Janaki Prasad Koirala

Identity Verification with Speech Recognition

A Study

Helsinki Metropolia University of Applied Sciences

Degree: Bachelor of Engineering

Degree Programme: Information Technology

Thesis

2 May 2013

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Janaki Prasad Koirala Identity Verification with Speech Recognition A Study 39 pages + 1 appendix 2 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Sakari Lukkarinen, Principal Lecturer

Since verification and authentication based on speech recognition are important developments in the field of information security, this study revolves around the topic of speech-based Identity verification. The goal was to create a speech recognition system that can be embedded to an existing user authentication system.

For this purpose, the study relied on a free speech modelling tool and modelled a limited dictionary for speech recognition. Speech samples were collected for training and testing purposes. The training sample was collected from three respondents and the testing data from seven, including those from previous three respondents. A freely available tool, HTK Toolkit, was used to train the model with speech samples recorded from respondents. The results showed that the accuracy of the model was dependent on whether the training dataset included the users' speech or not.

The study supports the significance of real use of the model. However, the scope of the study is not large enough for authentication and verification system.

Keywords	Speech Recognition, Identity Verification, HMM

Abbreviations

AI	Artificial Intelligence
ARPAbet	Advanced Research Projects Agency Alphabet
CMU	Carnegie Mellon University
DARPA	Defense Advanced Research Projects Agency
FFT	Fast Fourier Transform
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
IVR	Interactive Voice Response
IP	Internet Protocol
MFCC	Mel Frequency Cepstral Coefficients
MLF	Master Label File
OS	Operating System
pdf	Probability Density Function
WAV / WAVE	Waveform Audio File Format

Contents

1 Introduction

Information security over the Internet is one of the major concerns for users ranging from individuals to the corporate users. Big corporate houses are usual targets for a large number of server and database attacks with the intent of causing serious damage to corporate espionage. Protection of confidential and secure information is a concern for organizations dealing with such information. Any breach of data might result in the huge losses to the company's finances as well as reputation. Thus, companies that deal with confidential information, for example in the fields of banking, energy, population registrar must have a robust security system that neutralizes such attacks and breaches. An unauthorised breach to the database or server can signal such attempts. Thus, authorising the access with the use of a password is a common practice. A step forward involves the access triggered by voice controlled authentication. In this study, an aim is look to apply tools that can implement voice controlled authentication.

Multilevel security is a common choice today. Most of the network transactions are encrypted, accounts are password protected and various measures are applied for securing the system from outsiders and unauthorised users. This study is also one such instance where we are interested in speech recognition as one of the major components a multi-level security system. The case company intended to use this speech recognition for authorising the clients' access to the organizational system.

The project was supposed to be carried out with co-operation with supervisor and a private company which was interested in the idea of identity verification with speech recognition. The customers of the company were banks. Since the company has customers in many locations, it has a policy which checks different measures including user name, password, pin code, public key, IP Address and Geographic Location and allows different level of access to the data. For example, if client is using another network to access the accounts, the access is denied. The company wanted to add another feature which would use a speech recognition system that could recognize words from a limited dictionary and embed them into the existing security features. The case company decided to use a commercial product for the purpose. Losing the company for the project was a big setback; the idea is still relevant and intriguing enough to provide interesting opportunity to learn the technology. Thus, with the case company in consid-

eration, the idea of identity verification with speech recognition, this study was carried forward.

Initially the goal of the project was to model and build a voice recognition system that understands words from limited dictionary. The scope of this study is confined to the selection of such tools, preparing a training dataset, and testing the tool to practise and evaluate the performance. The major limitation is expected to lie in the small sample size and limited dictionary for the training data.

2 Speech Recognition

Humans have always tried to communicate with objects in natural language. Communication has been the integral aspect of human life; a strong tool for sharing and building the knowledge that is passed on from generation to generation. Speech, in addition to being a tool of communication, is also a symbol of identity and authorization. The conception of speech-based recognition comes from the human imagination and creativity that have been frequently used in several movies and television programs. Speech recognition-based authentication has been presented as the symbol of technological advancement as well as a secure system. This imagination of secure system combined with the superior digital and mathematical knowledge has resulted in new technologies that made such technology a day light reality. [1, 2; 2, 1-5]

Operating a new system is always a complicated procedure for a naïve user. For example, a user, new to an operating system and using it for the first time, is sure to find the procedure of using a browser following a series of step-wise procedure. A voice based command control would ease the task for the user. It would mean that the system would take commands from the user and perform on behalf of him and the only thing a user would need to know is to give commands. This has important implications as it simplifies the task of the user and hides the unnecessary complexity and operates in a reliable manner. Thus, an application with voice command would be far better performing and powerful tool to existing GUI commands making it a superior alternative. Many electronics devices can now be activated using voice commands instead of graphical user interface. [1, 2-3]

Google has been in the leading front as a leader in the field of operationalizing speech based capabilities in its products and services. Google embedded voice search tools into its products including Android, Google.com and YouTube. Google has also made search applications for other platforms including BlackBerry and Symbian OS. Most smartphones accept voice commands. A user can play music, pause it, jump to next songs, search for contact, make a call, and navigate in a map using voice commands in smartphones. The next level of completion in the smartphone markets may boil down to the superior performance of the speech based command tools. For example, the ability of creating, writing and sending messages using voice commands, with a higher accuracy rate, could be the battle winning functionality for smartphones. Google uses a service to dictate search queries in both browsers and applications in different operat-

ing systems. Saying “Hi Galaxy!” activates the smartphones manufactured by Samsung (Galaxy series), a technology developed by Sensory, Inc.[3] Google has released a new product called “Google Glass” which is controlled by voice or natural languages.[4] Microsoft's Windows 7 and 8 can be controlled with voice after some training. [5] Furthermore, there are some applications to understand speech and they are available for PC and smartphones. Mobile OSs Android, Windows Phone and iOS Platforms themselves support voice inputs. Some of the open source speech recognitions are GnomeVoiceControl [6], Open Mind Speech [7] and PerlBox [8].

The speech applications are getting better day by day. The developers are generating more data which is used further to improve the application. One article from 2010 in TechCrunch, a technology magazine, claims that the best Speech-to-Text technologies are 86 percent accurate. [9] However, it depends upon whether the user is a native speaker or not and how close the users’ language commands are to the language used in the training dataset. The accuracy is bound to increase in the future as the speech based models will be generating more data from users. These data can be used to train a model to cope with the complexity of language accent and dialects. Furthermore, with the increase in processing power and the availability of more data will only lead to improved performance. Google's privacy policy allows it to keep user data and it is processed to improve its results. [10]

The market growth of voice recognition software signifies its growing importance that cannot be overlooked. According to a report published in March 2012, the current market value of the Voice Recognition System was $49.2 billion, which was projected to reach $64.4 billion by 2015. [11] More Interactive Voice Response (IVR) systems are expected in the coming days. With the more application areas being uncovered and an improved data set being built, significant future growth could be expected in this field.

2.1 Human Speech System

Speech is produced through different parts of the mouth that creates air pressure (outside of the mouth) to change. The changes can then be sampled periodically and recorded in a digital wave form. The wave form carries all the information of the spoken word. All speech signals are produced in a similar manner. Since we can record speech signals or wave forms, one can think that it is easy to abstract the information.

Surely, the information can be abstracted but the procedure is not simple and straightforward. [1, 24]

The physical shape of a human vocal tract is different from person by person. Hence, each human speaks in a different way. [12; 13; 14] If a person is asked to utter the same word twice, the speech signal will not be exactly same as the frequency and other sound properties may differ from time to time. The environment where human speaks, the dialect of the language, differences in the vocal track length of males, female and children provide the speech variation and thus make it difficult to understand speech signals. [12] However, there are still some features in the human speech which can be mathematically modeled and used for predicting words from it but it demands tremendous amount of time and effort.

Air source is the elementary requirement to produce sounds. Humans, most of the time, produce sounds while breathing out. Sound is produced due to the obstruction of air in the organs in the respiratory track (vocal cords, vocal cavity, nasal cavity, tongue, teeth, lips, and velum). The speech produced changes the air pressure that creates wave forms. The vibrating air pressure is perceived through air and further processed by different internal organs of the internal ear and brain. [15; 16]

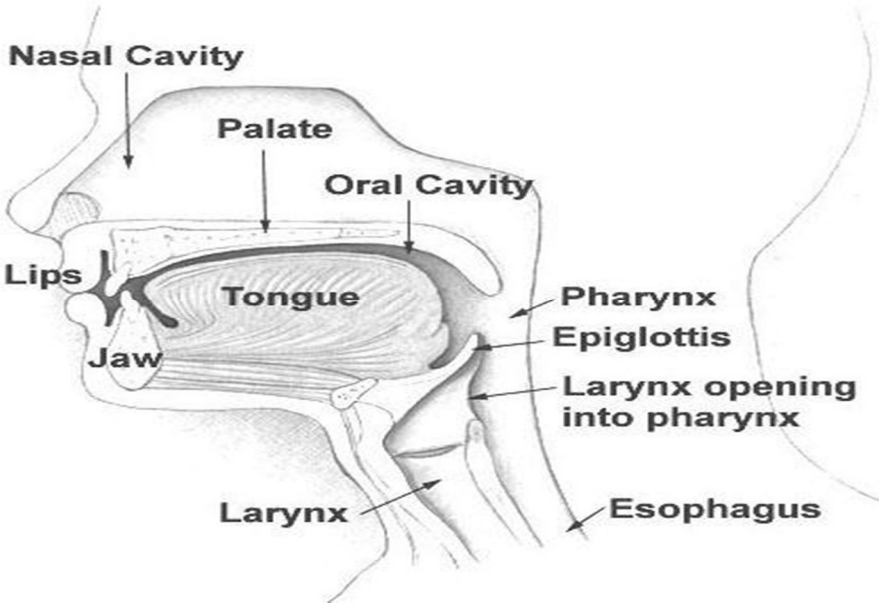

Figure 1. Human Speech Production Organs. Reprinted from Wikipedia Commons [17] verified by The vocal tract and larynx [18]

The vocal tract, a cross-section picture in figure 1, is the main organ responsible for speech production that creates resonances. The resonance depends on the shape of the mouth. The location of resonances determines the phoneme pronounced. This feature is taken into consideration to identify phonemes for speech recognition.

Every spoken language has certain basic features that are common across the languages. The smallest unit of phonetics is a phoneme. A phoneme is created by vowels and consonants. A phoneme can be diphthongs or monophthongs. The movement in the vocal tract organs causes different phonemes. Fricatives are those phonemes that require some friction of the upper tongue to palate or upper teeth to lower lips etc. A sequence of silence, burst, friction and aspiration cause plosives. The plosives are followed by fricatives, affricates are produced. [1, 39-46]

There are no distinct differences between phonemes, especially vowels. However, when the speaking process occurs, the dynamics on mouth changes. Some people speak slowly while some speak very fast. Duration of a single spoken phoneme changes as the speed of speech changes, duration of phoneme, syllabic stress, emphasis given to the word spoken, etc. [1,39-46; 12]

2.2 Limitations

Many scientists have introduced many theories or models about human speech recognition. But almost all of theories have received fair share of criticism. Even though there have been a significant number of researches being carried on speech recognition, none of those researches yet has succeeded in building a perfect speech recognition system or to be able to understand fully how speech be recognized by given speech signal. Various trial and error approaches have been used but still the best way, so far, have been data driven approaches. [12]

Most of the currently (partially) working solutions are based on data-driven training of phoneme-specific models. These models are based on the duration and phoneme of identity. The models are connected according to vocabulary constraints using Hidden Markov Model (HMM). The statistical model is the best model for speech recognition. [19]

2.3 Brief History of Speech Recognition

Humans have always been passionate about the intelligent systems. Early works on speech recognition are very limited in number. One of the early recognition applications made was Radio Rex, a toy dog activated by the word “Rex!” (basically a loud sound). More serious work began only after World War II. A recognition system to recognize digits was developed using an acoustic pattern matching feature at AT&T Bell Labs (1952). A Defense Advanced Research Projects Agency (DARPA) funded a competition (1971) of projects to develop a high-performance recognition system. The winner was Harpy developed at CMU (1980). Harpy was derived from a system called Dragon (1975), initially developed by a CMU graduate student James Baker (who used the HMMs for speech for the first time). Jelinek in IBM had developed another system using HMMs for speech simultaneously (1976). Most of the modern recognizers today use HMM. [2,919-923]

2.4 Statistical Models

Most of the speech recognition systems today use the statistical models. These systems use probability and other mathematical functions to calculate the most likely output given by the speech signals. A large set of model training data is used to calculate the features. [19; 20]

The statistical model requires acoustic modelling. Acoustic modelling is represented by the Hidden Markov Model. These models are tuned parameters with speech signals and acoustic topology. The speech waveforms (observations) are converted into numeric representation calculated using different techniques including Mel frequency cepstral coefficients (MFCC). These are all numbers. In the recognition process, the most likelihood of a sequence is calculated/searched from an already available model. The most likely word with the largest probability is produced as the result of the given speech waveform. [21; 25-26].

Figure 2 graphically explains the basic procedure for speech recognition statistically.

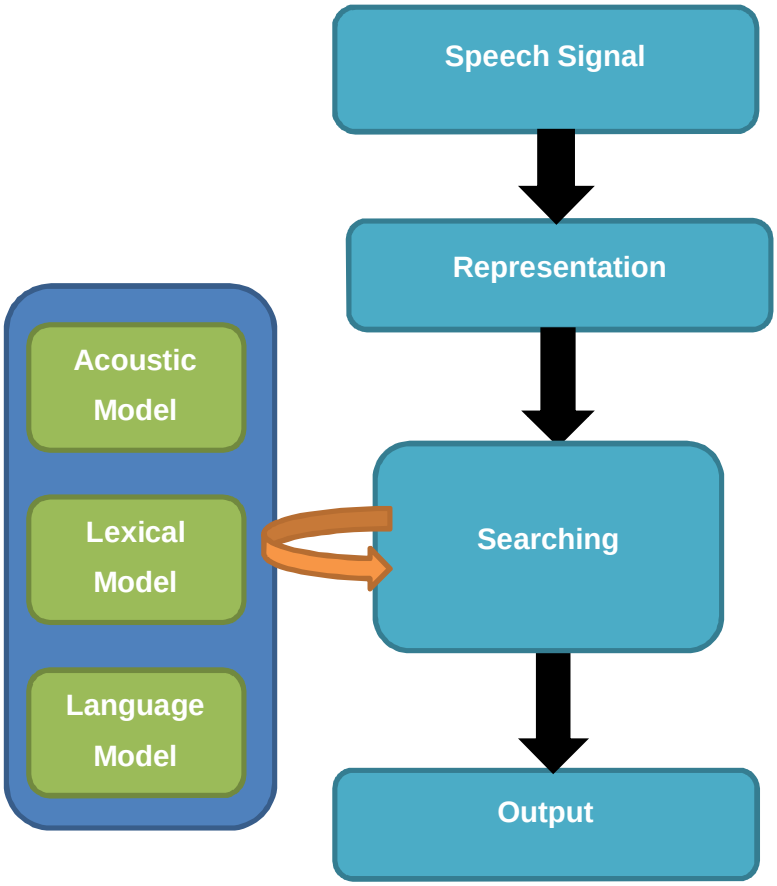

Figure 2. High-level procedure for a statistical recognition system. Reprinted from Automatic speech recognition. [22]

Some of the major theoretical aspects involved in HMM-based recognition systems are described below.

2.4.1 Acoustic Modelling

Linguists are able to find about 100 phones from the known languages from around the world that can be used to compose any verbal word. A phone is the smallest sound unit; it can be a vowel or a consonant with some exceptions. A compound of consonants can produce a single phone, for example 'ng', or sometimes a single alphabet can produce different phones, for example 'a' in 'arm', 'cat' and 'away', etc. Listing of all the phones used in American English, Advanced Research Projects Agency Alphabet (ARPAbet), has 48 phones. Appendix I contains the ARPAbet phones. [2, 914-915]

Each speech signal is sliced into frames, usually each 10 ms length. All the frames are then summarized using feature vectors. The features include an amount of acoustic energy between certain frequencies, MFCC for each frequency, total energy in the frame and other features, which are then used to train HMM. Figure 3 is an English phone model for [m]. The phone is represented by 5-state model with first and last state representing silence. [2, 915]

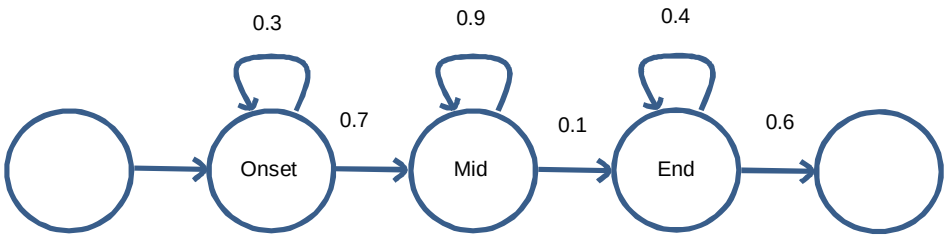

Figure. 3. Phone model for English Phone [m]. Reprinted from Artificial Intelligence: A modern approach. [2 , 916]

The average duration of the phone is about 50-100 ms, equivalent to 5-10 frames. The phone model contains self-loops to compensate the variation in the length of the phone. If some speaker pronounces “ohhh”, the self-loop allows “hhh” to fit into the model while ignoring if the loop allows just “h” (likewise in the middle state of figure 3). Each word needs to be modeled to achieve a higher accuracy. However variation in dialect and co-articulation might force to make multiple models of the words or cause inaccuracy. [2, 915]

2.4.2 Signal Processing

When a speaker dictates the word, the variation in air pressure is measured by a microphone and digitized with an A/D converter. Here, during the process of converting an analog sound into digital format, the sampling rate is an important variable. The fundamental frequency of an ordinary male is between 85 to 155 Hz and a female is between 165 to 255 Hz. In case of infants, it is between 250 to 650 Hz. [23] The applications include telephone sample speech at 8 kHz while CD recording is sampled at 44.1 kHz and DVD recording is sampled at 48 kHz. [2, 914; 25] However, for convenience higher sampling frequency is used generally. The digitized audio signal needs further

processing. Since how the words are spoken matters more than how they sound for recognition purpose, so not all the audio information is crucial for further processing.

A digital filter helps to remove unwanted information from a signal. In a real spoken system, each vocal tract organ acts as a filter. While modeling speech organs phoneme classification depends on the digital filter. The filtered signal is applied to a window function. A window function allows separating a particular slice of the signal. Usually different features then are abstracted from the fixed-length signal. [2, 916]

The fetched features are used to build a model or recognize words from given observations.

2.4.3 Hidden Markov Model

Hidden Markov Model (HMM) is used to predict or analyze time series using probability. Whenever a time series is used the HMM can easily be applied. Most of the intelligent systems use HMM extensively. Robotics, medicine, finance, machine translation and speech recognition are some examples. [24]

In probability, two events are independent if the first event does not affect the outcome of the second event or vice versa. In contrary to independent events, one event affects the outcome of other events in dependent events. Markov invented a stochastic process called Markov Chain, also known as (Simple) Markov Model, where each state is dependent on a fixed number of previous states. The common and simplest Markov Chain, First Order Markov Chain (First Order Markov Process), is the chain where current state depends only on the previous state. The current state is enough to give (probabilistic) future conditionally independent of the past. [1, 378-380] Figure 4 represents a two-state Markov chain with transition probabilities a_{ij}.

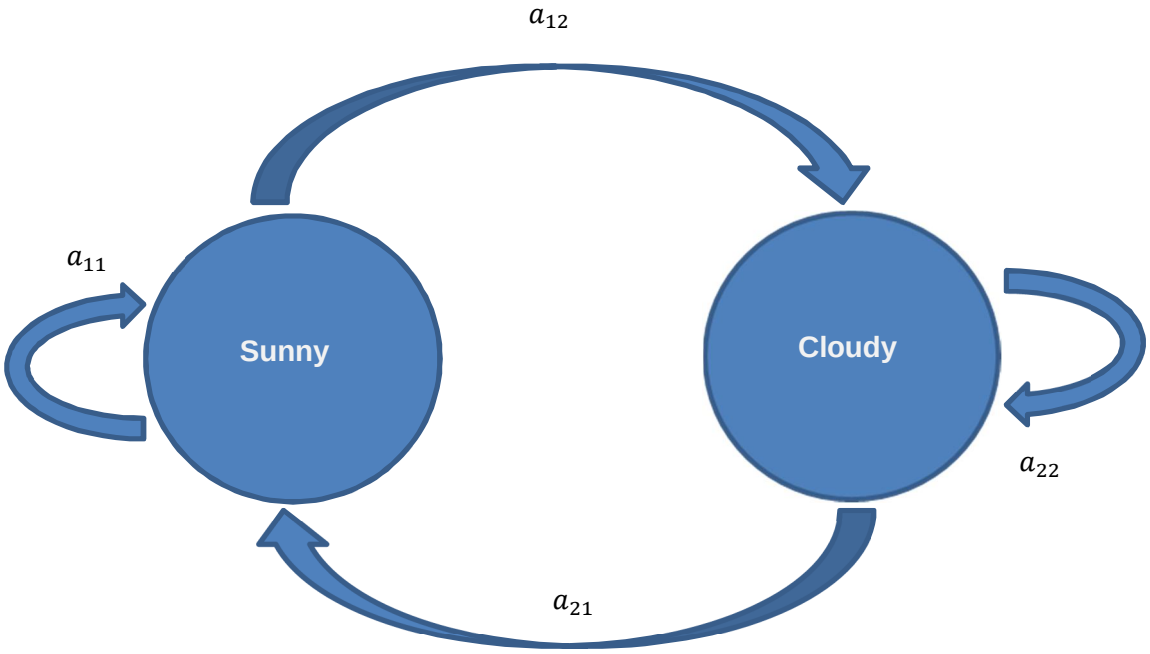

Figure 4. Example of Markov chain representing transition probability of the weather of the following day given the probability of present day.

In the example, given the initial distribution (π), the probability for any number of sequences of states can be calculated. For example, let us assume the initial probability,

$$\pi_i = \begin{bmatrix} P(Sunny) \\ P(Cloudy) \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \text{ and } a_{ij} = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix};$$

then the probability of three consecutive sunny days can be calculated in the following way:

$$P(Sunny, Sunny, Sunny) = \pi i \times (0.8)^3 = 0.3584.$$

Markov chain is deterministically an observable event. Many real-life applications have a feature that is not deterministic. A natural extension of Markov chain is Hidden Markov Model (HMM), the extension where the internal states are hidden and any state produces observable symbols or evidences. The observable symbols are random variables and the probabilistic function of the internal stochastic states. This model is known as HMM. [1, 380]

The use of HMM in speech recognition and HMM itself is not a new concept. The concept of HMM was presented by L.E. Baum and Petrie in late 1966. [2, 604] Figure 5 is the HMM version of figure 4.

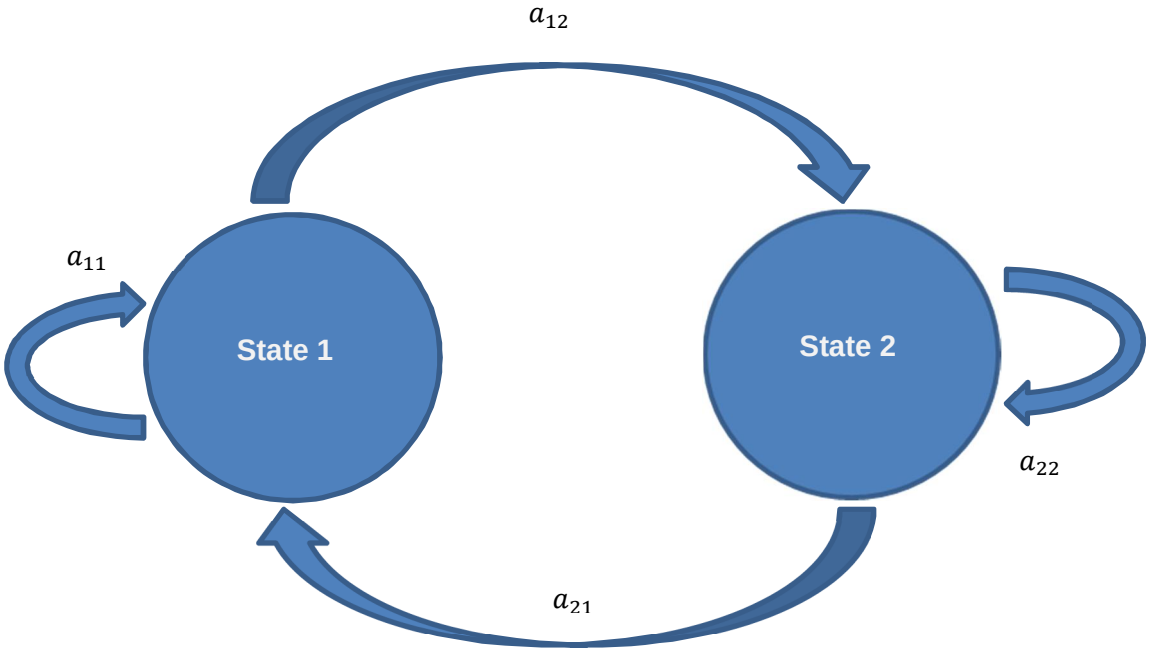

Figure 5. Hidden Markov Model version of figure 4.

In the example above, the states are not visible. An extension to this example could be, if a person likes sunny sunny day and hates the cloudy day, to observe the happiness or sadness of the person on that particular day and guess the hidden internal state.

In HMM, since there is no direct correspondence between outputs to states, the sequence of states cannot be produced given the sequence of outputs. Mathematically Hidden Markov Model contains five elements: [1, 380-383; 25]

i) Internal States ($S = \{ 1, 2, 3, .., N \}$)

In this sample space, each state is noted as s_t. These states are hidden and give the flexibility to model different applications. Although they are hidden, usually there is some kind of relation between the physical significance to hidden states. In the example provided the model assumes that there are two states, a day being sunny or cloudy.

Figure 6 is an example of HMM where blue circles represent hidden (sunny or cloudy) states.

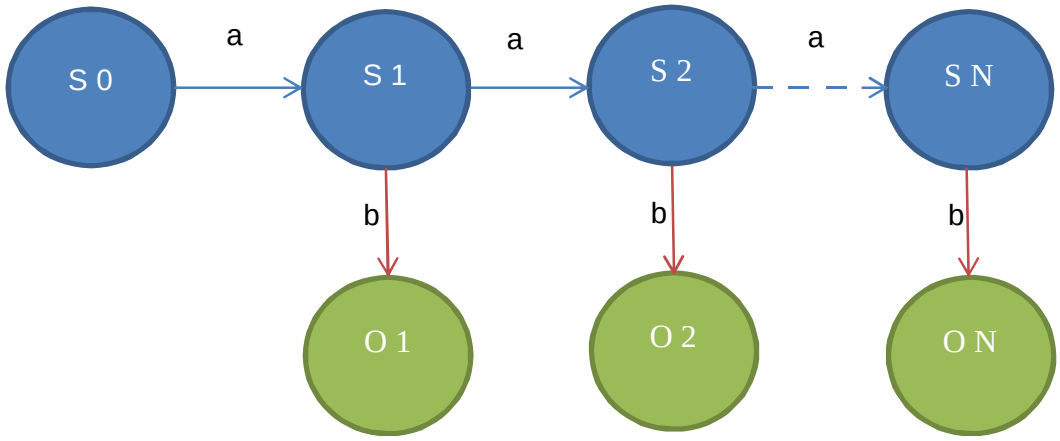

Figure 6. Hidden and Observable states in Hidden Markov Model.

ii) Output ($O = \{o_1, o_2, o_3, ..., o_m\}$)

An output observation alphabet. In the example in figure 6, somebody being happy or sad based on the weather corresponds to the observation alphabet, circle in green in the example figure.

iii) Transition Probability Distribution $A = a_{ij}$ is a matrix. The matrix defines what the probability to transition from one state to another is

$$a_{ij} = P(s_t = j \mid s_{t-1} = i).$$

iv) Output Observation Alphabet. Probability Distribution $B = b_i(k)$ is probability of generating observation symbol ok while entering to state i is entered. Mathematically it can be expressed as follows:

$$b_i(k) = P(O_t = o_k \mid s_t = i).$$

v) The initial state distribution ($\pi = \{\pi_i\}$) is the distribution of states before jumping into any state.

π_i = P(s_t = i) $1 \leq i \leq N$.

Here all three symbols A, B and π represent probability distributions. So they must satisfy the property of probability.

$a_{ij} \geq 0$, $b_i(k) \geq 0$, $\pi_i \geq 0$ for all i, j, k.

Similarly,

$$\sum_{j=1}^{N} a_{ij} = 1 ,$$

$$\sum_{k=1}^{M} b_i(k) = 1 ,$$

$$\sum_{i=1}^{N} \pi_i(k) = 1 ,$$

The probability distributions A, B and π are usually written in HMM as a compact from denoted by lambda as

$\lambda = (A, B, \pi)$.

In the HMM based speech recognition, HMMs are used to represent phones. Figure 3 is an example of phone level HMM.

2.4.4 Gaussian Distributions

Researchers and scientists have found that most of the distribution involving random variables is very close to Gaussian. A continuous random variable X with mean π and variance σ^2 has a pdf in the following form:

$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$ [1, 92].

Figure 7 is an example of Gaussian distribution, also known as normal distribution.

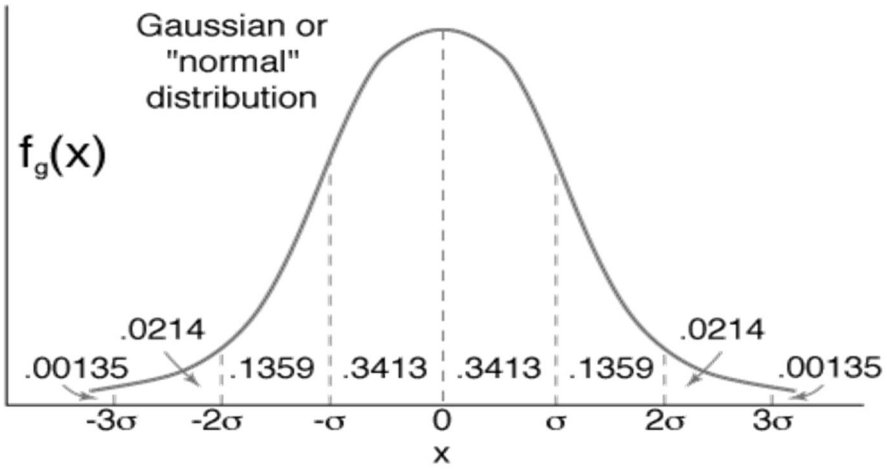

Figure 7. Gaussian distribution where each floating point value is referring to probability for the random variable for that region. Reprinted from Gaussian distribution function [26]

The 68 percent of the distribution of X lies in an interval of ±1 standard deviation (σ) about its mean value μ. [27] For the n-dimensional continuous random vector *X = (X1, X2, …., Xn)*, the multivariate Gaussian probability distribution function (pdf) has the following form:

$$f(X = x|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]$$

where $\boldsymbol{\mu}$ is the n-dimensional mean vector, $\boldsymbol{\Sigma}$ dimensional is the $n \times n$ covariance matrix and $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix $\boldsymbol{\Sigma}$. [1, 93]

In an HMM-based speech recognition system, the mixture of Gaussian is used to output the distribution of HMM.

3 Methods and Materials

Initially the main idea of this project was to build an identity verification system with speech recognition (not speaker recognition). The project was not carried out as a case study focused on the requirement of one particular company. However, sticking to the initial idea, the project was carried out considering the requirements stated by the company, but not for the company. The company wanted a recognition system that could understand digits. The company already had a credential management system based on user name, password and other. They wanted to use a smartphone with recording capability, so that they could record the voice of a user (a special passcode) and send it to server for verification where the speech was recognized. The access would be given if the user's speech was recognized.

3.1 Tools Used

Different commercial products on the speech recognition are available in the market today. Vlingo, Sensory, Naunce are some of the companies that provide commercial product for speech recognition. A few open source tools like Carnegie Mellon University (CMU) Sphinix, Spraak and Hidden Markov Model Toolkit (HTK) are freely available in the internet. The open source tools were best for project. HTK is one of the most widely used tools for speech recognition research and teaching-learning, including Aalto University which made me to choose it.

Hidden Markov Model Toolkit

The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models. HTK is primarily used for speech recognition research although it has been used for numerous other applications including research into speech synthesis, character recognition and DNA sequencing. HTK is in use at hundreds of universities worldwide.

The HMM Toolkit was originated in Machine Intelligence Laboratory in the Cambridge University Engineering Department. In 1993, the rights to sell and develop HTK was acquired by Entropic Research Laboratory, Inc. Microsoft bought Entropic in 1999,

hence acquired rights on HTK. However HTK was licensed back to Cambridge University Engineering Department with the support to redistribute it. [28]

Figure 8 illustrates the complexity and architecture of the HMM Toolkit.

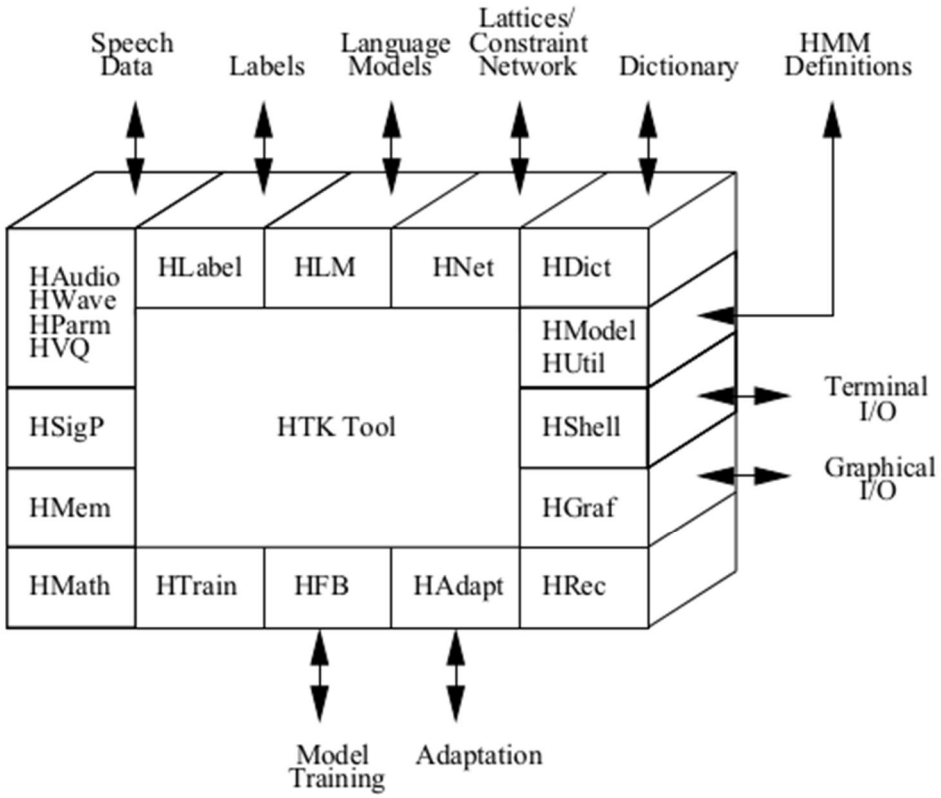

Figure 8. Architecture of HTK Tool. Reprinted from HTK Book [29, 15

The C-source code is available after registration and accepting the license term and conditions. A currently available stable release is version 3.4.1. The HTK needs to have a C compiler. The GNU C compiler works perfectly. Some Perl scripts, a how-to book are also included in the HTK Package. The scripts are intended for automating text processing and other helper tasks. However the core functions are in C. The tools, visualized in figure 8, and its sub-tools, are generated when compiled.

The HTK is a collection of different tools. It does not have Graphical User Interface (GUI) but runs on the traditional command-line interface. The developers believe that the command-line interface has more advantages over the modern GUI. The main advantage is that the command-line interface allows writing shell-scripts to execute HTK commands allowing automating large-scale systems. It speeds up the training and testing experiments. It also helps to record and document the entire process. The HTK

tools can be divided into four categories based on their functions: data preparation, training, testing and analysis tools.

The requirements of the recognition systems vary in practice. Some applications include continuous speech recognition from a large set of vocabulary while some applications needs to recognize discrete (isolated) words. The data preparation largely depends on kind of application being made.

Figure 9 sums up the process for isolated words recognition process.

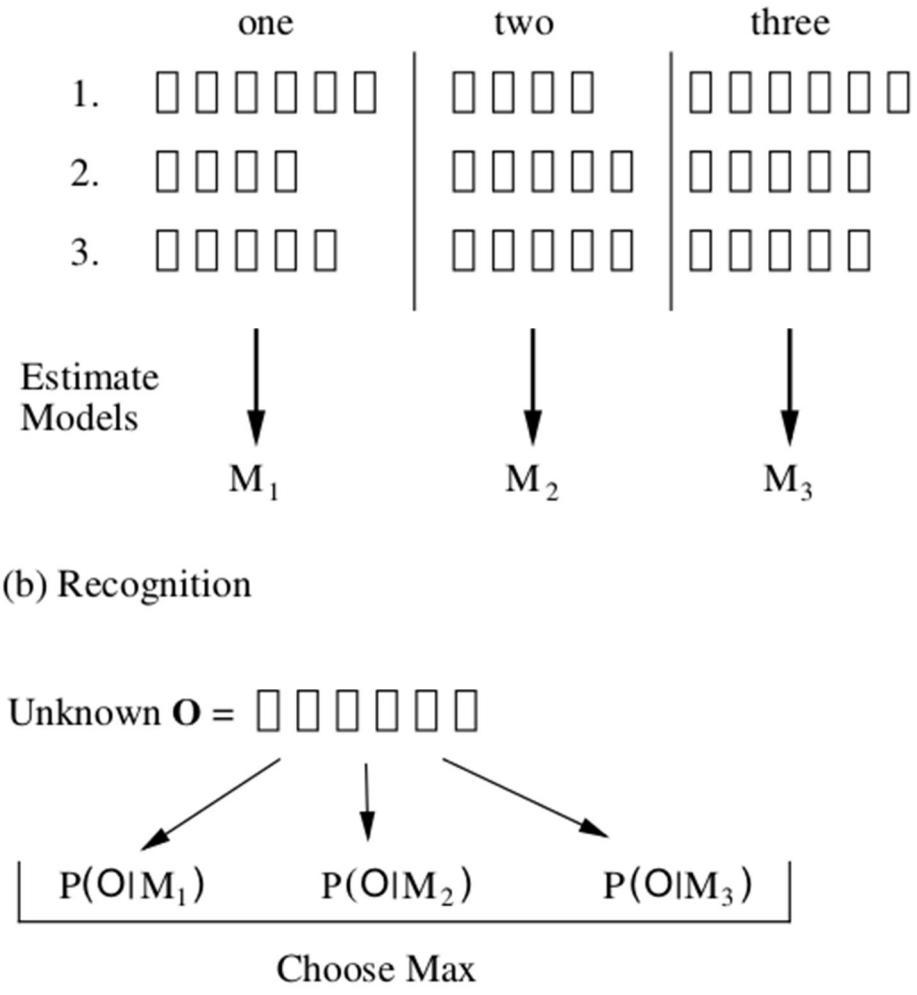

Figure 9. Using HMMs for isolated word recognition. Reprinted from HTK Book [29,5]

First, the model of the application must be well defined so that it functions like the grammar for the application. The grammar is basically a set of rules that include what kinds of words and sentences are expected for the application. Then the sample sen-

tences, which follow the model, are generated. For the complex systems the word list is built using the words in the training data. The dictionary is a sorted list of words the required for recognition process. Since the recognition process is a data-driven approach, more data, phonetically balanced words and sentences always result in better training and, hence, produce better results. The transcript of utterances for training data should strictly come from the word-list required for the application. The system needs a dictionary of words and their phonetic equivalent symbols. For this purpose, each word in the dictionary should be phonetically transcribed. If available, a large dictionary can be used to generate a sub-dictionary for the word list. If the data set is small, the training data can be generated using the HTK tools. The HTK tools and scripts are helpful to generate the phone-level transcription of the training data. Of course the recording of the transcription is needed in a lossless format. [29, 33-51]

The training phase begins when all the data is ready. The HTK itself is Hidden Markov Model training kit. The user of HTK does not have to take care of the HMM model while training it. However parameters can be changed according to the needs. HMM training includes processes of creating a flat prototype, and then the prototype is used to create a new version of prototype with *HCompV command* for the first time. Later most of the training is performed using *HERest* command which loads the training files into the memory and uses the associated label files to construct a composite HMM of utterances. It uses the Forward-Backward Algorithm to estimate the values and outputs the updated HMMs. The training process includes adding the silence models, aligning training data, making triphones and finally tied-state triphones. [29, 24-43]

The recognition command *HVite* is used to recognize words, which requires the model and codded speech as its parameters and outputs the results in various ways. The output can be transcripts of the recognized words. [29, 43-45]

Audacity

Audacity is a free, open source for recording and editing audio files for Windows, Mac and Linux platforms. The computer used was not compatible with the sound drivers to run HTK Tool to record the speech data. It has all the required functionalities to edit the sound files. It is possible to change the parameters to record the voice with desired sampling rate and quantization bits. It can export the recorded sounds to most common type file format used nowadays.

3.2 Design Process

No prior experience in the field of the project made it difficult to get started. I spent a huge amount of time on reviewing theory and literature. All the information was found using Google.

I had some options available to choose an available set of tool kits from CSLU Toolkit, SRI Language Modeling Toolkit or HTK Toolkit. Many well-renown universities used HTK for modeling languages. The documentation of HTK was easy to follow. It not only introduced the tools but also explained the underlying theories for each tool making it easier for beginners with examples. I followed the example provided in the book for the process described in section 3.2.

Requirements

The goal for the project was to model and analyzes the system with limited words, related to the numeric system, i.e. 0-9.

Model / Grammar

The application had to recognize some repeated numbers of a fixed length. HTK requires to have model the possible recognition pattern. The modeling was not difficult. The dictionary had only 10 words, which could be repeated. The length of the sequence or number repetition was not stricted. But for this project, I have used sequence of five digits. So the grammar was quite simple, as shown in listing 1.

```
$digit = ZERO | ONE | TWO | THREE | FOUR | FIVE | SIX
| SEVEN | EIGHT | NINE;

(SENT-START $digit $digit $digit $digit $digit SENT-
END)
```

Listing 1. The grammar.

In the listing 1, SENT-START and SENT-END mean silence before and after the speech. The HTK tools require an HTK Standard Lattice Format file. HTK command, *HParse*, then was executed to create a word network.

Ten sample sentences were written into a file sample_sentences.txt.

Dictionary

A Perl script prompts2wlist to create a word list from a file is available within HTK Package. Running the script, shown in listing 2,

```
prompts2wlist sample_sentence.txt word_list.txt
```

Listing 2. A Perl script to generate words from given sample sentences.

generates an alphabetically ordered list of words which are in the sample sentences. The word_list.txt does not include SENT-START and SENT-END words used in the grammar. The two words need to be manually inserted into the word_list.txt in alphabetical order.

HTK tools needs a list of phonemes and a word-to-phoneme dictionary. The command 'HDMan' can create a phoneme list and a word-to-phoneme dictionary for limited words given a word-to-phoneme dictionary. The dictionary was saved as english_phonemes.txt. The command 'HDMan' creates the monophone list as well as dictionary. However the command needs the HTK “internal commands” to perform the task and add pauses and silence into the phonemes. The internal commands, shown in listing 3, are stored on *global.ded* file.

```
AS sp
RS cmu
MP sil sil sp
global.ded
```

Listing 3. The internal commands inside *global.ded* file.

```
HDMan -m -w word_list.txt -n monophones1 -l
log.dictionary_created dictionary english_phonemes.txt
```

Listing 4. The HDMan command

The command, shown listing 4, creates three more files where *log.dictionary_created* is a log file, *monophones1* is list of phonemes and dictionary is dictionary of word-to-phonemes. The log file contains the statistics of phonemes in the dictionary.

The application will be able to recognize only 10 digits. So instead of recording large vocabulary and transcribing it, the command *HSGen*, shown in listing 5, will be used to create 100 transcriptions to be recorded.

```
HSGen -l -n 100 wordnet.slf dictionary > train-
ing_prompts.txt
```

Listing 5. Command to generate the prompts to be recorded.

The audio files could not be recorded using the HTK tool in the computer because of the incompatibility of the sound driver. So they were recorded using a free and open source tool Audacity. The files were stored in “Records” and stored using waveform audio file format (WAVE / WAV).

The transcription file training_prompts.txt should also be converted into Master Label File *(.mlf)* format. The *.mlf* file can be a single file or can be multiple files equivalent to the number of audio recordings. I have stored them in single file for easiness. Another Perl script, shown in listing 6, to convert from *training_prompts* file to *transcripts.mlf* is also available.

```
prompts2mlf transcripts.mlf training_prompts.txt
```

Listing 6. Command to convert transcripts into *mlf* format.

The output *(transcripts.mlf)* is the word-level transcript of training prompts. The HTK tools need a phone-level transcript of every sentence in *training_prompts.txt*. The command *HLEd* creates the phone-level transcripts. Similar to an *HDMan* command, it

also requires to have a file with internal HTK commands. These commands are stored in *make_phone.led* file which are shown on listing 7.

```
EX
IS sil sil
DE sp
```

Listing 7. The internal commands required for *HLEd*.

Where *EX* replaces each word in *transcripts.mlf* to phone-level transcription. *IS* puts the silence phoneme model `'sil'` to the beginning and end of each word and *DE* removes the short-pause models from the words if there are any.

```
HLEd -l '*' -d dictionary -i phone.mlf make_phone.led
transcripts.mlf
```

Listing 8. Command HLEd with all required parameters.

The command produces phone-level transcriptions on *phones.mlf*.

Recording

The sound driver in the computer that was used did not support to record using HTK command *HSLab*. This led to record the training and test voice using Audacity. The sound files were recorded in 48000 Hz as a sampling rate and was stored on a 16-bit lossless WAV file format.

All the generated test prompts were recorded first and exported to wav format. They were checked multiple times to ensure the correct audio files were produced. It was a tedious process.

The command *HCopy* copies the audio data into sequences of feature vectors. The feature vectors are Mel Frequency Cesptral Cofficients, the log-spectra of FFT. These vectors are used to train the HMM model. The command requires a script file *code-train.scp* contains the locations of raw audio files (wav files) and location for output files. The config.1 file contains the parameters for the command *HCopy*.

Training Process

The training begins with defining a prototype model for topology. For the this part, I used the model from HTK Book. The model was saved as proto file. The topology is 3-state HMM left to right right with no escape.

The HTK command *HCompV* scans the data files (MFCCs) and computes the global mean and variance set all Gaussians in the given HMM to have the same mean and variance.

```
HCompV -C config.2 -f 0.01 -m -S train_mfcc.scp -M
hmm0 proto
```

Listing 9. Command *HCompV* with all required parameters.

The command in listing 9 creates two files in the *hmm0* directory. One is new version of *proto*, and another one *vFloors*. The proto is used to model each phone's HMM and also define macros.

The HMM for each phonemes except “sp”, a flat-start-monophone is defined in the *hmmdefs* file with the prototype in a newly created *proto*. The HTK tool HERest is executed using the command in listing 10 to re-estimate the *monophones1*.

```
HERest -C config.2 -I phones.mlf -t 250.0 150.0 1000.0
-s train_mfcc.scp -H hmm0/macros -H hmm0/hmmdefs -m
hmm1 monophones0
```

Listing 10. Command *HERest* with all required parameters.

Running the command re-estimates the *monophones* in another directory. This command is run again two times changing the input and output directories. The final result is stored in the *hmm3* directory.

The model is further developed by adding the short-pause model sp. The contents of the *hmm3* is copied to *hmm4* and *hmmdefs* is edited by copying the *sil* model and making it sp. The HTK Tool *HHEd*, shown on listing 11, can make changes to the model by executing commands on a separate script file.

```
HHEd -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed
monophones1
```

Listing 11. Command *HHEd* with all required parameters.

The *sil.hed* script, listed in Listing 12, contains commands to change the model by given parameters.

```
AT 2 4 0.2 {sil.transP}
AT 4 2 0.2 {sil.transP}
AT 1 3 0.3 {sp.transP}
TI silst {sil.state[3],sp.state[2]}
```

Listing 12. Internal Commands in *sil.hed* file.

The new model is now stored in the *hmm5* directory .The model is further re-estimated with the command *HERest* twice more, now using the *monophone1*. The results are stored in *hmm7* by now.

Input word-level transcription, *transcripts.mlf,* is transformed into new phone-level transcription called *aligned.mlf* by using the currently available version of HMMs stored in *hmm7* and *dictionary*. The output, resulted by executed command in listing 13, is improved from the previous transformation of transcripts.mlf into phones.mlf.

```
HVite -l '*' -o SWT -b SENT-END -C config.2 -a -H
hmm7/macros -H hmm7/hmmdefs -i aligned.mlf -m -t 250.0
-y lab -I transcripts.mlf -S train_mfcc.scp dictionary
monophones1
```

Listing 13. Command *HVite* with all required parameters.

The HMM is re-estimated twice more, using the *HERest* command using the newly created *aligned.mlf*.

The current monophone HMM model needs to be transformed into triphone HMM. Triphones are context-dependent models. This can be done with command shown in listing 14. The HLEd command needs a script shown in listing 15.

```
HLEd -n triphones1 -l '*' -i wintri.mlf mktri.led
aligned.mlf
```

Listing 14. Command *HLEd* with all required parameters.

The command generates *wintri.mlf* from *aligned.mlf* with the editor script *mktri.led*.

```
WB sp
WB sil
TC
```

Listing 15. Internal Commands required by *HLEd*

The existing HMM model needs to be cloned using mktri.hed. *mktri.hed* can be generated with a Perl script, shown in listing 16.

```
maketrihed monophone1 triphones1
```

Listing 16. Perl script to make triphones

The HMM is once more transformed using *mktri.hed* file.

```
HHEd -B -H hmm9/macros -H hmm9/hmmdefs -m hmm10
mktri.hed monophones1
```

Listing 17. Command HHEd with all required parameters.

The model is re-estimated again twice using *wintri.mlf* and *triphones1*.

The *fulllist* file contains all the *monophones* and *triphones* created. The *tree.hed* is the script that contains all the contexts to test for the clustering of the data. The script contains English phone models. For this purpose the rules were copied from Julias website. [30]

The final time The HMM is re-estimated using newly created *tiedlist* twice. The model is now ready for recognition. The HTK command, HVite, was used for the recognition.

4 Results

The speech samples for the training data set were collected from direct observations. First, three respondents were chosen based on random sampling :-) and asked to pronounce the sequence of five digit numbers that formed the code used for authentication. The code consisted of five digit numbers between zero and nine pronounced in the English and individually. These responses were recorded to produce training data sets that would be used to train the model. Using the training data the model was trained. These speech data were used to train HMM models in three different ways. Once the model was trained, it was tested with the actual testing data. Again testing data was collected from a sample of seven respondents which included respondents from the training data set as well. The responses were tested with ten different sequences of digits. All of the respondents were non-native speaker of English. Furthermore, the transcription for the recorded data was auto-generated. To include the heterogeneity of the dialect and the accent, test data included speech from native Nepalese speakers and a Finnish speaker and both male and female speakers. The details of data used in this study are summarized in table 1.

S.N.	Respondent/ Speaker	Sample Count		Mother Tongue	Sex
		Training	Testing		
1.	Bikesh	50	10	Nepali	Male
2.	Janaki	100	10	Nepali	Male
3.	Krishna	-	10	Nepali	Male
4.	Prashamsha	-	10	Nepali	Female
5.	Purushottam	50	10	Nepali	Male
6.	Sakari	-	10	Finnish	Male
7.	Udeep	-	10	Nepali	Male

Table 1. The demography of the respondents.

In the first step, the training data set included 100 different 5-digit codes from a single speaker, Janaki. The model was trained using this data set and then tested with 10 responses from each of the respondents. The test result indicated the case of under-

sampling in the training data set. Thus, it was decided to increase the size of the training data set for the second test run. An additional 100 speech data were added, making it a total of 200, to the training dataset from two more respondents; 50 each from both Bikesh and Purushottam. Then, the model was trained with the new data set of 200 new speech data points and then tested with all the available test data sets. The final two models were trained with 50 speech data from both Bikesh and Purushottam then was tested and compared with the first model.

The following formula was used to calculate the accuracy.

$$Accuracy = \frac{Correctly\ recognized\ words}{Total\ number\ of\ words} \times 100\%$$

First Test

The trained HMM was tested for all the seven test sets later. The outcome was a somewhat mixed type. Nine out of ten sample sequences from Janaki were correctly recognized. Two digits in a sequence were incorrectly recognized. The digit recognition was 96 percent accurate.

However the sample sequences from other speakers than Janaki produced poor results. Not a single sequence was recognized correctly. However some of the digits in the sequence were correctly recognized. The histogram in figure 10 explains the results.

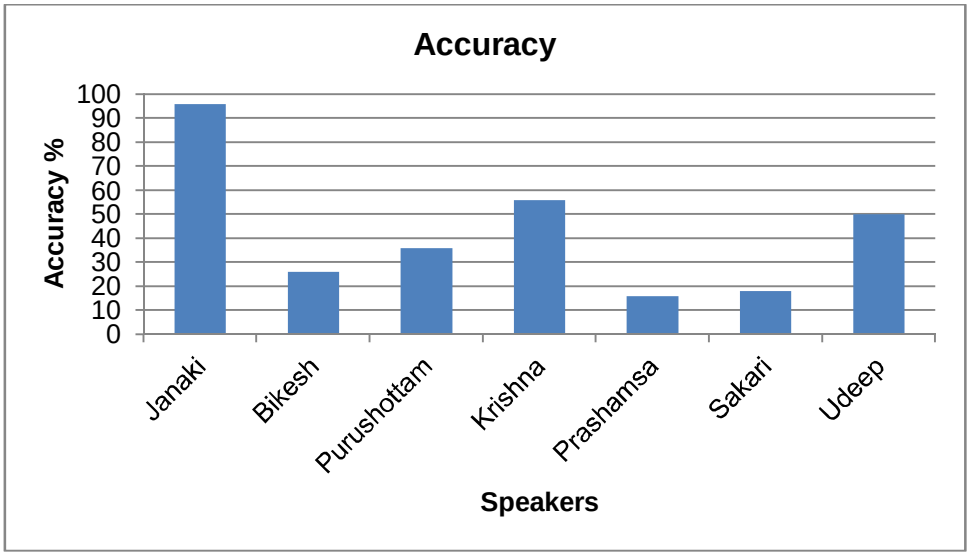

Figure 10. The graph for the first test.

The accuracy of results for the speakers except Janaki was between 16 to 56 percent. The results for respondent Prashamsha, Female Speaker, and Sakari, Non-Nepali Speaker, was the poorest, with the accuracy of 16 percent and 18 percent respectively. The word digit “SIX” was repeated eight times throughout the sample. Out of eight recognized digits, six were the digit “SIX” for Prashamsha and out of nine correctly recognized word digits for Sakari, eight were “SIX”. The result “SIX” was produced for all the 10 digits (0-9) for both of the speakers.

Second Test

All the availalbe speech data for training (200 sequences or 1000 words altogether) were used to train the HMM. The trained HMM was used to test all the available samples of seven speakers. The accuracy now varied from 38 to 76 percent. The results compared with first test are shown on figure 11.

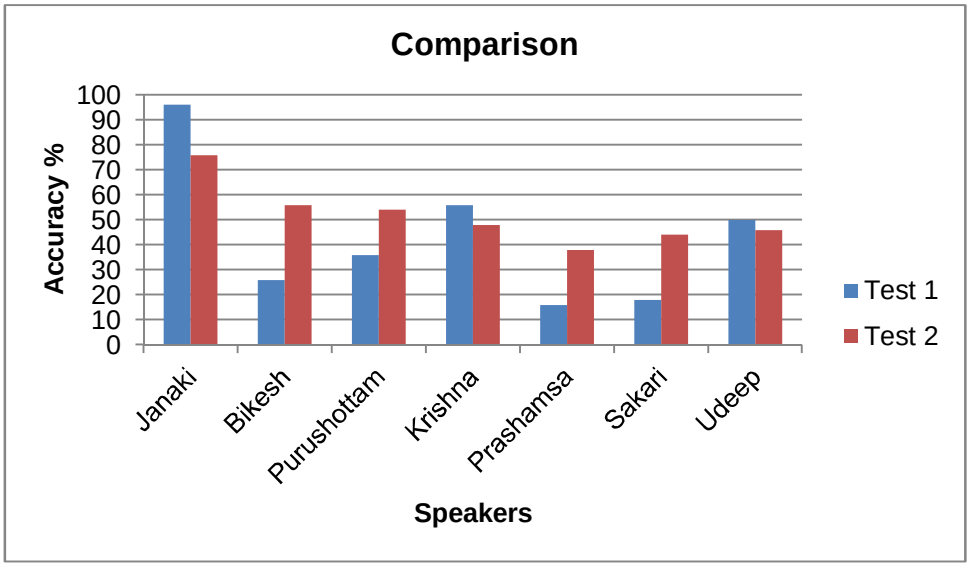

Figure 11. Bar graph comparing two results produced from the first and second test.

The results dropped from the previous test for Janaki, Krishna and Udeep, while improving other speakers' accuracy. The accuracy for Sakari and Prashamsha were improved by 22 percent and 26 percent respectively.

Third Test

The speech data from respondent Purushottam and Bikesh was used to train HMM individually. Only 50 sequences were available for both speakers compared to 100 sequences for Janaki for first test. The accuracy was 72 percent and 88 percent respectively. The recording environment for sample and test speech was different for respondent Purushottam. The pie charts in figure 12 show the results for individual respondents.

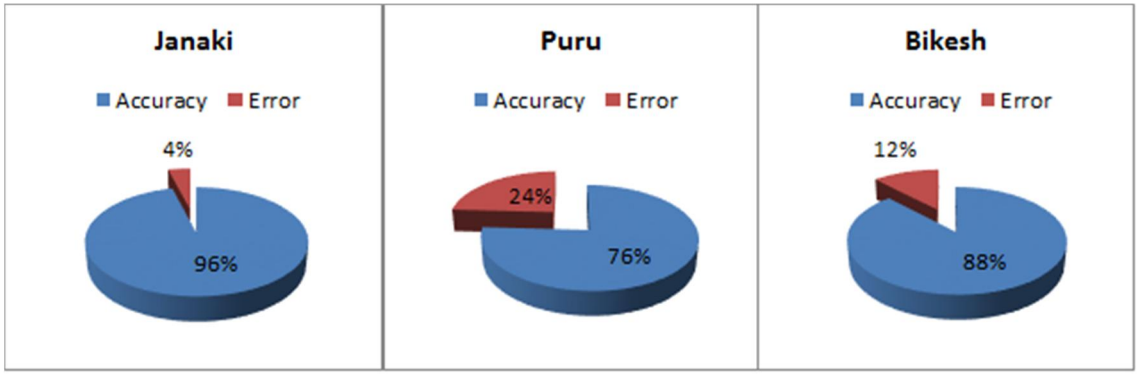

Figure 12. Pie chart for accuracy of words for respondents

Each pie chart shows the accuracy versus error for the test performed on a model that was trained and tested using the individual's speech datasets.

5 Discussion

The tested model of identification offered several advantages. However some limitations were also imminent. First, the model worked quite well with high accuracy. However, the accuracy was the function of the speaker recognition; i.e. when the same speaker's speech data was used for training the HMM and testing the result, the accuracy was 96 percent accurate. The accuracy of 96 percent was achieved by training only 100 sequences of digits (500 digits). When the model was tested from other respondents' speech data, the accuracy of the results dropped. This can be well explained by the variance in the speech of two different speakers. Even though there is a shortcoming in the model, the concept can still be implemented for the system required.

The clients who have the right to authenticate with the system, would record hundreds of digits or words to train the model individually. A mobile application could be used to train the model first, showing fixed number of sequences (say 100) of digits or words. A database should be created of trained HMM for each client. If enough data is used, a higher accuracy can be achieved. For the authentication, the coded voice sample from the user, for randomly generated digits, is matched against the database to authenticate. Furthermore, building the training dataset from real client data allows for the customized authentication and model learning process based on each individual client. The required size of the training dataset would be directly proportional to the desired level of accuracy. The least accuracy required should be carefully chosen. This requires large-scale testing. An accuracy higher than the calibrated number can be determined for positive authentication. For example if the accuracy is greater than 95.0 percent, the request is authenticated otherwise rejected. Since the system was supposed to be embedded in a multi-level authentication system, the database could be searched for who the client claims to be, reducing the computational power. Faking the authentication could be very difficult if the approach described was used. Clearly, there exists tradeoff between the accuracy and security against efficiency of the model. Furthermore, the system requires an alternative authentication process when the client is in an adversary situation, for example if there is damage to the voice due to sickness.

Modeling a limited dictionary and training HMM for it is much easier modeling a larger dictionary (for continues speech recognition). The test also showed another application of the limited dictionary recognition system. If larger set of training data is used to train

the model and a sample for the third speaker is recognized from the model, the accuracy increases. By training model for limited dictionary from large training set, the application can be used in gaming. For example, for a kick-boxing game, the words like “kick”, “sit”, “stand”, “punch” could be used. This allows a voice controlled gaming. This kind of system works not only for gaming, but major home appliances. A light could be switched on by saying “on” or turned off by saying “off”. Small robots could be commanded from voice.

Native speakers speak very similarly. In the first test, respondent Krishna's sample had 56 percent accurate results querying over trained HMM by respondent Janaki's speech. International clients could be encouraged to use their local language for authentication. However, this would require to model acoustic features for multiple languages.

The major advantage of the application would be simplicity to use it. If a mobile program was developed implementing the authentication system, a single button would be enough to verify the authentication. The application can be used as a spoken password. The potential user will be given a fixed-length number as a password, that could be tested against the accuracy of as well as the spoken content to verify the identity.

Despite the fact that the model offers decent performance to the given problem, there are some limitations within this study that one needs to consider when implementing the model. First, the results are based on a sample size that was relatively small. To rephrase it, the speech data points were too few for testing a real life application. If the application is to be used for security purposes, it requires much more testing, data, attention, specialists and a big fund. The populations in this project giving samples of speech were friends who were easily reachable. The speech data was taken in random locations, some were quite noisy. Several experts including linguists, statisticians, signal processing specialists, artificial intelligence specialists and programmers would be required for real application. However, the results produced lay down basic foundation for building such model.

For the project of this nature, the number of respondents and the training dataset were too small. A larger-scale implementation may have yielded different results. The results have been hardly tested and security authentication should be very robust. A minor wrong evaluation would result unimaginable loss. To prove something statistically

needs hundreds of gigabytes of data so that the test could be statistically tested. One possible option described in second paragraph in Discussion (section 5) supposes that trained word models for each client could be made. The company might have an idea which requires a single trained model. This might lead to less accurate recognition and might not be applicable to the system.

The automated voice initiated model of authentication inherits some basic fallacies in themselves from the conception. For example, users may find it difficult to verify their identity if there is some modulation in their voices due adversaries like sickness, noisy environment, etc. Researchers and model developers should consider such circumstances and thrive for preparing an alternative method of authentication to address such adversaries.

Security is one of the most important issues these days. In comparison to the original problem of security, the application is too simple, and the application was based on a numeric model only. Hence, the level of security is not high enough for a real-life application to be used in critical processes. A more robust model would allow the researcher to build in alphanumeric codes. However, one cannot deny the utility of this simple model, and with some additional effort, this model could be extended to make the system secure. This extension issue will be discussed in the next section.

Further Developments

For the project, the digit-pronunciation was limited to the English language only. The model could be further developed to incorporate digits from other languages; most preferably the local languages of the clients. Furthermore, the dictionary size could be increased using alphabets, so the large test data could be generated and trained.

In the second test, the application was trained with a male Nepali language speaker. When the test speech samples from a female and a non-Nepali speaker were tested, the accuracy was dropped to less than 20 percent. It would reduce the accuracy when the speaker was not from the same language group.

Since the security was involved in the application, it could be more secure if there existed a hidden pronunciation for each word in the dictionary. For example, “one” could be pronounced with as “hello”. This concept is not very secure if the digits are visible to

the 'hearer' and could easily be decoded. However if the digits would expire on a fixed time, this would add a little security feature.

It was discussed in the chapter 2 that each human has different voice finger prints, which could be utilized as speaker recognition. The concept of identity verification would be more concrete if a speaker recognition system was applied. Building customized training data set and then applying an individual level security system would increase the security provided by the model.

6 Conclusion

The goal of the project was to test a model for speech recognition that could be used for authentication of client's access. Some free tools were available for simulating the application. HTK tool was chosen because of its uses in research and also teaching-learning. The documentation available for a beginner user was found to be greatly helpful. Then, the training dataset was built. Using this training dataset the model was tested with the sample data. When the application was simulated, various results were observed. Even though the scale for experiment was very small, the HTK Tool appeared to be simple and reliable.

The necessary data was generated. Although only few samples were used to carry out the experiment, multiple comparisons were made with the limited data. The tests were carried out in such a way that the changes in the results were easily observed while changing the testing parameters.

The speech recognition system for the authentication system requires a higher level of accuracy. The experiments/test carried out showed that a higher level of accuracy can be achieved if the language model was designed for limited dictionary and trained the word model with a large set of speech data from the user. The database of each user could be created and used for authentication. Thus, it can be concluded that an identity verification system by speech recognition is possible. However the results produced were not successful to ensure the system could be used in real life mostly because of the small scale of the study carried out.

It can be concluded that the project was partly successful to achieve its goal.

References

1 Huang X, Acero A, Hon HW. Spoken Language Processing. Upper Saddle River, NJ: Pretince-Hall, Inc; 2001.

2 Russell S, Norvig Peter. Artificial Intelligence: A Modern Approach, Third Edition. Upper Saddle River, NJ: Pearson Education, Inc; 2010.

3 Press Release. Sensory Brings Voice Activation to the Samsung Galaxy - “Hi Galaxy” [Online]. Santa Clara, CA: Sensory, Inc.; 3 May 2012.
URL: http://www.sensoryinc.com/company/pr12_05.html.
Accessed 2 May 2013.

4 Newman J. Google’s ‘Project Glass’ Teases Augmented Reality Glasses [Online]. PCWorld.
URL:
http://www.pcworld.com/article/253200/googles_project_glass_teases_augmented_reality_glasses.html.
Accessed 2 May 2013.

5 What can I do with Speech Recognition? [Online]. Microsoft.
URL: http://windows.microsoft.com/en-au/windows7/what-can-i-do-with-speech-recognition.
Accessed 2 May 2013.

6 GNOME Community. Gnome-Voice-Control [Online]. Groton, MA: Gnome Foundation; 16 May 2011.
URL: https://live.gnome.org/GnomeVoiceControl.
Accessed 2 May 2013.

7 Open Mind Initiative. Open Mind Speech [Online].
URL: http://freespeech.sourceforge.net/.
Accessed 2 May 2013.

8 Mason SC, Michaelis A, Winne C. Perlbox Voice [Online]. Perlbox Community.
URL: http://perlbox.sourceforge.net/pbtk/.
Accessed 2 May 2013.

9 Schonfeld E. PhoneTag Voice-To-Text Is only 85 Percent Accurate, But That’s Better Than Google Voice [Online]. TechCrunch.
URL: http://techcrunch.com/2010/01/28/phonetag-voice-to-text-86-percent-accurate-google-voice/.
Accessed 2 May 2013.

10 Privacy Policy [Online]. Mountain View, CA: Google, Inc.; 27 July 2012.
URL: http://www.google.com/policies/privacy/.
Accessed 2 May 2013.

11 2012 Market Leaders [Online]. New York, NY: Speech Technology Media; 10 July 2012.
URL: http://www.speechtechmag.com/Articles/Editorial/Feature/The-2012-Market-Leaders-83629.aspx.
Accessed 3 May 2013.

12 Forsberg M. Why is Speech Recognition Difficult? Department of Computing Science, Chalmers University of Technology; 24 February 2003.

13 Scheips D. Voice recognition – benefits and challenges of this biometric application for access control.
URL: http://www.sourcesecurity.com/news/articles/co-3108-ga.4100.html.
Accessed 3 May 2013.

14 Speaker Recognition [Online]. National Science and Technology Council.
URL: http://www.biometrics.gov/Documents/speakerrec.pdf.
Accessed 3 May 2013.

15 Diamantini TM. The Physics of Sound: How We Produce Sounds [Online]. Yale-New Haven Teachers Institute.
URL: http://www.yale.edu/ynhti/curriculum/units/2003/4/03.04.04.x.html.
Accessed 3 May 2013.

16 Newman DL. The Physiology of Speech Production. Elvet Riverside, Durham: Department of Arabic, School of Modern Languages & Cultures, University of Durham.
URL: http://www.dur.ac.uk/daniel.newman/phon6.pdf.
Accessed 3 May 2013.

17 Head and Neck Overview [Image]. Wikipedia Commons.
URL: http://en.wikipedia.org/wiki/File:Illu01_head_neck.jpg.
Accessed 3 May 2013.

18 Coleman J.The vocal tract and larynx [Online]. Wellington Square, Oxford: Phonetics Laboratory, University of Oxford.
URL: http://www.phon.ox.ac.uk/jcoleman/phonation.htm.
Accessed 3 May 2013.

19 Sun DX. Statistical Models for Speech Recognition [Online]. Murray Hill, NJ: Bell Labs; 18 June 1997.
URL: http://cm.bell-labs.com/cm/ms/departments/sia/project/speech/index.html.
Accessed 3 May 2013.

20 Grabianowski E. Speech Recognition and Statistical Modeling [Online]. Atlanta, GA, HowStuffWorks.
URL: http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition2.htm.
Accessed 3 May 2013.

21 He X, Deng L. Discriminative Learning for Speech Recognition: Theory and Practice [pdf]. Morgan & Claypool.
URL: http://books.google.fi/books?id=yKSUUOQvbXoC.
Accessed 3 May 2013.

22 Glass J. Automatic Speech Recognition. MIT Open Courseware, Massachusetts Institutes of Technology.
URL: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-automatic-speech-recognition-spring-2003/index.htm.
Accessed 2 May 2013.

23 Sample-rate [Online]. Benchmark Media Syracuse, NY: Benchmark Media Systems, Inc.; 23 October 2006.
URL: http://www.benchmarkmedia.com/wiki/index.php/Sample-rate.
Accessed 3 May 2013.

24 Thrun S, Norvig P. Introduction to Artificial Intelligence [Online Course].
URL: ai-class.com.
Accessed 2 May 2013.

25 Rabiner LR. A Tutorial on Hidden Markov Models and Selected Applications on Speech Recognition [pdf].
URL:
http://www.cs.cornell.edu/Courses/cs4758/2012sp/materials/hmm_paper_rabiner.pdf.
Acessed 2 May 2013.

26 Gaussian Distribution Function [Online]. Atlanta, GA: Georgia State University.
URL: http://hyperphysics.phy-astr.gsu.edu/hbase/math/gaufcn.html.
Accessed 2 May 2013.

27 Bigelow C. The Normal Distribution [pdf/Lecture Slide]. Amherst; MA: University of Massachusetts.
URL: http://people.umass.edu/biep540w/pdf/normal.pdf.
Accessed 2 May 2013.

28 What is HTK? [Online]. Cambridge, United Kingdom: Cambridge University Engineering Department.
URL: http://htk.eng.cam.ac.uk/.
Accessed 2 May 2013.

29 Young S, Evermann G, Gales M, Gales M, Hain T, Kershaw D, Liu X, Moore G, Odell J, Ollason D, Povey D, Valtchev V, Woodland P. The HTK Book [pdf]. Microsoft Corporation (1995-1999); Cambridge University Engineering Department.
URL: http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml.
Accessed 2 May 2013.

30 Making Tied-State Triphones [Online]. VoxForge Community.
URL:
http://www.voxforge.org/home/dev/acousticmodels/linux/create/htkjulius/tutorial/triphones/step-10.
Accessed 2 May 2013.

The ARPA phonetic alphabet [2,914]

Listing of all the phones used in American English

Vowels		Consonants B-N		Consonants P-Z	
Phone	Example	Phone	Example	Phone	Example
[iy]	b**ea**t	[b]	**b**et	[p]	**p**et
[ih]	b**i**t	[ch]	**Ch**et	[r]	**r**at
[eh]	b**e**t	[d]	**d**ebt	[s]	**s**et
[æ]	b**a**t	[f]	**f**at	[sh]	**sh**oe
[ah]	b**u**t	[g]	**g**et	[t]	**t**en
[ao]	b**ough**t	[hh]	**h**at	[th]	**th**ick
[ow]	b**oa**t	[hv]	**h**igh	[dh]	**th**at
[uh]	b**oo**k	[jh]	**j**et	[dx]	bu**tt**er
[ey]	b**ai**t	[k]	**k**ick	[v]	**v**et
[er]	B**er**t	[l]	**l**et	[w]	**w**et
[ay]	b**uy**	[el]	bott**le**	[wh]	**wh**ich
[oy]	b**oy**	[m]	**m**et	[y]	**y**et
[axr]	din**er**	[em]	bott**om**	[z]	**z**oo
[aw]	d**ow**n	[n]	**n**et	[zh]	mea**s**ure
[ax]	**a**bout	[en]	butt**on**		
[ix]	ros**e**s	[ng]	si**ng**		
[aa]	c**o**t	[eng]	wash**ing**	[-]	*silence*