BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

PHAN QUỐC TUẤN

PHÁT HIỆN TỰ ĐỘNG MỘT SỐ LỖI PHÁT ÂM TIẾNG ANH CỦA NGƯỜI HỌC

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công nghệ Thông Tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 3 năm 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

PHAN QUỐC TUẤN

PHÁT HIỆN TỰ ĐỘNG MỘT SỐ LỖI PHÁT ÂM TIẾNG ANH CỦA NGƯỜI HỌC

LUẬN VĂN THẠC SĨ

Chuyên ngành: Công nghệ Thông Tin

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: TS. Đặng Thanh Dũng

TP. HỒ CHÍ MINH, tháng 3 năm 2016

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học:TS. Đặng Thanh Dũng

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày … tháng … năm …

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ Luận văn Thạc sĩ)

TT	Họ và tên	Chức danh Hội đồng
1		Chủ tịch
2		Phản biện 1
3		Phản biện 2
4		Ủy viên
5		Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày..… tháng….. năm 20…..

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Phan Quốc Tuấn Giới tính:Nam

Ngày, tháng, năm sinh: 04/01/1988 Nơi sinh:Bến Tre

Chuyên ngành: Công nghệ Thông Tin MSHV:1341860030

I- Tên đề tài:

Phát hiện tự động một số lỗi phát âm Tiếng Anh của người học

II- Nhiệm vụ và nội dung:

Tìm hiểu các kiến thức về ngữ âm học, âm vị học, các kỹ thuật xử lý tiếng nói để xây dựng một cơ chế xử lý tiếng nói thích hợp giúp phát hiện một cách tự động một số lỗi phát âm Tiếng Anh của người học.

III- Ngày giao nhiệm vụ: *15/8/2014*

IV- Ngày hoàn thành nhiệm vụ: 15/06/2015

V- Cán bộ hướng dẫn:*(Ghi rõ học hàm, học vị, họ, tên)* Tiến Sĩ Đặng Thanh Dũng

CÁN BỘ HƯỚNG DẪN
(Họ tên và chữ ký)

KHOA QUẢN LÝ CHUYÊN NGÀNH
(Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

(Ký và ghi rõ họ tên)

LỜI CÁM ƠN

Với lòng biết ơn sâu sắc nhất , tôi xin gửi tới tập thể quý thầy cô khoa Công nghệ Thông tin trường Đại học Công nghệ TP. HCM, những người đã truyền đạt cho tôi rất nhiều kiến thức quý báu trong thời gian tôi học tập tại trường.

Tôi cũng xin chân thành bày tỏ lòng biết ơn sâu sắc tới TS. Đặng Thanh Dũng – người thầy trực tiếp hướng dẫn và chỉ bảo cho tôi thực hiện luận án này. Thầy là người đã định hướng, giúp đỡ tôi rất nhiều trong nghiên cứu khoa học. Nếu không có sự hướng dẫn tận tình của thầy thì sẽ rất khó khăn để tôi có thể hoàn thành luận văn thạc sỹ này. Một lần nữa, tôi xin chân thành cảm ơn thầy.

Tôi xin chân thành cảm ơn bạn bè và đặt biệt là gia đình đã luôn ở bên tôi; động viên, khích lệ, tạo điều kiện và giúp đỡ tôi trong suốt quá trình thực hiện và hoàn thành luận án này.

Phan Quốc Tuấn

TÓM TẮT

Trong luận văn này, tác giả khảo sát một phương pháp phát hiện tự động lỗi phát âm tiếng Anh. Để đạt được mục tiêu này, tác giả tìm hiểu một số kiến thức về âm vị học, trên cơ sở đó, chỉ ra một số lỗi phát âm thường gặp của người Việt. Tác giả sử dụng các bộ nhận dạng SVM đã được huấn luyện dựa trên vector đặc trưng gồm 39 hệ số đặc trưng ngữ âm và 3 formant (tổng cộng 42 hệ số) trên một frame có chiều dài 25ms. Việc tính toán vetor đặc trưng được thực hiện sau mỗi 10ms. Các thư viện được sử dụng trong luận văn này gồm: HTK, SVM-Light Toolkit, Praat. Kết quả từ thí nghiệm cho thấy rằng dùng các SVM với vector đặc trưng nêu trên cho phép đạt được độ chính xác phát hiện lỗi tương đối cao trên hai tập dữ liệu Buckeye (tập dữ liệu huấn luyện) và TIMIT (tập dữ liệu đánh giá).

ABSTRACT

In this thesis, the author presents a method that automatically detects English pronunciation errors. To achieve this goal, the author investigates knowledge of phonology, based on that, pointing out some common English pronunciation errors of the Vietnamese learners. The author uses the trained SVM classifiers based on feature vectors that contains 39 acoustic feature coefficients and 3 formants (total of 42 coefficients) on a 25ms frame. The feature vectors is calculated after each 10ms. The libraries are used in this thesis include HTK, SVM-Light Toolkit, Praat. The result from the experiment suggests that using the SVMs based on the feature vectors can achieve relatively high error detection accuracy on the two datasets: Buckeye corpus (training data set) and TIMIT corpus(testing data set).

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CÁM ƠN ii
TÓM TẮT iii
ABSTRACT iv
MỤC LỤC v
DANH MỤC CÁC TỪ VIẾT TẮT viii
DANH MỤC CÁC BẢNG ix
DANH MỤC CÁC BIỂU ĐỒ, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH x
CHƯƠNG 1 - MỞ ĐẦU 1
1.1 Đặt vấn đề 1
1.2 Tính cấp thiết của đề tài 1
1.3 Mục tiêu, đối tượng và phạm vi nghiên cứu 3
1.3.1 Mục tiêu của đề tài 3
1.3.2 Đối tượng và phạm vi nghiên cứu 3
CHƯƠNG 2: TỔNG QUAN 4
2.1 Các nghiên cứu liên quan 4
2.1.1 Phát hiện lỗi dựa trên xác suất (likelihood-based scoring) 4
2.1.2 Phương pháp độc lập ngôn ngữ thứ nhất (L1-independent) 5
2.1.3 Phương pháp phụ thuộc ngôn ngữ thứ nhất (L1-dependency) 5
2.1.4 Phát hiện lỗi dựa trên bộ phân loại (classifier-based scoring) 5
2.1.5 Mô hình tiếng nói do người nước ngoài phát âm (non-native acoustic modeling) 6
2.1.6 Phát hiện lỗi phát âm độc lập với văn bản(text independence) 6
2.1.7 Phát hiện và phản hồi lỗi về nhịp điệu phát âm(prosodic pronunciation error) 7

2.1.8Thiết kế hệ thống CAPT có tính tương tác (Interactive CAPT system design) 7
2.2Các vấn đề còn tồn tại 8
2.3Phương hướng giải quyết của nghiên cứu này 9
CHƯƠNG 3: CƠ SỞ LÝ THUYẾT 10
3.1Cơ bản về ngữ âm học và âm vị học 10
3.1.1Ngữ âm học và âm vị học 10
3.1.2Âm vị (phoneme) và âm tố (phone hay speech sound) 11
3.1.3Phụ âm (consonant) và nguyên âm (vowel) 11
3.1.4Ví trí phát âm (place of articulation) 12
3.1.5Cách thức phát âm (manner of articulation) 15
3.1.6Hình thang nguyên âm 17
3.1.7Âm hữu thanh (voice) và âm vô thanh (voiceless) 18
3.1.8Tha âm vị (allophone) 19
3.1.9Hệ thống âm vị tiếng Việt 19
3.1.10Hệ thống âm vị tiếng Anh 22
3.2Xác định một số lỗi sai thường gặp của người Việt học tiếng Anh 22
3.3Cơ bản về xử lý tiếng nói 23
3.3.1Spectrogram 24
3.3.2Formant 25
3.3.3Đặc trưng ngữ âm (Acoustic feature) 27
3.4Support Vector Machine 27
3.4.1Các khái niệm cơ bản 28
3.4.2Cực đại hóa bộ phân loại hậu nghiệm (classifier posterior) 30
3.4.3Cực tiểu hóa rủi ro về mặt cấu trúc 30
CHƯƠNG 4: THÍ NGHIỆM VÀ ĐÁNH GIÁ 37
4.1Mô tả các kho dữ liệu được sử dụng trong thí nghiệm 37
4.1.1Kho dữ liệu TIMIT 37
4.1.2Mô tả bộ dữ liệu mẫu của TIMIT 38

4.1.3Kho dữ liệu Buckeye 38
4.2Các thư viện và công cụ dùng trong thí nghiệm.....42
4.2.1Thư viện HTK và công cụ HCopy.....42
4.2.2Thư viện SVM 44
4.2.3Praat 45
4.3Huấn luyện các SVM.....46
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....52
DANH MỤC TÀI LIỆU THAM KHẢO.....56
PHỤ LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
AF	Acoustic feature (đặc trưng ngữ âm)
HTK	Hidden Markov Model Toolkit
SVM	Support Vector Machine
L1	Ngôn ngữ mẹ đẻhay ngôn ngữ thứ nhất
L2	Ngoại ngữ hay ngôn ngữ thứ 2 (không phải ngôn ngữ mẹ đẻ)
ESL	English as a Second Language
FAR	False Acceptance Rate
SAR	Successful Acceptance Rate
RBF	Radial Basis Function
VPM	Voice, Place, Manner

DANH MỤC CÁC BẢNG

Bảng 3.1– Bảng tổng hợp vị trí phát âm và cách thức phát âm của các âm vị . **Error! Bookmark not defined.**
Bảng 3.2 – Hệ thống âm đầu tiếng Việt 19
Bảng 3.3 – Hệ thống nguyên âm tiếng Việt 20
Bảng 3.4 – Hệ thống âm cuối tiếng Việt 20
Bảng 3.5 – Các phụ âm trong tiếng Anh (được phân loại dựa vào VPM) 22
Bảng 3.6 – Các âm vị tiếng Anh không có trong tiếng Việt 23
Bảng 3.7 – Một số lỗi phát âm sẽ khảo sát trong luận văn. 23
Bảng 4.1 – Các loại tập tin trong kho dữ liệu Buckeye 39
Bảng 4.2 – Ý nghĩa các tham số được dùng để tính AF dùng thư viên HTK 43
Bảng 4.3 – Ý nghĩa các tham số phụ đi kèm với tham số TARGETKIND 44
Bảng 4.4 - Độ chính xác phát hiện lỗi sai khi huấn luyện dữ liệu trên Buckeye 49
Bảng 4.5 - Độ chính xác phát hiện lỗi sai khi huấn luyện dữ liệu trên TIMIT 51
Bảng 4.6 - So sánh độ chính xác phát hiện lỗi trên các mô hình khác nhau 51
Bảng 5.1 – Các kho dữ liệu đã tìm hiểu 54
Bảng 7.1 - Kí hiệu nhấn âm 60
Bảng 7.2 – Nguyên âm đơn 61
Bảng 7.3 – Nguyên âm đôi 62
Bảng 7.4 – Phụ âm dừng (stop) 63
Bảng 7.5 – Phụ âm tắt sát (affricate) 63
Bảng 7.6 – Phụ âm sát (fricative) 64
Bảng 7.7 – Âm mũi (nasal) 64
Bảng 7.8 – Âm nước (liquid) 65
Bảng 7.9 – Bán nguyên âm (semivowel) 65

DANH MỤC CÁC BIỂU ĐỒ, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH

Hình 3.1 – Vị trí phát âm của âm môi....12
Hình 3.2 – Vị trí phát âm của cuối lưỡi 13
Hình 3.3 – Các vị trí khác nhau trong hệ thống phát âm 14
Hình 3.4 – Các loại phụ âm tương ứng với các vị trí phát âm....14
Hình 3.5 – Sự khác nhau giữa âm mũi (phải) và âm miệng (trái) 15
Hình 3.6 – Hình thang nguyên âm 17
Hình 3.7 – Sự khác nhau giữa 2 âm tiếng Anh [iy] (trái) và âm [uw] (phải). 18
Hình 3.8 – Sự khác nhau giữa hai âm vị tiếng Anh [ae] (trái) và [aa] (phải) 18
Hình 3.9 – Sơ đồ về 3 tiêu chí khu biệt cho sáu âm vị thanh điệu....21
Hình 3.10 – Lăng trụ thanh điệu 21
Hình 3.11 – Biểu đồ thanh điệu 21
Hình 3.12 – Spectrogram gồm 2 chiều: tần số (spectrum) và thời gian 24
Hình 3.13 – Spectrogram của câu nói "She came back and started again" 24
Hình 3.14 – Sóng âm, spectrogram, và phiên âm ở mức âm vị và mức landmark. ..25
Hình 3.15 – Ba formants được thể hiện trong spectrogram....26
Hình 3.16 – Hai formant trong spectrogram của ba từ "bad", "dad" và "gag" 26
Hình 3.17 – Mel-scale spectrogram của phone /b/....28
Hình 3.18 – Véc tơ hóa mel-scale spectrogram của phone /b/ 29
Hình 3.19 – Kết quả sử dụng SVM tuyến tính trên dữ liệu kiểm tra (test data)....34
Hình 3.20 – So sánh kết quả SVM tuyến tính trên dữ liệu huấn luyện và dữ liệu kiểm tra....34
Hình 3.21 – Đường ranh giới (boundary) của phân loại RBF-SVM 36
Hình 4.1 – Giao diện trang web tải kho dữ liệu Buckeye....40
Hình 4.2 – Hệ thống tập tin đã được tải về đĩa 41
Hình 4.3 – Nội dung của một tập tin phiên âm ở mức âm vị (.phones)....42
Hình 4.4 – Ví dụ minh họa tập tin SVM đầu vào 45
Hình 4.5 – Sơ đồ tổng quát của quá trình xử lý tiếng nói trong thí nghiệm 46
Hình 4.6 – Quá trình huấn luyện một SVM và các dữ liệu cần thiết....47

CHƯƠNG 1 - MỞ ĐẦU

1.1 Đặt vấn đề

Các hệ thống CAPT (Computer-Assisted Pronunciation Training) có thể cung cấp nhiều lợi ích cho người học tiếng Anh. Chúng có thể cung cấp thông tin phản hồi (feedback) cho người học mà không đòi hỏi thời gian và công sức của giáo viên. Chúng cũng có thể hỗ trợ quá trình tự học và khuyến khích người học sử dụng tiếng Anh bất kỳ khi nào người học có thời gian rảnh và giúp người học vượt qua rào cản của sự thiếu tự tin, mắc cỡ vì sợ phát âm sai.

Để có thể mang lại lợi ích lớn nhất đối với người học, CAPT cần có khả năng chẩn đoán (tự động) một cách nhanh chóng, chính xác các lỗi phát âm của người học, đồng thời chỉ ra và điều chỉnh lỗi này để người học nhận biết chỗ sai của mình và định hướng được làm thế nào để phát âm đúng. Điều này đặc biệt có ích cho người tự học, vì thông thường họ sẽ không tự nhận biết được các lỗi trong phát âm của họ để khắc phục. Việc phát âm sai gây khó hiểu cho người nghe, dẫn đến giao tiếp (bằng tiếng Anh) kém hiệu quả.

Trong phạm vi luận văn này, tác giả sẽ giải quyết các vấn đề sau:

-Xác định một số lỗi phát âm tiếng Anh thường gặp của người học tiếng Anh, đặc biệt là người Việt.

- Sử dụng các kỹ thuật xử lý tiếng nói, khảo sát mô hình xác định tự động các lỗi cơ bản nêu trên.

- Tiến hành thử nghiệm mô hình trên các tập dữ liệu lớn đáng tin cậy.

1.2 Tính cấp thiết của đề tài

Việc phát âm đúng tiếng Anh sẽ giúp người học giao tiếp hiệu quả và tự tin hơn. Tuy nhiên, do bị ảnh hưởng bởi ngôn ngữ mẹ đẻ và các thói quen hình thành khi phát âm tiếng Việt, chúng ta thường có khuynh hướng rơi vào một số lỗi chung khi

phát âm tiếng Anh. Chẳng hạn bỏ sót âm vị cuối (ví dụ bỏ âm vị /t/ trong từ ‘mount’), phát âm sai âm vị /r/ trong từ ‘right’ (vì tiếng Việt không có âm vị này), v.v…

Để có thể khắc phục các lỗi này, cần phải có người phát âm đúng thường xuyên chỉ ra các lỗi phát âm sai của người học, từ đó người học có thể rèn luyện và bỏ các thói quen dẫn đến phát âm sai. Việc này đòi hỏi nhiều thời gian, đặc biệt là khi người học không có điều kiện để giao tiếp với người phát âm đúng và không sống trong môi trường nói tiếng Anh. Đặc biệt, đối với những người tự học, không có điều kiện để nhận được sự chỉ dẫn từ giáo viên, việc khắc phục các lỗi phát âm sẽ trở nên khó khăn hơn rất nhiều.

Do vậy, một phần mềm hỗ trợ người học phát hiện ra các lỗi sai trong phát âm của mình sẽ giúp ích rất nhiều trong việc nâng cao kỹ năng nói tiếng Anh cho người học, nâng cao hiệu quả học tập (phát âm), góp phần giảm chi phí và thời gian học tập. Điều này đặc biệt có ích trong bối cảnh toàn cầu hoá hiện nay, khi số lượng người Việt học tiếng Anh ngày càng gia tăng nhanh chóng, khi tiếng Anh là một trong những ngôn ngữ quan trọng nhất và là hành trang không thể thiếu đối với những người muốn tiến xa hơn trong sự nghiệp, học tập, nghiên cứu.

Tuy nhiên, việc xây dựng một phần mềm đáng tin cậy với chức năng nêu trên đòi hỏi một mô hình xử lý tiếng nói thích hợp để có thể tự động phát hiện được chính xác một số lỗi phát âm đặc thù của người Việt khi phát âm tiếng Anh. Xây dựng mô hình này là một trong những mục tiêu của đề tài nghiên cứu. Cụ thể, nghiên cứu này sẽ giải quyết các câu hỏi sau đây:

- Các lỗi phát âm tiếng Anh đặc thù của người Việt là gì?

- Cơ chế xử lý tiếng nói thích hợp để có thể nhận dạng tự động các lỗi phát âm này khi người học phát âm các từ (hoặc cụm từ ngắn) trong tiếng Anh.

- Làm thế nào để định hướng người học khắc phục các lỗi trên?

Trong nghiên cứu này, tác giả chấp nhận các giả thuyết sau đây:

- Mỗi nước trên thế giới đều có các lỗi phát âm đặc thù khi giao tiếp bằng ngoại ngữ. (Các lỗi này là do mỗi ngôn ngữ có một tập hợp nhất định các âm vị, và cách phát âm của ngôn ngữ đó tạo thành một số thói quen nhất định ở các cơ quan phát âm như lưỡi, mũi, môi, v.v… Các thói quen phát âm tiếng mẹ đẻ được chuyển tải qua quá trình phát âm tiếng nước ngoài, tạo ra các lỗi đặc trưng của từng quốc gia).

- Việc chỉ ra các lỗi phát âm, giúp người học nhận biết lỗi sai, từ đó họ tự định hướng cách sửa lỗi phát âm sai, dần dần khắc phục được các lỗi này. (Như vậy, nếu luyện tập thường xuyên, người học sẽ nhanh chóng tiến bộ).

- Luyện tập phát âm với một phần mềm sẽ giúp người học chủ động và thoải mái hơn về giờ giấc so với việc luyện tập với một giáo viên.

1.3 Mục tiêu, đối tượng và phạm vi nghiên cứu

1.3.1 Mục tiêu của đề tài

Mục tiêu tổng quát của đề tài là thực nghiệm để khảo sát việc tự động phát hiện các lỗi phát âm tiếng Anh thường gặp của người học trên các kho dữ liệu Buckeye và TIMIT.

Mục tiêu cụ thể của đề tài gồm:

1) Tìm hiểu các kiến thức nền tảng về ngữ âm học, âm vị học, và các kỹ thuật xử lý tiếng nói.
2) Tìm hiểu mô hình xử lý âm thanh phù hợp để có thể phát hiện được các lỗi trong phạm vi nghiên cứu.
3) Tiến hành thử nghiệm mô hình xử lý trên các tập dữ liệu lớn đáng tin cậy.

1.3.2 Đối tượng và phạm vi nghiên cứu

Nghiên cứu sẽ được tiến hành trên một tập xác định các lỗi phát âm tiếng Anh của người học, xét trên trường hợp cụ thể là người Việt và người Tây Ban Nha học tiếng Anh. Cụ thể là lỗi phát âm khi phát âm các âm vị:[ae], [p], [aa], [sh], [iy] trong tiếng Anh.

CHƯƠNG 2: TỔNG QUAN

2.1 Các nghiên cứu liên quan

Các nghiên cứu về phát hiện lỗi phát âm và đánh giá phát âm bắt đầu từ những năm 1990 và phát triển dữ dội vào cuối thập kỷ 90 đến đầu năm 2000. Có thể kể ra vài nghiên cứu tiêu biểu trong thời kỳ này như các công trình từ (Cucchiarini, De Wet, et al. 1998),(Cucchiarini, Strik, et al. 1998a), (Cucchiarini, Strik, et al. 1998b), (Eskenazi 1999),(Franco, Abrash, et al. 2000), (Kim et al. 1997), (Neumeyer et al. 2000), (Franco, Neumeyer, et al. 2000). Khoảng đầu thế kỷ 20, các phần mềm thương mại CAPT ra đời ngày càng nhiều đã cho thấy nhiều vấn đề khó khăn, kéo theo các hoạt động nghiên cứu cũng dần hạ nhiệt. Tuy nhiên, cùng với sự phát triển mạnh mẽ của khoa học máy tính, các thiết bị di động, và sự cải tiến đáng kể trong lĩnh vực nhận dạng giọng nói, lĩnh vực này lại tiếp tục nhận được sự chú ý của các nhà nghiên cứu, đầu tiên là sự ra đời của tổ chức ISCA với tên gọi là SlaTE (Speech & Language Technology for Education) vào năm 2007. Các nghiên cứu có thể kể đến như (Eskenazi 2009), (Delmonte 2011), (Levis 2007), trong đó cung cấp cái nhìn rất rõ về hướng nghiên cứu trong giai đoạn trước 2009. Do việc phát hiện lỗi phát âm là một bài toán khó nên những nghiên cứu trước đây thường chỉ hướng đến một số thành phần như phát hiện lỗi phát âm ở mức độ âm vị (phoneme) hoặc mức độ nhịp điệu (prosodic). Những năm gần đây, các nghiên cứu bắt đầu đề cập đến các thành phần khác có ảnh hưởng đến phát âm. Thông qua các nghiên cứu đã được công bố, có thể tóm tắt sơ lược một số phương pháp đã được sử dụng để nhận dạng lỗi sai trong phát âm theo từng giai đoạn trong các phần sau.

2.1.1 Phát hiện lỗi dựa trên xác suất (likelihood-based scoring)

Các nghiên cứu đầu tiên trong lĩnh vực này vào những năm 90 đã đưa ra một số thuật toán phát hiện lỗi phát âm ở mức độ âm vị dựa trên xác suất (likelihood). Một số nghiên cứu dựa trên phương pháp này có thể kể đến như: (Kim et al. 1997)(three HMM-based scores),(Witt 1999) (GOP score – Goodness of Pronunciation score),

(Kawai and Hirose 1998) (và phiên bản mở rộng của thuật toán này do (Neumeyer et al. 2000) đề xuất cũng cho kết quả tốt).

2.1.2 Phương pháp độc lập ngôn ngữ thứ nhất (L1-independent)

Một trong những điểm quan trọng trong bài toán dò tìm lỗi phát âm là có nên xây dựng một hệ thống "L1 dependent" (phụ thuộc ngôn ngữ mẹ đẻ) hay không. Hệ thống "L1 independent" (độc lập ngôn ngữ mẹ đẻ) mang về những lợi ích về kinh tế trong khi "L1 dependent" sẽ mang lại hiệu quả vận hành cao hơn. Về hướng "L1 independent", có thể kể ra một số nghiên cứu tiêu biểu như: (Cucchiarini et al. 2011) sử dụng một kho dữ liệu gồm tiếng nói của người nước ngoài học tiếng Hà Lan, được gán nhãn bởi chuyên gia để làm thống kê giữa những lỗi phát âm thường gặp với những lỗi phát âm do ngữ cảnh; (Li et al. 2011) kết hợp giữa việc đánh giá dựa trên xác suất và đánh giá độ trôi chảy(fluency scores); (Cincarek et al., 2009) sử dụng phương pháp dựa trên phân loại (classifier-based), kết hợp giữa đánh giá dựa trên xác suất và đánh giá dựa trên độ dài đoạn ngữ âm tương ứng với âm vị đang xét (different duration) để tính xác suất phát âm sai một số âm vị trên các phát âm.

2.1.3 Phương pháp phụ thuộc ngôn ngữ thứ nhất (L1-dependency)

Bên cạnh hướng độc lập ngôn ngữ mẹ đẻ thì cũng có rất nhiều nghiên cứu theo phương pháp phụ thuộc ngôn ngữ mẹ đẻ vì độ chính xác cao hơn mà nó mang lại. (Ito et al. 2007) đưa ra một số luật phát âm sai cho một cặp L1/L2 cho trước và dùng chúng để nhóm các mẫu lỗi (error rules) bằng cách sử dụng cây quyết định (decision tree). Phương pháp này đã đem lại sự cải tiến đáng kể về độ chính xác trong dò tìm lỗi phát âm.

2.1.4 Phát hiện lỗi dựa trên bộ phân loại (classifier-based scoring)

Mặc dù các phương pháp dựa trên xác suất có ưu điểm là độc lập ngôn ngữ mẹ đẻ và dễ tính toán, nhưng các nhà nghiên cứu cho thấy rằng phương pháp này không thể giúp xác định chính xác loại lỗi phát âm (error type). Rất nhiều nghiên cứu được tiến hành để làm rõ luận điểm này. Tuy nhiên, bằng việc sử dụng bộ phân loại cho

từng cặp âm vị cụ thể, ta có thể xác định được loại lỗi phát âm. (van Doremalen et al. 2009) đã xây dựng một tập các bộ phân loại cho các cặp nguyên âm tương phản trong tiếng Hà Lan. Kết quả từ nghiên cứu này cho thấy rằng việc dùng MFCC cùng với các đặc trưng ngữ âm (phonetic features) để huấn luyện các bộ phân loại sẽ cho kết quả phân loại tốt nhất. Tương tự, (Truong et al. 2004) đã phát triển một bộ phân loại độc lập với ngôn ngữ mẹ đẻ sử dụng một số các đặc trưng âm-ngữ âm (acoustic-phonetic features) đặc thù cho từng loại lỗi phát âm. Bộ phân loại này đã cho kết quả vượt trội so với những nghiên cứu trước đó. Tuy nhiên nhược điểm của phương pháp này là các lỗi thường gặp đặc trưng cho từng L2 phải được biết trước và đòi hỏi các bộ phân loại riêng biệt cho từng loại lỗi phát âm. Những nghiên cứu gần đây theo hướng này có thể kể đến như (Strik et al. 2009), trong đó nhóm tác giả so sánh độ chính xác (khi cho điểm tự động) của 4 bộ phân loại khác nhau cho một tập các cặp âm vị thường bị lẫn lộn khi người nước ngoài phát âm tiếng Hà Lan. Nghiên cứu này cho thấy phương pháp đánh giá dựa trên bộ phân loại có kết quả vượt trội so với đánh giá dựa trên xác suất.

2.1.5 Mô hình tiếng nói do người nước ngoài phát âm (non-native acoustic modeling)

Khi hệ thống CAPT cho phép sinh viên phát âm tự do, ta cần phải có mô hình ngữ âm không phải bản xứ (non-native acoustic modeling). (Ye and Young 2005) cho thấy việc sử dụng thuật toán tương thích chuẩn (standard adaptation algorithm) cho phép tăng độ chính xác trong phát hiện lỗi. Tương tự, (Saz et al. 2009) cũng cho thấy việc đi từ nhận dạng không phụ thuộc người nói (speaker independent) tới phụ thuộc người nói (speaker dependent) hầu như giảm được một nửa tỉ lệ lỗi nhận dạng âm vị.

2.1.6 Phát hiện lỗi phát âm độc lập với văn bản(text independence)

Tính đến hiện tại, có rất ít nghiên cứu đánh giá chất lượng phát âm của các phát âm đàm thoại tự do (unconstrained spontaneous speech). Tuy nhiên, đối với các hoạt động học phát âm nâng cao, việc để sinh viên nói một đoạn văn bản một cách tự

nhiên so với đọc đoạn văn bản là rất cần thiết. Để làm được điều này, các nhà nghiên cứu đề xuất phương pháp dùng tuần tự hai nhiệm vụ nhận dạng khác nhau. Hai công trình tiêu biểu cho phương pháp này là (Moustroufas and Digalakis 2007) và (Chen et al. 2009). Trước tiên, giọng nói ngoại ngữ(của người không phải là người bản xứ) (non-native) sẽ được nhận dạng mà không cần quan tâm tới bất kì lỗi phát âm nào. Việc này được thực hiện với các mô hình ngữ âm(acoustic model) tương thích với các đặc điểm cụ thể của người nói. Tiếp theo đoạn văn bản nhận dạng được sử dụng để tiến hành nhận dạng trong chế độ đặt các phân cách thời gian (forced-alignment) trong bản phiên âm (transcription) của tín hiệu tiếng nói và để tính toán mức độ phát âm đúng dựa trên một trong các thuật toán được đề xuất cho nhiệm vụ này.

2.1.7 Phát hiện và phản hồi lỗi về nhịp điệu phát âm(prosodic pronunciation error)

Gần đây có rất nhiều nghiên cứu dựa trên phương pháp này. (Levow 2009) dùng một bộ phân loại dựa trên SVM (SVM based classifier) cho việc nhận dạng giọng nói (pitch accent). (Hönig et al. 2009) sử dụng một tập lớn các đặc tính dựa trên duration (thời lượng phát âm), energy (năng lượng dùng phát ra âm thanh), pitch (giọng) và pauses (khoảng dừng) để dò tìm các accent(trọng âm). Gần đây hơn (Hönig et al. 2012) sử dụng phương pháp phân biệt, trong đó tác giả dùng một tập lớn các đặc tính nhịp(nhịp điệu) đặc biệt như là đặc tính nhịp điệu tổng quát(general prosodic) để tạo ra một độ đo phù hợp thích hợp cho phát âm có nhịp điệu(prosodic pronunciation).

2.1.8 Thiết kế hệ thống CAPT có tính tương tác (Interactive CAPT system design)

Tạo các bài luyện phát âm đòi hỏi nhiều thời gian. Ý tưởng tự động hoá quá trình tạo các bài luyện được đề xuất trong (Liu et al. 2009) và (Saz and Eskenazi 2011).(Saz and Eskenazi 2011) tự động đưa ra các bài luyện gồm một câu gốc và một câu được tự động phát sinh có một số âm vị dễ nhầm lẫn (phát âm sai) giữa hai

câu (ta gọi là minimal pair difference). Việc này giúp sinh viên tập trung vào các lỗi phát âm nghiêm trọng có thể gây ra mức hiểu lầm cao hơn so với các lỗi khác. Gần đây nhất, (Rossetti et al. 2011) xây dựng một hệ thống dạy học kết hợp các lý thuyết về học ngoại ngữ và các kỹ thuật dạy phát âm. Đây là một trong nhiều ví dụ về học phát âm trong các hệ thống tương tác đa phương tiện (multimedia dialog).

2.2 Các vấn đề còn tồn tại

Đã có nhiều phương pháp tự động đánh giá phát âm bằng cách sử dụng độ tin cậy (confidence scores) được tính toán từ hệ thống nhận dạng giọng nói. Độ tin cậy đo mức độ giống nhau giữa phát âm của người nói với âm được nhận dạng. Kết quả sai sẽ dẫn tới độ tin cậy thấp, điều này cung cấp thông tin về lỗi phát âm của người nói.

Tuy nhiên, độ chính xác của việc đánh giá dựa trên độ tin cậy không phải lúc nào cũng cao. Hơn nữa, việc đo lường được tính toán theo cùng một cách cho tất cả các âm vị (phoneme) nên khó để đo lường cụ thể cho các âm vị đặc biệt mà người học thường phát âm sai. Lúc bắt đầu học, người học có khuynh hướng phát âm sai các âm vị không tồn tại trong ngôn ngữ mẹ đẻ của họ (L1), và họ thậm chí vẫn phát âm sai một vài trong số các âm vị ấy đến tận vài năm học sau đó. Các phương pháp luyện phát âm cần phải phát hiện được lỗi và định hướng tập luyện các âm vị này theo cách đặc biệt riêng.

Phương pháp phân loại đạt hiệu quả cao hơn trong trường hợp đánh giá các phát âm sai các âm vị đặc biệt. (Felps et al. 2009) đã xây dựng mô hình bộ phân loại cho âm tắc – vòm mềm – vô âm (voiceless velar fricative) /x/, thường bị phát âm sai thành âm bật – vòm mềm – vô âm (voiceless velar stop) /k/ cho người Hà Lan học tiếng Anh. Tác giả huấn luyện một cây quyết định bằng cách sử dụng đặc trưng âm – ngữ âm chuyên cho việc phân biệt phụ âm bật (stop) và phụ âm tắc (fricative), và đã đạt độ chính xác trong khoảng từ 75% → 91%. (Eskenazi 2009) xây dựng hai bộ phân loại sử dụng đặc trưng âm – ngữ âm trong (Felps et al. 2009) (bộ phân loại A.P) và các hệ số ceptral (cepstral coefficients) (bộ phân loại MFCC). Cả hai bộ phân loại này đều cho kết quả với độ chính xác cao hơn so với phương pháp dựa trên độ tin

cậy, nhưng bộ phân loại AP thậm chí còn cho kết quả tốt hơn cả bộ phân loại MFCC khi có sự sai lệch kho dữ liệu dùng để huấn luyện và kho dữ liệu dùng để đánh giá. Tuy nhiên bộ phân loại MFCC lại dễ cài đặt hơn bộ phân loại AP vì các đặc trưng MFCC đã có sẵn trong hệ thống nhận dạng giọng nói.

2.3 Phương hướng giải quyết của nghiên cứu này

Luận văn này sử dụng bộ phân loại SVM(SVM classifier based) trong hệ thống tự động phát hiện lỗi phát âm sai. Theo lý thuyết ESL, chọn ra các âm vị mà người học thường phát âm sai, sau đó cho các bộ phân loại SVM học trên tất cả các âm vị này. Phương pháp này không giới hạn cho các nguyên âm và phụ âm đặc biệt.

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

Trong chương này, tác giả trình bày các kiến thức cơ sở liên quan đến ngữ âm học và âm vị học (phần 3.1), cũng như các kiến thức cơ sở về xử lý tiếng nói (phần 3.3), nhằm cung cấp nền tảng kiến thức cần thiết để có thể trình bày và thảo luận về các vấn đề liên quan đến thí nghiệm được trình bày trong chương 4. Cũng trong chương này, sau khi trình bày các kiến thức cơ sở về ngữ âm học, âm vị học, hệ thống âm vị tiếng Anh, hệ thống âm vị tiếng Việt, tác giả chọn ra một số lỗi sai được giả định là thường gặp của người Việt phát âm tiếng Anh (phần 3.2). Giả định này dựa trên giả thuyết rằng những âm vị tiếng Anh không có mặt trong hệ thống âm vị tiếng Việt sẽ dễ bị phát âm sai do ảnh hưởng bởi thói quen phát âm tiếng mẹ đẻ.

3.1 Cơ bản về ngữ âm học và âm vị học

Phần này trình bày một số kiến thức cơ bản về ngữ âm học và âm vị học làm cơ sở lý luận cho luận văn. Trong phần này, tác giả dùng xen lẫn hai hệ thống ký hiệu âm vị IPA và ARPAbet (xem chi tiết hệ thống ký hiệu ARPAbet trong phụ lục). Khi dùng hệ thống ký hiệu IPA, tác giả dùng ký hiệu "/./" để chỉ đó là ký hiệu theo hệ thống IPA. Khi dùng hệ thống ARPAbet, tác giả dùng "[.]".

3.1.1 Ngữ âm học và âm vị học

Cách phát âm (pronunciation) của mo□t ngôn ngữ luôn đu□ợc nghiên cứu du□ới 2 khía cạnh ngữ âm học (phonetic) và âm vị học (phonology). Mạ□c dù 2 ngành này đều nghiên cứu âm thanh, nhu□ng giữa chúng có mo□t số điểm khác bie□t co□ bản nhu□ sau:

- Âm vị học là ngành khoa học nghiên cứu về sự khác nhau trong cách phát âm của cùng một âm vị hoặc của những âm vị khác nhau, ngữ điệu của từ và câu, qua các khái niệm âm vị, hình thang nguyên âm, tha âm vị (allophone), ngữ điệu (intonation), nhấn giọng (stress), đọc lướt (weak form).

- Ngữ âm học có tính phổ quát (universal) hơn. Nó nghiên cứu các vấn đề sau: các thuộc tính âm thanh có tính chất loài, các âm tố (speech sound hoặc là phone, xem chi tiết trong phần 3.1.2). Ngữ âm học không những nghiên cứu quá trình tạo ra âm thanh (speech production), mà còn nghiên cứu quá trình nhận thức âm thanh (sound perception) cũng như quá trình truyền âm thanh (transmission of sounds).

3.1.2 Âm vị (phoneme) và âm tố (phone hay speech sound)

Âm vị là một đơn vị cơ bản nhỏ nhất của ngôn ngữ (ở khía cạnh âm vị học), có thể gây ra sự thay đổi về ý nghĩa. Nghĩa là chỉ cần thay đổi một âm vị trong một từ ta có thể tạo ra một từ có ý nghĩa khác. Ví dụ: xét từ "kiss" (phát âm là /kɪ s/) và "kill" (phát âm là /kɪ l/). Hai từ có ý nghĩa khác nhau này hình thành bằng cách thay âm vị /s/ bằng /l/.

Âm tố (phone) là âm thanh được phát ra với mục đích thể hiện âm vị. Cần lưu ý sự khác biệt giữa âm vị (phoneme) và âm tố (phone hay speech sound): âm vị là một đơn vị trừu tượng còn âm tố là một thể hiện cụ thể của âm vị. Âm vị được thể hiện ra bằng các âm tố và âm tố là sự thể hiện của âm vị. Những âm tố cùng thể hiện một âm vị được gọi là các biến thể của âm vị hay còn gọi là tha âm vị (allophone – xem chi tiết trong phần 3.1.8).

3.1.3 Phụ âm (consonant) và nguyên âm (vowel)

Trong quá trình phát âm, luồng hơi từ phổi sẽ được thoát ra ngoài. Trên đường thoát ra ngoài, luồng hơi có thể bị nghẽn nhiều hoặc ít, tạo ra phụ âm hoặc có sự điều chỉnh nhỏ để tạo ra nguyên âm.

Sự phân biệt giữa nguyên âm và phụ âm được dựa trên 3 đặc điểm cơ bản sau đây:

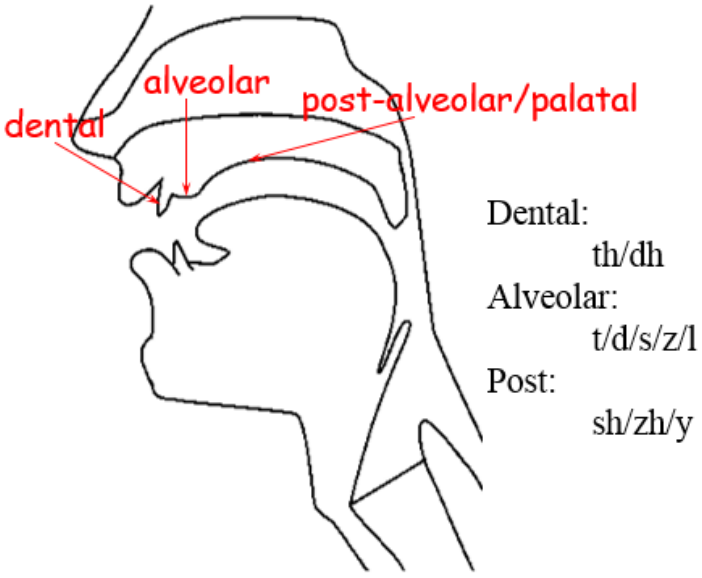
- Đặc điểm sinh lý (physiological): Khi phát âm, luồng hơi bị chặn lại (trong trường hợp phụ âm) và thoát ra tự do (trong trường hợp nguyên âm).
- Đặc điểm ngữ âm (acoustic): Nguyên âm thường được nghe rõ hơn, nổi bật hơn, nhiều năng lượng hơn phụ âm.

- Đặc điểm âm vị học (phonological): Nguyên âm tạo ra được âm tiết, phụ âm không thể tạo ra được âm tiết. Một âm tiết bắt buộc phải có một nguyên âm.

Các phụ âm được phân biệt với nhau dựa chủ yếu vào vị trí phát âm (xem phần 3.1.4) và cách thức phát âm (xem phần 3.1.5). Nhưng để phân biệt một cách đầy đủ các phụ âm, người ta dùng một bộ 3 tham số Voicing/Unvoicing (xem phần 3.1.7), vị trí phát âm (place of articulation), và cách thức phát âm (manner of articulation). Bộ 3 tham số này thường được viết tắt là VPM (Voice, Place, Manner).

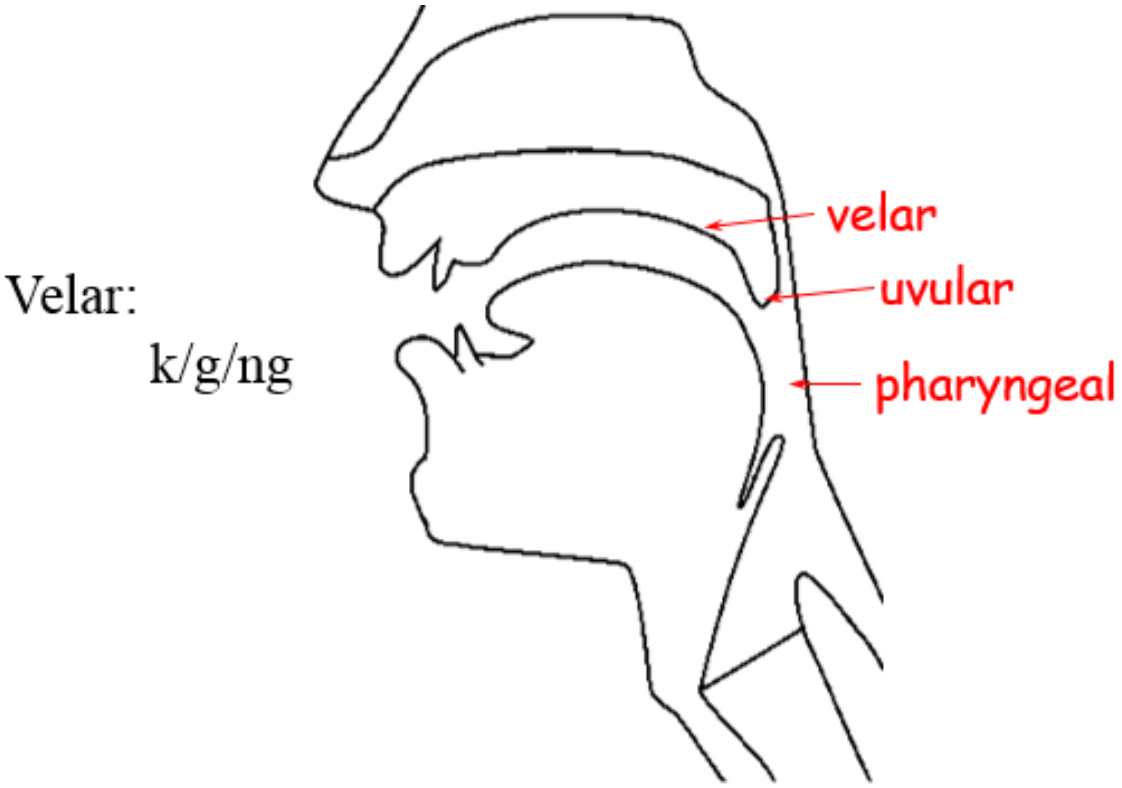
3.1.4 Ví trí phát âm (place of articulation)

Các phụ âm có thể được phân loại dựa vào vị trí nơi luồng khí đi trong hệ thống phát âm (articulation) bị hạn chế hay thu hẹp (constricted) nhất. Một cách tổng quát nhất, có thể chia vị trí phát âm thành 3 loại: vị trí môi (labial), vị trí đầu lưỡi (coronal), vị trí cuối lưỡi (dorsal). Âm đầu lưỡi là các phụ âm được hình thành bằng cách tạo khe hẹp ở vị trí đầu lưỡi. Đối với âm đầu lưỡi, có thể được chia nhỏ thành các loại: âm răng (dental), âm chân răng (alveolar), âm chân răng sau (post-aveolar). Hình sau đây được trích từ bài giảng về ngữ âm học của giáo sư Daniel Jurafsky tại đại học Stanford. Trong Hình 3.1, tác giả dùng kí hiệu phiên âm ARPAbet (xem phần phụ lục).



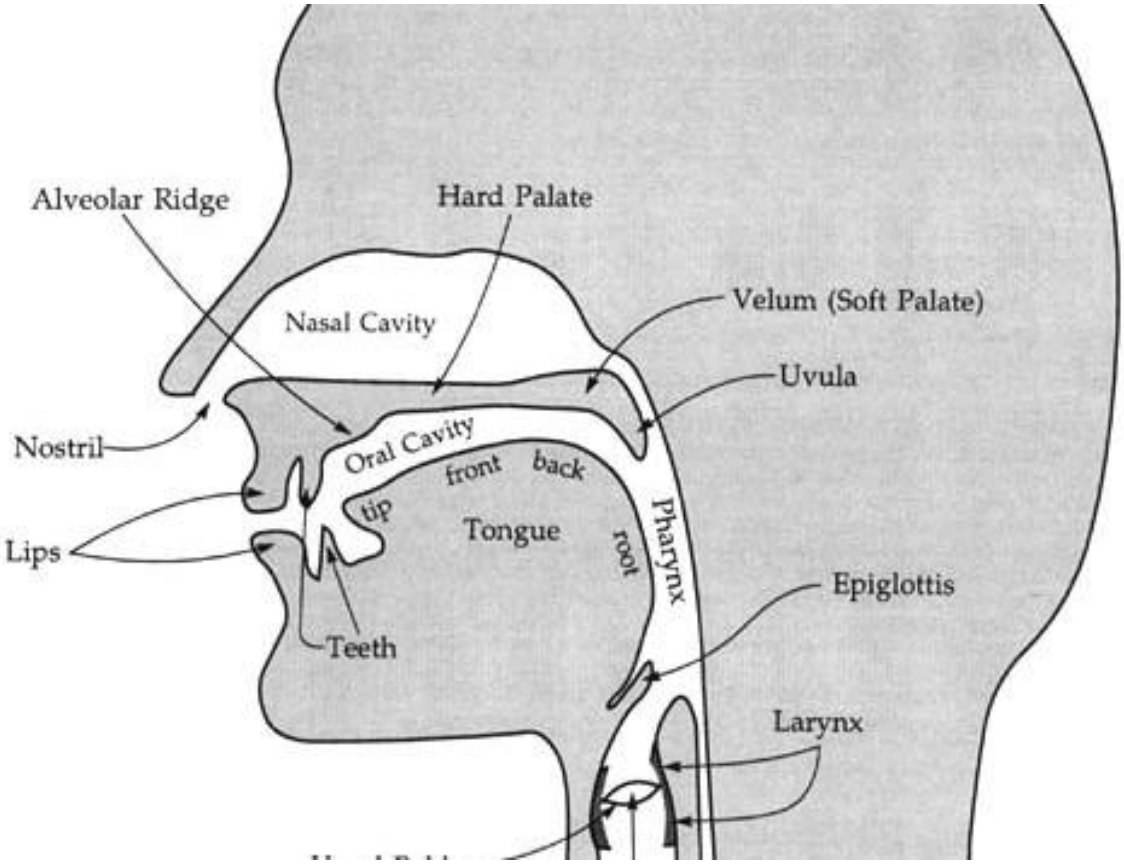
Hình 3.1 – Vị trí phát âm của âm môi

Âm cuối lưỡi là các phụ âm được hình thành dựa trên khe hẹp ở cuối lưỡi. Cũng theo bài giảng nêu trên, âm cuối lưỡi được chia thành 3 loại: âm vòm mềm (velar), âm lưỡi nhỏ (uvular), âm yết hầu (pharyngeal). Vị trí phát âm của các âm này được mô tả bằng Hình 3.2:

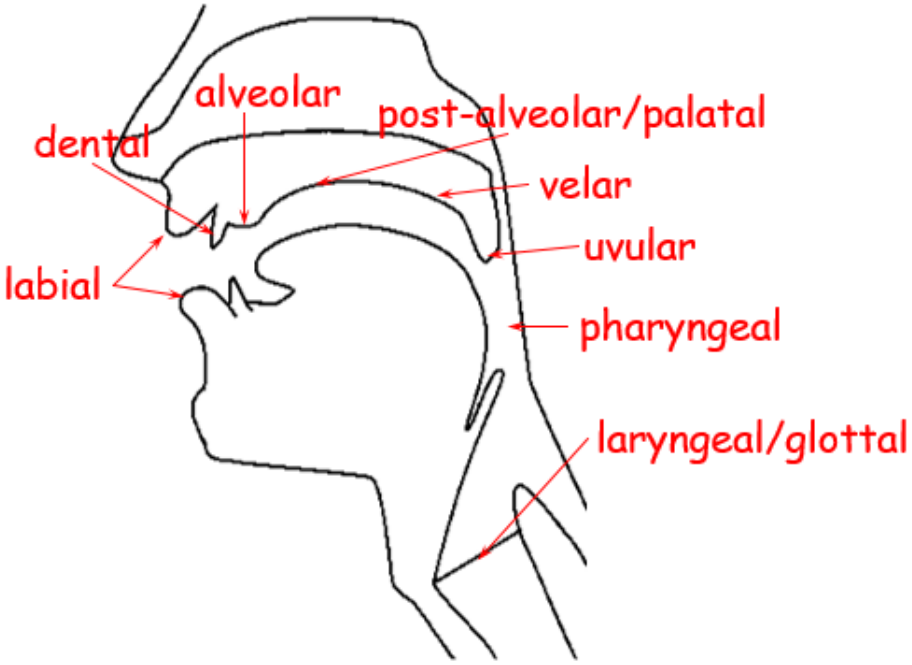


Hình 3.2 – Vị trí phát âm của cuối lưỡi

Âm môi được hình thành bởi khe hẹp tạo ra ở vị trí môi. Hình 3.3 (*Source: Department of Linguistics, University of Pennsylvania*) và Hình 3.4 mô tả các vị trí khác nhau trong hệ thống phát âm và các loại phụ âm tương ứng tại các vị trí đó.



Hình 3.3 – Các vị trí khác nhau trong hệ thống phát âm

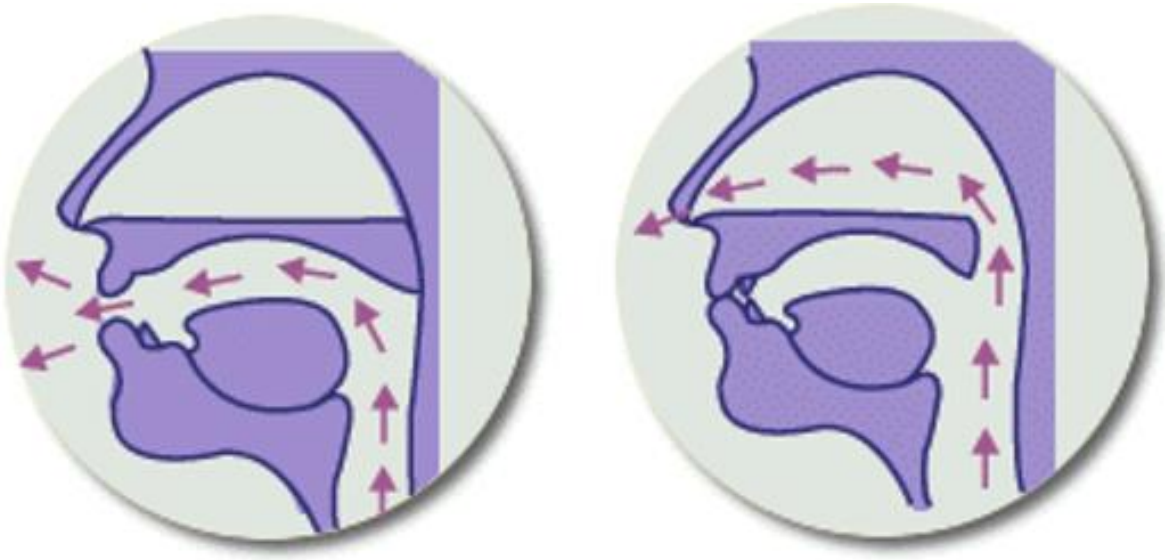


Hình 3.4 – Các loại phụ âm tương ứng với các vị trí phát âm

3.1.5 Cách thức phát âm (manner of articulation)

Ngoài việc phân loại các phụ âm dựa trên vị trí phát âm, người ta còn phân loại các phụ âm dựa vào cách thức phát âm (manner of articulation). Theo đó các phụ âm được chia thành các loại sau:

- Âm mũi (nasal sound): được hình thành bằng cách điều khiển luồng hơi đi ra ngoài bằng đường mũi chứ không phải đường miệng.
- Âm miệng (oral sound): được hình thành bằng cách cho luồng hơi thoát hoàn toàn qua đường miệng. Hình 3.5 minh họa sự khác nhau giữa âm mũi và âm miệng.
- Âm tiệm cận (approximant sound): được hình thành bằng cách để các bộ phận phát âm gần nhau nhưng không thực sự đủ gần để tạo ra khe hẹp (constricted). Ví dụ, cho âm vị này là /y/ và /r/ (kí hiệu theo ARPAbet).
- Âm tiệm cận cạnh (lateral approximant): được hình thành bằng cách điều khiển luồng hơi tập trung vào giữa lưỡi và thoát qua hai bên lưỡi và đi ra ngoài (không thoát qua đầu lưỡi).
- Âm sát (fricative): được hình thành bằng cách tạo khe hẹp trong bộ phận phát âm đủ nhỏ để tạo thành âm thanh tương tự như âm /s/ của tiếng Việt.



Hình 3.5 – Sự khác nhau giữa âm mũi (phải) và âm miệng (trái)

Error! Reference source not found. **(*Soure The International Phonetic Alphabel 2005*) tóm tắt các âm vị được phân loại dựa trên vị trí phát âm và cách thức**

phát âm.

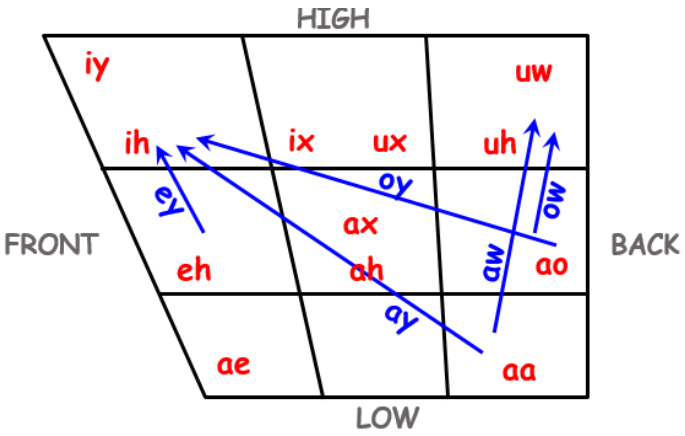
VỊ TRÍ PHÁT ÂM (PLACE OF ARTICULATION)

CÁCH THỨC PHÁT ÂM (MANNER OF ARTICULATION)

	Môi (labial)		Đầu lưỡi (coronal)				Mặt lưỡi (dorsal)				Gốc lưỡi		Họng
	Môi (bilabial)	Môi răng (labio-dental)	Răng (dental)	Chân răng (Alveolar)	Chân răng sau (Palato-alveolar)	Quặt lưỡi (retroflex)	Lợi vòm miệng (alveolo-palatal)	Vòm miệng (palatal)	Vòm mềm (velar)	Lưỡi nhỏ (Uvular)	Yết hầu (pharyngeal)	Thanh quản (epi-glottal)	Thanh hầu (glottal)
Mũi (nasal)	m	ɱ	n			ɳ	ɲ		ŋ	N			
Bật (plosive)	p b		t d			ʈ ɖ	c ɟ		k g	q G		ʡ	ʔ
Sát/xát (fricative)	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ɕ ʑ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	ʜ ʢ	h ɦ
Tiếp cận (approximant)		ʋ	ɹ			ɻ	j		ɰ				
Vỗ (tap, flap)		ⱱ	ɾ			ɽ							
Rung (trill)	ʙ		r							ʀ			
Sát cạnh (lateral fricative)			ɬ ɮ										
Tiếp cận cạnh (lateral approximant)			l			ɭ	ʎ		ʟ				

3.1.6 Hình thang nguyên âm

Các nguyên âm có thể được phân loại dựa vào độ mở của miệng và vị trí trong bộ phận phát âm nơi luồng hơi bị chặn lại một phần khi phát âm. Để minh họa sự khác biệt giữa các loại nguyên âm, người ta dùng hình ảnh được gọi là hình thang nguyên âm có dạng là một hình thang ngược (tức cạnh dưới nhỏ hơn cạnh trên) như Hình 3.6. Cạnh dưới của hình thang nguyên âm đại diện cho hàm dưới, cạnh trên đại diện cho hàm trên. Cạnh bên trái đại diện cho đầu lưỡi, cạnh bên phải đại diện cho cuốn lưỡi.

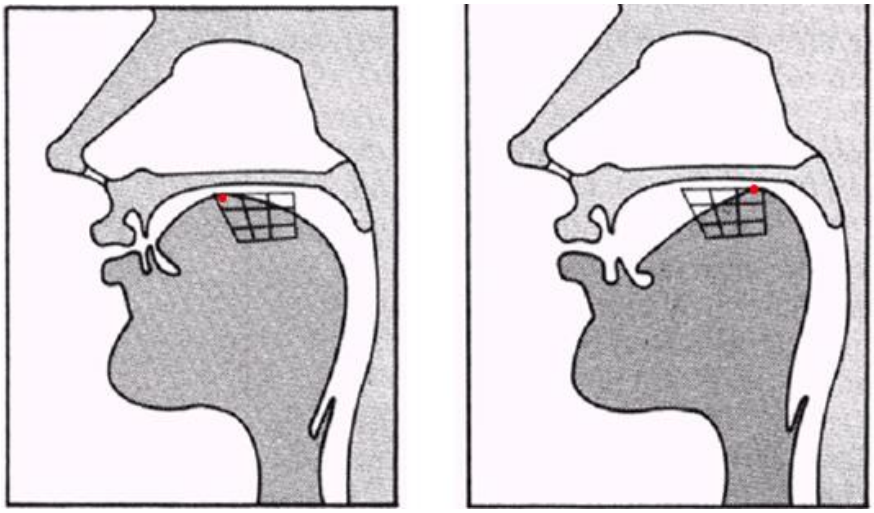


Hình 3.6 – Hình thang nguyên âm

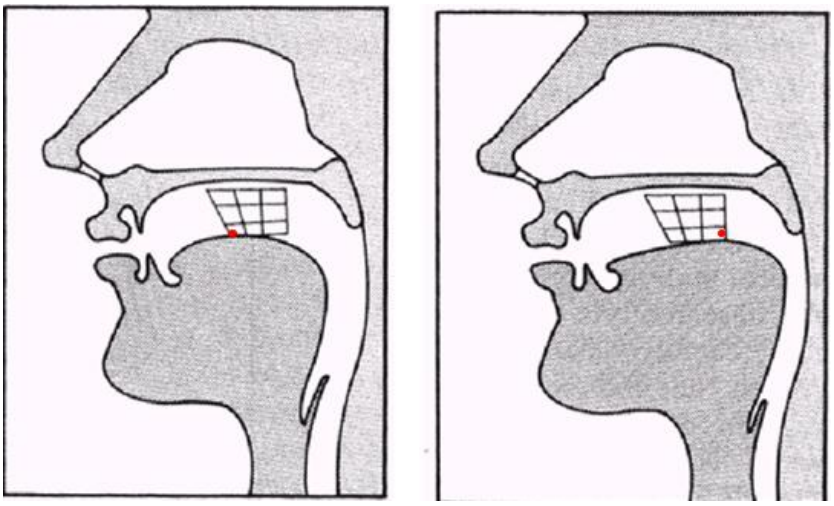
Trong Hình 3.6 các âm vị được kí hiệu dùng ARPAbet. Nhìn chung, người ta chia ra một số loại âm vị sau:

- Âm trước (front): được hình thành bằng cách tạo khe hẹp ở phía ngoài cùng của khoang miệng.
- Âm sau (back): được hình thành bằng cách thụt lưỡi sâu vào trong họng
- Âm cao (high): được hình thành bằng cách đóng hẹp hàm trên và hàm dưới gần nhau.
- Âm thấp (low): được hình thành bằng cách mở rộng miệng để hàm trên và hàm dưới xa nhau.

Hình 3.7minh họa sự khác biệt giữa hai âm vị tiếng Anh: [iy] (âm trước, như trong từ “eat”) và [uw] (âm sau, như trong từ “school”).



Hình 3.7 – Sự khác nhau giữa 2 âm tiếng Anh [iy] (trái) và âm [uw] (phải).



Hình 3.8 – Sự khác nhau giữa hai âm vị tiếng Anh [ae] (trái) và [aa] (phải)

3.1.7 Âm hữu thanh (voice) và âm vô thanh (voiceless)

Âm hữu thanh (voice) là âm được tạo với sự rung của dây thanh âm (vocal fold/cord), luồng khí từ phổi đi qua dây thanh âm liên tục bị đóng, mở thông qua cơ chế rung của dây thanh âm. Âm vô thanh là âm được tạo ra với sự mở rộng của dây thanh âm (vocal cord/fold) để cho luồng khí đi qua dây thanh âm trong cổ họng một cách tự do.

3.1.8 Tha âm vị (allophone)

Trong âm vị học, một tha âm vị (allophone) là một âm tố trong một tập nhiều âm tố (phone hoặc speech sound) được sử dụng để phát âm một âm vị duy nhất trong một ngôn ngữ cụ thể. Ví dụ, [p^h] (trong từ"pin") và [p] (trong từ "spin") là các tha âm vị của âm vị /p/ trong tiếng Anh. Các tha âm vị cụ thể được chọn trong một tình huống xác định thường có thể đoán được dựa vào ngữ cảnh âm (những tha âm vị như vậy được gọi là các biến thể vị trí (positional variantion) – tức do vị trí của âm vị trong từ thay đổi. Đôi khi tha âm vị xảy ra trong sự biến đổi tự do (free variantion), tức do cách phát âm khác nhau trong các ngữ cảnh khác nhau hoặc ảnh hưởng bởi môi trường, tiếng ồn. Thay thế một âm tố bằng một âm tố khác trong cùng một tập các tha âm vị thường sẽ không làm thay đổi từ được nhận thức bởi người nghe, mặc dù đôi khi kết quả nghe không giống giọng bản xứ hoặc thậm chí là khó hiểu. Người bản ngữ của một ngôn ngữ nhất định thường nhận thức được một âm tố trong ngôn ngữ đólà một âm thanh đặc biệt duy nhất, và sẽ ngạc nhiên khi thấy các biến thể tha âm vị dùng để phát âm các âm vị tương ứng với âm tố đó.

3.1.9 Hệ thống âm vị tiếng Việt

Phần lý thuyết trình bày bên dưới trích từ nguồn *http://ngonngu.net/*.

3.1.9.1 Hệ thống âm đầu

Tiếng Việt có 22 phụ âm đầu, bao gồm

/b, m, f, v, t, t', d, n, z, ʐ, s, ʂ, c, ʈ, ɲ, l, k, χ, ŋ, ɣ, h, ʔ/

Bảng 3.1 – Hệ thống âm đầu tiếng Việt

Phương thức \ Vị trí				Môi	Đầu lưỡi		Mặt lưỡi	Gốc lưỡi	Thanh hầu
					Bợt	Lưỡi			
Tắc	Ồn	Bật hơi			t'				
		Không bật hơi	Vô thanh		t	ʈ	c	k	ʔ
			Hữu thanh	b	d				
	Vang			m	n		ɲ	ŋ	
Xát	Ồn	Vô thanh		f	s	ʂ		χ	h
		Hữu thanh		v	z	ʐ		ɣ	
	Vang				l				

Âm đệm /w/ có chức năng làm trầm hoá âm sắc của âm tiết.

3.1.9.2 Hệ thống âm chính

Tiếng Việt có 13 nguyên âm đơn và 3 nguyên âm đôi làm âm chính:

/i, e, ɛ, ɤ, ɤ˘, a, ɯ, ă, u, o, ɔ, ɔ˘, ɛ˘, ie, ɯɤ, uo/

Bảng 3.2 – Hệ thống nguyên âm tiếng Việt

Âm sắc	Vị trí lưỡi, hình dáng môi / Độ mở của miệng	Trước, không tròn môi	Sau	
			Không tròn môi	Tròn môi
Cố định	Nhỏ	i	ɯ	u
	Lớn vừa	e	ɤ/ɤ̆	o
	Lớn	ɛ/ɛ̆	a/ă	ɔ/ɔ̆
Không cố định		i̯e	ɯ̯ɤ	u̯o

3.1.9.3 Hệ thống âm cuối

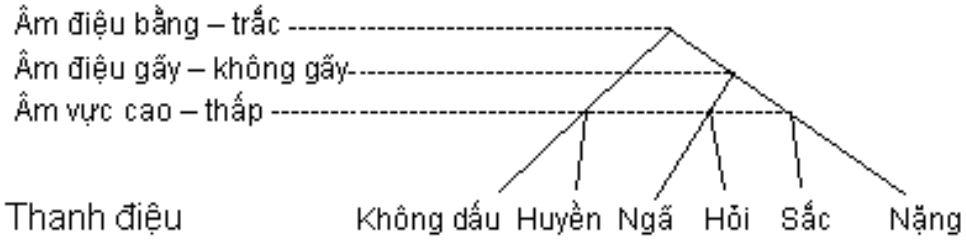
Ngoài âm cuối /rezo/, tiếng Việt còn có 8 âm cuối có nội dung tích cực, trong đó có 6 phụ âm /m, n, ŋ, p, t, k/ và hai bán nguyên âm /-w, -j/.

Bảng 3.3 – Hệ thống âm cuối tiếng Việt

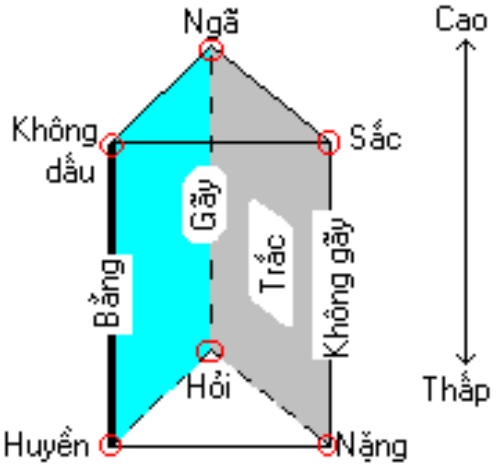
Phương thức / Vị trí		Môi	Lưỡi	
			Đầu lưỡi	Gốc lưỡi
Ồn		p	t	k
Vang	Mũi	m	n	ŋ
	Không mũi	-w	-j	

3.1.9.4 Hệ thống thanh điệu

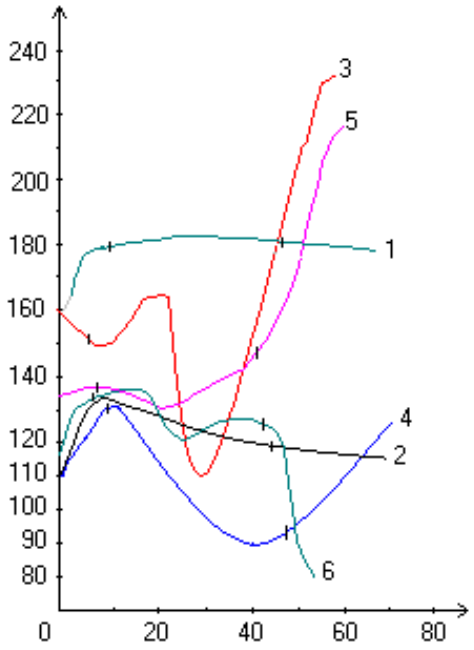
Tiếng Việt có 6 thanh điệu.



Hình 3.9 – Sơ đồ về 3 tiêu chí khu biệt cho sáu âm vị thanh điệu



Hình 3.10 – Lăng trụ thanh điệu



Hình 3.11 – Biểu đồ thanh điệu

Quy luật hình thành thanh điệu tiếng Việt

Trong quá trình lịch sử phát triển nhóm ngôn ngữ Việt Mường đã có một chuyển đổi quan trọng mang tính quy luật: ban đầu chúng là những ngôn ngữ/ phương ngữ không thanh điệu, về sau hệ thống thanh điệu xuất hiện và có diện mạo như ngày nay. Chuyển đổi mang tính quy luật này thường được các nhà nghiên cứu gọi là quy luật hình thành thanh điệu và do A.G. Haudricourt giải thích từ năm 1954. Sơ đồ dưới đây cho chúng ta biết rằng sự xuất hiện các thanh xảy ra là do các biến đổi của âm cuối (rụng đi) và phụ âm đầu (lẫn lộn vô thanh với hữu thanh).

Bản chất của quá trình này là vấn đề đường nét các thanh điệu có liên quan đến cách kết thúc âm tiết. Bản chất của quá trình này cũng là sự xuất hiện âm vực của từ và sau đó là độ cao của thanh điệu nhằm giải quyết mối tương ứng hữu thanh và vô thanh lẫn lộn.

3.1.10 Hệ thống âm vị tiếng Anh

Theo (Jurafsky and Martin 2014), tiếng Anh (Mỹ) có 43 âm vị, bao gồm 26 phụ âm được liệt kê trong Bảng 3.5 và 17 nguyên âm được liệt kê trong Hình 3.6. Lưu ý là ở đây, các âm vị được ký hiệu theo hệ thống ARPAbet. Trong số 17 nguyên âm của tiếng Anh, có 12 nguyên âm đơn và 5 nguyên âm đôi ([ey], [oy], [ow], [aw], [ay]).

Bảng 3.4 – Các phụ âm trong tiếng Anh (được phân loại dựa vào VPM)

MANNER OF ARTICULATION \ PLACE OF ARTICULATION	bilabial		labio-dental		inter-dental		alveolar		palatal		velar		glottal	
stop	p	b					t	d			k	g	q	
fric.			f	v	th	dh	s	z	sh	zh			h	
affric.									ch	jh				
nasal		m						n				ng		
approx		w						l/r		y				
flap							dx							

VOICING: voiceless | voiced

3.2 Xác định một số lỗi sai thường gặp của người Việt học tiếng Anh

Trong phần này, tác giả sẽ xác định một số lỗi phát âm thường gặp để khảo sát trong phần thí nghiệm ở Chương 4. Tác giả chọn một số lỗi phát âm được đề cập trong (Witt and Young 2000) và một số lỗi phát âm thường gặp của người Việt. Các lỗi phát âm của người Việt được chọn lựa dựa trên hai giả thuyết/quan sát sau đây:

- Tiếng Việt có đặc điểm là viết sao đọc vậy, nhưng tiếng Anh thì cách đọc khác với cách viết (ví dụ trong "delete", ký tự "e" được phát âm thành [ih]. Do thói quen phát âm của người Việt, nhiều người phát âm các âm vị [ih] thành [eh].
- Một số âm vị trong tiếng Anh không có trong tiếng Việt, chẳng hạn [aa], khiến người Việt phát âm [aa] thành cách phát âm ký tự A ([ae]) của người Việt (vì có cách phát âm khá gần [aa]). Xem một số ví dụ trong Bảng 3.6

Bảng 3.5 – Các âm vị tiếng Anh không có trong tiếng Việt

IPA	ARPAbet	Ví dụ	Âm vị tiếng Việt gần giống
/ð/	[dh]	Father /fɑðɚ/	đ, th
/ɑ :/	[aa]	Father /fɑðɚ/	A
/dʒ /	[dz]	Jump /dʒ ʌ mp/	d, ch, gi

Trong luận văn này, tác giả giới hạn phạm vi nghiên cứu chỉ dừng ở mức độ khảo sát phương pháp tự động phát hiện các lỗi trong Bảng 3.7

Bảng 3.6 – Một số lỗi phát âm sẽ khảo sát trong luận văn.

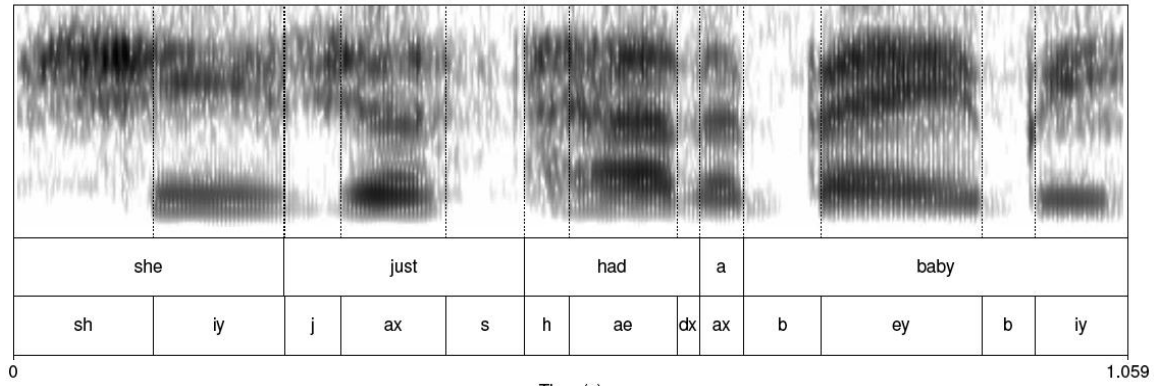
Âm vị tiếng Anh (L2) đúng	Âm vị thay thế do phát âm sai	Từ gốc	Phát âm gốc	Phát âm sai	L1
[ae]	[eh]	bad	[baed]	[behd]	Tiếng Tây Ban Nha
[p]	[b]	pause	[pows]	[bows]	Tiếng Việt
[aa]	[ae]	father	[faadhuh]	[faedhuh]	Tiếng Việt
[sh]	[s]	show	[show1]	[sow1]	Tiếng Việt
[iy]	[ih]	sheep	[shiyp]	[shihp]	Tiếng Việt

3.3 Cơ bản về xử lý tiếng nói

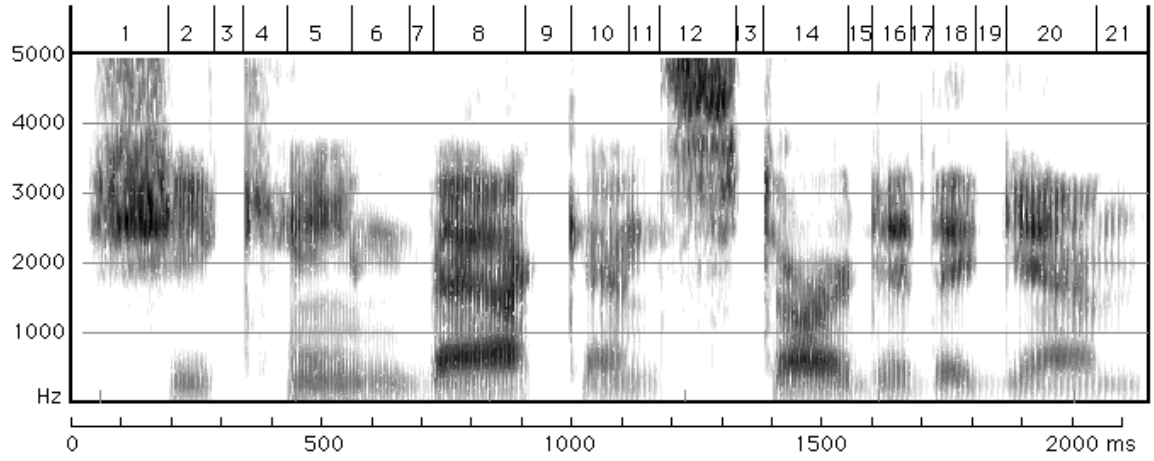
Phần này, tác giả sẽ trình bày một số khái niệm cơ bản trong xử lý tiếng nói.

3.3.1 Spectrogram

Spectrogram là một sự thể hiện trực quan bằng đồ thị của các tần số quang phổ của âm thanh hoặc một loại tín hiệu nào đó mà có sự biến đổi theo thời gian hoặc là theo một biến số khác. Các ví dụ về spectrogram được trình bày trong các hình từ Hình 3.12, Hình 3.13(nguồn http://www.phonetics.ucla.edu) và Hình 3.14.



Hình 3.12 – Spectrogram gồm 2 chiều: tần số (spectrum) và thời gian

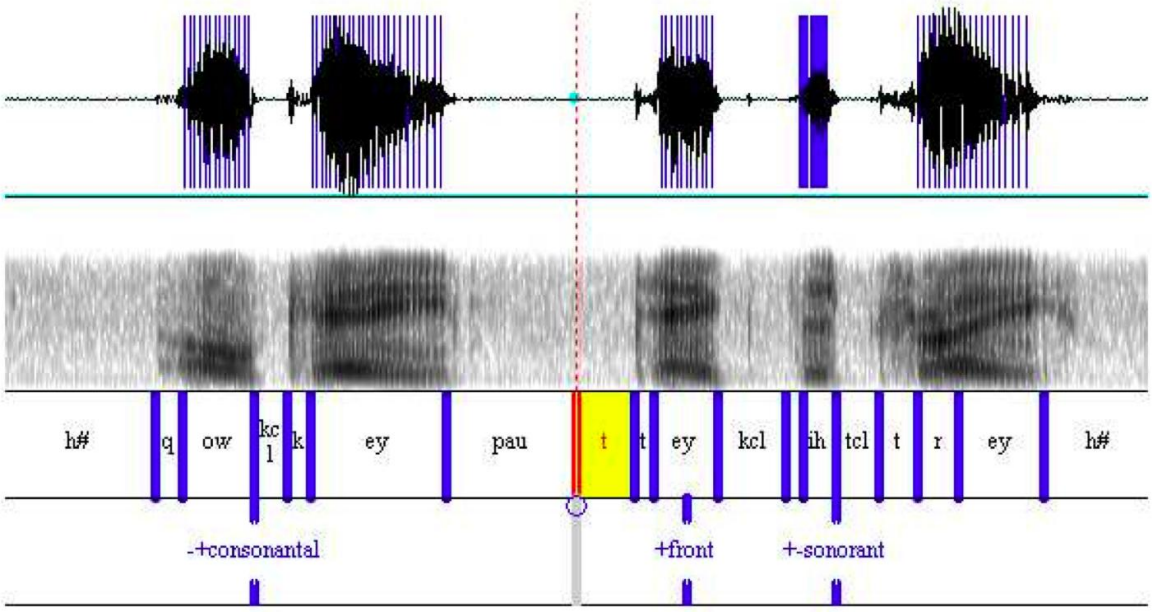


Hình 3.13 – Spectrogram của câu nói "She came back and started again"

Nhìn vào Hình 3.13 ta rút ra các nhận xét sau:

- Trong đoạn 1: Chứa nhiều năng lượng ở mức tần số cao
- Đoạn 3: Tương ứng với khoảng thời gian đóng miệng, chuẩn bị phát âm /k/
- Đoạn 4: Âm thanh nhiễu (burst) gây ra trong đoạn bắt đầu phát âm /k/

- Đoạn 5: Nguyên âm [ey]; phần formant 1100 Hz xuất hiện do chuẩn bị phát âm âm mũi.
- Đoạn 6: Âm mũi môi (bilabial nasal), tức âm /m/
- Đoạn 7: Giai đoan đóng miệng chuẩn bị phát âm /b/ (b closure).
- Đoạn 8: Tương ứng âm [ae]. Lưu ý phần chuyển tiếp sau âm môi dừng (bilabial stop – tức âm /b/).
- Đoạn 9: Hai formant trong /k/.



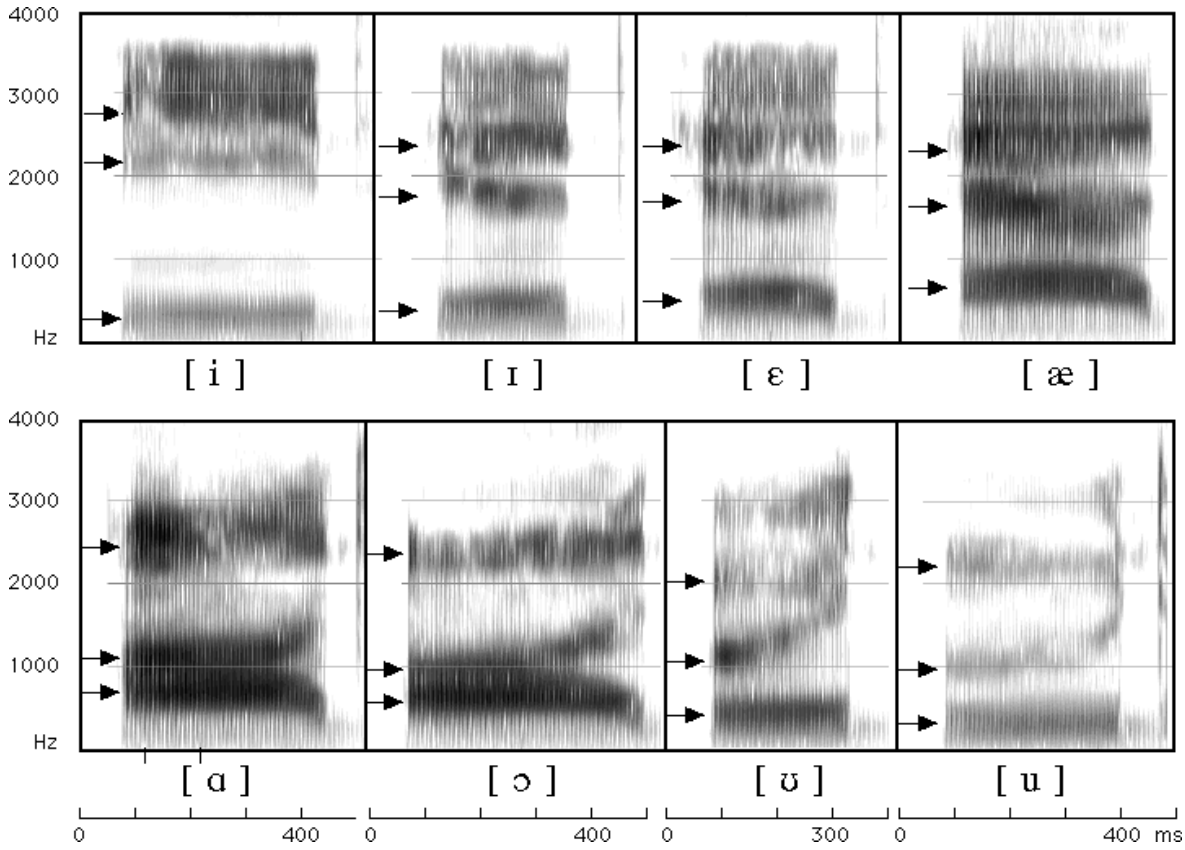
Hình 3.14 – Sóng âm, spectrogram, và phiên âm ở mức âm vị và mức landmark.

Câu nói trong Hình 3.14 là "Okay, take the tray" (Sarah Borys and Mark Hasegawa-Johnson, 2009). Trong hình này, ba landmark ví dụ được trình bày: hai landmark stop closure và một landmark vowel center (landmark là thời điểm khi có biến động lớn về cách phát âm xảy ra).

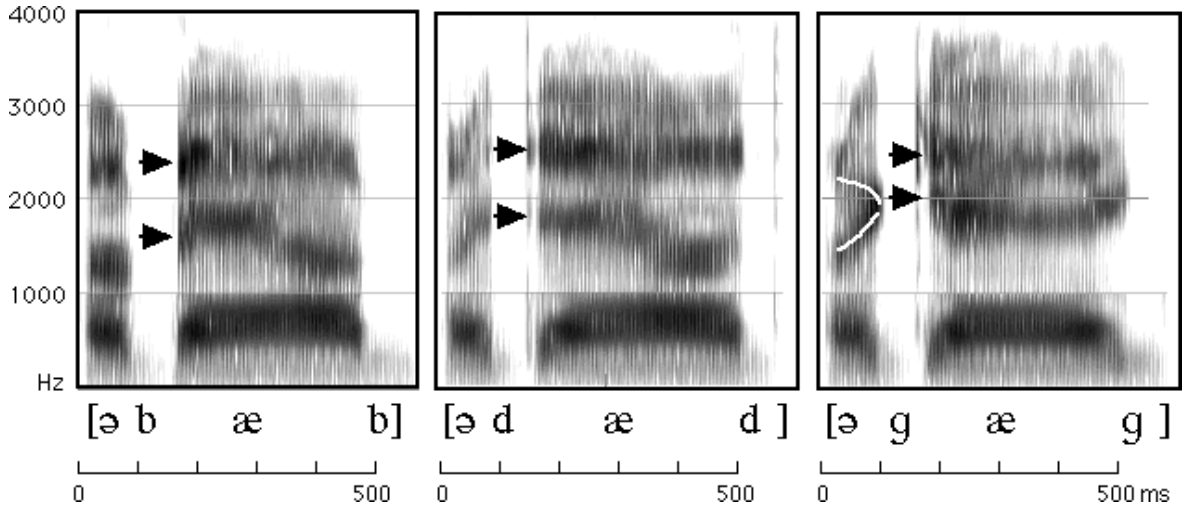
3.3.2 Formant

Formant là mức năng lượng âm thanh xung quanh một tần số cụ thể trong sóng âm. Có một số formant mà mỗi âm thanh ở một tần số khác nhau, khoảng một phần trong mỗi băng tần 1000Hz. Hay nói cách khác, formant xảy ra trong khoảng thời

gian khoảng 1000Hz. Mỗi formant tương ứng với một sự cộng hưởng trong thanh quản. Các nguyên âm có thể được nhận dạng dựa vào 2 formant (F1 và F2).



Hình 3.15 – Ba formants được thể hiện trong spectrogram



Hình 3.16 – Hai formant trong spectrogram của ba từ "bad", "dad" và "gag" (*http://www.phonetics.ucla.edu*)

- Từ “bad”: khép kín môi lại làm giảm tất cả các formants: làm tăng nhanh tất cả các formant ở vị trí đầu của từ “bad”
- Từ “dad”: formant đầu tăng nhưng F2 và F3 lại giảm nhẹ.
- Từ “gag”: F2 và F3 chạm nhau: đây là một đặc điểm của velar. Các chuyển đổi formant có thể mất nhiều thời gian trong âm vòm mềm (velar) hơn trong âm chân răng(alveolar) hoặc âm môi labial.

3.3.3 Đặc trưng ngữ âm (Acoustic feature)

Tất cả các đặc trưng ngữ âm (acoustic feature) thường được sử dụng trong nhận dạng tiếng nói tự động là dựa trên spectrogram. Một spectrogram là một biểu đồ về mặt thời gian-tần số của tín hiệu tiếng nói, $X(t, f)$, với t được tính bằng giây và f được tính bằng Hertz. Đặc biệt, $X(t_0, f)$ là logaritcủa biên độ (magnitude) của phép biến đổi Fourier, tại tần số f, của tín hiệu tiếng nói nhân với một hàm cửa sổ ngắn (short window)$w(t)$ ($w(t)$ là một hàm cửa sổ thường có độ dài khoảng 20-30ms):

$$X(t0, f) = |F\{x(t + t_0)w(t)\}| \quad (3.1)$$

$$SPEC(t_0, f) = logX(t_0, f) \quad (3.2)$$

Spectrogram có khả năng cho thấy được các âm vị một cách rõ ràng nếu nó được xây dựng dựa trên các tính toán giá trị $X(t_0, f)$ sau mỗi 2ms, nhưng để tiết kiệm bộ nhớ và độ phức tạp tính toán, các thuật toán nhận dạng giọng nói thông thường chỉ tính $X\ (t_0, f\ )$ sau mỗi 10ms. Một số trường hợp đặc biệt, việc tính toán $X\ (t_0, f\ )$ ở tần số 10ms/lần thì không đủ để nhìn thấy rõ (nhận dạng được rõ) âm vị, ví dụ các âm sát (fricative). Lí do là vì phần nhiễu (burst) của những âm này chứa ít thông tin để nhận dạng.

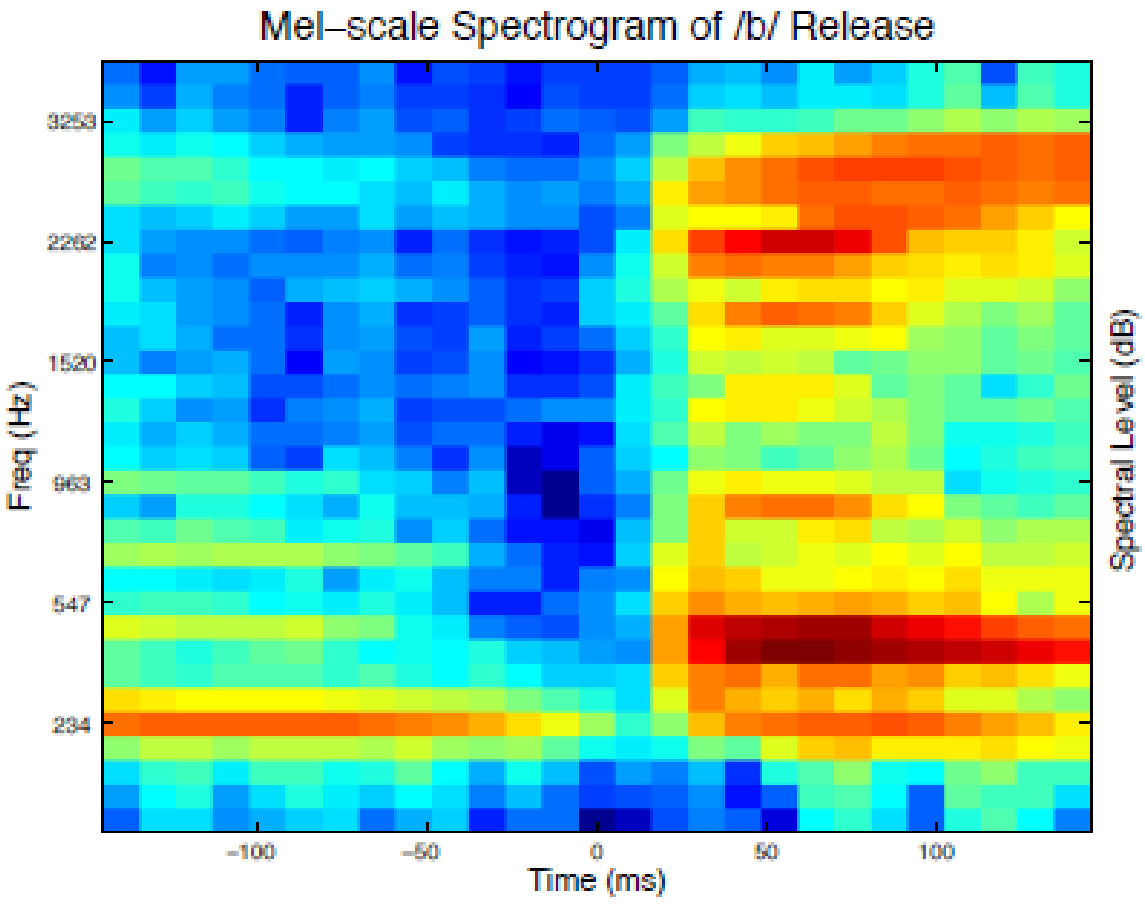
3.4 Support Vector Machine

Phần này trình bày các kiến thức cơ bản về bộ phân loại nhị phân SVM (Mark Hasegawa- Johnson, 2005) .

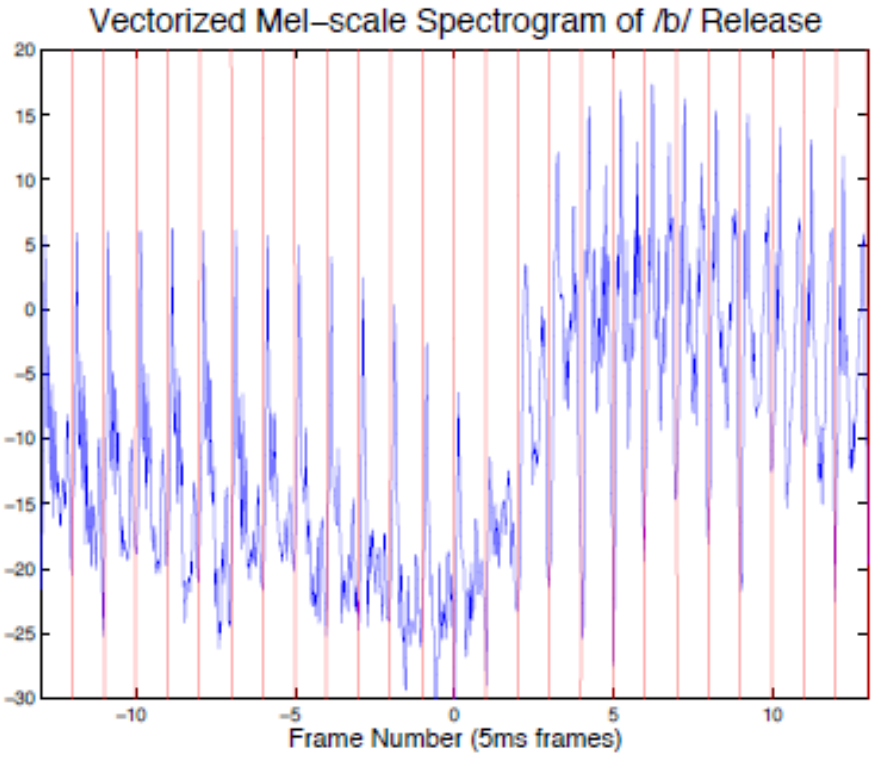
3.4.1 Các khái niệm cơ bản

Mẫu (observation)	$\vec{x} \in \Re^K$, mẫu thứ *k* là x_k
Nhãn (label)	$y \in \{-1,1\}$
Các tham số có thể học (trainable parameters)	$\vec{\theta} \in \Re^D$
Mục học (training token)	$(\vec{x}_m, y_m), \vec{x}_m = [x_{m1}, \dots, x_{mK}]'$

Một mẫu có thể chỉ là một spetral, hoặc có thể là toàn bộ spectrogram. Ví dụ, nối 27 vector MFCC (từ vector thứ t-13 đến t+13), mỗi vector gồm 32 số, ta thu được một vetor mẫu $\vec{x}_t$ có kích thước 32x27=864 số.



Hình 3.17 – Mel-scale spectrogram của phone /b/



Hình 3.18 – Véc tơ hóa mel-scale spectrogram của phone /b/

Bộ phân loại nhị phân là một hàm $h(\vec{x})$, trong đó quan sát một vector $\vec{x}$ ($\vec{x}$ có thể là một spectrum hoặc toàn bộ spectrogram) và kết quả xuất ra của hàm là một giá trị nhị phân (giá trị của một đặc trưng âm vị học riêng biệt tại một thời điểm cụ thể nào đó):

$$h(\vec{x}|\vec{\theta}) \in \{-1, 1\} \tag{3.3}$$

Giả sử rằng các nhãn và mẫu có phân phối xác suất đồng thời không đổi (constant joint probability distribution), $p(\vec{x}, y)$ thì chất lượng của một bộ phân loại được đánh giá dựa trên "rủi ro dự kiến", "rủi ro", hay "tỷ lệ lỗi dự kiến trên kho dữ liệu đánh giá " của nó.

$$R(\vec{\theta}) = E|y - h(\vec{x}|\vec{\theta})| = \sum_{y} \int |y - h(\vec{x}|\vec{\theta})| p(\vec{x}, y) dx \tag{3.4}$$

Trong các trường hợp thực tế nhất thì giá trị $p(\vec{x}, y)$ là không biết trước, nên rủi ro dự kiến cũng ko thể biết được. Chúng ta chỉ có các mục học (training token) đã

được gắn nhãn $(y_m, \vec{x}_m)$. Cho trước một mục học (training token) thì ta có thể tính được các rủi ro thực nghiệm hoặc lỗi trên kho dữ liệu huấn luyện.

$$R_{emp}(\vec{\theta}) = \frac{1}{M}\sum_{m=1}^{M} |y_m - h(\vec{x}|\vec{\theta})| \tag{3.5}$$

3.4.2 Cực đại hóa bộ phân loại hậu nghiệm (classifier posterior)

Nếu biết được mật độ xác suất hợp (joint probability) $p(\vec{x}, y)$, thì rủi ro của bộ phân loại có thể được giảm thiểu một cách rõ ràng. Bộ phân loại tối ưu là bộ phân loại cho phép lựa chọn nhãn, kí hiệu là y, nhằm cực đại hóa xác suất hậu nghiệm (posteriori probability) $(y|\vec{x})$:

$$\begin{aligned} E|y - h(\vec{x})| = &\int_{h(\vec{x})=-1} p(y = 1|\vec{x})dx \\ &+ \int_{h(\vec{x})=1} p(y = -1|\vec{x})dx \end{aligned} \tag{3.6}$$

$$h_{opt}(\vec{x}) = \arg\max p(y|\vec{x}) = \{^{1 \;\; p(y=1|\vec{x}) lớn hơn}_{-1 \;\; p(y=-1|\vec{x}) lớn hơn} \tag{3.7}$$

3.4.3 Cực tiểu hóa rủi ro về mặt cấu trúc

3.4.3.1 Lỗi tổng quát hóa và số chiều VC

Mục tiêu của học máy (machine learning) là để giảm thiểu rủi ro về mặt cấu trúc chứ không phải rủi ro thực nghiệm:

$$(\vec{v}, b) = \arg\min R(\vec{v}, b) \tag{3.8}$$

với

$$R(\vec{v}, b) = E_{p(\vec{x},y)}|y - h(\vec{x}|\vec{v}, b)| \tag{3.9}$$

trên phân phối đúng chưa biết (unknown true distribution)$p(\vec{x}, y)$.

Sự khác biệt giữa rủi ro dự kiến và rủi ro thực nghiệm được gọi là “lỗi tổng quát hóa” (generalization error).

$$R(\vec{v}, b) = R_{emp}(\vec{v}, b) + GeneralizationError, R_{emp} = Known, GeneralizationError = Unknown \quad (3.10)$$

(Burges 1998)và (Vapnik 1998) đã cho thấy lỗi tổng quát hóa (generalization error) được chặn trên bởi hàm cận tuyến tính (nearly linear) của số chiều VC, ký hiệu D_{VC}:

$$P\{R(\vec{v}, b\} \leq R_{emp}(\vec{v}, b) + f\left(\frac{D_{VC} - log\delta}{M}\right)\} \geq 1 - \delta \quad (3.11)$$

trong đó hàm $f(\cdot)$ là đơn điệu tăng và cận (roughly) tuyến tính. Số chiều của VC là thước đo sự linh hoạt (flexibility) của các hàm phân loại $h(\vec{x}|\theta)$. Nếu, bằng cách thay đổi các tham số vector θ, ta có thể gắn nhãn cho một kho dữ liệu huấn luyện lớn theo rất nhiều cách khác nhau, thì bộ phân loại sẽ có số chiều VC lớn (a high VC dimension).

Số chiều VC của bộ phân loại tuyến tính hoàn toàn nhỏ hơn số tham số có thể huấn luyện (traianble parameter):

Bộ phân loại tuyến tính (Linear Classifier):

$$D_{VC} \leq K + 1, K = length(\vec{v}) = length(\vec{x}) \quad (3.12)$$

Công thức (3.11) kết hợp với (3.12) đưa ra luật sau: để giảm thiểu $R(\vec{\theta})$, chúng ta nên chọn bộ phân loại có các điểm sau:

(1) rủi ro thực nghiệm thấp nhất có thể,

(2) có số tham số nhỏ nhất có thể.

Sự cân bằng giữa số lượng tham số của bộ phân loại với rủi ro thực nghiệm thu được bằng BIC (Bayesian Information Criterion). Bằng cách tối ưu hóa BIC, ta có thể lựa chọn giữa các bộ phân loại với nhiều mức độ phức tạp khác nhau.

Vapnik chứng minh rằng, trong nhiều trường hợp, cận trên trong biểu thức (3.12) là quá lớn; số chiều VC thật của một bộ phân loại tuyến tính có thể thấp hơn nhiều. Xét các tình huống sau đây.Với kho dữ liệu huấn luyện X bất kỳ, ta chuẩn hóa (normalize) v, b bằng công thức:

$$\min|\vec{v}^T\vec{x}_m - b| = 1 \tag{3.13}$$

Biểu thức 3.13 cho thấy khoảng cách tối thiểu giữa siêu phẳng (hyperplane) và một điểm dữ liệu riêng lẻ bất kì là $r = \frac{1}{|\vec{v}|}$. Đặt R là bán kính của hình tròn chứa tất cả các điểm dữ liệu $\vec{x}_m$. Khi đó:

$$D_{VC} \leq \left(\frac{R}{r}\right)^2 = (R|\vec{v}|)^2 \tag{3.14}$$

Theo biểu thức 3.14, ta có thể kiểm soát các lỗi tổng quát hóa của bộ phân loại bằng cách kiểm soát độ lớn của $\vec{v}$ bằng điều kiện trong biểu thức 3.14:

$$R(\vec{v}, b) \leq R_{emp}(\vec{v}, b) + R^2|\vec{v}|^2 \tag{3.15}$$

3.4.3.2 Linear Support Vector Machine

Lưu ý rằng, trong bộ phân loại tuyến tính bất kỳ, lỗi phân loại xảy ra nếu $y_m\ (\vec{v}^T\vec{x}_m - b) < 0$. Ta định nghĩa lỗi bộ phận (partial error) là một biểu thức $y_m\ (\vec{v}^T\vec{x}_m - b) < 1$, và định nghĩa biến bù (slack variable) ξ_m như sau:

$$\xi_m = \max\big(0{,}1 - y_m(\vec{v}^T\vec{x}_m - b)\big) \tag{3.16}$$

Lỗi trên kho dữ liệu huấn luyện được ràng buộc bằng công thức sau:

$$R_{emp}(\vec{v}, b) \leq \sum_{m=1}^{M} \xi_m \tag{3.17}$$

Kết hợp hai biểu thức (3.15) với (3.17), chúng ta thấy rằng rủi ro về mặt cấu trúc của một bộ phân loại tuyến được chặn theo biểu thức sau:

$$R(\vec{v}, b) \leq R^2|\vec{v}|^2 + \sum_{m=1}^{M} \xi_m \tag{3.18}$$

Phần bên trái của biểu thức 3.18 là tiêu chí tối ưu được giảm thiểu bằng một SVM (support vector machine). Do đó, mục đích huấn luyện một SVM là để giảm thiểu biểu thức 3.18, với ràng buộc như trong biểu thức 3.16. Bài toán tối ưu hóa ràng

buộc cụ thể này (particular constrained optimization problem) được gọi là bài toán tối ưu hóa bậc hai (quadratic programming).

Cực tiểu hóa (minimize):

$$(\vec{v}, b) = \arg\min \frac{1}{2}|\vec{v}|^2 + C\sum_{m=1}^{M}\xi_m \tag{3.19}$$

Với

$$\xi_m = \max\left(0,1 - y_m(\vec{v}^T\vec{x}_m - b)\right) \tag{3.20}$$

Biểu thức 3.19 và 3.20 có thể được chuyển đổi tương đương thành bài toán tối ưu hóa bậc hai, tìm các hệ số α_m như sau:

$$\vec{v} = \sum_{m=1}^{M}\alpha_m\,\vec{x} \tag{3.21}$$

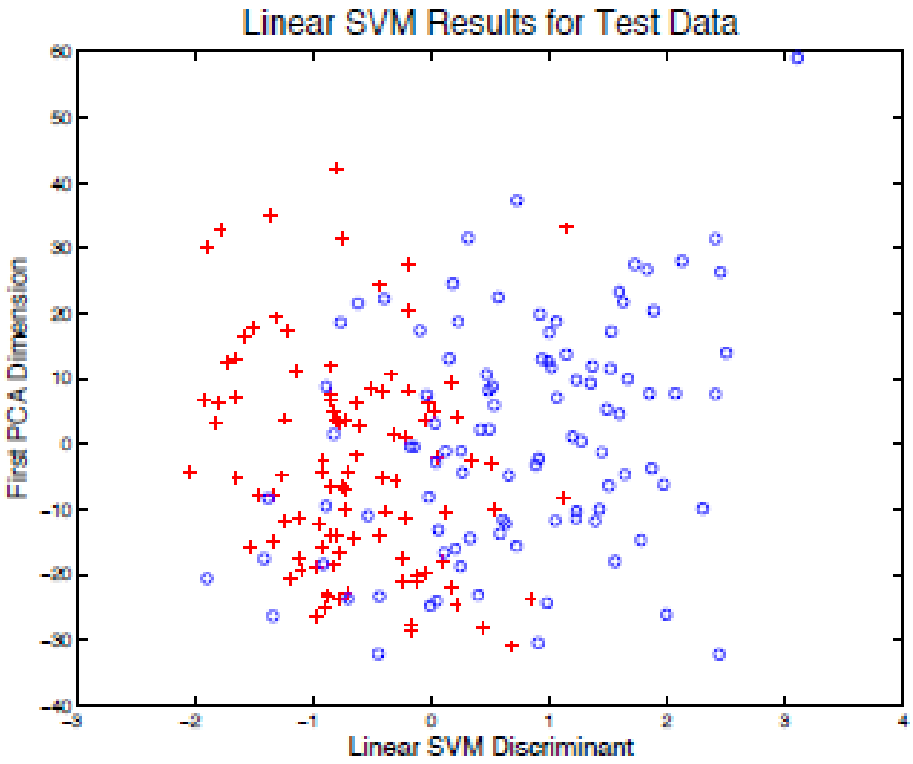
Với

$$\alpha_m = \arg min \sum_{m}\sum_{m}\alpha_m\vec{x'}_m\vec{x}_n\alpha_n - \sum_{m}y_m\alpha_m \tag{3.22}$$

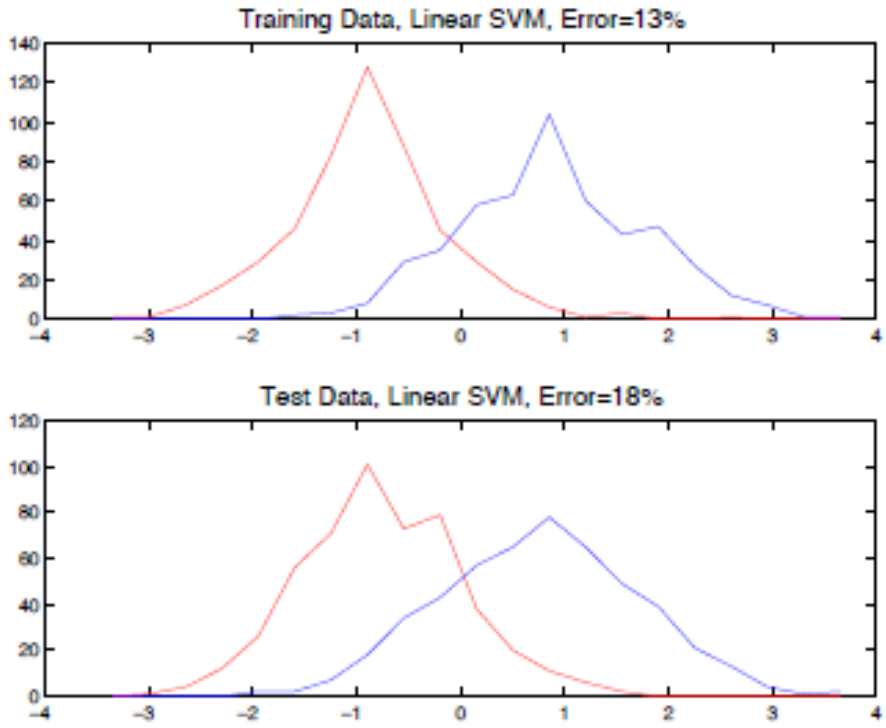
Và

$$\sum_{m=1}^{M}\alpha_m = 0, \qquad 0 \le y_m\alpha_m \le C \tag{3.23}$$

Bằng cách giảm thiểu các rủi ro về mặt cấu trúc (biểu thức 3.18), SVM tránh được việc huấn luyện quá mức vào dữ liệu, trong đó hoặc là (1) có quá ít mục huấn luyện, hoặc là (2) các vector mẫu (observation) là quá lớn để một bộ phân loại tiêu chuẩn (LDA hoặc neural network) có thể hoạt động tốt.



Hình 3.19 – Kết quả sử dụng SVM tuyến tính trên dữ liệu kiểm tra (test data)



Hình 3.20 – So sánh kết quả SVM tuyến tính trên dữ liệu huấn luyện và dữ liệu kiểm tra

3.4.3.3 Từ SVM tuyến tính đến SVM phi tuyến tính

Lưu ý rằng, theo biểu thức 3.21, các vector riêng biệt tối ưu $\vec{v}$ là một kết hợp có trọng số của các mục huấn luyện. Do đó, các hàm phân loại $h(\vec{x})$ có thể viết như sau:

$$h(\vec{x}) = |sign(\vec{v}^T\vec{x} - b) = sign(-b + \sum_{m=1}^{M} \alpha_m \vec{x}_m^T \vec{x}) \qquad (3.24)$$

Biểu thức 3.24 cho thấy rằng $h(\vec{x})$ chỉ phụ thuộc vào các tích vô hướng (dot-product) giữa vectơ huấn luyện $\vec{x}_m$ và vector kiểm tra $\vec{x}$ mà ta chưa biết. Trong thực tế, có thể sử dụng hàm xác định dương (positive-definite function) $(\vec{x}_m, \vec{x})$ bất kỳ: (any symmetric, positive-definite function $(\vec{x}_m, \vec{x})$).

$$h(\vec{x}) = sign(-b + \sum_{m=1}^{M} \propto_m K(\vec{x}_m, \vec{x})) \qquad (3.25)$$

Một trường hợp phổ biến, linh hoạt, và cực kỳ hữu dụng là RBF SVM, được định nghĩa bởi hàm nhân (kernel function) như sau:

$$K(\vec{x}_m, \vec{x}) = e^{-\gamma|\vec{x}_m - \vec{x}|^2} \qquad (3.26)$$

Hàm phân loại kết quả là

$$h(\vec{x}) = sign(g(\vec{x}) - b) \qquad (3.27)$$

với hàm phân biệt phi tuyến tính $g(\vec{x})$ được định nghĩa bằng biểu thức:

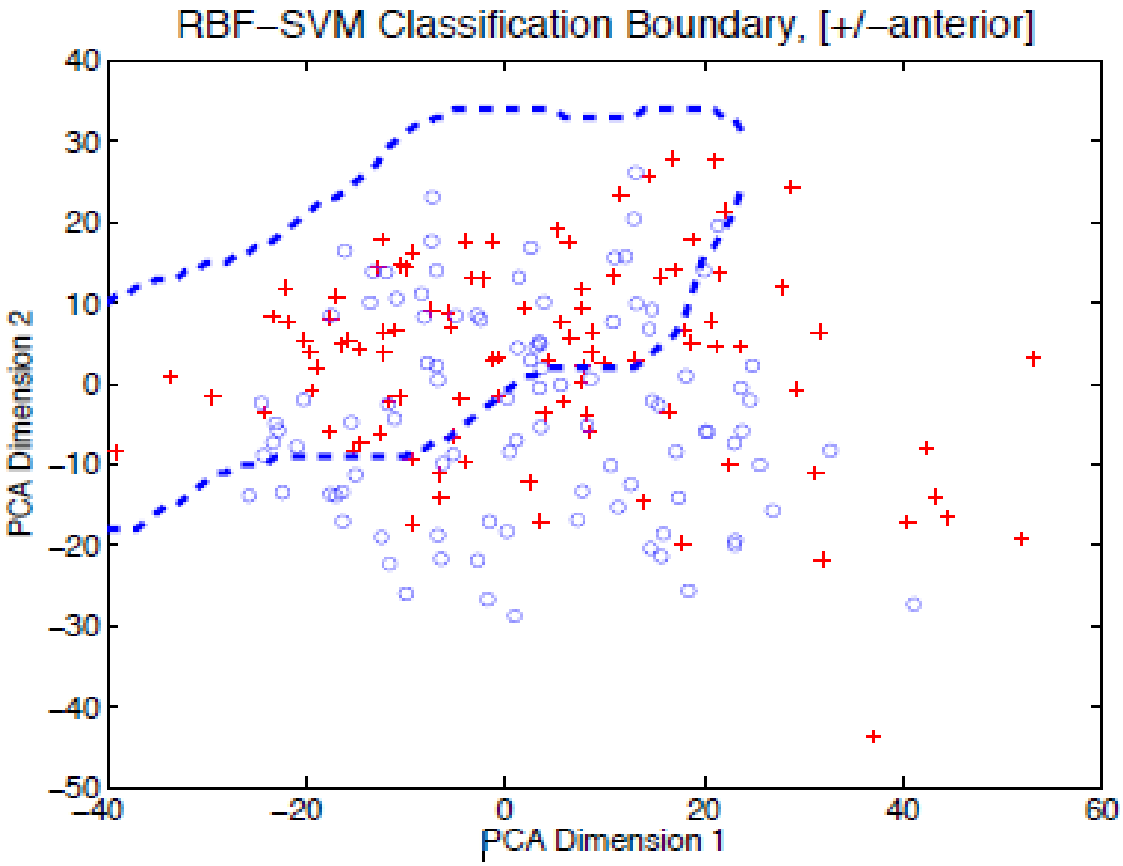
$$g(\vec{x}) = \sum_{m-1}^{M} \propto_m K(\vec{x}_m, \vec{x}) \qquad (3.28)$$

Một bộ phân loại RBF tổng quát rất có tính chất rất linh hoạt ở chỗ nó có thể cho bộ phân loại RBF "phân mảnh" (fracture) một kho dữ liệu huấn luyện bằng vô số cách khác nhau, do đó không có khái niệm chặn trên(upper bound) về số chiều VC của nó. Bộ phân loại RBF được huấn luyện bằng cách sử dụng các tiêu chí SVM (công thức 3.14), tuy nhiên, số chiều VC bị hạn chế, do đó có thể sử dụng các phương

pháp huấn luyện cho SVM học bộ phân loại RBF với tỷ lệ lỗi tổng quát (generalization error) rất thấp:

$$D_{VC} \leq f(|\vec{v}|^2) = f\left(\sum_m \sum_n \propto_m \propto_n K(\vec{x}_m, \vec{x}_n)\right) \qquad (3.29)$$

Dưới đây là một ví dụ về ranh giới phân loại được tính bởi bộ phân loại RBF



Hình 3.21 – Đường ranh giới (boundary) của phân loại RBF-SVM

CHƯƠNG 4: THÍ NGHIỆM VÀ ĐÁNH GIÁ

Phương pháp được trình bày có thể được mở rộng cho người học ngoại ngữ ở các nước khác với điều kiện phải cung cấp các mẫu phát âm sai tiếng Anh của họ.

Trong luận văn này, tác giả thực hiện thí nghiệm trên hai bộ dữ liệu: TIMIT và Buckeye. Bộ dữ liệu Buckeye được dùng để huấn luyện các SVM, bộ dữ liệu TIMIT dùng để đánh giá các SVM đã được huấn luyện. Thí nghiệm này được thực hiện nghiệm dựa trên công trình của (Yoon et al. 2009). Tuy nhiên, tác giả sử dụng Buckeye corpus để huấn luyện SVMs thay vì TIMIT và dùng 3 formants thay vì 2 formants. Ngoài ra thì tập các âm vị cũng khác nhau.

4.1 Mô tả các kho dữ liệu được sử dụng trong thí nghiệm

4.1.1 Kho dữ liệu TIMIT

TIMIT là kho dữ liệu giọng đọc được thiết kế để cung cấp dữ liệu tiếng nói cho các nghiên cứu về ngữ âm và nhận dạng tiếng nói tự động. TIMIT bao gồm các đoạn ghi âm tiếng nói với dải băng tần rộng của 630 người (speaker) với 8 giọng địa phương chính của tiếng Anh Mỹ, mỗi người đọc 10 câu ngữ âm được thiết kế một tập cân bằng các âm vị của tiếng Anh (nghĩa là trong mỗi câu chứa đủ 144 âm vị tiếng Anh với tần số đồng đều, không có âm vị nào xuất hiện quá nhiều hoặc quá ít). Các câu này được gọi là “SX” (“SX” sentence). Bộ dữ liệu này chứa phần phiên âm ở mức độ câu, từ và âm vị của tất cả các câu đã được đọc bởi 630 người nêu trên. Giá trị tại các thời điểm lấy mẫu được lưu bởi 16 bit (2 bytes). Kho dữ liệu này được thiết kế và xây dựng bởi Viện Công nghệ Massachusetts (MIT) và Texas Instruments (TI). Dữ liệu tiếng nói được ghi âm tại TI, sao chép lại tại MIT và được kiểm nghiệm, chuẩn bị cho việc ghi đĩa CD-ROM tại Viện Tiêu chuẩn và Công nghệ (NIST). Các phiên âm của TIMIT được kiểm tra bởi các chuyên gia ngữ âm học tại viện công nghệ MIT. Ngoài ra, bộ dữ liệu này cũng xác định rõ ràng các tập con dùng để huấn luyện và đánh giá các bộ phân loại (classifier). Cả hai tập này đều

chứa một tập cân bằng các âm vị của tiếng Anh (nghĩa là không có âm vị nào xuất hiện quá nhiều hoặc quá ít). Ngoài ra cả hai tập này đều bao phủ tất cả các giọng đọc từ các vùng miền được xác định bởi người thiết kế. Tập dữ liệu dùng để đánh giá được gọi là tập dữ liệu đánh giá trong quá trình xây dựng (development test dataset). Các tập tin phiên âm (cấp độ câu – từ - âm vị) là các tập tin văn bản dạng bảng (tabular text) giúp thuận tiện trong việc tìm kiếm thông tin và xử lý các phiên âm tự động bằng máy tính.

4.1.2 Mô tả bộ dữ liệu mẫu của TIMIT

Kho dữ liệu này chứa một phần của kho TIMIT, bao gồm các tập tin tiếng nói, các phiên âm ở cấp độ câu, từ và âm vị. Nó chứa âm thanh của 16 người từ 8 miền khác nhau. Mỗi miền gồm một nam và một nữ phát âm. Tổng cộng có 130 câu (mỗi người phát âm 10 câu; lưu ý rằng một số câu được phát âm bởi nhiều người, trong đó 2 câu sa1 và sa2 được tất cả 16 người phát âm). Tổng cộng có 160 câu được ghi âm (mỗi người 10 câu). Các tập tin âm thanh được lưu dưới định dạng wav, 1 kênh (single channel), tần số lấy mẫu là 16kHz (16kHz sampling), giá trị của mỗi mẫu được lưu bằng 2 bytes (16 bit sample), giá trị các mẫu được mã hóa theo PCM (PCM encoding).

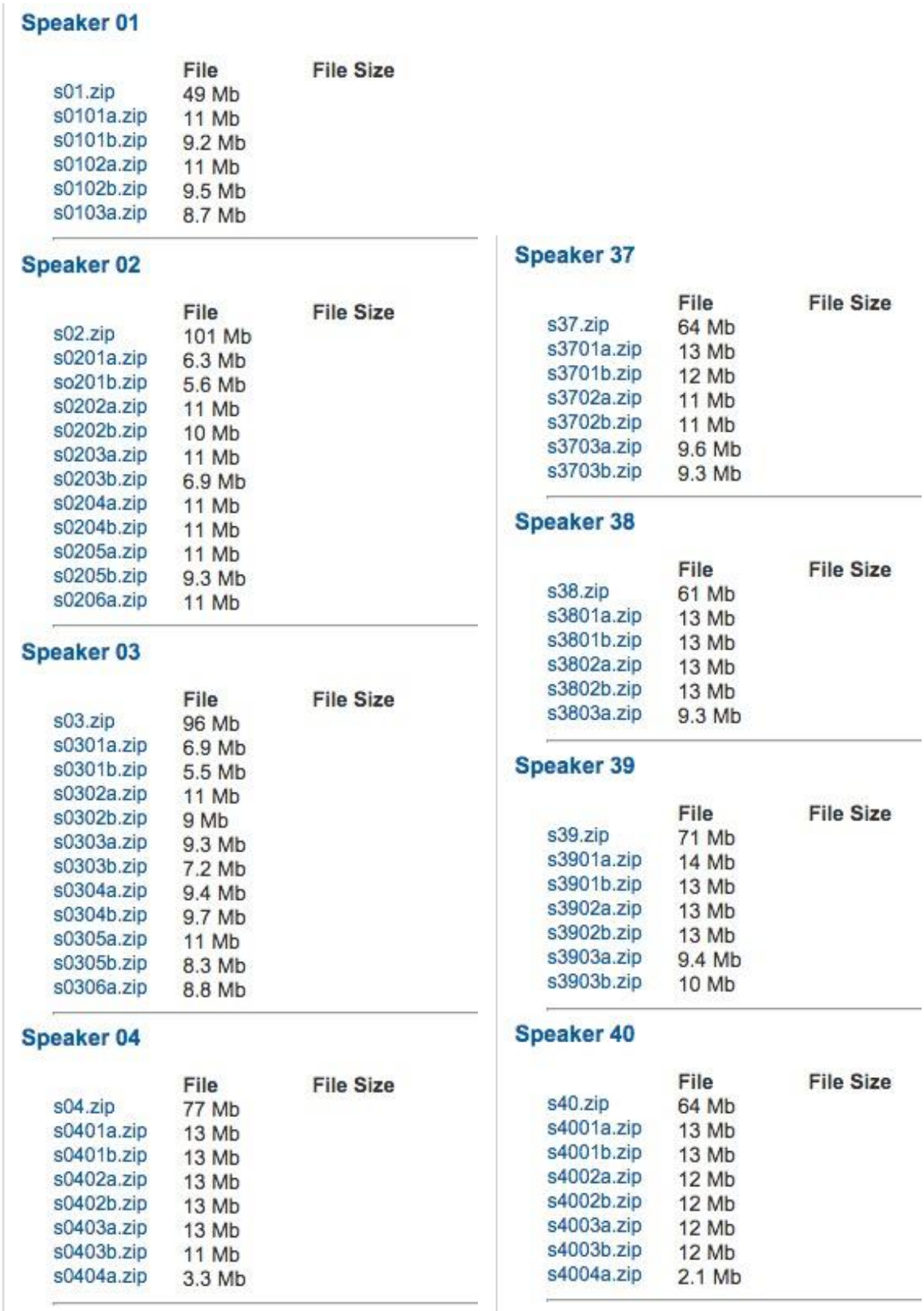
4.1.3 Kho dữ liệu Buckeye

Buckeye (http://www.buckeyecorpus.osu.edu) chứa tập tin âm thanh không có kịch bản, thu từ các cuộc phỏng vấn từ 40 người ở các độ tuổi khác nhau, sinh sống tại Columbus, Ohio. Phiên âm ở mức độ âm vị bao gồm dạng nguyên gốc từ điển (citation form) và âm vị phát âm thực tế bị biến đổi bởi cách phát âm nhanh trong trong đàm thoại (actual pronunciation) được lưu trữ trong các tập tin đi kèm. Thời điểm tương ứng với bắt đầu và kết thúc của các âm vị cũng được cung cấp. Tổng kích thước của bộ dữ liệu là 3GB. Bộ dữ liệu chứa khoảng 300,000 từ. Để ghi âm các cuộc đàm thoại, những người tham gia ghi âm được hỏi về các chủ đề hàng ngày, nhưng không biết rằng phần đàm thoại này đã được ghi âm.

Mỗi file âm thanh (.wav) đi kèm với 3 file văn bản: .words, .phones, .txt. File .words chứa phiên âm của âm thanh ở mức độ từ và thời điểm (tính bằng giây, kể từ thời điểm bắt đầu file .wav, tức là offset) bắt đầu và kết thúc của từ đó trong file .wav (time-aligned word labels). Tương tự, file .phones chứa phiên âm và các thời điểm bắt đầu và kết thúc của các âm tố (phones).

Bảng 4.1 – Các loại tập tin trong kho dữ liệu Buckeye

Loại file	Nội dung
.wav	Chứa âm thanh tiếng nói.
.words	Phiên âm ở mức từ và canh thời gian của các từ.
.phones	Chứa phiên âm và canh thời gian ở mức âm tố (phone).
.log	Các ghi chú của người thực hiện phiên âm (labeler), đánh dấu các thời điểm bất thường về chất lượng âm thanh và cách thức phát âm.
.txt	Chứa phiên âm ở mức câu.



Speaker 01

File	File Size
s01.zip	49 Mb
s0101a.zip	11 Mb
s0101b.zip	9.2 Mb
s0102a.zip	11 Mb
s0102b.zip	9.5 Mb
s0103a.zip	8.7 Mb

Speaker 02

File	File Size
s02.zip	101 Mb
s0201a.zip	6.3 Mb
so201b.zip	5.6 Mb
s0202a.zip	11 Mb
s0202b.zip	10 Mb
s0203a.zip	11 Mb
s0203b.zip	6.9 Mb
s0204a.zip	11 Mb
s0204b.zip	11 Mb
s0205a.zip	11 Mb
s0205b.zip	9.3 Mb
s0206a.zip	11 Mb

Speaker 03

File	File Size
s03.zip	96 Mb
s0301a.zip	6.9 Mb
s0301b.zip	5.5 Mb
s0302a.zip	11 Mb
s0302b.zip	9 Mb
s0303a.zip	9.3 Mb
s0303b.zip	7.2 Mb
s0304a.zip	9.4 Mb
s0304b.zip	9.7 Mb
s0305a.zip	11 Mb
s0305b.zip	8.3 Mb
s0306a.zip	8.8 Mb

Speaker 04

File	File Size
s04.zip	77 Mb
s0401a.zip	13 Mb
s0401b.zip	13 Mb
s0402a.zip	13 Mb
s0402b.zip	13 Mb
s0403a.zip	13 Mb
s0403b.zip	11 Mb
s0404a.zip	3.3 Mb

Speaker 37

File	File Size
s37.zip	64 Mb
s3701a.zip	13 Mb
s3701b.zip	12 Mb
s3702a.zip	11 Mb
s3702b.zip	11 Mb
s3703a.zip	9.6 Mb
s3703b.zip	9.3 Mb

Speaker 38

File	File Size
s38.zip	61 Mb
s3801a.zip	13 Mb
s3801b.zip	13 Mb
s3802a.zip	13 Mb
s3802b.zip	13 Mb
s3803a.zip	9.3 Mb

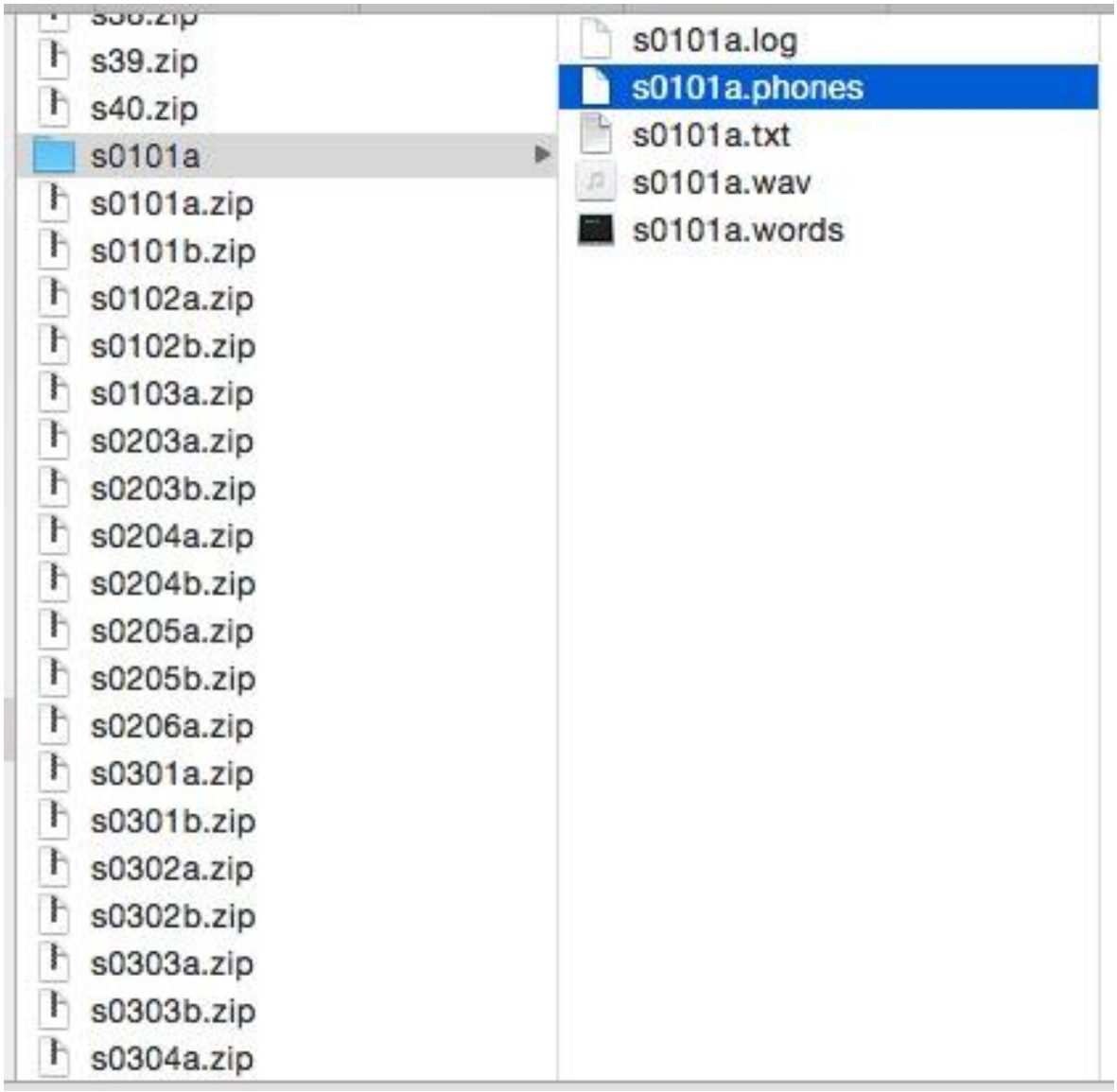
Speaker 39

File	File Size
s39.zip	71 Mb
s3901a.zip	14 Mb
s3901b.zip	13 Mb
s3902a.zip	13 Mb
s3902b.zip	13 Mb
s3903a.zip	9.4 Mb
s3903b.zip	10 Mb

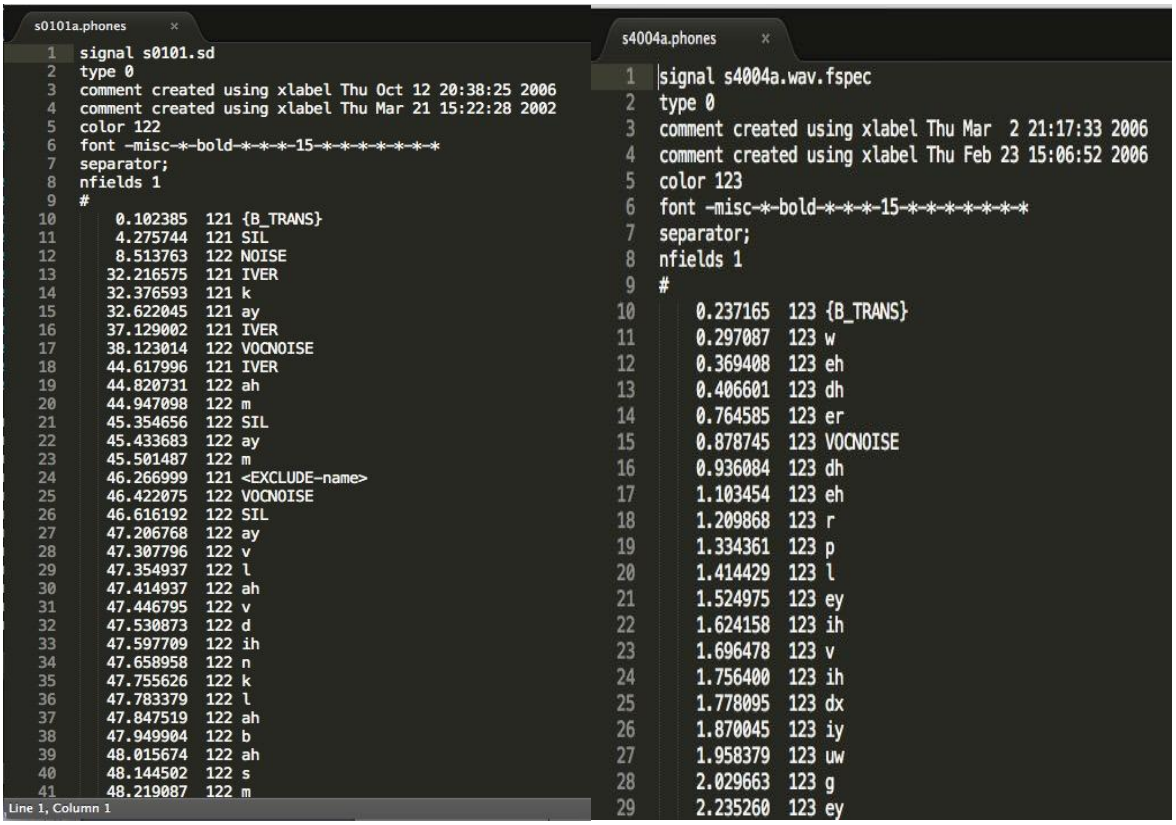
Speaker 40

File	File Size
s40.zip	64 Mb
s4001a.zip	13 Mb
s4001b.zip	13 Mb
s4002a.zip	12 Mb
s4002b.zip	12 Mb
s4003a.zip	12 Mb
s4003b.zip	12 Mb
s4004a.zip	2.1 Mb

Hình 4.1 – Giao diện trang web tải kho dữ liệu Buckeye



Hình 4.2 – Hệ thống tập tin đã được tải về đĩa



Hình 4.3 – Nội dung của một tập tin phiên âm ở mức âm vị (.phones)

4.2 Các thư viện và công cụ dùng trong thí nghiệm

4.2.1 Thư viện HTK và công cụ HCopy

Trong luận văn này, tác giả tính vector đặc trưng ngữ âm bằng cách dùng công cụ Hcopy của HTK. HCopy nhận đầu vào một file âm thanh và một file cấu hình. Các thông số cần thiết để tính vector đặc trưng ngữ âm được cung cấp trong file cấu hình như sau (trong file config.txt):

- SOURCEFORMAT = NIST
- SOURCEKIND = WAVEFORM
- TARGETKIND = PLP_E_D_A_Z_C
- TARGETRATE = 100000.0
- WINDOWSIZE = 250000.0

- NUMCHANS = 32
- CEPLIFTER = 27
- NUMCEPS = 12

Ý nghĩa các cấu hình trên như sau:

Bảng 4.2 – Ý nghĩa các tham số được dùng để tính AF dùng thư viên HTK

Tên tham số	Ý nghĩa của tham số	Đơn vị tính	Giá trị có thể nhận
SOURCEFORMAT	Định dạng của file âm thanh	N/A	NIST, WAV, HTK
TARGETFORMAT	Định dạng của file chứa kết quả	N/A	N/A
SOURCEKIND	Chỉ định dữ liệu vào là dữ liệu waveform	N/A	N/A
TARGETKIND	Xác định loại đặc trưng ngữ âm	N/A	PLP, MELSPEC, FBANK, MFCC, LPCEPSTRA
WINDOWSIZE	Chiều dài của cửa sổ w(t) tính bằng đơn vị 100ns	100ns	$25ms = 25000\mu s = 25000 \times 100ns$
TARGETRATE	Khoảng cách (thời gian)giữa hai vị trí đầu của 2 frame liên tiếp. Hoặc tần số tính toán AF cho frame (cho biết sau bao lâu thì tính toán AF một lần).	10ms	$10ms = 1000\mu s = 10000 \times 100ns$
NUMCEPS	Số hệ số cepstra được dùng	N/A	12 (do dó AF sẽ gồm 12 hệ số cepstra, năng lượng trên frame, đạo hàm bậc 1 và bậc 2, tổng cộng là 39 hệ số)

Bảng 4.3 – Ý nghĩa các tham số phụ đi kèm với tham số TARGETKIND

Tham số phụ	Ý nghĩa
_E	Thêm năng lượng (energy) của frame vào AF.
_D	Thêm đạo hàm bậc 1 (delta) của các hệ số cepstra và của năng lượng vào AF.
_A	Thêm đạo hàm bậc hai (delta-delta) của các hệ số cepstra và của năng lượng vào AF.
_Z	Thực hiện phép trừ giá trị trung bình: trừ từng AF cho vector trung bình của tất cả các AF trên file âm thanh. Tức là làm cho giá trị trung bình của các AF mới bằng không (zero mean).
_C	Nén (compress) file chứa các hệ số.

4.2.2 Thư viện SVM

SVMLight_Binary (Thorsten Joachims) là một chương trình hữu ích cho việc huấn luyện các support vector machines (SVM) . Thư viện này gồm hai công cụ, svm_classify.exe, svm_learn.exe. Công cụ svm_learn được sử dụng để huấn luyện các vector đầu vào ,và svm_classify để phân lớp dữ liệu vào lớp 1, hoặc -1.

Tập tin dữ liệu đầu vào có định dạng sau:

$$O_1\ 1{:}\,value1\ 2{:}\,value2\ 3{:}\,value3\ldots$$

$$O_2\ 1{:}\,value1\ 2{:}\,value2\ 3{:}\,value3\ldots$$

$$O_3\ 1{:}\,value1\ 2{:}\,value2\ 3{:}\,value3\ldots$$

$$\ldots$$

$$O_N\ 1{:}\,value1\ 2{:}\,value2\ 3{:}\,value3\ldots$$

với $O_{1....}O_N$ là một loạt các observation và các dòng "1: $value1$ 2: $value2$ 3: $value3$ .." đại diện cho các vector tương ứng với mỗi mẫu. Các con số 1, 2, 3, ở phía trước dấu ":" là những chỉ số của các số, mỗi chỉ số đại diện bởi "valueindex" chứa trong vector. Ví dụ tập tin đầu vào được thể hiện trong Hình 4.4.

```
-1    1:-0.100951 2:-0.0237191 3:-0.0392793 4:0.515768
1     1:-0.780494 2:0.667376 3:-0.0564404 4:0.530383
-1    1:0.508571 2:0.0834868 3:0.104343 4:0.475912
-1    1:0.171674 2:-0.522371 3:-0.348407 4:0.550595
-1    1:0.518083 2:-0.723123 3:-0.445102 4:0.189794
1     1:-0.204792 2:-0.00798731 3:-0.163721 4:0.441962
```

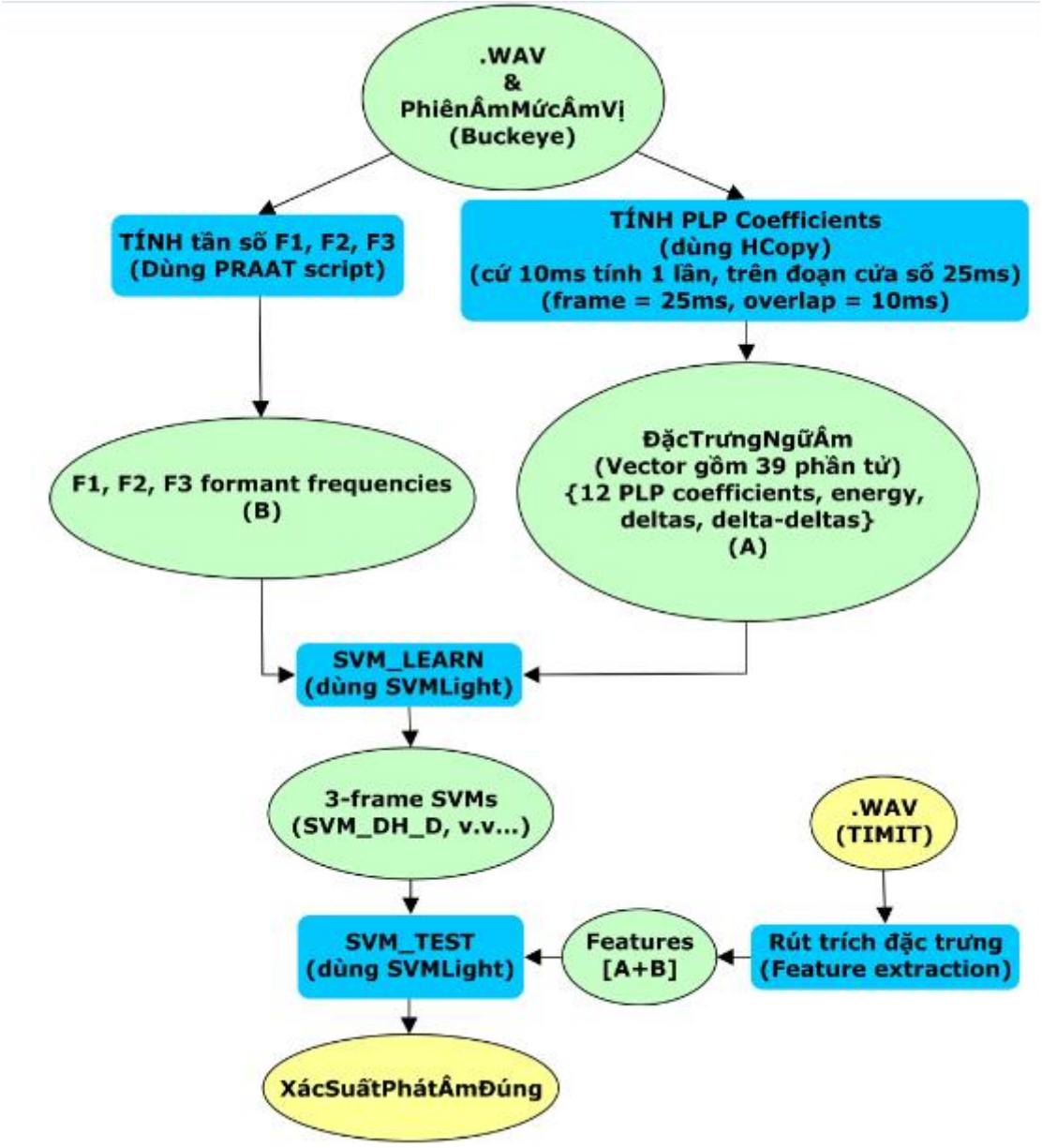
Hình 4.4 – Ví dụ minh họa tập tin SVM đầu vào

Trong hình này, các vector đầu vào có chiều dài là 4. Tập tin đầu vào thực sự được sử dụng chứa các vector có chiều dài là 429.

4.2.3 Praat

Praat là phần mềm được dùng để phân tích một cách linh động và ghi lại lời phát biểu. Đây là phần mềm miễn phí tương thích nhiều hệ điều hành, được tải về tại địa chỉ www.praat.org.Ngoài việc cung cấp giao diện đồ họa cho phép thao tác với các tập âm thanh, thực hiện việc phân tích tiếng nói và phiên âm ở mức độ từ, âm vị, Praat còn cung cấp bộ dịch ngôn ngữ script cho phép thực hiện các phân tích và phiên âm bằng cách lập trình (chứ không phải thao tác trên giao diện đồ họa). Ngoài ra, Praat cũng cung cấp chương trình chạy trên Windows có tên PraatCon cho phép thực hiện việc trích các formant từ tập tin tiếng nói. Trong thí nghiệm này, tác giả dùng PraatCon để tiến hành rút trích ba thành phần formant sẽ được dùng cùng với 39 thành phần của đặc trưng tiếng nói trong việc huấn luyện các SVM.

4.3 Huấn luyện các SVM

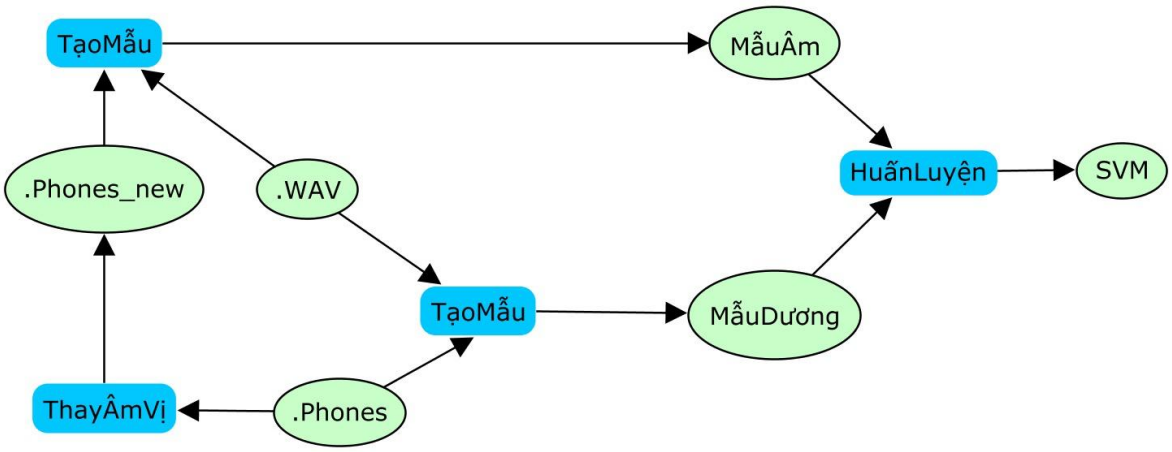


Hình 4.5 – Sơ đồ tổng quát của quá trình xử lý tiếng nói trong thí nghiệm

Trong nghiên cứu này, tác giả huấn luận 3 SVM. Mỗi SVM được huấn luyện để phân biệt phát âm đúng và phát âm sai ứng với từng âm vị. Các ví dụ về phát âm đúng trong kho dữ liệu được dùng để huấn luyện SVM nhận biết đâu là phát âm đúng. Trong luận văn này, tác giả dùng cụm từ "mẫu dương" (possitive sample) để chỉ các phát âm đúng này. Ngược lại, các mẫu phát âm sai trong bộ dữ liệu huấn

luyện được dùng để huấn luyện các SVM nhận biết đâu là phát âm sai, và cụm từ "mẫu âm" (negative sample) được dùng để chỉ các mẫu sai này. Ứng với mỗi SVM, số lượng mẫu dương và mẫu âm được dùng để huấn luyện là như nhau.

Hình 4.6 sau đây mô tả quá trình huấn luyện một SVM và các dữ liệu cần thiết cho việc huấn luyện này.



Hình 4.6 – Quá trình huấn luyện một SVM và các dữ liệu cần thiết

Các SVM được huấn luyện dùng tập dữ liệu Buckeye và được đánh giá trên tập dữ liệu TIMIT mẫu (xem phần 4.1.2).

Để huấn luyện các SVM (Hình 4.6), ta dùng các đặc trưng ngữ âm (acoustic feature) gồm 39 hệ số, trong đó: 12 hệ số PLP (PLP coefficients), năng lượng của tín hiệu ngữ âm (energy), đạo hàm bậc 1 và bậc 2 của các hệ số trên (deltas và acceleration). Trong thí nghiệm tác giả sử dụng đặc trưng ngữ âm PLP vì là một trong những đặc trưng phù hợp với phương pháp phát hiện lỗi phát âm âm vị, ngoài ra còn có đặc trưng phổ biến khác trong nhận dạng giọng nói như hệ số MFCC. Các vectơ đặc trưng này được trích chọn qua quá trình biến đổi đặc trưng của âm thanh tiếng nói, quá trình này được tích hợp trong bộ công cụ Hcopy của bộ thư viện HTK được tính trên từng đoạn có chiều dài 25ms (window size = 25ms, hay frame = 25ms) và cứ 10ms ta thực hiện tính vector đặc trưng một lần (tức các frame chồng nhau một đoạn 10ms), và 3 giá trị formants (f1, f2, f3 formant frequency) được tính bằng phương pháp nêu trong (Boersma and Weenink 2010). So với mô hình của

ông Yoon đề xuất là chỉ dùng 2 giá trị format thì thí nghiệm tác giả tiến hành trích chọn thêm 1 giá trị formant f3 để tăng thêm dữ liệu cho quá trình nhận dạng phát âm, vì qua tìm hiểu thì tác giả nhận thấy rằng tần số f3 vẫn còn mang nhiều năng lượng ngữ âm có thể khai thác cho quá trình nhận dạng phát âm.

Như vậy vec-tơ đặc trưng cho một đoạn (frame) ngữ âm gồm 42 hệ số, trong đó có 39 hệ số đại diện cho spectral feature và 3 hệ số là 3 giá trị của formant 1, 2 và 3. Các hệ số của spectral feature được tính bằng cách dùng thư viện HTK và ba giá trị formant được tính như đã đề cập ở trên.

Để tính vec-tơ đại diện cho một âm tố (phone), ta kết hợp 3 vec-tơ đặc trưng cho n frame nằm trong đoạn ngữ âm của âm tố đó. Như vậy, vec-tơ đại diện cho một âm vị sẽ có chiều dài là nx42. Giá trị của n chạy từ 1 đến 3, tuỳ theo độ dài của âm tốt, hay nói cách khác, vec-tơ đại diện cho âm tố sẽ có độ dài là một trong các giá trị sau: 42 (n = 1), 84 (n = 2), 126 (n = 3).

Đối với nguyên âm, frame/các frame được chọn để tính vec-tơ đại diện cho âm tố nằm ở trung tâm của nguyên âm. Đối với phụ âm, frame/các frame được chọn để tính vec-tơ đặc trưng nằm ở phần rìa của phụ âm trong trường hợp C-V (phụ âm đứng trước nguyên âm) hoặc V-C (nguyên âm đứng trước phụ âm), và nằm ở trung tâm của phụ âm trong trường hợp C-C (hai phụ âm đứng cạnh nhau). Cách chọn các frame này được giải thích trong (Hasegawa-Johnson et al. 2005). Theo đó, các frame nằm ở trung tâm nguyên âm chứa nhiều thông tin nhất về vị trí tương đối của các bộ phận phát âm khi phát âm (place of articulation feature) nguyên âm đó, và như vậy, chứa nhiều thông tin nhất về tính đúng sai của âm tố được phát ra cho một âm vị. Tương tự, đối với hai phụ âm liền nhau C-C. Đối với phụ âm trong trường hợp C-V hoặc V-C, các frame nằm ở rìa phụ âm (rìa trước hoặc rìa sau) chứa đựng nhiều thông tin nhất về khả năng phát âm đúng/sai của âm tố. Đối với trường hợp C-V, frame được sử dụng là các frame nằm cuối phụ âm (các frame này được gọi là onset frame hay prevocalic frame). Đối với trường hợp V-C, các frame được dùng

nằm ở rìa trước của phụ âm (các frame này được gọi là offset frame hay postvocalic frame).

Thư viện được dùng để huấn luyện các SVM là SVM-Light Toolkit (Joachims 1999) và hàm nhân (kernel) của SVM là hàm RBF (Radial Basic Function).

Kết quả và đánh giá:

Bảng 4.4 và Bảng 4.5 trình bày độ chính xác của các SVM được huấn luyện để phát hiện các lỗi phát âm cho các âm vị tiếng Anh nêu trong phần 3.1.10. Theo kết quả này tác giả thấy độ chính xác đạt được là tương đối cao và có thể áp dụng được trong thực tiễn.

Bảng 4.4 - Độ chính xác phát hiện lỗi sai khi huấn luyện dữ liệu trên Buckeye

Tên âm vị	SAR	FAR
[ae]	62	56
[P]	83	5
[aa]	85	17
[sh]	78	22
[iy]	74	26

Huấn luyện trên bộ dữ liệu TIMIT(chuẩn):

Qúa trình huấn luyện SVM trên bộ dữ liệu TIMIT cũng giống như quá trình huấn luyện trên bộ dữ liệu Buckeye, Trên bộ dữ liệu TIMIT chỉ chọn ra các file .sx, huấn luyện trên 2310 câu SX và học trên 840 SX còn lại. Kết quả như sau:

Bảng 4.5 - Độ chính xác phát hiện lỗi sai khi huấn luyện dữ liệu trên TIMIT

Tên âm vị	SAR	FAR
[ae]	83	6
[P]	84	7
[aa]	87	15
[sh]	76	26
[iy]	85	12

Nhận thấy quá trình huấn luyện trên bộ dữ liệu TIMIT có độ chính xác cao hơn khi huấn luyện dữ liệu trong bộ Buckeye.

So sánh độ chính xác của thí nghiệm so với các mô hình khác:

Bảng 4.6 – So sánh độ chính xác phát hiện lỗi trên các mô hình khác nhau.

Method	Description	SAR	FAR
Yoon 2009	SVM + landmark	79	8
Witl 2000	Gop Scoring	90	8
This Thesis	SVM	83	13

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong luận văn này, tác giả đã thực hiện các công việc sau:

- Tổng hợp và hệ thống hoá các kiến thức nền tảng làm cơ sở cho việc phát hiện tự động các lỗi phát âm tiếng Anh của người học.
- Tìm hiểu các nghiên cứu liên quan đến lĩnh vực này để có cơ sở chọn được phương pháp xử lý phù hợp áp dụng cho mục đích của luận văn.
- Chọn lựa phương pháp xử lý, tiến hành thử nghiệm phương pháp đã chọn trên hai tập dữ liệu Buckeye và TIMIT, đánh giá kết quả.
- Để minh hoạ cho phương pháp này tác giả đã áp dụng để phát hiện một số lỗi phát âm thường gặp của người Việt. Các lỗi phát âm này được tác giả chọn lựa dựa trên giả thuyết rằng thói quen phát âm mẹ đẻ của các âm vị gần giống các âm vị tương ứng trong tiếng Anh (nhưng không tồn tại trong tiếng Anh) sẽ gây ra các lỗi phát âm. Cần lưu ý rằng phương pháp này có thể áp dụng được với bất kỳ ngôn ngữ L1 (ngôn ngữ mẹ đẻ) nào, chứ không chỉ riêng tiếng Việt. Kết quả thí nghiệm cho thấy mô hình xử lý tiếng nói được sử dụng cho phép phát hiện tự động một số lỗi phát âm tiếng Anh nêu trên với độ chính xác tương đối cao. Khả năng ứng dụng thực tế của phương pháp trong việc xây dựng các công cụ hỗ trợ luyện tập phát âm. Ngoài ra, ta cũng có thể sử dụng phương pháp này cho các ngôn ngữ L2 (ngoại ngữ) khác nhau, chứ không chỉ riêng tiếng Anh. Chỉ cần có kho dữ liệu để huấn luyện các SVM, chúng ta có thể dùng các SVM đã được huấn luyện để phát hiệntự động lỗi phát âm người học phát âm trong L2. Đề tài này chọn tập trung phân tích và thí nghiệm trên tiếng Anh là vì tính phổ dụng của tiếng Anh và sự phong phú của các kho dữ liệu tiếng Anh, cho phép dễ dàng thực hiện được các thí nghiệm. Nếu có dữ liệu đáng tin cậy, chúng ta có thể tìm ra các lỗi phát âm khi người nước ngoài phát âm các âm vị tiếng Việt. Hoặc phát hiện các âm vị do chính người Việt phát âm sai (ví dụ /l/ được phát âm thành /n/). Tuy nhiên, việc xây dựng kho dữ liệu mẫu và gán nhãn âm vị cho từng

đoạn ngữ âm (tương ứng với từng âm vị) trong một câu đòi hỏi rất nhiều thời gian, công sức, kiến thức về ngữ âm học, đặc biệt là kinh nghiệm của các chuyên gia (phonetician) thẩm định các phát âm.

Việc phát hiện tự động các lỗi này sẽ cung cấp thông tin phản hồi cần thiết cho người học tiếng Anh, giúp họ hoàn thiện hơn kỹ năng phát âm. Khi các lỗi này được khắc phục, ta có thể thu thập thông tin và mẫu lỗi khác của người học để huấn luyện các SVM mới , từ đó tạo ra bộ phát hiện lỗi tự động cho các loại lỗi mới. Nhờ vậy, phương pháp này có thể được dùng trong việc huấn luyện kỹ năng phát âm tiếng Anh cho người học ở các trình độ khác nhau.

Vì thời gian thực hiện có hạn, nên luận văn còn một số hạn chế sau:

- Không xây dựng tập mẫu dữ liệu thật gồm các phát âm của người Việt đại diện cho các lỗi sai nêu trong phần 3.2. Việc xây dựng tập dữ liệu mẫu và tiến hành gán nhãn cho từ đoạn ngữ âm (dùng công cụ tương tự Praat) đòi hòi rất nhiều thời gian, và cần đến kiến thức chuyên ngành về ngữ âm học (phonetics) và âm vị học (phonology), nằm ngoài phạm vị của một đề tài thạc sĩ. Trong khi đó, ở Việt Nam chưa có một công trình khoa học nào xây dựng bộ dữ liệu này để có thể dùng được vào trong thử nghiệm phương pháp được chọn lựa trong luận văn.
- Phần thí nghiệm chỉ thực hiện việc thử nghiệm bằng cách giả lặp các lỗi sai trên tập dữ liệu gồm các phát âm do người bản xứ (Mỹ) phát âm, chưa thử nghiệm trên tập dữ liệu là các phát âm thật của người Việt.
- Phần thí nghiệm chỉ thực hiện cho một số ít loại lỗi phát âm. Thực tế người học ngoại ngữ có thể có nhiều lỗi phát âm hơn rất nhiều. Ngoài ra, các lỗi phát âm không chỉ là lỗi phát âm về âm vị (phonemic errors), mà có thể là lỗi phát âm về nhấn (stress errors), lỗi phát âm về ngữ điệu (intonation errors). Đề tài này chỉ tập trung vào một số lượng nhỏ lỗi phát âm thuộc loại lỗi phát âm âm vị.

- Tác giả cũng đã tìm hiểu nhiều bộ dữ liệu khác bao gồm Radio Speech, WS97, NTIMIT (các bộ dữ liệu này có phiên âm ở mức âm vị), nhưng giá thành các bộ dữ liệu này rất đắc, nằm ngoài khả năng tài chính của tác giả. Một số bộ dữ liệu khác đã được tìm hiểu bao gồm: Broastcast New, AVICAR. Tuy nhiên những bộ dữ liệu này không có phiên âm ở mức âm vị và vì vậy không dùng được trong thí nghiệm này. Có thể nói hạn chế này xuất phát từ nguyên nhân khách quan, tuy nhiên phương pháp được sử dụng có thể được thử nghiệm trên các bộ dữ liệu khác trong tương lai khi có thể mua được các bộ dữ liệu nêu trên. Bảng 4.6 tóm tắt thông tin về các bộ dữ liệu tác giả đã tìm hiểu.

Do các hạn chế nêu trên, luận văn có thể được phát triển theo hướng khắc phục các hạn chế này. Cụ thể hơn, nên thực hiện nghiên cứu việc xây dựng một kho dữ liệu của người Việt phát âm tiếng Anh, bao gồm các việc gán nhãn và cho điểm từng đoạn ngữ âm tương ứng với các âm vị trong đoạn âm thanh tiếng nói. Kho dữ liệu này thường được gọi là kho dữ liệu đã được cho điểm (rated corpus). Nghiên cứu theo hướng này đòi hỏi các chuyên gia thẩm định phát âm tham gia vào nhóm nghiên cứu.

- Hướng mở rộng thứ hai là thực hiện việc phát hiện tự động lỗi phát âm cho nhiều loại lỗi phát âm khác nhau, chứ không chỉ hạn chế trong một số lỗi phát âm như trong luận văn. Các lỗi phát âm được mở rộng có thể là lỗi nhấn, lỗi ngữ điệu. Tuy nhiên khi mở rộng tìm các lỗi phát âm liên quan đến ngữ điệu và nhấn câu, đòi hỏi phải hiệu chỉnh phương pháp hoặc đòi hỏi một phương pháp hoàn toàn mới.

Bảng 4.6 – Các kho dữ liệu đã tìm hiểu

Tên	Loại	Bandwidth	Số giờ	Số người phát âm	Mức độ phiên âm	Giá thành
TIMIT	Đọc	7kHz	14	640	Câu – từ – âm vị	250$
NTIMIT	Đọc	3.5 kHz	14	640	Câu – từ – âm vị	500$
WS97	Hội thoại	3.5 kHz	3.5	2283	Câu – từ – âm vị – nhấn – TOBI	Miễn phí
Switchboard 1	Hội thoại	3.5 kHz	349	4188	Câu – từ	3000$
Broastcast New	TV	7 kHz	Lớn	Lớn	Câu	2400$
Radio Speech	Đọc	7 kHz	3.5	7	Câu – từ – TON – BRK – LBL=PHN	1200$
AVICAR	Đọc	7 kHz	50	100	Câu	Miễn phí

Hướng mở rộng thứ ba là hiệu chỉnh phương pháp đã trình bày sao cho tăng độ chính xác của việc phát hiện lỗi (tăng SAR và giảm FAR).

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Âm vị và các hệ thống âm vị tiếng Việt [online] . Available from:<http://ngonngu.net?p=64>.[Accessed 5 Jun 2015]

[2]Boersma, P. and Weenink, D., 2010. Praat: doing phonetics by computer. [online]. Available from: http://www.citeulike.org/group/14233/article/8146799 [Accessed 5 Jun 2015].

[3]Burges, C. J., 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2 (2), 121–167.

[4]Chang, C.-C. and Lin, C.-J., 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2 (3), 27.

[5]Chen, L., Zechner, K., and Xi, X., 2009. Improved pronunciation features for construct-driven assessment of non-native spontaneous speech. *In*: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* [online]. Association for Computational Linguistics, 442–449. Available from: http://dl.acm.org/citation.cfm?id=1620819 [Accessed 5 Jun 2015].

[6]Cucchiarini, C., Van Den Heuvel, H., Sanders, E., and Strik, H., 2011. Error Selection for ASR-Based English Pronunciation Training in'My Pronunciation Coach'. *In*: *INTERSPEECH* [online]. 1165–1168. Available from: http://lands.let.ru.nl/literature/catia.2011.1.pdf [Accessed 5 Jun 2015].

[7]Cucchiarini, C., Strik, H., and Boves, L., 1998a. Automatic pronunciation grading for Dutch. *In*: *Proc. STiLL* [online]. 95–98. Available from: http://hstrik.ruhosting.nl/wordpress/wp-content/uploads/2013/04/a45.pdf [Accessed 4 Jun 2015].

[8]Cucchiarini, C., Strik, H., and Boves, L., 1998b. Quantitative assessment of second language learners' fluency: an automatic approach. *In*: *ICSLP* [online]. Available from: http://www.mirlab.org/conference_papers/International_Conference/ICSLP%201998/PDF/AUTHOR/SL980752.PDF [Accessed 4 Jun 2015].

[9]Cucchiarini, C., De Wet, F., Strik, H., and Boves, L., 1998. Assessment of dutch pronunciation by means of automatic speech recognition technology. *In*: *ICSLP* [online]. Citeseer, 1739–1742. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.7646&rep=rep1&type=pdf [Accessed 4 Jun 2015].

[10]Delmonte, R., 2011. *Exploring Speech Technologies for Language Learning* [online]. INTECH Open Access Publisher. Available from: http://www.intechopen.com/source/pdfs/16006/InTech-Exploring_speech_technologies_for_language_learning.pdf [Accessed 5 Jun 2015].

[11]Van Doremalen, J., Cucchiarini, C., and Strik, H., 2009. Automatic detection of vowel pronunciation errors using multiple information sources. *In*: *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on* [online]. IEEE, 580–585. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5373335 [Accessed 5 Jun 2015].

[12]Eskenazi, M., 1999. Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype. *Language learning & technology*, 2 (2), 62–76.

[13]Eskenazi, M., 2009. An overview of spoken language technology for education. *Speech Communication*, 51 (10), 832–844.

[14]Felps, D., Bortfeld, H., and Gutierrez-Osuna, R., 2009. Foreign accent conversion in computer assisted pronunciation training. *Speech communication*, 51 (10), 920–932.

[15]Franco, H., Abrash, V., Precoda, K., Bratt, H., Rao, R., Butzberger, J., Rossier, R., and Cesari, F., 2000. The SRI EduSpeakTM system: Recognition and pronunciation scoring for language learning. *Proceedings of InSTILL 2000*, 123–128.

[16]Franco, H., Neumeyer, L., Digalakis, V., and Ronen, O., 2000. Combination of machine scores for automatic grading of pronunciation quality. *Speech Communication*, 30 (2), 121–130.

[17] Peter Ladefoged. *A course in phonetics* [online]. Available from:

http://www.phonetics.ucla.edu/course/chapter8/figure8.html [Accessed 5 Jun 2015].

[18]Hasegawa-Johnson, M., Baker, J., Borys, S., Chen, K., Coogan, E., Greenberg, S., Juneja, A., Kirchhoff, K., Livescu, K., Mohan, S., and others, 2005. Landmark-based speech recognition: Report of the 2004 Johns Hopkins summer workshop. *In*: *Proceedings of the... IEEE International Conference on Acoustics, Speech, and Signal Processing/sponsored by the Institute of Electrical and Electronics Engineers Signal Processing Society. ICASSP* [online]. NIH Public Access, 1213. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2638080/ [Accessed 5 Jun 2015].

[19]Hönig, F., Batliner, A., and Nöth, E., 2012. Automatic assessment of non-native prosody annotation, modelling and evaluation. *In*: *Proc. of ISADEPT–*

International Symposium on Automatic Detection of Errors in Pronunciation Training, June [online]. 6–8. Available from: http://www.academia.edu/download/30921816/Hoenig12-AAO.pdf [Accessed 5 Jun 2015].

[20]Hönig, F., Batliner, A., Weilhammer, K., and Nöth, E., 2009. Islands of failure: employing word accent information for pronunciation quality assessment of English L2 learners. *In*: *SLaTE* [online]. Citeseer, 41–44. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.377.5729&rep=rep1&type=pdf [Accessed 5 Jun 2015].

[21]Ito, A., Lim, Y.-L., Suzuki, M., and Makino, S., 2007. Pronunciation error detection for computer-assisted language learning system based on error rule clustering using a decision tree. *Acoustical science and technology*, 28 (2), 131–133.

[22]Joachims, T., 1999. *Making large scale SVM learning practical* [online]. Universität Dortmund. Available from: https://eldorado.tu-dortmund.de/handle/2003/2596 [Accessed 5 Jun 2015].

[23]Jurafsky, D. and Martin, J. H., 2014. *Speech and language processing* [online]. Pearson. Available from: http://www.cs.colorado.edu/~martin/SLP/Updates/1.pdf [Accessed 14 Jun 2015].

[24]Kawai, G. and Hirose, K., 1998. A CALL system using speech recognition to teach the pronunciation of Japanese tokushuhaku. *In*: *STiLL-Speech Technology in Language Learning* [online]. Available from: http://www.isca-speech.org/archive_open/still98/stl8_073.html [Accessed 5 Jun 2015].

[25]Kim, Y., Franco, H., and Neumeyer, L., 1997. Automatic pronunciation scoring of specific phone segments for language instruction. *In*: *Eurospeech* [online]. Available from: http://mc-10136-1356568960.us-west-2.elb.amazonaws.com/sites/default/files/publications/automatic_pronunciation_scoring_of_specific_phone_segments.pdf [Accessed 4 Jun 2015].

[26]Levis, J., 2007. Computer technology in teaching and researching pronunciation. *Annual Review of Applied Linguistics*, 27, 184–202.

[27]Levow, G.-A., 2009. Investigating pitch accent recognition in non-native speech. *In*: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* [online]. Association for Computational Linguistics, 269–272. Available from: http://dl.acm.org/citation.cfm?id=1667666 [Accessed 5 Jun 2015].

[28]Li, H., Huang, S., Wang, S., and Xu, B., 2011. Context-dependent duration modeling with backoff strategy and look-up tables for pronunciation assessment and mispronunciation detection. *In*: *Twelfth Annual Conference of the International Speech Communication Association*.

[29]Liu, L., Mostow, J., and others, 2009. Automated Generation of Example Contexts for Helping Children Learn Vocabulary. [online]. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.8340 [Accessed 5 Jun 2015].

[30]Moustroufas, N. and Digalakis, V., 2007. Automatic pronunciation evaluation of foreign speakers using unknown text. *Computer Speech & Language*, 21 (1), 219–230.

[31] Mark Hasegawa- Johnson, 2005. Phonetic Features, Neural Nets, and Support Vector Machines. Available from: http://www.ifp.uiuc.edu/speech/courses/minicourse/ [Accessed 5 Jun 2015]

[32]Neumeyer, L., Franco, H., Digalakis, V., and Weintraub, M., 2000. Automatic scoring of pronunciation quality. *Speech communication*, 30 (2), 83–93.

[33]Rossetti, A. G., dos Santos Albuquerque, A., Bastos, R. C., and da Silva Filho, V. P., 2011. *Multimedia Authorship Tool for the Teaching of Foreign Languages and Distance Learning in a Multiagent Environment* [online]. INTECH Open Access Publisher. Available from: http://cdn.intechopen.com/pdfs-wm/14521.pdf [Accessed 5 Jun 2015].

[34] Sarah Borys and Mark Hasegawa-Johnson,2009. *SVM-HMM Landmark Based Speech Recognition* [online]. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.3726&rep=rep1&type=pdf [Accessed 5 Jun 2015].

[35]Saz, O. and Eskenazi, M., 2011. Identifying confusable contexts for automatic generation of activities in second language pronunciation training. *In*: *SLaTE* [online]. 121–124. Available from: https://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/max/mainpage_files/Saz-Eskenazi_SLaTE2011.pdf [Accessed 5 Jun 2015].

[36]Saz, O., Lleida, E., and Rodrıguez, W. R., 2009. Acoustic Phonetic Decoding for assessment of mispronunciations in speakers with cognitive disorders. *In*: *Proceedings of the 3rd Advanced Voice Function Assessment International Workshop (AVFA09)* [online]. 129–132. Available from: http://dihana.cps.unizar.es/~oscar/data/2009%20-%20oskarsaz%20-%20AVFA.pdf [Accessed 5 Jun 2015].

[37]Strik, H., Truong, K., De Wet, F., and Cucchiarini, C., 2009. Comparing different approaches for automatic pronunciation error detection. *Speech Communication*, 51 (10), 845–852.

[38]Truong, K., Neri, A., Cucchiarini, C., and Strik, H., 2004. Automatic pronunciation error detection: an acoustic-phonetic approach. *In*:

InSTIL/ICALL Symposium 2004 [online]. Available from: http://www.isca-speech.org/archive_open/icall2004/iic4_032.html [Accessed 5 Jun 2015].

[39]Vapnik, V. N. and Vapnik, V., 1998. *Statistical learning theory* [online]. Wiley New York. Available from: http://ai.ato.ms/MITECS/Articles/pednault.html [Accessed 29 Jun 2015].

[40]Witt, S. M., 1999. Use of speech recognition in computer-assisted language learning. [online]. University of Cambridge. Available from: ftp://svr-www.eng.cam.ac.uk/pub/reports/auto-pdf/witt_thesis.pdf [Accessed 5 Jun 2015].

[41]Witt, S. M. and Young, S. J., 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30 (2), 95–108.

[42]Ye, H. and Young, S., 2005. Improving the speech recognition performance of beginners in spoken conversational interaction for language learning. *In*: *INTERSPEECH* [online]. 289–292. Available from: http://mi.eng.cam.ac.uk/~sjy/papers/yeyo05b.pdf [Accessed 5 Jun 2015].

PHỤ LỤC

7.1 Hệ thống ký hiệu ARPAbet, IPA, và ví dụ về cách sử dụng ARPAbet trong tư điển phát âm của CMU

Arphabet là mã phiên âm được tổ chức Advanced Research Projets Agency (ARPA) phát triển như là một phần dự án Speech Understanding Project (1971-1976) của họ. Arpabet biểu diễn từng âm vị trong hệ thống tiếng Anh kiểu Mỹ bằng một chuỗi riêng biệt các mã ASCII. Arpabet đã được sử dụng trong một vài hệ thống tổng hợp giọng nói như: hệ thống Computalker dành cho máy S-100 (Altair), SAM dành cho máy tính Commodore 65, SAY dành cho hệ điều hành Amiga, TextAssist dành cho SCO1 tổng hợp âm vị IC. Nó cũng được sử dụng trong từ điển phát âm của CMU (CMU Pronouncing Dictionary).

7.1.1 Ký hiệu

Trong hệ thống Arpabet, mỗi âm vị được biểu diễn bằng một hoặc hai ký tự in hoa. Chữ số được dùng như các dấu nhấn và được đặt ở cuối nguyên âm tiết được nhấn. Dấu chấm câu được sử dụng như trong các ngôn ngữ viết, để đại diện cho những thay đổi ngữ điệu ở cuối mệnh đề và câu. Các giá trị nhấn bao gồm:

7.1.1.1 Nhấn (Stress)

Bảng 0.1 - Kí hiệu nhấn âm

Giá trị	Mô tả
0	Không phải âm nhấn
1	Âm nhấn chính
2	Âm nhấn phụ

7.1.2 Nguyên âm (Vowels)

Nguyên âm đơn (Monophthongs)

Bảng 0.2 – Nguyên âm đơn

Arpabet	IPA	Ví dụ
AO	ɔ	off (AO1 F); fall (F AO1 L); frost (F R AO1 S T)
AA	ɑ	father (F AA1 DH ER), cot (K AA1 T)
IY	i	bee (B IY1); she (SH IY1)
UW	u	you (Y UW1); new (N UW1); food (F UW1 D)
EH	ɛ	red (R EH1 D); men (M EH1 N)
IH	ɪ	big (B IH1 G); win (W IH1 N)
UH	ʊ	should (SH UH1 D), could (K UH1 D)
AH	ʌ	but (B AH1 T), sun (S AH1 N)
AX	ə	sofa (S OW1 F AH0), alone (AH0 L OW1 N)
		discus (D IH1 S K AX0 S); phân biệt với discuss (D IH0 S K AH1 S)
AE	æ	at (AE1 T); fast (F AE1 S T)

Nguyên âm đôi (Diphthongs)

Bảng 0.3 – Nguyên âm đôi

Arpabet	IPA	Ví dụ
EY	eɪ	say (S EY1); eight (EY1 T)
AY	aɪ	my (M AY1); why (W AY1); ride (R AY1 D)
OW	oʊ	show (SH OW1); coat (K OW1 T)
AW	aʊ	how (HH AW1); now (N AW1)
OY	ɔ ɪ	boy (B OY1); toy (T OY1)
Arpabet	IPA	Ví dụ
ER	ɝ	her (HH ER0); bird (B ER1 D); hurt (HH ER1 T), nurse (N ER1 S)
AXR	ɚ	father (F AA1 DH ER); coward (K AW1 ER D)
EH R	ɛ r	air (EH1 R); where (W EH1 R); hair (HH EH1 R)
UH R	ʊ r	cure (K Y UH1 R); bureau (B Y UH1 R OW0), detour (D IH0 T UH1 R)
AO R	ɔ r	more (M AO1 R); bored (B AO1 R D); chord (K AO1 R D)
AA R	ɑ r	large (L AA1 R JH); hard (HH AA1 R D)
IH R **IY R**	ɪ r	ear (IY1 R); near (N IH1 R)
AW R	aʊ r	Nguyên âm r-controlled này rất hiếm khi dùng. Trong một vài ngữ điệu địa phương (F L AW1 R; theo giọng địa phương khác F L AW1 ER0)

7.1.3 Phụ âm (Consonants)

Phụ âm dừng (stop)

Để tạo ra âm Stop, ta để cho luồng hơi di chuyển, nhưng trước khi luồng hơi ra khỏi miệng thì ngừng lại một chút, rồi cho luồng hơi di chuyển trở lại.

Bảng 0.4 – Phụ âm dừng (stop)

Arpabet	IPA	Ví dụ
P	p	pay (P EY1)
B	b	buy (B AY1)
T	t	take (T EY1 K)
D	d	day (D EY1)
K	k	key (K IY1)
G	ɡ	go (G OW1)

Phụ âm tắc sát (affricate)

Kết hợp giữa một âm dừng (stop) và một âm sát (fricative).

Bảng 0.5 – Phụ âm tắt sát (affricate)

Arpabet	IPA	Ví dụ
CH	tʃ	chair (CH EH1 R)
JH	dʒ	just (JH AH1 S T); gym (JH IH1 M)

Phụ âm sát (fricative)

Để tạo ra âm sát, ta để luồng hơi bị {chận}{chặn} lại nhưng không hoàn toàn như âm dừng, do đó tạo ra âm thanh giống như tiếng ồn.

Bảng 0.6 – Phụ âm sát (fricative)

Arpabet	IPA	Ví dụ
F	f	for (F AO1 R)
V	v	very (V EH1 R IY0)
TH	θ	thanks (TH AE1 NG K S); Thursday (TH ER1 Z D EY2)
DH	ð	that (DH AE1 T); the (DH AH0); them (DH EH1 M)
S	s	say (S EY1)
Z	z	zoo (Z UW1)
SH	ʃ	show (SH OW1)
ZH	ʒ	measure (M EH1 ZH ER0); pleasure (P L EH1 ZH ER)
HH	h	house (HH AW1 S)

Âm mũi (nasal)

Để tạo ra âm mũi, ta để luồng hơi đi theo đường ống từ miệng lên mũi.

Bảng 0.7 – Âm mũi (nasal)

Arpabet	IPA	Ví dụ
M	m	man (M AE1 N)
EM	m□	keep 'em (K IY1 P EM)
N	n	no (N OW1)
EN	n□	button (B AH1 T EN)
NG	ŋ	sing (S IH1 NG)
ENG	ŋ□	Washington (W AO1 SH ENG T EN)

Âm nước (liquid)

Bảng 0.8 – Âm nước (liquid)

Arpabet	IPA	Ví dụ
L	ɬ	late (L EY1 T)
EL	□□	bottle (B AO1 DX EL)
R	r hoặc ɹ	run (R AH1 N)
DX	ɾ	wetter (W EH1 DX AXR)
NX	□̃	wintergreen (W IY2 NX AXR G R IY1 N)

Bán nguyên âm(Semivowels)

Bảng 0.9 – Bán nguyên âm (semivowel)

Arpabet	IPA	Ví dụ
Y	j	yes (Y EH1 S)
W	w	way (W EY1)
Q	ʔ	glottal stop (uh-oh - ʔ ʌ ʔ oʊ)
(missing)	hw hoặc ʍ	"when" etc. in some dialects