

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CUỐI KÌ MÔN XỬ LÝ TIẾNG NÓI
ỨNG DỤNG NỀN TẢNG CMUSPHINX
TRONG XỬ LÝ TIẾNG NÓI

Người hướng dẫn: TS NGUYỄN CHÍ THIỆN

Người thực hiện: HỒNG QUANG VINH – 186005004

NGUYỄN ĐẠI THỊNH – 186005035

Lớp: 18600531

Khoá: 2018-2020

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CUỐI KÌ MÔN XỬ LÝ TIẾNG NÓI
ỨNG DỤNG NỀN TẢNG CMUSPHINX
TRONG XỬ LÝ TIẾNG NÓI

Người hướng dẫn: TS NGUYỄN CHÍ THIỆN

Người thực hiện: HỒNG QUANG VINH – 186005004

NGUYỄN ĐẠI THỊNH – 186005035

Lớp: 18600531

Khoá: 2018-2020

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn Thầy Nguyễn Chí Thiện đã giúp đỡ chúng em hoàn thành đồ án. Những hướng dẫn của Thầy giúp chúng em có một nền tảng lý thuyết đủ để có thể ứng dụng và nghiên cứu phát triển đề tài này. Xin chân thành cảm ơn Thầy.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Nguyễn Chí Thiện;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Hồng Quang Vinh

Nguyễn Đại Thịnh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Ngày nay, với sự phát triển của cuộc Cách mạng Công nghiệp 4.0, tiên phong là sự phát triển của lĩnh vực Trí tuệ nhân tạo, con người đang tiến gần hơn tới mục đích làm cho máy tính mô phỏng được các hành vi của con người. Có thể kể đến một số lĩnh vực như nhận dạng hình ảnh, tiếng nói, chữ viết... Trong đó, nhận dạng tiếng nói được xem như lĩnh vực khó khăn nhất cũng như thành tựu đạt được tương đối ít nếu so với các lĩnh vực khác.

Sở dĩ nói là khó bởi vì nó bị ảnh hưởng bởi nhiều yếu tố. Tiếng nói được phát ra dưới dạng sóng âm, để nghe được tiếng nói ta cần phải qua hai bước là thu nhận và xử lý sóng âm đó. Tuy nhiên, tùy vào môi trường khác nhau, chúng ta có thể thu nhận được các dạng âm thanh khác nhau, kể cả khi nói cùng một từ giống hệt nhau. Đây chính là yếu tố đầu tiên và khó khăn nhất trong quá trình Xử lý tiếng nói.

Ngoài ra, tiếng Việt còn là một ngôn ngữ cực kỳ đa dạng và phong phú, chính sự đa dạng này làm cho việc phân tích dữ liệu trở nên khó khăn hơn. Trên thế giới đã có nhiều hệ thống nhận dạng tiếng nói dựa trên tiếng Anh, tuy nhiên đối với tiếng Việt vẫn còn hạn chế, cũng như không thể áp dụng cùng mô hình của các ngôn ngữ khác.

Mục đích của đề tài này là xây dựng được một hệ thống có thể nhận dạng tiếng Việt dựa trên lời nói, và chuyển đổi nó sang dạng chữ viết. Dựa trên những yêu cầu của bài toán, nhóm lựa chọn công cụ CMUSphinx làm công cụ hỗ trợ chính trong quá trình nghiên cứu. Các khái niệm và cách sử dụng được trình bày ở các phần sau của đồ án này.

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC	1
DANH MỤC CÁC BẢNG BIẾU, HÌNH VẼ, ĐỒ THỊ	4
CHƯƠNG 1 – MỞ ĐẦU	6
1.1 Giới thiệu đề tài	6
1.2 Mục tiêu và phạm vi đề tài.....	6
1.3 Cơ sở dữ liệu	7
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT	9
2.1 Phương pháp rút trích đặc trưng Mel Frequency Cepstral Coefficient (MFCC).....	9
2.1.1 Tiền nhấn (Pre-emphasis).....	10
2.1.2 Cửa sổ hoá (Windowing).....	11
2.1.3 Biến đổi Fourier nhanh (Fast Fourier Transform - FFT)	11
2.1.4 Lọc qua bộ lọc Mel-scale.....	11
2.1.5 Tính log năng lượng phô	12
2.1.6 Biến đổi Cosine rời rạc	12
CHƯƠNG 3 – GIỚI THIỆU CÔNG CỤ CMUSPHINX	13
3.1 Giới thiệu	13
3.2 Các mô hình trong CMUSphinx	13
3.3 Một số khái niệm khác.....	14
CHƯƠNG 4 – CÀI ĐẶT, CẤU HÌNH VÀ XÂY DỰNG ỨNG DỤNG VỚI CMUSPHINX	15
4.1 Cài đặt CMUSphinx	15
4.1.1 Môi trường cài đặt	15

4.1.2 Chuẩn bị các gói của CMUSphinx	15
4.1.3 Cài đặt CMUSphinx	16
4.1.4 Xây dựng bộ dữ liệu nhận dạng.....	18
4.1.4.1 Xây dựng bộ từ điển (Dictionary)	19
4.1.4.2 Xây dựng bộ mô hình ngôn ngữ (Language Model)	20
4.1.4.3 Xây dựng bộ mô hình âm thanh (Acoustic Model)	22
4.2 Triển khai ứng dụng lên Android	30
CHƯƠNG 5 – KẾT QUẢ THỬ NGHIỆM	34
5.1 Đánh giá thông qua sphinxtrain	34
5.2 Đánh giá thông qua thử nghiệm thực tế.....	36
5.2.1 Đối với môi trường yên tĩnh không tiếng ồn	36
5.2.2 Đối với môi trường nhiều tiếng ồn	37
5.3.3 Kết luận.....	37
CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	38
6.1 Các hạn chế.....	38
6.2 Những điều đã đạt được.....	38
6.3 Hướng phát triển	39
TÀI LIỆU THAM KHẢO	40

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

f Tân số (Hz)

CÁC CHỮ VIẾT TẮT

MFCC Mel Frequency Cepstral Coefficient

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1: Sơ đồ rút trích đặc trưng tổng quát	9
Hình 2: Hình các bước rút trích đặc trưng MFCC	10
Hình 3: Thư mục chứa các gói của CMUSphinx.....	16
Hình 4: Kết quả sau khi chạy install gói sphinxbase	17
Hình 5: Đường dẫn cho shared library.....	18
Hình 6: Chi tiết bộ từ điển	19
Hình 7: Hình chi tiết file ngữ âm	20
Hình 8: Hình chi tiết file text cho mô hình ngôn ngữ.....	21
Hình 9: Hình thư mục chứa mô hình ngôn ngữ sau khi generate	22
Hình 10: Chi tiết thư mục chứa file ghi âm	23
Hình 11: Hình chi tiết file “.transcription”	24
Hình 12: Hình chi tiết file “.fileids	25
Hình 13: Chi tiết file “.filter”	26
Hình 14: Hình thư mục etc chứa các mô hình đã xây dựng.....	27
Hình 15: Kết quả sau khi huấn luyện	28
Hình 16: Hình file report quá trình huấn luyện	29
Hình 17: Thư mục chứa bộ mô hình âm thanh sau khi được huấn luyện	30
Hình 18: Thư mục asset trong Android source chứa bộ dữ liệu đã được huấn luyện ...	31
Hình 19: Hình các bộ dữ liệu lên application	32
Hình 20: Hình file grammar giúp nhận diện cụm từ hoặc câu.....	33
Hình 21: Hình phương thức Listening để thu nhận tiếng nói bên ngoài	33
Hình 22: Hình vị trí của file “tuvung.align”	34
Hình 23: Hình kết quả kiểm tra của từng tập dữ liệu sphinxtrain.....	35
Hình 24: Hình kết quả kiểm tra sau cùng của sphinxtrain.....	35

DANH MỤC BẢNG

Bảng 1: Bảng bộ dữ liệu dùng để huấn luyện và kiểm tra của đề tài.....	8
Bảng 2: Bảng kết quả thu được trong môi trường yên tĩnh	36
Bảng 3: Bảng kết quả thu được trong môi trường tiếng ồn	37

CHƯƠNG 1 – MỞ ĐẦU

1.1 Giới thiệu đề tài

Xử lý tiếng nói bao gồm 2 công đoạn chính, đó là hiểu (thu nhận, phân tích) tiếng nói và xử lý đầu ra theo yêu cầu của bài toán. Trong đó công đoạn thu nhận và phân tích tiếng nói là công đoạn khó khăn nhất. Đối với con người, việc nghe và hiểu tiếng nói tương đối đơn giản, tuy nhiên với máy tính thì đó là một thách thức lớn. Nguyên nhân chủ yếu tới từ sự đa dạng, phong phú của từng ngôn ngữ khác nhau. Một câu khi nói ra là sự kết hợp của nhiều từ, và một từ lại là sự kết hợp của nhiều âm, .v.v.

Ý tưởng chính của phương pháp nhận dạng giọng nói có thể được tóm tắt bởi các bước: Thu nhận tiếng nói đầu vào dưới dạng sóng âm → Chia sóng âm ra thành các phần nhỏ → Rút đặc trưng của các vùng nhỏ này → Dựa vào đặc trưng có thể dự đoán được từ được nói.

Ngoài ra cần áp dụng kết hợp thêm các mô hình ngôn ngữ, mô hình HMM, .v.v. để tăng độ chính xác của việc dự đoán từ.

1.2 Mục tiêu và phạm vi đề tài

Việc xây dựng hệ thống có thể hiểu được tiếng Việt là cực kỳ khó khăn dựa trên số lượng từ vựng rất lớn cũng như sự kết hợp của các từ này. Vì thế mục tiêu chính của đề tài này là xây dựng hệ thống nhận dạng một vài từ cơ bản, ví dụ: Chữ số (Một → Mười), Động vật (Chó, mèo...), Phép tính cơ bản (Cộng, trừ...). Sau đó chuyển các từ nhận dạng được sang dạng chữ viết.

Những khó khăn phải đối mặt với đề tài này bao gồm:

- Cần một lượng dữ liệu lớn phục vụ cho quá trình huấn luyện, cụ thể là file ghi âm các từ cần nhận dạng.
- Để xây dựng một cụm từ/câu bao gồm nhiều từ, cần phải có một mô hình kết hợp giữa các từ, càng nhiều từ thì sự kết hợp càng lớn.

- Môi trường ghi âm và thực tế là khác nhau, dẫn đến kết quả thu được không thật sự chính xác.

1.3 Cơ sở dữ liệu

Bộ cơ sở dữ liệu huấn luyện bao gồm các đặc điểm sau:

- 20 từ vựng, mỗi từ ghi âm khoảng 50 lần, trong đó huấn luyện 40 tập và 10 tập dùng để kiểm tra.
- Dữ liệu âm thanh thu trong khoảng thời gian 1 - 2 giây.
- Chỉ thu âm 1 người trong không gian ít tiếng ồn, nhiễu.
- File ghi âm dưới dạng file .wav, sampling rate 8 kHz, 16 bit mono.

	Huấn luyện	Kiểm tra	Tổng
Không	40	10	50
Một	40	10	50
Hai	40	10	50
Ba	40	10	50
Bốn	40	10	50
Năm	40	10	50
Sáu	40	10	50
Bảy	40	10	50
Tám	40	10	50
Chín	40	10	50
Mười	40	10	50
Cộng	40	10	50

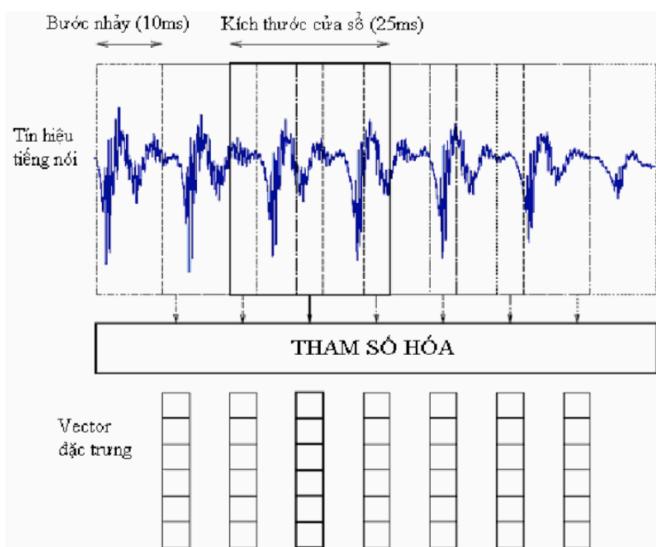
Trù	40	10	50
Nhân	40	10	50
Chia	40	10	50
Chó	40	5	45
Mèo	40	5	45
Gà	40	5	45
Chuột	40	5	45
Heo	40	4	44
Tổng cộng	800	174	974

Bảng 1: Bảng bộ dữ liệu dùng để huấn luyện và kiểm tra của đề tài

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

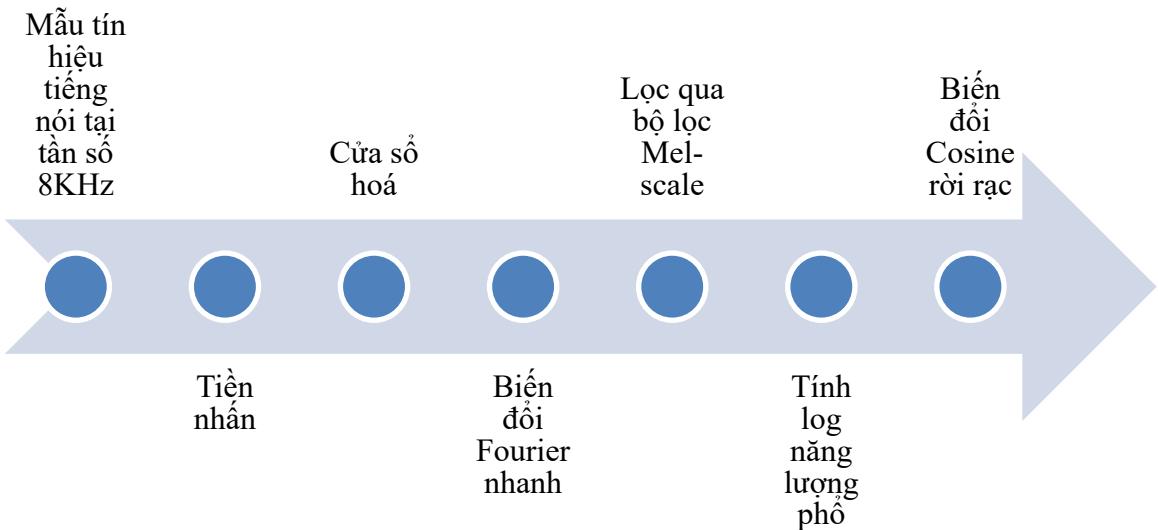
2.1 Phương pháp rút trích đặc trưng Mel Frequency Cepstral Coefficient (MFCC)

Quá trình rút trích đặc trưng là quá trình phân tích phổ (spectral analysis) nhằm xác định các thông tin quan trọng, đặc trưng của tín hiệu tiếng nói, tối thiểu hoá ảnh hưởng của nhiễu, xúc cảm, trạng thái của con người, cách phát âm của người nói, ...



Hình 1: Sơ đồ rút trích đặc trưng tổng quát

MFCC là phương pháp trích đặc trưng dựa trên đặc điểm cảm thụ tần số âm của tai người: tuyến tính đối với tần số nhỏ hơn 1kHz và phi tuyến đối với tần số trên 1kHz (theo thang tần số Mel, không phải theo Hz).



Hình 2: Hình các bước rút trích đặc trưng MFCC

2.1.1 Tiền nhấn (Pre-emphasis)

Chúng ta biết rằng phổ tiếng nói hữu thanh có khuynh hướng suy giảm toàn bộ - 6 dB/octave khi tần số tăng lên. Điều này là do khuynh hướng suy giảm -12 dB/octave của nguồn kích âm hữu thanh và tăng lên +6 dB/octave do phát âm miệng. Do đó cần phải bù +6 dB/octave trên toàn bộ băng tần. Điều này được gọi là pre-emphasis tín hiệu. Trong xử lý tín hiệu số, chúng ta dùng bộ lọc thông cao có tần số cắt 3 dB ở tần số trong phạm vi từ 100 Hz đến 1k Hz. Phương trình sai phân:

$$y(n) = x(n) - a * x(n) \quad (2.1)$$

Trong đó $y(n)$ là mẫu ra hiện tại của bộ lọc pre-emphasis, $x(n)$ là mẫu vào hiện tại, $x(n-1)$ là mẫu vào trước đó và a là hằng số thường được chọn giữa 0.9 và 1.

Lấy biến z của phương trình trên:

$$Y(z) = X(z) - az^{-1}X(z) = (1 - az^{-1})X(z) \quad (2.2)$$

Trong đó z^{-1} là toán tử trễ mẫu đơn vị. Suy ra hàm truyền $H(z)$ của bộ lọc:

$$H(z) = \frac{Y(z)}{X(z)} = 1 - az^{-1} \quad (2.3)$$

2.1.2 Cửa sổ hoá (Windowing)

Đầu tiên tín hiệu tiếng nói $x(n)$ sẽ được chia thành từng frame (có thực hiện chồng phủ một phần lên nhau - overlap) để được T frame $x_t'(n)$. Công việc cửa sổ hoá này sẽ được thực hiện bằng cách nhân tín hiệu tiếng nói với một hàm cửa sổ. Gọi phương trình cửa sổ hoá là $w(n)$ ($0 \leq n \leq N-1$; N : số mẫu trong 1 frame tín hiệu), khi đó tín hiệu sau khi được cửa sổ hoá là $X_t(n)$:

$$X_t(n) = x_t'(n) \cdot w(n) \quad (2.4)$$

Hàm cửa sổ thường được dùng là hàm cửa sổ Hamming:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N}\right); n = 0..N-1 \quad (2.5)$$

2.1.3 Biến đổi Fourier nhanh (Fast Fourier Transform - FFT)

Phổ tín hiệu sau khi nhân với cửa sổ Hamming sẽ sử dụng phép biến đổi Fourier nhanh Ta thu được biên độ phổ chứa các thông tin có ích của tín hiệu tiếng nói. Biến đổi Fourier nhanh - FFT (Fast Fourier Transform) là thuật toán rất hiệu quả để tính DFT của một chuỗi số. Ưu điểm là ở chỗ nhiều tính toán được lặp lại do tính tuần hoàn của số hạng Fourier $e^{-j\frac{2\pi}{N}kn}$. Dạng của DFT là:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}kn} \quad (2.6)$$

2.1.4 Lọc qua bộ lọc Mel-scale

Các nghiên cứu về hệ thống thính giác của con người cho thấy, tai người có cảm nhận đối với độ lớn các tần số không theo thang tuyến tính. Các đặc trưng phổ tần số của tiếng nói được tai người tiếp nhận như ngõ ra của một dãy các bộ lọc. Tần số trung tâm của các bộ lọc này không phân bố tuyến tính dọc theo trục tần số. Thành phần phổ dưới 1 kHz thường được tập trung nhiều bộ lọc hơn vì nó chứa nhiều thông tin về âm thanh hơn. Ở tần số thấp các bộ lọc bao gồm hẹp được sử dụng để tăng độ phân giải tần số để có được tần số cơ bản và họa tần vốn ổn định, còn ở tần số cao các bộ lọc thông rộng được sử dụng để thu được các thành phần tần số cao vốn biến động rất nhanh. Với nỗ lực nhằm mô tả chính xác sự tiếp nhận tần số của tai người, một thang tần số được xây

dụng - thang tần số Mel dựa trên cơ sở thực nghiệm cảm nhận nghe của người. Tần số 1 kHz được chọn là 1000 Mel. Mối quan hệ giữa thang tần số thực (vật lý) và thang tần số Mel (sinh lý) được cho bởi công thức:

$$F_{Mel} = 2595 \log_{10}(1 + \frac{F_{Hz}}{700}) \quad (2.7)$$

Với F_{Mel} là tần số sinh lý, đơn vị là Mel, F_{Hz} là đơn vị tần số thực, đơn vị Hz.

2.1.5 Tính log năng lượng phô

Sau khi qua bộ lọc Mel, phô tín hiệu $Y_t(m)$ sẽ được tính \log_{10} theo:

$$\log \{|Y_t(m)|^2\} \quad (2.8)$$

2.1.6 Biến đổi Cosine rời rạc

Bước cuối cùng để thu được các hệ số MFCC là lấy biến đổi Cosine rời rạc của kết quả cho bởi:

$$y_t^{(m)}(k) = \sum_{m=1}^M \log \{|Y_t(m)|^2\} \cos(k \left(m - \frac{1}{2}\right) \frac{\pi}{M}) \quad (2.9)$$

Thông thường số điểm rời rạc k của biến đổi ngược này được chọn $1 \leq k \leq 12$. Các hệ số MFCC chính là số điểm rời rạc này, ta có thể có 1-12 hệ số MFCC.

CHƯƠNG 3 – GIỚI THIỆU CÔNG CỤ CMUSPHINX

3.1 Giới thiệu

CMUSphinx, thường được gọi tắt là Sphinx, bao gồm một nhóm các hệ thống nhận dạng giọng nói, được phát triển tại Đại học Carnegie Mellon. CMUSphinx bao gồm các bộ:

- Pocketsphinx: Bộ thư viện được viết bằng C.
- Sphinxtrain: Công cụ huấn luyện mô hình âm thanh.
- Sphinxbase: Các thư viện hỗ trợ cho Pocketsphinx và Sphinxtrain.
- Sphinx4: Bộ thư viện được viết bằng Java.

Trong đề tài này nhóm sử dụng bộ thư viện Pocketsphinx dựa trên sự nhỏ gọn và tốc độ xử lý của bộ thư viện C.

3.2 Các mô hình trong CMUSphinx

Mô hình âm thanh (Acoustic Model): Chứa các đặc tính của âm thanh, các vector đặc trưng có thể xảy ra của mỗi âm vị. Bộ mô hình âm thanh có được sau quá trình huấn luyện, nó dùng để so sánh đặc trưng của các âm thanh được đưa vào, sau đó đưa ra dự đoán về các từ có thể xảy ra.

Từ điển ngữ âm (Phonetic Dictionary): Chứa bộ ánh xạ các từ tới các ngữ âm. Nó ghi nhận một từ được cấu tạo bởi các âm nào. Đây được xem là từ điển chứa sự kết hợp ở mức độ ngữ âm.

Vd: “Một” => “M” - “ột”

Mô hình ngôn ngữ (Language Model): Được xem là một bộ từ điển chứa sự kết hợp của các từ. Nó ghi nhận một từ thường đi cặp với từ nào, từ đó đưa ra dự đoán về từ, cụm từ có thể xảy ra. Nó giúp hạn chế đáng kể không gian tìm kiếm từ.

Vd: “Mèo” thường đi với “Con” => “Con mèo”

3.3 Một số khái niệm khác

Hidden Markov Models (HMM): Mô hình Markov ẩn là một mô hình dùng để phân tích các âm theo đặc trưng và giúp hệ thống xác định sự khác biệt giữa các âm thanh.

Linguist: Cơ sở dữ liệu của Sphinx, lưu trữ các âm thanh chuẩn làm mẫu để so sánh, phân tích và nhận dạng. Gồm 2 phần là: acoustic model và language model.

Lattice: lưới – là một đồ thị có hướng diễn tả các biến trong quá trình nhận dạng. Thường là chọn ra các tổ hợp không có trong thực tế nhất; Trong trường hợp đó đó, lưới (lattice) đóng vai trò trung gian để trình bày kết quả.

N-best list: danh sách các biến tốt nhất giống như lattice – lưới, nhưng các tổ hợp kết quả thưa hơn lattice.

Speech database: Cơ sở dữ liệu lời nói – là một tập hợp các đoạn ghi âm điển hình lấy từ các cơ sở dữ liệu nhiệm vụ. Nếu chúng ta phát triển hệ thống đối thoại, nó có thể đối thoại với các đoạn ghi âm của người dùng. Đối với hệ thống chính tả, nó có thể đọc các đoạn ghi âm. CSDL lời nói thường được dùng để huấn luyện, đồng điệu và kiểm tra hệ thống giải mã.

Text databases: các CSDL văn bản là các mẫu văn bản thu gom để dùng cho việc huấn luyện mô hình ngôn ngữ và nhiều ứng dụng khác. Thông thường, CSDL văn bản được thu gom trong một mẫu văn bản. Vấn đề là chuyển các tài liệu hiện tại (PDF, trang web, bản scan) thành các mẫu văn bản nói. Nghĩa là, bạn cần phải loại bỏ các thẻ và tiêu đề, mở rộng số hình thức nói của họ, và mở rộng các chữ viết tắt.

Word confusion networks: mạng các từ nghi vấn là các lưới lattice mà trật tự chắt chẽ của các nốt được lấy trong các cạnh của lưới lattice.

CHƯƠNG 4 – CÀI ĐẶT, CẤU HÌNH VÀ XÂY DỰNG ỦNG DỤNG VỚI CMUSPHINX

4.1 Cài đặt CMUSphinx

4.1.1 Môi trường cài đặt

Linux là môi trường hệ điều hành thích hợp nhất để cài đặt Sphinx và thực hiện huấn luyện. Ngoài ra Sphinx cũng có thể được cài đặt trong môi trường MacOS và Window, tuy nhiên sẽ có khác biệt giữa các hệ điều hành. Trong đề tài này nhóm sử dụng môi trường MacOS.

Các gói cần thiết cho việc cài đặt môi trường:

- Python - Ngôn ngữ lập trình Python, dùng để chạy các đoạn script như huấn luyện dữ liệu, .v.v. Có thể download ở trang chủ: <https://cmusphinx.github.io/wiki/download/>
- Homebrew - Công cụ quản lý gói trên MacOS
- Các gói để chạy bộ huấn luyện, bao gồm: gcc, automake, autoconf, libtool, bison, swig, python-dev, libpulse-dev. Sử dụng homebrew để cài đặt
 - homebrew install gcc
 - Chạy tương tự với các gói khác.

4.1.2 Chuẩn bị các gói của CMUSphinx

Các gói cần thiết cho quá trình cài đặt bao gồm:

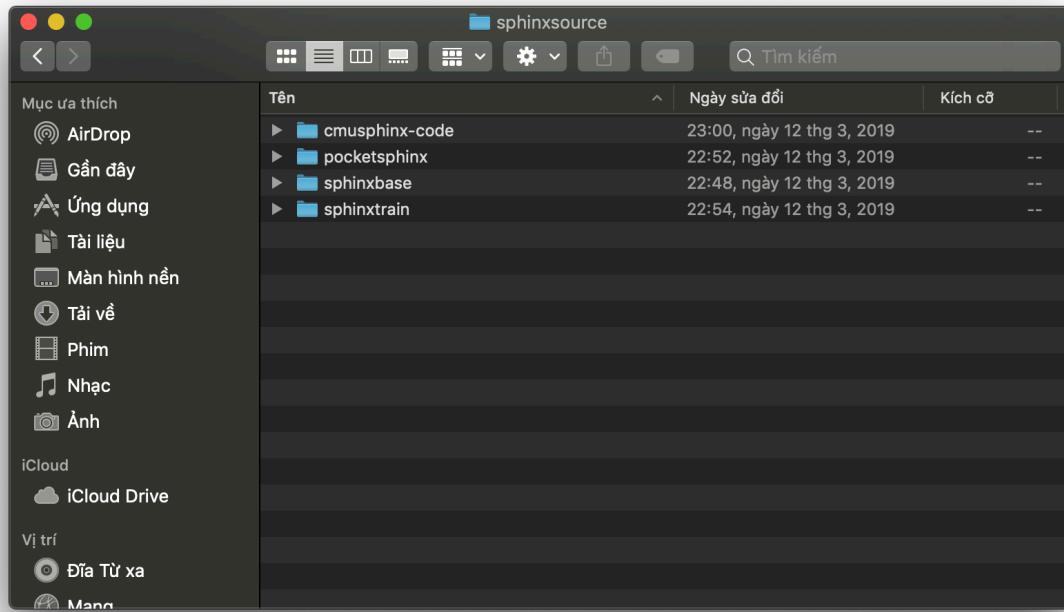
- Pocketsphinx - Bộ thư viện viết bằng C
- Sphinxbase - Thư viện hỗ trợ
- Sphinxtrain - Bộ công cụ dùng để huấn luyện
- Cmuclmtk - Bộ công cụ để xây dựng mô hình ngôn ngữ

Các gói trên có thể download trực tiếp trên trang chủ của CMU:

<https://cmusphinx.github.io/wiki/download/>

4.1.3 Cài đặt CMUSphinx

Bước 1: Di chuyển tới thư mục chứa các gói CMUSphinx đã tải về, giả sử chúng ta có thư mục sphinxsource như hình sau:

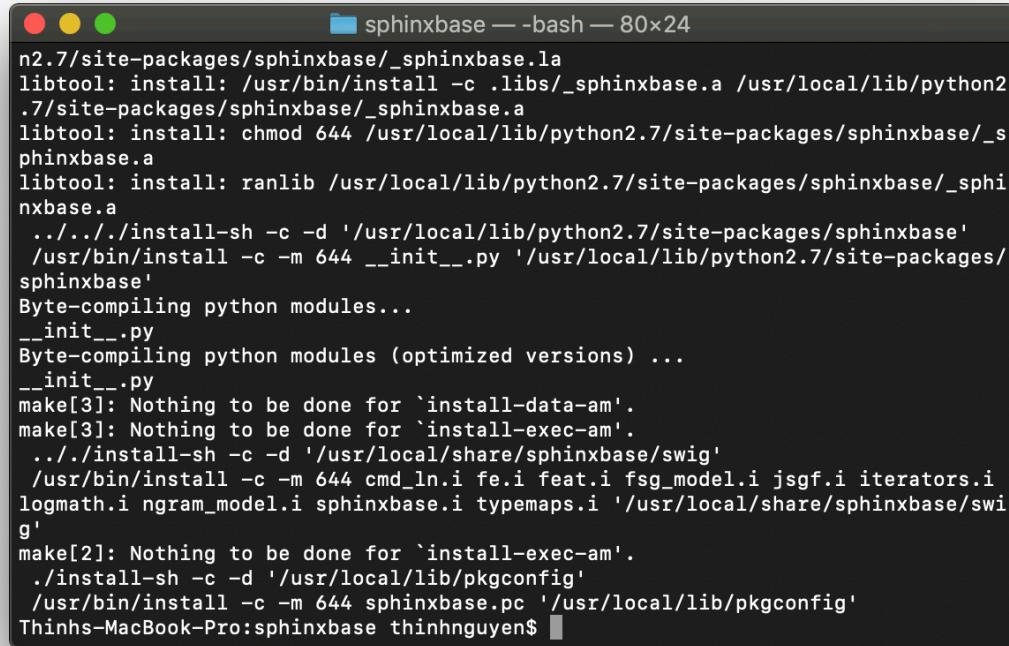


Hình 3: Thư mục chứa các gói của CMUSphinx

Bước 2: Sử dụng terminal trên MacOS và chạy các câu lệnh sau

- Di chuyển tới thư mục sphinxbase:
 - cd sphinxbase
- Cài đặt gói sphinxbase:
 - ./autogen.sh
 - make
 - sudo make install

Chạy xong 3 câu lệnh ta được như hình bên dưới:



```

n2.7/site-packages/sphinxbase/_sphinxbase.la
libtool: install: /usr/bin/install -c .libs/_sphinxbase.a /usr/local/lib/python2
.7/site-packages/sphinxbase/_sphinxbase.a
libtool: install: chmod 644 /usr/local/lib/python2.7/site-packages/sphinxbase/_s
phinxbase.a
libtool: install: ranlib /usr/local/lib/python2.7/site-packages/sphinxbase/_sphi
nxbase.a
..././install-sh -c -d '/usr/local/lib/python2.7/site-packages/sphinxbase'
/usr/bin/install -c -m 644 __init__.py '/usr/local/lib/python2.7/site-packages/
sphinxbase'
Byte-compiling python modules...
__init__.py
Byte-compiling python modules (optimized versions) ...
__init__.py
make[3]: Nothing to be done for `install-data-am'.
make[3]: Nothing to be done for `install-exec-am'.
..././install-sh -c -d '/usr/local/share/sphinxbase/swig'
/usr/bin/install -c -m 644 cmd_ln.i fe.i feat.i fsg_model.i jsgf.i iterators.i
logmath.i ngram_model.i sphinxbase.i typemaps.i '/usr/local/share/sphinxbase/swi
g'
make[2]: Nothing to be done for `install-exec-am'.
./install-sh -c -d '/usr/local/lib/pkgconfig'
/usr/bin/install -c -m 644 sphinxbase.pc '/usr/local/lib/pkgconfig'
Thinhs-MacBook-Pro:sphinxbase thinhnguyen$ █

```

Hình 4: Kết quả sau khi chạy install gói sphinxbase

Bước 3: Thực hiện tương tự các bước trên đối với các gói: sphinxtrain, pocketsphinx và cmuclmtk.

Bước 4: Thực hiện cài đặt đường dẫn shared library, sử dụng câu lệnh:

- export LD_LIBRARY_PATH=/usr/local/lib

Hoặc có thể tạo file “ld.so.conf”

- sudo nano /etc/ld.so.conf

Edit với nội dung như hình sau:

```
sphinxbase — nano ~ sudo — 80x24
GNU nano 2.0.6          File: /etc/ld.so.conf

include /etc/ld.so.conf.d/*.conf
/usr/local/lib

[ Read 3 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U Uncut Text ^T To Spell
```

Hình 5: Đường dẫn cho shared library

Việc cài đặt đã hoàn tất, tiến hành xây dựng bộ cơ sở dữ liệu cho quá trình huấn luyện.

4.1.4 Xây dựng bộ dữ liệu nhận dạng

Tuỳ vào mục tiêu bài toán mà xây dựng bộ dữ liệu nhận dạng phù hợp. Đối với đề tài này, nhóm xây dựng bộ dữ liệu khoảng 20 từ tiếng Việt, cụ thể như sau:

- Chữ số, từ “Không” tới “Mười”
- Phép toán cơ bản, bao gồm: “Cộng, Trừ, Nhân, Chia”
- Các con vật, bao gồm “Chó, Mèo, Gà, Chuột, Heo”

Các mô hình cần chuẩn bị cho bộ dữ liệu trên bao gồm:

- Bộ từ điển (Dictionary)

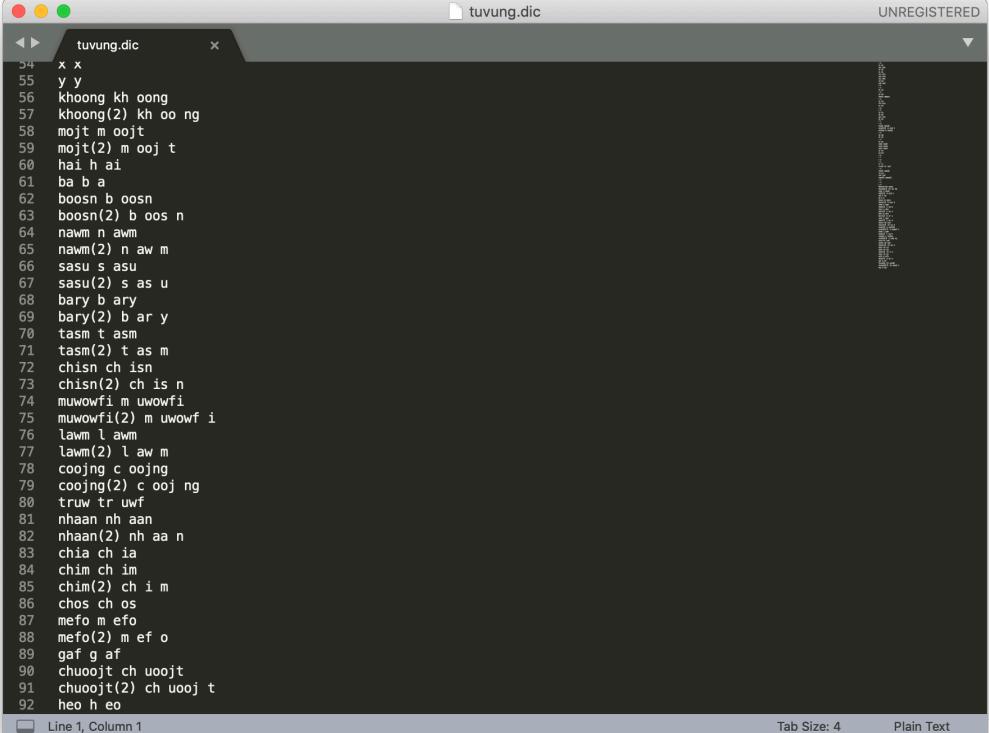
- Bộ mô hình ngôn ngữ (Language Model)
- Bộ mô hình âm thanh (Acoustic Model)

4.1.4.1 Xây dựng bộ từ điển (Dictionary)

Bộ từ điển bao gồm 2 file chính là file từ điển có đuôi “.dic” và file ngữ âm có đuôi “.phone”.

Đối với file từ điển “.dic”, ta cần xây dựng các từ cần nhận dạng, đi với nó là các ngữ âm câu tạo nên từ đó, mỗi từ cách nhau 1 hàng.

Đối với bảng mã unicode, việc nhận dạng còn chưa tốt, chính vì vậy nhóm sử dụng bảng mã ASCII cho từ, cũng như giữ nguyên kiểu gõ dấu Telex thay thế cho bảng mã unicode. Chi tiết file từ điển như hình sau:



```

tuvung.dic UNREGISTERED
54 x x
55 y y
56 khoong kh oong
57 khoong(2) kh oo ng
58 mojt m oojt
59 mojt(2) m ooj t
60 hai h ai
61 ba b a
62 boosn b oosn
63 boosn(2) b oos n
64 nawm n awm
65 nawm(2) n aw m
66 sasu s asu
67 sasu(2) s as u
68 bary b ary
69 bary(2) b ar y
70 tasm t asm
71 tasm(2) t as m
72 chisn ch isn
73 chisn(2) ch is n
74 muuwolfi m uwolfi
75 muuwolfi(2) m uwolf i
76 lawm l awm
77 lawm(2) l aw m
78 coojng c oojng
79 coojng(2) c ooj ng
80 truw tr uwf
81 nhaan nh aan
82 nhaan(2) nh aa n
83 chia ch ia
84 chim ch im
85 chim(2) ch i m
86 chos ch os
87 mefo m efo
88 mefo(2) m ef o
89 gaf g af
90 chuoojt ch uoojt
91 chuoojt(2) ch uooj t
92 heo h eo

```

Line 1, Column 1 Tab Size: 4 Plain Text

Hình 6: Chi tiết bộ từ điển

Đối với file ngữ âm, đây là file chưa toàn bộ các âm vị cấu tạo nên các từ nằm trong file từ điển, mỗi âm vị chỉ cần ghi 1 lần, cách nhau một hàng, cuối file là ký hiệu “SIL” tượng trưng cho khoảng nhiễu. Chi tiết file ngữ âm như hình dưới.

```
tuvung.phone
29  n̩
29  ch
30  tr
31  nh
32  oong
33  oojt
34  ai
35  a
36  oosn
37  awm
38  asu
39  ary
40  asm
41  isn
42  uwowfi
43  oojng
44  uwf
45  aan
46  ia
47  im
48  os
49  efo
50  af
51  uoojt
52  eo
53  SIL
```

Hình 7: Hình chi tiết file ngữ âm

4.1.4.2 Xây dựng bộ mô hình ngôn ngữ (Language Model)

Bộ mô hình ngôn ngữ được xây dựng dựa trên file text “.txt”. File này chứa các sự kết hợp có thể của từ này với từ khác. Ví dụ: Từ “Con” thường sẽ đi chung với từ “Mèo” để tạo thành từ “Con Mèo”. Tuy nhiên, trong phạm vi đồ án này, các số không có kết hợp với nhau, từ đó không có mô hình ngôn ngữ cho các số đó. Chính vì vậy, ở mô hình ngôn ngữ nhóm sử dụng lại luôn từ vựng đó. Chi tiết như hình dưới

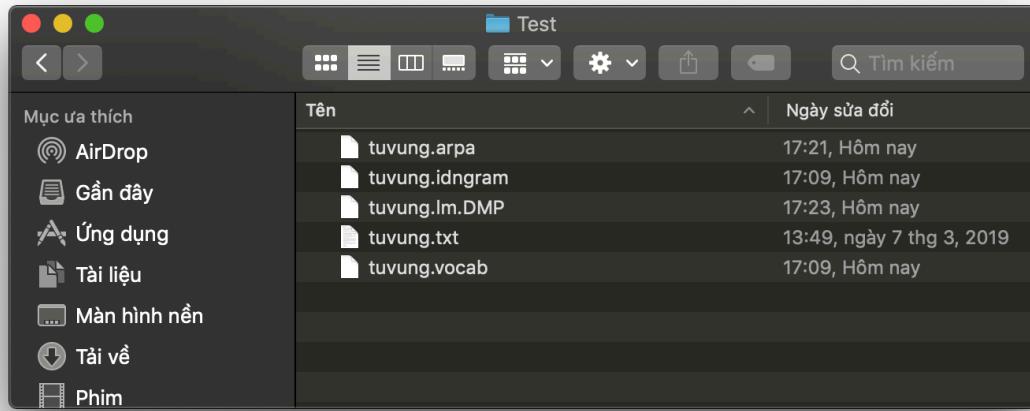
```
tuvung.txt
1 <s> khoong </s>
2 <s> moojt </s>
3 <s> hai </s>
4 <s> ba </s>
5 <s> boosn </s>
6 <s> nawm </s>
7 <s> sasu </s>
8 <s> bary </s>
9 <s> tasm </s>
10 <s> chisn </s>
11 <s> muwomi </s>
12 <s> muwowfi </s>
13 <s> khoong moojt hai ba boosn nawm sasu bary tasm chisn muwowfi </s>
14 <s> moojt coong mootj </s>
15 <s> moojt coong hai </s>
16 <s> moojt coong ba </s>
17 <s> moojt coong boosn </s>
18 <s> moojt coong nawm </s>
19 <s> moojt coong sasu </s>
20 <s> moojt coong bary </s>
21 <s> moojt coong tasm </s>
22 <s> moojt coong chisn </s>
23 <s> moojt coong muwowfi </s>
24 <s> moojt truwf mootj </s>
25 <s> moojt truwf hai </s>
26 <s> moojt truwf ba </s>
27 <s> moojt truwf boosn </s>
```

Hình 8: Hình chi tiết file text cho mô hình ngôn ngữ

Tiếp theo cần xây dựng mô hình ngôn ngữ dựa trên file text, sử dụng bộ công cụ cmuclmtk bằng các lệnh sau:

- text2wfreq < “tên_file”.txt | wfreq2vocab > “tên_file”.vocab
- text2idngram -vocab “tên_file”.vocab -idngram “tên_file”.idngram < “tên_file”.txt
- idngram2lm -vocab_type 0 -idngram “tên_file”.idngram -vocab “tên_file”.vocab -arpa “tên_file”.arpa
- sphinx_lm_convert -i “tên_file”.arpa -o “tên_file”.lm.DMP

Sau khi hoàn thành, ta được bộ mô hình ngôn ngữ chứa các file sau



Hình 9: Hình thư mục chứa mô hình ngôn ngữ sau khi generate

4.1.4.3 Xây dựng bộ mô hình âm thanh (Acoustic Model)

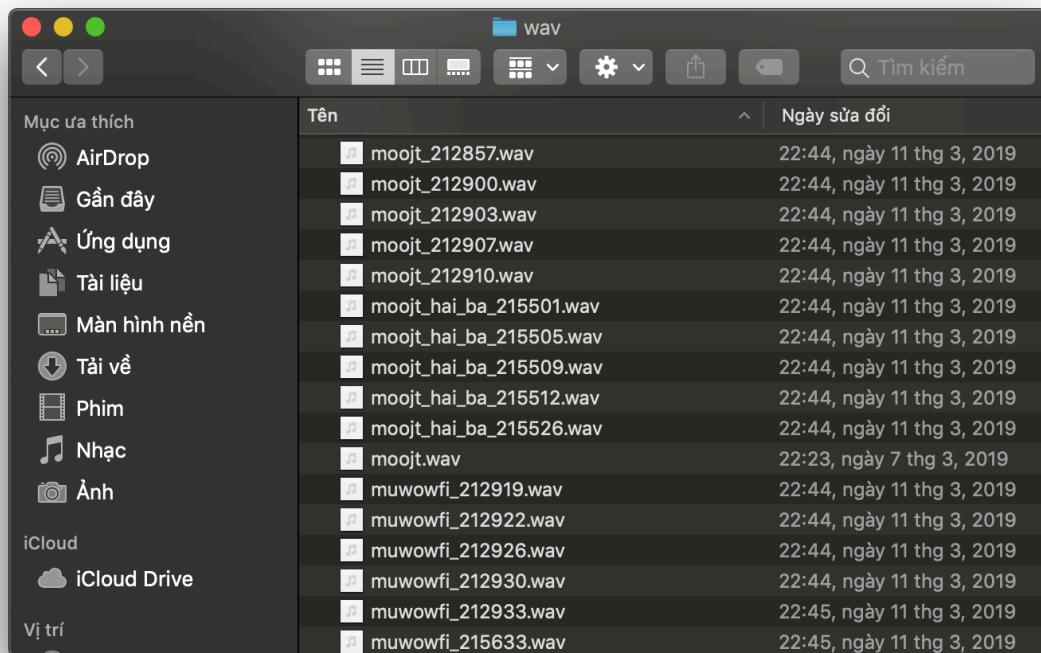
Mô hình âm thanh chứa các đặc trưng về âm thanh phục vụ cho quá trình nhận dạng. Để có mô hình âm thanh, việc đầu tiên ta cần xây dựng các file dữ liệu huấn luyện, cụ thể là các file ghi âm từ cần nhận dạng, cùng với đó là các file nhãn tương ứng để máy có thể học.

Các thành phần cần xây dựng để có thể huấn luyện bộ mô hình âm thanh bao gồm:

- Bộ file ghi âm các từ cần nhận dạng.
- File “.transcription” chứa tên các file ghi âm và nhãn dán tương ứng của nó.
- File “.fileids” chứa tất cả tên của các file ghi âm.
- File “.filter” chứa khoảng nhiều

Đầu tiên là xây dựng file ghi âm, một số điều cần lưu ý:

- Cần ghi âm lượng lớn các file để quá trình nhận dạng được chính xác, ở đây nhóm ghi âm khoảng 50 lần cho mỗi từ.
- Các file ghi âm cần được convert sang dạng file “.wav”, 16 bit mono, sampling rate 8kHZ (cho điện thoại) và lưu trong thư mục wav.



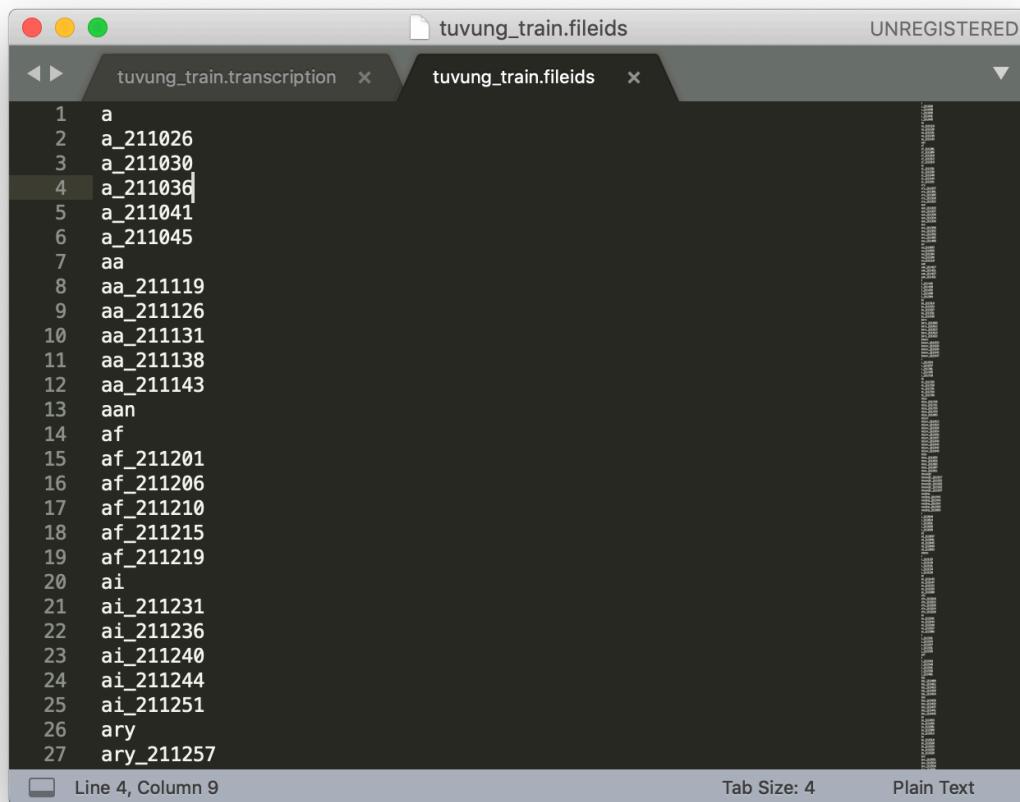
Hình 10: Chi tiết thư mục chứa file ghi âm

Tiếp theo là xây dựng file “.transcription” để dán nhãn các âm tương ứng với từ, chi tiết file như hình dưới

```
tuvung_train.transcription x
242 <s> moojt </s> (moojt)
243 <s> moojt </s> (moojt_212857)
244 <s> moojt </s> (moojt_212900)
245 <s> moojt </s> (moojt_212903)
246 <s> moojt </s> (moojt_212907)
247 <s> moojt </s> (moojt_212910)
248 <s> muwowfi </s> (muwowfi)
249 <s> muwowfi </s> (muwowfi_212919)
250 <s> muwowfi </s> (muwowfi_212922)
251 <s> muwowfi </s> (muwowfi_212926)
252 <s> muwowfi </s> (muwowfi_212930)
253 <s> muwowfi </s> (muwowfi_215633)
254 <s> muwowfi </s> (muwowfi_215636)
255 <s> muwowfi </s> (muwowfi_215638)
256 <s> muwowfi </s> (muwowfi_215641)
257 <s> muwowfi </s> (muwowfi_215643)
258 <s> muwowfi </s> (muwowfi_215646)
259 <s> muwowi </s> (muwowi)
260 <s> muwowi </s> (muwowi_212940)
261 <s> muwowi </s> (muwowi_212943)
262 <s> muwowi </s> (muwowi_212947)
263 <s> muwowi </s> (muwowi_212950)
264 <s> muwowi </s> (muwowi_212954)
265 <s> n </s> (n)
266 <s> n </s> (n_212958)
267 <s> n </s> (n_213001)
268 <s> n </s> (n_213006)
```

Hình 11: Hình chi tiết file “.transcription”

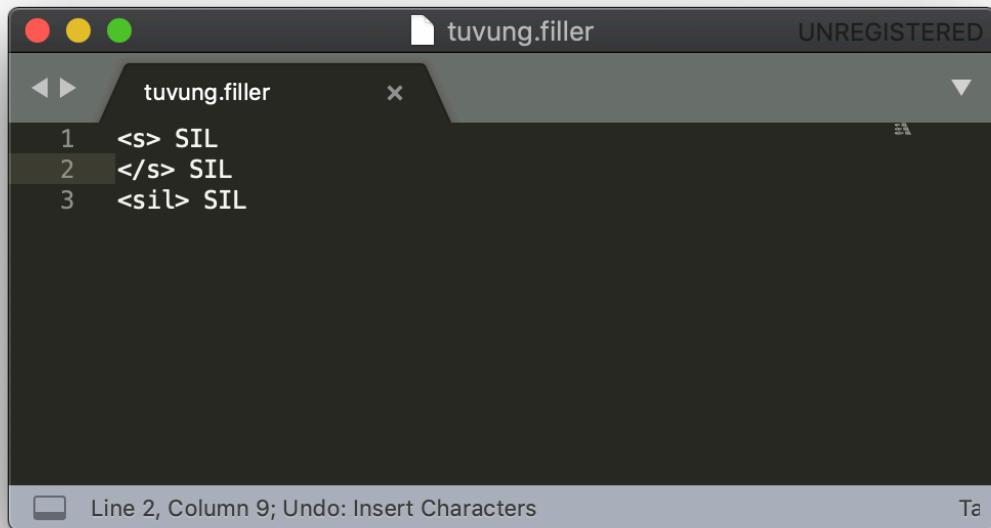
Tiếp theo là file “.fileids” chứa danh sách tên các file ghi âm đã tạo, chi tiết như hình dưới đây



```
tuvung_train.transcription x tuvung_train.fileids x UNREGISTERED
1 a
2 a_211026
3 a_211030
4 a_211036
5 a_211041
6 a_211045
7 aa
8 aa_211119
9 aa_211126
10 aa_211131
11 aa_211138
12 aa_211143
13 aan
14 af
15 af_211201
16 af_211206
17 af_211210
18 af_211215
19 af_211219
20 ai
21 ai_211231
22 ai_211236
23 ai_211240
24 ai_211244
25 ai_211251
26 ary
27 ary_211257
```

Hình 12: Hình chi tiết file “.fileids”

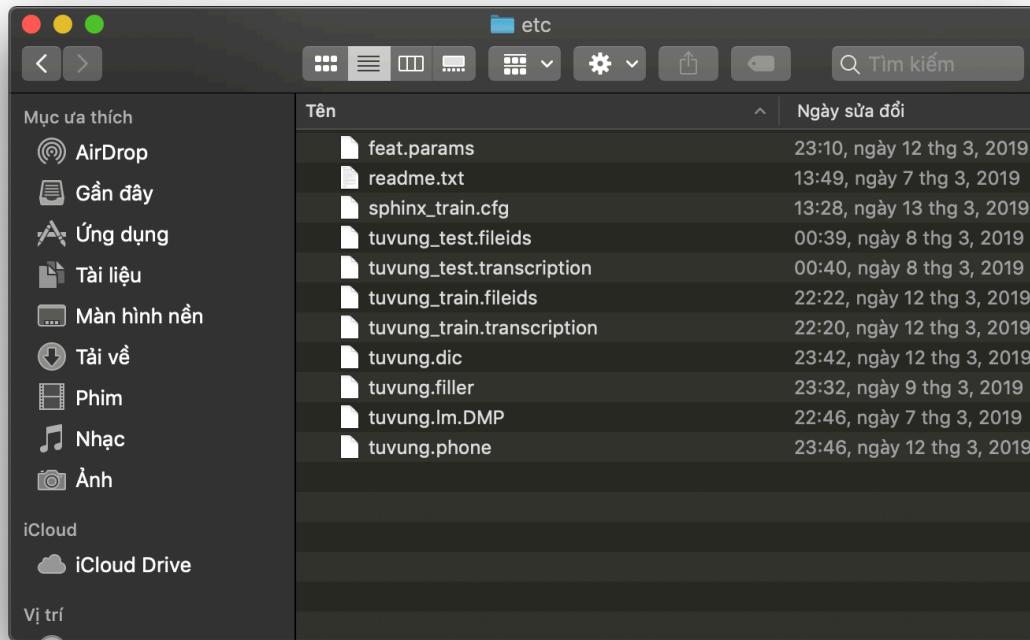
File “.filter” dùng để lọc ra các khoảng trắng giữa các lần nói, nội dung file này bao gồm các dòng như sau:



```
tuvung.filler
1 <s> SIL
2 </s> SIL
3 <sil> SIL
```

Hình 13: Chi tiết file “.filter”

Bước cuối cùng của công đoạn chuẩn bị, ta cần gom các file đã generate trước đó vào chung một thư mục, đặt tên là etc, cụ thể là các file sau:



Hình 14: Hình thư mục etc chứa các mô hình đã xây dựng

Tiếp theo cần sinh ra file config sử dụng cho quá trình huấn luyện, sử dụng terminal di chuyển tới thư mục chứa etc và wav, và chạy câu lệnh sau:

- sphinxtrain -t “tên_thư_mục” setup

Sau khi chạy sẽ sinh ra 2 file “feat.params” và “sphinx_train.cfg” như hình x ở trên. Ta cần quan tâm tới file config. Sử dụng text editor để chỉnh sửa file, một số biến đường dẫn cần lưu ý như sau:

- Đổi các biến sang “wav” nếu sử dụng định dạng này cho file ghi âm.
- Các đường dẫn tới các file “.dic”, “.phone”, “.filter”, “.fileids” và “.transcription” đã tạo ở bước trên:
- Thay đổi các giá trị sampling rate cho việc huấn luyện

Bước tiếp theo là tiến hành huấn luyện bằng dòng lệnh sau:

- sphinxtrain run

```

0% 10% 20% 30% 40% 50% 60% 70% 80% 90%
Normalization for iteration: 6
Current Overall Likelihood Per Frame = -142.304481529361
Training completed after 6 iterations
Skipped (set $CFG_CD_TRAIN = 'yes' to enable)
MODULE: 60 Lattice Generation
Skipped: $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 61 Lattice Pruning
Skipped: $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 62 Lattice Format Conversion
Skipped: $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 65 MMIE Training
Skipped: $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 90 deleted interpolation
Skipped for continuous models
MODULE: DECODE Decoding using models previously trained
        Decoding 174 segments starting at 0 (part 1 of 1)
        0%
        Aligning results to find error rate
        SENTENCE ERROR: 1.7% (3/174) WORD ERROR RATE: 1.7% (2/174)
Thinh-MacBook-Pro:tuvung thinhnguyen$ 

```

Hình 15: Kết quả sau khi huấn luyện

Sau khi huấn luyện sẽ cho ra kết quả

- Sentence Error: 1.7%
- Word Error Rate: 1.7%

Trong đó Word Error Rate (WER): Là tỉ lệ lỗi từ, thể hiện mức độ đúng của mô hình nhận dạng, được tính theo công thức:

$$\text{WER} = (I + D + S) / N \quad (4.1)$$

- I (Insert): Số từ thêm vào so với văn bản gốc
- D (Delete): Số từ bị bỏ ra so với văn bản gốc
- S (Substituted): Số từ bị thay thế

Tỉ lệ lỗi từ càng thấp chứng tỏ hiệu suất của mô hình càng cao.

Quá trình huấn luyện sẽ được in ra file “.html”, nếu như có bước nào báo lỗi, cần khắc phục những lỗi đó mới có thể train thành công.

```

file:///Users/thinhnguyen/Desktop/SpeechProcessing/TrainingData/tuvung/tuvung.html
Facebook | Training an acoustic model for CMUSphinx - CMUSphinx... | Speech Processing - Google Drive | tuvung | +
/Users/thinhnguyen/Desktop/SpeechProcessing/TrainingData/tuvung

Training tuvung on Thinks-MacBook-Pro.local

MODULE: 000 Computing feature from audio files (2019-03-13 13:33)
Extracting features from segments starting at (part 1 of 1)
    sphinx_fe Log File completed
Extracting features from segments starting at (part 1 of 1)
    sphinx_fe Log File completed
Feature extraction is done

MODULE: 00 verify training files (2019-03-13 13:33)
Phase 1: Checking to see if the dict and filler dict agrees with the phonelist file.
    Found 83 words using 53 phones passed
Phase 2: Checking to make sure there are not duplicate entries in the dictionary passed
Phase 3: Check general format for the fileids file; utterance length (must be positive); files exist passed
Phase 4: Checking number of lines in the transcript file should match lines in fileids file passed
Phase 5: Determine amount of training data, see if n_tied_states seems reasonable.
    Estimated Total Hours Training: 0.0740194444444444
    This is a small amount of data, no comment at this time WARNING
Phase 6: Checking that all the words in the transcript are in the dictionary
    Words in dictionary: 80
    Words in filler dictionary: 3 passed
Phase 7: Checking that all the phones in the transcript are in the phonelist, and all phones in the phonelist appear at least once passed

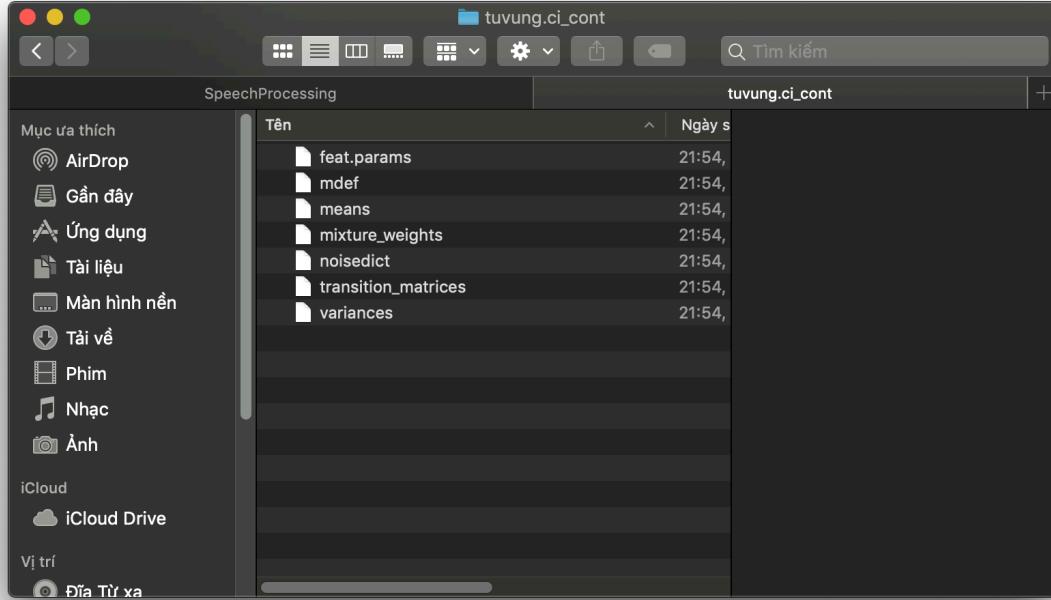
MODULE: 0000 train grapheme-to-phoneme model (2019-03-13 13:33)
Skipped (set $CFG_G2P_MODEL = 'yes' to enable)

```

Hình 16: Hình file report quá trình huấn luyện

Nếu như tất cả các bước đều complete thì quá trình huấn luyện đã hoàn tất. Đồng thời, quá trình huấn luyện sẽ sinh ra bộ mô hình âm thanh, được lưu trong thư mục “model_parameters/tên_file.ci_cont” mà sphinxtrain đã generate ra.

Các file giống như hình bên dưới:



Hình 17: Thư mục chứa bộ mô hình âm thanh sau khi được huấn luyện

Tổng kết lại, chúng ta đã có các bộ sau:

- Bộ từ điển: File “.dic”
- Bộ mô hình ngôn ngữ: File “.DMP”
- Bộ mô hình âm thanh: Toàn bộ thư mục “model_parameters/tên_file.ci_cont”

Quá trình xây dựng bộ nhận dạng đã hoàn tất, tiếp theo là nhúng các bộ này vào trong ứng dụng. Trong demo này nhóm xây dựng ứng dụng Android biến đổi các từ cơ bản từ tiếng nói sang dạng chữ và hiện ra màn hình.

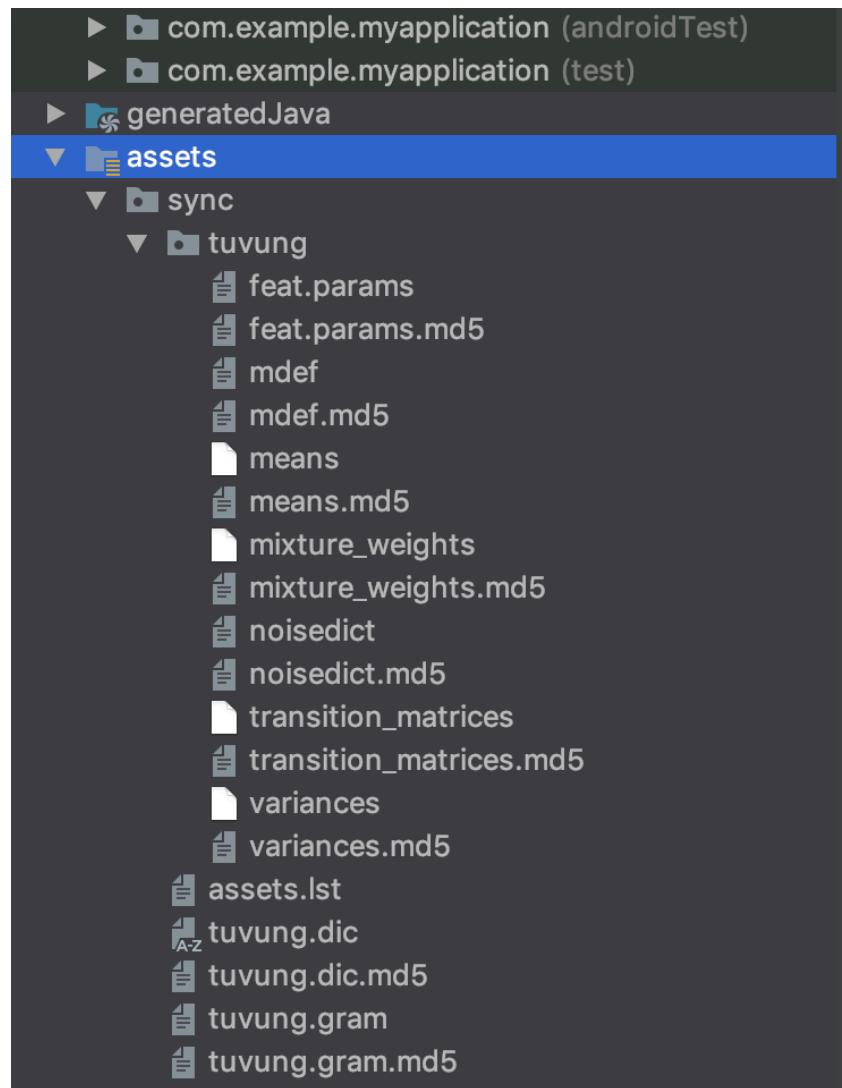
4.2 Triển khai ứng dụng lên Android

Để xây dựng ứng dụng trên Android, cần chuẩn bị những thứ sau đây:

- Android Studio/Android SDK: Có thể download trên trang: <https://developer.android.com/studio>

- Bộ thư viện **pocketsphinx** dưới dạng module “.aar”
 - Quá trình cài đặt và import thư viện vào Android Project được hướng dẫn trên trang chủ CMUSphinx:
- <https://cmusphinx.github.io/wiki/tutorialandroid/>

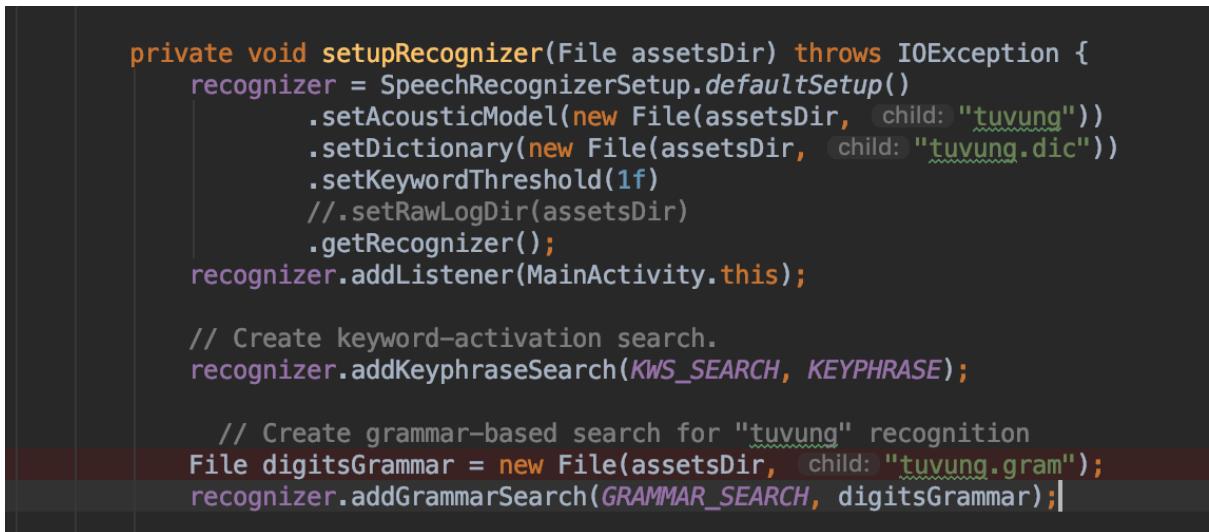
Trong source folder của project, tạo thư mục **asset-sync** và copy file từ điển “**.dic**” cùng bộ mô hình âm thanh vào thư mục đó.



Hình 18: Thư mục asset trong Android source chứa bộ dữ liệu đã được huấn luyện

Các file “.md5” là file được sinh ra sau quá trình build project, ta không cần quan tâm tới chúng.

Bước tiếp theo trong code ta cần load các mô hình âm thanh và từ điển trong asset và đưa vào bộ Recognizer trong thư viện pocketsphinx.



```

private void setupRecognizer(File assetsDir) throws IOException {
    recognizer = SpeechRecognizerSetup.defaultSetup()
        .setAcousticModel(new File(assetsDir, "child: " + "tuvung"))
        .setDictionary(new File(assetsDir, "child: " + "tuvung.dic"))
        .setKeywordThreshold(1f)
        // .setRawLogDir(assetsDir)
        .getRecognizer();
    recognizer.addListener(MainActivity.this);

    // Create keyword-activation search.
    recognizer.addKeyphraseSearch(KWS_SEARCH, KEYPHRASE);

    // Create grammar-based search for "tuvung" recognition
    File digitsGrammar = new File(assetsDir, "child: " + "tuvung.gram");
    recognizer.addGrammarSearch(GRAMMAR_SEARCH, digitsGrammar);
}

```

Hình 19: Hình các bộ dữ liệu lên application

Để nhận dạng chính xác 1 từ, vd: “Một”, ta sử dụng hàm:

- recognizer.addKeyphraseSearch(KWS_SEARCH, KEYPHRASE);

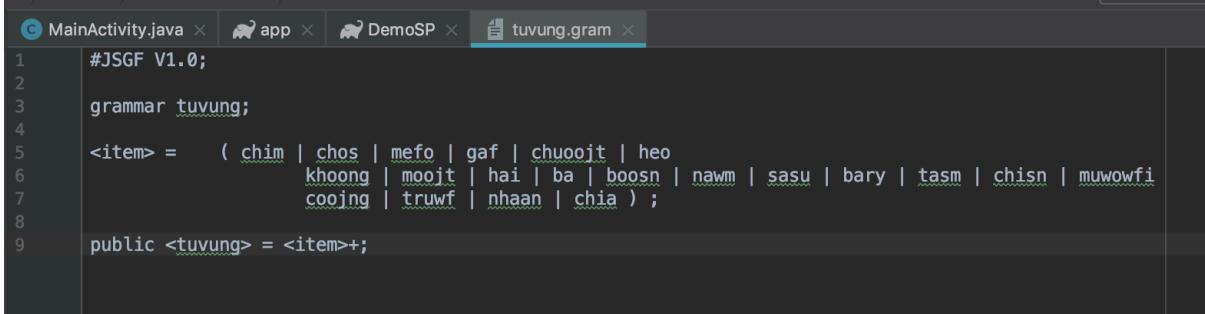
Trong đó:

- KWS_SEARCH: Là key để nhận dạng.
- KEYPHRASE: Là từ chính xác cần nhận dạng, giả sử đưa vào từ “Một”, hệ thống sẽ nhận dạng từ ta nói vào có khớp với “Một” hay không.

Để nhận dạng nhiều từ hoặc 1 cụm từ trong 1 câu, ta có thể sử dụng hàm:

- recognizer.addGrammarSearch(GRAMMAR_SEARCH, digitsGrammar);

Trong đó đưa vào 1 key và 1 file “.gram” được xây dựng bằng tay.

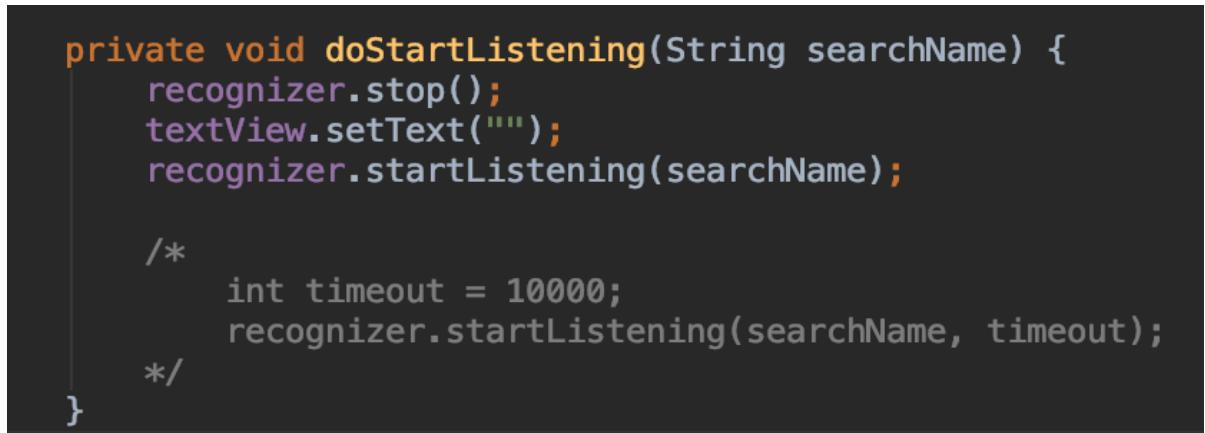


```

1 #JSGF V1.0;
2
3 grammar tuvung;
4
5 <item> = ( chim | chos | mefo | gaf | chuooit | heo
6           | khoong | moojt | hai | ba | boosn | nawm |
7           | coojng | truwf | nhaan | chia ) ;
8
9 public <tuvung> = <item>+;

```

Hình 20: Hình file grammar giúp nhận diện cụm từ hoặc câu
Sử dụng hàm **Listening** để nghe tiếng nói bên ngoài vào, trong đó **searchName** chính là các key tương ứng với các kiểu nhận dạng.



```

private void doStartListening(String searchName) {
    recognizer.stop();
    textView.setText("");
    recognizer.startListening(searchName);

    /*
        int timeout = 10000;
        recognizer.startListening(searchName, timeout);
    */
}

```

Hình 21: Hình phương thức Listening để thu nhận tiếng nói bên ngoài
Ta cần có 1 listener dung cho việc callback mỗi khi hệ thống nhận dạng được tiếng nói, cần implement các phương thức sau:

Trong đó quan trọng là 2 phương thức:

- **onPartialResult(Hypothesis hypothesis)**: Trả về kết quả là 1 Hypothesis, trong đó chưa kết quả ở dạng text nếu hệ thống nhận dạng tiếng nói đầu vào đúng với từ cần tìm kiếm ở hàm startListening. Hàm này được gọi liên tục trong quá trình listening.
- **onResult(Hypothesis hypothesis)**: Trả về kết quả cuối cùng sau khi hàm stop được gọi.

CHƯƠNG 5 – KẾT QUẢ THỬ NGHIỆM

Thử nghiệm là một quá trình quan trọng, nó giúp chỉ ra độ chính xác của mô hình khi áp dụng vào thực tế. Chính vì vậy, quá trình thực nghiệm cần phải chi tiết và khách quan để có thể đánh giá mô hình một cách cẩn kẽ nhất. Trong đồ án này, nhóm chia ra 2 hình thức đánh giá chính, bao gồm:

- Đánh giá bằng công cụ Sphinxtrain.
- Đánh giá bằng thử nghiệm thực tế.

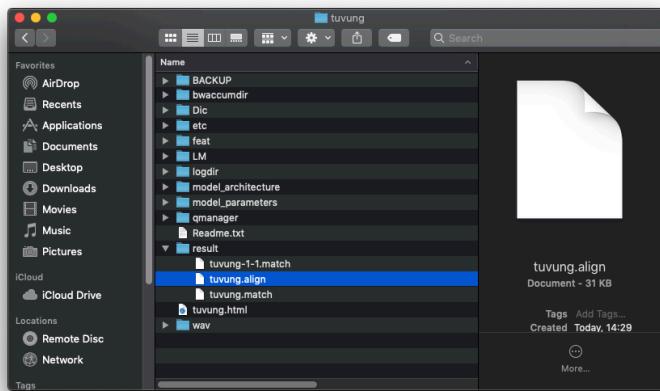
Ngoài ra nhóm còn áp dụng thêm đánh giá chéo (cross validation) để tăng thêm độ tin cậy của mô hình.

Ở cả 2 phương pháp, nhóm sử dụng chung một giá trị để thể hiện kết quả của mô hình, được gọi là độ chính xác. Được thể hiện theo công thức sau:

$$\text{Độ chính xác} = \frac{\text{Số lượng từ đúng}}{\text{Tổng số từ}} \times 100\% \quad (5.1)$$

5.1 Đánh giá thông qua sphinxtrain

Sau khi quá trình huấn luyện hoàn tất, sphinxtrain sẽ thực hiện ghi kết quả ra file “tuvung.align” nằm trong thư mục result.



Hình 22: Hình vị trí của file “tuvung.align”

File này sẽ thống kê toàn bộ quá trình chạy test của bộ sphinxtrain, gồm các thông tin như hình sau:

```

59
60  BARY      (user-BARY34)
61  CHUOOJT  (user-BARY34)
62  Words: 1 Correct: 0 Errors: 1 Percent correct = 0.00% Error = 100.00% Accuracy = 0.00%
63  Insertions: 0 Deletions: 0 Substitutions: 1
64

```

Hình 23: Hình kết quả kiểm tra của từng tập dữ liệu sphinxtrain

Giải thích hình x:

- Ở dòng đầu tiên, sphinxtrain đưa vào 1 file ghi âm tên là “BARY34” file này chưa dữ liệu từ “Bảy” và đã được dán nhãn “BARY”.
- Dòng thứ hai, khi thực hiện test đưa vào file “BARY34”, hệ thống đưa ra kết quả là “CHUOOJT” thể hiện cho từ “Chuột”.
- Khi so sánh thấy kết quả khác với nhãn đã dán trước đó, sphinxtrain sẽ đưa ra kết quả sai và thể hiện ở dòng thứ 3 (Error = 100.00%).
- Nếu kết quả hệ thống khớp với nhãn đã dán trước đó, kết quả ở dòng thứ 3 sẽ hiện ra Percent correct = 100.00% và Error = 0.00%.

Sau khi chạy hết tất cả các tập kiểm tra, hệ thống sẽ đưa ra kết quả như hình dưới đây:

```

698
699  TOTAL Words: 174 Correct: 171 Errors: 3
700  TOTAL Percent correct = 98.28% Error = 1.72% Accuracy = 98.28%
701  TOTAL Insertions: 0 Deletions: 1 Substitutions: 2
702

```

Hình 24: Hình kết quả kiểm tra sau cùng của sphinxtrain

Kết quả thử nghiệm bao gồm các thông số:

- Words: 174 tương ứng với 174 tập dữ liệu dùng để kiểm tra.
- Correct: 171 tương ứng với 171 từ chính xác.
- Errors: 3 tương ứng với 3 từ sai.
- Percent correct = 98.28% tương ứng với độ chính xác.

- Error = 1.72% tương ứng với tỉ lệ lỗi từ.

5.2 Đánh giá thông qua thử nghiệm thực tế

Thử nghiệm thực tế (thực nghiệm) chính là phương pháp quan sát nhất để đánh giá độ chính xác của mô hình. Phương pháp này là sử dụng tiếng nói của nhiều người trên nhiều môi trường khác nhau để kiểm tra.

Cụ thể, nhóm chia thành 2 người để nói, và mỗi người nói trong 2 môi trường khác nhau (yên tĩnh / ồn ào).

Người 1 là người trực tiếp thu âm huấn luyện, người 2 là người ngoài để giúp cho đánh giá trở nên khách quan hơn.

Mô hình được đánh giá thông qua khái niệm độ chính xác được mô tả ở đầu Chương 5.

5.2.1 Đối với môi trường yên tĩnh không tiếng ồn

	Người 1	Người 2
Lần 1	97% (29/30)	90% (27/30)
Lần 2	90% (27/30)	83% (25/30)
Lần 3	93% (28/30)	83% (25/30)
Lần 4	93% (28/30)	87% (26/30)
Lần 5	97% (29/30)	83% (25/30)
Trung bình	94%	85.2%

Bảng 2: Bảng kết quả thu được trong môi trường yên tĩnh

Trên bảng x, Đối với người trực tiếp thu âm huấn luyện, độ chính xác trung bình tương đối cao: 94%

Người 2 là người không trực tiếp huấn luyện, kết quả thực nghiệm cho thấy tỉ lệ chính xác không cao bằng, độ chính xác trung bình: 85.2%

5.2.2 Đối với môi trường nhiều tiếng ồn

	Người 1	Người 2
Lần 1	90% (27/30)	76% (23/30)
Lần 2	90% (27/30)	83% (25/30)
Lần 3	87% (26/30)	83% (25/30)
Lần 4	93% (28/30)	80% (24/30)
Lần 5	97% (29/30)	76% (23/30)
Trung bình	91.4%	79.6%

Bảng 3: Bảng kết quả thu được trong môi trường tiếng ồn

Đối với môi trường nhiều tiếng ồn, kết quả thu được là khá thấp, ở Người 1 có độ chính xác là 91.4%, trong khi Người 2 có độ chính xác 79.6%

5.3.3 Kết luận

Dựa trên bảng đánh giá và kết quả thu được, có thể thấy mô hình phụ thuộc rất nhiều vào 2 yếu tố: Người nói và môi trường xung quanh. Chính vì âm thanh khi phát ra ở mỗi lần không giống nhau, cộng với việc ảnh hưởng của môi trường, mà kết quả thu về có thể rất thấp.

CHƯƠNG 6 – KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Các hạn chế

Dựa trên những đánh giá từ kết quả thu được, có thể thấy những mặt hạn chế của ứng dụng như sau:

- Bộ cơ sở dữ liệu về từ vựng còn ít, ứng dụng chỉ mới nhận dạng được khoảng 20 từ, rất ít nếu so với bộ từ vựng hiện có của tiếng Việt.
- Ứng dụng chỉ có thể nhận dạng tốt ở một người nói, cụ thể là người trực tiếp thu âm và huấn luyện, đối với những người khác, ứng dụng cho ra kết quả sai khá nhiều, thậm chí không thể nhận dạng ở một số trường hợp đặc biệt (vd: tiếng địa phương .v.v)
- Việc nhận dạng còn kém khi ở trong môi trường nhiễu, cụ thể là môi trường nhiều tiếng ồn.

6.2 Những điều đã đạt được

Tuy ứng dụng còn nhiều mặt hạn chế, nhưng thông qua quá trình nghiên cứu và áp dụng, nhóm chúng em cơ bản đã thực hiện được một số điểm:

- Tìm hiểu về tiếng nói, các phương pháp rút trích, xử lý tiếng nói.
- Tìm hiểu và thực hiện huấn luyện mô hình sử dụng công cụ CMUSphinx, áp dụng cho tiếng Việt.
- Xây dựng bộ cơ sở dữ liệu nhỏ về xử lý tiếng nói tiếng Việt.
- Xây dựng thành công chương trình nhận dạng tiếng Việt trên thiết bị Android.

6.3 Hướng phát triển

Những kết quả thu được chính là nền tảng quý báu, từ đó có thể phát triển thêm một số hướng cho đồ án như:

- Mở rộng bộ cơ sở dữ liệu, hướng đến mục tiêu nhận dạng nhiều người, nhiều ngữ cảnh... để hệ thống đạt độ chính xác cao hơn.
- Tìm hiểu và áp dụng thêm các kỹ thuật học máy thích hợp vào quá trình xây dựng mô hình, tiến hành thực nghiệm để tìm ra mô hình tối ưu với độ chính xác cao.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Huỳnh Thanh Giàu (2012), “Nghiên cứu về nhận dạng tiếng nói tiếng Việt và ứng dụng thử nghiệm trong điều khiển máy tính”, Đồng Nai.
2. Huỳnh Thanh Huy – Nguyễn Hoàng Vũ (2015), “Nghiên cứu về nhận dạng tiếng Việt”, Thành phố Hồ Chí Minh.

Tiếng Anh

1. CMUSphinx Tutorial, <https://cmusphinx.github.io/wiki/tutorialconcepts/>.