ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN
CS337.O11 - XỬ LÝ ÂM THANH VÀ TIẾNG NÓI
ĐỀ TÀI:

NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT TỰ ĐỘNG VỚI MÔ HÌNH TRANSFORMER

TRANSFORMER-BASED MODEL IN VIETNAMESE AUTOMATIC SPEECH RECOGNITION

Giảng viên hướng dẫn: TRỊNH QUỐC SƠN

Sinh viên thực hiện:
21522628- PHAN VĂN THIỆN
21522509- HOÀNG ANH ĐỨC ĐĂNG QUANG

TP.HỒ CHÍ MINH, 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN
CS337.O11 - XỬ LÝ ÂM THANH VÀ TIẾNG NÓI
ĐỀ TÀI:

NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT TỰ ĐỘNG VỚI MÔ HÌNH TRANSFORMER

TRANSFORMER-BASED MODEL IN VIETNAMESE AUTOMATIC SPEECH RECOGNITION

Giảng viên hướng dẫn: TRỊNH QUỐC SƠN

Sinh viên thực hiện:
21522628- PHAN VĂN THIỆN
21522509- HOÀNG ANH ĐỨC ĐĂNG QUANG

TP.HỒ CHÍ MINH, 2023

LỜI CẢM ƠN

Lời đâu tiên,chúng tôi xin gửi lời cảm ơn chân thành và tri ân sâu sắc đến thầy Trịnh Quốc Sơn- người đã không ngừng truyền đạt kiến thức chuyên môn quý báu và trực tiếp hỗ trợ chúng tôi trong quá trình thực hiện đồ án này. Sự nhiệt huyết và tận tâm của thầy là sự đóng góp và động viên to lớn vào sự thành công của đồ án này.

Xin được bày tỏ lòng biết ơn đến các thành viên trong nhóm vì sự cống hiến và nhiệt tình trong suốt quá trình tìm hiểu và hiện thực hóa đề tài. Không thể không nhắc đến gia đình và bạn bè, những người đã luôn ủng hộ và khuyến khích chúng tôi trong mọi bước tiến của hành trình học tập.

Vì kiến thức bản thân và kinh nghiệm còn hạn chế nên không thể tránh khỏi những sai sót, chúng tôi rất mong nhận được lời nhận xét và góp ý của thầy để đề tài có thể phát triển hơn nữa.

Chúng tôi xin trân trọng cảm ơn.

Thành phố Hồ Chí Minh, 12/2023

Sinh viên

Phan Văn Thiện Hoàng Anh Đức Đăng Quang

MỤC LỤC

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH ẢNH VÀ BIỂU ĐỒ

DANH MỤC TỪ VIẾT TẮT

NST Noisy Student Training

ASR Nhận dạng tiếng nói tự động

CTC Connectionist Temporal Classification

pre-train tiền huấn luyện

self-training tự huấn luyện

speech-to-text Chuyển đổi tiếng nói thành văn bản

finetune tinh chỉnh

WER Word Error Rate (Tỉ lệ lỗi từ)

prefix search decoding Bộ giải mã tìm kiếm tiền tố

product quantization lượng tử hóa tích

temperature giá trị nhiệt độ

contrastive loss hàm mất mát tương phản

diversity loss hàm mất mát đa dạng

contrastive task tác vụ tương phản

HMM Hidden Markov Model

GMM Gaussian Mixture Model

RNN Recurrent neural network

CRF Conditional Random Field

"blank" nhãn rỗng

VUS Voiced-Unvoiced-Slience

LER Label Error Rate

WER Word Error Rate

TÓM TẮT ĐỒ ÁN

Đồ án thực hiện cài đặt mô hình học tự giám sát Wav2vec2.0 trên khung huấn luyện bán giám sát Noisy Student Traning với bộ dữ liệu 100h VLSP2020-VinAI. Kết quả mô hình được đánh giá dựa trên tỉ lệ lỗi WER(Word Error Rate). Đồ án nhằm mục đích:

1. Tìm hiểu kiến thức về các phương pháp dựa trên Transformer mà cụ thể ở đây là Wav2vec2.0 để giải quyết bài toán Nhận diện tiếng nói tự động.

2. Cài đặt và so sánh mô hình Wav2vec2.0 trên huấn luyện cơ bản và huấn luyện bán giám sát.

Kết quả thực hiện đạt kết quả cuối cùng là 8.29% trên tập VIVOS và 9.048 % trên tập Common Voice Vi. Xu hướng mô hình vẫn còn có thể cải thiện nếu được huấn luyện thêm và đạt kết quả đúng với kỳ vọng: mô hình student đạt kết quả tốt dần theo các thế hệ về sau.

Chương 1. TỔNG QUAN

1.1 Đặt vấn đề

Trong lĩnh vực trí tuệ nhân tạo nói chung và xử lý âm thanh nói riêng, bài toán nhận diện tiếng nói tự động đang trở thành một lĩnh vực nghiên cứu và ứng dụng ngày càng quan trọng và được chú ý nhiều hơn trong thời gian gần đây. Sự phát triển nhanh chóng của trí tuệ nhân tạo và các công nghệ liên quan mở ra nhiều phương pháp và hướng đi cho việc vải thiện khả năng nhận diện tiếng nói. Từ đó, các ứng dụng của nhận dạng tiếng nói cũng trở nên đa dạng hơn bao giờ hết, đặc biệt là trong lĩnh vực giao tiếp người máy, cụ thể hơn là các trợ lý ảo như Siri của Apple, Google Assistant của Google hay Alexa của Amazon,...Mặc dù khả năng hiểu ngôn ngữ của các mô hình rất mạnh mẽ trên tiếng Anh nhưng khả năng hiểu tiếng Việt lại vẫn còn rất nhiều hạn chế. Từ đó đặt ra nhu cầu về một mô hình có khả năng hiểu được tiếng Việt có thể tùy biến và cải tiến

Xu hướng hiện nay của bài toán nhận dạng tiếng nói tự động tập trung chủ yếu vào việc sử dụng các mô hình học sâu để đạt được độ chính xác đáng kinh ngạc. Đặc biệt, sự ra đời của kiến trúc Transformer đã đặt nền tảng cho rất nhiều bước tiến lớn không chỉ trong lĩnh vực Xử lý ngôn ngữ tự nhiên mà còn cả những lĩnh vực khác, trong đó bao gồm cả Xử lý âm thanh. Tuy nhiên trong ngôn ngữ tiếng Việt, bài toán nhận dạng tiếng nói vẫn còn gặp khá nhiều khó khăn, đặc biệt là vấn đề thiếu dữ liệu được gán nhãn. Vì những lý do đó mà nhóm lựa chọn đề tài là Nhận diện tiếng nói tự động với kiến trúc Transformer.

1.2 Mục tiêu đồ án

- Tìm hiểu mô hình tiền huấn luyện Wav2Vec2.0 [1]
- Tìm hiểu kỹ thuật học bán giám sát Noisy Student Training[2]
- Tìm hiểu Beam Search[3] và Connectionist Temporal Classification [4] cho decoder.
- Áp dụng Noisy Student Training để huấn luyện Wav2Vec2.0

1.3 Phạm vi nghiên cứu

- Kiến thức hiểu biết về tiền huấn luyện,tự huấn luyện.
- Sử dụng bộ dữ liệu có gán nhãn từ VLSP2020

1.4 Ý nghĩa thực tiễn

Nghiên cứu góp phần cải thiện mô hình nhận dạng tiếng nói trong ngôn ngữ tiếng Việt bằng cách sử dụng kỹ thuật học bán giám sát.Từ đó giúp thu được kết quả tốt hơn với lượng dữ liệu có gán nhãn hạn chế.

Chương 2. Cơ sở lý thuyết

2.1 Bài toán nhận diện tiếng nói tự động

Nhận dạng tiếng nói tự động (ASR) hay còn biết tới với những cái tên khác như Chuyển đổi tiếng nói thành văn bản (speech-to-text) là tác vụ chuyển đổi âm thanh thành văn bản. Có rất nhiều ứng dụng của bài toán này, một trong số đó có thể kể đến như trợ lý ảo, tạo phụ đề tự động,...

Một hệ thống nhận diện tiếng nói tự động sẽ nhận đầu vào là một đoạn tín hiệu âm thanh và sau đó chuyển đổi nó thành văn bản hoặc biểu diễn ngôn ngữ tương ứng. Quá trình này gồm nhiều bước như: thu thập và xử lý dữ liệu âm thanh, trích xuất đặc trưng và chuyển đổi thành văn bản.

2.2 Các nghiên cứu liên quan

Những ý tưởng về một hệ thống có thể hiểu và nhận diện giọng nói đã bắt đầu từ rất sớm, từ những năm 1950, bài toán nhận dạng tiếng nói tự động đã được đề cập đến trong một bài báo mang tên “Automatic Recognition of Spoken Language” [5] của hai nhà nghiên cứu Davis và Biddulph được đăng trên tạp chí "Proceedings of the IRE" (Institute of Radio Engineers). Các phương pháp được đề cập ở bài báo này còn khá sơ khai nhưng nó đã đặt nền tảng cho sự phát triển của các phương pháp phức tạp khác trong tương lai.Từ những năm 70 80, các phương pháp thống kê như Hidden Markov Model (HMM), Gaussian Mixture Model (GMM) được áp dụng trong Nhận dạng tiếng nói để mô hình hóa chuỗi thời gian của các trạng thái Voiced-Unvoiced-Slience (VUS) và đặc trưng thống kê của các tín hiệu âm thanh. Những năm 2000, bài toán nhận diện tiếng nói đạt được nhiều bước tiến lớn nhờ sự ra đời lần lượt của các kỹ thuật và mô hình như CTC, DNN, Transformer,...

Xu hướng của những năm gần đây đang tập trung đến các mô hình tiền huấn hay mô hình học tự giám sát. Học tự giám sát là một dạng học máy trong đó mô hình học từ dữ liệu mà không yêu cầu các nhãn (labels) đúng cho từng điểm dữ liệu đầu vào. Thay vào đó mô hình sẽ học biểu diễn dữ liệu từ các dữ liệu không được gán nhãn và sau đó tinh chỉnh (finetune) trên một tập dữ liệu được gán nhãn. Kỹ thuật này thường được áp dụng nhằm tận dụng nguồn dữ liệu không được gán nhãn dồi dào mà vẫn đem lại hiệu quả lớn. Một trong số các mô hình học tự giám sát nổi bật thời gian gần đây có thể kể đến wav2vec 2.0. Để tận dụng tốt nguồn dữ liệu không gán nhãn cũng như để khai thác tốt các mô hình học tự giám sát, các kỹ thuật học bán giám sát cũng rất được quan tâm và liên tục có nhiều

công trình nghiên cứu giúp thúc đẩy giới hạn của mô hình.

2.3 Hàm mục tiêu Connectionist Temporal Classification

2.3.1 Tổng quan

Bài toán đánh nhãn dữ liệu chuỗi mà không được phân đoạn sẵn là một tác vụ phổ biến trong việc mô hình hóa các chuỗi trong thế giới thực. Các bài toán này thường là các tác vụ liên quan đến nhận diện (Ví dụ: nhận diện giọng nói, nhận diện cử chỉ,..), mà đầu vào ở đây thường có các nhiễu, giá trị thực từ luồng đầu vào thường được ký hiệu bởi chuỗi các ký hiệu rời rạc như là các ký tự hay từ.

Năm 2006, lúc A.Graves cùng cộng sự viết bài về Connectionist Temporal Classification (CTC)[4] với các mô hình dạng đồ thị như HMM, Conditional Random Field (CRF),.. Cách tiếp cận này đã chứng minh được hiệu quả của mình thông qua sự thành công ở nhiều bài toán, tuy nhiên một số điểm hạn chế:

1. Yêu cầu lượng kiến thức về tác vụ cần giải quyết nhằm thiết kế trạng thái hoặc chọn đặc trưng đầu vào cho mô hình.

2. Đòi hỏi các giả định phụ thuộc phải tường minh để việc suy luận kết quả trở nên dễ hiểu hơn.

3. Với HMM tiêu chuẩn, quá trình huấn luyện mang tính sinh dù việc gán nhãn chuỗi là quá trình phân loại.

Các mô hình Recurrent neural network (RNN) có thể huấn luyện phân loại và cung cấp cơ chế đặc biệt giúp mô hình hóa chuỗi thời gian. Mô hình mạnh mẽ khi đối mặt với nhiễu cả về không gian lẫn thời gian. Tuy nhiên để áp dụng RNN một cách trực tiếp vào gãn nhãn chuỗi là điều không thể vì hàm mục tiêu của mạng nơ-tron nhân tạo được định nghĩa riêng cho mỗi điểm dữ liệu. Nói cách khác, RNN chỉ được dùng để huấn luyện cho việc tạo ra nhãn chuỗi độc lập với nhau và dữ liệu cần được phân đoạn từ trước.

Tính đến thời điểm 2006, hệ thống kết hợp RNN và HMM để mô hình hóa cấu trúc tuần tự tầm xa của dữ liệu, mạng nơ-ron giúp phân loại cục bộ. Thành phần HMM có khả năng tự động phân đoạn trong quá trình huấn luyện. Tuy nhiên hệ thống này vẫn chưa khai thác được kết tiềm năng của RNN trong bài toán mô hình hóa chuỗi tuần tự.

A.Graves cùng cộng sự vào năm 2006 đã tổng hợp các vấn đề tồn đọng trên và trình bày một nghiên cứu về CTC. Theo như tác giả, đây là một phương pháp mới dành cho dữ liệu

dạng chuỗi tuần tự, nếu áp dụng cho RNN thì sẽ không cần phải phân đoạn dữ liệu đầu vào và xử lý sau khi có đầu ra nữa, mô hình sẽ tự mô hình hóa toàn bộ chuỗi tuần tự trong một kiến trúc mạng duy nhất. Ý tưởng đơn giản là thông dịch đầu ra của mạng nơ-ron như là một phân phối xác suất qua toàn bộ các chuỗi nhãn có thể xảy ra. Đưa trước phân phối này, ta có thể đưa ra một hàm mục tiêu nhằm tối đa hóa xác suất những chuỗi có nhãn đúng. Bởi vì hàm mục tiêu này có thể đạo hàm được, nên mạng nơ-ron có thể được huấn luyện bằng lan truyền ngược theo thời gian.

2.3.2 Kỹ thuật

2.3.2.1 Bài toán Temporal Classification

Gọi S là tập dữ liệu được lấy từ phân phối cố định $\mathcal{D}_{\mathcal{X}\times\mathcal{Z}}$. Không gian đầu vào $\mathcal{X} = (\mathbb{R}^m)*$ là tập tất cả các chuỗi véc-tơ số thực có m chiều. Không gian mục tiêu $\mathcal{Z} = L*$ là tập tất cả các chuỗi tuần tự qua một bảng chữ cái L. Một cách tổng quát, ta đề cập mỗi phần từ của $L*$ là một nhãn. Mỗi mẫu trong S chứa một cặp chuỗi $(\mathbf{x}, \mathbf{y})$. Chuỗi mục tiêu $\mathbf{z} = (z_1, .., z_U)$ có độ dài nhất bằng với độ dài của chuỗi $\mathbf{x} = (x_1, x_2, .., x_T)$ có nghĩa là $U \leq T$. Vì chuỗi đầu vào và chuỗi mục tiêu không có cùng độ dài, không có cách nào tiên nghiệm để căn chỉnh 2 chuỗi này. Mục tiêu là dùng S để huấn luyện một mô hình phân loại theo thời gian h: $\mathcal{X} \mapsto \mathcal{Z}$ để phân loại đầu vào chưa nhìn thấy trước đây theo mục tiêu là giảm thiểu một thang đo độ lỗi cụ thể.

Label Error Rate (LER): Trong bài toán Temporal Classification, với một tập $S' \subset \mathcal{D}_{\mathcal{X}\times\mathcal{Z}}$ khác S, định nghĩa LER của mô hình phân loại theo thời gian h là khoảng cách để chỉnh sửa một chuỗi nhãn được phân loại với mục tiêu trên S':

$$LER(h, S') = \frac{1}{Z} \sum_{((\mathbf{x},\mathbf{z})\in S'} ED(h(\mathbf{X})) \tag{2.1}$$

Mà Z là tổng số lượng nhãn của mục tiêu thuộc S' và $ED(\mathbf{p}, \mathbf{q})$ là khoảng cách chỉnh sửa giữa hai chuỗi $\mathbf{p}$ và $\mathbf{q}$- số lượng thêm, xóa, sửa giữa 2 chuỗi. Trong báo cáo đồ án này, LER sẽ là Word Error Rate (WER).

2.3.3 Connectionist Temproral Classification

Một mạng CTC có môt lớp đầu ra softmax với nhiều hơn một nhãn so với L. Các giá trị của $|L|$ đơn vị đầu tiên có thể được thông dịch như xác suất để quan sát được nhãn vị trí tương ứng tại một thời điểm cụ thể. Giá trị kích hoạt của đơn vị cộng thêm là xác suất quan sát

một nhãn rỗng ("blank") hoặc có thể xem là không có nhãn. Với bộ phân phối xác suất này, đầu ra có thể tạo ra tất cả các cách căn chỉnh khác nhau giữa chuỗi nhãn đầu ra và chuỗi đầu vào. Tổng xác suất của bất kỳ một chuỗi nhãn dàn nào đó đều có thể tính được bằng tổng tất cả xác suất của các cách căn chỉnh của nó. Cụ thể hơn, với mỗi chuỗi đầu vào $\mathbf{x}$ có độ dài T, định nghĩa một RNN với m đầu vào , n đầu ra và vector trọng số w như một hàm ánh xạ liên tục $\mathcal{N}_w : (\mathbb{R}^m)^T \mapsto (\mathbb{R}^n)^T$. Gọi $\mathbf{y} = \mathcal{N}_w(\mathbf{x})$ là chuỗi đầu ra của mạng và ký hiệu y_k^t là giá trị kích hoạt của đầu ra k tại thời điểm t hay xác suất quan sát được của nhãn k tại thời điểm t. Ta có định nghĩa một phân phối qua tập L'^T có độ dài chuỗi là T qua bảng chữ cái $L' = L \cup \{blank\}$:

$$p(\pi|\mathbf{x}) = \Pi_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L'^T \tag{2.2}$$

Trong đó π là một phần tử của L'^T, gọi là đường dẫn(path). Công thức trên ngầm giả định rằng đầu ra của mạng tại thời điểm khác nhau là độc lập có điều kiện. Điều này đảm bảo không tồn tại kết nối từ lớp đầu ra của mạng tới chính nó. Định nghĩa một hàm ánh xạ nhiều-một $\mathcal{B} : L'^T \mapsto L^{leT}$ là một tập các cách để gán nhãn. Việc tìm các chuỗi gán nhãn này đơn giản là loại bỏ tất cả ký tự trống và hợp những nhãn trùng thành một. Nhờ hàm ánh xạ, mạng CTC có thể xuất ra nhiều cách căn chỉnh khác nhau. Ta có thể dùng hàm $\mathcal{B}$ để định nghĩa xác suất có điều kiện khi biết trước các chuỗi gán nhãn $\mathbf{I} \in L^{\leq T}$ là tổng xác suất các đường dẫn tương ứng với nó:

$$p(\mathbf{I}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1} p(\pi|\frown)} \tag{2.3}$$

Sau khi đã có những thông tin trên, đầu ra của bộ phân loại h là chuỗi có xác suất cao nhất cho chuỗi đầu vào:

$$h(\mathbf{x}) = argmax_{I \in L^{\leq T}} p(\mathbf{I}|\mathbf{x} \tag{2.4}$$

Lúc này ta có thể giả định chuỗi tốt nhất là sự kết hợp của các nhãn có giá trị xác suất cao nhất tại mỗi thời điểm:

$$h(x) \approx \mathcal{B}(\pi*) \tag{2.5}$$

$$\text{mà } \pi* = argmax_{\pi \in N^t p(\pi|\mathbf{x}} \tag{2.6}$$

Tất nhiên, giả định này chưa chắc chính xác. Một cách khác để lấy chuỗi tối ưu hơn là dùng Bộ giải mã tìm kiếm tiền tố (prefix search decoding), ví dụ như Beam Search[3]. Trong báo cáo đồ án này, nhóm chúng tôi sử dụng Beam Search [3] để tìm kiếm chuỗi âm vị của câu nói

2.4 Wav2Vec 2.0

2.4.1 Tổng quan

Mạng nơ-ron dù mang lại hiệu suất cao nhưng yêu cầu một lượng lớn dữ liệu có gán nhãn để huấn luyện. Trong nhiều tình huống mà việc thu thập dữ liệu có gán nhãn khó hơn nhiều so với dữ liệu được gán nhãn: những hệ thống nhận diện tiếng nói gần đây yêu cầu hàng ngàn giờ ghi âm để đạt được hiệu suất có thể chấp nhận được, việc này không phải điều dễ dàng với tất cả các ngôn ngữ. Việc học từ các biểu diễn âm thanh thông qua dữ liệu không gán nhãn cũng như việc trẻ sơ sinh học cách hiểu người lớn xung quanh chúng nói chuyện.

Trong máy học, học bán giám sát cũng đạt được nhiều thành công lớn trong xử lý ngôn ngữ tự nhiên và là một lĩnh vực nghiên cứu tích cực trong xử lý ảnh.

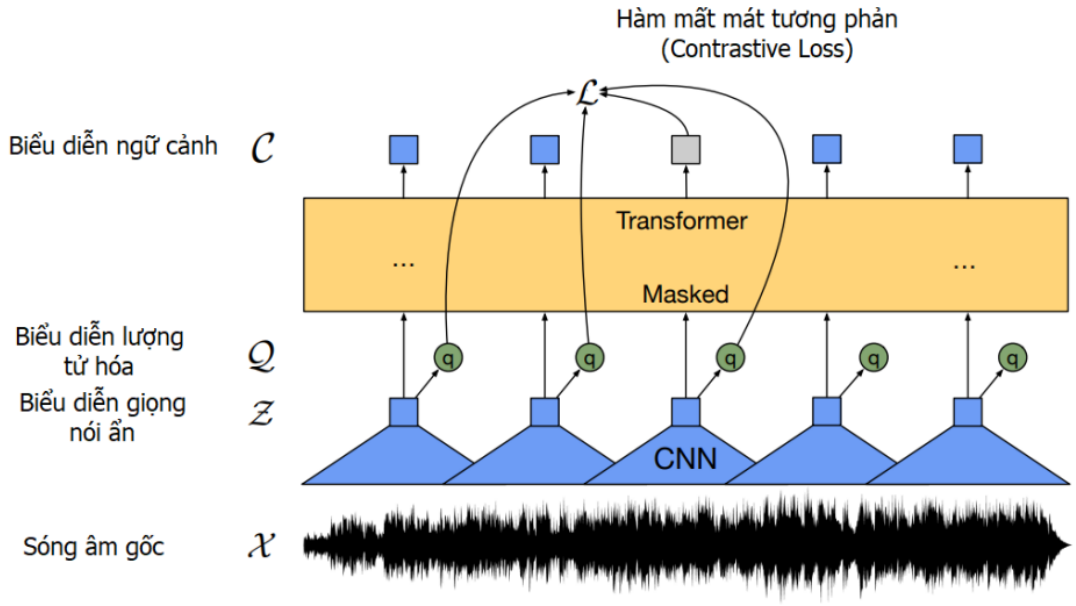
Trong bài báo của mình, nhóm tác giả giới thiệu một framework giúp biểu diễn âm thanh từ dữ liệu âm thanh gốc thông qua một mạng tích chập đa lớp và sau đó che đi một phần kết quả biểu diễn ngầm như cách Masked Language Model hoạt động. Những biểu diễn sẽ được đưa tới một mạng Transformer để xây dựng biểu diễn ngữ cảnh và sau đó mô hình sẽ được huấn luyện thông qua tác vụ tương phản (contrastive task). Sau khi tiền huấn luyện, mô hình sẽ được tinh chỉnh với dữ liệu được gán nhãn với hàm mục tiêu CTC.

Kết quả tốt nhất đạt được của nhóm tác giả với 960h dữ liệu có gán nhãn Librispeech lên tới 1.8/3.3 % WER

2.4.2 Mô hình

2.4.2.1 Bộ mã hóa đặc trưng

Bộ mã hóa bao gồm các khối chứa temporal convolution theo sau là lớp Normalization và một hàm kích hoạt GELU. Đầu vào dạng sóng thô với bộ mã hóa được chuẩn hóa thành giá trị trung bình bằng 0 và phương sai đơn vị. Tổng bước trượt (stride) của bộ mã hóa quyết định số time-step T được đưa vào Transformer[6].



Hình 2.1. Cách thức học của Mô hình Wav2vec2.0

2.4.2.2 Biểu diễn ngữ cảnh với Transformer

Dữ liệu đầu ra của bộ mã hóa đặc trưng sẽ được đưa vào mạng ngữ cảnh theo kiến trúc Transformer. Thay vì sử dụng bộ nhúng vị trí tuyệt đối như kiến trúc gốc, Wav2Vec2 sử dụng một lớp tích chập tương tự để đóng vai trò như một bộ nhúng vị trí tương đối. Đầu ra của lớp tích chập được theo sau bởi một hàm kích hoạt GELU và được áp dụng một lớp Normalization.

2.4.2.3 Mô-đun Lượng tử hóa (Quantization Modules)

Để học tự huấn luyện (self-training), nhóm tác giả đã lượng tử hóa đầu ra của bộ mã hóa đặc trưng z thành tập hữu hạn các biểu diễn âm thanh thông qua lượng tử hóa tích (product quantization). Số lượng product quantization để chọn các biểu diễn được lượng tử hóa từ nhiều codebooks và ghép lại với nhau. Với G codebook hoặc nhóm , V mục $e \in \mathbb{R}^{V \times d/G}$, ta chọn một mục thuộc mỗi codebooks và ghép các véc-tơ kết quả $e_1, \ldots, e_G$ và áp dụng phép biến đổi tuyến tính $\mathbb{R}^d \mapsto \mathbb{R}^f$ để ra được kết quả $q \in \mathbb{R}^f$.

Gumbel softmax cho phép lựa chọn các entry rời rạc theo cách hoàn toàn khác biệt. Ta sử dụng một phép tính xuyên suốt và cài đặt phép tính Gumbel Softmax G. Đầu ra **z** của bộ mã hóa đặc trưng được ánh xạ tới các logits $I \in \mathbb{R}^{G \times V}$ và xác suất để chọn codebook

entry thứ v cho nhóm g là:

$$p_{g,v} = \frac{\exp\left(l_{g,v} + n_v\right)/\tau}{\sum_{k=1}^{V} \exp\left(l_{g,k} + n_k\right)/\tau} \tag{2.7}$$

với τ là một giá trị nhiệt độ (temperature) không âm, $n = -\log(-\log(u))$ và u là các mẫu đồng nhất từ phân phối đều $U(0, 1)$. Trong quá trình lan truyền xuôi, mục thứ i được chọn khi $i = argmax_j p_{g,j}$. Giá trị đạo hàm thực của Gumbel softmax được sử dụng trong quá trình đạo hàm cho lan truyền ngược

2.4.3 Quá trình huấn luyện

Để dùng mô hình này cho quá trình huấn luyện, ta cần che đi một phần tỉ lệ các time-step nhất định của bộ mã hóa đặc trưng, tương tự như mask language modeling của BERT. Mục tiêu huấn luyện sẽ yêu cầu xác định chính xác vector lượng tử hóa ẩn cho biểu diễn âm thanh trong mỗi một tập các bộ phân tâm cho mỗi time-step bị che. Mô hình sau khi tiền huấn luyện (pre-train) có thể được finetune trên bộ dữ liệu có gán nhãn.

2.4.3.1 Ẩn dữ liệu (Masking)

Để huấn luyện, mô hình sẽ phải dùng cơ chế mặt nạ che đi một tỷ lệ đầu ra của bộ mã hóa đặc trưng, hoặc nói cách khác là che đi các time-step trước khi đưa nó vào mạng học ngữ cảnh (Transformer) và thay thế giá trị ở các time-step này bằng cách véc-tơ đặc trưng được học và chia sẻ véc-tơ này cho tất cả các time-step bị che, nhưng bộ mã hóa đặc trưng sẽ không bị che khi đi qua mô-đun lượng tử hóa. Để che đầu ra của bộ mã hóa đặc trưng, mô hình sẽ ngẫu nhiên một tỷ lệ p từ tất cả các time-step để làm vị trí đầu tiên, và sau đó dùng cơ chế mặt nạ để che liên tục M time-step tiếp theo từ vị trí được chọn đó, việc này có thể chồng lên nhau.

2.4.3.2 Hàm mục tiêu

Trong quá trình pre-train, mô hình học biểu diễn âm thanh bằng cách giải quyết tác vụ tương phản $\mathcal{L}_m$ - tác vụ yêu cầu xác định chính xác tầng biểu diễn lượng tử ẩn cho mỗi time-step bị che lại trong một tập các bộ phân tâm. Hàm mục tiêu sẽ được tăng cường bởi contrastive loss các codebook $\mathcal{L}_d$ để khuyến khích mô hình sử dụng các mục trong codebook đều nhau.

$$\mathcal{L} = \mathcal{L}_m + \alpha\mathcal{L}_d \tag{2.8}$$

α là tham số có thể điều chỉnh được. Ở hàm mục tiêu số (2), ta thấy có hai thành phần là hàm mất mát tương phản (contrastive loss) và hàm mất mát đa dạng (diversity loss).

Constrastive Loss: Với một đầu ra của mạng ngữ cảnh c_t tại time-step bị ẩn t, mô hình cần nhận biết chính xác biểu diễn âm thanh ẩn được lượng tử hóa $\mathbf{q}_t$ trong tập $K+1$ các ứng viên $\tilde{q} \in \mathbf{Q}_t$ bao gồm $\mathbf{q}_t$ và K bộ phân tâm. Các bộ phân tâm được lấy mẫu từ các time-step bị ẩn khác của cùng một câu nói. Hàm mất mát định nghĩa như sau:

$$\mathcal{L}_m = -\log \frac{\exp(sim(c_t, \mathbf{q}_t/K))}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(c_t, \tilde{\mathbf{q}}/K))} \tag{2.9}$$

mà $sim(a,b) = a^T b/||a||||b||$ là thang đo độ tương đồng cosine giữa biểu diễn ngữ cảnh và biểu diên lượng tử hóa ẩn.

Diveristy Loss: Tác vụ tương phản phụ thuộc vào các codebooks để biểu diễn các mẫu dương (positive), âm(negative) và hàm mất mát đa dạng $\mathcal{L}_d$ này được thiết kế để tăng việc sử dụng các mẫu codebook đã được lượng tử hóa. Điều này giúp mô hình được khuyến khích sử dụng các mục V một cách đồng đều bằng cách tối đa hóa giá trị entropy của phân phối trung bình softmax I thông qua các mục cho từng codebook $\hat{p}_g$ qua mỗi batch; phân phối softmax không bao gồm các nhiễu gumbel cũng như tham số temperature :

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\hat{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \hat{p}_{g,v} \log \hat{p}_{g,v} \tag{2.10}$$

2.4.3.3 Điều chỉnh (Finetune)

Mô hình được điều chỉnh cho nhận diện giọng nói bằng cách thêm một phép chiếu tuyến tính được khởi tạo ngẫu nhiên ở đầu mạng ngữ cảnh lên C lớp để đại diện các từ vựng của tác vụ. Với Libriseech, ta có 29 tokens cho các mục tiêu ký tự cộng với 1 token ranh giới từ. Mô hình được tối ưu bằng cách tối thiểu hóa một hàm mất mát CTC và ta áp dụng phiên bản được finetune của SpecAugment bằng cách che đi các tim-step và kênh âm thanh trong suốt quá trình huấn luyện nhằm trì hoãn tình trạng quá khớp và cải thiện đáng kể tỉ lệ lỗi cuối cùng.

2.5 Học bán giám sát với Noisy Student Training

2.5.1 Học bán giám sát

Semi-supervised learning hay Học bán giám sát là một thể loại đặc biệt của giám sát yếu mà ý tưởng chủ đạo là kết hợp một lượng nhỏ dữ liệu đã được đánh nhãn với một lượng lớn dữ liệu không có nhãn trong quá trình huấn luyện. Học bán giám sát nằm giữa học không giám sát (unsupervised learning – với việc huấn luyện mà dữ liệu không cần nhãn) và học giám sát (supervised learning – với việc huấn luyện mà dữ liệu toàn bộ đều có nhãn). Một lượng lớn dữ liệu không có nhãn khi sử dụng kết hợp với dữ liệu một lượng nhỏ dữ liệu có nhãn có thể tạo ra một sự cải tiến đáng kể trong độ chính xác của việc huấn luyện mô hình. Việc thu thập dữ liệu có nhãn cho quá trình huấn luyện đòi hỏi kỹ năng của con người (ví dụ mô tả bản dịch của một đoạn âm thanh giọng nói, xác định cấu trúc 3D của một protein,...). Chi phí liên quan tới quá trình gắn nhãn có thể khiến việc gắn nhãn một tập dữ liệu lớn có thể không khả thi, bởi vì quá trình gắn nhãn này rất tốn kém. Trong những trường hợp như thế, học bán giám sát có thể có giá trị thực tế lớn. Học bán giám sát cũng được quan tâm về mặt lý thuyết trong Machine Learning vì cách học của nó dựa trên con người.

Cho biết một tập n các mẫu $x_1, .., x_n \in \mathbf{X}$ với nhãn tương ứng $y_1, ..., y_n \in \mathbf{Y}$ và u mẫu dữ liệu không có nhãn $x_{n+1}, ..., x_{n+u} \in \mathbf{X}$ đã được xử lý (cách xử lý tùy theo bài toán). Học bán giám sát kết hợp các thông tin này để vượt qua khả năng dự đoán của mô hình khi chỉ học giám sát (bỏ dữ liệu không nhãn) hoặc mô hình khi chỉ học không giám sát (bỏ dữ liệu có nhãn)

2.5.2 Tự huấn luyện sử dụng Noisy Student Training

Self-training (tự huấn luyện) hay được biết tới với các tên self-labeling (tự đánh nhãn) hay decision-directed leraning (huấn luyện định hướng quyết định), là một trong những cách tiếp cận sớm nhất trong học bán giám sát, nhưng lại phát triển khá phổ biến những năm gần đây. Thuật toán tự huấn luyện bắt đầu với việc huấn luyện một mô hình giám sát trên tập dữ liệu có nhãnS. Sau đó, với mỗi lần lặp, mô hình hiện tại tại lựa chọn một phần dữ liệu không có nhãn X_u và gán nhãn giả (pseudo-label) bằng dự đoán của mô hình này. Bộ dữ liệu với nhãn giả này sau đó được dùng để huấn luyện bộ phân lớp mới cùng với bộ dữ liệu có nhãn cũ, tức là huấn luyện trên $S \cup X_u$ Nói tổng quát hơn, trong khuôn khổ tự huấn luyện lặp đi lặp lại (iterative self- training), một loạt các mô hình được huấn luyện mà mô hình trước sẽ là teacher của mô hình sau, bằng cách teacher sẽ sinh nhãn giả cho mô hình student

học. Noisy Student Training (NST)[2] là một phương pháp được trình bày lần đầu tiên năm 2020 trong lĩnh vực Thị giác máy tính để giải quyết bài toán phân loại trên ImageNet. NST cải thiện self-training và chắt lọc mô hình theo hai cách: thứ nhất, việc cài đặt thực nghiệm NST sẽ bao gồm tăng kích thước mô hình student bằng hoặc lớn hơn teacher, vì thế mô hình student sẽ được lợi hơn khi học từ một lượng dữ liệu (không gán nhãn) lớn hơn, nhưng chỉ đơn thuần tăng kích thước sẽ không đạt hiệu quả mà phải sử dụng kết hợp với tự huấn luyện. Thứ hai, NST cũng bao gồm việc thêm nhiễu vào student để ép student học khó hơn từ nhãn giả. Để làm nhiễu đầu vào, có nhiều cách như dropout, Stochastic Depth hay các phương pháp tăng cường dữ liệu.

Trong bài nghiên cứu, nhóm tác giả đã trình bày cách ứng dụng NST cho bài toán nhận dạng giọng nói, bằng cách giới thiệu phiên bản cải tiến của phương pháp tăng cường dữ liệu SpecAugment là Adaptive SpecAugment[7] để ứng dụng vào bước làm nhiễu dữ liệu cho mô hình student, kết quả tốt nhất của họ đạt WER 1.6%/3.4% trên dev-clean/-other và 1.7%/3.4% trên test-clean/-other của LibriSpeech. Tuy nhiên, kết quả này vẫn có thể cải thiện, nhóm tác giả của nghiên cứu [22] đã trình bày một phương pháp kết hợp tiền huấn luyện với tự huấn luyện. Sử dụng Wav2Vec2.0 để tiền huấn luyện Conformer, sau đó thay vì khởi tạo ngẫu nhiên student hoặc teacher thì sử dụng trọng số đã được tiền huấn luyện để đi huấn luyện tiếp theo khuôn khổ NST. Kết quả hiện tại đang là SOTA của thế giới với WER trên dev-clean/-other là 1.3%/2.6% và trên test-clean/-other là 1.4%/2.6%. Bài nghiên cứu của nhóm chúng tôi dựa vào ý tưởng của nghiên cứu này để thực hiện.

2.5.3 Áp dụng Noisy Student Training vào bài toán

Theo luồng NST ở bài báo gốc [8], gọi tập dữ liệu có nhãn là S, dữ liệu không nhãn là U, quy trình từ nghiên cứu của tác giả sẽ bao gồm:

1. Điều chỉnh mô hình trên dữ liệu huấn luyện M_0 trên S với Specaugment. Gán $M = M_0$.
2. Gán M với mô hình ngôn ngữ và đo hiệu năng của mô hình.
3. Tạo nhãn giả M(U) với mô hình đã được gán mô hình ngôn ngữ.
4. Trộn M(U) và S.
5. Điều chỉnh mô hình tiền huấn luyện M’ với Specaugment trên dữ liệu đã trộn lẫn
6. Gán M = M’ và quay lại bước 2

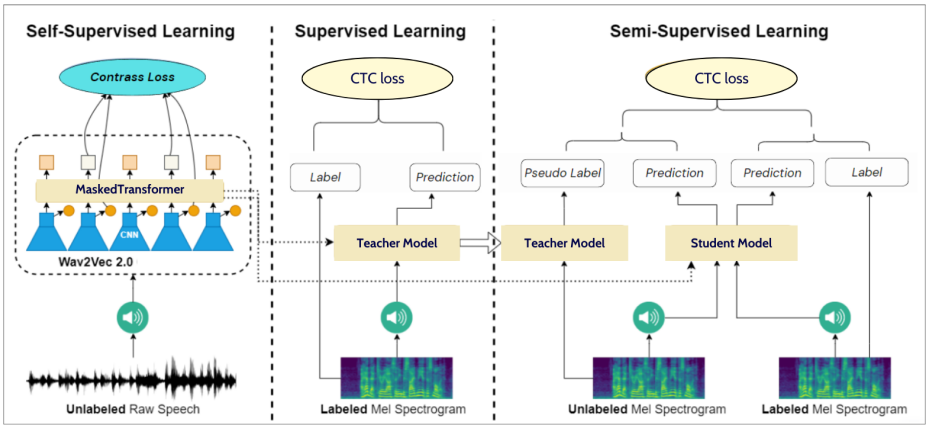
Tuy nhiên trong báo cáo của nhóm có một số điểm khác biệt : Mô hình của tác giả sử dụng Conformer encoder và RNN Transducer decoder- có lợi trong trường hợp tài nguyên huấn luyện lớn. Còn trong báo cáo này nhóm vẫn sử dụng kiến trúc Masked Transformer với hàm mất mát CTC vì tính đơn giản và hiệu quả trong trường hợp tài nguyên huấn luyện hạn chế.

2.6 Kết hợp các kỹ thuật để hiện thực hóa bài toán

Tổng quan lại, ý tưởng của báo cáo như sau:

1. Tiền huấn luyện mô hình Wav2Vec2.0 trên 13k giờ youtube audio
2. Huấn luyện tinh chỉnh mô hình Teacher với dữ liệu có nhãn và sau đó dùng mô hình teacher để tạo ra nhãn giả trên tập dữ liệu không có nhãn để đa dạng hóa nguồn dữ liệu huấn luyện. Từ đó mô hình Student được huấn luyện trên tập dữ liệu lớn hơn.

Hình bên dưới trình bày tương quan lại toàn bộ quay trình bài toán được sử dụng trong báo cáo:



Hình 2.2. . Sơ đồ cấu trúc được sử dụng trong báo cáo. Đường "." mô tả rằng trọng số của teacher và student sẽ được khởi tạo bằng tiền huấn luyện bằng Wav2Vec2.0

2.7 Kỹ thuật đánh giá giọng nói

Để đánh giá hiệu năng nhận diện giọng nói của mô hình, nhóm chúng tôi sử dụng Word Error Rate (WER) được tính bằng công thức:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \tag{2.11}$$

Với :

- S là số lượng từ cần sửa
- D là số lượng từ cần xóa
- I là số lượng từ cần thêm
- C là số lượng từ dự đoán đúng
- N là tổng số từ trong chuỗi thực tế (N = S+D+C)

Giá trị WER càng nhỏ vàng tốt, được tính theo giá trị phần trăm (%). Giá trị tốt nhất là 100% và tệ nhất là 0%.

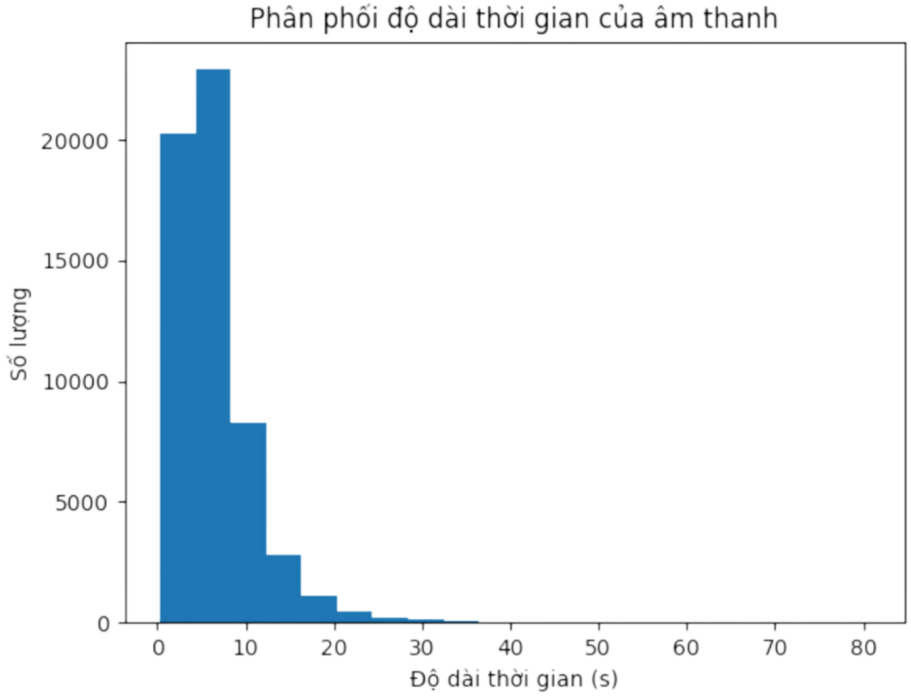
Chương 3. Thực nghiệm và kết quả

3.1 Cài đặt thực nghiệm

3.1.1 Dữ liệu

3.1.1.1 VLSP2020-VINAI 100 giờ dữ liệu

Trong bài báo cáo này nhóm sử dụng 100h của VINAI đóng góp công khai tại hội thảo VLSP2020.Bộ dữ liệu chứa hơn 100 giờ âm thanh đã được làm sạch được thu thập từ các nguồn mở và phiên âm thủ công với độ chính xác 96%. Với tổng số lượng mẫu khoảng 56,400 mẫu, dữ liệu có phân phối theo độ dài âm thanh như sau:



Hình 3.1. Phân phối dựa theo độ dài của âm thanh.

Tuy nhiên để đáp ứng vấn đề thiếu tài nguyên phần cứng trong quá trình huấn luyện. Nhóm đã lược bỏ đi khoảng hơn 2000 mẫu có độ dài lớn hơn 10s.

3.1.1.2 Dữ liệu không gán nhãn

Nhóm thu thập thêm 10h dữ liệu không gán nhãn từ các nguồn mở để huấn luyện cho mô hình student.

3.1.1.3 Dữ liệu đánh giá

Nhóm sử dụng tập test của hai bộ dữ liệu là VIVOS và Commonvoice Vi để tiện so sánh với các mô hình được hiện thực hóa đã có.

3.1.2 Tiền huấn luyện

Mô hình được sử dụng trong báo cáo này được tiền huấn luyện trên tập dữ liệu 13,000 giờ âm thanh youtube tiếng Việt (13k hours of Vietnames youtube audio) trong 30 ngày với TPU-V3-8. Tuy nhiên vì vấn đề thời gian thực hiện đồ án có hạn và phần cứng không cho phép nên nhóm sử dụng lại mô hình đã được huấn luyện sẵn được công bố trên huggingface nguyenvulebinh/wav2vec2-base-vi

3.1.3 Tinh chỉnh mô hình

Mô hình được huấn luyện 5 thế hệ, bao gồm thế hệ 0 được huấn luyện trên bộ dữ liệu gán nhãn VLSP2020-VINAI 100h trên bộ xử lý GPU NVIDIA TESLA P100 với lưu trữ tối đa 15.9GB. Ở các mô hình thế hệ sau, mô hình thế hệ trước sẽ dùng để tạo nhãn giả phục vụ huấn luyện cho thế hệ sau và kế thừa trọng số khởi tạo từ mô hình thế hệ trước.

```
Layer (type:depth-idx)                                  Param #
==========================================================
| Wav2Vec2Model: 1-1                                    --
|      Wav2Vec2FeatureEncoder: 2-1                      --
|    |      ModuleList: 3-1                             4,200,448
|      Wav2Vec2FeatureProjection: 2-2                   --
|    |      LayerNorm: 3-2                              1,024
|    |      Linear: 3-3                                 393,984
|    |      Dropout: 3-4                                --
|      Wav2Vec2Encoder: 2-3                             --
|    |      Wav2Vec2PositionalConvEmbedding:3-5         4,719,488
|    |      LayerNorm: 3-6                              1,536
|    |      Dropout: 3-7                                --
|    |      ModuleList: 3-8                             85,054,464
| Dropout: 1-2                                          --
| Linear: 1-3                                           84,590
```

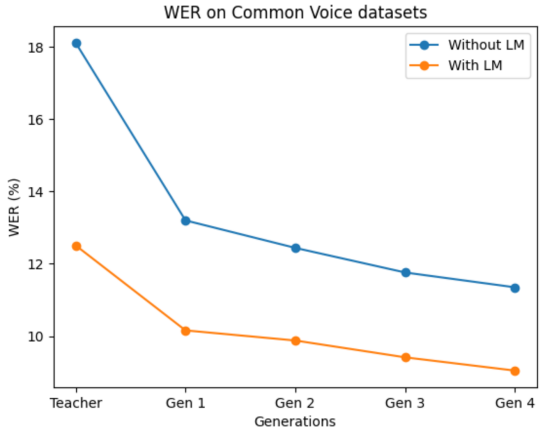
```
================================================================
Total params: 94,455,534
Trainable params: 90,255,086
Non-trainable params: 4,200,448
```

3.1.4 Mô hình ngôn ngữ

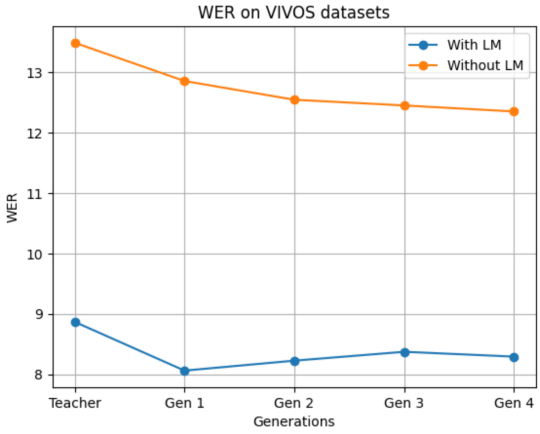
Để cải thiện kết quả của mô hình chúng tôi sử dụng thêm một mô hình 4-gram.

3.2 Kết quả thực nghiệm

Mô hình có sự cải thiện với độ lỗi giảm dần theo từng thế hệ.



Hình 3.2. WER trên tập Common Voice Vi



Hình 3.3. WER trên tập VIVOS

		Teacher	Gen1	Gen2	Gen3	Gen4
Common Voice Vi	**With LM**	12.51	10.16	9.88	9.415	9.048
	W/O LM	18.11	13.2	12.44	11.76	11.35
VIVOS	**With LM**	8.87	8.06	8.22	8.37	8.29
	W/O LM	13.49	12.86	12.55	12.45	12.35

Bảng 3.1. Bảng WER theo từng thế hệ mô hình trên Common voice vi và VIVOS.

Chương 4. Kết luận và hướng phát triển

4.1 Kết quả đạt được

4.1.1 Kết quả bài toán

Từ phần 3, ta có thể thấy kết quả rất khả quan. Mô hình đạt tỉ lệ lỗi khá nhỏ và vẫn có xu hướng có thể tốt hơn nếu được huấn luyện thêm. Kết quả đạt được sau 4 thế hệ là tỉ lệ lỗi từ 9.05% trên bộ dữ liệu Common voice vi và 8.3% trên bộ VIVOS.

4.1.2 Kiến thức

Qua báo cáo, nhóm hiểu được công thức, kiến trúc và cách thức cài đặt, huấn luyện mô hình Wav2Vec2.0 với kỹ thuật học bán giám sát.

4.1.3 Kỹ năng

Việc thực hiện báo cáo đã góp phần nâng cao khả năng đọc, tham khảo và hiện thực hóa các nghiên cứu khác của các thành viên nhóm. Đồng thời nhóm học được cách trình bày báo cáo, luận văn.

4.2 Hướng phát triển

Trong tương lai, nhóm sẽ nghiên cứu và thực hiện tối ưu mô hình hơn cho bài toán nhận diện tiếng nói cũng như tinh chỉnh để có thể áp dụng cho nhiều bài toán khác tương tự. Nói cách khác là tối ưu mô hình bằng cách tăng kích thước mô hình và số lượng dữ liệu có gán nhãn cũng như không gán nhãn, đồng thời thực hiện train các thế hệ đến gần hội tụ trước khi chuyển sang thế hệ tiếp theo thay vì chỉ trên 20 epochs/thế hệ như ở hiện tại. Đồng thời nhóm cũng có một số đề xuất cải thiện mô hình như sau:

1. Thay thế khối Transformer bằng Conformer để nắm bắt đặc trưng cục bộ tốt hơn
2. Mở rộng mô hình lên subword-level.

TÀI LIỆU THAM KHẢO

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, và Michael Auli. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". Trong: *CoRR* abs/2006.11477 (2020). arXiv: 2006.11477. URL: https://arxiv.org/abs/2006.11477.

[2] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, và Quoc V. Le. "Self-training with Noisy Student improves ImageNet classification". Trong: *CoRR* abs/1911.04252 (2019). arXiv: 1911.04252. URL: http://arxiv.org/abs/1911.04252.

[3] Markus Freitag và Yaser Al-Onaizan. "Beam Search Strategies for Neural Machine Translation". Trong: *CoRR* abs/1702.01806 (2017). arXiv: 1702.01806. URL: http://arxiv.org/abs/1702.01806.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, và Jürgen Schmidhuber. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". Trong: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, trang 369–376. ISBN: 1595933832. DOI: 10.1145/1143844.1143891. URL: https://doi.org/10.1145/1143844.1143891.

[5] Robert H. Davies, Richard Biddulph, và Irving Pollack, eds. *Automatic Recognition of Spoken Language*. 1950.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Illia Polosukhin, Lukasz Kaiser, Illia Polosukhin, Jakob Uszkoreit, Aidan N. Gomez, Illia Polosukhin, Lukasz Kaiser, Illia Polosukhin, Jakob Uszkoreit, Aidan N. Gomez, Illia Polosukhin, Jakob Uszkoreit, Aidan N. Gomez, Illia Polosukhin, Jakob Uszkoreit, Aidan N. Gomez, Illia Polosukhin, Jakob Uszkoreit, Aidan N. Gomez, Illia Polosukhin, và Jakob Uszkoreit. "Attention Is All You Need". Trong: *Advances in Neural Information Processing Systems*. 2017, trang 3004–3017.

[7] Rohan Jain, Thomas Unterthiner, Jakob Uszkoreit, và Alexander M. Rush. "Adaptive SpecAugment: A Data Augmentation Framework for Speech Recognition". Trong: *CoRR* abs/2108.02002 (2021). arXiv: 2108.02002. URL: https://arxiv.org/abs/2108.02002.

[8] Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, và Yonghui Wu. *Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition*. 2022. arXiv: 2010.10504 [eess.AS].