**CHƯƠNG 1. TỔNG QUAN**

**1.1. Giới thiệu đề tài**

Đề tài này mang tên “NGHIÊN CỨU CÔNG NGHỆ NHẬN DẠNG GIỌNG NÓI VỚI MÔ HÌNH HIDDEN MARKOV MODEL VÀ ỨNG DỤNG TRONG VIỆC ĐIỀU KHIỂN TRÌNH DUYỆT”. Để thực hiện được việc nhận dạng giọng nói, ta phải xây dựng một hệ thống gọi là Hệ thống nhân dạng tiếng nói tự động (Automatic Speech Recognition -ASR), đây là hệ thống chuyển đổi chuỗi âm thanh tiếng nói thành chuỗi từ. Việc xây dựng một hệ nhận dạng tiếng nói không phải là một công việc đơn giản, đòi hỏi nhóm phát triển phải am hiểu các kỹ thuật, lý thuyết từ nhiều kiến thức khác nhau như: âm học - vật lý, ngữ âm học, ngôn ngữ học, lý thuyết xác suất thống kê, máy học, trí tuệ nhân tạo, ... Trên thế giới, nhiều nhóm nghiên cứu đã phát triển thành công hệ nhận dạng tiếng nói cho các ngôn ngữ lớn như: tiếng Anh, tiếng Trung Quốc, tiếng Nhật, ... nhưng giải pháp nhận dạng cho tiếng Việt vẫn còn nhiều mặt hạn chế.

**1.2. Các nghiên cứu có liên quan đến khóa luận**

**1.2.1. Trên thế giới**

Giao tiếp người-máy là một lĩnh vực nghiên cứu lớn và khó nhưng lại có nhiều ứng dụng thực tiễn. Tiếng nói là một phương tiện giao tiếp tự nhiên nhất của con người và vì vậy, nghiên cứu để máy tính có thể hiểu tiếng nói của con người, hay còn gọi là nhận dạng tiếng nói tự động (Automatic Speech Recognition –ASR), đã trải qua quá trình 70 năm phát triển. Những nỗ lực nghiên cứu đầu tiên về ASR đã được tiến hành trong thập niên 50 với ý tưởng chính là dựa trên ngữ âm. Do kĩ thuật xử lí tín hiệu số cũng như khả năng máy tính còn giới hạn, các hệ thống nhận dạng lúc đó chỉ tập trung khai thác đặc trưng phổ cộng hưởng (spectral resonances) đối với các nguyên âm của tín hiệu, sau khi đi qua các bộ lọc tương tự. Trong giai đoạn này, có các hệ thống đáng chú ý như: hệ thống nhận dạng ký số rời rạc của Bell-lab (1952), bộ nhận dạng 13 âm vị của trường đại học College–Anh (1958) [1, p. 8].

Trong thập kỉ 1960, điểm đáng ghi nhận nhất là ý tưởng của tác giả người Nga, Vintsyuk khi ông đề xuất phương pháp nhận dạng tiếng nói dựa trên qui hoạch động theo thời gian (Dynamic Time Warping –DTW) [2, p. 1]. Đáng tiếc là mãi đến những năm 1980, phương pháp này mới được thế giới biết đến. Cuối những năm 1960, Reddy ở trường đại học CMU (Mỹ) đã đề xuất những ý tưởng đầu tiên về nhận dạng tiếng nói liên tục bằng kĩ thuật đánh dấu đường đi và truy vết lùi tìm kết quả [2, p. 2].

Đến những năm 70, nghiên cứu về nhận dạng tiếng nói đã bước đầu thu được các kết quả khích lệ, làm nền tảng cho những phát triển sau này. Trước tiên là bài toán nhận dạng từ rời rạc được giải quyết dựa trên ý tưởng của các nhà khoa học người Nga và Nhật. Velichko và Zagoruyko (ở Nga) là những người đi tiên phong trong việc áp dụng ý tưởng về phân lớp mẫu cho ASR. Sakoe và Chiba (Nhật) đề xuất các kĩ thuật sử dụng phương pháp qui hoạch động. Và Itakura, trong thời gian ở Bell-lab, đã đưa ra phương pháp mã hoá dự báo tuyến tính (Linear Predictive Coding –LPC) làm tiền đề cho việc áp dụng các tham số phổ LPC vào ASR. Các hệ thống ASR đáng chú ý của giai đoạn này gồm: Harpy và Hearsay-II của trường đại học CMU-Mỹ, hệ thống HWIM của BBN... [2, p. 2]

Nghiên cứu về ASR trong thập kỉ 80 đánh dấu phép dịch chuyển trong phương pháp luận: từ cách tiếp cận đối sánh mẫu sang cách tiếp cận sử dụng mô hình thống kê.

Ngày nay, hầu hết các hệ thống ASR đều dựa trên mô hình thống kê được phát triển ở thập kỉ này, cùng với những cải tiến ở thập kỉ 90. Một trong những phát minh quan trọng nhất ở thập kỉ 80 là mô hình Markov ẩn (Hidden Markov Model –HMM). Mặc dù HMM được áp dụng thành công ở một số phòng LAB (chủ yếu là IBM và viện nghiên cứu thuộc bộ Quốc phòng Mỹ) nhưng phải đợi đến vài năm sau đó, mô hình này mới được xuất bản và phổ biến trên thế giới. Ngoài ra, hai đề xuất quan trọng khác của giai đoạn này là tổ hợp Cepstrum và mô hình ngôn ngữ:

* Furui đề xuất sử dụng tổ hợp của các hệ số phổ cùng vớI đạo hàm bậc 1 và bậc 2 của chúng, như là những đặc trưng cơ sở cho ASR. Phương pháp này dù được đề xuất cuối những năm 70 nhưng đã không được áp dụng suốt một thời gian dài. Ngày nay, hầu hết các hệ thống nhận dạng tiếng nói đều sử dụng tổ hợp đặc trưng này. [2, p. 4]
* Những nhà khoa học thuộc công ty IBM là những người đi tiên phong trong việc phát triển mô hình ngôn ngữ (Language Model –LM). Đây là một công cụ hiệu quả trong việc lựa chọn chuỗi từ nhận dạng và đã được áp dụng thành công trong tất cả các hệ thống ASR ngày nay, đặc biệt là các hệ thống nhận dạng tiếng nói liên tục với bộ từ vựng lớn.

Các hệ thống ASR ra đời trong thời gian này có thể kể đến: hệ thống Sphinx của trường đại học CMU, Byblos của công ty BBN, Decipher của viện SRI, và các hệ thống khác của Lincoln Labs, MIT và AT&T Bell Labs.

Thập niên 90 ghi nhận một số kết quả nghiên cứu mới trong lĩnh vực phân lớp mẫu. Cụ thể, bài toán phân lớp theo mô hình thống kê (dựa trên luật quyết định Bayes), đòi hỏi phương pháp ước lượng các phân bố cho dữ liệu, được chuyển thành bài toán tối ưu, bao gồm phép cực tiểu lỗi phân lớp bằng thực nghiệm. Sự chuyển đổi này về mặt bản chất xuất phát từ ý tưởng sau đây: mục tiêu của phân lớp là cực tiểu lỗi chứ không phải cung cấp hàm phân bố phù hợp với dữ liệu nhận dạng. Khái niệm cực tiểu lỗi này đã làm nảy sinh một số kĩ thuật như phương pháp huấn luyện riêng biệt (Discriminative Training). Hai dạng điển hình của phép huấn luyện này là: CME (Minimum Classification Error) và MMI (Maximum Mutual Information). Kết quả thực nghiệm cho thấy các phương pháp huấn luyện mới đưa ra kết quả nhận dạng tốt hơn so với phương pháp huấn luyện cực đại xác suất trước đó. Ngoài ra, nhận dạng tiếng nói trong môi trường nhiễu cũng rất được quan tâm. Để nâng cao hiệu suất nhận dạng cho dữ liệu nhiễu, một số kỹ thuật đã được đề xuất như: MLLR (Maximum Likelihood Linear Regression), PMC (Parallel Model Combination). [2, p. 3]

Cuối cùng, các ứng dụng được phát triển trong giai đoạn này gồm: hệ thống trả lời thông tin tự động cho các chuyến bay (Air Travel Information Service –ATIS), hệ thống ghi lại các bản tin phát thanh (Broadcast News Transcription System).

Đến những năm đầu của thế kỷ 21, các nghiên cứu tập trung vào việc nâng cao kết quả nhận dạng tiếng nói, thông qua chương trình có tên gọi EARS (Effective Affordable Reusable Speech-to-Text) [2, p. 3]. Tới thời điểm này, tiếng nói được giả thiết là đã được thu âm trong môi trường bình thường, không bị ràng buộc bởi bất cứ điều kiện nào (các điều kiện giả thiết trước đây thường bao gồm tiếng nói được thu trong phòng sạch cách âm và do người bản xứ đọc). Đích hướng tới của chương trình này là khả năng nhận dạng, tóm tắt và chuyển ngữ các đoạn audio, giúp cho người đọc hiểu nhanh nội dung của chúng thay vì phải nghe toàn bộ.

Hiện nay, với các ngôn ngữ phổ biến như tiếng Anh, tiếng Pháp, tiếng Tây Ban Nha, các công trình nghiên cứu về nhận dạng tiếng nói đã thu được những kết quả rất tốt, có nhiều ứng dụng thực tiễn được triển khai như:

* Hệ thống giao tiếp hỏi đáp thông tin tự động qua điện thoại.
* Hệ thống truy vấn thông tin thoại.
* Hệ thống thông dịch tiếng nói xuyên ngữ tự động.
* Các trạm kiểm soát, hệ thống điều khiển sử dụng tiếng nói.
* Các ứng dụng tiếng nói trên thiết bị di động.

Về mặt kinh tế và thương mại, công nghệ nhận dạng tiếng nói đã thay đổi cách con người tương tác với hệ thống và thiết bị, không còn bó buộc trong cách thức tương tác truyền thống (như thông qua bàn phím của máy tính hay điện thoại) mà chuyển sang tương tác trực tiếp bằng giọng nói. Trong môi trường kinh tế cạnh tranh, các ứng dụng dần dần đã chuyển sang tích hợp tính năng tương tác âm thanh. Việc ứng dụng và khách hàng có thể tương tác với nhau thông qua âm thanh không có nghĩa là loại bỏ giao diện đồ họa truyền thống mà nó cung cấp thêm một cách truy cập thông tin và dịch vụ tiện lợi, tự nhiên hơn.

Về mặt nghiên cứu khoa học, các hệ thống nhận dạng tiếng nói hiện tại đều dựa trên phương pháp thống kê và so khớp mẫu. Phương pháp này đòi hỏi các tri thức về ngữ âm và một lượng lớn dữ liệu huấn luyện, bao gồm cả dạng âm thanh và dạng văn bản, để huấn luyện bộ nhận dạng. Lượng dữ liệu huấn luyện càng lớn, bộ dạng dạng càng có nhiều khả năng đưa ra kết quả chính xác hơn.

**1.2.2. Trong nước**

Tại Việt Nam, có 2 nhóm nghiên cứu chính về bài toán nhận dạng tiếng nói liên tục với bộ từ vựng lớn (LVCSR). Nhóm đầu tiên thuộc Viện Công nghệ Thông tin do PGS. Lương Chi Mai đứng đầu, với phương pháp ANN và công cụ CSLU [3] được sử dụng. Nhóm thứ hai thuộc trường Đại học Khoa học Tự nhiên thành phố Hồ Chí Minh do PGS. Vũ Hải Quân đứng đầu, với phương pháp HMM và công cụ HTK được sử dụng, các nghiên cứu của nhóm tập trung vào bài toán truy vấn thông tin tiếng Việt, nhận dạng tiếng nói, hệ thống giao tiếp giữa người và máy, tìm kiếm bằng giọng nói. Ngoài ra, gần đây có nghiên cứu của LIG (Laboratoire Informatique de Grenoble) hợp tác với phòng thí nghiệm MICA ở Hà Nội về sự khả chuyển của các mô hình ngữ âm (acoustic model portability).

Ở trong nước còn có các đề tài liên quan như: “Chương trình đọc chính tả”, sử dụng lượng hóa vector VQ, hạn chế về nhận dạng tiếng nói liên tục. “Phát triển các kết quả tổng hợp, nhận dạng câu lệnh, chuỗi số tiếng Việt liên tục trên môi trường điện thoại di động” [4], “Tăng cường độ chính xác của hệ thống mạng neuron nhận dạng tiếng Việt” [5], “Chương trình nhận dạng lệnh 10 chữ số liên tục qua điện thoại” của Viện công nghệ thông tin sử dụng công cụ CSLU [6], phương pháp mô hình Artificial neural network - ANN, giải mã bằng thuật toán Viterbi, cơ sở dữ liệu mẫu âm học của CSLU.

**1.3. Mục tiêu của khóa luận**

Mục tiêu chung nhất: tìm hiểu, vận dụng các kiến thức về nhận dạng tiếng nói đề xây dựng mộ chương trình nhận dạng tiếng nói tiếng Việt và ứng dụng trong điều khiển thiết bị mô phỏng và thiết bị thật.

Mục tiêu chi tiết:

1. Tìm hiểu các khái niệm có liên quan đến hệ nhận dạng tiếng nói để làm rõ  hơn một số yếu tố quan trọng trong việc sử dụng công cụ hỗ trợ.
2. Tìm hiểu phương pháp cài đặt công cụ hỗ trợ xây dựng hệ nhận dạng tiếng nói.
3. Tìm hiểu xây dựng mô hình âm học, mô hình ngôn ngữ thích hợp cho tiếng Việt.
4. Xây dựng chương trình mô phỏng, thực nghiệm, thử nghiệm giữa các mô hình với nhau từ đó đưa ra kết luận và nhận xét.

**1.4. Phạm vi**

Giải quyết bài toán nhận dạng các câu lệnh điều khiển cơ bản qua thực nghiệm (demo) đó là:

Điều khiển trình duyệt web Google Chrome bằng giọng nói với tổ hợp câu lệnh điều khiển gồm 45 từ.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Tổng Quan Về Âm Học Và Tiếng Nói**

**2.1.1. Âm học**

**2.1.1.1. Khái niệm**

Khi có nguồn phát ra âm thanh (như tiếng trống, tiếng nhạc cụ, tiếng nói), ta sẽ nghe và cảm nhận được âm thanh phát ra. Vật tạo ra được âm thanh còn được gọi là nguồn phát âm, âm thanh chính là sự dao động cơ của các thành phần vật chất trong một môi trường nào đó lan truyền và đến tai ta và khi đó ta cảm nhận được âm thanh. Trong môi trường không có vật chất tồn tại như chân không, không có dao động song cơ do đó cũng không có âm thanh tồn tại. Trong đời sống xã hội, âm thanh là phương tiện giao tiếp, truyền đạt thông tin phổ biến và xấu hiện từ lâu đởi nhất của con người. Khi nghiên cứu về âm thanh, người ta thường quan tâm đến 2 đặc điểm: đặc trưng vật lý và đặt trưng sinh học.

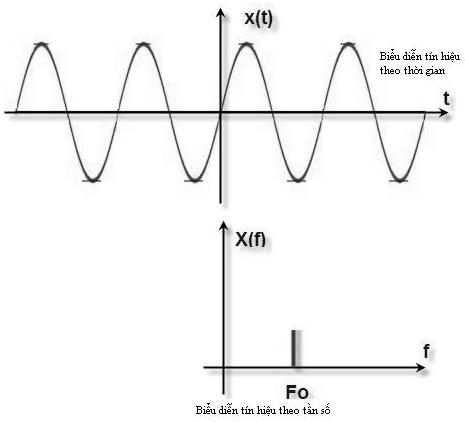
**2.1.1.2. Biểu diễn tín hiệu âm thanh trong miền thời gian và tần số.**

Thông thường, người ta dùng hàm toán học x(t) để biểu diễn âm thanh trong miền thời gian. Trong đó:

- t: thời gian

- x: biên độ biến thiên, hay còn gọi là ly độ.

Như vậy, ta có thể biểu diễn x(t) bằng đồ thị theo thời gian.Đặt x(t) = A.sin t = A. sin 2 F0t



Hình 2.1 Biểu diễn tín hiệu âm thanh

Phổ tính hiệu: là cách biểu diễn các thành phần cấu tạo nên x(t) theo tần số.

**2.1.1.3. Các loại âm thanh**

Những dao động cơ mà con người nghe được gọi âm thanh (sound).

Âm thanh có thể biểu diễn theo thời gian, song cũng có thể biểu diễn theo tần số do có thể phân tích một tín hiệu âm thanh thành tổ hợp các thành phần tần số khác nhau (Chuỗi Fourier, tích phân Fourier). Hoặc nói một cách đơn giản thực tiễn hơn, một âm thanh có thể là tổ hợp từ nhiều đơn âm, từ nhiều nhạc cụ, mà mỗi cái có một tần số dao động nhất định.

Dải tần số nghe được là từ 20 Hz - 20000 Hz. Siêu âm là âm dao động ngoài 20000 Hz. Hạ âm là các âm dao động dưới 20 Hz. Tai người không nghe được siêu âm và hạ âm.

Tiếng nói (voice, speech) là âm thanh phát ra từ miệng người, được truyền đi trong không khí đến tai người nghe. Dải tần số của tiếng nói đủ nghe rõ là từ 300 Hz đến 3500 Hz, là dải tần tiêu chuẩn áp dụng cho điện thoại. Còn dải tần tiếng nói có chất lượng cao có thể là từ 200 Hz-7000 Hz.

**2.1.1.4. Đơn vị đo âm thanh**

Người ta thấy rằng con người cảm nhận độ to của âm thanh không tỉ lệ thuận với cường độ âm thanh mà theo hàm số mũ.

Bel = 10lg P2/P1. (Phát âm là Ben)

decibel = 20lg I2/I1 (Phát âm là Đề xi ben)

**2.1.2. Tiếng nói**

Tiếng nói là âm thanh phát ra từ miệng (người). Nghiên cứu tiếng nói gồm: Bộ máy phát âm của con người. Thụ cảm âm thanh của tai người. Phân loại tiếng nói.

Bộ máy phát âm của con người gồm:

* **Phổi** đóng vai trò là cái bơm không khí, tạo năng lượng hình thành âm.
* **Đôi dây thanh** (vocal fold, vocal cord) là hai cơ thịt ở trong cuống họng, có hai đầu dính nhau, còn hai đầu dao động với tần số cơ bản là Fo, tiếng Anh gọi là pitch, fundamental frequency. Fo của nam giới nằm trong khoảng 100-200 Hz, của nữ giới là 300-400 Hz, của trẻ em là 500-600 Hz.
* **Thanh quản và vòm miệng:** đóng vai như là hốc cộng hưởng, tạo ra sự phân biệt tần số khi tín hiệu dao động từ đôi dây thanh phát ra. Đáp ứng tần số của hốc công hưởng này có nhiều đỉnh cộng hưởng khác nhau được gọi là các formant.
* **Miệng** đóng vai trò phát tán âm thanh ra ngoài.
* **Lưỡi** thay đổi để tạo ra tần số formant khác nhau.

Các âm khác nhau là do vị trí tương đối của formants.

* 1. Phân loại tiếng nói theo thanh:
* **Âm hữu thanh (**voiced, tiếng Pháp là voisé) là âm khi phát ra có sự dao động của đôi dây thanh, nên nó tuần hoàn với tần số Fo. Vì vậy phổ của nguyên âm là phổ vạch, khoảng cách giữa các vạch bằng chính Fo.
* **Âm vô thanh** (unvoiced, tiếng Pháp là non voisé) phát ra khi đôi dây thanh không dao động. Thí dụ phần cuối của phát âm English, chữ sh cho ra âm xát. Phổ tín hiệu có dạng là nhiễu trắng, phổ phân bổ đều.
  1. Phân loại tiếng nói:
* **Nguyên âm** (vowel) là âm phát ra có thể kéo dài. Tất cả nguyên âm đều là âm hữu thanh, nghĩa là tuần hoàn và khá ổn định trong một đoạn thời gian vài chục ms.
* **Phụ âm** (consonant) là âm chỉ phát ra một nhát, không kéo dài được. Có phụ âm hữu thanh và phụ âm vô thanh.

Thanh điệu của tiếng Việt tương ứng với các dấu: không dấu, huyền, hỏi, ngã, sắc, nặng khi viết

**Giọng bổng (high voiced pitch, hay high pitched)** hay **giọng trầm (low voiced pitch)** là Fo cao hay thấp. Như vậy Fo đóng vai trò rất quan trọng trong cảm nhận, trong thụ cảm âm thanh của con người.

**Tiếng bổng** hay **tiếng trầm** tương ứng với dải tần số cao hay thấp.

**2.2. Hệ Thống Ngữ Âm Tiếng Việt**

**2.2.1. Đặc điểm của tiếng Việt**

Tiếng Việt là ngôn ngữ đơn âm tiết, tức là các từ khi viết ra chỉ đọc lên thành một tiếng

Một từ có cấu tạo gồm 2 phần là: nguyên âm V (vowel) và phụ âm C (consonant) và được kết hợp theo 3 cách để tạo nên từ trong tiếng Việt:

- C+V (phụ âm + nguyên âm). Ví dụ: ba, mẹ, đi

- C+V+C (phụ âm + nguyên âm + phụ âm). Ví dụ: bàn, con, mong

- V+C (nguyên âm + phụ âm). Ví dụ: an, ông, én

Trong tiếng Việt, ngoài 2 thành phần chính là nguyên âm, phụ âm, còn có các thành phần khác giúp cho Việt phân loại trong âm tiết trở nên rõ ràng như nhị hợp âm, tam hợp âm, phụ âm đơn, phụ âm kép. Khi học tiếng Việt, ngay từ đọc phải học thuộc các nguyên âm, phụ âm, nhị hợp âm, tam hợp âm, phụ âm đơn, phụ âm kép, quy tác ghép nối các thành phần đó để tạo thành âm tiết hoặc một từ, khi đó một từ tiếng Việt được Việt ra, sẽ kèm theo các đọc của từ đó bằng quy tắc kết hợp trên.

Tiếng Việt có yếu tố đa thanh. Đa thanh tức là có nhiều thanh điệu, nhiều dấu giọng. Cụ thể là có 6 thanh điệu, được ghi bằng 5 ký hiệu khác nhau: dấu sắc (Á), dầu huyền (À), dầu hỏi (Ả), dầu ngã (Ã), dấu nặng (Ạ).

**2.2.2. Hệ thống mẫu tự và ngữ âm tiếng Việt**

**2.2.2.1. Hệ thống mẫu tự tiếng Việt**

Bảng chữ cái tiếng Việt có 29 chữ cái [7], theo thứ tự:

**[a, ă, â, b, c, d, đ, e, ê, g, h, i, k, l, m, n, o, ô, ơ, p, q, r, s, t, u, ư, v, x, y]**

chia làm hai phần: mẫu tự chính (khi phát âm thì gọi là nguyên âm) và mẫu tự phụ (khi phát âm thì gọi là phụ âm):

**Nguyên âm:** Trong tiếng Việt, ngoài nguyên âm đơn còn có nguyên âm đôi, nguyên âm ba. Có mối liên hệ phức tạp giữa nguyên âm và cách phát âm của chúng. Một nguyên âm có thể biểu thị cho vài cách phát âm khác nhau, tùy theo nó nằm trong nguyên âm đơn, đôi hay ba; và nhiều khi các cách viết nguyên âm khác nhau tượng trưng cho cùng một cách phát âm.

**Phụ âm:**

tiếng Việt có 17 phụ âm đơn trong tập trên gồm:

**[b, c, d, đ, g, h, k, l, m, n, p, q, r, s, t, v, x]**

và 11 phụ âm ghép:

**[gi, gh, qu, ch, kh, ng, ngh, nh, ph, th, tr]**

Trong đó chỉ có 8 phụ âm có thể nằm ở cuối từ:

**[c, m, n, p, t, ng, nh, ch]**

**2.2.2.2. Hệ thống ngữ âm tiếng Việt**

Trong tiếng Việt, âm tiết có cấu trúc chặt chẽ, mỗi âm vị có một vị trí nhất định trong âm tiết. Theo một số nhà nghiên cứu âm vị học trong tiếng Việt, âm tiết tiếng Việt có cấu tạo như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thanh điệu** | | | |
| **Âm đầu** | **Vần** | | |
| **Âm đệm** | **Âm chính** | **Âm cuối** |

**a. Âm đầu:**

Tại vị trí thứ nhất trong âm tiết, âm đầu có chức năng mở đầu âm tiết. Những âm tiết mà chính tả không ghi âm đầu như an, ấm, êm... được mở đầu bằng động tác khép kín khe thanh, sau đó mở ra đột ngột, gây nên một tiếng bật. Động tác mở đầu ấy có giá trị như một phụ âm và người ta gọi là âm tắc thanh hầu (kí hiệu: /?/). Như vậy, âm tiết trong tiếng Việt luôn luôn có mặt âm đầu (phụ âm đầu). Với những âm tiết mang âm tắc thanh hầu như vừa nêu trên thì trên chữ viết không được ghi lại, và như vậy vị trí xuất hiện của nó trong âm tiết là zero, trên chữ viết nó thể hiện bằng sự vắng mặt của chữ viết.

**b. Âm đệm**

Âm đệm là yếu tố thứ hai trong âm tiết, nó thường nằm trong các âm tiết để tạo nên sự khác nhau giữa âm tròn môi (như “toàn”) và âm không tròn môi (như “tàn”). Âm đệm trong tiếng Việt được miệu tả gồm 2 dạng: âm vị bán nguyên âm /u/ (trong “toàn”) và âm vị trống (trong “tàn”). Trên chữ viết, âm đệm trống thể hiện bằng sự vắng mặt của chữ viết, âm đệm /u/ thể hiện bằng chữ “u” (như “tuấn”) và chữ “o” (như “loan”).

**c. Âm chính**

Đứng ở vị trí thứ ba trong âm tiết, âm chính được xem như là đỉnh của âm tiết, mang âm sắc chủ yếu của âm tiết và luôn là nguyên âm. Do được xem là thành phần hạt nhận của âm tiết, nên không bao giờ có một từ nào đọc được lại không có âm chính, trong âm tiết, âm chính cũng đóng vai trò là âm mang thanh điệu của âm tiết.

**d. Âm cuối**

Âm cuối nằm cuối cùng trong âm tiết, nó có chức năng kết thúc một âm tiết, trong các âm tiết tiếng Việt ta thường thấy có sự đối lập bằng các cách kết thúc khác nhau. Một số âm tiết có âm cuối kết thúc bằng sự kéo dài và giữ nguyên (như “má”,”ba”), số khác lại có âm cuối kết thúc bằng cách biến đổi âm tiết ở phần cuối do có sự tham gia của âm cuối (như “bộn”, “bàn”,”bài”). Trong trường hợp đầu, ta gọi các âm cuối đó là âm vị rỗng, trường hợp còn lại âm cuối là những âm vị bán nguyên âm hay phụ âm.

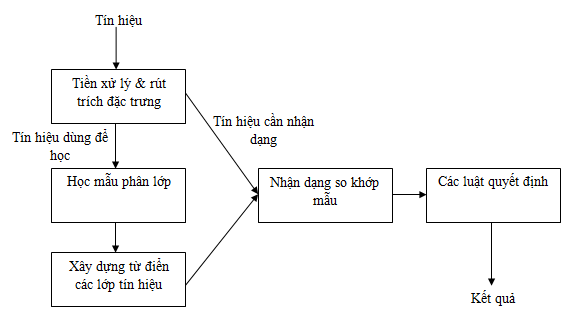
**e. Thanh điệu**

Thanh điệu là một yếu tố thể hiện độ cao và sự chuyển biến của độ cao trong mỗi âm tiết. Mỗi âm tiết tiếng Việt nhất thiết phải được thể hiện với một thanh điệu. Thanh điệu có chức năng phân biệt vỏ âm thanh, phân biệt nghĩa của từ. Có nhiều ý kiến khác nhau về vị trí của thanh điệu trong âm tiết. Nhưng ý kiến cho rằng thanh điệu nằm trong cả quá trình phát âm của âm tiết (nằm trên toàn bộ âm tiết) là đáng tin cậy nhất về vị trí của thanh điệu.

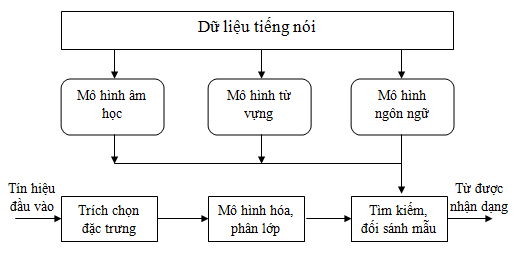
**2.3. Hệ Nhận Dạng Tiếng Nói**

**2.3.1. Tổng quan**

Nhận dạng tiếng nói là một hệ thống tạo khả năng để máy nhận biết ngữ nghĩa của lời nói. Về bản chất, đây là quá trình biến đổi tín hiệu âm thanh thu được của người nói qua Micro, đường dây điện thoại hoặc các thiết bị khác thành một chuỗi các từ. Kết quả của quá trình nhận dạng có thể được ứng dụng trong điều khiển thiết bị, nhập dữ liệu, soạn thảo văn bản bằng lời, quay số điện thoại tự động hoặc đưa tới một quá trình xử lý ngôn ngữ ở mức cao hơn.



Hình 2.2 Sơ đồ nhận dạng tiếng nói tổng quát



Hình 2.3 Các phần tử cơ bản của một hệ thống nhận dạng tiếng nói

Hình 2.3 là cấu trúc của một hệ nhận dạng tiếng nói. Tính hiệu tiếng nói đầu tiên được tiền xử lý và rút trích đặc trưng. Kết quả thu được sau quá trình này là tập các đặc trưng âm học (acoustic features), được tạo dựng thành 1 hay nhiều véctơ được gọi là vector đặc trưng.

Để có thể thực hiện việc so sánh, trước hết hệ thống phải được huấn luyện và xây dựng các đặc trưng, sau đó mới có thể dùng để so sánh với các tham số đầu vào để nhận dạng.

Trong quá trình huấn luyện, hệ thống dùng các vector đặc trưng được đưa vào để ước lượng, tính toán các tham số cho các mẫu (được gọi là mẫu tham khảo). Một mẫu tham khảo chính là bản mẫu dùng để so sánh và nhận dạng, các mẫu tham khảo này mô phỏng cho một từ, một âm tiết, hoặc thậm chí một âm vị.

Trong quá trình nhận dạng, dãy các vector đặc trưng được đem so sánh với các mẫu tham khảo (được xây dựng ở trên). Sau đó, hệ thống tính toán độ tương đồng (likelihood) của dãy vector đặc trưng và mẫu tham khảo. Việc tính toán độ tương đồng được được thực hiện bằng cách áp dụng các thuật toán đã được chứng minh hiệu quả như thuật toán Vitertbi (trong Hidden Markov Model). Mẫu có độ tương đồng cao nhất được cho là kết quả của quá trình nhận dạng.

**2.3.2. Phân loại các hệ thống nhận dạng tiếng nói**

**2.3.2.1. Nhận dạng từ liên tục và nhận dạng từ tách biệt**

Một hệ nhận dạng tiếng nói có thể là một trong hai dạng: nhận dạng liên tục và nhận dạng từng từ

**2.3.2.2. Nhận dạng phụ thuộc người nói và độc lập người nói**

Nhận dạng phụ thuộc người nói thì mỗi một hệ nhận dạng chỉ phục vụ được cho một người, và nó sẽ không hiểu người khác nói gì nếu như chưa được huấn luyện lại từ đầu. Do đó, hệ thống nhận dạng người nói khó được chấp nhận rộng rãi vì không phải ai cũng đủ khả năng kiến thức và nhất là kiên nhẫn để huấn luyện hệ thống. Đặc biệt là hệ thống loại này không thể ứng dụng ở nơi công cộng. Ngược lại, hệ thống nhận dạng độc lập người nói thì lý tưởng hơn, ứng dụng rộng rãi hơn, đáp ứng được hầu hết các yêu cầu đề ra. Nhưng không may là hệ thống lý tưởng như vậy gặp một số vấn đề, nhất là độ chính xác của hệ thống.

Do đó, trong thực tế có một cách giải quyết là bán độc lập người nói. Phương pháp này thực hiện bằng cách thu mẫu một số lượng lớn các giọng nói khác biệt nhau. Khi sử dụng, hệ thống sẽ được điều chỉnh cho phù hợp với giọng của người dùng, bằng cách nó học thêm một vài câu có chứa các từ cần thiết

**2.3.3. Một số phương pháp nhận dạng tiếng nói**

Có 3 phương pháp phổ biến được sử dụng trong nhận dạng tiếng nói hiện nay:

- phương pháp âm học- ngữ âm học.

- phương pháp nhận dạng mẫu (Markov ẩn - HMM)

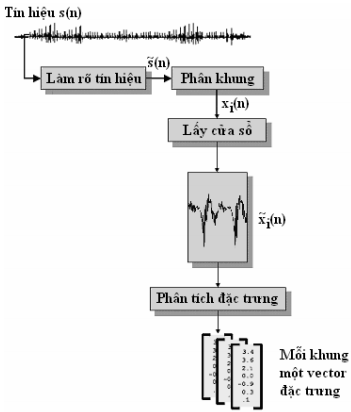
- phương pháp ứng dụng trí tuệ nhân tạo.

**2.4. Rút Trích Đặt Trưng Tín Hiệu Tiếng Nói**

**2.4.1. Giới thiệu**

Rút trích đặc trưng của tiếng nói là một trong những khâu quan trọng trong quá trình nhận dạng tiếng nói. Dữ liệu tiếng nói thông thường dưới dạng sóng âm đã lưu trữ trong máy tính là loại dữ liệu khó xử lý, học mẫu huấn luyện, và so sánh, vì thế việc rút trích đặc trưng tiếng nói là cần thiết. Kết quả của quá trình rút trích đặc trưng là 1 hoặc nhiều vector đặc trưng, các vector này chứa các tham số mang giá trị quan trọng của tín hiệu tiếng nói, làm giảm đi rất nhiều số lượng tính toán cần thực hiện, làm rõ ràng hơn sự khác biệt giữa 2 tín hiệu tiếng nói. Hình bên dưới minh họa cho quá trình rút trích đặc trưng.

Có nhiều phương pháp để thực hiện rút trích đặc trưng, 2 trong số đó là phương pháp MFCC và LPC.

**

*Hình 2.4 Công đoạn rút trích đặc trưng*

Hình 2.4 mô tả quá trình của việc rút trích đặt trưng, tín hiệu âm thanh lưu trong máy tính là tín hiệu digital [9], mô hình hóa tín hiệu âm thanh trong máy tính dưới dạng toán học là một hàm s(n), trong đó n chỉ thời gian (thông thường là ms) và s(n) là biên độ âm.

**2.4.2. Làm rõ tín hiệu (pre-emphasis - tiền khuếch đại)**

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:

(2.1)

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 tren 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. Hệ thống số được dùng trong khối tiền khuếch đại vừa cố định vừa thích nghi chậm (ví dụ, để lấy trung bình các điều kiện chuyển, các nền nhiễu, hoặc thậm chí lấy trung bình phổ tín hiệu).

Trong trường hợp sử dụng bộ lọc áp theo công thức 2.1, 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:

s'(n) s(n) apres(n 1) (2.2)

**2.4.3. Tách từ**

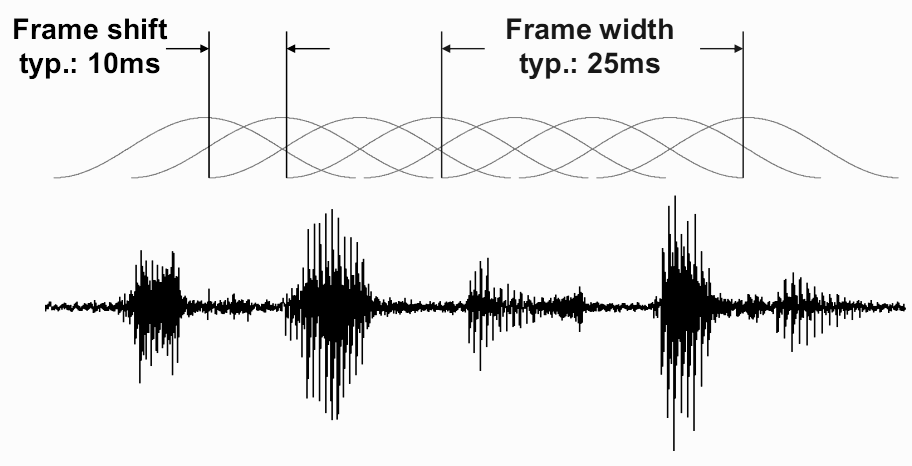
Tín hiệu tiếng nói s(n) sau khi được làm nỗi tín hiệu, sẽ được chuyển sang để tách từ, tách từ là công đoạn chia toàn bộ tính hiệu thu được thành những đoạn tính hiệu mà trong đó chỉ chứa nội dung của một từ

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 thứ m, hàm năng lượng thời gian ngắn E(m) được xác định:

**2.4.3.1. Phân đoạn thành các khung (Framing)**

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, s (n) , được chia khối thành các khung N mẫu, với các khung kề nhau được ngăn cách bởi M mẫu.

**

*Hình 2.5 Ví dụ phân khung đoạn tín hiệu*

Khung đầu tiên chứa N mẫu tiếng nói đầu tiên. Khung thứ hai bắt đầu sau khung thứ nhất M mẫu, và chồng lên nó N - M mẫu. Thông thường, khung thứ 3 bắt đầu sau 2M so với khung đầu tiên (hoặc M mẫu sau khung thứ 2) và chồng lên khung đầu N - 2M mẫu. Quá trình này tiếp tục cho đến khi toàn bộ tiếng nói được tính hết cho một hay nhiều khung. Dễ thấy là nếu MN thì các khung chồng lên nhau và ước đoán phổ LPC kết quả sẽ là tương quan từ khung này đến khung khác; nếu M<<N thì các ước đoán phổ LPC từ khung này đến khung khác sẽ khá trôi chảy. Nói cách khác, nếu M>N, sẽ không có chồng lấp giữa các khung kề nhau; thực tế, một số tín hiệu tiếng nói sẽ hoàn toàn bị mất (tức là không bao giờ xuất hiện trong bất cứ khung phân tích nào), và mối tương quan giữa các ước đoán phổ LPC của các khung kề nhau sẽ không chứa một thành phần nhiễu mà cường độ của nó tăng như M (nghĩa là, khi có nhiều tiếng nói bị bỏ qua không phân tích). Tình trạng này là không thể chấp nhận trong phân tích LPC cho nhận dạng tiếng nói. Nếu ta biểu thị khung tiếng nói thứ  là x (n) và có L khung trong toàn bộ tín hiệu tiếng nói thì

x(n)s(Mn) n0,1,...,N1 0,1,...L1 (2.4)

Tức là, khung tiếng nói đầu tiên, x0(n), chứa các mẫu tiếng nói s(0), s(1), ..,

s(N 1), khung tiếng nói thứ hai, x1(n), chứa các mẫu s(M),s(M 1),...,

s(M N 1),và khung thứ L, xL-1(n), chứa các mẫu s(M(L1)), s(M(L1)1), ...,

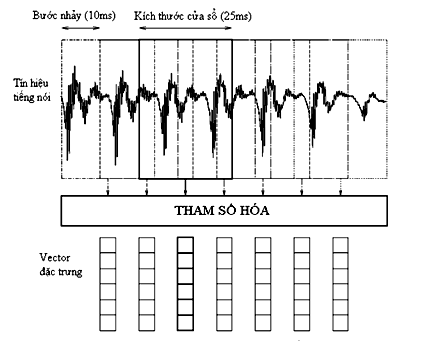
s ( M ( L 1) N 1) .

**2.4.5. Rút trích đặc trưng**

**2.4.5.1. Phương pháp rút trích đặc trưng MFCC**

Trong lĩnh vực xử lý và nhận dạng tiếng nói, việc tiền xử lý các tín hiệu thu được và rút trích đặc trưng là một kỹ thuật thiết yếu mà bất cứ hệ thống nhận dạng nào cũng bắt buộc phải có. Trích rút đặc trưng có vai trò quan trọng quyết định hiệu suất của quá trình nhận dạng mẫu(cả trong quá trình nhận dạng và trong quá trình huấn luyện). Công việc của bước này là phân tích phổ spectral nhằm mục đích xác định các thông tin quan trọng, đặc trưng của tiếng nói, cắt giảm bớt các yếu tố không cần thiết trong quá trình nhận dạng để làm giảm khối lượng dữ liệu cần xử lý.

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 Cosine rời rạc ( Discrete Cosin Transform ) thành một thuật toán hoàn chỉnh ứng dụng trong lĩnh vực nhận dạng tiếng nói liên tục. Động thời định nghĩa khái niệm hệ số Cepstral và thang đo tần số Mel (Mel scale).

**

*Hình 2.7 Tổng quát phương pháp rút trích đặc trưng MFCC*

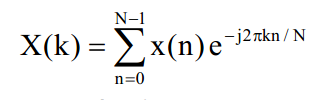
Tóm tắt quá trình rút trích đặc trưng theo MFCC sẽ như sau: Ban đầu tín hiệu sau khi qua tiền xử lý sẽ được chia thành các Frame có khoảng thời gian ngắn. Từ mỗi frame đó sau khi áp dụng các bước chuyển đổi và lọc sẽ ra được một vecto tương ứng. Và xong quá trình này, ta sẽ có đặc trưng của dãy tín hiệu input vào là một dãy vecto đặc trưng output ra.

**f. 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):

k=0,1,2,..., N-1 (2.8)

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

n=0,1,2,..., N-1 (2.9)

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

**g. Lọc qua bộ lọc Mel-scale**

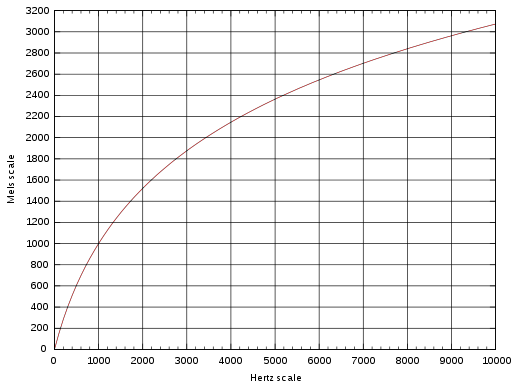
Trong lĩnh vực nghiên cứu về nhận dạng tiếng nói, đòi hỏi chúng ta phải hiểu và mô phỏng chính xác khả năng cảm thụ tần số âm thanh của tai người. Chính vì thế các nhà nghiên cứu đã xây dựng một thang tần số - hay gọi là thang tần số Mel (Mel scale) dựa trên cơ sở thực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

(2.10)

Trong đó:

m là tần số trong thang Mel, đơn vị là Mel

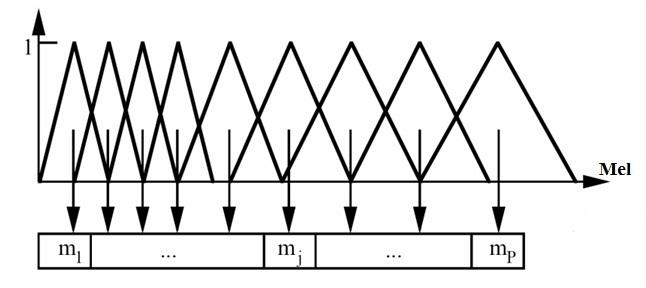
f là tần số thực, đơn vị là Hz.

**

*Hình 2.8 Biểu đồ thang tần số Mel theo tần số thực.*

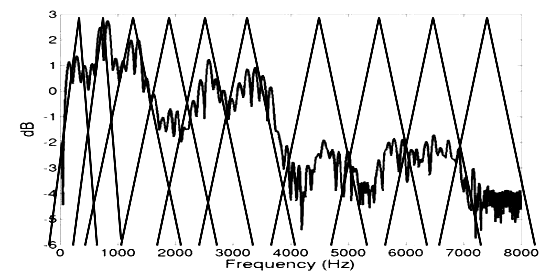
Theo biểu độ trên thì trong khoảng tần số thấp hơn 1kHz thì đồ thị trên gần như là tuyến tính, nghĩa là trong khoảng tần số dưới 1kHz, tần số Mel và tần số thực. Trong khoảng tần số trên 1kHz thì mối quan hệ này la quan hệ Logarit.

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

**

*Hình 2.9 Băng lọc tần số Mel.*

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

**

*Hình 2.10 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)

**h. 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.



**i. 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.

(2.11)

Trong đó: N là số kênh lọc.

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\*k).cn (2.12)

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.

**2.4.5.2. Phương pháp trích đặc trưng LPC**

**j. 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.

**k. 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.

**l. 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) a1s(n-1) + a2s(n-2) + ... + aps(n-p) (2.13)

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:

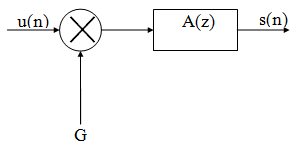
s(n) a s(n i) Gu(n) (2.14)

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) a ziS(z) GU(z) i (2.15)

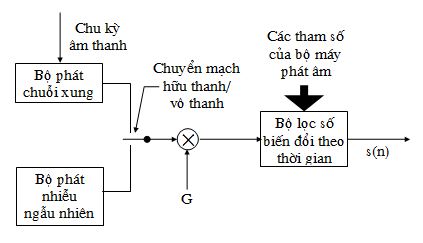
dẫn đến hàm chuyển:

Hình 2.11 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) 1 , tạo ra tín hiệu tiếng nói s(n).

**

*Hình 2.11 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 2.12. Mẫu phân tích tiếng nói theo phương pháp LPC*

Dựa vào mẫu trên hình 1, quan hệ chính xác giữa s(n) và u(n) là

 s(n)a s(nk)Gu(n) (2.17)

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

s(n) a (n k) (2.18)

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

e(n)s(n)s(n)s(n)a (nk) (2.19)

Với hàm chuyển đổi lỗi:

E(z) p A(z)S(z) 1akzk (2.20)

Rõ ràng là khi s(n) được tạo thực sự bởi một hệ tuyến tính kiểu như hình 13 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.

**m. Phân tích tự tương quan**

Sau khi tiến hành công đoạn nhận với hàm cửa sổ (áp dung công thức 2.7). Mỗi khung của tín hiệu được chia cửa sổ là tự tương quan với khung tiếp theo để cho:

r(m)n0 x (n)x (nm) m0,1,...,p (2.21)

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.

**n. 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 l nhỏ ở dưới rl(m) ):

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.

**r. Nhận xét phương pháp LPC**

Tóm lại, trong mô hình phân tích LPC trên, chúc ta cần phải đặc 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).

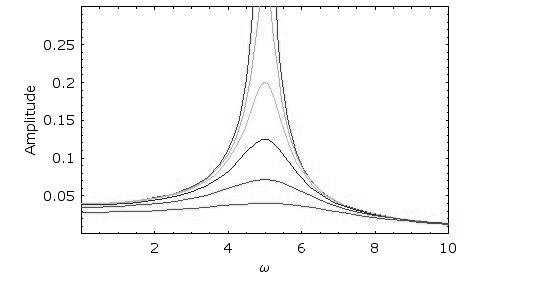
Bảng 3 Các giá trị đặc trưng cho các tham số phân tích LPC đối với hệ nhận dạng tiếng nói

|  |  |  |  |
| --- | --- | --- | --- |
| 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 ch ta một xấp xỉ tốt đối với đường bao phổ cơ quan phát âm. Với miềng 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 trê phần cứng và phần mềm.

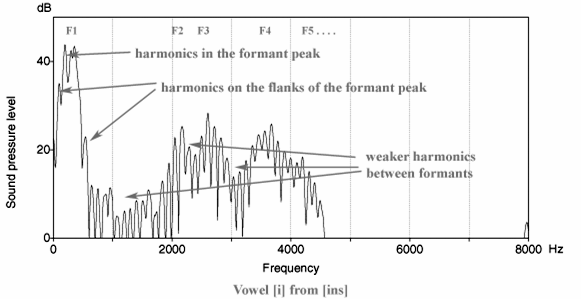
**2.4.6. Tìm hiểu về Formant**

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 2.13 Minh họa 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 2.14 Minh họa 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

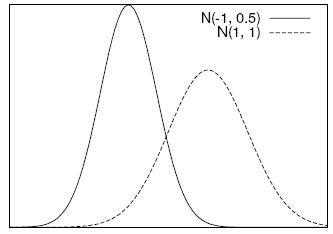
F2: ứng với cộng hưởng khoang miệng

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.

**2.5. Gaussian Mixture Model**

Mô hình hợp 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 2.15: Hàm mật độ Gausss*

Về cơ bản, mô hình GMM xấp xỉ một hàm mật độ xác suất bằng hợp các hàm mật độ Gauss. Hình 2.15 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, μ, σ2) được cho bởi công thức:

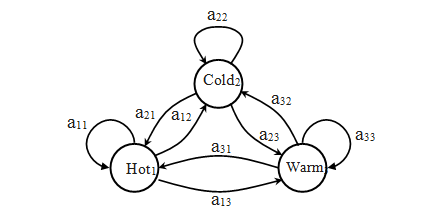
Như vậy, một mô hình GMM có M phân phối Gauss sẽ được đại diện bởi bộ tham số λ={wi,μi,Σi},i[1,M].Tùy thuộc vào cách tổ chức của ma trận hiệp phương sai (covariance matrix), GMM có thể có một số biến thể khác nhau:

* 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.

**2.6. Hidden Markov Model**

**2.6.1. Giới thiệu chuỗi Makov**



Hình 2.18 Mô hình Markov chain thời tiết

Một mô hình Markov chain được cụ thể hóa bằng các thành phần:

* S = {S1, S2, ..., SN} Tập các trạng thái, gọi qt là trạng thái đạt đến được ở thời điểm t
* A = { aij } Ma trận xác suất chuyển trạng thái, mỗi aij thể hiện xác xuất di chuyển từ trạng thái Si sang trạng thái Sj,

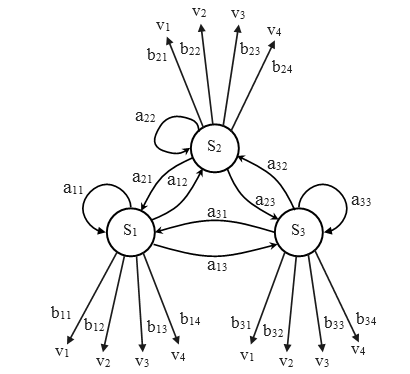
aijpqt1Sj|qtSi, 1i,jN,saocho a i j 0 , N a i j 1

* π= {πi} . πi là xác suất khởi đầu của trạng thái Si – xác suất rơi

vào trạng thái Si ở thời điểm t = 1:pq S,1iN,1

**2.6.2. Mô hình Markov ẩn**

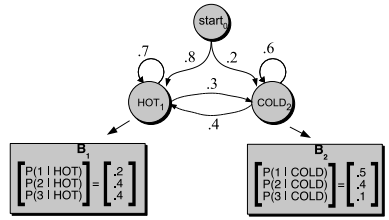
Trong mô hình Markov ẩn, các sự kiện quan sát được nằm trong mỗi trạng thái và phụ thuộc vào vào hàm mật độ xác suất trong các trạng thái đó.



Hình 2.19 Mô hình Markov ẩn 3 trạng thái

Hình 2.19 minh họa một mô hình Markov ẩn 3 trạng thái với các sự kiện có thể quan sát được trong mỗi trạng thái là V = {v1, v2, v3, v4}. Khả năng quan sát được sự kiện vk trong trạng thái Sj phụ thuộc vào xác suất quan sát bj(k). Hàm b được gọi là hàm mật độ xác suất của các sự kiện được quan sát.

Hình 2.20 minh họa một ví dụ HMM đơn giản về mối liên hệ giữa số lượng que kem với thời tiết. Số lượng que kem được ăn mỗi ngày là các quan sát, ví dụ: O = {1,2,3}. Thời tiết nóng hay lạnh tương ứng với hai trạng thái. Giả sử chúng ta biết được số lượng que kem được ăn, không biết thời tiết, liệu chúng ta có dự đoán được thời tiết hôm ấy thế nào không. Thời tiết (trạng thái) được xem là “ẩn” so với số que kem được ăn (kết quả quan sát được). Chính vì vậy, mô hình này được gọi là mô hình Markov ẩn (hidden) – Hidden Markov Model (HMM).



Hình 2.20 Ví dụ HMM đơn giản

Một mô hình Markov ẩn được cụ thể hóa bằng các thành phần:

* S = {S1, S2, ..., SN} Tập các trạng thái, qt là trạng thái đạt đến được ở thời điểm t.
* V = {v1, v2, ..., vM} Tập các tín hiệu quan sát, gọi Ot là tín hiệu quan sát được ở thời điểm t
* A = { aij } Ma trận xác suất chuyển trạng thái, mỗi aij thể hiện xác xuất di chuyển từ trạng thái Si sang trạng thái Sj,

aijpqt1Sj|qtSi, 1i,jN,saocho N j1 aij0, a1

* B = { bj(k) } Ma trận các hàm mật độ xác suất trong mỗi trạng thái,

bj(k)pvk att|qt Sj, 1jN, 1kM k1

Thỏa ràng buộc M bj (k) 1

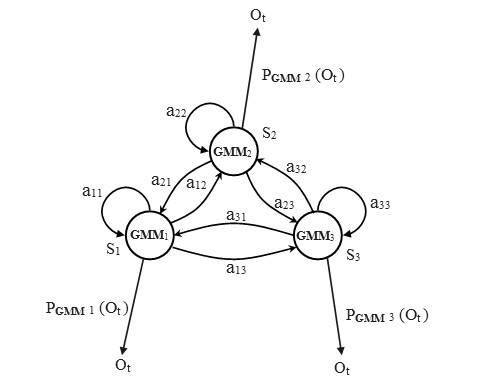
* π= {πi} πi là xác suất khởi đầu của trạng thái Si – xác suất rơi vào trạng thái Si ở thời điểm t = 1: p q S , 1 i N , N i 1

Để thuận tiện cho việc trình bày, mỗi mô hình HMM sẽ được quy ước đại diện bởi bộ tham số λ = (π , A, B).

**2.7. Mixture of Gaussians Hidden Markov Model**

**2.7.1. Đặc tả mô hình**

Phần 2.6 đã trình bày về mô hình HMM và 3 bài toán cơ bản của nó, trong đó hàm mật độ xác suất của các tín hiệu quan sát là rời rạc. Mô hình HMM này được gọi là HMM rời rạc. Khi hàm mật độ xác suất trong mỗi trạng thái là hàm liên tục, ta có HMM liên tục. Thông thường, có hai dạng chính của HMM liên tục:

1. Gaussian Hidden Markov Model (GHMM): hàm mật độ xác suất trong mỗi trạng thái là hàm mật độ Gauss.
2. Mixture of Gaussians Hidden Markov Model (MGHMM): hàm mật độ xác suất trong mỗi trạng thái là hợp các hàm mật độ Gauss (mô hình GMM như đã trình bày trong mục 2.7 ở trên).
3. 

Hình 2.23: Mô hình MGHMM 3 trạng thái

Phần này trình bày về mô hình MGHMM. Đây là một dạng của HMM liên tục, trong đó hàm mật độ xác suất của vector quan sát Ot được cho bởi mô hình GMM:

bj(Ot)pGMMj (Ot)  (2.64)

trong đó pGMM j chính là xác suất đầu ra của mô hình GMM trong trạng thái Sj.

Như vậy, khả năng quan sát được các vector trong mỗi trạng thái sẽ bị chi phối bởi GMM của trạng thái đó. Hình 2.23 minh họa mô hình MGHMM có 3 trạng thái.

Cũng như trong định nghĩa của HMM, một mô hình MGHMM có N trạng thái và M phân phối Gauss trong mỗi trạng thái sẽ được đại diện bởi bộ tham số λ = {π, A, B}, trong đó:

* A = { aij }, aij là xác suất chuyển từ trạng thái Si sang trạng thái Sj.
* π = { πi }, πi là xác suất khởi đầu của trạng thái Si.
* B = { bj }, bj là hàm mật độ xác suất trong trạng thái Sj.

aij và πi thì không có gì thay đổi so với HMM, điểm khác biệt chính nằm ở bj. Từ (2.38) ta có:

bj(Ot)c GO,,U (2.65)

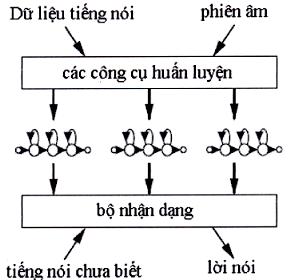
trong đó trạng thái thứ j  [1, N], Ot là vector quan sát được ở thời điểm t, G(μjm, Ujm) là hàm mật độ Gauss thứ m trong trạng thái Sj với vector trung bình μ và ma trận hiệp phương sai U tương ứng (dùng U thay cho Σ để tránh nhầm lẫn với ký hiệu Σ tổng), cjm là trọng số (không âm) của phân phối Gauss thứ m trong trạng thái Sj thỏa ràng buộc xác suất c 1

**CHƯƠNG 3. CÁC FRAMEWORK NHẬN DẠNG GIỌNG NÓI**

**3.1. HTK Framework**

**3.1.1. Giới thiệu**

HTK (Hidden Markov Model Toolkit) là một bộ các công cụ để xây dựng mô hình Markov ẩn (HMM). HTK được thiếu kế chủ yếu để xây dựng các mô hình xử lý, nhận dạng tiếng nói dựa trên HMM, do đó các hỗ trợ mả HTK cung cấp đều phục vụ cho mục đích trên.

**

*Hình 3.1 Nhận dạng tiếng nói sử dụng HTK*

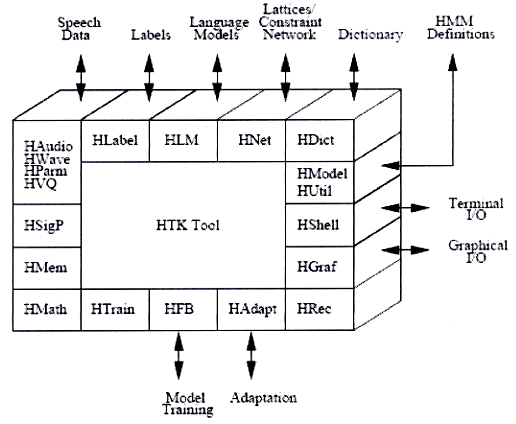
Trong hình 23 có mô tả một hệ thống nhận dạng tiếng nói có sử dụng HTK, nó bao gồm 2 giai đoạn xử lý chính:

* Các công cụ huấn luyện HTK được sử dụng để ước lượng các tham số của một tập các HMM sử dụng các câu nói và kết hợp các phát âm của chúng để huấn luyện.
* Sau đó, một câu nói chưa biết thông qua các công cụ nhận dạng HTK sẽ cho kết quả là các phát âm (phiên âm) của những tiếng nói này.

HTK được sử dụng như một bộ thư viện, dễ mở rộng và phát triển. Đây là một công cụ lý tưởng để nghiên cứu mô hình và nhận dạng dưa trên HMM và qua đó, chúng ta có thể dành nhiều thơi gian hơn để phát triển các công cụ riêng cho mình nhằm giúp cho việc nhận dạng tốt hơn. Như chúng ta có thể dễ dàng áp dụng công cụ này cho việc nhận dạng tiếng Việt, không phải mất thời gian để xây dựng lại công cụ mộ hình HMM, HTK thực sự đã được phát triển rất tốt cho công việc này.

**3.1.2. Tổng quan về HTK**

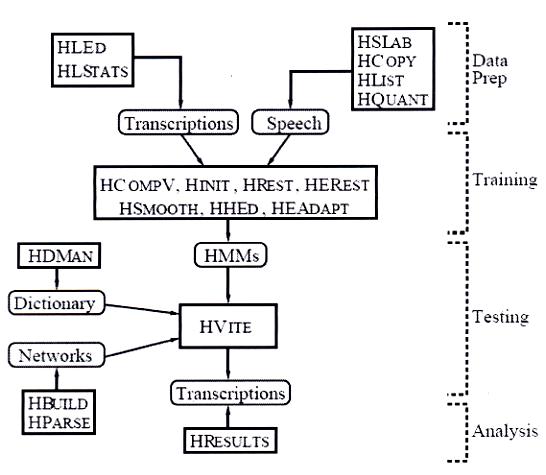
HTK cung cấp một tập các công cụ để thực hiện các nhiệm vụ khác nhau trong mô hình Markov ẩn trong hệ thống nhận dạng. Các công cụ này được viết bằng ngôn ngữ C và sử dụng một tập các hàm cơ bản để trong một thư viện chung để xây dựng HMM

**

*Hình 3.2 Kiến trúc của HTK*

Trong Hình 24, mô tả kiến trục tổng quát của HTK, bao gồm nhiều môdun được phân chi theo từng mục đích khác nhau, trong đó có các công cụ hỗ trợ đặc biệt cho việc nhận dạng. Hình 25 dưới trình bày các công đoạn của việc xây dựng hệ thống nhận dạng tiếng nói, và các công cụ mà HTK cung cấp qua từng công đoạn cụ thể như: chuẩn bị dữ liệu, huấn luyện, kiểm tra kết quả, đánh giá và cuối cùng là phân tích.



**

*Hình 3.3 Các công đoạn xây dựng hệ nhận dạng tiếng nói với HTK*

HTK sử dụng những tập tin dữ liệu khác nhau để chuyển đổi dữ liệu cho các công cụ khác nhau. Các tập tin này có thể chứa dữ liệu âm thanh (dạng sóng âm hoặc dãy các vector đặc trưng âm thanh), dữ liệu âm thanh được đánh nhãn (phiên âm), các HMM (các tham số định nghĩa mô hình HMM) hoặc các mạng nhận dạng. những tham số của HMM có thể được chia sẽ giữa các tập hợp, các trạng thái, hay các HMM,... Trong HTK, đơn vị nhận dạng nhỏ nhất (do người dùng định nghĩa, có thể là âm vị, âm tiết hoặc một từ) được mô hình bằng một HMM được gọi là một phone. Việc kết hợp các mô hình trong các bộ phận nhận dạng dựa trên từ con (sub-word), HTK có cung cấp các HMM phụ thuộc ngữ cảnh, do đó mô hình nhị âm (bi-phone) hay tam âm (tri-phone) được sữ dụng cho mỗi ngữ cảnh có khả năng giữa các phone

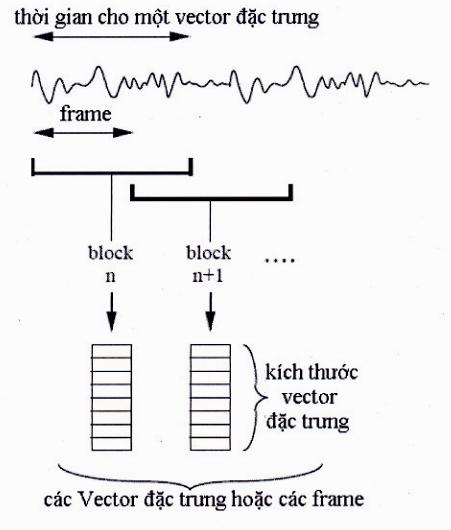
Sau đây là các công cụ chính trong HTK

* -  HCode: đây là công cụ được sử dụng để mã hóa một tập tin tiếng nói ở dạng  sóng âm thành tập tin ở dạng tham số (dãy các vector quan sát).
* -  HInit: công cụ này được dùng để khởi tạo một HMM dựa trên một tập các  phân đoạn dữ liệu huấn luyện được đánh nhãn.
* -  HRest: đây là công cụ dùng đề tái ước lượng tham số dựa trên thuận toán Baum-Welch, nó cũng làm việc trên một tập các phân đoạn dữ liệu huấn luyện được đánh nhãn.
* -  HERest: đây là công cụ dựa trên thuật toán Baum-Welch huấn luyện đồng thời (embedded training). Nó huấn luyện một tập các HMM cùng lúc sử dụng một tập dữ liệu huấn luyện (bao gồm các câu được nói liên tục cùng phiên âm của chúng).
* -  HVite: công cụ này sử dung thuật toán Viterbi đề nhận dạng tiếng nói liên tục, dựa trên các ràng buộc về cú pháp và tìm kiếm tín hiệu.
* -  HResults: công cụ này sử dụng 2 tập tin lưu trữ dữ liệu phiên âm, sử dụng một kết hợp lập trình động (Dynamic programming), sau đó xuất tra thống kê kết quả nhận dạng.
* -  HSLab: công cụ này dùng để hiện thị tiếng nói dưới dạng sóng âm và tập phiên âm của nó dưới dạng đồ họa.
* **3.1.3. Chi tiết về HTK**
* **3.1.3.1. Xử lý tín hiệu tiếng nói**

HTK hổ trợ việc rút trích đặc trưng (quá trình chuyển tín hiệu âm thanh dạng sóng thành dãy các vector tham số). Dữ liệu đầu vào của bước này là các tập tin được lưu trữ dưới dạng file .wav và kết quả thu được là dãy các tham số sử dụng phương pháp nén MFCC\_C (compressed mel-ceptral coefficients). Thật ra, HTK cho phép dữ liệu dầu vào và ra ở nhiều dạng khác nhau. Ví dụ, dữ liệu vào còn có thể là tín hiệu âm thanh được thu trực tiếp bằng công cụ HAudio và dữ liệu đầ ra là dãy các tham số của một phương pháp rút trích đặc trưng khác như LPC, PLP,..

Quá trình này được thể hiện trong hình 26. Đối với HTK, các tập tin dạng sóng âm và các tập tin tham số hóa là như nhau, chúng đều dùng để biểu diễn cho cùng một dãy mẫu tín hiệu, chỉ khác là một loại thì ở dạng nhị phân, loại kia ở dạng vector nhiều thành phần. Thông thường một vector tham số chỉ thể hiện cho một khoản thời gian nhất định trong dãy tin hiệu âm thanh, và khoảng thời gian này là từ 10-25 msec (người dùng HTK có thể định nghĩa lại giá trị này).





*Hình 3.4 Xử lý mã hóa tiếng nói*

Cơ chế của quá trình Xử lý tiếng nói trong HTK tương tự như đã giới thiệu trong phần *Rút trích đặc trưng* của báo cáo này. Trong HTK ta có thể sự dụng lện HCopy để thực hiện việc này.

**3.1.3.2. Ước lượng tham số cho các mô hình**

HTK cung cấp 4 công cụ cơ bản cho việc ước lượng tham số cho các HMM, đó là: HCompV, HInit, HRest. HCompV và HInit được dùng để khởi tạo giá trị cho các tham số. HCompV sẽ tính kỳ vọng và phương sai của mỗi thành phần Gausssian trong định nghĩa mô hình Markov ẩn, để cho chúng gần bằng với kỳ vọng và phương sai của dữ liệu huấn luyện tiếng nói. HInit sẽ tính giá trị cho các HMM mới sử dụng công thức ước lượng như thuật toán Viterbi.

**3.1.3.3. Mô hình ngôn ngữ được dùng trong nhận dạng tiếng nói HTK**

Cũng giống như Sphinx, để nhận dạng tiếng nói đặt kết quả tốt, chúng ta cần phải áp dụng, xây dựng trong đó các văn phạm và ngữ nghĩa của ngôn ngữ. Đặc biệt, trong trường hợp nhận dạng tiếng nói liên tục dựa trên phone, kết quả nhận dạng được là một chuỗi các phone liên tiếp nhau. Vì thế để có kết quả ra các câu nói thông thường, chúng ta không thể đơn giản chỉ là tìm tương ứng với mỗi dãy phone là một từ đơn trong từ điển, bởi vì với chuỗi các phone tương ứng, kết quả tìm được có thể là một dãy các từ khác nhau.

Một mạng ngữ nghĩa biểu diễn cho dãy các từ có thể được nhận dạng thông qua hệ thống. Một mạng ngữ nghĩa biểu diễn ở mức độ từ là định nghĩa thứ tự có thể có của tất cả các từ trong từ điển với những câu khác nhau, nó chính là văn phạm của các từ trong câu. Trong mạng ngữ nghĩa, thông thường các đường truyền trạng thái giữa các từ với nhau được kèm theo một công thức xác suất, nó cho biết khả năng từ này tiếp sau từ kia là bao nhiêu trong ngôn ngữ. khi đó dựa vào mạng mà chúng ta có thể klấy được kết quả có nhiều khả năng nhất với một tập các phone được nhận dạng.

Trong HTK có nhiều công cụ để xây dựng mạng nói trên như: HParse, HSGen, HLStats. Trong đó:

* HParse: dùng để chuyển một tập tin văn phạm của từ điển thành một mạng ngữ nghĩa và khả năng xảy ra thứ tự các từ theo một trật tự.
* HGen: ngược lại với HParse, không những thế nó còn giúp phát sinh ra một dãy các câu dựa trên tập văn bản thu được.
* HLStats: đọc một danh sách các HMM và tập phiên âm lời nói. Công cụ này sẽ tính toán các thống kê khác nhau để phân tích dữ liệu huấn luyện ngữ âm và phát sinh các mô hình ngôn ngữ đơn giản dùng để nhận dạng.

**3.2. CMU Sphinx Framework**

**3.2.1. Giới thiệu CMU Sphinx.**

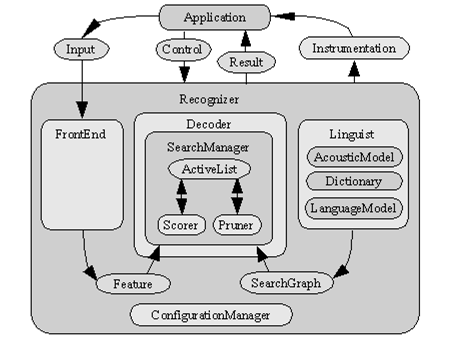
Sphinx là một hệ thống nhận dạng tiếng nói hoàn chỉnh được viết trên ngôn ngữ Java. Sphinx được công bố và chia sẽ mã nguồn lần đầu tiên vào năm 2010 do nhóm nghiên cứu Sphinx ở Carnegie Mellon University nghiên cứu và phát triển. Sau đó với sự giúp đỡ của các tổ chức (Sun Microsystems Laboratories, Hewlett Packard, Mitsubishi Electric Research Labs) và các trường đại học (University of California at Santa Cruz, Massachusetts Institute of Technology) Sphinx tiếp tục nghiên cứu và bổ sung các phiên bản mới cho Sphinx (Sphinx 2, Sphinx 3,..) và mới nhất là Sphinx 4, làm cho hệ thống nhận dạng ngày càng tối ưu hơn. Trong phạm vi báo cáo này, chúng ta sẽ tập trung tìm hiểu Sphinx 4.

**3.2.2. Các đặc điểm của CMU Sphinx**

Sphinx tuy mới được công bố để đưa vào các ứng dụng nhận dạng chưa lâu nhưng Sphinx đang dần trở thành một framwork nhận dạng tiếng nói mạnh mẽ và được sử dụng rất nhiều trong các ứng dụng trong cuộc sống. Đó là nhờ các đặc điểm sau:

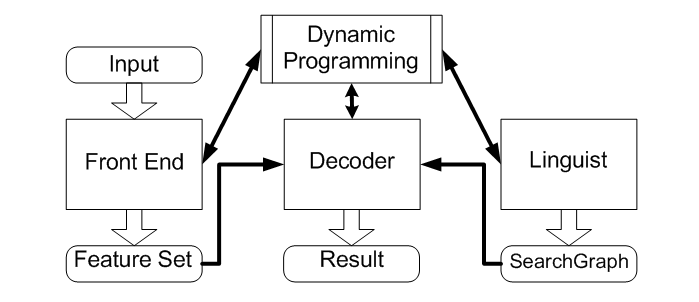
* -  Hỗ trợ nhận dạng tiếng nói ở chế độ trực tiếp hoặc chia lô, có khả năng nhận dạng  tiếng nói rời rạc và liên tục.
* -  Là một hệ thống nhận dạng đồ sộ nhưng có khả năng tháo lắp rất linh động. Hỗ trợ  sẵn đầy đủ các tính năng đáp ứng nhu cầu nhận dạng như xây dựng các bộ lọc, các hàm cửa sổ, các phép biến đổi, các công cụ hỗ trợ rút trích đặc trưng theo nhiều phương pháp khác nhau.
* -  Hỗ trợ nhiều mô hình ngôn ngữ dạng ASCII và các phiên bản nhị phân của unigram, bigram, trigram, Java Speech API Grammar Format(JSGF) và ARPA- format FST grammars.
* -  Xây dựng sẵn các thuật toán tìm kiếm tối ưu (breath first, word pruning), dễ dàng tinh chỉnh cho phù hợp với nhu cầu nhận dạng.
* **3.2.3. Kiến trúc Sphinx4**

Sphinx 4 là một framework nhận dạng âm thanh được thiết kế khá đồ sộ và phức tạp. Nó bao gồm các thành phần được ghép nối rất linh động với nhau và được công cụ hóa thành các bộ phận có chức năng riêng. Các thành phần này có thể được chỉnh sửa và kết nối với nhau theo nhu cầu của ứng dụng mà không phá vỡ cấu trúc của hệ thống.

**

*Hình 3.5 Kiến trúc tổng quát sphinx 4*

Có ba thành phần cơ bản cấu tạo nên Sphinx 4: Bộ ngoại vi (FrontEnd), Bộ giải mã (Decoder) và bộ ngôn ngữ (Linguist). Để dễ hình dung cách hoạt động tổng quát của Sphinx ta sẽ tìm hiểu sơ lược chức năng của các bộ phận trên. Bộ ngoại vi với các công cụ hổ trợ cho việc thu nhận và tiền xử lý các tín hiệu sẽ nhận và thực hiện tham số hóa chúng thành tập hợp các vecto đặc trưng (Feature). Bộ ngôn ngữ sẽ có nhiệm vụ đọc vào các mô hình ngôn ngữ với các thông tin cách phát âm trong từ điển và thông tin cấu trúc của các mô hình âm học rồi mô hình hóa chúng vào mộ đồ thị tìm kiếm (Search Graph). Bộ phận còn lại là bộ giải mã (Decoder) có nhiệm vụ quan trọng nhất kết nối hai bộ phận còn lại. Cụ thể là thành phần quản lý tìm kiếm (Search manager) trong bộ giải mã sẽ lấy các thông tin đặc trưng từ bộ ngoại vi rồi kết hợp với đồ thị tìm kiếm được phát sinh từ bộ ngôn ngữ để giả mã và tính toán ra kết quả nhận dạng.

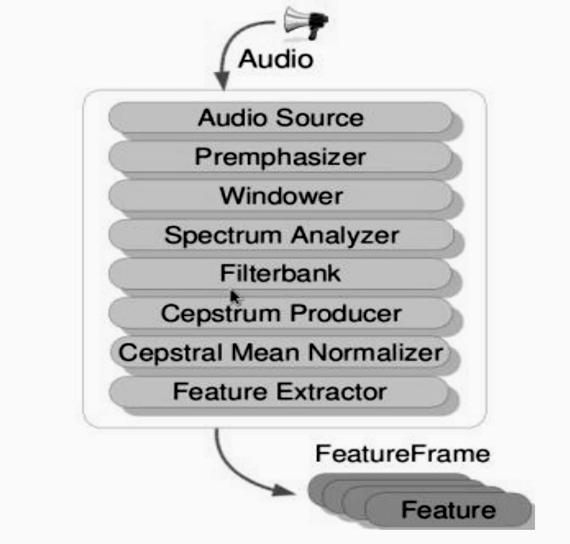
**

*Hình 3.6 Mô phỏng hoạt động của Sphinx*

Khi xây dựng Sphinx, các nhà phát triển đã nghiên cứu khá kỹ về lĩnh vực âm học nên đã xây dựng một số lượng lớn các hiệu số cấu hình để hiệu chỉnh chung cho phù hợp với nhu cầu của các ngôn ngữ và âm thanh khác nhau. Chúng ta có thể thay đổi, điều chỉnh chúng như thay đổi các cách rút trích đặc trưng, điều chỉnh các phương pháp tìm kiếm trên file config của hệ thống mà không cần phải can thiệp sâu vào bên trong. Bên cạnh đó nó còn xây dựng các tool hỗ trợ cho quá trình nhận dạng như các tool huấn luyện, các tool giám sát và báo cáo hệ thống.

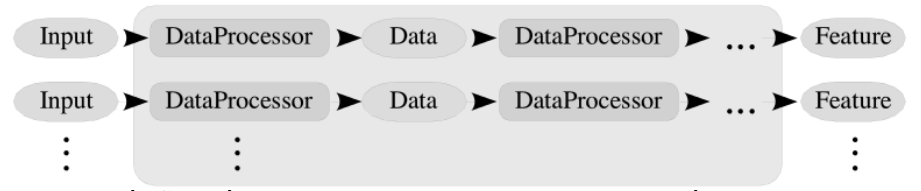
* **3.2.3.1. Bộ ngoại vi – FrontEnd**

**Chức năng:** Nhận vào các tín hiệu từ bên ngoại, thực hiện qua một số bộ lọc và xử lý dữ liệu để cho kết quả là một tập các Vecto đặc trưng.

**

*Hình 3.7 Kiến trúc tổng quát Bộ ngoại vi*

**Cấu tạo:** Bên trong của bộ ngoại vi này thưc chất là một chuỗi các modun con có khả năng xử lý tín hiệu giao tiếp được nối ghép với nhau. Các chuỗi này có thể hoạt động song song cùng một lúc với nhau gọi là Dataproccessor.

**

*Hình 3.8 Chuỗi các Dataprocessor trong quá trình xử lý*

Việc xử lý bằng các chuỗi Dataprocessor nối ghép với nhau sẽ giúp chúng ta linh động trong việc rút trích các thông tin đặc trưng, với một tín hiệu ta có thể rút trích đặc trưng bằng các phương pháp khác nhau để so sánh tìm các đặc trưng tối ưu hoặc áp dụng liên tiếp các phương pháp với nhau để tìm ra đặc trưng tốt nhất. Đồng thời cho phép các chuỗi này thực hiện song song với nhau làm cho hệ thống nhận diện nhanh chóng theo thời gian thực với lượng tín hiệu lớn thu vào.

* **3.2.3.2. Bộ ngôn ngữ - Linguist**

**Chức năng:** Bằng các công cụ và phương pháp ngôn ngữ, bộ này sẽ đọc vào các file cấu trúc của một ngôn ngữ rồi mô hình hóa chúng vào đồ thị tìm kiếm để sử dụng trong việc tìm kiếm nhận dạng.

**Cấu tạo:** Ở bộ này cấu tạo khá phức tạp vì nó quy định hầu như toàn bộ phạm vi ngôn ngữ mà chúng ta cần nhận dạng, nó gồm các thành phần nhỏ sau:

**a. Thành phần mô hình ngôn ngữ**

Thành phần này sẽ có chức năng đọc vào file cấu trúc ngôn ngữ ở cấp độ là các từ (word level). Thành phần này có vai trò quan trọng để xác định những thứ mà hệ thống cần nhận dạng. Cấu trúc ngôn ngữ sẽ được mô hình hóa ở thành phần này theo hai mô hình: graph-driven grammar và Stochastic N-Gram.

* -  Mô hình graph-driven grammar : biễu diễn một đồ thị từ có hướng trong đó mỗi nút biểu diễn một từ đơn và mỗi cung biễu diễn xác suất dịch chuyển sang một từ.
* -  Mô hình stochastic N-Gram : mô hình này cung cấp các xác suất cho các từ được cho dựa vào việc quan sát n-1 từ đứng trước.
  + 1. Sphinx hỗ trợ nhiều định dạng ngôn ngữ khác nhau như:
* -  SimpleWordListGrammar: định nghĩa một từ dựa trên một danh sách các từ.  Một tham số tùy chọn chỉ ra ngữ pháp có lặp hay không. Nếu ngữ pháp không lặp, ngữ pháp sẽ được dùng cho một nhận dạng từ tách biệt. Nếu ngữ pháp lặp, nó sẽ được dùng để hỗ trợ liên kết nhận dạng từ tầm thường, tương đương với một unigram grammar với xác suất bằng nhau.
* -  JSGFGrammar: Hỗ trợ JavaTM Speech API Grammar Format (JSGF), định nghĩa một biểu diễn theo BNF, độc lập nền tảng, Unicode của các ngữ pháp.
* -  LMGrammar: định nghĩa một ngữ pháp dựa trên một mô hình ngôn ngữ  thống kê. LMGrammar phát sinh một nút ngữ pháp mỗi từ và làm việc tốt với  các unigram và bigram, xấp xỉ 1000 từ.
* -  FSTGrammar: hỗ trợ một bộ chuyển đổi trạng thái giới hạn (finite-state  tranducer) trong định dạng ngữ pháp ARPA FST.
* -  SimpleNGramModel: cung cấp hỗ trợ cho các mô hình ASCII N-Gram  trong định dạng ARPA. SimpleNGramModel không cố làm tối ưu việc sử dụng  bộ nhớ, do đó nó làm việc tốt với các mô hình ngôn ngữ nhỏ.
* -  LargeTrigramModel: cung cấp hỗ trợ các mô hình N-Gram đúng được phát sinh bởi CMU-Cambridge Statictical Language Modeling Toolkit. LargeTrigramModel tối ưu việc lưu trữ bộ nhớ, cho phép nó làm việc với các  tập tin rất lớn, trên 100MB.

**b. Thành phần từ điển**

Thành phần này cung cấp cách phát âm cho các từ ta đã xây dựng trong mô hình ngôn ngữ và hỗ trợ phân lớp các từ thành các lớp khác nhau để hỗ trợ cho việc tìm kiếm.

**c. Thành phần mô hình âm học**

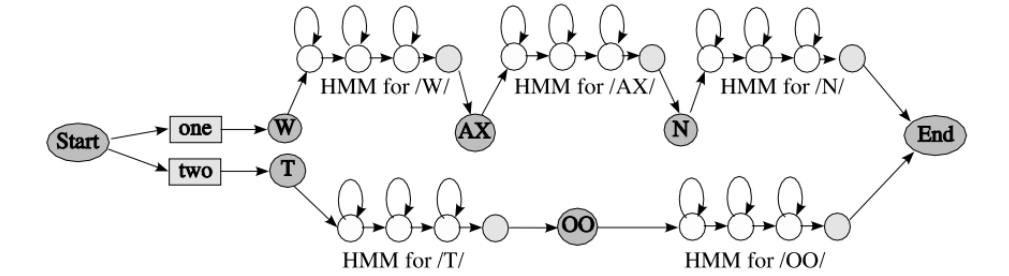
Thành phần này mô hình âm học cung cấp một ánh xạ giữa một đơn vị tiếng nói và một HMM (Hidden Markov Model) có thể được đánh giá dựa vào các đặc trưng được cung cấp bởi bộ ngoại vi. Các ánh xạ có thể đưa thông tin vị trí của từ và ngữ cảnh từ thành phần mô hình ngôn ngữ. Định nghĩa ngữ cảnh này được xây dựng từ cấu trúc ngữ pháp của mô hình ngôn ngữ

Khác với các hệ nhận dạng khác, mô hình HMM trong Sphinx có cấu trúc cố định trong bộ nhớ và đơn thuần chỉ là một đồ thị có hướng của các đối tượng. Trong đồ thị này, mỗi nút tương ứng với một trạng thái HMM và mỗi cung biễu diễn xác suất biến đổi từ trạng thái này sang trạng thái khác trong HMM. Bằng cách biểu diễn HMM như là các đồ thị có hướng của các đối tượng thay vì một cấu trúc cố định, một bổ sung của mô hình âm học có thể dễ dàng cung cấp các HMM với các dạng hình học tôpô khác.

Mỗi trạng thái HMM có khả năng phát sinh một đánh giá từ một đặc trưng quan sát. Quy tắc để tính toán điểm số được thực hiện bởi chính trạng thái HMM, do đó che dấu các thực thi của nó đối với phần còn lại của hệ thống, thậm chí cho phép các hàm mật độ xác suất khác nhau được sử dụng trên mối trạng thái HMM. Mô hình âm học cũng cho phép chia sẻ các thành phần khác nhau trên tất cả các cấp độ. Nghĩa là các thành phần tạo nên một trạng thái HMM như các hợp Gaussian (Gaussian mixture), các ma trận biến đổi và các trọng số hỗn hợp (mixture weight) có thể được chia sẽ bởi bất kỳ trạng thái HMM nào.

**d. Đồ thị tìm kiếm – Search Graph**

Đây là kết quả mà bộ ngôn ngữ phát sinh được cuối cùng để đưa vào sử dụng trong bộ giải mã. Độ thị tìm kiếm này là một đồ thị có hướng, trong đó mỗi nút được gọi là một trạng thái tìm kiếm (SearchState): biểu diễn một trong hai trạng thái: phát hoặc không phát (emitting state hay non-emitting state). Và các đường cung biểu diễn các trạng thái biến đổi có thể, trên các cùng này có các giá trị xác suất được tình toán từ mô hình âm học: biểu diễn khả năng chuyển từ trạng thái này đến trạng thái kia. Một trạng thái có thể có nhiều cung hướng đến trạng thái khác.

**

*Hình 3.9 Ví dụ một đồ thị tìm kiếm đơn giản*

**3.2.3.3. Bộ giải mã – Decoder**

**Chức năng:** Bộ giải mã có nhiệm vụ chính là sử dụng các đặc trưng (Features) từ bộ ngoại vi kết hợp với đồ thị tìm kiếm được phát sinh từ bộ ngôn ngữ để tiến hành giải mà và áp dụng các thuật toán suy ra kết quả nhận dạng

**Cấu tạo:** Thành phần của bộ này tuy đơn giản nhưng rất phức tạp về thuật toán vì ảnh hưởng tới tốc độ tìm kiếm, gồm một thành phần quản lý tìm kiếm (SearchManager) có khả năng tháo lắp và các mã hỗ trợ khác để đơn giản hóa quá trình giải mã cho một ứng dụng. Nên trong báo cáo chỉ trung giới thiệu thành phần quản lý tìm kiếm.

Nhiệm vụ của thành phần quản lý tìm kiếm là nhận dạng các tập các vecto đặc trưng để tìm ra ánh xạ tương ứng của nó trong đồ thị tìm kiếm. Để đáp ứng tìm ra kết quả chính xác trong đồ thị tìm kiếm khi xử lý kết quả, Sphinx cung cấp các tiện ích có khả năng phát sinh lưới và các đánh giá độ tin cậy từ kết quả. Và thêm đặc điểm nữa khác các hệ thống khác là không gian tìm kiếm trong Sphinx có thể được tinh chỉnh thay đổi trong quá tình tìm kiếm để tăng hiệu suất tìm kiếm.

Ngoài ra để nâng cao hiệu suất của kết quả nhận dạng, Sphinx con bổ sung thêm các công cụ hỗ trợ cho việc đánh giá kết quả nhận được, đó là thành phần đánh giá (Scorer) và thành phần cắt tỉa (Pruner). Nói về thành phần Scorer thì nó là một module dùng để ước lượng xác suất của trạng thái khi cung cấp các giá trị mật độ trạng thái xuất hiện. Khi thành phần quản lý tìm kiếm yêu cầu đánh giá điểm số cho một trạng thái, nó sẽ gọi đến thành phần Scorer, nó sẽ phân tích các thông tin đặc trưng của trạng thái đó rồi áp dụng các phép toán để tính điểm số.

**CHƯƠNG 4. XÂY DỰNG CHƯƠNG TRÌNH THỬ NGHIỆM**

**4.1. Cài đặt Sphinx**

**4.1.1. Chuẩn bị hệ điều hành**

**4.1.2. Chuẩn bị các gói cài đặt Sphinx**

**4.1.3. Cài đặt Sphinx**

**4.2. Chuẩn bị bộ huấn luyện cho Sphinx**

**4.3. Tiến hành huấn luyện mô hình bằng Sphinx**

**4.4. Cài đặt HTK**

**4.5. Triển khai ứng dụng demo**

**4.5.1. Ứng dụng điều khiển Google Chrome**

**a. Giớithiệu**

Đây là ứng dụng demo sử dụng giọng nói để điểu khiển trình duyệt web Google Chrome, người dùng sẽ thao tác với trình duyệt thông qua giọng nói, demo hỗ trợ thực hiện khoảng 20 câu lệnh điều khiển, được ghép từ 47 từ đơn.

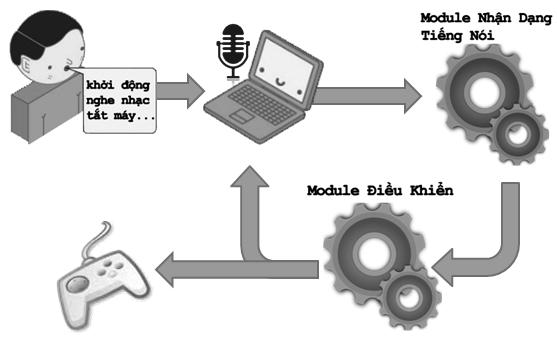
*Bảng 5 Danh sách 47 từ trong điều khiển Google Chrome*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bản | duyệt | mở | sang | to | xóa |
| chuyển | hãy | mới | sau | tới | xuống |
| cửa | khóa | nghe | sổ | tra |  |
| cuối | kiếm | ngừng | sử | trái |
| cuộn | kiểm | nhạc | tải | trang |
| đầu | lại | nhỏ | táp | trình |
| đi | lên | phải | thu | trở |
| đồ | lịch | phóng | thư | trước |
| đóng | lưu | quép | tìm | từ |

Từ các từ đơn này, ta có thể ghéo lại thành các câu lệnh điều khiển có nghĩa, vd: mở táp (tab) mới, mở cửa sổ, mở trang quép (web). Danh mục các câu điều khiển có thể tham khảo thêm trong phần phụ lục.

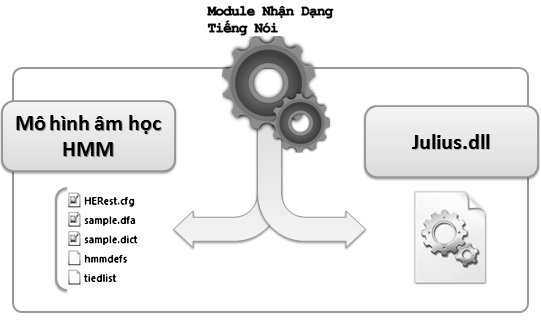
**b. Chương trình**

Chương trình được viết bằng ngôn ngữ C#, kết hợp thư viện Julius.dll và mô hình âm học được huấn luyện từ công tụ HTK, bao gồm 2 module chính đó là: module nhận dạng và module điều khiển trình duyệt.



*Hình 4.2 Mô hình hoạt động của chương trình demo*

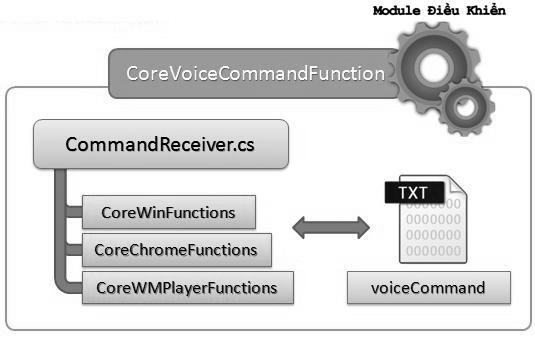
Module nhận dạng sử dụng các hàm do bộ thư viện Julius.dll cung cấp để thực hiện công đoạn nhận dạng, sử dụng mô hình âm học được huấn luyện phần trên (sử dụng từ phần huấn luyện của HTK), kết hợp với mô hình ngôn ngữ. Sau khi nhận dạng ra kết quả sẽ chuyễn sang dạng text (ký tự) và cuối cùng chuyễn nội dung nhận dạng được sang *module điều khiển.*



Hình 4.3 Mô hình sơ đồ Module nhận dạng

Module điều khiểu sau khi được gọi và truyền vào tham số là kết quả tiếng nói nhận dạng được sẽ thực hiện công đoạn điều khiển dựa vào nội dung đó. Module này được xây dựng như một thư viện liên kết động (dll) để tiện cho việc nâng cấp điều khiển sau này. Hiện tại, chương trình demo chỉ có thể điểu khiển được vài chức năng cơ bản của Hệ điều hành, trình duyệt Google Chrome và trình nghe nhạc WMPlayer. Trong Module này gồm 1 class chính là CommandReceiver.cs. Class này đảm nhận nhiệm vụ nhận lệnh dưới dạng text để thực hiện lệnh điều khiển. Để hổ trợ việc ánh xạ các câu lệnh điều khiển thành lệnh điều khiển thực sự, module này liên kết với một tập tin danh sách lệnh mang tên voiceCommand.txt. Danh sách các câu lệnh trong đây sẽ được nạp vào chương trình lúc khởi động, sau khi nhận được nội dung dưới dạng text cho người dùng ra lệnh, module nhận dạng sẽ tiến hành tra khớp các lệnh có trong cơ sở dữ liệu lệnh vừa được nạp vào, từ đó đưa ra được lệnh cần tiến hành thực sự. Nội dung danh sách lệnh được bổ sung trong phần phụ lục.

Trong Module này còn bao gồm 3 class chính thực hiện nhiệm vụ điều khiển cho 3 đối tượng chính của demo này đó là window, chrome và trình nghe nhạc WMPlayer. Để điều khiển được trình duyệt Chrome cũng như các chương trình khác, chương trình truyền cho hệ điều hành Windows những sự kiện bàn phím, những sự kiện bàn phím này chính là các tổ hợp phím tắt đề sử dụng Chrome (vd: Ctrl + T: mở tab mới, Ctrl + W: đóng tab).

**

*Hình 4.4 Mô hình cấu tạo của Module điều khiển*

**

*Hình 4.5 Giao diện chương trình điều khiển máy tính*

**c. Nhận xét**

Kết quả nhận dạng được từ module nhận dạng quyết định độ chính xác rất cao của chương trình, nhiễu khi nhận dạng sẽ gây ra câu lệnh bị sai lệnh đôi chút so với câu lệnh thực tế, ví dụ: người dùng nói “mở táp mới”, những do nhiễu môi trường nên kết quả nhận dạng có thể trả về “hãy mở táp mới”. Biện pháp khác phục đơn giản nhất đó chính là so sánh độ “giống nhau” giữa câu lệnh nhận dạng được và câu lệnh trong mẫu câu thay vì so khớp chính xác từng từ. Nhìn chung kết quả nhận dạng khá chính xác trong môi trường không quá ồn (~90%).

**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1. Kết quả đạt được**

Sau quá trình tìm hiểu, nghiên cứu phương pháp nhận dạng giọng nói tiếng Việt, khóa luận đã đạt được những mục tiêu đề ra như sau:

* Tìm hiểu các khái niệm có liên quan đến hệ nhận dạng tiếng nói từ đó hiểu và vận dụng được một số yếu tố quan trọng trong việc sử dụng công cụ hỗ trợ.
* Tìm hiểu phương pháp cài đặt công cụ hỗ trợ xây dựng hệ nhận dạng tiếng nói Sphinx và HTK. Trong đó bao gồm việc chi tiết hóa từng bước cài đặt và thực hiện. Các bước xây dựng mô hình huấn luyện.
* Xây dựng bộ huấn luyện với dữ liệu thu âm ~ 15 giờ đồng hồ.
* Thực nghiệm so sánh khả năng áp dụng vào tiếng Việt của 2 công cụ Sphinx và HTK, trong đó bao gồm cả thử nghiệm các công cụ decode khác nhau như Hvite, Sphinx4 và Julius để từ đó rút ra được kết luận và nhận xét.
* Xây dựng chương trình mô phỏng nhận dạng tiếng nói tiếng Việt trên máy tính sử dụng mô hình âm học được huấn luyện bằng công cụ Sphinx, chương trình nhận dạng được trên 60 từ đơn, được ghép thành khoảng 100 câu lệnh điều khiển. Ngoài ra, nhóm còn thực hiện demo trên mô hình thật, vận dụng khả năng của mạch điều khiển vào việc điểu khiển mô hình thật. Kết quả đạt được trong demo đạt được tỉ lệ chính xác rất cao.

**5.2. Những điểm còn hạn chế**

Do đây là một đề tài tương đối khó, và nhóm cũng chưa được trang bị các kiến thức về xử lý tín hiệu số, xử lý tiếng nói, các mô hình toán học, cũng như tài liệu về âm học, ngữ âm học còn hạn chế nên khóa luận không tránh khỏi nhiều thiếu sót.

Những mặt còn hạn chế của khóa luận:

* Bộ từ vựng còn quá ít, so vớI tất cả từ đơn của tiếng Việt (hơn7000từ). Để có được bộ tự vựng lớn hơn đòi hỏi phải bỏ ra rất nhiều công sức bao gồm cả thu thập dữ liệu, xây dựng mô hình phiên âm chính xác, thu âm, phân tích ngữ pháp
* Mô hình ngữ âm còn hạn chế, hệ thống chỉ nhận dạng được với độ chính xác cao đối với 2 thành viên trong nhóm, đối với một người chưa được huấn luyện thì mô hình vẫn có thể nhận dạng được tiếng nói của họ nhưng độ chính xác không cao.
* demo của khóa luận chỉ tập trung xây dựng nhằm mục đích thể hiện khả năng ứng dụng của mô hình nhận dạng tiếng nói tiếng Việt tự động được xây dựng từ các framework. Tính ứng dụng thực tế vẫn còn hạn chế, tuy nhiên, để xây dựng 1 ứng dụng mang tính thực tiễn thật sự không quá khó, vấn đề quan trọng là đặt ra được mục tiêu thực tiển của 1 ứng dụng cụ thể, từ đó xây dựng mô hình âm học thích hợp.

**5.3. Hướng nghiên cứu và phát triển**

Hiện tại, nhóm đã thành công trong việc thử nghiệm mô hình từ điển tiếng Việt do nhóm tham khảo được từ nhiều nguồn (mô hình này chưa thật sự chuẩn xác, nhưng khả năng áp dụng ở mức chấp nhận được) vào các công cụ xây dựng hệ thống nhận dạng tiếng nói tự động, bao gồm HTK và Sphinx. Nhóm đã thực hiện thí nghiệm trên các công cụ decode khác nhau như Hvite, Julius, Sphinx4, và nhận thấy kết quả rất khả quan, đứng tại mức thành quả đạt được này sẽ có rất nhiều hướng nghiên cứu mới có thể được thực hiện cụ thể như:

* Tìm hiểu, nghiên cứu xây dựng mô hình phát âm tiếng Việt, điều này có ý nghĩa to lớn cho cả việc tổng hợp và nhận dạng giọng nói. Mô hình này nếu được phát triển một cách khoa học và đúng đắn sẽ đưa độ chính xác nhận dạng của các mô hình âm học được xây dựng bởi các công cụ lên rất nhiều. Công việc này đòi hỏi sự nghiên cứu của các nhà âm học, ngữ âm học, nghiên cứu về tiếng Việt.
* Tìm hiểu sâu hơn về cấu tạo của các framework nhận dạng giọng nói trên, để từ đó, nắm được quy trình hoạt động một cách rõ ràng hơn, ta có thể xây dựng được một mô hình âm học thích hợp nhất cho tiếng Việt.
* Mở rộng vốn từ vựng của bộ tự điển, thực hiện thu âm số với quy mô rộng rãi hơn, đa dạng giọng nói hơn, nhằm mục đích xây dựng một hệ nhận dạng tiếng nói đọc lập người nói.
* Xây dựng các ứng dụng cụ thể hóa hơn sử dụng các mô hình đã được huấn luyện. Các ứng dụng tương tác giữa người và thiết bị bằng giọng nói, hỗ trợ sự thông minh cho thiết bị hoặc hỗ trợ hoạt động của người khuyết tật.

**TÀI LIỆU THAM KHẢO**

[1]  B.H. Juang, Lawrence R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology".

[2]  S. Furui, "50 years of progress in speech and speaker recognition".

[3]  [Online]. Available: http://www.cslu.ogi.edu/toolkit/. [Accessed 7 2012].

[4]  L. C. Mai, "Phát triển các kết quả tổng hợp, nhận dạng câu lệnh, chuỗi số tiếng Việt liên tục trên môi trường điện thoại di động," 2006.

[5]  Đăng Ngọc Đức, Lương Chi Mai, "Tăng cường độ chính xác của hệ thống mạng neuron nhận dạng tiếng Việt," 2003.

[6]  B. H. Khang, "Báo cáo tổng kết Khoa học và Kỹ thuật đề tài Nghiên cứu phát triển công nghệ nhận dạng, tổng hợp và xử lý ngôn ngữ tiếng Việt," 2004.

[7]  "Vietnamese alphabet," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Vietnamese\_alphabet. [Accessed 7 2012].