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

LỰA CHỌN ĐƠN VỊ ÂM KHÔNG ĐỒNG NHẤT TRONG TỔNG HỢP TIẾNG NÓI TIẾNG VIỆT

Sinh viên thực hiện : **Đỗ Văn Thảo**

Lớp: CNPM – K51

Giáo viên hướng dẫn: TS. **Trần Đỗ Đạt**

HÀ NỘI 05-2011

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Đỗ Văn Thảo

Điện thoại liên lạc: 01226397323 Email: thaodv.bkit@gmail.com

Lớp: Công nghệ phần mềm K51 Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Trung tâm nghiên cứu Mica – Trường Đại học Bách Khoa Hà Nội.

Thời gian làm ĐATN: Từ ngày 21/02/2011 đến 28/05/2011

2. Mục đích nội dung của ĐATN

Tìm hiểu phương pháp tối ưu hóa lựa chọn đơn vị trong tổng hợp tiếng nói tiếng Việt và cài đặt thử nghiệm.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu các vấn đề trong tổng hợp mức thấp của hệ thống tổng hợp tiếng nói và xác định vấn đề mình tập trung giải quyết.
- Đề xuất phương pháp chọn lựa đơn vị âm tối ưu và thực thi, đánh giá phương pháp.
- Tổ chức cơ sở dữ liệu cho tìm kiếm đơn vị âm.

4. Lời cam đoan của sinh viên:

Tôi – *Đỗ Văn Thảo* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Trần Đỗ Đạt* .

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 05 năm 2011

Tác giả ĐATN

Đỗ Văn Thảo

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

Hà Nội, ngày 28 tháng 05 năm 2011

Giáo viên hướng dẫn

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong những năm gần đây, các phương thức giao tiếp người máy được chú trọng nghiên cứu và phát triển. Các phương thức giao tiếp mới như qua cử chỉ, ánh mắt, tiếng nói hay suy nghĩ của con người đều hứa hẹn giúp con người nâng cao sự thuận tiện trong giao tiếp với máy. Tổng hợp tiếng nói là một bài toán áp dụng trong lĩnh vực này. Trong đó, con người sẽ được nghe máy đọc những đoạn văn bản mong muốn. Với mong muốn tìm hiểu và phát triển bộ tổng hợp tiếng nói cho tiếng Việt, đồ án đã chọn lĩnh vực tổng hợp tiếng nói làm hướng nghiên cứu. Đồ án tập trung vào phần tổng hợp mức thấp trong tổng hợp tiếng nói, cụ thể là quá trình tìm kiếm và lựa chọn đơn vị âm. Với mong muốn cải thiện chất lượng tiếng nói tổng hợp, thuật toán lựa chọn đơn vị không đồng nhất được sử dụng với mục đích chọn ra đơn vị âm dài nhất, giảm thiểu số điểm ghép nối.

Trong đồ án này, tác giả tập trung đi tìm hiểu bài toán tổng hợp tiếng nói nói chung và áp dụng cho tiếng Việt nói riêng. Sau đó, đồ án tập trung vào vấn đề tìm kiếm và lựa chọn đơn vị âm trong tổng hợp ghép nối. Phương pháp lựa chọn đơn vị âm không đồng nhất được đề xuất và áp dụng cho tiếng Việt. Tác giả cũng tiến hành cài đặt và đánh giá hiệu quả của phương pháp. Từ đó đưa ra hướng phát triển tiếp theo cho đồ án.

LỜI CẢM ƠN

Trước hết, em xin được gửi lời cảm ơn chân thành tới các thầy cô giáo trong trường Đại học Bách Khoa Hà Nội cũng như các thầy cô trong Viện Công nghệ thông tin và truyền thông đã truyền dạy cho em những kiến thức và kinh nghiệm quý giá trong suốt quá trình học tập tu dưỡng trong suốt 5 năm qua.

Em xin được gửi lời cảm ơn tới TS. Trần Đỗ Đạt – Cán bộ nghiên cứu, Trung tâm nghiên cứu Mica và ThS. Nguyễn Thị Thu Trang - Giảng viên bộ môn Công nghệ phần mềm, Viện Công nghệ thông tin và truyền thông, trường Đại học Bách Khoa Hà Nội đã hết lòng giúp đỡ, hướng dẫn và chỉ dạy tận tình trong quá trình em làm đồ án tốt nghiệp.

Em cũng bày tỏ lòng biết ơn tới trung tâm nghiên cứu Mica đã tạo điều kiện về cơ sở vật chất cho em trong quá trình học tập và nghiên cứu.

Em cũng muốn gửi lời cảm ơn tới tập thể lớp Công nghệ phần mềm K51 đã tạo một môi trường thi đua học tập lành mạnh, tạo điều kiện cho sự phát triển của các thành viên trong lớp.

Cuối cùng, em xin được gửi lời cảm ơn chân thành tới gia đình, bạn bè đã quan tâm, động viên, đóng góp ý kiến và giúp đỡ trong quá trình học tập, nghiên cứu và hoàn thành đồ án tốt nghiệp.

Hà Nội, ngày 27 tháng 05 năm 2011

Đỗ Văn Thảo

Lớp CNPM – K51

Viện CNTT & TT – ĐH Bách Khoa HN

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

THTN	**T**ổng **h**ợp **t**iếng **n**ói
PSOLA	**P**itch **S**ynchronous **O**ver**l**ap and **A**dd
FFT	**F**ast **F**ourier **T**ransform
IFFT	**I**nverse **F**ast **F**ourier **T**ransform
CSDL	**C**ơ **s**ở **d**ữ **l**iệu
XML	e**X**tensible **M**arkup **L**anguage
JNI	**J**ava **N**ative **I**nterface
HT	**H**ệ **T**hống

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ

ĐẶT VẤN ĐỀ

Máy vi tính là một trong những phát minh ảnh hưởng nhiều nhất tới đời sống con người trong thế kỉ vừa qua. Với máy vi tính, con người có thể làm được nhiều việc mà trước đó người ta không nghĩ tới. Lĩnh vực tương tác người máy ra đời giúp con người dễ dàng tương tác hơn với máy tính. Trước đây, con người có thể tương tác với máy tính bằng mắt, bằng tay thông qua các thiết bị như bàn phím, chuột, màn hình. Càng ngày yêu cầu về tính tiện dụng trong tương tác của con người ngày càng cao. Các hình thức tương tác mới ra đời như tương tác bằng cử chỉ, giọng nói…

Tổng hợp tiếng nói là một lĩnh vực quan trọng trong giao tiếp người máy và được nghiên cứu, phát triển từ khá sớm trên thế giới. Tại Việt Nam đã có nhiều bộ tổng hợp tiếng nói được phát triển như bộ tổng hợp "*Sao Mai*" của trung tâm Sao Mai, "*Hoa Súng*" của trung tâm nghiên cứu Mica – ĐH BKHN, "*Tiếng nói phương Nam*" của ĐHQG-TPHCM. Tuy nhiên, các bộ tổng hợp trên vẫn còn cần cải thiện hoặc về chất lượng tiếng nói, hoặc về kích thước CSDL. Với mong muốn xây dựng một bộ tổng hợp tiếng nói có chất lượng tốt, kích thước CSDL không quá lớn, đồ án này quyết định chọn phương pháp lựa chọn đơn vị không đồng nhất để tìm hiểu và áp dụng vào chương trình tổng hợp tiếng nói. Đồ án được thực hiện tại trung tâm nghiên cứu quốc tế MICA Trong quá trình thực hiện đồ án, tác giả đã được tiếp cận những kiến thức bổ ích từ các cán bộ nghiên cứu của trung tâm phục vụ cho quá trình làm đồ án.

Trong các bộ tổng hợp, tiếng nói được tổng hợp bằng cách ghép nối các đơn vị âm lại với nhau,các đơn vị âm này là cùng một loại duy nhất, ví dụ cùng là âm vị kép, cùng là bán âm tiết … Đây là cách tiếp cận lựa chọn đơn vị đồng nhất – tức là chỉ dùng một loại đơn vị âm. Và kích thước CSDL dùng trong các bộ tổng hợp này thường bị giới hạn.

Trong vài năm trở lại đây, sự phát triển của khoa học công nghệ đã nâng cao khả năng lưu trữ và xử lý của máy tính. Kích thước CSDL của bộ tổng hợp tiếng nói có thể được tăng lên để cải thiện chất lượng tiếng nói tổng hợp. Phương pháp lựa chọn đơn vị không đồng nhất được thử nghiệm. Nhiều loại đơn vị âm được sử dụng với tư tưởng sử dụng đơn vị âm càng dài thì chất lượng càng cao. Với mỗi ngôn ngữ khác nhau, phương pháp này được áp dụng theo các cách thức khác nhau và đồ án này tập trung áp dụng phương pháp này cho ngôn ngữ tiếng Viêt.

Đồ án này bao gồm bốn chương:

- Chương một: giới thiệu về tổng hợp tiếng nói và những vấn đề trong tổng hợp tiếng nói.
- Chương hai: trình bày chi tiết về bài toán lựa chọn đơn vị trong tổng hợp ghép nối.
- Chương ba: mô tả chi tiết hệ thống tổng hợp tiếng Việt theo thuật toán lựa chọn đơn vị không đồng nhất.
- Chương bốn: đánh giá kết quả đạt được và chưa được, đồng thời đưa ra hướng phát triển tiếp theo.

Chương 1. Tổng hợp tiếng nói

Trong chương này, luận văn sẽ giới thiệu:

- *Tổng quan về bài toán tổng hợp tiếng nói.*
- *Các vấn đề cần giải quyết trong tổng hợp ghép nối.*
- *Vấn đề cụ thể đồ án tập trung tìm hiểu và giải quyết.*

1.1 Tổng quan về bài toán tổng hợp tiếng nói

Tổng hợp tiếng nói là quá trình tạo ra tiếng nói nhân tạo của người từ văn bản đầu vào. Đây là lĩnh vực nghiên cứu có tính ứng dụng thực tiễn cao nên được quan tâm trên thế giới và Việt Nam. Ứng dụng của tổng hợp tiếng nói có thể dễ dàng thấy trong nhiều hệ thống, như hệ thống hỗ trợ đọc văn bản cho người khuyết tật, hệ thống trả lời tự động tại các tổng đài hay robot, hệ thống chỉ đường trong các phương tiện vận tải... Có thể phân chia các hệ thống tổng hợp tiếng nói theo phương pháp tiếng nói được tổng hợp gồm ba loại chính [9] :

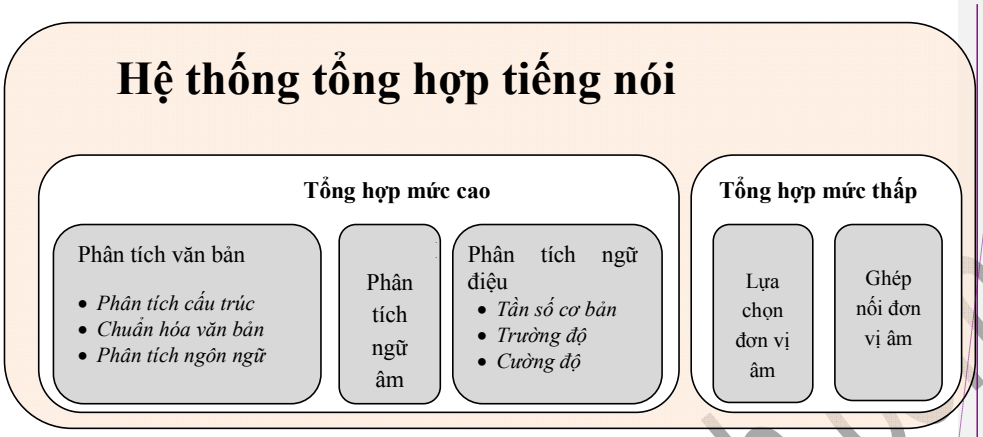
- Tổng hợp cấu âm.
- Tổng hợp formant.
- Tổng hợp theo phương pháp ghép nối (tổng hợp ghép nối).

Phương pháp tổng hợp cấu âm hứa hẹn mang lại kết quả tốt nhất nhưng do quá phức tạp nên khó thực hiện nhất. Phương pháp tổng hợp formant có thể tạo ra được tiếng nói với số lượng câu vô hạn nhưng chất lượng tiếng nói chưa được tự nhiên. Phương pháp tổng hợp được dùng rộng rãi nhất và cho kết quả tốt nhất hiện nay là tổng hợp ghép nối [7] .

Một hệ thống tổng hợp tiếng nói (THTN) gồm hai phần chính: tổng hợp mức cao và tổng hợp mức thấp (Hình 1.1 Mô hình hệ thống THTN.Nhiệm vụ phần tổng hợp mức cao là chuẩn hóa văn bản, phát sinh thông tin về ngữ âm, ngữ điệu. Phần tổng hợp mức thấp (trong phương pháp ghép nối) dựa vào các thông tin phía trên sẽ tiến hành tìm kiếm và lựa chọn đơn vị âm, thực hiện ghép nối và làm trơn tín hiệu, cho ra tiếng nói cần tổng hợp.

Đầu vào của bộ THTN là văn bản và nhiệm vụ của bộ THTN là phải đọc một cách chính xác văn bản này. Văn bản thực tế được viết theo nhiều cách khác nhau, không theo một tiêu chuẩn cụ thể nào và nội dung cũng rất phong phú, đa dạng, bao gồm các chữ số, ngày tháng, từ viết tắt, từ phiên âm nước ngoài … Tất cả đều phải được qua bước chuẩn hóa văn bản, cho đầu ra là các âm tiết đúng theo quy tắc tiếng Việt. Các từ viết tắt phải được thay thế bởi dạng đầy đủ, các con số phải được

chuyển về chữ cái phù hợp, sự nhập nhằng trong các trường hợp phát âm phải được xử lý. Kết quả của bước chuẩn hóa văn bản này ảnh hưởng trực tiếp tới việc đọc đúng hay không đúng văn bản cần tổng hợp.



Comment [TDD1]: Cần có mũi tên kết nối giữa các thành phần cấu thành

Hình 1.1 Mô hình hệ thống THTN [9] .

Văn bản được chuẩn hóa sẽ là đầu vào cho bộ phân tích cú pháp tiếp theo. Phân tích cú pháp chuẩn xác sẽ đưa ra cho hệ thống một cái nhìn toàn cảnh về cấu trúc của văn bản, các cụm từ trong văn bản từ phức tạp cho đến đơn giản nhất, đồng thời các vị trí âm tiết trong cụm từ cũng được đưa ra. Việc này tạo điều kiện thuận lợi cho việc mô hình hóa trường độ, cao độ của câu cần tổng hợp, đồng thời, độ chính xác của câu được phân tích cũng ảnh hưởng tới quá trình lựa chọn đơn vị tiếp theo.

Tiếp theo là quá trình mô hình hóa trường độ, cao độ và cường độ của câu cần tổng hợp. Việc này sẽ phát sinh ngữ điệu cho câu, ảnh hưởng trực tiếp tới mức độ dễ nghe của tiếng nói tổng hợp. Trường độ và cao độ cũng là các tham số trong hàm chi phí dùng trong quá trình lựa chọn đơn vị.

Sau khi văn bản đầu vào được phân tích cú pháp và phát sinh ngữ điệu, văn bản sẽ được tổng hợp bởi quá trình lựa chọn và ghép nối đơn vị âm. Trong quá trình này, tập các đơn vị âm trong cơ sở dữ liệu khớp với đơn vị âm đích nhất sẽ được lựa chọn và ghép nối. Tiếng nói từ văn bản sẽ được sinh ra trong quá trình này. Do từng đoạn tiếng nói vốn hoàn toàn tự nhiên nên ta có thể hy vọng tiếng nói tổng hợp được cũng có tính tự nhiên cao. Tuy nhiên, theo [11] , các đoạn tiếng nói bị ảnh hưởng lớn bởi hiện tượng đồng cấu âm, nên nếu ta ghép nối hai đoạn tín hiệu tiếng nói không liền nhau có thể xảy ra hiện tượng không liên tục về phổ hoặc ngữ điệu. Do sự không liên tục này mà chất lượng tiếng nói tổng hợp có thể giảm đáng kể mặc dù các đoạn được ghép là hoàn toàn tự nhiên.

1.2 Các vấn đề trong tổng hợp tiếng nói bằng phương pháp ghép nối

Trong tổng hợp ghép nối, theo [9] , các vấn đề cần giải quyết để đạt được tiếng nói tổng hợp chất lượng tốt bao gồm:

- Lựa chọn loại đơn vị âm.
- Xây dựng kho đơn vị âm.
- Tìm kiếm đơn vị âm tối ưu.
- Ghép nối đơn vị âm.

1.2.1 Lựa chọn loại đơn vị âm

Tiếng Việt là ngôn ngữ đơn âm tiết có thanh điệu, Cấu trúc đầy đủ của một âm tiết gồm 5 thành phần như sau:

Âm tiết = [Âm đầu][Âm đệm]<Âm chính>[Âm cuối][Thanh điệu]

Trong đó những thành phần nằm trong cặp dấu <> là bắt buộc phải có, những thành phần nằm trong cặp dấu [ ] thì có thể có hoặc không.

Trong tổng hợp tiếng nói tiếng Việt, các loại đơn vị âm được phân tích từ âm tiết có thể dùng trong tổng hợp bao gồm: âm vị, âm vị kép, bán âm tiết, âm đầu/vần, âm tiết [9] .

Số lượng các loại đơn vị âm trong tiếng Việt được tổng hợp theo bảng sau [9] :

Bảng 1.1 Số lượng các loại đơn vị âm trong tiếng Việt

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

Trên đây là các loại đơn vị âm có thể dùng trong THTN tiếng Việt, việc lựa chọn loại đơn vị âm nào sẽ được trình bày chi tiết hơn trong mục 2.1.

1.2.2 Xây dựng kho đơn vị âm

Trong thời kì đầu phát triển tổng hợp tiếng nói ghép nối, kích thước của kho dữ liệu không lớn, và mỗi đơn vị âm chỉ có một mẫu. Cho tới những năm 1990, các

hệ thống tổng hợp ghép nối dựa trên kho đơn vị âm kích thước lớn mới được phát triển, và số lượng mẫu của một đơn vị âm cũng tăng lên.

Để xây dựng kho đơn vị âm, các việc cơ bản cần làm là ghi âm các đoạn tiếng nói từ một người thu âm duy nhất và gán nhãn các đoạn tiếng nói với văn bản tương ứng. Theo [11] , do việc ghi âm thường diễn ra trong nhiều phiên nên một điều quan trọng là duy trì điều kiện thu âm không thay đổi trong suốt quá trình. Việc này có mục đích là tránh sự không liên tục về phổ và biên độ gây ra bởi điều kiện thu âm thay đổi.

Chúng ta có thể thu được tiếng nói tổng hợp chất lượng cao hơn nếu như văn bản được thu âm có nội dung tương đồng với văn bản cần tổng hợp. Việc này làm cho chúng ta có thể sử dụng đơn vị âm dài hơn, và số điểm ghép nối cần thiết sẽ giảm đi.

Sau khi thu âm dữ liệu văn bản, việc tiếp theo là phân đoạn tín hiệu thành các đoạn tương ứng với đơn vị âm. Quá trình phân đoạn có thể thực hiện tự động hoặc thủ công. Vấn đề lớn nhất đối với quá trình phân đoạn thủ công là đòi hỏi công sức lớn trong việc xác định ranh giới giữa các đơn vị âm. Đối với phân đoạn tự động, việc kiểm tra thủ công sau khi phân đoạn là cần thiết để đảm bảo rằng quá trình phân đoạn là đúng trong tất cả các trường hợp.

Bước tiếp theo là gán nhãn cho đoạn âm thanh. Các thông số liên quan như trường độ, tần số cơ bản, điểm đánh dấu đường biên của tín hiệu cũng được gán cho đơn vị âm. Việc lựa chọn các thông số để gán cho đơn vị âm tùy vào từng hệ thống và ngôn ngữ. Trong tiếng Việt, theo [9] các tham số được dùng là tần số cơ bản, năng lượng trung bình, trường độ, các hệ số khoảng cách phổ MFC … Đây sẽ là các tham số dùng trong việc tính toán khoảng cách ngữ điệu và ngữ âm giữa các đơn vị âm.

1.2.3 Tìm kiếm đơn vị âm tối ưu

Văn bản đầu vào được phân tích thành chuỗi các đơn vị âm đích. Các đơn vị âm đích này sẽ được dùng để tìm kiếm trong cơ sở dữ liệu. Mục đích của việc tìm kiếm là chọn ra chuỗi đơn vị tối ưu khớp với ngữ điệu mong muốn nhất. Trong cơ sở dữ liệu thường lưu trữ nhiều mẫu của một đơn vị âm. Hệ thống phải tìm kiếm các đơn vị âm tương ứng tốt nhất sao cho khi ghép nối chúng lại với nhau được tiếng nói tổng hợp có chất lượng tốt nhất có thể. Các đơn vị âm tốt nhất là các đơn vị thỏa mãn sao cho độ méo tiềm tàng giữa chúng là tốt nhất.

Hai phương pháp được dùng để lựa chọn các đơn vị âm tối ưu là dựa trên mô hình cây quyết định và tối ưu hóa hàm chi phí.

Chọn lựa dựa trên mô hình cây quyết định

Trong phương pháp này, dữ liệu học được nhóm lại trong một cây bằng cách phân đoạn mỗi nút thành các nút con dựa trên dữ liệu âm học, bằng cách sử dụng các tiêu chuẩn được gợi ý theo nhãn ngữ cảnh của dữ liệu. Điều này tạo ra một số lượng lớn nhóm, mỗi nhóm chứa các phân đoạn giống nhau ở mức độ ngữ cảnh và âm học. Nhóm được sử dụng cho một ngữ cảnh đặc biệt khi tổng hợp sau đó có thể được suy ra từ cây thích hợp hoặc sử dụng các kết quả ngữ cảnh tương đương nhằm mục đích so sánh nhãn của ngữ cảnh yêu cầu với nhãn của các nhóm có thể sử dụng được.

Chọn lựa dựa trên việc tối ưu hóa hàm chi phí

Trong phương pháp trên, mỗi nhóm thường được biểu diễn bởi điểm trung tâm hoặc phân đoạn gần điểm trung tâm nhất. Tuy vậy, các đơn vị âm có cùng ngữ cảnh vẫn có thể có sự khác nhau về phổ hoặc ngữ điệu, ghép nối những đơn vị âm này vẫn có thể gây ra sự không liên tục. Một phương pháp khác được đưa ra. Sau khi văn bản đầu vào được phân tích ngữ điệu và phiên âm, hệ thống sẽ tìm kiếm các đơn vị âm tốt nhất trong số các mẫu dựa trên việc tối thiểu hóa hàm chi phí.

Nội dung của phương pháp này là sẽ chọn ra đơn vị âm có hàm chi phí nhỏ nhất trong số các mẫu đơn vị âm. Hàm chi phí là tổng có trọng số của hai loại chi phí:

- Chi phí đích thể hiện bằng sự khác nhau giữa đơn vị âm được lựa chọn với đơn vị âm cần tổng hợp.
- Chi phí ghép nối được thể hiện bằng khoảng cách giữa đơn vị âm được chọn so với đơn vị âm trước đó.

Theo các nghiên cứu và thực nghiệm cho tiếng Việt [9] , việc chọn lựa đơn vị âm dựa trên hàm chi phí cho kết quả tốt hơn mô hình cây quyết định. Chi tiết nội dung phương pháp này sẽ được trình bày trong mục 2.2.

1.2.4 Phương pháp ghép nối đơn vị âm.

Tổng hợp ghép nối là ghép nối các đoạn tiếng nói với nhau, chính vì vậy sẽ dẫn tới hiện tượng không liên tục tại điểm ghép nối giữa các đơn vị âm (về cao độ, về phổ, về pha). Sự không liên tục này xảy ra do sự khác nhau về ngữ cảnh của các đơn vị âm hoặc do quá trình phân đoạn tiếng nói. Ngoài ra, chúng ta không thể có đầy đủ các đơn vị âm khớp đúng với ngôn điệu ta mong muốn. Chúng ta cần một kĩ thuật cho phép điều khiển các tham số ngữ điệu của đơn vị âm cần tổng hợp để khi ghép nối giảm được tối thiểu sự không liên tục giữa chúng. Cụ thể mục tiêu là thay đổi biên độ, trường độ và cao độ của đoạn tiếng nói. Việc sửa đổi biên độ có thể dễ dàng được thực hiện bởi bộ nhân trực tiếp, tuy nhiên trường độ và cao độ không

đơn giản như vậy. Kĩ thuật được đề xuất là PSOLA (Pitch Synchronous Overlap and Add). Đây là một kĩ thuật dùng rất phổ biến trong các chương trình tổng hợp tiếng nói tiếng Việt và các tiếng khác.

1.2.4.1 Phương pháp PSOLA

Phương pháp PSOLA 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 cửa sổ tập trung tại các vị trí chu kì 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:

- 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 được biến đổi

1.2.4.2 Các phiên bản của PSOLA

- **TD-PSOLA** (Time Domain - PSOLA) là phiên bản miền thời gian của PSOLA (TD-PSOLA). Phương pháp này thao tác với tín hiệu trên miền thời gian nên được sử dụng nhiều vì hiệu quả trong tính toán của nó.
- **FD-PSOLA** (Frequency Domain - PSOLA) là phương pháp bao gồm các bước giống như TD-PSOLA nhưng thao tác trên miền tần số. Phương pháp này có chi phí tính toán cao hơn TD-PSOLA do cần ít nhất một phép biến đổi FFT và IFFT cho mỗi đoạn tín hiệu.
- **LP-PSOLA** (Linear Prediction – PSOLA). Phương pháp dự đoán tuyến tính được thiết kế để mã hoá tiếng nói nhưng phương pháp này cũng có thể dùng cho tổng hợp.

PSOLA là một phương pháp được sử dụng trong xử lý tiếng nói từ rất sớm và đã được trình bày rất chi tiết trong các tài liệu và luận văn về tổng hợp tiếng nói [9] [11] . Vì vậy, luận văn này sẽ không trình bày chi tiết nội dung phương pháp.

1.2.4.3 Vấn đề không liên tục trong ghép nối

Khi sử dụng kỹ thuật PSOLA cho việc ghép nối các đơn vị âm, sẽ vẫn tồn tại ba khả năng về sự không liên tục có thể xảy ra: không liên tục về pha, tần số cơ bản và phổ [9] .

Sự không liên tục về pha: xảy ra do có sự khác nhau về vị trí của các điểm đánh dấu cao độ giữa các đoạn tín hiệu trái và phải. Để loại bỏ sự không liên tục này, ta cần phải xác định lại vị trí các điểm đánh dấu cao độ theo cùng một chuẩn và đồng nhất cho tất cả các mẫu của tín hiệu.

Sự không liên tục về tần số cơ bản: xảy ra do các đoạn tín hiêu cần ghép nối có các tần số cơ bản khác nhau. Khi thu âm dữ liệu tiếng nói, nếu người thu âm nói với một tần số cơ bản không đổi thì có thể giảm thiểu sự không liên tục này. Tuy nhiên đối với ngôn ngữ có thanh điệu thì đây không phải là một biện pháp thích hợp. Phương pháp TD-PSOLA có thể dùng để chuẩn hóa theo tần số cơ bản. Trong hệ thống tổng hợp ghép nối, sự không liên tục này được biểu diễn bởi một chi phí kết nối đo sự méo ngữ điệu tiềm tàng giữa hai đoạn tiếng nói. Quá trình lựa chọn đơn vị sẽ chọn ra đoạn tín hiệu có chi phí thấp để tổng hợp.

Sự không liên tục về phổ: xảy ra do hiện tượng đồng cấu âm, gây ra những ảnh hưởng khác nhau lên các đoạn tín hiệu tiếng nói phía trái và phía phải mà xuất phát từ ngữ cảnh khác nhau. Phương pháp TD-PSOLA không phải là phương pháp có thể loại bỏ sự không liên tục này mà ta có thể sử dụng một trong hai cách sau:

- Liên kết TD-PSOLA với một mô hình tham số dạng LPC và thực hiện làm trơn phổ trong miền tham số.
- Áp dụng kĩ thuật MBR-PSOLA.

Trong hệ thống tổng hợp theo phương pháp ghép nối, sự không liên tục này cũng được biểu diễn bởi một chi phí kết nối đo sự méo phổ tiềm tàng giữa hai phân đoạn tiếng nói. Nhờ có chi phí này mà những đoạn tín hiệu tiếng nói có sự không liên tục về phổ thấp sẽ được lựa chọn để ghép nối.

1.3 Kết luận

Tổng hợp ghép nối có bốn vấn đề cần giải quyết để thu được tiếng nói tổng hợp có chất lượng cao. Đây đều là những vấn đề lớn, đòi hỏi kiến thức lẫn thời gian thực hiện. Do đó, trong giới hạn của thời gian làm đồ án tốt nghiệp, tác giả được giới hạn phạm vi, tập trung vào giải quyết vấn đề lựa chọn loại đơn vị âm và tìm kiếm đơn vị âm tối ưu. Đối với vấn đề xây dựng kho đơn vị âm và ghép nối đơn vị âm, tác giả sử dụng lại CSDL và chương trình tổng hợp tiếng nói trên mức bán âm tiết của tác giả Trần Đỗ Đạt [9] tại trung tâm nghiên cứu Mica. Trong chương tiếp

theo, luận văn sẽ tập trung trình bày chi tiết vấn đề lựa chọn và tìm kiếm đơn vị âm tối ưu.

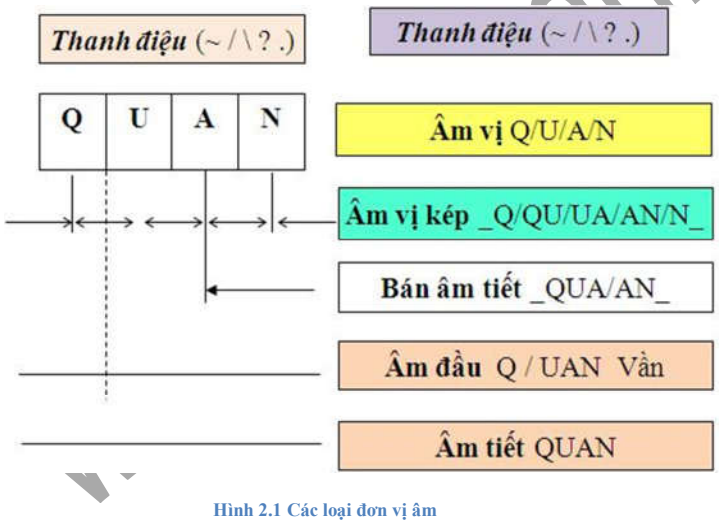
Chương 2. Lựa chọn và tìm kiếm đơn vị âm trong tổng hợp ghép nối

Trong chương này, luận văn sẽ trình bày về các vấn đề:

- *Các loại đơn vị âm và loại được lựa chọn trong tổng hợp ghép nối cho tiếng Việt*
- *Phương pháp lựa chọn đơn vị âm tối ưu*

2.1 Lựa chọn loại đơn vị âm

Tiếng Việt có các loại đơn vị âm có thể dùng cho tổng hợp tiếng nói là âm vị, âm vị kép, bán âm tiết, âm đầu/vần, âm tiết, cụm từ. Hình 2.1 mô tả các loại đơn vị âm của âm tiết "QUAN".



Hình 2.1 Các loại đơn vị âm

2.1.1 Âm vị

Âm vị là loại đơn vị nhỏ nhất trong hệ thống các đơn vị của ngôn ngữ. Hệ thống ghép nối sử dụng âm vị về mặt lý thuyết có thể ghép nối được tất cả các âm tiết. Trong tiếng Việt có 40 âm vị không có thanh điệu, 130 âm vị có thanh điệu [9] . Vì số lượng khá nhỏ nên kích thước cơ sở dữ liệu của hệ thống sẽ được thu gọn lại. Tuy nhiên, do có nhiều sự thay đổi về ngữ cảnh, sự không liên tục trong ghép nối xảy ra thường xuyên. Âm thanh tổng hợp được vì thế sẽ có chất chưa tốt và tương đối khó nghe.

2.1.2 Âm vị kép

Âm vị kép (diphone) là một đoạn tín hiệu cấu thành từ nửa cuối một đơn vị âm và nửa đầu đơn vị âm tiếp theo. Do đó, âm vị kép giữ được sự chuyển tiếp giữa các đơn vị âm. Hình 2.1 chỉ ra cấu trúc của một âm tiết theo các âm vị kép. Các biên giữa âm vị kép trong khi tổng hợp là điểm giữa các đơn vị âm, điều này làm giảm đi sự không liên tục trong ghép nối, bởi những điểm này thường có vùng phổ ổn định và bền hơn với các ngữ cảnh âm học.

2.1.3 Bán âm tiết

Bán âm tiết là một phân đoạn tín hiệu của một nửa đầu và nửa cuối của một âm tiết. Như vậy, để tạo thành một âm tiết, ta chỉ cần ghép nối hai bán âm tiết với nhau, số điểm ghép nối chỉ là một. So với âm vị hoặc âm vị kép thì rõ ràng việc sử dụng bán âm tiết hứa hẹn tín hiệu tổng hợp có chất lượng tốt hơn do giảm thiểu được sự không liên tục trong ghép nối. Theo Bảng 1.1, số lượng bán âm tiết trong tiếng Việt không nhiều, đây là một lợi thế trong tổng hợp dựa theo bán âm tiết.

2.1.4 Âm đầu và vần

Âm tiết cũng có thể chia thành hai thành phần: âm đầu và vần. Âm đầu là phần phụ âm bắt đầu một âm tiết, phần này là tùy chọn và không mang thông tin về thanh điệu. Vần là sự kết hợp của ba thành phần: âm đệm, âm chính và âm cuối. Phần này là phần bắt buộc và mang thông tin về thanh điệu của âm tiết. Ưu điểm của loại đơn vị âm này là nó giữ lại đặc tính thanh điệu của âm tiết. Tuy nhiên, nhược điểm của loại này giống với loại đơn vị âm kiểu âm vị, có nhiều sự không liên tục tại điểm ghép nối giữa âm đầu và vần.

Tiếng Việt có 22 âm đầu và 155 vần khi không xét đến thanh điệu, 661 vần nếu xét đến thanh điệu. Số lượng đơn vị âm loại này không lớn và có thể chấp nhận được đối với một hệ thống tổng hợp. Tuy nhiên, vấn đề lớn là sự không liên tục tạo ra trong quá trình ghép nối là lớn hơn so với loại bán âm tiết. Vì vậy, loại đơn vị âm này không được ưu tiên sử dụng.

2.1.5 Âm tiết

Âm tiết là đơn vị phát âm nhỏ nhất của lời nói, mang những sự kiện ngôn điệu như thanh điệu, trọng âm. Ưu điểm của loại đơn vị âm này là ta không cần cắt âm tiết ra thành nhiều phần, vì vậy nó bảo toàn đầy đủ sự đồng cấu âm giữa các âm vị bên trong của đơn vị âm. Xét về mặt chất lượng, âm tiết là loại đơn vị âm lý tưởng cho tổng

hợp. Tuy nhiên, với ngôn ngữ có thanh điệu và đơn âm tiết như tiếng Việt, số lượng âm tiết là khá lớn. Vì vậy ta khó có thể xây dựng cơ sở dữ liệu bao phủ đầy đủ tất cả các âm tiết của tiếng Việt. Muốn đạt được độ phủ CSDL lớn, hệ thống THTN thường được xây dựng cho các lĩnh vực giới hạn, ví dụ như [10] áp dụng trong lĩnh vực tường thuật bóng đá.

2.1.6 Cụm từ

Các đơn vị âm có thể là các 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.

2.1.7 Nhận xét

Trong các loại đơn vị âm nhỏ hơn âm tiết, theo [9] , bán âm tiết là loại được sử dụng trong tiếng Việt mang lại kết quả tổng hợp tốt so với các loại đơn vị âm còn lại. Kích thước cơ sở dữ liệu chấp nhận được (khoảng dưới 10M), chương trình có thể chạy trên máy tính cá nhân, trên DSP [1] , có thể tổng hợp được hầu hết âm tiết tiếng Việt. Ngày nay, sự phát triển của phần cứng cho phép ta có thể nghĩ tới chương trình tổng hợp với kích thước cơ sở dữ liệu lớn hơn, thời gian thực thi chương trình nhanh hơn. Việc sử dụng kết hợp các loại đơn vị âm bao gồm bán âm tiết, âm tiết, cụm từ được đề xuất, gọi là lựa chọn đơn vị không đồng nhất. Trong đồ án này, tác giả đi theo hướng lựa chọn đơn vị không đồng nhất.

Bảng 2.1 Các loại đơn vị âm sử dụng

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

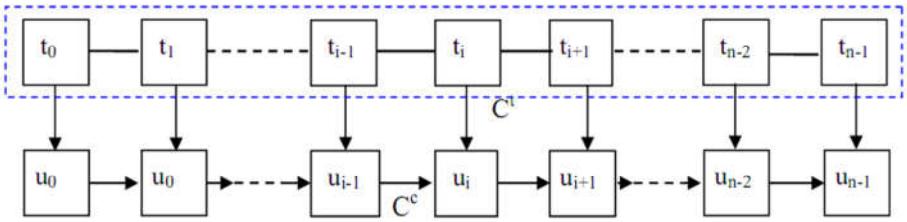
Như bảng trên đã chỉ ra ưu nhược điểm của từng loại đơn vị âm. Đơn vị âm càng dài thì số điểm ghép nối càng giảm, tuy nhiên xác suất tìm thấy đơn vị âm này trong CSDL cũng nhỏ hơn so với đơn vị âm ngắn hơn. Lựa chọn đơn vị không đồng

nhất sẽ 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 hầu hết âm tiết trong tiếng Việt bằng việc sử dụng 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.

2.2 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 hành 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 (t_i, i = 0…n-1) chứa những thông tin về ngữ điệu cần thiết. Từ đích này, ta cần tìm ra chuỗi n đơn vị âm (u_i, 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 2.2 Hàm chi phí giữa các đơn vị âm

Hai hàm chi phí được sử dụng:

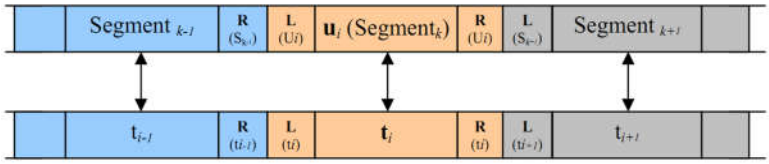
- Chi phí đích $C^t(u_i,t_i)$ là sự khác nhau giữa đơn vị âm trong cơ sở dữ liệu u_i và đích t_i.
- Chi phí ghép nối $C^c(u_i,t_i)$ là sự khác nhau tại điểm ghép nối giữa hai đơn vị âm liên tiếp (u_{i-1},u_i).

Với đặc tả về đích và chuỗi n đơn vị âm $T_1^n = (t_1,t_2,...t_n)$, hệ thống cần chọn ra n đơn vị âm $U_1^n = (u_1,u_2,...u_n)$ 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. Nội dung hai bước này sẽ được trình bày chi tiết trong chương tiếp theo.

2.2.1 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 âm 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. Sau đó, trong những đơn vị âm này, hệ thống sẽ chọn ra những đơn vị âm có độ méo thấp nhất dựa trên hàm chi phí đích [9] .



Hình 2.3 Chi phí đích

Sự khác nhau giữa đích t_i và đơn vị âm u_i được ước lượng bởi tính toán chi phí đích bao gồm các chi phí phụ sau:

- Sự khác nhau về ngữ cảnh giữa mẫu và đơn vị âm đích: sự khác nhau này được tính bằng cách so sánh những thông tin của các segment (k-1) và (k+1) lần lượt với đích là t_{i-1} và t_{i+1}. Các thông tin liên quan gồm có phiên âm và thanh điệu. Nếu hai giá trị của cùng một tham số của t_i và u_i là như nhau, thì sự khác nhau là 0, nếu không thì sự khác nhau bằng 1.
- Sự khác nhau về ngữ điệu giữa mẫu và đích: trường độ, tần số cơ bản, năng lượng. Giá trị của vector thể hiện sự khác nhau giữa mỗi tham số sẽ được chuẩn hóa để nhận các giá trị là 0 hoặc 1. Thông thường, giá trị trung bình của F0 được sử dụng. Tuy nhiên, đối với tiếng Việt, giá trị trung bình là chưa đủ [9] . Để so sánh F0 hoặc tính toán sự khác biệt giữa hai thanh điệu, ta gán vào mỗi thanh điệu hai tham số: hướng và độ phức tạp của thanh điệu.

- Tham số về hướng thể hiện hướng đường cong F0 của thanh điệu. Chúng ta coi thanh ngang, có đường cong F0 nằm ngang, có giá trị tham số là 0. Nếu đường cong F0 có giá trị hướng xuống thì tham số bằng -1, nếu có giá trị hướng lên thì tham số bằng 1.
- Tham số về độ phức tạp được thể hiện bằng độ phức tạp của thanh điệu.

Các giá trị của hai tham số tương ứng với 6 thanh điệu được thể hiện trong bảng dưới đây. Các giá trị này được đưa ra bằng cách so sánh với thanh ngang, được xem là thanh điệu tham chiếu.

Bảng 2.2 Hướng và độ phức tạp của các thanh điệu [9]

Thanh điệu	Hướng d^D	Độ phức tạp d^C
1	0	0
2	-1	1
3	1	3
4	-1	2
5a	1	1
6a	-1	2
5b	1	1
6b	-1	1

Công thức tính sự khác nhau về thanh điệu giữa hai đơn vị âm u_i và t_i:

$$C^t_{ton}(t_i,u_i) = w_D * \left|d^D_{t_i} - d^D_{u_i}\right| + w_C * (d^C_{t_i} + d^C_{u_i}) \qquad (2.1)$$

Chi phí đích tổng cộng được sử dụng để đánh giá khoảng cách giữa một mẫu và đích được tính bằng tổng sự khác nhau giữa hai vector đặc trưng tương ứng với mỗi đơn vị âm.

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w^t_j C^t_j(t_i, u_i) \qquad (2.2)$$

Trong đó:

$C^t_j(t_i,u_i)$*: chi phí đích phụ.*

w^t_j*: hệ số cân bằng*

Sau giai đoạn tiền lựa chọn, với mỗi đơn vị âm đích t_i, hệ thống chọn ra N(~20) mẫu tốt nhất. Tuy nhiên với số mẫu như vậy sẽ tạo ra N^n các tập hợp mẫu. Hệ thống không thể chọn ra một cách ngẫu nhiên chuỗi đơn vị âm dùng để tổng hợp mà nó cần phải tìm ra chuỗi n mẫu tốt nhất trong đó. Nhiệm vụ này được thực hiện trong gian đoạn lựa chọn cuối cùng dưới đây.

2.2.2 Chọn lựa 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 và chi phí ghép nối. Chi phí ghép nối được tính theo công thức dưới đây:

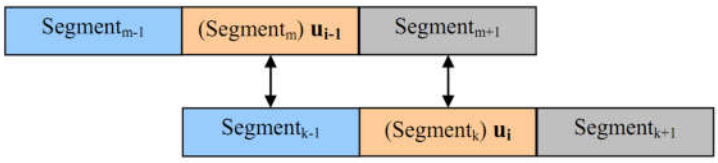
$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w^c_j C^c_j(u_{i-1}, u_i) \qquad (2.3)$$

Trong đó: $C^c_j(u_{i-1},u_i)$: chi phí ghép nối phụ.

Chi phí ghép nối phụ tương ứng với khoảng cách ngữ cảnh và khoảng cách tại điểm ghép nối giữa hai đơn vị âm:

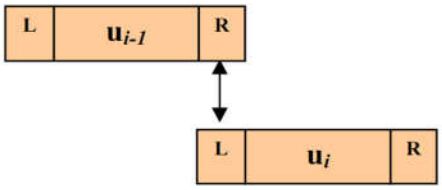
- Sự khác nhau giữa segment bên phải của u_{i-1} và u_i : $d(segment_{m+1}, u_i)$.
- Sự khác nhau giữa segment bên trái của u_i và u_{i-1}: $d(u_{i-1}, segment_{k-1})$.

Nếu hai giá trị của cùng một tham số của u_{i-1} và u_i là giống nhau thì sự khác nhau là 0, nếu không thì khoảng cách bằng 1.



Hình 2.4 So sánh sự khác nhau về ngữ cảnh.

Khoảng cách tại điểm kết nối thu được bằng cách tính các khoảng cách ngữ âm của các vùng tín hiệu được sử dụng ghép nối như là khoảng cách F0, và khoảng cách phổ. Khoảng cách phổ được sử dụng để tính toán sự không liên tục về phổ. Đó là khoảng cách Euclid giữa 12 hệ số MFCC (Mel-Frequency Cepstral Coefficients) của 2 cửa sổ 10ms (cửa sổ cuối cùng của segment u_{i-1} và cửa sổ đầu tiên của segment u_i).



Hình 2.5 So sánh sự khác nhau về phổ

Vậy khoảng cách tổng của một chuỗi n đơn vị âm chính là tổng của chi phí đích và chi phí ghép nối:

$$C\left(t_1^n, u_1^n\right) = \sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=2}^{n} C^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad (2.4)$$

$$C\left(t_1^n, u_1^n\right) = \sum_{i=1}^{n}\sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) + \sum_{i=2}^{n}\sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) + C^c(S, u_1) + C^c(u_n, S) \quad (2.5)$$

Trong đó, S mô tả khoảng lặng, $C^c(S,u_1)$ và $C^c(u_n,S)$ xác định các điều kiện ban đầu và kết thúc để cho việc ghép nối 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.

$$\overline{u}_1^n = \min_{u_1,\ldots,u_n} C\left(t_1^n, u_1^n\right)$$

Trong khi tính toán hàm chi phí, chi phí tổng của dãy các đơn vị âm là một tổng có trọng số của chi phí đích và chi phí ghép nối. Các chi phí này cũng là tổng có trọng số của các chi phí con. Việc xác định các trọng số trong đó rất quan trọng đối với chất lượng chung của tiếng nói tổng hợp. Tuy nhiên, việc tìm một cách khách quan để so sánh chất lượng tiếng nói tổng hợp bằng cách sử dụng các trọng số khác nhau là rất khó. Vì vậy, chúng ta cần các cách khác nhau để xác định các trọng số. Thông thường, các trọng số được xác định căn cứ vào thực nghiệm dựa trên kiến thức và bài đánh giá cảm thụ [9] [6] .

Việc lựa chọn dãy đơn vị âm tối ưu được thực hiện bằng cách áp dụng thuật toán Viterbi [10] [6] .

2.3 Kết luận

Qua nội dung được trình bày trong chương này, luận văn đã làm sáng rõ việc lựa chọn loại đơn vị âm và phương pháp lựa chọn đơn vị âm tối ưu trong tổng hợp tiếng nói tiếng Việt. Việc sử dụng kết hợp ba loại đơn vị âm là bán âm tiết, âm tiết, cụm từ đòi hỏi có những thay đổi trong cách áp dụng phương pháp đã trình bày ở trên. Trong chương sau, luận văn sẽ tổng hợp các nghiên cứu liên quan và đề xuất cách áp dụng phương pháp lựa chọn đơn vị không đồng nhất trong tổng hợp tiếng nói tiếng Việt.

Chương 3. Đề xuất cách áp dụng phương pháp lựa chọn đơn vị âm không đồng nhất cho tổng hợp tiếng nói tiếng Việt

Trong chương này, luận văn sẽ trình bày về:

- Phương pháp lựa chọn đơn vị không đồng nhất và áp dụng cho tiếng Việt
- Mô hình tổng thể của hệ thống tổng hợp tiếng nói tác giả phát triển

3.1 Tìm kiếm đơn vị âm không đồng nhất

Trong phần 2.2, luận văn đã trình bày chi tiết quá trình tìm kiếm và lựa chọn đơn vị âm. Đây là phương pháp áp dụng khi hệ thống sử dụng một loại đơn vị âm duy nhất. Trong các nghiên cứu gần đây [2] [5] [10] , một phương pháp mới dựa trên phương pháp trong 2.2 được sử dụng là lựa chọn đơn vị không đồng nhất. Mục đích là cải thiện chất lượng tiếng nói tổng hợp bằng cách giảm thiểu số điểm ghép nối và số lần xử lý tín hiệu. Mỗi loại ngôn ngữ có cách áp dụng và thực thi phương pháp này theo cách khác nhau. Dưới đây, luận văn sẽ tổng kết các nghiên cứu có liên quan tới phương pháp này.

3.1.1 Tổng kết các nghiên cứu liên quan

Đối với nghiên cứu [2] cho tiếng Hà Lan, đơn vị âm dùng để tìm kiếm là âm vị kép. Trong nghiên cứu này, một loại chi phí khác được bổ sung vào hàm chi phí tổng là chi phí phụ cận. Nếu hai diphone là liền kề nhau thì chi phí bằng 0, nếu khác thì chi phí bằng 1. Bằng việc thiết lập trọng số cao cho chi phí này so với các chi phí khác, dãy đơn vị được lựa chọn thường cho số lượng nhỏ hơn các điểm kết nối. Tuy nhiên, các chi phí cho tất cả khả năng ghép nối có thể vẫn được tính toán mặc dù những sự ghép nối này thường không được chọn do trọng số cao của chi phí phụ cận.

Đầu tiên tìm kiếm trong CSDL cho những đơn vị khớp về ngữ âm với diphone đích. Kết quả là một số lượng rất lớn các đơn vị ứng viên tiềm năng. Sau đó, bỏ bớt số lượng ứng viên và chỉ giữ lại những đơn vị có diphone liền kề trong CSDL tương ứng với diphone đích thứ hai. Kết quả là những đơn vị có chiều dài lớn hơn đã khớp với các diphone đích liền kề nhau. Quá trình này tiếp tục cho tới khi đơn vị dài nhất có thể được tìm thấy. Nếu có đơn vị nào mà không khớp với diphone đích, quá trình tìm kiếm bắt đầu lại để lựa chọn những đơn vị ứng viên khớp với những diphone không khớp đó.

Thuật toán trên có thể dẫn tới giảm thiểu số điểm kết nối. Tuy nhiên, các đơn vị ứng viên có độ dài càng lớn thì càng ít khả năng được tìm thấy. Việc này làm giảm số lượng ứng viên tiềm năng cho việc lựa chọn, ảnh hưởng tới chất lượng ghép nối và ngữ điệu. Vì vậy, một phương pháp được đề xuất là không dùng đơn vị ứng viên dài nhất có thể mà có thể dùng đơn vị ngắn hơn. Vào thời điểm tìm thấy ứng viên lớn nhất, ta quay lui và lựa chọn những đơn vị khớp với số lượng đơn vị nhỏ hơn. Trong hầu hết trường hợp, kết quả là có nhiều ứng viên tiềm năng hơn. Việc này dừng lại khi đạt tới ranh giới của âm tiết cuối cùng của đơn vị ứng viên lớn nhất. Nếu ứng viên dài nhất không chứa bất kì ranh giới âm tiết nào, đơn vị ứng viên sẽ không bị giảm chiều dài.

Sau khi tập các đơn vị âm tối ưu được lựa chọn, các đơn vị được ghép nối lại với nhau mà không thay đổi tham số ngữ điệu của đơn vị âm. Sự thay đổi chỉ được thực hiện tại biên khi các đơn vị được kết nối bởi thuật toán PSOLA.

Đối với nghiên cứu [5] cho tiếng Trung, đơn vị âm cơ sở là âm tiết có thanh điệu. CSDL âm thanh có độ dài 15 giờ, đảm bảo phủ gần hết số lượng âm tiết trong tiếng Trung – khoảng 1600 âm tiết, tương đối nhỏ so với số lượng hơn 7000 âm tiết trong tiếng Việt [9] . Từng âm tiết được tìm kiếm trong CSDL. Các hàm chi phí được sử dụng để chọn ra tập đơn vị âm tối ưu là chi phí đích và chi phí phụ cận. Chi phí đích là sự sai khác giữa hai vector bao gồm 6 thành phần:

- PinP: vị trí của âm tiết hiện tại trong cụm từ chứa nó.
- PinW: vị trí của âm tiết hiện tại trong từ chứa nó.
- 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.
- RightT: thanh điệu của âm tiết bên phải.

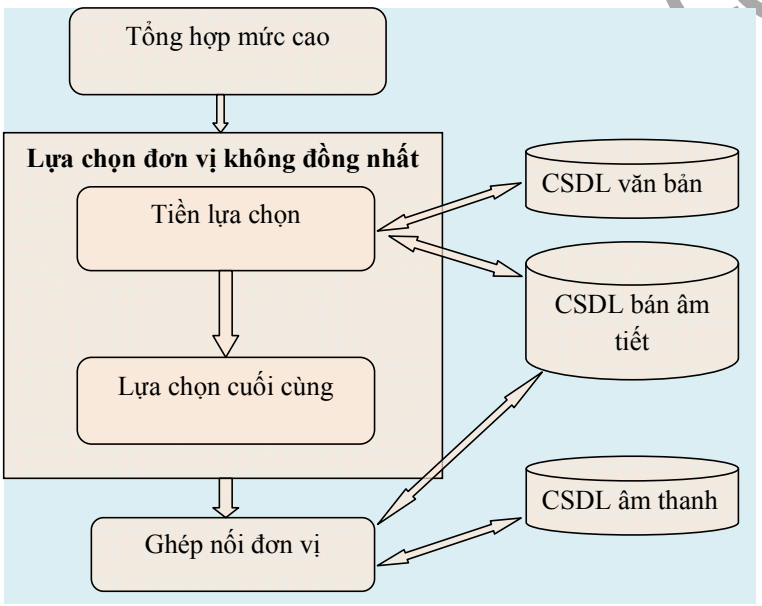
Chi phí phụ cận nhận hai giá trị 0 hoặc 1, là 0 khi hai đơn vị âm là hai đoạn âm thanh liền kề nhau trong CSDL. Bằng việc sử dụng chi phí này, các cụm từ có độ dài lớn có thể được lựa chọn, điều này theo đúng mục đích của phương pháp tìm kiếm đơn vị không đồng nhất.

Đối với nghiên cứu [10] cho tiếng Việt, tập các đơn vị ngữ âm được phân đoạn theo cấu trúc cây phân cấp. Mức lá là các âm tiết, rồi đến từ, cụm từ và nút gốc là câu. Cây phân cấp này được xây dựng theo phương pháp thống kê các cụm từ phổ biến trong một lĩnh vực nhỏ là tường thuật bóng đá. Âm tiết là loại đơn vị âm nhỏ nhất. Với việc xây dựng CSDL có kích thước lớn – 11 giờ tiếng nói, bộ từ vựng gồm 3479 tiếng đã phủ gần hết toàn bộ ứng dụng được giới hạn trong một lĩnh vực hẹp. Tuy nhiên, hệ

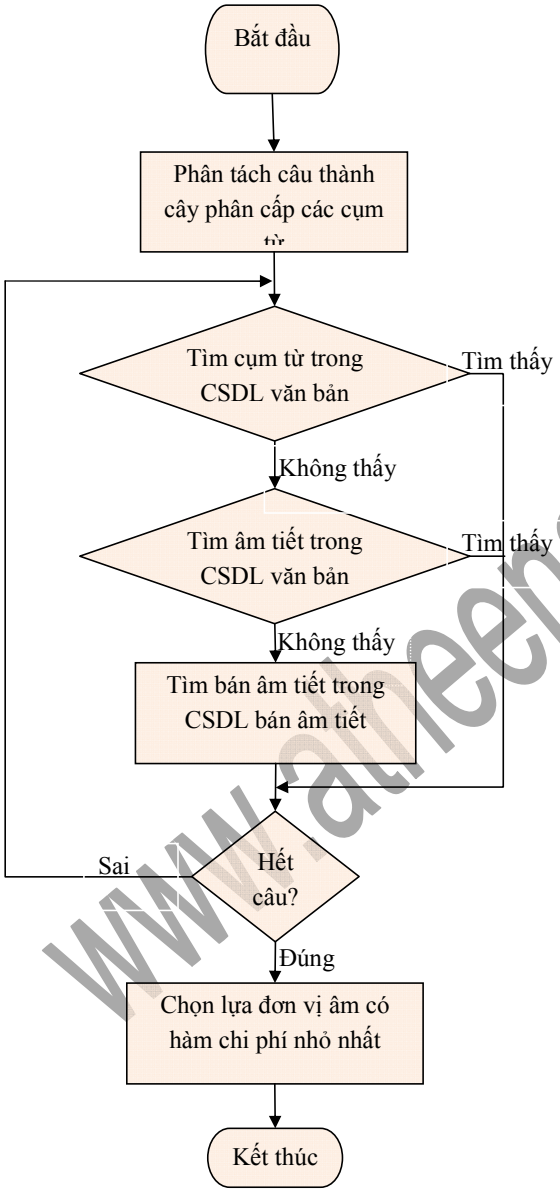
thống này cũng có nhược điểm là kích thước bộ từ vựng chỉ bằng một nửa số lượng âm tiết tiếng Việt, và nếu gặp âm tiết không có trong bộ từ vựng, hệ thống sẽ không tổng hợp được.

3.1.2 Mô hình thuật toán

Như đã trình bày trong mục 2.1, các loại đơn vị âm được lựa chọn là cụm từ, âm tiết, bán âm tiết. Với mục đích giảm thiểu số điểm ghép nối, loại đơn vị âm được ưu tiên chọn lựa sẽ theo thứ tự như trên. Hình 3.1 chỉ ra mô hình tổng quan của quá trình lựa chọn đơn vị âm. Dựa trên phương pháp đã được trình bày, quá trình lựa chọn đơn vị cũng được chia thành hai bước là tiền lựa chọn và lựa chọn cuối cùng. Nhiệm vụ của bước tiền lựa chọn là chọn ra các đơn vị âm dài nhất có thể, bước lựa chọn cuối cùng sẽ chọn ra dãy đơn vị âm tốt nhất.



Hình 3.1 Mô hình lựa chọn đơn vị âm không đồng nhất.



Hình 3.2 Quá trình tìm kiếm đơn vị

3.1.2.1 Tiền lựa chọn

CSDL được dùng trong bước tiền lựa chọn là CSDL văn bản và CSDL bán âm tiết. Các bước chi tiết của quá trình tìm kiếm đơn vị âm được mô tả trong Hình 3.2 Quá trình tìm kiếm đơn vị.

Bắt đầu của quá trình lựa chọn đơn vị, văn bản cần tổng hợp sẽ được chia thành các câu để tìm kiếm. Mỗi câu được phân tách thành các cụm từ và âm tiết và tìm kiếm chúng trong CSDL 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 tính 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 CSDL bán âm tiết. Tại mức này hầu như không xảy ra sự kiện không tìm thấy bán âm tiết [9] . Nếu không tìm thấy thì âm tiết đó không được tổng hợp.

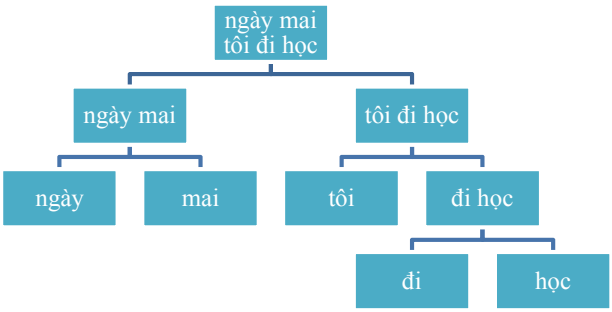
Vấn đề đặt ra là làm sao có thể phân tích được một câu thành các cụm từ và âm tiết sao cho tối đa hóa xác suất tìm thấy cụm từ được phân tích. Bởi nếu không chọn được cụm từ thích hợp để tìm kiếm, tỉ lệ đơn vị âm sẽ phần lớn là âm tiết và bán âm tiết, việc này ảnh hưởng trực tiếp tới hiệu quả của thuật toán lựa chọn đơn vị không đồng nhất. Ví dụ đối với một câu đơn giản "*Xin cám ơn mọi người*" và với các cách phân tách cụm từ như sau:

- *Xin cám* | *ơn mọi* | *người.*
- *Xin* | *cám ơn* | *mọi người.*

Nhìn vào hai cách phân tách trên, rõ ràng ta có thể nhận thấy với cách phân tách thứ hai, cụm từ được tìm kiếm sẽ có khả năng xuất hiện trong CSDL cao hơn. Một giải pháp được đề xuất để giải quyết vấn đề trên là sử dụng cây phân tích cú pháp. Câu cần tổng hợp sẽ được chia ra thành các cụm từ theo các mức khác nhau nhờ quá trình phân tích cú pháp. Ví dụ như hình minh họa dưới đây cho câu "*Ngày mai tôi đi học*".

Quá trình tìm kiếm sẽ được bắt đầu từ gốc, sau đó đi xuống các nhánh. 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 để tìm kiếm này làm tăng xác suất tìm thấy của những cụm từ có độ dài lớn 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 không đồng nhất.

Trong trường hợp không tìm thấy ứng viên nào ở mức lá, âm tiết còn lại sẽ được tổng hợp ở mức bán âm tiết. Theo [9] , việc tổng hợp ở mức bán âm tiết có thể tổng hợp được hầu hết các âm tiết trong tiếng Việt.



Hình 3.3 Cây phân cấp để tìm kiếm

3.1.2.2 Lựa chọn cuối cùng

Kết quả của bước tiền lựa chọn thường cho ra nhiều đơn vị ứng viên với cùng một đơn vị âm đích. Đối với việc sử dụng một loại đơn vị âm duy nhất, việc chọn ra tập đơn vị âm để ghép nối có thể thực hiện như 2.2.2. Tuy nhiên, trong trường hợp này có sự kết hợp của 3 loại đơn vị âm nên cần thiết phải có một cơ chế lựa chọn khác. Một giải pháp được đề xuất là tối ưu hóa cục bộ hàm chi phí. Nội dung của giải pháp này như sau:

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 phí lớn nhất.

- *Đối với dãy con chứa bán âm tiết:*
  - *Tính toán hàm chi phí cho dãy con như công thức trong 2.2.2.*
  - *Giữ lại $N_{halfSyl}$ chuỗi đơn vị âm có hàm chi phí nhỏ nhất trong dãy con này ($N_{halfSyl}$ được xác định bằng 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.*
    - *RightSl: 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.*
- *RightT: 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, nếu không hàm có giá trị bằng 1.*
  - *Hàm chi phí tổng là kết hợp của chi phí ghép nối và chi phí phụ cận. Trọng số của các hàm chi phí và tham số được xác định trong quá trình thực nghiệm.*
  - *Giữ lại N_{syl} chuỗi âm tiết có hàm chi phí nhỏ nhất trong dãy con này.*

- *Đối với dãy con chứa cụm từ:*
  - *Vẫn sử dụng hai loại hàm chi phí như đối với mức bán âm tiết. Giữ lại N_{phrase} dãy cụm từ có hàm chi phí nhỏ nhất trong dãy con này.*

Bước 3: chọn ra dãy đơn vị âm có hàm chi phí nhỏ nhất sử dụng thuật toán Viterbi.

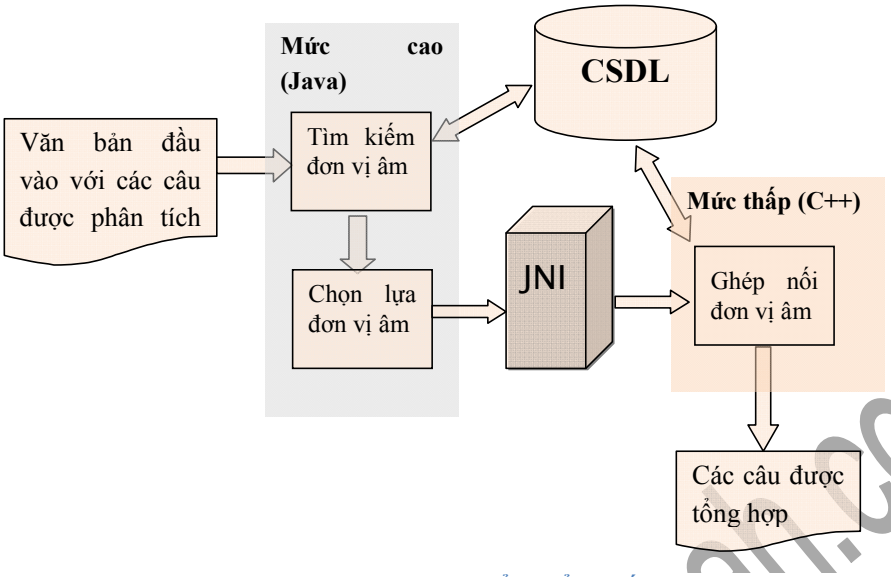
3.2 Mô hình tổng thể hệ thống

Chương trình được viết trên hai ngôn ngữ là Java và C++. Mục đích của việc sử dụng hai ngôn ngữ là muốn tận dụng ưu điểm của hai ngôn ngữ này. Java được dùng trong việc trong việc xử lý văn bản, còn C++ được dùng trong xử lý tín hiệu âm thanh. JNI được sử dụng để ghép nối phần mã được viết bởi C++ với Java. Dưới đây là mô hình tổng thể của hệ thống.

Đầu vào: văn bản cần tổng hợp với các câu đã được phân tích cú pháp, tổ chức thành cây phân cấp các cụm từ.

Đầu ra: file âm thanh tổng hợp từ văn bản đầu vào.

Chức năng phần mức cao là tìm kiếm và lựa chọn đơn vị âm tốt nhất để tổng hợp. Chức năng phần mức thấp là ghép nối các đơn vị âm.



Hình 3.4 Mô hình tổng thể hệ thống

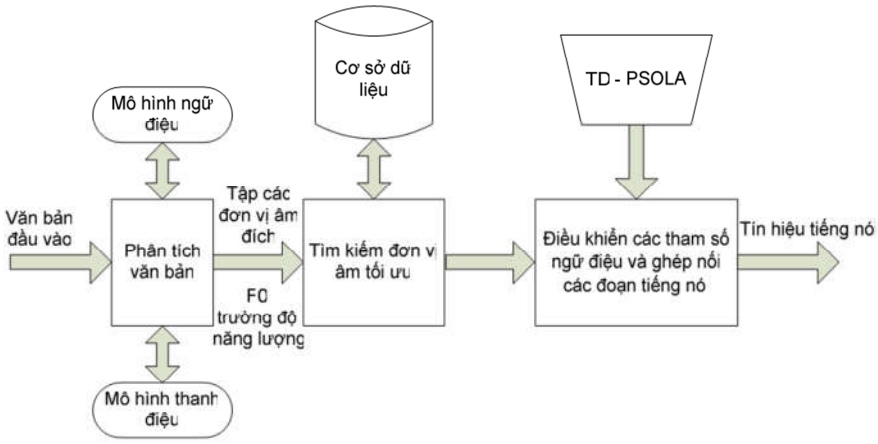
3.3 Kết luận

Trong chương này, luận văn đã tổng kết các nghiên cứu liên quan tới phương pháp lựa chọn đơn vị không đồng nhất. Dựa trên các nghiên cứu đó, một phương pháp lựa chọn đơn vị không đồng nhất cho tiếng Việt được đề xuất. Việc xây dựng và thử nghiệm hệ thống THTN dựa trên phương pháp này sẽ được trình bày vào chương tiếp theo.

Chương 4. Phát triển hệ thống tổng hợp tiếng nói tiếng Việt theo phương pháp lựa chọn đơn vị âm không đồng nhất

4.1 Giới thiệu chương trình tổng hợp Hoa Súng

Chương trình tổng hợp tiếng nói Hoa Súng là chương trình tổng hợp tiếng Việt được nghiên cứu và phát triển bởi các chuyên gia nghiên cứu của trung tâm nghiên cứu MICA. Chương trình được viết trên IDE Visual C++ 6.0. Phiên bản đầu tiên của chương trình được hoàn thành vào năm 2007 với sự tham gia của Lê Xuân Hùng và TS. Trần Đỗ Đạt. Chương trình cho kết quả tiếng nói tổng hợp khá tự nhiên và dễ hiểu. Chương trình này sẽ được tích hợp sử dụng trong hệ thống phát triển trong đồ án.

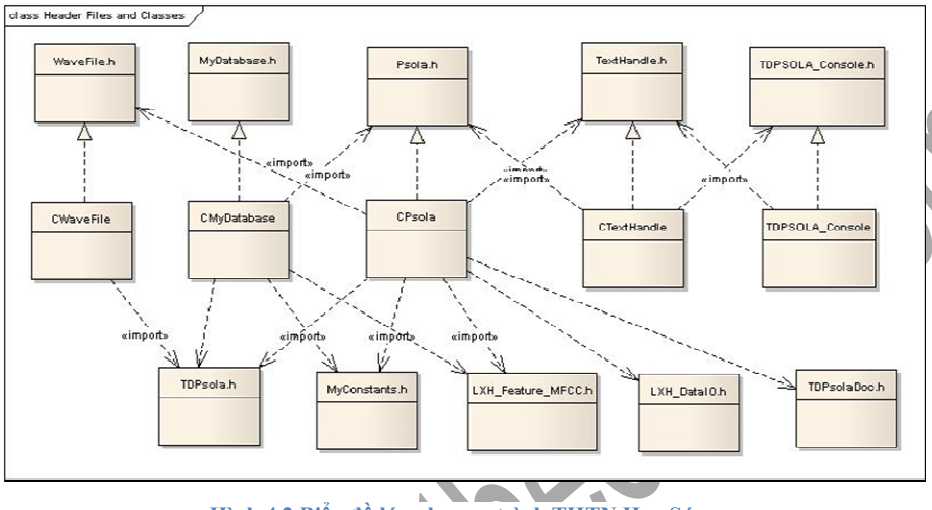


Hình 4.1 Sơ đồ hoạt động tổng quát của chương trình

Chương trình được thiết kế theo phương pháp hướng đối tượng, bao gồm các lớp chính liên quan tới module tổng hợp tiếng nói:

- CPsola: chứa các phương thức dùng cho việc điều khiển các tham số ngữ điệu của tiếng nói và ghép nối các đơn vị âm, làm trơn năng lượng giữa các âm tiết.
- CTextHandle: chứa các phương thức về phân tích văn bản đầu vào, lấy ra các thông tin về âm tiết...

- CMydabase: chứa các phương thức để thao tác với file cơ sở dữ liệu, tìm kiếm, lấy ra thông tin về các đơn vị âm... cũng như phương thức chọn lựa đơn vị âm tối ưu.
- CWaveFile: chứa các phương thức thao tác với file .wav như tạo mới, thêm dữ liệu vào một file .wav đã tồn tại...



Hình 4.2 Biểu đồ lớp chương trình THTN Hoa Súng

Chức năng các hàm của các lớp chính trong chương trình

Tên lớp	Tên hàm	Nội dung
TextHandle	GetSylCount	Lấy số lượng âm tiết cần để tổng hợp.
	GetSylInfo	Lấy ra các thông tin về âm tiết tại vị trí truyền vào.
	ParseSylInfo	Phân tích cấu trúc âm tiết, tìm xem có bao nhiêu thành phần cấu thành âm tiết.
	ToneInPhrase	Tính F0Array theo loại thanh điệu của đơn vị âm, fNorF0, và độ dài đơn vị âm.
	UpdateSylInfo	Cập nhật lại thông tin âm tiết.
Mydatabase	ToneDistance	Tính khoảng cách thanh điệu giữa hai thanh

Tên lớp	Tên hàm	Nội dung
	ContextDistance	Tính khoảng cách ngữ điệu giữa hai đơn vị âm ghép nối, phụ thuộc vào ToneDistance
	GetUnitContext	Lấy ra thông tin về ngữ điệu của đơn vị âm, đầu vào là âm tiết hiện tại, âm tiết bên trái, âm tiết bên phải; đầu ra là một UNITINFO
	DistanceMatrix	Tính ma trận khoảng cách ngữ điệu giữa các đơn vị âm thuộc hai lớp liền kề nhau. Gọi tới hàm ContextDistance để tính.
	ComputeShortestPathArray	Tính đường đi ngắn nhất trong số ma trận khoảng cách
	ListSelUnit	Gọi tới các hàm DistanceMatrix và ComputeShortestPathArray. Liệt kê các đơn vị âm có thể dùng để tổng hợp.
	BestUnitSelection	Lựa chọn các mẫu có thể cho các đơn vị âm dựa trên ngữ điệu.
	GetPosBestUnit	Lấy ra đường đi ngắn nhất của các đơn vị âm mẫu.
	Find	Trả về vị trí của một đơn vị âm trong cơ sở dữ liệu.
	FindAll	Trả về vị trí của một đơn vị âm trong cơ sở dữ liệu.
	FindNext	Tìm kiếm vị trí một đơn vị âm trong cơ sở dữ liệu kể từ vị trí tham số truyền vào.
	GetAUnitInfo	Lấy ra các thông tin về đơn vị âm trong cơ sở dữ liệu
	GetElementLen	Lấy độ dài của đơn vị âm trong cơ sở dữ liệu, giá trị trả về kiểu DWORD

Tên lớp	Tên hàm	Nội dung
	GetMFCCBegin	Lấy 12 hệ số đầu của đơn vị âm.
	GetMFCCEnd	Lấy 12 hệ số cuối của đơn vị âm.
	MFCCdistance	Tính khoảng cách MFCC giữa 12 hệ số MFCC đầu và cuối.
	SpectralDistance	Tính toán khoảng cách phổ giữa hai đơn vị âm, phụ thuộc vào hàm MFCCdistance.
Psola	GetMinValue	Lấy giá trị nhỏ nhất trong mảng từ vị trí đầu mảng tới vị trí chỉ số tham số truyền vào.
	GetMaxValue	Lấy giá trị lớn nhất trong mảng từ vị trí đầu mảng tới vị trí chỉ số tham số truyền vào.
	GetMaxAbsValue	Lấy giá trị tuyệt đối lớn nhất trong mảng từ vị trí đầu mảng tới vị trí chỉ số tham số truyền vào.
	GetMaxPositiveValuePos	Trả về vị trí có giá trị dương lớn nhất của lpSignal.
	GetMaxNegativeValuePos	Trả về vị trí có giá trị âm lớn nhất của lpSignal.
	GetMaxTo	Lấy ra giá trị lớn nhất của chu kì T0.
	GetAvgTo	Lấy ra giá trị trung bình của chu kì T0.
	HanningWnd	Hàm lấy cửa sổ Hanning tại vị trí giữa chu kì.
	HanningLeft	Hàm lấy cửa sổ Hanning tại vị trí bên trái chu kì.
	HanningRight	Hàm lấy cửa sổ Hanning tại vị trí bên phải chu kì.
	ChangeAmplitude	Thay đổi biên độ của lpSignal theo hệ số truyền vào.
	SansAccentFromPh	Xác định vị trí các điểm pitch mark khi tần

Tên lớp	Tên hàm	Nội dung
	one	số F0 không đổi.
	AccentGraveFromPhone	Xác định vị trí các điểm pitch mark khi tần số F0 giảm đều.
	AccentAiguFromPhone	Xác định vị trí các điểm pitch mark khi tần số F0 tăng đều.
	CreateNewPhone	Ghép nối các đơn vị âm để tạo thành âm tiết.
	CreateSyllable	Ghép nối các đơn vị âm để tạo thành âm tiết.
	EnergySmooth	Làm mượt sự gián đoạn giữa năng lượng của hai âm tiết liền nhau bằng phương pháp TD-PSOLA.
WaveFile	CreateWaveFile	Tạo ra file wave với các thông số đầu vào là đặc tính file wave.
	AppendToWaveFile	Thêm dữ liệu vào cuối một file wave có sẵn.
	SeparateData	Lấy dữ liệu ra hai kênh.
	Convert16To8MuLaw	Chuyển đổi định dạng file wave từ 16 bit về 8 bit theo luật Mu.

4.2 Tổ chức cơ sở dữ liệu

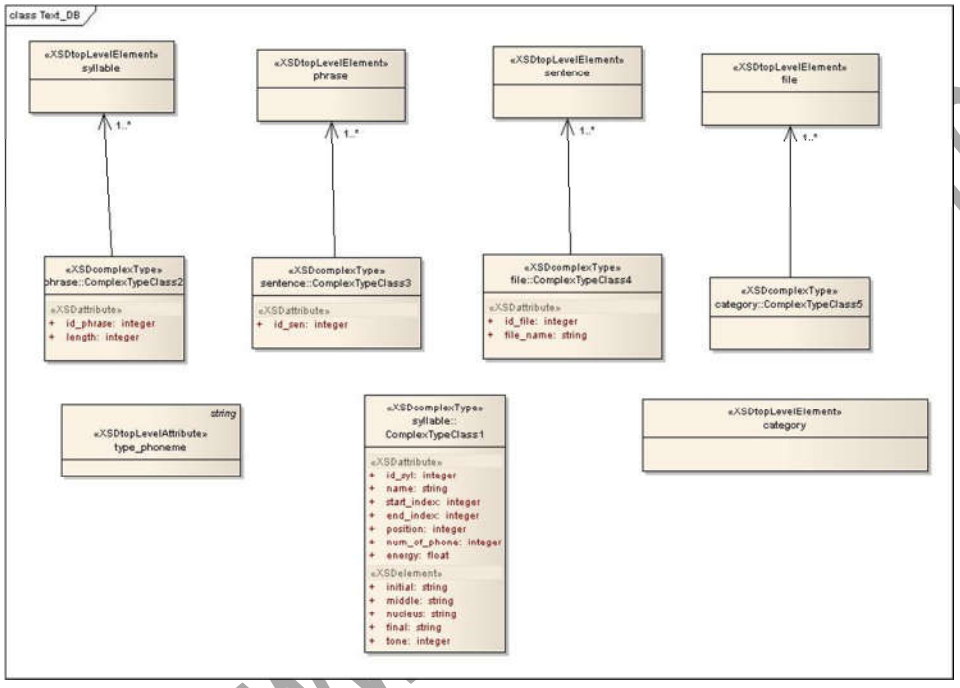
Trong phương pháp tìm kiếm đơn vị âm không đồng nhất, CSDL bao gồm các loại: CSDL âm thanh, CSDL văn bản tương ứng và CSDL bán âm tiết.

4.2.1 Cơ sở dữ liệu âm thanh.

CSDL âm thanh được lưu thành các file wav theo từng đoạn văn hoặc đoạn hội thoại tương ứng với CSDL văn bản. Các file wav này có chung một định dạng trong phần header để tạo điều kiện dễ dàng khi ghép nối các phần dữ liệu nhỏ trong đó. Kích thước CSDL âm thanh là 68M, độ dài 37 phút. Kích thước cơ sở dữ liệu như vậy được đánh giá là có thể chấp nhận được đối với bộ tổng hợp tiếng nói.

4.2.2 Cơ sở dữ liệu văn bản.

Dữ liệu được dùng trong tổng hợp là các đoạn văn, hội thoại tiếng Việt được thu âm bởi một giọng đọc duy nhất. Dữ liệu văn bản bao gồm 250 đoạn của 630 câu với 10852 mẫu của khoảng 1600 âm tiết phân biệt, được tổ chức theo cấu trúc XML như bảng dưới đây. Các thông tin của âm tiết như về thành phần cấu tạo, năng lượng, thanh điệu, trường độ được phân tích offline và lưu trữ để phục vụ cho quá trình lựa chọn và ghép nối đơn vị. Dưới đây là cấu trúc file XML dùng để lưu trữ CSDL văn bản.



Hình 4.3 Cấu trúc CSDL XML

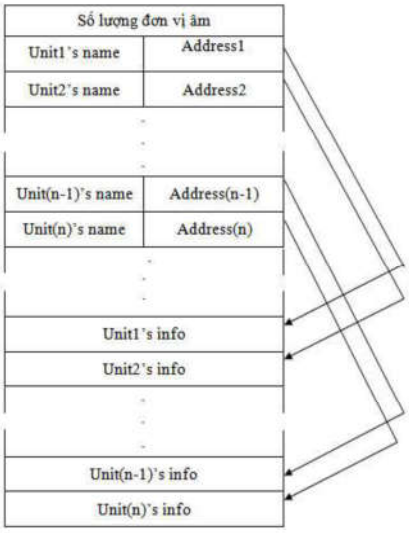
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
	<xs:attribute name="type_phoneme" use="optional" type="xs:string"/>
	<xs:element name="syllable">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="initial" type="xs:string"/>
				<xs:element name="middle" type="xs:string"/>
				<xs:element name="nucleus" type="xs:string"/>
				<xs:element name="final" type="xs:string"/>
				<xs:element name="tone" type="xs:integer"/>
			</xs:sequence>
			<xs:attribute name="id_syl" use="optional" type="xs:integer"/>
			<xs:attribute name="name" use="optional" type="xs:string"/>
			<xs:attribute name="start_index" use="optional" type="xs:integer"/>
			<xs:attribute name="end_index" use="optional" type="xs:integer"/>
			<xs:attribute name="position" use="optional" type="xs:integer"/>
			<xs:attribute name="num_of_phone" type="xs:integer"/>
			<xs:attribute name="energy" use="optional" type="xs:float"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="phrase">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="syllable" maxOccurs="unbounded"/>
			</xs:sequence>
			<xs:attribute name="id_phrase" use="optional" type="xs:integer"/>
			<xs:attribute name="length" use="optional" type="xs:integer"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="sentence">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="phrase" maxOccurs="unbounded"/>
			</xs:sequence>
			<xs:attribute name="id_sen" use="optional" type="xs:integer"/>
		</xs:complexType>
	</xs:element>
	<xs:element name="file">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="sentence" maxOccurs="unbounded"/>
			</xs:sequence>
```

```
                    <xs:attribute name="id_file" use="optional" type="xs:integer"/>
                    <xs:attribute name="file_name" use="optional" type="xs:string"/>
                </xs:complexType>
        </xs:element>
        <xs:element name="category">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="file" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
        </xs:element>
</xs:schema>
```

4.2.3 Cơ sở dữ liệu bán âm tiết

Cơ sở dữ liệu bán âm tiết là CSDL được dùng trong chương trình tổng hợp tiếng nói Hoa Súng đã được trình bày tại 4.1. Cấu trúc tổng quát của CSDL bao gồm các phần như sau [1] :

- Mở đầu CSDL là 2 bytes chứa số lượng đơn vị âm có trong CSDL.
- Tiếp theo, CSDL được chia thành các khối 8 bytes, 4 bytes đầu chứa tên đơn vị âm, 4 bytes còn lại chứa địa chỉ của phần dữ liệu của đơn vị âm tương ứng. Các khối này tạo thành phần header, có độ lớn là 50000 bytes.
- Tiếp theo đến phần dữ liệu của các đơn vị âm.

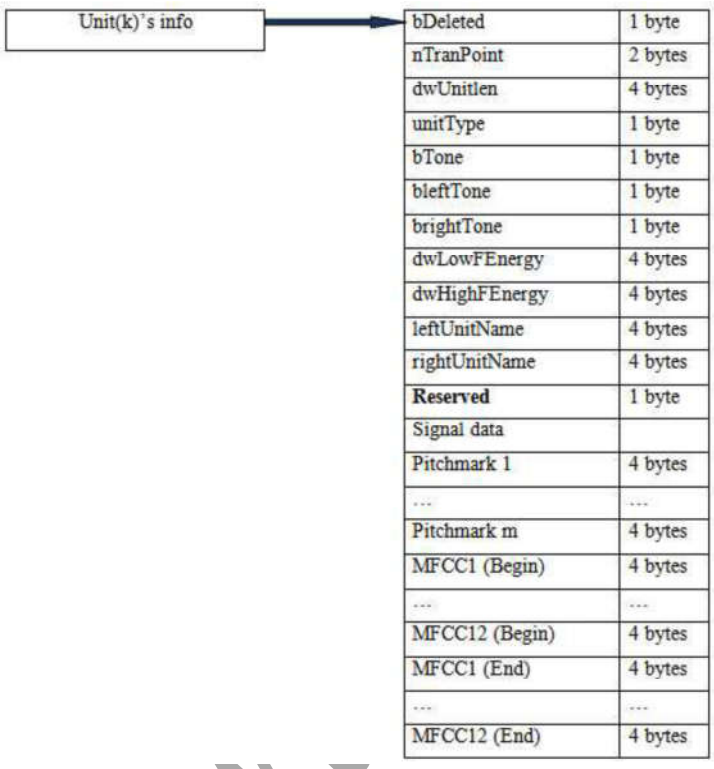


Hình 4.4 Cấu trúc CSDL bán âm tiết

Dữ liệu về một đơn vị âm trong CSDL bao gồm các thành phần:

Thành phần	Kích thước (byte)	Nội dung
bDeleted	1	Cho biết đơn vị âm có trong CSDL hay không
nTranPoint	2	Vị trí điểm chuyển giao giữa thành phần âm hữu thanh và vô thanh
dwUnitLen	4	Độ dài của đơn vị âm
unitType	1	Loại đơn vị âm
bTone	1	Loại thanh điệu của đơn vị âm
bleftTone	1	Loại thanh điệu của đơn vị âm bên trái
brightTone	1	Loại thanh điệu của đơn vị âm bên phải
dwLowFEnergy	4	Năng lượng thành phần tần số thấp của đơn vị âm
dwHighEnergy	4	Năng lượng thành phần tần số cao của đơn vị âm
leftUnitName	4	Tên của đơn vị âm bên trái
rightUnitName	4	Tên của đơn vị âm bên phải
Reversed	1	Byte dự trữ
Signal data		Dữ liệu của đơn vị âm, tùy vào từng đơn vị âm mà kích thước khác nhau
Pitchmark i	4	Các giá trị pitch mark, gồm m giá trị

MFCCi	4	12 hệ số MFCC cuối của đơn vị âm.

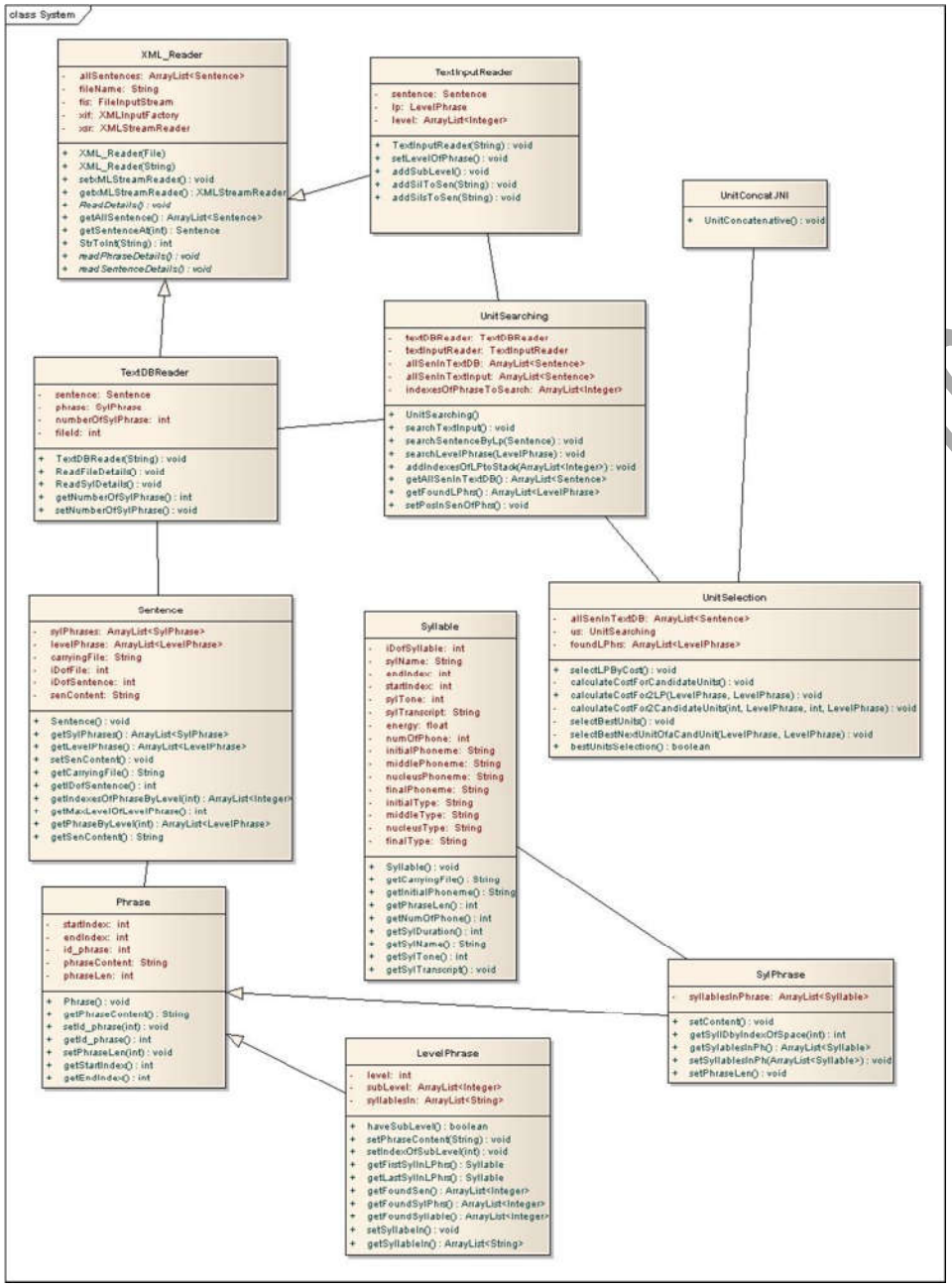


Hình 4.5 Thông tin của một đơn vị âm trong CSDL

4.3 Thiết kế lớp

4.3.1 Biểu đồ lớp

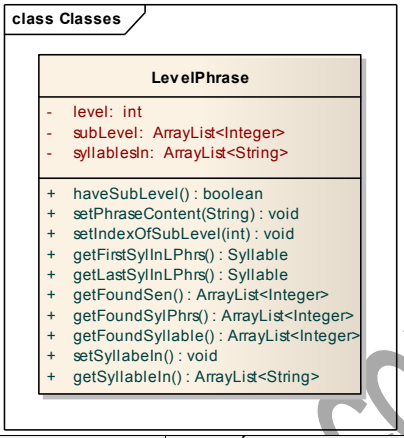
Chương trình được thiết kế theo phương pháp hướng đối tượng trên công cụ là Enterprise Architect 7.0. Biểu đồ lớp của chương trình như hình Hình 4.6 dưới đây.



Hình 4.6 Biểu đồ lớp của chương trình

4.3.2 Thiết kế chi tiết lớp

Chương trình được cài đặt với các lớp chính như sau:



Tên lớp: LevelPhrase

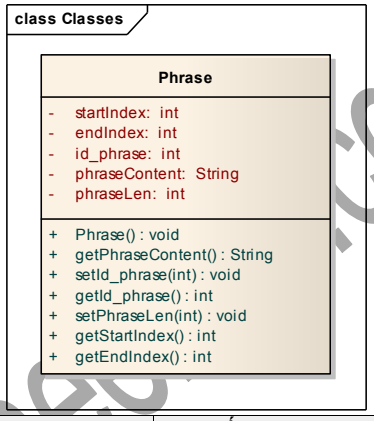
Chức năng: Lưu trữ các thông tin về các cụm từ trong câu văn bản đầu vào.

Phương thức	Ghi chú	Tham số
haveSubLevel() boolean Public	Kiểm tra xem cụm từ này có con là cụm từ khác hay không.	
setPhraseContent() void Public	Thiết lập nội dung của cụm từ.	**String** [in] phraseCont
getFirstSylInLPhrs() Syllable Public	Lấy âm tiết đầu tiên trong cụm từ	
getLastSylInLPhrs() Syllable Public	Lấy âm tiết cuối cùng của cụm từ	
getFoundSen() ArrayList<Integer> Public	Trả về mảng chỉ số của các câu trong CSDL mà tìm thấy cụm từ này trong đó	
getFoundSylPhrs() ArrayList<Integer> Public	Trả về mảng chỉ số của các SylPhrase trong CSDL mà tìm thấy cụm từ này trong đó	
getFoundSyllable() ArrayList<Integer> Public	Trả về mảng chỉ số của âm tiết đầu tiên trong SylPhrase tìm thấy cụm từ trong đó	
setSyllabeIn() void	Từ nội dung của cụm từ, thiết lập nội dung của	

Phương thức	Ghi chú	Tham số
Public	các âm tiết trong cụm từ.	
getSyllableIn() ArrayList<String> Public	Trả về nội dung của các âm tiết trong cụm từ	

Tên lớp: Phrase

Chức năng: Lưu trữ thông tin về các cụm từ, là superclass của LevelPhrase và SylPhrase.



Phương thức	Ghi chú	Tham số
Phrase() void Public	Hàm khởi tạo	
getPhraseContent() String Public	Trả về nội dung của cụm từ	
setId_phrase() void Public	Thiết lập Id của cụm từ trong câu chứa nó	**int** [in] id_phrase
getId_phrase() int Public	Trả về Id của cụm từ	
setPhraseLen() void Public	Thiết lập chiều dài của cụm từ, dựa trên số lượng âm tiết.	**int** [in] phraseLen
getStartIndex() int	Trả về vị trí của frame đầu tiên chứa cụm từ	

Phương thức	Ghi chú	Tham số
Public	trong CSDL âm thanh.	
getEndIndex() int Public	Trả về vị trí của frame cuối cùng chứa cụm từ trong CSDL âm thanh.	

Tên lớp: Sentence

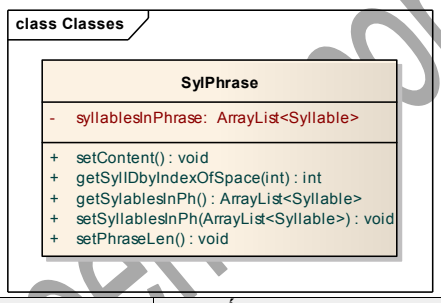
Chức năng: Lưu trữ câu trong CSDL và văn bản đầu vào.

Phương thức	Ghi chú	Tham số
Sentence() void Public	Hàm khởi tạo	
getSylPhrases() ArrayList<SylPhrase> Public	Trả về mảng các SylPhrase có trong câu, dùng trong trường hợp câu này là câu trong CSDL	
getLevelPhrase() ArrayList<LevelPhrase> Public	Trả về mảng các LevelPhrase có trong câu, dùng trong trường hợp câu này là câu đầu vào để tổng hợp	
getPhraseByLevel() ArrayList<LevelPhrase> Public	Lấy các cụm từ cùng mức trong câu đầu vào	**int** [in] level
setSenContent() void Public	Thiết lập nội dung của câu dựa vào nội dung của các SylPhrase	
getCarryingFile() String Public	Trả về tên file trong CSDL chứa câu	
getIDofSentence() int Public	Trả về Id của câu	
getIndexesOfPhraseByLevel() ArrayList<Integer> Public	Trả về mảng chỉ số của các LevelPhrase trong câu có cùng mức	**int** [in] level

Phương thức	Ghi chú	Tham số
getMaxLevelOfLevelPhrase() int Public	Trả về chỉ số mức lớn nhất, tức là chỉ số của cụm từ sâu nhất trong cây phân cấp	
getSenContent() String Public	Trả về xâu chứa nội dung của câu	

Tên lớp: SylPhrase

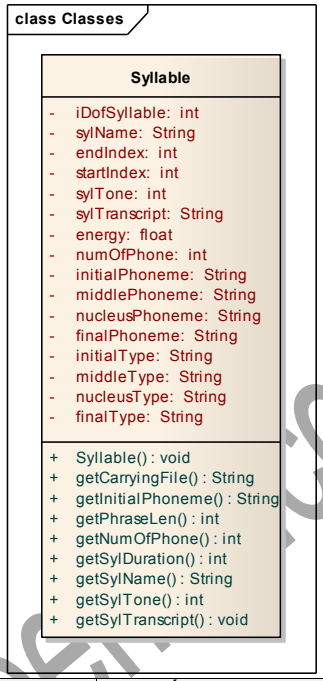
Chức năng: Lưu trữ các cụm từ trong CSDL.



Phương thức	Ghi chú	Tham số
setPhraseLen() void Public	Thiết lập chiều dài của cụm từ dựa trên số âm tiết trong cụm từ, không tính khoảng lặng và dấu câu	
setContent() void Public	Thiết lập nội dung của cụm từ dựa vào các âm tiết trong đó	
getSylIDbyIndexOfSpace() int Public	Trả về Id của âm tiết theo vị trí dấu cách truyền vào	**int** [in] indexOfSpace
getSylablesInPh() ArrayList<Syllable> Public	Trả về mảng các âm tiết trong cụm từ	
setSyllablesInPh() void Public	Thiết lập mảng các âm tiết trong cụm từ	**ArrayList<Syllable>** [in] syllabes

Tên lớp: Syllable

Chức năng: Lưu trữ thông tin về âm tiết, các thành phần, thuộc tính của âm tiết trong CSDL.

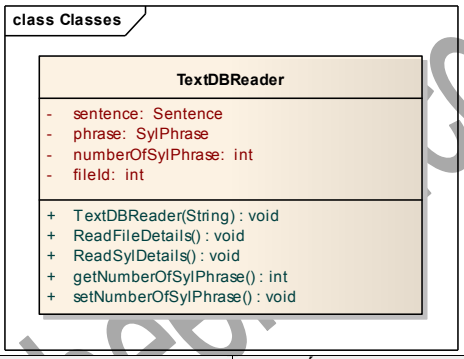


Phương thức	Ghi chú	Tham số
Syllable() void Public	Hàm khởi tạo	
getCarryingFile() String Public	Trả về tên file chứa âm tiết này	
getInitialPhoneme() String Public	Trả về âm đầu của âm tiết	
getPhraseLen() int Public	Trả về chiều dài của cụm từ chứa âm tiết này	
getNumOfPhone() int Public	Trả về số lượng âm vị trong âm tiết	
getSylDuration() int Public	Trả về trường độ của âm tiết	
getSylName() String	Trả về tên của âm tiết, là nội dung của âm tiết	

Phương thức	Ghi chú	Tham số
Public		
getSylTone() int Public	Trả về thanh điệu của âm tiết	
getSylTranscript() void Public	Trả về phiên âm của âm tiết	

Tên lớp: TextDBReader

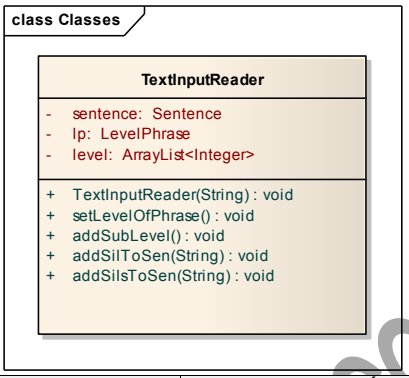
Chức năng: Đọc CSDL văn bản, lưu vào bộ nhớ trong.



Phương thức	Ghi chú	Tham số
TextDBReader() void Public	Hàm khởi tạo, đọc file có tên từ biến truyền vào	**String** [in] textDBLocation
ReadFileDetails() void Public	Được gọi tới khi gặp thẻ có tên "file", đọc Id và tên của file	
ReadSylDetails() void Public	Được gọi tới khi gặp thẻ có tên "syllable", đọc các thông tin chi tiết của âm tiết trong CSDL	
getNumberOfSylPhrase() int Public	Trả về số lượng của cụm từ trong CSDL	
setNumberOfSylPhrase() void Public	Thiết lập số lượng các cụm từ trong toàn bộ CSDL	

Tên lớp: TextInputReader

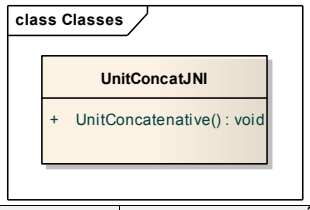
Chức năng: đọc văn bản đầu vào, lưu vào bộ nhớ trong.



Phương thức	Ghi chú	Tham số
TextInputReader() void Public	Bắt đầu quá trình đọc file văn bản đầu vào có tên là tham số truyền vào	**String** [in] str
setLevelOfPhrase() void Public	Tổ chức các cụm từ thành cấu trúc cây phân cấp, gán chỉ số	
addSilToSen() void Public	Thêm khoảng lặng của dấu phẩy vào câu	**String** [in] str
addSilsToSen() void Public	Thêm khoảng lặng của dấu chấm vào câu	**String** [in] sils

Tên lớp: UnitConcatJNI

Chức năng: kết nối code C với phần code Java, gọi tới module tổng hợp mức thấp.

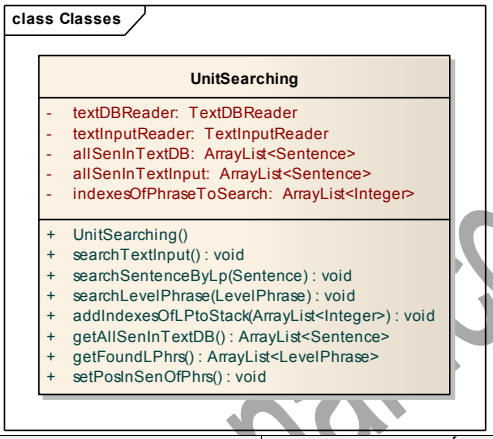


Phương thức	Ghi chú	Tham số
UnitConcatenative() void	Gọi tới module tổng hợp mức thấp viết trên C++	**String** [in] filelocation

Phương thức	Ghi chú	Tham số
Public		

Tên lớp: UnitSearching

Chức năng: Tìm kiếm cụm từ trong CSDL.

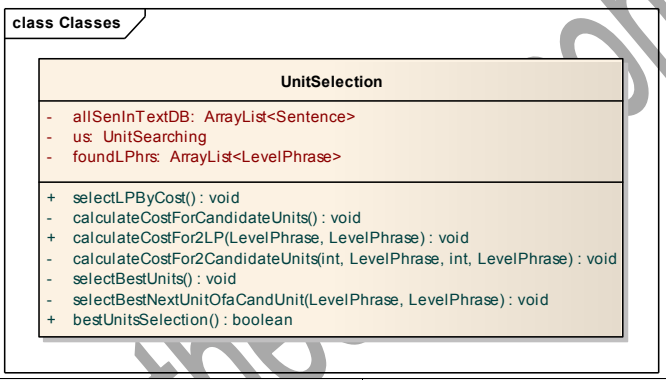


Phương thức	Ghi chú	Tham số
UnitSearching() Public	Hàm khởi tạo. Đọc CSDL văn bản và văn bản đầu vào vào bộ nhớ trong và bắt đầu tìm kiếm	
searchTextInput() void Public	Tìm kiếm văn bản đầu vào bằng cách duyệt qua các câu và tìm kiếm từng câu một	
searchSentenceByLp() void Public	Tìm kiếm câu s theo từng cụm từ	**Sentence** [in] s
searchLevelPhrase() void Public	Tìm kiếm cụm từ trong CSDL	**LevelPhrase** [in] lp
addIndexesOfLPtoStack() void Public	Thêm chỉ số của các cụm từ con của cụm từ đang tìm kiếm vào trong stack để tìm kiếm các cụm từ con	**ArrayList<Integer>** [in] indexes
getAllSenInTextDB() ArrayList<Sentence>	Lấy mảng tất cả các câu trong CSDL văn bản	

Phương thức	Ghi chú	Tham số
Public		
getFoundLPhrs() ArrayList<LevelPhrase> Public	Trả về các cụm từ được tìm thấy trong CSDL	
setPosInSenOfPhrs() void Public	Thiết lập vị trí của cụm từ được tìm thấy trong câu chứa nó	

Tên lớp: UnitSelection

Chức năng: lựa chọn đơn vị tối ưu trong CSDL.

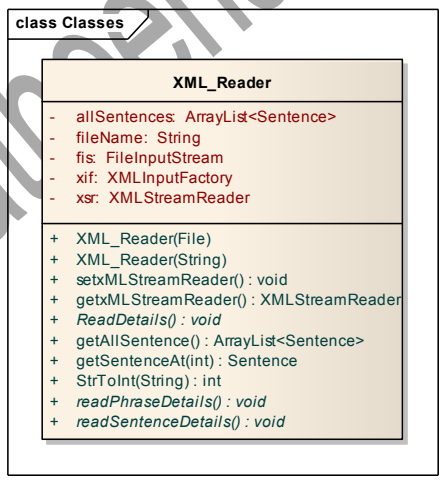


Method	Notes	Parameters
selectLPByCost() void Public	Lựa chọn chuỗi đơn vị âm tối ưu theo tối thiểu hóa hàm chi phí. Bắt đầu quá trình tìm kiếm đơn vị âm tối ưu	
calculateCostForCandidateUnits() void Private	Tính toán hàm chi phí giữa tất cả các đơn vị ứng viên	
calculateCostFor2LP() void Public	Tính toán hàm chi phí cho các đơn vị ứng viên của hai đơn vị âm đích	**LevelPhrase** [in] rightLP Đơn vị âm đích bên phải **LevelPhrase** [in] leftLP Đơn vị âm đích bên trái
calculateCostFor2CandidateUnits() void Private	Tính toán hàm chi phí giữa hai đơn vị ứng viên	**int** [in] indexOfCandOfRightLP **LevelPhrase** [in] rightLP

Method	Notes	Parameters
		int [in] indexOfCandOfLeftLP **LevelPhrase** [in] leftLP
selectBestUnits() void Private	chọn lựa chuỗi đơn vị âm tối ưu	
selectBestNextUnitOfaCandUnit() void Private	Thiết lập chỉ số của đơn vị ứng viên tốt nhất liền sau đơn vị ứng viên của đơn vị âm đích đang xét	**LevelPhrase** [in] nextUnit **LevelPhrase** [in] currentUnit
bestUnitsSelection() boolean Public	Lựa chọn đơn vị ứng viên tối ưu. Trả về true nếu tìm thấy.	

Tên lớp: XML_Reader

Chức năng: Đọc file XML, là superclass cho class **TextInputReader** và **TextDBReader**.



Phương thức	Ghi chú	Tham số
XML_Reader() Public	Hàm khởi tạo, bắt đầu đọc file XML với đầu vào là một File	**File** [in] inputFile
XML_Reader() Public	Hàm khởi tạo, bắt đầu đọc file XML với đầu vào là tên file	**String** [in] nameFile

Phương thức	Ghi chú	Tham số
setxMLStreamReader() void Public	Thiết lập XMLStreamReader	
getxMLStreamReader() XMLStreamReader Public	Trả về XMLStreamReader	
ReadDetails() void *abstract* Public	Hàm abstract, đọc các thông tin chi tiết, sẽ được cài đặt trong lớp con	
getAllSentence() ArrayList<Sentence> Public	Trả về mảng các câu được đọc	
getSentenceAt() Sentence Public	Trả về câu có chỉ số được truyền vào trong mảng các câu	**int** [in] indexOfSentence
StrToInt() int Public	Chuyển đổi số được lưu ở dạng String về dạng int	**String** [in] str
readPhraseDetails() void *abstract* Public	Hàm abstract, đọc các thông tin về cụm từ, sẽ được cài đặt trong lớp con	
readSentenceDetails() void *abstract* Public	Hàm abstract, đọc các thông tin chi tiết về câu, sẽ được cài đặt trong lớp con	

4.4 Kết quả và đánh giá

Chương trình đã được cài đặt bằng hai ngôn ngữ Java và C++. Tính tới thời điểm viết đồ án này, tác giả đã thực hiện tổng hợp thành công ở mức âm tiết và cụm từ. Nhằm đánh giá chất lượng của tiếng nói tổng hợp, tác giả đã chuẩn bị dữ liệu để làm các bài đánh giá cảm thụ của người nghe đối với tiếng nói.

4.4.1 Bài đánh giá cảm thụ

4.4.1.1 Mục tiêu

Đánh giá chất lượng của tiếng nói tổng hợp được bởi chương trình của tác giả dựa trên hai tiêu chí là *độ rõ ràng trong phát âm* và *độ tự nhiên* của tiếng nói tổng hợp. Các tiêu chí được đánh giá theo thang điểm từ 1 tới 5 theo như bảng sau:

Tiêu chí đánh giá	Thang điểm và giải thích
Độ rõ ràng trong phát âm	1. *Không thể phân biệt* 2. *Không phân biệt rõ* 3. *Hơi rõ* 4. *Đủ rõ để phân biệt* 5. *Rất rõ*
Độ tự nhiên của tiếng nói	1. *Không hề tự nhiên* 2. *Không tự nhiên lắm* 3. *Có tự nhiên* 4. *Tự nhiên* 5. *Rất tự nhiên*

Nhiệm vụ của người tiến hành đánh giá là nghe các câu tổng hợp và cho điểm đánh giá các câu theo hai tiêu chí trên.

4.4.1.2 Phương pháp thực hiện

Dữ liệu đánh giá

Dữ liệu được dùng để đánh giá là các câu hoặc đoạn văn ngắn tiếng Việt được lấy ngẫu nhiên từ trên web. Các đoạn này được đưa vào hai hệ thống tổng hợp để so sánh và đánh giá. Hệ thống thứ nhất là bộ tổng hợp tiếng nói Hoa Súng, hệ thống thứ hai là hệ thống tác giả phát triển trong đồ án.

Các đoạn văn hoặc câu tổng hợp dùng để làm bài đánh giá	
Đoạn 1	Hôm nay tôi muốn gửi tâm sự của mình , rất mong các bạn cho tôi

	những lời khuyên bổ ích.
Đoạn 2	Các nhà khoa học vừa nghiên cứu một loại túi thông minh. Túi thông minh sẽ giải quyết vấn đề đau đầu trên. Khi túi thông minh phát hiện thực phẩm quá hạn sử dụng , túi sẽ biến sắc , hơn nữa , túi còn có chức năng bảo đảm sự tươi mới của thực phẩm.
Đoạn 3	Cuộc sống đồng bào dân tộc vùng cao là đề tài hấp dẫn cho sáng tác ảnh , đặc biệt là sáng tác ảnh nghệ thuật.
Đoạn 4	Những chính sách không đến được với dân , thường là do cách làm hay cách nói khác. Đó là sự quan tâm nửa mùa của những người ra chính sách.
Đoạn 5	Ông đã tham dự nhiều hội thảo , đọc báo cáo kinh nghiệm ở nhiều nơi , được mời đến giảng bài ở một số trường đại học.
Đoạn 6	Quả thuốc phiện khô có trọng lượng từ sáu trăm ki lô gam đến dưới một nghìn năm trăm ki lô gam. Quả thuốc phiện tươi có trọng lượng từ một trăm năm mươi ki lô gam đến dưới bốn trăm năm mươi ki lô gam.
Đoạn 7	Nhà nước có nhiều chính sách quan tâm đến hoạt động của sinh viên tại nước ngoài.

Bài thí nghiệm được tiến hành trong phòng thu âm của trung tâm MICA với số lượng tám người, đều là người ở những vùng miền phía Bắc và nói giọng phía Bắc chuẩn. Để tiến hành bài đánh giá, tác giả sử dụng một chương trình viết bằng Java có giao diện như Hình 4.7. Hai mẫu của cùng một câu sẽ được sắp xếp liền kề nhau nhưng người thực hiện sẽ được nghe với thứ tự ngẫu nhiên.

Hình 4.7 Giao diện chương trình đánh giá

4.4.1.3 Phân tích kết quả

Kết quả đánh giá của người tham gia được lưu trữ trong file XML. Sau khi xử lý, tác giả thu được các bảng kết quả dưới đây Bảng 4.1 và Bảng 4.2.

Theo kết quả được thống kê, HT 2 cho kết quả tốt hơn HT 1 về cả hai tiêu chí đánh giá. Điều này có thể được lý giải do HT 2 dùng âm tiết và cụm từ để ghép nối nên không có nhiều điểm ghép nối như HT 1. Phần lớn các câu được tổng hợp trong HT 2 đều được đánh giá cao hơn, chỉ duy đoạn 5, khi được tổng hợp trong HT 2 cho kết quả thấp hơn theo tiêu chí 2. Điều này do trong đoạn 5, âm tiết "*nhiều*" không được tổng hợp tốt nên không nghe được rõ. Ngoài ra, âm tiết "*sáng*" trong đoạn 3, âm tiết "*của*" trong đoạn 7 cũng không nghe được rõ.

Đối với HT 2, điểm cho độ rõ ràng được 4.00, mức này được đánh giá là cao trong thang điểm 5, còn điểm cho độ tự nhiên thì thấp hơn, được 3.64. Việc này được giải thích là do HT 2 chưa sử dụng phương pháp điều khiển các tham số ngữ điệu, ngữ điệu của câu là ngữ điệu của các âm tiết trong các câu khác nhau, nên khi ghép lại, ngữ điệu của câu tổng hợp không khớp với ngữ cảnh của câu.

Tập dữ liệu dùng để đánh giá là các câu trần thuật, vì vậy ngữ điệu của câu tổng hợp trong HT 2 khá là giống tự nhiên. Tuy nhiên, nếu dùng các câu khác như câu hỏi, câu cầu khiên, câu cảm thán, ngữ điệu của các câu tổng hợp của HT 2 sẽ không giống vì chưa áp dụng các mô hình ngữ điệu và kĩ thuật điều khiển tham số ngữ điệu.

Bảng 4.1 Kết quả về độ rõ ràng

Người đánh giá	Câu 1		Câu 2		Câu 3		Câu 4		Câu 5		Câu 6		Câu 7	
	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2
Người 1	3	5	3	5	3	4	3	4	3	4	3	5	3	4
Người 2	3	5	3	5	2	4	2	4	3	4	3	5	3	4
Người 3	4	4	4	4	4	2	3	4	3	3	3	4	4	5
Người 4	3	3	4	5	2	3	3	3	2	3	2	5	2	3
Người 5	2	2	3	3	1	2	3	3	1	2	2	3	2	4
Người 6	5	5	4	5	4	5	4	5	5	4	3	5	4	5
Người 7	3	4	3	5	3	4	3	4	3	5	3	4	4	4
Người 8	3	3	2	4	2	4	1	4	3	4	3	4	2	5
Điểm trung bình	3.25	3.88	3.25	4.50	2.63	3.50	2.75	3.88	2.88	3.63	2.75	4.38	3.00	4.25

HT 1: hệ thống tổng hợp tiếng nói Hoa Súng.

HT 2: hệ thống đồ án phát triển.

Điểm trung bình HT1: 2.93

Điểm trung bình HT2: 4.00

Bảng 4.2 Bảng kết quả về độ tự nhiên

Người đánh giá	Câu 1		Câu 2		Câu 3		Câu 4		Câu 5		Câu 6		Câu 7	
	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2	HT 1	HT 2
Người 1	3	4	3	4	3	3	3	3	3	2	2	5	3	3
Người 2	5	4	4	4	3	3	3	3	4	3	4	3	5	3
Người 3	3	4	3	4	4	4	3	4	4	4	3	4	4	4
Người 4	3	3	3	4	2	4	3	3	3	3	3	5	3	3
Người 5	1	2	2	2	2	3	2	3	2	2	2	3	2	3
Người 6	5	5	4	5	3	5	3	5	5	5	3	4	4	5
Người 7	3	3	2	4	3	4	2	4	4	4	2	3	3	4
Người 8	2	2	2	4	3	5	2	4	2	3	3	4	2	4
Điểm trung bình	3.13	3.38	2.88	3.88	2.88	3.88	2.63	3.63	3.38	3.25	2.75	3.88	3.25	3.63

Điểm trung bình HT1: 2.98

Điểm trung bình HT2: 3.64

4.5 Kết luận chương

Trong chương này, tác giả đã trình bày về thiết kế cơ sở dữ liệu và thiết kế lớp để phát triển chương trình của tác giả. Sau khi thực hiện các bài đánh giá, kết quả cho thấy chương trình phát triển đã đạt được kết quả ban đầu khả quan. Tuy nhiên, hạn chế lớn nhất của chương trình là chưa tổng hợp được mức bán âm tiết và sử dụng phương pháp điều khiển tham số ngữ điệu của hệ thống trong 4.1. Đây sẽ là nhiệm vụ trong thời gian sắp tới để chương trình hoàn thiện hơn.

Kết luận và hướng phát triển

Mục tiêu đồ án đặt ra là đề xuất phương pháp chọn lựa đơn vị âm tối ưu cho tổng hợp tiếng nói tiếng Việt và thực thi phương pháp. Phương pháp được tác giả đề xuất và áp dụng trong đồ án là "*Lựa chọn đơn vị không đồng nhất*". Tác giả cũng đã phát triển chương trình thực thi phương pháp với việc sử dụng CSDL và bộ tổng hợp mức bán âm tiết của Trung tâm nghiên cứu Mica. Điều này cho thấy sự đúng đắn trong hướng đi và cách áp dụng phương pháp đề xuất của tác giả. Mặc dù đánh giá kết quả bước đầu tương đối khả quan nhưng chương trình mới chỉ làm được những phần việc rất nhỏ trong một bộ tổng hợp tiếng nói.

Phần sau là tổng kết về những gì đã làm được trong đồ án, những điểm hạn chế và hướng đi trong tương lai của đề tài:

- Những điểm đã đạt được:
  - Tìm hiểu lý thuyết về tổng hợp tiếng nói và tổng hợp mức thấp.
  - Đề xuất cách áp dụng phương pháp lựa chọn đơn vị không đồng nhất cho tổng hợp tiếng nói tiếng Việt.
  - Tổ chức CSDL văn bản và âm thanh thuận lợi cho việc tìm kiếm và mở rộng sau này.
  - Cài đặt chương trình trên ngôn ngữ Java và C++, kết nối hai phần với nhau qua JNI.
- Những điểm còn hạn chế
  - Chưa thực hiện được phần ghép nối bán âm tiết và điều khiển tham số ngữ điệu trên C++;
  - Việc lựa chọn đơn vị âm tối ưu chưa thực hiện đối với bán âm tiết.
  - Số lượng người tham gia bài thực nghiệm còn ít.
  - Chưa dùng các tham số ngữ điệu trong tính toán hàm khoảng cách để tối ưu hóa, hiện mới chỉ dùng các tham số ngữ âm.
  - Chưa áp dụng mô hình ngữ điệu về trường độ và cao độ.
- Hướng đi trong tương lai
  - Xây dựng một bộ dữ liệu lớn hơn, đảm bảo độ phủ cao hơn đối với âm tiết tiếng Việt.
  - Nghiên cứu kĩ hơn việc ảnh hưởng của các tham số trong hàm khoảng cách ở mức âm tiết và cụm từ.
  - Áp dụng các mô hình ngữ điệu cho các loại câu khác nhau trong tiếng Việt.

- Chỉnh sửa thiết kế để chương trình có khả năng thích nghi với cơ sở dữ liệu mới.
- Giảm bớt sự phụ thuộc của việc lựa chọn đơn vị âm vào kết quả của cây phân tích cú pháp, có thể cho kết quả đúng khi cây phân tích cú pháp cho kết quả sai.
- Xem xét cách áp dụng mô hình ngôn ngữ và thống kê để lựa chọn cụm từ.

Tài liệu tham khảo

[1] Lại Hoàng Nam, Quách Đại Quang, "*Xây dựng chương trình tổng hợp tiếng nói trên DSP*", đồ án tốt nghiệp K49, ĐH Bách Khoa Hà Nội, 2009.

[2] Lukas Latacz, Yuk On Kong, Werner Verhelst, "*Unit Selection Synthesis Using Long Non-Uniform Units and Phonemic Identity Matching*", Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, 2007.

[3] Marcello Balestri, Alberto Pacchiotti, Silvia Quazza, Pier Luigi Salza, Stefano Sandri, "*Choose the best to modigy the least: a new generation concatenative synthesis system*", CSELT - Centro Studi e Laboratori Telecomunicazioni S.p.A., Torino, Italy.

[4] Mark Tatham, Katherine Morton, "*Development in Speech Synthesis*", Wiley, 2005.

[5] Min Chu,Hu Peng, Hong-yun Yang, Eric Chang, "*Selecting non-uniform units from a very large corpus for cancatenative speech synthesizer*", Microsoft Research China, Beijing.

[6] Minghui Dong, Kim-Teng Lua, Haizhou Li, "*A Unit Selection-based Speech Synthesis Approach for Mandarin Chinese*", Institute for Infocomm Research.

[7] Paul Taylor, "*Text-to-SpeechSynthesis",* University of Cambridge, Cambridge University Press, 2006.

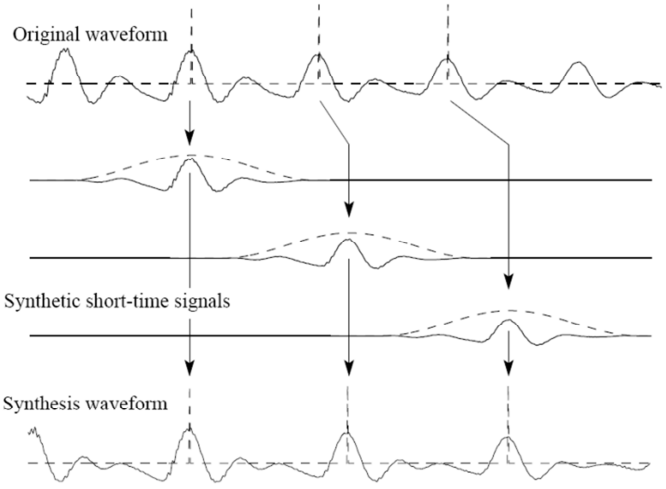
[8] Tian-Swee Tan and Sh-Hussain, "*Implementation of Phonetic Context Variable Length Unit Selection Module for Malay Text to Speech",* Faculty of Biomedical Engineering and Health Science, University Teknologi Malaysia,, Malaysia, 2008.

[9] Trần Đỗ Đạt, "*Synthèse de la parole a partir du texte en langue Vietnamienne*", Ph.D. Thesis, Thèse en cotutelle international MICA, Hanoi, 2007.

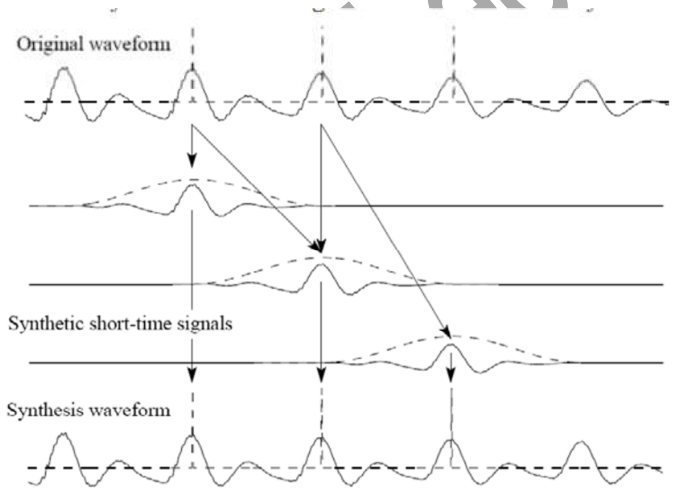
[10] Vũ Hải Quân, Cao Xuân Nam, "*Tổng hợp tiếng nói tiếng Việt, theo phương pháp ghép nối cụm từ*". Tập V-1, Số 1, tháng 04/2009

[11] Xuedong Huang, Alejandro Acero, Hsiao-Wuen Hon, "*Spoken language processing*", Prentice Hall, 2001.

Phụ lục



Hình 4.8 Biến đổi cao độ tín hiệu bằng TD-PSOLA



Hình 4.9 Biến đổi trường độ với TD-PSOLA