AI Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

|  |
| --- |
| Dự án xây dựng mô hình Speech-to-Text với ngôn ngữ tiếng Việt |

16/08/2024

Người thực hiện: Tạ Hà Khoa

Giảng viên hướng dẫn: Thầy Phí Văn Lâm

Contents

[**1. Giới thiệu** 5](#_Toc174647615)

[1.1. Thông tin nền tảng 5](#_Toc174647616)

[1.2. Động lực và mục tiêu 5](#_Toc174647617)

[1.3. Lập kế hoạch 5](#_Toc174647618)

[**2. Thực hiện dự án** 6](#_Toc174647619)

[2.1. Thu thập dữ liệu 6](#_Toc174647620)

[2.2. Phương pháp huấn luyện 6](#_Toc174647621)

[2.3. Quy trình làm việc 7](#_Toc174647622)

[2.4. Thiết kế hệ thống 7](#_Toc174647623)

[**3. Kết quả** 8](#_Toc174647624)

[3.1. Xử lý dữ liệu 8](#_Toc174647625)

[3.2. Phân tích dữ liệu khám phá (EDA) 9](#_Toc174647626)

[3.3. Mô hình hóa 9](#_Toc174647627)

[3.3.1. Xủ lý dữ liệu 9](#_Toc174647628)

[3.3.2. Xây dựng mô hình tokenizer 10](#_Toc174647629)

[3.3.3. Trích xuất đặc trưng 12](#_Toc174647630)

[3.3.4. Tạo processor 12](#_Toc174647631)

[3.3.5. Tạo trình đo metrics 13](#_Toc174647632)

[3.3.6. Tạo mô hình huấn luyện 14](#_Toc174647633)

[3.4. Giao diện người dùng 15](#_Toc174647634)

[3.5. Kiểm thử và cải tiến 16](#_Toc174647635)

[3.5.1. Kết quả lần 1: 16](#_Toc174647636)

[3.5.2. Kết quả lần 2: 17](#_Toc174647637)

[4. Tác động của dự án 19](#_Toc174647638)

[4.1. Các hiệu quả đạt được 19](#_Toc174647639)

[4.2. Các mở rộng của dự án trong tương lai 19](#_Toc174647640)

[**5. Nhận xét và đánh giá của giảng viên** 20](#_Toc174647641)

# **1. Giới thiệu**

## 1.1. Thông tin nền tảng

Nhận dạng giọng nói, hay còn gọi là Speech-to-Text, là một trong những lĩnh vực nghiên cứu và ứng dụng đang phát triển mạnh mẽ trong công nghệ trí tuệ nhân tạo (AI) và xử lý ngôn ngữ tự nhiên (NLP). Công nghệ này cho phép chuyển đổi giọng nói của con người thành văn bản, tạo ra nhiều ứng dụng thực tiễn như trợ lý ảo, hệ thống điều khiển bằng giọng nói, dịch vụ khách hàng tự động, và hỗ trợ cho người khuyết tật.

Công nghệ nhận diện giọng nói đã có những bước tiến đáng kể trong những năm gần đây, được thúc đẩy bởi sự phát triển của các mô hình học máy phức tạp và sự sẵn có của một lượng lớn dữ liệu. Trong số các mô hình hàng đầu trong lĩnh vực này là Wav2Vec2, một mô hình tiền huấn luyện được phát triển bởi Facebook AI, nổi bật trong việc nhận diện giọng nói tự động (ASR). Wav2Vec2 đặc biệt phù hợp với các ngôn ngữ ít tài nguyên, thường thiếu các bộ dữ liệu phong phú cần thiết để huấn luyện các hệ thống nhận diện giọng nói hiệu suất cao.

Tiếng Việt, một ngôn ngữ có dấu với hơn 90 triệu người sử dụng, đặt ra những thách thức độc đáo cho việc nhận diện giọng nói do tính chất có dấu và sự hạn chế về số lượng dữ liệu giọng nói có chú thích. Sự ra đời của Wav2Vec2 mở ra những khả năng mới cho việc phát triển các hệ thống nhận diện giọng nói hiệu quả cho tiếng Việt.

## 1.2. Động lực và mục tiêu

Động lực chính của dự án này là giải quyết các thách thức hiện tại trong việc phát triển hệ thống nhận diện giọng nói cho tiếng Việt, một ngôn ngữ phức tạp với nhiều dấu thanh và đặc điểm ngữ âm độc đáo. Hiện nay, các công nghệ nhận diện giọng nói tiếng Việt còn hạn chế, đặc biệt là trong bối cảnh dữ liệu ngôn ngữ còn thiếu hụt và chưa được xử lý đầy đủ. Wav2Vec2, với khả năng học biểu diễn từ dữ liệu chưa gán nhãn và hiệu suất cao trên các ngôn ngữ ít tài nguyên, là một giải pháp tiềm năng cho vấn đề này.

Mục tiêu của dự án là xây dựng và tinh chỉnh một mô hình Wav2Vec2 cho nhận diện giọng nói tiếng Việt, sử dụng bộ dữ liệu từ Common Voice và các kỹ thuật xử lý dữ liệu tiên tiến. Dự án không chỉ nhằm đạt được hiệu suất cao trong việc chuyển đổi giọng nói tiếng Việt thành văn bản mà còn nhằm mở rộng ứng dụng của mô hình này trong các hệ thống AI khác, bao gồm dịch thuật và tương tác người-máy.

Thông qua dự án này, tôi mong muốn đóng góp vào sự phát triển của công nghệ nhận diện giọng nói tiếng Việt, tạo ra một công cụ hữu ích cho cộng đồng và thúc đẩy nghiên cứu trong lĩnh vực này.

## 1.3. Lập kế hoạch

Tuần 1: Chuẩn bị đề tài và tìm hiểu các thông tin liên quan

Tuần 2: Chuẩn bị dữ liệu và code các chương trình cần thiết, làm báo cáo

# **2. Thực hiện dự án**

## 2.1. Thu thập dữ liệu

Dữ liệu được sử dụng trong dự án này được lấy từ tập dữ liệu Common Voice, một tập dữ liệu mở do Mozilla phát triển nhằm phục vụ cho các nghiên cứu và ứng dụng về nhận diện giọng nói. Đối với dự án, tôi sử dụng phiên bản 11.0 của Common Voice, tập trung vào các đoạn ghi âm tiếng Việt. Tập dữ liệu này bao gồm các bản ghi âm từ nhiều người nói khác nhau, với đa dạng về giọng điệu, giới tính, và vùng miền, giúp tạo ra một bộ dữ liệu phong phú và đại diện cho tiếng Việt. Mỗi một điểm dữ liệu sẽ có cấu trúc như sau:

|  |  |
| --- | --- |
| client\_id | ‘c14ad590de005be3a512e187900c5ad5c76921…’ |
| path | ‘/root/.cache/huggingface/datasets/downloads/extracted/…’ |
| audio | {'path': '/root/.cache/huggingface/datasets/downloads/extracted…} |
| sentence | 'quả nhiên trúng tuyển vào trường Quốc Lập' |
| up\_votes | 2 |
| down\_votes | 0 |
| age | '' |
| gender | '' |
| accent | '' |
| locale | vi' |
| segment | '' |

Tập dữ liệu được chia thành hai phần: phần tập huấn luyện (train) và phần kiểm tra (validation). Tôi đã tiến hành các bước xử lý dữ liệu, bao gồm loại bỏ các cột không cần thiết, chuẩn hóa âm thanh về cùng tần số lấy mẫu, và xử lý văn bản để loại bỏ các ký tự đặc biệt nhằm đảm bảo chất lượng đầu vào tốt nhất cho mô hình.

## 2.2. Phương pháp huấn luyện

Tôi sử dụng mô hình Wav2Vec2, một mô hình nhận diện giọng nói tự động tiên tiến đã được huấn luyện trước trên dữ liệu đa ngôn ngữ. Mô hình này được tinh chỉnh (fine-tuning) trên tập dữ liệu tiếng Việt để tối ưu hóa khả năng nhận diện giọng nói trong ngữ cảnh ngôn ngữ cụ thể.

Phương pháp huấn luyện bao gồm việc đóng băng các lớp trích xuất đặc trưng của mô hình để duy trì các biểu diễn âm thanh chung đã học từ trước, đồng thời tập trung vào việc huấn luyện các lớp mạng nơ-ron phía trên để thích ứng với tiếng Việt. Tôi sử dụng trình tối ưu hóa AdamW và điều chỉnh tốc độ học (learning rate) để đảm bảo quá trình huấn luyện ổn định và đạt hiệu suất cao.

## 2.3. Quy trình làm việc

Quy trình làm việc của dự án được chia thành các giai đoạn chính như sau:

1. **Chuẩn bị dữ liệu**: Thu thập và xử lý dữ liệu từ Common Voice, bao gồm chuyển đổi định dạng, chuẩn hóa âm thanh, và làm sạch văn bản.
2. **Xây dựng mô hình**: Tạo và tinh chỉnh bộ tokenizer, cấu hình mô hình Wav2Vec2 và các thành phần cần thiết khác như feature extractor và processor.
3. **Huấn luyện mô hình**: Sử dụng bộ dữ liệu đã chuẩn bị để huấn luyện mô hình Wav2Vec2, theo dõi quá trình huấn luyện thông qua các chỉ số như loss và WER (Word Error Rate).
4. **Đánh giá và cải thiện**: Đánh giá mô hình trên tập dữ liệu kiểm tra, phân tích các lỗi để thực hiện các điều chỉnh cần thiết, và tiếp tục huấn luyện để cải thiện hiệu suất.

## 2.4. Thiết kế hệ thống

Hệ thống nhận diện giọng nói được thiết kế bao gồm các thành phần chính sau:

* **Mô hình Wav2Vec2**: Đóng vai trò chính trong việc chuyển đổi âm thanh thành văn bản. Mô hình này được tinh chỉnh để nhận diện các đặc điểm giọng nói tiếng Việt và có khả năng mở rộng để áp dụng cho các tác vụ ngôn ngữ khác.
* **Bộ xử lý dữ liệu**: Bao gồm các thành phần như tokenizer, feature extractor, và processor, chịu trách nhiệm tiền xử lý âm thanh và văn bản trước khi đưa vào mô hình.
* **Cơ sở hạ tầng huấn luyện**: Sử dụng GPU để tăng tốc quá trình huấn luyện, cùng với các kỹ thuật như gradient checkpointing để tối ưu hóa việc sử dụng bộ nhớ và giảm thời gian huấn luyện.
* **Giao diện người dùng**: Hệ thống có thể được tích hợp vào các ứng dụng AI khác, chẳng hạn như dịch thuật hoặc hệ thống trợ lý ảo, để cung cấp một trải nghiệm người dùng mượt mà và trực quan

A diagram of a process

Description automatically generated

# **3. Kết quả**

## 3.1. Xử lý dữ liệu

Ở bài toán này, ta chỉ quan tâm tới 3 thuộc tính là path, audio và sentence. Các thuộc tính còn lại không có giá trị sử dụng cho bài toán nên ta sẽ loại bỏ đi để tiết kiệm bộ nhớ.Do dữ liệu được chia ở tập test có số lượng khoảng một nửa tập training, cho nên để tối ưu nhất ta sẽ gộp tập train và test làm 1 rồi dùng validation để kiểm thử ở bước cuối cũng như kiểm tra sự dao động của độ đo sai số trong quá trình huấn luyện.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Việc kết hợp này được áp dụng do data hiện có quá ít. Và trong quá trình thực nghiệm tôi nhận thấy khi kết hợp tập test vào bộ training độ chính xác của mô hình tăng đáng kể so với việc không chia hoặc chia theo tỉ lệ 80/20

Ngoài ra, các kỹ thuật như giảm tiếng ồn và cắt bỏ các phần âm thanh không chứa lời nói cũng được áp dụng để cải thiện chất lượng dữ liệu đầu vào. Sau khi hoàn thành quá trình xử lý, dữ liệu đã sẵn sàng để được sử dụng trong quá trình huấn luyện mô hình.

## 3.2. Phân tích dữ liệu khám phá (EDA)

Quá trình phân tích dữ liệu khám phá (EDA) giúp tôi hiểu rõ hơn về cấu trúc và đặc điểm của tập dữ liệu. Tôi đã tiến hành phân tích phân phối độ dài của các đoạn âm thanh, cũng như tần suất xuất hiện của các từ và ký tự trong tập dữ liệu văn bản. Những thông tin này giúp xác định các thách thức tiềm ẩn trong việc huấn luyện mô hình, chẳng hạn như sự không đồng đều trong phân phối độ dài câu hoặc sự phổ biến của các từ hiếm.

Ngoài ra tôi cũng xem các thành phần chính của dữ liệu và loại bỏ các thông tin không cần thiết để dữ liệu huấn luyện được đống bộ và nhẹ. Sau đó loại bỏ các yếu tố như tiếng ồn để âm thanh đưa vào huấn luyện được “sạch” hơn.

## 3.3. Mô hình hóa

### 3.3.1. Xủ lý dữ liệu

Như đã biết thì Sampling Rate càng cao thì số lượng mẫu âm thanh lưu trữ càng nhiều, nhờ vậy mà âm thanh trở nên trong trẻo, rõ ràng và chính xác với thực tế hơn. Tuy vậy để tối ưu trong học máy ta không cần đến mức đó, với bài toán này ta chỉ cần mức Sampling Rate nhỏ vừa đủ để có thể nghe, với lại khi chỉnh nhỏ giúp mô hình huấn luyện nhanh hơn và giảm chi phí tính toán. Vì vậy ta sẽ dùng hàm **cast\_column** để chuyển audio có sampling rate từ 48.000 về 16.000.

Sau khi đã xử lý xong trên cột âm thanh ta sẽ tiếp tục xử lý trên dữ liệu văn bản (**sentence**). Ta nhận ra rằng, từ âm thanh chuyển sang văn bản, những dấu câu thường sẽ không có ý nghĩa, vì đơn giản dấu câu dùng để thể hiện trong văn viết, với lại khi thêm dấu câu mô hình sẽ rất khó có thể học được và xác định nó, nếu một vài dấu câu sai cũng sẽ gây rất nhiều phiền toái về ngữ nghĩa làm người đọc hiểu sai. Vì vậy để đơn giản chúng ta sẽ xoá hết những dấu câu này cũng như những ký tự đặc biệt.

A close-up of a computer code

Description automatically generated

### 3.3.2. Xây dựng mô hình tokenizer

Máy tính không thể hiểu được các văn bản, ta cần chuyển đổi các từ trong văn bản này sang dạng số (encoding) để máy tính có thể tính được xác suất của từng từ một. Thông thường ta có thể làm nhanh bước này bằng cách sử dụng một bộ thư viện có sẵn để chuyển đổi, tuy nhiên ta sẽ làm thủ công để có thể hiểu rõ hơn cách nó hoạt động, với lại đối với bài toán liên quan tới âm thanh, bộ Tokenizer này sẽ đặc biệt hơn một chút.

A screen shot of a computer code

Description automatically generated

Ta sẽ tiến hành nối tất cả các hàng trong cột sentence thành một hàng văn bản duy nhất rồi sẽ trích xuất các ký tự xuất hiện trong văn bản đó ra. Lưu ý ở đây ta sẽ chỉ lấy ký tự chứ không lấy từ vựng như các bộ tokenizer LLM thông thường. Vì âm thanh không cần dự đoán ngữ nghĩa nhiều, mô hình chỉ cần xác định đúng từ vựng được nói là được, CTC là một phương pháp học máy thường được sử dụng trong các nhiệm vụ như nhận dạng tiếng nói và nhận dạng văn bản. Nó cho phép mô hình dự đoán chuỗi ký tự mà không cần cung cấp độ dài chuỗi cố định. Trong học máy, việc xử lý dữ liệu một cách đúng đắn rất quan trọng để đảm bảo mô hình được đào tạo hiệu quả, đây được gọi là cơ chế CTC. Việc lấy theo ký tự này giúp ta có thể trích lọc ra gần như tất cả các ký tự chữ cái xuất hiện trong Tiếng Việt hiện nay

A white background with black text

Description automatically generated

Việc lấy ký tự cũng giúp ta chỉ cần lưu trữ một lượng nhỏ **96 phần tử** so với số lượng từ vựng đồ sộ trong Tiếng Việt thì lưu ký tự sẽ tối ưu hơn rất nhiều. Tuy nhiên ta cũng sẽ không quên thêm 2 phần tử là [UNK] và [PAD] để cho từ không xác định và padding.

Sau khi xong ta sẽ lưu lại bộ vocab này với định dạng json và tạo tokenizer theo cơ chế CTC mặc định với mô hình Wav2Vec2. Để tạo được bộ tokenizer ta sẽ cần load config từ mô hình gốc, tại đây ta sẽ lấy mô hình **facebook/wav2vec2-large-xlsr-53** và đặt tên cho mô mới là **wav2vec2-vi**.

A close up of words

Description automatically generated

Ta sẽ dùng hàm AutoConfig để lấy các thông tin cài đặt mặc định từ mô hình gốc về nhằm cài lên tokenizer của Tiếng Việt

A screenshot of a computer code

Description automatically generated

Cuối cùng ta sẽ gán các giá trị này vào hàm tạo tokenizer của Wav2Vec2.

A screenshot of a computer program

Description automatically generated

### 3.3.3. Trích xuất đặc trưng

A screenshot of a computer code

Description automatically generated

### 3.3.4. Tạo processor

A close-up of a computer code

Description automatically generated

Sau khi đã có bộ xử lý ta sẽ tiến hành áp dụng hàm biến đổi này lên toàn bộ các điểm dữ liệu, trong đó ở từng dòng ta sẽ biến đổi *đầu vào* và *đầu ra* của phần tử, với âm thanh đầu vào hàm processor sẽ trích xuất đặc trưng và biến đổi, với văn bản đầu ra thì sẽ tokenizer.

A screen shot of a computer code

Description automatically generated

Sau khi đã hoàn thành, ta sẽ tiến hành mapping nó trên tập train và test của dữ liệu.

Để tối ưu tốc độ training cho mô hình ta sẽ giới hạn độ dài audio đầu vào trên dataset bằng cách cắt với độ dài tối đa cho phép trên dữ liệu âm thanh là **15 giây.**

A screen shot of a computer code

Description automatically generated

### 3.3.5. Tạo trình đo metrics

Đối với bài toán Speech-to-Text thì ngoài sai số loss ra chúng ta sẽ dùng độ đo Word error rate (WER) để ước lượng hiệu xuất của mô hình



Lưu ý là chỉ số WER này chỉ được tính trên tập dữ liệu kiểm thử (test) để kết quả khách quan nhất có thể. Ngoài ra đối với token có giá trị **-100** tức là pad dùng để điền đầy độ dài văn bản ta đã cài đặt lúc đầu sẽ được chuyển thành pad\_token\_id của tokenizer

A screenshot of a computer code

Description automatically generated

### 3.3.6. Tạo mô hình huấn luyện

Ta sẽ tiến hành load mô hình từ **facebook/wav2vec2-large-xlsr-53** về rồi thiết đặt các thông số như dropout và các thông số khác như bên dưới, ta cũng cần phải chuyển mô hình sang device là **cuda** để các trọng số mô hình sẽ được lưu trên GPU, đảm bảo việc tính toán và lưu trữ tận dụng được tối đa sức mạnh của phần cứng.

A screenshot of a computer program

Description automatically generated

Sau đó ta sẽ đóng băng lớp trích xuất đặc trưng để đảm bảo lớp này không bị thay đổi các giá trị trọng số.

A screenshot of a computer program

Description automatically generated

A close-up of a computer screen

Description automatically generated

Cuối cùng ta sẽ tổng hợp lại tất cả rồi đưa vào hàm trainer rồi chạy hàm train( ) để huấn luyện mô hình.

A screenshot of a computer program

Description automatically generated

## 3.4. Giao diện người dùng

A screenshot of a computer

Description automatically generated

Giao diện gồm những đoạn audio để thử cũng như cho phép người dùng ghi âm giọng nói của mình và chương trình sẽ sinh ra đoạn văn bản tương ứng

## 3.5. Kiểm thử và cải tiến

Tổng số lần huấn luyện là 2 lần và huấn luyện trên cùng một tập dữ liệu như đề cập ở trên

### 3.5.1. Kết quả lần 1:

A screenshot of a graph

Description automatically generatedA graph of a graph with numbers and a line

Description automatically generated with medium confidence

Lần huấn luyện đầu cho thấy một số điểm quan trọng sau:

* Training Loss: Ban đầu, training loss giảm dần từ 7.9837 xuống 3.3282 ở bước 400. Điều này cho thấy mô hình đã học và cải thiện dần qua các bước huấn luyện.
* Validation Loss: Validation loss cũng giảm, nhưng dao động quanh mức 3.3016 ở bước 400. Mặc dù có sự cải thiện, validation loss không giảm đáng kể sau bước 150, cho thấy mô hình có thể đang dần hội tụ.
* WER (Word Error Rate): WER duy trì ở mức 1.0 trong suốt quá trình huấn luyện, nghĩa là mô hình không cải thiện về khả năng nhận diện chính xác từ. Đây là dấu hiệu cho thấy mô hình chưa học được cách nhận diện tốt từ ngữ và cần phải cải thiện.
* Hiệu suất đánh giá (Evaluation Performance): Thời gian chạy đánh giá là 26.42 giây với tốc độ xử lý khoảng 9.387 mẫu trên giây. Điều này cho thấy mô hình hoạt động khá nhanh, nhưng chất lượng chưa đạt yêu cầu.

Kết luận: Mặc dù training loss và validation loss có sự giảm nhẹ, WER không giảm cho thấy mô hình chưa đạt được hiệu suất nhận diện từ ngữ mong muốn.

### 3.5.2. Kết quả lần 2:

A screenshot of a graph

Description automatically generated

A graph of different colored lines

Description automatically generated

Lần huấn luyện thứ hai của bạn cho thấy những cải thiện đáng kể so với lần đầu tiên:

* Training Loss: Training loss giảm mạnh từ 3.3295 xuống còn 0.4660 sau 400 bước huấn luyện. Đây là một dấu hiệu tích cực, cho thấy mô hình đang học và cải thiện dần.
* Validation Loss: Validation loss cũng giảm từ 3.2241 xuống còn 0.6196. Sự giảm này cho thấy mô hình đang tổng quát hóa tốt hơn trên dữ liệu mới.
* WER (Word Error Rate): WER giảm dần từ 1.0 xuống 0.5750, cho thấy mô hình đang ngày càng chính xác hơn trong việc nhận diện từ ngữ. Đặc biệt, WER ở bước đánh giá cuối cùng là 0.5619, đây là một cải thiện đáng kể so với WER ban đầu.
* Hiệu suất đánh giá (Evaluation Performance): Thời gian đánh giá là 27.52 giây với tốc độ xử lý khoảng 9.012 mẫu trên giây. Dù thời gian tăng nhẹ so với lần đầu, hiệu suất và độ chính xác của mô hình đã được cải thiện đáng kể.

Kết luận: Lần huấn luyện thứ hai cho thấy mô hình của bạn đã học tốt hơn, với cả training loss và validation loss giảm mạnh, đồng thời WER giảm đáng kể, chứng tỏ mô hình đang tiến gần hơn đến việc nhận diện chính xác giọng nói. Những kết quả này cho thấy mô hình đang hội tụ và hiệu suất nhận diện từ ngữ đã được cải thiện rõ rệt so với lần huấn luyện đầu tiên.

# 4. Tác động của dự án

## 4.1. Các hiệu quả đạt được

Nghiên cứu này không chỉ giải quyết một vấn đề quan trọng mà còn đóng góp vào sự phát triển của lĩnh vực và cung cấp một cơ sở cho các nghiên cứu và ứng dụng trong tương lai. Tóm lại, trong đồ án này, chúng ta đã thực hiện thành công quá trình fine tuning model facebook/wav2vec2-large-xlsr-53 cho tiếng Việt bằng thư viện Wav2vec2 có CTC. Kết quả độ đo WER trên tập validation là 0.575039. Kết quả này cho thấy hệ thống đã đạt được độ chính xác tương đối ở mức kỳ vọng. Để đạt được kết quả này, ta đã thực hiện qua các bước bao gồm:

* Tiền xử lý dữ liệu âm thanh tiếng Việt trong bộ common\_voice\_11
* Tạo bộ processor, áp dụng các thay đổi lên trên bộ dữ liệu.
* Finetuning model mô hình với các siêu tham số được tối ưu hóa.
* Đánh giá độ chính xác của model trên tập validation. Tuy nhiên, vẫn còn một số vấn đề cần được cải thiện, chẳng hạn như:
* Độ chính xác của hệ thống có thể được cải thiện bằng cách tăng kích thước tập dữ liệu huấn luyện.
* Có thể sử dụng các kỹ thuật finetuning khác để cải thiện độ chính xác của hệ thống.
* Có thể sử dụng các mô hình ngôn ngữ lớn (LLM) để cải thiện khả năng hiểu ngôn ngữ tự nhiên của hệ thống.

## 4.2. Các mở rộng của dự án trong tương lai

Nghiên cứu có thể mở rộng để áp dụng mô hình Speech-to-Text vào các ứng dụng thực tế như tạo phụ đề cho video, triển khai trợ lý ảo hoặc tích hợp vào các sản phẩm và dịch vụ liên quan đến ngôn ngữ tiếng Việt. Để cải thiện độ chính xác của model, chúng ta có thể thực hiện các bước sau:

* Tăng kích thước của tập dữ liệu tiếng Việt. Điều này có thể được thực hiện bằng cách thu thập thêm dữ liệu từ các nguồn khác nhau, chẳng hạn như các cuộc hội thoại, bản ghi âm, v.v. 41
* Sử dụng các kỹ thuật tăng cường dữ liệu để tạo ra thêm dữ liệu từ tập dữ liệu hiện có. Một số kỹ thuật tăng cường dữ liệu phổ biến bao gồm:
  + Phản chiếu: Lặp lại các mẫu dữ liệu hiện có theo chiều ngược lại.
  + Trộn: Kết hợp các mẫu dữ liệu hiện có với nhau.
  + Thêm tiếng ồn: Thêm tiếng ồn vào các mẫu dữ liệu hiện có.

• Sử dụng các kỹ thuật tối ưu hóa khác nhau. Các kỹ thuật tối ưu hóa khác nhau có thể được sử dụng để cải thiện độ chính xác của model. Một số kỹ thuật tối ưu hóa phổ biến bao gồm:

* + AdamW: Một biến thể của Adam sử dụng trọng số giảm động.
  + Adagrad: Một kỹ thuật tối ưu hóa dựa trên độ dốc.
  + RMSProp: Một kỹ thuật tối ưu hóa dựa trên bình phương sai số trung bình.

Ngoài ra, chúng ta cũng có thể thử nghiệm các model khác nhau để xem liệu chúng có thể đạt được độ chính xác cao hơn hay không. Hy vọng rằng những ý tưởng này sẽ giúp cải thiện độ chính xác của model và làm cho nó trở nên hữu ích hơn cho các ứng dụng thực tế.

# **5. Nhận xét và đánh giá của giảng viên**

|  |  |  |
| --- | --- | --- |
| Danh mục | Điểm số | Nhận xét |
| Ý tưởng | \_\_/10 |  |
| Tính ứng dụng | \_\_/30 |  |
| Kết quả | \_\_/30 |  |
| Quản lý dự án | \_\_/10 |  |
| Thuyết trình | \_\_/20 |  |
| Tổng | \_\_/100 |  |