Isolated Tagalog Vowel Recognition using the Wall Street Journal Speech Corpus and Hidden Markov Models

Marl Aldwin C. Bermudo, Omar Job Abesamis and Joel Addawe

Abstract— **Speech recognition is the conversion of speech into text. Over the years, the predominant method of doing this has been the use of a mathematical framework (hidden Markov models) to map the (acoustic) signal features of the speech to the (phonetic) transcription of the text. This mapping results in the creation of a speech corpora (Wall Street Journal) trained with transcribed speeches of certain speakers of a language (English). This speech corpora can then be used by a software framework (Sphinx 4) for the actual task of speech recognition.**

The question of how well a speech corpora can recognize languages it was not natively trained in - its domain independence - is revealing of certain deep theoretical questions in linguistics, as well as practical needs brought by globalization. In particular, testing the domain independence of the Wall Street Journal (WSJ) Speech Corpora by modeling a minimal subset of a non-native language (Tagalog vowels) is feasible, useful, and novel. This is the task the thesis undertakes.

I. Introduction

Speech is the primary means of communication between people [42]. In recent years, its fundamental nature has been a subject of academic study, from the philosophical, to the communicative, and in particular the computational.

This paper will study one computational approach in the study of speech called speech recognition. The task is essentially the conversion of speech (spoken language) into text (written language). There is also the related field (speech synthesis) which does the reverse of recognition - conversion of text into speech. This paper will only tackle speech recognition.

History of Speech Recognition

The idea of giving computers the ability to process human speech is as old as the idea of computers themselves, and has intrigued engineers and scientists for centuries [34 42].

The first recorded modern experiment in speech recognition was in the 1930s, when Homer Dudley of Bell Laboratories proposed a system model for speech analysis and synthesis [42].

The problem of automatic speech recognition has since then been approached progressively. It started with simple machines that respond to a small set of sounds. Now, sophisticated systems respond fluently to spoken natural language, while taking into account the varying statistics of the language in which the speech is produced [42].

The introduction of statistical methods has been the most significant progress in speech recognition, in particular the Hidden Markov Models (HMM). The HMM methodology represented a major step forward from the simple pattern recognition and acoustic-phonetic methods used earlier in automatic speech recognition systems. Together with the (stochastic) language model, they enabled powerful new methods for handling virtually any continuous speech recognition problem efficiently and with high performance. It has now become the preferred method for speech recognition, especially because of the steady stream of improvements and refinements of the technology. [37 42]

Phonemes and the Tagalog Language

The basic unit of speech that can be spoken distinguishably is called a phoneme. Phonemes can be represented with characters enclosed by / /. There are two major classes of phonemes namely vowels and consonants. A relatively open tract produces vowels. On the other hand, a relatively closed vocal tract, resulting in an audible effect on the airflow produces consonants [45].

The Tagalog phoneme is similar to other languages composed of vowels and consonants. In 1940, the renowned Lope K. Santos adapted the Abakadang Tagalog where he categorically described 21 phonemes as follows [45]:

Vowels: /a/ /e/ /i/ /o/ /u/

Consonants: /b/ /k/ /d/ /g/ /h/ /l/ /m/ /n/ /nan/ /p/ /r/ /s/ /t/ /w/ /y/ /?/

The character enclosed by / / is the alphabet that represent the phoneme with the exception of /nan/ which corresponds to the digraph ng and /?/ is identified with (') as in bat'a and (-) as in may-ari does not correspond to a specific alphabet but it is characterized by the closure of the glottis [45].

Presented to the Faculty of the Department of Mathematics and Computer Science, University of the Philippines Baguio in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

Vowel	Phoneme Sequence
A	/a/
E	/e/
I	/i/
O	/o/
U	/u/

Consonant Phoneme Sequence

B	/ba/
K	/ka/
D	/da/
G	/ga/
H	/ha/
L	/la/
M	/ma/
N	/na/
NG	/nan/
P	/pa/
R	/ra/
S	/sa/
T	/ta/
W	/wa/
Y	/ya/

Tagalog alphabets are also classified either a vowel or consonants. Vowel alphabets are pronounced similar to its phoneme. Consonants are a combination of two phonemes: the letter's corresponding phoneme followed by /a/. The alphabet NG is an exception, which is pronounced as /nan/ [45].

Acoustics Models and the Wall Street Journal Speech Corpus

Acoustic models are statistical models that estimate the probability that a certain phoneme has been uttered in a recorded audio segment. The models are trained on several hours worth of pre-recorded speech. To give generality to the model the material includes speakers of different age and sex. [47]

Most acoustic models in speech recognition are mono-lingual - they are trained from speakers using a single language [47]. This is the case for the Wall Street Journal (WSJ) Speech corpus, which is a first general-purpose English, large vocabulary, natural language, high perplexity, corpus containing significant quantities of both speech data (400 hrs) and text data (47M words) [48].

With globalization and immigration, people of different nationalities travel more to different parts of the world and increasingly communicate with each other using a common language, specifically English. This is particularly true for the Filipino people, who use a lot of loan words from other languages like English. Let us define domain-independence as the degree of multilinguiality of a speech corpus - the degree to which it can model languages in which it was not trained. An interesting research problem, therefore, is to measure how well an English-based speech corpus like the WSJ can model the Filipino (Tagalog) language. Specifically, testing the domain independence of the WSJ Speech Corpora by modeling a minimal subset of the Tagalog language (phonemes) is both useful and feasible.

For this paper, we created a program to recognize Tagalog phonemes (vowels) using the WSJ acoustic model and the HMM-based Sphinx 4 framework. We did an experiment consisting of a male and a female who uttered and recorded voices for the program we created to recognize. We recorded the results of the recognition, and computed the word error rate. Finally, recommendations for future work are provided.

II. Materials and Methods

The task of speech recognition requires the mathematical framework of hidden Markov models. This method has now become preferred for most speech recognition tasks [37 42]. Modern software frameworks for speech recognition, in particular the Sphinx 4, is based on this model. The experiments conducted in this thesis will be based on the Sphinx 4 hidden Markov model framework. **Probability Theory**

In order to understand Hidden Markov Model, a background in probability theory is needed to begin the study.

Probability theory is the branch of mathematics concerned with the study of uncertainty [2]. It deals with the analysis of phenomena with random behavior [44]. It has applications in the sciences, gambling, business, health, research, and in particular, speech recognition.

To mathematically define probability, some basic elements must be introduced:

- Sample Space Ω - the set of all outcomes of a random experiment. Each outcome $\omega \subset \Omega$ can be thought of a state of the world after the experiment
- Event Space ζ - a set who elements $A \in \zeta$ are subsets of Ω. A is a measurable set of points or sets
- Probability Measure - a function $P : \zeta \rightarrow R$ that satisfies the following properties
  - $P(A) > 0$, for all $A \in \zeta$
  - $P(\Omega) = 1$
  - If $A_1, A_2, \ldots,$ are disjoint events - $A_i \cap A_j = \emptyset$, then

$$P(\cup_i A_i) = \sum_i P(A_i) \tag{1}$$

Thus, P(A) is the probability that an event A may happen on a number of possibilities in the set Ω. An example would be coin toss: if A is the probability of heads, the sample space Ω is heads, tails, and P(A) = 0.5.

More formally,

$$P(A) = \frac{n(A)}{n(\Omega)} \tag{2}$$

Where $n(X)$ is the cardinality, or number of elements of X in the sample space Ω (Applies only for the discrete case).

The following properties follow:

- $P(\emptyset) = 0$
- $P(A \cup B) = P(A) + P(B) = P(A \cap B)$
- $P(\Omega) = 1 - P(A)$

Definition: Mutual Exclusivity
Let $A_1, ..., A_k$ be a set of events on Ω.
Then A_i and A_j are mutually exclusive if and only if

$$P(A_i \cup A_j) = 0; \forall i, j, i \neq j \tag{3}$$

Definition: Independence
Let $A_1, \ldots, A_k$ be a set of events on Ω.
Then A_i and A_j are mutually exclusive if and only if

$$P(A_1 \cap A_2 \cap ... \cap A_k) = P(A_1) \cap P(A_2) \cap \ldots \cap P(A_k) \tag{4}$$

Definition: Conditional Probability
Let B be an event with non-zero probability. The conditional distribution of any event A given event B is

$$P(A|B) = \frac{P(A \cup B)}{P(B)} \tag{5}$$

Conditional probabilities are useful in many areas of application. In particular, the key formula of most speech recognition systems relies on a conditional probability.

In the coin toss example, assume we toss the coin twice. The value of the first coin is say, heads, and the value of the second toss of the same coin is say, tails. Observing the value of the first coin, we know that the second coin if not at all influenced by whatever value the first coin has; A and B are known to be independent. If we let A be the first coin and B the second coin, we get

$$P(A|B) = \frac{P(A \cup B)}{(P(B)} = \frac{P(A)P(B)}{(P(B)} = P(A) \tag{6}$$

This is a general result.

There are important classes of conditional probabilities whose values are not independent but depend only on the immediate predecessor event. We tackle such probabilities later.

Definition: Random Variables

A random variable X is a function $X : \Omega \to R^2$

Often we are not interested in probabilities of specific cases, like of heads or tails. We are more interested in questions like the number of heads in x tosses, or the longest tail sequence. And we want this function to be real-valued to be useful [1]. This is the random variable (RV)

In the coin tossing example, although it will not be derived, the probability of x heads in n tosses is a random variable X of the form:

$$P(X = x; n) = \binom{n}{x} 0.5^x 0.5^{n-x} \tag{7}$$

This random variable takes on a finite set of possible values this is called a discrete random variable.

In general, we can assign a probability mass function (pmf) $p_X(x)$ to any discrete random variable X,

$$p_X(x) = P(X = x) \tag{8}$$

The following properties must be satisfied by the pmf:

- $0 \leq p_X(x) \leq 1$
- $\sum_x p_X(x) = 1$

In the previous example, summing the equation over x,

$$\sum_x p_X(x) = \sum_x \binom{n}{x} 0.5^x 0.5^{n-x} \tag{9}$$

By the Binomial Theorem

$$(x + a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k} \tag{10}$$

Thus,

$$\sum_x p_X(x) = \sum_x^n \binom{n}{x} 0.5^x 0.5^{n-x} = (0.5 + 0.5)^n = 1 \tag{11}$$

A. Markov Chains

Real life experiments that use probability often deal with groups of random variables. Of particular interest is a sequence of random variables, all sharing a certain property.

Denote this finite sequence of RVs by $(X_1, X_2, \ldots, X_n)$, defined on a sample space Ω.

The probabilities involving $(X_1, X_2, ..., X_n)$ can be prescribed by the joint probability mass function

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = p_{12\ldots n}(x_1, x_2, \ldots, x_n) \tag{12}$$

We are interested in joint conditional probability mass functions of the form,

$$P(X_{n+1} = x_n | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \tag{13}$$

The certain property they share is that the value of every RV depends only on the immediately previous RV

$$\begin{aligned} &P(X_{n+1} = x_n | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \\ &\quad = P(X_{n+1} = x_n | X_n = x_n), n \geq 0 \end{aligned} \tag{14}$$

This is called the Markov property.

A theorem important in establishing the class of Markov chains is given. This theorem will not be proven as it uses

advanced mathematical analysis and measure theory.

Theorem 1 :
Let $(X_1, X_2, \ldots, X_n)$ be a sequence of RVs having this joint pmf:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = p_{12\ldots n}(x_1, x_2, \ldots, x_n) \tag{15}$$

If

$$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \sum_{x_{k+1}\ldots x_n} P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) \tag{16}$$

Coincides with

$$p_{12\ldots n}(x_1, x_2, ..., x_n) \tag{17}$$

for all n and $k < n$, then the system is consistent.

The sequence $(X_1, X_2, ..., X_n)$ can be viewed as a random variable going through time. We denote this sequence a Markov chain, q_t.

The values $x_1, x_2, \ldots, x_n$ are all taken from the sample space Ω. We can think of this sample space as a set of states of a particular system letters in a poem, phonemes in speech, etc. Each value of q_t at t = 1,2, . . . , n can take on any of those values with a specific probability [19]. In particular, because of the Markov property, the set of all such probabilities, or state transition probabilities, can be represented in a matrix of the form $a_{ij} = P(q_t = x_j | q_{t-1} = x_i) 1 \leq i, j \leq n$

a_{ij} is also a probability mass function, so it must satisfy

- $a_{ij} \geq 0$
- $\sum_j^n a_{ij} = 1$

The values $x_1, x_2, ..., x_n$ of the sample space Ω are to be called states, and are to be denoted $S_1, S_2, ..., S_n$. This is done for notational standard in speech recognition, where states are usually denoted so.

An example Markov chain with N states (N=5) and corresponding state transition probabilities can be represented in a diagram:
The state transition probabilities is a matrix A:
Let us prove the form of the probabilities of the Markov chain of interest.

Theorem 2:

A sequence of discrete random variables $(X_1, X_2, ..., X_n)$ is a Markov chain if and only if for all $i_1, i_2, ..., i_n \in \Omega, n \geq 0$

$$P(X_1 = i_1, X_2 = i_2, ..., X_n = i_n) = p_{i_1} p_{i_1} p_{i_2} ... p_{i_{n-1}} p_{i_n} \tag{18}$$

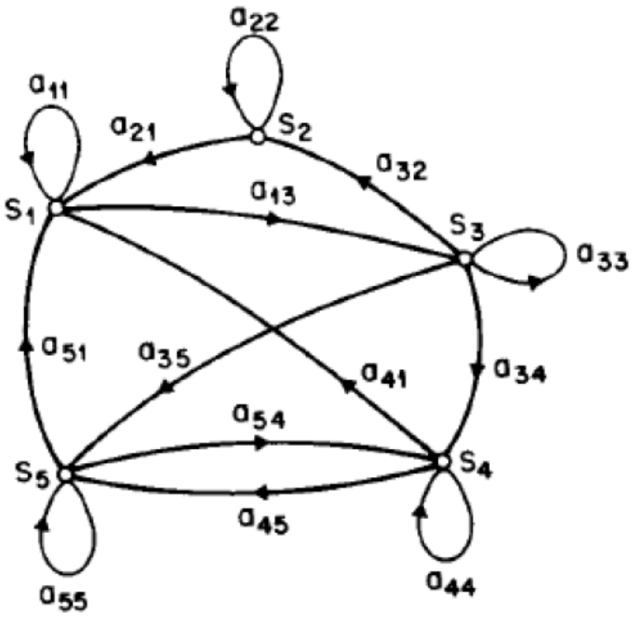

Fig. 1. Markov Chain

$$A = \{a_{ij}\} = \begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{matrix}$$

Fig. 2. Transition Matrix

Proof:
Let $(X_1, X_2, ..., X_n)$ have the Markov property. Then,

$$\begin{aligned} &P(X_1 = i_1, X_2 = i_2, ..., X_n = i_n) = \\ &P(X_1 = i_1) P(X_2 = i_2 | X_1 = i_1) ... \\ &P(X_n = x_n | X_1 = x_1, X_2 = x_2, ..., X_{n-1} = x_{n-1}) \\ &= p_{i_1} p_{i_1} p_{i_2} ... p_{i_{n-1}} p_{i_n} \end{aligned} \tag{19}$$

Now define:

$$P(X_1 = i_1) = p_{i_1} \tag{20}$$

if p_{i_1} is summed over $i_{k+1}, i_{k+2}, ..., i_n$, we obtain $p_{i_1} p_{i_1} p_{i_2} ... p_{i_{k-1}} p_{i_k}$ since the Markov matrix is a PMF.

Thus, the sequence of RVs is consistent, and by Theorem 1, we can write the probabilities of the Markov chain using the formula of Theorem 2

B. Hidden Markov Models

In a Markov chain (see Appendix), we can correspond an observable (physical) event with a particular state. But as will be demonstrated, the Markov property alone can be too restrictive for many problems of interest [19].

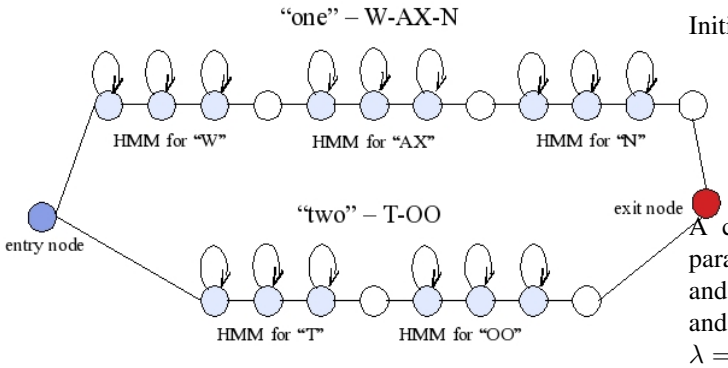

Fig. 3. Hidden Markov Model of the words "one" and "two"

If we have a 3-state Markov chain, and each state stands for an event in the weather - sunny, cloudy, or rainy - we can simply compute the probability of the upcoming weather based on the previous weather(s). But in the particular and often real-world case that we cannot observe the state but must infer it from another set of states, the Markov chain would not be able to explain this without additional information. In our weather example, if the states of "sunny, cloudy, rainy" are 'hidden', but a certain piece of evidence exists - a piece of seaweed whose sogginess is informative of the weather - then we can probabilistically deduce the weather from the states of the seaweed's sogginess [6].

This is an example of a Hidden Markov Model (HMM). More formally, an HMM can be characterized by the following [19]:

N - number of states in the model.

- The states that are being modeled in the real-world are hidden, but in the model, assumptions have to be made as to the number of them.
- In the weather example, the states can be "sunny, cloudy, or rainy"
- The individual states are denoted $S = S_1, S_2, \ldots, S_n$

M - number of distinct observation symbols per state

- Correspond to the physical output of the system being modeled
- In the weather example, the sogginess of the seaweed can be "dry, damp, wet" and are the observation symbols
- The individual symbols are denoted $V = v_1, v_2, \ldots, v_n$

State transition probability distribution $A = a_ij$ where

$$a_{ij} = P(q_t = x_j | q_{(t-1)} = x_i); 1 \leq i, j \leq n \tag{21}$$

Observation symbol probability distribution in state j, $B = b_j(k)$, where

$$b_j(k) = P[v_k \; at \; t | q_t = S_j] \tag{22}$$
$$1 \leq j \leq N; 1 \leq k \leq M \tag{23}$$

Initial state distribution $\pi = \pi_i$ where

$$p_i = P(q_1 = S_i) \tag{24}$$
$$1 \leq i \leq N \tag{25}$$

A complete specification of an HMM requires two model parameters (N and M), specification of observation symbols, and the specification of three probability measures A, B, and π [19]. For compact notation, designate the HMM as $\lambda = (A, B, \pi)$.

This diagram highlights in particular how an HMM works for speech recognition [45]. Note that there is more than one HMM - groups of them are used to 'recognize' either the words "one" or "two". At the entry node, an unknown word (an observation sequence) enters both the "one" and "two" sets of HMMs. They will get evaluated at both sides, resulting in a probability at the exit node of the form

$$P(< unknownword > | model) = ? \tag{26}$$

Whichever of "one" or "two" has the highest probability will be the output word. In the methodology, the computation of $P(< unknownword > | model)$ will be discussed further.

C. Sphinx 4

The high-level architecture of a sphinx 4 program is given in figure 2.

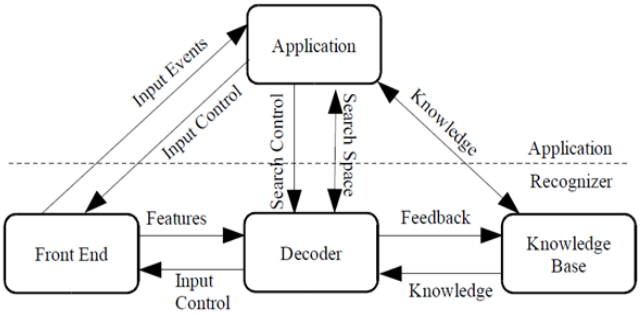

Fig. 4. The high-level architecture of a sphinx 4

The front end gathers, annotates, and processes the input data. It also extracts features from the input data to be read by the decoder, and provides the annotations to include the beginning and ending of a data segment. Operations performed include preemphasis, noise cancellation, automatic gain control, end pointing, Fourier analysis, Mel spectrum filtering, cepstral extraction, etc [49].

The knowledge base provides the information the decoder needs to do its job, including the the acoustic model and the language model. This knowledge base can also receive feedback from the decoder, permitting the knowledge base to dynamically modify itself based upon successive search

TABLE I
TABLE OF RECOGNIZED VOWEL

Vowel	Male Speaker		Female Speaker	
	Live Recognition	**Recorded Recognition**	**Live Recognition**	**Recorded Recognition**
/a/	/u/	/a/	/e/	/e/
	/a/	/u/	/e/	/e/
	/o/	/u/	/u/	/i/
	/o/	/u/	/e/	/o/
	/o/	/e/	/u/	/o/
	/o/	/o/	/u/	/i/
	/u/	/o/	/u/	/u/
	/o/	/o/	/o/	/e/
	/u/	/o/	/u/	/i/
	/o/	/i/	/e/	/o/
/e/	/e/	/i/	/e/	/e/
	/u/	/i/	/e/	/e/
	/e/	/i/	/e/	/u/
	/e/	/i/	/u/	/u/
	/e/	/i/	/o/	/o/
	/e/	/i/	/u/	/u/
	/e/	/o/	/u/	/u/
	/e/	/i/	/u/	/o/
	/e/	/i/	/e/	/o/
	/e/	/u/	/u/	/o/
/i/	/i/	/u/	/i/	/u/
	/i/	/u/	/i/	/i/
	/i/	/u/	/i/	/u/
	/i/	/i/	/u/	/u/
	/u/	/u/	/i/	/i/
	/i/	/e/	/i/	/i/
	/i/	/u/	/u/	/i/
	/i/	/u/	/i/	/i/
	/i/	/u/	/u/	/i/
	/i/	/i/	/u/	/i/
/o/	/a/	/u/	/o/	/e/
	/a/	/o/	/o/	/e/
	/o/	/o/	/o/	/u/
	/o/	/o/	/o/	/i/
	/o/	/o/	/o/	/u/
	/u/	/o/	/o/	/o/
	/a/	/o/	/o/	/u/
	/u/	/o/	/o/	/i/
	/a/	/o/	/o/	/o/
	/a/	/o/	/o/	/u/
/u/	/u/	/u/	/o/	/e/
	/o/	/i/	/o/	/u/
	/u/	/e/	/o/	/e/
	/o/	/u/	/o/	/i/
	/o/	/o/	/i/	/e/
	/o/	/o/	/u/	/u/
	/o/	/u/	/o/	/i/
	/o/	/u/	/o/	/o/
	/o/	/o/	/o/	/i/
	/u/	/i/	/o/	/o/

Table 1: Table of actual vowel uttered by the speakers against the outputted result vowel recognized by the program.

TABLE II
VOWEL ERROR RATE

Vowel	**Male Speaker**		**Female Speaker**	
	Live Recognition	**Recorded Recognition**	**Live Recognition**	**Recorded Recognition**
/a/	9/10 (90%)	9/10 (90%)	10/10 (100%)	10/10 (100%)
/e/	1/10 (10%)	10/10 (100%)	6/10 (60%)	8/10 (80%)
/i/	1/10 (10%)	8/10 (80%)	4/10 (40%)	3/10 (30%)
/o/	7/10 (70%)	1/10 (10%)	0/10 (0%)	8/10 (80%)
/u/	7/10 (70%)	6/10 (60%)	9/10 (90%)	8/10 (80%)

Table2: Vowel Error Rate as computed from Table 2.

In the recorded recognition phase of the experiment a recording device (Samsung Star Wifi) is used to capture each utterance of the speaker, and this is done during the live recognition phase while the speaker is uttering a vowel. After the live recognition, the recording device is played and it is positioned directly to the microphone. Every time the recording device was played, the program will recognize and output a corresponding result vowel. This data is then tabulated in table 1. The distance from the mouth to the recording device, as well as the device to the microphone, is also estimated to be 7 cm.

E. Word Error Rate

The word error rate (WER) is computed as:

$$WER = \frac{S + D + I}{N} \tag{27}$$

Where

* S is the number of substitutions
* D is the number of deletions
* I is the number of insertions
* N is the number of words in the reference

In this paper, only substitutions occurs. Thus, deletions and insertions are equal to zero.

Since only vowels were tested in the experiment, it is understood that all the words in queston is a vowel, and the word error rate may be substituted by the vowel error rate.

III. Results and Discussion

Based on the table (Figure 2), the highest vowel error rate is /a/ with 0.950, while the lowest vowel error rates are /i/ and /o/ at .400.

On the one hundred samples given to each speaker, the male speaker has a vowel error rate of .590, while the female speaker has a vowel error rate of .660.

Also, on the one hundred samples allocated to each method of recognition, the live recognition has a vowel error rate of .540, and the recorded recognition has a vowel error rate of .710.

The overall vowel error rate, based on 125 errors out of 200 utterances, is .625.

Based on the overall vowel error rate of the 200 samples, the domain independence of the Wall street Journal (WSJ) speech corpus on modeling the Tagalog vowels is generally of poor quality.

IV. Conclusion and Future Work

This work can be extended to the recognition of consonants, isolated words, and continuous speech of the Filipino (Tagalog) language.

Determining the sample size of speakers necessary to represent the (Tagalog) language must also be considered.

The nature of the application of interest is a determining factor in what error rates are acceptable [3]. There are some applications for which even 0.05 word error rate is unacceptable; this constraint should be accounted for in further research.

Most speech recognition systems are far from achieving human level language comprehension; this work, in the context of a given speech corpora (Wall Street Journal) and a subset of a language to recognize (Tagalog vowels), can be an inch step closer to further understanding the nature of this problem.

Appendix I
Sphinx 4 Class Structure

This is the Java class structure of the Tagalog vowel recognizer program, using the Sphinx 4 framework [49].

* Recognizer The main interface that will be tap to get the result
  * accuracyTracker - Tracks and reports recognition accuracy.
  * memoryTracker - Monitors a recognizer for memory usage.
  * speedTracker - Monitors a recognizer for speed.
  * Decoder - An abstract decoder which implements all functionality which is independent of the used decoding-paradigm
  * SimpleBreadthFirstSeachManager - Provides the breadth first search. To perform recognition an application should call initialize before recognition begins, and repeatedly call ”recognize” until Result.isFinal() returns true. Once a final result has been obtained, ”terminate” should be called. All scores and probabilities are maintained in the log math log domain.
  * SearchManager - primary role is to execute the search for a given number of frames. The SearchManager will return interim results as the recognition proceeds and when recognition completes a final result will be returned.
  * StatisticsVariable - Variable generally used to label, and view a variable anytime (implementation side)
  * SearchStateArc - Represents a single state in a language search space
  * Token - Represents a single state in the recognition trellis. Subclasses of a token are used to represent various emitting state. All scores are maintained in LogMath log base
  * Linguist - The linguist is responsible for representing and managing the search space for the decoder. The role of the linguist is to provide, upon request, the search graph that is to be used by the decoder.

    * Dictionary - The dictionary is responsible for determining how a word is pronounced. The dictionary is a file that maps words to their phonetic transcriptions, that is, it maps words to sequences of phonemes
    * AcousticModel (TiedStateAcousticModel)
        * Recognizer The main interface that will be tap to get the result
* FrontEnd - a wrapper class for the chain of front end processors. It provides methods for manipulating and navigating the processors.
    * Microphone - captures audio data from the system's underlying audio input systems.
    * DataBlocker - A DataProcessor which wraps incoming DoubleData objects into equally size blocks of defined length.
    * AbstractVoiceActivityDetector - An abstract analyzer that signals about presense of speech in last processing frame. This information is used in noise filtering components to estimate noise spectrum for example.
    * Preemphasizer - Implements a high-pass filter that compensates for attenuation in the audio data.
    * RaisedCosineWindower - Slices up a Data object into a number of overlapping windows (usually referred to as "frames" in the speech world).
    * Discrete Fourier Transform - Computes the Discrete Fourier Transform (FT) of an input sequence, using Fast Fourier Transform (FFT).
    * MelFrequencyFilterBank - Filters an input power spectrum through a bank of number of mel-filters.
    * DiscreteCosineTransform - Applies a logarithm and then a Discrete Cosine Transform (DCT) to the input data.
    * LiveCMN - Subtracts the mean of all the input so far from the Data objects.
    * DeltasFeatureExtractor - Computes the delta and double delta of input cepstrum (or plp or ...).

REFERENCES

[1] Robert B. Ash. *Basic Probability Theory*. New York: Dover Publications, 2008.

[2] Arian Maleki, Tom Do. "Review of Probability Theory." Stanford University. Web.

[3] Jay G. Wilpon, Lawrence R. Rabiner, Chin-Hui Lee, E. R. Goldman."Automatic Recognition of Keywords in Unconstrained Speech Using HMM". IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol 38, No. 11, (Nov., 1990). 1-9.

[4] P. Varga, R. K. Moore. "Hidden Markov Model Decomposition of Speech and Noise." Speech Research Unit, Royal Signal and Radar Establishment, (1989). 1-4.

[5] H. Juang, L. R. Rabiner. "Hidden Markov Models for Speech Recognition." Technometrics, Vol. 33, No. 3 (Aug., 1991). 251-272.

[6] "Viterbi Algorithm and Forward Algorithm." www.comp-leeds.com. Web.

[7] Eric Xing. "Machine Learning - Hidden Markov Model." Web.

[8] Igor Bolshakov, Alexander Gelbukh. *Computational Linguistics - Models, Resouces, Applications*. Mexico: Fondo De Cultura Economica, 2004.

[9] Narada Dilp Warakagoda. "A Hybrid ANN-HMM ASR System with NN based Adaptive Preprocessing." Web.

[10] Dirk Husmeier. "A Tutorial on Hidden Markov Models." Biomathematics and Statistics Scotland. Web.

[11] Philip Jackson. "HMM Tutorial." Centre for Vision Speech Signal Processing, University of Surrey. Web.

[12] Rakesh Dugad, U. B. Desai. "A Tutorial on Hidden Markov Models." Signal Processing and Artificial Neural Networks Laboratory. Web.

[13] Andrew W. Moore. "Hidden Markov Models." School of Computer Science, Carnegie Mellon University. Web.

[14] Tapas Kanungo. "Hidden Markov Models." Center for Automation Research, University of Maryland. Web.

[15] Sam Roweis. "Hidden Markov Models." University of Toronto. Web.

[16] Barbara Resch. "Hidden Markov Models." Signal Processing and Speech Communication Laboratory. Web.

[17] Jia Li. "Hidden Markov Model". Department of Statistics, The Pennsylvania State University. Web.

[18] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models." Marcin Marszalek, Visual Geometry Group. Web.

[19] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." Proceedings of the IEEE, Vol 77, No. 2, (Feb 1989).

[20] Javier R. Movellan. "Tutorial on Hidden Markov Models". Web.

[21] Eric Fosler-Lussier. "Markov Models and Hidden Markov Models: A Brief Tutorial." International Computer Science Institute, (1990). Web.

[22] Junichi Yamagishi, Korin Richmond, Simon King. "Hidden Markov Model-based Speech Synthesis." Center for Speech Technology Research, University of Edinburgh, UK. Web.

[23] Phil Blunsom. "Hidden Markov Models.".

[24] Sung-Jung Cho. "Introduction to Hidden Markov Model and Its Application." Samsung Advanced Institute of Technology, (Apr., 2005). Web.

[25] Nikolai Shokhirev. "Hidden Markov Models." Web.

[26] Catherine Sweeney-Reed. "Hidden Markov Models." Web.

[27] Larry Reeve. "Hidden Markov Models Applied to Information Extraction." Web.

[28] Valery A. Petrushin. "Hidden Markov Models - Fundamentals and Applications." Center for Strategic Technology Research, Accenture. Web.

[29] Ed Grabianowski. "How Speech Recognition Works." www.howstuffworks.com. Web.

[30] "Speech Recognition." www.speech.cs.cmu.edu/comp.speech. Web.

[31] John Paul Hosom. "Speech Recognition with Hidden Markov Models". Web.

[32] Christine Englund. "Speech Recognition in the JAS 39 Gripen aircraft." Department of Speech, Music and Hearing, (Mar., 2004) Web.

[33] James H. Martin, Daniel Jurafsky. *Speech and Language Processing*. New Jersey: Pearson Prentic Hall.

[34] J. M. Baker, L. Deng, J. Glass, S. Khundanpur, C. H. Lee, N. Morgan, D. O'Shaughnessy. "Research Developments and Directions in Speech Recognition and Understanding, Part 1." Web.

[35] S. Furui, K. Shikano, S. Matsunaga, T. Matsuoka, S. Takahashi, T. Yamada. "Recent Topics in Speech Recognition at NTT Laboratories." NTT Human Interface Laboratories. Web.

[36] J. M. Baker, L. Deng, S. Khundanpur, C. H. Lee, J. Glass, N. Morgan. "Historical Development and Future Directions in Speech Recognition and Understanding." Report of the Speech Understanding Working Group. Web.

[37] Schliep, B. Georgi, W. Rungsarityotin, I. G. Costa, A. Schonhuth. "The General Hidden Markov Model Library - Analyzing Systems with Unobservable States." Web.

[38] Sadik Kapadia. "Discriminative Training of Hidden Markov Models." Downing College, (1998).

[39] IEEE Computer Society. *Computer Science Curriculum 2008*.

[40] S. Umesh. "Automatic Speech Recognition - Research and Standards." Department of Electrical Engineering, Indian Institute of Technology Madras, (May 2010). Web.

[41] H. Juang, L. R. Rabiner. "Automatic Speech Recognition - A Brief History of the Technology Development." Web.

[42] Dan Ellis. "An Overview of Speech Recognition research at ICSI." International Computer Science Institute, Berkeley California. Web.

[43] L. Buchsbaum, R. Giancarlo. "Algorithmic Aspects in Speech Recognition - An Introduction." Assocation for Computing Machinery. Web.

[44] W. Walker, P. Lamere, P. K. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, J. Woelfel. "Sphinx-4: An Open Source Framework for Speech Recognition". Sun Microsystems, (2004). Web.

[45] Rolando P. Navarro Jr. "Recognition of Tagalog Alphabets Using The Hidden Markov Model." University of the Philippines, Diliman, (Oct., 2007).

[46] D. A. Liauw Kie Fa. "Topics in Speech Recognition." Web.

[47] Andre Mansikkaniemi. "Acoustic Model and Language Model Adaptation for a Mobile Dictation Service." School of Science and Technology, Aalto University, (2010).

[48] Douglas B. Paul, Janet M. Baker. "The Design for the Wall Street Journal-based CSR Corpus." Defense Advanced Research Projects Agency (DARPA).

[49] J. Hajik. ”Linguistics meet exact sciences.” Web.