Phân đoạn Âm thanh Nâng cao: Hướng dẫn Kỹ thuật về Dóng hàng Cưỡng bức và Chỉnh sửa Ngữ điệu bằng AI

Phần I: Mô thức Dóng hàng theo Hướng dẫn của Văn bản: Thiết lập Chân lý Nền tảng

Phần đầu tiên của báo cáo này thiết lập các công nghệ nền tảng để giải quyết nửa đầu của vấn đề người dùng đặt ra: xác định chính xác thời gian bắt đầu và kết thúc của mỗi từ trong âm thanh, bất kể các khoảng lặng bị lỗi giữa chúng. Chúng ta sẽ đi từ các phương pháp cổ điển, có độ chính xác cao đến các mô hình đầu cuối hiện đại, nhanh hơn nhưng kém chính xác hơn, và cuối cùng là các giải pháp lai (hybrid) mang lại những ưu điểm tốt nhất của cả hai thế giới.

Chương 1: Nguyên tắc của Dóng hàng Cưỡng bức

Mục tiêu

Chương này nhằm giới thiệu khái niệm cốt lõi của dóng hàng cưỡng bức (Forced Alignment - FA), được định nghĩa là quá trình đồng bộ hóa một bản ghi chép văn bản (transcript) với tín hiệu âm thanh tương ứng để xác định các ranh giới thời gian cho các đơn vị ngôn ngữ như từ hoặc âm vị (phone).[1] Đây là khối xây dựng cơ bản cho bất kỳ tác vụ chỉnh sửa âm thanh nào được dẫn dắt bởi văn bản. Quá trình dóng hàng này tạo điều kiện thuận lợi cho việc xử lý các tệp âm thanh ở các bước tiếp theo bằng cách cung cấp vị trí nhanh chóng và chính xác của các đơn vị lời nói trong một tệp âm thanh dài hơn.[1]

Khuôn khổ GMM-HMM Cổ điển

Trong lịch sử, các thuật toán dóng hàng cưỡng bức đã thống trị lĩnh vực này thông qua việc sử dụng kiến trúc Mô hình Markov Ẩn (Hidden Markov Model - HMM) kết hợp với Mô hình Hỗn hợp Gaussian (Gaussian Mixture Model - GMM).[1] Khuôn khổ này, mặc dù đã có sự chuyển dịch sang các kiến trúc đầu cuối, vẫn là phương pháp chủ đạo để đạt được FA với độ chính xác cao.[1]

Kiến trúc này hoạt động bằng cách phân chia nhiệm vụ: HMM mô hình hóa khía cạnh thời gian của lời nói, cụ thể là chuỗi các trạng thái âm vị và xác suất chuyển đổi giữa chúng. Trong khi đó, GMM mô hình hóa các đặc trưng âm học—chẳng hạn như Hệ số Cepstral Tần số Mel (Mel-frequency cepstral coefficients - MFCCs)—liên quan đến từng trạng thái HMM.[1] Ưu điểm trực quan của hệ thống dựa trên HMM nằm ở mối quan hệ thời gian trực tiếp giữa các khung âm học (acoustic frames) và các trạng thái được gán nhãn, cho phép xác định ranh giới thời gian một cách chính xác.[1]

Quá trình dóng hàng được thực hiện thông qua thuật toán Viterbi, thuật toán này tìm ra đường đi có xác suất cao nhất của các trạng thái HMM dựa trên các khung âm học và bản ghi chép văn bản được cung cấp.[1] Quá trình này là một vòng lặp: đầu tiên, các phân phối GMM được khởi tạo, sau đó thuật toán dóng hàng các đặc trưng âm thanh với các trạng thái HMM. Dựa trên kết quả dóng hàng này, các tham số của phân phối GMM được cập nhật. Quá trình này, được gọi là thuật toán Cực đại hóa Kỳ vọng (Expectation-Maximization - EM), được lặp lại nhiều lần cho đến khi mô hình hội tụ, đảm bảo độ chính xác cao.[1]

Phân biệt với Nhận dạng Tiếng nói Tự động (ASR)

Cần làm rõ một sự khác biệt quan trọng: Nhận dạng Tiếng nói Tự động (Automatic Speech Recognition - ASR) có nhiệm vụ *dự đoán* văn bản từ âm thanh, trong khi dóng hàng cưỡng bức (FA) *dóng hàng* một văn bản đã biết với âm thanh.[1] Thuật ngữ "cưỡng bức" trong FA đề cập đến việc chúng ta cung cấp văn bản tham chiếu làm chân lý nền tảng (ground truth), buộc mô hình phải tìm ra sự dóng hàng tốt nhất dựa trên giả định rằng văn bản này chính xác là những gì đã được nói.[7] Điều này giới hạn không gian tìm kiếm của thuật toán và chuyển nhiệm vụ từ nhận dạng sang dóng hàng.

Sự phân biệt này là nền tảng cho việc lựa chọn công nghệ phù hợp. Các hệ thống ASR hiện đại được tối ưu hóa để đạt được tỷ lệ lỗi từ (Word Error Rate - WER) thấp nhất, nhưng không

nhất thiết phải duy trì các biểu diễn phản ánh chính xác sự dóng hàng thời gian.[8] Ngược lại, các hệ thống FA được thiết kế đặc biệt cho mục đích này, khiến chúng trở thành công cụ không thể thiếu cho các nghiên cứu âm vị học và các ứng dụng đòi hỏi độ chính xác về mặt thời gian.[1]

Đầu vào và Đầu ra

Một hệ thống FA cổ điển đòi hỏi ba đầu vào thiết yếu để hoạt động chính xác:

1. **Tệp âm thanh:** Tín hiệu giọng nói cần được xử lý, thường ở định dạng .wav.[1]
2. **Bản ghi chép văn bản (Transcript):** Văn bản chính tả (orthographic) của những gì được nói trong tệp âm thanh.[1]
3. **Từ điển phát âm (Lexicon):** Một tệp ánh xạ các từ trong văn bản sang chuỗi âm vị tương ứng của chúng, cung cấp cho mô hình kiến thức về cách phát âm của từ.[1]

Đầu ra của quá trình này là một tệp được dóng hàng theo thời gian, chẳng hạn như tệp TextGrid của phần mềm Praat, chứa các mốc thời gian bắt đầu và kết thúc chính xác cho mỗi từ và mỗi âm vị trong bản ghi chép.[11] Độ chính xác này là nền tảng cho việc phân tích và chỉnh sửa âm thanh một cách chi tiết.

Sự phát triển của công nghệ giọng nói đã tạo ra một sự đánh đổi cơ bản: một bên là độ chính xác thời gian cao của các hệ thống dựa trên HMM mô-đun cũ, và bên kia là độ chính xác nhận dạng vượt trội và sự dễ sử dụng của các mô hình nơ-ron đầu cuối mới hơn. Các hệ thống HMM như Montreal Forced Aligner (MFA), được xây dựng dựa trên mối quan hệ thời gian trực tiếp giữa các khung âm học và trạng thái âm vị, vốn dĩ được thiết kế để đạt độ chính xác cao trong việc dóng hàng.[1] Ngược lại, các mô hình transformer đầu cuối như Whisper, được huấn luyện với hàm mất mát cross-entropy, tối ưu hóa cho việc dự đoán chuỗi token đầu ra chính xác chứ không phải thời gian chính xác của chúng.[13] Điều này giải thích tại sao Whisper có thể nhận dạng văn bản với "độ chính xác tuyệt đối" nhưng lại cung cấp các dấu thời gian rất thiếu chính xác.[15] Do đó, để giải quyết bài toán của người dùng—vốn đòi hỏi phải sửa chữa thời gian bị lỗi—việc chỉ dựa vào dấu thời gian gốc của một mô hình ASR hiện đại là không đủ. Một giải pháp mạnh mẽ hơn là cần thiết, dẫn đến sự lựa chọn giữa việc sử dụng một công cụ cổ điển như MFA hoặc một công cụ lai như WhisperX.

Chương 2: Dóng hàng Độ chính xác cao với Montreal Forced Aligner (MFA)

Mục tiêu

Chương này giới thiệu Montreal Forced Aligner (MFA) như một triển khai mã nguồn mở tiêu chuẩn vàng của mô thức FA cổ điển. MFA được xây dựng dựa trên bộ công cụ ASR Kaldi mạnh mẽ, cung cấp một giải pháp toàn diện để dóng hàng âm thanh với văn bản ở cấp độ âm vị và từ.[1]

Quy trình và Kiến trúc của MFA

MFA cung cấp một quy trình làm việc có cấu trúc để đạt được sự dóng hàng chính xác, yêu cầu người dùng chuẩn bị các tệp đầu vào một cách cẩn thận.

- **Đầu vào:** Như đã đề cập, MFA yêu cầu các tệp âm thanh (ví dụ: .wav), bản ghi chép văn bản (thường ở định dạng .TextGrid của Praat hoặc các tệp .lab/.txt đơn giản), một mô hình âm học được huấn luyện trước cho ngôn ngữ mục tiêu, và một từ điển phát âm.[10]
- **Quy trình Dóng hàng Đa lượt (Multi-pass):** MFA sử dụng một quy trình dóng hàng phức tạp, tinh chỉnh kết quả qua nhiều lượt để tăng độ chính xác [9]:
    - *Lượt 1: Mô hình Đơn âm vị (Monophone):* Ở giai đoạn đầu, mỗi âm vị được mô hình hóa một cách độc lập, không xét đến ngữ cảnh âm vị học của nó. Đây là một bước khởi tạo thô.
    - *Lượt 2: Mô hình Tam âm vị (Triphone):* Ở giai đoạn này, các mô hình phụ thuộc vào ngữ cảnh được sử dụng. Mô hình âm học của một âm vị sẽ tính đến các âm vị đứng trước và sau nó, cho phép mô hình hóa hiện tượng đồng cấu âm (co-articulation), một yếu tố quan trọng trong lời nói tự nhiên.
    - *Lượt 3: Thích ứng Người nói (Speaker Adaptation - SAT):* Đây là một tính năng chính của MFA. Ở lượt cuối cùng, mô hình âm học được điều chỉnh để thích ứng với các đặc điểm âm học cụ thể của người nói trong tệp âm thanh. Quá trình này, thường sử dụng các kỹ thuật như LDA+MLLT, tính toán một phép biến đổi các đặc trưng MFCC cho mỗi người nói, giúp mô hình trở nên chính xác hơn với giọng nói cụ thể đó.[9]

Triển khai Thực tế

MFA là một công cụ dòng lệnh, và việc sử dụng nó đòi hỏi một số kiến thức cơ bản về terminal.

- **Sử dụng Dòng lệnh:** Quy trình làm việc điển hình bao gồm các lệnh chính như mfa validate và mfa align. Lệnh mfa validate được sử dụng để kiểm tra sự tương thích của kho dữ liệu (corpus), từ điển và mô hình âm học. Nó sẽ báo cáo các vấn đề tiềm ẩn, đặc biệt là các từ nằm ngoài từ vựng (Out-Of-Vocabulary - OOV).[12] Lệnh mfa align là lệnh chính thực hiện quá trình dóng hàng.[20]
- **Xử lý Từ OOV:** Một trong những thách thức lớn nhất khi sử dụng MFA là xử lý các từ OOV—những từ có trong bản ghi chép nhưng không có trong từ điển phát âm. MFA cung cấp hai giải pháp chính:
  1. **Thêm thủ công:** Người dùng có thể chỉnh sửa tệp từ điển và thêm các mục mới cho các từ OOV, cung cấp phiên âm âm vị học của chúng.[12]
  2. **Sử dụng Mô hình Grapheme-to-Phoneme (G2P):** MFA tích hợp các mô hình G2P có thể tự động dự đoán cách phát âm của các từ OOV dựa trên cách viết của chúng. Người dùng có thể sử dụng lệnh mfa g2p để tạo một từ điển mới từ một danh sách từ.[10]

Điểm mạnh và Hạn chế

- **Điểm mạnh:**
  - **Độ chính xác thời gian rất cao:** MFA có thể đạt được độ chi tiết thời gian ở mức 10ms, làm cho nó trở thành công cụ lý tưởng cho các nghiên cứu âm vị học, phân tích ngữ điệu, và bất kỳ ứng dụng nào mà độ chính xác thời gian ở cấp độ dưới từ là không thể thiếu.[4]
  - **Khả năng huấn luyện và tùy chỉnh:** Không giống như các hệ thống đóng, MFA cho phép người dùng huấn luyện các mô hình âm học và từ điển mới cho các ngôn ngữ, phương ngữ hoặc giọng nói chưa được hỗ trợ, miễn là có đủ dữ liệu.[11]
- **Hạn chế:**
  - **Độ phức tạp:** Việc cài đặt và sử dụng MFA có thể phức tạp, đòi hỏi người dùng phải làm quen với môi trường Conda và chuẩn bị cẩn thận các tệp đầu vào, đặc biệt là từ điển phát âm.[10]
  - **Phụ thuộc vào chất lượng đầu vào:** Độ chính xác của kết quả dóng hàng phụ thuộc rất nhiều vào chất lượng của mô hình âm học và sự đầy đủ của từ điển phát âm. Việc sử dụng một mô hình được huấn luyện trên giọng người lớn để dóng hàng cho giọng trẻ em, hoặc một mô hình tiếng Anh Mỹ cho tiếng Anh Anh, có thể dẫn đến kết quả sai lệch.[9]

Chương 3: Cuộc cách mạng Đầu cuối: Điểm mạnh và Thiếu sót về Dấu

thời gian của Whisper

Mục tiêu

Chương này phân tích mô hình Whisper của OpenAI như một đại diện tiêu biểu cho các mô hình ASR đầu cuối, quy mô lớn hiện đại. Mục tiêu là giải thích rõ ràng kiến trúc của nó và lý do tại sao cơ chế đánh dấu thời gian của nó lại không đáng tin cậy cho các tác vụ đòi hỏi độ chính xác cao.

Kiến trúc và Huấn luyện của Whisper

Whisper là một mô hình transformer dựa trên kiến trúc bộ mã hóa-bộ giải mã (encoder-decoder), được huấn luyện trên một tập dữ liệu khổng lồ gồm 680,000 giờ âm thanh đa dạng và được giám sát yếu (weakly supervised).[21] Phương pháp huấn luyện này cho phép Whisper đạt được khả năng nhận dạng mạnh mẽ trên nhiều miền, ngôn ngữ và điều kiện nhiễu khác nhau, thường đạt hiệu suất ở mức con người mà không cần tinh chỉnh (fine-tuning).[21]

Tuy nhiên, mục tiêu huấn luyện của nó—hàm mất mát cross-entropy—được thiết kế để tối ưu hóa việc dự đoán chuỗi token văn bản chính xác nhất có thể.[13] Điều này có nghĩa là toàn bộ kiến trúc và quá trình tối ưu hóa của mô hình đều tập trung vào việc trả lời câu hỏi "những từ nào đã được nói?", chứ không phải "chúng được nói chính xác vào lúc nào?". Sự tập trung vào độ chính xác của nội dung văn bản này chính là nguyên nhân gốc rễ dẫn đến sự thiếu chính xác về mặt thời gian của nó.

Vấn đề về Dấu thời gian

Hệ quả của phương pháp huấn luyện này là một vấn đề cố hữu về dấu thời gian.

- **Dấu thời gian không chính xác ở cấp độ đoạn:** Dấu thời gian gốc của Whisper được tạo ra ở cấp độ đoạn (segment-level) và thường không chính xác, đôi khi có thể lệch đi vài giây so với thực tế.[13] Điều này làm cho chúng không phù hợp cho các ứng dụng như chỉnh sửa video, tạo phụ đề chính xác theo từng từ, hoặc bất kỳ tác vụ nào cần đồng bộ

hóa chặt chẽ.

- **Hạn chế của dấu thời gian cấp độ từ:** Mặc dù OpenAI đã phát hành các tính năng để tạo ra dấu thời gian ở cấp độ từ, các hạn chế về mặt kiến trúc vẫn còn đó, và phương pháp được sử dụng không được công bố một cách minh bạch.[23] Người dùng liên tục báo cáo rằng ngay cả những dấu thời gian cấp độ từ này cũng không đủ chính xác cho việc chỉnh sửa hoặc phân đoạn chính xác.[14] Các lỗi dự đoán có thể dao động từ 20 đến 120 ms đối với ASR và khoảng 200 ms đối với dịch nói tự động (AST).[23]
- **Xử lý theo khối (Chunking):** Whisper xử lý các tệp âm thanh dài bằng cách chia chúng thành các khối 30 giây. Ranh giới của các khối này không nhất thiết phải trùng khớp với ranh giới của từ, điều này càng làm tăng thêm sự sai lệch về thời gian.[13]

Tóm lại, trong khi Whisper là một công cụ đột phá về độ chính xác nhận dạng, nó không được thiết kế để cung cấp dấu thời gian đáng tin cậy. Sự đánh đổi này đã thúc đẩy sự phát triển của một loạt các công cụ của bên thứ ba nhằm kết hợp khả năng nhận dạng xuất sắc của Whisper với các kỹ thuật dóng hàng chính xác hơn.

Chương 4: Các Giải pháp Dóng hàng Lai (Hybrid) Tiên tiến

Mục tiêu

Chương này trình bày và so sánh các công cụ hàng đầu đã xuất hiện để giải quyết vấn đề về dấu thời gian của Whisper. Các giải pháp này kết hợp khả năng nhận dạng mạnh mẽ của Whisper với các kỹ thuật dóng hàng chính xác hơn, tạo ra một phương pháp lai tận dụng những điểm mạnh của cả hai thế giới.

4.1. WhisperX - Cách tiếp cận Hai mô hình

- **Công nghệ Cốt lõi:** WhisperX cải tiến Whisper bằng cách thực hiện một lượt dóng hàng cưỡng bức thứ hai sau khi nhận dạng. Quá trình này sử dụng một mô hình ASR riêng biệt dựa trên âm vị, cụ thể là mô hình wav2vec2, để đạt được độ chính xác thời gian cao.[21]
- **Quy trình làm việc:**
    1. **Tiền xử lý bằng VAD:** Đầu tiên, WhisperX sử dụng tính năng Phát hiện Hoạt động Giọng nói (Voice Activity Detection - VAD) để phân đoạn âm thanh dài thành các đoạn nhỏ hơn, có thể quản lý được. Điều này không chỉ cải thiện hiệu quả bằng cách

cho phép xử lý hàng loạt (batch processing) mà còn giúp giảm thiểu hiện tượng "ảo giác" (hallucination) của Whisper—tức là tạo ra văn bản ở những đoạn không có lời nói.[21]

2. **Nhận dạng bằng Whisper:** Tiếp theo, nó sử dụng một mô hình Whisper để tạo ra một bản ghi chép văn bản có độ chính xác cao cho mỗi đoạn âm thanh đã được phân đoạn.[24]
3. **Dóng hàng dựa trên Âm vị:** Cuối cùng, nó sử dụng mô hình wav2vec2 để thực hiện dóng hàng cưỡng bức bản ghi chép do Whisper tạo ra với âm thanh ở cấp độ âm vị. Vì các mô hình wav2vec2 được tinh chỉnh để nhận dạng các đơn vị âm thanh nhỏ nhất, chúng có thể xác định ranh giới từ với độ chính xác rất cao.[24]

- **Triển khai Thực tế:** WhisperX cung cấp một API Python đơn giản, thể hiện rõ ba bước riêng biệt: load_model (để tải mô hình Whisper), align (để thực hiện dóng hàng với mô hình wav2vec2), và tùy chọn diarize (để xác định người nói).[24]
- **Hiệu suất:** WhisperX đã chứng tỏ hiệu suất hàng đầu, vượt trội đáng kể so với các phương pháp chỉ sử dụng Whisper trong các bài kiểm tra về phân đoạn từ.[21] Tuy nhiên, một số người dùng vẫn báo cáo sự khác biệt khi so sánh với tiêu chuẩn vàng là MFA, cho thấy rằng đối với các ứng dụng đòi hỏi độ chính xác tuyệt đối, MFA vẫn có thể là lựa chọn ưu tiên.[28]

4.2. stable-ts & whisper-timestamped - Cách tiếp cận Cơ chế Nội tại

- **Công nghệ Cốt lõi:** Các thư viện này áp dụng một cách tiếp cận khác. Thay vì sử dụng một mô hình dóng hàng bên ngoài, chúng khai thác một cách thông minh các trọng số chú ý chéo (cross-attention weights) bên trong của Whisper. Đây là cơ chế mà bộ giải mã của Whisper sử dụng để tập trung vào các phần âm thanh có liên quan cho mỗi token văn bản mà nó dự đoán.[29]
- **Quy trình làm việc:**
  1. **Nhận dạng bằng Whisper:** Một lượt nhận dạng tiêu chuẩn được thực hiện để tạo ra văn bản.
  2. **Trích xuất Chú ý Chéo:** Trong hoặc sau quá trình giải mã, các bản đồ chú ý (attention maps) giữa đầu ra của bộ mã hóa âm thanh và các token của bộ giải mã văn bản được trích xuất. Các bản đồ này về cơ bản cho thấy phần nào của âm thanh tương ứng với phần nào của văn bản.
  3. **Dóng hàng qua DTW:** Thuật toán Co giãn Thời gian Động (Dynamic Time Warping - DTW) được sử dụng để tìm ra đường đi dóng hàng tối ưu qua ma trận chú ý, ánh xạ các token văn bản với các khung âm thanh một cách hiệu quả.[8]
- **Kỹ thuật Tinh chỉnh:** stable-ts nổi bật với các tính năng hậu xử lý tiên tiến. Nó có thể sử dụng VAD để xác định và loại bỏ các khoảng lặng, từ đó tinh chỉnh dấu thời gian một cách chính xác hơn. Nó cũng có thể nhóm các từ lại thành các phân đoạn tự nhiên hơn

dựa trên dấu câu và các khoảng lặng, và thậm chí cung cấp một phương pháp tinh chỉnh lặp đi lặp lại để cải thiện hơn nữa độ chính xác của dấu thời gian.[29]

- **Triển khai Thực tế:** stable-ts cung cấp một API Python rất thân thiện với người dùng. Một ví dụ mã đơn giản sẽ bao gồm việc sử dụng stable_whisper.load_model() và model.transcribe(), trong đó các tham số chính như word_timestamps=True và suppress_silence=True được bật.[29] Một tính năng đặc biệt mạnh mẽ là hàm model.align(), cho phép dóng hàng trực tiếp một văn bản đã có sẵn với một tệp âm thanh, rất phù hợp với kịch bản đầu vào của người dùng.[32]

4.3. Phân tích So sánh và Khuyến nghị

Phần này tổng hợp các phát hiện từ các chương 2-4 thành một khuôn khổ ra quyết định rõ ràng, được hỗ trợ bởi một bảng so sánh chi tiết. Việc lựa chọn công cụ phù hợp phụ thuộc vào sự cân bằng giữa các yếu tố như độ chính xác, tốc độ, sự dễ sử dụng và các yêu cầu cụ thể của dự án. Bảng dưới đây cung cấp một cái nhìn tổng quan để hỗ trợ quá trình ra quyết định này, chắt lọc các nghiên cứu thành một tài liệu tham khảo nhanh chóng, so sánh các công cụ trên các trục quyết định quan trọng nhất.

Bảng 1: Phân tích So sánh các Công cụ Dóng hàng Âm thanh bằng AI

Tính năng	**Montreal Forced Aligner (MFA)**	**WhisperX**	**stable-ts / whisper-timestamped**
Công nghệ Cốt lõi	HMM-GMM với backend Kaldi [1]	Whisper (Transformer) để nhận dạng + Wav2Vec2 (ASR Âm vị) để dóng hàng [24]	Whisper (Transformer) với chú ý chéo nội tại + DTW để dóng hàng [29]
Đầu vào Chính	Âm thanh, Bản ghi chép, Mô hình Âm học, Từ điển Phát âm [11]	Tệp âm thanh (bản ghi chép được tạo ra nội bộ) [24]	Tệp âm thanh (bản ghi chép được tạo ra nội bộ) hoặc Âm thanh + Bản ghi chép có sẵn cho hàm align() [33, 35]
Độ chính xác Dấu	**Rất cao (Tiêu**	**Cao.** Tốt hơn đáng	**Tốt đến Cao.** Độ

thời gian (Tương đối)	**chuẩn Vàng).** Độ chi tiết 10ms. Được coi là chính xác nhất trong các bài kiểm tra và báo cáo người dùng.[4, 28]	kể so với Whisper gốc. Vượt trội các phương pháp dựa trên Whisper khác trong một số bài kiểm tra [21], nhưng vẫn có thể kém chính xác hơn MFA.[28]	chính xác đang được cải thiện với các kỹ thuật tinh chỉnh tốt hơn (VAD, v.v.).[31] Một số người dùng báo cáo nó chính xác hơn WhisperX trong một số trường hợp.[36]
Phụ thuộc vào Nhận dạng	**Chỉ dóng hàng.** Yêu cầu một bản ghi chép chính xác có sẵn từ trước.[1]	**Nhận dạng + Dóng hàng.** Tự tạo ra bản ghi chép, đây là một lợi thế lớn.[24]	**Nhận dạng + Dóng hàng.** Cũng có thể dóng hàng một bản ghi chép có sẵn, mang lại sự linh hoạt.[35]
Mức độ Dễ triển khai	**Phức tạp từ Trung bình đến Cao.** Yêu cầu thiết lập môi trường (Conda) và chuẩn bị cẩn thận nhiều tệp đầu vào.[12, 18]	**Dễ.** Cài đặt đơn giản bằng pip và API Python dễ hiểu.[24]	**Dễ.** Cài đặt đơn giản bằng pip và API rất thân thiện với người dùng.[29, 33]
Điểm mạnh Chính	Độ chính xác vô song, có thể huấn luyện cho ngôn ngữ/giọng nói mới, mạnh mẽ và đã được thiết lập tốt.[11]	Độ chính xác nhận dạng xuất sắc, suy luận hàng loạt nhanh, phân loại người nói, cân bằng tốt giữa độ chính xác và sự dễ sử dụng.[21]	Độc lập (không cần mô hình thứ hai), hàm align() linh hoạt, các tùy chọn hậu xử lý và tinh chỉnh nâng cao.[29, 32]
Trường hợp Sử dụng Lý tưởng	Nghiên cứu âm vị học thuật, phân tích pháp lý/pháp y, các ứng dụng mà độ chính xác thời gian dưới cấp độ từ là không thể thiếu.	Nhận dạng và tạo phụ đề đa dụng, chỉnh sửa podcast/video, các ứng dụng cần nhãn người nói và một giải pháp	Tạo mẫu nhanh, các ứng dụng cần dóng hàng một bản ghi chép có sẵn, các quy trình làm việc đòi hỏi kiểm soát chi tiết về việc

		tất-cả-trong-một nhanh chóng, chính xác.	tinh chỉnh dấu thời gian.

Phần II: Biên giới của Phân đoạn dựa trên Ngữ điệu: Tạo ra Dòng chảy Tự nhiên

Phần này của báo cáo giải quyết khía cạnh thứ hai, tinh tế hơn trong truy vấn của người dùng: làm thế nào để sửa các khoảng lặng không chính xác. Sau khi đã xác định được ranh giới từ trong Phần I, chúng ta sẽ khám phá cách AI có thể hiểu được "âm nhạc" của lời nói để xác định nơi các khoảng lặng *nên* xuất hiện một cách tự nhiên, cho phép tái phân đoạn một cách thông minh.

Chương 5: Vượt ra ngoài Từ ngữ: Tầm quan trọng của Ngữ điệu Lời nói

Mục tiêu

Chương này nhằm định nghĩa ngữ điệu (prosody) và giải thích vai trò quan trọng của nó trong việc làm cho lời nói trở nên tự nhiên và dễ hiểu. Điều này cung cấp cơ sở lý thuyết cho các kỹ thuật tiên tiến được thảo luận trong phần này.

Các Yếu tố của Ngữ điệu

Ngữ điệu là sự kết hợp của nhiều khía cạnh của ngôn ngữ, thường được mô tả là "âm nhạc" hoặc "dòng chảy" của một ngôn ngữ.[37] Các thành phần chính bao gồm:

- **Cao độ/Ngữ điệu (Pitch/Intonation):** Sự thay đổi về tần số cơ bản (F_0) ảnh hưởng đến cách cảm nhận ngữ điệu. Ví dụ, một người đặt câu hỏi thường sẽ nâng cao độ ở cuối câu.[37]

- **Thời gian/Nhịp điệu (Timing/Rhythm):** Thời lượng của các âm vị, âm tiết và các khoảng lặng kiểm soát nhịp độ của lời nói. Việc sử dụng các khoảng lặng là rất quan trọng để tạo ra sự hồi hộp hoặc nhấn mạnh ý.[38]
- **Nhấn mạnh/Trọng âm (Emphasis/Stress):** Các mẫu trọng âm và nhấn mạnh làm nổi bật các từ quan trọng trong một câu, giúp truyền tải ý nghĩa và thu hút sự chú ý.[38]

Vấn đề của người dùng về "ngắt nghỉ chưa chính xác" về cơ bản là một vấn đề về thời gian ngữ điệu bị lỗi. Ngữ điệu không chỉ làm cho lời nói nghe tự nhiên hơn mà còn truyền tải các thông tin quan trọng ngoài từ ngữ, chẳng hạn như cảm xúc, cấu trúc cú pháp và ý định của người nói.[37]

Ngữ điệu và Phân đoạn

Người nghe một cách tự nhiên sử dụng các tín hiệu ngữ điệu—chẳng hạn như các khoảng lặng, thay đổi cao độ và thay đổi thời lượng—để phân đoạn dòng lời nói liên tục thành các đơn vị có ý nghĩa như cụm từ và mệnh đề.[41] Ví dụ, một khoảng dừng ngắn hoặc một sự giảm cao độ thường báo hiệu sự kết thúc của một đơn vị cú pháp. Đây chính là nguyên tắc mà chúng ta muốn AI mô phỏng để có thể tái phân đoạn âm thanh một cách thông minh. Bằng cách phân tích các đặc trưng ngữ điệu, một hệ thống AI có thể xác định các ranh giới tự nhiên trong lời nói, thay vì chỉ dựa vào sự im lặng tuyệt đối.

Chương 6: AI cho Hiểu biết Âm thanh-Ngữ nghĩa

Mục tiêu

Chương này giới thiệu các nghiên cứu tiên tiến về việc sử dụng AI để phân đoạn âm thanh dựa trên các tín hiệu sâu hơn, phi từ vựng, vượt ra ngoài việc phát hiện im lặng đơn giản hoặc dóng hàng từ.

Phân đoạn Không giám sát với Mô hình Ngôn ngữ Lời nói (SLM)

Một hướng tiếp cận mới lạ được trình bày trong bài báo arXiv 2501.03711 đề xuất một phương pháp phân đoạn lời nói không giám sát, tập trung vào sự thay đổi trong "phong cách âm thanh-ngữ nghĩa" (acoustic-semantic style).[43] Thay vì tập trung vào những thay đổi phổ trong tín hiệu (như phân đoạn âm vị), phương pháp này nhằm mục đích phân chia lời nói thành các đoạn có các đặc điểm khác nhau như cảm xúc, giới tính, hoặc người nói, mà không cần đến bản ghi chép văn bản.[43]

Quy trình ba phần được đề xuất như sau:

1. **Bộ phân câu (Sentencer):** Âm thanh đầu tiên được chia thành các đoạn nhỏ, có kích thước cố định được gọi là "câu âm thanh" (acoustic-sentences).
2. **Bộ tính điểm (Scorer):** Một Mô hình Ngôn ngữ Lời nói (Speech Language Model - SLM) được sử dụng để tính toán xác suất các câu âm thanh liên tiếp xuất hiện cùng nhau. Một xác suất thấp cho thấy một điểm ranh giới tự nhiên, nơi phong cách có thể đã thay đổi.
3. **Bộ chọn khoảng (Span-Selector):** Dựa trên các điểm số này, hệ thống quyết định giữ lại các ranh giới nào, hợp nhất các câu âm thanh thành các phân đoạn lớn hơn, có ý nghĩa.

Mặc dù phương pháp này được thiết kế cho các tác vụ không giám sát như phân loại cảm xúc, logic cơ bản của nó—sử dụng một mô hình ngôn ngữ để tìm các điểm gián đoạn tự nhiên trong dòng âm thanh—có thể được điều chỉnh để xác định các điểm dừng ngữ điệu tự nhiên.

Mô hình hóa Ngữ điệu cho TTS và ASR

Trong lĩnh vực Tổng hợp Tiếng nói từ Văn bản (Text-to-Speech - TTS), việc tạo ra ngữ điệu tự nhiên là một thách thức lớn. Các mô hình AI, đặc biệt là các mạng LSTM (Long Short-Term Memory), có thể được huấn luyện để dự đoán các đặc trưng ngữ điệu như thời lượng khoảng lặng, cao độ và cường độ dựa trên ngữ cảnh ngôn ngữ.[38]

Ví dụ, một mô hình có thể học được rằng các danh từ và động từ (từ nội dung) thường được nhấn mạnh hơn các mạo từ và giới từ (từ chức năng).[38] Tương tự, nó có thể học cách chèn các khoảng lặng ngắn sau dấu phẩy và các khoảng lặng dài hơn ở cuối câu. Phân tích cú pháp có thể giúp xác định cấu trúc của câu, cho phép hệ thống thêm các khoảng lặng giữa các mệnh đề, bắt chước các mẫu nói của con người.[38] Mặc dù các mô hình này thường được sử dụng để *tạo ra* lời nói tổng hợp, logic dự đoán bên trong chúng chứa đựng kiến thức cần thiết để xác định các mẫu ngắt nghỉ tự nhiên từ văn bản. Logic này có thể được tái sử dụng để thông báo cho quá trình tái phân đoạn của chúng ta.

Chương 7: Một Quy trình làm việc Hai giai đoạn được Đề xuất để Tái phân đoạn Tự nhiên

Mục tiêu

Chương này tổng hợp các phát hiện từ toàn bộ báo cáo thành một quy trình làm việc mới, có thể hành động, giải quyết trực tiếp vấn đề phức tạp của người dùng.

Giải pháp mạnh mẽ nhất cho vấn đề của người dùng không nằm ở một công cụ duy nhất, mà là một quy trình hai giai đoạn tận dụng các thế mạnh riêng biệt của các công nghệ đã được thảo luận. Phần I cung cấp các công cụ để xác định *cái gì* đã được nói và *khi nào*, trong khi Phần II cung cấp các nguyên tắc để xác định *cách* nó nên được điều chỉnh nhịp độ. Bằng cách kết hợp chúng, chúng ta có thể sửa chữa âm thanh nguồn bị lỗi.

Quá trình này bắt đầu bằng việc chấp nhận rằng các khoảng lặng trong âm thanh gốc là không đáng tin cậy. Do đó, bước đầu tiên là phải bỏ qua chúng hoàn toàn và thiết lập một chân lý nền tảng chỉ dựa trên nội dung lời nói. Các công cụ dóng hàng chính xác cao như WhisperX hoặc MFA, khi được cung cấp bản ghi chép văn bản, sẽ thực hiện chính xác điều này. Chúng sẽ xác định thời gian bắt đầu và kết thúc của mỗi từ, tạo ra một danh sách các "viên gạch" âm thanh không có các khoảng lặng sai. Khi đã có các khối xây dựng cơ bản này, giai đoạn thứ hai bắt đầu. Giai đoạn này sử dụng trí thông minh về ngữ điệu để quyết định cách lắp ráp lại các viên gạch này. Bằng cách phân tích cấu trúc ngôn ngữ của văn bản và các đặc trưng âm học của các đoạn từ đã được dóng hàng, một mô hình ngữ điệu có thể dự đoán thời lượng và vị trí của các khoảng lặng tự nhiên. Cuối cùng, âm thanh được tái tạo bằng cách ghép các đoạn từ đã được cắt ra với các khoảng lặng được tạo ra một cách nhân tạo, tạo ra một sản phẩm cuối cùng có dòng chảy tự nhiên và dễ hiểu.

Các bước của Quy trình làm việc được Đề xuất

1. **Giai đoạn 1: Dóng hàng Từ với Độ chính xác cao.**
    - **Đầu vào:** Tệp âm thanh gốc của người dùng và bản ghi chép văn bản tương ứng.
    - **Quy trình:** Sử dụng một công cụ dóng hàng có độ chính xác cao. **WhisperX** được khuyến nghị vì sự cân bằng giữa tính dễ sử dụng và độ chính xác. Đối với các ứng dụng đòi hỏi độ chính xác tối đa, **MFA** là lựa chọn thay thế. Công cụ này sẽ xử lý âm

thanh và văn bản để tạo ra một danh sách cuối cùng của tất cả các từ cùng với thời gian bắt đầu và kết thúc chính xác của chúng trong âm thanh gốc.
    - **Đầu ra:** Một định dạng dữ liệu có cấu trúc, ví dụ như JSON, chứa thông tin về từng từ. Ví dụ: [{"word": "Xin", "start": 0.5, "end": 0.8}, {"word": "chào", "start": 2.1, "end": 2.5},...]. Lưu ý khoảng cách thời gian không tự nhiên giữa các từ, phản ánh các khoảng lặng bị lỗi trong bản gốc.
2. **Giai đoạn 2: Dự đoán Khoảng lặng dựa trên Ngữ điệu và Tái tạo Âm thanh.**
    - **Đầu vào:** Dữ liệu từ-dấu thời gian có cấu trúc từ Giai đoạn 1 và bản ghi chép văn bản gốc.
    - **Quy trình A (Phân tích Ngôn ngữ):** Phân tích bản ghi chép văn bản để xác định các ranh giới cú pháp, chẳng hạn như dấu phẩy, dấu chấm, và cuối mệnh đề. Đây là những chỉ báo mạnh mẽ cho thấy nơi cần có các khoảng lặng.
    - **Quy trình B (Trích xuất Đặc trưng Ngữ điệu):** Đối với mỗi đoạn từ được xác định trong Giai đoạn 1, sử dụng một thư viện như librosa [47] để trích xuất các đặc trưng ngữ điệu như đường cong cao độ (F_0) và cường độ. Một cao độ giảm dần ở cuối một phân đoạn là một chỉ báo mạnh mẽ khác cho thấy cần có một khoảng lặng.
    - **Quy trình C (Mô hình hóa Thời lượng Khoảng lặng):** Áp dụng một mô hình để dự đoán thời lượng khoảng lặng thích hợp. Mô hình này có thể là một hệ thống dựa trên quy tắc đơn giản (ví dụ: 200ms cho dấu phẩy, 500ms cho dấu chấm) hoặc một mạng nơ-ron được huấn luyện để dự đoán thời lượng dựa trên các đầu vào từ Quy trình A và B.
    - **Quy trình D (Cắt và Tái tạo Âm thanh):** Tạo một tệp âm thanh mới, sạch. Điều này được thực hiện bằng cách lập trình trích xuất từng đoạn âm thanh của từ từ tệp gốc bằng cách sử dụng các dấu thời gian chính xác của nó, sau đó ghép chúng lại với nhau, chèn vào giữa các khoảng lặng im lặng có thời lượng đã được dự đoán từ Quy trình C.
    - **Đầu ra:** Một tệp âm thanh mới với cùng các từ được nói nhưng có các khoảng lặng tự nhiên, do AI tạo ra, sửa chữa ngữ điệu bị lỗi của bản ghi âm gốc.

Phần III: Hướng dẫn Triển khai Thực tế

Phần cuối cùng này cung cấp mã nguồn có thể hành động và các bước tiền xử lý cần thiết để triển khai các giải pháp đã thảo luận, giúp người dùng chuyển từ giai đoạn lên ý tưởng sang thực thi.

Chương 8: Bước Đầu tiên Quan trọng - Chuẩn hóa Văn bản

Mục tiêu

Chương này giải thích tại sao chuẩn hóa văn bản là một bước tiền xử lý không thể thiếu để dóng hàng chính xác và cung cấp hướng dẫn để triển khai nó.

Vấn đề về các Dạng không Khớp

Các mô hình ASR tạo ra đầu ra ở dạng nói (ví dụ: "một trăm hai mươi ba đô la"), trong khi các bản ghi chép văn bản thường chứa các dạng ký hiệu (ví dụ: "$123").[49, 50] Sự không khớp này sẽ khiến bất kỳ thuật toán dóng hàng nào cũng thất bại, vì nó không thể tìm thấy sự tương ứng giữa "đô la" trong âm thanh và "$" trong văn bản.

Để giải quyết vấn đề này, chúng ta cần thực hiện Chuẩn hóa Văn bản (Text Normalization - TN), là quá trình chuyển đổi văn bản viết sang dạng nói của nó. Ngược lại, Chuẩn hóa Văn bản Nghịch đảo (Inverse Text Normalization - ITN) chuyển đổi dạng nói sang dạng viết, thường được sử dụng trong hậu xử lý ASR.[49] Đối với dóng hàng cưỡng bức, chúng ta cần TN.

Các Tác vụ Chuẩn hóa Chính

Một quy trình chuẩn hóa văn bản toàn diện bao gồm nhiều bước để đảm bảo tính nhất quán:

- **Chuyển đổi chữ hoa/thường:** Chuyển đổi toàn bộ văn bản sang chữ thường để loại bỏ sự khác biệt về cách viết hoa.[51]
- **Xử lý Dấu câu:** Loại bỏ hoặc thay thế dấu câu không mang thông tin ngữ nghĩa quan trọng cho việc dóng hàng.[52]
- **Mở rộng các từ viết tắt:** Chuyển đổi các từ viết tắt sang dạng đầy đủ của chúng (ví dụ: "Dr." -> "Doctor").[54]
- **Chuẩn hóa Số, Tiền tệ và Ngày tháng:** Chuyển đổi các ký hiệu số thành từ ngữ (ví dụ: "123" -> "one hundred twenty-three", "$50" -> "fifty dollars", "June 3rd" -> "June third").[49]

Các Thư viện Python cho Chuẩn hóa Văn bản

Nhiều thư viện Python có thể tự động hóa quá trình này.

- **NVIDIA NeMo Text Processing:** Đây là một bộ công cụ mạnh mẽ, cấp độ sản xuất, cung cấp cả các tùy chọn dựa trên quy tắc (WFST) và dựa trên mạng nơ-ron. Nó hỗ trợ nhiều ngôn ngữ và có khả năng tùy chỉnh cao, làm cho nó trở thành một lựa chọn tuyệt vời cho các quy trình làm việc đòi hỏi sự mạnh mẽ.[49]
- **whisper-normalizer:** Đây là một thư viện nhẹ, triển khai chính xác các quy tắc chuẩn hóa được sử dụng trong bài báo gốc của Whisper. Điều này làm cho nó trở thành một lựa chọn lý tưởng để tiền xử lý văn bản một cách nhất quán cho các quy trình làm việc dựa trên Whisper, đảm bảo rằng văn bản đầu vào khớp với những gì mô hình Whisper mong đợi.[60]

Chương 9: Các Mẫu Triển khai

Mục tiêu

Chương này cung cấp các ví dụ mã Python rõ ràng, có chú thích, từng bước cho các công cụ dóng hàng được khuyến nghị, cho phép triển khai nhanh chóng.

Mẫu 1: Dóng hàng Độ chính xác cao với WhisperX

Mã nguồn sau đây minh họa quy trình làm việc hoàn chỉnh để nhận dạng, dóng hàng và tùy chọn phân loại người nói bằng WhisperX.

Python

```
import whisperx
import gc
```

```
# Chọn thiết bị (GPU được khuyến nghị)
device = "cuda"
audio_file = "path/to/your/audio.wav"
batch_size = 16 # Giảm nếu bộ nhớ GPU thấp
compute_type = "float16" # Thay đổi thành "int8" nếu bộ nhớ GPU thấp

# 1. Nhận dạng với mô hình Whisper gốc (xử lý hàng loạt)
# Tải mô hình Whisper (ví dụ: 'large-v2')
model = whisperx.load_model("large-v2", device, compute_type=compute_type)

# Tải âm thanh
audio = whisperx.load_audio(audio_file)

# Thực hiện nhận dạng
result = model.transcribe(audio, batch_size=batch_size)
print("Bản ghi chép ban đầu (trước khi dóng hàng):")
print(result["segments"])

# Giải phóng bộ nhớ nếu cần
# gc.collect(); torch.cuda.empty_cache(); del model

# 2. Dóng hàng đầu ra của Whisper để có dấu thời gian cấp độ từ
# Tải mô hình dóng hàng cho ngôn ngữ đã phát hiện
model_a, metadata = whisperx.load_align_model(language_code=result["language"],
device=device)

# Thực hiện dóng hàng
result = whisperx.align(result["segments"], model_a, metadata, audio, device,
return_char_alignments=False)

print("\nBản ghi chép đã dóng hàng (với dấu thời gian cấp độ từ):")
# In ra các phân đoạn với các từ đã được dóng hàng
for segment in result["segments"]:
    print(f"Segment: {segment['start']:.2f}s -> {segment['end']:.2f}s")
    for word in segment['words']:
        print(f"  - {word['word']} ({word['start']:.2f}s -> {word['end']:.2f}s)")

# Giải phóng bộ nhớ nếu cần
# gc.collect(); torch.cuda.empty_cache(); del model_a

# 3. (Tùy chọn) Gán nhãn người nói (Phân loại người nói)
# Yêu cầu mã thông báo truy cập Hugging Face
# YOUR_HF_TOKEN = "hf_..."
```

```
# diarize_model = whisperx.DiarizationPipeline(use_auth_token=YOUR_HF_TOKEN, device=device)
# diarize_segments = diarize_model(audio)
# result = whisperx.assign_word_speakers(diarize_segments, result)
# print("\nBản ghi chép với người nói được xác định:")
# print(result["segments"])
```

Dựa trên các ví dụ trong [24]

Mẫu 2: Dóng hàng Linh hoạt với stable-ts

Mã nguồn sau đây minh họa hai trường hợp sử dụng chính của stable-ts: nhận dạng và dóng hàng từ một tệp âm thanh, và dóng hàng một bản ghi chép văn bản đã có sẵn.

Python

```
import stable_whisper

# Tải mô hình (ví dụ: 'base')
model = stable_whisper.load_model('base')

# --- Trường hợp 1: Nhận dạng và Dóng hàng từ Âm thanh ---
print("--- Trường hợp 1: Nhận dạng và Dóng hàng ---")
audio_file = 'path/to/your/audio.mp3'

# Thực hiện nhận dạng, bật VAD để có kết quả tốt hơn
# suppress_silence=True là mặc định và giúp tinh chỉnh dấu thời gian
result_transcribe = model.transcribe(audio_file, vad=True)

# In ra các từ với dấu thời gian
for segment in result_transcribe.segments:
    for word in segment.words:
        print(f"[{word.start:.2f}s -> {word.end:.2f}s] {word.word}")

# Lưu kết quả vào tệp phụ đề
result_transcribe.to_srt_vtt('output_transcribed.srt')
print("\nĐã lưu kết quả nhận dạng vào 'output_transcribed.srt'")
```

```
# --- Trường hợp 2: Dóng hàng một Bản ghi chép Văn bản có sẵn ---
print("\n--- Trường hợp 2: Dóng hàng Văn bản có sẵn ---")

# Giả sử bạn có một bản ghi chép đã được chuẩn hóa
existing_transcript = "đây là văn bản đã được chuẩn hóa để dóng hàng với tệp âm thanh."

# Sử dụng hàm align() để dóng hàng văn bản này với âm thanh
# Điều này nhanh hơn nhiều so với việc nhận dạng lại từ đầu
result_align = model.align(audio_file, existing_transcript, language='vi') # Chỉ định ngôn ngữ nếu cần

# In ra các từ đã được dóng hàng
for segment in result_align.segments:
    for word in segment.words:
        print(f"[{word.start:.2f}s -> {word.end:.2f}s] {word.word}")

# Lưu kết quả dóng hàng
result_align.to_srt_vtt('output_aligned.srt')
print("\nĐã lưu kết quả dóng hàng vào 'output_aligned.srt'")
```

Dựa trên các ví dụ trong [29]

Chương 10: Đánh giá và Xác thực Chất lượng Phân đoạn

Mục tiêu

Chương này giới thiệu ngắn gọn các phương pháp để đánh giá chất lượng của kết quả dóng hàng và âm thanh cuối cùng, cung cấp một khuôn khổ để xác thực kết quả.

Các Chỉ số Định lượng để Phát hiện Ranh giới

Trong nghiên cứu học thuật, chất lượng của việc phát hiện ranh giới từ thường được đo lường bằng các chỉ số tiêu chuẩn. Mặc dù người dùng cuối có thể không cần phải tự tính toán các chỉ số này, việc hiểu chúng sẽ cung cấp bối cảnh để so sánh hiệu suất của các công cụ khác

nhau.

- **Precision, Recall, và F1-score:** Đây là các chỉ số phổ biến nhất. Một ranh giới từ được dự đoán được coi là "đúng" (true positive) nếu nó nằm trong một cửa sổ dung sai nhất định (ví dụ: 200ms) so với ranh giới trong dữ liệu chân lý nền tảng.[21]
- **R-value:** Chỉ số này được sử dụng để đo lường mức độ phân đoạn quá mức (over-segmentation) hoặc phân đoạn dưới mức (under-segmentation), cung cấp một cái nhìn về sự cân bằng của thuật toán.[61]

Xác thực Định tính

Đối với hầu hết các mục đích thực tế, phương pháp xác thực hiệu quả nhất là kiểm tra bằng thị giác và thính giác.

- **Kiểm tra bằng Thị giác và Thính giác:** Cách tiếp cận thực tế nhất là sử dụng một trình chỉnh sửa âm thanh như Audacity hoặc một công cụ phân tích âm vị học như Praat.[12] Người dùng có thể tải tệp âm thanh cùng với tệp TextGrid được tạo ra (ví dụ: từ MFA hoặc được chuyển đổi từ đầu ra của WhisperX). Bằng cách phóng to dạng sóng, người dùng có thể kiểm tra trực quan xem các ranh giới từ được đánh dấu có khớp với các sự kiện âm thanh thực tế hay không.
- **Đánh giá Âm thanh Tái tạo:** Đối với quy trình làm việc hai giai đoạn được đề xuất trong Chương 7, bước xác thực cuối cùng là nghe tệp âm thanh đã được tái tạo. Mục tiêu là đánh giá xem các khoảng lặng được chèn vào có tạo ra một dòng chảy tự nhiên, dễ nghe hay không, và liệu nhịp điệu tổng thể có được cải thiện so với bản gốc hay không.

Kết luận: Tổng hợp Giải pháp và Hướng phát triển trong Tương lai

Tóm tắt các Phát hiện

Báo cáo này đã khám phá sâu vào các phương pháp tiên tiến do AI cung cấp để phân đoạn và chỉnh sửa âm thanh, tập trung vào việc giải quyết thách thức về các khoảng lặng không chính xác trong một bản ghi âm. Phân tích cho thấy một sự đánh đổi cốt lõi giữa các mô hình cổ điển và hiện đại: các hệ thống dựa trên HMM như MFA cung cấp độ chính xác thời gian vượt

trội, trong khi các mô hình đầu cuối như Whisper mang lại độ chính xác nhận dạng hàng đầu. Các giải pháp lai, chẳng hạn như WhisperX và stable-ts, đã nổi lên như một con đường thực tế phía trước, kết hợp những điểm mạnh của cả hai cách tiếp cận để cung cấp cả bản ghi chép chính xác và dấu thời gian đáng tin cậy.

Hơn nữa, báo cáo đã đề xuất một quy trình làm việc hai giai đoạn sáng tạo để giải quyết hoàn toàn vấn đề của người dùng. Giai đoạn đầu tiên tập trung vào việc đạt được sự dóng hàng từ chính xác bằng cách sử dụng các công cụ lai này, thiết lập một chân lý nền tảng về nội dung và thời gian của lời nói. Giai đoạn thứ hai đi sâu vào lĩnh vực ngữ điệu, sử dụng các nguyên tắc phân tích ngôn ngữ và âm học để dự đoán và chèn các khoảng lặng tự nhiên. Quy trình tổng hợp này—**Dóng hàng Chính xác + Chỉnh sửa Ngữ điệu**—đại diện cho một giải pháp toàn diện nhất, có khả năng biến một bản ghi âm có nhịp điệu kém thành một sản phẩm âm thanh có dòng chảy tự nhiên và chuyên nghiệp.

Các Khuyến nghị theo Từng cấp độ

Dựa trên phân tích, các khuyến nghị sau đây được đưa ra, được điều chỉnh cho các nhu cầu và ưu tiên khác nhau của dự án:

- **Đối với Độ chính xác Tối đa:** Nếu độ chính xác thời gian tuyệt đối là ưu tiên hàng đầu và thời gian phát triển ít quan trọng hơn, hãy sử dụng **Montreal Forced Aligner (MFA)**. Phương pháp này đòi hỏi sự chuẩn bị dữ liệu cẩn thận nhưng mang lại các ranh giới từ đáng tin cậy nhất, phù hợp cho nghiên cứu học thuật hoặc các ứng dụng pháp lý.
- **Đối với Sự cân bằng Tốt nhất giữa Độ chính xác và Tính dễ sử dụng:** Giải pháp chính được khuyến nghị là **WhisperX**. Nó cung cấp khả năng nhận dạng xuất sắc, dấu thời gian chất lượng cao và các tính năng bổ sung như phân loại người nói trong một gói duy nhất, dễ sử dụng. Đây là lựa chọn lý tưởng cho hầu hết các ứng dụng đa dụng như tạo phụ đề, chỉnh sửa podcast và video.
- **Đối với Sự linh hoạt và Tinh chỉnh Tối đa:** Nếu người dùng cần dóng hàng một bản ghi chép đã có sẵn hoặc muốn kiểm soát chi tiết quá trình hậu xử lý, **stable-ts** là một lựa chọn xuất sắc. Hàm align() mạnh mẽ của nó và các tùy chọn tinh chỉnh nâng cao (ví dụ: VAD, triệt tiêu im lặng) làm cho nó trở thành một công cụ cực kỳ linh hoạt để tạo mẫu nhanh và các quy trình làm việc tùy chỉnh.

Hướng phát triển trong Tương lai

Lĩnh vực công nghệ giọng nói đang phát triển với tốc độ chóng mặt. Những tiến bộ nhanh chóng trong các Mô hình Ngôn ngữ Lời nói (SLM) và mô hình hóa ngữ điệu cho thấy rằng

trong tương lai, một mô hình đầu cuối duy nhất có thể thực hiện đồng thời cả việc dóng hàng có độ chính xác cao và chỉnh sửa ngữ điệu. Một mô hình như vậy sẽ hợp nhất hai giai đoạn của quy trình làm việc được đề xuất, đơn giản hóa đáng kể quá trình tạo ra âm thanh tự nhiên, được phân đoạn hoàn hảo từ các bản ghi âm thô. Khi các mô hình này trở nên tinh vi hơn trong việc hiểu không chỉ nội dung từ vựng mà cả các sắc thái âm thanh-ngữ nghĩa của lời nói, ranh giới giữa nhận dạng, dóng hàng và tổng hợp sẽ tiếp tục mờ đi, mở ra những khả năng mới cho việc tạo và chỉnh sửa nội dung âm thanh tự động.

Nguồn trích dẫn

1. Tradition or Innovation: A Comparison of Modern ASR Methods for Forced Alignment - arXiv, truy cập vào tháng 11 2, 2025, https://arxiv.org/html/2406.19363v1
2. docs.pytorch.org, truy cập vào tháng 11 2, 2025, https://docs.pytorch.org/audio/main/tutorials/ctc_forced_alignment_api_tutorial.html#:~:text=The%20forced%20alignment%20is%20a,Speech%20Technology%20to%201%2C000%2B%20Languages.
3. Glossary — Montreal Forced Aligner 3.0.0 documentation, truy cập vào tháng 11 2, 2025, https://montreal-forced-aligner.readthedocs.io/en/v3.2.3/user_guide/glossary.html
4. The Mason-Alberta Phonetic Segmenter: a forced alignment system based on deep neural networks and interpolation - PMC - NIH, truy cập vào tháng 11 2, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11449383/
5. Human Assisted Speaker Recognition Using Forced Alignments on HMM - International Journal of Engineering Research & Technology, truy cập vào tháng 11 2, 2025, https://www.ijert.org/research/human-assisted-speaker-recognition-using-forced-alignments-on-hmm-IJERTV2IS90739.pdf
6. Forced alignment HMM - hidden markov model - Stats StackExchange, truy cập vào tháng 11 2, 2025, https://stats.stackexchange.com/questions/270141/forced-alignment-hmm
7. How does forced alignment work? - Conversational AI - Research at NVIDIA, truy cập vào tháng 11 2, 2025, https://research.nvidia.com/labs/conv-ai/blogs/2023/2023-08-forced-alignment/
8. Whisper Has an Internal Word Aligner - arXiv, truy cập vào tháng 11 2, 2025, https://arxiv.org/html/2509.09987v1
9. Performance of Forced-Alignment Algorithms on Children's Speech - PMC - NIH, truy cập vào tháng 11 2, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC8740721/
10. Montreal Forced Aligner Tutorial - Miao ZHANG, truy cập vào tháng 11 2, 2025, https://miaozhang.org/2025-04-28-montreal-forced-aligner-tutorial/
11. User Guide — Montreal Forced Aligner 3.0.0 documentation, truy cập vào tháng 11 2, 2025, https://montreal-forced-aligner.readthedocs.io/en/v3.3.0/user_guide/index.html
12. Introduction to Montreal Forced Aligner - Scott Nelson, truy cập vào tháng 11 2, 2025, https://www.scott-nelson.net/MFA.html

13. Why Whisper's Timestamps Are Inaccurate and How WhisperSync Solves It, truy cập vào tháng 11 2, 2025, https://whispersync.unicornplatform.page/blog/why-whispers-timestamps-are-inaccurate-and-how-whispersync-solves-it/
14. [D] Speech to Text Word Level Timestamps Accuracy Issue : r/MachineLearning - Reddit, truy cập vào tháng 11 2, 2025, https://www.reddit.com/r/MachineLearning/comments/1cbd8x1/d_speech_to_text_word_level_timestamps_accuracy/
15. Is it possible to achieve the transcript accuracy of Whisper with the timestamp accuracy of Vosk in speech-to-text tasks? - AI Stack Exchange, truy cập vào tháng 11 2, 2025, https://ai.stackexchange.com/questions/42014/is-it-possible-to-achieve-the-transcript-accuracy-of-whisper-with-the-timestamp
16. Montreal-forced-aligner - montrealcorpustools.github.io, truy cập vào tháng 11 2, 2025, https://montrealcorpustools.github.io/Montreal-Forced-Aligner/
17. MontrealCorpusTools/Montreal-Forced-Aligner: Command line utility for forced alignment using Kaldi - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner
18. 3 Montreal Forced Aligner | Corpus Phonetics Tutorial - Eleanor Chodroff, truy cập vào tháng 11 2, 2025, https://eleanorchodroff.com/tutorial/montreal-forced-aligner.html
19. LingMethodsHub - Montreal Forced Aligner - Linguistics Methods Hub, truy cập vào tháng 11 2, 2025, https://lingmethodshub.github.io/content/tools/mfa/mfa-tutorial/
20. All commands — Montreal Forced Aligner 2.0.0 documentation - Read the Docs, truy cập vào tháng 11 2, 2025, https://montreal-forced-aligner.readthedocs.io/en/v2.1.7/user_guide/commands.html
21. WhisperX: Time-Accurate Speech Transcription of ... - ISCA Archive, truy cập vào tháng 11 2, 2025, https://www.isca-archive.org/interspeech_2023/bain23_interspeech.pdf
22. openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/openai/whisper
23. Word Level Timestamp Generation for Automatic Speech Recognition and Translation, truy cập vào tháng 11 2, 2025, https://arxiv.org/html/2505.15646v1
24. m-bain/whisperX: WhisperX: Automatic Speech ... - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/m-bain/whisperX
25. WhisperX: Word-level timestamps, diarization (new), batch inference within file(new) · openai whisper · Discussion #684 - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/openai/whisper/discussions/684
26. Interview transcription using WhisperX model, Part 1. - Valor Software, truy cập vào tháng 11 2, 2025, https://valor-software.com/articles/interview-transcription-using-whisperx-model-part-1
27. WhisperX - AI Cloud Automation, truy cập vào tháng 11 2, 2025,

https://aicloudautomation.net/projects/whisperx/
28. Word-level timestamps from WhisperX are inaccurate compared to ..., truy cập vào tháng 11 2, 2025, https://github.com/m-bain/whisperX/issues/1247
29. jianfch/stable-ts: Transcription, forced alignment, and audio ... - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/jianfch/stable-ts
30. kullup/whisper-timestamped - Hugging Face, truy cập vào tháng 11 2, 2025, https://huggingface.co/kullup/whisper-timestamped
31. Stabilizing Timestamps for Whisper - stable-ts · PyPI, truy cập vào tháng 11 2, 2025, https://pypi.org/project/stable-ts/2.5.3/
32. stable-ts - Stabilizing Timestamps for Whisper - PyPI, truy cập vào tháng 11 2, 2025, https://pypi.org/project/stable-ts/2.13.3/
33. rpayanm/stable-ts - Hugging Face, truy cập vào tháng 11 2, 2025, https://huggingface.co/rpayanm/stable-ts
34. Improving Timestamp Accuracy · openai whisper · Discussion #435 - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/openai/whisper/discussions/435
35. How to map word level timestamps to text of a given transcript? - Stack Overflow, truy cập vào tháng 11 2, 2025, https://stackoverflow.com/questions/76574488/how-to-map-word-level-timestamps-to-text-of-a-given-transcript
36. Fluency - Prosody Features, truy cập vào tháng 11 2, 2025, https://docs.soapboxlabs.com/technical-docs/online-technical-documentation/fluency-(online)/fluency-prosody-features
37. Mastering Prosody Modeling in AI Voiceovers: A Comprehensive Guide for Video Producers, truy cập vào tháng 11 2, 2025, https://kveeky.com/blog/ai-voiceover-prosody-modeling-for-video-producers
38. Voice Synthesis Improvement by Machine Learning of Natural Prosody - PMC - NIH, truy cập vào tháng 11 2, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10934073/
39. The Role of Prosody in Spoken Question Answering - ACL Anthology, truy cập vào tháng 11 2, 2025, https://aclanthology.org/2025.findings-naacl.471.pdf
40. Speech Segmentation - Artificial Intelligence, truy cập vào tháng 11 2, 2025, https://schneppat.com/speech-segmentation.html
41. Speech Prosody Serves Temporal Prediction of Language via Contextual Entrainment - PMC - PubMed Central, truy cập vào tháng 11 2, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC11236583/
42. Unsupervised Speech Segmentation: A General Approach Using Speech Language Models, truy cập vào tháng 11 2, 2025, https://arxiv.org/html/2501.03711v1
43. Unsupervised Speech Segmentation: A General Approach Using Speech Language Models - arXiv, truy cập vào tháng 11 2, 2025, https://arxiv.org/pdf/2501.03711?
44. [2501.03711] Unsupervised Speech Segmentation: A General Approach Using Speech Language Models - arXiv, truy cập vào tháng 11 2, 2025, https://arxiv.org/abs/2501.03711
45. [Literature Review] Unsupervised Speech Segmentation: A General Approach

Using Speech Language Models - Moonlight, truy cập vào tháng 11 2, 2025, https://www.themoonlight.io/en/review/unsupervised-speech-segmentation-a-general-approach-using-speech-language-models
46. librosa.display.specshow — librosa 0.11.0 documentation, truy cập vào tháng 11 2, 2025, http://librosa.org/doc/0.11.0/generated/librosa.display.specshow.html
47. librosa 0.11.0 documentation, truy cập vào tháng 11 2, 2025, https://librosa.org/doc/
48. Text Normalization and Inverse Text Normalization with NVIDIA NeMo | NVIDIA Technical Blog, truy cập vào tháng 11 2, 2025, https://developer.nvidia.com/blog/text-normalization-and-inverse-text-normalization-with-nvidia-nemo/
49. Text Normalization for Voice AI: Complete Guide to Speech Preprocessing in 2025 - Vapi, truy cập vào tháng 11 2, 2025, https://vapi.ai/blog/text-normalization
50. Text Normalization in NLP: Techniques & Best Practices - Galaxy, truy cập vào tháng 11 2, 2025, https://www.getgalaxy.io/learn/glossary/text-normalization-for-nlp
51. Normalizing Textual Data with Python - GeeksforGeeks, truy cập vào tháng 11 2, 2025, https://www.geeksforgeeks.org/python/normalizing-textual-data-with-python/
52. Systematic Review on Text Normalization Techniques and its Approach to Non-Standard Words - ResearchGate, truy cập vào tháng 11 2, 2025, https://www.researchgate.net/publication/374166354_Systematic_Review_on_Text_Normalization_Techniques_and_its_Approach_to_Non-Standard_Words
53. Text Normalization - Devopedia, truy cập vào tháng 11 2, 2025, https://devopedia.org/text-normalization
54. NeMo text processing for ASR and TTS - GitHub, truy cập vào tháng 11 2, 2025, https://github.com/NVIDIA/NeMo-text-processing
55. Text_Normalization_Tutorial.ipynb - Colab - Google, truy cập vào tháng 11 2, 2025, https://colab.research.google.com/github/NVIDIA/NeMo/blob/r1.0.0rc1/tutorials/tools/Text_Normalization_Tutorial.ipynb
56. Neural Text Normalization Models — NVIDIA NeMo Framework User Guide, truy cập vào tháng 11 2, 2025, https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/nlp/text_normalization/nn_text_normalization.html
57. Text (Inverse) Normalization — NVIDIA NeMo Framework User Guide, truy cập vào tháng 11 2, 2025, https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/nlp/text_normalization/wfst/wfst_text_normalization.html
58. whisper-normalizer · PyPI, truy cập vào tháng 11 2, 2025, https://pypi.org/project/whisper-normalizer/
59. arXiv:2411.10423v1 [cs.LG] 15 Nov 2024, truy cập vào tháng 11 2, 2025, https://arxiv.org/pdf/2411.10423