trị cực đại địa phương. Mặc dù giải thuật được mô tả cơ bản dựa trên 1 chuỗi quan sát huấn luyện, nó có thể dễ dàng phát sinh ra nhiều chuỗi quan sát huấn luyện khác với giả thiết là các chuỗi độc lập với nhau. Việc huấn luyện mô hình HMM từ M chuỗi dữ liệu tương đương với việc tìm ra 1 vecto thông số HMM sao cho tỉ lệ joint - likelihood lớn nhất :  .

Hàm huấn luyện thể hiện trên giải thuật trên mỗi chuỗi quan sát để tính kì vọng trong biểu thức (2.32) và (2.33) . Biến đếm này trong tử số và mẫu số có thể được thêm vào thông qua M chuỗi dữ liệu. Cuối cùng tổng số thông số mô hình được tính tổng lại thành một. Điều này tạo ra mỗi bước lặp của sự ước lượng lại Baum-Welch, bước lặp sẽ tiếp tục đến khi hội tụ. Nó sẽ cho phép huấn luyện một mô hình HMM tốt trong hệ thống nhận dạng với một số lượng lớn dữ liệu. Ví dụ gọi  là dịch chuyển từ chuỗi dữ liệu thứ m và Tm là độ dài tương ứng , thì biểu thức được mở rộng như sau :

 (2.34)

#### 2.2.2.2. Mô hình Markov liên tục (Continuous density HMM)

Để đảm bảo cấu trúc ban đầu của tín hiệu nguyên thủy, người ta dùng mô hình Markov liên tục :



Hình 2.14. Mô hình Markov ẩn liên tục

Các hệ số B = {bi(k)} giờ được tính như sau :

 (2.35)

trong đó : M là số vecto đặc trưng cho trạng thái j của mô hình HMM

bik(x) là hàm mật độ Gaussian đơn với giá trị trung bình  , ma trận hiệp phương sai , của trạng thái i. Trong đó bik(x) được tính như sau :

 với  (2.36)

cik là trọng số cho thành phần trộn thứ k (xác suất xuất hiện vectơ thứ k trong trạng thái j) trong đó 

Mô hình HMM liên tục là một mô hình tốt cho lí thuyết HMM. Do khối lượng tính toán khá lớn nên đặt hệ thống nhận dạng trên 1 máy chủ (server) là phù hợp. Mỗi trạng thái trong HMM liên tục cần có M vector riêng đặc trưng. Mỗi vector có thêm 1 ma trận hội tụ tương ứng với nó. Vector có số chiều càng lớn thì kích thước ma trận hội tụ càng lớn, tính toán càng phức tạp.

#### 2.2.2.3. Mô hình ngôn ngữ

Một phần quan trọng của module nhận dạng là mô hình ngôn ngữ. Trong mô hình ngôn ngữ mỗi chuỗi từ W được gắn với một phân bố xác suất P(W), phản ánh tần suất của chuỗi từ đó xuất hiện như 1 câu. Mô hình ngôn ngữ hay được sử dụng để nhận dạng tiếng nói là mô hình N-gram. Ý tưởng mô hình như sau :

 (2.37)

Nhìn vào biểu thức ta thấy từ wi chỉ phụ thuộc vào toàn bộ chuỗi từ đã xuất hiện trong quá khứ. Đối với từ vựng có kích thước là v, và có vi-1 các từ đã xuất hiện trước đó để xác định  thì vi giá trị phải được ước lượng. Tuy nhiên trong thực tế thì không thể xác định được chính xác giá trị i. Một giải pháp đưa ra là chỉ phụ thuộc vào vài lớp tương đương. Các lớp tương đương này đơn giản dựa trên một vài từ xuất hiện trước đó. Đó là mô hình ngôn ngư N-gram. Nếu 1 từ chỉ phụ thuộc vào 2 từ trước nó thì ta có mô hình tri-gram  . Tương tự ta cũng có mô hình uni-gram  và bi-gram  .

Trong mô hình tri-gram, xác suất xuất hiện của từ phụ thuộc vào 2 từ trước đó. Mô hình tri-gram có thể ước lượng bằng cách quan sát xem số lần cặp C(wi-2,wi-1) và

C(wi-2,wi-1,wi) xuất hiện bao nhiêu lần và sau đó tính như sau :

 (2.38)

Mô hình N-gram có cấu trúc khá đơn giản và có thể luyện trên nhiều dữ liệu, và rất tốt khi muốn mô hình hóa trong hệ thống nhận dạng có nhiễu.

# CHƯƠNG 3 : XÂY DỰNG HỆ THỐNG NHẬN DẠNG TRÊN SPHINX4

## 3.1. Giới thiệu về Sphinx4

### 3.1.1. Tổng quan về Sphinx4

Sphinx4 là công cụ nhận dạng tiếng nói liên tục với bộ từ vựng lớn, không phụ thuộc vào người nói do trường đại học Carnegie Mellon University phát triển. Nó là tập hợp của các công cụ miễn phí và mã nguồn mở cho phép nhà nghiên cứu và phát triển xây dựng các hệ thống nhận dạng cho riêng mình. Bộ công cụ này gồm có :

1. Sphinx4

2.PocketSphinx

3.SphinxBase

4.SphinxTrain

5.Cmuclmtk

Trong đó Sphinx4 và PocketSphinx đóng vai trò các bộ giải mã (decoder) trong hệ thống nhận dạng. SphinxBase là thư viện các hàm chức năng cơ bản cần được cài đặt khi muốn sử dụng bất kì bộ giải mã nào trong hệ thống. SphinxTrain là công cụ huấn luyện mô hình âm học cho hệ thống. Cmuclmtk là bộ công cụ để tạo ra mô hình ngôn ngữ.

Sphinx4 là công cụ viết bằng ngôn ngữ lập trình Java, được phát triển trên nền tảng Sphinx2, Sphinx3. Nó cung cấp mô hình linh hoạt hơn trong việc phát triển hệ thống nhận dạng tiếng nói. Cụ thể, mục tiêu phát triển của Sphinx4 là :

- phát triển một bộ huấn luyện mới (về mô hình âm thanh).

- thích ứng với người nói.

- cải thiện về quản lí cấu hình.

### 3.1.2.Những công cụ cần trang bị

Danh sách các gói cần thiết cho đào tạo cơ sở dữ liệu :

- sphinxbase - 0.8

- Sphinxtrain - 0.8

- Pocketsphinx

- cmuclmtk - 0.7

- sphinx4- 1.0 beta6

- ActivePerl (nếu thực hiện trên Windows, Linux không cần thiết)

- ActivePython (thực hiện trên Windows)

Tất cả các gói trên có thể tìm thấy và download tại địa chỉ :

<http://cmusphinx.sourceforge.net/wiki/download/>

## 3.2. Xây dựng mô hình nhận dạng trên Sphinx4

Mô hình được chọn để xây dựng hệ thống nhận dạng là mô hình Markov ẩn liên tục (Continous Density HMM). Cấu trúc mô hình đã trình bày cụ thể tại phần 2.

Phương pháp trích chọn đặc trưng được sử dụng là MFCC (Mel-scale Frequency Cepstral Coefficients). Phương pháp này cũng đã được trình bày tại phần 2.

### 3.2.1. Chuẩn bị cơ sở dữ liệu

**Tạo dữ liệu cho mô hình âm học**

Để tiến hành xây dựng cơ sở dữ liệu cho các file wav phục vụ việc nhận dạng, ta sẽ tạo ra 1 văn bản Corpus chứa toàn bộ các câu xoay quanh các chủ đề phục vụ cho ứng dụng . Nội dung mỗi chủ đề sẽ được ghi vào các file riêng biệt nhau.

- Tin nhắn : các câu lệnh, thao tác người dùng hay thực hiện.

**Ví dụ** : hãy nhắn tin đến tùng

hãy nhắn tin đến bảo

gửi tin nhắn tới khoa

gửi tin nhắn tới số điện thoại của khoa với nội dung là

ngày mai đi chơi không

gửi tin nhắn đến hiển...

Danh sách đầy đủ các câu thoại về chủ đề tin nhắn chứa khoảng gần 100 câu, được ghi lại trong file message.txt

- Tra cứu thông tin trên Internet : thời tiết, giá vàng, ngoại tệ, đô la Mỹ...

**Ví dụ** : giá vàng hôm nay thế nào

cho tôi xem tỉ giá đô la hôm nay

hôm nay thị truờng thế nào

giá vàng thế nào

thị truờng vàng thế nào

cho xem thông tin ngoại tệ

cho tôi biết thời tiết hà nội hôm nay

ngày mai hà nội có mưa không

hiện tại thời tiết bao nhiêu độ

thời tiết hôm nay thế nào

Danh sách đầy đủ các câu thoại về chủ đề tra cứu thông tin online chứa khoảng hơn 150 câu, được ghi lại trong file information.txt và weather.txt.

- Ngoài ra, để phục vụ hoạt động cho thiết bị, cơ sở dữ liệu còn có các câu nói về hội thoại thông thường, các lệnh thường sử dụng, một số địa điểm (thành phố), số đếm...

**Ví dụ** : tạm biệt nhé

chào nhé

hẹn gặp lại

thôi nhé

có

không

hủy

thoát

chấp nhận

một

hai ...

Các câu này được liệt kê đầy đủ tại các file conversation.txt, number.txt, command.txt. Tổng số khoảng 150 câu.

Sau khi đã xây dựng toàn bộ kịch bản các câu nói hay sử dụng, ta tiến hành thu âm. Thiết bị dùng để thu âm thường chọn là các smartphone, tab... chạy hệ điều hành Android. Ta sẽ xây dựng một chương trình thu âm Recording thực hiện công việc này.

Chương trình này khi tiến hành thu âm sẽ tạo ra 1 file âm thanh và 1 file text đi kèm ghi lại thông tin về : họ tên người thu, tuổi , vùng miền, giới tính, chủ đề, thông tin thiết bị, tần số lấy mẫu (mặc định 16KHz), mã hóa (16 bits), kênh đơn (mono), nội dung câu nói. Hai file này được đặt tên giống nhau để dễ dàng cho việc quản lí sau này.

**Ví dụ** : thông tin ghi trong file text đi kèm file âm thanh

Nguyen Thanh Tung

23

male

Hà Nội

command

samsung, GT-P1000, Android version 2.3.3 - GINGERBREAD\_MR1

16000

16

1

signed

vào ...

Bảng dưới đây thống kê chi tiết về dữ liệu ghi âm :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Tin nhắn  (câu) | Hội thoại  (câu) | Thông tin  (câu) | Thời tiết  (câu) | Lệnh  (câu) | Bổ sung  (câu) |
| Đào Kim Anh - F | 84 | - | 32 | 67 | 38 | - |
| N.Hạnh Nguyên- F | 81 | 57 | 29 | 67 | 37 | 112 |
| N.Thùy Dung - F | 82 | 52 | 31 | 65 | 37 | 110 |
| N.Thanh Nga - F | 89 | 56 | 27 | 66 | - | - |
| N. Thị Nga - F | 88 | - | - | - | 37 | 110 |
| Vũ Thị Thơm - F | 81 | 57 | 31 | 65 | - | 110 |
| A - M | 64 | 56 | 29 | 65 | - | 112 |
| N.Văn Bảo - M | 85 | 56 | 30 | 68 | - | 111 |
| N. Hữu Hiển - M | 75 | 52 | 29 | 65 | 37 | 110 |
| N. Thanh Tùng -M | 71 | 55 | 27 | 63 | 38 | 112 |
| Vũ Thế Mạnh - M | - | - | - | 58 | - | - |
| Trần Thành Vinh-M | 79 | 56 | 28 | 58 | - | 110 |

### 3.2.2. Xây dựng thư viện cơ sở dữ liệu và các mô hình

Ta tổ chức thư viện theo cấu trúc sau :

- etc : + an4.dic : file chứa tất cả các từ có mặt trong Corpus và phiên âm tiếng Việt

+ an4.lm.DMP : file mô hình ngôn ngữ .

+an4.filler : Chứa thông tin các khoảng lặng đầu, giữa hay cuối câu

+an4.phone : chứa danh sách tất cả các phone chứa trong từ điển phiên âm

+an4\_train.fileids : đường dẫn tới các file âm thanh dùng để đào tạo +an4\_train.transcription : chứa nội dung các file wav và tên file tương ứng.

Chú ý : tên file trong fileids và transcription phải giống hệt nhau.

+an4\_test.fileids : đường dẫn tới các file âm thanh để test.

+an4\_test.transcription : chứa nội dung các file âm thanh đó.

+feat.params: chứa các thông số trong việc huấn luyện

+sphinx\_train.cfg : chứa thông tin của bộ huấn luyện

- wav : thư mục chứa file âm thanh, sắp xếp thành từng thư mục con theo chủ đề

+chủ đề 1 :

+chủ đề 2 : ...

Một số chú ý trong quá trình xây dựng thư viện chứa dữ liệu :

+ File từ điển phiên âm ( an4.dic ) : mỗi từ nên được xếp thành 1 dòng, theo sau là phiên âm tiếng Việt của từ đó. Từ có thể chia tách thành từ đơn hoặc từ ghép. Đối với từ ghép, cần có dấu \_ để nối giữa các từ đơn.

Do Sphinxtrain không hỗ trợ các kí tự đặc biệt (như các dấu và các thanh trong tiếng Việt) nên ta cần có một bảng mã để chuyển đổi các tất cả các kí tự từ định dạng UTF-8 sang ASCII. Bảng mã này được ghi trong file 2ascii.def và tone.def.

Ta có ví dụ dạng từ trong từ điển :

ba5ch\_khoa b eh5 k0 kh uh a1

ba6n b a6 n0

bao\_nhieeu b a1 uf nh iee1 uf

ba4o b a4 uf

ba4y b aw4 if

+ File từ điển phụ (an4.filler) : chứa kí hiệu của các khoảng lặng, ngắt câu

<s> SIL

</s> SIL

<sil> SIL

+ File phoneset (an4.phone) : chứa tất cả các âm của từ điển phiên âm, không có âm nào được trùng nhau.

**Ví dụ** : a1

a2

aa1

aa2

e5

e6

ee1

ee2...

+ File mô hình ngôn ngữ (an4.lm.DMP) : chứa mô hình ngôn ngữ. Tất cả các công cụ nằm trong thư mục cmuclmtk . File này được xây dựng qua một số bước sau :

* Tạo một văn bản chứa tất cả các kịch bản có thể xảy ra. Các câu được bắt đầu với kí hiệu <s> và kết thúc bởi </s>. Tên file có thể như weather.txt.

**Ví dụ :** <s> cho tozi biee5t thow2i\_tiee5t ha2\_noz6i hozm\_nay </s>

<s> nga2y mai ha2\_noz6i co5 muwa khozng </s>

<s> hiee6n ta6i thow2i\_tiee5t bao\_nhieeu ddoz6 </s>

* Tạo ra các tập tin từ vựng. Đây là một danh sách của tất cả các từ trong tập tin:

text2wfreq <weather.txt | wfreq2vocab> weather.tmp.vocab

* Tạo ra mô hình định dạng ngôn ngữ arpa :

% text2idngram -vocab weather.vocab -idngram weather.idngram < weather.txt

% idngram2lm -vocab\_type 0 -idngram weather.idngram -vocab \

weather.vocab -arpa weather.arpa

* Chuyển đổi định dạng ngôn ngữ DMP :

sphinx\_lm\_convert -i weather.arpa -o weather.lm.DMP

Sau khi đã có được thư mục etc chứa đầy đủ các file cần thiết như trên, ta tiến hành chỉnh sửa một số thông tin trong filwe cấu hình việc nhận dạng (sphinx\_train.cfg) :

* Định dạng file âm thanh :

$CFG\_WAVFILES\_DIR = "$CFG\_BASE\_DIR/wav";

$CFG\_WAVFILE\_EXTENSION = 'wav';

$CFG\_WAVFILE\_TYPE = 'mswav'; # one of nist, mswav, raw

* Chọn mô hình .cont. (liên tục) do sử dụng công cụ sphinx4

$CFG\_HMM\_TYPE = '.cont.'; # Sphinx 4, Pocketsphinx

#$CFG\_HMM\_TYPE = '.semi.'; # PocketSphinx

#$CFG\_HMM\_TYPE = '.ptm.'; # PocketSphinx

* Số lượng các trạng thái để tạo ra cây quyết định :

# Number of tied states (senones) to create in decision-tree clustering

$CFG\_N\_TIED\_STATES = 1000;

Dữ liệu đã được chuẩn bị đầy đủ, ta tiến hành đào tạo cơ sở dữ liệu :

* Đầu tiên, ta chuyển đến thư mục chứa dữ liệu :

cd an4

* Trên Windows, ta chạy lệnh :

python ../sphinxtrain/scripts/sphinxtrain run

* Quá trình đào tạo thực hiện lần lượt các bước, ví dụ :

Baum welch starting for 2 Gaussian(s), iteration: 3 (1 of 1)

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Normalization for iteration: 3

Current Overall Likelihood Per Frame = 30.6558644286942

Convergence Ratio = 0.633864444461992

Baum welch starting for 2 Gaussian(s), iteration: 4 (1 of 1)

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Normalization for iteration: 4

Khi quá trình huấn luyện kết thúc, kết quả mô hình âm học là thư mục an4.ci\_cont (Context-Independent) hoặc có thể sử dụng an4.cd\_cont\_3000 (Context-Dependent).

Kết quả được lưu trong thư mục model\_parameters. Trong phạm vi ứng dụng, do lượng từ vựng không lớn nên ta có thể chọn mô hình âm học bối cảnh độc lập (CI).

Nó gồm các file sau :

+ mdef

+ means

+ variances

+ mixture\_weights

+transition\_matrices

+noisedict

+feat.params

Như vậy, quá trình đào tạo cơ sở dữ liệu đã hoàn tất. Ta sẽ sử dụng các file và thư mục sau như là kết quả để xây dựng chương trình nhận dạng :

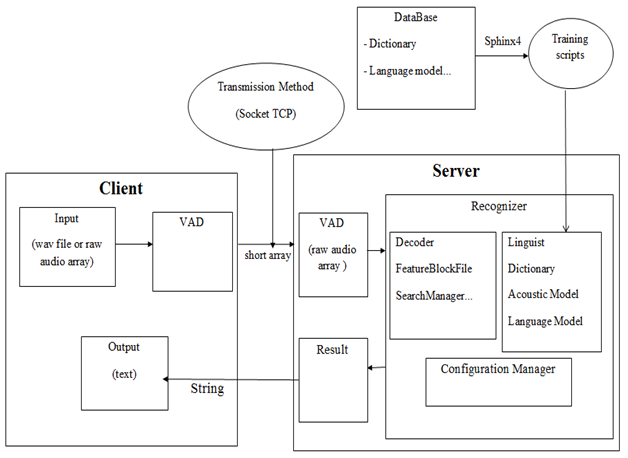
+ an4.dic

+ an4.lm.DMP

+ an4.ci\_cont

## 3.3.Xây dựng chương trình nhận dạng

### 3.3.1. Sơ đồ khối chương trình



Hình 3.1.Sơ đồ khối chương trình nhận dạng

### 3.3.2. Mô tả hoạt động

Chú ý : Module truyền nhận trong chương trình này được thực hiện bởi bạn Nguyễn Hữu Hiển. Hai module nhận dạng và truyền nhận kết hợp với nhau tạo nên server nhận dạng đầy đủ.

Môi trường truyền là mạng: Internet ( mạng dây, 3G ), LAN ( mạng dây, wifi )

Kiểu kết nối: Multi Client – Server

- Client gửi cho server một mảng kiểu short là tín hiệu âm thanh đã được lấy mẫu

- Server nhận dạng và trả lại kết quả dạng text

Phương thức truyền : Socket TCP

Cách thức hoạt động của cả hệ thống Multi client – server nhận dạng tiếng nói được mô tả như sau :

Sau khi thu âm ở client, client sẽ tự động ngắt thu âm và mở kết nối socket với server, số mẫu tiếng nói ban đầu được ép kiểu từ int thành short và được lưu vào một mảng short và được gửi toàn bộ lên server để tăng tốc độ gửi.

Server khi chạy luôn mở socket để lắng nghe và kết nối khi có một client muốn kết nối. Với mỗi một client đến server tạo một thread riêng để đáp ứng với client đó :

- Nhận mảng short client gửi lên

- Chuyển mảng short thành mảng int

- VAD mảng int để bỏ đi khoảng lặng và những xung âm thanh ngắn

- Ghi thành file âm thanh định dạng wav làm đầu vào cho khối nhận dạng

- Nhận dạng file âm thanh

- Trả ra kết quả là một string

- Gửi lại string xuống client

- Đóng kết nối với client tương ứng và kết thúc thread

Vì mỗi thread trên server được đặc trưng bởi 1 socket riêng tương ứng với client nên server luôn đảm bảo việc nhận đúng và trả đúng client .

Server khi khởi động sẽ tạo song song 3 bộ regconizer tương ứng là 3 bộ decoder, trong mỗi bộ regconizer đó đều có mô đun VAD và mô đun nhận dạng riếng sử dụng Sphinx 4. Ba yêu cầu đầu tiên sẽ lần lượt được phân vào 3 bộ recognizer theo thứ tự 1, 2, 3. Mỗi khi một regconizer nhận yêu cầu từ một client nó sẽ bật cờ báo bận, sau khi nhận dạng xong sẽ bật cờ báo rỗi, với một recognizer sẽ có một biến thời gian đi kèm để đo thời gian rảnh rỗi. Kể từ yêu cầu thứ 4 đến từ client, dữ liệu âm thanh nhận được từ client sẽ được phân vào bộ regconizer đang rảnh rỗi và có thời gian rảnh rỗi lớn nhất.

Quá trình nhận dạng qua một số bước chính : sau khi qua bộ VAD để bỏ các khoảng lặng, các file âm thanh sẽ được trích chọn đặc trưng MFCC để tạo ra các hệ số đặc trưng như đã mô tả ở các phần trước, rồi đưa đến bộ giải mã (decoder). Các hệ số đặc trưng đưa vào bộ giải mã để ánh xạ đến mô hình ngôn ngữ (file dict và DMP) và mô hình âm học (folder an4.ci\_cont) đã thu được sau quá trình huấn luyện. Bộ giải mã sẽ hoạt động và đưa ra kết quả nhận dạng.

## 3.4. Kết quả:

### 3.4.1.Đánh giá tốc độ gửi nhận client-server

Tốc độ gửi nhận client – server phụ thuộc vào :

* Môi trường truyền : Lan, 3g, Internet
* Dung lượng dữ liệu: dữ liệu càng lớn thì càng mất nhiều thời gian để gửi.

Dưới đây là bảng đánh giá kết quả :

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lan(mica)** | **Lan(Connectify)** | **Internet** |
| Không nói (2s thu âm) | 800 ms | 600 ms | 600 ms |
| 10 từ (5s thu âm) | 1200 ms | 1100 ms | 1200 ms |
| 22 từ (9s thu âm) | 2000 ms | 1800 ms | 3000 ms |

### 3.4.2. Tốc độ nhận dạng

Các yếu tố ảnh hưởng đến thời gian nhận dạng bao gồm :

- Cấu hình server: CPU, RAM,…

- Độ dài của câu nói nhận dạng

Dưới đây là bảng đánh giá kết quả :

|  |  |  |
| --- | --- | --- |
|  | Server Ubuntu  MICA ( RAM 32 Gb) | Server Laptop  (core i5 - RAM 4 Gb) |
| Không nói (2s thu âm) | 8 ms | 7 ms |
| 10 từ ( 5s thu âm) | 500 ms | 800 ms |
| 22 từ ( 9s thu âm) | 1800 ms | 2000 ms |

Qua bảng nhận dạng, ta thấy tốc độ nhận dạng trên chấp nhận được với ứng dụng online. Tổng thời gian chờ đợi của người dùng với 1 câu nói thông thường (khoảng 10 từ) tính từ lúc thu âm xong đến khi nhận dược kết quả khoảng 2s.

### 3.4.3. Tỉ lệ nhận dạng

Huấn luyện mô hình với tập dữ liệu gồm 12 giọng nói khác nhau, tất cả là giọng miền Bắc . Trong đó có 6 giọng nam và 6 giọng nữ, độ tuổi tập trung trong khoảng 20 -25 tuổi (ngoài ra có 1 giọng nam 15 tuổi và 1 giọng nữ 45 tuổi). Tổng số câu dùng để huấn luyện mô hình là 3822 câu, chia làm 6 chủ đề, từ điển bao gồm 251 từ (cả từ đơn và từ ghép). Tập dữ liệu chia làm 2 phần :

- Phần 1 : gồm 3664 câu dùng để huấn luyện.

- Phần 2 : gồm 158 câu dùng để kiểm tra.

Để đánh giá sai số nhận dạng, ta xây dựng 1 chương trình đánh giá như sau : người nói nói theo 1 kịch bản cho trước (số lượng nhất định, khoảng câu). Câu nói sẽ được nhận dạng trên server, trả về kết quả dạng text. Kết quả này sẽ được so sánh với dòng text kịch bản ban đầu. Sai số nhận dạng được tính theo công thức :



trong đó : Nword  là tổng số từ trong văn bản

Ins là số từ bị thêm vào do sai số

Del là số từ bị mất khi nhận dạng

Sub là số từ nhận dạng sai.

Thông qua kết quả tỉ lệ sai số nhận dạng, ta có thể đánh giá được tỉ lệ nhận dạng chính xác của mô hình : Acc = 100 - WER (%).

Ta lấy tập câu nói để kiểm tra gồm 50 câu (chứa cả câu ngắn chỉ có 1 từ đơn, câu trung bình từ 3 đến 7 từ; câu dài chứa khoảng 20 từ). Các câu này được lấy ở tất cả các chủ đề đã xây dựng cơ sở dữ liệu (tin nhắn, thời tiết, truy vấn, lệnh, số,... )

Sau đây là bảng tổng kết đánh giá :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Ins | Del | Sub | Nword | WER (%) | Acc (%) |
| Nguyễn Văn Hiếu | 0 | 20 | 20 | 219 | 18,3 | 81.7 |
| Trần Mạnh Tú | 1 | 17 | 35 | 219 | 24,2 | 75,8 |
| Nguyễn Quang Toàn | 7 | 10 | 36 | 219 | 24,2 | 75,8 |
| Vũ Thế Mạnh | 8 | 5 | 16 | 219 | 17,8 | 82,2 |
| Nguyễn Thanh Tùng | 0 | 8 | 19 | 219 | 12,3 | 87,7 |

Qua bảng tổng kết, ta nhận thấy sai số nhận dạng của hệ thống nằm trong khoảng từ 12% đến 25% , nói cách khác độ chính xác nhận dạng khoảng 75% đến 88%. Đây là kết quả chấp nhận được đối với một mô hình nhận dạng trung bình (từ vựng khoảng 300 từ).



**3.4.3. Đánh giá kết quả**

Kết quả nhận dạng đạt kết quả tốt khi thực hiện trong môi trường có nhiễu nhỏ (tiếng ồn, quạt...) vì trong quá trình thu thập cơ sở dữ liệu cho huấn luyện , môi trường cũng có những nhiễu nhỏ như vậy, không có nhiều tiếng nói khác. Tuy nhiên, mô hình nhận dạng còn có nhiều lỗi như nhận dạng sai, thêm từ với câu ngắn có 1 âm ( như số : một, ba , bốn ; lệnh : mở, dừng, ...) hoặc với những câu, từ xuất hiện với tần xuất nhỏ (như : cám ơn bạn, bạn khỏe không, hay đấy ... ). Các nguyên nhân chính dẫn đến sai số trong nhận dạng :

- Dữ liệu huấn luyện chưa đủ lớn, để việc nhận dạng có thể không phụ thuộc vào người nói. Những giọng nói đặc biệt (miền Trung, ngọng, tốc độ nói nhanh...) sẽ có sai số nhận dạng cao hơn những giọng nói trong cơ sở dữ liệu đã đào tạo.

- Những câu ngắn, có 1 âm duy nhất có tỉ lệ sai cao hơn những câu có nhiều âm tiết, do mô hình ngôn ngữ ta sử dụng là mô hình tri-gram (xác suất xuất hiện của 1 từ chỉ phụ thuộc và 2 từ trước nó). Những câu có 1 hay 2 âm tiết thì mô hình không thể xác định được xác suất xuất hiện của 2 từ đứng trước được.

- Nhiễu từ môi trường, đặc biệt là tiếng nói khác ảnh hưởng lớn đến chất lượng nhận dạng. Chất lượng mạng cũng ảnh hưởng một phần.

- Việc sử dụng các thiết bị khác nhau, các loại micro khác nhau cũng là một yếu tố ảnh hưởng sai số nhận dạng.

# CHƯƠNG 4 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Với các kết quả đã đạt được, đồ án đã tích hợp được hệ thống nhận dạng tiếng nói tiếng Việt lên server và cho phép nhận dạng tiếng nói trực tuyến . Tập mẫu dùng để huấn luyện đã khá lớn (khoảng 3800 câu, từ điển khoảng 250 từ) nhưng vẫn chưa đủ để thích hợp với việc nhận dạng độc lập với người nói. Để mở rộng qui mô và cải thiện chất lượng nhận dạng, người xây dựng hệ thống cần phải tạo ra thêm nhiều kịch bản trong mô hình ngôn ngữ và thêm số lượng mẫu thu âm phục vụ mô hình âm học.

**Hướng phát triển** :

- Phát triển thêm những chức năng khác vào ứng dụng. Ví dụ : quay số điện thoại bằng tiếng nói , hỏi đường trên google map, tra cứu thông tin trên Internet...

- Mở rộng bộ cơ sở dữ liệu với số lượng mẫu thu âm và số người nói lớn hơn. Mở rộng mô hình ngôn ngữ, từ điển phiên âm, thêm nhiều chủ đề để huấn luyện .

- Phát triển thêm mô đun tiền xử lí để lọc nhiễu , lọc bỏ tiếng vọng để nâng cao chất lượng của câu nói.

# Tài liệu tham khảo

[1]LSA 352 Speech Recognition and Synthesis Lec5 - Dan Jurafsky.

[2]LSA 352 Speech Recognition and Synthesis Lec6 - Dan Jurafsky.

[3]Hidden Markov Models and the Baum-Welch Algorithm, Lioyd R.Welch (2003)

[4] Introduction Basics of signal processing-Pr. Eric Castelli-January 2013

[5] Analysis of Speech signal – Pr.Eric Castelli-February, 2013

Website tham khảo :

<http://www.cslu.ogi.edu/tutordemos/nnet_training/tutorial.html>

<http://en.wikipedia.org/wiki/Speech_recognition>

<http://www.clear.rice.edu/elec532/PROJECTS98/speech/cepstrum/cepstrum.html>

<http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/>

<http://cmusphinx.sourceforge.net/wiki/tutorialam>