TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**TÌM HIỂU VÀ XÂY DỰNG ENGINE TỔNG HỢP TIẾNG NÓI TỪ VĂN BẢN TRÊN ANDROID**

Sinh viên thực hiện: **Lê Trọng Thịnh**

Lớp CNTT 4 – K54

Giáo viên hướng dẫn: ThS. **Lê Tấn Hùng**

HÀ NỘI 5-2014

**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

1. Thông tin về sinh viên

Họ và tên sinh viên: . . . .Lê Trọng Thịnh . . . . . . . . . . . . . . . . . . . . . . .

Điện thoại liên lạc: 01652 639 102 Email: VictoriaVinky@gmail.com

Lớp: CNTT 4 - K54 Hệ đào tạo: Kỹ sư

Đồ án tốt nghiệp được thực hiện tại:

Thời gian làm ĐATN: Từ ngày 13/01/2014 đến 30/05/2014

2. Mục đích nội dung của ĐATN

Tìm hiểu phương pháp tổng hợp tiếng nói và ứng dụng vào xây dựng engine TTS trên Android

3. Các nhiệm vụ cụ thể của ĐATN

* Tìm hiểu lý thuyết tiếng nói và các đặc điểm ngữ âm của Tiếng Việt
* Tìm hiểu lý thuyết tổng hợp tiếng nói, phương pháp tổng hợp tiếng nói và mô hình tổng hợp tiếng nói từ văn bản
* Tìm hiểu phương pháp xây dựng bộ tổng hợp Tiếng Việt bằng phương pháp ghép nối
* Tìm hiểu cơ chế hoạt động và quy trình xây dựng một Engine Text-To-Speech
* Xây dựng một Engine Text-To-Speech sử dụng modul tổng hợp Tiếng Việt bằng phương pháp ghép nối
* Thử nghiệm Engine Text-To-Speech với các ứng dụng sử dụng tính Text-To-Speech của Android. Đánh giá Engine về mặt tính năng và hiệu năng
* Kết luận và định hướng phát triển

4. Lời cam đoan của sinh viên:

Tôi – *Lê Trọng Thịnh* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Lê Tấn Hùng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 20 tháng 5 năm 2014*  Tác giả ĐATN  *Lê Trọng Thịnh* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Giáo viên hướng dẫn  *ThS. Lê Tấn Hùng* |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Mục tiêu của khuôn khổ đồ án là xây dựng được một Engine Text-To-Speech chạy trên nền tảng hệ điều hành Android với ngôn ngữ là Tiếng Việt. Trong quá trình nghiên cứu và làm việc tại Viện Nghiên cứu Quốc Tế MICA đã thu được kết quả là Engine Text-To-Speech sử dụng bộ tổng hợp Tiếng Việt do MICA nghiên cứu và phát triển. Qua quá trình làm việc và nghiên cứu tại Viện MICA và sự hướng dẫn của thầy ThS. Lê Tấn Hùng, TS. Trần Đỗ Đạt cùng TS. Mạc Đăng Khoa, kết quả thu được khá khả quan. Và báo cáo đồ án này sẽ trình bày các vấn đề đã tìm hiểu và thực hiện được trong suốt quá trình đồ án.

Báo cáo Đồ án tốt nghiệp sẽ trình bày về các vấn đề sau:

* Các khái niệm cơ bản về tiếng nói và các đặc tính âm học của tiếng nói
* Biểu diễn tiếng nói dưới dạng tín hiệu số
* Giới thiệu 3 phương pháp tổng hợp tiếng nói đang được sử dụng hiện nay
* Nêu mô hình tổng hợp tiếng nói từ văn bản
* Quy trình xây dựng bộ tổng hợp Tiếng Việt hiện đang được nghiên cứu và phát triển tại Viện nghiên cứu quốc tế MICA
* Trình bày về xây dựng Engine Text-To-Speech trên Android
* Thực nghiệm xây dựng một Engine Text-To-Speech sử dụng bộ tổng hợp Tiếng Việt do MICA phát triển
* Triển khai và thử nghiệm Engine Text-To-Speech được xây dựng
* Đánh giá và kết luận, đưa ra định hướng

# MỤC LỤC

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3](#_Toc389132335)

[MỤC LỤC 4](#_Toc389132336)

[DANH MỤC CÁC BẢNG 6](#_Toc389132337)

[DANH MỤC CÁC HÌNH 7](#_Toc389132338)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 8](#_Toc389132339)

[LỜI CẢM ƠN 9](#_Toc389132340)

[LỜI NÓI ĐẦU 10](#_Toc389132341)

[CHƯƠNG 1. TIẾNG NÓI VÀ TỔNG HỢP TIẾNG NÓI 12](#_Toc389132342)

[1.1. Giới thiệu chương 12](#_Toc389132343)

[1.2. Tiếng nói và các đặc tính của tiếng nói 12](#_Toc389132344)

[1.2.1. Tiếng nói và cơ chế phát âm 12](#_Toc389132345)

[1.2.2. Biểu diễn tín hiệu tiếng nói 13](#_Toc389132346)

[1.2.3. Đặc tính âm học của tiếng nói 14](#_Toc389132347)

[1.3. Các phương pháp tổng hợp tiếng nói 16](#_Toc389132348)

[1.3.1. Phương pháp mô phỏng hệ thống phát âm 16](#_Toc389132349)

[1.3.2. Phương pháp tổng hợp Formant 16](#_Toc389132350)

[1.3.3. Phương pháp ghép nối 18](#_Toc389132351)

[1.3.4. So sánh các phương pháp tổng hợp tiếng nói 18](#_Toc389132352)

[1.4. Mô hình tổng hợp tiếng nói từ văn bản 18](#_Toc389132353)

[1.4.1. Tổng hợp mức cao 19](#_Toc389132354)

[1.4.2. Tổng hợp mức thấp 19](#_Toc389132355)

[1.5. Kết chương 19](#_Toc389132356)

[CHƯƠNG 2. TỔNG HỢP TIẾNG VIỆT BẰNG PHƯƠNG PHÁP GHÉP NỐI CÁC ĐƠN VỊ ÂM KHÔNG ĐỒNG NHẤT 20](#_Toc389132357)

[2.1. Giới thiệu chương 20](#_Toc389132358)

[2.2. Tiếng Việt và đặc điểm ngữ âm của Tiếng Việt 20](#_Toc389132359)

[2.3. Tổng hợp Tiếng Việt bằng phương pháp ghép nối các đơn vị âm không đồng nhất 22](#_Toc389132360)

[2.3.1. Cơ sở xây dựng 22](#_Toc389132361)

[2.3.2. Lựa chọn loại đơn vị âm 23](#_Toc389132362)

[2.3.3. Xây dựng cơ sở dữ liệu 25](#_Toc389132363)

[2.3.4. Tìm kiếm đơn vị âm tối ưu 26](#_Toc389132364)

[2.3.5. Ghép nối các đơn vị âm và biến đổi tần số cơ bản 30](#_Toc389132365)

[2.4. Kết chương 34](#_Toc389132366)

[CHƯƠNG 3. XÂY DỰNG ENGINE TEXT-TO-SPEECH TRÊN NỀN TẢNG ANDROID 35](#_Toc389132367)

[3.1. Giới thiệu chương 35](#_Toc389132368)

[3.2. Hệ điều hành Android 35](#_Toc389132369)

[3.2.1. Giới thiệu về hệ điều hành Android 35](#_Toc389132370)

[3.2.2. Linux – Nền tảng hệ điều hành Android 36](#_Toc389132371)

[3.2.3. Kiến trúc hệ điều hành Android 37](#_Toc389132372)

[3.3. Engine Text-To-Speech trên Android 39](#_Toc389132373)

[3.3.1. Giới thiệu về tính năng Text-To-Speech và Engine Text-To-Speech 39](#_Toc389132374)

[3.3.2. Cơ chế hoạt động của Engine Text-To-Speech 40](#_Toc389132375)

[3.3.3. Xây dựng Engine Text-To-Speech 41](#_Toc389132376)

[3.4. Xây dựng Engine Text-To-Speech sử dụng bộ Tổng hợp Tiếng Việt bằng phương pháp ghép nối 43](#_Toc389132377)

[3.4.1. Bộ Tổng hợp Tiếng Việt bằng phương pháp ghép nối 43](#_Toc389132378)

[3.4.2. Xây dựng lớp xử lý văn bản đầu vào 43](#_Toc389132379)

[3.4.3. Tích hợp bộ Tổng hợp Tiếng Việt 44](#_Toc389132380)

[3.4.4. Xây dựng và đóng gói Engine Tổng hợp Tiếng Việt 48](#_Toc389132381)

[CHƯƠNG 4. TRIỂN KHAI VÀ ĐÁNH GIÁ ENGINE VIVA TTS TRÊN SMARTPHONE ANDROID 51](#_Toc389132382)

[4.1. Giới thiệu chương 51](#_Toc389132383)

[4.2. Triển khai Engine tổng hợp Tiếng Việt trên Smartphone Android 51](#_Toc389132384)

[4.2.1. Môi trường, thiết bị thử nghiệm 51](#_Toc389132385)

[4.2.2. Cài đặt và thiết lập Engine Text-To-Speech 51](#_Toc389132386)

[4.3. Thử nghiệm Engine tổng hợp Tiếng Việt bằng tính năng Talkback 53](#_Toc389132387)

[4.3.1. Ứng dụng Google TalkBack 53](#_Toc389132388)

[4.3.2. Các trường hợp thử nghiệm 55](#_Toc389132389)

[4.3.3. Kết quả thử nghiệm 56](#_Toc389132390)

[4.4. Đánh giá 57](#_Toc389132391)

[4.4.1. Tính năng 57](#_Toc389132392)

[4.4.2. Hiệu năng 58](#_Toc389132393)

[4.5. Kết chương 58](#_Toc389132394)

[CHƯƠNG 5. KẾT LUẬN 59](#_Toc389132395)

[5.1. Kết luận chung 59](#_Toc389132396)

[5.2. Những khó khăn gặp phải 59](#_Toc389132397)

[5.3. Định hướng phát triển 59](#_Toc389132398)

[TÀI LIỆU THAM KHẢO 60](#_Toc389132399)

# DANH MỤC CÁC BẢNG

[Bảng 1. Bảng giá trị tần số cơ bản 16](#_Toc389132400)

[Bảng 2. Số lượng các loại đơn vị âm trong Tiếng Việt 24](#_Toc389132401)

[Bảng 3. Ưu, nhược điểm của từng loại đơn vị âm sử dụng trong bộ tổng hợp Tiếng Việt 25](#_Toc389132402)

[Bảng 4. API bộ Tổng hợp Tiếng Việt 43](#_Toc389132403)

[Bảng 5. Các trường hợp thử nghiệm Engine Viva TTS 56](#_Toc389132404)

[Bảng 6. Kết quả thử nghiệm Engine Viva TTS 57](#_Toc389132405)

# DANH MỤC CÁC HÌNH

[Hình 1.1. Bộ máy phát âm của con người 12](#_Toc389132406)

[Hình 1.2. Phương pháp biểu diễn tín hiệu tiếng nói theo dạng số 13](#_Toc389132407)

[Hình 1.3. Cấu trúc cơ bản của bộ tổng hợp formant nối tiếp 17](#_Toc389132408)

[Hình 1.4. Cấu trúc cơ bản của một bộ tổng hợp formant song song 17](#_Toc389132409)

[Hình 1.5. Mô hình tổng hợp tiếng nói từ văn bản 18](#_Toc389132410)

[Hình 2.1. Mô hình một âm tiết Tiếng Việt 20](#_Toc389132411)

[Hình 2.2. Biểu đồ thanh điệu Tiếng Việt 22](#_Toc389132412)

[Hình 2.3. Hàm chi phí giữa các đơn vị âm 26](#_Toc389132413)

[Hình 2.4. Sơ đồ tìm kiếm đơn vị âm 27](#_Toc389132414)

[Hình 2.5. So sánh đích ti và đơn vị âm ui 28](#_Toc389132415)

[Hình 2.6. So sánh sự khác nhau về ngữ cảnh của 2 đơn vị âm ui và ui-1 29](#_Toc389132416)

[Hình 2.7. So sánh sự khác nhau về phổ của hai đơn vị âm ui và ui-1 29](#_Toc389132417)

[Hình 2.8. Ví dụ về ghép nối 2 đơn vị âm 32](#_Toc389132418)

[Hình 2.9. Cộng xếp chồng các tín hiệu thành phần 32](#_Toc389132419)

[Hình 2.10. Quá trình biến đổi tần số cơ bản theo thời gian 33](#_Toc389132420)

[Hình 2.11. Biểu đồ thanh điệu Tiếng Việt 33](#_Toc389132421)

[Hình 3.1. Liên minh di động mở 35](#_Toc389132422)

[Hình 3.2. Chiếc điện thoại đầu tiên chạy hệ điều hành Android: HTC Dream 36](#_Toc389132423)

[Hình 3.3. Kiến trúc hệ điều hành Android 37](#_Toc389132424)

[Hình 3.4. Sao chép cơ sở dữ liệu cho bộ Tổng hợp Tiếng Việt 46](#_Toc389132425)

[Hình 3.5. Sao chép thư viện bộ Tổng hợp Tiếng Việt 47](#_Toc389132426)

[Hình 4.1. Màn hình yêu cầu quyền truy cập của Viva TTS 52](#_Toc389132427)

[Hình 4.2. Thiết lập sử dụng Engine Viva TTS (1) 52](#_Toc389132428)

[Hình 4.3. Thiết lập sử dụng Engine Viva TTS (2) 53](#_Toc389132429)

[Hình 4.4. Thiết lập sử dụng Engine Viva TTS (3) 53](#_Toc389132430)

[Hình 4.5. Bật TalkBack 54](#_Toc389132431)

[Hình 4.6. TalkBack Settings và tính năng Khám phá bằng cách chạm 55](#_Toc389132432)

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt và thuật ngữ** | **Tiếng Anh** | **Nghĩa** |
| MICA | Multimedia, Infomation, Communication and Application | Viện Nghiên cứu Quốc tế về Thông tin đa phương tiện, Truyền thông và Ứng dụng |
| Pitch |  | Tần số cơ bản của âm thanh |
| PSOLA | Pitch Synchronous Overlap Add | Kỹ thuật chồng đồng bộ cao độ tần số cơ bản |
| TD-PSOLA | Time-domain pitch synchronous overlap add | Kỹ thuật chồng đồng bộ cao độ tần số cơ bản trên miền thời gian |
| FD-PSOLA | Frequent-domain pitch synchronous overlap add | Kỹ thuật chồng đồng bộ cao độ tần số cơ bản trên miền tần số |
| LP-PSOLA | Linear-prediction pitch synchronous overlap add | Kỹ thuật chồng đồng bộ cao độ tần số cơ bản với dự đoán tuyến tính |
| TTS | Text To Speech | Tổng hợp tiếng nói từ văn bản |
| MFCC | Mel-Frequency Cepstral Coeficients |  |

# LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn tới thầy, cô giáo và cán bộ giảng dạy tại trường Đại học Bách Khoa nói chung và các thầy cô trong Viện Công nghệ Thông tin và Truyền Thông đã tận tâm dạy dỗ, truyền đạt kiến thức chúng em trong suốt những năm học tại trường. Với các bài học và sự chỉ dạy quý giá từ các thầy cô mà chúng em, những sinh viên ngành Công nghệ thông tin có được những kiến thức chuyên ngành để có thể tiếp tục chặng đường học tập, vận dụng và sáng tạo ra những sản phẩm công nghệ có ích và thiết thực trong cuộc sống.

Em xin bày tỏ lòng biết ơn sâu sắc tới thầy giáo **ThS. Lê Tấn Hùng** đã giúp đỡ trong việc chọn đề tài cũng như tạo điều kiện thuận lợi cho em trong suốt quá trình làm đồ án tốt nghiệp, giúp em hoàn thành đồ án tốt nghiệp đúng hạn.

Em xin chân thành cảm ơn **TS. Trần Đỗ Đạt** và **TS. Mạc Đăng Khoa** đã tạo điều kiện cho phép em được tham gia nghiên cứu trên viện MICA và hướng dẫn em hoàn thành đồ án tốt nghiệp.

Tôi cũng xin gửi lời cảm ơn đến các bạn trong nhóm nghiên cứu đã giúp đỡ trong quá trình hoàn thiện đồ án.

*Hà Nội, ngày 20 tháng 5 năm 2014*

Tác giả ĐATN

*Lê Trọng Thịnh*

# LỜI NÓI ĐẦU

Trong thời đại kỷ nguyên số, các thiết bị điện tử cùng hàng loạt các công cụ trợ giúp, cuộc sống con người ngày càng trở nên văn minh và hiện đại hơn. Con người không ngừng tìm tòi và sáng tạo nên các thiết bị thông minh, các hệ thống tự động phục vụ trong cuộc sống hàng ngày có khả năng hiểu các yêu cầu của con người. Cách thức tương tác giữa người và máy càng trở nên dễ dàng và thân thiện. Con người đang chuyển dần phương thức tương tác người – máy bằng các thiết bị phần cứng như bàn phím, chuột, màn hình… sang các phương thức tương tác bằng cử chỉ, giọng nói. Và một trong các hướng nghiên cứu sử dụng tiếng nói trong tương tác người – máy ngày càng được nhiều người quan tâm. Những nghiên cứu này có liên quan trực tiếp tới các kết quả của chuyên ngành xử lý tiếng nói.

Tổng hợp tiếng nói là một phần trong xử lý tiếng nói đang được nghiên cứu khá rộng rãi trên thế giới và đã có những kết quả đáng khả quan. Viện Nghiên cứu Quốc tế MICA đã và đang định hướng tổng hợp tiếng nói là một trong những nghiên cứu lâu dài của viện. Với các bài báo nghiên cứu cùng các kết quả thực nghiệm gần đây, MICA đã xây dựng nên một thư viện tổng hợp Tiếng Việt có chất lượng tốt và gần giống với tiếng nói tự nhiên. Với ý định tìm hiểu về các phương pháp tổng hợp tiếng nói nói chung và Tiếng Việt nói chung để phát triển các ứng dụng và dịch vụ tổng hợp tiếng trên các thiết bị di động, Viện Nghiên cứu Quốc Tế MICA là địa điểm phù hợp cho công việc nghiên cứu tổng hợp tiếng cũng như phát triển các ứng dụng.

Với đề tài đồ án tốt nghiệp ***“Tìm hiểu và xây dựng engine tổng hợp tiếng nói từ văn bản trên android”***, sự giới thiệu của **ThS. Lê Tấn Hùng** tới Viện Nghiên cứu Quốc tế MICA, được sự giúp đỡ nhiệt tình và tận tâm của **TS. Trần Đỗ Đạt** và **TS. Mạc Đăng Khoa** cùng với nhóm sinh viên nghiên cứu trên viện MICA về xử lý tiếng nói, tôi đã tìm hiểu được phần nào về tổng hợp tiếng nói và ứng dụng để xây dựng nên một Engine Text-To-Speech chạy trên nền tảng Android.

Với nội dung đồ án cùng quá trình nghiên cứu, làm việc trên viện MICA, bài báo cáo đồ án tốt nghiệp được chia làm 5 chương với các phần cụ thể như sau:

* **Chương 1. Tiếng nói và tổng hợp tiếng nói.** Chương trình bày lý thuyết về tiếng nói, các đặc tính của tiếng nói và biểu diễn tiếng nói dưới dạng tín hiệu số. Và giới thiệu qua 3 phương pháp tổng hợp tiếng nói phổ biến cùng với mô hình tổng hợp tiếng nói từ văn bản.
* **Chương 2. Tổng hợp tiếng việt bằng phương pháp ghép nối các đơn vị âm không đồng nhất.** Chương này trình bày các đặc điểm ngữ âm Tiếng Việt cùng với quy trình xây dựng bộ tổng hợp Tiếng Việt mà MICA đã thực hiện.
* **Chương 3. Xây dựng Engine Text-To-Speech trên nền tảng Android** sẽ giới thiệu về hệ điều hành Android và tính năng Text-To-Speech. Chương trình bày cơ chế hoạt động của tính năng Text-To-Speech trên Android và cách thức xây dựng một engine Text-To-Speech. Và thực nghiệm xây dựng một Engine Text-To-Speech sử dụng bộ Tổng hợp Tiếng Việt.
* **Chương 4. Triển khai và đánh giá Engine tổng hợp Tiếng Việt trên Smartphone Android** sẽ trình bày kết quả thực nghiệm của Engine Text-To-Speech được xây dựng ở chương 2 với ứng dựng Talkback. Đánh giá về tính năng và hiệu năng của Engine sau khi thử nghiệm.
* **Chương 5. Kết luận.** Trình bày kết quả và định hướng phát triển sản phẩm.

Cụ thể và chi tiết từng chương, tôi xin được phép trình bày ở các phần tiếp theo của báo cáo.

# TIẾNG NÓI VÀ TỔNG HỢP TIẾNG NÓI

## Giới thiệu chương

Trong chương này, bao cáo trình bày về các vấn đề sau:

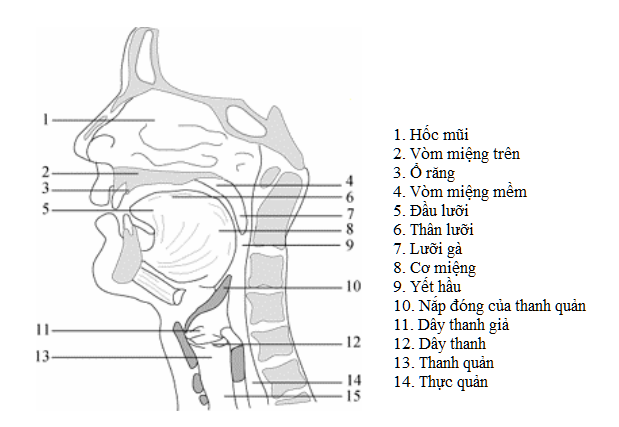
* Lý thuyết về tiếng nói và các đặc tính âm học của tiếng nói
* Biểu diễn tiếng nói dưới dạng tín hiệu số
* Giới thiệu 3 phương pháp tổng hợp tiếng nói
* Mô hình tổng hợp tiếng nói từ văn bản

## Tiếng nói và các đặc tính của tiếng nói

### Tiếng nói và cơ chế phát âm

Tiếng nói là một chuỗi các thanh được phát ra từ bộ máy phát âm của con người và là một phương tiện trao đổi thông tin quan trọng nhất của con người. Tiếng nói có những điểm giống với âm thanh khác trong thế giới tự nhiên nhưng tinh vi và phức tạp hơn nhiều so với các loại âm thanh đó. Tiếng nói được phân biệt với các âm thanh đó bởi các đặc tính âm học có nguồn gốc từ cơ chế tạo tiếng nói.

Tiếng nói là một quá trình đặc biệt tạo ra âm thanh của con người từ bộ máy phát âm, trong đó các dây thanh là những nguồn âm thanh ban đầu. Bộ máy phát âm của con người bao gồm các thành phần riêng rẽ như phổi, khí quản, thanh quản và các cơ quan phát âm (bao gồm lưỡi, vòm miệng, má, môi…)



Hình 1.1. Bộ máy phát âm của con người

Cơ chế để tạo ra tiếng nói có thể được chia làm ba phần:

* Phổi: tạo ra một luồng không khí đầy đủ có áp suất để làm rung dây thanh. Có thể nói áp suất từ luồng không khí này là nguyên liệu của giọng nói.
* Khi các luồng không khí từ phổi đi qua, dây thanh âm đóng vai trò như một van rung băm nhỏ các luồng không khí này và đưa vào các xung nghe được tạo ra từ nguồn âm thanh quản. Các bắp thịt của thanh quản điều chỉnh độ dài và độ căng của dây thanh để “tinh chỉnh” cao độ và giai điệu của âm.
* Các cơ quan phát âm lọc các âm thanh phát ra từ thanh quản và ở mức độ nào đó có thể tương tác với các luồng không khí trong thanh quản để tăng cường hay làm suy yếu nó như một nguồn âm thanh và tạo ra tiếng nói.

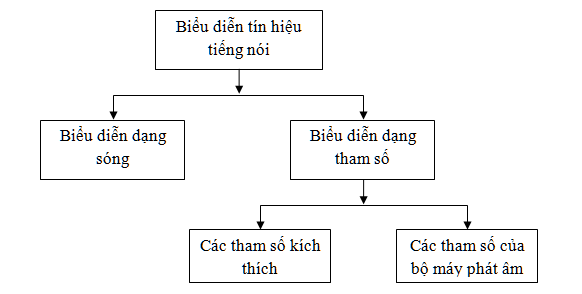
Các dây thanh âm kết hợp với cơ quan phát âm có khả năng tạo ra mảng âm thanh phức tạp. Những giai điệu của tiếng nói có thể được điều biến cho phù hợp với những cảm xúc như giận dữ, ngạc nhiên hay hạnh phúc.

### Biểu diễn tín hiệu tiếng nói

Tín hiệu tiếng nói là tín hiệu tương tự. Do đó khi biểu diễn tín hiệu tiếng nói trong môi trường tính toán của tín hiệu số, việc biểu diễn và lưu trữ sao cho không bị mất mát thông tin là vấn đề hết sức quan trọng trong các hệ thống thông tin có sử dụng tín hiệu tiếng nói.

Biểu diễn tín hiệu tiếng nói theo dạng số chịu ảnh hưởng của lý thuyết lấy mẫu, theo đó các trạng thái của tín hiệu có dải tần giới hạn có thể được biểu diễn dưới dạng các mẫu lấy tuần hoàn theo một chu kỳ cố định gọi là chu kỳ lấy mẫu. Việc lấy mẫu này sẽ cung cấp cho hệ thống những mẫu tín hiệu với tỷ lệ đủ lớn để xử lý.

Có hai phương pháp chính biểu diễn tín hiệu tiếng nói theo dạng số:



Hình 1.2. Phương pháp biểu diễn tín hiệu tiếng nói theo dạng số

Vì tiếng nói là tín hiệu liên tục nên để áp dụng các phương pháp xử lý tín hiệu thì tiếng nói phải được biểu diễn dưới dạng rời rạc. Quá trình rời rạc hóa tín hiệu tiếng nói bao gồm các bước sau:

* Lấy mẫu tín hiệu tiếng nói với tần số lấy mẫu *f0*
* Lượng tử hóa các mẫu với các bước lượng tử q
* Mã hóa và nén tín hiệu

### Đặc tính âm học của tiếng nói

#### Âm hữu thanh và âm vô thanh

##### Âm hữu thanh

Âm hữu thanh được tạo ra khi dây thanh âm căng lên và rung khi áp suất không khí tăng lên, làm cho thanh môn mở ra rồi đóng lại khi luồng không khí đi qua. Cơ quan phát âm hoạt động giống như hộp cộng hưởng khuếch đại những thành phần này và làm suy giảm những thành phần khác để tạo ra âm hữu thanh. Mức độ rung của dây thanh âm tùy thuộc vào áp suất không khí ở phổi và sức căng của dây thanh âm. Người ta có thể điểu khiển hai yếu tố trên để thay đổi tần số cơ bản của âm thanh (được gọi là pitch).

##### Âm vô thanh

Âm vô thanh được tạo ra khi dây thanh âm không rung. Âm vô thanh có hai loại cơ bản:

* Âm xát (ví dụ như âm ‘s’, ‘x’): được tạo ra khi một số điểm trên cơ quan phát âm bị co lại khi luồng không khí đi ngang qua nó. Luồng không khí đó sẽ chuyển thành chuyển động hỗn loạn tạo nên nhiễu ngẫu nhiên. Thông thường điểm co thắt xảy ra gần miệng nên sự cộng hưởng của tuyến âm ít ảnh hưởng đến đặc tính của âm xát được tạo ra.
* Âm tắc (ví dụ như âm ‘p’): được tạo ra khi tuyến âm đóng tại một số điểm làm cho áp suất không khí tăng lên và sau đó được giải phóng đột ngột. Sự giải phóng đột ngột này tạo ra kích thích nhất thời của tuyến âm. Sự kích thích này có thể xảy ra với sự cộng hưởng và không cộng hưởng của dây thanh tương ứng với âm tắc hữu thanh hoặc vô thanh.

#### Âm vị

Âm vị là đơn vị nhỏ nhất của ngôn ngữ. Tùy theo từng ngôn ngữ mà số lượng các âm vị có nhiều hay ít. Các âm vị được chia thành hai loại: nguyên âm và phụ âm.

##### Nguyên âm

Nguyên âm là âm hữu thanh được tạo ra bằng sự cộng hưởng của dây thanh khi dòng khí được thanh môn đẩy lên. Khoang miệng được tạo thành nhiều hình dạng nhất định tạo thành các nguyên âm khác nhau. Số lượng các nguyên âm phụ thuộc vào từng ngôn ngữ nhất định.

##### Phụ âm

Phụ âm được tạo ra bởi các dòng khí hỗn loạn được phát ra gần những điểm co thắt của đường dẫn âm thanh do các phát âm tạo thành. Phụ âm có đặc tính hữu thanh hay vô thanh tùy thuộc vào dây thanh có dao động để tạo nên cộng hưởng hay không. Dòng không khi tại chỗ đóng của vòm miệng tạo ra âm tắc. Âm xát được phát ra từ chỗ co thắt lớn nhất.

#### Âm tiết

Chuỗi lời nói mà con người phát ra gồm nhiều khúc đoạn dài ngắn khác nhau. Đơn vị phát âm ngắn nhất là âm tiết.

Về phương diện phát âm, âm tiết có tính chất toàn vẹn, không thể phân chia được bởi nó được phát âm bằng một đợt căng của cơ thịt của bộ máy phát âm. Khi phát một âm tiết, các cơ thịt của bộ máy phát âm đều phải trải qua ba giai đoạn: tăng cường độ căng, đỉnh điểm căng và giảm độ căng.

Dựa vào cách kết thúc, các âm tiếtđược chia thành hai loại lớn: mở và khép. Trong mỗi loại có hai loại nhỏ. Như vậy, có bốn loại âm tiết như sau:

* Âm tiết nửa khép: những âm tiết được kết thúc bằng một phụ âm vang
* Âm tiết khép: những âm tiết được kết thúc bằng một phụ âm không vang
* Âm tiết nửa mở: những âm tiết được kết thúc bằng một bán nguyên âm
* Âm tiết mở: những âm tiết được kết thúc bằng cách giữ nguyên âm của sắc của nguyên âm ở đỉnh âm tiết

#### Các đặc tính khác

##### Tỷ suất thời gian

Trong khi nói chuyện, khoảng thời gian nói và khoảng thời gian nghỉ xen kẽ nhau. Tỷ lệ % thời gian nói trên tổng số thời gian nói và nghỉ được gọi là tỷ suất thời gian. Giá trị này biến đổi tùy thuộc vào tốc độ nói và từ đó ta có thể phân loại thành nói nhanh, nói chậm hay nói bình thường

##### Tần số cơ bản

Dạng sóng của tiếng nói gồm hai phần: phần gần giống nhiễu (biên độ biến đổi ngẫu nhiên) và phần có tính chu kỳ (tín hiệu lặp lại gần như tuần hoàn). Phần tín hiệu có tính chu kỳ chứa các thành phần tần số có dạng điều hòa. Tần số thấp nhất chính là tần số cơ bản và cũng chính là tần số dao động của dây thanh. Đối với những người nói khác nhau, tần số cơ bản cũng khác nhau.

|  |  |
| --- | --- |
| **Người nói** | **Giá trị tần số cơ bản** |
| Nam giới | 80 – 200 Hz |
| Nữ giới | 150 – 450 Hz |
| Trẻ em | 200 – 600 Hz |

Bảng 1. Bảng giá trị tần số cơ bản

##### Formant

Với phổ tín hiệu tiếng nói, mỗi đỉnh có biên độ lớn nhất xét trong một khoảng nào đó tương ứng với một formant. Ngoài tần số, formant còn được xác định bởi biên độ và giải thông. Về mặt vật lý, formant tương ứng với các tần số cộng hưởng của tuyến âm. Trong xử lý tiếng nói và nhất là trong tổng hợp tiếng nói, để mô phỏng lại tuyến âm, người ta phải xác định được các tham số formant đối với từng loại âm vị, do đó việc đánh giá, ước lượng các formant có ý nghĩa quan trọng.

Tần số formant biến đổi trong một khoảng rộng phụ thuộc vảo giới tính của người nói và phụ thuộc vào các dạng âm vị (nguyên âm và phụ âm) tương ứng với formant đó. Đồng thời, formant còn phụ thuộc vào các âm vị trước và sau. Về cấu trúc tự nhiên, tần số formant có liên hệ chặt chẽ với hình dạng và kích thước tuyến âm. Thông thường phổ của tín hiệu tiếng nói có khoảng 5 formant nhưng chỉ có 3 formant đầu tiên ảnh hưởng quan trọng đến các đặc tính của âm vị, các formant còn lại cũng có ảnh hưởng song rất ít.

Thêm vào đó, tần số formant đặc trưng cho các nguyên âm biến đổi tùy thuộc vào người nói trong điều kiện phát âm nhất định. Mặc dù phạm vi của các tần số formant tương ứng với mỗi nguyên âm có thể trùm lên nhay nhưng vị trí giữa các formant là không đổi vì sự xê dịch của các formant là song song.

## Các phương pháp tổng hợp tiếng nói

### Phương pháp mô phỏng hệ thống phát âm

Phương pháp mô phỏng hệ thống phát âm (articulatory synthesis) cố gắng mô phỏng hệ thống phát âm của con người một cách hoàn hảo nhất, do đó có thể đạt tới chất lượng cao trong tổng hợp tiếng nói. Nhưng cũng chính vì vậy mà phương pháp này khó có thể thực hiện được vì việc mô phỏng hệ thống phát âm của con người rất khó thực hiện.

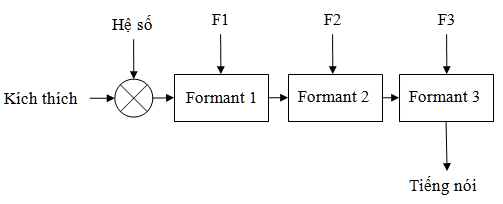
Sau khi phương pháp tổng hợp Formant ra đời thì phương pháp mô phỏng hệ thống phát âm ít khi được sử dụng trong các hệ thống. Nhưng từ khi có sự xuất hiện của máy tính thì nó lại được phát triển.

### Phương pháp tổng hợp Formant

Phương pháp tổng hợp Formant yêu cầu phải tổng hợp được tối thiểu 3 formant để hiểu được tiếng nói. Và để có được tiếng nói chất lượng thì cần tới 5 formant. Tiếng nói được tạo ra từ các bộ tổng hợp formant với thành phần chính là các bộ cộng hưởng. Tùy theo cách bố tí các bộ cộng hưởng mà ta có bộ tổng hợp formant là nối tiếp hay song song.

#### Bộ tổng hợp formant nối tiếp

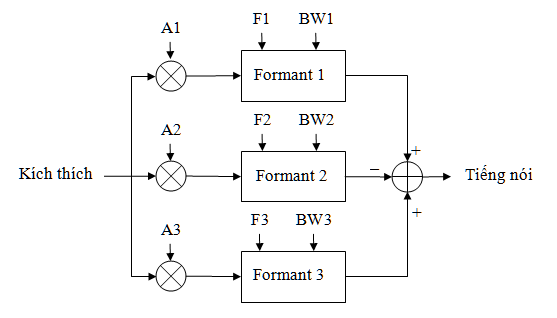
Bộ tổng hợp formant nối tiếp là một bộ tổng hợp formant có các tầng nối tiếp, đầu ra của bộ cộng hưởng này là đầu vào của bộ cộng hưởng kia.



Hình 1.3. Cấu trúc cơ bản của bộ tổng hợp formant nối tiếp

#### Bộ tổng hợp formant song song

Bộ tổng hợp formant song song bao gồm các bộ cộng hưởng mắc song song. Đầu ra là kết hợp của tín hiệu nguồn và tất cả các formant. Cấu trúc song song cần nhiều thông tin để điều khiển hơn.



Hình 1.4. Cấu trúc cơ bản của một bộ tổng hợp formant song song

Tổng hợp formant là một phương pháp tổng hợp cho chất lượng chấp nhận được nhưng nếu yêu cầu chất lượng vao thì phương pháp này chưa đáp ứng được.

### Phương pháp ghép nối

Tổng hợp bằng phương pháp ghép nối các âm được tổng hợp từ các lời nói tự nhiên đã được thu âm từ trước là cách đơn giản nhất để sản sinh lời nói. Phương pháp tổng hợp ghép nối cho chất lượng cao và tương đối tự nhiên. Phương pháp này rất phù hợp với các hệ thống phát thanh và các hệ thống thông tin. Tuy nhiên phương pháp này thường chỉ áp dụng cho một giọng và phải sử dụng nhiều bộ nhớ hơn các phương pháp khác do số lượng từ vựng rất lớn. Để khắc phục nhược điểm này, người ta xây dựng các phương pháp tổng hợp ghép nối từ những đơn vị nhỏ như âm vị, âm tiết, âm vị kép… Các đơn vị âm được cắt ra từ tín hiệu rồi sau đó được tổng hợp lại theo yêu cầu dựa trên một thuật toán ghép nối.

### So sánh các phương pháp tổng hợp tiếng nói

#### Về chất lượng của tiếng nói tổng hợp

Phương pháp mô phỏng bộ máy phát âm về nguyên tắc sẽ cho chất lượng tốt nhất. Tuy nhiên chi phí để mô phỏng bộ máy phát âm là khá lớn và hoàn toàn không đơn giản để thực hiện. Tiếp đó là chất lượng tiếng nói tổng hợp từ phương pháp ghép nối tốt hơn và tự nhiên hơn từ phương pháp formant.

#### Về hiệu quả tính toán

Phương pháp mô phỏng bộ máy phát âm đòi hỏi chi phí tính toán lớn nhất vì mô phỏng chính xác bộ máy phát âm của con người là vô cùng khó và không dễ thực hiện. Hai phương pháp còn lại thì chi phí tính toán thấp hơn nhiều.

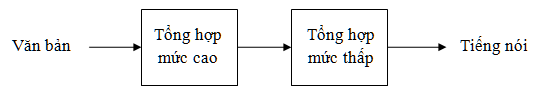
#### Về kích thước dữ liệu

Phương pháp ghép nối có kích thước dữ liệu lớn nhất do số lượng từ vựng lưu trữ là rất lớn. Hai phương pháp còn lại không phải lưu trữ các mẫu nên kích thước dữ liệu nhỏ hơn.

## Mô hình tổng hợp tiếng nói từ văn bản

Một trong những nhu cầu quan trọng của tổng hợp tiếng nói là quá trình tổng hợp tiếng nói từ văn bản (Text To Speech). Quá trình này được chia làm hai mức xử lý:

* Tổng hợp mức cao
* Tổng hợp mức thấp



Hình 1.5. Mô hình tổng hợp tiếng nói từ văn bản

### Tổng hợp mức cao

Tổng hợp mức cao là giai đoạn đầu của quá trình tổng hợp, giai đoạn chuyển đổi các văn bản thành các đơn vị tiếng nói. Tổng hợp mức cao gồm 3 bước:

* Xử lý văn bản đầu vào với các chữ số, các ký tự đặc biệt, chữ viết tắt và những từ viết tắt được ghép bằng các chữ cái đầu của các từ đầy đủ
* Phân tích cách phát âm của từ, kể cả từ đồng âm khác nghĩa và các tên riêng
* Phân tích ngữ điệu của tiếng nói

Sau khi tổng hợp mức cao, thông tin được cung cấp cho hệ thống mức thấp để điều khiển. Chẳng hạn, với bộ tổng hợp formant thì cần các thông tin như tần số cơ bản, tần số formant, khoảng thời gian và biên độ của mỗi đoạn âm thanh.

### Tổng hợp mức thấp

Tổng hợp mức thấp là quá trình kết hợp các đoạn tín hiệu. Các đoạn tín hiệu này đã được phân tích và xử lý ở mức cao.

Đối với phương pháp tổng hợp bằng cách mô phỏng hệ thống phát âm của con người thì sự lựa chọn dữ liệu và thực thi các luật là rất phức tạp. Hầu như không thể mô phỏng dưới dạng mô hình khối, sự chuyển động của lưỡi… một cách hoàn hảo.

Với bộ tổng hợp formant thì tập luật để điều khiển tần số cơ bản, biên độ và đặc trưng của tín hiệu nguồn lại rất lớn. Vì vậy làm mất đi tính tự nhiên vốn có.

Còn đối với phương pháp tổng hợp ghép nối thì việc thu thập các mẫu tín hiệu và gán nhãn mất rất nhiều thời gian và có thể làm cho cơ sở dữ liệu rất lớn. Tuy nhiên số lượng dữ liệu có thể giảm xuống đáng kể nếu sử dụng những phương pháp nén dữ liệu thích hợp. Bên cạnh đó sự không đồng bộ các điểm ghép nối cũng có thể làm tín hiệu tổng hợp bị méo. Đối với những đơn vị ghép nối dài như từ thì hiệu quả kết hợp là một vấn đề, ngoài ra bộ nhớ và hệ thống cũng là một khó khăn cần giải quyết.

## Kết chương

Trong chương 1, báo cáo đã trình bày khái quát sơ lược về các khái niệm cơ bản của tiếng nói, các biểu diễn tiếng nói ở dạng tín hiệu số cùng với đặc điểm âm học của tiếng nói.

Tiếp đó, báo cáo cũng giới thiệu ba phương pháp cơ bản được sử dụng phổ biến để tổng hợp tiếng nói hiện nay:

* Phương pháp mô phỏng hệ thống phát âm
* Phương pháp tổng hợp Formant
* Phương pháp ghép nối

Và mô hình tổng hợp tiếng nói từ văn bản.

# TỔNG HỢP TIẾNG VIỆT BẰNG PHƯƠNG PHÁP GHÉP NỐI CÁC ĐƠN VỊ ÂM KHÔNG ĐỒNG NHẤT

## Giới thiệu chương

Viện nghiên cứu Quốc Tế MICA đã và đang nghiên cứu và phát triển hệ thống thư ký ảo VIVA (**Vi**etnamese **V**oice **A**ssistant) với mục tiêu hỗ trợ các thiết bị Smartphone tương tác với người dùng bằng Tiếng Việt với các chức năng chính:

* Nhận dạng tiếng nói (Tiếng Việt)
* Hiểu ngữ nghĩa: hiểu được câu lệnh của người dùng
* Tổng hợp tiếng nói: phản hồi lại người dùng thông qua tiếng nói

Trong khuôn khổ đồ án, trong chương này, tôi xin giới thiệu về bộ tổng hợp Tiếng Việt mà nhóm nghiên cứu, phát triển dự án VIVA tại Viện MICA đã xây dựng thành công. Bộ tổng hợp Tiếng Việt được xây dựng bằng phương pháp ghép nối với loại đơn vị âm là bán âm tiết không đồng nhất.

## Tiếng Việt và đặc điểm ngữ âm của Tiếng Việt

Tiếng Việt là ngôn ngữ đơn âm tiết khác với các ngôn ngữ khác như tiếng Anh, tiếng Pháp… là các ngôn ngữ đa âm tiết. Âm tiết Tiếng Việt bao giờ cũng thể hiện khá đầy đủ, rõ ràng, được tách và ngắt thành từng khúc đoạn riêng biệt. Và tuyệt đại đa số các âm tiết đều có ý nghĩa. Mối quan hệ giữa âm và nghĩa trong âm tiết cũng chặt chẽ và thường xuyên như trong từ của các ngôn ngữ Châu Âu và đó chính là một nét đặc trưng loại hình chủ đạo của Tiếng Việt.

Mô hình âm tiết Tiếng Việt là một cấu trúc hai bậc bao gồm 3 phần chính: phần đầu, phần sau và thanh điệu. Phần đầu của âm tiết được xác định là âm đầu vì ở vị trí này chỉ có một âm vị tham gia cấu tạo. Phần sau của âm tiết được gọi là phần vần. Trong đó phần vần lại được chia làm ba phần nhỏ: âm đệm, âm chính và âm cuối. Từ đó, có thể hình dung mô hình một âm tiết Tiếng Việt ở dạng đầy đủ nhất gồm 5 thành tố, mỗi thành tố có một chức năng riêng.

|  |  |  |  |
| --- | --- | --- | --- |
| **THANH ĐIỆU** | | | |
| **ÂM ĐẦU** | **VẦN** | | |
| Âm đệm | Âm chính | Âm cuối |

Hình 2.1. Mô hình một âm tiết Tiếng Việt

##### Âm đầu

Có những cách mở đầu âm tiết khác nhau (âm tắc, âm xát), có chức năng tạo ra âm sắc cho âm tiết lúc mở đầu. Âm đầu do âm vị phụ âm đảm nhiệm. Ví dụ: ***t****oán –* ***h****oán.*

##### Âm đệm

Âm đệm có chức năng làm thay đổi âm sắc của âm tiết lúc khởi đầu và làm phân biệt âm tiết này với âm tiết khác. Âm đệm do âm vị bán nguyên âm đảm nhiệm. Ví dụ: *t****o****án – tán.*

##### Âm chính

Âm chính mang âm sắc chủ đạo của âm tiết và là hạt nhân của âm tiết. Âm chính trong Tiếng Việt do âm vị nguyên âm đảm nhiệm. Nguyên âm của Tiến Việt chỉ có chức năng làm âm chính và nó không bao giờ vắng mặt trong âm tiết. Vì mang âm sắc chủ yếu của âm tiết nên âm chính là âm mang thanh điệu. Ví dụ: *tú****y*** *– t****ú****i.*

##### Âm cuối

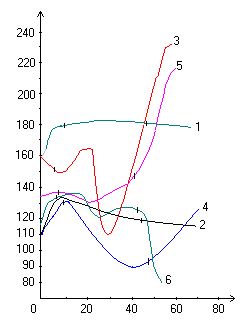
Âm cuối có vị trí cuối cùng của âm tiết có chức năng kết thúc một âm tiết. Do vậy khi có mặt âm cuối thì âm tiết không có khả năng kết hợp thêm với âm vị nào khác ở phía sau nó. Âm cuối là bán nguyên âm hoặc âm vị trống. Ví dụ: *bà****n*** *– bà****i.***

##### Thanh điệu

Thanh điệu là một yếu tố thể hiện độ cao và sự chuyển biến của độ cao trong mỗi âm tiết. Mỗi âm tiết Tiếng Việt nhất thiết phải được thể hiện với một thanh điệu. Thanh điệu có chức năng phân biệt vỏ âm thanh và phân biệt nghĩa của từ.

Tiếng Việt bao gồm 6 thanh điệu:

* Thanh ngang (1): xuất hiện trong tất cả các âm tiết, trừ âm tiết khép
* Thanh huyền (2): thấp hơn thanh ngang một bậc, xuất hiện trong các âm tiết không phải là âm tiết khép
* Thanh ngã (3): là thanh điệu thuộc âm vực cao (bắt đầu thấp hơn và kết thúc cao hơn), có thêm động tác nghẽn thanh hầu. Thanh ngã cũng không thể xuất hiện trong các âm tiết khép
* Thanh hỏi (4): là thanh điệu thuộc âm vực thấp. Khi phát âm, điểm bắt đầu và kết thúc thanh điệu đều ở âm vực thấp. Thanh này xuất hiện trong tất cả âm tiết không phải là âm tiết khép
* Thanh sắc (5): là thanh điệu thuộc âm vực cao. Khi phát âm, điểm xuất phát thấp hơn thanh ngang một chút và điểm kết thúc ở âm vực cao. Ngoài ra, khi kết thúc còn phải có thêm động tác nghẽn thanh hầu. Thanh này có thể xuất hiện trong tất cả kiểu âm tiết
* Thanh nặng (6): là thanh điệu thuộc âm vực thấp. Khi phát âm, điểm xuất phát gần với độ cao xuất phát của thanh huyền nhưng kết thúc đột ngột ở độ cao thấp hơn. Thanh nặng xuất hiện ở tất cả các kiểu âm tiết.



Hình 2.2. Biểu đồ thanh điệu Tiếng Việt

## Tổng hợp Tiếng Việt bằng phương pháp ghép nối các đơn vị âm không đồng nhất

### Cơ sở xây dựng

Căn cứ vào đặc điểm của ba phương pháp tổng hợp tiếng nói: phương pháp mô phỏng bộ máy phát âm, phương pháp formant và phương pháp ghép nối được trình bày trong chương 1, nhóm nghiên cứu, phát triển dự án VIVA tại viện MICA đã lựa chọn phương pháp ghép nối trong việc xây dựng bộ tổng hợp Tiếng Việt.

Trong tổng hợp tiếng nói theo phương pháp ghép nối, một âm tiết được tổng hợp bằng cách lấy ra đoạn sóng âm tương ứng với đơn vị âm và ghép nối lại với nhau. Và một câu văn, đoạn văn được tổng hợp bằng cách ghép nối các âm tiết được tổng hợp đó lại với nhau. Tuy nhiên, khi ghép nối hai đoạn sóng âm không liền kề nhau sẽ dẫn đến việc không liên tiếp về phổ và không liên tiếp về ngôn điệu. Sự không liên tiếp về phổ xảy ra khi formant tại điểm nối không hợp nhau. Sự không liên tiếp về ngôn điệu xảy ra khi cao độ tại điểm nối không hợp nhau. Chất lượng âm thanh sẽ tồi khi mà có nhiều điểm không liên tiếp trong ghép nối, cho dù các đơn vị âm là hoàn toàn tự nhiên.[[1]](#footnote-1)

Đánh giá theo mức tổng quát, chất lượng đầu ra của các bộ tổng hợp tiếng nói bằng phương pháp ghép nối có thể tốt trong hệ thống này nhưng lại tồi trong hệ thống khác. Nếu có đủ lượng đơn vị âm cần thiết, chất lượng âm thanh tổng hợp được gần với tiếng nói tự nhiên. Nhưng khi đó cơ sở dữ liệu lưu trữ các đơn vị âm sẽ tương đối lớn.

Trong các nghiên cứu về đặc điểm âm học của tiếng nói, có thể thấy rằng bất kỳ đoạn tín hiệu tiếng nói nào ngoài việc liên quan chặt chẽ với âm vị còn liên quan đến luật ngôn ngữ, trường độ, biên độ và tần số cơ bản F0 của đoạn tín hiệu. Bên cạnh đó, Tiếng Việt là ngôn ngữ có thanh điệu và khi thanh điệu thay đổi sẽ dẫn tới sự thay đổi ngữ nghĩa của từ. Và các nghiên cứu về thanh điệu trong Tiếng Việt cho thấy rằng sự thay đổi thanh điệu là kết quả của sự thay đổi tần số cơ bản của âm. Do đó với việc biến đổi tần số cơ bản của một âm không dấu, ta có thể tạo ra các thanh điệu từ các âm tiết không dấu đầu vào hay có thể nói một cách khác là ta có thêm được 5 âm khác từ một âm không dấu đầu vào. Điều này cho phép giảm kích thước dữ liệu cần lưu trữ.[[2]](#footnote-2)

Và trong quá trình nghiên cứu, phân tích các giải thuật ghép nối của các nghiên cứu viên ở viện MICA, bộ tổng hợp Tiếng Việt trong dự án VIVA được xây dựng bằng phương pháp ghép nối sử dụng giải thuật PSOLA (Pitch Synchronous Overlap and Add). Giải thuật này sẽ được trình bày ở mục sau của chương.

Tóm lại, với các cơ sở như trên, để xây dựng một bộ tổng hợp Tiếng Việt với chất lượng tốt theo phương pháp ghép nối cần quan tâm đến các vấn đề quan trọng sau:

* Lựa chọn loại đơn vị âm phù hợp
* Xây dựng cơ sở dữ liệu với loại đơn vị âm được chọn
* Tìm kiếm đơn vị âm tối ưu trong cơ sở dữ liệu để ghép nối
* Ghép nối các đơn vị âm

### Lựa chọn loại đơn vị âm

Cấu trúc âm tiết của Tiếng Việt gồm 5 thành tố: âm đầu, âm đệm, âm chính, âm cuối và thanh điệu. Mỗi thành tố của âm tiết (ngoại trừ thanh điệu) được cấu thành từ một hoặc một vài âm vị. Từ đó, chúng ta có thể phân chia một âm tiết Tiếng Việt thành các đơn vị âm có thể sử dụng trong quá trình tổng hợp tiếng nói:

* Âm vị: là đơn vị nhỏ nhất của ngôn ngữ, bao gồm âm vị phụ âm và âm vị nguyên âm
* Âm vị kép: là một cặp âm vị liền kề nhau
* Âm tiết: là đơn vị phát âm ngắn nhất trong tiếng nói
* Bán âm tiết
* Âm đầu và vần

Với đơn vị âm là âm vị, về mặt lý thuyết có thể tổng hợp được tất cả các câu. Số lượng âm vị trong Tiếng Việt cũng khá nhỏ (23 âm vị phụ âm, 16 âm vị nguyên âm và 2 âm vị bán nguyên âm)[[3]](#footnote-3). Tuy nhiên, sự thay đổi ngữ cảnh thường xuyên trong Tiếng Việt tạo ra sự không liên tục giữa các điểm kết nối của các đơn vị âm. Âm thanh tổng hợp có thể chất lượng sẽ kém và đôi khi còn khó hiểu.

Với đơn vị âm là âm tiết, mặc dù nó bảo toàn đầy đủ sự đồng cấu âm giữa các âm vị bên trong của đơn vị âm nhưng yêu cầu cơ sở dữ liệu ban đầu là rất lớn vì số lượng từ vựng trong Tiếng Việt rất đa dạng và phong phú.

Còn trường hợp đơn vị âm là âm đầu và vần dù giữ lại được đặc tính của thanh điệu nhưng tại các điểm ghép nối giữa âm đầu và vần có sự không liên tục tương tự như với đơn vị âm là âm vị.

Với đơn vị âm là âm vị kép và bán âm tiết, biên giữa các đơn vị âm vị này khi tổng hợp lại sẽ chồng lên nhau. Do đó làm giảm đi sự không liên tục trong ghép nối và phù hợp với giải thuật PSOLA. Hơn nữa, Tiếng Việt là ngôn ngữ đơn âm tiết nên theo cách phát âm, bất kỳ từ nào cũng có thể phân tích được thành hai đơn vị âm tương ứng. Ví dụ, từ *“minh”* có thể được tách thành hai âm vị kép *“mi”* và *“inh”*. Ngoài ra, cơ sở dữ liệu đầy đủ các âm vị kép và bán âm tiết nhỏ hơn rất nhiều so với âm tiết. Ví như tổng số âm vị kép cần có để tạo nên tất cả các từ Tiếng Việt là 389 âm vị kép[[4]](#footnote-4).

Số lượng các loại đơn vị âm được thống kê tổng hợp theo bảng sau[[5]](#footnote-5):

|  |  |  |
| --- | --- | --- |
| **Loại đơn vị âm** | **Số lượng** | |
| **Không có thanh điệu** | **Có thanh điệu** |
| Âm vị | 40 | 130 |
| Âm vị kép | 620 | 2976 |
| Bán âm tiết | 590 | 2809 |
| Âm đầu/vần | 22/161 | 22/661 |
| Âm tiết | 2466 | 7088 |

Bảng 2. Số lượng các loại đơn vị âm trong Tiếng Việt

Ngoài các đơn vị âm nhỏ hơn âm tiết, còn có thể kể đến loại đơn vị âm là cụm từ. Sử dụng các đơn vị âm này có thể tăng mức độ tự nhiên của tiếng nói tổng hợp do giảm thiểu điểm ghép nối. Tuy nhiên việc đảm bảo đơn vị âm này khớp với ngữ điệu mong muốn là rất khó. Việc dùng các cụm từ cũng có thuận lợi là khi tìm kiếm, ta không cần dạng phiên âm của âm tiết mà có thể tìm trực tiếp bản thân cụm từ đó. Điều này làm giảm thời gian thực thi của chương trình.

Từ các ưu điểm và nhược điểm của các đơn vị âm được phân tích ở trên, âm vị kép và bán âm tiết hiện vẫn đang là những đơn vị âm được sử dụng phổ biến nhất trong các hệ thống tổng hợp tiếng nói bằng phương pháp ghép nối. Viện nghiên cứu MICA cũng đang nghiên cứu và sử dụng chủ yếu hai loại đơn vị âm này trong các bộ tổng hợp tiếng nói đang được phát triển tại viện. Trong đó, cơ sở dữ liệu cho loại đơn vị âm là bán âm tiết được xây dựng và phát triển đầy đủ nhất.

Và bộ tổng hợp Tiếng Việt được dùng trong khuôn khổ đồ án này sử dụng kết hợp các loại đơn vị âm bao gồm bán âm tiết, âm tiết và cụm từ, gọi chung là lựa chọn đơn vị âm không đồng nhất[[6]](#footnote-6).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Loại đơn vị âm | Độ dài đơn vị âm | | Số điểm ghép nối | | Xác suất tìm thấy trong CSDL | |
| Cụm từ | Dài |  | Ít |  | Cao |  |
| Âm tiết |  |  |  |
| Bán âm tiết | Ngắn | Nhiều | Thấp |

Bảng 3. Ưu, nhược điểm của từng loại đơn vị âm sử dụng trong bộ tổng hợp Tiếng Việt

Lựa chọn đơn vị âm không đồng nhất kết hợp ưu điểm của cả ba loại đơn vị âm trên: giảm thiểu số điểm ghép nối bằng việc sử dụng đơn vị âm mức cụm từ và âm tiết, đồng thời đảm bảo khả năng tổng hợp hàu hết âm tiết trong Tiếng Việt bằng việc sử dụng đơn vị âm bán âm tiết. Nhược điểm của phương pháp này là sự rắc rối trong việc sử dụng ba loại đơn vị âm đòi hỏi cách xử lý linh hoạt, chuyển đổi qua lại giữa các loại đơn vị âm.

### Xây dựng cơ sở dữ liệu

Với phương pháp tổng hợp tiếng nói bằng ghép nối thì cơ sở dữ liệu là yếu tố quan trọng tạo nên chất lượng của tiếng nói tổng hợp. Cơ sở dữ liệu được tạo ra từ các đoạn âm thanh mẫu đầu vào. Sau đó được tách thành các âm tiết. Một âm tiết có thể được tạo ra từ nhiều bản âm thanh mẫu để có thể chọn được mẫu với chất lượng tốt nhất.

Sau khi có được các âm tiết, chúng ta tiếp tục tách thành các đơn vị âm. Đây là công việc khó khăn nhất trong khi xây dựng dữ liệu. Chúng ta cần cắt các đoạn tín hiệu từ các mẫu âm tiết đã có. Điểm cắt phải nẳm trong trong phần tín hiệu ổn định và tại đỉnh cao nhất trong một chu kỳ cùa phần tín hiệu tương ứng với nguyên âm (gần như tuần hoàn). Điều này đảm bảo rằng hai âm vị kép dùng để ghép nối không bị lệch pha nhiều.

Cuối cùng, ta cần lưu trữ lại thông tin của đơn vị âm vào một tệp dữ liệu cùng với phần dữ liệu của đoạn âm thanh mẫu tương ứng. Tệp dữ liệu cần được tổ chức để việc quản lý, cập nhật dữ liệu thuận tiện dễ dàng.

Tóm lại, để xây dựng cơ sở dữ liệu cho loại đơn vị âm được lựa chọn ở mục trước, chúng ta cần thực hiện ba công việc sau:

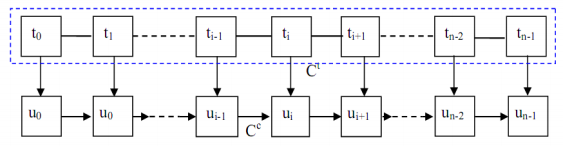
* Thu mẫu âm thanh
* Tách đoạn mẫu âm thanh thu được thành các đơn vị âm
* Lưu trữ thông tin các đơn vị âm và đoạn âm thanh mẫu tương ứng

Trong bộ tổng hợp Tiếng Việt sử dụng trong khuôn khổ đồ án này, cơ sở dữ liệu các đơn vị âm sử dụng lại kho đơn vị âm đã được Viện nghiên cứu MICA thu âm và lưu trữ.

### Tìm kiếm đơn vị âm tối ưu

Khi hệ thống có thông tin về đoạn văn bản cần tổng hợp, hệ thống tổng hợp sẽ chuyển đổi đoạn văn bản đầu vào thành một đặc tả đích. Đặc tả đích của một đoạn văn bản định nghĩa một tập các đơn vị âm cần thiết để tổng hợp tiếng nói từ đoạn văn bản đó. Các đơn vị âm trong tập sẽ được gán thêm các tham số ngữ điệu như tần số cơ bản, năng lượng, trường độ.

Giả sử một câu đầu vào được phân tích thành một chuỗi gồm n đơn vị âm để tổng hợp. Đích của câu này là chuỗi n đơn vị (ti, i = 0…n-1) chứa những thông tin về ngữ điệu cần thiết. Từ đích này, ta cần tìm ra chuổi n đơn vị âm (ui, i = 0…n-1) trong cơ sở dữ liệu, cho phép hệ thống tổng hợp ra đoạn âm thanh với chất lượng tốt nhất có thể.



Hình 2.3. Hàm chi phí giữa các đơn vị âm

Hàm chi phí được sử dụng:

* Chi phí đích Ct(ui, ti) là sự khác nhau giữa đơn vị âm trong cơ sở dữ liệu ui và đích ti gồm 6 thành phần
* Chi phí ghép nối Cc(ui-1, ui) là sự khác nhau tại điểm ghép nối giữa hai đơn vị âm liên tiếp ui-1 và ui

Với đặc tả về đích và chuỗi n đơn vị âm T1n = (t1, t2, …, tn), hệ thống cần chọn ra n đơn vị âm U1n = (u1, u2, …, un) mà gần với đích nhất.

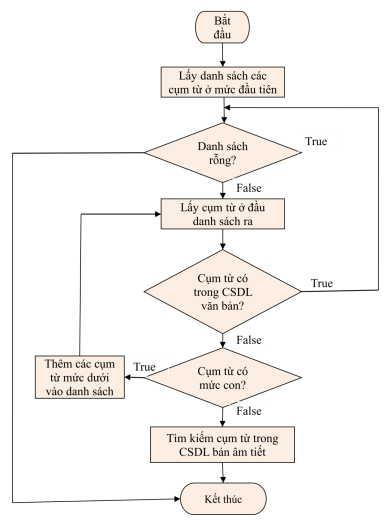
Trong phương pháp này, chi phí tính toán là đáng kể nếu như số lượng các đơn vị âm trong cơ sở dữ liệu là lớn. Vì vậy, để giảm thời gian và chi phí tính toán, quá trình lựa chọn đơn vị được chia làm hai bước: tiền lựa chọn và chọn lựa cuối cùng.

#### Tiền lựa chọn

Trong cơ sở dữ liệu mỗi đơn vị âm có thể có một hoặc nhiều mẫu tín hiệu, mỗi mẫu được sử dụng trong những ngữ cảnh khác nhau. Giai đoạn tiền lựa chọn có mục tiêu là tìm kiếm trong cơ sở dữ liệu các mẫu tương ứng với đơn vị đích. Giai đoạn này sẽ giúp làm giảm thời gian thực hiện của hệ thống và nó cũng giảm đi việc giảm chất lượng tiếng nói tổng hợp.

Đầu tiên, ta tìm trong cơ sở dữ liệu tất cả các đơn vị âm có khoảng cách ngữ âm nhỏ nhất với đơn vị âm đích. Cơ sở dữ liệu được dùng trong bước tiền lựa chọn là cơ sở dữ liệu văn bản và cơ sở dữ liệu bán âm tiết. Và với mục đích giảm thiểu số điểm ghép nối, loại đơn vị âm ưu tiên chọn lựa sẽ theo thứ tự: cụm từ, âm tiết và bán âm tiết

Bắt đầu quá trình tìm kiếm, văn bản đầu vào được chia thành các câu để tìm kiếm. Mỗi câu được phân tách thành các cụm từ và âm tiết rồi tìm kiếm chúng trong cơ sở dữ liệu văn bản. Nếu tìm thấy, vị trí tìm thấy và các thông tin về ngữ cảnh và ngữ âm của đơn vị âm tìm thấy được trả về để dùng cho việc tình toán hàm chi phí. Nếu âm tiết không được tìm thấy, âm tiết sẽ được phân tích thành hai bán âm tiết đầu và cuối. Các bán âm tiết này được tìm kiếm trong cơ sở dữ liệu bán âm tiết. Nếu không tìm thấy thì âm tiết đó không được tổng hợp.

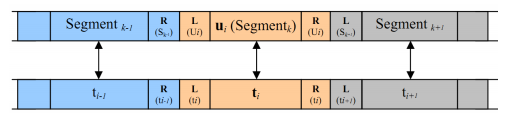


Hình 2.4. Sơ đồ tìm kiếm đơn vị âm

Vấn đề đặt ra là làm sao có thể phân tích được một câu thành các cụm từ và âm tiết sao cho tối đa hóa xác suất tìm thấy cụm từ được phân tích. Bởi nếu không chọn được cụm từ thích hợp để tìm kiếm, tỉ lệ đơn vị âm phần lớn sẽ là âm tiết và bán âm tiết, việc này ảnh hưởng trực tiếp tới hiệu quả của thuật toán lựa chọn đơn vị âm không đồng nhất. Một giải pháp được đề xuất để giải quyết vấn đề này là sử dụng cây phân tích cú pháp. Câu cần được tổng hợp sẽ đươc chia ra thành các cụm từ theo các mức khác nhau nhờ quá trình phân tích cú pháp. Quá trình tìm kiếm sẽ được bắt đầu từ gốc, sau đó đi xuống các nhanh. Việc tìm kiếm sẽ dừng lại ở mức cao nhất có thể ngay khi tìm thấy cụm từ hoặc đi tới mức lá là các âm tiết. Cách thức phân chia này làm tăng xác suất tìm thấy của những cụm từ có độ dài hơn một âm tiết hơn là việc chọn ngẫu nhiên cụm từ theo một độ dài xác định nào đó để tìm kiếm. Đây là ý tưởng chủ đạo trong thuật toán lựa chọn đơn vị âm không đồng nhất.

Trong trường hợp không tìm thấy ứng viên ở mức lá, âm tiết sẽ được tổng hợp ở mức bán âm tiết. Và trong các nghiên cứu, việc tổng hợp ở mức bán âm tiết có thể tổng hợp được hầu hết các âm tiết trong Tiếng Việt.

Sau khi tìm được các đơn vị âm từ trong cơ sở dữ liệu, sẽ chọn ra những đơn vị âm có độ méo thấp nhất dựa trên hàm chi phí đích Ct(ui, ti). Chi phi đích Ct(ui, ti) được tính bằng tổng của khoảng cách về ngữ cảnh và ngữ điệu của đích ti và đơn vị âm ui.



Hình 2.5. So sánh đích ti và đơn vị âm ui

Khoảng cách về ngữ cảnh và ngữ điệu này được ước lượng bằng sự khác nhau của đích ti và đơn vị âm ui trên các phương diện:

* Ngữ cảnh: thông tin được so sánh ở đây là phiên âm và thanh điệu. Nếu hai giá trị này của ti và ui là như nhau thì chi phí là 0, ngược lại chi phí là 1.
* Ngữ điệu: thông tin được so sánh ở đây là trường độ, tần số cơ bản và năng lượng. Sự khác nhau về mặt ngữ điệu được biểu diễn thành vector. Giá trị của vector sẽ được chuẩn hóa để nhận các giá trị 0 và 1.

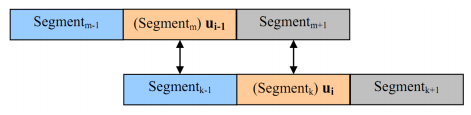
Sau giai đoạn tiền lựa chọn, với mỗi đích ti, sẽ chọn ra N (~20) mẫu tốt nhất. Tuy nhiên với số mẫu như vậy sẽ tạo ra Nn các tập hợp mẫu. Và giai đoạn tiếp theo – giai đoạn chọn lựa cuối cùng sẽ chọn ra chuỗi n mẫu tốt nhất trong Nn tập hợp mẫu đó.

#### Lựa chọn cuối cùng

Mục đích của giai đoạn này là chọn ra chuỗi các đơn vị âm sao cho sự không liên tục là nhỏ nhất có thể. Tiêu chí lựa chọn là dựa trên hàm chi phí bao gồm chi phí đích Ct(ui, ti) và chi phí ghép nối Cc(ui-1, ui). Chi phí đích Ct(ui, ti) được tính toán trong giai đoạn tiền lựa chọn. Chi phí ghép nối Cc(ui-1, ui) được tính bằng tổng của khoảng cách về ngữ cảnh và khoảng cách về phổ của hai đơn vị âm ui-1 và ui.

Khoảng cách về ngữ cảnh của hai đơn vị âm ui-1 và ui được ước lượng bằng cách so sánh:

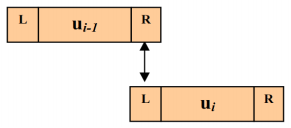
* Sự khác nhau giữa đơn vị âm bên phải của ui-1 và ui­ trong 2 chuỗi khác nhau trong Nn tập hợp mẫu ở giai đoạn tiền lựa chọn
* Sự khác nhau giữa đơn vị âm bên trái của ui và ui­-1 trong 2 chuỗi khác nhau trong Nn tập hợp mẫu ở giai đoạn tiền lựa chọn



Hình 2.6. So sánh sự khác nhau về ngữ cảnh của 2 đơn vị âm ui và ui-1

Nếu hai giá trị so sánh trên là giống nhau thì khoảng cách bằng 0, ngược lại khoảng cách bằng 1.

Khoảng cách về phổ giữa hai đơn vị âm ui và ui­-1 là khoảng cách Euclid giữa 12 hệ số MFCC của hai của sổ 10ms (của sổ cuối cùng của ui-1 và cửa sổ đầu tiên của ui).



Hình 2.7. So sánh sự khác nhau về phổ của hai đơn vị âm ui và ui-1

Khi đó, hàm chi phí dùng để so sánh ở giai đoạn lựa chọn cuối cùng này được tính theo công thức:

Trong đó, S mô tả khoảng lặng, và xác định các điều kiện ban đầu và kết thúc để cho việc ghép nối các đơn vị âm đầu và cuối có khoảng lặng.

Quy trình chọn lựa tập hợp các đơn vị âm phải thỏa mãn tổng chi phí tính toán phải được nhỏ nhất:

Tuy nhiên, trong trường hợp xây dựng bộ tổng hợp tiếng Việt sử dụng kết hợp ba loại đơn vị âm nên cần thiết phải có cơ chế lựa chọn khác. Giải pháp được đề xuất ở đây là tối ưu hóa cục bộ hàm chi phí[[7]](#footnote-7):

* Bước 1: chia dãy đơn vị âm cần tối ưu thành các dãy con sao cho các loại đơn vị âm trong dãy con là cùng một loại bán âm tiết, âm tiết hoặc cụm từ
* Bước 2: tính toán hàm chi phí cho các dãy con và loại bỏ một số ứng viên có hàm chi phi lớn nhất.
  + Đối với dãy con chứa bán âm tiết
    - Tính toán hàm chi phí cho dãy con theo công thức ở trên
    - Giữ lại NhalfSyl chuỗi đơn vị âm có hàm chi phí nhỏ nhất trong dãy con này (NhalfSyl được xác định bàng thực nghiệm, thường có giá trị nhỏ hơn nhiều so với số khả năng kết hợp của các đơn vị âm trong dãy)
  + Đối với dãy con chứa âm tiết
    - Tính hàm chi phí ghép nối dựa vào các tham số
      * LeftSyl: âm tiết liền kề bên trái trong CSDL
      * RightSyl: âm tiết liền kề bên phải trong CSDL
      * LeftPh: âm cuối của âm tiết liền kề bên trái
      * RightPh: âm đầu của âm tiết liền kề bên phải
      * LeftT: thanh điệu của âm tiết bên trái
      * Right: thanh điệu của âm tiết bên phải
    - Hàm chi phí đích được thay bằng hàm chi phí phụ cận. Hàm này có giá trị bằng 0 nếu hai âm tiết ứng viên là hai đoạn âm thanh liên tiếp nhau trong CSDL, ngược lại có giá trị bằng 1
    - Hàm chi phí tổng là kết hợp của chi phí ghép nối với chi phí phụ cận.
    - Giữ lại Nsyl chuỗi âm tiết có hàm chi phí nhỏ nhất trong dãy con
  + Đối với dãy con chứa cụm từ
    - Vẫn sử dụng hai loại hàm chi phí như đối với mức bán âm tiết. Giữ lại Nphrase dãy cụm từ có hàm chi phí nhỏ nhất trong dãy con
* Bước 3: chọn ra dãy đơn vị âm có hàm chi phí nhỏ nhất sử dụng thuật toán Viterbi.

### Ghép nối các đơn vị âm và biến đổi tần số cơ bản

Với mỗi từ Tiếng Việt, ta xác định được đơn vị âm bắt đầu và đơn vị âm kết thúc. Để tạo thành tiếng nói tổng hợp của âm tiết, ta cần ghép nối các đơn vị âm này lại và biến đổi tần số cơ bản để tạo thanh điệu. Và như đã trình bày ở trên, giải thuật được đề xuất là giải thuật PSOLA.

#### Giải thuật PSOLA

PSOLA (Pitch Synchronous Overlap and Add) còn được gọi là kỹ thuật chồng đồng bộ cao độ tần số cơ bản. Nó sẽ phân tích tín hiệu tiếng nói đầu vào thành một chuỗi các sóng cơ bản riêng biệt. Các sóng cơ bản này thu được nhờ sử dụng các hàm của sổ tạp trung tại các vị trí chu kỳ cơ bản của tín hiệu. Các sóng cơ bản này được biến đổi về trường độ và cao độ một cách riêng biệt rồi sau đó cộng xếp chồng chúng lên nhau.

Phương pháp PSOLA bao gồm 3 bước cơ bản sau:

* Phân tích tín hiệu thành các sóng cơ bản
* Tính toán các điểm đánh dấu cao độ: bước này sẽ thực hiện biến đổi trường độ và cao độ của tín hiệu. Việc biến đổi cao độ được thực hiện bằng cách thay đổi khoảng cách giữa các sóng cơ bản thu được ở bước phân tích. Việc biến đổi trường độ tín hiệu được thực hiện bằng việc lặp lại hoặc bỏ bớt các sóng cơ bản. Lặp lại thì sẽ làm tăng trường độ, bỏ bớt làm giảm trường độ
* Tổng hợp lại các đoạn tín hiệu đã biến đổi

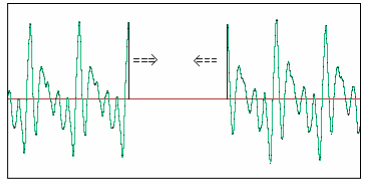
PSOLA có ba phiên bản chính:

* TD-PSOLA (Time Domain PSOLA) là phiên bản trên miền thời gian của PSOLA. Phương pháp này thao tác với tín hiệu trên miền thời gian nên được sử dụng nhiều vì hiệu quả trong tính toán
* FD-PSOLA (Frequency Domain PSOLA) tương tự như TD-PSOLA nhưng thao tác trên miền tần số. Phương pháp này có chi phí tính toán cao hơn TD-PSOLA
* LP-PSOLA (Linear Prediction PSOLA) được thiết kế để mã hóa tiếng nói nhưng cũng có thể dùng cho tổng hợp.

Bộ tổng hợp Tiếng Việt sử dụng trong đồ án sử dụng phiên bản TD-PSOLA để ghép nối và biến đổi tần số cơ bản. Trong báo cáo này không trình bày chi tiết giải thuật TD-PSOLA mà chỉ dừng ở mức giới thiệu khái quát về phương pháp PSOLA được sử dụng trong bộ tổng hợp Tiếng Việt.

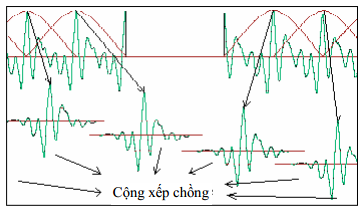
#### Ghép nối các đơn vị âm

Sau khi truy xuất trong cơ sở dữ liệu theo tên của hai đơn vị âm bắt đầu và kết thúc, chúng ta có được hai đoạn tín hiệu tương ứng với hai đơn vị âm này. Việc ghép được thực hiện giữa phần kết thúc của âm vị kép bắt đầu và phần bắt đầu của âm vị kép kết thúc.



Hình 2.8. Ví dụ về ghép nối 2 đơn vị âm

Muốn thay đổi độ dài tín hiệu thu được, trước hết các đơn vị âm cần được phân tích thành các tín hiệu thành phần có độ dài xác định được. Sau đó dùng TD-PSOLA cộng xếp chồng cho tín hiệu thu được bởi việc ghép hai đoạn tín hiệu để thu được một tín hiệu mới có độ dài mong muốn.



Hình 2.9. Cộng xếp chồng các tín hiệu thành phần

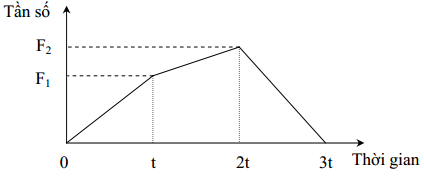
Sau khi thực hiện ghép nối, có thể biến đổi tần số cơ bản của tín hiệu theo quy luật biến đổi tần số cơ bản của các thanh điệu trong Tiếng Việt.

#### Biển đổi tần số cơ bản

Tín hiệu thu được sau khi ghép nối có tần số cơ bản là tần số cơ bản của tín hiệu ban đầu.

Để biến đổi tần số cơ bản của tín hiệu ta cần biết:

* Khoảng thời gian của tiếng nói tổng hợp
* Quá trình biến đổi của tần số cơ bản

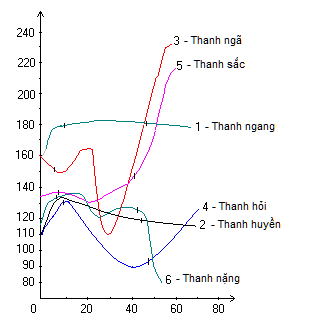


Hình 2.10. Quá trình biến đổi tần số cơ bản theo thời gian

Một cách gần đúng, có thể coi tần số cơ bản biến đổi theo đường gấp khúc như Hình 2.10. Tín hiệu tổng hợp được chia thành các đoạn với độ dài (theo thời gian) bằng nhau, tần số cơ bản trong mỗi đoạn thời gian biến thiên theo một đường thẳng. Nhờ vậy có thể xác định được tần số cơ bản tại tất cả các thời điểm.

Tín hiệu được chia thành các đoạn bằng nhau với độ dài (theo thời gian) nhỏ hơn, tần số cơ bản của mỗi đoạn không đổi và bằng tần số tại thời điểm giữa của đoạn trong quá trình biến đổi. Áp dụng TD-PSOLA, ta có thể tổng hợp được đoạn tín hiệu với tần số và độ dài cho trước.

Trong Tiếng Việt, ngữ nghĩa của một từ phụ thuộc vào thanh điệu. Khi thanh điệu thay đổi, nghĩa của từ cũng thay đổi theo. Có 6 thanh điệu trong Tiếng Việt: thanh ngang (không dấu), thanh huyền, thanh sắc, thanh nặng, thanh hỏi, thanh ngã. Tương ứng với mỗi thanh điệu, tần số cơ bản thay đổi theo một quy luật riêng. Hình 2.11 là quy luật của từng thanh điệu.



Hình 2.11. Biểu đồ thanh điệu Tiếng Việt

Ngoài ra, việc biến đổi tần số cơ bản của tín hiệu còn tạo ra các loại câu khác nhau trong Tiếng Việt: câu trần thuật, câu hỏi, câu cảm thán, câu cầu khiến… Tuy nhiên, báo cáo đồ án chỉ dừng lại ở mức trình bày về việc biến đổi tần số cơ bản tạo thanh điệu của âm tiết.

## Kết chương

Trong chương 2, báo cáo đã trình bày tổng quát về phương pháp tổng hợp Tiếng Việt bằng phương pháp ghép nối sử dụng giải thuật TD-PSOLA với lựa chọn đơn vị âm không đồng nhất.

Trước khi bắt đầu trình bày tổng hợp Tiếng Việt, chương có điểm qua vài đặc điểm ngữ âm của Tiếng Việt là ngôn ngữ được sử dụng để xây dựng bộ tổng hợp tiếng nói.

Báo cáo đã trình bày cơ sở lựa chọn phương pháp ghép nối cùng với đơn vị âm không đồng nhất để xây dựng bộ tổng hợp Tiếng Việt. Trình bày tổng quát về các công việc cần thực hiện khi xây dựng bộ tổng hợp Tiếng Việt:

* Lựa chọn loại đơn vị âm
* Xây dựng cơ sở dữ liệu
* Tìm kiếm đơn vị âm tối ưu
* Ghép nối các đơn vị âm và biến đổi tần số cơ bản để tạo thanh điệu

Chương này được viết dựa trên tổng hợp kiến thức thu được từ việc tham khảo các báo cáo của các nhà nghiên cứu phương pháp tổng hợp tiếng nói bằng phương pháp ghép nối và các báo cáo đồ án tốt nghiệp của các anh chị khóa trước nên chương chỉ trình bày ở mức độ lý thuyết. Phần công việc xây dựng bộ tổng hợp Tiếng Việt ở đây do 2 sinh viên Nguyễn Tiến Thành và Đỗ Kim Lân trong nhóm sinh viên nghiên cứu tại viện MICA thực hiện. Phần tiếp theo trình bày phần công việc bản thân được giao trong quá trình làm đồ án: xây dựng Engine Text-To-Speech sử dụng bộ tổng hợp Tiếng Việt được MICA xây dựng.

# XÂY DỰNG ENGINE TEXT-TO-SPEECH TRÊN NỀN TẢNG ANDROID

## Giới thiệu chương

Với bộ tổng hợp tiếng Việt do MICA nghiên cứu và phát triển, một trong những mục tiêu của đồ án tốt nghiệp là xây dựng một Engine Text-To-Speech trên Android có sử dụng bộ tổng hợp Tiếng Việt của MICA.

Trong chương này, báo cáo sẽ trình bày cái nhìn tổng quan về hệ điều hành Android và kiến trúc của nó. Tiếp đó, báo cáo tiếp tục trình bày khái quát về tính năng Text-To-Speech được tích hợp trên hệ điều hành Android cùng cơ chế hoạt động và quy trình xây dựng một Engine Text-To-Speech trên Android.

Và phần cuối chương sẽ trình bày thực nghiệm xây dựng một Engine Text-To-Speech trên Android với tên gọi là Viva TTS.

## Hệ điều hành Android

### Giới thiệu về hệ điều hành Android

**Android** là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như Smartphone (điện thoại thông minh) hay Tablet (máy tính bảng).

Android được phát triển bởi tổng công ty Android với sự hỗ trợ tài chính từ Google. Đến năm 2005, Google mua lại hệ điều hành này và ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở (Open Handset Alliance) - một hiệp hội gồm các công ty phần cứng, phần mềm và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động.



Hình 3.1. Liên minh di động mở

Chiếc điện thoại đầu tiên chạy hệ điều hành Android là HTC Dream, phát hành vào ngày 22 tháng 10 năm 2008.



Hình 3.2. Chiếc điện thoại đầu tiên chạy hệ điều hành Android: HTC Dream

Android là hệ điều hành mã nguồn mở và Google phát hành mã nguồn theo giấy phép Apache. Điều đó giúp cho các nhà phát triển thiết bị và mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do bằng một loại ngôn ngữ lập trình Java có sửa đổi.

Android đang trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới với khoảng 700.000 ứng dụng và ước tính khoảng 25 tỷ lượt tải qua Google Play vào tháng 10 năm 2012. Android đã vượt qua hệ điều hành Symbian và trở thành một hệ điều hành được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Tính vào thời điểm quý III năm 2012, Android chiếm 75% thị phần điện thoại thông minh với tổng cộng 500 triệu thiết bị được kích hoạt và 1,3 triệu lượt tải mỗi ngày.

Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong các “cuộc chiến điện thoại thông minh” giữa các công ty công nghệ.

### Linux – Nền tảng hệ điều hành Android

Android được Google tự phát triển riêng cho đến khi những thay đổi và cập nhật đã hoàn thiện, khi đó mã nguồn mở mới được công khai. Mã nguồn này nếu không sửa đổi chỉ chạy trên một số thiết bị, thường là thiết bị thuộc dòng Nexus. Có nhiều thiết bị có chứa những thành phần được giữ bản quyền do nhà sản xuất đặt vào thiết bị Android của họ.

Android có một hạt nhân dựa trên nhân Linux phiên bản 2.6, kể từ Android 4.0 – Ice Cream Sandwich trở về sau là phiên bản 3.x, với middleware, thư viện và API viết bằng C, còn phần mềm ứng dụng chạy trên một nền tảng ứng dụng gồm các thư viện tương thích với Java dựa trên Apache Harmony. Android sử dụng máy ảo Dalvik với một trình biên dịch tự dộng để chạy “mã dex” (Dalvik Executable) của Dalvik, thường được biên dịch sang Java bytecode. Nền tảng phần cứng chính của Android là kiến trúc ARM. Ngoài ra còn hỗ trợ x86 thông qua dự án Android x86. Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android

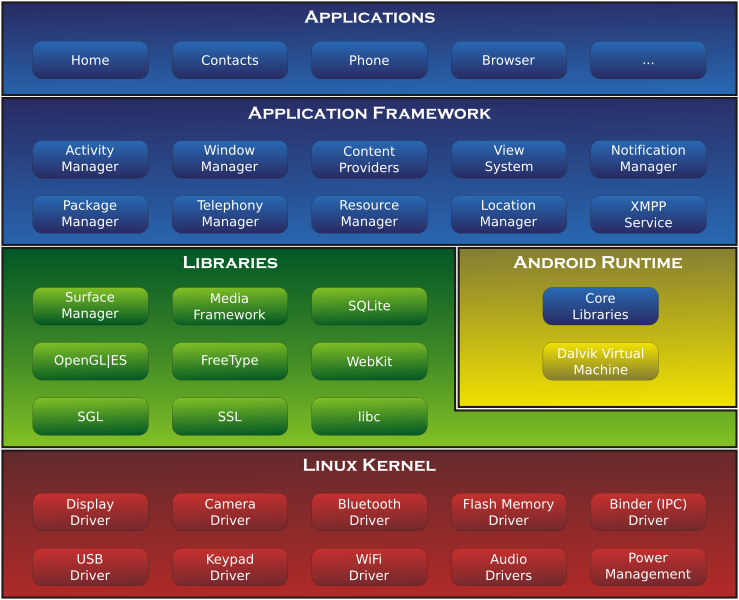
Nhân Linux dùng cho Android đã được Google thực hiện nhiều thay đổi về kiến trúc so với nhân Linux gốc. Android ko có sẵn X Window System cũng không hỗ trợ các thư viện GNU chuẩn nên việc chuyển các ứng dụng hoặc thư viện Linux có sẵn sang Android rất khó khăn. Các ứng dụng C đơn giản và SDK cũng được hỗ trợ bằng cách chèn những đoạn shim Java và sử dụng tương tự JNI.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux như tính năng quản lý nguồn điện wakelock nhưng không được chấp nhận.

### Kiến trúc hệ điều hành Android

Hệ điều hành Android được chia làm 5 tầng:

* Applications
* Applications Framework
* Libraries
* Android Runtime
* Linux Kernel



Hình 3.3. Kiến trúc hệ điều hành Android

##### Tầng Applications

Tầng Applications – tầng ứng dụng là tầng chứa các ứng dụng được phát triển bằng ngôn ngữ Java có chỉnh sửa. Ngoài một số ứng dụng cơ bản được tích hợp sẵn bời Google như Email Client, SMS, Calender, Map, Web Browser, Contact… thì tầng này là nơi chứa các ứng dụng khác được viết bởi các lập trình viên, các nhà phát triển hay các nhà phát hành thiết bị di động.

##### Tầng Application Framework

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin điểm truy cập, các dịch vụ chạy nền…

Nhà phát triển có thể truy cập và sử dụng các API được dùng trong các ứng dụng tích hợp sẵn của Android. Kiến trúc ứng dụng của Android được thiết kế nhằm mục đích đơn giản hóa việc tái sử dụng các component. Qua đó bất kì ứng dụng nào cũng có thể công bố các tính năng mà nó muốn chia sẻ cho các ứng dụng khác như tính năng gửi mail của ứng dụng Email được tích hợp sẵn. Phương pháp tương tự cho phép các thành phần được thay thế bởi người dùng.

Tầng Application Framework bao gồm một tập các Services và các thành phần:

* Một tập hợp các View có khả năng kế thừa lẫn nhau, dùng để thiết kế giao diện ứng dụng như: gridview, tableview, linearlayout…
* Conten Provider: cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (như Contact) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
* Resource Manager: cung cấp truy xuất tới các tài nguyên không phải mã nguồn, như strings, graphics, layouts…
* Notification Manager: cho phép tất cả ứng dụng hiển thị các custom alert trên thanh status bar.
* Activity Manager: giúp quản lý vòng đời của một ứng dụng

##### Libraries

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android.

Một số các thư viện:

* Surface Libraries: quản lý việc truy xuất vào hệ thống hiển thị
* Media Framework: hỗ trợ chạy các file hình ảnh, âm thanh và video dựa trên PacketVideo’s OpenCORE.
* SQLite: hỗ trợ việc truy xuất dữ liệu trong các ứng dụng
* System C library: thư viện C chuẩn được nhúng vào các thiết bị chạy trên nền tảng Linux, giúp hỗ trợ việc lập trình và phát triển ứng dụng cũng như các thiết bị di động
* Freetype: hỗ trợ hiển thị phông chữ theo kiểu vector hay bitmap
* LibWebCore: thư viện cho trình duyệt web
* SGL: hỗ trợ đồ họa 2D
* 3D libraries: hỗ trợ đồ họa 3D dựa trên API của OpenGL ES 1.0

##### Tầng Android Runtime

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong một tiến trình riêng với máy ảo Dalvik. Máy ảo Dalvik là một biến thể của máy ảo Java được sửa đổi, bổ sung các công nghệ đặc trưng của thiết bị di động. Nó được xây dựng với mục đích làm cho các thiết bị di động có thể chạy nhiều máy ảo, tương đương với nhiều tiến trình một cách hiệu quả. Khi nhà phát triển viết một ứng dụng cho Android, anh ta thực hiện các đoạn mã trong môi trường Java. Sau đó, nó được biên dịch sang các bytecode của Java. Tuy nhiên để thực thi được ứng dụng này trên Android thì nhà phát triển dùng công cụ dx để chuyển đồi bytecode sang dạng dex bytecode (.dex). Định dạng này được thiết kế để phù hợp với các thiết bị hạn chế về bộ nhớ cũng như tốc độ xử lý. Ngoài ra máy ảo Dalvik sử dụng bộ nhân Linux để cung cấp các tính năng như Thread, Low-level Memory Management.

##### Tầng Linux Kernel

Hệ điều hành Android được xây dựng trên bộ nhân Linux 2.6 cho những dịch vụ hệ thống cốt lõi như: sercurity, memory management, process management, network stack, driver model. Bộ nhân này làm nhiệm vụ trung gian kết nối phần cứng thiết bị và phần ứng dụng.

Bằng cách sử dụng cơ chế JNI, Android cho phép các ứng dụng chạy trên máy ảo Dalvik có thể sử dụng những phương thức được viết bằng các ngôn ngữ cấp thấp như: C, C++, Assembly. Qua đó các nhà phát triển ứng dụng có thể xây dựng ứng dụng dựa trên các bộ thư viện viết bằng C, C++, Assembly nhằm tăng tốc độ thực thi của các ứng dụng hoặc sử dụng những tính năng mức thấp mà ngôn ngữ Java không hỗ trợ.

## Engine Text-To-Speech trên Android

### Giới thiệu về tính năng Text-To-Speech và Engine Text-To-Speech

Text-To-Speech (TTS) là một khái niệm đề cập đến khả năng chuyển đổi một văn bản đầu vào thành giọng nói nhân tạo giống tiếng nói của con người. Nó sẽ nhận đầu vào là một văn bản và chuyển đổi văn bản đó thành các đơn vị âm và tiếp tục chuyển các đơn vị âm thu được thành dạng sóng để được đầu ra là âm thanh.

Và tính năng Text-To-Speech đã được Google tích hợp và trang bị sẵn cho hệ điều hành Android từ phiên bản 1.6 Donut (API 4).

Với sự phát triển ngày càng mạnh mẽ của các thiết bị di động chạy hệ điều hành Android và sự đa dạng của các ứng dụng trên Android, tính năng Text-To-Speech ngày càng thể hiện rõ khả năng và tính hữu ích của mình trong nhiều trường hợp, đặc biệt là đối với các phần mềm từ điển được tích hợp tính năng Text-To-Speech. Ngoài ra, các nhà phát triển ứng dụng trên nền tảng Android đã xây dựng và đưa ra rất nhiều các sản phẩm tích hợp tính năng Text-To-Speech như các ứng dụng đọc nội dung hiển thị trên màn hình, ứng dụng đọc tin nhắn, số điện thoại đang gọi hay ứng dụng hướng dẫn chỉ đường bằng giọng nói… Tính năng Text-To-Speech đã khiến cho người dùng không còn quá lệ thuộc vào việc nhìn vào màn hình các thiết bị để đọc mà chỉ còn đơn giản là “nghe”.

Và để hỗ trợ tính năng Text-To-Speech, mỗi thiết bị di động chạy hệ điều hành Android được tích hợp một Engine Text-To-Speech tùy theo các hãng điện thoại khác nhau. Với các dòng thiết bị di động chạy hệ điều hành Android gốc của Google như các dòng Nexus, mặc định Engine TTS là Google Text-To-Speech Engine. Hay các thiết bị của hãng điện thoại HTC là Pico TTS…

Ngoài ra, các thiết bị chạy Android chấp nhận có thể cài đặt được Engine của hãng thứ 3. Vì thế người dùng có thể cài thêm các Engine TTS khác được phát triển với chất lượng tốt hơn và hỗ trợ nhiều ngôn ngữ hơn tùy theo nhu cầu của mình.

Có thể kể tên đến hai Engine TTS được đánh giá khá cao về chất lượng:

* IVONA Text-to-Speech HQ: là một Engine TTS miễn phí hỗ trợ 13 ngôn ngữ khác nhau.
* Classic Text To Speech Engine: là sản phẩm thương mại với giá $ 2.99 với hơn 40 giọng nam/nữ hỗ trợ hơn 25 ngôn ngữ khác nhau.

Mặc dù có rất nhiều Engine TTS được phát triển hỗ trợ rất nhiều ngôn ngữ nhưng cho đến thời điểm hiện tại, vẫn chưa có một Engine TTS nào hỗ trợ Tiếng Việt dù hiện nay có rất nhiều trung tâm đang nghiên cứu và phát triển các bộ tổng hơp tiếng nói.

### Cơ chế hoạt động của Engine Text-To-Speech

Một Engine Text-To-Speech hoạt động tương tự như một Service:

* Luôn luôn chạy ẩn ở chế độ nền và không có giao diện người dùng.
* Cung cấp các chức năng cho các ứng dụng khác sử dụng. Ở đây là chức năng tổng hợp văn bản thành tiếng nói.

Từ phiên bản Android 4.0 Ice Cream Sandwich (API 14), Google cung cấp cho các nhà phát triển ứng dụng Android một Service mới – TextToSpeechService. Với Service này, các nhà phát triển có thể xây dựng một engine Text-To-Speech độc lập với các ứng dụng. Các nhà phát triển ứng dụng không cần tích hợp các bộ tổng hợp tiếng vào trong các ứng dụng. Engine TTS sẽ hoạt động độc lập và luôn luôn “lắng nghe” các yêu cầu tổng hợp tiếng nói từ các ứng dụng.

Các nhà phát triển ứng dụng có tích hợp thêm tính năng tổng hợp tiếng chỉ cần sử dụng hai phương thức đã được Google cung cấp trong lớp TextToSpeech từ phiên bản API 4:

* ***public******int*** *speak (****String*** *text,* ***int*** *queueMode,* ***HashMap****<****String****,* ***String****> params)*
* ***public******int*** *synthesizeToFile (****String*** *text,* ***HashMap****<****String****,* ***String****> params,* ***String*** *filename)*

Khi các ứng dụng đó khởi tạo đối tượng TextToSpeech và gọi đến một trong hai phương thức *speak( )* hay *synthesizeToFile( )* ở trên, Engine TTS sẽ lấy các thông số đầu vào và chuyển cho bộ tổng hợp tiếng chuyển đoạn văn bản thành tiếng nói và phát ra âm thanh.

Các nhà phát triển sẽ sử dụng đối tượng TextToSpeech để trao đổi các thông tin với Engine Text-To-Speech. Và chúng ta cần cài đặt lại Service TextToSpeechService để sử dụng lại bộ bộ tổng hợp tiếng nói đã xây dựng cho các thông tin nhận được từ đối tượng TextToSpeech.

Từ phân tích trên, chúng ta có định hướng xây dựng Engine Text-To-Speech:

* Xây dựng các lớp kiểm tra dữ liệu Text-To-Speech
* Xây dựng Service – thành phần chính của Engine
* Đóng gói Engine

### Xây dựng Engine Text-To-Speech

#### Xây dựng các lớp kiểm tra cơ sở dữ liệu Text-To-Speech

Với Engine Text-To-Speech, cơ sở dữ liệu đầu vào cho bộ tổng hợp là phần quan trọng nhất để có thể tổng hợp ra âm thanh. Trước khi gọi đến phương thức tổng hợp tiếng, Engine cần phải kiểm tra đã có cơ sở dữ liệu chưa. Nếu chưa có sẽ đưa ra yêu cầu tải về cơ sở dữ liệu cho bộ tổng hợp tiếng.

Với các nhiệm vụ đó, chúng ta cần xây dựng các lớp sau:

* CheckVoiceData: là một lớp kế thừa lớp Activity có cài đặt các phương thức kiểm tra cơ sở dữ liệu của Engine Text-To-Speech
* DownloadVoiceData: sau khi lớp CheckVoiceData kiểm tra thấy chưa có cơ sở dữ liệu cần dùng cho bộ tổng hợp tiếng thì sẽ khởi tạo đối tượng của lớp này. Lớp DownloadVoiceData cài đặt các phương thức để điều hướng đến kho dữ liệu trực tuyến để tải về cơ sở dữ liệu cần dùng.
* VoiceDataInstallerReceiver: là một lớp kế thừa lớp BroadcastReceiver thực hiện chức năng thông báo khi có một cơ sở dữ liệu được tải về và gói ngôn ngữ tương ứng được thêm vào hệ thống.

Ngoài ra chúng ta có thể viết thêm lớp GetSampleText để thử nghiệm cơ sở dữ liệu tải về đã đúng hay chưa.

Sau khi thực hiện việc kiểm tra đầu vào cho bộ tổng hợp tiếng, chúng ta cần xây dựng Service nhận các thông tin từ ứng dụng, xử lý và chuyển đến cho bộ tổng hợp tiếng được xây dựng.

#### Xây dựng Service Text-To-Speech

Để xây dựng Engine Text-To-Speech, chúng ta cần cài đặt lại lớp TextToSpeechService. Lớp này là trung tâm xử lý của Engine, nó nhận đầu vào và kiểm tra, xử lý các tham số trước khi chuyển đến bộ tổng hợp tiếng đã được xây dựng.

Chúng ta cần phải viết đè lại các phương thức trừu tượng của lớp TextToSpeechService sao cho phù hợp với bộ tổng hợp tiếng chúng ta đã xây dựng:

* onIsLanguageAvailable(String, String, String)
* onLoadLanguage(String, String, String)
* onGetLanguage( )
* onSynthesizeText(SynthesisRequest, SynthesisCallback)
* onStop( )

Ba hàm đầu tiên chủ yếu là để quản lý ngôn ngữ và được sử dụng để kiểm tra xem engine có hỗ trợ ngôn ngữ đầu vào không.

Phương thức onSynthesizeText(SynthesisRequest, SynthesisCallback) là trung tâm cốt lõi của service TextToSpeechService và của Engine. Phương thức này sẽ yêu cầu service tổng hợp tiếng nói từ đoạn văn bản đầu vào với các thông tin được đóng gói trong đối tượng TextToSpeech ở ứng dụng.

Chúng ta sẽ lấy các thông tin từ đối tượng SynthesisRequest bằng các hàm:

* getLanguage( )
* getCountry( )
* getVariant( )
* getText( )

Sau đó sẽ gửi đoạn văn bản cần tổng hợp được lấy từ SynthesisRequest. getText( ) đến bộ tổng hợp tiếng để tổng hợp thành âm thanh và phát ra.

#### Đóng gói Engine Text-To-Speech

Sau khi thực hiện xong hai bước ở trên chúng ta cần khai báo trong Manifest của Engine. Thêm các quyền truy cập cần thiết cho Engine. Và đóng gói lại thành bản cài đặt.

## Xây dựng Engine Text-To-Speech sử dụng bộ Tổng hợp Tiếng Việt bằng phương pháp ghép nối

### Bộ Tổng hợp Tiếng Việt bằng phương pháp ghép nối

Viện Nghiên cứu Quốc tế về Thông tin đa phương tiện, Truyền thông và Ứng dụng (tên tiếng Anh viết tắt là MICA – Multimedia, Infomation, Communication and Application) được thành lập năm 2002 nhằm đáp ứng nhu cầu phát triển công nghệ thông tin ở Việt Nam.

Các lĩnh vực đang được nghiên cứu và phát triển ở Viện MICA:

* Xử lý các tín hiệu phức tạp (âm thanh, hình ảnh)
* Phát triển các ứng dụng đa phương tiện
* Xây dựng các thiết bị đo ảo và phân tán

Trong đó xử lý tiếng nói được Viện chú trọng nghiên cứu và phát triển trên các phương diện:

* Nhận dạng tiếng nói
* Hiểu ngữ nghĩa
* Tổng hợp tiếng nói

Trong quá trình thực tập và nghiên cứu tại Viện MICA, nhóm sinh viên đã nghiên cứu và xây dựng bộ Tổng hợp Tiếng Việt bằng phương pháp ghép nối chạy trên nền tảng Java. Nhóm đã đóng gói thành thư viện libVSSNUU.jar cùng với cơ sở dữ liệu cung cấp các API tiện dụng để sử dụng bộ Tổng hợp Tiếng Việt này.

|  |  |
| --- | --- |
| **package** | **Mica.SynthesisVNUU** |
| **public** **class** | **MainSynthesis** |
| **Public methods** | **Mô tả** |
| **boolean** loadDatabase (Context context); | Hàm nạp cơ sở dữ liệu từ tệp Text\_DB\_Creator.xml. Hàm trả về giá trị true nếu nạp cơ sở dữ liệu thành công. |
| **void** readTextToSpeech (String textInput); | Hàm tổng hợp cả một đoạn văn bản đầu vào. Đồng thời sẽ phát ra các câu nói tổng hợp được. |
| **void** saveTextToWavFile (String nameOfWav, String textInput); | Hàm lưu lại tệp âm thanh vừa được tổng hợp từ đoạn văn bản với tên do người dùng tự đặt: nameOfWav |

Bảng 4. API bộ Tổng hợp Tiếng Việt

### Xây dựng lớp xử lý văn bản đầu vào

Theo mô hình tổng hợp tiếng nói từ văn bản được trình bày ở trên, một văn bản đầu vào cần trải qua hai giai đoạn xử lý:

* Tổng hợp mức cao
  + Xử lý văn bản
  + Phân tích cách phát âm
  + Phân tích ngữ điệu
* Tổng hợp mức thấp

Nhiệm vụ đầu tiên của Engine Text-To-Speech là chuyển đổi văn bản đầu vào về một khuôn dạng thích hợp cho bộ tổng hợp tiếng. Trong giai đoạn này, tất cả các đặc tính như chữ cái, chữ số, chữ viết tắt… phải được chuyển đổi theo một khuôn dạng rõ ràng và đầy đủ.

Với bộ Tổng hợp Tiếng Việt được xây dựng ở trên, khuôn dạng dữ liệu đầu vào:

* Các chữ số sẽ được bỏ qua do không tìm được âm vị kép tương ứng
* Các từ có dấu câu đi kèm phía sau sẽ được bỏ qua do không tìm được âm vị kép tương ứng
* Các từ không phải Tiếng Việt cũng sẽ được bỏ qua

Chúng ta xây dựng lớp TextInputStandardized có nhiệm vụ để chuẩn hóa văn bản Tiếng Việt đầu vào:

* Chuyển các chữ số thành chữ, ví dụ chuyển số ‘9’ thành chữ “chín”, để bộ tổng hợp tiếng có thể tổng hợp được
* Chèn thêm ký tự trắng vào giữa từ và dấu câu để tách từ và dấu cấu đang liền nhau thành 2 phần riêng biệt trước khi đưa vào bộ tổng hợp tiếng. Ví dụ: “hiện-” chuyển thành “hiện -”
* Với các từ không phải Tiếng Việt (ở đây tạm thời xử lý với ngôn ngữ là tiếng Anh) sẽ chèn thêm ký tự trắng vào sau từng ký tự để bộ tổng hợp tiếng sẽ đánh vần từng ký tự. Ví dụ “English” sẽ được chuyển thành “E n g l i s h”

### Tích hợp bộ Tổng hợp Tiếng Việt

#### Xây dựng lớp kiểm tra cơ sở dữ liệu

Engine Text-To-Speech chúng ta xây dựng sẽ đóng gói cả cơ sở dữ liệu cho bộ Tổng hợp Tiếng Việt vào bên trong. Vì thế chúng ta sẽ bỏ qua không cần viết các lớp:

* DownloadVoiceData
* VoiceDataInstallerReceiver

Chúng ta sẽ xây dựng lớp CheckVoiceData. Do cơ sở dữ liệu đi kèm bên trong bản cài đặt của Engine nên đầu ra của lớp này luôn thông báo thành công.

Trong lớp CheckVoiceData, biến kết quả kiểm tra được gán:

**int** result = TextToSpeech.Engine.CHECK\_VOICE\_DATA\_PASS;

sẽ luôn trả kết quả thành công khi ứng dụng yêu cầu Engine kiểm tra cơ sở dữ liệu.

Chúng ta khởi tạo hai biến đối tượng ArrayList<String>:

ArrayList<String> available = **new** ArrayList<String>();

ArrayList<String> unavailable = **new** ArrayList<String>();

Hai biến này có nhiệm vụ hiển thị các ngôn ngữ mà Engine hỗ trợ mà đã được cài đặt cơ sở dữ liệu hay chưa được cài đặt cho phép người dùng lựa chọn. Ở đây, Engine chỉ hỗ trợ Tiếng Việt và đã có cơ sở dữ liệu đính kèm nên chúng ta khai báo:

available.add("vi");

unavailable.add("");

Sau đó chúng ta đưa hai biến đó vào Intent returnData:

returnData.putStringArrayListExtra("availableVoices", available);

returnData.putStringArrayListExtra("unavailableVoices", unavailable);

Và chuyển kết quả khởi tạo ra màn hình cài đặt của Engine:

setResult(result, returnData);

Kết thúc Activity CheckVoiceData:

finish();

Trong Android Manifest, chúng ta khai báo để Activity CheckVoiceData chạy khi khởi chạy Engine hay khi ứng dụng có yêu cầu Engine kiểm tra cơ sở dữ liệu:

<application >

<activity

android:name=*"edu.mica.vivatts2.CheckVoiceData"*

android:label=*"@string/app\_name"*

android:theme=*"@android:style/Theme.Translucent.NoTitleBar"*>

<intent-filter>

<action android:name=*"android.speech.tts.engine.CHECK\_TTS\_DATA"*/>

<category android:name=*"android.intent.category.DEFAULT"* />

</intent-filter>

</activity>

</application>

Dòng <action android:name=*"android.speech.tts.engine.CHECK\_TTS\_DATA"*/> khai báo Activity CheckVoiceData sẽ được gọi để kiểm tra cơ sở dữ liệu của Engine Text-To-Speech.

Bên cạnh đó chúng ta có thể viết thêm lớp GetSampleText để thử nghiệm kiểm tra Engine Text-To-Speech. Lớp GetSampleText là một Activity trong phương thức onCreate()khai báo một đối tượng Intent:

Intent returnData = **new** Intent();

Đối tượng này sẽ được đóng gói thêm một đoạn văn bản thử nghiệm:

returnData.putExtra("sampleText", getString(R.string.*vi\_sample*));

với *vi\_sample* là đoạn văn bản thử nghiệm được khai báo trong /res/values/strings.xml:

<string name=*"vi\_sample"*>Đây là chương trình chuyển văn bản thành tiếng nói phiên bản 2.</string>

Gửi đối tượng Intent này tới phương thức xử lý rồi kết thúc:

setResult(TextToSpeech.*LANG\_AVAILABLE*, returnData);

finish();

Chúng ta cũng cần khai báo trong AndroidManifest:

<application >

<activity

android:name=*"edu.mica.vivatts2.GetSampleText"*

android:label=*"@string/app\_name"*

android:theme=*"@android:style/Theme.Translucent.NoTitleBar"* >

<intent-filter>

<action android:name=*"android.speech.tts.engine.GET\_SAMPLE\_TEXT"* />

<category android:name=*"android.intent.category.DEFAULT"* />

</intent-filter>

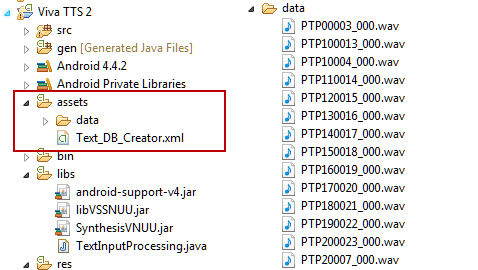
</activity>

</application>

Dòng <action android:name=*"android.speech.tts.engine.GET\_SAMPLE\_TEXT"* /> khai báo khi người dùng ấn vào “Listen to an example” thì engine sẽ khởi tạo đối tượng GetSampleText và phát ra đoạn tiếng nói thử nghiệm cho đoạn văn: “Đây là chương trình chuyển văn bản thành tiếng nói phiên bản 2.”.

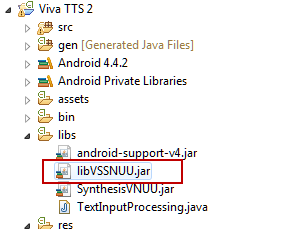
#### Sao chép thư viện và cơ sở dữ liệu

Sao chép cơ sở dữ liệu vào thư mục /assets. Cơ sở dữ liệu này bao gồm tệp Text\_DB\_Creator.xml và thư mục data chứa các tệp âm thanh định dạng wav.



Hình 3.4. Sao chép cơ sở dữ liệu cho bộ Tổng hợp Tiếng Việt

Sao chép thư viện libVSSNUU.jar vào thư mục /libs.



Hình 3.5. Sao chép thư viện bộ Tổng hợp Tiếng Việt

#### Sử dụng thư viện bộ Tổng hợp Tiếng Việt để xây dựng Service

Chúng ta sẽ xây dựng lớp TTSService kế thừa lớp TextToSpeechService. Lớp TextToSpeechService là một lớp trừu tượng có các phương thức trừu tượng:

* onIsLanguageAvailable(String, String, String)
* onLoadLanguage(String, String, String)
* onGetLanguage( )
* onSynthesizeText(SynthesisRequest, SynthesisCallback)
* onStop( )

Do engine chúng ta hiện tại chỉ hỗ trợ một ngôn ngữ là Tiếng Việt và có một cơ sở dữ liệu duy nhất nên chúng ta sẽ bỏ qua không viết đè ba phương thức:

* onIsLanguageAvailable(String, String, String)
* onLoadLanguage(String, String, String)
* onGetLanguage( )

Chúng ta cần viết đè phương thức onSynthesizeText(SynthesisRequest, SynthesisCallback) để gọi đến phương thức tổng hợp tiếng của bộ Tổng hợp Tiếng Việt libVSSNUU.jar.

Trước tiên cần import package:

**import** Mica.VSSNUU.MainSynthesis;

Sau đó ta khai báo và khởi tạo đối tượng MainSynthesis và nạp cơ sở dữ liệu từ tệp Text\_DB\_Creator.xml:

**final** MainSynthesis mainSynthesis = **new** MainSynthesis();

mainSynthesis.loadDataBase(getApplicationContext());

Trong phương thức onSynthesizeText( ), chúng ta lấy dữ liệu văn bản cần chuyển đổi từ đối tượng SynthesisRequest request:

String text = request.getText().toLowerCase();

Kế đó, chúng ta cần khởi tạo đối tượng TextInputStandardized để chuẩn hóa văn bản đầu vào:

TextInputStandardized textIS = **new** TextInputStandardized();

String modifiedText = textIS.standardize(text);

Với đoạn văn bản được chuẩn hóa, ta gọi tới hàm tổng hợp tiếng nói:

mainSynthesis.readTextToSpeech(modifiedText);

Trong AndroidManifest ta cần khai báo lớp TTSService là lớp cài đặt của Text-To-Speech Service:

<application >

<service

android:name=*"edu.mica.vivatts2.TTSService"*

android:label=*"@string/app\_name"* >

<intent-filter>

<action android:name=*"android.intent.action.TTS\_SERVICE"* />

<category android:name=*"android.intent.category.DEFAULT"* />

</intent-filter>

</service>

</application>

Khi đó một ứng dụng gọi tới Engine để tổng hợp âm thanh thì đoạn âm thanh tương ứng với đoạn văn bản sẽ được tổng hợp và phát ra thành tiếng nói.

### Xây dựng và đóng gói Engine Tổng hợp Tiếng Việt

#### Thêm lớp Activity VivaTTS

**package** edu.mica.vivatts2;

**import** android.app.Activity;

**import** android.os.Bundle;

**public** **class** VivaTTS **extends** Activity {

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

finish();

}

}

Trong AndroidManifest:

<application >

<activity

android:name=*"edu.mica.vivatts2.VivaTTS"*

android:enabled=*"true"*

android:hasCode=*"false"*

android:label=*"@string/app\_name"*

android:theme=*"@android:style/Theme.Translucent.NoTitleBar"* >

<intent-filter>

<action android:name=*"android.intent.action.START\_TTS\_ENGINE"* />

<category android:name=*"android.intent.category.tts\_engine.TTS"* />

<category android:name=*"android.intent.category.tts\_lang.vi"* />

</intent-filter>

</activity>

</application>

#### Thêm lớp SettingsProvider

**package** edu.mica.vivatts2.providers;

**import** java.io.File;

**import** android.content.ContentProvider;

**import** android.content.ContentValues;

**import** android.database.Cursor;

**import** android.database.MatrixCursor;

**import** android.net.Uri;

**import** android.os.Environment;

**public** **class** SettingsProvider **extends** ContentProvider {

**private** **class** SettingsCursor **extends** MatrixCursor {

**private** String settings;

**public** SettingsCursor(String[] columnNames) {

**super**(columnNames);

}

**public** **void** putSettings(String settings) {

**this**.settings = settings;

}

@Override

**public** **int** getCount() {

**return** 1;

}

@Override

**public** String getString(**int** column) {

**return** settings;

}

}

@Override

**public** **int** delete(Uri uri, String selection, String[] selectionArgs) {

**return** 0;

}

@Override

**public** **int** update(Uri uri, ContentValues values, String selection, String[] selectionArgs) {

**return** 0;

}

@Override

**public** String getType(Uri uri) {

**return** **null**;

}

@Override

**public** Uri insert(Uri uri, ContentValues values) {

**return** **null**;

}

@Override

**public** **boolean** onCreate() {

**return** **true**;

}

@Override

**public** Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs,String sortOrder) {

**final** File dataPath = Environment.*getExternalStorageDirectory*();

**final** String[] dummyColumns = {

"", ""

};

**final** SettingsCursor cursor = **new** SettingsCursor(dummyColumns);

cursor.putSettings(dataPath.getPath());

**return** cursor;

}

}

Trong AndroidManifest:

<application >

<provider

android:name=*"edu.mica.vivatts2.providers.SettingsProvider"*

android:authorities=*"edu.mica.vivatts2.providers.SettingsProvider"* />

</application>

#### Khai báo thêm quyền người dùng

Bộ Tổng hợp Tiếng Việt sử dụng thẻ nhớ ngoài làm nới lưu trữ tệp âm thanh tổng hợp được nên cần quyền truy cập sử dụng thẻ nhớ. Chúng ta khai báo trong AndroidManifest:

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*/>

<uses-permission android:name=*"android.permission.READ\_EXTERNAL\_STORAGE"*/>

# TRIỂN KHAI VÀ ĐÁNH GIÁ ENGINE VIVA TTS TRÊN SMARTPHONE ANDROID

## Giới thiệu chương

Engine Viva TTS sử dụng bộ tổng hợp Tiếng Việt do MICA phát triển đã được xây dựng như đã trình bày ở chương trước. Kết quả thu được là tệp cài đặt định dạng apk dùng để cài lên các thiết bị chạy hệ điều hành Android hay có môi trường giả lập hệ điều hành Android.

Sản phẩm thu được đã được cài đặt chạy thử trên Smartphone chạy hệ điều hành Android và thử nghiệm với tính năng TalkBack của thiết bị.

Chương này sẽ trình bày công việc triển khai và thử nghiệm engine bằng tính năng TalkBack trên Smartphone chạy hệ điều hành Android cùng với đánh giá về mặt tính năng và hiệu năng của Engine.

## Triển khai Engine tổng hợp Tiếng Việt trên Smartphone Android

### Môi trường, thiết bị thử nghiệm

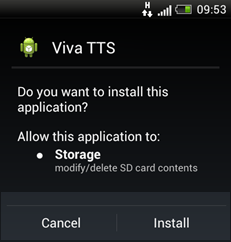
Thiết bị dùng để thử nghiệm và đánh giá Engine Viva TTS được xây dựng là Smartphone HTC One V với cấu hình như sau:

* Tốc độ chip xử lý: 1GHz
* Hệ điều hành: Android 4.0.3 với HTC Sense 4
* Bộ nhớ trong: 4GB
* Ram: 512 MB

Với cấu hình tầm trung của HTC One V, việc cài đặt và thử nghiệm Engine Viva TTS hoàn toàn ổn định.

### Cài đặt và thiết lập Engine Text-To-Speech

Cài đặt file apk của engine: **Viva TTS.apk**. Xuất hiện màn hình yêu cầu xác nhận quyền truy cập của engine. Ở đây yêu cầu quyền cho phép engine sử dụng thẻ nhớ ngoài SD Card làm nơi lưu trữ dữ liệu:

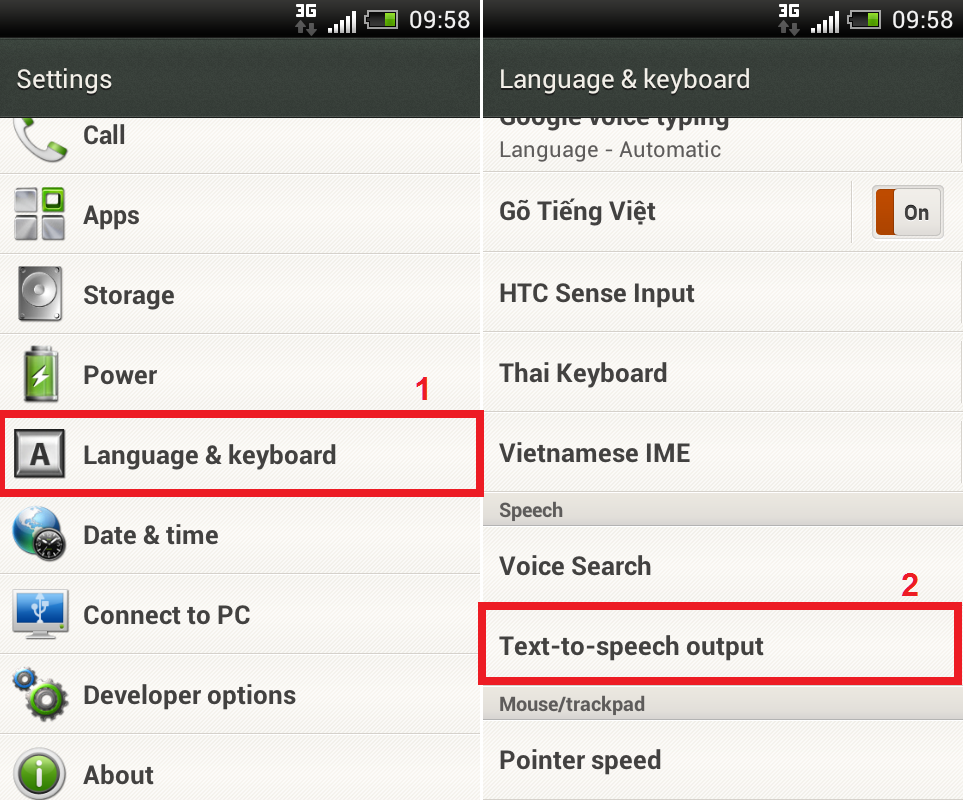


Hình 4.1. Màn hình yêu cầu quyền truy cập của Viva TTS

Chọn Install để xác nhận cài đặt engine.

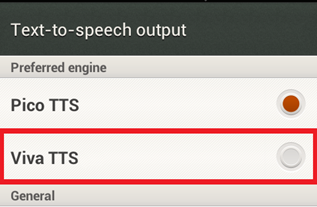
Sau khi cài đặt Engine Viva TTS lên Smartphone, cần thiết lập để Engine Viva TTS là Engine Text-To-Speech ngầm định của hệ thống:

* Vào **Android Setting → Language & keyboard** (có thể khác nhau tùy theo dòng điện thoại) **→ Text-to-speech output**



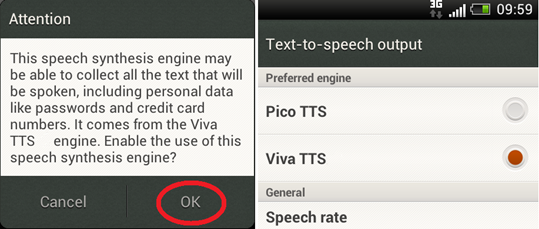
Hình 4.2. Thiết lập sử dụng Engine Viva TTS (1)

* Lựa chọn Viva TTS.



Hình 4.3. Thiết lập sử dụng Engine Viva TTS (2)

* Xuất hiện hộp thoại cảnh báo, ấn OK để đồng ý lựa chọn Engine Viva TTS là Engine Text-To-Speech ngầm định của hệ thống



Hình 4.4. Thiết lập sử dụng Engine Viva TTS (3)

## Thử nghiệm Engine tổng hợp Tiếng Việt bằng tính năng Talkback

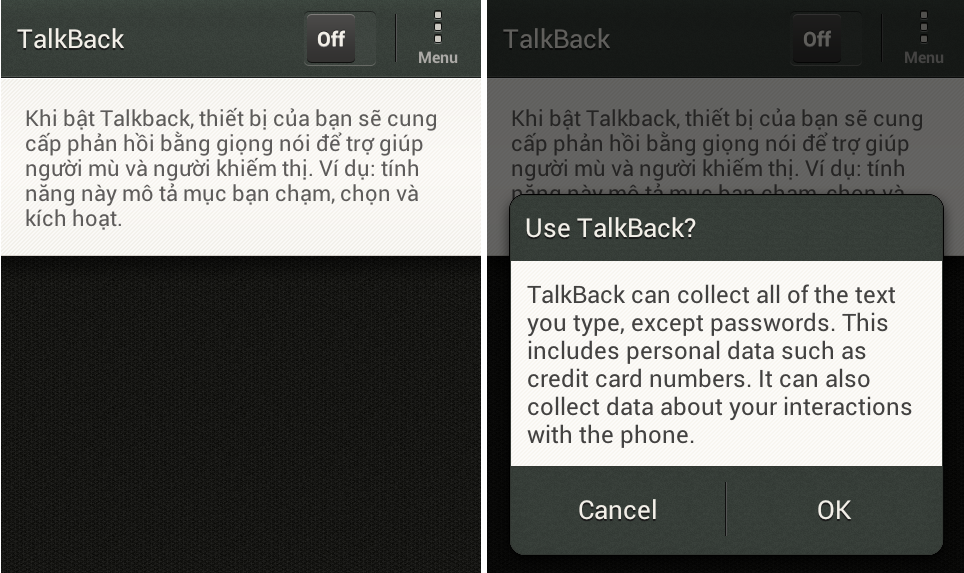
### Ứng dụng Google TalkBack

TalkBack là dịch vụ trình đọc màn hình được cài đặt sẵn do Google cung cấp giúp người dùng khiếm thị tương tác với các thiết bị của họ. Ứng dụng này sử dụng phản hồi bằng giọng nói và báo rung để mô tả kết quả của tác vụ (ví dụ như mở ứng dụng), điều hướng các thiết bị, mô tả những gì người dùng chạm vào hay kích hoạt cũng như mô tả các sự kiện xảy ra (ví dụ như các thông báo).

Đây là một ứng dụng hệ thống đã được cài đặt sẵn trên hầu hết các thiết bị và được cập nhật ngay khi có phiên bản mới.

***Bật tính năng TalkBack***

Vào **Android Setting > Accessibility > TalkBack**. Trượt nút TalkBack đến vị trí **On** để bật TalkBack.

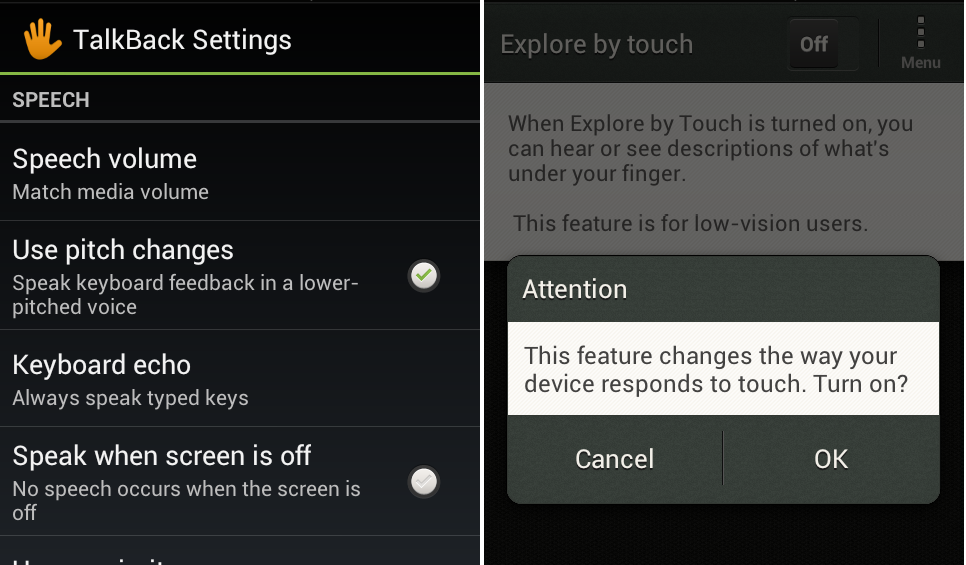


Hình 4.5. Bật TalkBack

Tiếp đó sẽ hiện ra một hộp thoại. Chạm vào OK để xác nhận bật tính năng TalkBack. Khi bật TalkBack, tính năng này sẽ cung cấp cho người dùng thông tin phản hồi bằng giọng nói, TalkBack sẽ giới thiệu các khả năng của TalkBack:

* Quan sát thao tác của bạn: nhận thông báo khi bạn đang tương tác với một ứng dụng
* Truy xuất nội dung trong cửa sổ: kiểm tra nội dung của cửa sổ đang tương tác
* Bật Khám phá bằng cách chạm: mục đã chạm sẽ được nói to và bạn có thể khám phá màn hình bằng cử chỉ. Thiết bị sẽ mô tả từng mục mà ngón tay di chuyển qua trên màn hình
* Bật trợ năng web nâng cao: tập lệnh có thể được cài đặt để làm cho nội dung ứng dụng dễ truy cập hơn
* Quan sát văn bản bạn nhập: bao gồm dữ liệu cá nhân chẳng hạn như số thẻ tín dụng và mật khẩu

Ngay khi bật TalkBack, phản hồi bằng giọng nói sẽ bắt đầu xuất hiện ngay lập tức. Khi người dùng điều hướng thiết bị, TalkBack mô tả tác vụ đang thực hiện và cảnh báo về thông báo và các trường hợp khác.



Hình 4.6. TalkBack Settings và tính năng Khám phá bằng cách chạm

Với khả năng phát ra giọng nói phản hồi có sử dụng Engine Text-To-Speech, TalkBack là một ứng dụng lý tưởng để thử nghiệm Engine Viva TTS được xây dựng trong quá trình làm đồ án tốt nghiệp.

### Các trường hợp thử nghiệm

|  |  |  |
| --- | --- | --- |
| **STT** | **Trường hợp thử nghiệm** | **Văn bản đầu vào** |
| 1 | Ấn nút nguồn để tắt màn hình | Màn hình tắt |
| 2 | Ấn nút nguồn để bật màn hình | 11:54 |
| 3 | Mở khóa thiết bị | Đã mở khóa thiết bị |
| 4 | Chạm vào một biểu tượng Clock Weather trên Laucher | Hình ảnh 73. Chưa được gán nhãn, Hoàng Mai, Th5, thg 5 29, Hoàng Mai, chưa có dữ liệu thời tiết sẵn sàng, chưa có dữ liệu thời tiết sẵn sàng |
| 5 | Vào danh sách ứng dụng | Danh sách ứng dụng |
| 6 | Lật trang danh sách ứng dụng | App page 4 of 6 |
| 7 | Truy cập một trang web tiếng việt | Nội dung web |
| 8 | Chạm vào một đoạn văn trong trang web | Tổng thống Obama cảnh báo sẵn sàng đáp trả sự gây hấn của Trung Quốc link |
| 9 | Vào danh bạ điện thoại | Danh bạ > Hộp chỉnh sửa Tìm số liên lạc |
| 10 | Vào một tên trong danh bạ | Hình ảnh 52. Chưa được gán nhãn, Hình ảnh 83. Chưa được gán nhãn, Hành động, Nút Gọi số di động 0983739609, Chính, Hình ảnh 24. Chưa được gán nhãn, Tùy chọn, Chỉnh sửa, Sửa số liên lạc, Nhạc chuông, Mặc định, Chặn người gọi đến, Tắt |
| 11 | Cham ngày trong ứng dụng Lịch Vạn Niên | 29, Trước phái đẹp và âm nhạc, thời gian sẽ trở thành vô nghĩa. (A.Xmit) |
| 12 | Chạm vào thanh thông báo | 12:19, Pin 100 phần trăm |
| 13 | Vào cài đặt | Cài đặt > Hiển thị mục từ 1 đến 10 trong tổng số 11 mục |
| 14 | Chạm vào Nghe thử ví dụ trong thiết lập Engine | Đây là chương trình chuyển văn bản thành tiếng nói. |
| 15 | Chạm vào các phím số điện thoại | 123456789 |
| 16 | Gõ đoạn văn bản | vnexpress.net |
| 17 | Chụp màn hình | Ảnh chụp màn hình |

Bảng 5. Các trường hợp thử nghiệm Engine Viva TTS

### Kết quả thử nghiệm

Do kết quả đánh giá đầu ra là âm thanh nên không thể hiện được bằng hình ảnh. Bảng dưới đây là kết quả thực nghiệm cho các trường hợp thử nghiệm đưa ra ở trên:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Văn bản đầu vào** | **Kết quả thu được** | | |
| **Đọc** | **Không đọc** | **Đọc không hoàn chỉnh** |
| 1 | Màn hình tắt | x |  |  |
| 2 | 11:54 | x |  |  |
| 3 | Đã mở khóa thiết bị | x |  |  |
| 4 | Hình ảnh 73. Chưa được gán nhãn, Hoàng Mai, Th5, thg 5 29, Hoàng Mai, chưa có dữ liệu thời tiết sẵn sàng, chưa có dữ liệu thời tiết sẵn sàng |  |  | x |
| 5 | Danh sách ứng dụng | x |  |  |
| 6 | App page 4 of 6 |  |  | x |
| 7 | Nội dung web |  |  | x |
| 8 | Tổng thống Obama cảnh báo sẵn sàng đáp trả sự gây hấn của Trung Quốc link |  |  | x |
| 9 | Danh bạ > Hộp chỉnh sửa Tìm số liên lạc | x |  |  |
| 10 | Hình ảnh 52. Chưa được gán nhãn, Hình ảnh 83. Chưa được gán nhãn, Hành động, Nút Gọi số di động 0983739609, Chính, Hình ảnh 24. Chưa được gán nhãn, Tùy chọn, Chỉnh sửa, Sửa số liên lạc, Nhạc chuông, Mặc định, Chặn người gọi đến, Tắt | x |  |  |
| 11 | 29, Trước phái đẹp và âm nhạc, thời gian sẽ trở thành vô nghĩa. (A.Xmit) |  |  | x |
| 12 | 12:19, Pin 100 phần trăm | x |  |  |
| 13 | Cài đặt > Hiển thị mục từ 1 đến 10 trong tổng số 11 mục |  |  | x |
| 14 | Đây là chương trình chuyển văn bản thành tiếng nói. | x |  |  |
| 15 | 123456789 | x |  |  |
| 16 | vnexpress.net | x |  |  |
| 17 | Ảnh chụp màn hình | x |  |  |

Bảng 6. Kết quả thử nghiệm Engine Viva TTS

Ý nghĩa phần kết quả thu được:

* **Đọc:** thiết bị phát ra âm thanh đúng với văn bản đầu vào
* **Không đọc:** thiết bị không phát ra âm thanh mặc dù có văn bản đầu vào
* **Đọc không hoàn chỉnh:** thiết bị phát ra âm thanh không đầy đủ so với văn bản đầu vào

## Đánh giá

### Tính năng

Với kết quả thử nghiệm bằng ứng dụng TalkBack, Engine Viva TTS đã đáp ứng các tính năng của một Engine Text-To-Speech:

* Phát ra giọng nói với văn bản đầu vào
* Giọng nói tổng hợp được có chất lượng dễ nghe, không trúc trắc

Tuy nhiên, trong một số trường hợp tiếng nói tổng hợp được không đầy đủ do hiện tại, bộ tổng hợp Tiếng Việt do MICA phát triển chỉ hỗ trợ các đoạn văn bản đầu vào là thuần việt. Và có những từ tiếng việt không được đọc do cơ sở dữ liệu đơn vị âm cho bộ tổng hợp tiếng việt còn chưa đầy đủ nên chưa thể tổng hợp được tất cả các âm tiết tiếng việt.

### Hiệu năng

Với các dữ liệu văn bản đầu vào được thử nghiệm, Engine Viva TTS đã tổng hợp được thành tiếng nói trong thời gian cho phép (không quá lâu khiến người dùng tưởng tính năng này bị treo) và sử dụng ít dung lượng bộ nhớ RAM (5MB~10MB).

## Kết chương

Chương này đã đưa ra các kết quả thực nghiệm cho Engine Viva TTS được xây dựng trong quá trình thực hiện đồ án. Kết quả thực nghiệm khá khả quan.

# KẾT LUẬN

## Kết luận chung

Đồ án tốt nghiệp với đề tài ***“Tìm hiểu và xây dựng engine tổng hợp tiếng nói từ văn bản trên android”*** đã cơ bản hoàn thành các mục tiêu đặt ra:

* Tìm hiểu lý thuyết tiếng nói và các đặc điểm ngữ âm của Tiếng Việt
* Tìm hiểu lý thuyết tổng hợp tiếng nói, phương pháp tổng hợp tiếng nói và mô hình tổng hợp tiếng nói từ văn bản
* Tìm hiểu phương pháp xây dựng bộ tổng hợp Tiếng Việt bằng phương pháp ghép nối
* Tìm hiểu cơ chế hoạt động và quy trình xây dựng một Engine Text-To-Speech
* Xây dựng một Engine Text-To-Speech sử dụng modul tổng hợp Tiếng Việt bằng phương pháp ghép nối
* Thử nghiệm Engine Text-To-Speech với các ứng dụng sử dụng tính Text-To-Speech của Android. Đánh giá Engine về mặt tính năng và hiệu năng

Đồ án đã giải quyết được các vấn để về tổng hợp tiếng nói theo phương pháp ghép nối và xây dựng Engine Viva TTS sử dụng thư viện libVSSNUU với cơ sở dữ liệu tổng hợp tiếng Việt của viện MICA xây dựng. Engine đã tạm chạy ổn với các chức năng chính của một Engine tổng hợp Tiếng Việt trên nền tảng Android.

## Những khó khăn gặp phải

Trong quá trình thực hiện đề tài, tài liệu về phương pháp xây dựng Engine khá là hạn chế nên chưa thể trình bày một cách chi tiết về việc xây dựng Engine nói chung và Engine Text-To-Speech nói riêng trên nền tảng Android.

Do lần đầu lập trình và xây dựng một Engine nên còn khá nhiều sai sót. Bộ dữ liệu thử nghiệm chưa đầy đủ nên chưa có một văn bản chuẩn hóa văn bản đầu vào cho bộ tổng hợp Tiếng Việt.

## Định hướng phát triển

Đề tài ***“Tìm hiểu và xây dựng engine tổng hợp tiếng nói từ văn bản trên android”*** trong khuôn khổ đồ án này chỉ dừng lại ở nghiên cứu và xây dựng một engine chưa thực sự đáp ứng đầy đủ yêu cầu về một Engine.

Có nhiều hướng mở để tiếp tục phát triển đề tài:

* Nghiên cứu thuật toán HMM để xây dựng bộ tổng hợp tiếng nói mới
* Với phương pháp tổng hợp ghép nối, xây dựng bộ cơ sở dữ liệu đầy đủ với 389 âm vị kép
* Tách phần xử lý tổng hợp tiếng nói của bộ tổng hợp Tiếng Việt với phần cơ sở dữ liệu, làm tiền đề cho việc xây dựng thêm nhiều bộ cơ sở dữ liệu tổng hợp tiếng.

# TÀI LIỆU THAM KHẢO

1. Jacob Benesty and M. Mohan Sondhi and Yiteng Huang, *Springer handbook of speech processing*, Springer, 2008
2. Mai Ngọc Chừ, *Cơ sở ngôn ngữ học và tiếng Việt*, Nxb Giáo dục, 1997
3. William A. Jr. Kretzschmar, *The Linguistics of Speech,* Cambridge University Press, 2009
4. Trịnh Văn Loan, *Tổng hợp TiếngViệt bằng giải thuật TD-PSOLA*
5. Trần Anh Bình và Đặng Ngọc Thiệu, *Tổng hợp tiếng nói trên hệ điều hành Android*, 2012
6. Đỗ Văn Thảo. *Lựa Chọn Đơn vị Âm Không Đồng Nhất Trong Tổng Hợp Tiếng Nói Tiếng Việt*, 2011.
7. Nguyễn Duy Bình. *Tương Tác Với Điện Thoại Thông Minh Bằng Tiếng Nói*, 2013
8. Nguyễn Phú Bình, *Slide: Xử lý tiếng nói*
9. *Android SDK: Using the Text to Speech Engine,* <http://code.tutsplus.com/tutorials/android-sdk-using-the-text-to-speech-engine--mobile-8540>
10. *Âm vị và các hệ thống âm vị Tiếng Việt,* <http://ngonngu.net/index.php?p=64>
11. *Ngữ âm tiếng Việt*, <http://ngnnghc.wordpress.com>
12. *Âm tiết Tiếng Việt,* <http://vnlp.net/?page_id=9>
13. *Human Voice,* <http://en.wikipedia.org/wiki/Human_voice>
14. *What Is a Diphone?,* <http://www.wisegeek.com/what-is-a-diphone.htm>
15. <https://support.google.com/talkback/>

1. Nguyễn Duy Bình, *Tương Tác Với Điện Thoại Thông Minh Bằng Tiếng Nói*, 21. [↑](#footnote-ref-1)
2. Trịnh Văn Loan, *Tổng Hợp Tiếng Việt Bằng Giải Thuật TD-PSOLA*, 46. [↑](#footnote-ref-2)
3. “Hệ Thống Âm vị.” [↑](#footnote-ref-3)
4. Trịnh Văn Loan, *Tổng Hợp Tiếng Việt Bằng Giải Thuật TD-PSOLA*, 49. [↑](#footnote-ref-4)
5. Đỗ Văn Thảo, *Lựa Chọn Đơn vị Âm Không Đồng Nhất Trong Tổng Hợp Tiếng Nói Tiếng Việt*, 3. [↑](#footnote-ref-5)
6. Đỗ Văn Thảo, *Lựa Chọn Đơn vị Âm Không Đồng Nhất Trong Tổng Hợp Tiếng Nói Tiếng Việt*, 11. [↑](#footnote-ref-6)
7. Đỗ Văn Thảo, *Lựa Chọn Đơn vị Âm Không Đồng Nhất Trong Tổng Hợp Tiếng Nói Tiếng Việt*, 23. [↑](#footnote-ref-7)