ỨNG DỤNG MÔ HÌNH WHISPER TRONG NHẬN DẠNG TIẾNG NÓI TỰ ĐỘNG (ASR)

# Chương 1. Tổng Quan Về Speech Technology

## Speech Technology là sự giao thoa của nhiều lĩnh vực khoa học, dùng tiếng nói để giao tiếp giữa người và máy. Đây là một dạng của ngôn ngữ tự nhiên nhằm giải quyết nhận dạng, hiểu và tạo ra tiếng nói con người. Speech Technology được phân ra nhiều loại như sau:

## 1.1. Speaker Recognition

Speaker Recognition là quá trình xác định hoặc xác minh danh tính của người nói từ giọng nói của họ. Có hai nhiệm vụ chính: (i) Speaker Identification (nhận dạng người nói trong tập hợp người dùng đã biết) và (ii) Speaker Verification (xác minh người nói có đúng là người đã khai báo hay không). Công nghệ này được ứng dụng trong bảo mật, xác thực truy cập và hệ thống ngân hàng giọng nói.

## 1.2. Speaker Diarization

Speaker Diarization là quá trình phân đoạn một đoạn âm thanh chứa nhiều người nói thành các vùng thời gian tương ứng với từng người. Mục tiêu là trả lời câu hỏi 'Ai nói khi nào?'. Đây là bước tiền xử lý quan trọng trong nhiều hệ thống nhận dạng giọng nói đa người nói như hội nghị, phỏng vấn, ghi chú y tế.

## 1.3. Speech Emotion Recognition

Speech Emotion Recognition (SER) là kỹ thuật nhằm nhận dạng cảm xúc của người nói như vui, buồn, giận dữ, bình tĩnh… dựa vào đặc trưng giọng nói. SER hữu ích trong chăm sóc khách hàng, giáo dục và giao tiếp người–máy có cảm xúc.

## 1.4. Automatic Speech Recognition

Automatic Speech Recognition (ASR) là quá trình chuyển đổi tiếng nói thành văn bản một cách tự động. ASR là thành phần cốt lõi của các ứng dụng như Google Assistant, Siri, và dịch vụ chuyển giọng nói thành văn bản trong hội nghị, video,...

## 1.5. Spoken Language Understanding

Hiểu Ngôn Ngữ Nói (SLU) là bước tiếp theo sau Nhận Dạng Giọng Nói (ASR), nhằm hiểu nội dung của câu nói. Có hai cách tiếp cận chính trong SLU:

Cách thứ nhất là chuyển giọng nói thành văn bản → xử lý ngôn ngữ tự nhiên (NLP) → hành động và chuyển giọng nói thành ý định . Tức là tiếp cận theo hướng tách rời ASR và NLP. Trong cách này, hệ thống đầu tiên sử dụng ASR để chuyển giọng nói thành văn bản. Sau đó, văn bản đầu ra được xử lý bởi các kỹ thuật NLP để xác định ý định (intent) của người nói và trích xuất các thông tin cần thiết (như địa điểm, thời gian, đối tượng,…).

Cách thứ 2 là tiếp cận end-to-end, tức là mô hình sẽ học trực tiếp từ âm thanh đầu vào đến ý định hoặc thông tin mà không cần chuyển thành văn bản. Trong trường hợp này, hệ thống nghe và hiểu luôn nội dung mà người dùng nói.

SLU rất cần thiết trong các ứng dụng như chatbot, trợ lý ảo và các hệ thống điều khiển bằng giọng nói.

## 1.6. Text-to-Speech Synthesis

Text-to-Speech (TTS) là quá trình chuyển đổi văn bản thành giọng nói. Các hệ thống TTS hiện đại như Tacotron 2, VITS giúp tạo ra giọng nói tự nhiên và biểu cảm hơn. Ứng dụng TTS bao gồm trợ lý ảo, thuyết minh video, sách nói và hỗ trợ người khiếm thị.

## 1.7. Mispronunciation Detection and Diagnosis

Mispronunciation Detection and Diagnosis (MD&D) là công nghệ nhận diện lỗi phát âm và chẩn đoán để phản hồi người học phát âm. Kỹ thuật này được ứng dụng rộng rãi trong dạy tiếng Anh, tiếng Trung và các nền tảng học ngoại ngữ sử dụng AI.

# Chương 2. Mô hình Whisper trong Automatic Speech Recognition (ASR)

## 2.1. Mô Hình Whisper.

Whisper là một mô hình mã hóa–giải mã do OpenAI phát triển, thuộc nhóm mô hình Transformer với kiến trúc attention. Whisper được huấn luyện trên 680.000 giờ dữ liệu tiếng nói đa ngôn ngữ và đa tác vụ, từ đó đạt khả năng nhận dạng tốt trên nhiều ngôn ngữ, trong đó có tiếng Việt. Điểm nổi bật của Whisper là sử dụng log-Mel spectrogram làm đầu vào, kết hợp với một bộ mã hóa (encoder) và giải mã (decoder) Transformer, hoạt động theo hướng end-to-end. Whisper còn hỗ trợ các tác vụ như phân đoạn lời nói (speech segmentation), phát hiện ngôn ngữ, nhận dạng từ khóa, và dịch ngôn ngữ nói.

Dữ liệu đầu vào là audio được resample về 16kHz, được trích xuất đặc trưng feature về log-mel-spectrogram ở năng lượng âm thanh trong 25ms, khoảng cách bắt đầu của 2 frame liên tiếp là 10ms, mỗi frame là các vector 80 chiều.

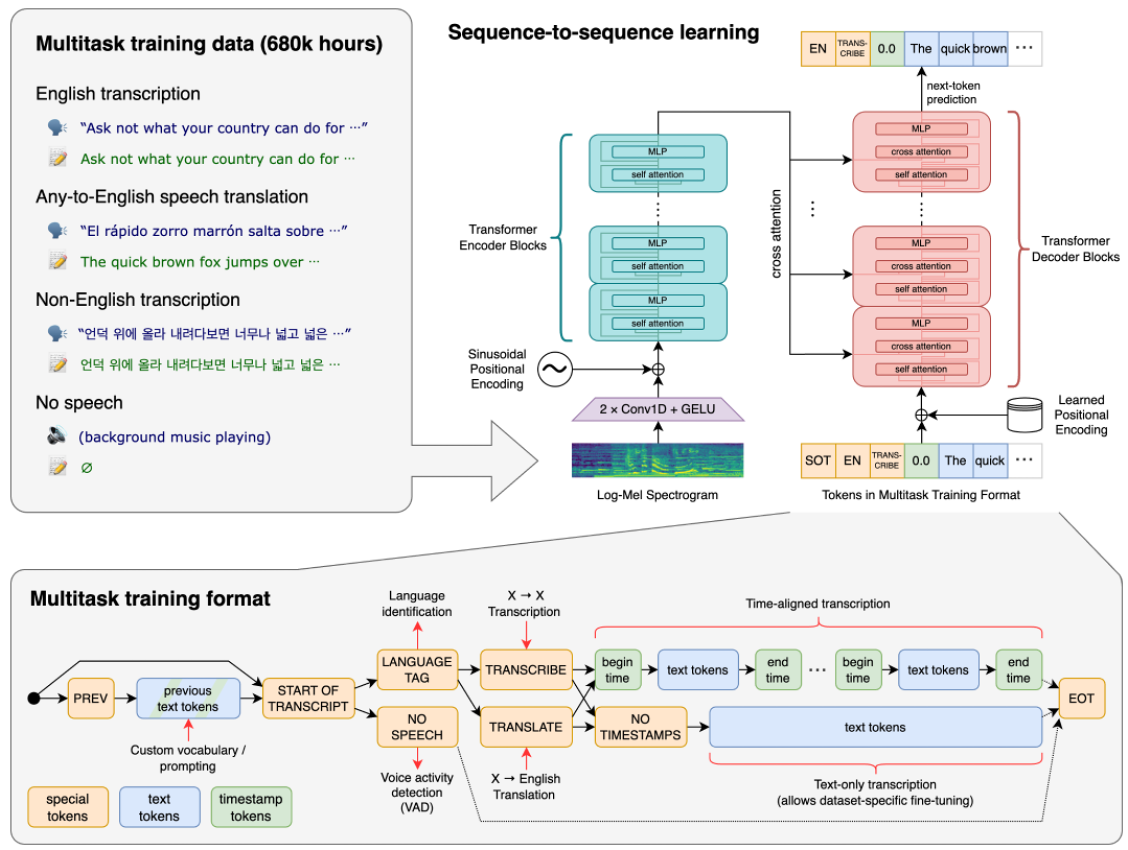
## 2.2. Cơ chế học Self-Supervised trong Whisper

Âm thanh đầu vào được mô hình Whisper xử lý bằng cách chuyển đổi thành log-Mel spectrogram – một dạng đặc trưng tần số-thời gian phản ánh đầy đủ thông tin của tín hiệu tiếng nói. Đặc trưng này sau đó được truyền qua hai lớp Convolution 1D kết hợp với hàm kích hoạt GELU, giúp trích xuất sâu hơn các đặc điểm quan trọng và giảm thiểu ảnh hưởng của nhiễu trong âm thanh. Để mô hình có thể nhận biết và hiểu được thứ tự thời gian của các khung âm thanh, sinusoidal positional encoding được áp dụng ngay tại bước đầu này.

Sau khi đặc trưng âm thanh đã được xử lý, dữ liệu sẽ được đưa vào thành phần Encoder gồm nhiều khối Transformer. Mỗi khối này bao gồm các lớp self-attention, cho phép mô hình tập trung vào những khung âm thanh quan trọng ở các vị trí khác nhau trong chuỗi, nhờ vậy có thể nắm bắt được ngữ cảnh dài hạn. Song song với đó là các lớp MLP (Feed-forward) có nhiệm vụ biến đổi phi tuyến tính, giúp tăng khả năng học của mô hình.

Tiếp theo, Decoder của mô hình cũng được xây dựng từ nhiều khối Transformer. Trong mỗi khối, các lớp self-attention sẽ tập trung vào những token văn bản đã sinh ra trước đó, đảm bảo việc dự đoán từ tiếp theo diễn ra hợp lý, đúng ngữ cảnh. Đặc biệt, các lớp cross-attention sẽ liên kết trực tiếp với output của encoder, khai thác toàn bộ thông tin đặc trưng đã được mã hóa từ tín hiệu âm thanh. Bên cạnh đó, các lớp MLP tiếp tục hỗ trợ mô hình trong việc biến đổi đặc trưng phi tuyến tính, trong khi learned positional encoding dành riêng cho chuỗi token đầu ra giúp mô hình hiểu được thứ tự vị trí trong văn bản.

Một điểm mạnh nổi bật của Whisper là định dạng huấn luyện đa nhiệm (multitask training format). Đầu vào của decoder không chỉ bao gồm token bắt đầu (start of transcript) mà còn chứa các token xác định nhiệm vụ (ví dụ: transcribe hay translate), token về ngôn ngữ đầu ra (ví dụ: EN cho tiếng Anh, VI cho tiếng Việt), cùng các thông tin về thời gian (timestamp), hoặc trạng thái có tiếng nói/không có tiếng nói (VAD). Cách thiết kế này giúp mô hình thích nghi nhanh chóng với nhiều nhiệm vụ khác nhau chỉ với một hệ thống duy nhất.



Bảng 1. WHISPER MODEL

## 2.3. Khả năng linh hoạt với đầu vào đa dạng

Một điểm nổi bật của mô hình Whisper là khả năng xử lý tốt với nhiều dạng đầu vào khác nhau như: Giọng nói đa dạng: Mô hình hỗ trợ hơn 97 ngôn ngữ và nhiều giọng địa phương khác nhau. Nhiễu âm: Whisper hoạt động hiệu quả ngay cả với tín hiệu âm thanh chứa tiếng ồn môi trường.Tốc độ nói khác nhau. Mô hình vẫn duy trì độ chính xác cao với tốc độ nói nhanh, chậm hoặc không đồng đều. Khả năng linh hoạt và robust cao của Whisper đến từ một số yếu tố quan trọng. Tập dữ liệu huấn luyện lớn và đa dạng: Gồm hơn 680.000 giờ dữ liệu từ nhiều nguồn thực tế như YouTube, podcast, TED Talk...

Đào tạo đa nhiệm (Multitask Learning): Mô hình học nhiều tác vụ song song (nhận dạng, dịch, phát hiện ngôn ngữ...), giúp tăng khả năng tổng quát hóa.Không cần chuẩn hóa đầu vào: Whisper có thể xử lý âm thanh 'thô' từ thực tế mà không yêu cầu tiền xử lý hoặc chuẩn hóa ngữ âm. Khả năng generalization mạnh: Mô hình không cần fine-tuning vẫn đạt hiệu quả cao trên tập test chưa thấy qua.

## 2.4. Mô hình Pho-Whisper

Pho-Whisper là một phiên bản được fine-tune từ mô hình Whisper gốc của OpenAI, được tối ưu hóa riêng cho bài toán nhận dạng tiếng nói tiếng Việt. Mô hình này được huấn luyện lại trên các tập dữ liệu lớn và chất lượng như VIVOS, Common Voice (Vietnamese), và VLSP, giúp tăng cường độ chính xác khi nhận diện giọng nói với ngữ âm và từ vựng đặc trưng của tiếng Việt. Pho-Whisper là kết quả của sự đóng góp từ các tổ chức và cộng đồng AI tại Việt Nam, như AIVIVN, BKAI, và nhiều nhóm nghiên cứu trong nước.

Về mặt kiến trúc, Pho-Whisper vẫn giữ nguyên cấu trúc Transformer Encoder–Decoder giống như mô hình Whisper gốc. Âm thanh đầu vào được chuẩn hóa về tần số lấy mẫu 16kHz, sau đó chuyển thành log-Mel spectrogram gồm 80 dải tần, cửa sổ 25ms và bước nhảy 10ms. Các đặc trưng này được đưa vào Transformer Encoder để trích xuất thông tin ngữ nghĩa, rồi được giải mã bởi Transformer Decoder để sinh ra transcript cuối cùng. Điểm khác biệt quan trọng là Pho-Whisper được fine-tune trên dữ liệu tiếng Việt, giúp mô hình thích ứng tốt hơn với phát âm, từ vựng và cấu trúc câu tiếng Việt.

Kết quả thử nghiệm cho thấy mô hình Pho-Whisper đạt hiệu suất vượt trội so với Whisper gốc. Cụ thể, trên tập dữ liệu VIVOS, Pho-Whisper đạt Word Error Rate (WER) chỉ khoảng 4–5%, trong khi Whisper gốc có WER vào khoảng 12–13%. Điều này chứng minh rằng việc tinh chỉnh mô hình trên dữ liệu tiếng Việt mang lại hiệu quả rõ rệt. Pho-Whisper cũng thể hiện khả năng nhận diện tốt với giọng nói mang accent vùng miền như miền Trung, miền Nam, hay tiếng Việt có pha trộn yếu tố ngoại ngữ.

# Chương 3. Triển Khai Mô Hình

Việc triển khai mô hình Whisper được thực hiện bằng Python, sử dụng thư viện Transformers của Hugging Face. Các bước triển khai cụ thể như sau:

## 3.1. Chuẩn bị dữ liệu

Tập dữ liệu gốc (~10 GB) được giảng viên cung cấp, được chia thành 3 batch để đảm bảo tốc độ xử lý và tránh gián đoạn trên Google Colab. Mỗi batch gồm 90% dữ liệu huấn luyện và 10% đánh giá. Dữ liệu âm thanh (.wav) và văn bản (.txt) được lưu trên Google Drive, sau đó đọc vào và chuyển thành datasets.Dataset của Hugging Face. Âm thanh được chuẩn hóa về mono 16kHz để phù hợp với mô hình Whisper, văn bản được token hóa, và đặc trưng âm thanh được trích xuất bằng Whisper Processor. Dữ liệu được xử lý riêng cho các tập train, validation và test. Với tập test, âm thanh được chuẩn hóa và đưa vào mô hình để dự đoán transcript phục vụ đánh giá hiệu suất.

Ngoài ra, để tăng độ đa dạng và độ phủ dữ liệu, nhóm đã bổ sung thêm các tập công khai gồm: **VIVOS, CommonVoice, VLSP, VinBigdata, FOSD, InfoRe, Bud500, LSVSC** và **VAIS-1000**.

| **Tên** | **Bộ dữ liệu** | **Link** |
| --- | --- | --- |
| AILAB-VNUHCM/vivos | 15 giờ | 11.660 train | 760 test | <https://huggingface.co/datasets/AILAB-VNUHCM/vivos> |
| doof-ferb/vlsp2020\_vinai\_100h | 100 giờ, 56.4k samples | <https://huggingface.co/datasets/doof-ferb/vlsp2020_vinai_100h> |
| doof-ferb/fpt\_fosd | 100h, 25.9k samples | <https://huggingface.co/datasets/doof-ferb/fpt_fosd> |
| doof-ferb/infore1\_25hours | 25h, 14.9k samples | <https://huggingface.co/datasets/doof-ferb/infore1_25hours> |
| linhtran92/viet\_bud500 | 500 giờ | 634K train | 7.5K val | 7.5K test | <https://huggingface.co/datasets/linhtran92/viet_bud500> |
| doof-ferb/LSVSC | 100h, 57k samples | <https://huggingface.co/datasets/doof-ferb/LSVSC> |
| doof-ferb/vais1000 | 1h40min, 1k samples | <https://huggingface.co/datasets/doof-ferb/vais1000> |
| NhutP/VSV-1100 | 1.03M samples | 16kHz | <https://huggingface.co/datasets/NhutP/VSV-1100> |
| doof-ferb/Speech-MASSIVE\_vie | 1 tiếng | <https://huggingface.co/datasets/doof-ferb/Speech-MASSIVE_vie> |
| doof-ferb/BibleMMS\_vie | 1 tiếng | <https://huggingface.co/datasets/doof-ferb/BibleMMS_vie> |
| capleaf/viVoice | 887K samples | 1,017h | <https://huggingface.co/datasets/capleaf/viVoice> |
| linhtran92/viet\_youtube\_asr\_corpus\_v2 | 100h | <https://huggingface.co/datasets/linhtran92/viet_youtube_asr_corpus_v2> |
| doof-ferb/VietMed\_labeled | 9.2k samples | <https://huggingface.co/datasets/doof-ferb/VietMed_labeled> |
| vlsp | 10k samples |  |

## 3.2. Cài đặt và tải mô hình Whisper

Cài đặt thư viện: transformers, datasets, torchaudio, evaluate, jiwer. Sử dụng mô hình huấn luyện đã được huấn luyện trước Whisper từ Hugging Face, cụ thể là openai/whisper-large-v3 đây là một mô hình đã được huấn luyện với tác vụ ASR đa ngôn ngữ với 1.55 tỉ tham số với bao gồm thành phần chính: model, tokenizer, processor. Chúng tôi sẽ tiếp fine-tuning mô hình với từ pretrained model này.

Sau đó nhóm có chạy thêm với mô hình Pho-Whisper Small ( pretrained Model) và Pho-Whisper Small (Fine-tune) để cải thiện điểm số WER

## 3.3. Huấn luyện mô hình (fine-tune)

Sử dụng Seq2SeqTrainer từ thư viện transformers để huấn luyện mô hình theo bài toán sequence-to-sequence. Các tham số được thiết lập bao gồm: kích thước batch\_size là 3, learning rate là 2e-5, số bước huấn luyện tối đa là 20000, tần suất đánh giá mỗi 1000 step một lần, và chiến lược lưu checkpoint mỗi 1000 step một lần. Do đây là một mô hình ngôn ngữ lớn với 1.55 tỉ tham số do đó cần yêu cầu phần cứng khá lớn. Với batch\_size = 2 thì mô hình đã chiếm 31G VRAM GPU, do đó mặc dù nhóm đã đăng kí Google colab pro + thì bao gồm cả quá trình dev và huấn luyện mô hình chỉ train được 10000 step thì bị giới hạn dung lượng và không thể tiếp tục train được nữa, lúc này không thể tiếp tục sử dụng server A100 với 50G VRAM, mà chỉ được cấp server 15G VRAM. Do đó checkpoint-10000 sẽ là checkpoint nhóm mang đi đánh giá các kết quả trên các tập dữ liệu public test và private test. Sau đó nhóm cũng thử nghiệm phương án khác sử dụng mô hình pho-whisper small để tiến hành finetune , nhóm rút kinh nghiệm từ whisper large thì train mô hình khá lâu, do đó nhóm lưạ chọn một mô hình cũng họ nhà whisper là whisper small để phù hợp hơn với chi phí thuê server bên ngoài tiếp tục tuning mô hình.

## 3.4. Suy luận và đánh giá

Sau khi huấn luyện, mô hình được sử dụng để suy luận (inference) trên từng file audio riêng lẻ hoặc cả tập public test. Kết quả suy luận là chuỗi văn bản, được giải mã từ token đầu ra bằng tokenizer và lưu vào file Transcript.txt. Đánh giá kết quả sử dụng chỉ số Word Error Rate (WER). File kết quả cuối cùng được nén thành định dạng .zip và nộp bài.

# Chương 4. Kết Quả Đạt Được

## 4.1. Kết Quả đạt được

Kết quả WER đạt được trên các tập Public test và Private test như sau.

| Tên mô hình | Public Test (WER) | Private Test (WER) |
| --- | --- | --- |
| Whisper Large V3 (pretrained model) | 17.91 | 55.00 |
| Whisper Large V3 ( fine-tune 12K steps) | 17.66 | 54.03 |
| Pho-Whisper Small ( pretrained model) | 9.08 | 40.7 |
| Pho-Whisper Small ( 500 steps) | 10.6 | 41.02 |

## 

*Hình ảnh kết quả cuối cùng của nhóm chúng tôi trên tập private test*

## 4.2. Nhận xét

Qua quá trình triển khai mô hình Whisper cho bài toán nhận dạng tiếng nói tiếng Việt, nhóm đã học được các triển khai huấn luyện và đánh giá một mô hình AI cho tác vụ nhận dạng tiếng nói tự động cho tiếng Việt, đặc biệt nhóm mạnh dạn triển khai thử sức với mô hình ngôn ngữ lớn của Whisper với 1.55 tỉ tham số để theo dõi kết quả, cũng như đo lường tài nguyên cần sử dụng để đào tạo một mô hình ngôn ngữ lớn như thế nào. Tuy nhiên do giới hạn phần cứng và chi phí khi sử dụng google colab, nhóm chỉ đào tạo được mô hình tới một giới hạn nhất định để đánh giá kết quả, chưa đào tạo mô hình được trên toàn bộ những tập dữ liệu đã tìm kiếm được. Kết quả chỉ có chút thay đổi nhẹ trên hai tập public test và private test. Sau đó chúng tôi lựa chọn mô hình pho whisper small để tiến hành finetuning tại một server vật lý thuê ở bên ngoài. Tại thời điểm thuê khoảng 10.000 đồng/giờ. Tuy nhiên trong quá trình finetune chúng tôi nhận thấy quá trình học của mô hình whisper này cũng rất chậm, có thể cho phần attention của mô hình gây ra vấn đề này. Nên với thời gian khoảng 5 tiếng nhóm train được khoảng 500 step tương ứng với 500 câu, do đó chúng tôi dự đoán vấn đề này có thể hoặc là do server chưa đủ mạnh hoặc là do tính chất của mô hình whisper nên nếu trong tương lai tinh chỉnh mô hình tiếp thì cần rất nhiều thời gian. Chính vì thời gian dài mà finetuning trên số lượng câu được ít như vậy, nên kết quả có chút tăng nhẹ về lỗi sai, điều này khá dễ hiểu khi finetune một mô hình ở những step đầ[u.](http://u.do)Do đó nhóm chấp nhận kết quả ở mức WER 40.7 trên tập private test.

Trong tương lai nếu có điều kiện hơn về phần cứng server và tài chính, tiếp tục huấn luyện mô hình trên những bộ dữ liệu đã thu thập được, nhóm bài tập lớn tin kết quả sẽ cải thiện đáng kể hơn khá nhiều.