TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN VĂN ANH TUẤN

KHÓA LUẬN TỐT NGHIỆP

SỬ DỤNG KỸ THUẬT HỌC BÁN GIÁM SÁT CHO TỰ ĐỘNG PHÁT HIỆN LỖI PHÁT ÂM

Chuyên ngành: Khoa học dữ liệu

Giảng viên hướng dẫn: PGS. TS Nguyễn Việt Linh

TP. Hồ Chí Minh, tháng 12 năm 2022

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY

FACULTY OF INFORMATION TECHNOLOGY

NGUYEN VAN ANH TUAN

GRADUATION THESIS

A SEMI-SUPERVISED LEARNING APPROACH FOR AUTOMATIC PRONUNCIATION ERROR DETECTION

Major: Data Science

Instructor: Assoc. Prof. PhD. Nguyen Viet Linh

Ho Chi Minh City, December 2022

CONTENT SUMMARY

Title: A semi-supervised learning approach for Automatic Pronunciation Error Detection

Abstract:

- Reason for writing: The demand for learning and using English in the world in general and in Vietnam in particular is increasing. This stimulates many artificial intelligence researches and applications to help non-native speakers practice pronunciation. However, current approaches require a lot of data that have been carefully labeled by the expert, which are very hard to get. On the other hand, the amount of unlabelled data is abundant and has not been fully exploited. Therefore, it is pressing to study a method for speech assessment that take advantage of unlabeled data, beside labeled data.
- Problem: build an artificial intelligent system to evaluate speaker's pronunciation. This includes two steps:
    - Build a deep learning model to translate speech into phoneme: The input is a reading of sentence, the output is the phoneme sequence of the reader.
    - Compare the output and the groudth truth phoneme sequence to analyze the accuracy/errors of the speaker's pronunication.
- Methods:
    - Training Conformer using Pre-training Wav2Vec2.0 Framework combined with Self-training technique Noisy Student Training for Phoneme Recognition problem.
    - Find the longest common subsequence between the ground truth phoneme sequence and model-predicted speaker phoneme sequence to detect miss-pronunciation
- Content:
    - Knowledge about Convolution Neural Network, Transformer and Conformer encoder model.

    - Knowledge about objective function Connectiontist Temporal Classification.
    - Knowledge about Wav2Vec2.0 self-supervised learning pre-training framework.
    - Knowledge about Semi-supervised learning technique Noisy Student Training self-training method.
    - Knowledge about dynamic programming Longest Common Subsequence.
    - Knowledge about Word Error Rate metric.
    - Knowledge about PyTorch library.
- Results:
    - Successfully apply the Semi-supervised learning technique for training the Conformer model to predict phoneme sequence with PER 12.66%.
    - Successfully detect phoneme error in speech from predicted phoneme sequence of Conformer using Longest Common Subsequence.
- Conclusion:
    - Deep understanding the knowledge of Wav2Vec2.0 framework, Conformer model, CTC objective function, Noisy Student Training technique and Longest Common Subsequence algorithm.
    - Successfully applied pretrained Conformer model to Noisy Student Training self-training method for improve the downstream task: predict phoneme sequence, final PER 12.66%.
    - Successfully applied Longest Common Subsequence for phoneme error detection.
    - Experience in research, experiment the result and using opensource.

LỜI CẢM ƠN

Lời đầu tiên em xin phép gửi lời cảm ơn chân thành đến PGS. TS Nguyễn Việt Linh. Thầy là người đã trực tiếp giảng dạy, chỉ bảo, dẫn dắt, góp ý em trong phương diện học vấn lẫn kinh nghiệm làm việc, nhờ thầy mà em có thể có cơ hội thử sức với một đề tài khó như thế này, và cũng nhờ thầy mà em có thể có cơ hội hoàn thành tốt hơn bài báo cáo này.

Em xin cảm ơn TS. Đặng Thị Phúc, Phó khoa Công Nghệ Thông Tin, đã giúp em hoàn thành đề tài trước kia liên quan đến lĩnh vực giọng nói, quá trình làm việc với cô đã giúp kiến thức của em ngày càng vững chắc hơn.

Em xin cảm ơn PGS. TS. Huỳnh Trung Hiếu, Trưởng khoa Công Nghệ thông tin. Thầy là người đầu tiên dạy cho em những kiến thức cốt lõi trong ngành Khoa Học Dữ Liệu. Em cũng cảm ơn thầy vì đã đồng ý nhận phản biện đề tài của em. Em tin rằng những đánh giá phản biện của thầy sẽ góp phần quan trọng cho việc hoàn thiện luận văn này.

Em cảm ơn thầy Nguyễn Hữu Tình, giáo viên chủ nhiệm lớp DHKHDL15A của em, là người thầy đã dõi theo em từ năm nhất đến hiện tại, đã giúp đỡ em rất nhiều trong quá trình định hình bản thân, thầy đã truyền lửa cho em để em biết được rằng, chỉ cần cố gắng thì bất kỳ điều gì mình cũng có thể làm được, mặc kệ xuất phát điểm của bản thân ở đâu.

Thêm nữa, em cũng xin gửi lời cảm ơn đến quý thầy, cô ở Khoa Công Nghệ Thông Tin – Trường Đại học Công Nghiệp Thành phố Hồ Chí Minh đã giảng dạy, và cùng với vốn liếng tri thức của mình để giúp em trong suốt quãng thời gian em học tập tại trường.

Em cũng xin bày tỏ lòng biết ơn đến ban lãnh đạo của Trường Đại học Công Nghiệp Thành phố Hồ Chí Minh và các Khoa, Phòng ban chức năng đã trực tiếp hoặc gián tiếp giúp đỡ em trong suốt quá trình em học tập và thực hiện báo cáo này.

Cuối cùng, em cảm ơn công ty WeAI đã cho em cơ hội thực hiện một số dự án nghiên cứu thú vị, đặc biệt là các dự án liên quan đến phân tích giọng nói. Các dự

án của công ty đã thực sự giúp em phát triển vượt bậc về chuyên môn. Ngoài ra, em xin cảm ơn các anh chị và các bạn trong team R&D vì những hỗ trợ và môi trường làm việc vui vẻ hoà đồng.

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

Luận văn giải quyết một bài toán rất khó trong xử lý âm thanh: phát hiện lỗi phát âm tiếng Anh của người nói. Đây là một bài toán có tính ứng dụng cao, là công nghệ cốt lõi của một số phần mềm hỗ trợ học tiếng Anh. Tuy nhiên số lượng các nghiên cứu được công bố rất ít. Luận văn có thể xem là nghiên cứu tiên phong về mô hình đầu cuối (end-to-end model) cho bài toán này.

Luận văn đã thực hiện thành công kỹ thuật huấn luyện mô hình học sâu bán giám sát, dựa trên huấn luyện nhiễu cho mô hình student (noisy student training - NST). Mặc dù ý tưởng dùng NST đã được thực hiện trong bài toán nhận diện giọng nói, cách triển khai đòi hỏi khả năng phần cứng rất lớn để thực hiện cả việc tiền huấn luyện (pretraining) và tự huấn luyện (self-training). Luận văn có cách tiếp cận sáng tạo để vượt qua các hạn chế về phần cứng: (i) tận dụng các mô hình tiền huấn luyện sẵn có, đã được thực hiện để phục vụ cho bài toán nhận dạng giọng nói, (ii) làm nhỏ mô hình tiền huấn luyện bằng cách chỉ giữ lại một số khối Conformer. Nối phần tiền huấn luyện với một đầu ra phù hợp cho bài toán dự đoán âm vị.

Sinh viên đã thực hiện rất nhiều thử nghiệm để lựa chọn các thông số và kỹ thuật xử lý phù hợp, đặc biệt là đưa lớp relative positional embedding vào trước chuỗi các khối Conformer. Ngoài ra sinh viên đã thực hiện được thuật toán phát hiện lỗi sai trong phát âm bằng thuật toán tìm chuỗi con chung dài nhất. Trong quá trình thực hiện đề tài, sinh viên đã thể hiện được sự hiểu biết tốt về các mô hình học máy, đặc biệt là các mô hình xử lý âm thanh. Học viên đã rất chủ động nghiên cứu tìm hiểu các phương pháp huấn luyện và các thuật toán để đưa vào ứng dụng cho bài toán. Đặc biệt, sinh viên đã rất dũng cảm chọn một đề tài khó và mang tính rủi ro cao. Điều này không thường gặp ở một sinh viên đại học.

Kết luận: đây là một luận văn ở mức xuất sắc. .....

.....

.....

.....

.....

.....

.....

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

MỤC LỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC TỪ VIẾT TẮT

TỪ NGỮ	Ý NGHĨA
ASR – Automatic Speech Recognition	Nhận diện giọng nói
APED - Automatic Pronunciation Error Detection	Tự động phát hiện lỗi phát âm
Attention – Attention mechanism	Cơ chế chú ý
Average Pooling	Toán tử tính trung bình các điểm giá trị trong cùng một vùng đang xét – thuộc CNN
Batch	Một lô các mẫu dữ liệu
Beam Search	Thuật toán tìm kiếm chùm cải tiến dựa trên tìm kiếm tham lam
Beam width	Kích cỡ của mỗi chùm trong Beam Search
CALL – Computer-assisted Language Learning	Học ngoại ngữ với sự trợ giúp của máy tính
CAPT – Computer-assisted Pronunciation Training	Máy tính hỗ trợ học phát âm
CNN – Convolution Neural Network	Mạng nơ-ron tích chập
Codebook	Một nhóm các vector nhúng có kích thước cố định được học bởi mô hình (Wav2Vec2.0)
CV – Computer Vision	Thị giác máy tính
Contrastive Loss	Hàm mất mát tương phản
Contrastive Task –	Tác vụ tương phản – Học tương phản

Contrastive Learning	
Convolution	Tích chập
Convolution Subsampling	Lớp tích chập giảm kích thước dữ liệu
Cosine Similarity	Thước đo sự tương đồng giữa hai vector
CRF – Conditional Random Field	Mô hình thống kê dành cho các bài toán nhận diện mẫu và học máy
CTC – Connectionist Temporal Classification	Hàm mất mát cho các bài toán phân loại chuỗi nhãn theo thời gian
Decision-directed Learning	Huấn luyện theo hướng quyết định là phương pháp tìm lời giải theo hướng lặp đi lặp lại
DBN – Deep Belief Network	Mạng niềm tin sâu là một mô hình huấn luyện theo hướng học không giám sát
Diversity Loss	Hàm mất mát đa dạng trong Wav2Vec2.0
Downstream task	Tác vụ phía sau, là tác vụ chính muốn giải khi tận dụng lại mô hình đã được tiền huấn luyện
DWT – Dynamic Time Warping	Thuật toán đo lường độ tương đồng giữa hai chuỗi tuần tự có chiều dài khác nhau
E2E – End-to-end	Đầu cuối, thường nói đến mạng học sâu
Encoder	Bộ mã hóa
Feature encoder	Bộ mã hóa tính năng (tầng tích chập trong Wav2Vec2.0)
Fine-tune, fine-tuning	Điều chỉnh mô hình sau khi tận dụng tiền huấn luyện
Global Statistics Supervision	Giám sát thống kê toàn cầu
GLU – Gated Linear Unit	Cơ chế cổng để kiểm soát luồng thông tin trong mạng, tương tự với cơ chế tự chú ý
GOP – Goodness of	Thuật toán tính tỉ lệ giống nhau giữa chuỗi âm vị của

Pronunication	người nói với chuỗi âm vị thực tế
Ground truth	Mẫu thực tế, luôn đúng
HMM – Hidden Markov Model	Mô hình thống kê dựa trên tính chất Markov
ImageNet	Cơ sở dữ liệu ảnh lớn thiết kế cho các tác vụ nhận diện ảnh và các nghiên cứu liên quan
Language Model	Mô hình ngôn ngữ
LCS – Longest Common Subsequence	Thuật toán quy hoạch động tìm chuỗi con chung dài nhất
LER – Label Error Rate	Tỉ lệ lỗi nhãn
Logits	Giá trị xác suất lớp cuối cùng của mạng nơ-ron (cung cấp phân phối xác suất trên tập từ điển)
Multiple-instance Learning	Là một loại học giám sát, thay vì nhận được một nhãn sẽ nhận được một nhóm nhãn
NST – Noisy Student Training	Là phương pháp huấn luyện bán giám sát, được mở rộng ý tưởng từ tự huấn luyện và chắt lọc mô hình thế hệ sau
Partial Supervision	Giám sát một phần
PER – Phoneme Error Rate	Tỉ lệ lỗi âm vị
Phoneme	Âm vị
Pointwise Convolution	Tích chập theo từng điểm, có kerkel size 1×1
Pre-training	Tiền huấn luyện mô hình
Pseudo Label	Nhãn giả được sinh ra từ mô hình teacher
Relative Sinusoidal Positional Encoding	Là một loại nhúng vị trí cho các mô hình kiểu Transformer (có cơ chế chú ý) mà giúp mô hình có khả năng nắm được thông tin vị trí tương đối
ReLU	Hàm kích hoạt ReLU

RNN – Recurrent Neural Network	Mạng nơ-ron hồi quy
Self-attention	Cơ chế tự chú ý
Self-labeling	Tên gọi khác của self-training
Self-training	Tự học giám sát
Semi-Supervised Learning	Huấn luyện bán giám sát
SOTA – state-of-the-art	Mức độ cao nhất của phát triển, thường chỉ đến kết quả tốt nhất theo một thang đo nào đó trong cùng một lĩnh vực
Stride	Bước nhảy trong CNN
Supervised Learning	Huấn luyện giám sát
Temperature	Giá trị nhiệt độ trong việc lựa chọn các mục của codebook (Wav2Vec2.0)
Temporal Classification	Bài toán phân loại các nhãn theo thời gian
Time-step	Bước thời gian
Transfer Learning	Học chuyển giao
Unsupervised Learning	Học không giám sát
Weak Supervision	Giám sát yếu
WER – Word Error Rate	Tỉ lệ lỗi từ

CHƯƠNG 1. GIỚI THIỆU

1.1. Tổng quan

1.1.1. Bối cảnh

Trong thời đại hội nhập ngày nay, việc sử dụng và giao tiếp tiếng Anh ngày càng trở nên thông dụng. Việc một người biết giao tiếp bằng tiếng Anh bây giờ không phải là chuyện lạ nữa. Đối với Việt Nam, một đất nước đang ngày càng mở rộng cánh cửa hội nhập thì tầm quan trọng trong việc học tiếng Anh giao tiếp lại được quan tâm hơn bao giờ hết. Toàn cầu hóa đã đem đến nhiều bước ngoặt trong đời sống, kinh tế, xã hội: các doanh nghiệp giao thương với nước ngoài ngày càng nhiều; các quỹ đầu tư, tập đoàn nước ngoài ồ ạt vào Việt Nam mang theo những cơ hội lớn về nghề nghiệp cho người Việt.

Đi cùng xu hướng đó, những yêu cầu trong tuyển dụng nhân sự của các doanh nghiệp cũng đã thiết lập những quy chuẩn cao hơn trong kĩ năng tiếng Anh giao tiếp. Tiếng Anh không còn là một yếu tố cộng thêm để xem xét ứng viên nữa mà là một yêu cầu bắt buộc khi bạn muốn gia nhập vào các tổ chức [1]. Theo trang Indeed [2], việc có một chứng chỉ tiếng Anh có thể hữu ích cho người nói tiếng Anh như ngôn ngữ thứ hai hay đang có nhu cầu đi du học, hay là tăng cơ hội ứng tuyển cho người có nhu cầu tìm việc.

Thế nên, việc học tiếng Anh đang rất cần thiết. Vì thế, ngày càng có nhiều ứng dụng giúp cho việc tự học cách phát âm tiếng Anh trở nên phổ biến hơn, có thể kể tên như Elsa [3] hay Duolingo [4]. Cũng như các nghiên cứu về bài toán đánh giá khả năng phát âm của người đọc (đánh giá giọng nói – Speech Verification) trong lĩnh vực Trí tuệ nhân tạo cũng đang nổi lên [5] [6] [7] [8] [9]. Tuy nhiên các nghiên cứu này đều sử dụng dữ liệu giọng nói đã được đánh nhãn để làm nguồn dữ liệu chính trong việc huấn luyện mô hình. Mà dữ liệu giọng nói đánh nhãn là một nguồn rất ít ỏi, so với lượng dữ liệu giọng nói không đánh nhãn có nguồn cung dồi dào trên Internet. Một kỹ thuật trong lĩnh vực Trí tuệ nhân tạo cho phương pháp tận dụng lượng dữ liệu không đánh nhãn để giải quyết bài toán có sẵn gọi là học bán giám sát (Semi-supervised Learning).

1.1.2. Lý do chọn đề tài

Lượng dữ liệu có nhãn không quá nhiều, nhưng lượng dữ liệu không có nhãn lại cực kỳ dồi dào, nên tôi chọn nghiên cứu kỹ thuật huấn luyện Semi-supervised Learning cho bài toán APED và đã chọn "Sử dụng kỹ thuật học bán giám sát cho tự động phát hiện lỗi phát âm" làm tiêu đề cho đề tài khóa luận tốt nghiệp của mình.

1.2. Mục tiêu nghiên cứu

- Tìm hiểu về kiến trúc Encoder Conformer thuộc bài toán nhận dạng giọng nói.
- Tìm hiểu về mô hình Pre-training self-supervised learning Wav2Vec2.0
- Tìm hiểu về kỹ thuật Semi-supervised Learning Self-training Noisy Student Training.
- Áp dụng kết hợp Wav2Vec2.0 để pre-training cho Conformer, sau đó đi huấn luyện tiếp Conformer dùng Noisy Student Training.
- Dùng Conformer đã được huấn luyện để dự đoán chuỗi âm vị của người nói.
- Tìm hiểu về thuật toán quy hoạch động Longest Common Subsequence (LCS) để so khớp giữa chuỗi âm vị thực tế và dự đoán để đánh giá giọng nói, mục tiêu là đưa ra được người nói hiện đang sai sót ở vị trí nào trong câu nói.

1.3. Phạm vi nghiên cứu

- Kiến thức và hiểu biết về Conformer, Pre-training, Self-training và Longest Common Subsequence.
- Sử dụng bộ dữ liệu có nhãn được kết hợp từ Libri-Light và LibriSpeech cùng với bộ dữ liệu không có nhãn LibriSpeech.

1.4. Ý nghĩa khoa học và thực tiễn

Nghiên cứu này góp phần làm tiền đề cho nghiên cứu về bài toán Automatic Pronunciation Error Detection (APED) sử dụng kỹ thuật Semi-supervised Learning trong việc cải thiện khả năng dự đoán chuỗi âm vị của mô hình Conformer, từ đó lấy chuỗi dự đoán này đi so sánh với chuỗi thực tế thông qua thuật toán LCS. Kết quả thu được sẽ phản ánh tốt hơn về nghiên cứu APED với ngôn ngữ tiếng Anh.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Bài toán Automatic Pronunciation Error Detection

2.1.1. Khái niệm

Với sự phát triển nhanh chóng của việc toàn cầu hóa và giáo dục, số lượng người cần học ngôn ngữ ngày càng tăng trưởng. Tuy nhiên, hầu hết người học đều gặp một vấn đề đó là không tìm được giáo viên hướng dẫn hoặc không có thời gian để đi theo một lộ trình học có hệ thống. Vì thế, những nghiên cứu về hệ thống Học ngoại ngữ với sự trợ giúp của máy tính (CALL) nổi lên nhiều hơn [10].

CALL được nghiên cứu với mục tiêu đưa ra một dịch vụ giáo dục linh hoạt, mà có thể được sử dụng để thay thế hoàn toàn cho yêu cầu của việc học một ngôn ngữ trong quãng thời gian bị ngắt quãng, không liền mạch [11]. Đặc biệt, việc luyện tập phát âm là một phần rất quan trọng trong việc giao tiếp thường ngày, và hệ thống Máy tính hỗ trợ phát âm (CAPT) được thiết kế cho việc này. Những hệ thống kể trên đóng vai trò quan trọng trong bài toán Tự động phát hiện lỗi phát âm (APED).

Một hệ thống APED đầu tiên sẽ đưa ra một đoạn chữ được định nghĩa sẵn (và nếu cần thì sẽ kèm thêm một đoạn giọng nói có từ trước để người học có thể nghe tham khảo). Nhiệm vụ của người học rất đơn giản: cố gắng đọc đúng đoạn chữ này nhất có thể. Ví dụ, người học muốn học cách phát âm từ “apple” (chuỗi âm vị của nó là “æ p l”), nhưng người học có thể đọc nhầm thành “ə p l”. Trong trường hợp này, chúng ta định nghĩa chuỗi “æ p l” là *chuỗi phát âm chuẩn* và chuỗi “ə p l” là *chuỗi của người đọc*. Hệ thống APED sẽ dự đoán chính xác được người dùng đọc từ “apple” bị sai ở vị trí cụ thể nào, từ đó đưa ra phản hồi cho người học biết để người học có thể kịp thời sửa sai, dần dần, người học sẽ cải thiện khả năng phát âm của mình [11].

2.1.2. Các nghiên cứu đã có

APED đã được nghiên cứu hàng thập kỷ. Dựa trên cách để đánh giá mức độ so khớp giữa câu phát âm từ người học và câu phát âm chuẩn, có một vài phương pháp so sánh dựa trên phương pháp Goodness of Pronunciation (GOP) đã được đề xuất để giải quyết bài toán APED. Ann Lee và cộng sự [12] đã trình bày kỹ thuật so

khớp cho đánh giá phát âm bằng cách căn chỉnh giữa câu nói của người học và câu nói chuẩn thông qua Dynamic Time Wraping (DWT). Một bài nghiên cứu khác của Ann Lee cùng cộng sự [13] sử dụng biểu đồ hậu nghiệm của Deep Belief Network (DBN) làm đầu vào cho DWT (nghiên cứu ở câu trước) để phát hiện sai sót cấp độ từ (thay vì theo âm vị như trong luận văn này), cụ thể hệ thống hoạt động bằng cách so sánh câu nói của người học (không phải bản xứ) với ít nhất một câu nói của người bản xứ, từ đó trích xuất các tính năng mô tả mức độ căn chỉnh (căn chỉnh giữa chuỗi thực và chuỗi dự đoán) sai sót. Kết quả của nghiên cứu này cho thấy việc thay thế MFCC [14] hay hậu nghiệm của Gauss bằng cách cài đặt thực nghiệm theo kiểu Unsupervised của hậu nghiệm DBN giúp hệ thống cải thiện tương đối khoảng 14%. Hơn nữa, hệ thống vẫn ổn định khi chỉ sử dụng khoảng 30% dữ liệu có đánh nhãn. Đây là một tiền đề cho việc sử dụng dữ liệu có nhãn kết hợp với không có nhãn trong bài toán APED. Ngoài ra, còn có các nghiên cứu liên quan cũng tương tự như hai bài nghiên cứu trên [15] [16] [17] [18]. Tuy nhiên, hạn chế của phương pháp này là hệ thống có nhiều thành phần, làm phức tạp hóa quy trình thực hiện bài toán.

Gần đây, với xu hướng gia tăng việc áp dụng Neural Network và sự phát triển của công nghệ Nhận diện giọng nói (ASR), có một vài nghiên cứu đã được đề xuất để làm giảm bớt các thành phần trong hệ thống APED (các phương pháp dựa trên GOP được trình bày ở đoạn trước cần nhiều thành phần phối hợp với nhau). Với thành phần cốt lõi vẫn là bài toán ASR, các phương pháp này dùng để nhận diện chuỗi âm vị từ câu nói của người học và căn chỉnh chuỗi này với chuỗi âm vị chuẩn, từ đó đưa ra lỗi phát âm. Có thể kể đến như nghiên cứu của Leung và cộng sự [19], nhóm tác giả kết hợp Convolution Neural Network (CNN), Recurrent Neural Network (RNN) và hàm mục tiêu Connectionist Temporal Classification (CTC). Hoặc như của Long và cộng sự [6], nhóm tác giả đề xuất một mô hình kết hợp giữa CTC và cơ chế Attention. Phương pháp APED dựa trên ASR này hoàn toàn giúp giảm nỗ lực trong việc triển khai mô hình trên thực tế khi so với các phương pháp sử dụng GOP. Đặc biệt, mô hình Conformer [20], kết hợp giữa CNN và Transformer [21] để

học đồng thời thông tin ngữ cảnh cục bộ lẫn toàn cục, giúp đẩy kết quả của bài toán ASR lên mức giới hạn, trở thành mô hình SOTA [22] của bộ dữ liệu đánh giá LibriSpeech [23] tiếng Anh. Vì thế, rất hứa hẹn khi sử dụng các phương pháp APED dựa trên ASR cho bài toán dự đoán chuỗi âm vị, vì lúc này, chỉ cần thay đổi đầu ra của mô hình từ dạng ký tự thành dạng phiên âm.

2.2. Mô hình nhận dạng giọng nói Conformer

2.2.1. Tổng quan

Các phương pháp nhận dạng giọng nói đầu cuối (End-to-end ASR) dựa trên Neural Network đã được cải thiện rất nhiều trong những năm gần đây. RNN đã từng là sự lựa chọn ưu tiên trong bài toán ASR [24] [25] [26] [27], bởi vì dạng mô hình này có thể mô hình hóa các phụ thuộc theo thời gian trong chuỗi âm thanh một cách hiệu quả (như trong model RNN-Transducer [28]). Gần đây, kiến trúc mô hình Transformer dựa trên cơ chế tự chú ý (self-attention) [21] [29] đã được áp dụng rộng rãi cho các bài toán cần mô hình hóa các chuỗi bởi vì khả năng nắm bắt được thông tin dài hạn và tỏ ra hiệu quả trong việc huấn luyện. Một mặt khác, CNN cũng được ứng dụng thành công cho bài toán ASR, mà cơ chế chủ yếu để nắm bắt thông tin là qua một cửa sổ nhỏ theo từng lớp. Ví dụ như Jasper [30] là một CNN đầu cuối, Quartznet [31] là một mạng tích chập 1 chiều với cấu trúc có thể tách rời kênh thời gian của giọng nói, toàn bộ mạng là sự kết hợp bởi tích chập 1 chiều (1-D CNN), Batch Normalization [32] và hàm kích hoạt ReLU [33], hay Contextnet [34] cải thiện CNN truyền thống cho bài toán ASR bằng cách thêm mô-đun Squeeze-and-excitation để thực hiện average-pooling toàn cục (global average pooling), mô hình này đạt được kết quả rất tốt, ngang ngửa so với Conformer. Bên cạnh đó còn một số nghiên cứu khác về CNN cho ASR [35] [36].

Tuy nhiên, những mô hình chỉ có cơ chế tự chú ý (self-attention) hay chỉ có tích chập (convolution) đều có những hạn chế riêng. Trong khi Transformer rất tốt trong việc nắm bắt những thông tin, ngữ cảnh toàn cục, thì nó lại khá hạn chế trong việc bắt những thông tin, mẫu cục bộ. Mặt khác, CNN lại rất được ưa chuộng trong những tác vụ liên quan đến thị giác máy tính (Computer Vision – CV) bởi vì khả

năng khai thác thông tin cục bộ của nó, nó học được rất tốt những thông tin cạnh, hình dạng, vị trí thông qua một cửa sổ nhỏ. Tuy nhiên một giới hạn của việc sử dụng kết nối cục bộ (thông tin trong một cửa sổ kết nối với nhau) này là mô hình sẽ phải cần nhiều lớp, nhiều tham số để bắt được hết thông tin toàn cục. Để khắc phục vấn đề này, Contextnet được đề cập ở trên đã thêm vào một mô-đun là Squeeze-and-Excitation [37] ở mỗi khối phần dư (residual block) để bắt được ngữ cảnh dài hơn. Tuy nhiên, việc này vẫn bị giới hạn ở những thông tin toàn cục phức tạp, bởi vì mô-đun này chỉ thực hiện trung bình toàn cục (global averaging) trên toàn bộ chuỗi câu.

Những nghiên cứu gần đây cho thấy rằng việc kết hợp tích chập và cơ chế tự chú ý cải thiện hơn việc sử dụng mỗi thành phần riêng lẻ [38]. Khi kết hợp cùng nhau, mô hình như thế có thể học cả các tính năng cục bộ theo từng vị trí và sử dụng cả thông tin nội dung toàn cục. Ngoài ra còn có các nghiên cứu khác liên quan đến cách kết hợp này [39] [40] [41].

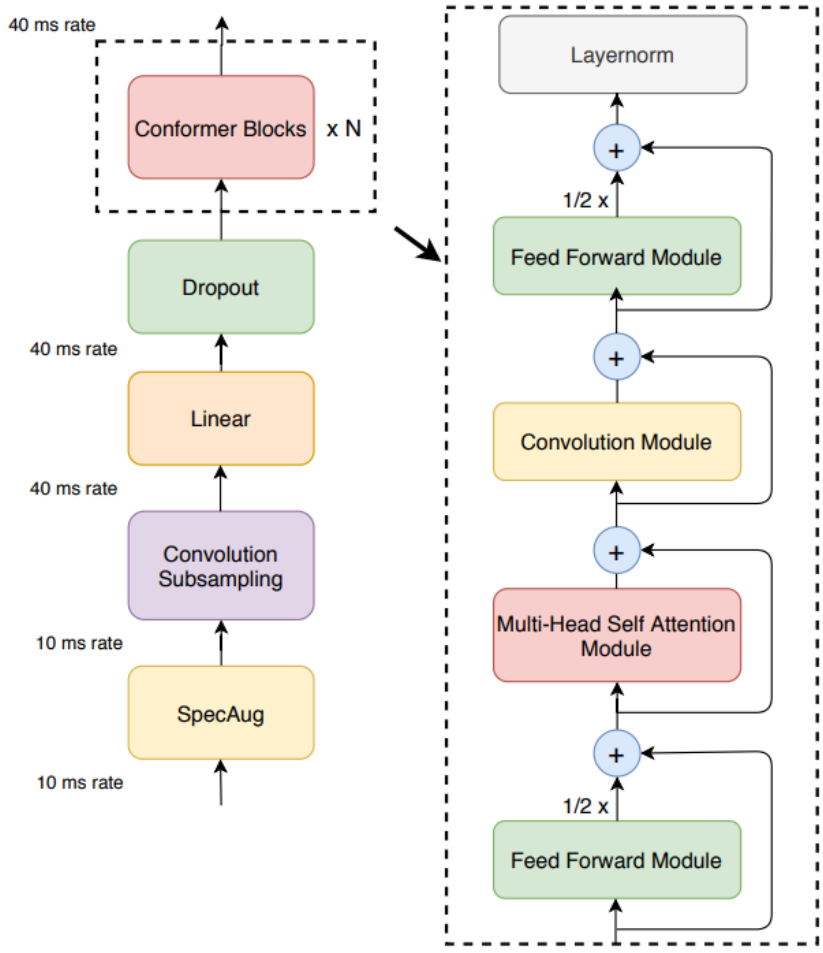
Anmol cùng cộng sự [20] đã trình bày một kiến trúc có tên là Conformer. Kiến trúc này kết hợp cơ chế tự chú ý với tích chập dành cho bài toán ASR. Họ giả định rằng cả thông tin cục bộ lẫn ngữ cảnh toàn cục đều quan trọng để tham số hóa, và cần tham số hóa một cách hiệu quả. Cơ chế tự chú ý tìm hiểu sự tương tác toàn cục của thông tin trong khi đó tích chập sẽ nắm bắt hiệu quả những tương quan cục bộ dựa trên độ lệch tương đối. Cách kết hợp mạng của họ dựa trên hình thức kẹp: kẹp giữa một cặp mô-đun feed forward là mô-đun convolution và mô-đun self-attention. Kết quả được trình bày vào năm 2020, đã đạt được state-of-the-art trên bộ dữ liệu đánh giá LibriSpeech [23], vượt qua bài nghiên cứu trước đó là Transformer Transducer [29], kết quả tốt nhất của họ đạt 1.9%/3.9% WER khi kết hợp thêm một mô hình ngôn ngữ.

Trong bài này, tôi sử dụng Conformer làm Encoder chính để thực hiện bước Pre-training sử dụng Framework Wav2Vec2.0 [42] (dùng bản sửa đổi Wav2Vec2.0 Conformer) và Self-training sử dụng Noisy Student Training [43], sau cùng đưa mô hình Conformer tốt nhất đi dự đoán chuỗi âm vị của người nói.

2.2.2. Kiến trúc

2.2.2.1. Conformer Encoder

Mô hình Conformer sẽ chỉ là phần Encoder trong một mạng Encoder-Decoder truyền thống. Nhiệm vụ của Conformer encoder này đầu tiên sẽ xử lý dữ liệu đầu vào với một lớp Convolution Subsampling, trước đó là sẽ tăng cường dữ liệu bằng SpecAugment [44], theo sau Convolution Subsampling là một lớp Linear để chuyển kích cỡ của dữ liệu đầu vào cho vừa với kích cỡ của các khối Conformer và



Hình 1. Kiến trúc của Conformer Encoder

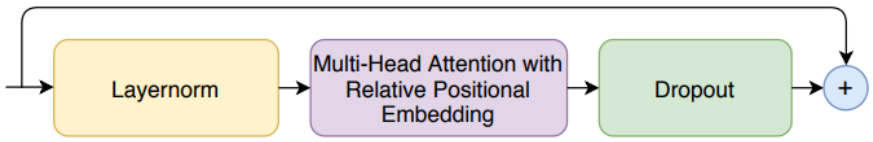
Dropout [45] để tránh học quá khớp (overfit) [46], sau đó dữ liệu được đưa qua một số khối Conformer, được mô tả trong Hình 1.

Như Hình 1, một khối Conformer kết hợp hai mô-đun feed forward với dữ liệu được kết nối theo residual nhưng giảm bớt một nửa, kẹp giữa là mô-đun multi-

headed self-attention và mô-đun convolution. Theo sau 4 khối là một lớp Layer Norm [47]. Phần 2.2.2.2, 2.2.2.3, 2.2.2.4 sẽ nói về các mô-đun self-attention, convolution và feed forward. Phần 2.2.2.5 sẽ nói về cách kết hợp các mô-đun này lại với nhau.

2.2.2.2. Mô-đun Multi-Headed Self-Attention

Tác giả của Conformer sử dụng lại multi-head self-attention (MHSA) kết hợp thêm một kỹ thuật từ mô hình Transformer-XL [48], là kỹ thuật mã hóa vị trí tương đối theo sinusoidal (relative sinusoidal positional encoding scheme). Việc mã hóa vị trí tương đối này cho phép mô-đun tự chú ý có thể tổng quát hóa tốt hơn trên nhiều chiều dài đầu vào khác nhau và encoder sau khi được huấn luyện có thể mạnh mẽ hơn trước sự biến thiên của chiều dài câu nói. Mô-đun này sử dụng cách kết nối đơn

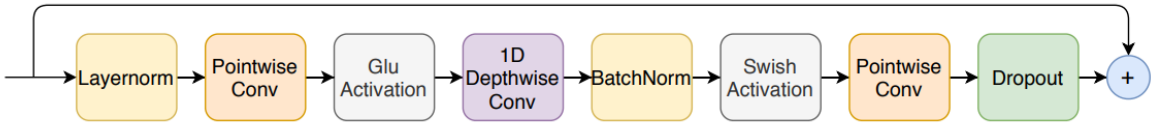


Hình 2. Mô tả cấu trúc của mô-đun Multi-headed self-attention

vị pre-norm (đầu vào sẽ cộng với đầu vào đã qua Layernorm – pre-norm residual units) [49] [50], kết hợp thêm Dropout để giúp quá trình huấn luyện trơn tru hơn và cũng để chỉnh hóa khi mô hình quá sâu. Hình 2 mô tả mô-đun Multi-Headed Self-Attention.

2.2.2.3. Mô-đun Convolution

Được tạo cảm hứng từ kiến trúc mô hình Lite Transformer của Z. Wu và cộng sự [41], mô-đun tích chập được bắt đầu với một cơ chế cổng (gating mechanism) [51], cụ thể là gồm tích chập theo điểm (pointwise convolution) [52] và GLU (Gated Linear Unit). Theo sau là một lớp tích chập theo chiều sâu 1-D [53], một lớp Batch Normalization, một hàm kích hoạt Swish [54], một tích chập theo điểm và cuối

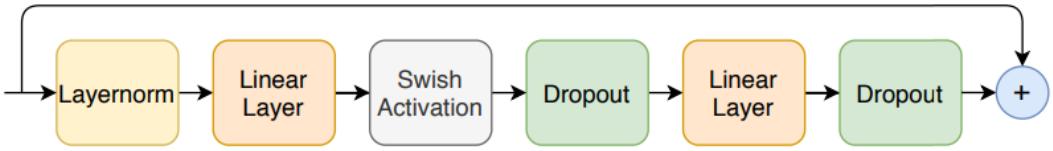


Hình 3. Cấu trúc của mô-đun Convolution

cùng là Dropout. Cách kết nối của mô-đun này cũng sử dụng prenorm residual units. Hình 3 mô tả mô-đun convolution.

2.2.2.4. Mô-đun Feed Forward

Mô-đun Feed Forward cũng sử dụng pre-norm residual units, theo sau Layer Normalization là một Linear layer, một hàm kích hoạt Swish, một Dropout để chỉnh



Hình 4. Cấu trúc của mô-đun Feed Forward

hóa, một Linear và một Dropout. Hình 4 mô tả cấu trúc của mô-đun Feed Forward.

2.2.2.5. Khối Conformer

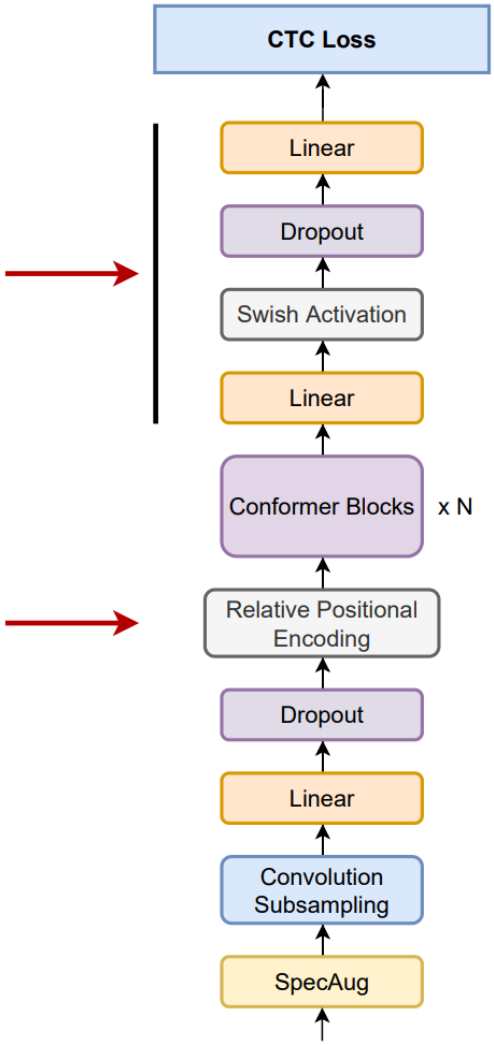
Ở Hình 1, một khối Conformer sẽ theo kiểu Sandwich với hai Feed Forward ở hai bên và kẹp giữa là Multi-headed Self-Attention và Convolution. Kiểu Sandwich này được gợi cảm hứng từ Macaron-Net [55], mà ở mạng này, tác giả đề xuất cách thay đổi lớp feed-forward truyền thống của khối Transformer thành hai nửa lớp feed-forward, một trước attention và một sau. Cũng như Macaron-Net, tác giả sửa khối Conformer để có hai nửa trọng số của mô-đun feed forward, sau mô-đun feed forward thứ hai sẽ có thêm một lớp Layer Normalization. Nếu theo biểu diễn toán học, đầu vào là x_i đi vào khối Conformer thứ i, thì đầu ra y_i của khối là:

$$\begin{aligned} \widetilde{x_i} &= x_i + \frac{1}{2}\mathrm{FFN}(\mathrm{x_i}) \\ x_i' &= \widetilde{x_i} + \mathrm{MHSA}(\widetilde{x_i}) \\ x_i'' &= x_i' + \mathrm{Conv}(\mathrm{x_i'}) \\ y_i &= \mathrm{Layernorm}(x_i'' + \frac{1}{2}\mathrm{FFN}(x_i'')) \end{aligned} \tag{1}$$

Ký hiệu FFN là dành cho mô-đun Feed Forward, MHSA là mô-đun Multi-headed Self-attention, và Conv là mô-đun Convolution.

2.2.2.6. Cải tiến cấu trúc của Conformer

Trong quá trình thực nghiệm, tôi đã thêm một số thành phần mới vào cấu trúc Conformer truyền thống. Cụ thể, tôi đã thêm một lớp Relative Positional Encoding vào trước các khối Conformer để mã hóa được thông tin, vị trí của chuỗi đầu vào. Ở sau các khối Conformer thì là sự kết hợp của 2 lớp Linear kẹp giữa là hàm kích hoạt Swish và Dropout. Trong quá trình thực nghiệm, tôi nhận thấy rằng cách kết hợp này phù hợp với bài toán và đưa ra kết quả khả quan. Ở cuối, hàm mất mát CTC là hàm mục tiêu mặc định. Hình 5 mô tả cấu trúc Conformer mới được chỉnh sửa, mũi tên màu đỏ chỉ vào những thành phần đã chỉnh sửa.



Hình 5. Kiến trúc Conformer được chỉnh sửa

2.3. Hàm mục tiêu huấn luyện giám sát: 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 thức (ví dụ: nhận diện chữ viết tay, nhận diện giọng nói, nhận diện cử chỉ), mà đầu vào ở đây thường là sẽ có nhiễu, giá trị thực từ các luồng đầu vào thường được ký hiệu bởi chuỗi các ký hiệu rời rạc như là các kí tự hay là các từ.

Năm 2006, lúc A. Graves cùng cộng sự viết bài nghiên cứu về CTC [56] các mô hình dạng đồ thị như mô hình Markov ẩn (Hidden Markov Models – HMM [57]), mô hình các trường ngẫu nhiên có điều kiện (conditional random fields – CRFs [58]) và các dạng tương tự của hai mô hình này đang đứng đầu trong các khung mô hình để giải quyết bài toán mô hình hóa các chuỗi. Tuy cách tiếp cận này đã được chứng minh rất thành công trong nhiều bài toán, nhưng nó có một số điểm hạn chế sau:

1. Các mô hình như HMM, CRF đòi hỏi phải có một lượng kiến thức đủ trong tác vụ muốn giải quyết, ví dụ việc thiết kế các trạng thái của mô hình HMM hay chọn các tính năng đầu vào cho CRF.
2. Các mô hình này đòi hỏi phải có giả định phụ thuộc một cách tường minh (và thường là câu hỏi mở) để khiến cho việc suy luận kết quả trở nên dễ hiểu hơn. Ví dụ giả định rằng các quan sát không phụ thuộc lẫn nhau trong HMM.
3. Đối với HMM tiêu chuẩn, quá trình huấn luyện mang tính sinh (tạo ra), mặc dù việc đánh nhãn các chuỗi là quá trình phân biệt (phân biệt các nhãn theo từng vị trí).

Mặt khác, các dạng mô hình RNN không yêu cầu kiến thức về dữ liệu từ trước, ngoài việc chọn dữ liệu đầu vào và đầu ra. Nó còn dùng để huấn luyện, và việc huấn luyện này mang tính phân biệt, và các trạng thái bên trong mô hình còn cung cấp một cơ chế đặc biệt để giúp mô hình hóa chuỗi thời gian. Thêm nữa, mô hình này

còn khá là mạnh mẽ khi đối mặt với dữ liệu có nhiễu cả mặt không gian lẫn thời gian.

Tuy nhiên việc áp dụng các mô hình RNN một cách trực tiếp vào việc đánh nhãn các chuỗi vẫn là điều không thể. Vấn đề là hàm mục tiêu của Neural Network truyền thống được định nghĩa riêng cho mỗi điểm dữ liệu trong chuỗi huấn luyện, nói cách khác, các mô hình RNN chỉ có thể dùng để huấn luyện cho việc tạo ra các chuỗi có nhãn độc lập với nhau. Có nghĩa là dữ liệu để huấn luyện phải được phân đoạn ra từ trước, và đầu ra của Neural Network cũng phải được xử lý để đưa ra chuỗi nhãn cuối cùng.

Tính đến 2006, phương pháp hiệu quả nhất trong việc ứng dụng các dạng mô hình RNN cho việc đánh nhãn các chuỗi là kết hợp nó với HMM, cách tiếp cận này gọi là cách tiếp cận kết hợp (Hybrid) [59]. Hệ thống Hybrid này sử dụng HMM để mô hình hóa cấu trúc tuần tự tầm xa của dữ liệu, mạng nơ-ron lúc này để đưa ra phân loại cục bộ. Thành phần HMM có khả năng tự động phân đoạn chuỗi tuần tự trong quá trình huấn luyện, và để chuyển đổi mạng phân loại thành các chuỗi nhãn. Tuy nhiên, như đã đề cập các hạn chế của HMM ở trên, hệ thống Hybrid này không thể khai thác hết được 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 [56] về Connectionist Temporal Classification (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 [60].

Phần 2.3.2 sẽ trình bày về tác vụ đánh nhãn chuỗi chưa được phân đoạn gọi tên là Temporal Classification (phân loại theo trình tự thời gian) [61], và sử dụng RNN cho bài toán này như một tác vụ phân loại theo trình tự thời gian liên kết (connectionist temporal classification). Việc phân loại này sẽ diễn ra độc lập theo từng bước thời gian (time-step) hay từng khung dữ liệu của chuỗi dữ liệu tuần tự.

2.3.2. Kỹ thuật

2.3.2.1. Bài toán Temporal Classification

Gọi S là tập dữ liệu huấn luyện 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 vector số thực có m chiều. Không gian mục tiêu $\mathcal{Z} = L^*$ là tập tất cả chuỗi tuần tự qua một bảng chữ cái L. Nói tổng quát, ta đề cập mỗi phần tử của L^* như là *chuỗi nhãn* hay *nhãn dán*. Mỗi mẫu trong S chứa một cặp chuỗi $(\mathbf{x}, \mathbf{z})$. Chuỗi mục tiêu $\mathbf{z} = (z_1, z_2, \cdots, z_U)$ có độ dài dài nhất bằng với độ dài của chuỗi $\mathbf{x} = (x_1, x_2, \cdots, x_T)$, có nghĩa là $U \leq T$. Bởi vì chuỗi đầu vào và chuỗi mục tiêu không có chung độ dài, không có cách tiên nghiệm nào để căn chỉnh hai chuỗi này.

Mục tiêu là dùng S để huấn luyện một mô hình phân loại theo thời gian (temporal classifier) $h: \mathcal{X} \mapsto \mathcal{Z}$ để phân loại đầu vào chưa nhìn thấy trước đây theo mục tiêu là giảm thiểu một thang đo độ lỗi nào đó (tùy theo tác vụ muốn giải quyết là gì).

<u>*Label Error Rate*</u>: Đối với bài toán Temporal Classification, một thang đo lỗi ta quan tâm có ngữ cảnh như sau: đưa trước một tập $S' \subset \mathcal{D}_{\mathcal{X}\times\mathcal{Z}}$ khác với S, định nghĩa *label error rate* (LER) của mô hình phân loại theo thời gian h là khoảng cách để chỉnh sửa một chuỗi nhãn đã được phân loại với mục tiêu trên S', có nghĩa là:

$$LER(h, S') = \frac{1}{Z} \sum_{(\mathbf{x},\mathbf{z})\in S'} ED(h(\mathbf{x})) \tag{2}$$

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 $\mathbf{q}$ – có thể hiểu là tối thiểu số lượng thêm, sửa, xóa để biến $\mathbf{p}$ thành $\mathbf{q}$. Đây là thang đo thường được dùng cho các tác vụ như thế này (như nhận dạng giọng nói hay nhận dạng chữ viết tay) mà mục tiêu là giảm thiểu số

lượng sai sót trên bản dịch thực tế. Trong bài luận này, LER sẽ là Phoneme Error Rate (PER), được trình bày trong phần 2.7.1.

2.3.2.2. Connectionist Temporal Classification

Một mạng CTC có một lớp xuất đầu ra softmax [62] với nhiều hơn một nhãn khi so với L. Các giá trị của $|L|$ đơn vị đầu tiên được thông dịch như là xác suất để quan sát được nhãn tương ứng tại một thời điểm cụ thể. Giá trị kích hoạt của đơn vị cộng thêm là xác suất để quan sát được một nhãn "blank" (rỗng) 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 nào đó đều có thể tính được bằng cách 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 là T, định nghĩa một RNN với m đầu vào, n đầu ra và vector trọng số w như là 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. y_k^t được thông dịch như là xác suất quan sát được nhãn k tại thời điểm t, điều này định nghĩa một phân phối qua tập L'^T có độ dài chuỗi là T qua bảng chữ cái $L' = L \cup \{\text{blank}\}$:

$$p(\pi|\mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t, \qquad \forall \pi \in L'^T \tag{3}$$

Ta sẽ ký hiệu π là một phần tử của L'^T, gọi là *đường dẫn* (path). Công thức số (3) ngầm giả định rằng đầu ra của mạng tại các thời điểm khác nhau là độc lập có điều kiện với điều kiện là các trạng thái bên trong của mạng. Điều này đảm bảo rằng không có tồn tại kết nối từ lớp đầu ra của mạng tới chính nó hay tới bản thân mạng.

Định nghĩa một hàm ánh xạ nhiều-sang-một $\mathcal{B} : L'^T \mapsto L^{\leq T}$ là tập các cách để đánh nhãn (có nghĩa là tập các chuỗi có độ dài bé hơn hoặc bằng T qua bảng chữ cái L). Việc tìm các chuỗi đánh nhãn này đơn giản bằng cách loại bỏ tất cả ký tự "blank" và gộp những nhãn trùng nhau lại thành một (ví dụ: $\mathcal{B}(a - ab -) = \mathcal{B}(-aa - -abb) = abb$). Nhờ vào hàm ánh xạ này, 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 đánh nhãn $\mathbf{l} \in L^{\leq T}$ là tổng xác suất của các đường dẫn tương ứng với nó:

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}} p(\pi|\mathbf{x}) \tag{4}$$

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

$$h(\mathbf{x}) = \text{argmax}_{\mathbf{l} \in L^{\leq T}} \, p(\mathbf{l}|\mathbf{x}) \tag{5}$$

Lúc này, ta có thể giả định chuỗi tốt nhất (giống với đầu vào nhất) là sự kết hợp của các nhãn có giá trị xác suất cao nhất tại mỗi thời điểm:

$$\begin{gathered} h(\mathbf{x}) \approx \mathcal{B}(\pi^*) \\ \text{mà } \pi^* = \text{argmax}_{\pi \in N^t} \, p(\pi|\mathbf{x}) \end{gathered} \tag{6}$$

Đương nhiên giả định này chưa chắc đúng, một cách khác để lấy chuỗi tối ưu hơn là dùng prefix search decoding, ví dụ như Beam Search [63]. Trong luận văn này, tôi sử dụng phương pháp như công thức (6), gọi là phương pháp tham lam (greedy) và cả Beam Search để thực hiện tìm kiếm chuỗi âm vị của câu nói.

2.4. Khung mô hình học biểu diễn giọng nói tự giám sát: Wav2Vec2.0

2.4.1. Tổng quan

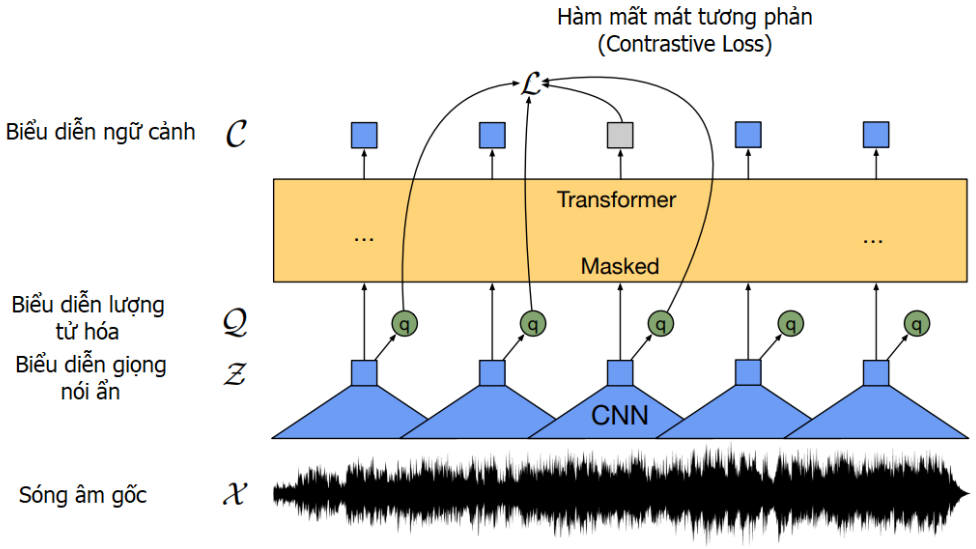
Neural Network rất được lợi từ việc huấn luyện trên một lượng lớn dữ liệu. Tuy nhiên, trong một số trường hợp thì dữ liệu có nhãn thường khó kiếm hơn dữ liệu không đánh nhãn: một hệ thống nhận dạng giọng nói hiện đại yêu cầu hàng nghìn giờ dữ liệu giọng nói đã được đánh nhãn một cách cẩn thận lại không thể thực hiện được cho hơn 7000 ngôn ngữ nói trên thế giới [64]. Việc chỉ học hoàn toàn trên các mẫu đã được đánh nhãn không giống với việc tiếp thu ngôn ngữ ở con người: trẻ em học ngôn ngữ bằng cách lắng nghe người lớn nói xung quanh chúng – một quá trình yêu cầu phải học được một cách biểu diễn tốt của lời nói.

Trong học máy, học tự giám sát (self-supervised learning) đã nổi lên như một mô hình học dữ liệu đại diện một cách tổng quát từ dữ liệu không đánh nhãn và sau đó

đem mô hình đi fine-tuning trên dữ liệu có nhãn. Điều này đã được chứng minh bằng những nghiên cứu rất thành công trong lĩnh vực Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) [65] [66] [67] và vẫn đang là một hướng nghiên cứu mở cho Thị giác máy tính (Computer Vision – CV) [68] [69] [70] [71] [72].

A. Baevski cùng cộng sự đã trình bày về wav2vec2.0 [42] vào năm 2020, đây là một khung mô hình thuộc tự giám sát để học được cách biểu diễn từ dữ liệu âm thanh gốc. Phương pháp này mã hóa các âm thanh giọng nói thông qua CNN nhiều lớp, sau đó dùng mặt nạ để che đi các dữ liệu giọng nói đại diện này [73] [74], tương tự như mô hình hóa ngôn ngữ dùng mặt nạ [65]. Các không gian biểu diễn ẩn này sẽ được đưa vào Transformer để xây dựng biểu diễn theo ngữ cảnh, lúc này mô hình được huấn luyện theo tác vụ tương phản (Contrastive task – Contrastive Learnig) nhằm mục đích phân biệt giữa các không gian ẩn với các bộ phân tâm [75] [76] [77] [78].

Trong phần huấn luyện, mô hình này sẽ học các đơn vị giọng nói rời rạc (quantized representations) [79] [80] [81] [82] thông qua hàm kích hoạt gumbel softmax [83] [84] để đại diện cho biểu diễn ẩn (latent speech representations) trong tác vụ tương phản (Hình 6 mô tả rõ hơn điều này), việc lượng tử hóa này được tác giả nghiên cứu rằng hiệu quả hơn việc sử dụng một đại diện biểu diễn không được lượng tử. Sau khi tiền huấn luyện (pre-training) trên dữ liệu giọng nói không có nhãn, mô hình được điều chỉnh (fine-tune) trên một tập dữ liệu có nhãn với CTC (phần 2.3) để dùng cho tác vụ nhận dạng giọng nói về sau.



Hình 6. Minh họa Wav2Vec2.0 và cách thức học của mô hình này

Các nghiên cứu trước nghiên cứu này thường là cố gắng học một cách lượng tử hóa của dữ liệu, theo sau là một bộ học đại diện ngữ cảnh với một mô hình có cơ chế tự chú ý [83] [85], còn phương pháp tác giả trình bày là một mô hình đầu cuối. Việc dùng cơ chế mặt nạ để che đi đầu vào cho Transformer trong mạng wav2vec2.0 cho giọng nói cũng đã được nghiên cứu từ trước [73] [85], tuy nhiên các nghiên cứu trước phụ thuộc vào mô hình có luồng chạy là hai bước hay mô hình của họ được huấn luyện để tạo ra các tính năng có bộ lọc (filter bank value, ví dụ như MFCC). Một vài nghiên cứu khác học cách biểu diễn tự động mã hóa dữ liệu đầu vào [86] [87] hoặc dự đoán trực tiếp các time-step trong tương lai [88].

Kết quả của nghiên cứu của A. Baevski cùng cộng sự về Wav2Vec2.0 đã cho thấy rằng việc học cách biểu diễn các đơn vị giọng nói rời rạc với biểu diễn ngữ cảnh (contextualized representations) đưa ra kết quả đáng kể hơn việc học một số lượng đơn vị cố định trong nghiên cứu của ông từ trước [85].

2.4.2. Kiến trúc

2.4.2.1. Cấu tạo mô hình

Mô hình Wav2Vec2.0 được kết hợp bởi nhiều lớp encoder tích chập (gọi là feature encoder) $f : \mathcal{X} \mapsto \mathcal{Z}$ mà nhiệm vụ của nó là sẽ nhận âm thanh thô $\mathcal{X}$ và chuyển thành biểu diễn giọng nói ẩn $\mathbf{z}_1, \cdots, \mathbf{z}_T$ cho T time-step. Sau đó nó đưa dữ liệu này

vào Transformer $g : \mathcal{Z} \mapsto \mathcal{C}$ để xây dựng $\mathbf{c}_1, \cdots, \mathbf{c}_T$ nhằm nắm được thông tin cả một chuỗi [85] [83] [65]. Đầu ra của feature encoder sẽ được rời rạc hóa thành $\mathbf{q}_t$ với mô-đun lượng tử hóa $\mathcal{Z} \longmapsto \mathcal{Q}$ để biểu diễn mục tiêu (Mô tả trong hình 6) trong mục tiêu của bài toán tự giám sát.

Feature Encoder: Encoder này chứa một vài khối chứa các lớp tích chập theo thời gian đi cùng với một lớp Layer Normalization và hàm kích hoạt GELU. Chuỗi sóng âm thanh gốc đưa vào encoder được chuẩn hóa lại thành trung bình bằng 0 và phương sai là 1. Tổng số lượng stride của encoder xác định số lượng time-step T, để rồi sau đó đưa vào Transformer.

Biểu diễn ngữ cảnh với Transformer: Đầu ra của feature encoder sẽ được đưa vào một mạng ngữ cảnh, mà mạng này sẽ là Transformer [65] [89] [81]. Thay vì cố định positional embeddings để biểu diễn thông tin vị trí như trong Transformer gốc, tác giả sử dụng một lớp tích chập tương tự như trong các nghiên cứu [85] [90] [91] để xem như relative positional embedding. Đầu ra của lớp tích chập này sẽ cộng với đầu vào cùng với hàm kích hoạt GELU và sau đó áp dụng thêm một lớp Layer Normalization.

Mô-đun lượng tử hóa: Đối với việc huấn luyện tự giám sát, mô hình đã được lượng tử hóa đầu ra của feature encoder **z** thành một tập biểu diễn giọng nói giới hạn thông qua lượng tử hóa tích (product quantization) [92]. Sự lựa chọn này đưa ra kết quả tốt trong nghiên cứu trước đó [83] mà cụ thể là học cách biểu diễn các đơn vị rời rạc sau đó kết hợp thêm học biểu diễn ngữ cảnh. Lượng tử hóa tích tương đương với việc lựa chọn các cách biểu diễn lượng tử từ nhiều bộ codebook và kết nối lại với nhau. Ta có G codebook hoặc là nhóm, với V mục $\mathrm{e} \in \mathbb{R}^{\mathrm{V} \times \mathrm{d}/\mathrm{G}}$, ta sẽ chọn một mục trong mỗi codebook và kết nối lại với nhau thành các vector $\mathrm{e}_1, \cdots, \mathrm{e}_\mathrm{G}$ và thực hiện một biến đổi tuyến tính $\mathbb{R}^{\mathrm{d}} \longmapsto \mathbb{R}^{\mathrm{f}}$ để tạo ra được $\mathbf{q} \in \mathbb{R}^{\mathrm{f}}$.

Gumbel softmax cho phép lựa chọn các mục rời rạc của codebook theo cách hoàn toàn khác nhau [93] [94] [95]. Tác giả sử dụng ước lượng trực tiếp [73] và cài đặt G toán tử Gumbel softmax cứng (hard Gumbel softmax operations) [94]. Đầu ra của

feature encoder **z** được ánh xạ tới các logit $\mathbf{l} \in \mathbb{R}^{G \times V}$ và xác suất cho việc chọn mục thứ v của codebook thứ g là:

$$p_{g,v} = \frac{\exp\left(l_{g,v} + n_v\right)}{\sum_{k=1}^{V} \exp \frac{\left(l_{g,k} + n_k\right)}{\tau}} \tag{7}$$

Mà τ là giá trị nhiệt độ (temperature) không âm, $n = -\log(-\log(u))$ và u được lấy mẫu từ phân phối đều $\mathcal{U}(0,1)$. Trong quá trình lan truyền xuôi (forward pass), mục thứ i được chọn khi $i = \text{argmax}_j p_{g,j}$. Giá trị đạo hàm thực của Gumbel softmax sẽ được sử dụng trong quá trình đạo hàm cho lan truyền ngược.

2.4.2.2. Quá trình huấn luyện

Để dùng mô hình này cho quá trình tiền huấn luyện, ta cần phải che đi một tỷ lệ các time-step nhất định của feature encoder, tương tự như mask language modeling của BERT [65]. Mục tiêu huấn luyện sẽ yêu cầu xác định chính xác vector lượng tử hóa ẩn cho biểu diễn âm thanh trong một tập các bộ phân tâm cho mỗi time-step bị che. Mô hình sau khi tiền huấn luyện có thể điều chỉnh trên dữ liệu có nhãn.

Masking hay che dữ liệu: Để huấn luyện, mô hình sẽ phải dùng cơ chế mặt nạ che đi một tỷ lệ đầu ra của feature encoder, 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 feature vector được học và chia sẻ vector này cho tất cả các time-step bị che, nhưng feature encoder sẽ không bị che khi đi qua mô-đun lượng tử hóa. Để che đầu ra của feature encoder, 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.

Hàm mục tiêu: Trong quá trình tiền huấn luyện, mô hình học các biểu diễn các âm thanh giọng nói bằng cách giải quyết tác vụ tương phản $\mathcal{L}_m$ mà yêu cầu của nó sẽ là 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 hàm mất mất đa

dạng codebook $\mathcal{L}_d$ để khuyến khích mô hình sử dụng các mục trong codebook thường xuyên như nhau (sử dụng đều các mục trong codebook).

$$\mathcal{L} = \mathcal{L}_m + \alpha\mathcal{L}_d \tag{8}$$

α là tham số có thể điều chỉnh được.

Ở hàm mục tiêu số (8), ta thấy được có hai thành phần là hàm mất mát tương phản (contrastive loss) và hàm mất mát đa dạng (diversity loss).

1. Contrastive Loss: Đưa đầu ra của mạng $\mathbf{c}_t$ bị che tại thời điểm t, mô hình cần phải xác định chính xác biểu diễn lượng tử hóa ẩn $\mathbf{q}_t$ trong một tập $K+1$ các ứng viên biểu diễn đã được lượng tử hóa $\tilde{\mathbf{q}} \in \mathbf{Q}_t$, mà bao gồm $\mathbf{q}_{\mathrm{t}}$ và K bộ phân tâm [81] [96]. *Bộ phân tâm* thường được lấy mẫu đều từ các time-step bị che khác của cùng một câu nói. Hàm mất mát định nghĩa như sau:

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_{\mathrm{t}}, \mathbf{q}_{\mathrm{t}})/K)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_{\mathrm{t}}, \tilde{\mathbf{q}}_{\mathrm{t}})/K)} \tag{9}$$

   Mà $sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$ là thang đo độ tương đồng cosine similarity giữa biểu diễn ngữ cảnh và biểu diễn lượng tử hóa ẩn [69] [70].
2. Diveristy Loss: Tác vụ tương phản phụ thuộc vào các codebook để biểu diễn cả những mẫu dương (positive) và mẫu âm (negative), và hàm mất mát đa dạng $\mathcal{L}_d$ này được thiết kế để tăng việc sử dụng các biểu diễn codebook đã được lượng tử hóa [97]. Điều này khiến mô hình được khuyến khích hơn trong việc sử dụng các mục V một cách đồng đều bằng cách tối đa hóa giá trị entropy của trung bình softmax phân phối $\mathbf{l}$ qua các mục trong codebook cho mỗi codebook $\overline{p_g}$ qua một batch các câu nói. Hàm mất mát có phân phối softmax không chứa nhiễu từ Gumbel hay tham số temperature:

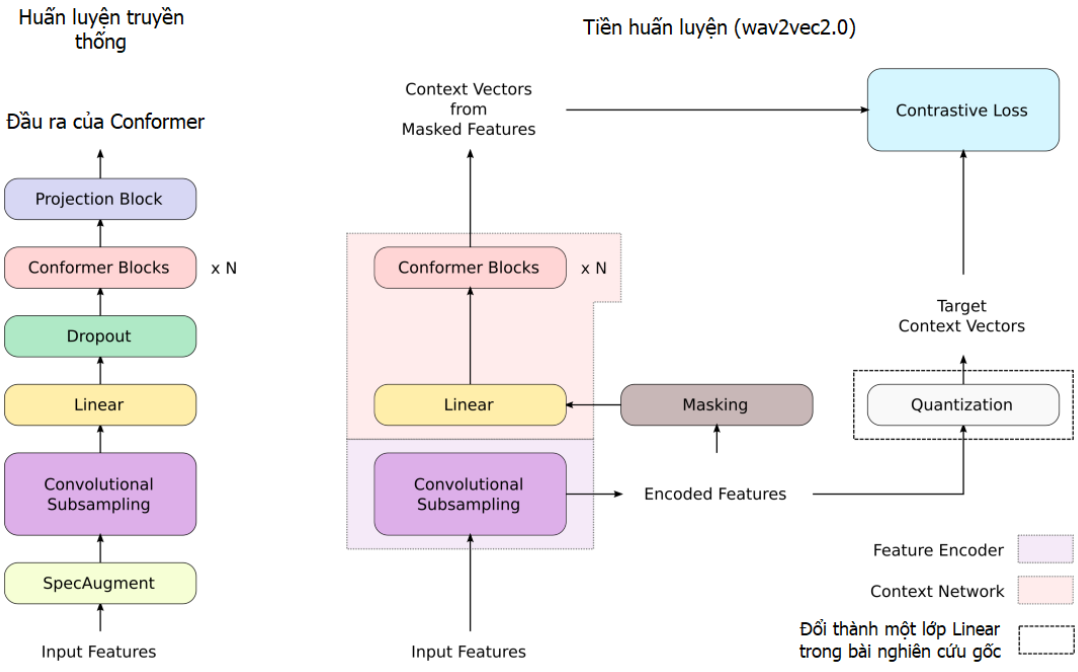
$$\mathcal{L}_d = \frac{1}{GV}\sum_{g=1}^{G} -H(\overline{p_g}) = \frac{1}{GV}\sum_{g=1}^{G}\sum_{v=1}^{V} \overline{p_{g,v}} \log \overline{p_{g,v}} \tag{10}$$

2.4.2.3. Điều chỉnh sau khi tiền huấn luyện

Mô hình Wav2Vec2.0 chỉ cung cấp một khung mô hình để tiền huấn luyện đối với dữ liệu không có nhãn. Để sử dụng ta có thể thêm một lớp Linear có đầu ra là softmax trên bảng chữ cái *L* và huấn luyện dùng hàm mất mát CTC. Hoặc có thể lấy bộ biểu diễn ngữ cảnh (context representation) bên trong Wav2Vec2.0 ra để gắn vào một kiến trúc khác để tận dụng bộ biểu diễn giọng nói đã được huấn luyện dùng Wav2Vec2.0. Trong bài này, tôi tận dụng cách sử dụng thứ hai, lấy bộ biểu diễn ngữ cảnh ra và đem đi huấn luyện dùng CTC. Nhưng thay vì các khối Transformer mô hình Wave2Vec2.0 dùng các khối Conformer, dựa theo [22].

2.4.2.4. Wav2Vec2.0 Conformer

Y. Zhang và cộng sự đã nghiên cứu cách để đẩy giới hạn của mô hình Conformer trong bài toán nhận dạng giọng nói lên một mức cao hơn, khi tận dụng cả tiền huấn luyện lẫn tự huấn luyện (self-training) trong tác vụ học bán giám sát [22]. Hình 7 mô tả sơ đồ bài toán của họ.



Hình 7. Cấu trúc Conformer truyền thống và Wav2Vec2.0 Conformer ở trong (thay cho Transformer) ở bài nghiên cứu của Y. Zhang và cộng sự

Trong bài này, tôi thừa kế nghiên cứu của họ để tận dụng Conformer sau khi trải qua bước tiền huấn luyện với dữ liệu giọng nói không có nhãn, sau đó dùng lại trọng số của Conformer đã được tiền huấn luyện (chủ yếu là N khối Conformer) cho Encoder Conformer như đã trình bày trong phần *2.2.2.6* rồi thực hiện tác vụ tự huấn luyện dùng Noisy Student Training.

2.5. Huấn luyện bán giám sát với Noisy Student Training

2.5.1. Học có giám sát

2.5.1.1. Vấn đề của dữ liệu huấn luyện được gắn nhãn

Những mô hình và kỹ thuật Machine Learning ngày càng dễ tiếp cận hơn đối với các nhà nghiên cứu và lập trình viên. Nó được trọng dụng bởi vì tính hữu ích thực tế của những mô hình này, sự hữu ích này còn phụ thuộc vào nguồn dữ liệu huấn luyện được gắn nhãn chất lượng cao. Việc đòi hỏi một lượng dữ liệu huấn luyện được gắn nhãn lớn khiến điều này trở thành cản trở đối với việc ứng dụng các mô hình Machine Learning trong các tổ chức hay công nghiệp. Sự hạn chế này có ở nhiều khía cạnh, bao gồm những ví dụ dưới đây:

1. *Không đủ số lượng dữ liệu được gắn nhãn*: Khi kỹ thuật Machine Learning mới được áp dụng trong công nghiệp, thường thì chưa có đủ dữ liệu để áp dụng quy trình truyền thống. Một số ngành có sẵn dữ liệu huấn luyện có giá trị hàng chục năm, một vài ngành khác thì không có sẵn như thế. Trong những trường hợp như thế này, việc có được dữ liệu huấn luyện có thể là không thực tế, đắt đỏ hay không thể có mà không chờ đợi hàng chục năm tích lũy.
2. *Không đủ người có chuyên môn để đánh nhãn dữ liệu*: Khi việc đánh nhãn dữ liệu huấn luyện đòi hỏi những kiến thức chuyên biệt, việc tạo ra hay đánh nhãn dữ liệu một cách nhanh chóng trở nên cực kỳ khó khăn [98]. Vấn đề này thường xảy ra trong các ứng dụng Machine Learning liên quan đến Y Sinh hoặc Bảo Mật.
3. *Không đủ thời gian để gắn nhãn và chuẩn bị dữ liệu*: Hầu hết thời gian để thực hiện một dự án Machine Learning thuộc khâu chuẩn bị dữ liệu [98]. Khi

các ngành công nghiệp hay nhóm nghiên cứu giải quyết một vấn đề mà đòi hỏi mức độ nhanh chóng, tăng trưởng nhanh thì việc có được một bộ dữ liệu hay quá trình thu thập dữ liệu nhanh chóng là điều hầu như không thể. Vấn đề này thường xảy ra trong bài toán Phát hiện gian lận hay An ninh mạng.

Vì thế, ngoài kỹ thuật học có giám sát thông thường phải đòi hỏi lượng dữ liệu được gắn nhãn lớn, các lĩnh vực Machine Learning khác cũng được thúc đẩy bởi nhu cầu tăng số lượng và chất lượng đào tạo. Bao gồm Học chủ động (active learning), Học chuyển giao (Transfer learning) hay Học bán giám sát (Semi-supervised learning).

2.5.1.2. Giám sát yếu

Giám sát yếu hay *Weak Supervision* là một nhánh của Machine Learning, mà các nguồn nhiễu, bị giới hạn hay không chính xác được sử dụng để làm nhãn (label) cho một lượng lớn dữ liệu huấn luyện trong phương diện học có giám sát [99]. Cách tiếp cận này giảm bớt gánh nặng của việc thu thập dữ liệu có nhãn thủ công (hand-labeled data sets) vì loại dữ liệu có nhãn rất tốn kém hoặc không thực tế để thu thập. Thay vì thế, việc sử dụng những nhãn yếu nhưng rẻ với giả định rằng việc sử dụng nó sẽ không hoàn hảo, nhưng dù sao cũng sẽ tạo ra được một mô hình dự đoán mạnh mẽ [100] [101] [102].

Nhãn yếu: Những nguồn nhãn yếu được tạo ra nhằm mục đích giảm thiểu chi phí và công sức và nỗ lực của con người trong việc đánh nhãn thủ công. Nhãn yếu có rất nhiều dạng, và có thể được chia thành các loại sau:

1. *Thống kê toàn cục về một nhóm dữ liệu đầu vào*: Loại nhãn này ám chỉ việc xem xét thông tin toàn cục trên một nhóm mẫu. Ví dụ: biết được một nửa số nhãn của tập mẫu dữ liệu cho trước, ta có thể dùng thông tin đó để làm mẫu yếu. Một số ví dụ trong giám sát thống kê toàn cục (global statistics supervision) bao gồm học nhiều phiên bản (multiple-instance learning) [103] hay học từ tỉ lệ nhãn [104].
2. *Bộ phân loại yếu*: Weak classifier là cách tiếp cận thứ hai để tạo ra nhãn yếu, bao gồm việc giả định các mô hình phân loại yếu sẽ tương quan thấp với mô

hình cần huấn luyện. Những mô hình phân loại này sẽ mô hình hóa nhãn từ các nền tảng cộng đồng, chuyên gia, các phép đo có nhiễu hay những quy tắc lập trình. Tổng quát hơn, lập trình viên sẽ được lợi từ các nguồn có sẵn (như là nguồn tri thức, nguồn dữ liệu khác, hay mô hình đã tiền huấn luyện) để tạo ra nhãn có ích, mặc dù rằng nó không hoàn hảo cho vấn đề muốn giải quyết hiện tại.

3. *Nhãn không đầy đủ*: Thứ ba, giám sát yếu có thể được hiểu là tiếp cận đến nguồn tri thức một phần trên mỗi nhãn. Tri thức một phần này có thể được hiểu là một quy trình bị ngắt [105]. Trong một vài trường hợp, các quan sát một phần còn có thể trở thành một tập các nhãn tiềm năng mà tương thích với quan sát một phần này, mà đây là một trong những tính chất của giám sát một phần (partial supervision) [106] [107]. Giám sát một phần là phương pháp tổng quát hóa của học bán giám sát, và đây cũng là phương pháp cổ điển trong việc tiếp cận với bài toán để vượt qua việc thiếu dữ liệu có nhãn đầy đủ.

2.5.1.3. 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, xác định dầu có ở vị trí cụ thể hay không, hay là phiên âm theo bảng mã IPA một đoạn âm thanh giọng nói tiếng Anh như trong luận văn này). 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 l các mẫu $x_1, \cdots, x_l \in \boldsymbol{X}$ với nhãn tương ứng $y_1, \cdots, y_l \in \boldsymbol{Y}$ và u mẫu dữ liệu không có nhãn $x_{l+1}, \cdots, x_{l+u} \in \boldsymbol{X}$ 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 dùng Noisy Student Training

2.5.2.1. Tổng quan

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 [108], 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 nhãn S. Sau đó, với mỗi lần lặp, mô hình hiện tại tại lựa chọn một phần dữ liệu không có nhãn $X_{\not u}$, 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 $S \cup X_{\not u}$.

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) 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 [43]. 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 có nhãn) lớn hơn, nhưng phải có lưu ý rằng, việc tăng kích thước đơn thuần sẽ không cải thiện kết quả, mà phải kết hợp với tự huấn luyện [109]. Thứ hai, NST cũng bao gồm việc thêm nhiễu vào mô hình student để ép student học khó hơn từ nhãn giả. Để làm nhiễu đầu vào, có nhiều cách bao gồm Dropout, Stochastic Depth [110] hay các phương pháp tăng cường dữ liệu [111].

Trong bài nghiên cứu [112], 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 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 state-of-the-art 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 tôi dựa vào ý tưởng của nghiên cứu này để thực hiện.

2.5.2.2. Kỹ thuật Noisy Student Training sửa đổi được sử dụng trong luận văn

Theo [22], họ sử dụng luồng huấn luyện NST cho ASR từ nghiên cứu [112] để huấn luyện các mô hình đã tiền huấn luyện dùng Wav2Vec 2.0. Trong khuôn khổ NST cho ASR, đối với mô hình teacher, họ sẽ gắn mô hình đã huấn luyện (Conformer có decoder là RNN-Transducer [28]) với mô hình ngôn ngữ đã huấn luyện trên tập nhãn của tập huấn luyện riêng để tạo nhãn giả tốt hơn cho dữ liệu không được tăng cường. Mô hình teacher sau khi tạo nhãn giả sẽ được thông qua bước lọc và cân bằng (filtering và balancing) sẽ được dùng để huấn luyện mô hình ASR thế hệ kế tiếp. Dữ liệu dùng cho mô hình student được tăng cường sử dùng Adaptive SpecAugment. Tuy nhiên, tác giả của nghiên cứu [22] lấy tất cả nhãn giả mà không lọc hay cân bằng, nhằm có nhiều dữ liệu hơn.

Theo luồng NST của [22], nếu gọi tập dữ có nhãn LibriSpeech là S, dữ liệu không có nhãn Libri-Light U và mô hình ngôn ngữ sẽ được huấn luyện trên bản dịch âm thanh của LibriSpeech, quy trình dưới đây là từ nghiên cứu của tác giả:

1. Điều chỉnh (fine-tune) mô hình đã tiền huấn luyện M_0 trên S với SpecAugment. Gán $M = M_0$.
2. Gắn M với mô hình ngôn ngữ và đo lường hiệu năng của mô hình.
3. Tạo nhãn giả $M(U)$ với mô hình đã được gắn mô hình ngôn ngữ.
4. Trộn lẫn $M(U)$ 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, bài toán của tôi khác với bài toán gốc nên trong luận văn này sẽ sử dụng quy trình khác. Bao gồm các ý sau:

1. Mô hình tác giả [22] sử dụng là Conformer với decoder là RNN Transducer, rất được lợi trong việc học từ một lượng lớn dữ liệu có nhãn (dữ liệu của họ lớn hơn trong bài luận văn này). Còn trong bài luận văn này tôi sẽ chỉ sử dụng Conformer với hàm mất mát CTC vì sự đơn giản nhưng hiệu quả của kiến trúc này.
2. Bỏ bước số (2) trong quy trình của tác giả, việc gắn thêm một language model vào sẽ khiến mô hình phụ thuộc vào ngữ cảnh ngôn ngữ, nên mô hình sẽ kém bị phụ thuộc vào ngữ cảnh giọng nói, mà cái này rất quan trọng trong việc xác định âm vị của người nói.
3. Phương pháp cài đặt NST của tác giả sẽ lặp đi lặp lại vài lần để đẩy độ chính xác lên mức cao nhất. Còn bài luận văn này sẽ chỉ lặp 1 lần, mô hình student sau khi huấn luyện xong sẽ là phiên bản cuối cùng, dùng để dự đoán chuỗi âm vị. Các mô hình student sau thường lớn hơn thế hệ trước và được huấn luyện trên các bộ dữ liệu lớn hơn, nên số lần lặp khá phụ thuộc vào độ lớn và khả năng của phần cứng.

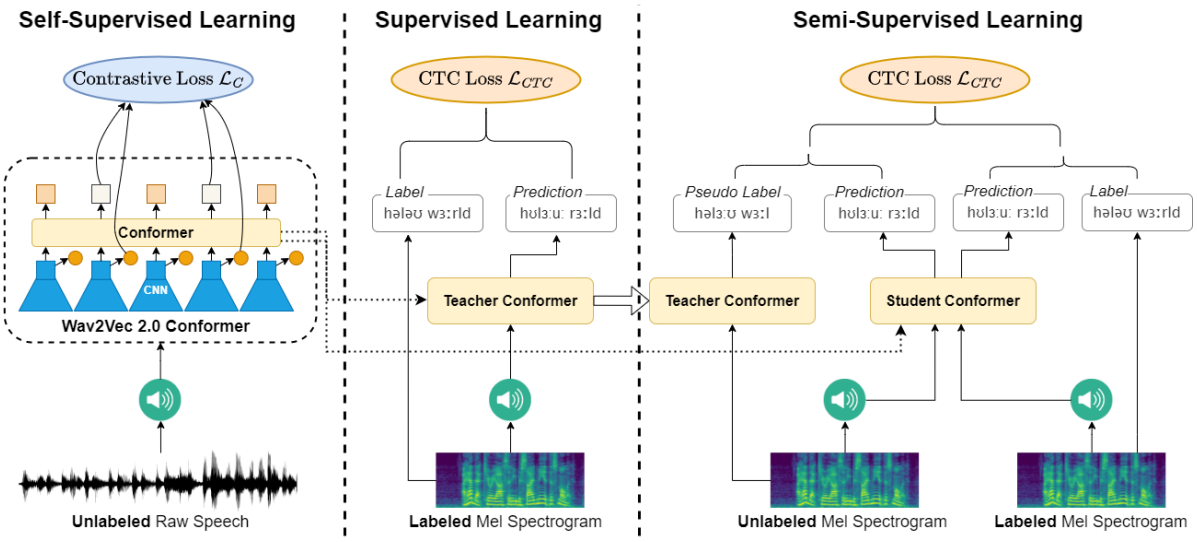
Vì thế, quy trình NST để huấn luyện mô hình student trong luận văn này sẽ trở thành:

1. Điều chỉnh (fine-tune) mô hình đã tiền huấn luyện M_{teacher} trên S với SpecAugment.
2. Tạo nhãn giả $M_{\text{teacher}}(U)$ với mô hình teacher M_{teacher}.
3. Trộn lẫn $M_{\text{teacher}}(U)$ và S.
4. Điều chỉnh mô hình tiền huấn luyện M_{student} với SpecAugment trên dữ liệu đã trộn lẫn.

Mô hình M_{student} sau khi huấn luyện sẽ là mô hình cuối cùng dùng để nhận diện chuỗi âm vị.

2.6. Kết hợp các kỹ thuật để thực hiện bài toán phát hiện chuỗi âm vị

Tổng quan lại, ý tưởng thực hiện của luận văn như sau: Huấn luyện mô hình teacher với dữ liệu có nhãn, sau đó dùng mô hình teacher để tạo ra nhãn giả trên tập dữ liệu không có nhãn nhưng dồi dào dữ liệu hơn. Từ đó, mô hình student sẽ được huấn luyện trên cả tập dữ liệu có nhãn bị giới hạn, kết hợp thêm tập dữ liệu có nhãn giả dồi dào để dùng cho downstream task dự đoán chuỗi âm vị. Nhưng trước đó, thay vì huấn luyện teacher và student từ đầu (khởi tạo ngẫu nhiên), các khối Conformer (thành phần chính của teacher và student) được tiền huấn luyện trước với bằng framework Wav2Vec2.0 với một lượng dữ liệu không có nhãn khổng lồ, vì thế, sẽ tạo ra các khối Conformer với trọng số đã được tối ưu lý tưởng trên các âm thanh giọng nói. Hình 8 trình bày về toàn bộ quy trình bài toán được sử dụng trong luận văn.



Hình 8. Sơ đồ cấu trúc luận văn. Đường "." mô tả rằng trọng số của teacher và student sẽ được khởi tạo bằng các khối Conformer được tiền huấn luyện bằng Wav2Vec2.0

2.7. Kỹ thuật đánh giá giọng nói

2.7.1. Đánh giá mô hình bằng Phoneme Error Rate

Để đánh giá hiệu năng dự đoán chuỗi âm vị của mô hình, tôi sẽ sử dụng Phoneme Error Rate (PER), được định nghĩa như Word Error Rate của chuỗi âm vị dự đoán ra và chuỗi âm vị thực tế. WER có thể tính bằng công thức (11) [113]:

$$\text{PER} = \text{WER} = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C} \quad (11)$$

Mà:

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

Các ký hiệu của việc thêm, xóa, sửa là cách để biến chuỗi dự đoán thành chuỗi âm vị thực tế. Giá trị PER này càng nhỏ càng tốt, được tính theo phần trăm %, tốt nhất là 0 và tệ nhất là 100%.

2.7.2. Phát hiện lỗi sai trong phát âm bằng thuật toán Tìm chuỗi con chung dài nhất

Bài toán tìm chuỗi con chung dài nhất (Longest Common Subsequence – LCS) là một bài toán tìm các ký tự liên tục dài nhất mà vẫn chung với các chuỗi được so sánh với nhau (thường là 2 – trong bài này là chuỗi thực tế và chuỗi dự đoán). Nó khác với bài toán chuỗi con dài nhất bởi vì các ký tự mà bài toán LCS tìm được nó sẽ không nhất thiết phải liên tục. Đây là một bài toán cổ điển trong khoa học máy tính, được ứng dụng nhiều trong hệ thống hay ngôn ngữ tính toán [114].

Gọi chuỗi âm vị thực tế là $X = (x_1, x_2, \cdots, x_n)$, chuỗi dự đoán là $Y = (y_1, y_2, \cdots, y_m)$. Tiền tố của X là $X_0, X_1, X_2, \cdots, X_n$; tiền tố của Y là $Y_0, Y_1, Y_2, \cdots, Y_m$. Gọi $LCS(X_i, Y_j)$ đại diện cho một tập chuỗi con chung dài nhất của các tiền tố X_i và Y_j. Tập chuỗi này có thể biểu diễn bằng công thức (12):

$$LCS(X_i, Y_j) = \begin{cases} \emptyset, & \text{khi } x = \text{hoặc } j = 0 \\ LCS(X_{i-1}, Y_{j-1}) + x_i, & \text{khi } i, j > 0 \text{ và } x_i = y_j \\ \max\left(LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)\right), & \text{khi } i, j > 0 \text{ và } x_i \neq y_j \end{cases} \quad (12)$$

Để tìm được LCS của X_i và Y_j, ta đi so sánh x_i và y_j. Nếu nó giống nhau, ta sẽ nở rộng $LCS(X_{i-1}, Y_{j-1})$ thêm phần tử x_i. Nếu không bằng nhau, ta sẽ so sánh độ dài của hai $LCS(X_i, Y_{j-1})$ và $LCS(X_{i-1}, Y_j)$, và chọn cái dài nhất để giữ lại.

Độ phức tạp của bài toán khi sử dụng thuật toán quy hoạch động là $O(nm)$.

Ta có thể mở rộng ý tưởng của bài toán để tìm được các âm vị của người nói bị sai. Để làm được như vậy, ta phải giả định chuỗi dự đoán từ mô hình là chuỗi âm vị mô tả đúng nhất giọng nói của người dùng. Lúc này, ta áp dụng thuật toán LCS để tìm chuỗi âm vị khớp nhau nhất, sau đó tìm ra những âm vị mà không được khớp (phần còn lại) và xuất ra kết quả đây là các vị trí mà người nói bị sai.

Dưới đây là mã giả cho LCS.

Pseudo Code

```
function LCS(s1, s2):
  n := size(s1)
  m := size(s2)
  L = tạo mảng 2D (0..n, 0..m) với mỗi phần tử là một mảng rỗng
  for i := 1 to :n
    for j := 1 to m:
      else if s1[i - 1] = s2[j - 1]:
        L[i][j] := thêm ký tự s1[i - 1] vào mảng L[i - 1][j - 1]
      else:
        // so sánh chiều dài của mỗi mảng, cái nào lớn hơn
        // thì lấy gán cho L[i][j]
        L[i][j] := max(L[i - 1][j], L[i][j - 1], key=size)
  return L[n][m]
s1 = "ɔ l ɪ z s ɛ d w ɪ ð aʊ t ə w ə d" độ dài n
s2 = "ɔ l w ɪ z s ɛ d w ɪ ð aʊ t ə w ə d" độ dài m
lcs = LCS(s1, s2)
print("Longest Common Phoneme Sequence:", lcs)
```

Output

```
Longest Common Phoneme Sequence: ɔ l ɪ z s ɛ d w ɪ ð aʊ t ə w ə d
```

CHƯƠNG 3: DỮ LIỆU

3.1. Libri-Light

Bộ dữ liệu Libri-Light, được trình bày năm 2020 của Facebook AI [115] là một tập các bộ âm thanh đọc sách người nói thích hợp cho các hệ thống nhận diện giọng nói mà cách cài đặt thuộc dạng giám sát giới hạn hay không có giám sát. Bộ dữ liệu này có nguồn gốc từ sách nói mã nguồn mở của dự án LibriVox. Nó chứa hơn 60 nghìn giờ dữ liệu âm thanh, mà theo họ được biết, đây là nguồn dữ liệu âm thanh miễn phí lớn nhất từ đó đến nay.

Bộ dữ liệu được chia thành 4 phần:

- Một tập huấn luyện với các âm thanh giọng nói không được đánh nhãn.
- Một tập huấn luyện có đánh nhãn nhưng giới hạn.
- Các tập dữ liệu âm thanh giọng nói đánh giá dev/test.
- Và một tập huấn luyện chứa dữ liệu chữ chưa được căn chỉnh.

3.1.1. Tập dữ liệu giọng nói dùng để huấn luyện không có nhãn

Tập dữ liệu này được lấy từ các tập tin âm thanh cho giọng nói tiếng Anh từ dự án LibriVox chứa các sách nói mã nguồn mở. Các tập tin được tải xuống và chuyển thành 16kHz dạng FLAC. Sau đó nhóm tác giả bỏ đi các tập tin bị hỏng, không biết các thông tin hay các tập tin thuộc LibriSpeech dev và test. Sau đó tập này được chia thành 3 tập nhỏ hơn: unlab-60k, unlab-6k, unlab-600.

Tên tập	Số giờ	Số sách	Số tập tin	Số giờ mỗi người nói	Tổng số người nói
unlab-60k	57706.04	9860	219041	7.84	7439
unlab-6k	5770.7	1106	21327	3.31	1742
unlab-600	577.2	202	2588	1.18	489

Bảng 1. Thông tin của tập huấn luyện không có nhãn của Libri-Light

3.1.2. Tập dữ liệu giọng nói giới hạn để huấn luyện có đánh nhãn

Nhằm phục vụ cho tác vụ giám sát có giới hạn, tác giả lựa chọn 3 tập của LibriSpeech: tập 10h, tập 1h và 6 tập 10 phút (6 tập này kết hợp với nhau để tạo ra

được tập 1h, tập 1h ở trong tập 10h). Trong mỗi tập, một nửa số lượng các câu nói được lấy từ tập huấn luyện clean và other. Tác giả dùng phonemizer [116] để tạo ra bản dịch chính tả và bản dịch âm vị từ bản dịch gốc.

Tên tập	Số giờ	Số sách	Số tập tin	Số giờ mỗi người nói	Tổng số người nói
train-10h	10	25	12	12	24
train-1h	1	2.5	12	12	24
train-10m	10 phút	2.5	2	2	4

Bảng 2. Thông tin tập dữ liệu giới hạn có nhãn của Libri-Light

3.1.3. Các tập dữ liệu giọng nói để đánh giá dev/test

Tập dữ liệu dev/test giống hệt như của LibriSpeech được đưa vào để cho mục đích đánh giá và điều chỉnh. Tất cả tập tin âm thanh thuộc tập dev và test được xóa bỏ khỏi tập huấn luyện. Các phoneme của tập dev và test cũng được tạo ra dùng phonemizer.

3.1.4. Tập chữ chưa căn chỉnh dùng để huấn luyện

Để huấn luyện mô hình ngôn ngữ cho cài đặt giám sát, tác giả sử dụng bộ dữ liệu dùng trong LibriSpeech, chứa hơn 800 triệu từ và bộ từ điển gồm 200 nghìn từ từ 14.5 nghìn sách công khai từ dự án Gutenberg.

3.2. LibriSpeech

Vào năm 2015, V. Panayotov và cộng sự đã trình bày một bộ dữ liệu dành cho bài toán nhận diện giọng nói, bộ dữ liệu tên là LibriSpeech [117]. Bộ dữ liệu này thích hợp cho việc huấn luyện và đánh giá các hệ thống nhận diện giọng nói.

Dữ liệu được lấy từ sách nói, là một phần của dự án LibriVox, chứa 1000 giờ giọng nói được lấy mẫu ở 16kHz. Bộ dữ liệu này được chia thành 2 phần: bộ huấn luyện (bao gồm train-clean-100, train-clean-360, train-other-500) và bộ đánh giá (các bộ dev/test clean/other).

Tên tập	**Số giờ**	**Số phút mỗi người nói**	**Số lượng nữ**	**Số lượng nam**	**Tổng số người nói**
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1106

Bảng 3. Thông tin các tập dữ liệu của LibriSpeech

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ

4.1. Cài đặt thực nghiệm

4.1.1. Dữ liệu

4.1.1.1. Dữ liệu huấn luyện

Để tiền huấn luyện Wav2Vec2.0 với các khối Conformer, bài nghiên cứu sử dụng bộ dữ liệu unlab-60k. Mô hình tiền huấn luyện Wav2Vec2.0 Conformer được lấy từ Github của Facebook Research, được huấn luyện trên bộ dữ liệu unlab-60k (xem mục 4.1.2).

Huấn luyện teacher sử dụng kết hợp các bộ dữ liệu của [train-10h + dev-clean + dev-test + test-other] = 25.8h dữ liệu có nhãn. Và tập train-clean-360 (loại bỏ nhãn) không nhãn kết hợp với tập huấn luyện có nhãn của teacher để huấn luyện bán giám sát, tổng cộng là 363.6h không nhãn và 25.8h có nhãn. Tỉ lệ giữa tập có nhãn và không nhãn trong huấn luyện student là 1: 14, trong khi các paper về NST cho ASR sử dụng tỉ lệ 1: 9 [22], 1:60 [112].

Giai đoạn huấn luyện	**Tên tập dữ liệu**	**Tổng số giờ âm thanh**	**Tỉ lệ có/không nhãn**
Tiền huấn luyện	- unlab-60k	57706.04	0: 1
Huấn luyện Teacher	- train-10h, dev-clean, dev-test, test-other	25.8	1: 0
Huấn luyện Student	- train-10h, dev-clean, dev-test, test-other - train-clean-360 (bỏ nhãn)	389.4	1: 14

Bảng 4. Thông tin mô tả dữ liệu dùng cho các giai đoạn huấn luyện

4.1.1.2. Dữ liệu đánh giá

Dữ liệu dùng cho việc đánh giá là tập test-clean, gồm 5.4h.

4.1.2. Tiền huấn luyện

Việc tiền huấn luyện Wav2Vec2.0 Conformer đòi hỏi kích thước mô hình đủ lớn để có thể học được 60 nghìn giờ từ unlab-60k. Nhưng trong luận văn này, việc có đủ kích cỡ phần cứng để huấn luyện là điều khó khăn. Nên tôi sử dụng mô hình đã được tiền huấn luyện và công bố công khai trên GitHub của Facebook Research, tên mô hình là *Wav2Vec 2.0 Large conformer - rel_pos (LV-60)* [118]. Các thông số của mô hình được trình bày trong bảng 5.

Loại tham số	**Giá trị**
Tầng Convolution [kích cỡ kernel, stride, channel đầu ra] × số lượng	$[10 \times 10, 5, 512] \times 1$ $[3 \times 3, 2, 512] \times 4$ $[2 \times 2, 2, 512] \times 2$
Số lượng khối Conformer	24
Kích thước các khối Conformer	1024
Kích thước lớp ẩn đầu ra, lớp lượng tử hóa	768
Tổng số lượng tham số mô hình	620 triệu

Bảng 5. Thông tin của mô hình Wav2Vec2.0 Conformer được lấy từ fairseq

4.1.3. Huấn luyện teacher và student

Sau khi bước tiền huấn luyện hoàn tất, sẽ thu được trọng số của các khối Conformer. Các khối này sẽ được đưa vào mô hình Conformer sửa đổi. Thông tin tham số được trình bày ở bảng dưới đây, hai mô hình teacher và student chỉ khác nhau ở số lượng khối Conformer (đây là thành phần quan trọng nhất).

Thông tin	**Teacher**	**Student**
Thông tin mô hình		
Kích thước đầu vào	128 (số lượng mel filterbank)	
Adaptive	- 2 freq mask.	

SpecAugment	- Kích thước freq mask $F = 27$. - Tỉ lệ time mask $p_s = 0.05$ - $\min\{10, \text{chiều dài MelSpectrogram} \times p_s\}$ số lượng time mask.	
Convolution Subsampling	$\text{Conv2D} - [3 \times 3, 2, 256] \times 2$	
Input Projection	- Kích thước đầu vào: 512 - Kích thước đầu ra: 1024	
Relative Positional Encoding	5000 vị trí	
Dropout Conformer	0.1	
Số lượng khối Conformer	**2**	**4**
Kích thước Conformer	1024	
Linear	- Đầu vào: 1024 - Đầu ra: 1024	
Hàm kích hoạt Swish		
Dropout	0.1	
Linear	- Đầu vào: 1024 - Đầu ra: 47 (số lượng các âm vị).	
Tổng số lượng tham số	≈ 57 triệu	≈ 107 triệu
Thời gian huấn luyện	5 giờ	51 giờ

Bảng 6. Thông tin các tham số của mô hình Conformer sửa đổi, cả của teacher và student

Dữ liệu âm thanh sau khi đọc lên sẽ được chuyển thành sạng Mel Spectrogram [119] với kích cỡ Fast Fourier Transform (n_fft) để tính là 1024 [120], số lượng mel filterbank (n_mels) [121] là 128 (cũng là kích cỡ đầu vào của mô hình), kích cỡ cửa sổ chạy khắp sóng âm thanh (win_length) là 40ms, kích cỡ một lần nhảy cửa sổ

(hop_length) là 20ms. Các tham số khác sẽ lấy giá trị mặc định của lớp MelSpectrogram của thư viện torchaudio [122]. Thông tin của Mel Spectrogram được trình bày ở bảng 7.

Tham số	**Giá trị**
n_fft	1024
n_mels	128
win_length	400
hop_length	200

Bảng 7. Các tham số của Mel Spectrogram

4.1.4. Mô hình ngôn ngữ

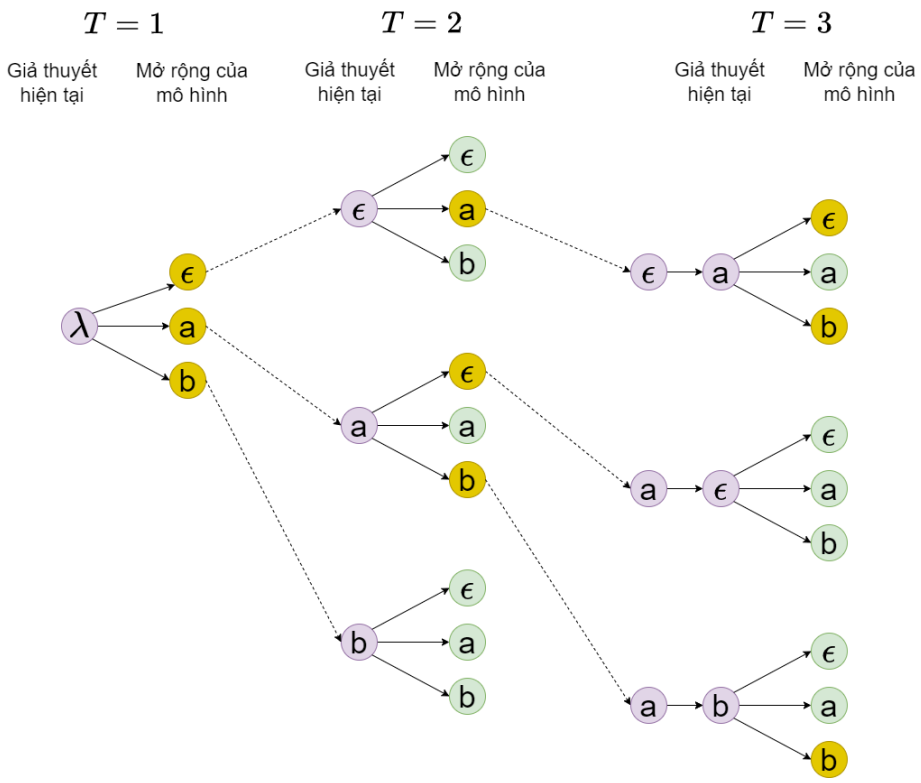
Ta có thể đặt ra giả định, để tối ưu kết quả dự đoán của mô hình thì sẽ cần một mô hình ngôn ngữ làm xác suất cộng thêm cho Beam Search [123] trong lúc dự đoán, công thức (13) mô tả mục tiêu lúc tìm chuỗi âm vị bằng Beam Search. Vì thế tôi đi huấn luyện mô hình ngôn ngữ 3-gram từ code của GitHub daandouwe/ngram-lm [124] theo phương pháp interpolate của Witten-Bell [125].

Dữ liệu dùng là toàn bộ các câu phiên dịch âm vị của tập huấn luyện giám sát cho teacher, được chia theo tỉ lệ $80:20$ cho tập huấn luyện và kiểm thử, thông tin được trình bày trong bảng 8, bao gồm perplexity [126] sau khi huấn luyện mô hình. Sau khi huấn luyện, mô hình ngôn ngữ sẽ đưa vào thư viện pyctcdecode [127] để dự đoán chuỗi âm vị có kết hợp Beam Search với tham số mặc định: kích thước beam (beam width) là 100, $\alpha = 0.5$ và $\beta = 1.5$.

Tập dữ liệu	**Số lượng câu**	**Perplexity**
Huấn luyện 80%	8906	10.17
Kiểm thử 20%	2241	10.47

Bảng 8. Perplexity trên hai tập dữ liệu âm vị của mô hình ngôn ngữ 3-gram

$$Q(c) = \log\big(P(c|x)\big) + \alpha \log\big(P_{lm}\,(c)\big) + \beta word\_count(c) \qquad (13)$$



Hình 9. Mô tả Beam Search qua 3 time-step, với Beam Width là 3 và số lượng thành phần trong bộ từ vựng là 3

Công thức (13) [128] mô tả rằng Beam Search sẽ tìm chuỗi âm vị tối đa hóa giá trị $Q(c)$, trên xác suất đầu ra của CTC $P(c|x)$ và xác suất từ mô hình ngôn ngữ $P_{lm}(c)$ cộng thêm số lượng các âm vị của chuỗi $word\_count(c)$ [129]. Hình 9 mô tả Beam Search với Beam Width là 3.

4.1.5. Phần cứng được xử dụng

Cả hai mô hình teacher và student được huấn luyện trên 2 GPU RTX 3090, thời gian huấn luyện được trình bày trong bảng 6.

Mô hình ngôn ngữ được huấn luyện trên 1 GPU RTX 3090 mất 1 phút.

4.2. Kết quả

4.2.1. Kết quả dự đoán chuỗi âm vị

Mô hình student sẽ là mô hình cuối cùng của bài toán, kết quả của mô hình student và teacher được trình bày trong bảng 9 dưới đây. Khi dự đoán, lớp SpecAugment sẽ được bỏ qua.

Mô hình	Greedy	Beam Search + LM
Teacher	17.13	22.31
Student	12.66	25.45

Bảng 9. Kết quả dự đoán chuỗi âm vị của teacher và student, có hay không có mô hình ngôn ngữ được tính theo PER (%)

Kết quả cho thấy mặc dù mô hình ngôn ngữ rất phù hợp cho bài toán ASR [6] [20] [22] [24] [30] [31] [34], nhưng áp dụng vào bài toán này thì lại bị tăng PER, không hiệu quả bằng phương pháp decode greedy (lựa chọn âm vị có xác suất cao nhất mỗi time-step). Điều này có thể giải thích bằng: lượng dữ liệu chữ của các bài nghiên cứu trên lớn, ngữ cảnh ngôn ngữ của mô hình ngôn ngữ được học rất hiệu quả, nên khi ứng dụng vào sẽ đạt kết quả tốt. Còn trong bài nghiên cứu này chỉ có xấp xỉ 9000 câu, ngữ cảnh của âm vị cũng rất khác so với chữ, perplexity trên tập test cũng khá cao, khoảng 10.47, nên có thể thấy ứng dụng mô hình ngôn ngữ vào bài toán này không hiệu quả.

Kết quả còn cho thấy thêm một điểm quan trọng: PER của student giảm tương đối 26% khi so với teacher, là mô hình đã tạo ra nhãn yếu cho student học, **thế nên kỹ thuật huấn luyện này có thể kết luận là thành công. Theo hiểu biết của tôi, chưa có nghiên cứu nào ứng dụng kỹ thuật học bán giám sát này cho bài toán dự đoán chuỗi âm vị. Vì thế đây có thể là tiền đề cho các nghiên cứu tương lai về kỹ thuật này.**

Dưới đây là một số kết quả đự đoán chuỗi âm vị trên tập test-clean, những âm vị nào bị sai sẽ được tô đỏ trên chuỗi dự đoán, tô xanh dương trên chuỗi thực tế nếu bị thiếu.

- Ví dụ 1, PER: 12.5%
  - *Thực tế:* w aɪ ə t ʌ ŋ ɪ m p ɹ ɛ s d w ɪ ð h ʌ n i f ɹ ʌ m ɛ v ɹ i w ɪ n d
  - *Dự đoán:* w aɪ ə t t ɑː ŋ ɪ m p ɹ ɛ s t w ɪ ð h ʌ n i f ɹ ʌ m ɛ v ɹ i w ɪ n d ɪ
- Ví dụ 2, PER: 0%
  - *Thực tế:* w aɪ ə n ɪ ɹ ə w ə l p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n
  - *Dự đoán:* w aɪ ə n ɪ ɹ ə w ə l p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n

- Ví dụ 3, PER: 6.25%
  - *Thực tế:* ɔ l ɪ z s ɛ d w ɪ ð aʊ t ə w ə d
  - *Dự đoán:* ɔ l w ɪ z s ɛ d w ɪ ð aʊ t ə w ə d
- Ví dụ 4, PER: 10.2%
  - *Thực tế:* aɪ s ɪ t b ə n i θ ð aɪ l ʊ k s æ z t ʃ ɪ l d ɹ ə n d uː ɪ n ð ə n uː n s ʌ n w ɪ ð s oʊ l z ð æ t t ɹ ɛ m b l θ ɹ uː ð ɛ ɹ h æ p i aɪ l ɪ d z f ɹ ʌ m ə n ʌ n ə v ə d j ɛ t p ɹ ɑː d ɪ g l ɪ n w ɚ d d ʒ ɔɪ
  - *Dự đoán:* aɪ s ɪ t b ə n i θ aɪ l ʊ k s æ z t ʃ ɪ l d ɹ ə n d uː ɪ n ð ə n uː n s ʌ n w ɪ ð s oʊ l z ð æ t ɹ ɛ m b l θ ɹ uː ð ɛ ɹ h æ p i aɪ ə l ɪ z f ɹ ʌ m ə n ʌ n ʌ v ə d j ɛ t p ɹ ɑː n ɪ k l ɪ n w ɚ d d ʒ ɔɪ ə ɪ ɪ
- Ví dụ 5, PER: 5.66%
  - *Thực tế:* aɪ l ʌ v ð i f ɹ i l i æ z m ɛ n s t ɹ aɪ v f ɔː ɹ ɹ aɪ t aɪ l ʌ v ð i p j ʊɹ l i æ z ð eɪ t ə n f ɹ ʌ m p ɹ eɪ z
  - *Dự đoán:* aɪ l ʌ v ð i f ɹ i l i æ z m ɛ n s t ɹ aɪ f f ɔː ɹ ɹ aɪ d aɪ l ʌ v ð i p j ʊɹ d l i æ z ð eɪ t ə n f ɹ ʌ m p ɹ eɪ z
- Ví dụ 6, PER: 10%
  - *Thực tế:* d ɪ ɹ ɪ s t t i t ʃ m i s oʊ t ə p o ɹ aʊ t g ɹ æ ɾ ɪ t uː d æ z ð aʊ d ɑː s t g ʊ d
  - *Dự đoán:* d ɪ ɹ ɪ s t t i t ʃ m i s oʊ t ə p o ɹ aʊ t k ɹ æ d ɪ t uː d æ z ð aʊ d ʌ s t g ʊ t

4.2.2. Kết quả dự đoán lỗi sai trong câu nói

Để kiểm tra khả năng phát hiện lỗi sai của mô hình, ta cần phải giả định chuỗi âm vị được dự đoán từ mô hình là *ground truth*, nghĩa là mô hình luôn mô tả đúng nhất giọng nói của người đọc (mặc dù ở bảng 9, ta thấy mô hình sai khoảng 12%) thì mới dùng thuật toán LCS để so khớp với chuỗi dự đoán với chuỗi thực tế được.

Để thực hiện đánh giá lỗi sai, ta sẽ dựa trên một mẫu trong bộ kiểm thử test-clean mà mô hình có khả năng dự đoán chính xác hoàn toàn, nghĩa là PER 0%, để mà tạo ra các phiên bản sửa đổi đọc sai của mẫu này. Bảng 10 dưới đây là thông tin của mẫu đó.

Thông tin	Giá trị
Dạng chữ cái	why an ear a whirlpool fierce to draw creations in
Dạng phiên âm IPA	w aɪ ə n ɪ ɹ ə w ə l p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n
Độ dài chuỗi phiên âm	32
Đường dẫn file đến bộ test-clean	test-clean/908/157963/908-157963-0030.flac

Bảng 10. Thông tin mẫu giọng đọc dành cho phần phát hiện lỗi sai

Để thực hiện kiểm tra lỗi sai, tôi sẽ lần lượt thay đổi dạng chữ cái của câu mẫu thành một từ khác, sau đó tạo lại dạng phiên âm IPA dùng phonemizer (như của tác giả) và dùng để phát hiện lỗi (so khớp dùng LCS), rồi dùng mô hình text-to-speech FastPitch của Nvidia để tạo giọng đọc [130]. Bảng 11 trình bày các mẫu thử được chỉnh sửa. Những vị trí được tô màu xanh dương là vị trí chỉnh sửa, màu đỏ là bị xóa đi. Các mẫu âm thanh có thể được nghe ở tham khảo này [131].

Tên mẫu	Dạng chữ cái	Chuỗi phiên âm gốc	Độ dài chuỗi phiên âm
error_1	why an ear a weir pool fierce to draw creations in	w aɪ ə n ɪ ɹ ə w ɪ ɹ p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	32
error_2	why an ear a whirlpool fear to draw creations in	w aɪ ə n ɪ ɹ ə w ə l p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	31
error_3	why an ear a whirl pole fierce to draw creations in	w aɪ ə n ɪ ɹ ə w ə l p oʊ l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	32

Bảng 11. Các mẫu được chỉnh sửa so với mẫu gốc nhằm mục đích đánh giá giọng nói

Cách đánh giá: một mẫu được coi là phát hiện lỗi sai đúng nếu mô hình phát hiện được lỗi sai (những chỗ màu xanh dương) hoặc độ dài đúng (cho mẫu mất chữ). Bảng 12 trình bày kết quả phát hiện lỗi bằng chuỗi âm vị dự đoán từ mô hình và đưa qua LCS. Những âm không được khớp bằng LCS sẽ được tô màu xanh dương.

Tên mẫu	Chuỗi phiên âm được dự đoán	Các âm không được khớp	Độ dài chuỗi phiên âm được dự đoán	PER	Kết quả
error_1	w aɪ ə n ɪ ɹ ə w ɪ ɹ p uː l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	ɪ, ɹ	32	6.25	Đúng
error_2	w aɪ ə n ɪ ɹ ə w ə k l f ɪ ɹ t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	k	29	12.5	Sai
error_3	w aɪ ə n ɪ ɹ ə w ə l p oʊ l f ɪ ɹ s t ə d ɹ ɔ k ɹ i eɪ ʃ ə n z ɪ n	oʊ	32	3.125	Đúng

Bảng 12. Kết quả phát hiện lỗi sai bằng LCS

Kết quả từ bảng 12 cho thấy mô hình có khả năng mô tả giống với giọng nói của người đọc, chỉ có mẫu error_2 bị lỗi do chữ “fierce” đọc khá tương tự với “fear”, chỉ khác ở âm “s” ở cuối, chứng tỏ mô hình student đã được huấn luyện tương đối ổn, để mô hình có khả năng nhận diện chính xác hơn nữa câu nói của người đọc, PER phải giảm xuống mức xấp xỉ $2\% - 3\%$. Thuật toán LCS cho thấy chỉ cần mô hình student nhận diện đúng được người đọc nói như thế nào thì có thể giải quyết phần còn lại của bài toán, phát hiện chuỗi âm vị sai (không được khớp).

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

5.1.1. Kết quả thu được

5.1.1.1. Kết quả bài toán

Từ phần 4, ta có thể thấy kết quả rất khả quan, luận văn đã thực hiện thành công kỹ thuật học bán giám sát với Noisy Student Training cho bài toán dự đoán chuỗi âm vị khi lỗi dự đoán âm vị PER của mô hình student chỉ là 12.66%, giảm PER tương đối 26%. Ngoài ra, mô hình còn cho thấy tiềm năng khi kết hợp với thuật toán tìm chuỗi con chung dài nhất để phát hiện lỗi phát âm của người đọc.

5.1.1.2. Kiến thức

Qua luận văn, tôi đã phần nào hiểu được công thức, kiến trúc, cách thức cài đặt, huấn luyện của mô hình tiền huấn luyện Wav2Vec2.0 Conformer, mô hình giám sát teacher Conformer và mô hình bán giám sát student Conformer. Thêm nữa, tôi còn biết thêm cách để kết hợp kết quả của bài toán dự đoán chuỗi âm vị với LCS để phát hiện lỗi phát âm, tạo tiền đề cho những nghiên cứu tương lai trong bài toán APED với kỹ thuật huấn luyện bán giám sát.

5.1.1.3. Kỹ năng

Việc thực hiện luận văn đã góp phần nâng cao khả năng thực nghiệm kết quả, đọc và tham khảo các nghiên cứu khác của tôi.

5.2. Hướng phát triển trong tương lai

Trong tương lai, tôi sẽ tiếp tục cải thiện, tối ưu bài toán dự đoán chuỗi âm vị, vì đây là thành phần chính trong bài toán tự động phát hiện lỗi phát âm. Nói cách khác là tiếp tục tối ưu mô hình student bằng cách huấn luyện trên nhiều dữ liệu không có nhãn hơn, kết hợp với kiến trúc khác ngoài CTC, tăng kích cỡ mô hình, tăng số lượng dữ liệu có nhãn hiện có, tăng số lượng thế hệ student lên để có thể đẩy kết quả lên mức cao nhất. Bên cạnh đó, tôi cũng sẽ tham khảo các hướng tiếp cận khác ngoài LCS để phát hiện lỗi phát âm.

TÀI LIỆU THAM KHẢO

[1] [Trực tuyến]. Available: https://pasal.edu.vn/tieng-anh-giao-tiep-chia-khoa-cho-gioi-tre-viet-nam-hoi-nhap-quoc-te-n575.html. [Đã truy cập 21 11 2022].

[2] "Indeed," [Trực tuyến]. Available: https://www.indeed.com/career-advice/career-development/english-language-certications. [Đã truy cập 21 11 2022].

[3] "Elsa," [Trực tuyến]. Available: https://vn.elsaspeak.com/. [Đã truy cập 21 11 2022].

[4] "Duolingo," [Online]. Available: https://www.duolingo.com/. [Accessed 21 11 2022].

[5] Tepperman, Joseph and Silva, Jorge and Kazemzadeh, Abe and You, Hong and Lee, Sungbok and Alwan, Abeer and Narayanan, Shrikanth, "Pronunciation verification of children's speech for automatic literacy assessment," 2006.

[6] Long Zhang, Ziping Zhao, Chunmei Ma, Linlin Shan, Huazhi Sun, Lifen Jiang, Shiwen Deng, Chang Gao, "End-to-End Automatic Pronunciation Error Detection Based on Improved Hybrid CTC/Attention Architecture," *Sensors*, tập 20, 2020.

[7] R. Ai, "Automatic Pronunciation Error Detection and Feedback Generation for CALL Applications," trong *Springer International Publishing*, 2015.

[8] Helmer Strik, Khiet Truong, Febe de Wet, Catia Cucchiarini, "Comparing different approaches for automatic pronunciation error detection," *Speech Communication*, tập 51, pp. 845-852, 2009.

[9] S. Witt, "Automatic Error Detection in Pronunciation Training: Where we are and where we need to go," 2012.

[10] K. Beatty, Teaching & Researching: Computer-Assisted Language Learning, London: Routledge, 2013.

[11] Zhan Zhang and Yuehai Wang and Jianyi Yang, "Text-conditioned Transformer for automatic pronunciation error detection," *Speech Communication*, tập 130, pp. 55-63, 2021.

[12] A. L. a. J. Glass, "Pronunciation assessment via a comparison-based system," trong *Proc. Speech and Language Technology in Education (SLaTE 2013)*, 2013.

[13] Lee, Ann and Zhang, Yaodong and Glass, James, "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," 2013.

[14] Md. Afzal Hossan, Sheeraz Memon, Mark A Gregory, "A novel approach for MFCC feature extraction," trong *2010 4th International Conference on Signal Processing and Communication Systems*, 2010.

[15] Lee, Ann and Chen, Nancy F. and Glass, James, "Personalized mispronunciation detection and diagnosis based on unsupervised error pattern discovery," trong *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[16] Ann Lee, James Glass, "A comparison-based approach to mispronunciation detection," trong *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012.

[17] S. M. Witt, "Use of speech recognition in computer-assisted language learning," 1999.

[18] S.M Witt and S.J Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, tập 30, pp. 95-108, 2000.

[19] Wai-Kim Leung, Xunying Liu, Helen Meng, "CNN-RNN-CTC Based End-to-end Mispronunciation Detection and Diagnosis," trong *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[20] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, Ruoming Pang, *Conformer: Convolution-augmented Transformer for Speech Recognition*, arXiv, 2020.

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

[22] Zhang, Yu and Qin, James and Park, Daniel S. and Han, Wei and Chiu, Chung-Cheng and Pang, Ruoming and Le, Quoc V. and Wu, Yonghui,

Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition, arXiv, 2020.

[23] Panayotov, Vassil and Chen, Guoguo and Povey, Daniel and Khudanpur, Sanjeev, "Librispeech: An ASR corpus based on public domain audio books," trong *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206-5210.

[24] Chiu, Chung-Cheng and Sainath, Tara N. and Wu, Yonghui and Prabhavalkar, Rohit and Nguyen, Patrick and Chen, Zhifeng and Kannan, Anjuli and Weiss, Ron J. and Rao, Kanishka and Gonina, Ekaterina and Jaitly, Navdeep and Li, Bo and Chorowski, Jan and Bacchia, *State-of-the-art Speech Recognition With Sequence-to-Sequence Models*, arXiv, 2017.

[25] Kanishka Rao, Haşim Sak, Rohit Prabhavalkar, *Exploring Architectures, Data and Units For Streaming End-to-End Speech Recognition with RNN-Transducer*, arXiv, 2018.

[26] He, Yanzhang and Sainath, Tara N. and Prabhavalkar, Rohit and McGraw, Ian and Alvarez, Raziel and Zhao, Ding and Rybach, David and Kannan, Anjuli and Wu, Yonghui and Pang, Ruoming and Liang, Qiao and Bhatia, Deepti and Shangguan, Yuan and Li, Bo and Punda, *Streaming End-to-end Speech Recognition For Mobile Devices*, arXiv, 2018.

[27] Tara N. Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziel Alvarez, Zhifeng Chen, Chung-Cheng Chiu, David Garcia, Alex Gruenstein, Ke Hu, Minho Jin, Anjuli Kannan, Qiao Liang, Ian McGraw, Cal Peyse, *A Streaming On-Device End-to-End Model Surpassing Server-Side Conventional Model Quality and Latency*, arXiv, 2020.

[28] A. Graves, *Sequence Transduction with Recurrent Neural Networks*, arXiv, 2012.

[29] Zhang, Qian and Lu, Han and Sak, Hasim and Tripathi, Anshuman and McDermott, Erik and Koo, Stephen and Kumar, Shankar, *Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss*, arXiv, 2020.

[30] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, Ravi Teja Gadde, *Jasper: An End-to-End Convolutional Neural Acoustic Model*, arXiv, 2019.

[31] Kriman, Samuel and Beliaev, Stanislav and Ginsburg, Boris and Huang, Jocelyn and Kuchaiev, Oleksii and Lavrukhin, Vitaly and Leary, Ryan and Li, Jason and Zhang, Yang, "Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions," trong *{ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[32] Sergey Ioffe, Christian Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv, 2015.

[33] A. F. Agarap, *Deep Learning using Rectified Linear Units (ReLU)*, arXiv, 2018.

[34] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, Yonghui Wu, *ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context*, arXiv, 2020.

[35] Tara N. Sainath, Abdel-rahman Mohamed, Brian Kingsbury, Bhuvana Ramabhadran, "Deep convolutional neural networks for LVCSR," trong *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[36] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, Dong Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, tập 22, pp. 1533-1545, 2014.

[37] Hu, Jie and Shen, Li and Sun, Gang, "Squeeze-and-Excitation Networks," trong *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[38] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, Quoc V. Le, *Attention Augmented Convolutional Networks*, arXiv, 2019.

[39] Baosong Yang, Longyue Wang, Derek Wong, Lidia S. Chao, Zhaopeng Tu, *Convolutional Self-Attention Networks*, arXiv, 2019.

[40] Yu, Adams Wei and Dohan, David and Luong, Minh-Thang and Zhao, Rui and Chen, Kai and Norouzi, Mohammad and Le, Quoc V., *QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension*, arXiv, 2018.

[41] Wu, Zhanghao and Liu, Zhijian and Lin, Ji and Lin, Yujun and Han, Song,

Lite Transformer with Long-Short Range Attention, arXiv, 2020.

[42] Baevski, Alexei and Zhou, Henry and Mohamed, Abdelrahman and Auli, Michael, *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,* arXiv, 2020.

[43] Xie, Qizhe and Luong, Minh-Thang and Hovy, Eduard and Le, Quoc V., *Self-training with Noisy Student improves ImageNet classification,* arXiv, 2019.

[44] Daniel S. Park and William Chan and Yu Zhang and Chung-Cheng Chiu and Barret Zoph and Ekin D. Cubuk and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," trong *ISCA*, 2019.

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* tập 15, pp. 1929-1958, 2014.

[46] X. Ying, "An Overview of Overfitting and its Solutions," *Journal of Physics: Conference Series,* tập 1168, p. 022022, 2019.

[47] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, *Layer Normalization,* arXiv, 2016.

[48] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, Ruslan Salakhutdinov, "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context," trong *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, 2019.

[49] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, Lidia S. Chao, *Learning Deep Transformer Models for Machine Translation,* arXiv, 2019.

[50] Nguyen, Toan Q. and Salazar, Julian, "Transformers without Tears: Improving the Normalization of Self-Attention," *Zenodo,* 2019.

[51] Yann N. Dauphin, Angela Fan, Michael Auli, David Grangier, *Language Modeling with Gated Convolutional Networks,* arXiv, 2016.

[52] "Paper With Code," [Trực tuyến]. Available: https://paperswithcode.com/method/pointwise-convolution. [Đã truy cập 22 11 2022].

[53] "Paper With Code," [Trực tuyến]. Available: https://paperswithcode.com/method/depthwise-convolution. [Đã truy cập 22 11 2022].

[54] Ramachandran, Prajit and Zoph, Barret and Le, Quoc V., *Searching for Activation Functions*, arXiv, 2017.

[55] Lu, Yiping and Li, Zhuohan and He, Di and Sun, Zhiqing and Dong, Bin and Qin, Tao and Wang, Liwei and Liu, Tie-Yan, *Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View*, arXiv, 2019.

[56] Alex Graves, Santiago Fernández, Faustino Gomez, Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," trong *Association for Computing Machinery*, New York, 2006.

[57] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, tập 77, pp. 257-286, 1989.

[58] John D. Lafferty, Andrew McCallum, Fernando C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," trong *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, 2001.

[59] Hervé A. Bourlard, Nelson Morgan, Connectionist Speech Recognition, New York: Springer New York, 1994.

[60] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, tập 78, pp. 1550-1560, 1990.

[61] M. W. Kadous, "Temporal classification: extending the classification paradigm to multivariate time series," 2002.

[62] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," trong *NATO Neurocomputing*, 1990.

[63] Markus Freitag, Yaser Al-Onaizan, "Beam Search Strategies for Neural Machine Translation," trong *Association for Computational Linguistics*, 2017.

[64] [Trực tuyến]. Available: https://www.ethnologue.com/. [Đã truy cập 24 11 2022].

[65] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.

[66] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, "Deep Contextualized Word Representations," trong *Association for Computational Linguistics*, New Orleans, Louisiana, 2018.

[67] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, "Improving language understanding by generative pre-training," 2018.

[68] Philip Bachman, R Devon Hjelm, William Buchwalter, "Learning Representations by Maximizing Mutual Information Across Views," trong *Advances in Neural Information Processing Systems*, 2019.

[69] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, *A Simple Framework for Contrastive Learning of Visual Representations,* arXiv, 2020.

[70] He, Kaiming and Fan, Haoqi and Wu, Yuxin and Xie, Saining and Girshick, Ross, *Momentum Contrast for Unsupervised Visual Representation Learning,* arXiv, 2019.

[71] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, Aaron van den Oord, *Data-Efficient Image Recognition with Contrastive Predictive Coding*, arXiv, 2019.

[72] Misra, Ishan and van der Maaten, Laurens, *Self-Supervised Learning of Pretext-Invariant Representations*, arXiv, 2019.

[73] Dongwei Jiang, Xiaoning Lei, Wubo Li, Ne Luo, Yuxuan Hu, Wei Zou, Xiangang Li, *Improving Transformer-based Speech Recognition Using Unsupervised Pre-training,* arXiv, 2019.

[74] Wang, Weiran and Tang, Qingming and Livescu, Karen, *Unsupervised Pre-training of Bidirectional Speech Encoders via Masked Reconstruction,* arXiv, 2020.

[75] Kazuya Kawakami, Luyu Wang, Chris Dyer, Phil Blunsom, and Aaron van den Oord, *Learning Robust and Multilingual Speech Representations,* arXiv, 2020.

[76] Rivière, Morgane and Joulin, Armand and Mazaré, Pierre-Emmanuel and

Dupoux, Emmanuel, *Unsupervised pretraining transfers well across languages,* arXiv, 2020.

[77] Schneider, Steffen and Baevski, Alexei and Collobert, Ronan and Auli, Michael, *wav2vec: Unsupervised Pre-training for Speech Recognition,* arXiv, 2019.

[78] Oord, Aaron van den and Li, Yazhe and Vinyals, Oriol, *Representation Learning with Contrastive Predictive Coding,* arXiv, 2018.

[79] Jan Chorowski and Ron J. Weiss and Samy Bengio and Aaron van den Oord, "Unsupervised Speech Representation Learning Using {WaveNet} Autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* tập 27, pp. 2041--2053, 2019.

[80] Liu, Alexander H. and Tu, Tao and Lee, Hung-yi and Lee, Lin-shan, *Towards Unsupervised Speech Recognition and Synthesis with Quantized Speech Representation Learning,* arXiv, 2019.

[81] van den Oord, Aaron and Vinyals, Oriol and kavukcuoglu, koray, "Neural Discrete Representation Learning," trong *Advances in Neural Information Processing Systems*, 2017.

[82] David Harwath, Wei-Ning Hsu, James Glass, *Learning Hierarchical Discrete Linguistic Units from Visually-Grounded Speech,* arXiv, 2019.

[83] Baevski, Alexei and Schneider, Steffen and Auli, Michael, *vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations,* arXiv, 2019.

[84] Eric Jang, Shixiang Gu, Ben Poole, *Categorical Reparameterization with Gumbel-Softmax,* arXiv, 2016.

[85] Alexei Baevski, Michael Auli, Abdelrahman Mohamed, *Effectiveness of self-supervised pre-training for speech recognition,* arXiv, 2019.

[86] Eloff, Ryan and Nortje, André and van Niekerk, Benjamin and Govender, Avashna and Nortje, Leanne and Pretorius, Arnu and van Biljon, Elan and van der Westhuizen, Ewald and van Staden, Lisa and Kamper, Herman, *Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks,* arXiv, 2019.

[87] Tjandra, Andros and Sisman, Berrak and Zhang, Mingyang and Sakti, Sakriani and Li, Haizhou and Nakamura, Satoshi, *VQVAE Unsupervised Unit Discovery and Multi-scale Code2Spec Inverter for Zerospeech*

Challenge 2019, arXiv, 2019.

[88] Yu-An Chung, Wei-Ning Hsu, Hao Tang, James Glass, *An Unsupervised Autoregressive Model for Speech Representation Learning,* arXiv, 2019.

[89] Liu, Yinhan and Ott, Myle and Goyal, Naman and Du, Jingfei and Joshi, Mandar and Chen, Danqi and Levy, Omer and Lewis, Mike and Zettlemoyer, Luke and Stoyanov, Veselin, *RoBERTa: A Robustly Optimized BERT Pretraining Approach,* arXiv, 2019.

[90] Mohamed, Abdelrahman and Okhonko, Dmytro and Zettlemoyer, Luke, *Transformers with convolutional context for ASR*, arXiv, 2019.

[91] Wu, Felix and Fan, Angela and Baevski, Alexei and Dauphin, Yann N. and Auli, Michael, *Pay Less Attention with Lightweight and Dynamic Convolutions,* arXiv, 2019.

[92] Jégou, Herve and Douze, Matthijs and Schmid, Cordelia, "Product Quantization for Nearest Neighbor Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* tập 33, pp. 117-128, 2011.

[93] F. Brouers, "Statistical Foundation of Empirical Isotherms," *Open Journal of Statistics*, tập 4, 2014.

[94] Jang, Eric and Gu, Shixiang and Poole, Ben, *Categorical Reparameterization with Gumbel-Softmax,* arXiv, 2016.

[95] Chris J. Maddison, Daniel Tarlow, Tom Minka, "A* Sampling," trong *Advances in Neural Information Processing Systems*, 2014.

[96] M. Gutmann, A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," *Journal of Machine Learning Research - Proceedings Track*, pp. 297-304, 2010.

[97] Dieleman, Sander and Oord, Aäron van den and Simonyan, Karen, *The challenge of realistic music generation: modelling raw audio at scale,* arXiv, 2018.

[98] Yuji Roh, Geon Heo, Steven Euijong Whang, *A Survey on Data Collection for Machine Learning: a Big Data -- AI Integration Perspective,* arXiv, 2018.

[99] "The Standford AI Lab Blog," [Trực tuyến]. Available: http://ai.stanford.edu/blog/weak-supervision/. [Đã truy cập 28 11 2022].

[100] Andrea Campagner, Davide Ciucci, Carl-Magnus Svensson, Marc Thilo Figge, Federico Cabitza, "Ground truthing from multi-rater labeling with three-way decision and possibility theory," *Information Sciences,* tập 545, pp. 771-790, 2021.

[101] Pierre Nodet, Vincent Lemaire, Alexis Bondu, Antoine Cornuéjols, Adam Ouorou, "From Weakly Supervised Learning to Biquality Learning: an Introduction," trong *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021.

[102] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review,* tập 5, pp. 44-53, 2018.

[103] Thomas G. Dietterich and Richard H. Lathrop and Tomás Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence,* tập 89, pp. 31-71, 1997.

[104] Novi Quadrianto, Alex J. Smola, Tibério S. Caetano, Quoc V. Le, "Estimating Labels from Label Proportions," *Journal of Machine Learning Research,* tập 10, pp. 2349-2374, 2009.

[105] Brendan Van Rooyen. Robert C. Williamson, "A Theory of Learning with Corrupted Labels," *Journal of Machine Learning Research,* tập 18, pp. 1-50, 2018.

[106] E. Hüllermeier, "Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization," *International Journal of Approximate Reasoning,* tập 55, pp. 1519-1534, 2014.

[107] Cabannnes, Vivien and Rudi, Alessandro and Bach, Francis, "Structured Prediction with Partial Labelling through the Infimum Loss," trong *Proceedings of the 37th International Conference on Machine Learning,* 2020.

[108] Stanley C. Fralick, "Learning to recognize patterns without a teacher," *IEEE Trans. Inf. Theory,* tập 13, pp. 57-64, 1967.

[109] Zoph, Barret and Ghiasi, Golnaz and Lin, Tsung-Yi and Cui, Yin and Liu, Hanxiao and Cubuk, Ekin D. and Le, Quoc V., *Rethinking Pre-training and Self-training,* arXiv, 2020.

[110] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, Kilian Weinberger, *Deep Networks with Stochastic Depth*, arXiv, 2016.

[111] Cubuk, Ekin D. and Zoph, Barret and Shlens, Jonathon and Le, Quoc V., *RandAugment: Practical automated data augmentation with a reduced search space*, arXiv, 2019.

[112] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, Quoc V. Le, "Improved Noisy Student Training for Automatic Speech Recognition," trong *Interspeech 2020*, 2020.

[113] "Wikipedia," [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Word_error_rate. [Đã truy cập 29 11 2022].

[114] "Wikipedia," [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Longest_common_subsequence_problem. [Đã truy cập 29 11 2022].

[115] J. Kahn and M. Riviere and W. Zheng and E. Kharitonov and Q. Xu and P.E. Mazare and J. Karadayi and V. Liptchinsky and R. Collobert and C. Fuegen and T. Likhomanenko and G. Synnaeve and A. Joulin and A. Mohamed and E. Dupoux, "Libri-Light: A Benchmark for ASR with Limited or No Supervision," trong *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[116] Mathieu Bernard and Hadrien Titeux, "Phonemizer: Text to Phones Transcription for Multiple Languages in Python," *Journal of Open Source Software*, tập 6, p. 3958, 2021.

[117] Panayotov, Vassil and Chen, Guoguo and Povey, Daniel and Khudanpur, Sanjeev, "Librispeech: An ASR corpus based on public domain audio books," trong *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[118] "GitHub," [Trực tuyến]. Available: https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec. [Đã truy cập 30 11 2022].

[119] Keunwoo Choi, György Fazekas, Kyunghyun Cho, Mark Sandler, *A Comparison of Audio Signal Preprocessing Methods for Deep Neural Networks on Music Tagging*, arXiv, 2017.

[120] "Wikipedia," [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Fast_Fourier_transform. [Đã truy cập 30 11 2022].

[121] "PyFilterbank," [Trực tuyến]. Available:

https://siggigue.github.io/pyfilterbank/melbank.html. [Đã truy cập 30 11 2022].

[122] "pytorch," [Trực tuyến]. Available: https://pytorch.org/audio/main/generated/torchaudio.transforms.MelSpectrogram.html. [Đã truy cập 30 11 2020].

[123] "Wikipedia," [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Beam_search. [Đã truy cập 1 12 2022].

[124] "GitHub," [Trực tuyến]. Available: https://github.com/daandouwe/ngram-lm. [Đã truy cập 1 12 2022].

[125] A. S. M Hasan, Saria Islam, M. Arifur Rahman, "A Comparative Study of Witten Bell and Kneser-Ney Smoothing Methods for Statistical Machine Translation," *JU Journal of Information Technology (JIT)*, tập 1, p. 1, 2012.

[126] "Wikipedia," [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Perplexity. [Đã truy cập 1 12 2022].

[127] "Github," [Trực tuyến]. Available: https://github.com/kensho-technologies/pyctcdecode. [Đã truy cập 1 12 2022].

[128] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng, *Deep Speech: Scaling up end-to-end speech recognition*, arXiv, 2014.

[129] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, Ronan Collobert, *End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures*, arXiv, 2019.

[130] "Huggingface," [Trực tuyến]. Available: https://huggingface.co/nvidia/tts_en_fastpitch. [Đã truy cập 2 12 2022].

[131] [Trực tuyến]. Available: https://www.kaggle.com/datasets/tuannguyenvananh/aped-sample. [Đã truy cập 2 12 2022].

[132] Staudemeyer, Ralf C. and Morris, Eric Rothstein, *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*, arXiv, 2019.

[133] “Wikipedia,” [Trực tuyến]. Available: https://en.wikipedia.org/wiki/Perplexity. [Đã truy cập 30 11 2022].

NHẬT KÝ LÀM VIỆC

Tuần	Từ ngày	Đến ngày	Nội dung
1	08/08/2022	14/08/2022	Tìm hiểu về từ khóa đề tài: Speech Verification. Đọc các tài liệu liên quan đến đề tài, cố gắng định nghĩa được hướng nghiên cứu.
2	15/08/2022	21/08/2022	Tiếp tục tìm hiểu về đề tài thông qua các nghiên cứu sẵn có.
3	22/08/2022	28/08/2022	Đọc sâu hơn một số tài liệu nhất định của hướng nghiên cứu, dần có được thông tin về đề tài.
4	29/08/2022	04/09/2022	Sau quá trình tìm hiểu, nhận ra từ khóa Speech Verification chưa được sử dụng phổ biến bằng Automatic Pronunciation Error Detection, mặc dù có cùng ý nghĩa. Nên đổi tên đề tài và tìm hiểu nhiều nghiên cứu liên quan hơn, từ khóa này đưa ra nhiều nghiên cứu hơn.
5	05/09/2022	11/09/2022	Tiếp tục nghiên cứu, đọc tài liệu liên quan đến APED.
6	12/09/2022	18/09/2022	Đọc tài liệu liên quan đến ASR để tìm hướng nghiên cứu cho APED dựa vào ASR.
7	19/09/2022	25/09/2022	Đọc tài liệu về ASR, tìm hiểu các hướng nghiên cứu khác nhau.
8	26/09/2022	02/10/2022	Phát hiện ra dữ liệu của bài toán này khá ít, đọc tài liệu về các hướng nghiên cứu sử dụng ít dữ liệu có nhãn.
9	03/10/2022	09/10/2022	Định nghĩa được hướng nghiên cứu, đọc tài liệu của hướng nghiên cứu. Tìm hiểu cách hiện thực mô hình như thế nào.
10	10/10/2022	16/10/2022	Tiếp tục đọc sâu hơn tài liệu của hướng nghiên

			cứu. Tìm kiếm dữ liệu cho phần Supervise (dữ liệu có nhãn âm vị).
11	17/10/2022	23/10/2022	Tìm hiểu cách sử dụng lại mô hình đã tiền huấn luyện của fairseq (facebook research). Thử tiền huấn luyện sử dụng mô hình của Hugging Face Transformers.
12	24/10/2022	30/10/2022	Thử nghiệm huấn luyện các teacher, cố gắng thử xem cách kết hợp cấu trúc như thế nào là tốt nhất cho teacher. Thử huấn luyện cho student, mặc kệ kết quả của teacher tốt hay xấu. Điều này dẫn đến kết quả của student khá tệ, do teacher không có khả năng cung cấp nhãn tốt.
13	31/10/2022	06/11/2022	Huấn luyện student, gặp vấn đề từ teacher, nhãn yếu được tạo ra từ teacher không sử dụng được, giá trị mất mát bị NaN, tìm cách sửa lỗi. Tìm cách để tăng cường nhiễu cho Noisy Student Training.
14	07/11/2022	13/11/2022	Thử huấn luyện lại teacher với nhiễu sử dụng Speed Perturbation và Multi Style Training (sử dụng bộ dữ liệu Musan). Huấn luyện giám sát với bộ dữ liệu Timit và không nhãn sử dụng LibriSpeech 100h. Kết quả của student không tốt hơn teacher nên phải thực nghiệm lại.
15	14/11/2022	20/11/2022	Đọc nhiều tài liệu hơn về nhiễu cho Noisy Student Training, phát hiện ra Multi Style Training không thể kết hợp với SpecAugment tốt được nên phải thực nghiệm lại, chỉ sử dụng SpecAugment, sử dụng thêm Adaptive SpecAugment để tăng cường nhiễu. Thử nhiều

			kiến trúc của teacher hơn với hy vọng teacher tốt sẽ tạo ra student tốt, đưa ra được kiến trúc tốt nhất cho teacher và student được trình bày trong luận văn. Thay đổi dữ liệu có nhãn và không nhãn thành dữ liệu trong bài luận. Kết quả huấn luyện của teacher tốt khiến student cũng tốt. Hoàn thiện việc huấn luyện mô hình.
16	21/11/2022	27/11/2022	Tập trung viết luận văn.
17	28/11/2022	04/12/2022	Tiếp tục viết luận văn, nhận phản hồi từ giáo viên hướng dẫn để chỉnh sửa bài luận. Gửi bản báo cáo bài luận cho giáo viên phản biện.
18	05/12/2022	11/12/2022	Hoàn thiện bài luận và slide thuyết trình.