

SPEAKNOW: BUILDING AN EFFECTIVE SPEECH-TO-TEXT SYSTEM FOR THE HILIGAYNON LANGUAGE USING A KALDI-BASED ASR MODEL

A Special Problem

Presented to

the Faculty of the Division of Physical Sciences and Mathematics

College of Arts and Sciences

University of the Philippines Visayas

Miag-ao, Iloilo

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science in Computer Science by

GONZALES, Benjie Jr.

PANIZALES, John Patrick

PIORQUE, Lester

Francis D. DIMZON

Adviser

June 5, 2023

Approval Sheet

The Division of Physical Sciences and Mathematics, College of Arts and
Sciences, University of the Philippines Visayas

certifies that this is the approved version of the following special problem:

THIS IS THE TITLE OF YOUR SPECIAL PROBLEM

Approved by:

Name	Signature	Date
_____	_____	_____
(Adviser)		
_____	_____	_____
(Co-Adviser)		
_____	_____	_____
(Reader)		
_____	_____	_____
(Division Chair)		

Division of Physical Sciences and Mathematics

College of Arts and Sciences

University of the Philippines Visayas

Declaration

I/We, GONZALES, PANIZALES and PIORQUE, hereby certify that this Special Problem, including the pdf file, has been written by me/us and is the record of work carried out by me/us. Any significant borrowings have been properly acknowledged and referred.

Name

Signature

Date

(Student)

(Student)

(Student)

Dedication

We, the SpeakNow team, would like to dedicate this special problem to all the individuals who have supported us throughout this journey. To our families, whose unwavering love and encouragement have been our pillars of strength, we extend our heartfelt gratitude. To our project adviser, Sir Francis D. Dimzon, for his guidance, expertise, and invaluable insights, we are deeply grateful. To our friends and colleagues, for their camaraderie and support, thank you for being by our side. Finally, we dedicate this work to the countless individuals whose lives may be positively impacted by the knowledge gained from our research. May this paper contribute to the advancement of knowledge, inspire future explorations, and make a meaningful difference in the world.

Acknowledgment

We would like to express our heartfelt gratitude to our esteemed project adviser, Sir Francis D. Dimzon, for his invaluable guidance, expertise, and unwavering support throughout the duration of this special problem. His deep knowledge, insightful feedback, and mentorship have been pivotal in shaping the direction and quality of our research. We are truly grateful for his dedication and commitment to our academic growth.

We also sincerely appreciate the participants of this project whose involvement and cooperation were integral to its success. Their dedication of time and effort made significant contributions to the results of our special problem. We are deeply grateful for their valuable contributions and trust in our research.

Furthermore, we would like to acknowledge the invaluable assistance and support provided by our colleagues and friends who have stood by us throughout this undertaking. Their encouragement, discussions, and exchange of ideas have played a crucial role in shaping and refining our analysis.

Lastly, we would like to thank our families for their unwavering support and understanding. Their encouragement, patience, and belief in our abilities have been a constant source of motivation during challenging times.

Together, the collective contributions and support of our project adviser, participants, colleagues, and families have made this research endeavor possible. We are deeply grateful for their involvement and contributions, as they have played a significant role in the successful completion of this special problem.

Abstract

This research paper presents a speech-to-text (STT) system for the Hiligaynon language using an Automatic Speech Recognition (ASR) model trained with Kaldi. The ASR model was trained on a corpus of approximately 3,500 Hiligaynon words. From this corpus, a subset of 1,000 words was randomly selected and recorded by 18 speakers (9 males and 9 females). The models trained included Monophone, Delta-based Triphone, LDA+MLLT, LDA+MLLT+SAT, and DNN, employing a six-fold cross-validation scheme. The DNN model yielded the lowest average Word Error Rate (WER) score and was chosen for the STT system development. In addition, the Whisper toolkit was utilized to transcribe the same audio dataset. Comparing different pre-trained model sizes, it was found that performance was proportional to the model size. The small model achieved a WER of 25.50%. This research contributes to the development of an effective STT system for Hiligaynon, emphasizing the significance of acoustic model selection and the influence of model size on transcription performance.

Keywords: Automatic Speech Recognition (ASR), Hiligaynon language, acoustic modeling, neural networks, Whisper, OpenAI, Word Error Rate (WER), Speech-to-Text, STT

Contents

1	Introduction	1
1.1	Overview of the Current State of Technology	1
1.2	Problem Statement	2
1.3	Research Objectives	3
1.3.1	General Objective	3
1.3.2	Specific Objectives	3
1.4	Scope and Limitations of the Research	3
1.5	Significance of the Research	4
2	Review of Related Literature	5
2.1	Automatic Speech Recognition	5
2.1.1	Lexicon Model	6

2.1.2	Acoustic Model	6
2.1.3	Language Model	7
2.1.4	Decoding	7
2.2	The Language Dilemma	7
2.2.1	Whisper by OpenAI	8
2.3	Hiligaynon Speech Recognition	9
2.4	The Hiligaynon Language	10
2.5	Kaldi ASR Toolkit	11
2.6	Acoustic Models in Kaldi	12
2.6.1	Monophone Model	12
2.6.2	Triphone Model	13
2.6.3	LDA + MLLT Model	13
2.6.4	LDA + MLLT + SAT Model	13
2.6.5	DNN Model	14
3	Research Methodology	15
3.1	Research Activities	15
3.1.1	Data Gathering and Preprocessing	15

<i>CONTENTS</i>	ix
3.1.2 Preprocessing	17
3.1.3 Training	17
3.1.4 Evaluation	19
3.1.5 System Development	20
4 Results and Discussions/Analyses	21
4.1 Results of Monophone model	22
4.2 Results of Triphone model	26
4.3 Result of LDA + MLLT model	30
4.4 Result of LDA + MLLT + SAT model	35
4.5 Result of DNN model	39
4.6 Summary of results	43
4.7 Results on the performance of Whisper	44
5 Conclusion	47
5.1 Recommendations	48
6 References	49
A Appendix	53

A.1	Command Line STT system	53
A.2	Code snippets	54

List of Figures

- A.1 Screenshot of the system when called without an argument. 53
- A.2 Screenshot of the system with an audio file's file name as an argument. 54

List of Tables

2.1	Table of the 25 significant sounds in Hiligaynon (Wolfenden, 2019)	10
4.1	Results of first fold decoding of monophone model	22
4.2	Results of second fold decoding of monophone model	23
4.3	Results of third fold decoding of monophone model	23
4.4	Results of fourth fold decoding of monophone model	24
4.5	Results of fifth fold decoding of monophone model	25
4.6	Results of sixth fold decoding of monophone model	25
4.7	Results of first fold decoding of triphone model	26
4.8	Results of second fold decoding of triphone model	27
4.9	Results of third fold decoding of triphone model	27
4.10	Results of fourth fold decoding of triphone model	28
4.11	Results of fifth fold decoding of triphone model	29

4.12	Results of sixth fold decoding of triphone model	29
4.13	Results of first fold decoding of LDA + MLLT model	30
4.14	Results of second fold decoding of LDA + MLLT model	31
4.15	Results of third fold decoding of LDA + MLLT model	32
4.16	Results of fourth fold decoding of LDA + MLLT model	33
4.17	Results of fifth fold decoding of LDA + MLLT model	33
4.18	Results of sixth fold decoding of LDA + MLLT model	34
4.19	Results of first fold decoding of LDA + MLLT + SAT model . . .	35
4.20	Results of second fold decoding of LDA + MLLT + SAT model .	36
4.21	Results of third fold decoding of SAT model	36
4.22	Results of fourth fold decoding of SAT model	37
4.23	Results of fifth fold decoding of SAT model	38
4.24	Results of sixth fold decoding of SAT model	38
4.25	Results of first fold decoding of DNN model	39
4.26	Results of second fold decoding of DNN model	40
4.27	Results of third fold decoding of DNN model	40
4.28	Results of fourth fold decoding of DNN model	41

4.29 Results of fifth fold decoding of DNN model	42
4.30 Results of sixth fold decoding of DNN model	42
4.31 Summary of WER means of acoustic models in each fold	43
4.32 WER and SER results when transcribing in Hiligaynon using dif- ferent sizes of pre-trained models trained with Whisper	44

Chapter 1

Introduction

1.1 Overview of the Current State of Technology

The technology known as Automatic Speech Recognition (ASR) enables computers to recognize spoken language and convert it into text. It is a fast developing field that might completely alter how we engage with technology. Since Hiligaynon is one of the most extensively used languages in the Philippines and is spoken in the Western Visayas area, it is a suitable topic for an ASR special problem. With over 7 million speakers, it is the third most spoken language in the Philippines, according to the Philippine Statistics Authority[1]. But since there aren't many ASR systems for Hiligaynon at the moment, creating and advancing ASR technology for this language could be very beneficial.

The Visayan language family includes Hiligaynon, which is spoken in the Philippines' central and southern regions. It is well-known for its extensive vocabulary

and intricate verbal structure, which incorporates numerous Spanish and English words. The language is used in education, the media, and government, among other places. However, due to limited availability of ASR systems for Hiligaynon, the alternatives for speech-to-text transcription and other ASR-related applications are currently constrained.

There are numerous advantages to developing an ASR system for Hiligaynon, including enhancing accessibility for Hiligaynon speakers and encouraging the language's preservation and development. Additionally, it would aid in the development of computational linguistics and ASR technology. The creation of an ASR system for Hiligaynon is, in general, a significant and pressing special issue with the potential to have a significant impact.

1.2 Problem Statement

As time has advanced, multiple ASR models have been developed, particularly for the most commonly spoken languages. Notably, ASR models have been created for languages such as English and Filipino. Considering that the Philippines is an archipelagic country, it is home to a diverse range of 187 languages based on the paper of Eberhard et al. (n.d.). The prospect of creating an ASR model for regional languages, particularly Hiligaynon, has piqued the interest of researchers. The notion of developing an ASR model specifically for a local language serves as a foundational milestone towards advancing speech-to-text innovation.

1.3 Research Objectives

1.3.1 General Objective

The aim of this project is to develop an Automatic Speech Recognition System for the Hiligaynon language using Kaldi.

1.3.2 Specific Objectives

Specifically, the project targets to:

1. Train an ASR model with Hiligaynon words using Kaldi.
2. Train different acoustic models and compare the results.
3. Evaluate the performance of the models in terms of the Word Error Rate (WER) via cross-validation.
4. Evaluate the current performance of OpenAI's Whisper when transcribing in the Hiligaynon language.
5. Construct a speech-to-text (STT) system that employs the best performing model trained using Kaldi.

1.4 Scope and Limitations of the Research

The system is specific to the Hiligaynon language. The words used in the audio data are limited to two-to-three-syllable Hiligaynon words taken from a personally

compiled collection of Hiligaynon words contributed by Francis D. Dimzon. The system is also limited to the features offered by Kaldi - an open source speech recognition toolkit used for training the models.

1.5 Significance of the Research

ASR systems offer a wide-range of benefits such as improved accessibility through the use of spoken language data. However for under resourced languages such as Hiligaynon, data used for training is limited. Focusing attention to developing ASR systems that cater to under resourced languages can help in extending these benefits to speakers of such languages. This project aims to take another step towards developing an automatic speech recognition (ASR) system for the Hiligaynon Language, adding more words to the vocabulary of the system and incorporating more speakers for variability.

Chapter 2

Review of Related Literature

2.1 Automatic Speech Recognition

ASR as we know it extends back to 1952 when the infamous Bell Labs created “Audrey,” a digit recognizer. Audrey could only transcribe spoken numbers, but a decade later, researchers improved upon Audrey so that it could transcribe rudimentary spoken words like “hello”.

For most of the past fifteen years, ASR has been powered by classical Machine Learning technologies like Hidden Markov Models. Though once the industry standard, accuracy of these classical models had plateaued in recent years, opening the door for new approaches powered by advanced Deep Learning technology that’s also been behind the progress in other fields such as self-driving cars.

Not only has accuracy skyrocketed, but access to ASR technology has also improved dramatically. Ten years ago, customers would have to engage in lengthy,

expensive enterprise software contracts to license ASR technology. Today, developers, startup companies, and Fortune 500s have access to State-of-the-Art ASR technology via simple APIs like AssemblyAI’s Speech-to-Text API.

Automatic speech recognition follows an order of combinations in predicting transcriptions. These include the lexicon model, acoustic model, language model, and decoding.

2.1.1 Lexicon Model

A lexicon model, also known as a pronunciation dictionary, maps words or units of speech to their corresponding pronunciations. For example, in English, the word “cat” might be represented with the phonemic transcription /kæt/. The lexicon model provides information about the pronunciation variants of words, including phonetic transcriptions or pronunciation rules. It serves as a reference for the acoustic and language models to accurately recognize and decode spoken words.

2.1.2 Acoustic Model

The acoustic model captures the acoustic properties of speech signals and converts them into a sequence of phonetic representations. It learns the relationship between the input audio features and the corresponding phonetic units. For example, given a speech input, the acoustic model analyzes the audio features and generates the most likely sequence of phonetic units.

2.1.3 Language Model

A language model incorporates linguistic knowledge to estimate the likelihood of word sequences in a given language. It helps distinguish between potential word sequences generated by the acoustic model. For example, given the sequence of phonetic units /k/ /æ/ /t/, the language model calculates the probability of different word sequences that could correspond to those phonetic units.

2.1.4 Decoding

Decoding is the process of determining the most likely word sequence or transcription given the outputs of the acoustic model and language model. It is a crucial step in speech recognition systems. During decoding, the system combines the information from the acoustic model, which produces a sequence of phonetic units, and the language model, which estimates the likelihood of word sequences.

2.2 The Language Dilemma

While there is a wealth of English-oriented ASR systems, other languages, especially lesser known languages tend to struggle in these situations. For instance, most leading tech companies tend to focus on developing speech recognition technologies for the English language such as DeepSpeech, Mozilla's speech to text engine and OpenAI's Whisper. In the context of Philippine languages, research in speech processing technologies for the Filipino language is not unheard of. However, developing efficient ASR systems for the such languages have yet to be seen

Dimzon & Pascual (2020) . For instance, Dimzon & Pascual (2020) was able to develop an “Automatic Phoneme Recognizer for Children’s Filipino Read Speech”. Additionally, Aquino et al. 2019 was able to develop a system using a grapheme to phoneme (G2P) approach, in conjunction with selected ASR models which have been found out to be just as effective as human transcribers. Other local languages, however, are challenged by limited resources but efforts are underway.

2.2.1 Whisper by OpenAI

Several open-source speech recognition systems have been launched which offer models trained using hundreds-to-thousands-hours-long audio data, yielding good accuracy.

OpenAI’s Whisper in particular, is trained using 680,000 hours of ”multilingual and multitask” weakly supervised audio data from the web Radford et al. (2022). Including a diverse dataset makes Whisper a robust system that is sensitive to accents and background noise and is able to support transcription to multiple languages, as well as translation from other languages to English.

What makes Whisper different from other speech recognition systems is its deviation from ”self-supervision and self-training techniques” typical of large-scale speech recognition systems. This makes Whisper efficient at dealing with large volumes of data.

Currently, the only Philippine language mentioned in Whisper’s documentation is Tagalog, where the system performs with a 13.8% word error rate (WER) with the Fleurs dataset using the large-v2 model Radford et al.. It was also noted

that Whisper’s performance is proportional to the amount of data trained on a particular language, leaving under-resourced languages at a disadvantage.

2.3 Hiligaynon Speech Recognition

Billones & Dadios (2014) conducted a study, where they created a 5-word vocabulary speech recognition system for Hiligaynon terms used as motion commands implemented for a breast self-examination (BSE) multimedia training system. The study aimed at raising awareness about breast cancer among the local female population of Western Visayas. Their focus was on developing a system that utilized Hiligaynon speech recognition with a limited vocabulary of five words. The researchers selected five commonly used Hiligaynon words (“*idalom*,” “*ibabaw*,” “*wala*,” “*tuo*,” and “*patiyog*”) as representative chromosomes for the system. They collected a total of 200 audio samples by recording 40 samples for each word. The system employed Mel frequency cepstrum coefficients (MFCC) for feature extraction and genetic algorithms for pattern recognition. Additionally, an adaptive database was integrated into the system to enhance training and classification accuracy for the Hiligaynon words. Through the combination of these models and the adaptive database, the system achieved an impressive accuracy rate of 97.50% in recognizing the distinct Hiligaynon words.

2.4 The Hiligaynon Language

Table 2.1: Table of the 25 significant sounds in Hiligaynon (Wolfenden, 2019)

Vowels	Consonants
i	p
e	b
a	t
o	d
u	k
	g
	c
	j
	f
	v
	s
	h
	m
	n
	ng
	l
	r
	w
	y
Stress (not symbolized)	

Hiligaynon, also known as Ilonggo, is an Austronesian language spoken in the Western Visayas region of the Philippines, particularly in the provinces of Iloilo, Guimaras, Negros Occidental, and Capiz. It is one of the major languages of the Philippines, spoken by millions of people as a first or second language.

Hiligaynon has a rich and varied vocabulary, with many loanwords from Spanish, English, and other languages. According to Hiligaynon Reference Grammar by Wolfenden 2019, Hiligaynon has a complex verb conjugation and tense system, with a range of tense markers including markers for past, present, and future tense, as well as markers for perfective and imperfective aspect. The book also notes that Hiligaynon has a number of mood markers, including markers for indicative, imperative, and subjunctive mood.

Additionally, Hiligaynon Reference Grammar by Wolfenden 2019 describes the phonemic alphabet of Hiligaynon as consisting of 28 letters: A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R, S, T, U, V, W, X, Y, and Z. The book notes that the letters C, F, J, Q, V, X, and Z are not used as frequently in Hiligaynon as in other Philippine languages, and that the letter Ñ is used to represent the Spanish sound "ny."

2.5 Kaldi ASR Toolkit

Povey et al. 2011 described Kaldi as a modern toolkit for speech recognition. It is designed to be extensible and has one of the least restrictive licenses making it more accessible. Several studies have incorporated Kaldi into their implementations.

For instance, Upadhyaya et al. 2017 , developed a continuous Hindi speech recognition model using Kaldi, citing the toolkit for its ability to create high quality lattices and sufficient speed for real time recognition. It also said that the mentioned toolkit is actively maintained and accessible.

Additionally, not only is Kaldi able to support conventional models such as Gaussian mixture models (GMMs) but is also able to implement deep neural network based structures. For example, Kipyatkova & Karpov 2016 developed a “DNN-Based Acoustic Modeling for Russian Speech Recognition Using Kaldi.” The paper mentioned using DNN implementations in Kaldi, ultimately choosing Dan’s implementation because of its support for parallel training on multiple CPUs.

2.6 Acoustic Models in Kaldi

In the realm of speech recognition research, the Kaldi toolkit has gained prominence as an open-source framework offering a diverse array of acoustic models. These models play a pivotal role in effectively capturing and representing speech data, thereby contributing to advancements in automatic speech recognition research. Notably, the following acoustic models are widely employed in Kaldi:

2.6.1 Monophone Model

The monophone model represents each phoneme individually, serving as an initial building block for speech recognition systems. For example, in the word ‘cat,’ the monophone model would have separate models for the /k/, /æ/, and /t/

phonemes.

2.6.2 Triphone Model

The triphone model enhances the representation of phonemes by incorporating context-dependent information. It captures variations in phoneme sounds based on neighboring phonemes. For instance, in the word 'cat,' the triphone model would consider the context of the previous and next phonemes, such as /k/ in the context of /æ/ and following /t/.

2.6.3 LDA + MLLT Model

The LDA + MLLT model combines Linear Discriminant Analysis (LDA) with Maximum Likelihood Linear Transformations (MLLT). LDA reduces the dimensionality of acoustic features and provides a discriminative representation. MLLT then refines this representation to account for speaker and channel variations. This can be visualized as a transformation of the acoustic feature space, aligning similar speech patterns while differentiating between different speakers.

2.6.4 LDA + MLLT + SAT Model

The SAT (Speaker Adaptive Training) model extends the capabilities of the triphone model by incorporating speaker adaptation techniques. It accommodates individual speaker characteristics by adapting the model parameters to match the characteristics of the target speaker. This adaptation can be imagined as adjust-

ing the model to fit the specific speaker’s speech patterns, reducing inter-speaker variability.

2.6.5 DNN Model

The DNN (Deep Neural Network) model employs deep neural networks, such as feed-forward or recurrent neural networks, to capture intricate representations of acoustic features. With multiple hidden layers, these models can learn complex patterns and relationships within the input data. The DNN model can be depicted as a series of interconnected nodes, simulating the human brain’s neural connections.

Chapter 3

Research Methodology

This chapter presents the methodology used to develop an automatic speech recognition system for the Hiligaynon language. This chapter is divided into the following major parts: Data Gathering and Preprocessing, Training, Evaluation and System Development.

3.1 Research Activities

3.1.1 Data Gathering and Preprocessing

Word Selection and Dictionary Creation

Initially, a set of one thousand words from the corpus were considered for recording. The selection was limited to two-to-three-syllable words only. From this set, a script was created for each speaker to read. Each script contained 500 words

which were randomly chosen from the previous 1000-word dictionary.

The same words were also included in the lexicon along with their phonetic transcriptions which were manually transcribed. Transcription was based off of the phonemes described in Wolfenden's Hiligaynon Reference Grammar.

Later, another set of words were added to the lexicon along with their phonetic transcriptions which were also manually transcribed. However, these words were not considered for recording. All in all, a total of about 3500 words were included in the project's lexicon.

Selection of Speakers and Equipment Used

A total of 18 speakers were gathered for audio recording. Nine of the speakers were male and the other nine were female. The speakers were either native speakers of Hiligaynon or are fluent in the said language. For recording, an external microphone was utilized alongside Audacity, a free and open-source digital audio editor.

Recording

Recording sessions were done in areas with minimal background noise. Each of the speakers uttered 25 words per recording. A total of 20 audio files were produced from each speaker. A naming system was developed for each audio file that identifies the gender of the speaker, the speaker number, and the audio file number in the following format: <gender>_<speaker-number>_<file-number >. The male gender was represented by a 1 and the number 2 was used to represent

the female gender.

3.1.2 Preprocessing

Audacity was used for preprocessing the audio files. Normalization was first applied, followed by compression. After which, noise reduction was applied. The audio files were then exported using the WAV format.

3.1.3 Training

Kaldi, an open-source speech recognition toolkit was used for training the ASR models.

A six-fold cross validation scheme was followed for training. Each fold contains data from three speakers. For each iteration, one fold was set for use as testing data while the remaining folds were set for use as training data. A total of 6 iterations were executed, with each fold acting as testing data exactly once.

For each iteration, a set of metadata files were created. These files contain information specific to the audio data within the context of each iteration. These files can be divided into two types:

Acoustic Data

These files contains information related to each audio file/data.

- **wav.scp:** This file contains information that maps a file id to the location

of the file. Each line in this file is written in the following format: `<file_id ><path_to_file >`.

- **text:** This file contains information about the file and the corresponding words uttered in that particular audio file. It is written in the following format `<utterance_id> word1 word2 word3...`
- **utt2spk:** This file contains information about the mapping of a specific file to it's corresponding speaker. It is written in the following format.
`<utterance_id><speaker_id>`.
- **spk2gender:** This file contains information indicating a specific speaker's gender. It is written in the following format `<speaker_id ><gender >`.
- **corpus.txt:** This file contains all the words uttered in all of the recordings. Each line represents the words uttered in a specific file.

Language Data

These files contain information which were used for language modeling.

- **lexicon.txt:** This file contains information about all the words considered in the project's dictionary together with their phonemic transcriptions. Silence phones are also included in the lexicon.
- **nonsilence_phones:** This file contains all of the non-silent phones included in the project.
- **silence_phones.txt and optional_silence.txt:** This files contains the silence phones included in the project.

Training scripts were sourced from Kaldi's builtin scripts for different acoustic models namely: monophone, triphone, LDA+MLLT, LDA+MLLT+SAT and DNN.

3.1.4 Evaluation

Evaluation for DNN model trained using Kaldi

Performance of the models were measured in terms of WER which was done by getting the average WER for each iteration of the training.

Evaluation of transcriptions using Whisper

The same dataset used in this project was used for transcription using Whisper. Transcription was done using pre-trained models of the following sizes: tiny, base and small.

The generated transcriptions were then formatted in the same format as Kaldi's transcriptions, specifically in the following format: `{file_id} word1 word2 word3...` for compatibility purposes.

The formatted reference transcription was then compared with the reference transcription for the calculation of WER using Kaldi's computer-wer tool.

3.1.5 System Development

A speech-to-text system with a command line interface was developed using the best-performing DNN model during training.

The system can be called with or without an argument. When called without an argument, the system starts recording. To stop the recording, the user is prompted to press q.

After recording, the system proceeds to decoding. The resulting transcriptions are then printed on the terminal. (see figure A.1)

The system also accepts one argument, which is the file name of the audio to be transcribed. Given this argument, the system proceeds to decoding and then it also prints the resulting transcriptions on the terminal. (see figure A.2)

Chapter 4

Results and Discussions/Analyses

This chapter provides a summary and analysis of the results obtained by decoding different training models, such as monophones, triphones, LDA + MLLT, SAT, and DNN. The information presented includes data extracted from each decoding model's WER files output, including the number of errors made (insertions, deletions, and substitutions), as well as the word and sentence error rates (WER and SER), and the average percentage of errors for each model.

The same information is also presented but for the evaluation of the transcriptions generated by Whisper using the same audio dataset used in the project.

To ensure dependable outcomes, this project underwent rigorous testing using a 6-fold cross-validation technique for each training model. The project involved a total of eighteen speakers, with three speakers assigned for testing in each fold, while the remaining speakers were used for training. The subsequent tables will display the outcomes obtained from this procedure.

4.1 Results of Monophone model

	WER	SER	Insert	Delete	Substitution
wer_7	9.07	81.67	5	20	111
wer_8	7.87	78.33	5	20	93
wer_9	6.87	75.00	5	22	76
wer_10	6.53	73.33	4	25	69
wer_11	6.27	71.67	4	25	65
wer_12	5.73	66.67	4	30	52
wer_13	5.67	65.00	4	32	49
wer_14	5.60	65.00	3	34	47
wer_15	5.73	63.33	3	38	45
wer_16	5.93	63.33	3	43	43
wer_17	5.60	63.33	3	43	38
MEAN	6.35	69.70			

Table 4.1: Results of first fold decoding of monophone model

Table 4.1 presents data for the first fold monophone model after acoustic model training. WER ranges from wer_7 to wer_17, with an average of 6.35%, highest at 9.07%, and lowest at 5.60%. The average sentence error rate is 69.70%, with the highest at 81.67% and lowest at 63.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	3.07	48.33	1	0	45
wer_8	2.47	41.67	1	0	36
wer_9	2.13	36.67	0	0	32
wer_10	1.73	35.00	0	0	26
wer_11	1.67	33.33	0	0	25
wer_12	1.67	33.33	0	0	25
wer_13	1.67	33.33	0	0	24
wer_14	1.60	33.33	0	0	24
wer_15	1.53	31.67	0	0	23
wer_16	1.53	31.67	0	0	23
wer_17	1.53	31.67	0	0	23
MEAN	1.87	35.45			

Table 4.2: Results of second fold decoding of monophone model

Table 4.2 displays the data for the second fold monophone model after acoustic model training. The WER ranges from wer_7 to wer_17, with an average of 1.87%, highest at 3.07%, and lowest at 1.53%. The Sentence Error Rate (SER) has an average of 35.45%, with the highest and lowest scores also at 3.07% and 1.53%, respectively.

	WER	SER	Insert	Delete	Substitution
wer_7	0.80	20.00	1	0	11
wer_8	0.67	16.67	1	0	9
wer_9	0.67	16.67	1	0	9
wer_10	0.67	16.67	1	0	9
wer_11	0.67	16.67	1	0	9
wer_12	0.67	16.67	1	0	9
wer_13	0.67	16.67	1	0	9
wer_14	0.67	16.67	1	0	9
wer_15	0.67	16.67	1	0	9
wer_16	0.67	16.67	1	0	9
wer_17	0.60	15.00	1	0	8
MEAN	0.68	16.82			

Table 4.3: Results of third fold decoding of monophone model

Table 4.3 presents the data for the third fold monophone model after acoustic model training. The WER ranges from wer_7 to wer_17, with an average of 0.68%, highest at 0.80%, and lowest at 0.60%. The average sentence error rate is 16.82%, with the highest score at 20.00% and the lowest at 15.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	1.00	18.33	3	2	10
wer_8	0.80	16.67	3	2	7
wer_9	0.67	13.33	3	2	5
wer_10	0.60	11.67	3	2	4
wer_11	0.60	11.67	3	2	4
wer_12	0.53	11.67	3	2	3
wer_13	0.53	11.67	3	2	3
wer_14	0.53	11.67	3	2	3
wer_15	0.53	11.67	3	2	3
wer_16	0.53	11.67	3	2	3
wer_17	0.53	11.67	3	2	3
MEAN	0.62	12.88			

Table 4.4: Results of fourth fold decoding of monophone model

Table 4.4 showcases the data for the fourth fold monophone model after acoustic model training. The WER ranges from wer_7 to wer_17, with an average of 0.62%, highest at 1.00%, and lowest at 0.53%. The average SER is 12.88%, with the highest score at 18.33% and the lowest at 11.67%.

	WER	SER	Insert	Delete	Substitution
wer_7	8.53	51.67	5	7	116
wer_8	7.40	40.00	5	7	99
wer_9	6.53	36.67	4	8	86
wer_10	5.33	33.33	2	9	69
wer_11	4.60	28.33	2	9	58
wer_12	4.27	26.67	1	9	54
wer_13	3.73	26.67	1	9	46
wer_14	3.47	23.33	1	9	42
wer_15	3.47	23.33	1	9	42
wer_16	3.40	23.33	1	9	41
wer_17	3.20	23.33	1	10	37
MEAN	4.90	30.61			

Table 4.5: Results of fifth fold decoding of monophone model

Table 4.5 has the data gathered for the fifth fold monophone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean of the word error rate is 4.90%. The highest word error rate score is 8.53% and the lowest is 3.20%. For the sentence error rate the mean is 30.61%. The highest sentence error rate score is 51.67% while the lowest is 23.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.67	13.33	3	0	7
wer_8	0.67	13.33	3	0	7
wer_9	0.47	10.00	1	0	6
wer_10	0.47	10.00	1	0	6
wer_11	0.47	10.00	1	0	6
wer_12	0.40	8.33	1	0	5
wer_13	0.33	6.67	1	0	4
wer_14	0.33	6.67	1	0	4
wer_15	0.33	6.67	1	0	4
wer_16	0.33	6.67	1	0	4
wer_17	0.33	6.67	1	0	4
MEAN	0.44	8.94			

Table 4.6: Results of sixth fold decoding of monophone model

Table 4.6 has the data gathered for the sixth fold monophone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean of the word error rate is 0.44%. The highest word error rate is 0.67% and the lowest is 0.33%. For the sentence error rate, the mean is 8.94%. The highest sentence error rate is 13.33% while the lowest is 6.67%.

4.2 Results of Triphone model

	WER	SER	Insert	Delete	Substitution
wer_7	2.40	35.00	2	0	34
wer_8	1.87	31.67	2	0	26
wer_9	1.73	31.67	1	0	25
wer_10	1.67	30.00	1	0	24
wer_11	1.47	26.67	1	0	21
wer_12	1.33	23.33	1	0	19
wer_13	1.20	21.67	1	0	17
wer_14	1.07	20.00	1	0	15
wer_15	1.07	20.00	1	0	15
wer_16	1.07	20.00	1	0	15
wer_17	1.07	20.00	1	0	15
MEAN	3.09	25.46			

Table 4.7: Results of first fold decoding of triphone model

Table 4.7 has the data gathered for the first fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean of the word error rate is 3.09%. The highest word error rate is 2.40% and the lowest is 1.07%. For the sentence error rate, the mean is 25.46%. The highest sentence error rate is 35.00% while the lowest is 20.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	5.07	63.33	4	0	72
wer_8	4.40	60.00	2	0	64
wer_9	4.13	56.67	2	0	60
wer_10	3.87	56.67	1	0	57
wer_11	3.80	55.00	1	0	56
wer_12	3.27	53.33	0	0	49
wer_13	3.07	51.67	0	0	46
wer_14	3.07	51.67	0	0	46
wer_15	2.87	50.00	0	0	43
wer_16	2.87	50.00	0	0	43
wer_17	2.73	46.67	0	0	41
MEAN	3.56	54.46			

Table 4.8: Results of second fold decoding of triphone model

Table 4.8 has the data gathered for the second fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean of the word error rate is 3.56%. The highest word error rate is 5.07% and the lowest is 2.73%. For the sentence error, the mean is 54.46. The highest sentence error rate is 63.33% while the lowest is 46.67%.

	WER	SER	Insert	Delete	Substitution
wer_7	2.40	35.00	2	0	34
wer_8	1.87	31.67	2	0	26
wer_9	1.73	31.67	1	0	25
wer_10	1.67	30.00	1	0	24
wer_11	1.47	26.67	1	0	21
wer_12	1.33	23.33	1	0	19
wer_13	1.20	21.67	1	0	17
wer_14	1.07	20.00	1	0	15
wer_15	1.07	20.00	1	0	15
wer_16	1.07	20.00	1	0	15
wer_17	1.07	20.00	1	0	15
MEAN	1.45	25.46			

Table 4.9: Results of third fold decoding of triphone model

Table 4.9 has the data gathered for the third fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 1.45%. The highest word error rate is 2.40% and the lowest is 1.07%. For the sentence error rate, the mean is 25.46. The highest sentence error rate is 35.00% while the lowest is 20.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	1.13	21.67	5	2	10
wer_8	1.00	20.00	5	2	8
wer_9	0.87	18.33	5	2	6
wer_10	0.80	16.67	4	2	6
wer_11	0.73	16.67	4	2	5
wer_12	0.73	16.67	4	2	5
wer_13	0.73	16.67	4	2	5
wer_14	0.67	15.00	4	2	4
wer_15	0.67	15.00	4	2	4
wer_16	0.67	15.00	4	2	4
wer_17	0.67	15.00	4	2	4
MEAN	0.79	16.97			

Table 4.10: Results of fourth fold decoding of triphone model

Table 4.10 has the data gathered for the fourth fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.79%. The highest word error rate is 1.13% and the lowest is 0.67%. For the sentence error rate, the mean is 16.97. The highest sentence error rate is 21.67% while the lowest is 15.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	14.20	63.33	11	31	171
wer_8	13.40	56.67	8	31	162
wer_9	12.67	55.00	5	34	151
wer_10	12.00	51.67	1	36	143
wer_11	11.93	51.67	1	37	141
wer_12	11.93	50.00	1	41	137
wer_13	11.73	48.33	1	48	127
wer_14	11.40	46.67	1	49	121
wer_15	11.27	45.00	1	50	118
wer_16	11.27	45.00	1	52	116
wer_17	11.20	43.33	1	56	111
MEAN	12.09	50.60			

Table 4.11: Results of fifth fold decoding of triphone model

Table 4.11 has the data gathered for the fifth fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean error rate is 12.09%. The highest word error rate is 14.20% and the lowest is 11.20%. For the sentence error rate, the mean is 50.60%. the highest sentence error rate is 63.33% while the lowest is 43.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.67	13.33	3	0	7
wer_8	0.67	13.33	3	0	7
wer_9	0.67	13.33	3	0	7
wer_10	0.47	10.00	1	0	6
wer_11	0.47	10.00	1	0	6
wer_12	0.40	8.33	1	0	5
wer_13	0.33	6.67	1	0	4
wer_14	0.33	6.67	1	0	4
wer_15	0.33	6.67	1	0	4
wer_16	0.33	6.67	1	0	4
wer_17	0.33	6.67	1	0	4
MEAN	0.45	0.24			

Table 4.12: Results of sixth fold decoding of triphone model

Table 4.12 has the data gathered for the sixth fold triphone model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.45%. The highest word error rate is 0.67% and the lowest is 0.33%. For the sentence error rate, the mean is 0.24%. the highest sentence error rate is 13.33% while the lowest is 6.67%.

4.3 Result of LDA + MLLT model

	WER	SER	Insert	Delete	Substitution
wer_7	52.53	100.00	1	661	126
wer_8	52.87	100.00	1	680	112
wer_9	53.40	100.00	0	702	99
wer_10	54.27	100.00	0	724	90
wer_11	56.13	100.00	0	760	82
wer_12	57.20	100.00	0	786	72
wer_13	58.53	100.00	0	813	65
wer_14	59.80	100.00	0	839	58
wer_15	60.60	100.00	0	862	47
wer_16	62.07	100.00	0	887	44
wer_17	62.53	100.00	0	896	42
MEAN	57.27	100.00			

Table 4.13: Results of first fold decoding of LDA + MLLT model

Table 4.13 has the data gathered for the first fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 57.27%. The highest word error rate is 62.53% and the lowest is 52.53%. For the sentence error rate, the mean is 100 since all values are 100.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	6.47	65.00	2	4	91
wer_8	6.13	65.00	2	4	86
wer_9	5.67	65.00	2	4	79
wer_10	5.53	61.67	2	4	77
wer_11	4.80	56.67	2	3	67
wer_12	4.47	56.67	1	3	63
wer_13	4.33	55.00	1	3	61
wer_14	4.07	55.00	0	4	57
wer_15	3.87	51.67	0	5	53
wer_16	3.80	50.00	0	5	52
wer_17	3.60	48.33	0	5	49
MEAN	4.79	48.33			

Table 4.14: Results of second fold decoding of LDA + MLLT model

Table 4.14 has the data gathered for the second fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 10.15%. The highest word error rate is 62.53% and the lowest is 52.53%. For the sentence error rate, the mean error rate is 61.97. the highest sentence error rate is 65.00% and the lowest is %.

	WER	SER	Insert	Delete	Substitution
wer_7	2.73	35.00	1	12	28
wer_8	2.67	33.33	1	13	26
wer_9	2.33	30.00	1	13	21
wer_10	2.20	30.00	1	13	19
wer_11	2.13	28.33	1	12	19
wer_12	2.13	28.33	1	12	19
wer_13	2.13	28.33	1	12	19
wer_14	2.13	28.33	1	12	19
wer_15	2.07	28.33	1	12	18
wer_16	1.93	28.33	1	12	16
wer_17	2.00	28.33	1	14	15
MEAN	2.21	29.69			

Table 4.15: Results of third fold decoding of LDA + MLLT model

Table 4.15 has the data gathered for the third fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean error rate is 2.21%. The highest word error rate is 2.73% and the lowest is 1.93%. For the sentence error rate, the mean is 29.69. The highest value is 35.00% and the lowest value is 28.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.93	21.67	4	2	8
wer_8	0.87	20.00	4	2	7
wer_9	0.87	20.00	4	2	7
wer_10	0.80	18.33	4	2	6
wer_11	0.73	18.33	4	2	5
wer_12	0.67	16.67	4	2	4
wer_13	0.67	16.67	4	2	4
wer_14	0.60	15.00	4	2	3
wer_15	0.60	15.00	4	2	3
wer_16	0.53	13.33	4	2	2
wer_17	0.53	13.33	4	2	2
MEAN	0.71	17.12			

Table 4.16: Results of fourth fold decoding of LDA + MLLT model

Table 4.16 has the data gathered for the fourth fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.71%. The highest word error rate is 0.93% and the lowest is 0.53%. For the sentence error rate, the mean is 17.12%. The highest is 21.67% and the lowest is 13.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	16.73	60.00	0	160	91
wer_8	16.73	58.33	0	165	86
wer_9	16.60	56.67	0	169	80
wer_10	16.47	56.67	0	169	78
wer_11	16.27	55.00	0	173	71
wer_12	16.27	53.33	0	174	70
wer_13	16.20	51.67	0	175	68
wer_14	16.13	50.00	0	178	64
wer_15	16.07	48.33	0	178	63
wer_16	15.93	46.67	0	179	60
wer_17	15.93	46.67	0	181	58
MEAN	16.30	53.03			

Table 4.17: Results of fifth fold decoding of LDA + MLLT model

Table 4.17 has the data gathered for the fifth fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 16.30%. The highest word error rate is 16.73% and the lowest is 15.93%. For the sentence error rate, the mean is 53.03%. The highest value is 60.00% and the lowest is 46.67%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.67	13.33	3	0	7
wer_8	0.67	13.33	3	0	7
wer_9	0.67	13.33	3	0	7
wer_10	0.47	10.00	1	0	6
wer_11	0.47	10.00	1	0	6
wer_12	0.40	8.33	1	0	5
wer_13	0.33	6.67	1	0	4
wer_14	0.33	6.67	1	0	4
wer_15	0.33	6.67	1	0	4
wer_16	0.33	6.67	1	0	4
wer_17	0.33	6.67	1	0	4
MEAN	0.45	9.24			

Table 4.18: Results of sixth fold decoding of LDA + MLLT model

Table 4.18 has the data gathered for the sixth fold LDA + MLLT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.45%. The highest word error rate is 0.67% and the lowest is 0.33%. For the sentence error rate, the mean is 9.24%. The highest value is 13.33% and the lowest is 6.67%.

4.4 Result of LDA + MLLT + SAT model

	WER	SER	Insert	Delete	Substitution
wer_7	2.87	48.33	6	0	37
wer_8	2.67	46.67	6	0	34
wer_9	2.67	46.67	6	0	33
wer_10	2.60	46.67	6	0	33
wer_11	2.27	45.00	5	0	29
wer_12	2.27	45.00	5	0	29
wer_13	2.27	45.00	5	0	29
wer_14	2.27	45.00	5	0	29
wer_15	2.20	45.00	5	0	28
wer_16	2.20	45.00	5	0	28
wer_17	2.13	43.33	5	0	27
MEAN	2.40	45.61			

Table 4.19: Results of first fold decoding of LDA + MLLT + SAT model

Table 4.19 has the data gathered for the first fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 2.40%. The highest word error rate is 2.87% and the lowest is 2.13%. For the sentence error rate, the mean is 45.61%. The highest value is 48.33% and the lowest is 43.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	2.00	40.00	0	0	30
wer_8	1.93	38.33	0	0	29
wer_9	1.87	38.33	0	0	28
wer_10	1.80	36.67	0	0	27
wer_11	1.67	33.33	0	0	25
wer_12	1.67	33.33	0	0	25
wer_13	1.67	33.33	0	0	25
wer_14	1.67	33.33	0	0	25
wer_15	1.60	33.33	0	0	24
wer_16	1.53	33.33	0	0	23
wer_17	1.47	31.67	0	0	22
MEAN	1.72	35.00			

Table 4.20: Results of second fold decoding of LDA + MLLT + SAT model

Table 4.20 has the data gathered for the second fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 1.72%. highest word error rate is 2.00% and the lowest is 1.47%. For the sentence error rate, the mean is 35.00%. The highest value is 40.00% and the lowest is 31.67%.

	WER	SER	Insert	Delete	Substitution
wer_7	1.80	21.67	1	12	14
wer_8	1.73	20.00	1	12	13
wer_9	1.73	20.00	1	12	13
wer_10	1.73	20.00	1	12	13
wer_11	1.73	20.00	1	12	13
wer_12	1.73	20.00	1	12	13
wer_13	1.67	18.33	1	12	12
wer_14	1.67	18.33	1	12	12
wer_15	1.67	18.33	1	12	12
wer_16	1.67	18.33	1	12	12
wer_17	1.67	18.33	1	12	12
MEAN	1.71	19.39			

Table 4.21: Results of third fold decoding of SAT model

Table 4.21 has the data gathered for the third fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 1.71%. The highest word error rate is 1.80% and the lowest is 1.67%. For the sentence error rate, the mean is 19.39%. The highest value is 21.67% and the lowest is 18.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.80	16.67	4	3	5
wer_8	0.80	16.67	4	3	5
wer_9	0.80	16.67	4	3	5
wer_10	0.67	16.67	4	3	3
wer_11	0.60	15.00	4	3	2
wer_12	0.60	15.00	4	3	2
wer_13	0.60	15.00	4	3	2
wer_14	0.60	15.00	4	3	2
wer_15	0.60	15.00	4	3	2
wer_16	0.60	15.00	4	3	2
wer_17	0.60	15.00	4	3	2
MEAN	0.66	15.61			

Table 4.22: Results of fourth fold decoding of SAT model

Table 4.22 has the data gathered for the fourth fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.66%. The highest word error rate is 0.80% and the lowest is 0.60%. For the sentence error rate, the mean is 15.61%. The highest value is 16.67% and the lowest is 15.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	12.73	40.00	2	133	56
wer_8	12.67	40.00	2	134	54
wer_9	12.73	38.33	2	137	52
wer_10	12.67	38.33	2	138	50
wer_11	12.60	38.33	2	139	48
wer_12	12.60	36.67	1	145	43
wer_13	12.60	36.67	1	148	40
wer_14	12.80	35.00	1	154	37
wer_15	12.93	35.00	1	157	36
wer_16	12.93	35.00	1	158	35
wer_17	13.00	35.00	1	159	35
MEAN	12.75	37.12			

Table 4.23: Results of fifth fold decoding of SAT model

Table 4.23 has the data gathered for the fifth fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 12.75%. The highest word error rate is 13.00% and the lowest is 12.73%. For the sentence error rate, the mean is 37.12%. The highest value is 40.00% and the lowest is 35.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	1.33	21.67	4	0	16
wer_8	1.20	20.00	4	0	14
wer_9	1.20	20.00	4	0	14
wer_10	1.07	20.00	2	0	14
wer_11	0.93	16.67	2	0	12
wer_12	0.93	16.67	2	0	12
wer_13	0.93	16.67	2	0	12
wer_14	0.87	16.67	2	0	11
wer_15	0.87	16.67	2	0	11
wer_16	0.80	15.00	2	0	10
wer_17	0.73	13.33	2	0	9
MEAN	0.99	17.58			

Table 4.24: Results of sixth fold decoding of SAT model

Table 4.24 has the data gathered for the sixth fold LDA + MLLT + SAT model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.99%. The highest word error rate is 1.33% and the lowest is 0.73%. For the sentence error rate, the mean is 17.58. The highest value is 21.67% and the lowest is 13.33%.

4.5 Result of DNN model

	WER	SER	Insert	Delete	Substitution
wer_7	3.40	56.67	6	0	45
wer_8	3.07	51.67	5	0	41
wer_9	2.80	51.67	5	0	37
wer_10	2.67	50.00	4	0	36
wer_11	2.13	38.33	4	0	28
wer_12	2.13	38.33	4	0	28
wer_13	1.93	36.67	4	0	25
wer_14	1.93	36.67	4	0	25
wer_15	1.73	35.00	4	0	22
wer_16	1.73	35.00	4	0	22
wer_17	1.73	35.00	4	0	22
MEAN	2.30	42.27			

Table 4.25: Results of first fold decoding of DNN model

Table 4.25 has the data gathered for the first fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 2.30%. The highest word error rate is 3.40% and the lowest is 1.73%. For the sentence error rate, the mean is 42.27%. The highest value is 56.67% and the lowest is 35.00%.

	WER	SER	Insert	Delete	Substitution
wer_7	2.40	41.67	1	0	35
wer_8	2.20	38.33	1	0	32
wer_9	2.13	38.33	1	0	31
wer_10	2.00	38.33	0	0	30
wer_11	1.93	38.33	0	0	29
wer_12	1.87	38.33	0	0	28
wer_13	1.73	35.00	0	0	26
wer_14	1.67	35.00	0	0	25
wer_15	1.67	35.00	0	0	25
wer_16	1.67	35.00	0	0	25
wer_17	1.60	33.33	0	0	24
MEAN	1.90	36.97			

Table 4.26: Results of second fold decoding of DNN model

Table 4.26 has the data gathered for the second fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 1.90%. The highest word error rate is 2.40% and the lowest is 1.60%. For the sentence error rate, the mean is 36.97%. the highest value is 41.67% and the lowest is 33.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.73	18.33	1	0	10
wer_8	0.73	18.33	1	0	10
wer_9	0.73	18.33	1	0	10
wer_10	0.73	18.33	1	0	10
wer_11	0.73	18.33	1	0	10
wer_12	0.73	18.33	1	0	10
wer_13	0.73	18.33	1	0	10
wer_14	0.73	18.33	1	0	10
wer_15	0.73	18.33	1	0	10
wer_16	0.73	18.33	1	0	10
wer_17	0.73	18.33	1	0	10
MEAN	0.73	18.33			

Table 4.27: Results of third fold decoding of DNN model

Table 4.27 has the data gathered for the third fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. all values for the word error rate is 0.73%. For the sentence error rate all the values are 18.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.47	11.67	3	2	2
wer_8	0.47	11.67	3	2	2
wer_9	0.47	11.67	3	2	2
wer_10	0.47	11.67	3	2	2
wer_11	0.47	11.67	3	2	2
wer_12	0.40	10.00	3	2	1
wer_13	0.33	8.33	3	2	0
wer_14	0.33	8.33	3	2	0
wer_15	0.33	8.33	3	2	0
wer_16	0.33	8.33	3	2	0
wer_17	0.33	8.33	3	2	0
MEAN	0.40	10.00			

Table 4.28: Results of fourth fold decoding of DNN model

Table 4.28 has the data gathered for the fourth fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.40%. The highest word error rate is 0.47% and the lowest is 0.33%. For the sentence error rate, the mean is 10.00%. The highest value is 11.67% and the lowest is 8.33%.

	WER	SER	Insert	Delete	Substitution
wer_7	11.20	36.67	24	6	138
wer_8	10.47	36.67	19	6	132
wer_9	9.60	33.33	13	6	125
wer_10	9.27	33.33	11	6	122
wer_11	8.33	31.67	9	6	110
wer_12	7.60	30.00	7	6	101
wer_13	7.47	30.00	7	6	99
wer_14	7.33	28.33	7	7	96
wer_15	7.00	26.67	6	9	90
wer_16	6.47	26.67	5	9	83
wer_17	6.47	26.67	5	9	83
MEAN	8.29	30.00			

Table 4.29: Results of fifth fold decoding of DNN model

Table 4.29 has the data gathered for the fifth fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 8.29%. The highest word error rate is 11.20% and the lowest is 6.47%. For the sentence error rate, the mean is 28.18%. The highest value is 36.67% and the lowest is 26.67%.

	WER	SER	Insert	Delete	Substitution
wer_7	0.93	20.00	4	0	10
wer_8	0.73	15.00	3	0	8
wer_9	0.67	13.33	3	0	7
wer_10	0.60	11.67	2	0	7
wer_11	0.53	10.00	2	0	6
wer_12	0.33	5.00	1	0	4
wer_13	0.33	5.00	1	0	4
wer_14	0.27	3.33	1	0	3
wer_15	0.20	3.33	0	0	3
wer_16	0.20	3.33	0	0	3
wer_17	0.20	3.33	0	0	3
MEAN	0.45	8.48			

Table 4.30: Results of sixth fold decoding of DNN model

Table 4.30 has the data gathered for the sixth fold DNN model after training the acoustic model. Word and sentence errors start from wer_7 to wer_17. The mean word error rate is 0.45%. The highest word error rate is 0.93% and the lowest is 0.20%. For the sentence error rate, the mean is 8.48%. The highest value is 20.00% and the lowest is 3.33%.

4.6 Summary of results

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Mean
Monophone	6.35	1.87	0.68	0.62	4.90	0.44	2.48
Triphone	3.09	3.56	1.45	0.79	12.09	0.45	3.57
LDA + MLLT	57.27	4.79	2.21	0.71	16.30	0.45	13.62
SAT	2.40	1.72	1.71	0.66	12.75	0.99	3.37
DNN	2.30	1.90	0.73	0.40	8.29	0.45	2.35

Table 4.31: Summary of WER means of acoustic models in each fold

Table 4.31 displays the compiled mean word error rates for each acoustic model across the different folds. The monophone model achieved the lowest word error rate of 0.44% on the sixth fold. Similarly, the triphone model obtained a word error rate of 0.45% on the sixth fold. The LDA + MLLT model exhibited optimal performance on the sixth fold, attaining a word error rate of 0.45%. Conversely, the SAT model yielded the lowest word error rate of 0.66% on the fourth fold. Notably, the DNN model showcased exceptional efficacy with the dataset, achieving a word error rate of 0.40% on the fourth fold and yielding the lowest mean of 2.35% among all the means obtained from different acoustic models.

Each model demonstrated satisfactory word error rates across different folds. Among these models, the Deep Neural Network or DNN model stood out with the

lowest word error rate of 0.40% on the fourth fold. Consequently, this model was selected for the development of an automatic speech recognition system designed for the Hiligaynon language.

4.7 Results on the performance of Whisper

Model size	WER	SER	Insertion	Deletion	Substitution
tiny	62.11	100.00	631	63	4896
base	49.02	100.00	429	19	3964
small	25.50	99.72	86	13	2196

Table 4.32: WER and SER results when transcribing in Hiligaynon using different sizes of pre-trained models trained with Whisper

It is seen that the performance of the transcription is proportional to the size of the pre-trained model used. The small model performed best with a WER of 25.50% and SER of 99.72%.

It is important to note that the language specified for decoding was Tagalog since it is the closest language available in the Whisper system. However, not specifying the language does lead Whisper to detect the language as Tagalog, based on the text. Moreover, Wolfenden (2019) mentions similarities between the phonology of Tagalog and Hiligaynon, hence the researchers found it reasonable to proceed with decoding.

Furthermore, the calculation of the WER scores is based on the assumption that we follow the spelling of the words as indicated in the source corpus for Hiligaynon used in this project. Hence, transcriptions made by Whisper may have influences based on the Tagalog language model leading to some words transcribed in a

variation of its spelling, to which a human may think of as similar words but different to a model. This is evident particularly on the 'i' and 'e' or 'o' and 'u' sounds where the difference may translate to an error.

Chapter 5

Conclusion

In this paper, the researchers trained different ASR models in Kaldi to build a speech-to-text (STT) system with a command line interface. The following acoustic models were considered for training: Monophone, Triphone, LDA + MLLT, SAT and DNN. Eighteen speakers (9 male and 9 female) were gathered for recording. Each speaker spoke 500 randomly chosen words from a dictionary of 1000 words which were considered for recording. However, there is a total of 3,500 words which were included in the system's lexicon. Each word in the lexicon comes with their manually transcribed phonemic transcriptions which is used for training. A six-fold cross validation scheme was followed during training. Results show that the DNN model yielded the lowest word error rate of 2.35% followed by the Monophone model at 2.48%. The following are the results of the remaining models in ascending order: SAT at 3.37%, Triphone at 3.57%, and LDA +MLLT at 13.62%. Considering this, the DNN model was used to build the speech-to-text (STT) system. Specifically, the model produced during the fourth iteration of the

training of the DNN model was used after yielding the lowest WER of 0.40%.

Additionally, Whisper was also used to generate transcriptions of the same audio dataset using different sizes of pre-trained models available in Kaldi; specifically the tiny, base and small models which contain multilingual data. It's important to note that the language parameter passed to the Whisper system was Tagalog. Considering that it is the only Philippine language officially supported by Whisper, as well as considering the phonetic similarities of Tagalog and Hiligaynon, the researchers found it reasonable to proceed with the transcription. Results show that the size of the model is proportional to the performance of the system. The small model yielded the best WER of 25.50%, followed by the base model at 49.02%, and finally by the tiny model at 62.11%.

5.1 Recommendations

Future researchers may conduct further studies by adding to the complexity of the words used in training the system. The researchers may add more words beyond 3 syllables, or train using grammatically correct sentences instead of training just by word-for-word. They may also try to investigate deeper into the linguistic properties of Hiligaynon and incorporate those insights, for instance, in a building a more nuanced phonetic dictionary for the system. Furthermore, other researchers may want to explore techniques on how to make the system more sensitive to the speakers' emotion.

Chapter 6

References

References

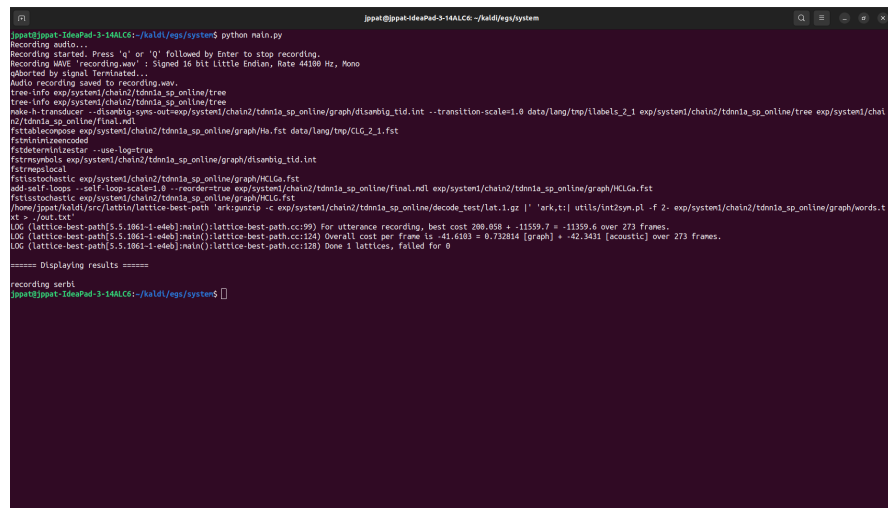
- Aquino, A., Tsang, J. L., Lucas, C. R., & de Leon, F. (2019, 8). G2P and ASR techniques for low-resource phonetic transcription of Tagalog, Cebuano, and Hiligaynon. *2019 International Symposium on Multimedia and Communication Technology (ISMAC)*. Retrieved from <http://dx.doi.org/10.1109/ismac.2019.8836168> doi: 10.1109/ismac.2019.8836168
- Billones, R. K. C., & Dadios, E. P. (2014, 11). Hiligaynon language 5-word vocabulary speech recognition using Mel frequency cepstrum coefficients and genetic algorithm. *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. Retrieved from <http://dx.doi.org/10.1109/hnicem.2014.7016247> doi: 10.1109/hnicem.2014.7016247
- Dimzon, F. D., & Pascual, R. M. (2020, 12). An Automatic Phoneme Recognizer for Children's Filipino Read Speech. *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. Retrieved from <http://dx.doi.org/10.1109/tale48869.2020.9368399> doi: 10.1109/tale48869.2020.9368399
- Eberhard, D. M., Simons, G. F., & Fennig, C. D. (n.d.). *Ethnologue: Languages*

- of the world. twenty-sixth edition.* Dallas, Texas: SIL International.
- Kipyatkova, I., & Karpov, A. (2016). DNN-Based Acoustic Modeling for Russian Speech Recognition Using Kaldi. *Speech and Computer*, 246–253. Retrieved from http://dx.doi.org/10.1007/978-3-319-43958-7_29 doi: 10.1007/978-3-319-43958-7_29
- Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. K., ... Vesely, K. (2011, 1). The Kaldi Speech Recognition Toolkit. *IEEE Automatic Speech Recognition and Understanding Workshop*. Retrieved from https://publications.idiap.ch/downloads/papers/2012/Povey_ASRU2011_2011.pdf
- Radford, A., Kim, J., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022, 12). Robust speech recognition via large-scale weak supervision.
- Upadhyaya, P., Farooq, O., Abidi, M. R., & Varshney, Y. V. (2017, 3). Continuous hindi speech recognition model based on Kaldi ASR toolkit. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Retrieved from <http://dx.doi.org/10.1109/wispnet.2017.8299868> doi: 10.1109/wispnet.2017.8299868
- Wolfenden, E. (2019). *Hiligaynon Reference Grammar* (Open Access ed.). University of Hawaii Press. Retrieved from <https://core.ac.uk/display/211329359>

Appendix A

Appendix

A.1 Command Line STT system



```
jppat@jppat-IdeaPad-3-14ALC6: ~/kaldi/egs/system$ python main.py
Recording audio...
Recording started. Press 'q' or 'Q' followed by Enter to stop recording.
Recording WAVE 'recording.wav' : Signed 16 Bit Little Endian, Rate 44100 Hz, Mono
Aborted by signal terminated...
Audio recording saved to recording.wav.
tree-info exp/system1/chain2/tdmia_sp_online/tree
tree-info exp/system1/chain2/tdmia_sp_online/tree
make-h-transducer --disambig-syms-out=exp/system1/chain2/tdmia_sp_online/graph/disambig_tld.int --transition-scale=1.0 data/lang/tmp/labels_2_1 exp/system1/chain2/tdmia_sp_online/tree exp/system1/chain2/tdmia_sp_online/final.mdl
fsttablecompose exp/system1/chain2/tdmia_sp_online/graph/Hs.fst data/lang/tmp/CLG_2_1.fst
fstminimizecoded
fsttobinlattice
fsttobinlattice --use-logtrue
fsttobinlattice exp/system1/chain2/tdmia_sp_online/graph/disambig_tld.int
fsttobinlattice
fsttostochastic exp/system1/chain2/tdmia_sp_online/graph/HCLGa.fst
add-self-loop --self-loop-scale=1.0 --read-er-tree exp/system1/chain2/tdmia_sp_online/final.mdl exp/system1/chain2/tdmia_sp_online/graph/HCLGa.fst
fsttostochastic exp/system1/chain2/tdmia_sp_online/graph/HCLG.fst
/home/jppat/kaldi/src/lattice/lattice-best-path 'ark:gunzip -c exp/system1/chain2/tdmia_sp_online/decode_test/lat.1.gz |' 'ark,tt:|' utils/int2syn.pl -f 2- exp/system1/chain2/tdmia_sp_online/graph/words.txt > ./out.txt
LOG (lattice-best-path[5.5.1061-1-e4eb]:main):lattice-best-path.ccc:99) For utterance recording, best cost 200.058 + -11559.7 = -11359.6 over 273 frames.
LOG (lattice-best-path[5.5.1061-1-e4eb]:main):lattice-best-path.ccc:124) Overall cost per frame is -45.6183 = 0.732834 [graph] + -42.3431 [acoustic] over 273 frames.
LOG (lattice-best-path[5.5.1061-1-e4eb]:main):lattice-best-path.ccc:128) Done 1 lattices, failed for 0
===== Displaying results =====
recording serbi
jppat@jppat-IdeaPad-3-14ALC6:~/kaldi/egs/system$
```

Figure A.1: Screenshot of the system when called without an argument.

```

jppat@jppat-ideaPad-3-144LC: ~/hald/egs/system$ python main.py recording.wav
tree-info exp/system1/chain2/tdmia_sp_online/tree
tree-info exp/system1/chain2/tdmia_sp_online/tree
make -s -f transducer -d /system1/chain2/tdmia_sp_online/graph/disambig_tid.int --transition-scale=1.0 data/lang/tmp/labels_2_1 exp/system1/chain2/tdmia_sp_online/graph/disambig_tid.int
fsttablecompose exp/system1/chain2/tdmia_sp_online/graph/HtG.fst data/lang/tmp/CLG_2_1.fst
fsttrmsymbols exp/system1/chain2/tdmia_sp_online/graph/disambig_tid.int
fstvln(int)zeencoded
fsttrmsymbols
fstddeterminizestar --use-log-trim
fststochastic exp/system1/chain2/tdmia_sp_online/graph/HtG.fst
add-self-loops --self-loop-scale=1.0 --reorder=true exp/system1/chain2/tdmia_sp_online/graph/HtG.fst
fststochastic exp/system1/chain2/tdmia_sp_online/graph/HtG.fst
/home/jppat/ideaPad/serbi/lattice-best-path.sh > exp/system1/chain2/tdmia_sp_online/decode_test/lat.1.gz 'ark,t:| uttl:/int2syn.pl -f 2- exp/system1/chain2/tdmia_sp_online/graph/words.t
ark> ./out.txt'
LOG (lattice-best-path[5.5.1061-1-e4eb]:main():lattice-best-path.ccc:99) For utterance recording, best cost 200.008 + -11559.7 = -11359.6 over 273 frames.
LOG (lattice-best-path[5.5.1061-1-e4eb]:main():lattice-best-path.ccc:124) Overall cost per frame is -41.6183 + 0.732834 [graph] + -42.3431 [acoustic] over 273 frames.
LOG (lattice-best-path[5.5.1061-1-e4eb]:main():lattice-best-path.ccc:128) Done 1 lattices, failed for 0

===== DIsplaying results =====
recording serbi
jppat@jppat-ideaPad-3-144LC: ~/hald/egs/system$

```

Figure A.2: Screenshot of the system with an audio file's file name as an argument.

A.2 Code snippets

Training the different ASR acoustic models

```

1  ### Mono Training
2
3  if [ $stage -le 0 ]; then
4
5  echo
6  echo "Starting stage 0"
7  echo
8  echo
9  echo "==== MONO TRAINING ====="
10 echo
11

```

```
12 steps/train_mono.sh --nj $nj --cmd "$train_cmd" data/
    train data/lang exp/system1/mono || exit 1
13 data/train data/lang exp/system1/mono || exit 1
14
15 # Graph compilation
16 utils/mkgraph.sh data/lang exp/system1/mono exp/system1/
    mono/graph || exit 1
17 fi
18
19 if [ $stage -le 3 ]; then
20 # Decoding
21 echo
22 echo "====_MONO_DECODING_===="
23 echo
24 steps/decode.sh --nj $decode_nj --cmd "$decode_cmd" exp/
    system1/mono/graph data/test exp/system1/mono/decode
25 echo -e "Mono_training_done.\n"
26
27 fi
```

```
1 if [ $stage -le 4 ]; then
2     echo
3     echo "Starting_stage_4"
4     echo
5
6     echo
```

```
7 echo "====_TRIPHONE_TRAINING_===="
8 echo
9
10 steps/align_si.sh --boost-silence 1.25 --nj $nj --cmd "
    $train_cmd" data/train data/lang exp/system1/mono exp
    /system1/mono_ali
11
12 data/train data/lang exp/system1/mono_ali exp/system1/
    tri1
13
14 steps/train_deltas.sh --boost-silence 1.25 --cmd "
    $train_cmd" 3200 30000 data/train data/lang exp/
    system1/mono_ali exp/system1/tri1
15
16 ## Graph compilation
17 utils/mkgraph.sh data/lang exp/system1/tri1 exp/system1
    /tri1/graph
18
19 # Decoding
20 echo
21 echo "====_TRIPHONE_DECODING_===="
22 echo
23
24 steps/decode.sh --nj $decode_nj --cmd "$train_cmd" exp/
    system1/tri1/graph data/test exp/system1/tri1/decode
```

```
25     /tri1/decode
26 fi

1 # train an LDA+MLLT system.
2 if [ $stage -le 5 ]; then
3     echo
4     echo "Starting stage 5"
5     echo
6     echo "====_TRIPHONE_LDA_MLLT_TRAINING_===="
7     steps/align_si.sh --nj $nj --cmd "$train_cmd" data/
        train data/lang exp/system1/tri1 exp/system1/
        tri1.ali
8     steps/train_lda_mllt.sh --cmd "$train_cmd" --splice-
        opts "--left-context=3--right-context=3" 2000
        20000 data/train data/lang exp/system1/tri1.ali
        exp/system1/tri2b
9
10    # Graph compilation
11    utils/mkgraph.sh data/lang exp/system1/tri2b exp/
        system1/tri2b/graph
12
13    # Decoding
14    echo "====_TRIPHONE_LDA_MLLT_DECODING_===="
15    steps/decode.sh --nj $decode_nj --cmd "$train_cmd" exp
        /system1/tri2b/graph data/test exp/system1/tri2b/
```



```
        decode
16 fi

1 # Train tri3b, which is LDA+MLLT+SAT
2 if [ $stage -le 6 ]; then
3     echo
4     echo "Starting stage 6"
5     echo
6
7     # Align utts using the tri2b model
8     steps/align_si.sh --nj $nj --cmd "$train_cmd" --use-
        graphs true data/train data/lang exp/system1/tri2b
        exp/system1/tri2b.ali
9     steps/train_sat.sh --cmd "$train_cmd" 2000 20000 data/
        train data/lang exp/system1/tri2b.ali exp/system1/
        tri3b
10    utils/mkgraph.sh data/lang exp/system1/tri3b exp/
        system1/tri3b/graph
11    steps/decode_fmllr.sh --nj $decode_nj --cmd "
        $train_cmd" exp/system1/tri3b/graph data/test exp/
        system1/tri3b/decode
12 fi
13
14 if [ $stage -le 8 ]; then
15     echo
16     echo "Starting stage 8"
```

```
17  echo
18  steps/align_fmllr.sh --nj $nj --cmd "$train_cmd" data/
    train data/lang exp/system1/tri3b exp/system1/
    tri3b_ali
19
20  # decode using the tri3b model
21  utils/mkgraph.sh data/lang exp/system1/tri3b exp/
    system1/tri3b/graph
22  steps/decode_fmllr.sh --nj $decode_nj --cmd "
    $decode_cmd" exp/system1/tri3b/graph data/test exp/
    system1/tri3b/decode
23  fi
```

```
1  # Train a chain model
2  if [ $stage -le 9 ]; then
3      echo
4      echo "Starting stage 9: Chain Model"
5      echo
6      local chain2/run_tdnn.sh
7  fi
8
9  if [ $stage -le 10 ]; then
10     echo
11     echo "Starting stage 10: Display results"
12     echo
13     for x in exp/system1/*/decode*;
```

```
14         do
15             [ -d $x ] && grep WER $x/wer_* | utils/
16                 best_wer.sh;
17
18         done
19
20     for x in exp/system1/chain2*/*/decode*;
21     do [ -d $x ] && grep WER $x/wer_* | utils/
22         best_wer.sh; done
23 fi
```

Code snippets for the speech-to-text system

```
1 # Prepare online decoding
2 os.makedirs("./exp/system1/chain2/tdnn1a_sp_online",
3             exist_ok=True)
4
5 bash_out = s.run("steps/online/nnet3/
6                 prepare_online_decoding.sh --mfcc-config conf/
7                 mfcc_hires.conf "data/lang exp/system1/nnet3/
8                 extractor exp/system1/chain2/tdnn1a_sp exp/system1/
9                 chain2/tdnn1a_sp_online", stdout=f, text=True, shell=
10                 True)
11
12 # Create decoding graph
13 os.makedirs("./exp/system1/chain2/tdnn1a_sp_online/graph
14             ", exist_ok=True)
```

```

7 bash_out = s.run("utils/mkgraph.sh --self-loop-scale 1.0
    " "data/lang_exp/system1/chain2/tdnn1a_sp_online_exp
    /system1/chain2/tdnn1a_sp_online/graph", stdout=f,
    text=True, shell=True)

8

9 # Decode using the created graph
10 os.makedirs("./exp/system1/chain2/tdnn1a_sp_online/
    decode_test", exist_ok=True)
11 bash_out = s.run("steps/online/nnet3/decode.sh --acwt 1.0
    --post-decode-acwt 10.0 --nj 1" "exp/system1/
    chain2/tdnn1a_sp_online/graph." "exp/system1/chain2
    /tdnn1a_sp_online/decode_test", stdout=f, text=True,
    shell=True)

12

13 # GET TRANSCRIPTION
14 gz_location = "exp/system1/chain2/tdnn1a_sp_online/
    decode_test/lat.1.gz"
15 words_txt_loc = "exp/system1/chain2/tdnn1a_sp_online/
    graph/words.txt"
16 command = "~/kaldi/src/latbin/lattice-best-path" \ "ark
    : 'gunzip -c {0} |' " \
17         "'ark,t:|utils/int2sym.pl -f 2 -" \
18         "{1}>./out.txt' ".format(gz_location,
    words_txt_loc)

```

```
19 bash_out = s.run(command, stdout=f, text=True, shell=
    True)
```

Transcribing using Whisper

```
1 #!/bin/bash
2
3 # Directory path
4 directory="./audio"
5
6 # Loop through each file in the directory
7 for file in "$directory"/*.wav; do
8     if [ -f "$file" ]; then
9         echo "transcribing␣file:␣"$file"."
10        whisper "$file" --language Tagalog --model base
11    fi
12 done
13
14 echo "transcribing␣done"
```