**­­­­­­­­­­ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÂM DUY THẮNG**

**NGUYỄN MINH CHÁNH**

**KHÓA LUẬN TỐT NGHIỆP**

**ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH**

**BẰNG GIỌNG NÓI THÔNG QUA ỨNG DỤNG ANDROID**

**KỸ SƯ NGÀNH KỸ THUẬT MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2017**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**LÂM DUY THẮNG - 12520704**

**NGUYỄN MINH CHÁNH – 12520544**

**KHÓA LUẬN TỐT NGHIỆP**

**ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH**

**BẰNG GIỌNG NÓI THÔNG QUA ỨNG DỤNG ANDROID**

**KỸ SƯ NGÀNH KỸ THUẬT MÁY TÍNH**

**GIẢNG VIÊN HƯỚNG DẪN**

**PHAN ĐÌNH DUY**

**TP. HỒ CHÍ MINH, 2017**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số ……...........… ngày ………………….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

* 1. …………………………………………. – Chủ tịch.
  2. …………………………………………. – Thư ký.
  3. …………………………………………. – Ủy viên.
  4. …………………………………………. – Ủy viên.

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA  TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc Lập - Tự Do - Hạnh Phúc** |
|  | *TP. HCM, ngày…..tháng…..năm 2017* |

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP**

**(CỦA CÁN BỘ HƯỚNG DẪN)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tên khóa luận:** | | | | | | |
| **ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH**  **BẰNG GIỌNG NÓI THÔNG QUA ỨNG DỤNG ANDROID** | | | | | | |
| **Nhóm SV thực hiện:** | | | | **Cán bộ hướng dẫn:** | | |
| Lâm Duy Thắng | |  | | | Ths.Phan Đình Duy | |
| Nguyễn Minh Chánh | |  | | |  | |
| **Đánh giá Khóa luận**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:       1. Về nội dung nghiên cứu:        1. Về chương trình ứng dụng:        1. Về thái độ làm việc của sinh viên:       **Đánh giá chung:**      **Điểm từng sinh viên:**  Lâm Duy Thắng**: ………../10**  Nguyễn Minh Chánh**: ………../10** | | | | | | | | |
|  | | | | | **Người nhận xét**  (Ký tên và ghi rõ họ tên)  ………………………………… | | |
| ĐẠI HỌC QUỐC GIA  TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN** | | | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc Lập - Tự Do - Hạnh Phúc** | | | | |
|  | | | *TP. HCM, ngày…..tháng…..năm 2017* | | | | |

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP**

**(CỦA CÁN BỘ PHẢN BIỆN)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên khóa luận:** | | | | | |
| **ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH**  **BẰNG GIỌNG NÓI THÔNG QUA ỨNG DỤNG ANDROID** | | | | | |
| **Nhóm SV thực hiện:** | | **Cán bộ phản biện:** | | | |
| Lâm Duy Thắng |  | | | ....................................... | |
| Nguyễn Minh Chánh |  | | | ....................................... | |
| **Đánh giá Khóa luận**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:       1. Về nội dung nghiên cứu:        1. Về chương trình ứng dụng:        1. Về thái độ làm việc của sinh viên:       **Đánh giá chung:**      **Điểm từng sinh viên:**  Lâm Duy Thắng**: ………../10**  Nguyễn Minh Chánh**: ………../10** | | | | | |
|  | | | **Người nhận xét**  (Ký tên và ghi rõ họ tên)  ………………………………… | |

**LỜI CẢM ƠN**

Lời đầu tiên, cho phép nhóm em xin gửi lời cảm ơn sâu sắc nhất đến các thầy, cô giáo của trường Đại Học Công Nghệ Thông Tin Thành phố Hồ Chí Minh, những người đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức quý báu và cần thiết trong suốt những năm học tập tại trường.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới Ths Phan Đình Duy, người đã quan tâm, động viên, tận tình hướng dẫn và giúp đỡ chúng em trong suốt quá trình chúng em làm khóa luận tốt nghiệp, để chúng em có thể hoàn thanh tốt khóa luận của mình.

Và cuối cùng, cảm ơn gia đình đã tạo điều kiện cho chúng em hoàn thành việc của mình trong suốt khoảng thời gian học tập trên giảng đường và có điều kiện thuận lợi nhất để nhóm em hoàn thành tốt khóa luận này.

***Xin chân thành cảm ơn.***

Nhóm sinh viên thực hiện:

***Lâm Duy Thắng***

***Nguyễn Minh Chánh***

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA  TP. HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC**  **CÔNG NGHỆ THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc Lập - Tự Do - Hạnh Phúc** |

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỀ TÀI: “ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH**  **BẰNG GIỌNG NÓI THÔNG QUA ỨNG DỤNG ANDROID”** | |
| **Cán bộ hướng dẫn: Phan Đình Duy** | |
| **Thời gian thực hiện:** Từ ngày 16/1/2017 đến ngày 1/7/2017 | |
| **Sinh viên thực hiện:**   1. Lâm Duy Thắng – 12520704 2. Nguyễn Minh Chánh – 12520544. | |
| **Nội dung đề tài:***(Mô tả chi tiết mục tiêu, phạm vi, đối tượng, phương pháp thực hiện, kết quả mong đợi của đề tài)*   * Xây dựng một hệ thống có thể điều khiển các thiệt bị điện trong gia đình bằng chính giọng nói của người dùng thông qua việc sử dụng điện thoại di động. Trên điện thoại di động (android) chúng ta sẽ xây dựng một ứng dụng có khả năng nhận diện được các câu lệnh điều khiển từ người dùng, ứng dụng mã nguồn Sphinx trong việc nhận diện tiếng Việt và các thuật toán xử lý ngôn ngữ tự nhiên trong việc xử lý câu lệnh, gửi chúng đến cloud server thông qua wifi. Các node được gắn trên các thiết bị điện nhằm đóng/ngắt nguốn điện cho thiết bị ngay khi nhận được lệnh của được truyền đến từ cloud server. * **Kết quả mong đợi**: * Ứng dụng chạy trên nền tảng di động Android, có thể nhận dạng được những câu lệnh đơn hoặc kép về điều khiển các thiết bị trong gia đình. Sau đó ứng dụng sẽ gởi kết quả nhận diện lên cloud server. * Một hệ thống board mạch theo mô hình server – client được đặt trong nhà, dùng để điều khiển các thiết bị. Trong đó, mỗi thiết bị sẽ là một client, còn server chính là board mạch trung tâm, sẽ tải những câu lệnh từ cloud server để điều khiển các client. | |
| **Kế hoạch thực hiện:***(Mô tả kế hoạch làm việc và phân công công việc cho từng sinh viên tham gia)*   |  |  | | --- | --- | | **Mô tả công việc** | **Phân công** | | 1. Thiết kế hệ thống ứng dụng trên Android về dạng giọng nói dựa trên bộ tool PocketSphinx. | Lâm Duy Thắng | | 1. Xây dựng kết nối từ ứng dụng đến cloud server | Nguyễn Minh Chánh | | 1. Training hệ thống và đi vào xây dựng các mô hình ngôn ngữ, mô hình Acoustic. | Lâm Duy Thắng  Nguyễn Minh Chánh | | 1. Xây dựng board mạch điện tử có thể tải dữ liệu từ server và gởi tín hiệu đến các node | Nguyễn Minh Chánh | | 1. Cải thiện độ chính xác của hệ thống nhận dạng | Lâm Duy Thắng  Nguyễn Minh Chánh | | 1. Hoàn thiện các tính năng rìa của ứng dụng Andorid | Lâm Duy Thắng | | 1. Kiểm thử , sửa lỗi, tối ưu và hoàn thiện hệ thống | Lâm Duy Thắng  Nguyễn Minh Chánh | | 1. Tổng hợp tài liệu và viết báo cáo | Lâm Duy Thắng  Nguyễn Minh Chánh | | |
| **Xác nhận của CBHD**  (Ký tên và ghi rõ họ tên) | **TP. HCM, ngày tháng năm 2017**  **Sinh viên**  (Ký tên và ghi rõ họ tên) |

# MỤC LỤC

[MỤC LỤC 11](#_Toc484647660)

[LỜI MỞ ĐẦU 14](#_Toc484647661)

[TỔNG QUAN ĐỀ TÀI 15](#_Toc484647662)

[1.1 Tình hình nghiên cứu 15](#_Toc484647663)

[1.2 Lý do chọn đề tài 16](#_Toc484647664)

[1.3 Mục tiêu và phạm vi đề tài 17](#_Toc484647665)

[1.3.1 Mục tiêu 17](#_Toc484647666)

[1.3.2 Phạm vi đề tài 17](#_Toc484647667)

[Chương 1 : Cơ sở lý thuyết về nhận diện giọng nói 19](#_Toc484647668)

[1.1 Hệ nhận dạng giọng nói 19](#_Toc484647669)

[1.1.1 Tổng quan 19](#_Toc484647670)

[1.1.2 Các hệ nhận dạng giọng nói 20](#_Toc484647671)

[1.1.3 Một số phương pháp nhận dạng giọng nói 21](#_Toc484647672)

[1.2 Rút trích đặc trưng tín hiệu giọng nói 22](#_Toc484647673)

[1.2.1 Giới thiệu 22](#_Toc484647674)

[1.2.2 Tiền khuếch đại 23](#_Toc484647675)

[1.2.3 Tách từ 24](#_Toc484647676)

[1.2.4 Phân đoạn 24](#_Toc484647677)

[1.2.5 Lấy cửa sổ khung tín hiệu 25](#_Toc484647678)

[1.2.6 Rút trích đặt trưng 27](#_Toc484647679)

[1.2.7 Tìm hiểu về format 40](#_Toc484647680)

[1.3 Gaussian Mixture Model 42](#_Toc484647681)

[1.4 Mô hình Markov ẩn 44](#_Toc484647682)

[1.4.1 Giới thiệu 44](#_Toc484647683)

[1.4.2 Ba bài toán cơ bản của mô hình Markov ẩn 45](#_Toc484647684)

[1.5 Mô hình HMM trong nhận dạng giọng nói 53](#_Toc484647685)

[1.5.1 Giới thiệu 53](#_Toc484647686)

[1.5.2 HMM trong nhận dạng giọng nói 53](#_Toc484647687)

[1.5.3 Các khái niệm và các thuật ngữ 53](#_Toc484647688)

[Chương 2 : Công cụ huấn luyện và nhận dạng giọng nói 54](#_Toc484647689)

[2.1 Tổng quan 54](#_Toc484647690)

[2.2 Huấn luyện và nhận dạng giọng nói với CMU SPHINX 54](#_Toc484647691)

[2.2.1 Giới thiệu 54](#_Toc484647692)

[2.2.2 Vận dụng mô hình HMM trong nhận dạng giọng nó với Sphinx4 55](#_Toc484647693)

[2.2.3 Các thành phần chính trong CMU Sphinx 56](#_Toc484647694)

[2.2.4 Các xử lý chính trong CMU Sphinx 56](#_Toc484647695)

[2.3 Kết luận 58](#_Toc484647696)

[2.3.1 Ưu điểm 58](#_Toc484647697)

[2.3.2 Nhược điểm 58](#_Toc484647698)

[2.4 Sphinx4 với ngôn ngữ Tiếng Việt 59](#_Toc484647699)

[2.4.1 Tổng quan 59](#_Toc484647700)

[2.4.2 Corpus 59](#_Toc484647701)

[Chương 3 : Phân tích và thiết kế hệ thống 61](#_Toc484647702)

[3.1 Thiết lập môi trường CMU SPHINX 61](#_Toc484647703)

[3.1.1 Chuẩn bị hệ điều hành 61](#_Toc484647704)

[3.1.2 Các gói thiết lập Sphinx 61](#_Toc484647705)

[3.1.3 Xây dựng bộ từ điển (Dictionary model) 61](#_Toc484647706)

[3.1.4 Xây dựng bộ huấn luyện ngôn ngữ (Language model) 62](#_Toc484647707)

[3.1.5 Xây dựng mô hình âm học (Aucostic model) 64](#_Toc484647708)

[3.1.6 Cấu hình Sphinx70 71](#_Toc484647709)

[3.1.7 Cấu hình thư mục huấn luyện 71](#_Toc484647710)

[3.1.8 Chuẩn bị data cho quá trình huấn luyện 73](#_Toc484647711)

[3.1.9 Điều chỉnh tham số huấn luyện 73](#_Toc484647712)

[3.1.10 Thực thi huấn luyện 75](#_Toc484647713)

[3.1.11 Sử dụng kết quả huấn luyện 76](#_Toc484647714)

[3.2 Xây dựng ứng dụng trên điện thoại Android 76](#_Toc484647715)

[3.2.1 Giới thiệu ứng dụng 76](#_Toc484647716)

[3.2.2 Cấu trúc ứng dụng 77](#_Toc484647717)

[3.2.3 Hình ảnh demo 78](#_Toc484647718)

[3.3 Board node 81](#_Toc484647719)

[3.3.1 Sơ đồ tổng quan 81](#_Toc484647720)

[3.3.2 Sơ đồ mạch 81](#_Toc484647721)

[3.3.3 Các thông số kỹ thuật của board node 81](#_Toc484647722)

[3.3.4 Sản phẩm demo 81](#_Toc484647723)

[Chương 4 : Thực nghiệm 82](#_Toc484647724)

[4.1 Giá thành sản phẩm 82](#_Toc484647725)

[4.2 Tỉ lệ nhận diện giọng nói 82](#_Toc484647726)

[4.3 Tính ổn định của hệ thống 82](#_Toc484647727)

[4.4 Tính linh hoạt/nâng cấp của hệ thống 82](#_Toc484647728)

[Chương 5 : Kết luận và kiến nghị 83](#_Toc484647729)

[5.1 Kết luận 83](#_Toc484647730)

[5.1.1 Các kết quả đạt được 83](#_Toc484647731)

[5.1.2 Các kết quả chưa đạt được 83](#_Toc484647732)

[5.2 Kiến nghị 83](#_Toc484647733)

[5.3 Hướng phát triển 83](#_Toc484647734)

[TÀI LIỆU THAM KHẢO 84](#_Toc484647735)

# LỜI MỞ ĐẦU

Giao tiếp là vấn đề quan trọng, và đặc biệt giao tiếp bằng giọng nói được sử dụng như một phương tiện đặt biệt quan trọng, trong vấn đề truyền tải thông tin hằng ngày. Với sự phát triển của Công Nghệ Thông Tin đòi hỏi vấn đề số hóa giọng nói trở nên rất quan trọng. Hiện nay, trên thế giới vấn đề nhận dạng giọng nói đặc biệt là tiếng Anh đã không còn quá mới mẻ và đạt được nhiều thành tựu có thể kể đến như Google Now, Cortana giúp cho con người có thể giao tiếp với máy tính một cách nhanh chóng và dễ dàng hơn, giúp thực hiện nhiều việc quan trọng có thể kể đến nhà thông minh, trợ lý ảo, giáo viên ngôn ngữ. Đối với việc nhận dạng Tiếng Việt còn một số khó khăn, do các API hiện tại ít hỗ trợ, chính vì thế nhóm chúng em muốn thực hiện khóa luận này với chủ để tìm hiểu nhận dạng giọng nói tiếng Việt và xây dựng ứng dụng demo quá trình nhận dạng , cùng với sự nỗ lực đó nhóm chúng em cũng muốn góp một phần nhỏ của mình vào việc tìm hiểu bộ nhận dạng giọng nói tiếng việt, góp một phần nhỏ sức lực của mình vào việc hoàn thành, nâng cấp nhận dạng giọng nói Tiếng Việt. Trong quá trình thực hiện, do có một số khó khăn chắc chắn không thể hoàn hảo trong quá trình nhận dạng, ứng dụng còn gặp nhiều lỗi nhận dạng không chính xác mong được thứ lỗi. Khóa Luận bao gồm 2 phần chính: Tìm hiểu nhận dạng thông qua CMU Sphinx và xây dựng ứng dụng demo.

# TỔNG QUAN ĐỀ TÀI

## Tình hình nghiên cứu

### Trong nước

Trong thời đại phát triển công nghệ vượt bậc như ngày nay , hưởng ứng xu thế của toàn cầu ở Việt Nam cụm từ “ Smarthome “ đã không còn quá xa lạ ; điển hình là tại các trường đại học , viện khoa học việc đầu tư cũng như mức độ quan tâm của các bạn sinh viên , nghiên cứu sinh về vấn đề này là cực lớn. Đề cử như tại Trường Đại học Công nghệ (ĐHQGHN) , để có được ngôi nhà thông minh (Smart Home), nhóm các bạn sinh viên đã nghiên cứu, thiết kế, chế tạo các cấu kiện và hệ thống tự động hóa phục vụ giám sát, điều khiển, điều hành cho các nhà cao tầng (nhà công ích và dân dụng). Đây cũng là nghiên cứu nằm trong đề tài KHCN  thuộc Chương trình KHCN cấp Nhà nước giai đoạn 2006-2010 (KC.03/06.-10) do PGS.TS.Trần Quang Vinh làm chủ nhiệm. Bên cạnh đó vào năm 2012 , các bạn sinh viên yêu thích công nghệ trong câu lạc bộ BKIT4U ( ĐHBK\_ĐHQGTPHCM) đã đạt giải nhất cuộc thi “ Thiết kế với TIMCU 2012 ” do Texas Intrument ( TI\_USA ) tổ chức với đề tài  “Smart Home” (ngôi nhà thông minh dựa trên vi điều khiển) . Ngoài ra còn rất nhiều các bạn sinh viên ở các trường Đại học ( ĐHSPKT TPHCM , ĐHBK Đà Nẵng,...) đang nghiên cứu về “ Smarthome “ . Tuy nhiên phần lớn các đề tài chỉ mô phỏng, và điều khiển từ xa qua điện thoại bằng SMS, điều khiên bằng giao diện và các nút bấm trên điện thoại chứ chưa có giải pháp điều khiển bằng giọng nói .

### Ngoài nước

Về đề tài “ Smarthome “ trên thế giới mọi người đã có ý tưởng và bắt tay vào nghiên cứu từ rất lâu , và đã đạt được những thành tựu đáng ngưỡng mộ . Có thể lấy ví dụ như tại trung tâm Tyndall thuộc trường đại học East Anglia đang có 1 dự án về smarthome được gọi là REFIT . Refit sẽ nghiên cứu các khái niệm Smart Home để cung cấp các giải pháp truyền thông , công nghệ thông tin, công nghệ năng lượng tái tạo tại chỗ cho các ngôi nhà tại Anh. Ngoài ra tại trường đại học Columbia đang có một dự án về ngôi nhà thông minh do tiến sỹ Javier Zamora chủ nhiệm , đề án đang nghiên cứu về mô hình ngôi nhà thông minh có khả năng tự dọn dẹp chính nó, điều chỉnh để thay đổi thời tiết và cắt giảm tiêu thụ năng lượng, cảm biến nhiệt độ và khí hậu, và vòi phun nước tự động, đèn tự động bật/tắt tùy vào ánh sáng của môi trường. Bên cạnh đó các đại gia công nghệ như Google, Apple, SamSung, Siemens, Scheneider cũng đang chạy đua cho ngôi nhà thông minh. Phần lớn các đề tài đều khá sáng tạo, có tính thực tiễn và đầu tư phát triển công nghệ cao , tuy nhiên giá thành cho sản phẩm đầu ra cũng sẽ rất đắt đỏ và hệ thống khi lắp đặt sẽ làm ảnh hưởng đến toàn bộ kiến trúc của ngôi nhà .

## Lý do chọn đề tài

Hiện nay, lĩnh vực “*Smarthome*” cũng đã khá phát triển và được đông đảo mọi người biết đến và dành sự quan tâm nhất định. Trên thị trường hiện nay đã có rất nhiều sản phẩm “Smarthome” được điều khiển bằng điện thoại, tablet, PC thông qua mạng internet trên nền tảng web; nhưng các thao tác khá phức tạp và mất thời gian khi ta phải đăng nhập vào web, lựa chon thiết bị và đưa ra mệnh lệnh điều khiển. Bên cạnh đó còn có các giải pháp điều khiển “*Smarthome*” bằng tin nhắn SMS từ điện thoại nhưng giải pháp này vẫn vấp phải những vấn để của riêng nó. Ví dụ như mỗi tin nhắn SMS gửi đi và nhận về thì người dùng đều phải tốn phí, cú pháp soạn tin SMS đôi khi phức tạp khó nhớ, và thao tác cũng khá mất thời gian. Từ đó đặt ra bài toán cấp bách cho các giải pháp nâng cao tính tiện lợi cho “Smarthome”. Xuất phát từ nhu cầu ấy chúng tôi đã xây dựng nên đề tài “ *ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG GIA ĐÌNH BẰNG GIỌNG NÓI THÔNG QUA ĐIỆN THOẠI DI ĐỘNG* ” để giải quyết các vấn đề còn tồn động của “ Nhà Thông Minh”.

Điểm mới trong đề tài này là thay vì phải thao tác trên các giao diện nút bấm hay SMS chúng ta sẽ tương tác với ngôi nhà bằng chính giọng nói của chúng ta, một cách giao tiếp cơ bản, đơn giản và nhanh chóng. Ngày nay điện thoại di động (*Smartphone*) ngày càng trở nên phổ biến và gần gũi với tất cả mọi người, vì thế tận dụng nó như một thiết bị điều khiển ngôi nhà là hết sức hợp lí. Bên cạnh đó mạng wifi đã có mặt ở khắp mọi nơi thuận tiện cho việc vận dụng vào đề tài này, thay vì dùng SMS tốn kém. Đặc biệt hơn nữa khi giao tiếp bằng giọng nói chúng ta sẽ cảm thấy thật sự tiện lợi. Ví dụ như khi lái ô-tô chúng ta chỉ cần ra lệnh bằng giọng nói không cần phải dùng tay thao tác, và khi đang làm các công việc trong gia đình như làm bếp, làm việc trên máy tính thì việc điều khiển các thiết bị trong nhà bằng giọng nói sẽ phát huy được sự tiện lợi cao nhất.

## Mục tiêu và phạm vi đề tài

### Mục tiêu

Xây dựng một hệ thống có thể điều khiển các thiệt bị điện trong gia đình bằng chính giọng nói của người dùng thông qua việc sử dụng điện thoại di động. Trên điện thoại di động (android) chúng ta sẽ xây dựng một ứng dụng có khả năng nhận diện được các câu lệnh điều khiển từ người dùng, ứng dụng mã nguồn Sphinx trong việc nhận diện tiếng Việt và các thuật toán xử lý ngôn ngữ tự nhiên trong việc xử lý câu lệnh, gửi chúng đến cloud server thông qua wifi. Các node được gắn trên các thiết bị điện nhằm đóng/ngắt nguốn điện cho thiết bị ngay khi nhận được lệnh của được truyền đến từ cloud server.

### Phạm vi đề tài

* Đối tượng ( thiết bị điện ) được điều khiển : tất cả các thiết bị điện dùng điện 220V trong gia đình có chân phích cắm cho việc cấp nguồn.
* Khả năng điều khiển : bật/tắt (on/off) nguồn điện.
* Số lượng node điều khiển tối đa có thể có : tùy vào service cloud server mà để tài dùng thì sẽ có số lượng tối đa khác nhau :
* Account free : 10 kết nối
* $19/month : 100 kết nối
* $99/month : 1000 kết nối
* $299/month : 10000 kết nối

Hiện tại ở đề tài nhóm chọn loại account free với tối đa 10 kết nối.

* Khả năng tăng thêm node điều khiển vào hệ thống : có thể linh hoạt add thêm node vào đến mức tối đa có thề có cloud server.
* Định dạng câu lệnh cho người dùng :

|  |  |  |
| --- | --- | --- |
| **Action**  *“bật, tắt, tăng, giảm, kiểm tra”* | **Device**  *“đèn, quạt, điều hòa, bếp”* | **Area**  *“phòng ngủ, phòng khách, nhà bếp, lầu, 1 … 100”* |

VD : Bật tất cả đèn lầu mười hai .

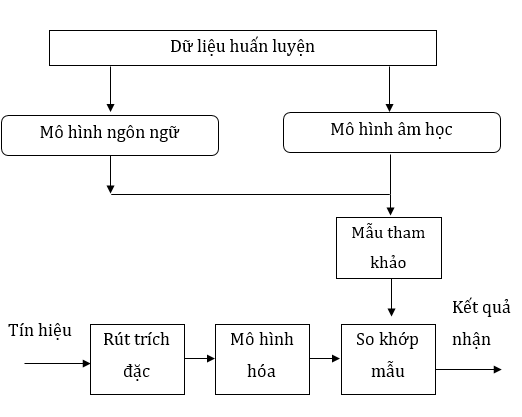
Kiểm tra quạt ở phòng ngủ năm

# Cơ sở lý thuyết về nhận diện giọng nói

## Hệ nhận dạng giọng nói

### Tổng quan

Nhận dạng giọng nói là quá trình biến đổi các tín hiệu âm thanh thành một chuỗi các từ.



***Hình 1. 1.1Sơ đồ nhận dạng giọng nói***

Tín hiệu giọng nói sẽ được rút trích đặc trưng, kết quả thu được của quá trình này gọi là các vector đặc trưng.

Để thực hiện việc so sánh, hệ thống phải được huấn luyện và xây dựng các đặc trưng. Trong quá trình huấn luyện, hệ thống sử dụng các vector đặc trưng để ước lượng, tính toán các tham số cho các mẫu gọi là các mẫu tham khảo. Một mẫu tham khảo sẽ được dùng để so sánh và nhận dạng.

Trong quá trình nhận dạng, dãy các vector đặc trưng của tín hiệu cần nhận dạng được so sánh với các mẫu tham khảo được xây dựng trong quá trình huấn luyện. Hệ thống sẽ tính toán độ tương đồng của dãy các vector đặc trưng với mẫu tham khảo. Việc tính toán độ tương đồng được thực hiện bằng cách áp dụng các thuật toán như thuật toán Viterbi trong mô hình Markov ẩn. Mẫu có độ tương đồng cao nhất được cho là kết quả của quá trình nhận dạng.

### Các hệ nhận dạng giọng nói

#### Nhận dạng từ liên tục và tách biệt

Một hệ nhận dạng giọng nói có thể là một trong hai dạng: nhận dạng liên tục và nhận dạng từng từ. Nhận dạng liên tục tức là nhận dạng giọng nói được phát liên tục trong một chuỗi tín hiệu chẳng hạn như một câu nói, một mệnh lệnh hoặc một đoạn văn. Các hệ thống dạng này rất phức tạp vì nếu người nói phát âm liên tục sẽ rất khó tách từ. Trong đó, kết quả tách từ ảnh hưởng rất lớn đến các quá trình sau.

Đối với mô hình nhận dạng từng từ, mỗi từ cần nhận dạng được phát âm một cách rời rạc. Các ứng dụng thực tiễn đối với mô hình dạng này có thể kể đến như hệ thống điều khiển bằng giọng nói, quay số bằng giọng nói, nhập văn bản bằng giọng nói, tìm kiếm bằng giọng nói,…

#### Nhận dạng phụ thuộc người nói và độc lập

Đối với nhận dạng phụ thuộc người nói thì mỗi hệ nhận dạng chỉ phục vụ được cho một người và nó sẽ không hiểu người khác nói gì nếu chưa được huấn luyện. Do đó, đối với những hệ thống dạng này không được áp dụng rộng rãi.

Hệ nhận dạng độc lập với người nói thì khắc phục được nhược điểm của hệ nhận dạng phụ thuộc nhưng hệ nhận dạng này lại gặp vấn đề về độ chính xác. Vì trong thực tế mỗi người có một giọng nói khác nhau, ở những vùng miền và độ tuổi khác nhau cũng ảnh hưởng đến quá trình nhận dạng. Dó đó, để khắc phục vấn đề này đòi hỏi hệ thống phải được huấn luyện với lượng dữ liệu lớn và phong phú.

### Một số phương pháp nhận dạng giọng nói

#### Phương pháp âm – ngữ học

Phương pháp này dựa trên lý thuyết về âm – ngữ học. Trong lời nói tồn tại các đơn vị ngữ âm xác định, có tính phân biệt. Các bước nhận dạng của phương pháp này bao gồm:

*Bước 1: Phân đoạn và gán nhãn*

Chia tín hiệu tiếng nói thành các đoạn có đặc tính âm học đặc trưng cho một đơn vị ngữ âm, đồng thời gán cho mỗi đoạn âm thanh đó một hay nhiều nhãn ngữ âm phù hợp.

*Bước 2: Nhận dạng*

Dựa trên một số điều kiện ràng buộc về từ vựng để xác định một hay một chuỗi từ đúng trong các chuỗi nhãn ngữ âm đã được tạo ra ở bước 1.

#### Phương pháp nhận dạng mẫu

*Bước 1:* Sử dụng tập mẫu tiếng nói hay còn gọi là cơ sở dữ liệu tiếng nói để huấn luyện. Kết quả thu được của bước này là các mẫu tiếng nói đặc trưng.

*Bước 2:* So sánh mẫu tiếng nói cần nhận dạng với các mẫu tiếng nói đặc trưng được tạo ra ở bước 1 để nhận dạng.Mô hình Markov ẩn (HMM), mạng nơ-ron nhân tạo (ANN), so sánh thời gian động (DTW),… áp dụng phương pháp nhận dạng mẫu này.

#### Phương pháp ứng dụng trí tuệ nhân tạo

Phương pháp ứng dụng trí tuệ nhân tạo kết hợp các phương pháp trên nhằm tận dụng tối đa các ưu điểm của chúng, đồng thời bắt chước các khả năng của con người trong phân tích và cảm nhận các sự kiện bên ngoài để áp dụng vào nhận dạng tiếng nói.

Đặc điểm của phương pháp này là sử dụng hệ chuyên gia để phân đoạn, gán nhãn ngữ âm. Sử dụng mạng nơ-ron nhân tạo để học mối quan hệ giữa các ngữ âm.

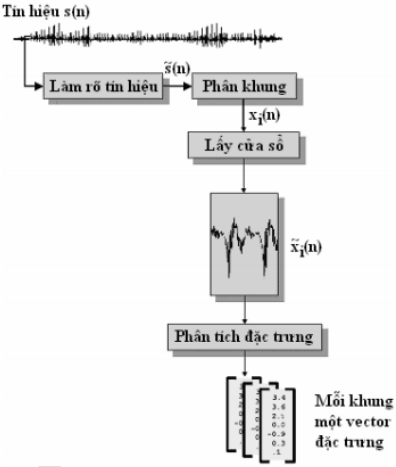
Việc sử dụng hệ chuyên gia nhằm tận dụng kiến thức con người vào hệ nhận dạng:

* Kiến thức về âm học: Để phân tích phổ và xác định đặc tính âm học của các mẫu tiếng nói.
* Kiến thức về từ vựng: sử dụng để kết hợp các khối ngữ âm thành các từ cần nhận dạng.
* Kiến thức về cú pháp: nhằm kết hợp các từ thành các câu cần nhận dạng.
* Kiến thức về ngữ nghĩa: nhằm xác định tính logic của các câu đã được nhận dạng.

## Rút trích đặc trưng tín hiệu giọng nói

### Giới thiệu

Rút trích đặc trưng của tiếng nói là một trong những công đoạn quan trọng trong quá trình nhận dạng tiếng nói, điều này giúp giảm thiểu số lượng dữ liệu trong việc huấn luyện và nhận dạng. Mặt khác, việc rút trích đặc trưng còn làm rõ sự khác biệt của tiếng nói này với tiếng nói khác hay làm mờ đi sự khác biệt của hai lần phát âm cho cùng một từ. Kết quả của quá trình rút trích đặc trưng là một hoặc nhiều vector đặc trưng. Có nhiều phương pháp để thực hiện rút trích đặc trưng, 2 trong số đó là phương pháp MFCC và LPC.



Tín hiệu âm thanh lưu trong máy tính là tín hiệu digital [13], mô hình hoá tín hiệu âm thanh trong máy tính dưới dạng toán học là một hàm s(n) với n chỉ thời gian (đơn vị thông thời là ms) và s(n) là biên độ âm.

### Tiền khuếch đại

Làm nổi rõ đặc trưng của tín hiệu và làm cho nó ít nhạy cảm hơn với các hiệu ứng do độ chính xác hữu hạn ở những bước sau.

Theo các nghiên cứu về âm học thì giọng nói có sự suy giảm 20dB/decade khi lên tần số cao do đặc điểm sinh lý của hệ thống phát âm con người. Để khắc phục sự suy giảm này, chúng ta sẽ phải tăng cường tín hiệu lên một giá trị gần 20dB/decade. Bên cạnh đó, hệ thống thính giác con người có xu hướng nhạy cảm hơn với vùng tần số cao. Dựa vào những đặc điểm trên, ta sẽ áp dụng bộ lọc thông cao để tiền xử lý các tín hiệu thu được nhằm làm rõ vùng tín hiệu mà tai người có thể nghe được. Bộ lọc áp dụng công thức sau:



Trong đó apre là hệ số nhấn mạnh, thường có giá trị là 0.9700002861.

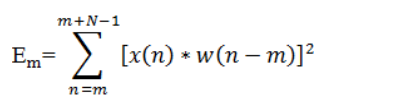
Bộ lọc này có chức năng tăng cường tín hiệu tại tần số cao (tần số trên 1KHz).Tín hiệu tiếng nói đã số hoá, s(n), được đưa qua một hệ số bậc thấp, để làm phẳng tín hiệu về phổ và làm nó ít bị ảnh hưởng bởi các hiệu ứng có độ chính xác hữu hạn sau này trong quá trình xử lý tín hiệu Trong trường hợp sử dụng bộ lọc áp theo công thức trên đầu ra của dãy tiền khuếch đại s’(n), liên quan đến đầu vào của dãy tín hiệu s(n), theo đẳng thức vi phân sau:



### Tách từ

Tín hiệu giọng nói s(n) sau khi được khuếch đại, sẽ được chuyển sang để tách từ. Đây là công đoạn chia toàn bộ tín hiệu thu được thành những đoạn tín hiệu mà trong đó chỉ chứa nội dung của một từ.

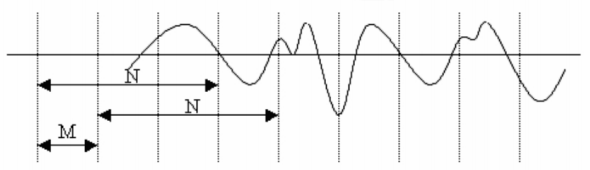
Có nhiều phương pháp để tách điểm đầu và điểm cuối của một ra khỏi toán bộ tín hiệu tiếng nói, trong đó phương pháp dùng hàm năng lượng thời gian ngắn là phương pháp được sử dụng phổ biến. Với một cửa sổ kết thức tại mẫu thứm, hàm năng lượng thời gian ngắn E(m) được xác định :



### Phân đoạn

Tín hiệu tiếng nói là một đại lượng biên thiên theo thời gian và không ổn định nên không thể xử lý trực tiếp trên đó được. Do đó, tín hiệu được chia ra thành các khung với chiều dài tương đối nhỏ để lấy được các đoạn tín hiệu tương đối ổn định và xử lý tiếp trong các bộ lọc tiếp theo. Theo các nghiên cứu đã có thì trong khoang thời gian 10-20ms, tín hiệu tiếng nói tương đối ổn định. Nên ở bước này, người ta thường phân tín hiệu thành các khung với kích thước 20-30ms. Nhưng để tránh mất mát và làm gián đoạn tín hiệu ban đầu, khi phân khung, người ta chồng lấp các khung lên nhau khoảng 10-15ms.

Trong bước này tín hiệu được tiền khuếch đại **,** được chia thành các khung, mỗi khung gồm N mẫu, khoảng cách giữa các khung là M mẫu.



***Hình 1. 1.2 Phân đoạn tiếng nói thành các khung.***

Hình 1.0.3 minh họa cách phân thành các khung trong trường hợp M = 1/3N.

Cụ thể khung thứ nhất gồm N mẫu tiếng nói đầu tiên. Khung thứ hai bắt đầu từ mẫu thứ M và kết thúc ở vị trí M+N-1. Tương tự khung thứ I bắt đầu từ mẫu thứ i\*M và kết thúc ở vị trí i\*M+N-1. Tiến trình này tiếp tục cho đến khi các mẫu tiếng nói đều đã thuộc về một hay nhiều khung. Như vậy, nếu MN thì các khung kề nhau sẽ có sự chồng lấp dẫn đến kết quả là các phép rút trích đặc trưng có tương quan với nhau từ khung này sang khung khác. Ngược lại, nếu M>N thì sẽ không có sự chồng lấp giữa các khung kề nhau.

### Lấy cửa sổ khung tín hiệu

Lấy cửa sổ khung tín hiệu cho mỗi khung để giảm thiểu sự gián đoạn ở đầu và cuối mỗi khung. Một dãy tín hiệu con được lấy ra từ một tín hiệu dài hơn hoặc dài vô hạn x(n) gọi là một cửa sổ tín hiệu. Việc quan sát tín hiệu x(n) bằng một đoạn XN(n) trong khoảng n0...(n0 + N – 1) tương đương với việc nhận x(n) với một hàm cửa sổ w(n-n0).

x(n) n0 nn0 + N - 1

0 (n<n0) v (n>n0 + N – 1)

(2.3)

XN(n) = x(n).w(n-n0) =

Trong xử lý tín hiệu số, các cửa sổ thường được biểu diễn thông qua cửa sổ Hamming tổng quát:



(2.4)

W(n) =

0

Tùy theo các giá trị khác nhau của  mà ta có các cửa sổ khác nhau:

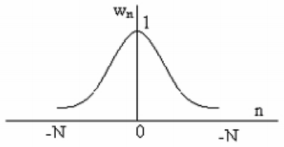
Với  = 0.54, ta có cửa sổ Hamming:



(2.5)

W(n) =

0



***Hình 1.1.2.1. Cửa sổ Hamming vớilà 0.54.***

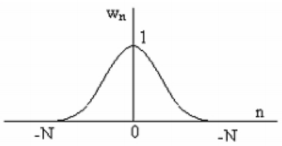
Với  = 0.5, ta có cửa sổ Hanning:



(2.6)

W(n) =

0



***Hình 1.1.2.2. Cửa sổ Hanninglà 0.5***

Với  = 1, ta có cửa sổ chữ nhật:

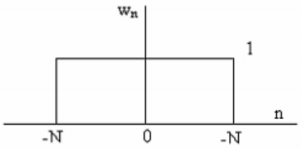


(2.7)

W(n) =

1

0



***Hình 1.1.2.3. Cửa sổ chữ nhật***

Ý nghĩa của việc áp cửa sổ : là nhằm mục đích có được dữ liệu theo miền tần số chuẩn để đưa vào phép biến đổi Fourier rời rạc

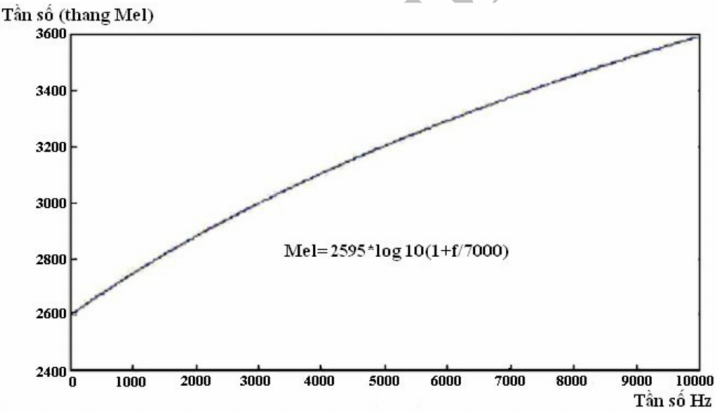
### Rút trích đặt trưng

#### Rút trích đặt trưng với MFCC

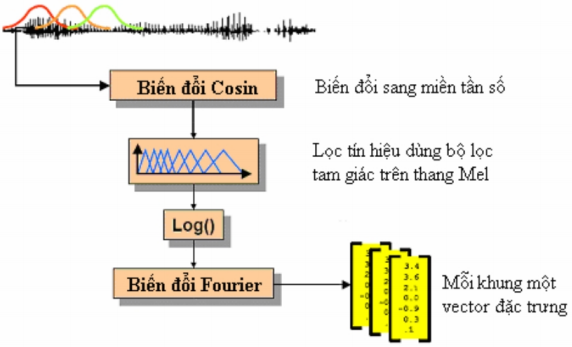
Các nghiên cứu cho thấy rằng hệ thống thính giác của con người thu nhận âm thanh với độ lớn các tần số âm thanh không theo thang tuyến tính. Do đó, các thang âm thanh đã ra đời cho phù hợp với sự tiếp nhận của thính giác con người.

Mel Scale Frequency Cepstral Coefficients (MFCC) là một phương pháp rút trích đặc trưng sử dụng dãy bộ lọc được Davis và Mermelstein đưa ra vào năm 1980 khi họ kết hợp các bộ lọc cách khoảng không đều với phép biến đổi Cosin rời rạc (Discrete Cosin Transform) thành một thuật toán hoàn chỉnh được ứng dụng trong lĩnh vực nhận dạng giọng nói liên tục. Đồng thời cũng định nghĩa khái niệm hệ số Cepstral và thang đo tần số Mel (Mel scale).

Các thang được xây dựng bằng thực nghiệm, cho nên người ta xây dựng các công thức để xấp xỉ sự chuyển đổi này. Trong các thang và công thức dạng đó thì đặc trưng MFCC sử dụng thang Mel. Thang Mel được thể hiện thông qua đồ thị sau:

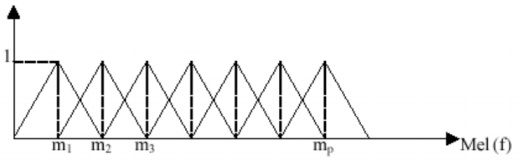


Hình *.* Đồ thì biểu diễn mối quan hệ giữa Mel và Hz.

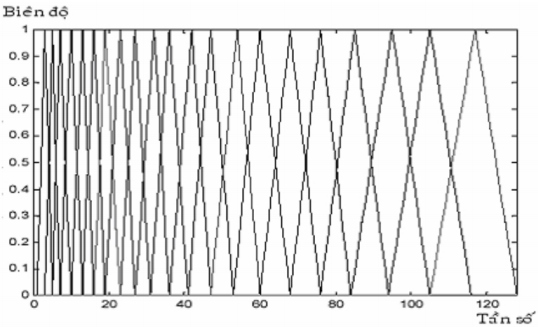


Hình Các bước rút trích đặc trưng MFCC.

Ta dùng phép biến đổi Fourier để chuyển đổi tín hiệu từ miền thời gian sang miền tần số sau đó ta dùng dãy bộ lọc tín hiệu đó là dãy bộ lọc tam giác có tần số giưa đều nhau trên thang Mel.

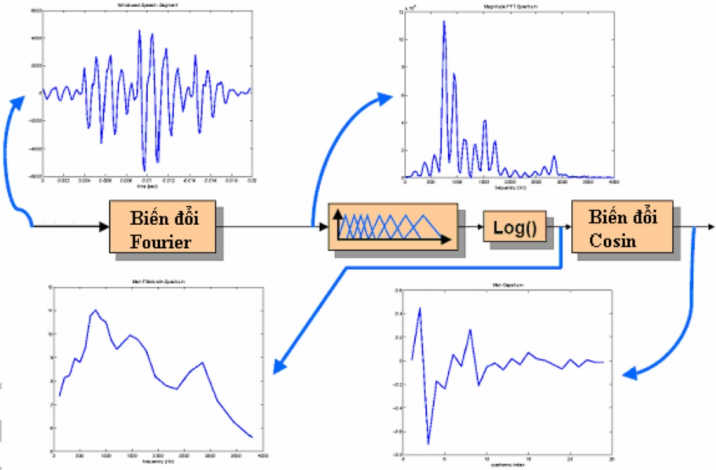


Hình *.* Bộ lọc trên thang Mel.



Hình Bộ lọc trên tần số thật*.*

Lấy log trên dãy kết quả từ dãy bộ lọc và thực hiện biến đổi cosin rời rạc ta thu được các hệ số đặc trưng MFCC.



Hình . Minh họa các bước biến đổi MFCC.

Như vậy, quá trình rút trích đặc trưng theo phương pháp MFCC sẽ được tiến hành như sau:

Tín hiệu sau khi qua tiền xử lý sẽ được chia thành các khung (frame) có khoảng thời gian ngắn. Dựa vào mỗi khung đó, sau khi áp dụng các bước biến đổi và lọc sẽ được một vecto tương ứng.

1. Biến đổi FFT (Fast Fourier Transform)

Biến đổi FFT thực chất là một biến đổi DFT (Discrete Fourier Transform ) nhưng được tối ưu bằng các thuật toán nhanh và gọn hơn để đáp ứng các yêu cầu xử lý theo thời gian thực trong các lĩnh vực như xử lý âm thanh, hình ảnh,...

Fast Fourier là một phép biến đổi thuận nghịch có đặc điểm bảo toàn tính tuyến tính bất biến, tuần hoàn và tính trễ. Dùng để biến đổi tín hiệu tương tự sang miền tần số, nó gồm các công thức như sau:

Công thức phép biến đổi thuận (dùng để phân tích tín hiệu):

X(k)=-j2/N k = 0,1,2,…,N-1

Công thức phép biến đổi nghịch (dùng để tổng hợp lại tín hiệu):

x(n) = 

n = 0,1,2,…N-1

Trong đó: x(n) = a(n) + b(n)

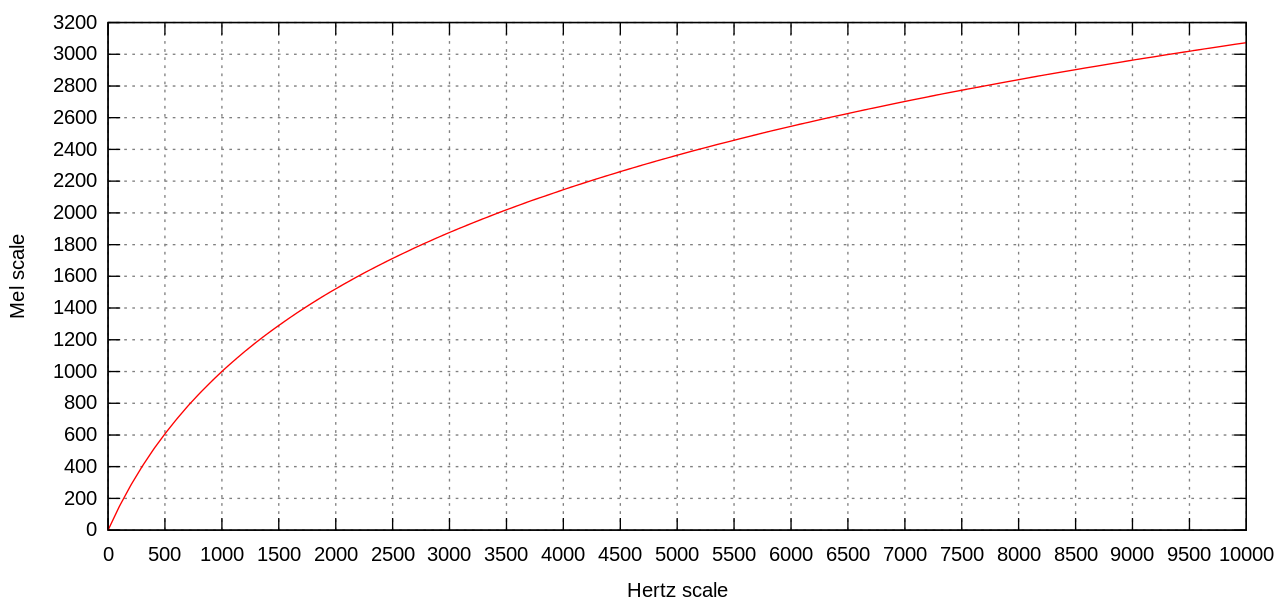
Kết quả chúng ta có được khi thực hiện FFT là dãy tín hiệu Xt(k) để đưa vào bộ lọc Mel-scale.

1. Lọc qua bộ lọc Mel-scale

Trong lĩnh vực nghiên cứu về nhận dạng tiếng nói, đòi hỏi chúng ta phải hiểu và mô phỏng chính xác khả năng cảm thụ tần số âm thanh của tai người. Chính vì thế các nhà nghiên cứu đã xây dựng một thang tần số - hay gọi là thang tần số Mel (Mel scale) dựa trên cơ sở thực nghiệm nhiều lần khả năng cảm nhận âm thanh của con người. Thang tần số Mel được định nghĩa trên tần số thực theo công thức:

m = 2595log10

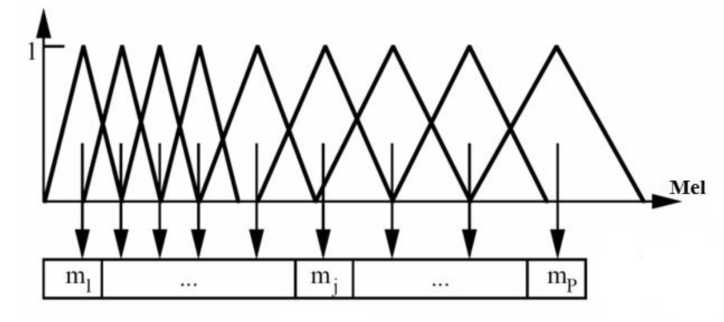
Trong đó: m là tần số trong thang Mel, đơn vị là Mel; f là tần số thực, đơn vị là Hz.



Hình Biểu đồ thang tần số Mel theo tần số thực.

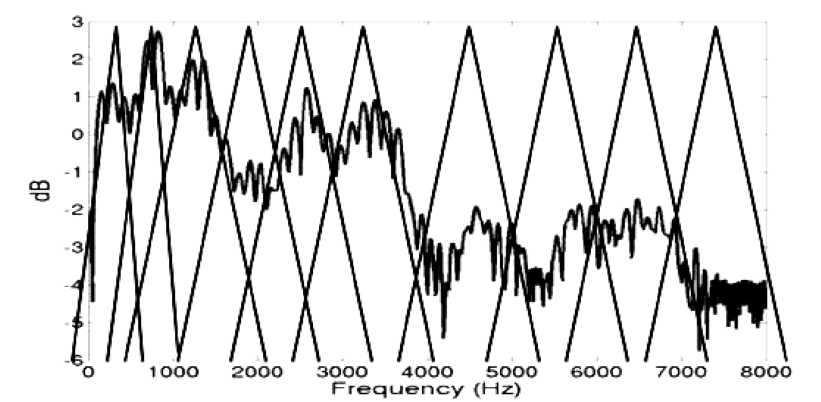
Theo biểu độ trên thì trong khoảng tần số thấp hơn 1kHz thì đồ thị trên gần như là tuyến tính, nghĩa là trong khoảng tần số dưới 1kHz, tần số Mel và tần số thực. Trong khoảng tần số trên 1kHz thì mối quan hệ này là quan hệ Logarit.

Dựa vào các thực nghiệm trên tai người, người ta đã xác định được các tần số thực mà tai người có thể nghe duoc va chứa đựng nhiều thông tin. Sau đó chuyển các tần số này sang tần số Mel va xây dựng một thang đo như sau:



Hình Băng lọc tần số Mel.

Ta dùng thang đo này để áp vào dãy sóng tín hiệu thu được sau khi thực hiện FFT.



Hình Đưa tín hiệu vào băng lọc tần số Mel.

Kết quả của bước này là chúng ta sẽ có được tập hợp các tần số Yt(m) là giao điểm của sóng tần số vời thang tần số Mel từ dãy tín hiệu Xt(k).

1. Logarit giá trị năng lượng (Logarit of filter energies)

Mục đích của bước này là nén các giá trị đã lọc được vào miền giá trị nhỏ hơn để xử lý nhanh hơn. Nên các giá trị thu được ở mỗi kênh lọc sẽ được lấy Logarit với công thức log{|Yt(m)|2}.

1. Biến đổi cosin rời rạc

Dựa vào phổ tín hiệu tiếng nói của con người trên miền tần số, ta có thể thấy rằng phổ tín hiệu khá trơn, nên khi lấy các giá trị năng lượng ra từ các bộ lọc, các giá trị này có sự tương quan khá gần nhau, dẫn đến các đặc trưng ta rút được sẽ không rõ ràng. Chính vì thế, ta thực hiên biến đổi DCT (Discrete Cosin Transform) để làm rời rạc các giá trị này ra cho nó ít tương quan với nhau, làm tăng tính đặc trưng của các tham số. Giá trị thu được sau bước này ta gọi là hệ số Cepstral.

Ci = 

N là số kênh lọc

Trong đó:

Mj là giá trị logarit năng lượng của mạch lọc thứ j.

i là bậc của hệ số cepstral

Thông thường người ta lấy i trong đoạn [1,12] là số lượng đặc trưng trong mỗi vecto đặc trưng. Trong các hệ nhận dạng, số lượng đặc trưng nằm trong khoảng (10,15) là đủ để cho kết quả nhận dạng tương đối mà dữ liệu xử lý lại không quá lớn.

Sau khi thực hiện biến đổi DCT, theo công thức trên ta thấy các hệ số thu được sẽ tăng tuyến tính theo số bậc của nó. Hệ số Cepstral có số bậc cao sẽ có giá trị rất cao, ngược lại các hệ số với số bậc thấp sẽ có giá trị rất thấp. Sự chênh lệch này sẽ gây khó khăn cho chúng ta trong qua trình mô hình hóa dữ liệu và xử lý sau này. Vì khi có sự chênh lệch cao, ta phải dùng miền giá trị lớn để biểu diễn dữ liệu, và gặp khó khăn khi đưa vào các mô hình xử lý xác suất,..Nên để có các hệ số tối ưu cho các qua trình sau, ta sẽ thực hiện việc điều chỉnh các hệ số này để giảm sự chênh lệch. Việc này thực hiện bằng công thức:

c'n = exp(n \* fc) . cn

Cuối cùng chúng ta sẽ thu được các giá hệ số Cepstral đã được tinh chế. Các hệ số này là đặc trưng MFCC mà chúng ta sẽ sử dụng để huấn luyện và nhận dạng.

#### Rút trích đặt trưng với LPC

1. *Giới thiệu*

LPC là chữ viết tắt của cụm từ: Linear Predictive Coding (mã hóa dự báo tuyến tính). Đây được xem là một trong những phương pháp được sử dụng rộng rãi trong việc rút trích đặt trưng của tín hiệu âm thanh (hay còn được gọi là tham số hóa tính hiệu âm thanh). Đóng vai trò quan trọng trong các kỹ thuật phân tích tiếng nói. Đây còn được xem là một phương pháp hiệu quả cho việc nén (mã hóa với chất lượng tốt) dữ liệu tiếng nói ở mức bit rate thấp.

1. *Cơ sở âm học của phương pháp LPC*

Tiếng nói hay còn được xem là âm thanh do con người phát ra từ miệng bắt nguồn từ từ sự rung động của dây thanh âm (do sự thay đổi áp suất không khí từ phổi đưa lên), sự rung động này mang 2 đặc tích là cường độ (intensity) và tần số (frequency). Âm thanh này sau đó được truyền qua cuống họng đến khoang miệng và khoan mũi. Tại đây dựa vào cấu tạo vòng miệng khi nói, cách đặt lưỡi, chuyễn động của lưỡi và cơ miệng... sẽ góp phần gây ra sự cộng hưởng của âm thanh (hay còn được gọi là các Formant), kết quả chính là tiếng nói mà ta nghe được. LPC phân tích những tín hiệu tiếng nói bằng cách ước tính các formant, loại bỏ đi những thành phần ảnh hưởng của nó (những thứ không mang giá trị tiếng nói trong âm phát ra), và ước lượng các đặc điểm về cường độ, tần số của phần âm thanh còn lại. Qúa trình loại bỏ ở trên còn được gọi là quá trình lọc nghịch đảo (inverse filtering) và phần âm thanh còn lại gọi là“cặn” (residue) mang những yếu tố và đặc trưng cốt lõi của âm thanh. Kết quả còn lại sau quá trình LPC là những con số, mà mô tả những đặc điểm quan trọng của các formant cũng như phần âm thanh còn lại. các con số này có thể được dùng đại diện như tín hiệu tiếng nói ban đầu, hiệu quả hơn trong việc lưu trữ, phân tích nội dung, truyền tải tiếng nói... LPC còn được dùng trong quá trình tổng hợp lại tiếng nói từ các con số đặc trưng trên.

1. *Nội dung phương pháp rút trích đặc trưng LPC*

Ý tưởng của LPC là một mẫu tiếng nói cho trước ở thời điểm n, s(n), có thể được xấp xỉ bằng một tổ hợp tuyến tính của p mẫu tiếng nói trước đó:

s(n)  a­­1(n-i) + a2s(n-2) + ... + aps(n-p)

Trong đó a1, a2, ..., an coi là các hằng trên toàn khuông phân tích tiếng nói. Ta chuyển đẳng thức (1) trên tương đương bằng cách thêm giới hạn kích thích, Gu(n), có:

s(n) = 

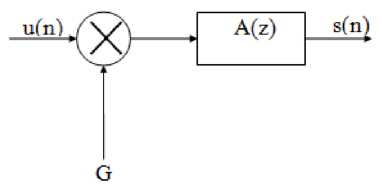
Trong đó u(n) là kích thích đã chuẩn hoá, G là độ khuếch đại của kích thích. Biểu diễn đẳng thức (2) trong miền z ta có quan hệ:

s(n) = 

dẫn đến hàm chuyển

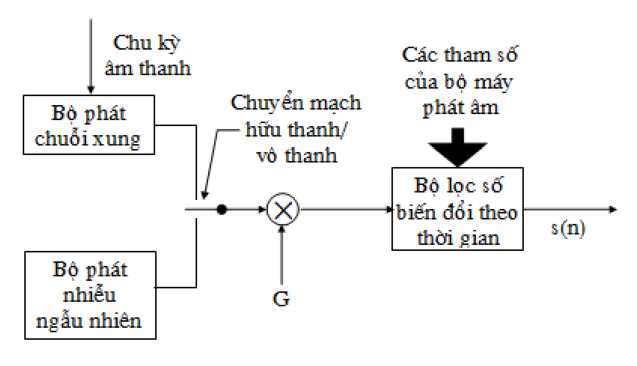
H(z) = 

Hình sau diễn tả đẳng thức bên trên, mô tả nguồn kích thích đã chuẩn hóa u(n), sau đó được nâng mức với độ khuếch đại G, và đóng vai trò đầu vào của hệ toàn cực, H(z) = , tạo ra tín hiệu tiếng nói s(n).



Hình Mẫu tiếng nói dự báo tuyến tính.

Chức năng kích thích thực tế đối với tiếng nói về bản chất là chuỗi liên tục các xung gần như điều hoà (với các âm hữu thanh) và là một nguồn nhiễu ngẫu nhiên (với các âm vô thanh), mẫu hoà âm thích hợp đối với tiếng nói, ứng với phân tích LPC được mô tả trong phía dưới. Tại đây, nguồn kích thích đã chuẩn hoá được chọn nhờ một chuyển mạch mà vị trí của nó được điều khiển bởi đặc tính hữu thanh/vô thanh của tiếng nói, chọn cả chuỗi xung gần điều hoà làm kích thích đối với các âm hữu thanh, chọn cả chuỗi nhiễu ngẫu nhiên cho các âm vô thanh. Độ khuếch đại G của nguồn được ước tính từ tín hiệu tiếng nói, và nguồn được nâng mức được dùng làm đầu vào của bộ lọc số (H(z)), điều khiển bởi đặc tính của các tham số dải phát âm của tiếng nói đang được tạo. Như vậy các tham số của mẫu này là phân loại hữu thanh/vô thanh, chu kỳ cao độ của âm hữu thanh, tham số khuếch đại, và các hệ số của bộ lọc số {ak}, các tham số này đều biến đổi rất chậm theo thời gian.



Hình Mẫu phân tích tiếng nói theo phương pháp LPC.

Dựa vào mẫu trên hình, quan hệ chính xác giữa s(n) và u(n) là



Ta coi tổ hợp tuyến tính của các mẫu tiếng nói trước đó là ước lượng (n), được định nghĩa là:



Bây giờ ta thiết lập lỗi dự báo, e(n), được định nghĩa là:



Với hàm chuyển đổi lỗi:



(2.21)

Rõ ràng là khi s(n) được tạo thực sự bởi một hệ tuyến tính kiểu thì lỗi dự báo e(n) sẽ bằng Gu(n), kích thích được khuếch đại.

Vấn đề cơ bản của phân tích dự báo tuyến tính là xác định tập các hệ số dự báo, {ak}, trực tiếp từ tín hiệu tiếng nói. Vì các đặc tính phổ của tiếng nói biến đổi theo thời gian nên các hệ số dự báo tại thời điểm cho trước n, phải được ước lượng từ một đoạn tín hiệu tiếng nói ngắn xuất hiện quanh thời điểm n. Như vậy cách cơ bản là tìm một tập các hệ số dự báo giảm thiểu lỗi dự báo trung bình bậc hai trong một đoạn dạng sóng tiếng nói. (Thường thì kiểu phân tích phổ thời gian ngắn nàyđược thực hiện trên các khuông tiếng nói liên tiếp, có dãn cách khuông khoảng 10ms).

Nội dung tiếp theo trình bày các bước chính trong công đoạn phân tích đặc trưng theo phương pháp LPC.

1. *Phân tích tự tương quan*

Sau khi tiến hành công đoạn nhận với hàm cửa sổ. Mỗi khung của tín hiệu được chia cửa sổ là tự tương quan với khung tiếp theo để cho:

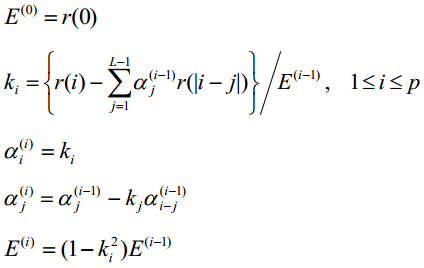


re(m) =

Trong đó giá trị tự tương quan cao nhất, p, là bậc của phép phân tích LPC. Các giá trị p thường dùng là từ 8 đến 16, với p=8 được sử dụng nhiều trong các hệ phân tích LPC.

1. *Phân tích LPC*

Bước xử lý tiếp theo là phân tích LPC, chuyển từng khuông của p+1 mối tự tương quan thành một "tập tham số LPC", đó là các hệ số LPC. Phương pháp chính thức để chuyển từ các hệ số tự tương quan sang tập tham số LPC (để dùng cho phương pháp tự tương quan LPC) được gọi là phương pháp Durbin và có thể cho một cách hình thức như thuật toán (để thuận tiện, ta sẽ bỏ qua  nhỏ ở dưới ):



Trong đó tổng trong đẳng thức được bỏ qua đối với i=1. Tập các đẳng thức trên được giải đệ qui với i=1,2,...,p, và lời giải cuối cùng được cho là

am = các hệ số LPC =a ,1 ≤ m ≤ p

km = các hệ số PARCOR

gm = các hệ số truyền miền logarit = log()

Lúc này ta có thể dùng các hệ số LPC làm vector đặc trưng cho từng khung. Tuy nhiên có một phép biến đổi tạo ra dạng hệ số khác có độ tập trung cao hơn từ các hệ số LPC, đó là phép phân tích Ceptral.

1. *Phân tích cepstral*

Một tập tham số LPC rất quan trọng, có thể suy trực tiếp từ tập hệ số LPC, là các hệ số Cepstral LPC, c(m). Dạng đệ qui là:

Cm = am + 

Cm = 

Trong đó ơ2 là độ khuếch đại trong mẫu LPC. Các hệ số cepstral, là các hệ số của biểu diễn chuyển đổi Fourier của logarit phổ cường độ, được mô tả là một tập đặc tính mạnh, đáng tin cậy hơn các hệ số LPC. trong đó Q.

1. Đặt trọng số cho các hệ số cepstral

Do tính nhạy cảm của các hệ số cepstral bậc thấp đối với sườn phổ tổng thể và do tính nhạy cảm cua hệ số cepstral bậc cao đối với nhiễu (và các dạng biến đổi giống nhiễu khác), nó trở thành kỹ thuật chuẩn để định trọng các hệ số cepstral nhờ một cửa sổ được làm hẹp sao cho giảm thiểu những nhạy cảm này. Chuyển sang ceptral có trọng số:

c’m = wm.cm với 1 ≤ m ≤ Q

(2.31)

(2.30)

Hàm trọng số thích hợp là bộ lộc thông dài (trong miền cepstral):

Wm = 1 + với 1 ≤ m ≤ Q

1. *Tính đạo hàm cepstral*



Với  là hằng số chuẩn và (2K+1) là số lượng Frame cần tính. K=3 là giá trị thích hợp để tính đạo hàm cấp.Vector đặc trưng của tín hiệu gồm Q hệ số cepstral va Q hệ số đạo hàm cepstral.

1. *Nhận xét phương pháp LPC*

Tóm lại, trong mô hình phân tích LPC trên, chúng ta cần phải đặ tả các tham số bao gồm:

* N: số mẫu trong mỗi Frame phân tích
* M: số mẫu cách nhau giữa 2 Frame kề nhau
* P: cấp phân tích LPC
* Q: số chiều của các vector cepstral dẫn xuất từ các hệ số LPC
* K: số frame được dùng tính các đạo hàm cepstral.

Tuy mỗi tham số đều có thể thay đổi trên dải rộng, bảng sau cho các giá trị đặc trưng đối với các hệ phân tích ở 3 tần số lấy mẫu khác nhau (6.67kHz, 8kHz và 10kHz).

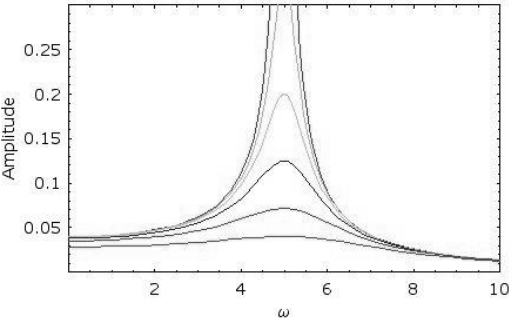
Bảng Các giá trị đặc trưng cho các tham số phân tích LPC.

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số | F1=6.67kHz | F1=8kHz | F1=10kHz |
| N | 300 (45ms) | 240 (30ms) | 300 (30ms) |
| M | 100 (15ms) | 80 (10ms) | 100 (10ms) |
| P | 8 | 10 | 10 |
| Q | 12 | 12 | 12 |
| K | 3 | 3 | 3 |

Mô hình LPC là mô hình thích hợp cho việc xử lý tín hiệu tiếng nói. Với miền tiếng nói hữu thanh có trạng thái gần ổn định, mô hình tất cả các điểm cực đại của LPC cho ta một xấp xỉ tốt đối với đường bao phổ cơ quan phát âm. Với miền tiếng nói vô thanh, mô hình LPC tỏ ra ít hữu hiệu hơn so với miền hữu thanh, nhưng nó vẫn là mô hình hữu ích cho các mục đích nhận dạng tiếng nói. Mô hình LPC đơn giản và dễ cài đặt trên phần cứng và phần mềm.

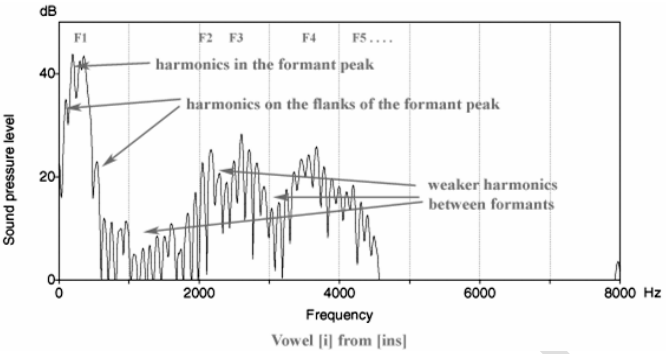
### Tìm hiểu về formant

Để hiểu rõ hơn về khái niệm Formant, trước tiên chúng ta cần hiểu khái niệm cộng hưởng là gì? Đó là một hiện tượng xảy ra trong dao động cưỡng bức khi một vật dao động được kích thích bởi một ngoại lực tuần hoàn có cùng tần số với giao động riêng của nó.



Hình. Minh hoạ hiện tượng cộng hưởng.

Formant là dải tần số được tăng cường do hiện tượng cộng hưởng trong ống dẫn thanh, đặc trưng cho âm sắc của mỗi nguyên âm. Trong mỗi dải tần như thế có một tần số được tăng cường hơn cả gọi là đỉnh formant (formant peak).



Hình 1*.* Minh hoạ Formant.

Formant là trong các thành phần quan trọng hình thành nên âm học, tạo ra sự đặc trưng cho từng âm thanh. Một nguyên âm do một người phát ra có nhiều formant:

F1: ứng với cộng hưởng vùng yết hầu.

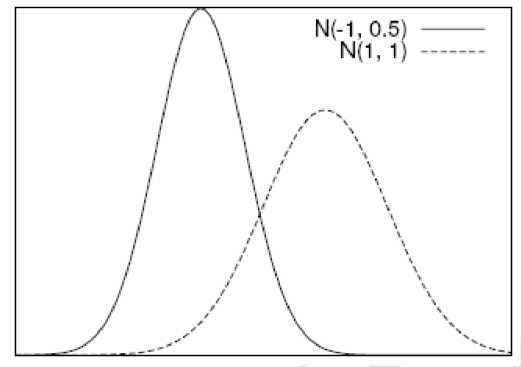
F2: ứng với cộng hưởng khoang miệng.

Khi ta nói, sẽ có các sự cộng hưởng khác ở khoang mũi tạo nên các formant khác (F3, F4, F5..) chính các formant này sẽ quy định các đặc trưng giọng nói riêng cho từng người vì khoang mũi của mỗi người là cố định, có cấu tạo tùy thuộc vào thể trạng của từng người. Ngoài ra còn có 2 nơi cộng hưởng khác là khoang miệng và yết hầu. Nhưng hộp cộng hưởng ở 2 nơi này có khả năng thay đổi linh động phụ thuộc vị trí của các cơ quan khác như môi, lưỡi, hàm. Khi 3 cơ quan này thay đổi vị trí sẽ tạo ra các hộp cộng hưởng khác nhau về thể tích, hình dáng, đường lưu thông của khí trong thanh quản,..Nên sẽ làm biến đổi các âm sắc, chính vì điều đó mà chúng ta thấy ở các lần nói khác nhau của một từ thì chưa chắc chúng được phát âm ra giống nhau. Nên trong lĩnh vực âm học, người ta tập trung nghiên cứu formant tạo ra ở 2 vùng này là F1, F2.

Formant được xác định dựa trên sự tập trung năng lượng âm học xung quanh một tần số đặc biệc trong dải sóng. Trong 1 dải sóng có thể xác định vài formant, mỗi formant có thể có cùng cường độ âm nhưng có tần số khác nhau.

## Gaussian Mixture Model

Mô hình Gauss (Gaussian Mixture Model - GMM) là một dạng mô hình thống kê được xây dựng từ việc huấn luyện các tham số thông qua dữ liệu học. Mô hình GMM còn có một số tên gọi khác như Weighted Normal Distribution Sums hay Radial Basis Function Approximations...

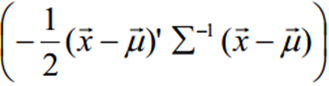


Hình Hàm mật độ Gauss.

Về cơ bản, mô hình GMM xấp xỉ một hàm mật độ xác suất bằng hợp các hàm mật độ Gauss. Hình phía dưới minh họa hai hàm mật độ Gauss với các tham số khác nhau. Một cách hình thức, hàm mật độ xác suất của phân phối Gauss fN(x,2) được cho bởi công thức:

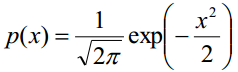
p(x) = 

(2.37)

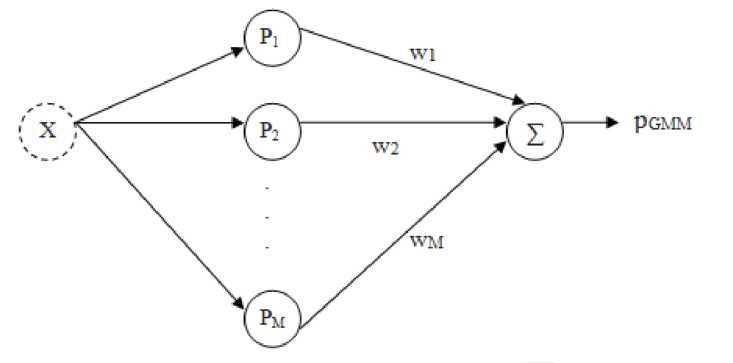
Trong đó, là giá trị trung bình,là độ lệch chuẩn. Trong trường hợp x là vector gồm D thành phần, hàm mật độ xác suất của phân phối Gauss fN (x, ,  được cho bởi công thức:

p() = exp

Khi đó,  là vector trung bình, là ma trận hiệp phương sai. Nếu chọn =0 và  =1, công thức (2.35) sẽ trở thành hàm mật độ chuẩn Gauss tiêu chuẩn:



Từ “Gauss” được đặt theo tên của nhà toán học người Đức Carl Friedrich Gauss. Ông đã định nghĩa hàm mật độ Gauss và áp dụng trong phân tích dữ liệu thiên văn.

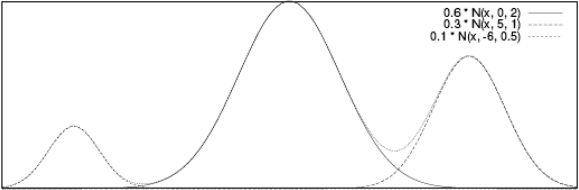


Hình Mô hình GMM.

Cho trước M phân phối Gauss có các hàm mật độ p1, p2, ..., pM, hàm mật độ xác suất của mô hình GMM được minh họa trong hình 2.20 chính là tổng trọng của M phân phối Gauss theo công thức:

pGMM() = ()

trong đó, wi là trọng số của phân phối Gauss thứ i, thỏa ràng buộc 0< wi <1 và . Các trọng số này thể hiện mức độ ảnh hưởng của mỗi phân phối Gauss đối với mô hình GMM. Như vậy, phân phối Gauss có phương sai và trọng số lớn bao nhiêu thì có mức độ ảnh hưởng lớn bấy nhiêu đối với kết xuất của mô hình. Hình 2.21 cho thấy mức độ ảnh hưởng của từng phân phối Gauss lên GMM.



Hình Hàm mật độ của GMM có 3 phân phối Gauss.

Như vậy, một mô hình GMM có M phân phối Gauss sẽ được đại diện bởi bộ tham số  = { wi, i, }, i . Tùy thuộc vào cách tổ chức của ma trận hiệp phương sai (covariance matrix), GMM có thể có một số biến thể khác nhau:

* Nodal covariance matrices GMM: mỗi phân phối Gauss trong GMM có một ma trận hiệp phương sai riêng.
* Grand covariance matrix GMM: mọi phân phối Gauss trong một GMM dùng chung một ma trận hiệp phương sai.
* Global covariance matrix GMM: mọi phân phối Gauss trong tất cả các GMM dùng chung một ma trận hiệp phương sai.

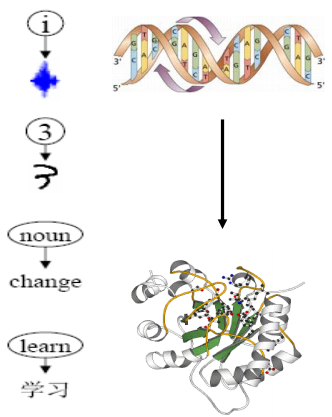
Ngoài ra, xét về dạng thức, ma trận hiệp phương sai gồm hai loại: full (dạng đầy đủ) và diagonal (dạng ma trận đường chéo). Thông thường, dạng nodal-diagonal covariance matrices GMM được sử dụng phổ biến nhất

## Mô hình Markov ẩn

### Giới thiệu

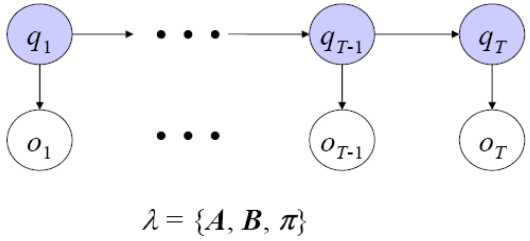
Mô hình Markov ẩn (tiếng Anh *Hidden Markov Model* – HMM) là mô hình thống kê trong đó hệ thống được mô hình hóa được cho là một quá trình Markov với các tham số không biết trước và nhiệm vụ là xác định các tham số ẩn từ các tham số quan sát được, dựa trên sự thừa nhận này. Các tham số của mô hình được rút ra sau đó có thể sử dụng để thực hiện các phân tích kế tiếp.

Các ứng dụng của mô hình Markov ẩn như nhận dạng giọng nói, nhận dạng chữ viết tay, xử lý ngôn ngữ thống kê, dịch máy, tin sinh học (khớp xấp xỉ nhiều chuỗi, tìm Motif, tìm kiếm tương tự).



***Hình Các ứng dụng của mô hình Markov ẩn.***

### Ba bài toán cơ bản của mô hình Markov ẩn



Bài toán 1 (Bài toán ước lượng): Cho trước chuỗi tín hiệu quan sát O = O1 O2 … OT và mô hình HMM được đại diện bởi bộ tham số λ = (π, A, B). Làm sao để tính toán một cách hiệu quả P(O|λ) – xác suất phát sinh O từ mô hình λ?

Bài toán 2 (Bài toán giải mã – nhận dạng): Cho trước chuỗi tín hiệu quan sát O = O1 O2 … OT và mô hình HMM đại diện bởi bộ tham số λ = (π, A, B). Cần tìm ra chuỗi trạng thái tối ưu nhất Q = q1 q2 … qT đã phát sinh ra O?

Bài toán 3 (Bài toán học): Cho trước chuỗi tín hiệu quan sát O = O1 O2 … OT. Làm thế nào để xác định các tham số mô hình λ = (π, A, B) sao cho cực đại hóa xác suất P(O|λ)? Đây chính là bài toán học hay còn gọi là huấn luyện mô hình. Bài toán này đem lại một khả năng rất quan trọng của HMM: khả năng mô hình hóa một đối tượng cụ thể trong thực tế, mô hình hóa dữ liệu học.

#### Bài toán 1 – Computing Likelihood

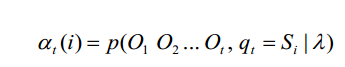
Mục tiêu của bài toán thứ nhất là tính p(O|λ) – xác suất phát sinh O từ mô

hình λ. Một giải pháp khả thi để tính p(O|λ) là thông qua thuật toán forwardbackward. Trước tiên, ta định nghĩa biến forward α t(j) là xác suất ở trạng thái

j tại thời điểm t và đã quan sát được đoạn O1, O

2, ..., Ot với mô hình λ cho

trước:



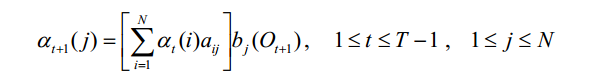
Các biến α t(i) có thể được tính theo qui nạp từng bước (hay thuật toán qui

hoạch động) như sau :

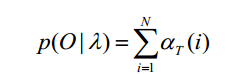
1. Khởi tạo:



1. Qui nạp:



1. Đầu ra:



Độ phức tạp tính toán của thuật toán forward là O(N2T).

Tương tự như trong thủ tục forward, thủ tục backward trước hết định nghĩa

biến backward βt(i) là xác suất quan sát được đoạn Ot+1, O

t+2, ..., O

T cho trước

trạng thái i ở thời điểm t và mô hình λ:



Các biến βt(i) cũng được tính theo qui nạp từng bước như sau:

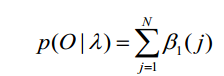
1. Khởi tạo:



2) Quy nạp :



3) Đầu ra:



Cũng giống như thuật toán forward, độ phức tạp của thuật toán backward là O(N2T). Như vậy, thủ tục forward-backward là khả thi với chi phí tính toán hoàn toàn có thể chấp nhận được.

Đối với việc tìm lời giải cho bài toán 1, ta chỉ cần đến phần forward trong thủ tục forward-backward. Tuy nhiên, phần backward giúp tìm lời giải cho bài toán 3.

#### Bài toán 2 – Decoding (Thuật toán Virterbi)

Mục tiêu của bài toán 2 là tìm ra chuỗi trạng thái “tối ưu” nhất Q = q1 q2 … qT đã phát sinh ra O. Một điều đáng lưu ý là có rất nhiều các tiêu chí “tối ưu” khác nhau cho việc xác định Q, nên lời giải cho bài toán này phụ thuộc vào tiêu chí ‘tối ưu” được chọn.

Thuật toán viterbi định nghĩa biến δt(i) là xác suất cao nhất của đoạn chuỗi trạng thái dẫn đến Si ở thời điểm t và đã quan sát được đoạn O1, O2, …, Ot cho trước mô hình λ:

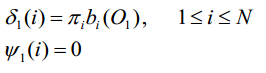


δt(i) có thể được tính theo qui nạp:

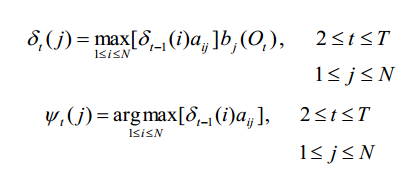


Tuy nhiên thuật toán viterbi còn cần đến mảng ψt(j) để lưu lại các tham số i làm cực đại hóa biểu thức trên, có thể hiểu tại thời điểm t trước j là i, nhằm mục đích truy vết (back trace). Chi tiết các bước của thuật toán viterbi như sau:

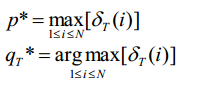
1. Khởi tạo:



1. Lặp qui nạp:



1. Kết thúc



1. quay lui – backtracking:



Kết thúc thuật toán, chuỗi Q = q1 \*q2\*..qT chính là lời giải thỏa đáng của bài toán 2

#### Bài toán 3 – Learning (Forward - Backward)

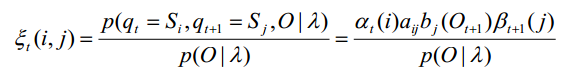
Mục tiêu của bài toán thứ 3, cũng là bài toán phức tạp nhất trong ba bài toán, là tìm cách cập nhật lại các tham số của mô hình λ = (π, A, B) sao cho cực đại hóa xác suất p(O|λ) – xác suất quan sát được chuỗi tín hiệu O từ mô hình. Với một chuỗi tín hiệu quan sát hữu hạn bất kỳ O làm dữ liệu huấn luyện, chưa có một phương pháp tối ưu nào cho việc ước lượng các tham số λ = (π, A, B) của mô hình theo hướng cực đại toàn cục. Tuy nhiên, bộ tham số λ có thể được chọn sao cho xác suất p(O|λ) đạt cực đại cục bộ bằng thuật toán Forward-Backward hoặc thuật toán Baum-Welch, hoặc thuật toán Expectation-Maximization (EM).Trước tiên, ta định nghĩa ξt(i,j) là xác suất ở trạng thái Si tại thời điểm t và rơi vào trạng thái Sj ở thời điểm t+1 cho trước mô hình λ và chuỗi tín hiệu quan sát O:

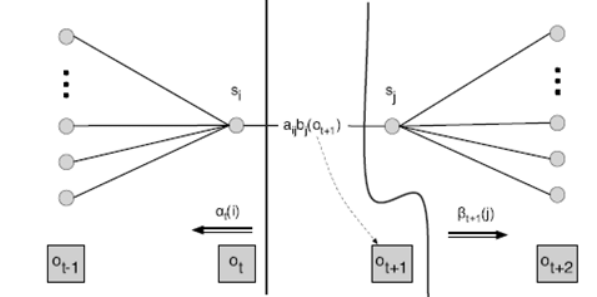


Áp dụng xác suất Bayes:



Suy ra:





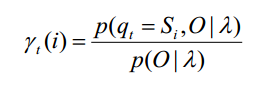


Gọi γt(i) là xác suất ở trạng thái Si vào thời điểm t cho trước chuỗi tín

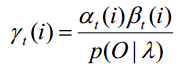
hiệu quan sát O và mô hình λ:

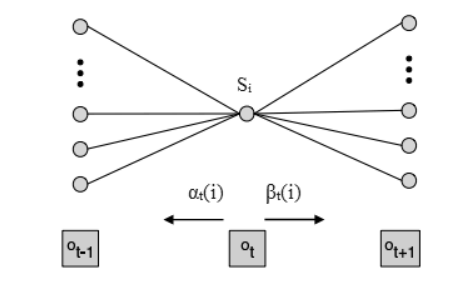


Áp dụng xác suất Bayes, suy ra:



Áp dụng xác suất forward và backward

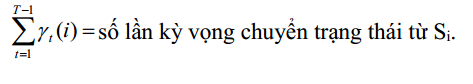


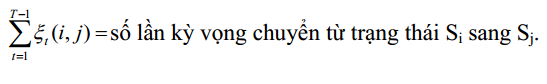




Nếu ta lấy tổng γt(i) theo t thuộc [1, T-1], kết quả nhận được là số lần kỳ

vọng chuyển trạng thái từ Si. Tương tự, lấy tổng ξt(i,j) theo t thuộc[1, T-1], ta sẽ có số lần kỳ vọng chuyển từ trạng thái Si sang Sj:

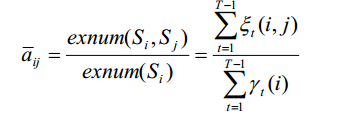


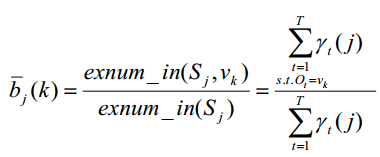


Với các đại lượng này, ta có các biểu thức cập nhật tham số của HMM

như sau:







exnum(Si,Sj): số lần kỳ vọng chuyển từ trạng thái Si sang trạng thái Sj exnum(Si): số lần kỳ vọng chuyển trạng thái từ Si.exnum\_in(Sj,vk): số lần kỳ vọng ở trạng thái Sj và quan sát được tín hiệu vk.exnum\_in(Sj): số lần kỳ vọng ở trạng thái Sj.Từ mô hình ban đầu λ = (A, B, π) và chuỗi tín hiệu quan sát O, ta tính được vế phải của các biểu thức trên, kết quả nhận được chính là các tham số mới của mô hình



Như vậy, thuật toán Baum-Welch sẽ được áp dụng qua nhiều bước lặp để ước lượng và cập nhật các tham số mô hình cho đến khi hội tụ. Tuy nhiên, kết quả cuối cùng chỉ đạt được tối ưu cục bộ mà thôi. Thông thường, nếu các tham số được khởi tạo với các giá trị thích hợp, thì có một phần khả năng nào đó có thể giúp mô hình đạt được tối ưu toàn cục khi huấn luyện.

## Mô hình HMM trong nhận dạng giọng nói

### Giới thiệu

### HMM trong nhận dạng giọng nói

### Các khái niệm và các thuật ngữ

# Công cụ huấn luyện và nhận dạng giọng nói

## Tổng quan

Huấn luyện và nhận dạng là hai công đoạn quan trọng trong một hệ thống nhận dạng giọng nói. Hiện nay có hai công cụ hỗ trợ cho việc này là HTK và CMU Sphinx.

Trích chọn   
đặc trưng

(Feature   
Extraction)

Mô hình huấn luyện

(Training Acoustic Model)

Bộ dữ liệu huấn luyện

(Training Data)

HMM

Bộ dữ liệu kiểm tra

(Testing Data)

So khớp

(Viterbi Decoder)

Trích chọn   
đặc trưng

(Feature   
Extraction)

Nhận dạng

***Hình 2.2.1.1. Mô hình tổng quát quá hình huấn luyện và nhận dạng giọng nói***

## Huấn luyện và nhận dạng giọng nói với CMU SPHINX

### Giới thiệu

Sphinx4 là một hệ thống được viết bằng ngôn ngữ lập trình Java. Nó được phát minh bởi nhóm nhà khoa học thuộc Trường Đại học Carnegie Mellon, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL) và Hewlett Packages (HP) cùng với sự đóng góp của Đại học California Santa Cruz (UCSC) và Viện Công nghệ Massachusetts (MIT).

Sphinx4 là hệ thống nhận dạng giọng nói được xây dựng với kỹ thuật cốt lõi là HMM. Sphinx4 cũng có thể được sử dụng để nhận dạng nhiều dạng ngôn ngữ khác nhau trên thế giới bằng cách thiết lập lại một số thông số của Sphinx4.

Sphinx4 được xem là phiên bản tiên tiến nhất của Sphinx. Nó vẫn giữ tất cả chức năng của phiên bản trước đó và phát triển thêm những tính năng mới.

Sphinx4 là một hệ nhận dạng giọng nói dựa trên mô hình HMM [19] được xem là phổ biến nhất bởi vì khả năng nhận dạng với độ chính xác cao ngay cả trong trường hợp nhận dạng với từ vựng lớn. Hơn nữa, Sphinx4 cũng còn có những điểm mạnh khác mà hiếm khi chúng ta tìm thấy trong những hệ nhận dạng khác. Từng bước, chúng ta sẽ phân tích lý do tại sao chúng ta nên sử dụng nó trong phần giới thiệu các thành phần chính của Sphinx4.

### Vận dụng mô hình HMM trong nhận dạng giọng nó với Sphinx4

Mô hình Markov ẩn (HMM) là một mô hình thống kê, trong đó hệ thống đang được mô hình giả định là một quá trình Markov với các tham số chưa biết và nhiệm vụ là xác định các tham số ẩn từ các tham số quan sát được [9].

Trong quá trình nhận dạng giọng nói, sau khi giọng nói được thu nó sẽ được chia thành các khung mà chúng ta cần xử lý để tạo ra các câu văn bản. Mỗi khung tượng trưng một trạng thái, nhóm các trạng thái tượng trưng cho âm vị và nhóm của nhiều âm vị được biểu diễn như một từ mà chúng ta cần nhận dạng.

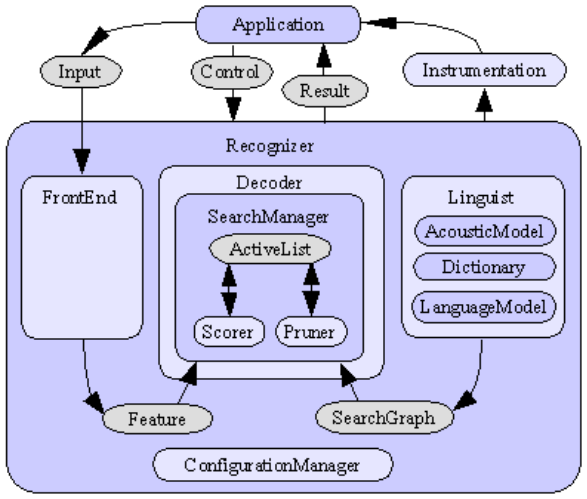
Trong cơ sở dữ liệu gọi là mô hình ngôn ngữ, chúng ta lưu trữ các giá trị tham chiếu của trạng thái, âm vị và từ để so sánh với các dữ liệu quan sát được (giọng nói).

Bằng cách sử dụng HMM, chúng ta sử dụng một mô hình thống kê trên mỗi âm vị. Và mục tiêu của hệ nhận dạng giọng nói là tìm ra dãy các trạng thái mà có xác suất cao nhất.

### Các thành phần chính trong CMU Sphinx

Có ba thành phần chính trong hệ thống nhận dạng Sphinx4. Đó là Frontend, Decoder và Linguist. Mỗi thành phần sẽ được mô tả sau. Trong luận văn này, chỉ tập trung giải thích thành phần Linguist bởi vì đây là thành phần cấp cao mà chúng ta có thể truy cập trực tiếp để thay đổi dữ liệu cần thiết cho việc xây dựng một ứng dụng mới.

Hai thành phần xử lý cốt lõi của Sphinx4 có thể kể đến là quá trình huấn luyện (training) và quá trình giải mã (decoding). Quá trình huấn luyện là một trong những bước cần thiết để xây dựng mô hình ngôn ngữ học (bao gồm mô hình ngôn ngữ và mô hình âm học). Ở bước này, chúng ta cần thu âm giọng nói của nhiều người làm cơ sở dữ liệu huấn luyện. Sau khi hệ thống được huấn luyện, nó có thể được sử dụng để nhận dạng giọng nói bằng cách giải mã giọng nói thành văn bản tương ứng. Quá trình nhận dạng được mô tả rất chi tiết trong biểu đồ dưới đây.



***Hình 1. 2.1 Kiến trúc hệ thống Sphinx4***

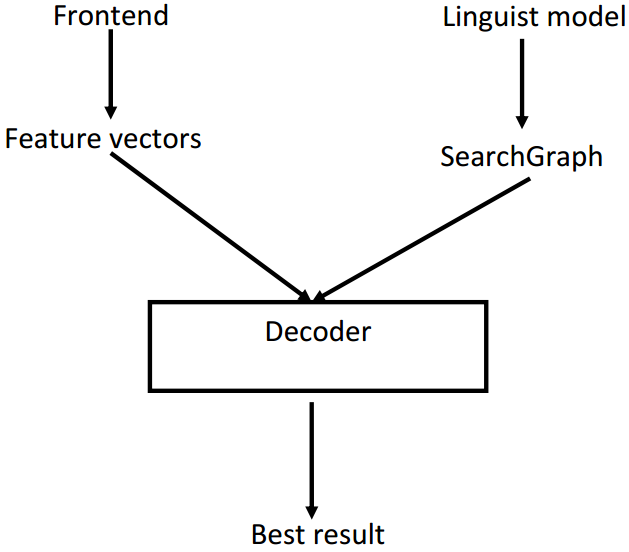
### Các xử lý chính trong CMU Sphinx

* Huấn luyện :

Quá trình huấn luyện được sử dụng để xây dựng dựng mô hình ngôn ngữ học (bao gồm mô hình ngôn ngữ và mô hình âm học) được coi là thành phần quan trọng nhất của một công cụ nhận dạng giọng nói xây dựng trên Sphinx4. Bước ban đầu của quá trình huấn luyện là định nghĩa một danh sách âm vị và từ điển cho mô hình. Sau đó, chúng ta sẽ thu âm giọng nói để làm dữ liệu huấn luyện (training data). Trong quá trình huấn luyện, giọng nói sẽ được chuyển thành dạng cepstral (những vector đặc trưng). Dựa trên những vector đặc trưng này, HMMs được xây dựng cho mỗi âm vị trong danh sách âm vị. Sau khi huấn luyện, chúng ta sẽ có một mô hình ngôn ngữ học được xây dựng trên HMM mà sẽ được sử dụng để nhận dạng giọng nói.

* Nhận dạng :

Nhiệm vụ chính của quá trình nhận dạng là tìm ra từ tốt nhất có thể mà sẽ phù hợp với giọng nói tương ứng. Giai đoạn đầu tiên của quá trình nhận dạng là giống với quá trình huấn luyện, giọng nói sẽ được thu âm và chuyển thành các vector đặc trưng bởi thành phần frontend. Mô hình ngôn ngữ học sẽ được sử dụng để định nghĩa từ cần giải mã. Tại bộ giãi mã, dựa vào thành phần Search graph và vector đặc trưng, nó sẽ áp dụng thuật toán tìm kiếm để tìm ra kết quả tốt nhất.



***Hình 1. 2.2 Quá trình nhận dạng.***

## Kết luận

Sphinx4 được coi là một framework tốt cho các sinh viên, các giáo sư sử dụng trong các nghiên cứu của họ. Sphinx4 cung cấp rất nhiều tính năng mà chúng ta không thể tìm thấy ở các framework khác.

### Ưu điểm

* Sphinx4 là một hệ thống nhận dạng tiếng nói liên tục
* Sphinx4 là một hệ thống có khả năng tùy chỉnh linh động
* Sphinx4 có thể thực hiện nhiều tác vụ nhận dạng
* Sphinx4 có thể làm việc độc lập với người nói
* Sphinx4 là hệ thống mã nguồn mở

### Nhược điểm

Sphinx4 là một hệ thống mã nguồn mở vì vậy nó không phải là hệ thống chuẩn mà có thể dễ dang đươc thực thi, nó bao gồm một loạt các gói mà người sử dụng cần phải cấu hình để chạy hệ thống.

Bên cạnh đó, Sphinx4 vẫn tồn tại một số lỗi mà chúng ta cần phải khắc phục nếu chúng ta muốn phát triển một công cụ nhận dạng giọng nói dựa trên Sphinx4. Bởi vì không có nhiều tài liệu hướng dẫn chi tiết về kiến trúc của Sphinx4 và cách mà hệ thống làm việc. Vì vậy chúng ta vẫn còn hiểu biết mơ hồ về nó và chính vì thế chúng ta có thể gặp phải những lỗi bất ngờ trong quá trình sử dụng.

## Sphinx4 với ngôn ngữ Tiếng Việt

### Tổng quan

Các hệ thống nhận dạng giọng nói yêu cầu hai loại tập tin tương ứng với hai mô hình để nhận dạng giọng nói. Thứ nhất là mô hình âm học (Acoustic Model), được tạo ra bằng cách ghi âm giọng nói kết hợp với phiên âm của chúng và sau đó biên dịch (compiling) để tạo thành một đại diện thống kê của âm thanh tạo nên mỗi từ. Thứ hai là mô hình ngôn ngữ (Language Model) hoặc tập tin Grammar. Một mô hình ngôn ngữ là một tập tin chứa các xác suất của chuỗi các từ. Một Grammar là một tập tin nhỏ hơn chứa một kết hợp của các từ đã được xác định trước.

Trong phần này, chúng tôi sẽ cung cấp một phương pháp xây dựng mô hình âm học cho tiếng Việt được sử dụng cho thành phần fontend của SphinxTrain.

### Corpus

Corpus là tập hợp các đơn vị âm thanh định nghĩa một từ cụ thể trong từ vựng. Đây là sự kết hợp giữa sự phát âm của âm thanh và từ điển.

Giọng nói

(Speech)

Danh sách âm vị

(Phone list)

Phát âm

(Audio utterances)

Thu âm (Recording)

Định nghĩa từ (Word definition)

Từ vựng

(Vocabulary)

Định nghĩa âm thanh

(Sound definition)

**Corpus**

Mô hình âm vị

(Acoustic model)

Huấn luyện (Training)

***Hình 2.2.4.1. Tổng quan về Corpus***

Corpus đóng một vai trò rất quan trọng trong hệ thống nhận dạng giọng nói. So với HMM, nó được coi là xương sống của hệ thống mà chúng ta có thể sửa đổi để cải thiện độ chính xác của hệ thống.

Trong Sphinx4, những người phát triển hệ thống nhận dạng cần phải quan tâm nhiều đến vai trò của corpus. Độ chính xác của nhận dạng thông thường tỷ lệ thuận với khả năng của corpus.

Để tạo một corpus, chúng ta có thể thu thập cách phát âm từ nhiều nguồn. Nó có thể từ tivi, tập tin âm thanh, hoặc ghi âm trực tiếp. Tuy nhiên, điều đó không có nghĩa rằng tất cả cách phát âm có thể được sử dụng để tạo ra các corpus, chỉ có những phát âm tốt sau khi được lọc có thể được sử dụng.

# Phân tích và thiết kế hệ thống

## Thiết lập môi trường CMU SPHINX

### Chuẩn bị hệ điều hành

Linux là môi trường hệ điều hành thích hợp nhất để cài đặt Sphinx và thực hiện huấn luyện. Trong các hệ điều hành Linux thì Ubuntu được xem như là hệ điều hành phổ biến. Sphinx dễ dàng được cài đặt trong hệ điều hành Ubuntu.

Nhóm đã sử dụng phiên bản Ubuntu 16.04.1 LTS để cho quá trình xây dựng bộ thư viện và cấu hình CMU SPHINX.

### Các gói thiết lập Sphinx

Các gói bao gồm:

* Pocketsphinx —bộ thư viện cấu hình cho Sphinx được viết bằng C. Mục đích là dành cho các ứng dụng đòi hỏi về tốc độ xử lý nhanh và nhẹ.
* Sphinxbase — gói thư viện nền, hỗ trợ các thư viện cần thiết cho các
* gói khác
* CMUclmtk — bộ công cụ xây dựng mô hình ngôn ngữ
* Sphinxtrain — bô công cụ huấn luyện mô hình ngữ âm

Các gói cài đặt có thể được tải trực tiếp từ trang chủ của CMU Sphinx

### Xây dựng bộ từ điển (Dictionary model)

Bộ từ điển bao gồm các ký tự hoặc các từ mong muốn chương trình nhận dạng. Do các công cụ huấn luyện của Sphinx chưa hỗ trợ tốt cho unicode nên các ký tự không thuộc bảng mã ASCII sẽ sử dụng phương pháp:

* Các ký tự không thuộc bảng mã ASCII sẽ được thay thế bằng kiểu gõ telex.
* Xây dựng bảng phiên âm tiếng Việt mức âm vị dưới dạng ASCII. (Tham khảo phụ lục).

Bảng phiên âm tiếng tiếng Việt mức âm vị được xây dựng dựa trên các tiêu chí:

* Biểu diễn được hết các âm vị có thể có trong tiếng Việt dưới dạng mã ASCII.
* Mọi âm vị đều được tổ hợp từ các ký hiệu sẵn có trên bàn phím để tiện lợi cho việc nhập liệu.
* Các thanh điệu được ký hiệu bằng các ký tự S, F, R, X, J và khoảng trắng.
* Thanh điệu được đặt cho các nguyên âm

Bộ từ điển được tổ chức như sau :

baajt b aajt : bật

car c ar : cả

casc c asc : các

quajt qu ajt : hòa

ngur ng ur : ngủ

ddefb dd efn : đèn

saan s aan : sân

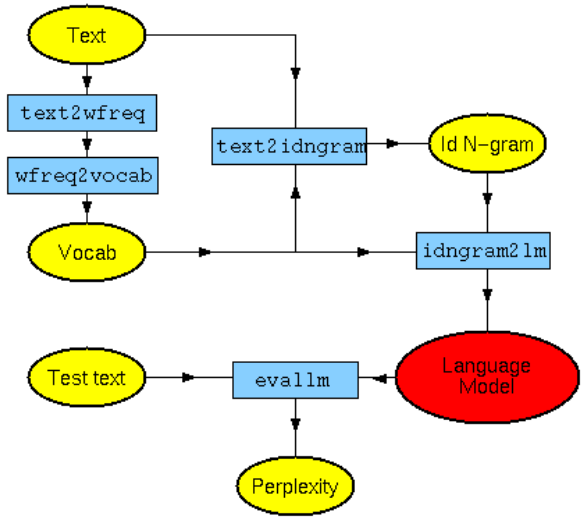
tawst t awst : tắt

ddefn dd efn : đèn

ddieefu dd ieefu

### Xây dựng bộ huấn luyện ngôn ngữ (Language model)

Quá trình tạo mô hình ngôn ngữ được mô tả như hình sau :



***Hình 1. 3.1 Sơ đồ quá trình tạo mô hình ngôn ngữ.***

Chuẩn bị tập tin văn bản chứa nội dung cần huấn luyện và đặt trong cặp thẻ <s></s>. Ví dụ tạo tập tin trainscription.txt có nội dung như sau:

<s> bật đèn và quạt ở phòng ngủ lầu một </s> (0001)

<s> tắt đèn và quạt ở phòng khách lầu hai </s> (0002)

<s> bật tất cả đèn ở sân lầu ba </s> (0003)

<s> tắt tất cả điều hòa ở bếp lầu bốn </s> (0004)

<s> bật quạt và đèn ở phòng ngủ lầu năm </s> (0005)

<s> tắt quạt và đèn ở phòng khách lầu sáu </s> (0006)

<s> bật tất cả các quạt và đèn ở bếp lầu bảy </s> (0007)

<s> tất tắt cả các quạt và đèn ở sân lầu tám trăm lẻ tám</s> (0008)

<s> bật điều hòa ở phòng ngủ lầu chín </s> (0009)

<s> tắt điều hòa ở phòng ngủ ba và phòng khách ba </s> (0010)

<s> baajt ddefn owr phofng ngur laafu moojt trawm nawm muwowi moost </s> (0011)

<s> tawst ddefn owr phofng ngur laafu moojt trawm nawm muwowi moost </s> (0012)

<s> baajt ddieefu hofa owr phofng khasch laafu hai trawm ba muwowi bary </s> (0013)

<s> tawst ddieefu hofa owr phofng ngur laafu chisn trawm muwowfi ba <<s> ghees </s>

<s> bi </s>

<s> sasch </s>

<s> gajch </s>

<s> xe </s>

<s> mefo </s>

<s> chuoojt </s>

<s> bosng </s>

<s> quaafn </s>

<s> bof </s>

<s> buwowsm </s>

<s> caso </s>

<s> cof </s>

<s> dieefu </s>

<s> eesch </s>

<s> gaf </s>

<s> hoor </s>

<s> nai </s>

<s> ong </s>

<s> rufa </s>

<s> saau </s>

<s> sosc </s>

<s> ve </s>

<s> vejt </s>

<s> voi </s>

<s> quaf </s>

<s> nguwja </s>

<s> hoa </s>

<s> bow </s>

<s> maajn </s>

<s> me </s>

<s> quyst </s>

<s> taso </s>

<s> bieern </s>

<s> boj </s>

<s> caay </s>

<s> chai </s>

<s> rawsn </s>

<s> roofng </s>

<s> ddaaju </s>

<s> desp </s>

<s> gajo </s>

<s> guoosc </s>

<s> goox </s>

<s> keso </s>

<s> las </s>

<s> lee </s>

<s> luwju </s>

<s> thufng </s>

<s> troosng </s>

<s> tusi </s>

Vì Sphinx chưa hỗ trợ unicode, nên các ký tự không thuộc ASCII sẽ được chuyển thành kiểu gõ telex:

<s> baajt ddefn vaf quajt owr phofng ngur laafu moojt </s> (0001)

<s> tawst ddefn vaf quajt owr phofng khasch laafu hai </s> (0002)

<s> baajt taast car casc ddefn owr saan laafu ba </s> (0003)

<s> tawst taast car casc ddieefu hofa owr beesp laafu boosn </s> (0004)

<s> baajt quajt vaf ddefn owr phofng ngur laafu nawm </s> (0005)

<s> tawst quajt vaf ddefn owr phofng khasch laafu sasu </s> (0006)

<s> baajt taast car casc quajt vaf ddefn owr beesp laafu bary </s> (0007)

<s> tawst taast car casc quajt vaf ddefn owr saan laafu tasm trawm ler tasm </s> (0008)

<s> baajt ddieefu hofa owr phofng ngur laafu chisn </s> (0009)

<s> ghees </s>

<s> bi </s>

<s> sasch </s>

<s> gajch </s>

<s> xe </s>

<s> mefo </s>

<s> chuoojt </s>

<s> bosng </s>

<s> quaafn </s>

<s> bof </s>

<s> buwowsm </s>

<s> caso </s>

<s> cof </s>

<s> dieefu </s>

<s> eesch </s>

<s> gaf </s>

<s> hoor </s>

<s> nai </s>

<s> ong </s>

<s> rufa </s>

<s> saau </s>

<s> sosc </s>

<s> ve </s>

<s> vejt </s>

<s> voi </s>

<s> quaf </s>

<s> nguwja </s>

<s> hoa </s>

<s> bow </s>

<s> maajn </s>

<s> me </s>

<s> quyst </s>

<s> taso </s>

<s> bieern </s>

<s> boj </s>

<s> caay </s>

<s> chai </s>

<s> rawsn </s>

<s> roofng </s>

<s> ddaaju </s>

<s> desp </s>

<s> gajo </s>

<s> guoosc </s>

<s> goox </s>

<s> keso </s>

<s> las </s>

<s> lee </s>

<s> luwju </s>

<s> thufng </s>

<s> troosng </s>

<s> tusi </s>…

### Xây dựng mô hình âm học (Aucostic model)

* Giới thiệu :

Mô hình âm học bao gồm một biểu diễn thống kê các âm thanh riêng biệt tạo nên mỗi từ trong mô hình ngôn ngữ hay bộ ngữ pháp. Quá trình huấn luyện mô hình âm học được sử dụng bằng công cụ sphinxtrain.

* Chuẩn bị dữ liệu :

Tất cả quá trình này được thực hiện trên môi trường Ubuntu.

Tạo thư mục dùng để huấn luyện mang tên beta, trong đó có 2 thư mục con là etc và wav.

Thư mục etc chứa các tập tin như sau:

beta.dic - bộ tự điển âm vị, âm tiết

beta. phone - tập tin chứa danh sách các âm vị

beta.lm.DMP - Mô hình ngôn ngữ

beta.filler - Danh sách các khoảng lặng

beta \_train.fileids - Danh sách các tập tin huấn luyện

beta \_train.transcription - Dữ liệu dạng văn bản của tập tin huấn luyện

beta \_test.fileids - Danh sách các tập tin test

beta \_test.transcription – Dữ liệu dạng văn bản của tập tin test

Tạo 2 thư mục con là train và test trong thư mục wav, trong đó thư mục train chứa dữ liệu dùng để huấn luyện, thư mục test chứa dữ liệu dùng để test.

Thư mục wav dùng để chứa các file.wav sẽ được sử dụng trong quá trình xây dựng Acoustic.model.

Thư mục test chứa các file âm thanh .wav được sử dụng cho quá trình test (kiểm tra) để đánh giá mức độ chính xác.

wav

| train

| từ\_huấn luyện\_1.wav – tập tin thu âm lần 1 của người huấn luyện dành cho dòng thứ nhất trong file transcription.txt của tập tin văn bản .

| từ\_huấn luyện\_2.wav – tập tin thu âm lần 2 của người huấn luyện dành cho dòng thứ hai trong file transcription.txt của tập tin văn bản.

| …

| từ\_huấn\_luyện\_n.wav – tập tin thu âm lần n của người huấn luyện dành cho dòng thứ n trong file transcription.txt của tập tin văn bản.

| từ\_huấn\_luyện\_n+1.wav – tập tin thu âm lần n+1 của người huấn luyện thứ hai dành cho dòng thứ n+1 trong file transcription.txt của tập tin văn bản.

| …

| từ\_huấn\_luyện\_n+m.wav tập tin thu âm lần n+m của người huấn luyện thứ hai dành cho dòng thứ n+m trong file transcription.txt của tập tin văn bản từ\_huấn\_luyện\_n+n +1.wav – tập tin thu âm lần n+m+1 của người huấn luyện thứ 3 dành cho dòng thứ n+m+1 trong file transcription.txt của tập tin văn bản.

| …

| test

| từ\_nhận\_dạng\_1.wav – tập tin thu âm lần 1 của người huấn luyện dành cho dòng thứ nhất trong file transcription\_test.txt của tập tin văn bản .

| từ\_nhận\_dạng\_2.wav – tập tin thu âm lần 2 của người huấn luyện dành cho dòng thứ nhất trong file transcription\_test.txt của tập tin văn bản .

| …

| từ\_nhận\_dạng\_n1.wav – tập tin thu âm lần 1 của từ cần nhận dạng dành cho từ thứ hai.

| từ\_nhận\_dạng\_n2.wav – tập tin thu âm lần 2 của từ cần nhận dạng dành cho từ thứ hai.

| …

**Tập tin beta.dic**

Tập tin này là tập tin từ điển đã chuẩn bị từ đầu. Nó chứa nội dung về cách phát âm của một từ trong bộ huấn luyện. Mỗi một dòng trong tập tin là định nghĩa cách đọc của một từ.

Trong tiếng Việt, cách đọc và cách viết một từ gần như gắng liền với nhau. Không cần có hướng dẫn cách đọc khi học tiếng Việt, trong tiếng Anh cách đọc và cách viết không phụ thuộc nhau, ví dụ “lead” (dẫn đầu) & “head” (cái đầu).

Để xây dựng tập tin này cho tiếng Việt, ta có thể định nghĩa các từ bằng nhiều cách như sau [3]:

NAM N A M

Với cách trên, ta xem từ “BAN” là một âm tiết với sự kết hợp của 3 âm vị là N, A, M.

NAM N AM

Với cách này, ta xem từ “BAN” là một âm tiết với sự kết hợp của 2 âm vị là N, AN.

Sphinx không hỗ trợ định nghĩa ở dạng word-base, nghĩa là cách đọc của một từ không được chính là từ đó. Ví dụ: NAM NAM là không được cho phép. Tuy nhiên có thể làm một phương pháp tương đương thay thế nếu muốn xây dựng theo kiểu word-base. Khi đó phải định nghĩa từ theo kiểu một từ có nhiều cách đọc, ví dụ: BAN BAN BANG

Ý nghĩa của dòng định nghĩa trên là từ “ban” có thể đọc theo hai cách là “ban” (cách đọc đúng chuẩn) hoặc đọc là “bang” (cách đọc người miền Nam).

Lưu ý, chỉ được dùng các ký hiệu a-z, A-Z, 0-9 để đảm bảo không gây lỗi cho tập tin này.

**Vấn đề thanh điệu**

Chúng ta sẽ xem các âm vị đi chung với thanh điệu sẽ là một âm vị độc lập. Khi đó thay vì xem thanh điệu như một âm vị khác theo cách định nghĩa sau (định nghĩa cho từ “nám”):

NASM N ASM

Ta sẽ xem âm ả là một âm vị khác, độc lập với âm a khi đó ta định nghĩa như sau:

NASM N AS M

**Tập tin beta.phone**

Tập tin này chứa tất các các âm vị (phiên âm) sử dụng trong tập tin trên, mỗi một dòng là một âm vị, nên sắp xếp các âm vị đó theo thứ tự để Sphinx dễ quản lý. Lưu ý thêm một âm vị đặc biệt vào tập tin này đó là SIL. Đây là âm vị đại diện cho khoảng lặng.

**Tập tin beta.lm.DMP**

Tập tin này là mô hình ngôn ngữ thống kê được xây dựng từ trước bằng công cụ CMUclmk, định dạng ARPA hoặc DMP.

**Tập tin beta.filler**

Tập tin này chứa các âm tiết dùng để “làm đầy”, thông thường là các khoảng lặng, được định nghĩa như sau:

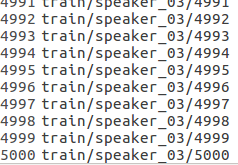
<s> SIL

</s> SIL

<sil> SIL

**Tập tin beta\_train.fileids**

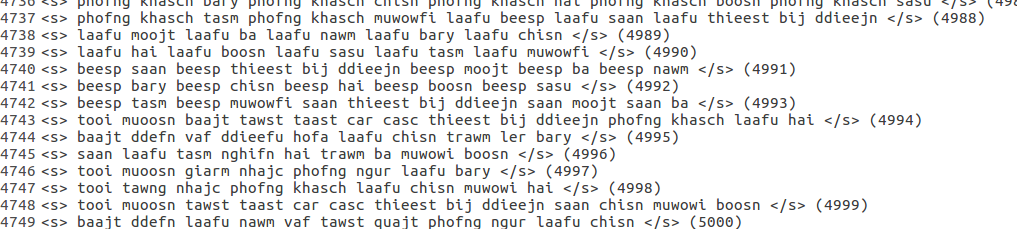
Tập tin này là tập tin liệt kê đường dẫn đến các tập tin ghi âm đuôi .wav, trên mỗi dòng tương ứng với một file thu âm trong thư mục và có trong sơ đồ thư mục trình bày phía trên. Lưu ý,có thêm phần .wav và mỗi dòng là một tập tin.



***Hình 1. 3.2 Ví dụ một phần của file beta\_train.fileids***

**Tập tin beta\_train.transcription**

Đây là phần nội dung mà tập tin âm thanh (.wav) đã thu âm được. Để huấn luyện cho Sphinx hiểu những gì chúng ta nói, cần cung cấp một tập tin dạng văn bản để giúp cho Sphinx hiểu và học từ đó. Cấu trúc một tập tin .transcript gồm nhiều dòng, mỗi một dòng là nội dung của một tập tin .wav kèm theo tên tập tin .wav đó.



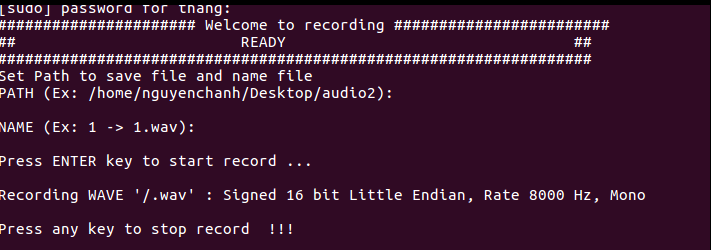
***Hình 1. 3.3 Ví dụ một phần của file beta\_train.transcription***

**Thiết lập định dạng âm thanh huấn luyện**

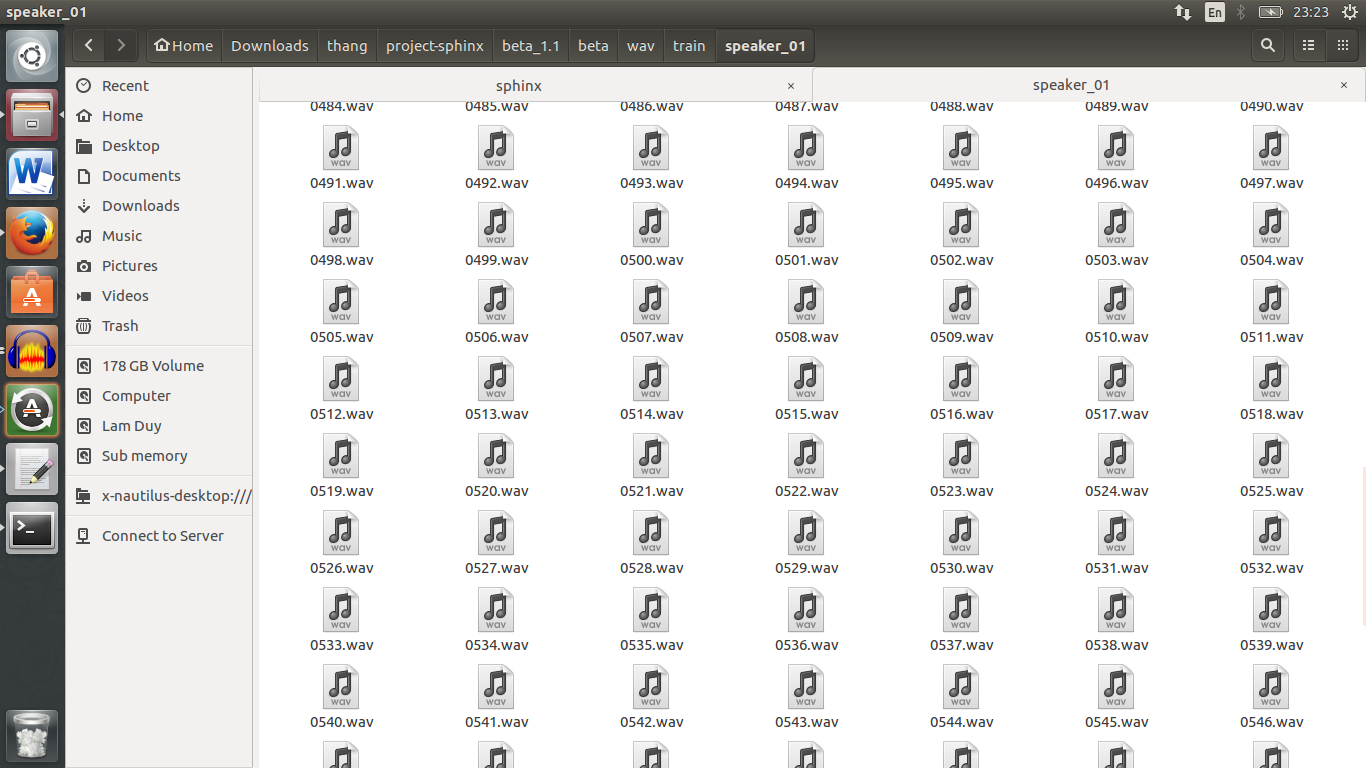
Dùng các chương trình ghi âm ***Audacity*** để ghi âm những câu đơn cho đến các câu phức tạp cần huấn luyện. Âm thanh được ghi vào với các thông số sau :

* Default Sample Rate Format: 8000Hz
* Default Sample Format: 16-bit
* Channels: 1(Mono)
* File Format: wav

Vì số lượng câu cần thu âm quá lớn, nên việc sử dụng các tool công cụ có sẵng để build chương tình sẽ gây nên sự bất tiện, do đó nhóm em đã quyết định tự viết một chương trinh riêng để phục vụ cho việc thu âm với số lượng âm thu sẽ rất lớn. Chương trình chạy trên nền tảng Ubuntu sẽ hỗ trợ thu âm và lưu file lần lượt chỉ cần qua thao tác nhấn Enter, các âm được thu sẽ được đặt tên theo số và có thứ tự liên tiếp nhau.Chương trình có sử dụng thư viện của ***Audacity*** và các âm thu được sẽ set theo các định dạng như ***Audacity.***



***Hình 1. 3.4 Giao diện của chương trình thu âm.***



***Hình 1. 3.3 Thiết lập định dạng lưu tập tin.***

### Cấu hình Sphinx

### Cấu hình thư mục huấn luyện

Sau khi đã cài đặt các gói cần thiết trong Ubuntu, chúng ta chép thư mục **beta** đã tạo ở bước trên vào cùng thư mục chứa thư mục **sphinx** đã tạo trước đó [10].

Trước khi bắt đầu quá trình huấn luyện, chúng ta sử dụng các lệnh của sphinxtrain và pocketsphinx để cấu hình thư mục huấn luyện:

*sphinxtrain -t beta setup*

Trong đó **beta** là tên của thư mục huấn luyện. Lệnh trên sẽ sao chép hai tập tin cấu hình cần thiết vào thư mục etc của thư mục huấn luyện:

**Tập tin feat.params**

-nfilt \_\_CFG\_NUM\_FILT\_\_

-lowerf \_\_CFG\_LO\_FILT\_\_

-upperf \_\_CFG\_HI\_FILT\_\_

-feat \_\_CFG\_FEATURE\_\_

-svspec \_\_CFG\_SVSPEC\_\_

-agc \_\_CFG\_AGC\_\_

-cmn \_\_CFG\_CMN\_\_

-varnorm \_\_CFG\_VARNORM\_\_

**Tập tin sphinx\_train.cfg**

# Configuration script for sphinx trainer

-\*-mode:Perl-\*-

$CFG\_VERBOSE = 1;

# Determines how much goes to the screen.

# These are filled in at configuration time

$CFG\_DB\_NAME = "beta";

# Experiment name, will be used to name model files and log files

$CFG\_EXPTNAME = "$CFG\_DB\_NAME";

# Directory containing SphinxTrain binaries

$CFG\_BASE\_DIR = "/home/user/beta";

$CFG\_SPHINXTRAIN\_DIR = "/usr/local/lib/sphinxtrain";

$CFG\_BIN\_DIR = "/usr/local/libexec/sphinxtrain";

$CFG\_SCRIPT\_DIR = "/usr/local/lib/sphinxtrain/scripts";

…

# Audio waveform and feature file information

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

$CFG\_WAVFILE\_EXTENSION = 'wav';

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

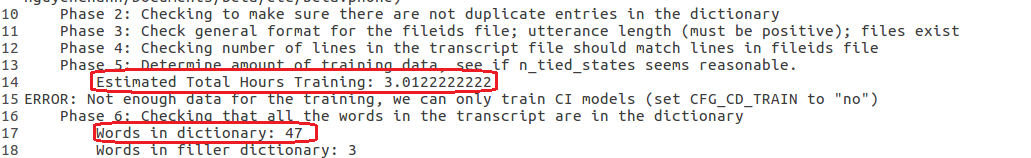
$CFG\_FEATFILES\_DIR = "$CFG\_BASE\_DIR/feat";

$CFG\_FEATFILE\_EXTENSION = 'mfc';

$CFG\_VECTOR\_LENGTH = 13;

### Chuẩn bị data cho quá trình huấn luyện

* Tập tin tư viện (beta.dic) : 47 từ
* Tập tin .Dump : được tổng hợp từ các bài báo tiếng việt.
* Thời lượng âm thu: Hơn 3 tiếng và được thu âm từ 10 người khác nhau. Trong đó, mỗi người sẽ được thu âm ở các môi trường khác nhau như: yên tĩnh, ồn ào…



***Hình 1. 3.5 File log tổng kết về quá trình huấn luyện data***

### Điều chỉnh tham số huấn luyện

Thông tin cấu hình nằm trong tập tin **sphinx\_train.cfg**. Một số cấu hình quan trọng:

Cấu hình định dạng của tập tin âm thanh dùng trong huấn luyện

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

$CFG\_WAVFILE\_EXTENSION = 'wav';

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

Điều chỉnh loại mô hình (huấn luyện HMM liên tục, bán liên tục), bỏ dấu # trước mô hình cần huấn luyện:

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

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

#$CFG\_HMM\_TYPE = '.ptm.'; # PocketSphinx (larger data sets)

Cấu hình tham số mật độ CFG có thể nhận các giá trị 4, 8, 16, 32, 64 tùy theo độ lớn của dữ liệu:

elsif ($CFG\_HMM\_TYPE eq '.cont.') {

$CFG\_DIRLABEL = 'cont';

# Single stream features - Sphinx 3

$CFG\_FEATURE = "1s\_c\_d\_dd";

$CFG\_NUM\_STREAMS = 1;

$CFG\_INITIAL\_NUM\_DENSITIES = 1;

$CFG\_FINAL\_NUM\_DENSITIES = 8;

die "The initial has to be less than the final number of densities"

if ($CFG\_INITIAL\_NUM\_DENSITIES > $CFG\_FINAL\_NUM\_DENSITIES);

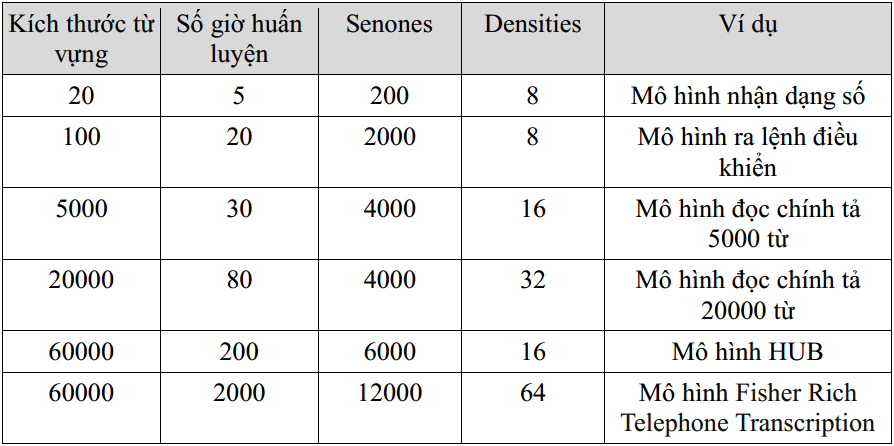
}

Cấu hình số lượng các senone để huấn luyện trong một mô hình. Số lượng senone càng lớn, sphinx phân biệt các âm càng chính xác. Nhưng mặt khác, nếu chúng ta có quá nhiều senone, mô hình sẽ không đủ tổng quát để nhận dạng các tiếng nói vô hình. Nghĩa là số từ lỗi sẽ tăng cao trên dữ liệu chưa huấn luyện. Đó là lý do quan trọng để không nên huấn luyện quá mức các mô hình. Trong trường hợp có quá nhiều senone mô hình sẽ phát sinh cảnh báo lỗi.

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

$CFG\_N\_TIED\_STATES = 200;

Theo nghiên cứu của nhóm CMUSphinx thì cấu hình dựa theo bảng sau:



Hình 1. 3.6 Bảng thống kê cấu hình

### Thực thi huấn luyện

Để bắt đầu quá trình huấn luận, chúng ta thực hiện chuyển đến thư mục huấn luyện (thư mục beta)

Gõ lệnh huấn luyện: **sphinxtrain run**

Lệnh trên sẽ duyệt qua các phần yêu cầu, phát sinh các thư mục feat, logdir, model\_parameters, model\_architecture, result, trees và qmanager bên trong thư mục beta.

Trong các giai đoạn, giai đoạn quan trọng nhất là kiểm tra tất cả mọi thứ được cấu hình đúng và dữ liệu đầu vào của chúng ta là phù hợp. Đừng bỏ qua các thông báo lỗi trong quá trình huấn luyện. Quá trình huấn luyện sẽ xuất ra các thông báo dạng:

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

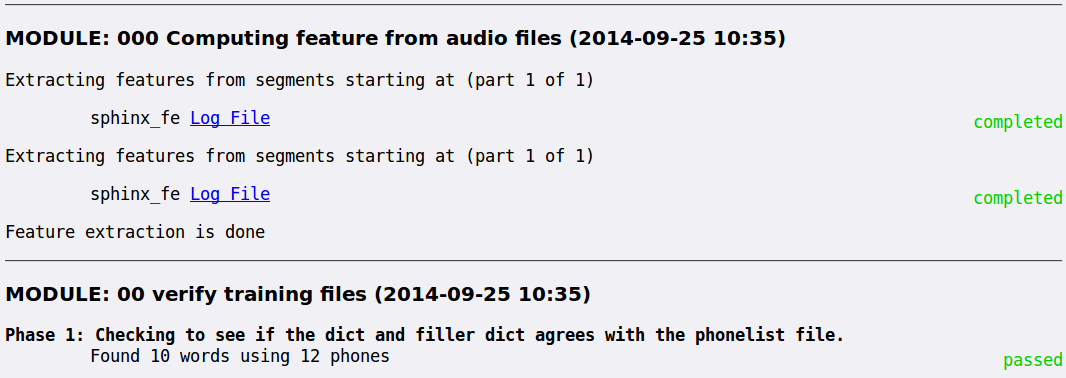
Convergence Ratio = 0.633864444461992

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

Kết thúc quá trình huấn luyện, chúng ta thu được tập tin beta.html là tập tin ghi nhận quá trình huấn luyện.



***Hình 1. 3.7 Thông báo trạng thái trong quá trình huấn luyện***

Kết thúc quá trình huấn luyện, mô hình âm học (acoustic model) được lưu trong thư mục **model\_parameters/beta.cd\_cont\_200**, bao gồm các tập tin sau:

mdef

feat.params

mixture\_weights

means

noisedict

transition\_matrices

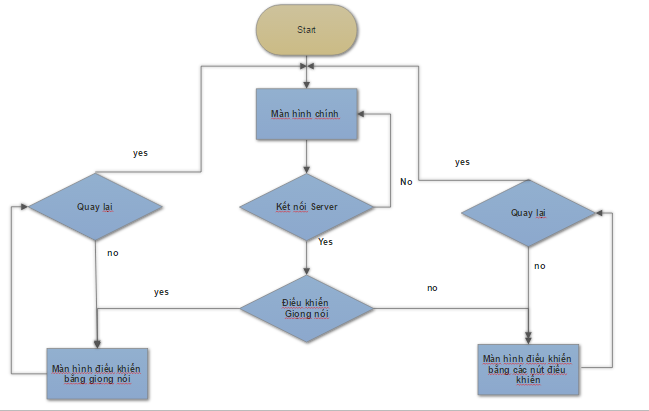
variances

## Xây dựng ứng dụng trên điện thoại Android

### Giới thiệu ứng dụng

Hiện tại trên thi trường Nhà Thông Minh, điều khiển các thiết bị bằng giọng nói giường như là một trong những yếu tố cần thiết và bắt buộc phải đó. Ứng dụng Smart Home của chúng em được ra đời và đáp ứng theo đúng nhu cầu đó. Ứng dụng Smart Home sẽ giúp cho người dùng vừa có thể tương tác bằng giọng nói hoặc bằng các thao tác để điều khiển thiết bị. Và ứng dụng còn có thể hỗ trợ cho người dùng cập nhật thêm thiết bị mới theo nhu cầu của người dùng

### Cấu trúc ứng dụng



Tổng quang các phần của màn hình:

+ Màn hình chính: Sẽ là màn hình bao gồm hai nút bấm giúp người dùng chọn lựa giữa việc điều khiển thiết bị trong gia đình bằng phương thức giọng nói hay nút điều khiển

+ Màn hình điều khiển bằng giọng nói: tại đây, chương trình nhận diện giọng nói sẽ được kích hoạt.

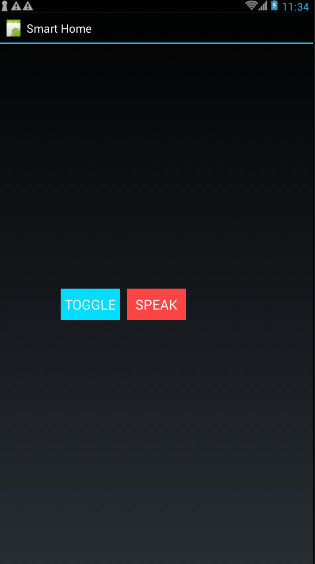
+ Màn hình điều khiển bằng nút điều khiển: danh sách cuộn hiển thị các thiết bị được sử dụng nhiều, giúp cho người dùng trong việc chọn lựa bằng việt bấm trượt trên Scroll View này và ấn vào button của đối tượng nào mong muốn điều khiển.

Sau khi các phương thức nhận diện được hoàn tất, hệ thống sẽ gởi đoạn dữ liệu vừa nhận diện đó lên server.

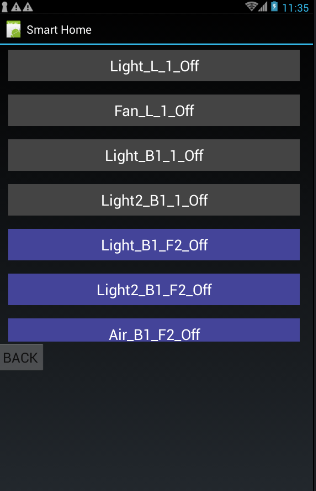
Người dùng có thể quay lại màn hình chính bằng nút “Back”.

### Hình ảnh demo

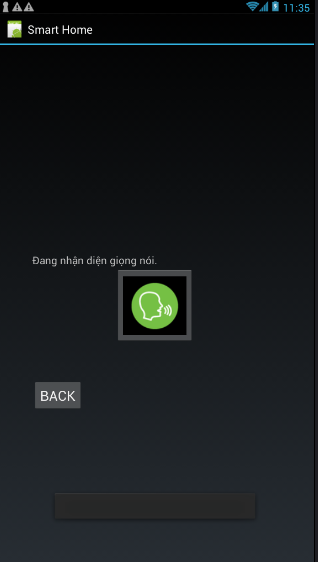
Sau đây hình ảnh demo ứng dụng



***Hình 1. 3.8 Màn hình giao diện đầu vào của ứng dụng***



***Hình 1. 3.9 Màn hình cuộn, điều khiển qua các Button***



***Hình 1. 3.10 Màn hình nhận diện giọng nói để điều khiển thiết bị***

## Board node

### Sơ đồ tổng quan

### Sơ đồ mạch

### Các thông số kỹ thuật của board node

### Sản phẩm demo

# Thực nghiệm

## Giá thành sản phẩm

## Tỉ lệ nhận diện giọng nói

## Tính ổn định của hệ thống

## Tính linh hoạt/nâng cấp của hệ thống

# Kết luận và kiến nghị

## Kết luận

### Các kết quả đạt được

### Các kết quả chưa đạt được

## Kiến nghị

## Hướng phát triển

# TÀI LIỆU THAM KHẢO