ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN

BÁO CÁO ĐỒ ÁN
MÔN XỬ LÝ THÔNG TIN GIỌNG NÓI
Đề tài: FAST CONFORMER WITH LINEARLY SCALABLE ATTENTION FOR EFFICIENT SPEECH RECOGNITION

GVHD: ThS. Nguyễn Thành Luân

Nhóm sinh viên thực hiện:

1. Phan Lê Chí Trung	MSSV: 21522725
2. Trần Nguyên Mẫn	MSSV: 21522325
3. Đỗ Phú Duy	MSSV: 21520205
4. Nguyễn Phúc Hào	MSSV: 21522047
5. Nguyễn Ngọc Lương	MSSV: 21522311

Tp. Hồ Chí Minh, 05/2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

……, ngày…….....tháng……năm 20…

Người nhận xét

(Ký tên và ghi rõ họ tên)

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:

Bảng 0.1 Bảng phân công, đánh giá thành viên

Họ và tên	MSSV	Phân công	Đánh giá
Phan Lê Chí Trung	21522725	- Tóm tắt nội dung bài báo - Chạy thực nghiệm - Kiểm tra nội dung báo cáo và thuyết trình	- Hoàn thành công việc được giao - Tỉ lệ đóng góp: 20%
Trần Nguyên Mẫn	21522325	- Tóm tắt nội dung bài báo - Chạy thực nghiệm - Làm slide	- Hoàn thành công việc được giao - Tỉ lệ đóng góp: 20%
Đỗ Phú Duy	21520205	- Tóm tắt nội dung bài báo - Chạy thực nghiệm - Soạn word	- Hoàn thành công việc được giao - Tỉ lệ đóng góp: 20%
Nguyễn Phúc Hào	21522047	- Tóm tắt nội dung bài báo - Chạy thực nghiệm - Soạn word	- Hoàn thành công việc được giao - Tỉ lệ đóng góp: 20%
Nguyễn Ngọc Lương	21522311	- Tóm tắt nội dung bài báo - Chạy thực nghiệm - Soạn word	- Hoàn thành công việc được giao - Tỉ lệ đóng góp: 20%

LỜI MỞ ĐẦU

Conformer là mô hình Transducer (RNNT) cho nhận diện giọng nói tự động được đề xuất bởi Gulati và các cộng sự. Conformer đã được ứng dụng nhiều trong công nghiệp, đặc biệt là về streaming ASR trên thiết bị hoặc trên cloud. Tuy nhiên conformer vẫn còn một số hạn chế nhất định, tiêu biểu là sử dụng nhiều bộ nhớ và tài nguyên hơn các mô hình ASR chỉ sử dụng các lớp convolution và các mô hình Scaling Conformer đòi hỏi phải điều chỉnh các kernel kernel size trong Conformer block để ổn định quá trình huấn luyện mô hình lớn.

Để giải quyết các thách thức kể trên, các tác giả của bài báo Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition đã thiết kế lại kiến trúc của Conformer và đề xuất một mô hình mới là Fast Conformer.

Để hiểu rõ hơn về mô hình Fast Conformer này, nhóm sẽ trình bày qua các chương sau:

- Chương 1: Giới thiệu chung
- Chương 2: Fast Conformer
- Chương 3: Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition
- Chương 4: Kết luận

DANH MỤC CÁC BẢNG, HÌNH ẢNH

Danh mục các bảng:

Danh mục hình ảnh:

MỤC LỤC

Chương 1: GIỚI THIỆU CHUNG

1.1 Giới thiệu về môn học

Tiếng nói là thứ làm loài người trở nên độc nhất cho với các loài động vật khác, khi con người có một tiếng nói cụ thể thay vì các tiếng kêu như các loại động vật khác.

Tiếng nói là thứ làm con người phát triển hơn so với các loài động vật khác, tiếng nói là thứ giúp con người được coi là động vật bậc cao. Tiếng nói có vai trò quan trọng đối với con người trong cả quá trình từ xã hội nguyên thủy đi qua các thời kỳ công xã nguyên thủy, công xã bộ lạc, phong kiến, đến thời trung đại hiện đại, qua các thời kỳ công nghiệp hóa, hiện đại hóa 1, 2, 3, 4 tiếng nói luôn đóng vai trò cực kỳ quan trọng. Nếu không có tiếng nói con người sẽ không thể đạt đến các thành tựu khoa học kỳ thuật...

Trong môn học Xử lý thông tin giọng nói giúp chúng ta tìm hiểu rõ hơn về tiếng nói của loài người, tại sao con người lại có tiếng nói và tại sao các loài động vật khác không có. Chúng ta được học về cách con người tạo ra giọng nói và tiếng nói, chuyên sâu về tiếng nói và đặc biệt là chúng ta có thể làm gì với dữ liệu giọng nói, tiếng nói, về những gì chúng ta có thể khai thác, có thể nhận được khi nghiên cứu về tiếng nói, các phương pháp xử lý giọng nói. Các cách thức tạo ra giọng nói trong mô hình máy học. Các tác vụ như Speech Recognition, Text To Speech, Speech To Text, Speech Translation....

1.2 Giới thiệu hướng làm nhóm lựa chọn

Nhóm sẽ trình bày lại một bài báo khoa học về tác vụ Speech Recognition dựa trên bài báo Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition.

1.3 Giới thiệu khái quát bài toán cụ thể nhóm lựa chọn

Bài toán nhận dạng giọng nói (speech recognition) là một trong những bài toán quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo. Bài toán này nhằm chuyển đổi dạng sóng âm thanh của giọng nói thành văn bản hoặc lệnh máy tính.

Không giống như văn bản viết, giọng nói không được biểu diễn dưới dạng chuỗi các ký tự mà thường được biểu diễn dưới dạng một dãy các tín hiệu sóng âm thanh. Do đó, để nhận dạng được giọng nói, cần phải có các bước tiền xử lý như chuyển đổi âm thanh thành dạng sóng, loại bỏ tiếng ồn, cắt thành các phần nhỏ hơn và trích xuất các đặc trưng âm thanh.

Sau đó, các mô hình máy học, như mạng nơ-ron sâu (deep neural networks) hoặc các mô hình học sâu khác, được sử dụng để học từ dữ liệu huấn luyện và dự đoán các chuỗi từ hoặc câu từ dữ liệu đầu vào. Cuối cùng, kết quả được sinh ra sau khi dự đoán sẽ được chuyển đổi thành văn bản hoặc lệnh máy tính.

Chương 2: FAST CONFORMER

2.1 Giới thiệu bài toán nhóm lựa chọn

Conformer là một mô hình Transducer được đề xuất bởi Gulati và các cộng sự cho tác vụ nhận dạng âm thanh. Conformer đã đạt được kết quả SOTA trên nhiều bộ dữ liệu về âm thanh nhờ vào kiến trúc encoder kết hợp giữa các lớp convolution sâu cho các features cục bộ và lớp self-attention cho các ngữ cảnh global. Tuy nhiên conformer sử dụng nhiều nhiều bộ nhớ và tài nguyên hơn để tính toán bởi vì các lớp self-attention tiêu tốn thời gian và độ phức tạp bộ nhớ theo phương trình bậc 2 độ dài của chuỗi đầu vào. Điều này đặt ra một vấn đề về độ dài giới hạn của chuỗi âm thanh đầu vào mà Conformer có thể xử lý được. Chính vì vậy bài báo này sẽ đề xuất phương pháp để cài thiện vấn đề trên.

2.2 Trình bày bài báo nhóm lựa chọn

Những mô hình với kiến trúc Conformer đã trở thành những mô hình end-to-end thống trị trong tác vụ xử lí âm thanh. Và với mục tiêu có thể cải thiện về thời gian huấn luyện và sử dụng của kiến trúc này, mô hình Fast Conformer đã ra đời với tốc độ nhanh hơn gấp 2.8 lần so với phương pháp Conformer truyền thống. Bài báo này đã thiết kế lại Conformer với một bộ schema dạng tiểu thuyết, giúp hỗ trợ mô hình scale hàng tỷ thông số mà không thay đổi kiến trúc gốc và đạt được sota trên accuracy của benchmark Automatic Speech Recognition.

2.3 Tóm tắt nội dung chính của bài báo

Bài báo thiết kế lại Conformer để giải quyết các vấn đề đã nêu trên. Trong đó có 2 ý chính bào gồm: thiết kế lại phần schema và các khối lấy mẫu để tăng số lượng lấy mẫu lên 8x; và thay thế các lớp self-attention bằng tổ hợp của các attention cục bộ và các token ngữ cảnh toàn cục trước khi train, tương tự như ý tưởng của LongFormer.

Kiến trúc encoder trên có ít hơn 2.9 lần các phép tích với phần attention toàn cục và có thể được làm để scale tuyến tính với chiều dài của chuỗi trước khi train. Thực nghiệm cho thấy mô hình nhanh hơn gấp 2.8 lần so với Conformer và scale đến một tỷ tham số mà không làm thay đổi kiến trúc lõi. Và Fast Conformer vẫn duy trì được độ cạnh tranh trên tỉ lệ lỗi của từ trên các benchmark của tác vụ speech recognition.

Về chi tiết kiến trúc của mô hình, để tăng tốc Conformer, các tác giả đã áp dụng 8x downsampling tại điểm bắt đầu của encoder, thay thế các lớp convolution sub-sampling nguyên bản thành depthwise separable convolutions, giảm số lượng của convolutional filters trong downsampling block còn 256 và giảm kích thước convolutional kernel xuống 9.

Các thay đổi này giúp tốc độ encoder tăng 2.8 lần mà vẫn giữ độ chính xác của mô hình. Để phiên âm các âm thanh dài, bài báo sử dụng một cách tiếp cận được lấy cảm hứng từ Longformer với việc sử dụng một attention token toàn cục đơn và việc chuyển đổi thành limited context attention giúp tăng thời gian mô hình có thể xử lý cùng một lúc trên một GPU A100 đơn lên 45 lần. Để tính toán hiệu quả cho attention này thì các tác giả sử dụng phương pháp tiếp cận các overlapping chunks được giới thiệu trong Longformer.

Bài báo tiến hành thực nghiệm trên bốn tác vụ, đó là Automatic Speech Recognigtion, Speech Translation, Spoken Language Understanding và Long-form audio transcription. Với tác vụ đầu tiên, mô hình được đánh giá trên các bộ dữ liệu LibriSpeech, Large-25k hours NeMo ASR Set. Kết quả trên bộ dữ liệu LibriSpeech cho thấy Fast Conformer có độ chính xác cao hơn Conformer và hiệu quả tính toán của nó nhanh khoảng hơn gấp 3 lần Conformer và cũng nhanh hơn Eff. Conformer và SqueezeFormer. Còn trên bộ dữ liệu Nemo ASR Set Fast Conformer có hiệu suất vượt trội hơn so với Conformer ở hầu hết các tiêu chuẩn nhưng Conformer đạt hiệu suất cao hơn Fast Conformer khi đi cùng với RNNT decoder trên test set LS test-other, MCV 8 test, nhưng hiệu quả chưa thực đáng kể nên Fast Conformer nhìn chung vẫn có hiệu suất cao. Tác vụ tiếp theo bài báo phân tích hiệu quả của Fast Conformer trên tác vụ dịch từ tiếng Anh sang tiếng Đức. Mặc dù kết quả chưa thực sự đáp ứng tốt đến từ mô hình, nhưng Fast Conformer-RNTT đem lại BLEU score 27.9 (cao hơn Conformer) và mô hình cũng học nhanh hơn Conformer gấp 1.84 lần và đối với Fast Conformer-Transformer cũng đưa ra hiệu quả tốt hơn so với Conformer thường. Về Spoken Langugae Understanding, các tác giả So sánh Fast Conformer với các mô hình SpeechBrain-SLU và ESPnet-SLU (cả 2 mô hình này đều sử dụng encoder pretrained HuBERT và được học tự giám sát trên toàn bộ dữ liệu LibriLight-60k). Fast Conformer biểu hiện vượt trội hơn hẳn so với các mô hình được chọn cũng như có thời gian học nhanh hơn 1.1 lần so với Conformer thường. Đối với tác vụ cuối, mô hình được đánh giá trên 2 bộ TED-LUIM v3 và Earnings21 và Fast Conformer với attention mechanism mới vượt trội hơn so với Conformer và Fast Conformer với global attention trên cả hai bộ long-form ASR benchmark này.

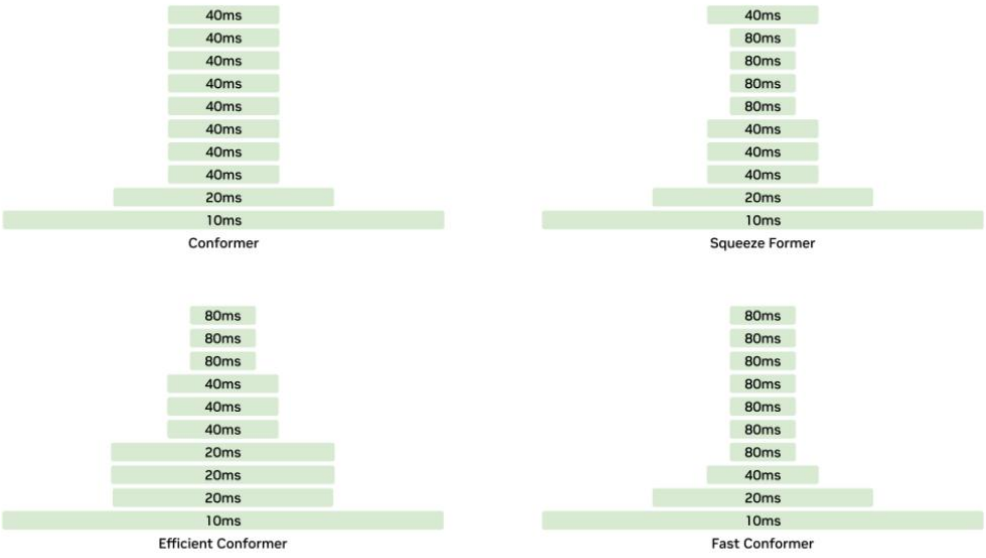
Tóm lại, bài báo đã giới thiệu Fast Conformer, một loại Conformer sử dụng ít hơn 2.9 lần tính toán mà vẫn giữ được WER gần bằng với Conformer gốc. Đánh giá trên các tác vụ speech translation và spoken language understanding cho thấy độ chính xác cao của mô hình trong khi đạt được độ tăng tốc đáng kể trong việc tính toán của encoder. Kết quả trên

long-form audio transcription được cải thiện với việc thêm vào một attention token toàn cục đơn. Và cuối cùng, các tác giả cũng cho thấy kiến trúc của Fast Conformer có thể dễ dàng tăng đến 1B tham số giúp tăng độ chính xác trong khi chịu đựng được nhiễu khi huấn luyện trên các bộ dữ liệu lớn.

Chương 3: FAST CONFORMER WITH LINEARLY SCALABLE ATTENTION FOR EFFICIENT SPEECH RECOGNITION

3.1 Giới thiệu chung

Conformer là mô hình Transducer (RNNT) cho nhận diện giọng nói tự động được đề xuất bởi Gulati và các cộng sự. Các mô hình Conformer đạt được kết quả state-of-the-art trên nhiều bộ dữ liệu về giọng nói nhờ vào kiến trúc encoder kết hợp giữa các lớp convolution sâu cho các features cục bộ và lớp self-attention cho các ngữ cảnh global. Conformers rất nhanh chóng đã được ứng dụng trong công nghiệp, đặc biệt là về streaming ASR trên thiết bị hoặc trên cloud. Tuy nhiên conformer sử dụng nhiều bộ nhớ và tài nguyên hơn các mô hình ASR chỉ sử dụng các lớp convolution như Quartznet, bởi vì các lớp self-attention tiêu tốn thời gian và độ phức tạp bộ nhớ theo phương trình bậc 2 độ dài của chuỗi đầu vào. Điều này đặt ra một vấn đề về độ dài giới hạn của chuỗi âm thanh đầu vào mà Conformer có thể xử lý được. Các mô hình Scaling Conformer đòi hỏi phải điều chỉnh các kernel kernel size trong Conformer block để ổn định quá trình huấn luyện mô hình lớn. Nhóm tác giả đã thiết kế lại Conformer để giải quyết các thách thức kể trên như sau: Thiết kế lại downsampling schema và subsampling block để tăng downsampling lên 8x (Xem Hình 3.1); Sau đó thay đổi các lớp self-attention ban đầu với kết hợp của local attention và global context token post-training, tương tự như LongFormer.



Hình 3.1 Lược đồ downsampling cho Conformer, SqueezeFormer, Efficiency Conformer và Fast Conformer

Encoder được đề xuất có ít hơn 2.9 lần multiply-add operations với global attention, và có thể scale tuyến tính với sequence length post-training. Nó nhanh hơn 2.8 lần so với Conformer tương ứng. Nó có thể scale tới 1 tỉ tham số mà không thay đổi kiến trúc lõi. Nhóm tác giả gọi mô hình này là Fast Conformer, Fast Conformer cũng có word error rates (WER) cao trên tiêu chuẩn ASR.

Nhóm tác giả đã làm thí nghiệm với cơ chế Longformer-based attention và nhận thấy bằng cách sử dụng limited context attention với một global attention token duy nhất nhóm tác giả có thể đạt được kết quả tốt trên long-form ASR, trong khi nhanh hơn 3 lần. Sau đó kiểm tra Fast Confermer trên hai tác vụ bổ sung là Speech Translation (ST) và Speech Language Understanding (SLU). Kết quả trên ST, cho thấy với với transformer và transducer decoders nhóm tác giả đạt được điểm cao cho En-De translation với tốc độ tăng đáng kể so với Conformer. Trên tác vụ SLU nhóm tác giả đạt được SOTA trên tác vụ Speech Intent Classification and Slot Filling, và tốc độ tăng không đáng kể so với Conformer.

3.2 Kiến trúc mô hình Fast Conformer

3.2.1 Downsampling schema

Conformer Encoder bao gồm chồng xen kẻ các multi-head attention, depth-wise separable convolutional và các lớp fully connected với residual connections. Bộ Encoder sẽ bắt đầu với sub-sampling module để tăng tốc độ khung hình từ 10ms lên 40ms. Giảm 4 lần độ dài chuỗi để giảm chi phí tính toán cũng như bộ nhớ tiêu hao của các lớp attention trong tất cả các block. Subsampling module tiêu hao tương đối, chiếm hơn 20% thời gian tính toán mỗi lần.

Cách để tăng tốc Conformer là tăng tỉ lệ downsampling từ 4x thành 8x. Ví dụ là EfficientConformer, giảm độ dài chuỗi xuống 8 lần bằng cách sử dụng progressive downsampling: down-sampling 2 lần ở lớp đầu tiên, sao đó 2 lần nữa ở lớp encoder ở giữa và cuối cùng 2 lần nữa ở lớp encoder cuối. Điểm hạn chế của progressive subsampling là sự mất cân bằng trong tính toán của giữa các lớp attention. Các lớp attention ban đầu có thể hoạt động trên các chuỗi dài hơn, do đó có chi phí tính toán cao hơn đến 16 lần so với các lớp attention hoạt động trên các chuỗi ngắn hơn (xem Hình 3.1). Squeezeformer kết hợp progressive downsampling với cấu trúc Temporal U-Net. Squeezeformer bổ sung thêm

downsampling ở giữa encoder và upsampling ở cuối encoder để khôi phục độ phân giải thời gian 4x (xem Hình 3.1). Một phương pháp tương tự đã được áp dụng ở Uconv-Conformer.

Một trong những lý do tại sao các nghiên cứu trước đó giới hạn encoder downsampling cuối cùng ở 4x liên quan tới việc sử dụng Conformer encoder với CTC loss. CTC yêu cầu đầu vào của hàm mất mát phải dài hơn độ dài chuỗi mục tiêu. Đầu ra của encoder có thể quá ngắn sau khi downsampling xuống 8 lần nếu mô hình sử dụng character tokenization. Ví dụ, hầu hết các mẫu của Librispeech sẽ không phù hợp với CTC nếu subsampling xuống 8 lần. Để có thể áp dụng downsampling xuống 8 lần cho Conformer-CTC nhóm tác giả đổi character tokenization thành Sentencepiece Byte Pair Encoding (BPE) với kích thước từ vựng dao động từ 128 đến 1024 token.

Để tăng tốc Conformer, nhóm tác giả đã thay đổi thiết kế ban đầu:

1. Downsampling xuống 8 lần khi bắt đầu encoder, nhờ đó các lớp attention tiếp theo giảm chi phí tính toán xuống 4 lần.
2. Thay thế các lớp convolution ban đầu với depthwise separable convolutions.
3. Giảm số lượng các convolutional filters trong downsampling block xuống 256
4. Giảm kích thước convolutional kernel xuống 9.

So sánh chi tiết được ghi lại ở Bảng 3.1:

Bảng 3.1 Lược đồ downsampling và loại lớp subsampling cho Conformer, SqueezeFormer, Efficiency Conformer và Fast Conformer

Model	**Subsampling schema**	**Type**	**K**
Conformer	2/4	2D Conv	31
Squeezeformer	progressive 2/4/8/4	Depth-wise sep	31
Eff. Conformer	progressive 2/4/8	Depth-wise sep	15
Fast Conformer	2/4/8	Depth-wise sep	9

Để xác định độ ảnh hưởng của từng thay đổi đối với độ chính xác của mô hình, nhóm tác giả đã áp dụng Conformer-RNNT Large (với 115 triệu tham số) như là baseline và áp dụng từng thay đổi. Đầu tiên thêm 1 lớp 2x convolutional subsampling. Sau đó là sử dụng depthwise-separable convolution ở lớp subsampling thứ 2 và thứ 3. Sau đó thay đổi số

kernel của lớp subsampling từ 512 thành 256. Cuối cùng giảm kích thước của convolutional kernel trong conformer blocks từ 31 thành 9. Tốc độ của encoder sẽ được đo ở batch size 128, GPU A100/80G và sử dụng mẫu giọng nói 20 giây. Kết quả được ghi lại ở Bảng 3.2. Tốc độ encoder tăng gấp 2.8 lần trong khi vẫn duy trì độ chính xác của mô hình.

Bảng 3.2 Độ chính xác và tốc độ của từng thành phần của lược đồ Downsampling Fast Conformer

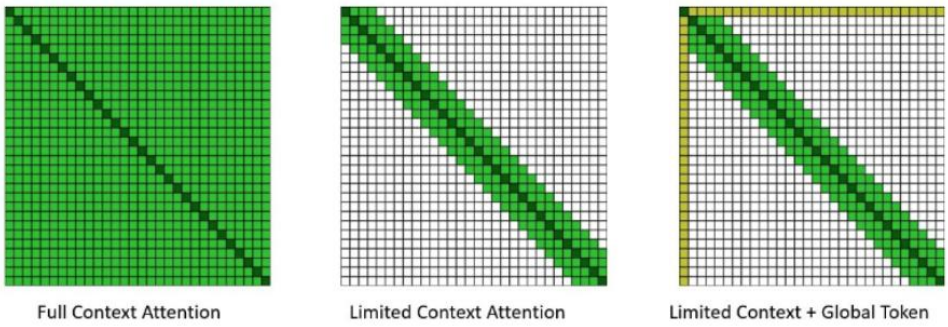
Encoder	**WER, %, test-other**	**Inference, samples/s**	**Params, M**	**GMACS**
Baseline Conformer	5.19	169	115	143.2
+8X Stride	5.11	303	115	92.5
+Depthwise conv	5.12	344	111	53.2
+256 channels	5.09	397	109	48.8
+Kernel 9	**4.99**	**467**	**109**	**48.7**

3.2.2 Long – form audio transcription

Dù các lớp multi-head attention tiêu chuẩn có thể xử lý tốt các câu nói ngắn, nhưng khả năng để mở rộng lên xử lý các chuỗi dài đã bị hạn chế bởi vì sự tăng theo bậc 2 của self-attention operation đối với các chuỗi dài. Ví dụ, Conformer có thể xử lý cùng lúc audio có độ dài tối đa 15 phút trên một A100 GPU. Để xử lý vấn đề này, một số phương pháp thay thế đã được khám phá. Một cách tiếp cận phổ biến là phiên âm đệm, chuỗi âm thanh được chia thành các đoạn ngắn hơn, sau đó được phiên âm riêng rồi hợp nhất lại để tạo thành một bản phiên âm hoàn chỉnh. Efficient-Conformer sử dụng grouped attention để giảm chi phí của các lớp attention trên các câu dài bằng cách nhóm các phần tử có thời gian gần nhau trước khi áp dụng scaled dot-product attention.

Nhóm tác giả quyết định sử dụng cách tiếp cận giống như Longformer, tăng cường local attention với global tokens. Nhóm tác giả sử dụng single global attention token, can thiệp vào tất cả các token khác và có tất cả các token khác can thiệp vào chính nó. Global attention token sẽ sử dụng một tập hợp các phép chiếu tuyến tính truy vấn, khóa và giá trị riêng biệt và các token khác sẽ can thiệp vào fixed-size window xung quanh token (Hình 3.2). Bằng cách chuyển sang sử dụng limited context attention, nhóm tác giả đã kéo dài thời gian mà mô hình có thể xử lý trên A100 GPU đơn lên gấp 45 lần, từ 15 phút đối với

Conformer ban đầu lên tới 675 phút với Fast Conformer và limited context (xem Bảng 3.3). Để tính toán một cách hiệu quả attention này, nhóm tác giả đã sử dụng overlapping chunks được giới thiệu trong Longformer.



Hình 3.2 Fast Conformer với local attention và global context token

Bảng 3.3 Thời lượng âm thanh tối đa có thể xử lý bởi A100 GPU với batch size 1

Model	**Max duration, min**
Conformer	15
Fast Conformer	25
Conformer + Limited Context	135
Fast Conf + Limited Context	675

3.3 Các thực nghiệm của nhóm tác giả

3.3.1 Automatic Speech Recognition

Mô hình Fast Conformer được đánh giá theo tiêu chuẩn English ASR benchmarks: LibriSpeech (LS), phần tiếng Anh của Multilingual LibriSpeech (MLS), Mozilla Common Voice (MCV), và Wall Street Journal (WSJ). Nhóm tác giả sử dụng Fast Conformer-RNNT và Fast Conformer-CTC cùng với baseline của mô hình Conformer với Large configuration.

3.3.1.1 LibriSpeech

Đầu tiên, nhóm tác giả huấn luyện mô hình trên bộ dữ liệu LibriSpeech. Sau đó sử dụng SentencePiece unigram tokenizer với 128 token cho CTC và 1024 token cho RNNT. Việc huấn luyện mô hình được thực hiện với optimizer AdamW và Noam learning rate scheduler và learning rate cao nhất lần lượt là 0,0025 và 0,001. Nhóm tác giả đặt lịch khởi động tuyến tính cho 15 nghìn bước trong tất cả các thử nghiệm. Các mô hình Fast Conformer được huấn luyện bằng cách sử dụng cosin scheduler với learning rate cao nhất lần lượt là 0,005

và 0,001. Cả hai mô hình Conformer và Fast Conformer-RNNT được huấn luyện với 80.000 steps và 380.000 với mô hình CTC. Các mô hình được huấn luyện trên 32 GPU và batch size là 2048. 5 checkpoint cuối cùng được tính trung bình. Nhóm tác giả đã sử dụng trình phân tích Deepspeed 4 để ước tính Multiply Accumulate operations (MACS) trên một audio dài 30 giây. Kết quả được ghi lại trong Bảng 3.4. Fast Conformer có độ chính xác cao hơn một chút so với Conformer thông thường. Bộ encoder được đề xuất có hiệu suất tính toán cao hơn gấp 3 lần so với encoder Conformer ban đầu và nhanh hơn đáng kể so với EfficiencyConformer và SqueezeFormer.

Bảng 3.4 Fast Conformer – Large trên tác vụ ASR với decoder RNTT và CTC trên bộ dữ liệu LibriSpeech

Encoder	WER, % test-other	Params, M	Compute, GMACS
RNNT decoder			
Conformer	5.19	115	143.2
Fast Conformer	**4.99**	**109**	**48.7**
CTC decoder			
Conformer	5.74	121	149.2
Eff. Conformer	5.79	125	101.3
SqueezeFormer	6.05	125	91.0
Fast Conformer	**5.64**	**115**	**51.5**

3.3.1.2 Large – 25k hours NeMo ASR set

Để kiểm tra khả năng của Fast Conformer với tập dữ liệu lớn hơn, nhóm tác giả đã huấn luyện Fast Conformer và Conformer trên bộ dữ liệu với 25000 giờ nói được soạn từ LibriSpeech, Mozilla Common Voice, National Singapore Corpus và các tập dữ liệu giọng nói tiếng Anh đã được công bố khác. mô hình RNNT đều được đào tạo cho 300.000 steps và mô hình CTC cho 1.000.000 steps. AdamW được sử dụng với cosin scheduler với 15.000 linear warm up, learning rates lần lượt là 0,0025 và 0,001 đối với RNNT và CTC. Các mô hình đã được thử nghiệm trên các bộ thử nghiệm LibriSpeech test-other, MLS, MCV và WSJ-93. Kết quả được trình bày trong Bảng 3.5. Fast Conformer vượt trội hơn Conformer truyền thống trên hầu hết tập test của các bộ dữ liệu.

Bảng 3.5 Fast Conformer – Large trên tác vụ ASR với decoder RNNT và CTC trên bộ dữ liệu NeMo

Encoder	LS test-other	MCV 8 test	MLS En	WSJ-92 test
RNNT decoder				
Conformer	**3.74**	**7.87**	**5.77**	**1.47**
Fast Conformer	3.79	8.18	5.76	1.42
CTC decoder				
Conformer	4.50	9.40	6.60	1.70
Fast Conformer	**4.19**	**9.00**	**6.42**	**1.59**

3.3.2 Speech Translation

Tiếp theo, nhóm tác giả phân tích sự hiệu quả của Fast Conformer trong tác vụ Speech Translation (ST) từ tiếng Anh sang tiếng Đức. Nhóm tác giả đã huấn luyện 2 kiến trúc với encoder giống Conformer và các autoregressive decoder khác nhau: RNNT hoặc 6-layer Transformer được huấn luyện với hàm mất mát cross entropy.

Trong tất cả các thí nghiệm, các tác giả đã khởi tạo tham số của encoder với các trọng số tương ứng từ mô hình ASR RNNT đã được huấn luyện trên 25.000 giờ nói. Tập từ vựng của nhóm tác giả bao gồm 16384 YouTokenToMe5 byte-pair-encodings được huấn luyện trên văn bản tiếng Đức. Các mô hình của tác giả được huấn luyện trên tất cả các bộ dữ liệu có sẵn ở cuộc thi IWSLT22 với 4000 giờ nói. Một số bộ dữ liệu không có bản dịch tiếng Đức nên nhóm tác giả đã tự tạo bản dịch tiếng Đức bằng mô hình text-to-text machine translation được huấn luyện trên WMT21 và tinh chỉnh trên Must-C v2.

Kết quả của tất cả các mô hình được trình bày trong Bảng 3.6. Nhóm tác giả phát hiện thấy RNNT loss không phù hợp với tác vụ speech translation vì các căn chỉnh ngầm định. Điều đáng ngạc nhiên là mô hình Fast Conformer-RNNT translation đạt BLEUscore 27,89. Ngoài ra, khả năng suy luận của mô hình này nhanh hơn tới 1,84× so với Conformer.

Bảng 3.6 Bảng so sánh Conformer và Fast Conformer trên tác vụ Speech Translation với bộ dữ liệu MUST-C V2 tst-COMMON

Encoder	BLEU	Time, sec	Speed-up
Transformer decoder			
Conformer	31.0	267	1X
Fast Conformer	31.4	161	1.66X
RNNT decoder			
Conformer	23.2	83	1X
Fast Conformer	27.9	45	1.84X

3.3.3 Spoken Language Understanding

Tiếp theo, nhóm tác giả đã áp dụng pre-trained Fast Conformer encoder cho tác vụ spoken language understanding (SLU). Nhóm tác đã cũng đã nghiên cứu tác vụ Speech Intent Classification and Slot Filling (SICSF) để xác định dự định của người dùng và trích xuất các phần bổ sung từ vựng tương ứng cho các vị trí thực thể được phát hiện. Ý định có thể là sự kết hợp của loại tình huống và loại hành động. Vị trí và phần bổ sung được thể hiện bằng cặp key-value. Các ý định và vị trí của đầu vào sắp xếp dưới dạng dictionary của Python và được biểu thị dưới dạng chuỗi. Tác vụ SICSF là dự đoán dictionary Python có cấu trúc này dưới dạng một chuỗi, dựa trên âm thanh đầu vào. Các thí nghiệm được tiến hành bằng cách sử dụng bộ dữ liệu SLURP và sử dụng accuracy và SLURP-F1 để làm độ đo đánh giá.

Nhóm tác giả đã sử dụng baseline Conformer cho bộ encoder, và các thông số được khởi tạo từ mô hình ASR với Transformer decoder. Nhóm tác giả cũng đã so sánh Fast Conformer với 2 mô hình state-of-the art là ESPNet-SLU và SpeechBrain. Cả ESPNet-SLU và SpeechBrain đều sử dụng pretrained encoder HuBERT thông qua self-supervised objective trên toàn bộ dữ liệu LibriLight-60k. ESPNet-SLU được tinh chỉnh thêm bộ encoder trên LibriSpeech trước khi được huấn luyện về trên bộ dữ liệu SLURP.

Kết quả về tác vụ SLURP được ghi lại ở Bảng 3.7. Mô hình với pre-trained Fast Conformer encoder tỏ ra vượt trội so với ESPNet-SLU và SpeechBrain. Fast Conformer đạt độ chính xác tương tự với Conformer nhưng decoder nhanh hơn 10%. Nhóm tác giả nhận thấy tốc độ không tăng nhiều như tác vụ ASR vì tỷ lệ độ dài tín hiệu âm thanh (sau khi

downsampling xuống 8 lần) so với token mục tiêu là khoảng **1:2.22** đối với tác vụ SICSF. Chi phí thực hiện của encoder được giảm thiểu bởi slow autoregressive Transformer decoder, vì vậy nhóm tác giả đã sử dụng batch 32 để cân bằng chi phí của encoder và decoder nhằm để tăng tốc.

Bảng 3.7 Kết quả của tác vụ Speech Intent Classification and Slot Filling trên bộ dữ liệu SLURP

Model	**Intent Acc.**	**SLURP F1**	**Inference, sec**	**Rel. Speed-up**
SpeechBrain-SLU	87.70	76.19	-	-
ESPnet-SLU	86.52	76.91	-	-
Conformer/Fast Conformer+Transformer Decoder				
Conformer	90.14	82.27	210	1X
Fast Conformer	90.68	82.04	191	1.1X

3.3.4 Long – form audio transcription

Fast Conformer với limited context và global token cho long-form audio được huấn luyện như sau: Mô hình có các shared query, key và các lớp value projection được sử dụng cho global và local attention. Bộ encoder sẽ được khởi tạo với pre-trained checkpoint với dữ liệu 25.000 giờ. Sau đó nhóm tác giả đã tinh chỉnh trên cùng bộ dữ liệu với limited context cho 10000 steps, learning rate tối đa là 1e-6, và cosin rate ủ về 0. Kích thước của limited context được đặt là 128 steps trên mỗi mặt của token tương ứng với khoảng 10 giây.

Hiệu suất của Fast Conformer được đánh giá trên 2 bộ dữ liệu long-from audio là TED-LIUM v3 và Earnings-21. Nhóm tác giả đã so sánh mô hình limited context với context Fast Conformer cũng như Conformer cơ sở trên cùng bộ dữ liệu. Nhóm tác giả sử dụng chuẩn hóa Whisper trên cả bản ghi lẫn dự đoán để đánh giá mô hình. Về Conformer và Fast Conformer với full context attention nhóm tác giả đã sử dụng bộ đệm 20 giây. Về Fast Conformer với limited context nhóm tác giả xử lý với chỉ một lần chuyển tiếp. Fast Conformer với cơ chế attention mới có kết quả áp đảo so với Conformer và Fast Conformer với global attention trên tiêu chuẩn long-form ASR (xem Bảng 3.8).

Bảng 3.8 Bảng so sánh Conformer và Fast Conformer trên long audio

Model	TED-LIUM v3	Earnings21
Conformer	9.18	18.26
Fast Conformer (buffered)	9.15	17.65
+ Limited Context	8.25	16.08
+ Global Token	7.5	11.85

3.4 Mở rộng mô hình Fast Conformer

Để cho thấy khả năng scaling của các mô hình Fast Conformer, nhóm tác giả đã thiết kế 3 mô hình với các kích thước Large (L), Extra Large (XL) and Extra Extra Large (XXL) tương tự như Fast Conformer (xem Bảng 3.9).

Bảng 3.9 Bảng thể hiện điều chỉnh tham số của mô hình FC-L, -XL và -XXL

Model	Hidden Dimension	Encoder Layers	RNNT Layers	Model Parameters
L	512	17	1	120 M
XL	1024	24	2	600 M
XXL	1024	42	2	1.1 B

Nhóm tác giả nhận thấy khi mở rộng mô hình từ XL sang XXL, pretrained encoder với Self Supervised learning sẽ giúp ổn định mô hình và cho phép learning rate cao. Nhóm tác giả đã áp dụng pretraining và tinh chỉnh của mô hình SSL dựa trên Wav2Vec 2.0. Không giống như các mô hình Conformer khác, nhóm tác giả đã không thay đổi conformer blocks và relative attention khi scaling mô hình. Từ L lên đến XXL kiến trúc lõi của mô hình được giữ nguyên.

Các mô hình XL và XXL mang lại kết quả vượt trội ở các bước đầu khi huấn luyện. Như trong Bảng 3.10 khi được kiểm tra trên HF-Leaderboard phân tích hiệu suất cho thấy tính hiệu quả của các mô hình XL và XXL được đào tạo trong 25000 giờ của NeMo ASR. Mô hình XL được huấn luyện trong 70000 bước và XXL là 100000 bước với cùng batch size là 2048. Với XL là optimizer AdamW với Noam learning rate scheduler với learning rate cao nhất là 6e-4 và linear warmup là 15000 bước, còn XXL là 25000 bước. Cả XL và

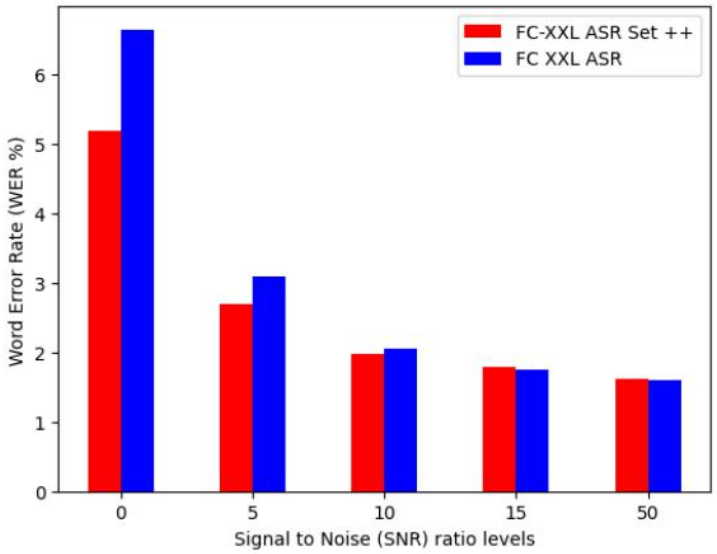
XXL đều được khởi tạo với checkpoint pretrained SSL. Nhóm tác giả nhận thấy việc tinh chỉnh mô hình RNNT FC-XL với CTC với 40000 bước cho hiệu suất tương tự với việc huấn luyện mô hình FC-XL CTC từ đầu với 200000 bước.

Bảng 3.10 Bảng so sánh XL và XXL trên ASR benchmarks

Model	LS Test-clean	LS Test-other	MLS Test	GMACS
Conformer-XL	1.49	2.80	5.32	686
FC-XL	1.50	2.88	4.90	253
FC-XXL	1.38	2.52	4.58	441

3.4.1 Mở rộng bộ dữ liệu

Mô hình FC-XXL RNNT được huấn luyện trên 25000 giờ của bộ dữ liệu ASR, đạt được hiệu suất state-of-the-art trên LS-test other cũng như đạt được hiệu suất tốt nhất trên nhiều bộ dữ liệu khác Tuy nhiên, để tận dụng hiệu quả các mô hình lớn, bắt buộc phải tăng kích thước tập dữ liệu theo tỷ lệ phù hợp với kích thước mô hình. Do đó, nhóm tác giả đã huấn luyện mô hình bằng cách thêm vào 40000 giờ vào bộ dữ liệu có sẵn ((ASR Set++). Việc thêm các bộ dữ liệu bổ sung này đã tạo điều kiện nâng cao độ chính xác và noise robustness trong cả hai mô hình XL và XXL. Bảng 3.11 và Bảng 3.12 cho thấy kết quả khi huấn luyện trên bộ dữ liệu lớn của mô hình 1.1B tham số. Hình 3.3 cho thấy noise robustness của mô hình XXL thông qua các mức signal-to-noise ratio (SNR) trên bộ dữ liệu Librispeech.



Hình 3.3 Noise robustness của mô hình XXL

Bảng 3.11 Hiệu suất của các mô hình FC-XL và FC-XXL với tập huấn luyện tăng cường

Model	Decoder	Train Dataset	LS Test-clean	LS Test-other	TED-LIUM V3	Vox Populi
FC-XL	RNNT	ASR Set	1.50	2.88	4.49	5.74
		ASR Set ++	1.63	3.06	3.86	6.05
	CTC	ASR Set	1.73	3.47	4.71	6.09
		ASR Set ++	1.87	3.76	3.78	7.00
FC-XXL	RNNT	ASR Set	**1.38**	2.52	4.74	5.56
		ASR Set ++	1.46	**2.47**	3.92	**5.39**
	CTC	ASR Set	1.69	3.4	4.64	6.45
		ASR Set ++	1.83	3.54	**3.54**	6.53

Bảng 3.12 Hiệu suất của các mô hình FC-XL và FC-XXL với tập huấn luyện tăng cường

Model	Decoder	Train Dataset	MCV 9 Test	AMI Test	Earnings 22	SPGI Speech	Giga Speech
FC-XL	RNNT	ASR Set	7.26	18.28	16.37	4.40	11.58
		ASR Set ++	8.07	17.55	14.78	3.47	10.07
	CTC	ASR Set	7.51	18.41	17.89	5.04	11.84
		ASR Set ++	10.57	16.3	14.14	4.11	10.35
FC-XXL	RNNT	ASR Set	6.07	18.81	16.66	4.98	11.95
		ASR Set ++	**5.79**	17.1	14.11	**3.11**	**9.96**
	CTC	ASR Set	8.31	17.62	16.44	4.91	11.61
		ASR Set ++	9.02	**15.62**	**13.69**	4.20	10.27

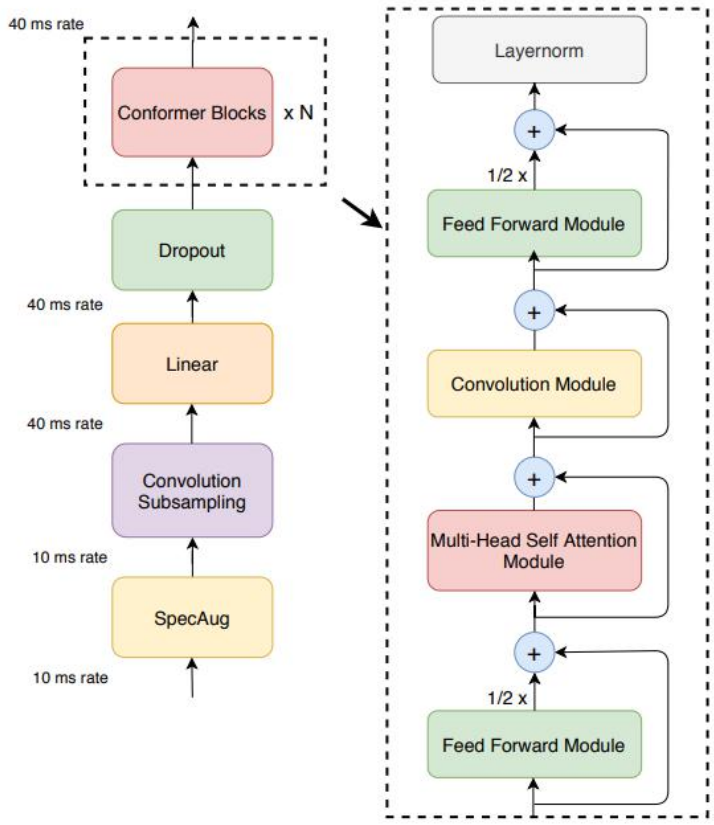
Chương 4: CÁC THỰC NGHIỆM CỦA NHÓM

4.1 Thiết kế thực nghiệm

Bởi vì các tác giả của bài báo này sử dụng tài nguyên rất lớn để huấn luyện Fast Conformer trên các bộ dữ liệu lớn nên nhóm không đủ khả năng để huấn luyện lại mô hình này. Thay vào đó, nhóm sẽ sử dụng checkpoint của mô hình Fast Conformer và một số mô hình khác như Conformer, Jasper, QuartzNet và Citrinet để dự đoán trên các bộ dữ liệu Libri – clean, Libri – other, Minds14, Speech COCO và Google Fleurs nhằm so sánh, đánh giá hiệu suất cũng như thời gian chạy của Fast Conformer so với các mô hình kể trên.

4.2 Các mô hình nhóm sử dụng

4.2.1 Conformer



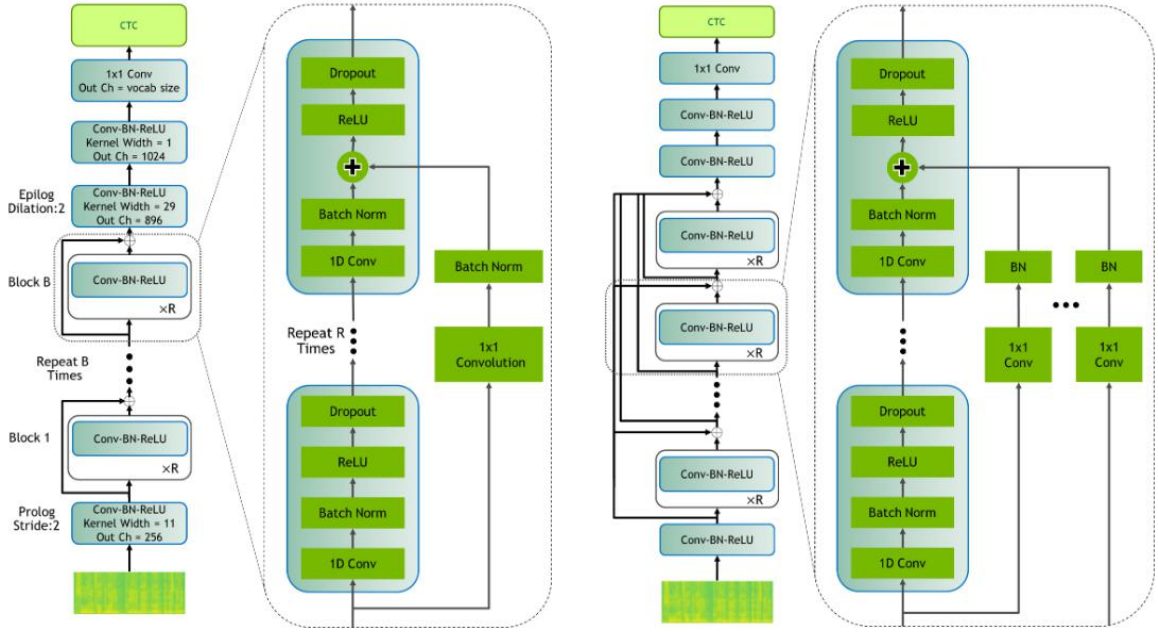
Hình 4.1 Kiến trúc mô hình Conformer

Mô hình Conformer được ra đời vào năm 2020 bởi Anmol Gulati và các cộng sự đã đề xuất một kiến trúc neural network dành cho tác vụ automatic speech recognition, kết hợp các lớp tích chập và các lớp transformer. Mô hình đã tận dụng khả năng nắm bắt được các sự phụ thuộc local của các lớp convolution và khả năng hiểu ngữ cảnh toàn cục của transformers. Mỗi Conformer block tích hợp bao gồm một module feedforward, cơ chế self-attention, một module convolution và cơ chế đặt cổng. Cách tiếp cận này đã tăng cường hiệu quả, độ

chính xác và khả năng mở rộng, làm cho Conformer đặc biệt hiệu quả cho tác vụ nhận diện giọng nói tự động và các ứng dụng dựa trên giọng nói khác.

4.2.2 Jasper

Jasper (Just Another Speech Recognition) là một deep time delay neural network (DTNN) được phát triển bởi NVIDIA để chuyển đổi giọng nói thành văn bản một cách chính xác và hiệu quả, giúp cải thiện các ứng dụng như trợ lý ảo, điều khiển bằng giọng nói, và các hệ thống dịch tự động.

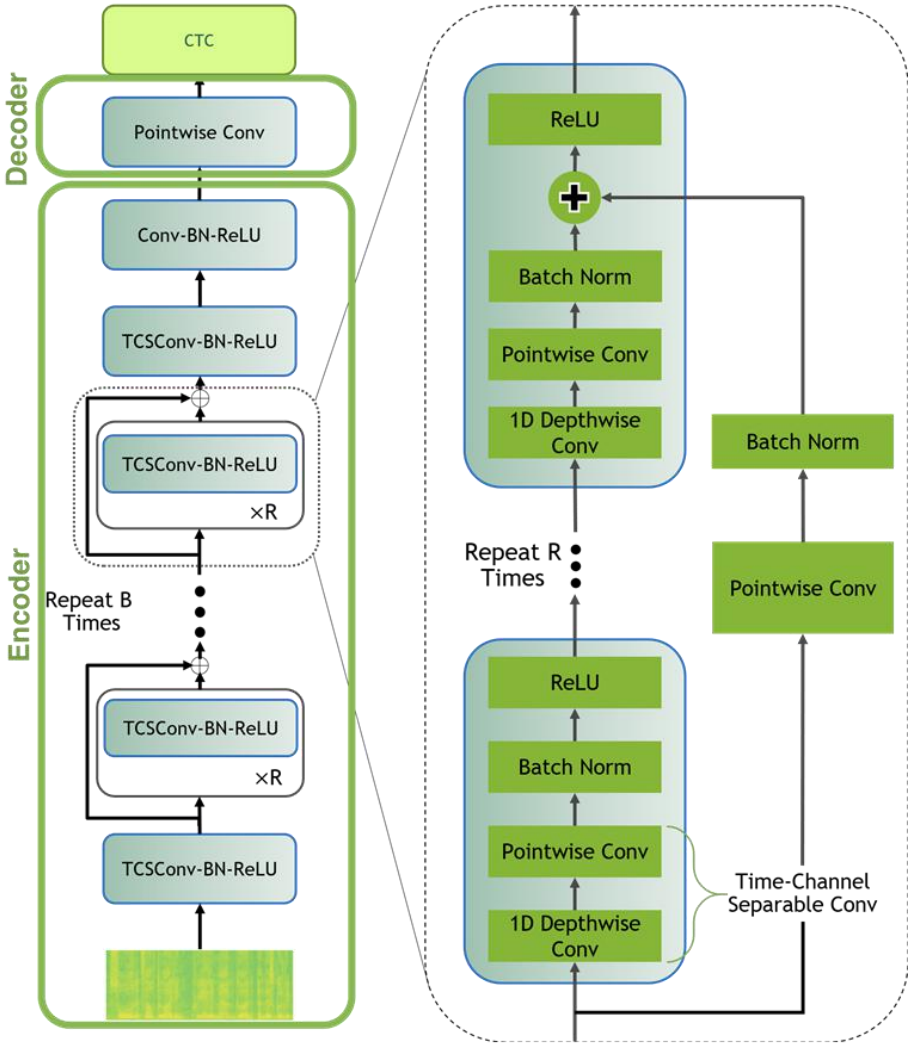


Hình 4.2 Kiến trúc của mô hình Jasper BxR và Jasper Dense Residual

Jasper sử dụng kiến trúc convolutional neural network (CNN) thay cho các mô hình acoustic và pronunciation, giúp mô hình huấn luyện nhanh hơn và dễ dàng song song hóa hơn trên các GPU hiện đại. Để cải thiện khả năng học và giảm độ sâu của mạng, Jasper sử dụng các kết nối residual giữa các block, tương tự như trong ResNet. Điều này giúp truyền thông tin hiệu quả hơn và giảm thiểu vấn đề vanishing gradient. Batch normalization cũng được áp dụng sau mỗi tầng convolution để ổn định quá trình huấn luyện và tăng tốc độ hội tụ. Kiến trúc của Jasper không chỉ tối ưu hóa quá trình huấn luyện mà còn giúp nó dễ dàng mở rộng và áp dụng trong các ứng dụng thực tế. Những điểm mạnh như khả năng huấn luyện nhanh chóng và hiệu quả, cùng với khả năng xử lý dữ liệu lớn, đã khiến Jasper trở thành một trong những mô hình hàng đầu trong lĩnh vực nhận dạng giọng nói.

4.2.3 QuartzNet

Quartznet là một mô hình nhận dạng giọng nói tự động (Automatic Speech Recognition - ASR) dựa trên kiến trúc mạng nơ-ron tích chập (Convolutional Neural Network - CNN). Được phát triển bởi NVIDIA, Quartznet là một trong những mô hình tiên tiến trong lĩnh vực xử lý giọng nói nhờ vào hiệu suất cao và khả năng xử lý trên nhiều ngôn ngữ khác nhau.



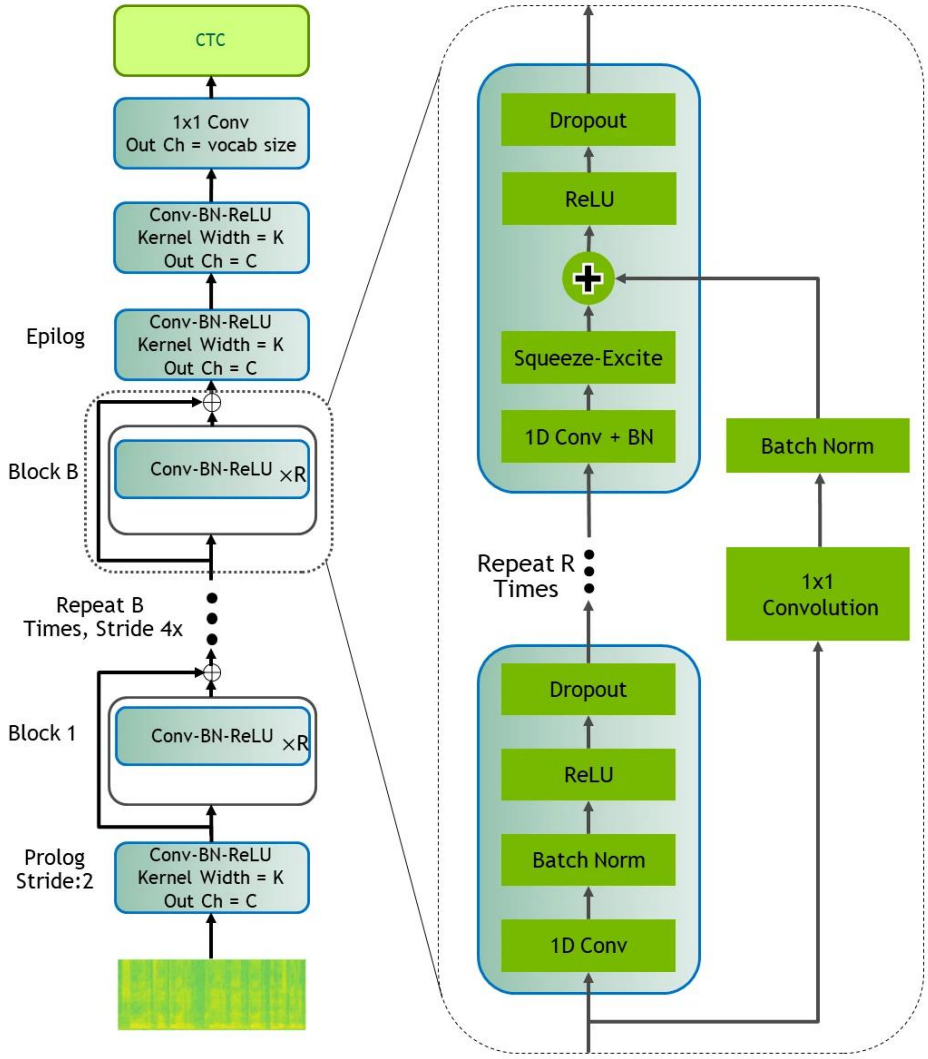
Hình 4.3 Kiến trúc của mô hình QuartzNet

Kiến trúc của Quartznet chủ yếu dựa trên các tầng mạng tích chập (Convolutional Layers) và có thiết kế tương tự như mô hình Jasper của NVIDIA, nhưng với một số cải tiến để tăng hiệu quả. Quartznet được xây dựng từ các khối (block), mỗi khối chứa một số lượng lớp tích chập và các lớp khác, thường bao gồm một số lớp tích chập 1D, theo sau là các lớp chuẩn hóa (Batch Normalization) và các hàm kích hoạt (ReLU activation). Các lớp tích chập 1D này để xử lý dữ liệu theo thời gian, giúp mô hình có thể nắm bắt thông tin theo

thời gian hiệu quả hơn. So với các mô hình khác như Transformer hoặc RNN, Quartznet có ít tham số hơn, giúp mô hình nhẹ hơn và nhanh hơn trong quá trình huấn luyện và suy luận.

4.2.4 Citrinet

Citrinet là một mô hình được phát triển bởi NVIDIA, nổi bật trong lĩnh vực xử lý ngôn ngữ tự nhiên. Citrinet là một phần của bộ công cụ NeMo (NVIDIA NeMo), được thiết kế để thực hiện các tác vụ như nhận dạng giọng nói, tổng hợp giọng nói và xử lý ngôn ngữ tự nhiên.



Hình 4.4 Kiến trúc của mô hình CItrinet

Citrinet là một mô hình nhận dạng giọng nói tự động (ASR) dựa trên kiến trúc CTC (Connectionist Temporal Classification). Citrinet là một phiên bản của QuartzNet mở rộng từ ContextNet, sử dụng mã hóa từ phụ (WordPiece tokenization) và cơ chế Squeeze-and-Excitation để đạt được bản ghi âm chính xác cao, đồng thời sử dụng sơ đồ giải mã dựa trên

CTC không tự hồi quy để suy luận hiệu quả. Kiến trúc này giúp giảm đáng kể khoảng cách giữa các mô hình không tự hồi quy và các mô hình sequence-to-sequence và transducer.

4.3 Các bộ dữ liệu nhóm sử dụng

4.3.1 LibriSpeech

Bộ dữ liệu Librispeech là một bộ bao gồm xấp xỉ 1000 giờ của các cuốn audiobook trong dự án librivox. Hầu hết các cuốn audiobook này đều nằm trong dự án Gutenberg. Tập train của bộ dữ liệu được chia thành 3 tập bao gồm 100 giờ, 360 giờ và 500 giờ. Trong khi đó tập dev và test được chia thành 2 loại là "clean" và "other". Các loại này được xác định dựa trên việc những hệ thống Automatic Speech Recognition sẽ hoạt động tốt hay gặp thử thách khi tính toán hiệu suất trên các tập. Mỗi tập dev và test sẽ bao gồm gần 5 giờ ghi âm. Do tập test của dataset này được chia theo độ khó và thách thức của âm thanh, nhóm sẽ tiến hành thực nghiệm và phân tích trên cả 2 loại "clean" và "other" của tập test.

4.3.2 PolyAI/Minds14

Minds14 là một bộ dữ liệu được phát triển bởi PolyAI, một công ty chuyên về công nghệ xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo. Mục đích chính của bộ dữ liệu này là cung cấp một tập hợp các đoạn hội thoại đa dạng để huấn luyện và đánh giá các mô hình NLP, đặc biệt là các mô hình dùng trong nhận dạng và phân loại ý định của người dùng trong các cuộc đối thoại. Bộ dữ liệu bao gồm 14 loại ý định khác nhau được trích xuất từ một hệ thống thương mại trong lĩnh vực ngân hàng điện tử, với 14 loại ngôn ngữ đa dạng nhưng chủ yếu vẫn là tiếng Anh. Nhóm sẽ sử dụng cả bộ dữ liệu với ngôn ngữ 'en-AU' để tiến hành dự đoán.

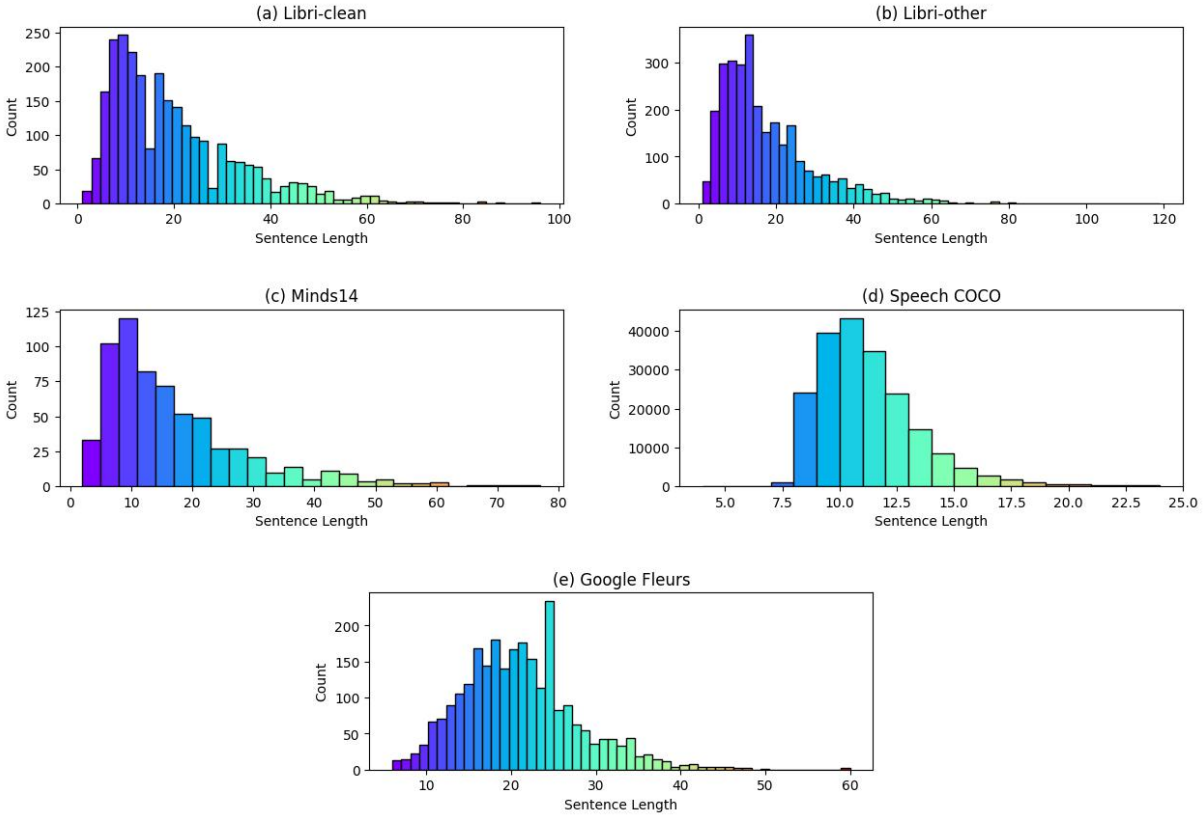
4.3.3 Speech COCO

Speech COCO là một bộ dữ liệu được thiết kế bởi William Havard, Laurent Besacier, và Olivier Rosec vào năm 2017, nhằm hỗ trợ nghiên cứu và phát triển trong lĩnh vực nhận dạng giọng nói tự động - ASR và các ứng dụng liên quan đến xử lý ngôn ngữ tự nhiên - NLP. Nó là một phần mở rộng của bộ dữ liệu COCO (Common Objects in Context), vốn nổi tiếng trong lĩnh vực thị giác máy tính (Computer Vision). Speech COCO được tạo bởi text-to-speech (TTS) synthesis từ bộ dữ liệu MSCOCO 2014, gồm 616,767 âm thanh chú thích ứng với mỗi hình ảnh, với hơn 600 giờ nói. Các tác giả sử dụng 8 giọng đọc khác nhau, điều chỉnh tốc độ và thêm các từ nói không lưu lướt như 'um', 'er' để âm thanh nghe tự nhiên hơn, giúp mô hình học được sự đa dạng và phức tạp của ngôn ngữ tự nhiên. Nhóm sẽ sử dụng tập val của bộ dữ liệu này với 202,654 dòng dữ liệu.

4.3.4 Google/Fleurs

Bộ dữ liệu Google Fleurs được phát triển bởi nhóm nghiên cứu tại Google, ra mắt vào tháng 4 năm 2022 với mục đích hỗ trợ nghiên cứu và phát triển hệ thống nhận dạng giọng nói tự động (ASR) cho nhiều ngôn ngữ khác nhau, đặc biệt là những ngôn ngữ ít được nghiên cứu và không có nhiều tài nguyên kỹ thuật số. Bộ dữ liệu bao gồm 102 ngôn ngữ, được xây dựng dựa trên tiêu chuẩn dịch máy FLoRes-101, với trung bình 12 tiếng cho mỗi ngôn ngữ. Nhóm sẽ sử dụng cả 3 tập train, val, dev của ngôn ngữ 'en_us' để tiến hành dự đoán.

4.3.4.1 Thống kê các bộ dữ liệu



Hình 4.5 Biểu đồ thể hiện phân phối độ dài câu của các bộ dữ liệu

Từ Hình 4.5, chúng ta có thể thấy hai bộ Libri clean và other có phân phối độ dài câu khá tương đồng, trải dài từ 1 từ cho đến cả trăm từ nhưng tập trung chủ yếu là dưới 40 từ. Minds14 cũng khá tương tự nhưng có phân phối thưa và ít lệch phải hơn. Speech COCO tuy số lượng rất nhiều, hơn 200,000 dòng dữ liệu nhưng mỗi câu chủ yếu chỉ trên dưới 10 từ. Cuối cùng là Google Fleurs, bộ dữ liệu này có phân phối độ dài câu gần như đối xứng với các câu có độ dài từ 10 đến 30 từ là phổ biến nhất. Tóm lại, các bộ dữ liệu như Librispeech, Minds14 và đặc biệt là Speech COCO đều chủ yếu gồm các câu ngắn, còn Google Fleurs có số lượng các câu dài nhiều hơn.

4.4 Kết quả và thảo luận

Bảng 4.1 Kết quả WER của các mô hình trên các bộ dữ liệu

Model	Libri - clean	Libri - other	Minds14	Speech COCO	Google Fleurs
Fast Conformer (RNNT)	0.0196	0.0386	0.1899	**0.0830**	**0.1024**
Conformer (RNNT)	**0.0193**	**0.0361**	**0.1862**	0.1001	0.1092
Fast Conformer (large)	0.0244	0.0510	0.1963	0.1082	0.1175
Conformer (large)	0.0251	0.0526	0.2073	0.1171	0.1186
Conformer (small)	0.0421	0.0964	0.2400	0.1671	0.1497
Jasper	0.0402	0.1152	0.2631	0.1485	0.1581
QuartzNet	0.0424	0.1137	0.2736	0.1684	0.1681
Citrinet	0.0355	0.0846	0.2234	0.1236	0.1368

Dựa trên kết quả dự đoán cho tác vụ automated speech recognition của các model trên bộ dữ liệu, ta có thể thấy FastConformer và Conformer sử dụng Recurrent Neural Network Transducer cho ra WER thấp nhất. Trong đó, WER thấp nhất được ghi nhận thuộc về ConFormer trên tập test của Libri-clean với. Điều này có thể được giải thích bởi sự "clean" của bộ Libri-clean, thứ chúng ta có thể quan sát được thông qua hiệu suất của các mô hình khác trên miền dữ liệu này khi hiệu suất mặt bằng chung WER của các mô hình sẽ rơi vào khoảng 0.0400 so với 0.0193 của mô hình. Thậm chí, hiệu suất của mô hình Conformer-small với CTC loss còn thấp hơn khi so sánh với các mô hình khác. Tiếp tục quan sát đến các bộ dữ liệu khó và phức tạp hơn, ta vẫn thấy được độ hiệu quả của mô hình Conformer trên 2 bộ Libri-Other và Minds-14 với WER lần lượt là 0.0361 và 0.1862. Ở dataset Libri-other, sự thống trị của Conformer trong các kiến trúc xử lý tác vụ âm thanh "end-to-end" đã được thể hiện rõ ràng hơn khi cả 4 mô hình Jasper, Quartznet, Citrinet và Conformer-small đều không đạt hiệu suất cao với tỉ lệ WER lớn hơn 0.08.Tuy nhiên, ta có

thể thấy được hiệu suất giảm mạnh ở tất cả mô hình trên dataset Minds14, kết quả này có thể được giải thích dựa trên đặc trưng của bộ dữ liệu khi được trích xuất từ 14 loại ý định khác nhau tại một hệ thống thương mại trong lĩnh vực ngân hàng điện tử, một phần các từ vựng chuyên môn có thể xuất hiện trong các đoạn hội thoại và làm giảm hiệu suất chung của các mô hình. Ở 2 bộ dữ liệu cuối là sự bức phá của Fast Conformer khi hiệu suất của mô hình này vượt trội so với hầu hết các mô hình khác và cao hơn mô hình dẫn đầu trong ba bộ dữ liệu trước từ 1-2% WER. Tuy nhiên, khi nhìn lại các mô hình kiến trúc Conformer, sự khác biệt có thể nhận thấy trên hiệu suất chính là ở RNNT decoder và CTC loss khi phương pháp RNNT có thể học được cách căn chỉnh lại đầu ra để tạo thành một câu hoàn chỉnh trong quá trình huấn luyện và khả năng căn chính đầu ra với đa dạng chiếu dài mà không cần định trước. Trong khi đó, CTC sẽ thực hiện công việc này bằng cách sử dụng một hàm loss đặc biệt để giải quyết vấn đề thông qua phân phối xác suất.

Bảng 4.2 Thời gian phiên âm trung bình (giây) của các mô hình trên các bộ dữ liệu

Model	**Libri - clean**	**Libri - other**	**Minds14**	**Speech COCO**	**Google Fleurs**
Fast Conformer (RNNT)	0.3115	0.2913	0.2701	0.0536	0.3418
Conformer (RNNT)	0.4434	0.3946	0.4103	0.0976	0.5256
Fast Conformer (large)	0.1252	0.1247	0.1242	0.0129	0.1126
Conformer (large)	0.1332	0.1273	0.1247	0.0172	0.1197
Conformer (small)	0.1581	0.1358	0.1029	**0.0113**	0.1056
Jasper	0.1922	0.1802	0.1418	0.0451	0.1226
QuartzNet	**0.0263**	**0.0219**	**0.0516**	0.0118	**0.0245**
Citrinet	0.0494	0.0443	0.0808	0.0208	0.0506

Tiếp đến, nhóm tiếp tục thí nghiệm trên thời gian các mô hình nhận diện văn bản từ âm thanh để đánh giá về tính thiết thực và khả năng cải thiện của mỗi mô hình. Nhìn vào kết quả, ta có thể nhận thấy tốc độ nhận diện của Quartznet là rất nhanh khi bỏ xa các mô hình còn lại với độ chênh lệch lên đến 0.1 giây. Đây là nhờ độ hiệu quả trong kiến trúc của Quartznet khi sử dụng 1D Depthwise Convolution giúp việc áp dụng Convolution hiệu quả hơn so với các 2D Convolution sử dụng trong các model ASR truyền thống. Bên cạnh đó, giúp giảm độ phức tạp của mô hình và tăng tốc độ tính toán. Bên cạnh đó, ta thấy được sự đánh đổi của các mô hình Conformer sử dụng kiến trúc RNNT decoder khi thời gian transcribe cao hơn nhiều lần so với phần còn lại vì độ phức tạp cao cũng như độ lớn của mô hình khi có hơn 100M tham số. Tuy nhiên, trên bộ dữ liệu Speech COCO, Conformer-small đã dẫn đầu về thời gian, vượt qua tốc độ của Quartznet và đồng thời là sự gia tăng đáng kể về tốc độ trên tất cả mô hình. Hiện tượng này được nhóm giải thích dựa trên bộ dữ liệu Speech COCO khi nhìn vào Hình 4.5, quan sát ta thấy độ dài nội dung của một đoạn ghi âm trên bộ dữ liệu này tương đối ngắn khi chỉ giao động chủ yếu trong khoảng từ 6 - 12 từ, khá thấp so với độ dài chung của các bộ dữ liệu còn lại. Điều này cũng cho thấy được tốc độ dự đoán của Conformer-small sẽ hiệu quả hơn khi độ dài câu ngắn đi.

Cuối cùng, để có thể đánh giá nghiên cứu quan trọng nhất của paper này, nhóm sẽ so sánh thời gian và hiệu suất giữa 2 model Conformer và Fast Conformer, và giữa 2 kiến trúc RNNT và CTC. Kết quả của Bảng 4.3 đã cho thấy hiệu quả trong việc cải thiện thời gian sản sinh phần script của mô hình Fast Conformer ở cả hai kiến trúc trên tất cả các bộ dữ liệu và vẫn giữ được hiệu suất của mô hình Conformer khi phần trăm trên lệch trên chỉ số WER ở 2 mô hình là rất thấp qua đó cũng chứng minh được độ hiệu quả trong phương pháp tác giả đã áp dụng và can thiệp lên mô hình.

Bảng 4.3 Bảng so sánh WER và thời gian phiên âm trung bình (giây) của Fast Conformer và Conformer với 2 loại decoder RNTT và CTC trên các bộ dữ liệu

	Fast Conformer		**Conformer**	
	WER	**Time (s)**	**WER**	**Time (s)**
RNTT				
Libri - clean	0.0196	**0.3115**	**0.0193**	0.4434
Libri - other	0.0386	**0.2913**	**0.0361**	0.3946
Minds14	0.1899	**0.2701**	**0.1862**	0.4103
Speech COCO	**0.0830**	**0.0536**	0.1001	0.0976
Google Fleurs	**0.1024**	**0.3418**	0.1092	0.5256
CTC				
Libri - clean	**0.0244**	**0.1252**	0.0251	0.1332
Libri - other	**0.0510**	**0.1247**	0.0526	0.1273
Minds14	**0.1963**	**0.1242**	0.2073	0.1247
Speech COCO	**0.1082**	**0.0129**	0.1171	0.0172
Google Fleurs	**0.1175**	**0.1126**	0.1186	0.1197

4.5 Phân tích lỗi

Nhóm sẽ tiến hành phân tích lỗi trong quá trình dự đoán của các mô hình để xác định và hiểu rõ hơn về các vấn đề mà các mô hình đang gặp phải.

4.5.1 Sai phạm ở ground truth

Nhóm đã tự nghe loại các đoạn audio để kiểm chứng và phát hiện được rằng ở bộ dữ liệu Minds14, các kết quả dự đoán mà mô hình đưa ra hoàn toàn đúng và ground truth đã bị gán sai. Do đó kết quả ở Bảng 4.1, các mô hình có thể dự đoán đúng nhưng do ground truth bị sai nên có WER cao, đây là lỗi và tồn tại của bộ dữ liệu này. Có 2 loại sai phạm chính là ‘Ground truth thiếu sót một số thông tin so với lời nói thực tế’ và ‘Ground truth chứa nội dung không chính xác so với lời nói thực tế’.

4.5.1.1 Ground truth thiếu sót một số thông tin so với lời nói thực tế

Điều này có thể do người ghi chép không nghe rõ hoặc ghi chép không đầy đủ đối với đoạn audio.

Bảng 4.4 Ví dụ cho Ground truth thiếu sót một số thông tin so với lời nói thực tế

Ground truh	**Prediction**
i like to pay my bill please thank you	**hi** i **would** like to pay my bill please **as it has an outstanding amount** thank you
could you show me my latest transaction please	um could you show me my latest transactions please **okay i recognize those payments**

Như ví dụ trên, mô hình đã dự đoán được các từ mà ground truth không có, và được các thành viên đánh giá là mô hình dự đoán đúng, đồng nghĩa với việc đồng ý về việc ground truth đã soạn thiếu sót thông tin.

4.5.1.2 Ground truth chứa nội dung không chính xác so với lời nói thực tế

Điều này có thể do nhiều nguyên nhân như người ghi chép đoạn audio sai, ghi sai chính tả hoặc hiểu sai ý nghĩa lời nói.

Bảng 4.5 Ví dụ cho Ground truth chứa nội dung không chính xác so với lời nói thực tế

Ground truh	**Prediction**
I dont want any more transaction I like hard please	I dont want anymore transactions **on my card** please
how about take someone today about my direct debit please	**id like to** speak to someone today about my direct abbots please

Như ví dụ trên, mô hình đã dự đoán được chữ ‘on my card’ thay vì ‘i like hard’ hoặc ‘id like’ thay vì ‘how about’, các từ mô hình đưa ra được các thành viên trong nhóm đánh giá là đúng, đồng nghĩa với việc đồng ý ground truth chứa nội dung sai lệch.

4.5.2 Sai phạm ở kết quả dự đoán

Đây là các sai phạm có ở tất cả các mô hình trên các bộ dữ liệu. Có hai loại thường gặp nhất là ‘Sai do audio không lưu loát’ và ‘Mô hình dự đoán sai từ’.

4.5.2.1 Sai do audio không lưu loát

Lỗi do đoạn audio chứa giọng ngập ngừng của người nói, như um, oh, er… dẫn đến việc mô hình tự nhận diện ngôn ngữ này và đưa ra kết quả dự đoán bị dư thừa.

Bảng 4.6 Ví dụ cho sai do audio không lưu loát

Ground truh	**Prediction**
hey there im just going to change my address	hey there **um** im just calling to change my address
yes i would like to open a joint account	yes i would like to **um er** open a joint account

4.5.2.2 Mô hình dự đoán sai từ

Lỗi do đoạn audio chứa các từ vựng dễ gây nhầm lẫn với các từ vựng khác (do có phiên âm, cách phát âm gần giống nhau) dẫn đến mô hình đưa ra các dự đoán chứa từ vựng sai đối với ground truth.

Bảng 4.7 Ví dụ cho mô hình dự đoán sai từ

Ground truh	**Prediction**
lock card immediately	lock **hard** immediately
im just wondering how i can deposit money into the account	im just wondering how i can deposit money into **this** account
existing line	existing **loan**

Chương 5: KẾT LUẬN

Tóm lại, trong bài báo này, các tác giả đã giới thiệu một mô hình mới là Fast Conformer với khả năng sử dụng ít hơn 2.9 lần tính toán mà vẫn giữ được WER gần như giống với Conformer gốc. Đánh giá trên các tác vụ speech translation và spoken language understanding cho thấy độ chính xác cao của mô hình trong khi đạt được độ tăng tốc đáng kể trong việc tính toán của encoder. Kết quả trên long-form audio transcription được cải thiện với việc thêm vào một attention token toàn cục đơn. Và cuối cùng, các tác giả cũng cho thấy kiến trúc của Fast Conformer có thể dễ dàng tăng đến 1B tham số giúp tăng độ chính xác trong khi chịu đựng được nhiễu khi huấn luyện trên các bộ dữ liệu lớn.

Thêm vào đó, nhóm đã phân tích và đánh giá hiệu suất của các mô hình Automatic Speech Recognition (ASR) khác nhau. bao gồm Conformer, Fast Conformer, Jasper, QuartzNet, CitriNet trên 5 bộ dữ liệu. Kết quả thu được cho thấy mô hình Conformer và Fast Conformer sử dụng Recurrent Neural Network Transducer cho ra kết quả tốt nhất theo độ đo WER, đặc biệt đối với tập test của bộ dữ liệu Libri-clean, mô hình Conformer cho kết quả WER 0.0193, cho thấy sự hiệu quả của các mô hình Conformer trên các tác vụ âm thanh. Fast Conformer cũng cho thấy hiệu suất rất cạnh tranh, mặc dù cho ra kết quả xấp xỉ so với mô hình Conformer, Fast Conformer lại có tốc độ dự đoán nhanh hơn đáng kể. Fast Conformer (RNNT) đạt WER là 0.1024 trên bộ dữ liệu Google Fleurs, và 0.0830 trên bộ dữ liệu Speech COCO, cho thấy sự cân bằng tốt giữa độ chính xác và tốc độ xử lý. Bên cạnh đó, QuartzNet mặc dù không cho kết quả tốt như 2 mô hình trên, song lại có tốc độ xử lý nhanh nhất trong tất cả các mô hình. Tóm lại, các mô hình Conformer và Fast Conformer có hiệu suất tốt nhất, cho thấy tiềm năng ứng dụng của các mô hình hình này trong các hệ thống nhận dạng giọng nói thực tế.

TÀI LIỆU THAM KHẢO

1. Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar1, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg, “Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition”, in ASRU, 2023
2. Li, J., Lavrukhin, V., Ginsburg, B., Leary, R., Kuchaiev, O., Cohen, J. M., ... & Gadde, R. T. (2019). Jasper: An end-to-end convolutional neural acoustic model. arXiv preprint arXiv:1904.03288.
3. Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., ... & Zhang, Y. (2020, May). Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6124-6128). IEEE.
4. NVIDIA, ‘Accelerating Conversational AI Research with New Cutting-Edge Neural Networks and Features from NeMo 1.0’, 2021 [Trực tuyến]. Địa chỉ: https://developer.nvidia.com/blog/accelerating-conversational-ai-research-with-new-cutting-edge-neural-networks-and-features-from-nemo-1-0/ [Truy cập lần cuối 25/05/2024]
5. Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5206-5210). IEEE.
6. Havard, W., Besacier, L., & Rosec, O. (2017). Speech-coco: 600k visually grounded spoken captions aligned to mscoco data set. arXiv preprint arXiv:1707.08435.
7. Gerz, D., Su, P., Kusztos, R., Mondal, A., Lis, M., Singhal, E., Mrksic, N., Wen, T., Vulic, I.: Multilingual and cross-lingual intent detection from spoken data. CoRR abs/2104.08524 (2021) 2104.0852416.
8. Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., Bapna, A.: Fleurs: Few-shot learning evaluation of universal representations of speech. arXiv preprint arXiv:2205.12446 (2022)