TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Tổng hợp tiếng nói tiếng Việt chất lượng cao và mô hình caching

LÊ MINH NGUYỄN
nguyen.lm162998@sis.hust.edu.vn
Ngành Kỹ Thuật Phần Mềm

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

Bộ môn: Kỹ thuật phần mềm
Viện: Công nghệ thông tin – Truyền thông

HÀ NỘI, 07/2021

Lời cam kết

Họ và tên sinh viên: Lê Minh Nguyễn

Điện thoại liên lạc: 0382334747 Email: nguyen.lm162998@sis.hust.edu.vn

Lớp: CNTT2.02 Hệ đào tạo: Kỹ Sư

Em – *Lê Minh Nguyễn* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân em dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng em, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Em xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 18 tháng 07 năm 2021

Tác giả ĐATN

Lê Minh Nguyễn

Lời cảm ơn

Chắc có lẽ với nhiều người tuổi thanh xuân là một khoảng thời gian đẹp nhất, với em điều này cũng vậy và càng đẹp hơn khi những tháng ngày thanh xuân đó dành trọn cho Bách Khoa. Nhớ những ngày đầu đi nhập học cùng mẹ, xen lẫn sự tự hào là bao điều bỡ ngỡ, bước vào trong cánh cổng parabol là một ngôi trường rộng lớn và đâu đâu cũng thấy các anh chị mặc áo trường, lúc đó trong đầu là sự thầm mong một ngày được như các anh chị và tốt nghiệp được Bách Khoa.

Thời gian thấm thoát trôi đưa, 5 năm đại học cũng gần kết thúc và bây giờ có lẽ là thời gian quan trọng nhất với mỗi sinh viên Bách Khoa – thời gian làm đồ án tốt nghiệp (ĐATN). ĐATN của em không thể hoàn thành nếu như không có sự hướng dẫn của thầy cô, sự giúp đỡ của anh em bạn bè, sự động viên từ gia đình và sự nỗ lực từ chính bản thân. Em xin chân thành cảm ơn cô Nguyễn Thị Thu Trang người đã hướng dẫn em nhiệt tình và chu đáo để em hoàn thành đồ án này. Cô là một người cô giáo có tâm và có tầm, em luôn cảm thấy thần tượng và nể phục cô không chỉ bởi sự tâm huyết với sinh viên mà còn là những hành động đẹp trong cuộc sống, được làm việc cùng cô trong vòng 2 năm vừa qua có lẽ là một điều may mắn đối với em ở thời sinh viên. Em xin cảm ơn anh Nguyễn Hoàng Kỳ một cựu sinh viên CNPM K59, người anh đã giúp đỡ em tận tình khi em gặp khó khăn, em cảm ơn những kiến thức và kỹ năng làm việc mà anh đã chỉ dạy cho em. Tớ xin cảm ơn Thành Và Duy, hai người bạn gắn bó nhất của tớ ở đại học, cảm ơn những tình cảm và sự giúp đỡ của các cậu dành cho tớ. Em xin cảm ơn tới những người anh em bạn bè ở phòng Lab914 thân thương, nơi mọi người luôn giúp đỡ nhau trong học tập. Con xin cảm ơn bố mẹ, chú dì, các em và người thân những người luôn đứng phía sau giúp đỡ, động viên con trong suốt 5 năm học. Và cuối cùng là lời cảm ơn tới chính bản thân tôi đã miệt mài, cố gắng trong suốt quá trình học tập để hoàn thành đồ án này

Năm năm học đã qua là những năm tháng cháy hết mình, dành trọn tuổi thanh xuân cho Bách Khoa và bây giờ nhìn lại không còn cảm thấy điều gì nuối tiếc. Cảm ơn Bách Khoa đã mang tới cho em những điều tuyệt vời nhất, những kỷ niệm sẽ không bao giờ quên, nếu có được chọn lại thì câu trả lời vẫn mãi là Bách Khoa.

Tóm tắt

Chuyển đổi văn bản sang giọng nói đang là một công nghệ phổ biến và có nhiều ứng dụng thực tiễn trong cuộc sống hiện nay như tổng đài nhân tạo, báo nói, thuyết minh phim … Với sự phát triển mạnh của những mô hình mạng học sâu giọng nói ngày càng trở nên tự nhiên hơn, tuy nhiên những mô hình này thường rất phức tạp dẫn tới tốc độ phản hồi còn chậm và tốn kém tài nguyên xử lý. Chính vì vậy đồ án này đã phát triển hệ thống tổng hợp tiếng nói chất lượng cao FHG có thời gian phản hồi nhanh đồng thời đề xuất mô hình TTS Caching nhằm tiết kiệm chi phi khi có thể tái sử dụng những phần nội dung đã tổng hợp. Mô hình này đặc biệt cần thiết với các hệ thống cần tổng hợp theo mẫu, như hệ thống tổng đài nhân tạo, callbot …

Hệ thống FHG được xây dựng dựa trên mô hình âm học phi hồi quy (Non Autoregressive) FastSpeech2 và mô hình sinh tiếng nói Hifi-Gan. Hệ thống này sử dụng một dạng dữ liệu trung gian gọi là mel-spectrogram trong quá trình xử lý, mô hình âm học Fastspeech2 sẽ dự đoán mel-spectrogram từ văn bản đầu vào, sau đó mô hình sinh tiếng nói Hifi-Gan tái tạo âm thanh từ mel-spectrogram đó. Hệ thống TTS Caching được xây dựng dựa trên thuật toán Viterbi – một thuật toán quy hoạch động trong tìm kiếm chuỗi tối ưu, thuật toán Viterbi sẽ tìm kiếm chuỗi các đoạn tiếng nói của những nội dung đã tổng hợp, sau đó tiến hành tính toán chi phí bao gồm chi phí lựa chọn và chi phí ghép nối của chuỗi các đoạn tiếng nói đó, từ đó đưa ra quyết định tái sử dụng cho văn bản đầu vào.

Hệ thống FHG hiện tại đã được triển khai thực tế ở trang web tổng hợp tiếng nói tiếng Việt tại địa chỉ https://vbee.vn/ và nhận được những phản hồi tích cực của người dùng, chất lượng giọng nói tổng hợp đạt sự tự nhiên 90% so với giọng người thật, tốc độ phản hồi tăng từ 4-5 lần so với những mô hình cũ. Hệ thống TTS Caching hiện tại đã được triển khai thử nghiệm cho hệ thống tổng đài nhân tạo - AI Call Center tại địa chỉ https://cp-dev.aicallcenter.vn/ , kết quả đánh giá cho thấy hệ thống giúp tối ưu hóa tốc độ tổng hợp tiếng nói của hệ thống FHG lên 2-3 lần với mẫu câu có nội dung lặp lại, chi phí tổng hợp tiết kiệm được 30% và chất lượng giọng nói sử dụng cơ chế caching đạt sự tương đồng tới 95% so với giọng nói tổng hợp từ toàn bộ văn bản.

Mục lục

Danh mục bảng

Danh mục hình vẽ

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
DNN	Deep Neural Network Mạng học sâu
HMM	Hidden Markov Model Mô hình Hidden Markov
NLP	Natural Language Processing Xử lý ngôn ngữ tự nhiên
Seq2Seq	Sequence to Sequence Chuyển đổi dãy sang dãy
TTS	Text To Speech Chuyển đổi văn bản thành tiếng nói
ĐATN	Đồ án tốt nghiệp
CSDL	Cơ sở dữ liệu

Danh mục thuật ngữ

End-To-End	Đầu cuối
Caching	Lưu trữ
Autoregressive	Mô hình hồi quy
Non-Autoregressive	Mô hình phi hồi quy
Training	Quá trình huấn luyện
Inference	Quá trình suy dẫn
Best Sequence of States	Chuỗi các trạng thái tốt nhất

Chương 1 Tổng hợp tiếng nói tiếng Việt

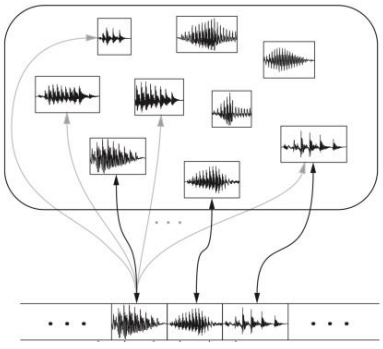
Chương này giới thiệu những vấn đề thực tế dẫn tới lý do chọn đề tài. Sau đó đưa ra những mục tiêu, phạm vi của đề tài, định hướng giải pháp và bố cục trình bày của đồ án.

1.1 Tổng quan về tổng hợp tiếng nói

Chuyển đổi văn bản sang giọng nói là một ứng dụng trí tuệ nhân tạo thuộc lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP)[1] đang được quan tâm và chú ý hiện nay. Một hệ thống tổng hợp tiếng nói dựa trên văn bản là một hệ thống tiếp nhận dữ liệu văn bản đầu vào, sau đó tiến hành tiền xử lý và cuối cùng chuyển hóa đoạn văn bản thành tiếng nói. Những hệ thống tổng hợp tiếng nói được bắt đầu nghiên cứu và phát triển từ những năm 1970, theo chiều dài lịch sự phát triển rất nhiều công nghệ và mô hình tổng hợp tiếng nói khác nhau đã được phát triển nhằm tăng chất lượng của giọng nói tổng hợp. Các mô hình đó có thể chia thành ba loại chính dựa theo công nghệ sử dụng: (i) Công nghệ tổng hợp dựa trên ghép nối, (ii) Công nghệ tổng hợp dựa trên tham số thống kê, và (iii) Công nghệ tổng hợp đầu cuối End-To-End.

1.1.1 Công nghệ tổng hợp dựa trên ghép nối

Những hệ thống tổng hợp tiếng nói ghép nối dựa là những hệ thống dựa trên những đoạn tiếng nói đã được thu âm, sau đó ghép nối lại với nhau để tạo thành tiếng nói tổng hợp. Tiếng nói có thể được thu âm theo nhiều mức đa dạng như câu, từ, âm tiết, ... Trong quá trình tổng hợp hệ thống sẽ tìm kiếm tiếng nói từ cơ sở dữ liệu dựa trên văn bản đầu vào, sau đó ghép nối lại để tạo thành tiếng nói đầu ra.



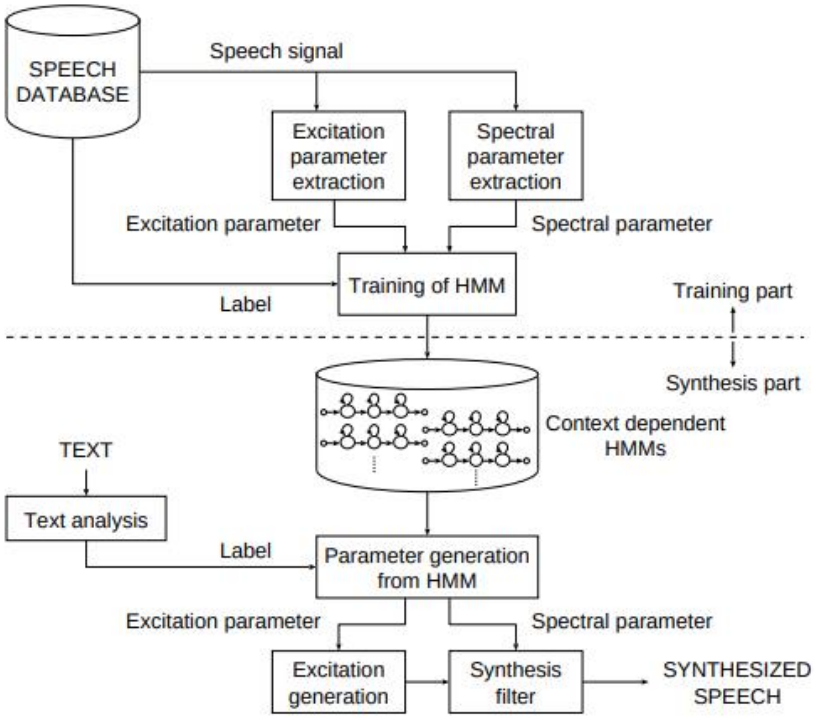
Hình 1.1 Hệ thống tổng hợp tiếng nói dựa trên ghép nối [7]

Những hệ thống này có ưu điểm là giữ lại được những đặc trưng riêng của người nói. Tuy nhiên hệ thống này tồn tại những nhược điểm lớn như (i) tại những vị trí ghép nối thường tiếng nói không được trơn tru, (ii) Việc thiết kế một cơ sở dữ liệu lớn chứa tất cả các đơn vị âm học ở tất cả các ngữ cảnh là việc làm tốn rất nhiều công sức và gần như không thể, (iii) tiếng nói ghép nối thường mất đi sự tự nhiên. Vậy nên nhu cầu đặt ra là cần có những hệ thống mới hơn để khắc phục điều này.

1.1.2 Công nghệ tổng hợp dựa trên tham số thống kê

Những hệ thống tổng hợp tiếng nói dựa trên tham số thống kê được phát triển để giải quyết những giới hạn của mô hình ghép nối. Ý tưởng của những hệ thống này là việc ước lượng tiếng nói bởi các tham số, sau đó huấn luyện mô hình tổng hợp để dự đoán ra những tham số đó và cuối cùng là bước xử lý để biến đổi những tham số đó thành tiếng nói. Hai hệ thống tổng hợp tiếng nói dựa trên kỹ thuật này là (a) Hệ thống tổng hợp tiếng nói dựa trên HMM và (b) Hệ thống tổng hợp tiếng nói dựa trên DNN

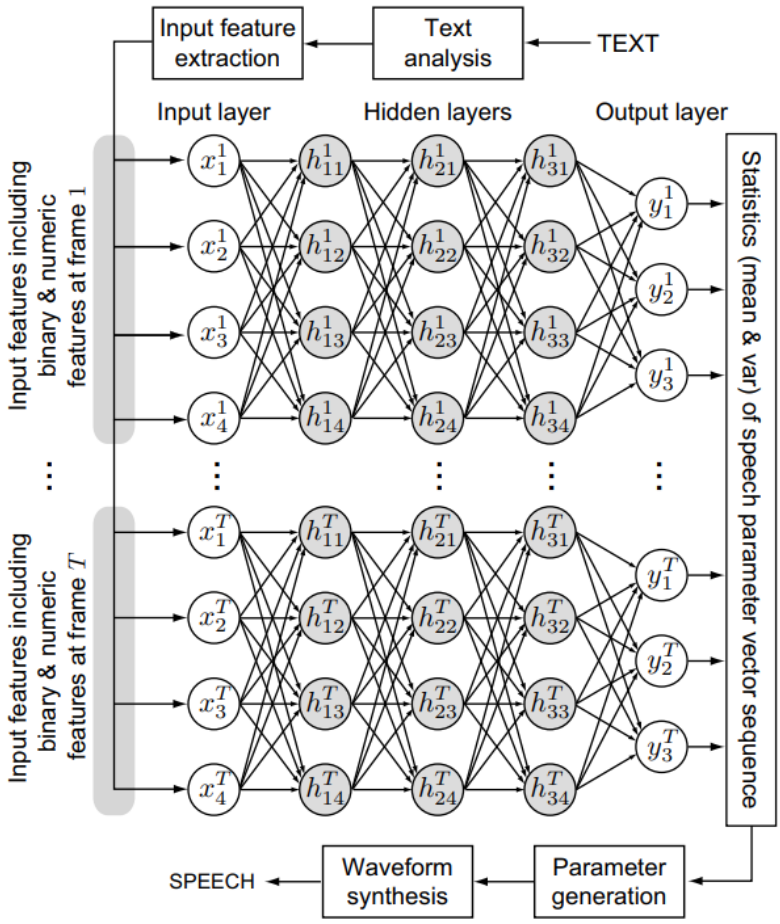
a) Hệ thống tổng hợp tiếng nói dựa trên HMM



Hình 1.2 Kiến trúc tổng quan hệ thống tổng hợp tiếng nói dựa trên HMM [19]

Mô hình tổng hợp tiếng nói dựa trên HMM đã phát triển mạnh mẽ trong những thập kỷ trước. Mô hình này có nhiều ưu điểm nổi trội so với những hệ thống tổng hợp tiếng nói dựa trên ghép nối trước đó như: (i) dễ dàng thay đổi những đặc trưng về tiếng nói, (ii) mô-đun ngôn ngữ nhỏ gọn, và (iii) có thể dễ dàng áp dụng cho nhiều loại ngôn ngữ khác nhau.

b) Hệ thống dựa trên DNN

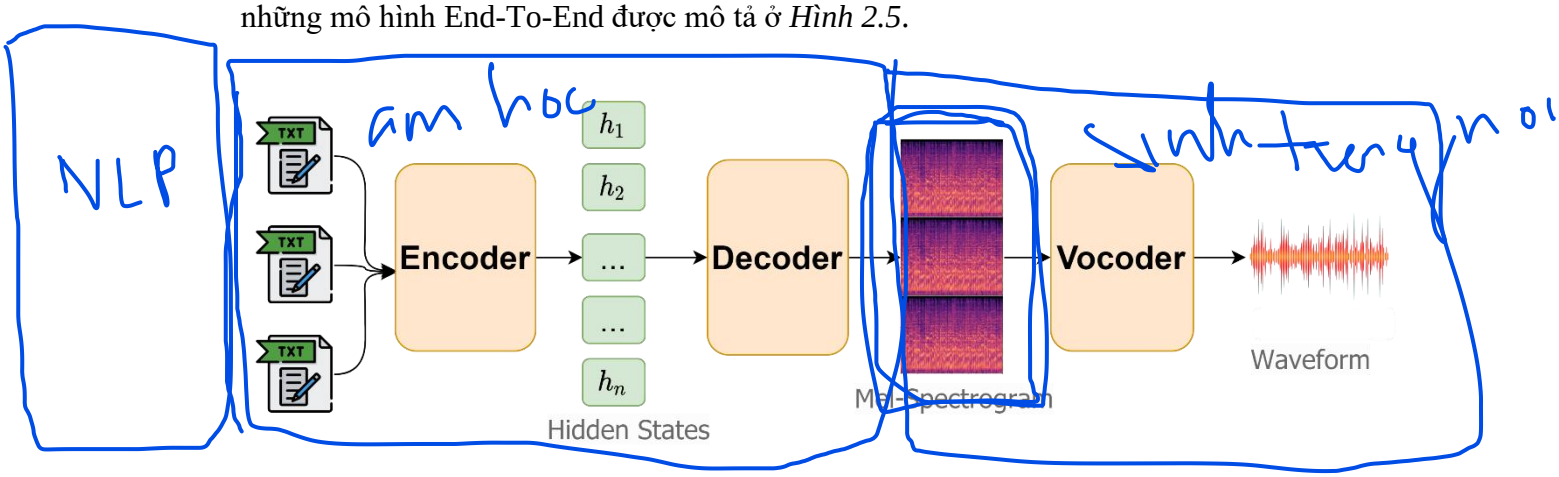


Hình 1.3 Kiến trúc tổng quan hệ thống tổng hợp tiếng nói dựa trên DNN [23]

Với những mô hình như HMM thì một hạn chế lớn nhất là về chất lượng của tiếng nói được tổng hợp. Vì vậy mô hình DNN đã được đề xuất để cải thiện chất lượng. Với mô hình DNN thì mối quan hệ giữa văn bản và những đặc trưng âm học sẽ được mô hình hóa bởi một cấu trúc mạng học sâu (Deep Neural Network). Việc sử dụng DNN có thể giải quyết được những giới hạn mà những mô hình như HMM gặp phải.

1.1.3 Công nghệ tổng hợp tiếng nói hiện đại End-To-End

Những mô hình tổng hợp tiếng nói dựa trên thống kê như HMM, DNN đã đánh dấu được những bước phát triển vượt trội trong lĩnh vực trong tổng hợp tiếng nói. Những mô hình này đã khắc phục được những nhược điểm của mô hình ghép nối trước đó và đã tạo ra được tiếng nói có chất lượng cao và biểu cảm hơn trước. Tuy nhiên những hệ thống như này thì khá phức tạp để có thể thiết kế dữ liệu cho quá trình huấn luyện và yêu cầu người dùng cần có hiểu biết sâu về mặt ngôn ngữ, điều này làm cho khó tiếp cận với nhiều người. Một vài năm trở lại đây với sự phát triển vượt bậc về những mô hình mạng học sâu đã có nhiều mô hình mới được phát triển, và điển hình là những mô hình End-To-End. Kiến trúc tổng quan về những mô hình End-To-End được mô tả ở *Hình 2.5*.



Hình 2.5 Kiến trúc tổng quan hệ thống tổng hợp tiếng nói End-To-End

Những mô hình End-To-End sử dụng một dạng dữ liệu trung gian gọi là mel-spectrogram giữa văn bản và âm thanh, dạng dữ liệu này biểu diễn các tín hiệu âm thanh dưới dạng tần số. Mô hình End-To-End có những ưu điểm lớn như: (i) dễ dàng tiền xử lý dữ liệu và huấn luyện mô hình, (ii) giọng nói tổng hợp có sự tự nhiên gần ngang bằng với con người.

Ở trên thế giới đã có nhiều hệ thống tổng hợp chất lượng cao được xây dựng như: Google TTS[1], Microsoft TTS[2], … Những công ty công nghệ lớn này đang tập trung vào việc nghiên cứu những mô hình tổng hợp tiếng nói hiện đại End-To-End nhằm mục đích tạo ra giọng nói có sự tự nhiên ngang bằng con người. Một vài mô hình End-To-End nổi tiếng có thể kể tới là Tacotron [2] (Google 2017), TransformerTTS [3] (Microsoft 2019), những mô hình này

[1] https://cloud.google.com/text-to-speech/

[2] https://www.microsoft.com/en-in/p/text-to-speech-tts

được xây dựng mô hình hồi quy (Autoregressive [8]) hay FastSpeech2 [5] (Microsoft 2021) được xây dựng theo mô hình phi hồi quy (Non Autoregressive [8]).

1.2 Tổng hợp tiếng nói tiếng Việt và những vấn đề

Tổng hợp tiếng nói tiếng Việt đã được quan tâm và nghiên cứu tại Việt Nam từ đầu những năm 2000 bởi sự cần thiết trong các ứng dụng cuộc sống. Máy đọc sách Sao Mai của trung tâm hỗ trợ người mù TP HCM phát triển vào năm 2004 được xem là hệ thống tổng hợp tiếng nói đầu tiên tại Việt Nam. Tiếp sau đó một vài hệ thống tổng hợp tiếng nói dựa trên ghép nối được được phát triển như [28], [29], [30] … tuy nhiên nhược điểm của những mô hình cũ này là chất lượng giọng nói tổng hợp còn kém tự nhiên. Hiện nay ở Việt Nam nhiều doanh nghiệp lớn đã bắt đầu xây dựng những hệ thống tổng hợp tiếng nói chất lượng cao sử dụng những mô hình End-To-End hồi quy (Autoregressive) hiện đại, có thể kể đến như: Vbee TTS[3], Viettel TTS [4], …

Việc áp dụng những mô hình tổng hợp tiếng nói End-To-End hồi quy (Autoregressive) có ưu điểm là sinh ra giọng nói có chất lượng cao, sự tự nhiên có thể ngang bằng với con người, tuy nhiên những hệ thống này có nhiều nhược điểm lớn như tốc độ tổng hợp chậm, mất từ, lặp từ, … dẫn tới những mô hình này không thể áp dụng được trong thực tế.

Một trong những ứng dụng phổ biến của tổng hợp tiếng nói tại Việt Nam đó là tổng đài nhân tạo. Tổng đài nhân tạo là hệ thống tổng đài mà con người bị thay thế một phần hoặc hoàn toàn bởi máy tính, đặc thù của những ứng dụng tổng đài nhân tạo là phản hồi theo thời gian thực và tổng hợp lặp lại nội dung theo mẫu nhất định. Vì vậy những hệ thống này yêu cầu tái sử dụng những nội dung đã tổng hợp để giảm thời gian xử lý và tiết kiệm chi phí.

1.3 Mục tiêu và phạm vi đề tài

Xuất phát từ những phân tích ở trên, mục tiêu của đồ án là tập trung nghiên cứu, phát triển hai hệ thống: (i) hệ thống tổng hợp tiếng nói chất lượng cao, tốc độ phản hồi nhanh và có sự ổn định, (ii) hệ thống caching giúp tối ưu tốc độ và tiết kiệm chi phí tổng hợp của những hệ thống ứng dụng tổng hợp giọng nói có nội dung lặp lại theo mẫu câu.

[3] https://vbee.vn/

[4] https://viettelgroup.ai/service/tts

Nhằm mục tiêu tổng hợp được tiếng nói chất lượng cao có tốc độ phản hồi nhanh và sự ổn định, đồ án sẽ tập trung nghiên cứu và phát triển hệ thống tổng hợp tiếng nói chất lượng cao - FHG sử dụng mô hình âm học End-To-End FastSpeech2 [5] và mô hình sinh tiếng nói Hifi-Gan [15]. Ngoài ra đồ án sẽ nghiên cứu và phát triển hệ thống caching nhằm tối ưu tốc độ tổng hợp và tiết kiệm chi phí cho những hệ thống ứng dụng tổng hợp giọng nói có nội dung lặp lại theo mẫu câu. Theo hiểu biết của em hiện nay thì ở trên thế giới và Việt Nam chưa có các nghiên cứu khoa học nào đề xuất xây dựng hệ thống caching cho những ứng dụng tổng hợp tiếng nói theo mẫu lặp lại.

1.4 Định hướng giải pháp

Với những mục tiêu và phạm vi đã nêu ra ở mục 1.3, đồ án sẽ đề xuất nghiên cứu và phát triển hệ thống tổng hợp tiếng nói tiếng Việt sử dụng công nghệ hiện đại End-To-End. Cụ thể hệ thống tổng hợp tiếng nói chất lượng cao - FHG được xây dựng dựa trên mô hình âm học phi hồi quy (Non Autoregressive [8]) FastSpeech2 [5] và mô hình sinh tiếng nói Hifi-Gan [15]. Hai mô hình này được huấn luyện với dữ liệu tiếng nói tiếng Việt, sau đó được tinh chỉnh để cải thiện chất lượng chất lượng và tối ưu hóa tham số để triển khai được trong thực tế. Thêm vào đó đồ án này sẽ đề xuất nghiên cứu và xây dựng một hệ thống caching cho những ứng dụng tổng hợp tiếng nói có đặc thù lặp lại theo mẫu và yêu cầu đầu ra theo thời gian thực. Hệ thống caching giúp tối ưu về mặt tốc độ tổng hợp và tiết kiệm được chi phí tổng hợp, hệ thống này được xây dựng dựa trên thuật toán Viterbi [6] nhằm mục đích tái sử dụng lại những nội dung đã được tổng hợp.

Hệ thống tổng hợp tiếng nói FHG sẽ có trách nhiệm tổng hợp tiếng nói chất lượng cao, ổn định từ văn bản đầu vào. Ngoài ra hệ thống này còn có khả năng dự đoán đặc trưng âm học từ văn bản và xử lý để sinh ra giọng nói từ những đặc trưng đó. Những đặc trưng âm học có thể nêu ra như: độ cao (pitch), năng lượng (energy), trường độ (duration), ... Hệ thống caching có nhiệm vụ lưu trữ những đoạn tiếng nói đã được tổng hợp cùng với các đặc trưng âm học tương ứng, tiến hành tìm kiếm, phân tích khả năng tái sử dụng lại những đoạn tiếng nói đã tổng hợp dựa trên các đặc trưng âm học cung cấp sao cho tỷ lệ tái sử dụng là cao nhất và chất lượng sát với âm thanh được tổng hợp từ toàn bộ văn bản nhất.

1.5 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Trong Chương 2 đồ án sẽ trình bày về cơ sở lý thuyết của các mô hình tổng hợp tiếng nói End-To-End, sau đó đồ trình bày về mô hình âm học FastSpeech [5] và mô hình sinh tiếng nói Hifi-Gan [15]. Cuối cùng là giới thiệu về thuật toán Viterbi và ứng dụng trong việc xây dựng hệ thống caching cho tổng hợp tiếng nói tiếng Việt.

Trong Chương 3 đồ án sẽ trình bày đề xuất mô hình tổng hợp tiếng nói chất lượng cao cho tiếng Việt sử dụng công nghệ hiện đại End-To-End. Cụ thể đồ án sẽ trình bày về kiến trúc tổng quan hệ thống, những thành phần chính trong hệ thống bao gồm: (i) thành phần xử lý ngôn ngữ tự nhiên, (ii) mô hình hóa âm học và (iii) mô hình sinh tiếng nói, sau đó sẽ trình bày chi tiết từng thành phần. Cuối cùng là phần trình bày về cách đánh giá chất lượng giọng nói và kết quả đạt được.

Trong Chương 4 đồ án sẽ đưa ra đề xuất về mô hình caching cho những ứng dụng tổng hợp tiếng nói có đặc thù lặp lại theo mẫu. Cụ thể đồ án này sẽ trình bày về mô hình tổng quan của một hệ thống caching, những thành phần chính trong mô hình sẽ bao gồm: (i) thành phần thuật toán tìm kiếm và (ii) thành phần quản lý và lưu trữ và (iii) Thành phần ghép nối các đoạn tiếng nói, sau đó đồ án sẽ đi vào trình bày chi tiết từng thành phần trong mô hình. Cuối cùng là phần trình bày về cách đánh giá hiệu năng, chất lượng giọng nói và kết quả đạt được.

Tiếp theo Chương 5 đồ án sẽ đưa giải pháp thiết kế, xây dựng, cách thức triển khai cho hệ thống tổng hợp tiếng nói FHG và hệ thống TTS Caching.

Cuối cùng trong Chương 6 sẽ đưa ra tổng kết về kết quả đạt được của đồ án, sau đó là những định hướng nghiên cứu trong tương lai.

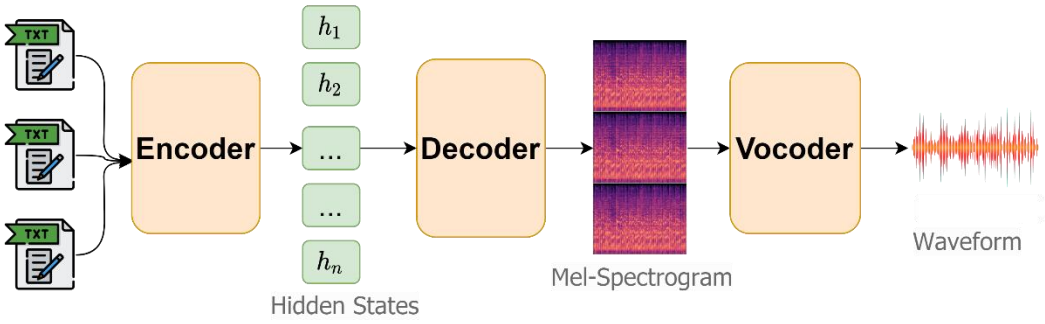
Sau đây là phần trình bày chi tiết từng phần của đồ án.

Chương 2 Cơ sở lý thuyết

Chương 2 này giới thiệu về kiến trúc chung và cách thức hoạt động của những mô hình tổng hợp tiếng nói hiện đại End-To-End. Sau đó sẽ là phần trình bày chi tiết về âm học End-To-End FastSpeech2 [5] và mô hình sinh tiếng nói End-To-End Hifi-Gan [15]. Phần cuối sẽ trình bày chi tiết về thuật toán Viterbi và cách ứng dụng nó vào các bài toán xử lý ngôn ngữ tự nhiên nói chung và bài toán caching cho những ứng dụng tổng hợp tiếng nói lặp theo mẫu nói riêng.

2.1 Tổng hợp tiếng nói End-To-End

Những mô hình tổng hợp tiếng nói dựa trên thống kê như HMM, DNN đã đánh dấu được những bước phát triển vượt trội trong lĩnh vực trong tổng hợp tiếng nói. Những mô hình này đã khắc phục được những nhược điểm của mô hình ghép nối trước đó và đã tạo ra được tiếng nói có chất lượng cao và biểu cảm hơn trước. Tuy nhiên những hệ thống như này thì khá phức tạp để có thể thiết kế dữ liệu cho quá trình huấn luyện và yêu cầu người dùng cần có hiểu biết sâu về mặt ngôn ngữ, điều này làm cho khó tiếp cận với nhiều người. Một vài năm trở lại đây với sự phát triển vượt bậc về những mô hình mạng học sâu đã có nhiều mô hình mới được phát triển, và điển hình là những mô hình End-To-End. Những mô hình này không chỉ dễ dàng huấn luyện, người sử dụng không cần có hiểu biết sâu về mặt ngôn ngữ học mà còn cho ra được tiếng nói có chất lượng gần ngang bằng với con người. Kiến trúc tổng quan về những mô hình End-To-End được mô tả ở *Hình 2.1*.



Hình 2.1 Kiến trúc tổng quan của mô hình tổng hợp tiếng nói End-To-End

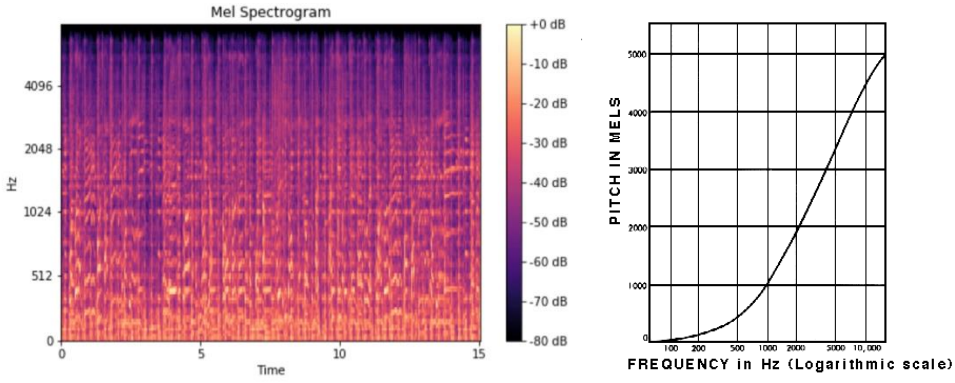
Các thành chính của mô hình tổng hợp tiếng nói End-To-End bao gồm: (i) Bộ mã hóa - Encoder, (ii) Bộ giải mã (Decoder) và (iii) Mô hình âm học - Vocoder.

Encoder: dữ liệu text ban đầu sẽ được chuyển thành dạng vector mang các giá trị số nguyên biểu diễn cho từng ký tự. Sau đó các véc-tơ sẽ được chuyển thành những giá trị trạng thái ẩn (hidden states). Những trạng thái này biểu diễn những tin về mặt ngôn ngữ học cho dữ liệu text đầu vào.

Decoder: những hidden states sẽ được giải mã (decode) thành dạng dữ liệu trung gian giữa văn bản và âm thanh gọi là mel-spectrogram (*Hình 2.2*) theo từng frame. Dạng dữ liệu mel-spectrogram này sẽ chứa những thông tin về mặt âm học để có thể tái tạo lại giọng nói.

Vocoder: là một mô hình sinh tiếng nói, mel-spectrogram đầu vào sẽ được chuyển thành các tín hiệu sóng âm thanh (waveform).

Với những mô hình theo kiến trúc End-To-End thì việc chuyển đổi giữa văn bản sang dạng giọng nói sẽ qua một dạng dữ liệu trung gian là mel-spectrogram [24]. Mel-spectrogram là dạng quang phổ âm thanh (spectrogram) dùng để biểu diễn cho tần số của tín hiệu và được tính toán dưới hệ đơn vị được gọi là mel-scale [24].



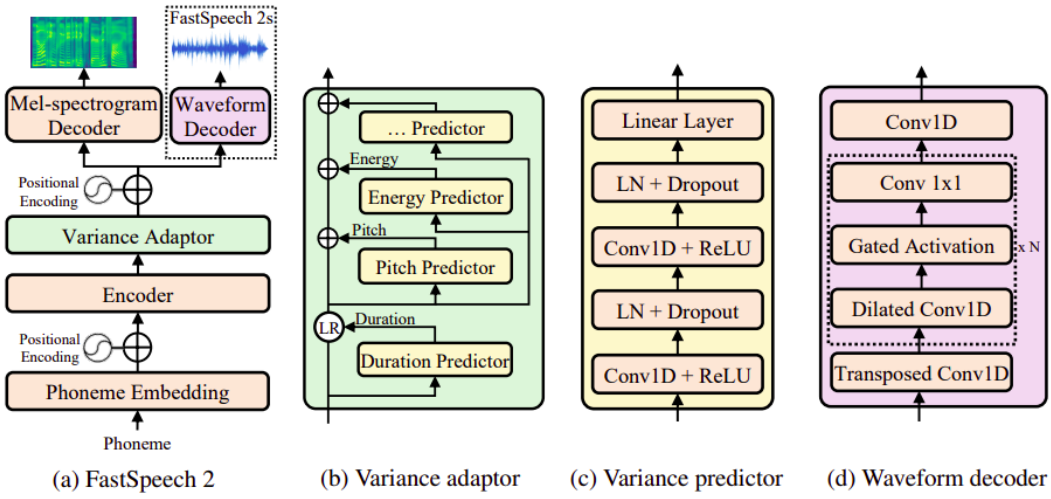
Hình 2.2 Mel-Spectrogram và Mel-Scale

Một vài mô hình tổng hợp tiếng nói End-To-End nổi tiếng trong những năm gần đây như: Tacotron [2] (Google-2017), TransformerTTS [3] (Microsoft-2019), FastSpeech [5] (Microsoft-2021), …

2.2 Mô hình âm học FastSpeech2

Xu hướng nghiên cứu về tổng hợp tiếng nói gần đây tập trung vào các mô hình âm học End-To-End nhằm mục đích đơn giản hóa quá trình huấn luyện và cải thiện chất lượng tiếng nói

tổng hợp. Tuy nhiên với một vài mô hình hồi quy (Autoregressive) [8] như Tacotron [2], TransformerTTS [3], ... thì những vấn đề mà các mô hình này còn gặp phải là lặp từ (word repeating), mất từ (word skipping) và tốc độ tổng hợp còn rất chậm (slow inference speed) [3]. Chính vì vậy đã có những mô hình End-To-End mới hơn được đề xuất để khắc phục những nhược điểm của mô hình Autoregressive, những mô hình này thường được gọi chung là phi hồi quy (Non Autoregressive) [8]. Một trong những mô hình Non Autoregressive nổi tiếng và được áp dụng rộng rãi là FastSpeech2 [5] được Microsoft phát triển trong năm 2021, đây được xem là một trong những mô hình mới và cho kết quả tốt nhất của tổng hợp tiếng nói tính tới thời điểm thực hiện đồ án này. Kiến trúc tổng quan của mô hình này được trình bày trong *Hình 2.3*.



Hình 2.3 Kiến trúc tổng quan của mô hình FastSpeech2 [5]

Các thành phần chính của FastSpeech2 bao gồm: (i) Phoneme Embedding, (ii) Encoder, (iii) Variance Adaptor và (iv) Mel-spectrogram Decoder. Một thành phần khác xuất hiện ở trong kiến trúc tổng quan ở *Hình 2.3* là Waveform Decoder, tuy nhiên thành phần này được sử dụng ở một mô hình nâng cao của FastSpeech2 gọi là FastSpeech 2s và không được sử dụng ở trong đồ án này.

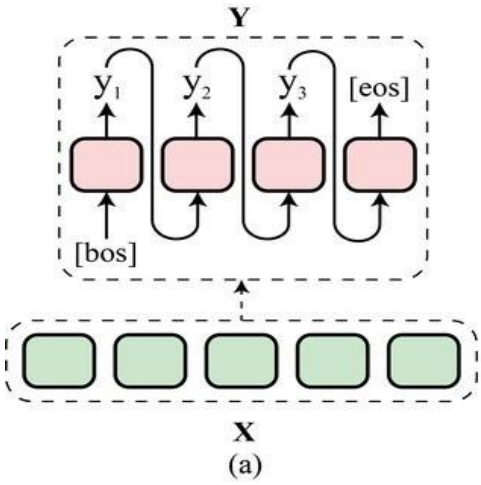
2.2.1 Phoneme Embedding

Đây là thành phần giúp biến đổi dữ liệu dạng văn bản đầu vào thành dạng dữ liệu véc tơ mang giá trị thực phục vụ cho quá trình huấn luyện mạng học sâu. Thành phần này sử dụng kỹ thuật Word Embedding [9] nhưng áp dụng cho mức âm vị (phoneme). Word Embedding là một lớp các kỹ thuật giúp biểu diễn mỗi từ (word) dưới dạng vector số thực trong một không gian vector được định nghĩa trước. Mỗi một từ sẽ được ánh xạ thành một vector và

những giá trị của vector này sẽ được học trong quá trình huấn luyện các mạng học sâu, những từ có cùng ý nghĩa sẽ được biểu diễn các giá trị vector tương đồng nhau. Ưu điểm của phương pháp này là số chiều biểu diễn của vector (thường 10-100) ngắn hơn nhiều so với phương pháp truyền thống là One-Hot Encoding [10] (1.000-1.000.000). Vì vậy sẽ giúp cho khối lượng tính toán giảm đi và quá trình huấn luyện thực hiện nhanh hơn. Phía sau của Phoneme Embedding sẽ có một thành phần được gọi là Positional Encoding, thành phần này giúp bổ sung thông tin về vị trí của các âm vị trong dữ liệu văn bản đầu vào cho những véc tơ nhúng (embedding vector).

2.2.2 Encoder

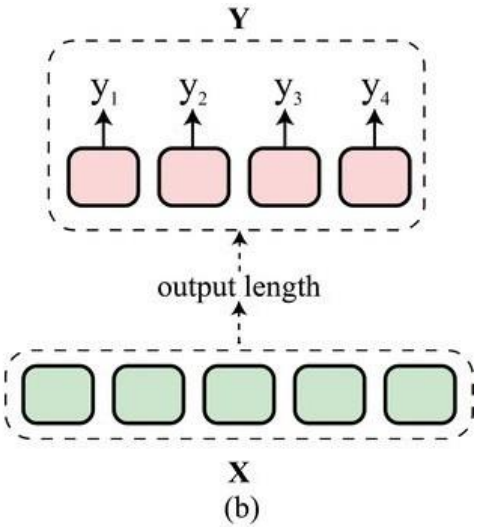
Thành phần này mã hóa các véc tơ nhúng từ bước 2.2.1 thành một dãy các trạng thái ẩn - hidden states [11] như trong các bài toán chuyển đổi từ dãy sang dãy (Seq2Seq - Sequence To Sequence [11]). Tuy nhiên với các mô hình Seq2Seq truyền thống thì kiến trúc thường bao gồm các mạng cổ điển như LSTM, RNN [12], ưu điểm của các mô hình này là tính trực quan và dễ dàng cài đặt, còn nhược điểm lớn là tại mỗi thời điểm chỉ xử lý được một trạng thái dựa vào đầu vào hiện tại và trạng thái đã được xử lý trước đó. Vì vậy những mô hình Seq2Seq dựa trên LSTM sẽ làm cho quá trình xử lý bị chậm. Mô hình Seq2Seq dựa trên LSTM được mô tả như ở *Hình 2.4*.



Hình 2.4 Mô hình Seq2Seq dựa trên mạng LSTM [13]

Đối với FastSpeech2 phần Encoder thay thế việc sử dụng những mô hình LSTM, RNN bằng một kiến trúc được gọi là Transformer [14]. Với kiến trúc này làm cho qúa trình xử lý không còn tuần tự mà được thực hiện một cách song song. Điều này đã làm cho tốc độ mã hóa các

véc tơ nhúng thành chuỗi các trạng thái ẩn diễn ra nhanh hơn rất nhiều so với LSTM hoặc RNN. Mô tả về mô hình Seq2Seq dựa trên kiến trúc Transformer được mô tả ở *Hình 2.5*.

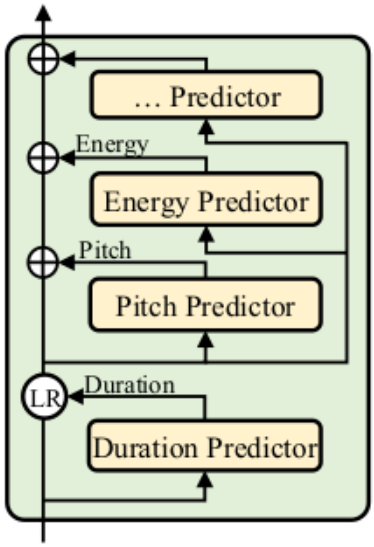


Hình 2.5 Mô hình Seq2Seq dựa trên mạng Transformer [13]

Sau bước 2.2.2 chúng ta sẽ có một dãy các trạng thái ẩn được mã hóa từ dữ liệu văn bản đầu dưới dạng véc tơ giá trị thực và mang đầy đủ thông tin về mặt ngữ cảnh. Những véc tơ đó còn được biết đến với cái tên là véc tơ ngữ cảnh.

2.2.3 Variance Adaptor

Như được mô tả ở trong mục 2.2.2, dữ liệu văn bản đầu vào sẽ được mã hóa và xử lý song song. Ở trong quá trình huấn luyện (training) thì việc này diễn ra bình thường nhờ những thông tin về mặt âm học như độ cao (pitch), trường độ (duration), năng lượng (energy) được trích xuất từ dữ liệu âm thanh tương ứng được cung cấp. Tuy nhiên trong quá trình suy dẫn (inference) thì chỉ có dữ liệu dạng văn bản, và những thông tin về mặt âm học thì mô hình cần phải dự đoán. Với những mô hình dựa trên Autoregressive thì việc dự đoán này sẽ được dựa vào trạng thái đã được dự đoán phía trước. Tuy nhiên với mô hình Non Autoregressive thì việc dự đoán sẽ được thực hiện song song trên nhiều trạng thái, vì vậy việc dự đoán cho trạng thái hiện tại có thể sẽ không có được thông tin của trạng thái phía trước. Vì vậy FastSpeech2 cung cấp thêm một thành phần phụ trợ, độc lập được gọi là Variance Adaptor nhằm mục đích dự đoán những thông tin về mặt âm học để có thể cung cấp thêm thông tin cho quá trình suy dẫn. Kiến trúc tổng quan của Variance Adaptor được mô tả ở *Hình 2.6*.



Hình 2.6 Kiến trúc của Variance Adaptor trong FastSpeech2 [5]

Kiến trúc của Variance Adaptor bao gồm 3 thành phần chính: (i) Thành phần dự đoán trường độ (Duration Predictor), (ii) Thành phần dự đoán độ cao (Pitch Predictor) và (iii) Thành phần dự đoán năng lượng (Energy Predictor). Các thành phần này có kiến trúc mạng nơ ron giống nhau và gọi chung là Variance Predictor (*Hình 2.3*). Tuy nhiên mỗi thành phần đầu vào và đầu ra khác nhau phụ thuộc vào thông tin mà thành phần đó dự đoán. Trong quá trình huấn luyện FastSpeech2 thì những thông tin về mặt âm học sẽ được trích xuất từ dữ liệu gốc cung cấp, những thông tin này cũng được đưa vào các Variance Predictor để huấn luyện song song. Ở quá trình suy dẫn thì những Variance Predictor này sẽ dự đoán thông tin âm học cho dữ liệu văn bản đầu vào để hỗ trợ cho FastSpeech2 dự đoán mel-spectrogram.

2.2.4 Mel-spectrogram Decoder

Thành phần này nhằm muc đích giải mã một chuỗi những trạng thái ẩn đã được bổ sung thông tin âm học ở bước 2.2.3 thành dạng mel-spectrogram. Kiến trúc của thành phần Decoder này cũng tương tự như của Encoder trong phần 2.2.2, dựa trên mô hình Transformer để giải quyết bài toán Seq2Seq. Và nhờ điều này Decoder có thể sử dụng tối ưu được khả năng tính toán song song của phần cứng máy tính và từ đó làm giảm thời gian giải mã đi nhiều lần so với mô hình Seq2Seq dựa trên LSTM hoặc RNN.

Qua những phần mô tả chi tiết về các thành phần chính trong mô hình FastSpeech2 ở trên, chúng ta có thể thấy rằng nhờ việc sử dụng cơ chế mã hóa và giải mã cho bài toán Seq2Seq dựa trên mô hình Transformer đã làm cho quá trình xử lý được thực hiện song song từ đó làm giảm thời gian xử lý đi rất nhiều so với những mô hình Autoregressive như Tacotron

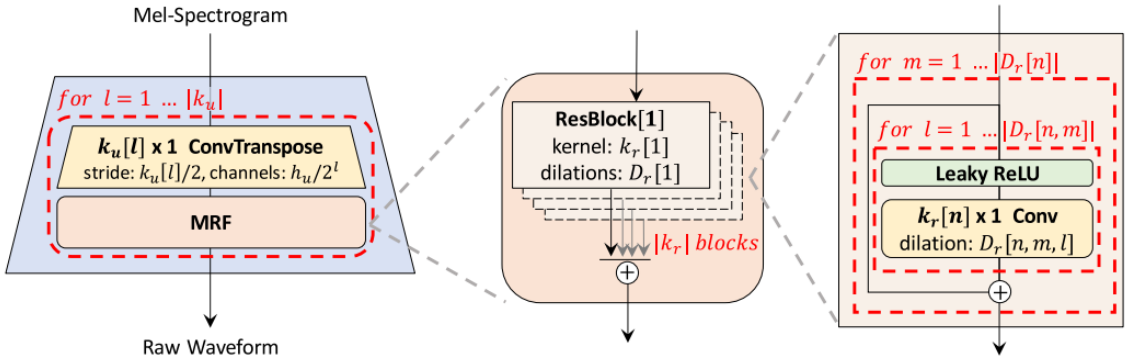
[2], Deep Voice 3 [16], ... Thêm vào đó tiếng nói tổng hợp có sự ổn định và sự tự nhiên tiệm cận tới tiếng nói con người.

2.3 Mô hình sinh tiếng nói Hifi-Gan

Sau khi giải mã được chuỗi mel-spectrogram thì để có thể chuyển đổi sang dạng âm thanh thì chúng ta cần đưa mel-spectrogram qua một mô hình sinh tiếng nói gọi là Vocoder (*Hình 2.1*). Vocoder có thể cài đặt bằng thuật toán như Griffin[26] hoặc huấn luyện thông qua những mô hình mạng. Đồ án sử dụng Vocoder dựa trên mô hình mạng học sâu gọi là Hifi-GAN [15] và huấn luyện cho dữ liệu tiếng nói tiếng Việt. Mô hình Hifi-Gan bao gồm hai thành phần chính: (i) Bộ sinh – Generator và (ii) Bộ phân tách – Discriminator.

2.3.1 Bộ sinh – Generator

Generator được sử dụng trong quá trình suy dẫn nhằm biến đổi quang phổ âm thanh (mel-spectrogram) về dạng sóng âm (waveform).

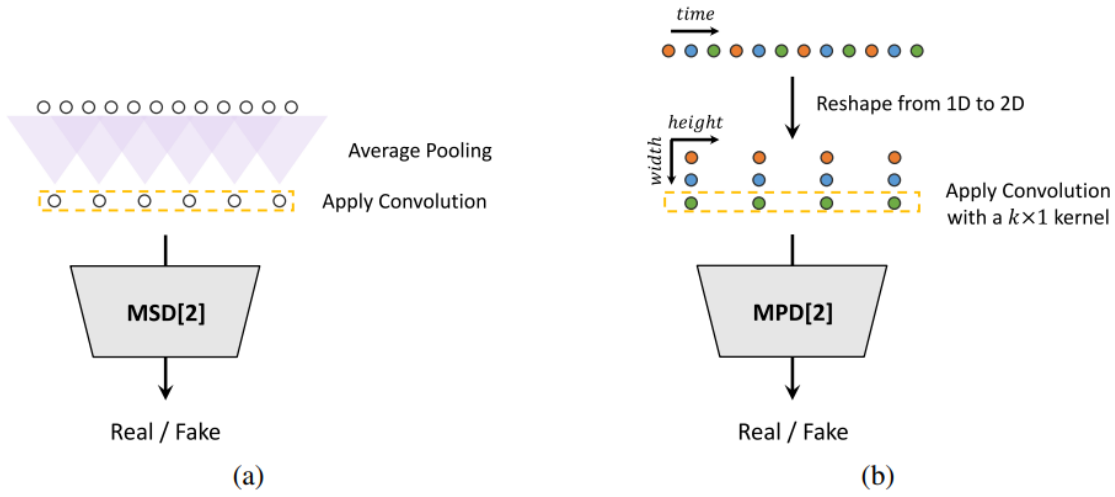


Hình 2.7 Kiến trúc Generator của mô hình sinh tiếng nói Hifi-Gan [15]

Generator trong mô hình sinh tiếng nói Hifi-Gan sử dụng mô hình mạng CNN [27] (convolutional neural network). Đầu vào của mô hình sử dụng mel-spectrogram được mở rộng (upsample) thông qua mô hình ConvTranspose [27] (transposed convolution) cho đến khi độ dài của chuỗi đầu ra bằng với độ phân giải về mặt thời gian của dạng sóng âm (waveform). Mỗi ConvTranspose sẽ được theo sau bởi một thành phần gọi là kết hợp đa cảm nhận – MRF (multi-receptive field fusion), thành phần này giúp quan sát các mẫu dữ liệu với độ dài khác nhau.

2.3.2 Bộ phân tách – Discriminator

Bộ phân tách – Discriminator giúp nhận diện được sự phụ thuộc dài hạn (long-term dependencies) ở trong tiếng nói. Bởi vì tiếng nói chứa nhiều tính hiệu sóng âm với những chu kỳ khác nhau, vậy nên để có thể mô hình hóa được giọng nói tự nhiên ta cần nhận diện được những chu kỳ này. Mô hình Hifi-Gan sử dụng bộ phân tách đa chu kỳ (multi-period discriminator - MPD) chứa nhiều bộ phân tách con, mỗi bộ phân tách sẽ chịu trách nhiệm xử lý một phần tín hiệu theo chu kỳ, và để có thể nhận diện được sự phụ thuộc dài hạn thì mô hình Hifi-Gan sử dụng thêm các bộ phân tách đa tỉ lệ (multi-scale discriminator - MSD), MSD sẽ đánh giá các mẫu âm thanh ở các cấp độ khác nhau.



Hình 2.8 Kiến trúc Discriminator của mô hình sinh tiếng nói Hifi-Gan [15]

2.4 Thuật toán Viterbi

Thuật toán Viterbi [6] là một thuật toán quy hoạch động với mục tiêu tìm được một chuỗi các trạng thái ẩn sao cho tối ưu hóa một hàm chi phí định nghĩa trước dựa trên một chuỗi trạng thái có thể quan sát được. Chuỗi trạng thái ẩn như vậy được gọi là đường dẫn Viterbi (Viterbi Path [6]). Ví dụ với lĩnh vực nhận dạng tiếng nói thì chuỗi quan sát được sẽ là một chuỗi các tính hiệu âm thanh và mục tiêu của thuật toán Viterbi là tìm ra một chuỗi văn bản có xác suất cao nhất dựa trên chuỗi âm thanh đó. Thuật toán Viterbi được đề xuất lần đầu tiên vào năm 1967 bởi nhà toán học Viterbi và có rất nhiều ứng dụng trong nhiều lĩnh vực, điển hình như: tổng hợp tiếng nói, nhận dạng tiếng nói, …

Công thức toán học của thuật toán Viterbi:

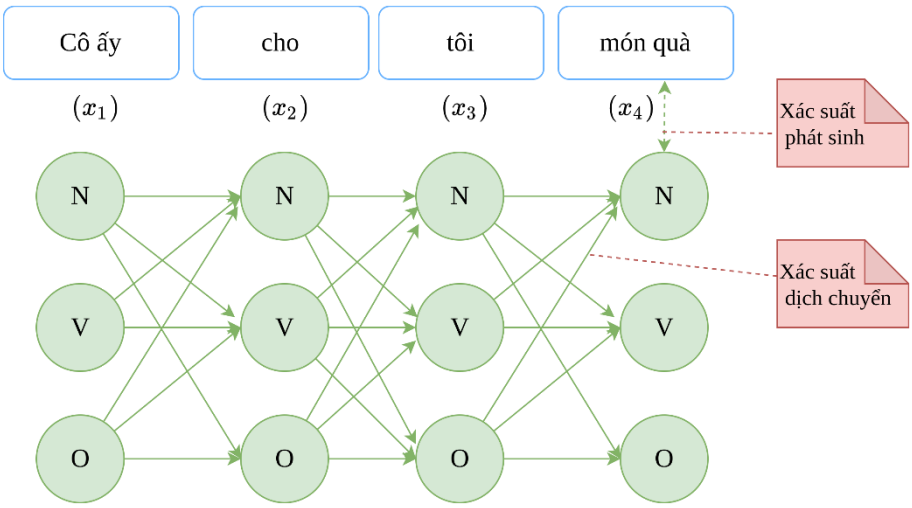
$$f(x_1 \dots x_n) = arg\ max_{(y_1 \dots y_n)} p(x_1 \dots x_n, y_1 \dots y_n)$$

$$p(x_1 \dots x_n, y_1 \dots y_n) = \prod_{i=1}^{n} q(y_i|y_{i-1}) \prod_{i=1}^{n} e(x_i|y_i)$$

Trong đó:

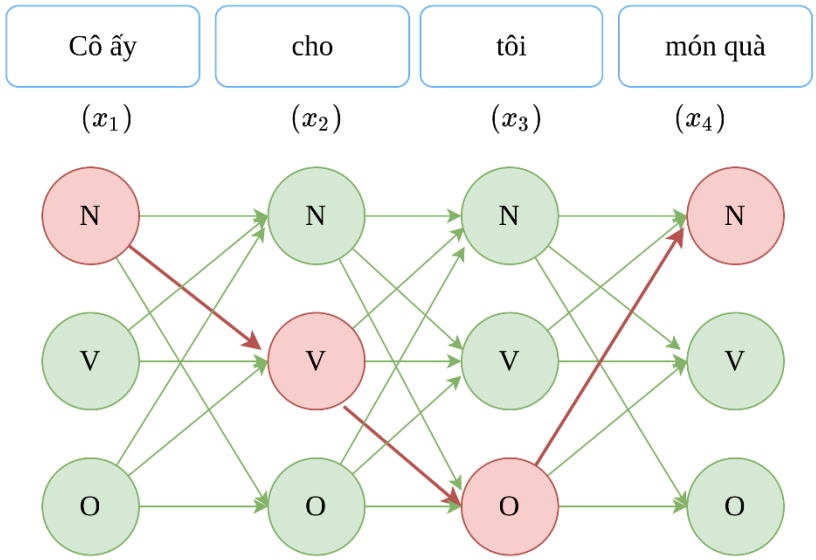
- $(x_1 \dots x_n)$ là chuỗi trạng thái mà chúng ta có thể quan sát được
- $(y_1 \dots y_n)$ là chuỗi trạng thái ẩn mà chúng ta cần tìm
- $q(y_i|y_{i-1})$ là xác suất chuyển dịch từ trạng thái y_{i-1} sang trạng thái y_i
- $e(x_i|y_i)$ là xác suất phát sinh ra x_i khi cho trạng thái ẩn hiện tại là y_i
- $p(x_1 \dots x_n, y_1 \dots y_n)$ là xác suất khi quan sát một chuỗi trạng thái $x_1 \dots x_n$ thì chuỗi trạng thái ẩn tương ứng là $y_1 \dots y_n$
- $f(x_1 \dots x_n)$ là chuỗi trạng thái ẩn có tích xác xuất lớn nhất

Thuật toán Viterbi là thuật toán quy hoạch động, thuật toán sẽ giải quyết các bài toán nhỏ trước, sau đó ghi nhớ kết quả các bài toán nhỏ rồi tiếp tục giải quyết các bài toán lớn hơn. Chính vì vậy tại mỗi bước thì thuật toán sẽ cho giải pháp tốt nhất cho hàm chi phí tổng thể của bài toán. Để có sự hình dung rõ hơn về thuật toán Viterbi, đồ án sẽ trình bày một ví dụ về ứng dụng thuật toán Viterbi trong bài toán gán nhãn từ loại (POS Tagging - Part Of Speech Tagging). Yêu cầu của bài toán này là cho trước một câu và yêu cầu cho biết mỗi từ trong câu thuộc từ loại nào, để đơn giản cho viêc mô tả thuật toán đồ án sẽ giới hạn các từ loại có thể như là: *N (noun - danh từ), V (verb - động từ) và O (Object - tân ngữ)*. Chúng ta có thể mô tả bài toán theo thuật toán Viterbi như sau: cho trước một câu bao gồm chuỗi các từ là các trạng thái mà ta có thể quan sát được$(x_1 \dots x_n)$, hãy tìm chuỗi trạng thái ẩn $(y_1 \dots y_n)$ là chuỗi từ loại tương ứng cho câu đầu vào. Khi đó $q(y_i|y_{i-1})$ là xác suất chuyển dịch từ từ loại y_{i-1} sang y_i và $e(x_i|y_i)$ là xác suất từ đó là x_i khi biết từ loại của nó là y_i. Ta có thể mô tả bài toán qua*Hình 2.9*, với câu đầu vào là "cô ấy cho tôi món quà":



Hình 2.9 Mô tả bài toán POS Tagging

Mỗi từ có thể thuộc một từ loại bất kỳ trong 3 từ loại đã cho. Chúng ta có thể tìm lời giải bằng cách xét hết các trường hợp có thể sau đó tính xác suất cho từng trường hợp, cuối cùng chọn ra chuỗi có xác suất cao nhất. Vậy nên nếu chúng ta vét cạn các trường hợp thì chúng ta sẽ có: $3^4 = 81$ chuỗi từ loại có thể. Vậy nếu độ dài câu và số từ loại tăng lên thì số trường hợp sẽ tăng theo hàm mũ, vì vậy việc vét cạn là không khả thi. Với thuật toán Viterbi, thuật toán sẽ giải quyết bài toán với chuỗi từ loại có độ dài bằng 1 trước, sau đó ghi nhớ giải pháp tốt nhất rồi tiếp tục giải quyết bài toán chuỗi có độ dài 2 và tiếp tục đến chuỗi có độ dài n. Độ phức tạp của thuật toán vét cạn sẽ là $O(m^n)$, còn thuật toán Viterbi là $O(m * n^2)$ (với m là số từ, còn n là số từ loại), vì vậy có thể áp dụng được trong thực tế. Với ví dụ ở trên thì chuỗi từ loại đúng sẽ là: N-V-O-N và với thuật toán Viterbi đây được gọi là đường dẫn Viterbi (Viterbi Path [6]).



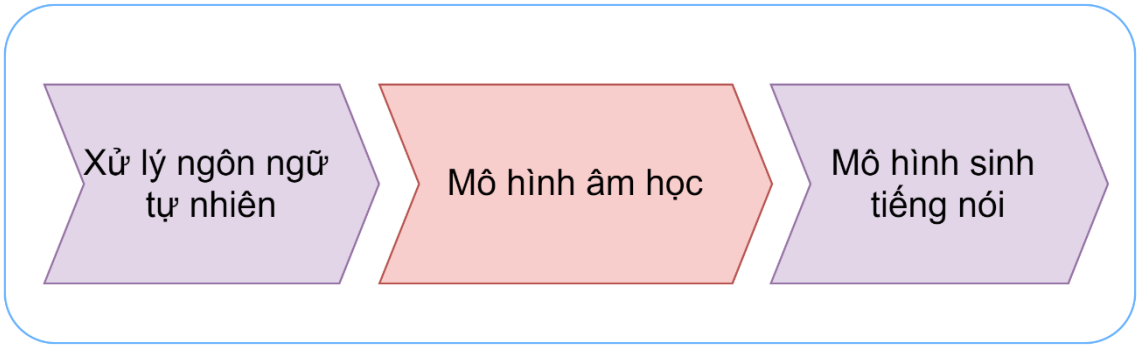
Hình 2.10 Ví dụ về đường dẫn viterbi (Viterbi Path)

Trong chương 2 này đồ án đã trình bày về sự phát triển và xu hướng của tổng hợp tổng nói hiện nay, xu hướng nghiên cứu hiện tại tập trung vào xây dựng những mô hình End2End nhằm giúp cho tiếng nói nhân tạo ngày càng có chất lượng gần với giọng nói con người và dễ dàng tiếp cận hơn. Sau đó đồ án có trình bày một trong những mô hình TTS End2End nổi tiếng gần đây là FastSpeech2 [5], mô hình này đã có nhiều ưu điểm như: tiếng nói tổng hợp tự nhiên, ít gặp lỗi và tốc độ tổng hợp nhanh. Cuối cùng là phần giới thiệu về thuật toán Viterbi và ứng dụng của nó trong nhiều lĩnh vực của xử lý ngôn ngữ tự nhiên. Trong chương 3 đồ án sẽ đồ án sẽ trình bày về kiến trúc đề xuất cho hệ thống tổng hợp tiếng nói Tiếng Việt chất lượng cao sử dụng mô hình FastSpeech2.

Chương 3 Tổng hợp tiếng nói chất lượng cao cho tiếng Việt

Chương 3 này đồ án sẽ trình bày về kiến trúc tổng quan của một hệ thống tổng hợp tiếng nói chất lượng cao cho tiếng Việt. Hệ thống này bao gồm các thành phần (i) Thành phần xử lý ngôn ngữ tự nhiên, (ii) Mô hình âm học và (iii) Mô hình sinh tiếng nói. Sau đó đồ án sẽ đi vào trình bày chi tiết từng thành phần này và cuối cùng là phần đánh giá kết quả đạt được.

3.1 Tổng hợp tiếng nói tiếng Việt chất lượng cao



Hình 3.1 Kiến trúc tổng quan của hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao

Một hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao sẽ bao gồm ba thành phần chính (i) Thành phần xử lý ngôn ngữ tự nhiên, (ii) Mô hình âm học và (iii) Mô hình sinh tiếng nói.

Thành phần xử lý ngôn ngữ tự nhiên

Đây là thành phần tiếp nhận văn bản đầu vào, sau đó tiến hành tiền xử lý và chuẩn hóa dữ liệu cho các bước phía sau, một vài bước tiền xử lý như: mở rộng từ viết tắt, bỏ ký tự đặc biệt, sửa lỗi chính tả, ... Ngoài ra thành phần này giúp cho việc phân tích những đặc trưng về mặt ngôn ngữ học của văn bản đầu vào như thông tin về vị trí âm vị, âm vực (tone), loại từ, …

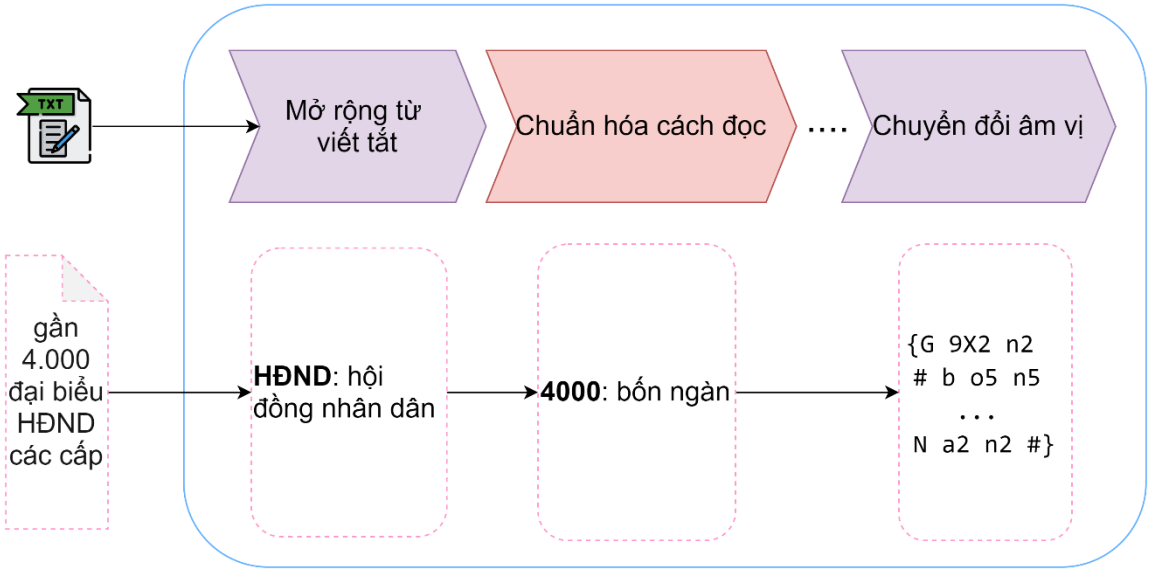
Mô hình âm học

Đây là thành phần giúp mô hình hóa giọng nói dưới dạng các tham số âm học đại diện cho những đặc trưng giọng nói của con người và có thể tái tạo lại giọng nói từ những giá trị âm học đó. Những đặc trưng âm học có thể kể đến như: cao độ (pitch), trường độ (duration), năng lượng (energy). Mô hình này có thể được huấn luyện để dự đoán những thông tin âm học này từ văn bản đã được tiền xử lý ở bước xử lý ngôn ngữ tự nhiên. Những đặc trưng này sau đó sẽ được kết hợp để chuyển đổi sang một dạng dữ liệu trung gian giữa văn bản và âm thanh được gọi là mel-spectrogram, một dạng dữ liệu biểu diễn các tính hiệu âm thanh dưới dạng tần số.

Mô hình sinh tiếng nói

Dữ liệu trung gian mel-spectrogram được dự đoán ở mô hình âm học sẽ được đưa vào mô hình sinh tiếng nói được gọi là Vocoder. Mô hình này sẽ chuyển đổi mel-spectrogram chứa những thông tin âm học dự đoán sang dạng tín hiệu âm thanh (waveform). Mô hình này có thể cài đặt bởi thuật toán như thuật toán Griffin-Lim [26] hoặc huấn luyện bằng các mô hình mạng học sâu như Hifi-GAN [15].

3.2 Xử lý ngôn ngữ tự nhiên



Hình 3.2 Thành phần xử lý ngôn ngữ tự nhiên

Thành phần xử lý ngôn ngữ tự nhiên có trách nhiệm tiền xử lý văn bản đầu vào như mở rộng từ viết tắt, xóa bỏ ký tự đặc biệt, chuẩn hóa cách đọc, … và phân tích những thông tin về mặt ngôn ngữ học như vị trí âm vị, từ loại, …Việc tiền xử lý dữ liệu sẽ giúp cho quá trình huấn luyện và suy dẫn được thực hiện dễ dàng và chính xác hơn. Trong đồ án này thành phần xử lý ngôn ngữ tự nhiên được sử dụng trên dựa một hệ thống[5] đã xây dựng sẵn dành cho tiếng Việt bởi TS. Nguyễn Thị Thu Trang.

a) Tiền xử lý văn bản

Dữ liệu văn bản đầu vào do người dùng đưa vào thường chứa những ký tự đặc biệt như: @, $, ... hoặc chứa những từ viết tắt như: BKHN (bách khoa hà nội), HĐND (hội đồng nhân dân), ... Vì vậy chúng ta cần phải có những bước tiền xử lý: loại bỏ các ký tự đặc biệt, mở rộng từ viết tắt, ... để chuẩn dữ liệu đầu vào. Một vài ví dụ về các thao tác trong tiền xử lý văn bản được mô tả ở *Bảng 3.1*.

Thao tác	**Văn bản đầu vào**	**Văn bản đầu ra**
Tách từ	tôi yêu mọi người	[tôi, yêu, mọi người]
Mở rộng từ viết tắt	HĐND	hội đồng nhân dân

Bảng 3.1 Ví dụ về các thao tác trong tiền xử lý văn bản

b) Chuyển đổi văn bản sang âm vị

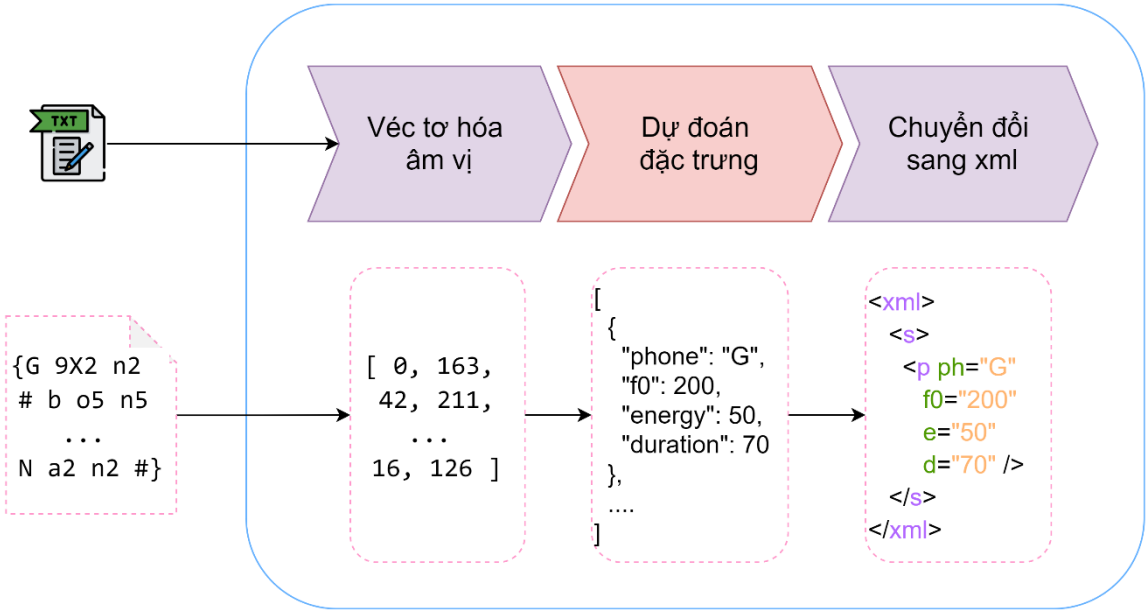
Với những mô hình tổng hợp tiếng nói End2End cho phép chúng ta tổng hợp tiếng nói ở mức ký tự, ví dụ như: a, ă, b, c, ... Tuy nhiên với tiếng Việt sẽ có những ký tự sẽ cho những cách phát âm khác nhau ở các từ khác nhau, ví dụ ký tự *"c"* ở từ *"cáu"* với từ *"cháu"* sẽ có cách phát âm khác nhau. Vì vậy nếu dữ liệu huấn luyện không đủ đa dạng thì khi tổng hợp sẽ gặp phải trường hợp nhập nhằng. Vì vậy cần phải có một cách biểu diễn khác cho văn bản đầu vào sao cho cách phát âm của những ký tự có sự phân biệt ở từng từ. Để làm được điều đó thì đồ án đã chuyển văn bản đầu vào sang dạng âm vị dựa trên hệ thống[5]. Âm vị là phiên âm cách đọc của mỗi từ và chi tiết hơn mức ký tự, vì vậy khi huấn luyện thì mô hình sẽ không bị nhầm về cách đọc. Một vài ví dụ về chuyển đổi từ sang âm vị được mô tả trong *Bảng 3.2* (dấu "-" là phân cách giữa 2 âm tiết, dấu "#" dùng để phân tách giữa các từ).

[5] TS. Nguyễn Thị Thu Trang, Hệ thống tổng hợp tiếng nói tiếng Việt HMM, https://lab-tts.vbeecore.com/

Mức từ	Mức âm vị
cáu	k aX5 w5
cháu	ts aX5 w5
công thức máu	k o1 Nm1 - th M6 k6 # m aX5 w5

Bảng 3.2 Ví dụ biến đổi từ mức từ sang mức âm vị

3.3 Mô hình âm học



Hình 3.3 Mô hình âm học

Để có thể dự đoán được những đặc trưng âm học từ văn bản đầu vào thì chúng ta cần phải đi qua nhiều bước trung gian như: (a) Tiền xử lý văn bản, (b) Chuyển đổi văn bản sang âm vị, (c) Véc tơ hóa âm vị, (d) Dự đoán đặc trưng và (e) Chuyển đổi sang định dạng xml. Hai bước (a) và (b) đã được trình bày ở phần 3.2.

a) Véc tơ hóa âm vị

Đây là quá trình biến đổi chuỗi âm vị đầu vào thành một véc tơ số nguyên, mỗi âm vị được đại diện bởi một số nguyên duy nhất. Việc chuyển đổi sẽ giúp cho các phép tính toán có thể

thực hiện được trên máy tính và quá trình huấn luyện dễ thực hiện hơn. Ví dụ với chuỗi âm vị: *{k aX5 w5}* sẽ được chuyển thành véc tơ *[27, 158, 63]*.

b) Dự đoán đặc trưng

Để có thể dự đoán đặc trưng âm học cho một văn bản đầu vào thì trước tiên cần huấn luyện mô hình FastSpeech2 [5] cho dữ liệu tiếng Việt. Dữ liệu được sử dụng trong đồ án là hai bộ dữ liệu tiếng Việt tương ứng với hai giọng đọc nữ thuộc hai vùng miền khác nhau. Bộ dữ liệu thứ nhất có tên là NgocHuyen-Vbee [18], được thu âm bởi phát thanh viên nữ miền Bắc. Bộ dữ liệu thứ hai có tên là ThaoTrinh-Vbee [18] được thu âm bởi phát thanh viên miền Nam. Hai bộ dữ liệu này bao gồm các tệp âm thanh chất lượng 48KHz và thu âm cho các lĩnh vực như tổng đài, sách, truyện, tin tức và hội thoại. Tất cả hai bộ dữ liệu trên đều được thu âm ở trong phòng âm chuyên nghiệp và đã được kiểm tra thủ công sự trùng khớp về mặt nội dung giữa văn bản với âm thanh. Thông tin chi tiết về từng bộ dữ liệu được mô tả ở *Bảng 3.3*.

Thông tin	**NgocHuyen-Vbee**	**ThaoTrinh -Vbee**
Tổng số lượng âm thanh	16.412	15.640
Tổng số lượng từ	409.106	380.131
Tổng số lượng từ riêng biệt	6253	6003
Tổng số lượng ký tự	1.706.009	1.583.994
Tổng thời gian	35.5 (giờ)	31.3 (giờ)
Thời gian trung bình trên mỗi đoạn tiếng nói	7.57 (giây)	7.2 (giây)
Số lượng từ trung bình trên mỗi đoạn tiếng nói	24.9	24.3
Thời gian âm thanh ngắn nhất	1.130 (giây)	0.292 (giây)
Thời gian âm thanh dài nhất	31.67 (giây)	25.63 (giây)

Bảng 3.3 Thông tin chi tiết về hai bộ dữ liệu huấn luyện cho tiếng Việt

Sau quá trình huấn luyện thì chúng ta sẽ có được mô hình có thể dự đoán những đặc trưng âm học từ văn bản đầu vào và có thể tổng hợp âm thanh từ đặc trưng âm học dự đoán. Những đặc trưng âm học sẽ được dự đoán theo từng âm vị, ví dụ về các giá trị đặc trưng âm học được dự đoán được mô tả ở *Bảng 3.4*.

Tên thuộc tính	**Giá trị**	**Mô tả**
cao độ - pitch	219.12 (Hz)	truyền đạt cảm xúc, sự nhấn mạnh của người nói
năng lượng - energy	40.5 (dB)	thể hiện độ to nhỏ của tiếng nói
trường độ - duration	100 (ms)	thời gian để phát âm một âm vị

Bảng 3.4 Ví dụ về giá trị đặc trưng âm học của tiếng nói

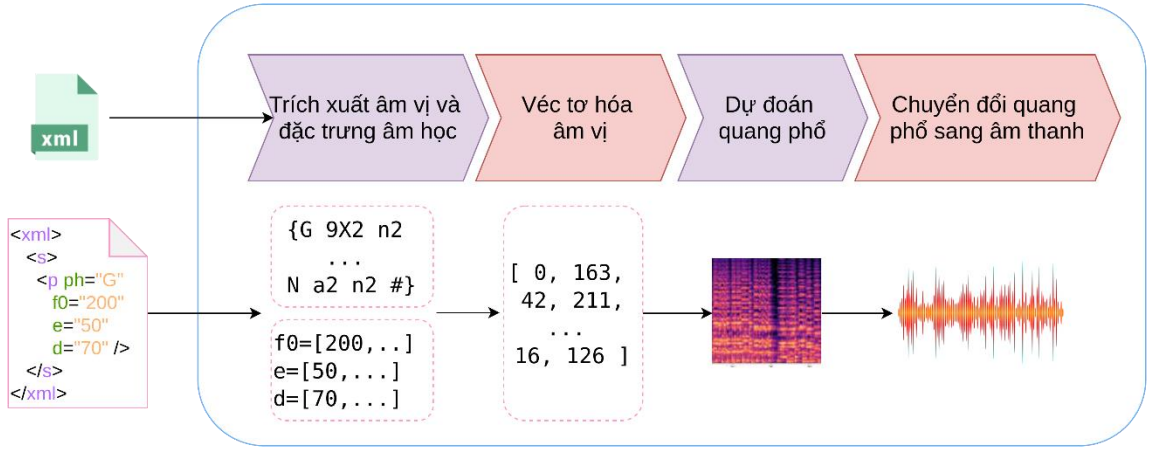
c) Chuyển đổi sang định dạng xml

Để thuận tiện cho việc lưu trữ và trích xuất dữ liệu các âm vị và các giá trị âm học tương ứng được lưu trữ dưới định dạng xml. Việc biến đổi có thể được thực hiện bởi các thư viện có sẵn trong các ngôn ngữ lập trình. Một ví dụ về cách lưu trữ các giá trị âm học dưới định dạng xml được mô tả ở trong *Hình 3.4*.

```
<t g2p_method="lexicon" ph="ts\\ u Nm _5a - t o j _1" pos="P">
   Chúng tôi
   <syllable ph="ts\\ u Nm" tone="_5a">
      <ph d="93" energy="7.102" f0="233.064" p="ts\\" />
      <ph d="46" energy="55.835" f0="245.893" p="u" />
      <ph d="35" energy="45.741" f0="237.897" p="Nm" />
   </syllable>
   <syllable ph="t o j" tone="_1">
      <ph d="46" energy="10.757" f0="236.195" />
      <ph d="46" energy="72.609" f0="292.897" />
      <ph d="58" energy="84.131" f0="284.235" />
   </syllable>
  </t>
```

Hình 3.4 Ví dụ về cách lưu trữ các thông tin âm học

3.4 Mô hình sinh tiếng nói



Hình 3.5 Kiến trúc chi tiết mô hình sinh tiếng nói từ đặc trưng âm học

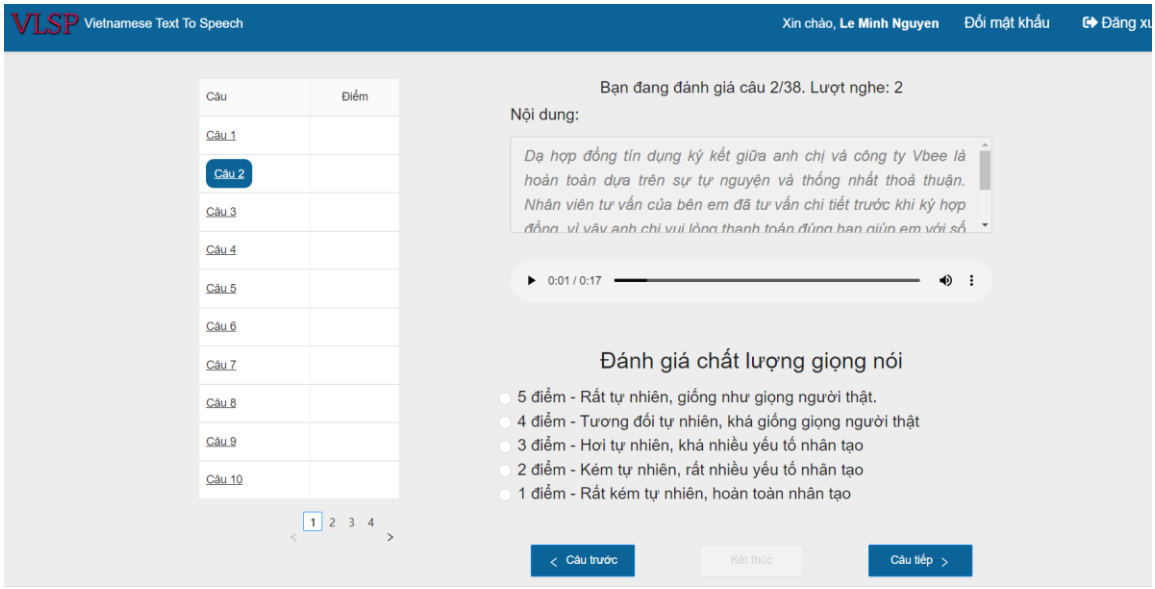
Dữ liệu đầu vào cho thành phần tổng hợp âm thanh sẽ dưới định dạng xml như đã được mô tả ở phần 3.3. Dữ liệu xml đó sẽ bao gồm các âm vị và các đặc trưng tương ứng, vì vậy để có thể tổng hợp thành âm thanh thì trước tiên ta sẽ trích xuất ra chuỗi âm vị và các tham số đặc trưng tương ứng. Sau đó để mô hình FastSpeech2 [5] đã được huấn luyện có thể sử dụng được thông tin âm vị thì đầu tiên chuỗi âm vị sẽ được véc tơ hóa thành dạng số nguyên, phần véc tơ hóa âm vị đã trình bày ở mục (a) phần3.2. Tiếp theo mô hình FastSpeech2 sẽ dự đoán quang phổ âm thanh từ véc tơ âm vị dựa vào các giá trị âm học tương ứng, cuối cùng quang phổ dự đoán sẽ được qua một bộ chuyển đổi Vocoder để biến thành âm thanh để đưa ra âm thanh tổng hợp cuối cùng. Trong mô hình sinh tiếng nói này, đồ án đã sử dụng thành phần biến đổi từ quang phổ sang âm thanh được gọi là Hifi-GAN [15], đã được trình bày chi tiết ở phần 2.3.

3.5 Đánh giá kết quả

3.5.1 Phương pháp đánh giá

Bài kiểm tra MOS (Mean Opinion Score) được lựa chọn để thực hiện việc đánh giá độ tự nhiên của giọng nói tổng hợp so với giọng nói thu âm gốc. Bài kiểm tra sẽ được tiến hành bằng cách trộn lẫn giọng nói tổng hợp với giọng nói gốc và mỗi bài kiểm tra sẽ bao gồm 40 câu (các câu ví dụ được trình bày ở phần phụ lục A.1). Các đối tượng tham gia sẽ lắng nghe từng câu sau đưa ra đánh giá về chất lượng giọng nói trên thang điểm 5, từ thấp đến cao bao gồm: (1) Rất kém tự nhiên, hoàn toàn nhân tạo, (2) Kém tự nhiên, rất nhiều yếu tố nhân tạo,

(3) Hơi tự nhiên, khá nhiều yếu tố nhân tạo, (4) Tương đối tự nhiên, khá giống giọng người thật, và (5) Rất tự nhiên, giống như giọng người thật. *Hình 3.6* mô tả giao diện mà những đối tượng tham gia đánh giá độ tự nhiên sử dụng để đưa ra điểm số cho đoạn tiếng nói mà họ vừa nghe. Kết quả sau cùng được tính bằng cách tính điểm đánh giá trung bình của mỗi câu theo từng giọng.



Hình 3.6 Giao diện công cụ đánh giá điểm MOS cho giọng nói

3.5.2 Kết quả đánh giá

Để đánh giá chất lượng của giọng nói tổng hợp của hệ thống thì đồ án tiến hành một bài kiểm tra MOS với bộ dữ liệu ThaoTrinh-Vbee cho ba loại giọng: (i) giọng thu âm gốc (ground truth), (ii) giọng mô hình End-To-End phi hồi quy (Fastspeech2) và (iii) giọng mô hình dựa trên tham số âm học (DNN). Dữ liệu đánh giá được lựa chọn ngẫu nhiên 40 câu từ tập dữ liệu gốc và không nằm trong tập dữ liệu huấn luyện, các câu có độ dài đoạn âm thanh thu âm gốc từ 8-15 giây. Sau đó các đoạn tiếng nói của các giọng được trộn lẫn và chia ngẫu nhiên tới từng đối tượng tham gia đánh giá chất lượng. Kết quả của bài đánh giá được mô tả ở *Bảng 3.5*.

Tên Giọng	**Điểm MOS**
ThaoTrinhGroundTruth	4.35
ThaoTrinhFastspeech2 (*)	**3.92**
ThaoTrinhDNN	3.06

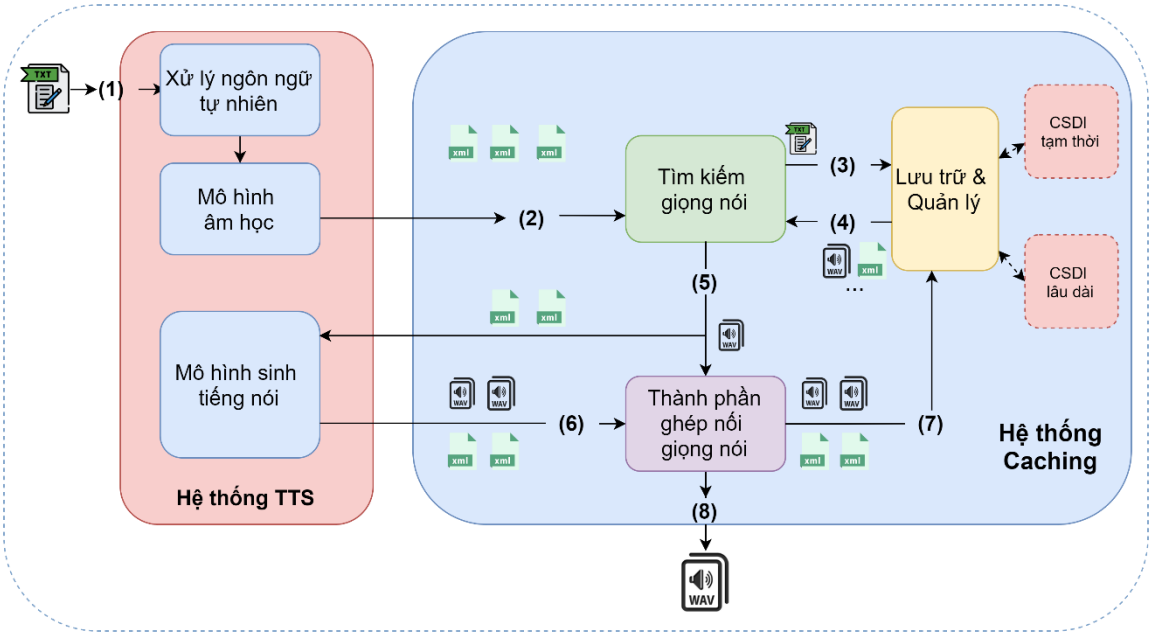
Bảng 3.5 Kết quả đánh giá MOS test cho giọng bộ dữ liệu ThaoTrinh-Vbee

Chúng ta có thể thấy chất lượng của giọng nói tổng hợp chất lượng cao FastSpeech đạt sự tự nhiên tới 3.92/4.35 ≈ 90% so với giọng nói thu âm gốc, trong khi đó hệ thống tổng hợp DNN chỉ cho kết quả 3.06/4.35≈70% so với giọng nói thu âm gốc. Đây là một kết quả khá tốt khi so sánh với những mô hình tổng hợp giọng nói được công bố hiện nay.

Chương 4 Đề xuất mô hình caching cho những ứng dụng tổng hợp tiếng nói lặp lại theo mẫu

Trong chương 4 này đồ án sẽ trình về giải pháp đề xuất và kiến trúc tổng quan của mô hình caching cho những ứng dụng tổng hợp tiếng nói lặp lại nội dung theo mẫu câu. Sau đó là những phần trình bày chi tiết về các thành phần trong mô hình bao gồm: (i) Thành phần tìm kiếm chuỗi tiếng nói (ii) Thành phần quản lý và lưu trữ dữ liệu và (iii) Thành phần ghép nối các đoạn tiếng nói. Cuối cùng là phần đánh giá kết quả đạt được.

4.1 Giải pháp tổng quan



Hình 4.1 Sơ đồ mô tả kiến trúc hệ thống caching kết hợp hệ thống tổng hợp tiếng nói

Như đã trình bày những mô hình hiện đại End-To-End phi hồi quy đã có những ưu điểm nổi trội hơn so với những mô hình trước đó như: ổn định, thời gian tổng hợp nhanh, chất lượng giọng nói gần bằng với con người. Tuy nhiên có nhiều hệ thống ứng dụng tổng hợp tiếng nói yêu cầu phản hồi theo thời gian thực và thường tổng hợp những nội dung lặp lại theo

mẫu câu, một ứng dụng điển hình có thể kể đến đó là tổng đài nhân tạo. Nhận thấy những điều này, trong chương 4 đồ án sẽ đưa ra đề xuất về giải pháp xây dựng một hệ thống caching kết hợp cùng với hệ thống tổng hợp tiếng nói chất lượng cao cho Tiếng Việt đã được trình bày ở Chương 3, với mục đích tái sử dụng được những nội dung đã được tổng hợp, từ đó tối ưu tốc độ tổng hợp và tiết kiệm chi phí. Giải pháp tổng quan được mô tả ở *Hình 4.1*.

Mô tả luồng hoạt động

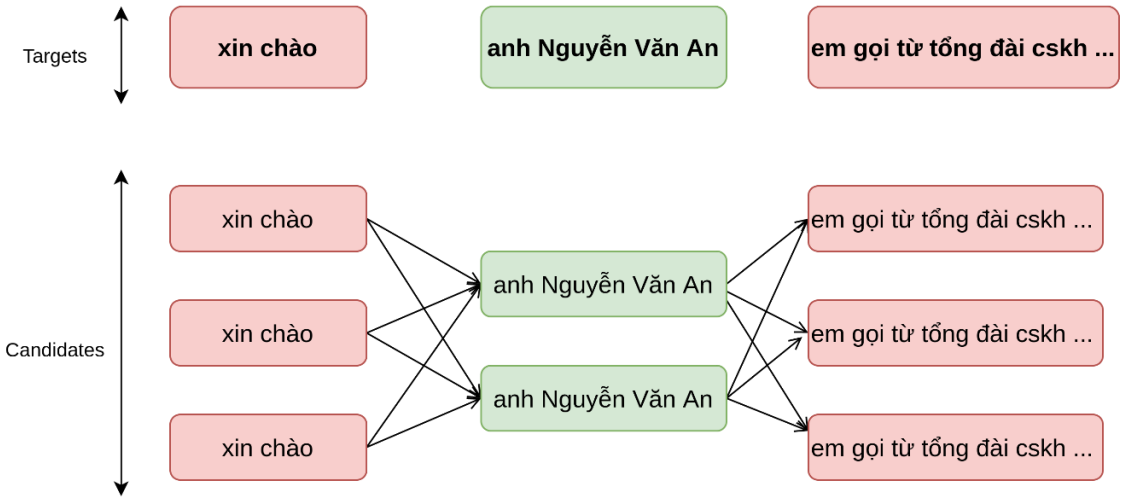
Tại bước (1) văn bản đầu vào sẽ được đưa qua thành phần xử lý ngôn ngữ tự nhiên của hệ thống TTS để tiến hành tiền xử lý và chuẩn hóa dữ liệu. Sau đó dữ liệu chuẩn hóa được đưa sang thành phần dự đoán đặc trưng âm học, ở đây các thông tin âm học dự đoán sẽ được lưu dưới dạng xml và gửi sang bên thành phần tìm kiếm giọng nói ở bước (2). Thành phần tìm kiếm giọng nói sẽ tiếp nhận các văn bản xml, sau đó tiến hành phân tách đặc trưng và nội dung văn bản. Tiếp theo thành phần tìm kiếm âm thanh sẽ truy vấn cơ sở dữ liệu âm thanh tại bước (3) dựa trên nội dung văn bản đã phân tách và nhận kết quả trả về bao gồm các cặp <*tiếng nói, đặc trưng*> của những văn bản có cùng nội dung tại bước (4). Tại đây thành phần tìm kiếm âm thanh sẽ tiến hành phân tích và tính toán để xác định những đoạn tiếng nói nào có thể tái sử dụng được cho văn bản vào, đoạn nào nào cần tổng hợp mới. Tại bước (5) những đoạn tiếng nói không tái sử dụng được thì thành phần tìm kiếm giọng nói sẽ gửi lại các thông tin âm học dưới dạng xml cho mô hình sinh tiếng nói của hệ thống TTS, còn đoạn tiếng nói tái sử dụng được thì sẽ được đưa sang thành phần ghép nối và chờ kết quả tổng hợp trả về từ hệ thống TTS. Hệ thống TTS sẽ trả về đoạn tiếng nói đã được tổng hợp cùng với thông tin âm học tương ứng cho thành phần ghép nối tại bước (6), mục đích của việc trả về cả đoạn tiếng nói và thông tin âm học nhằm mục đích lưu trữ để có thể tái sử dụng cho những văn bản tiếp theo. Thành phần ghép nối sẽ tiến hành làm trơn và ghép nối các đoạn tiếng nói nhận được sau đó trả về giọng nói kết quả tại bước (8), trước đó tại bước (7) thì hệ thống ghép nói sẽ chuyển tiếp các cặp <*tiếng nói, đặc trưng*> sang bên thành phần quản lý lưu trữ để lưu trữ thông tin vào cơ sở dữ liệu. Với ví dụ mô tả ở *Hình 4.1,* chúng ta có thể nhận thấy với văn bản đầu vào chúng ta có thể phân tách thành ba thành phần âm học và sau khi tiến hành tính toán thì chúng ta có thể tái sử dụng lại được một đoạn tiếng nói, tỷ lệ tái sử dụng sẽ là 1/3.

Sau đây là phần trình bày chi tiết từng thành phần trong hệ thống caching - hệ thống màu xanh bên phải của *Hình 4.1*.

4.2 Thành phần tìm kiếm đoạn tiếng nói

Mô hình hóa với thuật toán Viterbi

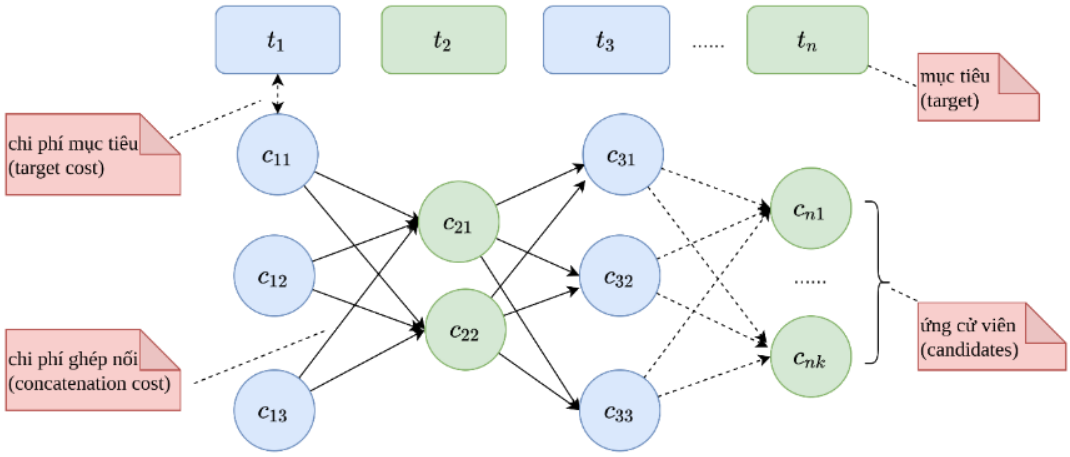
Việc xây dựng thành phần tìm kiếm giọng nói dựa trên thuật toán Viterbi [6] nhằm mục đích tìm kiếm được một chuỗi các đoạn tiếng nói có thể tái sử dụng được sao cho tối ưu một hàm chi phí định nghĩa trước. Để có hình dung dễ hơn về bài toán và cách áp dụng thuật toán, phần này sẽ xét một ví dụ thực tế từ vấn đề của các hệ thống tổng đài nhân tạo. Với các hệ thống tổng đài nhân tạo thì có rất nhiều câu nói với khách hàng thường có cấu trúc cá nhân hóa và chỉ khác ở phần danh xưng hoặc tên người. Ví dụ cấu trúc: *"xin chào @danhxung @ten, em gọi từ tổng đài cskh của công ty ABC", @danhxung* ở đây có thể là: ông, bà, anh, chị, em, ..., *@ten* ở đây có thể là tên người bất kỳ: An, Bình, ... Ta có thể nhận thấy rằng ở những khách hàng khác nhau thì chỉ có danh xưng hoặc tên người là có thể thay đổi, còn các phần còn lại thì nội dung giữ nguyên. Nếu chúng ta tổng hợp trước những đoạn như *"xin chào", "em gọi từ tổng đài cskh của công ty ABC"*, sau đó với mỗi khách hàng mới, chúng ta tổng hợp tên người đó rồi ghép lại với nội dung đã thu âm thì giọng nói đầu ra sẽ rất mất tự nhiên và gây cho người nghe cảm giác khó chịu. Còn nếu như với mỗi khách hàng mà chúng ta đều tổng hợp câu lại từ đầu thì điều đó sẽ làm quá trình tổng hợp bị lặp dẫn tới gây lãng phí về tài nguyên, thời gian.



Hình 4.2 Ví dụ về một mẫu câu lặp lại của hệ thống tổng đài nhân tạo

Chúng ta sẽ mô hình hóa bài toán tổng đài nhân tạo này với thuật toán Viterbi như sau: Giả sử với mỗi câu đầu vào chúng ta sẽ tiến hành tách câu thành các phần $\{t_1, t_2, ..., t_n\}$ dựa vào

cấu trúc cá nhân hóa, mỗi phần đó sẽ được gọi là mục tiêu (target) cần tìm kiếm âm thanh, những đoạn tiếng nói đã được tổng hợp có cùng nội dung sẽ được gọi là ứng cử viên (candidates). Mỗi mục tiêu ti sẽ có thể có nhiều ứng cử viên *[c_{i1}, c_{i2}, ...]*. Sự tương đồng về mặt ngữ cảnh giữa mục tiêu *ti* và mỗi ứng cử viên tương ứng *[c_{i1}, c_{i2}, ...]* được xem như xác suất phát sinh trong thuật toán Viterbi được trình bày ở mục 2.4 và được gọi là hàm chi phí mục tiêu (target cost). Sự tương đồng về mặt ngữ cảnh giữa hai ứng cử viên liên tiếp (c_{i-1}, c_i) được xem như xác suất chuyển dịch trong thuật toán Viterbi và được gọi là hàm chi phí ghép nối (concatenation cost). Hàm chi phí dùng để đánh giá trong quá trình tìm kiếm chuỗi các đoạn tiếng nói sẽ là tổng của 2 hàm chi phí mục tiêu và ghép nối.



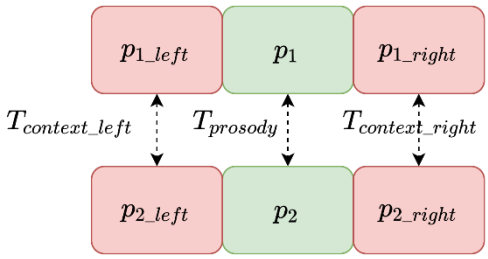
Hình 4.3 Thuật toán tìm kiếm đoạn tiếng nói đề xuất

Sau đây em sẽ trình bày chi tiết về công thức toán học của hai hàm chi phí đánh giá: (i) chi phí mục tiêu và (ii) chi phí ghép nối.

Hàm chi phí mục tiêu (target cost)

Hàm chi phí mục tiêu giữa mục tiêu ti và ứng cử viên ci được tính toán dựa vào sự khác biệt giữa âm vị đầu vào âm vị cuối của mục tiêu c_i với âm vị đầu và âm vị cuối của ứng cử viên c_i. Sự khác biệt giữa hai âm vị (giả sử p_1, p_2) với nhau được xác định bằng cách:

- Tính độ sai khác về mặt ngữ cảnh như: dấu câu, giá trị âm vị, ... giữa âm vị trái và âm vị phải của p_1 với âm vị trái và âm vị phải của p_2, và được ký hiệu là $T_{context}$
- Tính độ sai khác về mặt đặc trưng âm học như: trường độ, cao độ, năng lượng giữa hai âm vị p_1 và p_2, và được ký hiệu là $T_{prosody}$



Hình 4.4 Đánh giá sự khác biệt giữa hai âm vị p_1 và p_2

Hàm chi phí mục tiêu giữa hai âm vị được tính theo công thức sau đây:

$$T(p_i, p_j) = T_{context}(p_{i\_left}, p_{j\_left}) + T_{prosody}(p_i, p_j) + T_{context}(p_{i\_right}, p_{j\_right})$$

Trong đó:

- $T_{context}(p_i, p_j) = \frac{\sum_{m=1}^{k} evaluate_c(feat_{p_i}^{c}[m], feat_{p_j}^{c}[m])}{k}$

  (Với $evaluate_c(x, y) = \begin{cases} 0, & nếu\ x == y \\ 1, & nếu\ x \neq y \end{cases}$ và $feat_{pi}^{c}$ là véc tơ ngữ cảnh của âm vị p_i)

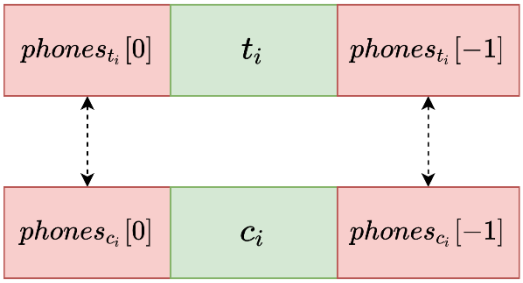
- $T_{prosody}(p_i, p_j) = \frac{\sum_{m=1}^{k} evaluate_c(feat_{p_i}^{p}[m], feat_{p_j}^{p}[m])}{k}$

  (Với $evaluate_p(x, y) = \begin{cases} 0, & nếu\ |x - y| \leq hằng\ số\ C \\ 1, & nếu\ |x - y| > hằng\ số\ C \end{cases}$ và $feat_{pi}^{p}$ là véc tơ âm học của âm vị p_i)

Cuối cùng chi phí mục tiêu giữa mục tiêu t_i và ứng cử viên c_i được xác định bằng:

$$C_t(t_i, c_i) = T(phones_{t_i}[0], phones_{c_i}[0]) + T(phones_{t_i}[-1], phones_{c_i}[-1])$$

(Với 0 là vị trí âm vị đầu tiên và -1 là vị trí âm vị cuối)

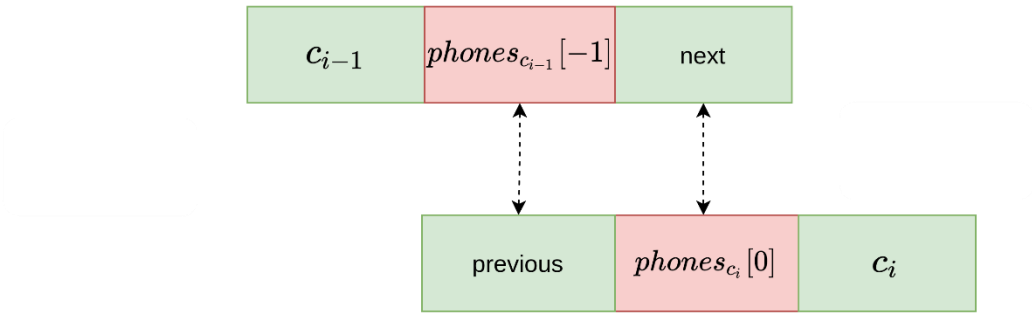


Hình 4.5 Đánh giá chi phí mục tiêu giữa mục tiêu ti và ứng cử viên c_i

Hàm chi phí ghép nối (concatenation cost)

Hàm chi phí ghép nối giữa hai ứng cử viên liên tiếp c_{i-1} và c_i được xác định dựa vào:

- Sự khác biệt giữa âm vị kế tiếp của âm vị cuối của c_{i-1} và âm vị đầu của c_i
- Sự khác biệt giữa âm vị cuối của c_{i-1} và âm vị trước của âm vị đầu của c_i



Hình 4.6 Đánh giá chi phí ghép nối giữa ứng cử viên c_{i-1} và ứng cử viên c_i

Cuối cùng chi phí ghép nối giữa ứng cử viên c_{i-1} và c_i được xác định bằng:

$$C_c(c_{i-1}, c_i) = T\left(phones_{c_{i-1}}[0], previous_{c_i}\right) + T\left(next_{c_{i-1}}, phones_{c_i}[0]\right)$$

(Với $previous_{c_i}$ biểu diễn cho âm vị trước c_i và $next_{c_i}$ biểu diễn cho âm vị sau c_i)

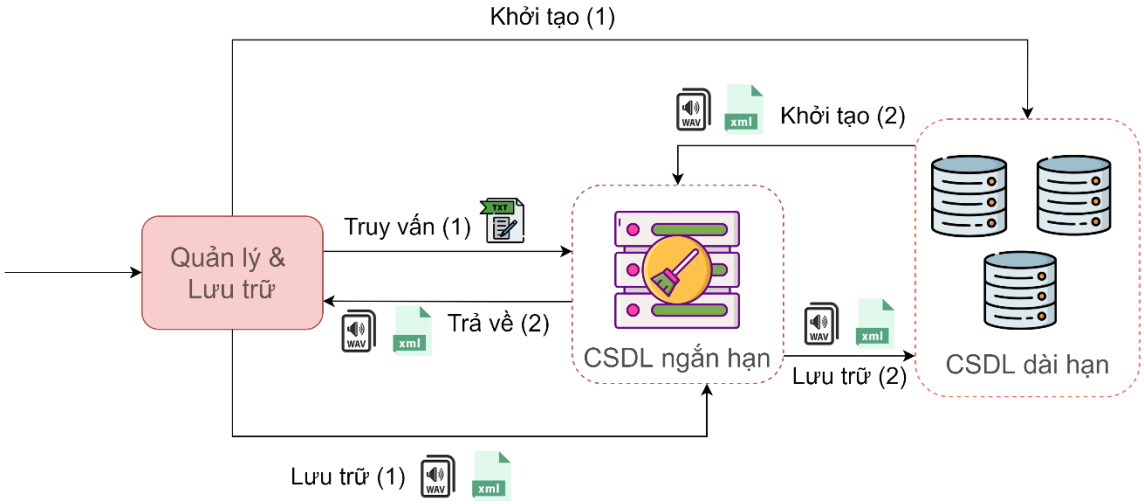
Hàm chi phí tổng cộng cho một chuỗi n mục tiêu và n ứng cử viên tương ứng sẽ bằng tổng của các chi phí mục tiêu và chi phí ghép nối:

$$C(t_1 \dots t_n, c_1 \dots c_n) = \sum_{i=1}^{n} C_t(t_i, c_i) + \sum_{i=2}^{n} C_c(c_{i-1}, c_i)$$

4.3 Thành phần quản lý và lưu trữ dữ liệu

Thành phần quản lý và lưu trữ dữ liệu là một trong ba thành phần chính của hệ thống caching trong tổng hợp tiếng nói tiếng Việt. Thành phần này chịu trách nhiệm lưu trữ và tìm kiếm những đoạn tiếng nói đã được tổng hợp cùng với các thông tin âm học của chúng. Để cho tối ưu về mặt hiệu năng khi triển khai, ở trong thành phần này đồ án đề xuất sử dụng kết hợp hai loại bộ nhớ: (1) bộ nhớ ngắn hạn nhưng có tốc độ đọc ghi nhanh, mục đích vùng nhớ này giúp trình truy vấn và lưu trữ được thực hiện một cách nhanh chóng, (2) bộ nhớ dài hạn

nhưng có tốc độ đọc ghi chậm hơn, mục đích vùng nhớ này này để lưu trữ dữ liệu lâu dài. Mô tả chi tiết về kiến trúc thành phần quản lý và lưu trữ dữ liệu được thể hiện ở *Hình 4.7*.



Hình 4.7 Mô tả quá trình hoạt động của thành phần quản lý và lưu trữ dữ liệu

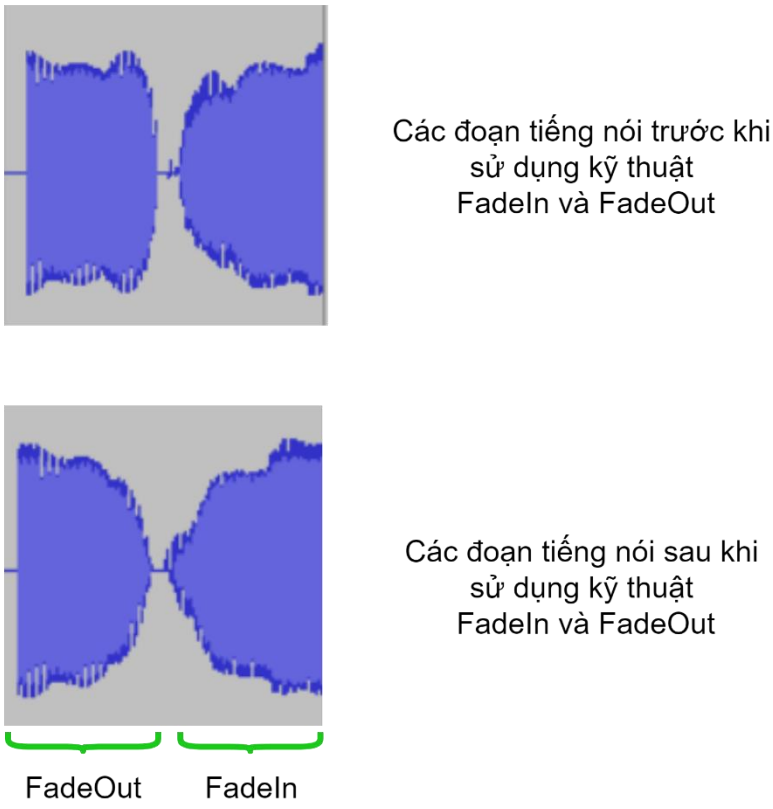
Khi bắt đầu quá trình khởi động hệ thống, thành phần quản lý và lưu trữ sẽ nạp dữ liệu từ vùng nhớ dài hạn vào trong vùng nhớ tạm. Dữ liệu nạp bao gồm các đoạn tiếng nói và thông tin âm học tương ứng. Trong quá trình hoạt động, khi có truy vấn tìm kiếm thông tin, thành phần quản lý và lưu trữ sẽ gửi nội dung văn bản cho vùng nhớ tạm, vùng nhớ tạm sẽ dựa vào nội dung đó để truy xuất ra thông tin. Việc sử dụng vùng nhớ tạm trong quá trình thực thi sẽ giúp cho quá trình truy xuất thông tin được diễn ra với tốc độ nhanh, làm giảm thời gian trễ so với việc đọc ghi vào cơ sở dữ liệu thông thường. Khi có yêu cầu lưu trữ thông tin, thành phần quản lý và lưu trữ sẽ lưu trữ thông tin vào vùng nhớ tạm đầu tiên, sau đó vùng nhớ tạm sẽ tự động kích hoạt gửi thông tin mới sang cho vùng nhớ dài hạn để lưu trữ lại.

4.4 Thành phần ghép nối các đoạn tiếng nói

Những đoạn tiếng nói bao gồm những đoạn tổng hợp mới và những đoạn tái sử dụng thì cần được ghép nối để trả về cho người dùng, tuy nhiên nếu tiến hành ghép nối trực tiếp thì sẽ dẫn tới sự khập khễnh tại những điểm nối và giọng nói tổng hợp không được tự nhiên. Vì vậy trong phần này thì đồ án đề xuất sử dụng hai kỹ thuật nhỏ là kỹ thuật FadeIn[6] và kỹ thuật

[6] FadeIn – một kỹ thuật xử lý âm thanh, giúp cho đoạn âm thanh sẽ có âm lượng bắt đầu từ thấp tới cao, https://hinative.com/en-US/questions/210525

FadeOut[7] nhằm làm mượt những điểm ghép nối. Kỹ thuật FadeIn sẽ được áp dụng tại đầu của những đoạn tiếng nói nhằm giúp cho âm lượng tăng dần từ thấp tới cao, kỹ thuật FadeOut sẽ được áp dụng tại cuối của những đoạn tiếng nói nhằm giúp cho âm lượng giảm dần từ cao tới thấp.



Hình 4.8 Kỹ thuật FadeIn và FadeOut áp dụng cho ghép nối các đoạn tiếng nói

Chương 4 này đã trình bày đề xuất giải pháp trong việc xây dựng hệ thống caching cho những hệ thống ứng dụng tổng hợp tiếng nói tiếng Việt có nội dung lặp lại theo mẫu câu. Đầu tiên là đưa ra kiến trúc tổng quan hệ thống, sau đó trình bày chi tiết từng thành phần chính trong hệ thống bao gồm: (i) Thành phần tìm kiếm âm thanh và (ii) Thành phần quản lý và lưu trữ dữ liệu. Trong chương kế tiếp đồ án sẽ trình bày về cách thức xây dựng và triển khai hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao và hệ thống caching cho ứng dụng tổng đài nhân tạo.

[7] FadeOut – một kỹ thuật xử lý âm thanh, giúp cho đoạn âm thanh sẽ có âm lượng bắt đầu từ cao tới thấp, https://hinative.com/en-US/questions/210525

4.5 Đánh giá kết quả

4.5.1 Phương pháp đánh giá

4.5.1.1 Đánh giá chất lượng giọng nói

Để đánh giá chất lượng giọng nói caching so với giọng nói được tổng hợp từ toàn bộ văn bản, đồ án tiến hành đánh giá bài kiểm tra MOS tương tự như phần 3.5.1 cho hai giọng caching và giọng tổng hợp từ toàn bộ văn bản.

4.5.1.2 Đánh giá hiệu năng

Bởi vì mỗi mẫu câu sẽ có sẽ có những khả năng tái sử dụng khác nhau phụ thuộc vào nội dung và ngữ cảnh của từng mẫu câu, vậy nên đồ án tiến hành đánh giá khả năng tái sử dụng cho 10 loại mẫu câu khác nhau thuộc lĩnh vực tổng đài nhân tạo. Để đánh giá được khả năng tái sử dụng, đầu tiên đồ án sẽ tiến hành tạo cơ sở dữ liệu giả lập – SampleDB, bao gồm 100 câu cho 10 mẫu câu này bằng cách chạy chương trình sinh dữ liệu. Sau đó với mỗi mẫu câu đồ án sẽ tạo 10 câu dữ liệu ngẫu nhiên không nằm trong cơ sở dữ liệu SampleDB và tiến hành đánh giá kết quả tái sử dụng của 10 câu này, sau đó lấy kết quả trung bình. Tỷ lệ tái sử dụng của từng câu được đánh giá bằng số đoạn giọng nói tái sử dụng được chia cho tổng số đoạn giọng nói tạo thành câu và số ký tự tái sử dụng chia cho tổng số ký tự trong câu.

4.5.2 Kết quả đánh giá

4.5.2.1 Kết quả đánh giá chất lượng giọng nói

Đồ án tiến hành đánh giá bài kiểm tra MOS cho bộ dữ liệu NgocHuyen-Vbee cho hai loại giọng: (i) giọng người tổng hợp từ toàn bộ văn bản (Full Synthesis) và (ii) giọng caching. 10 loại mẫu câu được chọn ngẫu nhiên từ hệ thống tổng đài nhân tạo (các mẫu câu này có thể xem ở phần phụ lục A.2), sau đó tiến hành tạo 40 câu dữ liệu một cách ngẫu nhiên từ các mẫu câu này. Tiếp theo tổng hợp các để đoạn giọng nói dựa trên hai chiến lược và tiến hành trộn lẫn giọng nói của hai giọng lại với nhau. Kết quả đánh giá được trình bày ở Bảng 4.1.

Tên giọng	**Điểm MOS**
NgocHuyenFullSynthesis	3.86
NgocHuyenCache (*)	**3.67**

Bảng 4.1 *Kết quả đánh giá điểm MOS của giọng tổng hợp toàn bộ và giọng caching*

Từ kết quả chúng ta có thể nhận thấy rằng giọng nói sử dụng caching có chất lượng gần như ngang bằng với giọng nói tổng hợp toàn bộ 3.84/3.86 ≈ 95%. Đây là một kết quả rất khả quan cho việc áp dụng caching vào hệ thống ứng dụng tổng hợp tiếng nói có đặc thù là lặp lại theo mẫu câu và có tốc độ phản hồi thời gian thực.

4.5.2.2 Kết quả đánh giá hiệu năng

Kết quả đánh giá trên 10 mẫu câu lựa chọn từ hệ thống tổng đài nhân tạo (chi tiết về nội dung từng mẫu câu được để ở phần phụ lục) được trình bày ở *Bảng 4.2*, các mẫu câu được đánh số từ SP001 tới SP010.

Mẫu câu	**Tỷ lệ tái sử dụng tiếng nói (%)**	**Tỷ lệ tái sử dụng ký tự (%)**
SP001	50	37
SP002	45	62
SP003	50	71
SP004	0	0
SP005	66	99
SP006	15	23
SP007	0	0
SP008	0	0
SP009	16	19
SP010	65	90
Trung bình	30.5	30

Bảng 4.2 Kết quả đánh giá tỷ lệ tái sử dụng của 10 mẫu câu

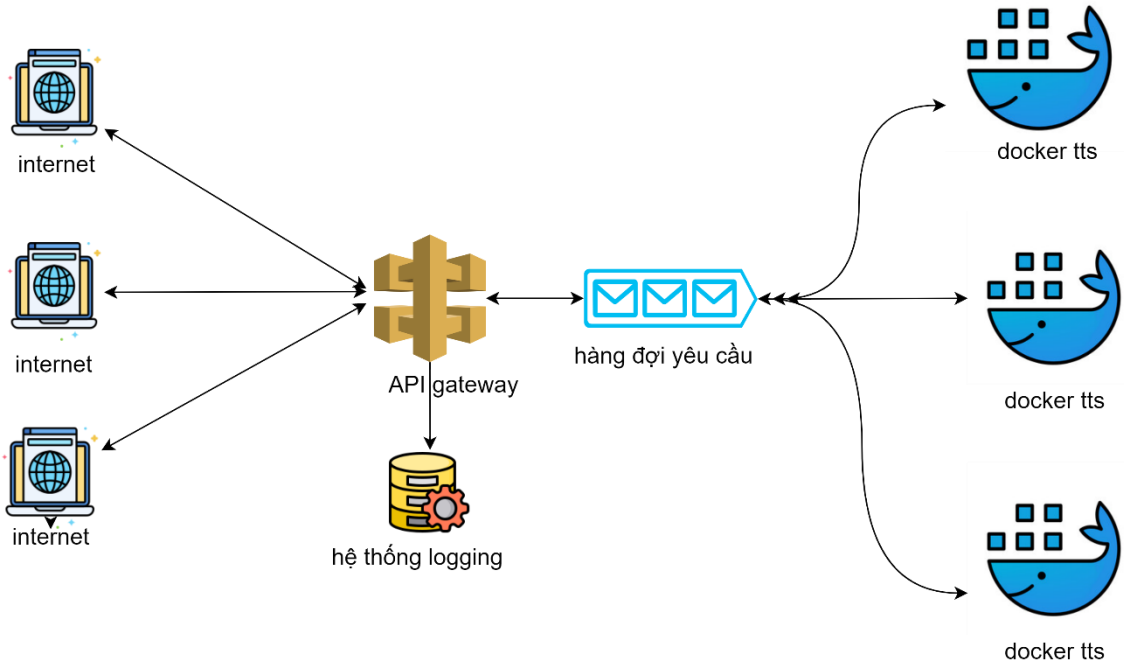
Dựa vào bảng kết quả đánh giá thì có thể thấy tỷ lệ tái sử dụng được của 10 mẫu câu đạt trung bình 30% về số lượng đoạn giọng nói và số số lượng ký tự. Trong bài đánh giá, cơ sở dữ liệu giả lập SampleDB chứa 100 câu cho 10 mẫu câu này, tuy nhiên trong thực tế thì số lượng câu trong cơ sở dữ liệu lớn hơn rất nhiều, vì vậy tỷ lệ tái sử dụng còn có thể cao hơn nhiều nữa so với bài đánh giá.

Chương 5 Phát triển và triển khai hệ thống tổng hợp tiếng nói và hệ thống caching

Chương này đồ án sẽ giới thiệu về kiến trúc triển khai tổng quan của hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao và hệ thống caching cho những hệ thống ứng dụng tổng hợp tiếng nói theo mẫu câu. Sau đó trình bày những nội dung liên quan tới triển khai và tích hợp của hai hệ thống này.

5.1 Hệ thống tổng hợp tiếng nói tiếng Việt

5.1.1 Kiến trúc hệ thống



Hình 5.1 Kiến trúc hệ thống tổng hợp tiếng nói tiếng Việt

Hệ thống tổng hợp tiếng nói được trên khai trên các máy chủ ảo (Virtual Private Server - VPS) bằng công cụ Docker[8] và hàng đợi yêu cầu RabbitMQ[12]. Những yêu cầu tổng hợp tiếng nói được tiếp nhận từ phía Client thông qua máy chủ gateway.

Luồng xử lý chính của hệ thống

Khách hàng gửi yêu cầu tổng hợp giọng nói tới máy chủ gateway, máy chủ gateway tiếp nhận yêu cầu sau đó gửi sang hàng đợi xử lý và lắng nghe kết quả trả về. Hàng đợi xử lý sẽ phân phối yêu cầu tới những Worker của mô hình TTS đang rảnh để xử lý. Sau khi xử lý xong Worker TTS trả kết quả cho hàng đợi, sau đó hàng đợi bắn trở về máy chủ gateway và cuối cùng máy chủ gateway trả kết quả về cho phía khách hàng.

5.1.2 Xây dựng hệ thống

Các công cụ và thư viện dùng để phát triển và triển khai hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao End-To-End được mô tả ở *Bảng 5.1*.

Mục đích	**Công cụ**	**Địa chỉ URL**
IDE	Visual Studio Code	https://code.visualstudio.com/
Huấn luyện mô hình	Pytorch	https://pytorch.org/
Ngôn ngữ	Python	https://www.python.org/
Triển khai dự án	Docker	https://www.docker.com/
Hàng đợi yêu cầu	RabbitMQ	https://www.rabbitmq.com/

Bảng 5.1 Thư viện và công cụ sử dụng để xây dựng dịch vụ tổng hợp tiếng nói

5.2 Hệ thống tổng hợp giọng nói sử dụng kỹ thuật caching

[8] Docker, https://www.docker.com/

5.2.1 Kiến trúc hệ thống

Hệ thống tổng hợp tiếng nói sử dụng caching cho mẫu câu lặp lại được thiết kế dựa trên kiến trúc microservices, các dịch vụ chính bao gồm: (i) dịch vụ tìm kiếm âm thanh, (ii) dịch vụ lưu trữ và quản lý và (iii) dịch vụ ghép nối.

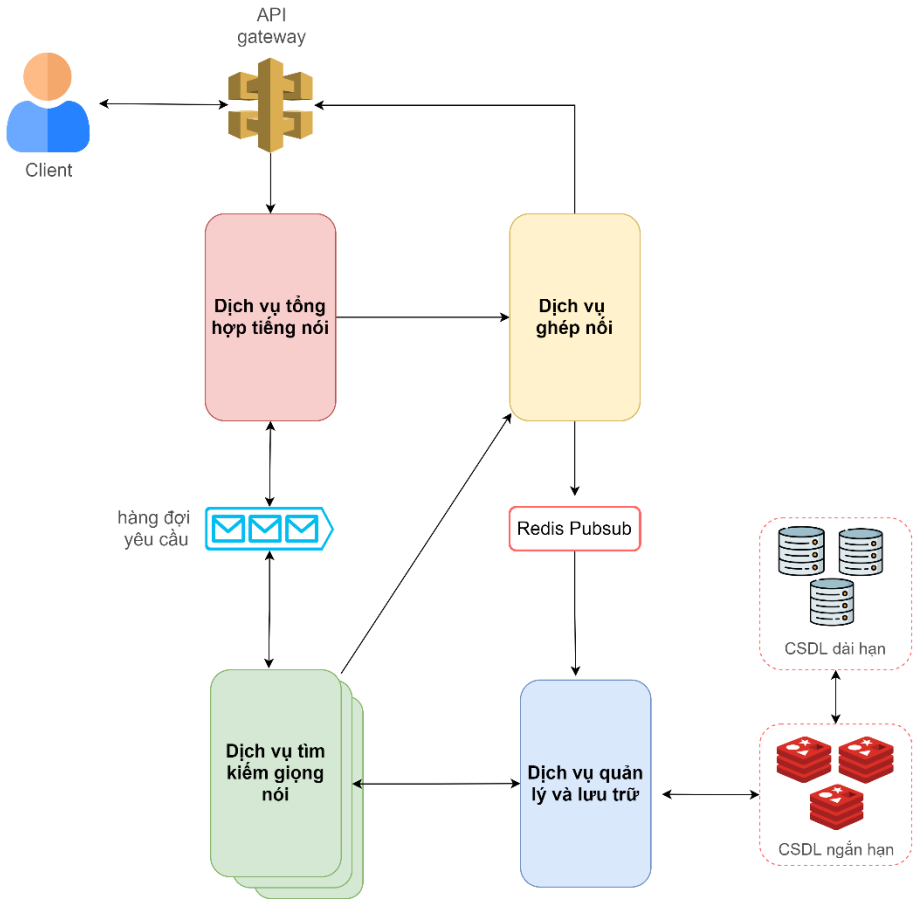
Luồng xử lý chính của hệ thống:

Phía Client (khách hàng) gửi yêu cầu tới máy chủ gateway, sau đó máy chủ gateway gửi yêu cầu cho dịch vụ tổng hợp tiếng nói. Dịch vụ TTS sẽ tiến hành tiền xử lý văn bản và dự đoán thông tin âm học từ văn bản đầu vào. Sau khi xử lý xong, dịch vụ TTS thêm thông tin dự đoán vào tin nhắn, sau đó gửi sang hàng đợi *"Tìm kiếm âm thanh"* và bắt đầu lắng nghe hàng đợi *"Kết quả tìm kiếm"*. Một trong những Worker đang rảnh rỗi của dịch vụ tìm kiếm âm thanh lắng nghe tin nhắn này, sau đó tiến hành phân tách thông tin đầu vào, những thông tin phân tách được gửi sang dich vụ quản lý lưu trữ để truy vấn thông tin. Sau khi tiếp nhận yêu cầu dịch vụ quản lý và lưu trữ sẽ tiến hành truy xuất dữ liệu ở cơ sở dữ liệu tạm dựa trên thông tin đó, rồi trả kết quả về cho dịch vụ tìm kiếm. Dịch vụ tìm kiếm giọng nói nhận dữ liệu trả về và tiến hành phân tích đánh giá, những đoạn văn bản có thể tái sử dụng âm thanh tổng hợp sẽ được thêm thông tin đường dẫn âm thanh tương ứng và gửi sang dịch vụ ghép nối, còn những đoạn văn bản còn lại sẽ được thêm thông tin âm học đã dự đoán và gửi sang dịch vụ TTS thông qua hàng đợi "*Kết quả tìm kiếm*". Sau khi tiếp nhận kết quả tìm kiếm dịch vụ TTS sẽ tổng hợp những đoạn văn bản không thể tái sử dụng được giọng nói dựa trên thông tin âm học trả về. Sau khi hoàn tất quá trình tổng hợp, dịch vụ TTS sẽ gửi kết quả sang dịch vụ ghép nối bao gồm những đoạn tiếng nói đã được tổng hợp mới và các đặc trưng tương ứng. Dịch vụ ghép nối sau khi tiếp nhận đủ các đoạn tiếng nói thì sẽ tiến hành ghép nối và trả về giọng nói tổng hợp cho phía khách hàng, đồng thời dịch vụ ghép nối cũng bắn những thông tin về giọng nói tổng hợp mới và những đặc trưng tương ứng cho dịch vụ quản lý lưu trữ thông qua cơ chế PubSub[9] của Redis[10] (phần cơ chế PubSub sẽ được giải thích chi tiết ở phần 0), ở đây dữ liệu sẽ được lưu trữ vào vùng nhớ tạm RedisDB và vùng nhớ dài hạn MongoDB[11].

[9] Redis PubSub, https://redis.io/topics/pubsub,

[10] Redis, https://redis.io/

[11] MongoDB, https://www.mongodb.com/



Hình 5.2 Kiến trúc hệ thống caching

Trên đây là phần trình bày toàn bộ về kiến trúc tổng quan và luồng hoạt động chính của hệ thống caching cho tổng hợp tiếng nói phục vụ cho những hệ thống ứng dụng tổng hợp tiếng nói theo mẫu câu. Tiếp theo đồ án sẽ trình bày về phần thiết kế và xây dựng các dịch vụ chính của hệ thống bao gồm: (i) Dịch vụ tìm kiếm giọng nói và (ii) Dịch vụ quản lý lưu trữ. Cuối cùng là phần trình bày về cách đóng gói và triển khai hệ thống.

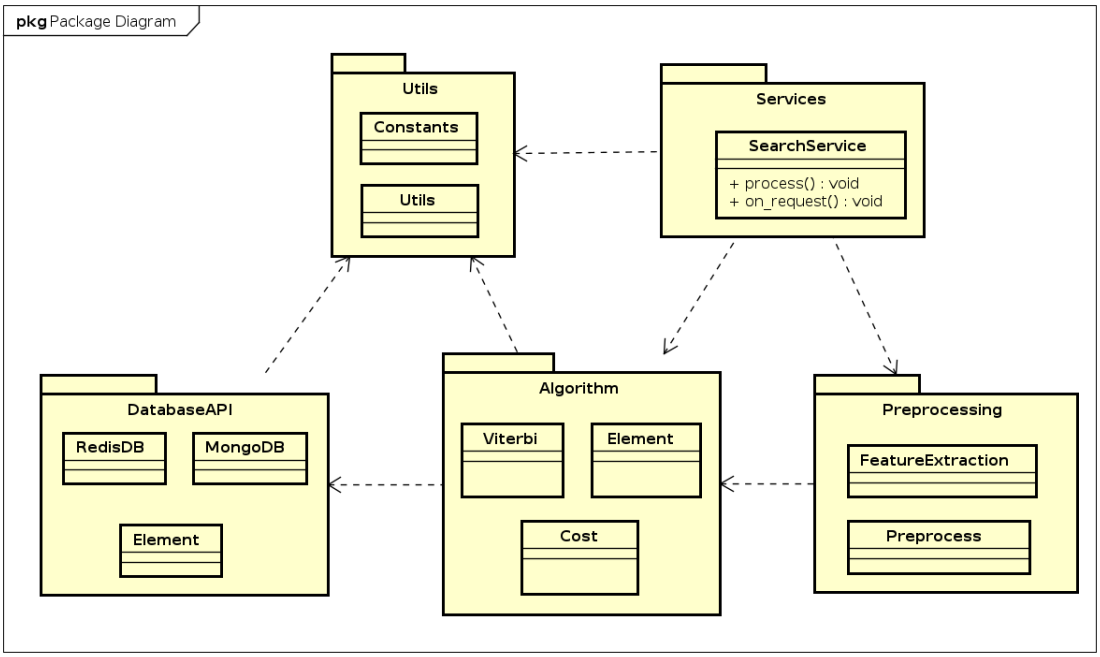
5.2.2 Dịch vụ tìm kiếm giọng nói

a) Thiết kế

Hình 5.3 mô tả thiết kế gói của dịch vụ tìm kiếm âm thanh. Gói Services chứa lớp SearchService có nhiệm vụ nhận yêu cầu đầu vào từ hàng đợi RabbitMQ[12] sau đó gửi sang Preprocessing để tiền xử lý đặc trưng đầu vào. Gói Algorithm chứa các lớp thực thi thuật

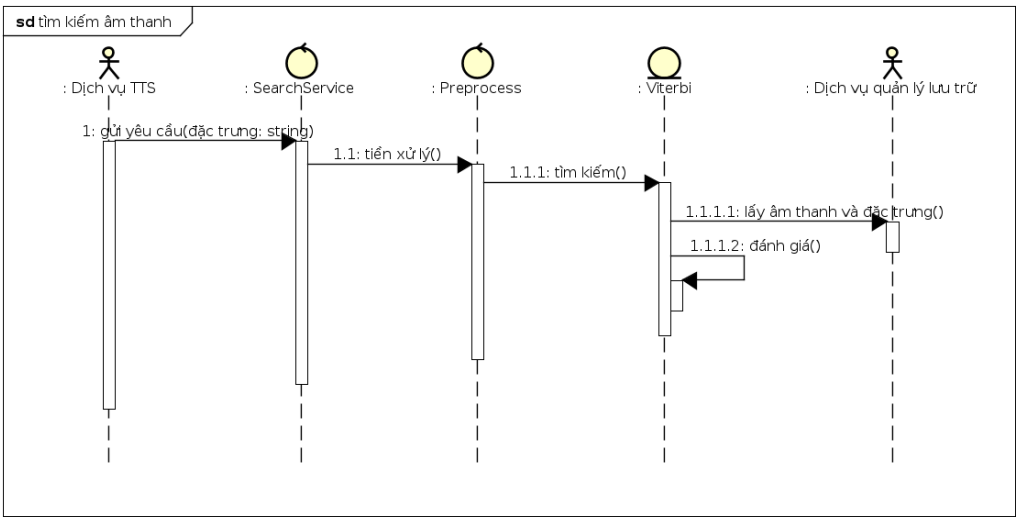
[12] RabbitMQ, https://www.rabbitmq.com/

toán Viterbi [6] như lớp Cost dùng để tính hàm đánh giá giữa các âm vị, lớp Viterbi và Element dùng để cài đặt thuật toán tìm kiếm. Gói DatabaseAPI chứa các lớp dùng để giao tiếp với dịch vụ quản lý và lưu trữ. Gói Utils chứa các lớp tiện dùng trong toàn hệ thống.



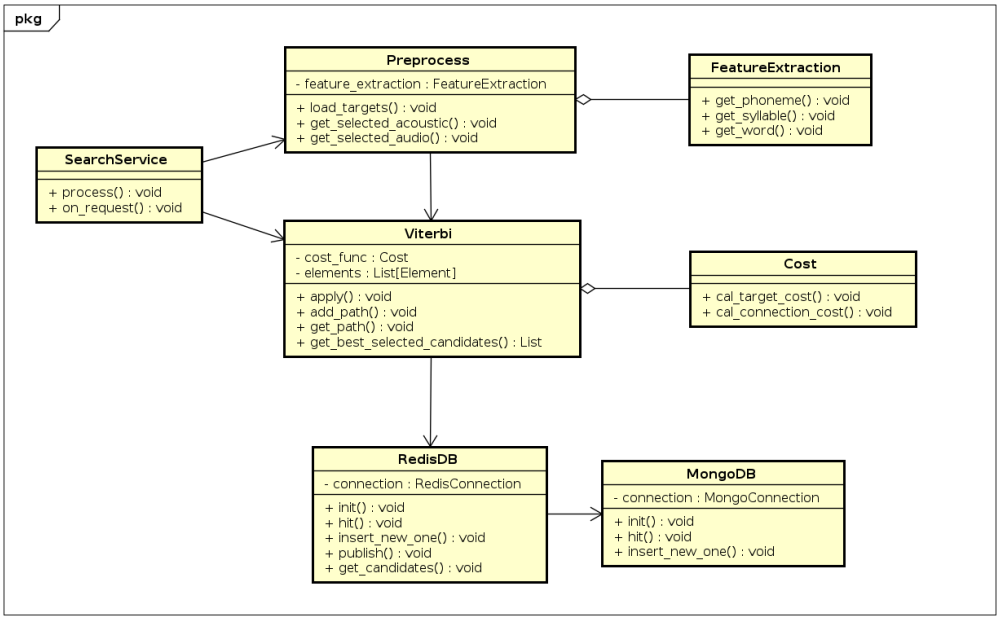
Hình 5.3 Sơ đồ kiến trúc gói của dịch vụ tìm kiếm âm thanh

Hình 5.4 mô tả luồng hoạt động của hệ thống tìm kiếm. Dịch vụ TTS sẽ gửi yêu cầu tìm kiếm âm thanh cùng với thông tin đặc trưng âm học của văn bản thông qua hàng đợi RabbitMQ, sau đó SearchService sẽ nhận và gửi yêu cầu sang Preprocess để tiến hàng tiền xử lý dữ liệu đầu vào. Tiếp theo Preprocess gửi thông tin tin đã được chuẩn hóa sang bên thực thể Viterbi để tiến hành đánh giá và tìm kiếm âm thanh, trong quá trình đó Viterbi sẽ truy vấn RedisDB thông qua cơ chế khóa-giá trị (key-value) để tiến hành truy vấn lấy thông tin từ cơ sở dữ liệu tạm.



Hình 5.4 Biểu đồ trình tự quá trình tìm kiếm âm thanh

Hình 5.5 trình bày sơ đồ thiết kế lớp chi tiết của một vài lớp quan trọng của hệ thống tìm kiếm âm thanh bao gồm: Preprocess, FeatureExtraction, Viterbi, Cost, SearchService, RedisDB và MongoDB.



Hình 5.5 Biểu đồ thiết kế lớp chi tiết của hệ thống tìm kiếm âm thanh

b) Xây dựng hệ thống

Như đã mô tả ở phần tổng quan, trong đồ án này đã sử dụng hai loại bộ nhớ là bộ nhớ tạm và bộ nhớ dài hạn, bộ nhớ tạm được em triển khai bằng phần mềm Redis với cơ chế truy vấn

dạng khóa-giá trị (key-value) để giúp cho việc truy vấn được thực hiện nhanh và giảm thời gian trễ của thao tác đọc ghi. Với bộ nhớ lưu trữ dài hạn em dùng cơ sở dữ liệu MongoDB, đây là cơ sở dữ liệu dạng phi cấu trúc (NoSQL[13]) với cơ chế lưu trữ theo dạng khóa-giá trị (key-value) cho phép việc truy vấn dễ dàng so với CSDL có cấu trúc (SQL[14]) và linh hoạt cho việc mở rộng. Để có thể tăng khả năng chịu tải và xử lý song song, em đã dùng RabbitMQ làm hàng đợi tin nhắn và triển khai hệ thống bằng Docker[8]. Docker là một công cụ cho phép đóng gói và triện khai hệ thống thành một hoặc nhiều tiến trình độc lập với nhau gọi là Worker, các yêu cầu gửi tới hàng đợi sẽ được gửi tới các Worker đang rảnh rỗi xử lý và sau đó trả kết quả về một hàng đợi "phản hồi" được định nghĩa trước. Cách xây dựng hệ thống này giúp cho hệ thống có thể xử lý đồng thời được nhiều yêu cầu gửi tới.

Các công cụ và thư viện dùng để phát triển và triển khai hệ thống tìm kiếm tiếng nói được mô tả ở *Bảng 5.2*

Mục đích	**Công cụ**	**Địa chỉ URL**
IDE	Visual Studio Code	https://code.visualstudio.com/
Hàng đợi tin nhắn	RabbitMQ	https://www.rabbitmq.com/
Cơ sở dữ liệu tạm	Redis	https://redis.io/
Cơ sở dữ liệu dài hạn	MongoDB	https://www.mongodb.com/
Triển khai hệ thống	Docker	https://www.docker.com/
Ngôn ngữ	Python	https://www.python.org/

Bảng 5.2 Thư viện và công cụ sử dụng để xây dựng dịch vụ tìm kiếm âm thanh

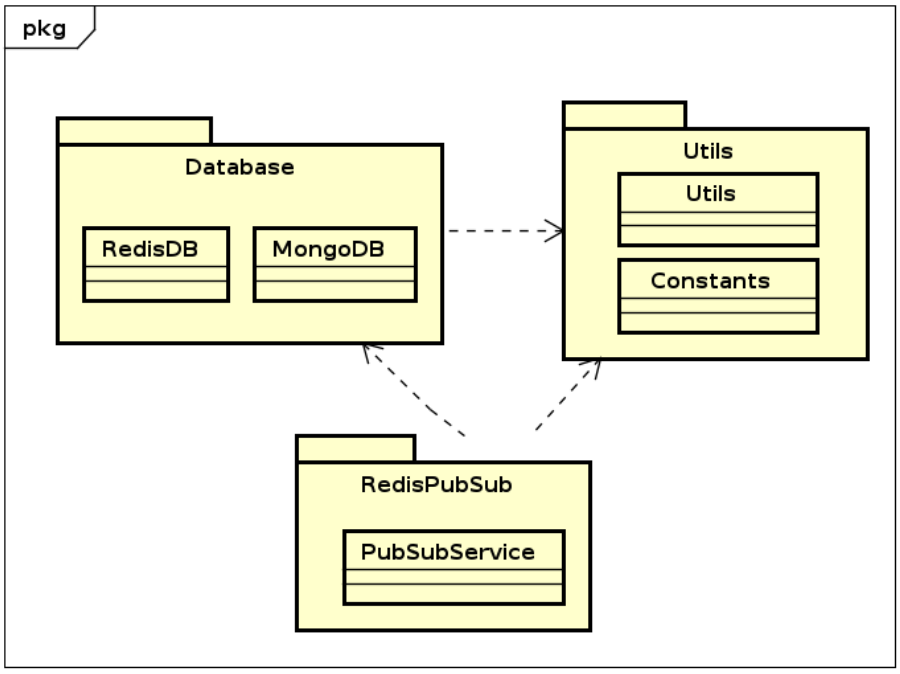
[13] Dạng dữ liệu phi cấu trúc, https://www.mongodb.com/nosql-explained

[14] Dạng dữ liệu có cấu trúc, http://www.sqlcourse.com/intro.html

5.2.3 Dịch vụ quản lý và lưu trữ

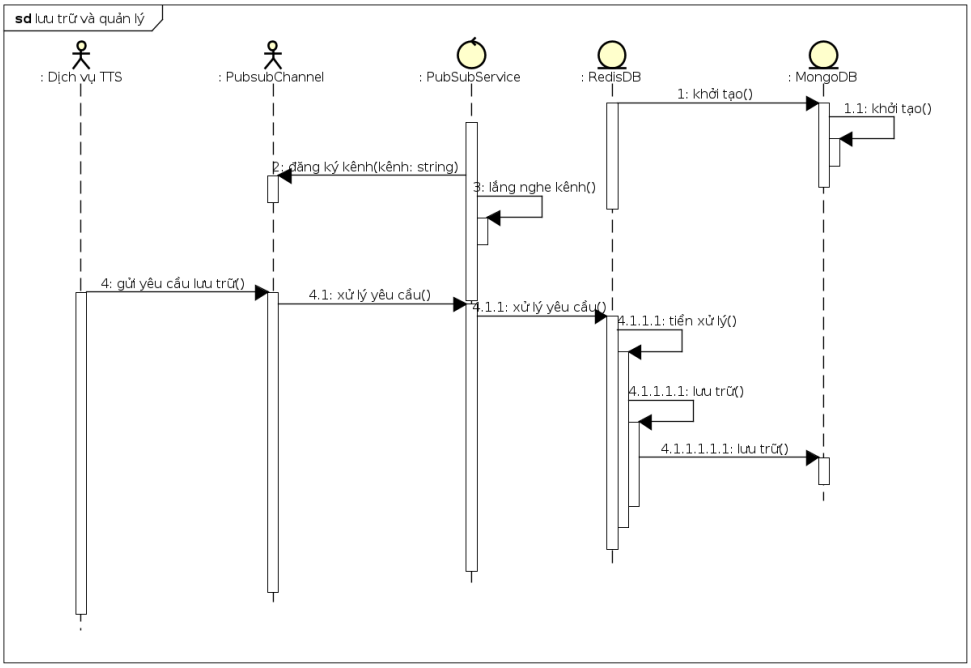
a) Thiết kế

Hình 5.6 mô tả sơ đồ thiết kế gói của hệ thống quản lý và lưu trữ. Hệ thống bao gồm gói Utils chứa các lớp tiện ích và hằng số sử dụng trong toàn bộ hệ thống, gói RedisPubSub chịu trách nhiệm đăng ký và lắng nghe các kênh để tiếp nhận yêu cầu. Yêu tiếp nhận sẽ được gửi đến lớp RedisDB của gói Database, ở đây yêu cầu được xử lý và lưu vào vùng nhớ tạm và vùng nhớ dài hạn.



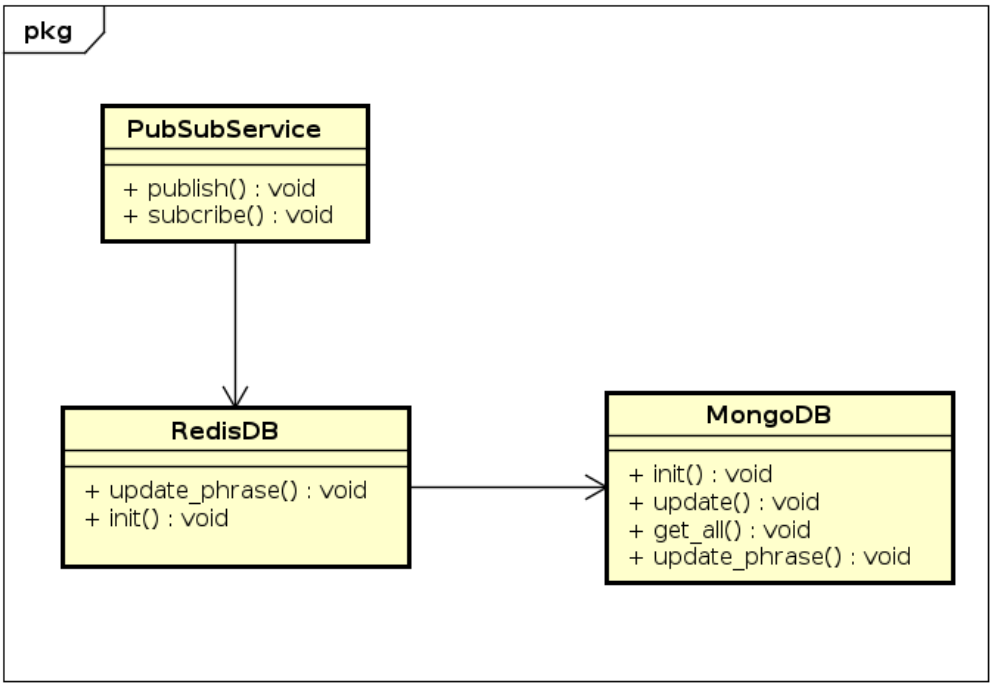
Hình 5.6 Biểu đồ thiết kế gói của hệ thống quản lý và lưu trữ

Hình 5.7 mô tả luồng hoạt động của hệ thống quản lý và lưu trữ khi nhận được yêu cầu bắn sang từ dịch vụ tổng hợp tiếng nói qua cơ chế PubSub. Sau khi tiếp nhận yêu cầu từ lớp PubSubService, lớp RedisDB sẽ tiến hành tiền xử lý để chuẩn hóa dữ liệu, sau đó lưu trữ vào vùng nhớ ngắn hạn và vùng nhớ dài hạn.



Hình 5.7 Biểu đồ hoạt động của hệ thống quản lý và lưu trữ

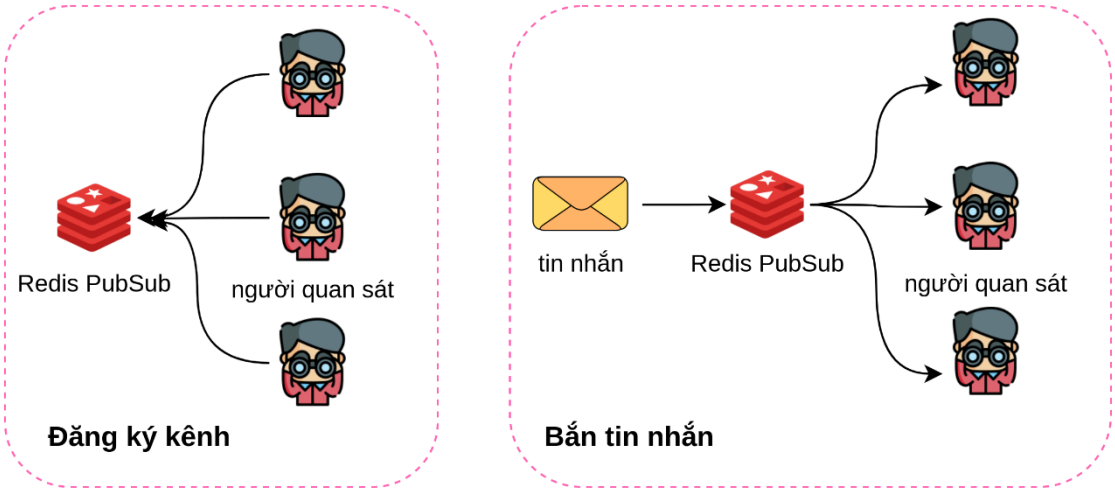
Hình 5.8 mô tả thiết kế chi tiết lớp của một số lớp quan trọng trong hệ thống quản lý và lưu trữ bao gồm: PubSubService, RedisDB và MongDB.



Hình 5.8 Biểu đồ lớp chi tiết của hệ thống quản lý và lưu trữ

b) Xây dựng hệ thống

Giống như mô tả ở phần tổng quan 0, ở hệ thống quản lý và lưu trữ đồ án sử dụng Redis cho cơ ở dữ liệu tạm thời và MongoDB cho cơ sở dữ liệu dài hạn. Ngoài ra ở hệ thống này đồ án còn dùng thêm cơ chế PubSub của Redis để tiếp nhận yêu cầu từ dịch vụ khác. PubSub là cơ chế tự động bắn yêu cầu được gửi tới một kênh cho những đối tượng đang lắng nghe kênh đó, ở hệ thống quản lý và lưu trữ thì ban đầu sẽ khởi tạo cơ sở dữ liệu tạm thời bằng cách nạp các bản ghi từ cơ sở dữ liệu dài hạn, sau đó hệ thống sẽ đăng ký và lắng nghe một kênh để nhận yêu cầu lưu trữ. Khi có một yêu cầu gửi tới kênh đang lắng nghe thì cơ chế PubSub sẽ bắn yêu cầu sang hệ thống để xử lý, *Hình 5.9* mô tả cách thức hoạt động của cơ chế Redis PubSub. Hệ thống được đóng gói và triển khai bằng công cụ Docker nhằm giúp việc dễ dàng triển khai và mở rộng.



Hình 5.9 Cơ chế hoạt động của Redis PubSub (publish-subcribe)

Các công cụ và thư viện sử dụng để xây dựng hệ thống quản lý và lưu trữ được mô tả trong *Bảng 5.3*.

Mục đích	**Công cụ**	**Địa chỉ URL**
IDE	Visual Studio Code	https://code.visualstudio.com/
Hàng đợi tin nhắn	RabbitMQ	https://www.rabbitmq.com/
Cơ sở dữ liệu tạm	Redis	https://redis.io/

Cơ sở dữ liệu dài hạn	MongoDB	https://www.mongodb.com/
Triển khai hệ thống	Docker	https://www.docker.com/
Ngôn ngữ	Python	https://www.python.org/

Bảng 5.3 Thư viện và công cụ sử dụng để xây dựng dịch vụ quản lý và lưu trữ

5.3 Đóng gói và triển khai

5.3.1 Cách thức triển khai

Đối với hệ thống tổng hợp tiếng nói chất lượng cao cho tiếng Việt, hệ thống được đóng gói bằng công cụ Docker và triển khai lên trên máy chủ ảo (VPS) của nền tảng điện toán đám mây Amazon Cloud. Để tăng cường khả năng chịu tải hệ thống sử dụng RabbitMQ một cơ chế tiếp nhận yêu cầu theo hàng đợi và triển khai nhiều Worker tổng hợp tiếng nói cùng lúc bằng cách sử dụng Docker để tăng cường khả năng xử lý song song.

Đối với hệ thống caching cho ứng dụng tổng đài nhân tạo, các dịch vụ tìm kiếm âm thanh và quản lý lưu trữ được đóng gói và triển khai trên các máy chủ vật lý bằng công cụ Docker, để tăng cường khả năng chịu tải thì đồ án có triển khai thêm cơ chế hàng đợi bằng công cụ RabbitMQ. Phần CSDL sử dụng Redis cho CSDL ngắn hạn và Mongo cho CSDL dài hạn. Còn dịch vụ tổng hợp tiếng nói được triển khai thành máy chủ Gateway bằng thư viện Flask[15] của ngôn ngữ Python sử dụng môi trường Anaconda[16].

5.3.2 Kết quả triển khai

Đối với hệ thống tổng hợp tiếng nói chất lượng cao cho Tiếng Việt, hiện tại hệ thống đã được triển khai thành công ở trang web vbee.vn**Error! Bookmark not defined.** và xử lý h àng chục ngàn yêu cầu mỗi ngày từ người dùng thực tế. Ngoài ra hệ thống còn cung cấp API cho nhiều ứng dụng thực tế khác như VietStudy**Error! Bookmark not defined.**, thuyết minh phim, … Hệ thống nhận được nhiều phản hồi tích cực từ phía người dùng thực tế.

[15] Flask - Framework lập trình web api, https://flask.palletsprojects.com

[16]Anaconda, https://www.anaconda.com/

Đối với hệ thống caching toàn bộ các dịch vụ đều được triển khai thành công ở trên máy chủ vật lý có cổng và địa chỉ IP thực sự. Tất cả các dịch vụ đều đã được tích hợp với nhau và kiểm thử thành công cho hệ thống tổng đài nhân tạo tại địa chỉ cp-dev.aicallcenter.vn[17].

[17] Dev-AI Call Center – https://cp-dev.aicallcenter.vn/

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Tổng kết lại, đồ án đã phát triển và xây dựng được hai hệ thống: (i) hệ thống tổng hợp tiếng nói tiếng Việt chất lượng cao - FHG, (ii) một hệ thống caching cho những ứng dụng tổng hợp giọng nói có tính chất lặp lại – TTS Caching. Hệ thống FHG bao gồm 2 thành phần: mô hình âm học End-To-End FastSpeech2 và mô hình sinh tiếng nói Hifi-Gan. Mô hình FastSpeech2 sử dụng kiến trúc phi hồi quy (Non Autoregressive) từ đó khắc phục những nhược điểm như mất từ lặp từ của mô hình hồi quy (Autoregressive), mô hình Hifi-Gan sử dụng kiến trúc Discriminator-Generator của mạng GAN giúp cho giọng nói có chất lượng cao và tốc độ tái tạo tiếng nói nhanh. Hệ thống TTS Caching được xây dựng dựa trên thuật toán tìm kiếm Viterbi – một thuật toán quy hoạch động trong tìm kiếm chuỗi tối ưu, thuật toán giúp tìm kiếm chuỗi các đoạn giọng nói của những nội dung đã tổng hợp, tiếp đó tính toán chi phí bao gồm: chi phí lựa chọn và chi phí trên chuỗi các đoạn giọng nói đó và cuối cùng đánh giá để đưa ra quyết định tái sử dụng giọng nói cho văn bản đầu vào.

Hệ thống FHG đã được triển khai thành công tại địa chỉ https://vbee.vn/ bằng cách sử dụng các máy chủ ảo (VPS) trên nền tảng điện toán đám mây của Amazon và đóng gói bằng công cụ Docker. Hệ thống TTS Caching đã được triển khai thử nghiệm thành công lên trên hệ thống tổng đài nhân tạo AI Call Center tại địa chỉ https://cp-dev.aicallcenter.vn/, bằng cách sử dụng các máy chủ vật lý và đóng gói bằng công cụ Docker.

Kết quả đánh giá cho thấy giọng nói tổng hợp có sự tự nhiên lên tới 90% so với giọng nói người thật và hệ thống nhận được nhận được những phản hồi tích cực từ phía người dùng. Thống kê cho thấy mỗi ngày hệ thống FHG tiếp nhận và xử lý hàng chục ngàn yêu cầu từ doanh nghiệp và người dùng thực tế. Với hệ thống TTS Caching, giải pháp đề xuất cho kết quả trung bình đạt tới 95% sự tương đồng so với văn bản tổng hợp toàn bộ và chi phí tổng hợp tiết kiệm được 30%, từ đó giúp giảm thời gian và chi phí tổng hợp một cách đáng kể cho ứng dụng có đặc thù nội dung lặp lại theo mẫu.

6.2 Hướng phát triển

Trong thời gian tiếp theo thì đồ án sẽ tiếp tục phát triển hệ thống tổng hợp tiếng nói chất lượng cao bằng việc tìm hiểu và thử nghiệm những mô hình tổng hợp tiếng nói mới nhằm cải thiện chất lượng giọng nói hơn nữa. Với hệ thống caching thì sẽ phát triển tiếp để tích hợp triển khai cùng với hệ thống tổng đài nhân tạo ở trên nền tảng Cloud, sau đó đồ án sẽ mở rộng nghiên cứu và phát triển thêm cho hệ thống để có thể xử lý được cho văn bản bất kỳ để tăng sự đa dạng của hệ thống, từ đó hệ thống có thể áp dụng cho nhiều ứng dụng khác.

Tài liệu tham khảo

[1] “What is Natural Language Processing?” [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html. [Accessed: 10-May-2021].

[2] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, Yonghui Wu: "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions", Feb 2018

[3] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, Ming Zhou: " Neural Speech Synthesis with Transformer Network ", Jan 2019

[4] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu: " FastSpeech: Fast, Robust and Controllable Text to Speech ", Nov 2019

[5] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu: " FastSpeech 2: Fast and High-Quality End-To-End Text to Speech", Mar 2021

[6] Andrew Viterbi: "a decoding algorithm for convolutional codes", 1967

[7] T. T. T. Nguyen, “HMM-based Vietnamese Text-To-Speech : Prosodic Phrasing Modeling, Corpus Design System Design, and Evaluation,” phdthesis, Université Paris Sud - Paris XI, 2015.

[8] Kainan Peng, Wei Ping, Zhao Song, Kexin Zhao, "Non-Autoregressive Neural Text-to-Speech", Jun 2020

[9] "What Are Word Embeddings? " [Online]. Available: https://machinelearningmastery.com/what-are-word-embeddings/ [Accessed 10-May-2021]

[10] "Why One-Hot Encode Data in Machine Learning?" [Online]. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ [Accessed 10-May-2021]

[11] Ilya Sutskever, Oriol Vinyals, Quoc V. Le , "Sequence to Sequence Learning with Neural Networks", Dec 2014

[12] Ralf C. Staudemeyer, Eric Rothstein Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks", Sep 2019

[13] Yixuan SuDeng, CaiYan, WangYan Wang, Nigel Collier, "Non-Autoregressive Text Generation with Pre-trained Language Models", Feb 2021

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", Dec 2017

[15] Jungil Kong, Jaehyeon Kim, Jaekyoung Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis", Oct 2020

[16] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, John Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning", Feb 2018

[17] TS Nguyễn Thị Thu Trang, "Hệ thống chuẩn hóa văn bản và chuyển văn bản sang âm vị", [Online] Available: https://lab-tts.vbeecore.com/ [Accessed 30-May-2021]

[18] Công ty cổ phần công nghệ và dịch vụ Vbee , "Dữ liệu huấn luyện chuyển đổi văn bản sang tiếng nói", Website: https://vbee.vn/, [Accessed 10-May-2021]

[19] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, “Hidden Markov models based on multi-space probability distribution for pitch pattern modeling,” in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258), Mar. 1999, vol. 1, pp. 229–232 vol.1, doi: 10.1109/ICASSP.1999.758104.

[20] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, An adaptive algorithm for mel-cepstral analysis of speech, vol. 1. 1992, p. 140 vol.1.

[21] A. W. Black, “D¡ epartment of Computer Science, Nagoya Institute of Technology Language Technologies Institute, Carnegie Mellon University,” p. 4.

[22] L. Dao and A. L. Society, “The Vietnamese classifiers ‘CON’, ‘CÁI’ and the Natural Semantic Metalanguage (NSM) approach: a preliminary study,” Sep. 2012, Accessed: Apr. 10, 2021. [Online]. Available: https://openresearch-repository.anu.edu.au/handle/1885/9327.

[23] H. Zen, A. Senior, and M. Schuster, “STATISTICAL PARAMETRIC SPEECH SYNTHESIS USING DEEP NEURAL NETWORKS,” p. 5.

[24] H. P. Combrinck and E. C. Botha, On The Mel-scaled Cepstrum. 1996.

[25] Mustapha Oloko-oba, Ibiyemi T.S, Osagie Samuel, “Text-to-Speech Synthesis Using Concatenative Approach”, October 2016

[26] D. W. Griffin and J. S. Lim, “Signal estimation from modified short-time Fourier transform,” IEEE Transactions on Acoustics, Speech and Signal Processing, pp. 236–243, 1984.

[27] Jianxin Wu, “Introduction to Convolutional Neural Networks”, May 2017

[28] Do Van Thao, Tran Do Dat, and Nguyen Thi Thu Trang, “Non-uniform unit selection in Vietnamese speech synthesis. In Proceedings of the Second Symposium on Information and Communication Technology” (SoICT 2011)

[29] Nguyen Thi Thu Trang, Pham Thanh Thi, and Tran Do-Dat, “A method for Vietnamese text normalization to improve the quality of speech synthesis. In Proceedings of the First Symposium on Information and Communication Technology (SoICT 2010)”

[30] Nguyen Thi Thu Trang, Tran Do Dat, Rilliard Albert, Alessandro Christophe, and Pham Thi Ngoc Yen, “Intonation issues in HMM-based speech synthesis for Vietnamese. In The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU’14)”, May 2014.

Phụ lục

A.1 Chi tiết các câu sử dụng cho bài kiểm thử chất lượng giọng nói

STT	Văn bản
01	Những dặm đường quả là ghê gớm, bởi vì mỗi bước tiến lên phía trước là mỗi bước liều mạng của người và chó, pe rôn đi đầu dò đường, đã hàng chục lần sụt cả người xuống mặt băng mỏng bị vỡ.
02	Thế là một tiếng kêu đau đớn làm xáo động buổi lễ yên tĩnh và người ta thấy con chó điên cuồng đau điếng quay mòng như một ngôi sao chổi trong nhà thờ cho tới khi chủ nó rứt được kẻ hành hạ.
03	Các Kỵ Sĩ Đen từ phía sau đang bị bỏ rơi, thậm chí những con chiến mã ghê gớm của chúng cũng không thể theo kịp tốc độ của con ngựa eo-f trắng của g-lô-phin-đơ.
04	Công ty Cổ phần Bảo hiểm Viễn Đông, trân trọng gửi đến Quý khách Giấy chứng nhận bảo hiểm điện tử trọn gói hộ gia đình, do đào thị mai liên có số điện thoại không chín tám hai, không bẩy chín, chín tám không mua gửi tới.
05	Hắn dốc hộc jét-từn, khoá máy, nhấn thử và mời tôi vào chơi, rồi ra đứng quầy đếm xía mấy cọc jét-từn tính tiền với quán rượu.
06	Nhưng một quả cầu po-li-gâu thì chắc chắn không phải là một quả cam, cho nên bọn trẻ vẫn nhìn thấy mọi người cựa quậy một cách nôn nóng trên chỗ ngồi như thể phòng y tế đột ngột giảm ô-xi.
07	Đừng nói tới việc nhà anh không hề ép anh cưới em, mà nếu, anh đã không muốn thì cha má có nói gì, có làm gì thì cũng không bao giờ ép được anh đâu, anh đâu phải là loại người dễ bị người ta sắp đặt.

08	Thực không cần đâu, cậu Tử Thần Ông chú thịt đưa lên miệng còn không chịu ăn, vừa miệng đây đẩy từ chối, vừa chân biết thời thế lùi ra sau.
09	Công ty Cổ phần Bảo hiểm Viễn Đông trân trọng gửi đến Quý khách Giấy chứng nhận bảo hiểm điện tử tai nạn cá nhân do nguyễn thị hoàng linh có số điện thoại không ba bẩy bẩy, không không không, năm bẩy tám mua gửi tới.
10	Người chủ mới đó anh, anh ta trông coi mảnh đất đó, trông anh ta không giống một người làm vườn cho lắm, nhưng cái cách anh ta chăm sóc cà-phê chứng tỏ anh ta cũng có kiến thức, và hình như anh ta đang muốn làm thay đổi mảnh đất đó của mình.
11	Tôi nói gì sai sao Lặng yên một chút nhìn người đối diện đang cúi đầu run lên cầm cập như sốt, Lăng Tử Thần có chút lo lắng, tay đưa về phía gò má ửng lên dữ dội có vẻ quan tâm lắm.
12	Cô bằng trạc tuổi bạn, nhưng cao hơn da cũng nâu quần áo cũng tầm thường như cậu, tóc buộc khăn mù soa đỏ hai tay đeo vòng bạc con, người coi mảnh dẻ, yết ớt, có lẽ cô cũng đã chịu nhiều nỗi gian truân.
13	Máy lọc nước na-nô gây-xờ, sử dụng công nghệ na-nô, là một phát mình nổi tiếng từ rất lâu trên thế giới ứng dụng vào Lõi lọc a-ra-gừn, là một sản phẩm công nghệ tuyệt vời về xứ lý nước mà chưa một công nghệ nào có thể sánh kịp tại thời điểm hiện tại.
14	Tình hình là mấy hôm trước, bạn lướt oát thấy có một bộ ê đít mà viu hơn một mờ, cũng đăng trăm mấy chương nghĩ nó hoàn rồi nên đọc.
15	Mô phin Chúng tôi đã cho bạn nguyễn thị tư là người thân của bạn vay số tiền hai triệu năm trăm nghìn Việt Nam đồng nhưng quá hạn chưa nhận lại được tiền vay.
16	Thêm một bằng chứng nữa về thái độ khó hiểu của người Mỹ đối với Nguyễn Khánh dù cho chỉ mới sáu mươi ngày trước đó, Bộ Ngoại giao Mỹ nhắc lại cam kết ủng hộ Chính phủ Nguỵ.
17	Bây giờ, cô con gái Hứa Oa Ni đã mười bẩy tuổi, vậy mà bệnh đau lưng vẫn chưa dứt hẳn, gặp hôm trái gió trở trời lại tái phát, Bố Đại phải đánh giác cho vợ.

18	Một số lượng khiêm tốn các doanh nghiệp biết áp dụng nguyên lý u ét pê, đã gặt hái thành công rực rỡ hơn rất nhiều, so với các doanh nghiệp không có u ét-x pê.
19	Anh không hề bổ sung thêm bất kỳ vật dụng nào ngoài những đồ đạc tối cần thiết mà bà trang bị cho căn phòng, không tranh ảnh, không đồ trang trí, không dấu vết ấm áp nào của bàn tay con người.
20	sau quá nhiều ngày sống dưới ánh đuốc, ánh đèn khí ga và ánh sáng mờ mờ ảo ảo của ngọn đèn phù thuỷ, ánh đèn huỳnh quang khiến mọi thứ tái nhách và không tự nhiên.
21	Họ máy móc đi đến cột bêu tội nhân, quần chúng xúm xít xung quanh, mỗi lúc một đông thêm, ắt hẳn họ đã quên phắt Cái Lỗ chuột, nếu như ơ-x-tách không nhắc.
22	Khi bơm dầu hoặc nước vào đường ống lần đầu thì tải trọng ở trên dây treo tăng lên đột ngột và sinh ra độ võng tương đối lớn, để bù đắp cho độ dư của đường ống, khi đặt ta giải quyết bằng phương pháp dồn ống để có khoảng cách hai đầu ống nhỏ nhất.
23	Thơ văn của họ tuy không phong phú, ít cách tân trong nghệ thuật thể hiện, nhưng tràn đầy nhiệt huyết, làm thức dậy ở nhân dân ý thức dân quyền, ý chí đánh giặc, tự cường, tinh thần độc lập.
24	Thưa trung uý, tôi tới để hỏi xem có thể xin một văn bản chính thức chứng nhận nạn nhân đúng là v-la-đi-mia y-a-cóp.
25	Cây chổi được treo toòng teng trên nhánh ổi bằng hai sợi dây cột ở hai đầu, trông từa tựa cái xích đu nhưng thế ngồi của Khoa thì đúng là thế ngồi hùng dũng của một kỵ sĩ đang xông ra chiến trường, ở đây là, tình trường.
26	Ngay cả cảng nổi tiếng zét-đơ cũng không thân quen với những gương mặt a-lơ, xắc-xơn trên đường phố đến thế, nhưng vào tuần thứ hai của tháng Tám, đa-ran đang chao đảo vì cuộc xâm lược.
27	Ngoài chuyện bò ra, hắn luôn tự biết mình chỉ là một thằng vụng, thằng khùng, luôn luông chịu yếm thế, bị coi thường, mất điểm.

28	Vả lại, hen-ri cũng đã chẳng đưa ra đề nghị ấy nếu như anh ta không chắc chắn là vụ vay mượn này không được sự tán thành của a-len.
29	Ông có quyển sách to đấy, nhưng tôi khuyên ông là hãy gập nó vào và để lại khách sạn, bởi với tôi giai-pua giống như một quyển sách mở.
30	Tuy vậy, cái lần cô ta nói chuyện riêng với pơ-tơ, cách ông ta mỉm cười dịu ngọt, việc bộ hồ sơ khách hàng thân thiết của phòng xêu bị cô ta tự ý điều chỉnh đã khiến tôi bắt đầu nghi ngại.
31	Lịch sử tái diễn, cái mặt thù lù như oan hồn của thằng em vừa đập thẳng vào mặt cô thì cũng với vận tốc ánh sáng như lúc xuất hiện, cánh cửa tàn tệ lại ngang nhiên đóng rầm, chính thức khai ngòi cho thế chiến thứ ba bùng nổ.
32	Thay vì được ăn ổi, tôi lấy làm lạ tại sao má tôi lại sắc một chén thuốc làm bằng vỏ lựu và vỏ chanh đắng nghét rồi bắt tôi uống.
33	Tất cả đều ra ngoài trong thời tiết sáng sủa nhẹ nhàng như để tích trữ ánh nắng và không khí dịu mát cho mùa đông khi họ không thể còn có gì.
34	Khó hiểu quá, jun thấy trong người đã bớt đau đớn nhiều, cơ thể cậu thường phục hồi cũng khá nhanh, cộng thêm đã được mun cho thuốc và dùng kháng sinh đặc trị của lũ người này.
35	Những hôm về làng, tối nào tôi cùng ra đứng trước hiên nhìn xuống chợ Đo Đo, tôi thấy hình ảnh bà ẩn hiện trong những chùm đèn lấp lánh.
36	Hét lên trong đầu cùng lũ hồn chiến tạo ra dòng âm thanh giàu nhạc điều, tôi nuốt nước bọn ừng ực, hai tay lại nắm chặt hơn, chống chế cơn run rẩy đang ngự trị.
37	Thi Thi, kỳ thật tôi cũng không phải cực kỳ hiểu rõ tới cùng là chuyện gì xảy ra, nhưng mà Hồng Tinh Huy nói Bảo Bảo trong bụng tôi là của hắn.
38	tôi cầm đèn soi lên, nhìn trân trân vào những dòng chữ khắc gỗ in mộc bản mà tôi đã đọc nhiều lần trên các tờ khẩu hiệu dán đầy ở những nẻo đường tôi đã vượt qua.

39	Cho dù là thế đi chăng nữa thì cũng chỉ là do bị sức ép của địa chủ ác bá bất đắc dĩ phải nghe theo nếu không tại sao cuối cùng bà lại treo cổ tự sát
40	Tôi bắt đầu phải chịu đựng ảnh hưởng của việc không chịu dùng chất thuỷ ngân đỏ của con người, không còn khoẻ mạnh như trước, di chuyển chậm chạp hơn, mắt bị mờ, mũi và tai cũng không còn thính nữa.

A.2 Chi tiết các mẫu câu sử dụng cho đánh giá hệ thống caching

STT	Mẫu câu
SP001	Lớp vẽ tranh Ta Vẽ xin chào {danh_xung} {ho_va_ten}. Đây là cuộc gọi xác nhận tham gia lớp chủ đề lúc {thoi_gian} sáng chủ nhật, tại địa chỉ {dia_chi}, của {danh_xung} . Xin vui lòng lắng nghe và phản hồi bằng cách bấm chọn. Để xác nhận tham gia, bấm phím 1. Để yêu cầu Ta Vẽ gọi lại, bấm phím 2.
SP002	Xin chào {danh_xung} {ho_va_ten}, khoản vay 10 triệu đồng {danh_xung} mượn đã lâu nhưng chưa thấy trả. Yêu cầu {danh_xung} {ho_va_ten} sắp xếp để trả khoản vay này trong 7 ngày tới. Ấn phím 1 để xác nhận sẽ trả khoản vay
SP003	Xin chào {danh_xung} {ho_va_ten}, vui lòng giúp chúng tôi thực hiện cuộc khảo sát nho nhỏ sau đây. Câu hỏi là {danh_xung} đã bao giờ trải nghiệm các dịch vụ tại Spa chưa ạ? Nhấn phím 1 để chọn Đã từng tham gia, phím 2 để chọn Chưa bao giờ tham gia.
SP004	Chào anh chị, xin hỏi có phải em đang trao đổi với {danh_xung_ho_va_ten} phải không ạ? Nếu đúng vui lòng bấm phím 1, nếu sai vui lòng bấm phím 2.
SP005	Dạ em xin lỗi vì những bức xúc và ghi nhận ý kiến đóng góp của anh chị. Anh chị có thể liên hệ hotline của công ty để phản ánh, bên em sẽ có biện pháp khắc phục nếu nhân viên vi phạm quy định. Tuy nhiên, hợp đồng của anh chị gần đến hạn thanh toán vào ngày {ngay_het_han}, em nghĩ sẽ tốt hơn nếu anh chị thanh toán đúng hạn để tránh ảnh hưởng đến hồ sơ tín dụng của anh chị. anh chị thu xếp thanh toán giúp em được không ạ?

SP006	Dạ hợp đồng tín dụng ký kết giữa anh chị và công ty {cong_ty} là hoàn toàn dựa trên sự tự nguyện và thống nhất thoả thuận. Nhân viên tư vấn của bên em đã tư vấn chi tiết trước khi ký hợp đồng. vì vậy anh chị vui lòng thanh toán đúng hạn giúp em với số tiền trả góp tháng này là {khoan_vay} đồng, vui lòng thanh toán trước ngày {ngay_het_han} được không ạ?
SP007	Dạ hồ sơ vay của bạn {ho_va_ten} cam kết thanh toán vào ngày đến hạn là {ngay_het_han} số tiền thanh toán kỳ này là {khoan_vay}, vì vậy anh chị cố gắng sắp xếp thanh toán giúp em. em cảm ơn.
SP008	Em chào {danh_xung_ho_va_ten}, em là nhân viên công ty {cong_ty}, gọi cho {danh_xung} để xác nhận thông tin về đơn hàng {don_hang} giá tiền {gia_tien} đồng mà {danh_xung} đã đặt tại địa chỉ: ngõ {dia_chi_ngo}, quận {dia_chi_quan_huyen}, thành phố {dia_chi_thanh_pho}. Anh vui lòng bấm phím 1 để xác nhận, bấm phím 2 để từ chối.
SP009	{danh_xung_ho_va_ten} chú ý, lịch khám bệnh của {danh_xung} tuần này là {thoi_gian}, {ngay_thang} tai bệnh viện {benh_vien}. {danh_xung} vui lòng tới đúng thời gian đã hẹn, em cảm ơn.
SP010	Em xin thông báo, {danh_xung_ho_va_ten}, số chứng minh nhân dân {cmnd} hiện nay đã mắc covid F1, {danh_xung_ho_va_ten} hãy đến ngay cơ sở y tế gần nhất để được cách ly kịp thời.