CHAPTER 3

LITERATURE REVIEW

3.1 Introduction

Speech recognition has of late become a practical technology. Speech recognition is used in real-world human language applications, such as information retrieval (Fujii et al., 2002). It is the most common means of the communication because the information contains the fundamental role in conversation. From the conversation or speech, it converts an acoustic signal that is captured by a microphone or a telephone, to a set of words (Zue et al., 1996). A set of word can either be the final result or it can then apply the synthesis to pronouns into sounds, which means speech-to-speech. Its means that, speech recognition can serve as the input to further linguistic processing in order to achieve speech understanding (Zue et al., 1996). Speech recognition systems can be characterized by speaking style, speaking mode, environment, vocabulary, acoustic model, language model, perplexity, SNR and transducer. In order to recognize the speech, many methods are used such as Neural Network (NN), Dynamic Time Warping (DTW), Vector Quantization (VQ), Expert System and Hidden Markov Model (HMM).
The research study was done by referring to the articles, books, conferences, internet, journals and databases. Mostly the articles are been selected from year 2000 to present. Only 20% of the data and information are referred in year 70s, 80s and 90s. Instead of finding the good and quality data, resources are finding in three (3) main channels, IEEE, Springer Link and ACM Portal besides Internet.

Overall, this chapter describes a review of speech recognition task, speech recognition approaches, current speech recognition system, Malay language speech recognition system as well as different type of methods apply in speech recognition system. Based on the review of the advantages and disadvantages, this thesis discusses the most suitable techniques and methods develop speech recognition system.

3.2 Automatic Speech Recognition

3.2.1 Current ASR System

Since 1930s, a simple speech machine that responds to a limited small set of words was invented. This machine is able to respond to spoken utterances and produce the speech. Since then, it has driven many-researched interest to invent a speech recognition system. One of the examples in 1950’s where Olson and Belar of RCA Laboratories who built a system to recognize 10 syllables of a single talker (Olson et al., 1956) and at MIT Lincoln Lab, Forgie and Forgie built a speaker-independent 10-vowel recognizer (Forgie et al., 1956). The study was continuously developed until by the mid 1970’s, the speech recognition based on LPC methods, were proposed by Itakura, Rabiner and Levinson (Itakura 1975; Rabiner et al., 1979) and others. This research brought major advantages
where research shift the methodology from the more intuitive template-based approach towards a more rigorous statistical modeling framework (Juang et al., 2004) in 1980s.

Diagram Figure 3.1 shows the milestones in speech and multimodal technology research. From a small vocabulary and acoustic-phonetics based in 1962s, speech technology has continuously developed and moving toward very large vocabulary by using semantic multimodal dialog technology.

**Figure 3.1**: The milestones of speech and multimodal technology research

Today speech technology plays an important role in many applications. Speech technology has moved from research to commercial application. Many human machine interfaces have been invented and applied today in telephone food ordering system, airport information system, ticketing system, restaurant reservation system, etc. Investigation has shown that
more than 85% of people are satisfied with the capability of the information inquiring service system of speech recognition (Jiang, 2009).

Many well-known English speech recognition systems are available in the market today. These include the speech recognition system from IBM, Microsoft, Dragon Natural Speaking 6 (currently known as ScanSoft), SPHINX, SRI/Nuance Communications, At&T Bell Labs, HTK, BBN – BYBLOS, Janus and so on. Table 3.1 shows the details of Euro-USA’s main speech recognition systems.

Table 3.1: English main speech recognition systems

<table>
<thead>
<tr>
<th>System</th>
<th>Conditions</th>
<th>Recognition Rate</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBN, BYBLOS System</td>
<td>Context dependent modeling of phonemes, continuous, vocabulary: 997 words</td>
<td>Recognition rate: 93%</td>
<td></td>
</tr>
<tr>
<td>(Chow et al., 1987)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bell Labs (Rabiner et al., 1989)</td>
<td>Speaker independent, connected digit recognition</td>
<td>Recognition rate: 97.1%</td>
<td>Using continuous HMM technology with multiple models and mixtures</td>
</tr>
<tr>
<td>CMU SPHINX (Lee et al., 1990)</td>
<td>Large vocabulary speaker independent, continuous speech recognition system, vocabulary: 997 words under the condition of grammar</td>
<td>Recognition rate: 96.8%, Phoneme recognition rate: 73.8%</td>
<td>Air Travel Information Service</td>
</tr>
<tr>
<td>DRAGON System of CMU</td>
<td>Uniform stochastic modeling, speaker-dependent, continuous, vocabulary: 194 words</td>
<td>Recognition rate: 84%</td>
<td>Latest Dragon Naturally Speaking 6 (Dictation software)</td>
</tr>
<tr>
<td>(Baker, 1975)</td>
<td></td>
<td></td>
<td>- available from ScanSoft Inc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- dictate 160 words per minute</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Windows XP, 2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Price – preferred = $200</td>
</tr>
</tbody>
</table>
Table 3.1: English main speech recognition systems, continued

<table>
<thead>
<tr>
<th>System</th>
<th>Conditions</th>
<th>Recognition Rate</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hild. H (Hild, 1993)</td>
<td>Speaker-dependent: 1000 sentences, multi-mode TDNN</td>
<td>Recognition rate: 98.5%</td>
<td>SPHIX: 96.0%</td>
</tr>
<tr>
<td></td>
<td>120 people when speaker-independent: 1680 words</td>
<td>Recognition rate: 92.0%</td>
<td>SPHIX: 90.4%</td>
</tr>
<tr>
<td>HTK toolkit (Young et al., 2002)</td>
<td>Continuous density Gaussian mixture HMMs, consists of a number of tools (programs) and a comprehensive set of library interface modules</td>
<td></td>
<td>The library modules ensure that all tools behave in a uniform way and they also simplify the development of new tools.</td>
</tr>
<tr>
<td>IBM, The Tangora System (IBM, 1985)</td>
<td>Speaker dependent Recognition, isolated word, vocabulary: 5000 words vocabulary with a natural language like grammar with perplexity 160.</td>
<td>Recognition rate: 97.1%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Speaker-independent, vocabulary: 20000 words</td>
<td>Recognition rate: 94.6%</td>
<td></td>
</tr>
<tr>
<td>IBM ViaVoice</td>
<td>Speaker-independent, vocabulary: 32000 Chinese words</td>
<td>Recognition rate: 95%</td>
<td>IBM ViaVoice is the Chinese version of Tangora system</td>
</tr>
<tr>
<td>INRS</td>
<td>Speaker-dependent, vocabulary: 75000 words</td>
<td>Recognition rate: 89.5%</td>
<td></td>
</tr>
<tr>
<td>Iso. K. (Iso, 1990)</td>
<td>Predictive neural network model, speaker-dependent, vocabulary: 5000 words</td>
<td>Recognition rate: 97.6%</td>
<td>It has a strong model-build capacity and can be used for person-independent continuous speech.</td>
</tr>
</tbody>
</table>
Table 3.1: English main speech recognition systems, continued

<table>
<thead>
<tr>
<th>System</th>
<th>Conditions</th>
<th>Recognition Rate</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMSI Lab (Gauvain et al., 1999)</td>
<td>Large vocabulary continuous speech recognition system, using CDHMMs with Gaussian mixture for acoustic modeling and back off N-gram statistics estimated, vocabulary: 65122 words (72788 phone transcription)</td>
<td>Overall word transcription error: 13.6%</td>
<td>Large vocabulary continuous speech recognition system in recognizing English broadcast-news speech.</td>
</tr>
<tr>
<td>Sawai. H. (Sawai, 1991)</td>
<td>A mixed method based on TDNN-LR-DP, 5000 words</td>
<td>Recognition rate: 92.6%</td>
<td></td>
</tr>
</tbody>
</table>

3.2.2 Malay ASR System

Malay is the national language for Malaysia, Indonesia and Brunei Darussalam. The term ‘Malay’ or ‘Melayu’ is an official nomenclature. Singapore and Brunei Darussalam called Bahasa Melayu (Malay language). In Malaysia and Indonesia, Malay language is known as Bahasa Malaysia (Malaysia language) and Bahasa Indonesia (Indonesia language) respectively.

More than 33,000,000 people speak Malay language in several countries and some smaller islands. It is also the national and official language used by Malaysia and Indonesia. People in Malaysia and Indonesia used Malay language to communicate in their daily life. The Malay language is different from English language. English words are formed due to the changes of phoneme in the based word itself according to its group of words (Asmah, 1989).
In recent years, many research studies have been conducted in developing the speaker independent for large vocabulary continuous speech recognition (LVCSR). Some of the system demonstrate very impressive accuracy and has some practical application (Ting, 2007), but most of the systems are based English language.

Today, there are also researches carried out on Malay isolated word recognition with accurate recognition rate. Most of these systems focus on digit recognition, which is identification of the digit from zero to nine either in Speaker Independent (SI) or Speaker Dependent (SD) manner. Most are isolated words and small vocabulary task. Thus, there are still limited research developed for speaker independent LVSCR in Malay speech domain.

Most of Malay words can be considered as a combination of syllables where syllables can be comprised of a vowel, or a vowel with a consonant or a vowel with several consonants (Asmah, 1989). For example, CV (C = consonant; V=Vowel), VC, CVC, CCVC, CVCC, CCCV and CCCVC (Ting et al., 2001). Among all the Malay syllables, the structure of consonant-vowel (CV) and consonant-vowel-vowel (CVV) are the most common found almost every Malay word. Example of Malay words recorded in the thesis are, ‘destinasi’, ‘tiket’, ‘taman’ and etc. A significant development of Malay ASR carried out by the researches and the summary are shown in Table 3.2.

-(Ting et al., 2001) used singular and modular Neural Network (NN) in classification of the Malay syllables in SI manner. All the speech conducted at sampling rate of 16KHz with 16bit. This research contains 1600 and 320 samples of speech in training and testing
respectively from 25 speakers. Linear predictive coding (LPC) is applied in the research to extract a set of speech features. The NN utilizes standard three layers Multi Layer Preceptron (MLP) as speech sound classification. Overall, singular NN achieves 92% of speech recognition rate.

- (Mohamad et al., 2004) conducts a study on Malay digit SD recognition based on improved NN. The study evaluates the recognition rate of Malay digits from zero to nine. It includes 50 speech samples from training and testing. Error Back-propagating (BP) is applied to improve NN. LPC is applied to extract a set of speech features. Result shows that speech recognition rate with improved BP (97.67%) is better and fast convergence rates compared to standard BP. BP is able to generate less epoch size compared to the mean square error function due to the enhancement of error function at the hidden layer.

-(Salam et al., 2001) This study is conducted on handcrafted and genetic algorithm based NN SD isolated Malay word recognition. As other studies, it recognize the digit isolated Malay words, zero to nine. LPC is used in feature extraction. A total of 360 utterances are recorded for training and testing purpose. From the experiment, recognition using genetic algorithm achieved 94% recognition rate and handcrafted NN achieved 95%. Even though the handcrafted method achieved 1% higher recognition rate, genetic algorithms is able to produce result within days while handcraft took weeks to recognize the words.

-(Ting et al., 2002) The isolated sounds are Malay plosive sounds, which comprised of /b/, /d/, /g/, /k/, /p/ and /t/. Based on this, a study that carries out by (Ting et al., 2002) using NN to recognize Malay isolated sounds of Malay Children in SI manner. Three-layers MLP is used to train and recognize speech sounds network parameter such as hidden neuron number and error function was investigated to achieve optimal performance of MLP. 30 Malay children aged 7-10 were involve in recognition. 600 and 300 speech tokens for train and test respectively. LPC is applied to extract the speech features. Overall, 84.67% recognition rate is achieved by using three-layers MLP.
-(Ting et al., 2004) study on the usage of NN in recognizing 6 Malay vowels of Malay Children in SI manner. This study collected a speech samples from the children ages between 7 (seven) to 10 (ten) years old in primary religious school. Total 40 speakers, which stand from 20 male, and 20 female are involved. There are 480 speech samples and 240 speech samples for training and testing respectively. One hidden layer of MLP was used to recognize the vowels. LPC is applied to extract the speech features. Overall, a 76.25% recognition rate is achieved in single frame of vowel signal of 70ms. Therefore longer vowel length was preferred over short signal length.

-(Ariff et al., 2005) design and implementation Malay speaker recognition system using Discrete HMM as classifier. 99 speakers (13 clients and 86 imposters) are involved in recording speech. Total of 250 utterances for each speaker in training and 50-isolated digit are used for testing. MFCC is used as speech signal processing and 24 MFCC coefficients were computed. As a result 7 digits long sequence, 0.96% error rate (EER) is achieved. It is reported that both longer cookbook size and longer digit sequence could be used to improve recognition performance.

-(Abd Manan, 2006) proposes a system for controlling automotive function using Malay speech recognition using Hidden Markov Modeling Toolkit (HTK) based on whole word and phone unit. This system is capable in controlling the radio tuning, wind screen function, air conditional control and cruise control. Total 8 speakers which stand from 4 males and 4 females continuous command sentences voice samples was recorded with 10 repetitions per utterance. Overall the HTK whole word unit scores 76.72% and 95.23% recognition rate for phone unit.

-(Al-Haddad et al., 2007) Another paper which carry out Malay isolated digits in SI manner using DTW. DTW is used to detect the nearest recorded voice with appropriate global constraint. Feature extraction method, MFCC is used to estimate the vocal tract filter. Result shown that 90.5% recognition rate achieved in Malay digit for all recorded words.
-(Al-Haddad et al., 2007) This paper studies the recognition rate between DTW and HMM for isolated Malay digit recognition. MFCC and vector quantization technology are used to process speech samples to accomplish the recognition. The difference between DTW and HMM are HMM used to emit a new feature vector for each frame access to an emission probability density function associated with that state. For HMM, recognition rate of 90.7% is achieved while 80.5% is achieved using DTW.

-(Ting, 2007) Applies HMM in recognizing the Malay digits as well. There are two type of HMM, DHMM and CDHMM. Different methods of HMM give different results. But both recognition rates are not significantly different. Whole word modeling is applied for each word in the system. This study conducts a speech recognition experiments between DHMM and CDHMM. Both are trained with different training algorithms. In DHMM, speech samples are trained by using Baum-Welch parameter re-estimation. While, HMM is trained by segmental k-mean. DHMM returns 96.62% word accuracy while HMM achieved 98.85% word accuracy. The less accuracy of DHMM is because DHMM models speech signals that has been vector quantized which cause loss of information in the subsequent modeling. While CDHMM directly model the continuous acoustic space without VQ, this maintains the information to be modeled (Ting 2007).

-(Ting, 2007) This study is focus on Malay connected digits recognition using whole-word based CDHMM connected word recognition system. Applied in speaker-dependent manner, this study apply segmental K-mean training procedure with Viterbi word segmentation follow by one pass Viterbi algorithm for recognition. Before training and testing, segmentation of the continuous training sentences (continuous digit strings) into individual tokens is needed. Then segmented words tokens in each files are used to rebuild all the corresponding word HMMs using typical word pattern building algorithm. Lastly, each models used to run the testing. Average recognition rate is 70.89% of word accuracy.

-(Fadhilah et al., 2008) This paper study on Malay isolated speech recognition using 5 states Hidden Markov Model (HMM) as acoustic model. Study focuses on Malay syllable especially CV and CVC syllables on 5 phonemes and structures such as ‘empat’, ‘lapan’, ‘tujuh’, ‘tidak’, ‘rekod’, and ‘tutup’. Recognition rate reaches 88.67% in the experiment.
<table>
<thead>
<tr>
<th>Paper</th>
<th>Type of Speech Recognition</th>
<th>Speaker manner</th>
<th>Database</th>
<th>Feature Extraction &amp; Methods</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ting et al., 2001)</td>
<td>Isolated words</td>
<td>SI</td>
<td>25 speakers</td>
<td>LPC</td>
<td>92.0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1600 and 320 speech samples</td>
<td>Singular and modular NN</td>
<td></td>
</tr>
<tr>
<td>(Mohamad et al.,)</td>
<td>Isolated words</td>
<td>SD</td>
<td>50 speech samples</td>
<td>LPC</td>
<td>97.67%</td>
</tr>
<tr>
<td>Malay digits 0-9</td>
<td></td>
<td></td>
<td></td>
<td>Error BP used to improve NN</td>
<td></td>
</tr>
<tr>
<td>(Salam et al., 2001)</td>
<td>Isolated words</td>
<td>SD</td>
<td>360 speech samples</td>
<td>LPC</td>
<td>94%</td>
</tr>
<tr>
<td>Malay digits 0-9</td>
<td></td>
<td></td>
<td></td>
<td>Handcraft and genetic algorithm NN</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Handcrafted NN</td>
<td></td>
</tr>
<tr>
<td>(Ting et al., 2002)</td>
<td>Isolated words</td>
<td>SI</td>
<td>20 speakers</td>
<td>LPC</td>
<td>84.67%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for 600 speech samples</td>
<td>3 layers MLP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10 speakers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for 300 speech samples</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Ting et al., 2004)</td>
<td>Isolated words</td>
<td>SI</td>
<td>480 speech samples</td>
<td>LPC</td>
<td>76.25%</td>
</tr>
<tr>
<td>6 Malay vowels</td>
<td></td>
<td></td>
<td>(training)</td>
<td>MLT + NN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>240 speech samples</td>
<td>with one hidden layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(testing)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Ariff et al., 2005)</td>
<td>Isolated words</td>
<td>SD</td>
<td>99 speakers</td>
<td>24 MFCC</td>
<td>0.96% EER</td>
</tr>
<tr>
<td>Malay digits 0-9</td>
<td></td>
<td></td>
<td>250 utterances for training</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50 for testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td>Type of Speech Recognition</td>
<td>Speaker manner</td>
<td>Database</td>
<td>Feature Extraction &amp; Methods</td>
<td>Performance</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------</td>
<td>----------------</td>
<td>---------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>(Adb Manan, 2006)</td>
<td>Continuous 56 Malay words</td>
<td>SI</td>
<td>8 speakers with 10 repetitions for training and testing</td>
<td>24 MFCC HTK HMM SVM</td>
<td>76.62% - whole word recognition 95.23% - phone recognition</td>
</tr>
<tr>
<td>(Ting, 2007)</td>
<td>Connected Words Malay digits 1-9, 'BELAS' &amp; 'PULUH'</td>
<td>SD</td>
<td>9 speakers each recorded 26 2-digit strings and 12 3-digit strings for training, 16 2-digit strings and 13 3-digit strings, separately recorded, for testing, Approximate 70 training tokens for each digit.</td>
<td>Segmental K-mean training procedure with Viterbi word segmentation. One pass Viterbi algorithm for recognition. 5 state whole word CDHMM with 4 mixture Gaussian densities.</td>
<td>70.89%</td>
</tr>
<tr>
<td>(Ting, 2007)</td>
<td>Isolated Words Malay digits 0-9</td>
<td>SD</td>
<td>26 speakers with 10 replication of each digit by each speaker (5 replication for training, another 5 for testing)</td>
<td>12th MFCCs Segmental k-mean parameter re-estimation. 5 state whole word CDHMM with 4 mixture Gaussian densities.</td>
<td>98.85%</td>
</tr>
<tr>
<td>Paper</td>
<td>Type of Speech Recognition</td>
<td>Speaker manner</td>
<td>Database</td>
<td>Feature Extraction &amp; Methods</td>
<td>Performance</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(Ting, 2007)</td>
<td>Isolated Words</td>
<td>SD</td>
<td>26 speakers with 10 replication of each digit by each speaker (5 replication for training, another 5 for testing)</td>
<td>12th MFCCs 5 state whole word DHMM with 128 codebook entries, Baum-Welch parameter re-estimation</td>
<td>96.3%</td>
</tr>
<tr>
<td>(Al-Haddad et al., 2007a)</td>
<td>Isolated words</td>
<td>SI</td>
<td>100 speech samples</td>
<td>MFCC DTW</td>
<td>90.5%</td>
</tr>
<tr>
<td>(Al-Haddad et al., 2007b)</td>
<td>Isolated words</td>
<td>SI</td>
<td></td>
<td>MFCC HMM</td>
<td>90.7%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MFCC DTW</td>
<td>80.5%</td>
</tr>
<tr>
<td>(Fadhilah et al., 2008)</td>
<td>Isolated 6 Malay vowels</td>
<td>SI</td>
<td>21 speakers (10 male and 11 female)</td>
<td>MFCC 5 states HMM with mixture Gaussian densities, Baum Welch algorithm for training Viterbi algorithm for recognition</td>
<td>88.67%</td>
</tr>
</tbody>
</table>
3.3 Speech Recognition Approaches

Automatic speech recognition system is used to transform or produce a sequence of message from a speech signal. This process is called decoding. Speech signal is decoded and then converted into writing (e.g. dictation machine) or commands to be processed (e.g. hands free dialing) (Murat, 2003). In general, there are three classical approaches as follows:

i. Acoustic-phonetic approach,

ii. Pattern recognition approach and

iii. Artificial intelligence approach.

Among the three approaches, the acoustic-phonetic approach has been studied and researched more in the past 40 years. This method is the oldest speech recognition approach originating from the 1950s. The AI approach is the youngest and least known. While pattern recognition approach is the most common approach applied in most current ASR systems.

3.3.1 Acoustic-Phonetic Approach

The acoustic-phonetic approach is based on the theory of acoustic phonetics. The theory proposes that there exist finite, distinct phonetic units in spoken language. The phonetic units are characterized by a set of properties that are embedded in the speech signal or its spectrum.
Even though the acoustic properties of phonetic units are highly variable, both with the speaker and with the neighboring phonetic units, it is assumed that the rules governing the variability are straightforward and can be readily learned and applied in practical situations.

**Figure 3.2:** Diagram of acoustic phonetic speech recognition system (Rabiner & Juang, 1993)

Figure 3.2 above shows the diagram of acoustic-phonetic speech recognition system. In order to recognize the speech, the first step is speech analysis or feature measurement method, which involves the filter bank processing. Each spectral represent the characteristics of time-varying speech signal. The most common spectral analysis methods are discrete Fourier Transform (DFT), Linear Predictive Coding (LPC), or Mel Frequency Cepstral Coefficients (MFCC) methods.

During the feature-detection process, the spectral is converted to a set of features. Among the features are nasality (nasal resonance), frication (random excitation), format
locations (frequencies of the first three resonances), voiced/unvoiced classification (periodic or a periodic excitation), and energy ratios.

The third process is segmentation and labeling stage. In this stage, system find the feature stable regions and then label those regions accordingly in order to match each individual phonetic units. The result of segmentation and labeling can be viewed as Figure 3.3. The figure shows a phoneme lattice for word string that result from segmentation and labeling process. This phoneme lattice represents a sequential set of phonemes that are likely matches to the spoken input speech.

![Figure 3.3: Segmentation and labeling for word sequence “seven-six” (Rabiner & Juang, 1993)](image-url)
Besides, there are also some problem and dilemma face by acoustic phonetic approach as discuss as the following (Rabiner & Juang, 1993).

i. Among the difficulties of this method is need for extensive knowledge of acoustic properties of phonetic units.

ii. Features are often based on non-optimal ad hoc considerations rather than based on intuition.

iii. The choice of features is likely based on suboptimal and so optimal implementation of classification and regression tree (CART) methods is rarely achieved.

iv. Furthermore, there is no well-defined, automatic procedure for tuning the labeled speech.

v. Moreover, no standard way in labeling the training speech. Naturally, these problems need to be solved for it to be utilized practically.

Due to the limitations, acoustic phonetic approach still needs much more research and understanding before it can successfully implemented in actual speech recognition system.

3.3.2 Statistical Pattern Recognition Approach

In pattern-recognition approach, the speech patterns are used directly without explicit feature determination and segmentation. There are two main steps: Training of speech patterns and recognition of patterns via pattern comparison. Speech knowledge is supplied into the system via the training procedure. Most of the current and modern ASR systems are based on the principles of statistical pattern recognition. As shown in Figure 3.4, speech recognition
using statistical pattern recognition paradigm has four steps in addition to the two main steps stated.

![Diagram of statistical pattern speech recognition system (Rabiner & Juang, 1993)](image)

**Figure 3.4:** Diagram of statistical pattern speech recognition system (Rabiner & Juang, 1993)

Firstly, features are extracted from the input signal and represented into a form of features. There are number of spectral analysis techniques, such as filter DFT, LPC and MFCC analysis.

Secondly, in pattern training one or more test patterns corresponding to speech sounds of the same class are used to create a pattern representative of the features of that class. The resulting pattern, generally called reference pattern, can be template, derived from some type of averaging technique, or it can be a model that characterizes the statistics of the features of the reference pattern.

This is followed by pattern classification process. Here, the unknown speech input is compared in pattern training, by measuring the similarity between the train and test pattern. Lastly, decision logic is applied to decide which reference best matches the unknown test pattern.
Based on (Rabiner & Juang, 1993) research, pattern recognition in speech recognition has its strengths and weaknesses which are defined as follows:

i. The system performance is sensitive to the amount of training data available for creating sound class reference patterns. The more training, the higher the performance of the system for virtually any task.

ii. The reference patterns are sensitive to the speaking environment and transmission characteristics of the medium used to create the speech; this is because the speech spectral characteristics are affected by transmission and background noise.

iii. Unlikely LPC, MFCC does not need speech-specific knowledge in the system; hence, the method is relatively insensitive to the choice of vocabulary words, task syntax, and task semantics.

iv. The computational load for both pattern training and pattern classification is generally linearly proportional to the number of patterns being trained or recognized; hence, computation of a large number of sound classes could and often does become prohibitive.

v. Because the system is insensitive to sound class, the basic techniques are applicable to a wide range of speech sound, including phrases, whole words, and sub-word units. A basic set of techniques developed for one sound class (e.g., words) can generally be directly applied to different sound classes (e.g., sub-word units) with little or no modifications to the algorithms.

vi. It is relatively straightforward to incorporate syntactic (and even semantic) constraints directly into the pattern recognition structure, thereby improving recognition accuracy and reducing computation.
3.3.3 Artificial Intelligence (AI) Approach

The artificial intelligence approach is a compound approach that utilizes the ideas of the first two approaches. The intention here is to mechanize the recognition procedure like the way a person applies his intelligence in analyzing and making decision on the acoustic knowledge. The aim is to integrate phonemic, lexical, semantic and pragmatic knowledge together (Rabiner & Juang, 1993).

Normally, there are three alternative ways often used in AI speech recognition system: “bottom up”, “top down” and “blackboard” approaches. In the standard “bottom up” processor as shown in Figure 3.5, the lower-level processes are applied before the higher-level processes. Lower-level process includes feature extraction and phonetic decoding while higher-level processes includes lexical decoding and language model.

Another alternative way is “top down” process. The processor integrates the word hypothesis matching, lexical decoding and syntactic analyses blocks into a consistent framework as shown in Figure 3.6.
Figure 3.5: Bottom-up approach to knowledge integration (Rabiner & Juang, 1993)

Figure 3.6: Top-down approach to knowledge integration (Rabiner & Juang, 1993)
The third alternative way is “blackboard” approach (as show in Figure 3.7). In this approach all knowledge sources (KS) are considered independent. A hypothesis-and test paradigm is applied as the main communication medium among KSs that are data driven, and based on the patterns (matching the KS templates) on the blackboard. The system operates asynchronously, and assigned cost and utility considerations are distributed across all levels. The approach was extensively studied at Carnegie Mellon University (CMU) in the 1970s, and it has been further researched for dialogue-based expert systems especially at Massachusetts Institute of Technology (MIT).

**Figure 3.7**: Blackboard approach to knowledge integration (Rabiner & Juang, 1993)

### 3.4 Speech Recognition Methods

The main task of speech recognition process is to match an input signal with a single or a set of words or sentences according to some optimality criteria (Vergin et al., 1999). Speech recognition is usually focused on pronunciation and intonation. Many
studies have been conducted in either automatic speech synthesis or automatic speech recognition.

Besides that, there are many types of techniques can be applied to recognize the speech. Among them is Hidden Markov Model (HMM), Neural Network (NN), Vector Quantization (VQ), Expert System and Dynamic Time Warping (DTW) are commonly the classical method for speech recognition. From the speech waveform, through feature extraction to spectral feature vectors, Gaussians models then follow by finding the phone likelihoods and decoding stage. A set of text or word(s) is recognized based on the highest probabilities. In section 3.6.1, DTW is described. Follow by ANN, VQ and HMM in section 3.6.2, 3.6.3 and 3.6.4 respectively.

3.4.1 Dynamic Time Warping (DTW)

Dynamic Time-Warping (DTW) is one of the common techniques in speech recognition systems. It is one of the oldest, method since 30 years ago and most important algorithms by matching the unknown speech input template to a pre-define reference template in speech recognition (Sakoe & Chiba, 1978).

DTW approach is a template matching method, where it compares the unknown pattern with its reference template to get the minimum score. The minimum score indicates that the unknown pattern is most likely to be matched onto the particular reference template compared to other reference. The algorithm finds the optimal nonlinear alignment between the unknown speech patterns with the reference pattern which may both vary in duration due to different speaking rate to obtain their global distance, indicating their similarity.
Overall, one of the reference templates is selected to test the recognition rate. If the recognition rate is high then this reference template is kept, otherwise another new template is selected for another round of recognition testing.

Traditionally, DTW algorithm needs long processing time and large pattern storage, which become a major problem for real time application as the number of speech patterns increases. As a result, DTW is widely used in the small scale speech recognition systems. For example, it provides a good recognition performance in small vocabulary, isolated word or speaker dependent.

Also, DTW approach is limited to word template. Ney (Ney, 1984) has extended the usage of DTW in isolated word to continuous speech recognition with the algorithm called One Stage DTW. Here the goal is to find the speech sample alignment and the best sequences of reference words.

Compared to HMM, DTW required high computational requirement. HMM can capture the statistical characteristics of word and sub-word units among different speakers even in large vocabulary and thus it is better than DTW in speaker independent large vocabulary speech recognition (Wong, 1998). The DTW techniques have been generally superseded by the more powerful and flexible HMM models.

3.4.2 Artificial Neural Network (ANN)

Neural Network or known as artificial neural network (ANN) is a mathematical model which based on biological neural networks. According to (Melin et al., 2009),
biological neural networks are made up of real biological neurons that are connected in the peripheral nervous system or the central nervous system. (Melin et al., 2009) also define that, artificial neural networks are made up of interconnecting artificial neurons. During the learning phase, ANN able to change the network structure based on internal and external information. This can see more details on speech recognition when the speech input is passing through the network. Figure 3.8 shows the network diagram of ANN from an input through hidden level and then produces the output.

![Neural Network Diagram](image)

**Figure 3.8**: Neural Network diagram

The network topology allows input from the input layer to the first hidden layer then from the first hidden layer to the second and until it reach at the last hidden layer to the output layer. Figure 3.19 illustrated the movement from input layer, first hidden layer and output layer. The hidden layer learns to recode or to provide a representation for the inputs (Smith, 2001). Based on (Smith, 2001), more than one hidden layer can be used in neuron network.
ANN is different with HMM, the speech input (sound) in HMM is model based on state sequence. In HMM, probabilities are assigned to the state transition and probability is associated with the observation of acoustic parameters, which correspond to states or state transition. ANN is a popular and has been successful in pattern recognition. Only recent years, ANN has been used for speech recognition. As described above, ANN is a structure, which is made up by biological which composed of many nonlinear computational elements operating parallel in patterns.

Moreover, ANN is a fault tolerance and nonlinear property when applied in speech recognition (Haykin, 1999) (Azam et al., 2007). And this encourages the development of speech recognition using neuron network. Even though neuron networks are excellent classifiers, but the performance is dependent on the quality and quantity of training samples presented to the network (Azam et al., 2007). For large vocabulary, ANN took longer time in recognition. According to (Azam et al., 2007), researches can reduced the number during neural network training.

ANN in speech recognition is done as the following steps:

i. During recognition process, input speech is recorded.

ii. MFCCs are used to extract the speech signal (Frames features are extracted by using MFCC coefficients).

iii. Get the highest probabilities of feature vector into the neural network.

iv. Based on that, neural network able to classify the unknown digit.

Figure 3.9 shows the example flow of speech recognition using neural network and the frame feature is extracted by MFCC coefficients. Neural network has three layers, the
input nodes (at input layer), hidden nodes (at hidden layer) and output nodes (output layer). One or more hidden layers of nodes is located between the input and output nodes (Alotaibi, 2004). The structure of the neural network is usable where it provides adequate training data (input nodes) and hidden nodes.

![Example flow of neural network in speech recognition](image)

**Figure 3.9:** Example flow of neural network in speech recognition

### 3.4.3 Vector Quantization (VQ)

Quantization is an important aspect of data compression. According to (Gray, 1984), the purpose of data compression is to reduce the bit rate to minimize communication channel capacity or digital storage memory requirements while maintaining the necessary fidelity of the data.
Vector quantization (VQ) is a classical quantization technique. It divides a large set of vectors into multiple groups. Each group is represented by its own centroid point. In Figure 3.10, the dots are called code vectors or centroid and each region is separated by the encoding regions. A codebook is composed of set of all code vectors while set of encoding regions form the space partitions.

![Figure 3.10: An example of codebook and codebook borderline of vector x](image)

Vector quantization codebook was used as an efficient means of characterizing the short-time spectral codebooks features of a speaker (Soong et al., 1985). A codebook is trained in training phase. During the testing phase, a set of cookbook was then used to recognize an unknown speaker from his spoken utterances based on a minimum distance between dot and centroid. Commonly VQ uses Euclidean distance algorithm to find the most nearest distance between code and the spoken utterances.

Most of the VQ speech recognition, one codebook is used for each word recognition. Each codebook is created from a training sequence containing repetitions of one vocabulary word (Savage & Herrera, 1991). For example, a codebook for the word “pulau” would be
designed upon several repetition of word “pulau” before the finalized codebook. Therefore each codebook needs to update always in order to get the precise recognition rate.

In speech recognition of VQ, a speech input is passed through each vector quantizer. A set of distortions $d_1(t), d_2(t), ..., d_M(t)$ upon quantize is produced (Savage & Herrera 1991). After that, distance is computed to find the shortest and best fit of frame $t$ of the voice signal. Each of the quantizer is stand from 1 until $M$, 1, ... , $M$. Thirdly, compare the global distortions $D_1, ... , D_m$ from each quantizer in order to identify the word spoken. A smallest global distortion is selected during $t=1, ... , T$ as Equation (1) and follow by Equation (2). Word $j$ is selected if the distance is minimum.

$$D_i = \sum_{t=1}^{T} d_i(t)$$  \hspace{1cm} (1)

$$D_j < D_i, \hspace{1cm} i = 1...M$$  \hspace{1cm} (2)

In conclusion the VQ process using K-means algorithm clustering is described as below. Figure 3.11 shows more details of the speech recognition process using VQ.

i. Initialize the codebook before the training phase start.

ii. Search the nearest centroid in the codebook.

iii. Update the centroid every time when adding a new feature vector into the codebook. This to ensure that, the feature vector able to find the nearest centroid.

iv. Repeat the flow of (2) and (3) until the distance fall below the threshold which would mean that the distance is unchanged.
3.4.4 Hidden Markov Model (HMM)

The Hidden Markov Models (HMMs) are widely used statistical tools in recognition system. It covers from isolated speech recognition to very large vocabulary unconstrained continuous speech recognition and speaker identification fields. Therefore most of the current speech recognitions are conducted based on Hidden Markov Model (HMMs).

The Hidden Markov Model is a statistical model where the system is modelled as Markov process which can be represented as a state machine with unknown parameter through it (Ting, 2007). The main important process is to determine the unknown parameter from the observable parameter. After determining the parameter, it then used to perform further analysis. One of the examples of the Hidden Markov Model is normally use in pattern application. For example, Hidden Markov Model in pattern application is

---

**Figure 3.11:** Example of speech recognition using VQ
speech, signature, handwriting, gesture recognition and bioinformatics and genomics in medical.

3.4.4.1 Actual HMM Topology

There are many speech recognition based HMM's method system that are being studied. Due to well present of sequential HMM, left-to-right models (Figure 3.12) was widely used for the speech model topology. Besides, it also provides a more rigid temporal structure where only transitions from left to right are allowed. Therefore, left to right model is used in this thesis.

For other speech-modeling tasks, the use of ergodic models (Levinson, 1987) as shown in Figure 3.13 is often more appropriate. Ergodic model also contain number of state and topology is configured based on prior knowledge without any mathematic computations.

![Figure 3.12: Left to right HMM Model](image-url)
For the left to right HMM model topology, the constraint is imposed so that the first frame of speech is allocated to the first state of the model and the last frame of speech is allocated to the last frame of the model. There are two probabilities for each frame. The frame can either be allocated to itself (a self transition) or it can be allocated to the next state. To improve the modeling of the temporal structure of speech signal, each state has the probabilities for these two types of transition, known as the transition probabilities. The two transition probabilities always sum to one.

### 3.4.4.2 HMM Architecture Type

HMM have several density architectures which are discrete HMM (DHMM) and continuous density HMM (CDHMM). The architecture of these models has its own advantages and disadvantages which are described as follows.
Discrete Hidden Markov Model (DHMM)

Discrete Hidden Markov Model (DHMM) is a type of HMM that model speech signals based on VQ technique to produce the speech observations. It is a combination of vector quantization and hidden Markov modeling. In 1982, Rabiner and his colleagues (Rabiner & Huang, 1993) was successful applied DHMM speech recognizer for a speaker independent isolated digit recognition task. Since then, research progress is made from isolated word recognition until large vocabulary recognition systems such as SPHINX system.

The main advantages of DHMM are its concise concepts and low computational costs. Compare to DTW, DHMM is more efficient and reliable technique. In DHMM, the VQ computation depends on the codebook size and computes the discrete output probability of an observation based on a lookup table. In addition, DHMM can easily model the phones, phonemes, and sub words. On the other hand, it is very difficult with DTW due to this technique needs to compute the boundary. DHMM used VQ codebook to represent the speech spectral vectors and creates an inherent spectral distortion in representing the actual analysis vector. The size of codebook can cause the quantization errors. On the other hand, large codebook size cause less training data for each codeword and therefore affects the recognition performance.

Continuous density Hidden Markov Models (CDHMM)

Continuous density Hidden Markov Models (CDHMM) models the acoustic observation by directly using estimated continuous probability density function (PDF) which is typically a mixture of Gaussian functions, without VQ. In this model, parameters
must be estimated for each state of each model. Thus, it is able to eliminate the VQ error effect of DHMM and increase the recognition accuracy. Although CDHMM requires longer training and recognition time it is a powerful model of acoustic variability with its Gaussian mixture densities.

3.4.4.3 HMM Representation Speech Units

It is important to identify the modeling unit for recognizer as it determines and affect the recognition result. Below describe the word and phone speech units.

**Word Model**

Word is a natural speech unit where it can produce the best performance when it adequately trained. For small vocabulary speech recognition, word model is suitable to trained and easy to collect speech samples for each word. However, if word model applied in large vocabulary, it might cause problems since each word cannot be shared in the training data. Therefore, memory usage grows with the number of words. Also, it is difficult to model word boundary coarticulation effects.

**Phone Model**

Phone model is more suitable compared to word model as it is trainable and sharable across different words. A phone unit can be shared among words and sentences. Unlike word model, monophone model has no problem in training. Different word that contain the same phone share the same phone model and utterance of one word provide training data for parts of the other words. A new word where phones already existing in the list can be added into the system easily without collecting utterance samples.
Context Dependent Phones

In speech recognition, HMMs are used to represent sub units of words or phones. Word models can be constructed from a combination of the sub word models. Monophone is a non-left or right context phone model. Biphone can stand from left or right context. A left context biphone contain left phone context. Similar with right-context biphone is dependent on its right context.

This is different for triphone model. Triphone model can take both contexts, left and right contexts. Each triphone is surrounded by two phones. However, it is difficult to train due to the large number of context. Figure 3.14 shows the difference between these three phone models where Malay word ‘dari’ is used as an example.
3.4.4.4 HMM Formulas

The Hidden Markov Models (HMMs) have been used successfully in a wide variety of recognition tasks ranging from small isolated word system assisted by heavily constrained grammars to very large vocabulary unconstrained continuous speech recognition (Odell et al., 1994). A HMM is a triple which consists of a vector and two matrices ($\pi$, $A$, $B$). These three sets of probabilities are vector ($\pi$), state transition matrix ($A$) and confusion matrix ($B$). Vector as Equation (3) contains the probability of the model in hidden state. State transition matrix as Equation (4) and confusion matrix as Equation (5).
contains the probability of observed state given that the hidden model is in hidden state. Besides that, HMM contain two sets of states, hidden state and observable state (visible state).

\[ \pi = (\pi_i) \]  
\[ A = (a_{ij}) \]  
\[ B = (b_{ij}) \]

Hidden Markov Model is so called hidden because it has an unknown parameter. The state is not directly visible, but the variables influenced by the state are visible (Măndoiu et al., 2008). HMM is a statistical model where model under the Markov process with unknown parameter, \( X \) through it. And the main important process is identified and determines the unknown parameter, \( X \) from the observable parameter, \( O \). After determining the parameter, it then used to perform further analysis. One of the examples of the Hidden Markov Model is normally used in pattern application. For example, Hidden Markov Model is applied pattern in application speech, signature, handwriting, gesture recognition and bioinformatics and genomics in the medical field.

As describe above, the observable parameter is represented by observation \( O \) or speech vectors as Equation (6). This speech vector is produced from each spoken word, \( W = w_1, w_2, w_3, \ldots, w_T \).

\[ O = o_1, o_2, \ldots, o_T \]  
, where \( o_t \) is the speech vector observed at time \( t \).
In speech recognition, the system must choose a word string $W$ (Equation 7) that maximizes the probability given acoustic observation $O$:

$$\hat{W} = \arg \max_w P(W | O)$$ (7)

However, this probability is not computable directly since most likely word sequence needs to be chosen. This problem can be solved by applying the Bayes’ rule as Equation (8).

$$\hat{W} = \arg \max_w \frac{P(O | W) P(W)}{P(O)} = \arg \max_w P(O | W) P(W)$$ (8)

The $P(O | W)$ (also known as acoustic model) consider how likely the observation acoustic signal are output by a word sequence and its subcomponents such as phonemes. The acoustic model here means hidden markov model. $P(W)$ (which is also known as Language model) is s prior of word sequence in sentence. Figure 3.15 shows the statistical overview of speech recognition using Bayes rule.
Figure 3.15: Overview of statistical speech recognition using Bayers’ rule

In HMM based speech recognition, a sequence of each observed word is generated by a Markov model. A Markov model is a finite state machine which changes state once time unit and each time $t$ that a state $j$ is entered, a speech vector $o_t$ is generated from the probability density $b_j(o_t)$ (Young et al., 2002). As an example, Figure 3.16 shows an example (calculate $P(O|W)$) where six states model moves through the state sequence $X = 1, 2, 3, 4, 5, 6$ in order to generate the sequence $o_1$ to $o_2$. The probability $O$ is generated by
model M moving through state sequence $X$ as product of transition probability and the output probability. The Equation (9) presents the above explanation.

$$P(O, X | M) = a_{i2} b_2 (o_1) a_{22} b_2 (o_2) a_{33} b_3 (o_3) \ldots$$

(9)

**Figure 3.16:** The Markov Generation model

Overall, only the observation sequence $O$ is known while state sequences $X$ is unknown (thus called Hidden Markov Model) therefore the required likelihood can be computed by summing over all possible sequences $X = x(1), x(2), x(3), \ldots, x(T)$ as Equation (10). As an alternative to Equation (10), the likelihood can be approximated by considering the most likely state sequence as Equation (11).

$$P(O|M) = \sum_X a_{x(t)j(t)} \prod_{t=1}^{T} b_{x(t)} (o_t) a_{x(t)x(t+1)}$$

(10)

$$\hat{P}(O|M) = \max_X \left\{ a_{x(t)j(t)} \prod_{t=1}^{T} b_{x(t)} (o_t) a_{x(t)x(t+1)} \right\}$$

(11)
However, if Equation (8) is computed, then speech recognition problem is solved. Therefore, assumes that Equation (12) is happen.

\[ P(O \mid W_i) = P(O \mid M_i) \]  

(12)

Thus, Figure 3.17 summarizes the overall process of choosing the maximize likelihood among the models, \( M \) during recognition process. First, recorded words (i.e. Ampang, KLCC, Universiti and etc.) are trained. During the recognition, the each model likelihood is calculated and the model is chosen to maximize probability.
3.4.4.5 HMM Training Algorithm

HMM modeling is provides a relatively easy and efficient training algorithm given a set of training data. These training algorithms are based on different parameter-estimation criterions. There are two well-known training algorithms; Baum Welch and Viterbi training algorithm.
Baum Welch Training

Baum Welch (BW) algorithm is based on maximum likelihood estimation (MLE) criterion. It involves maximization of $P(O|M)$ over $M$ to find optimal model to fit the observation sequence $O$. BW algorithm is used to find the unknown parameter of HMM ($A$, $B$, and $\pi$).

The CDHMM training based on BW algorithm is shown in Figure 3.18. Firstly, a set of training set of $n$ utterances is used to estimate the parameter of single CDHMM where each $n^{th}$ utterance $X^k$ is represented by a sequence of continuous observation vectors. Secondly, compute $\alpha^k_t(i)$ and $\beta^k_t(i)$ using forward and backward algorithm for each training utterance. The output probability distribution is calculated using continuous pdf.

Figure 3.18: Overview process of Baum Welch training algorithm
This is followed by computing a posterior state probability state $\gamma$ and $\zeta$ using $\alpha$ and $\beta$ values above. The $\xi^k_t (i, j)$ is calculated in states $S_i$ and $S_j$ at time $t$, given $X^n$ and $\lambda$ is express as Equation (13) while $\gamma$ probability as Equation (14) respectively.

$$\xi^k_t (i, j) = \frac{\alpha^k_t (i)a_{ij}b_j (x^n_{i+1})\beta^k_{t+1} (j)}{\sum_{i=1}^{N} \alpha^k_t (i)\beta^k_t (i)}$$  \hspace{1cm} (13)

$$\gamma^k_t (i) = \frac{\alpha^k_t (i)\beta^k_t (i)}{\sum_{i=1}^{N} \alpha^k_t (i)\beta^k_t (i)}$$  \hspace{1cm} (14)

Next, implement HMM parameters re-estimation using Gamma terms. Finally, an update model is obtained. The iterative re-estimation procedure continues until no improvement in model is achieved.

**Viterbi Training**

Viterbi algorithm is used to find the most likely state sequence in a hidden Markov model given for a sequence of observed outputs. The Viterbi algorithm can be regarded as the dynamic programming algorithm applied to HMM or as a modified forward algorithm (Fadhlilah 2008). A different with forward algorithm is Viterbi algorithm able to choose and remember the best path.

Same as Baum Welch algorithm, model parameters are initialized. After initialization, apply the state sequence segmentation using Viterbi algorithm to find the most likely path. Each observation sequence is segmented into N segment with each
segment $S_j$, correspond to each HMM state, $j$ (Ting, 2007). Thirdly, to apply HMM parameter re-estimation for $a_{ij}$ and $B (\cdot)$ via segmental K-mean or highest likelihood. The overall process flow of Viterbi algorithm is shown in Figure 3.19.

**Figure 3.19**: Overview process of Viterbi algorithm

### 3.4.4.6 Three Problems of HMM

There are three problem need to be solved before applying HMM into speech recognition system.
Problem 1: Evaluation

*Given:*
- Model $\lambda = (A, B, \pi)$ and;
- Testing observation sequence for $O = O_1 \ldots O_T$.

*Action:*
- Compute the probability of the observation sequence given the model, $P(O/\lambda)$.

This is an evaluation problem which find the probability of producing a given observation $O$ by a given model $\lambda$. The solution of problem is allowed to find the best model among multiple solution given the observation for the purpose of classification and recognition.

Problem 2: Decoding

*Given:*
- Model $\lambda = (A, B, \pi)$ and;
- Testing or training observation sequence $O = O_1, O_2, O_3, \ldots, O_{T-1}, O_T$.

*Action:*
- Track the optimum state sequence $Q = q_1, q_2, q_3, \ldots, q_{T-1}, q_T$ that most likely produce the given observations, using the given model.

This purpose of the decoding problem is to find the most likely sequence. The decoding procedure allows detecting or unhiding the state sequence of a given observation. It is used to find the optimal state sequence for continuous speech recognition.

Problem 3: Training

*Given:*
- Model $\lambda = (A, B, \pi)$
- Training observation sequence $O = O_1, O_2, O_3, \ldots, O_{T-1}, O_T$ where $k$ is the number of examples for training the model.

*Action:*
- Adjust the model parameters $\lambda = (A, B, \pi)$ and maximize $P(O/\lambda)$. 
This is estimation (or often called training) problem where unable to find the correct model parameter values. The training procedure is used to optimize the model parameters to obtain the best model that represent certain set of observations belonging to one spoken entity.

### 3.4.4.7 Solution to Three HMM Problems

**A) Problem 1**

Problem 1 is on evaluating how well a given model matches a given observation sequence. A most straightforward way to determine $P(O \mid \lambda)$ to find $P(O \mid I, \lambda)$ for fixed state sequence $I = i_1, i_2, \ldots, i_T$ is to multiply it by $P(I \mid \lambda)$ and then sum up over all possible $I$'s.

For

$$P(O \mid I, \lambda) = b_{i_{1}}(O_{1}) b_{i_{2}}(O_{2}) \ldots b_{i_{T}}(O_{T})$$

$$P(I \mid \lambda) = \pi_{i_{1}} a_{i_{1}i_{2}} a_{i_{2}i_{3}} \ldots a_{i_{T-1}i_{T}}$$

Therefore

$$P(O \mid \lambda) = \sum_{I} P(O \mid I, \lambda) P(I \mid \lambda)$$

$$P(I \mid \lambda) = \sum_{I} \pi_{i_{1}} b_{i_{1}}(O_{1}) a_{i_{1}i_{2}} b_{i_{2}}(O_{2}) \ldots a_{i_{T-1}i_{T}} b_{i_{T}}(O_{T}),$$

where $I = i_{1}, i_{2}, \ldots, i_{T}$.

The calculation of $P(O \mid \lambda)$ is involved in the order of $2T \cdot N^T$ calculations. This calculation is computationally unfeasible, even for small value of $N$ and $T$. Thus a more efficient procedure is required to solve problem 1. The method is called forward-backward procedure.
The Forward-Backward Procedure: Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, i_t = i \mid \lambda)$$  \hspace{1cm} (19)

where the probability of the partial observation sequence up to time $t$ and the state $i$ at time $t$, given the model $\lambda$. $\alpha_t(i)$ can be computed as follows:

Step 1:

$$\alpha_t(i) = \pi_i b_i(O_1), \hspace{0.5cm} 1 < i < N$$  \hspace{1cm} (20)

Step 2:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1})$$  \hspace{1cm} (21)

Step 3:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$  \hspace{1cm} (22)

Step 1 initializes the forward probabilities as the joint probability of state $S_i$ and initial observation $O_1$. In Step 2 induction, compute the probability of partial observation sequence up to time $t+1$ and state $j$ at time $t+1$. State $j$ ($S_j$) can be reached with probability $a_{ij}$ independently from any of the $N$ states at time $t$ as show in Figure 3.20. This illustrates the summation for Equation (21). Once the $S_j$ is known, it is easy to see that $\alpha_{t+1}(j)$ is obtained by accounting for observation $O_{t+1}$ in state $j$, i.e., by multiplying the summed quantity by the probability $b_j(O_{t+1})$. The computation of Equation (22) is performed for all states $j$, $1 < j < N$, for a given $t$ and iterated for $t = 1, 2, \ldots, T-1$. Finally final step give the desired calculation of $P(O \mid \lambda)$ as the sum of terminal forward variables $\alpha_T(i)$. 

70
Figure 3.20: Sequence of operation in computing Forward variable

The Forward-Backward Procedure: Consider the backward variable $\beta_t(i)$ defined as

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots, O_T | i_t = i, \lambda)$$  \hspace{1cm} (23)

where the probability of the observation sequence from $t+1$ to $T$ given the state $i$ at time $t$ and the model $\lambda$. This distinction has been made to be able to combine the forward and backward variables to produce useful result. As for forward variable $\alpha_t(i)$, backward variable $\beta_t(i)$ can be defined as follows:

**Step 1:**
$$\beta_t(i) = 1, \ 1 < i < N$$  \hspace{1cm} (24)

**Step 2:**

$$\beta_{t+1}(j) = \left\{ \sum_{i=1}^{N} a_{ij} b_j(O_{t+1},) \beta_{t+1}(j) \right\}$$  \hspace{1cm} (25)
Step 3:

\[ P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_i(i) \]  

(26)

The proof of Equation (24) and Equation (25) is similar to the one in Equation (21) and Eq. (22). The computation of \( P(O \mid \lambda) \) using \( \beta_i(i) \) is in the order of \( N^2T \) calculations. Hence both the forward as well as backward method is equally efficient for the computation of \( P(O \mid \lambda) \). Therefore this solves the problem 1.

B) Problem 2

Problem 2 is to find the “optimal” sequence associated with the given observation sequence, \( I \) that will maximize \( P(O, I \mid \lambda) \). The famous algorithm to solve this is called Viterbi algorithm. This algorithm is able to keep possible state sequence for each of the \( N \) states as the intermediate state for the desired observation sequence \( O = O_1, O_2, \ldots, O_T \). This is able to find the best path for each of the \( N \) states as the last state for the desired observation sequence. Lastly, identify and select the highest probability path.

As for example in Figure 3.21, assume an observation sequence \( O_1, \ldots, O_4 \) of length four. At time \( t=2 \), state 1 can be visited from any of the three states that time \( t=1 \). The total weight on each state is the summation of previous state and path. The process is always start from state 1 to state 2 and same procedure is repeated for state 2 and state 3. Finally, select out the minimum cost of all the states. As shown in the diagram below, total cost of state 1 is 11. Some to back tracking the sequence of states through state 1 at time \( t=4 \) the
observation sequence has highest probability of occurrence. Therefore, this sequence is state 3, state 1, state 3 and state 1 as shown in red color line.

**Figure 3.21:** Minimum path is traced using Viterbi algorithm. The final minimum cost path is shown in red color.

Viterbi algorithm is defined as follows:

**Step 1:**
For $1 < i < N$

$$
\delta_1 (i) = - \ln (\pi_i) - \ln (b_i (O_1))
$$

$$
\psi_1 (i) = 0
$$

**Step 2:**
For $2 < t < T$ for $1 < j < N$

$$
\delta_t (j) = \min_{1 \leq i \leq N} [\delta_{t-1} (i) - \ln (a_{ij})] - \ln (b_j (O_t))
$$

$$
\psi_t (j) = \arg \min_{1 \leq i \leq N} [\delta_{t-1} (i) - \ln (a_{ij})]
$$

**Step 3:**

$$
P^* = \min_{1 \leq i \leq N} [\delta_T (i)]
$$
\[ q_T^* = \arg \min_{1 \leq i < N} [\delta_T (i)] \quad (32) \]

Step 4:
For \( t = T-1, T-2, ..., 1 \)

\[ q_T^* = \psi_{t+1} (q_{t+1})^* \quad (33) \]

The \( \exp (-P^*) \) gives the required state-optimized probability, and \( Q^* = \{q_1^*, q_2^*, ..., q_T^* \} \) is the optimal state sequence. The computation of Viterbi algorithm is to find the maximum value of \( P(Q, O | \lambda) \) over all \( Q \). Besides that, it also possible to obtain the state sequence at the same time.

C) Problem 3

The problem 3 is to adjust parameters \((A, B, \pi)\) to maximize the probability of the observation sequence given the model. It deals with training the HMM such that it encodes the observation sequence in such a way that it includes many characteristics similar to the given one be encountered later it should be able to identify it. It means that in order to determine these parameters of a given HMM, it is first necessary to make a rough guess at the parameter. Once this is done, more accurate (in maximum likelihood) parameters can be found by applying Baum-Welch algorithm.

By using Baum-Welch algorithm, the parameters of the model \( \lambda = (A, B, \pi) \) are adjusted so to increase \( P(O | \lambda) \) until a maximum value is reached. This optimization criterion is called the maximum likelihood criterion. The function \( P(O | \lambda) \) is called the likelihood function.
However, Baum-Welch algorithm is strictly related to forward-backward algorithm which it tries to reach the local maximum of the probability function $P(O, \lambda)$. Let the probability of being in state $S_i$ at time $t$, and state $S_j$ at time $t+1$ as Equation (34).

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$  \hspace{1cm} (34)

Its relation with forward and backward variables is;

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+2}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+2}) \beta_{t+1}(j)}$$  \hspace{1cm} (35)

Now define the posteriori probability, $\gamma$ of being in state $S_i$ at time $t$, given the observation sequence and the model as Equation (36). This equation and its relationship with forward-backward variables as Equation (37).

$$\gamma_t(i) = P(q_t = S_i | O, \lambda)$$  \hspace{1cm} (36)

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}$$  \hspace{1cm} (37)

The sum $\gamma_t(i)$ over index $t$ to get a quantity can be interpreted as the expected number of times that state $S_i$ is visited. The summation of $\xi_t(i,j)$ over $t$ can be interpreted as the expected number of transitions made from $S_i$ to $S_j$. Finally, the formulas for re-estimating the parameters $A$, $B$ and $\pi$ is given as Equation (38).

$$\bar{\pi}_i = \text{expected number of times in state } S_i \text{ at time } (t=1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from } S_i \text{ to } S_j}{\text{expected number of transitions from } S_i}$$
After the re-estimation of the model parameters, observation sequence $O$ is produced. Stop the iterative re-estimation procedure if value of $P(O|\lambda)$ is not higher than the value at the previous step. Otherwise repeat above process by using the new re-estimated parameter values is obtained.

3.5 Feature Extraction

Features extraction is a process of deriving meaningful features from the input sound wave (Deller et al., 2000). In Hidden Markov model the input are not phones but is a feature vector of spectral and acoustic features. This means that, user speech and a speech waveform is records. Then an input of sound wave is digitized or sliced up into frames of usually 10, 15 or 20 milliseconds. The digitation includes two processes which are sampling (number of samples taken per second) and quantization (real-valued number as an integer). After that, a set of spectral feature vectors is transformed. These spectral features give the information about how much energy in the signal is at different frequencies (Deller...
et al., 2000). For this reason, LPC and MFCC are among the popular features set in feature extraction. LPC and MFCC feature set are described as following sub chapter.

### 3.5.1 Linear Predictive Coding (LPC)

As mention above in section 3.0, LPC is one of the speech analysis techniques which is widely used in speech recognition area. According to (Howitt, 1995) speech analysis technique is used to digitize and compress the input, sound wave as in Figure 3.22. The LPC is based on the speech production source filter and so it assumes that this model is all-pole model (Karpov, 2003). Linear prediction model is used in LPC because it can exactly preserve magnitude spectral dynamics, which means information in the speech is not retain the phase characteristic (Deller et al., 2000).

![Figure 3.22: A simple speech production from an input sound through Linear Prediction System](image)

In order to know further about LPC, the paragraph below describes the LPC features. LPC is the speech analysis technique widely used in signal processing and speech processing. It provides a good model of the speech signal in encoding the quality of speech
at low bit rate and provides very accurate estimates of speech parameters. Based on (Rabiner & Juang, 1993) the voiced regions in all-pole model of LPC it is able to provide a good approximation to the vocal tract spectral envelope. Even though LPC is less effective in unvoiced speech it is still useful in speech recognition.

The main task of LPC is to extract the features by using linear prediction analysis and convert it into Cepstral Coefficients (LPPC). The idea behind LPC is that a given speech sample can be approximated with a linear combination of past $p$ speech samples (Karpov, 2003). The formula of LPC depends only on past values and is represented as Equation (39). Levinson-Durbin algorithm is used in LPC to estimate the linear prediction coefficients from a given speech input. Figure 3.23 show the diagram of LPC processor until produce the LPCC.

$$\hat{s}(n) = -\sum_{k=1}^{P} a_k \cdot s(n - k)$$  \hspace{1cm} (39)

![Figure 3.23: The diagram of LPC processor](image)
Firstly, speech input is input into system through a low-order digital system. The process here is to flatten the signal and to make it less susceptible to finite precision effects that occur later in the signal processing (Rabiner & Juang, 1993). Secondly, frame blocking is the process of pre emphasized signal into a block of frame. Overlapping is maintained in the frames to smoothen the frames. Thirdly, each individual frame is windowed to minimize discontinuous signal at the beginning and the end of each frame. Among all the techniques, Hamming window is the most common technique used in LPC.

Fourthly, autocorrelation analysis is applied to auto correlated each windowed frame. Fifthly, in LPC analysis process an autocorrelation frame is converted into a LPC parameter set. Sixthly, from one of the LPC parameter set above, LPC coefficients set is used to produce cepstral coefficients. Next, the LPC cepstral coefficients are weighed to minimize the sensitivities cause by low order cepstral coefficients and for high order cepstral coefficients cause to noise. Then temporal derivative is applied to improve the cepstral representation. Finally, cepstral coefficients or LPCC is successfully produced. LPCC is then used as the input for in training and speech recognition purpose.

3.5.2 Mel Frequency Cepstral Coefficients (MFCC)

Extracting speech input signal with either the information of words or sentences pronounced by the speaker is the main idea of speech recognition. Mel Frequency Cepstral Coefficients (MFCC) is a popular front end choice for several state of the art speech recognition systems and extracts the speaker phonemes in the pre segmented speech sentences (Seddik et al., 2004). This operation allows us to obtain a set of parameter or coefficients fewer in the number than the input samples, while maintaining, in most of the
cases, the correct representation of different units that constitute speech (Vergin et al., 1999). Besides LPC features set described above, one of the most popular set of parameters is the MFCC developed by (Davis & Mermelstein, 1980).

White and Neely (White & Neely, 1967) research showed that the choice of speech representations will affects the recognition results in an isolated word recognition system. Therefore a good choice of feature analysis model is important. Figure 3.24 show the overall process of MFCC.

![Figure 3.24: Mel Frequency Cepstral Coefficients (MFCC) extraction process](image)

As show diagram above, MFCC extraction is a multistage process which consists of multiple processes. First, pre-emphasis is applied in the speech signal to extract the speech signal. Pre-emphasis filter is used to decrease the voice's lower frequency branch, which often comes from the surrounding background during recording. Second, the pre-emphasized speech is blocked into frames using Hanning window. A Hamming Window is applied on each block in order to decrease the edge effects due to the windows cutting (Seddik et al., 2004). Frames typically overlap each other with 30-50% to avoid losing
information and smoothen the speech. Thirdly, Discrete Fourier Transform (DFT) will transform the windowed input signal from the time domain into frequency domain. For speech signal in frequency domain, coefficients called spectral coefficients or spectrum can be obtained. Fourthly, spectrum is filtered into a bank of mel-filters called mel-spectrum. Here, a fast Fourier transform is applied on the signal. A series of triangular filters is produced and distributed on a mel scale to smooth the signal.

Next, logarithm is applied to convert the multiplication of magnitude of Fourier transform into addition (Abushariah, 2006). The main task of logarithm energy computation is to compute the logarithm of the magnitude of the output of the mel filter bank into logarithm mel spectrum. Following this step by inverse discrete Fourier transform (IDFT) process to produce mel cepstrum on the logarithm of magnitude of mel filter bank output. Finally, MFCC is ready to form a feature vectors. MFCC are then calculated as show in Equation (40). This feature vectors is applied and useful for the next operation, training feature vector before the recognition phase in the speech.

\[
M = \frac{1000}{\log 2} \log \left(1 + \frac{f}{1000}\right)
\]

Equation (40)

For example, 20 triangular band pass filters were illustrated in Equation (41). MFCC were computed where \(M\) is the number of cepstrum coefficients, and \(x, k = 1, 2, \ldots 20\), represents the log-energy output of the \(k\)th filter (Davis & Mermelstein, 1980). More details on each process of MFCC are described in the following sub section.

\[
\text{MFCC}_i = \sum_{k=1}^{20} X_k \cos \left( i \left( k - \frac{1}{2} \right) \frac{\pi}{20} \right), \quad i = 1, 2, \ldots , M
\]

Equation (41)
3.6 **Hidden Markov Model Toolkit (HTK)**

Hidden Markov Model Toolkit is a tool for building the Hidden Markov Models. It is a toolkit that used to process a set of HMM model by using all the speak utterances for training. HTK is a tool most applied in speech recognition research activities and involve activities from speech analysis, training, testing and results analysis. HTK is also used in speech synthesis, DNA sequencing and character recognition system.

HTK is a toolkit first developed by Speech Vision and Robotics Group of the Cambridge University Engineering Department (CUED) in 1989 by Steve Young. It consists of a set of C library modules and tools which able to support variety types of speech input formats (i.e. wav, pcm and etc.), different feature extraction techniques (i.e. PLP, MFCC, LPC and etc) and speech recognition technologies (i.e. Baum Welch, Viterbi, Vector Quantization and etc.). The HTK version has been improved from version 1 in 1989 to version 3.4 today.

There are four phases in building different types of sub-word continuous speech recognition units. These phases cover data preparation, training, testing and analysis. During the training and building the HMM based speech recognition model, C commands in HTK library are called to execute the HTK activities. The trained HMM model will then be used as a HTK recognizer tool to recognize all the unknown utterances in the application during testing.
3.6.1 Data Preparation

There are three important files need to prepare and develop during data preparation stage. First, acoustic model is built by collecting all the known utterances. These utterances will be extracted into sound statistical representations for training purpose. All the utterances are extracted using HCopy tool into MFCC or LPC features based on the application configuration. Secondly, language model or word network is produced to store the probabilities sequences of each word. The toolkit HParse is used to produce the word network. Thirdly, a grammar file must be created before training. This grammar file contains of pre-defined words combination as show in Figure 3.25. Finally, a set of unknown speech utterances which consists of different genders and aged will be recorded for training and testing purpose at a later stage.

$$\textit{food} = \text{FRIED RICE} \mid \text{FRIED NOODLES} \mid \text{CHICKEN SANDWISHES};$$

$$\textit{number} = \text{ONE} \mid \text{TWO} \mid \text{THREE} \mid \text{FOUR} \mid \text{FIVE} \mid \text{SIX} \mid \text{SEVEN} \mid \text{EIGHT} \mid \text{NINE} \mid \text{TEN};$$

(SEND-START (I WANT TO ORDER $\textit{food} \mid \textit{number}$ (SET | ORDER)) SEND-END)

**Figure 3.25:** Sample of task grammar in HTK

3.6.2 Training

Next, HTK is used to build the HMMs for any desire topology. Figure 3.26 shows the overall process during HMM training. In order to build the HMMs, HMMs is initiated from flat start, initiate Viterbi search and forward and backward techniques.

During the flat start, a set of unknown utterances are segmented into uniform segmentation. All the phone models are initialized using HCompV to make sure each state means and variances are equal to global means and variance. All the uniform segmentation will be
stored into Master Label Files (MLF) which stored all the labeled utterances phone-level with segmentation information.

Figure 3.26: HMM training process in HTK

Immediately after flat start activity, Viterbi search for these labeled utterances is in process. \textit{HInit} and \textit{HRest} tools are applied to provide isolated word style training using the fully labeled data. During the \textit{HInit process}, a set of parameter are computed using segmental k-means procedure. Conversely, HCompV and HERest are used for continuous speech recognition. In the first cycle, the mean and variance are estimated for training. While in the second cycle, this uniform segmentation is replaced by Viterbi alignment and further re-estimated by \textit{HERest} tool.
To construct the sub-word units, HMM definition editor (*HHEd* tool) will be used to clone the models into context-dependent and context-independent. After that, apply *HERest* tool for multiple times to re-estimate the modified set above.

### 3.6.3 Recognition

During the recognition stage, Viterbi search based on speech recognition is applied. This process will perform the Viterbi search by passing the algorithm to search the best match sequence of speech input from unknown speakers. *HVite* takes the trained HMMs monophones or tied-state triphones, word sequence network, dictionary and pre-defined phones list as an input for recognition process. Based on the input files, *HVite* tool will convert the word network into phone network and matches against each phone with HMM definition. Finally, speech input will be recognized and for each successful recognition the words will be stored in MLF file.

### 3.6.4 Analysis

HTK also has the tool to analysis the recognition result. *HResults* tool is used to evaluate the recognition performance. List of phones and MLF file which contain all the possible labeled utterances are used as an input during analysis. The overall recognition performance is calculated by using *HResults* tool based on word substitution, deletion and insertion errors found during recognition. Number of error counts will be displayed upon recognition.
There are two ways to compute the recognition rate; words correctness percentage and words accuracy percentage. Both results are compared against original word level transcriptions with speech input and output various statistics for analysis.

3.7 Summary

From the study, most of Malay speech recognition systems are focused on isolated digit recognition, which is recognized zero to nine and developed under speaker dependent. For this study, continuous speech recognition system for medium vocabulary based on Malay sentences is developed. There are many ways to recognize the speech input. The main modeling and classification technique consists of DTW, NN and HMM. DTW is suitable for use in small vocabulary and extend the usage of isolated to continuous speech recognition besides limited to word template. While NN is a static data classification and not suitable for temporal varying speech signal. Therefore, HMM is a statistical model temporal and acoustic variability of speech is the most powerful and widely used in speech recognition. Among all the HMM architecture, CDHMM is used in this study, because CDHMM gives the best result in large database. It is able to eliminate the VQ error effect of DHMM and increase the speech recognition accuracy along with powerful modeling of acoustic variability with its Gaussian mixture densities. While in speech unit modeling, sub-word unit modeling is used to present HMM. There are two sub-word units used in this study which are monophone and triphone. Compared to other sub-word unit, monophone is easier to train whereas triphone is shareable across different words and is surrounded by two phones.