AN AUTOPOIETIC APPROACH TO THE DEVELOPMENT OF SPEECH RECOGNITION

(PENDEKATAN AUTOPOIETIC DALAM PEMBANGUNAN PENGECAMAN SUARA)

ABD MANAN BIN AHMAD

RESEARCH VOT NO:
71703

Jabatan Kejuruteraan Perisian
Fakulti Sains Komputer dan Sistem Maklumat
Universiti Teknologi Malaysia

2006

Lampiran 20

UTM/RMC/F/0024 (1998)

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : **AN AUTOPOIETIC APPROACH TO THE DEVELOPMENT OF SPEECH RECOGNITION**

Saya **ABD MANAN BIN AHMAD**
(HURUF BESAR)

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. * Sila tandakan (/)

[ ]	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).
[ ]	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).
[X]	TIDAK TERHAD	

TANDATANGAN KETUA PENYELIDIK

Abd. Manan Ahmad
Fakulti Sains Komputer Dan Sistem Maklumat
Universiti Teknologi Malaysia

Nama & Cop Ketua Penyelidik

Tarikh : 01 - 12 - 2006

CATATAN : **Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/ organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan*

ABSTRACT

The focus of research here is on the implementation of speech recognition through an autopoietic approach. The work done here has culminated in the introduction of a neural network architecture named Homunculus Network. This network was used in the development of a speech recognition system for Bahasa Melayu. The speech recognition system is an isolated-word, phoneme-level speech recognizer that is speaker independent and has a vocabulary of 15 words. The research done has identified some issues worth further work later. These issues are also the basis for the design and the development of the new autopoietic speech recognition system.

ABSTRAK

Kajian yang dijalankan disini berfokus kepada implementasi pengecaman suara melalui kaedah autopoiesis. Hasil daripada kerja yang dilaksanakan itu telah membawa satu rangkaian neural yang dinamakan Homunculus Network. Rangkaian ini telah digunakan dalam pembangunan satu sistem pengecaman suara untuk Bahasa Melayu. Sistem pengecaman suara ini adalah berdasarkan ayat-terasing dan pada paras suku-kata. Sistem ini juga bebas dari pentutur dan mempunyai 15 patah perkataan. Kajian yang dijalankan juga telah mengenalpasti beberapa isu yang memerlukan perhatian seterusnya. Isu-isu ini yang telah dijadikan tapak bagi rekaan dan pembangunan satu sistem pengecaman suara autopoiesis baru.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AbS	-	analysis-by-synthesis
A/D		analog to- digital
ANN		Artificial Neural Networks
ART		Adaptive Resonance Theory
AR		autoregressive
Bps		bits per second
BSB		Brain-state-in a- box
CELP	-	Code Excited Linear Predictive Coding
dB	-	Decibels
DTW	-	Dynamic Time Warping
FFT	-	Fast Fourier Transform
Hz	-	Hertz
PLP	-	Perceptual Linear Prediction
PDP		Parallel Distributed Processing
LTF	-	long-term filter
LPC	-	Linear Predictive Coding
MPE-LPC		Multipulse-excitation linear predictive coding
MFCC	-	Mel Frequency Cepstrum Coefficients
MLP		Multilayer Perceptron
MT	-	Machine Translation
RELP	-	Residual Excited Linear Predictive Coding

RASTA	-	Relative Spectrum
RPE-LTP		Regular Pulse Excited-Long Term Prediction Coding
STF	-	short-term filters
SOM		Self Organizing Map
VSELP		Vector Sum Excited Linear Predictive Coding

LIST OF APPENDICE

CHAPTER I

PROJECT OVERVIEW

1.0 Introduction

The art of human-to-human communication has been honed to almost perfection through the millennia of man s progress in civilization. However, this cannot be said about the human to computer interaction. The interface between man and computer is still very much in a primitive phase. The keyboard, mouse and scanner are still the most important input devices. However, what of devices such as the microphone and the digital camera? Can all these devices ever be made into input devices that play a more important role apart from being just data streaming devices?

1.1.1 Preface

Many technological innovations rely upon user interface design to elevate their technical complexity to a usable product. Technology alone may not win user acceptance and subsequent marketability. The user experience, or how the user experiences the end product, is the key to acceptance. And that is where user interface enters the design

process. When applied to computer software, user interface design, also known as Human-Computer Interaction or HCI, refers to many products where the user interacts with controls or displays. Military aircraft, vehicles, airports, audio equipment, and computer peripherals, are a few products that require vast improvements into their user interface design. Therefore, speech recognition is but one of the many technological breakthroughs in the development of better, improved user interfaces.

1.2 Cognition and Perception

The subjective nature of perception, and hence of cognition, has attracted the attention of philosophers since days of Aristotle (1952). Perception is one of the oldest fields within scientific psychology, and there are correspondingly many theories about its underlying processes. The oldest quantitative law in psychology is the Weber-Fechner Law, which quantifies the relationship between the intensity of physical stimuli and their perceptual effects. It was the study of perception that gave rise to the Gestalt school of psychology, with its emphasis on holistic approaches. Many cognitive psychologists hold that, as we move about in the world, we create a model of how the world works. That is, we sense the objective world, but our sensations map to percepts, and these percepts are provisional, in the same sense that scientific hypotheses are provisional. As we acquire new information, our percepts shift. Pais (1997) refers to the 'esemplastic' nature of imagination. In the case of visual perception, some people can actually see the percept shift in their mind's eye. Others, who are not picture thinkers, may not necessarily perceive the 'shape-shifting' as their world changes. The 'esemplastic' nature has been shown by experiment: an ambiguous image has multiple interpretations on the perceptual level. Just as one object can give rise to multiple percepts, so an object may fail to give rise to any percept at all: if the percept has no grounding in a person's experience, the person may literally not perceive it.

1.3 Background on Speech

Human speech is basically an acoustic signal produced by the human vocal mechanism. The workings of this mechanism depend solely on the differential in air pressure of the chest cavity. Voiced sounds of speech are actually produced by the vibratory action of the vocal cords (Flanagan, 1972). This vibration is the effect of the oscillation of the cord due to the air flowing through the vocal tract and the local pressures that this air flow causes. Therefore, human speech could be easily analyzed from the aspects of frequency, amplitude, harmonic structure and resonance of the oscillation produced (Markowitz, 1996). Frequency and amplitude refers to the physical nature of the acoustic signal of speech. Frequency is the rate at which the signal oscillates and is measured in Hertz (Hz); or cycles per second. Amplitude, on the other hand, refers to the strength, or loudness, of the signal propagated across a medium and is measured in decibels (dB). These two measurements of speech would form the basis of the speech recognition model proposed. Speech recognition is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words (Zue et. al, 1995).

These recognized words may be used directly as in command and control applications and data entry applications or may be further processed, linguistically, in order to achieve objectives such as Machine Translation (MT). Therefore, speech recognition systems can be characterized by many parameters such as the mode of speech recognized, the style of speech to be processed, the need for speaker dependency, the size of the vocabulary of words involved, the language model to be used, the adverse conditions in which the speech was produced, and the perplexity of the grammar involved.

Basically, there are two aspects of spoken language used that are critical to its effectiveness as a mode of communication among humans. The first, being that the redundancy in representation of a message provides a robust method of transmitting information. There is also a need for extensive knowledge of linguistic structures, methods and styles of communication as well as knowledge of the environment of the

listener and general concepts of the topics discussed in order for a good speech recognizer to function as human-like as possible. The second aspect is that spoken language is often used interactively by the speaker and listener. At times, the success of speech communication depends, too, on the ability of speaker and listener to exchange roles and interact by providing feedback (Nusbaum et. al, 1995).

Therefore, a better understanding of the representations and processes that mediate human speech perception will bring significant impact on the development of more effective speech recognition systems. Besides that, current speech recognition systems employ single representation of speech and single approach to segmentation and pattern comparison of speech (Nusbaum et. al, 1995). As compared to humans, who are adaptive and flexible, current speech recognition lack the human touch to truly enable the technology to reach the capabilities of that of human listeners. It is this aspect of the failings of current speech recognition systems that is addressed here. It is hoped that with the introduction of autopoiesis, the return of the human factor into, what was once the domain of humans-only, speech recognition, by machines.

1.4 Problem Definition

The inherent problem with many speech recognition systems is that too much emphasis is given to the digital signal processing aspect of speech recognition and too little is understood about the human learning process on speech. Arguably, it is easier to treat the problem of speech recognition as a signal processing issue than that of a human cognitive process. Yet, the ultimate goal of a successful recognition system is that of attaining human-like recognition of speech rather than achieving advance capabilities in processing sound. Therefore, the answer to a speech recognition system that is robust, automatic and general-purpose lies in the ability to understand the human cognitive process.

1.5 Objective

The main objective of this research is to develop a working model of a speech recognition system. This is done through the analysis of the human speech process and identifying a representation of the human speech in a computer-generated environment. The proposed system could provide a new perspective into this field of study that has reached great maturity. After all, it does no harm to look at the feasibility of speech recognition from a cognitive processing point of view. It is important to note that there are an abundance of functional models in both the academia and commerce world, yet this should not be the stumbling block to creating another speech recognition model. After all, the research carried out is to bring improvement into speech recognition.

It is also the aim of this research to infuse autopoiesis into speech recognition. However the choice of autopoiesis is merely based on the notions of treating this recognition problem as one that uses self-organization as a solution perspective. Autopoietic theory promises an unconventional approach to the engineering of speech recognition because of the introduction of the possibilities of autonomy and self organization into speech recognition. With the incorporation of autopoiesis into conventional speech recognition methods, it is hoped that the resulting performance would pave the way for further and better understanding into both subject matters of speech recognition and autopoietic theory.

Autopoiesis is perhaps apt at handling the regulation of the learning process involved in the training of a speech recognition system. However, this does not mean that other important disciplines such as signal processing and cognitive processing techniques are to be ignored. It is in the treatment of the recognition system as a whole, with its autopoietic regulatory mechanisms, spectral analysis components and neural networked clusterings, that a new order of speech recognition systems would immerge and perhaps help bring the field of speech recognition to greater heights. This too is the aim of the research carried out; that is to produce mathematical models as well as functional

prototypes that could strengthen and justify the application of all related disciplines and theories put forward here.

1.6 Scope

The focus of the paper would be on the development of a speech recognition engine with autopoietic theory as the underlying foundation. Based on the highly phonetic Malay language, the engine is designed to accommodate a phoneme level recognition of the spoken input before the more complicated task of word matching is undertaken. The employment of autopoiesis is to facilitate a self-regulating system in terms of learning the inputs and generating the outputs. A design capable of implementation on a computer that is moderately fast with plenty of memory resources would be the product of this research. Therefore, the resulting working model would be a prototype implemented on a Pentium IV based PC with moderate processing power.

For a more focus approach to this research, the speech recognition engine to be developed would meet the dimensions of speaker-independence, discrete speech, and a vocabulary of eight to fifteen words in an environment that has tolerable conditions such as low noise levels and slight acoustical distortions. As the scope of vocabulary is an important issue, it has been decided that the target domain for this particular prototype of the system would be on automotive control. Therefore, the related words used to train and test the system would be based on commonly used words such as digits (in Malay) and simple commands such as kiri (left), kanan (right), and brek (brake).

Therefore, the scope defined in this research is suffice, if not enough, to guarantee completion and success. As such, it is important to stress on the fact that some trade off in performance is made in the process of developing the prototype for prompt results. Therefore, any limitation that is imposed on the prototype will be considered as an issue worth embarking upon in future work on the subject matter. After all, the work carried out here should be seeing as on-going and is filled with potential for improvements.

1.7 Summary

Speech recognition systems can be characterized by many parameters. An isolated-word speech recognition system requires that the speaker pause briefly between words, whereas a continuous speech recognition system does not. Spontaneous, or extemporaneously generated, speech contains disfluencies, and is much more difficult to recognize than speech read from script. Some systems require speaker enrollment; that is a user must provide samples of his or her speech before using them, whereas other systems are said to be speaker-independent, in that no enrollment is necessary. Some of the other parameters depend on the specific task. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words.

When speech is produced in a sequence of words, language models or artificial grammars are used to restrict the combination of words. This falls more into the domain of applying a strong human touch to the problem of speech recognition. After all, speech is the one gift of God that is unique only to man. As man go future, climb higher, run swifter, we need to look into ourselves for the answers we seek. Thus it is important to know enough to be inspired by the human cognitive process in order to recreate such a powerful and wonderful gift as speech recognition.

CHAPTER 2

SPEECH RECOGNITION AND AUTOPOIESIS

2.0 Introduction

This chapter will discuss about the fundamentals of speech recognition and the technologies that have evolved to provide for better speech recognition performance. Focus will be given to the main components that make up a speech recognition system; namely the feature extraction and acoustic modeling components.

Beside that we will discuss about Autopoiesis. It is somewhat a understated, underestimated theory. It was first developed to explain the essence of life and what it is that makes up a living organism. However, this theory can be developed to a degree that far exceeds that of a simple cognitive or learning theory. This type of promises is what makes autopoiesis such an interesting concept to explore.

The further description about these two parts will be describes in the next sections.

2.1 The Introduction of Speech Recognition

Developing a generic speech recognition system would require the primary components of an input mechanism, a signal processing engine, a cognitive mechanism, and an output device. The input mechanism receives the raw speech for the system and prepares this raw input for further processing from the signal processing engine. In general, most speech recognition system samples the raw speech signal at a rate between 4 kHz and 20 kHz (Hunt, 1995; Spina & Zue, 1997; Tebelskis, 1995). The speech is then transformed to frames at intervals of 10 or 20 ms for the purpose of simplification of contributory factors and compression of the signal produced. This transformation is done by the signal-processing engine and it involves techniques such as Fast Fourier Transform (FFT), Perceptual Linear Prediction (PLP), and Linear Predictive Coding (LPC) (Hunt, 1995; Tebelskis, 1995). Further references on LPC would be discussed in Section 2.6. From the speech frames generated, cognition of the frames would involve two aspects; i.e. the acoustic variability and the temporal variability (Tebelskis, 1995).

The acoustic variability refers to the difference in pitch, volume, pronunciation, and so forth. The current approaches to handling cognition from the acoustic variability point of view are the template-based approach, the knowledge-based approach and the statistical-based approach (Tebelskis, 1995). On the other hand, temporal variability refers to the rate of speech and has been proven that Dynamic Time Warping (DTW) algorithm, which would be discussed further in Section 2.5, is the solution to the issue. The output of this recognition system can be directly or indirectly applied to the areas of robotic control, machine translation, automated dictation and other language dependent systems.

2.1.1 Fundamentals of Speech Recognition System

Basically, speech recognition is made out of multiple disciplines such as artificial intelligence, linguistics, signal processing, cognitive science, statistics, and other hybrid, inter-related disciplines of knowledge in speech and learning. In recent years, some speech recognition systems could achieve performances beyond the accuracy of 90% (Bose and Liang, 1996; Zue, *et. al.*, 1995). However, these high performance systems manage their success through trade-offs in certain criteria of the speech recognition dimensions; be it speaker-dependence, limited vocabulary, discrete speech or low signal-to-noise ratio. Generally, the model of the speech recognition engine has three main components; namely a signal analysis engine, an acoustic modeling mechanism and a word matching function (Liew & Manan, 2000). The relationship of these components can be observed in Figure 2.1.

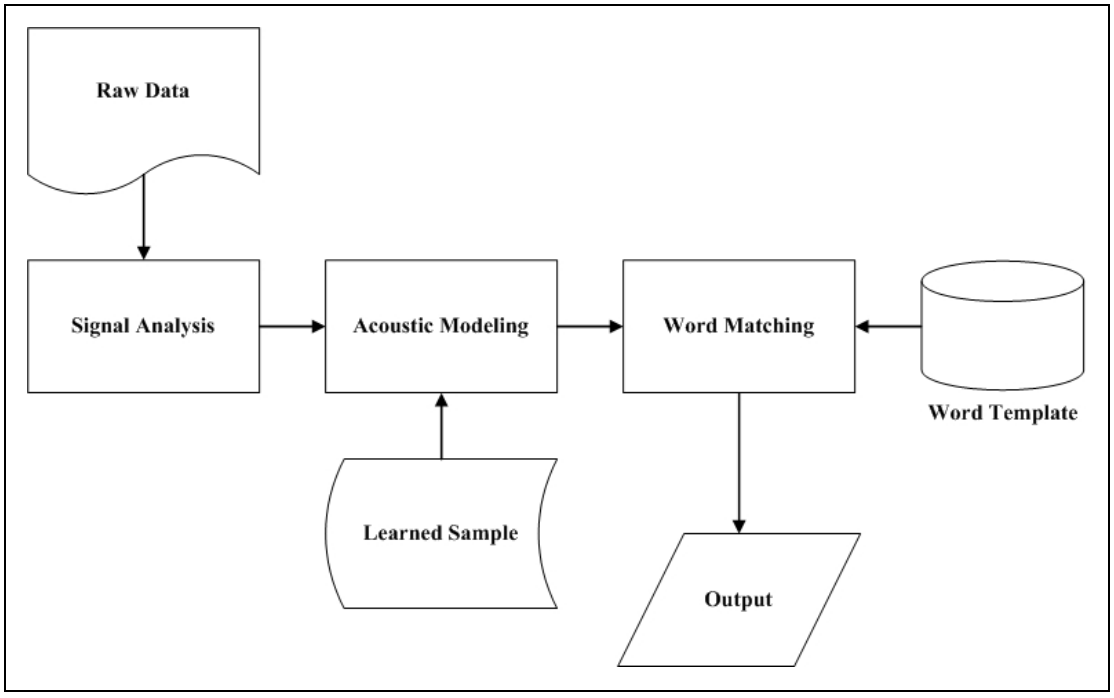


Figure 2.1: Components and their Relationship in the System

The signal analysis engine of the system is aimed at the feature extractions of the raw input speech, that is captured through audio acquisition devices such as microphones, to input vectors for the purpose of recognition. Initially, the speech input will be processed from digital signal processing techniques discussed in Section 2.5 to form the signal coefficients. Later, the speech frames, generated from the coefficients, will be produced in accordance with the segmentations of the speech input. These coefficients will be the input vectors for the acoustic modeling component of the system.

In the acoustic modeling mechanism, the derived speech frames will be utilized to learn the spoken input. This mechanism will determine whether the speech frames are for training or testing as the particular frame may carry a phoneme that has being learned prior to this. With the regulation of the learning process from autopoiesis, the modeling of the language of the input speech from a phonetic perspective can easily be carried out. Hence, the emphasis of the project will be on this component of the speech recognition system. This component will determine the approach of the development and also the essence of the research. The learning and recognition of the speech also happens within the functionality of this component. On the other hand, the word matching function of the system plays a secondary role to the acoustic modeling component. In the word matching component, the samples of speech, that has been learned and identified, are matched with current speech samples in the database for correctness and clarity of output. This stage in the recognition process tries to enforce a controlling mechanism on the lexical content of the possible outputs. Besides, word matching can lessen spelling error because of the way the acoustic model recognizes the speech from a phoneme to alphabet approach. Thus, word matching exists as a complementary to the main component of acoustic modeling. Nevertheless, proper steps must be taken to ensure that this component does not impose too great a constraint on the performance of the acoustic model.

The word templates and the learned samples of the system are basically derived from prior inputs. Therefore, these components will comprise speech databases of varying degrees of representation of the speech signal in accordance to the lexical content of the

signal. Generically, the database will be stored in a format that most benefits the system in terms of space and time complexity. After all, in developing such speech database, the emphasis is on the performance of the retrieval of the information stored. Accuracy and efficiency in retrieval will affect the response of the system on the whole. Thus, a well planned database is required before development of the system begins.

2.1.2 Handling Speech Signals

Speech is a varied and complex multidimensional signal whose acoustic properties include energy, frequency, and time (Kaplan, 1960). In a speech recognition system, the representation of speech is carried out for the purpose of sound discrimination rather than speech reproduction (O Shaughnessy, 1995). Therefore, a lossy digital compression of the speech signal for the speech recognition engine proposed is tolerable; as long as there exist an accurate representation of the signal. However, this highlights an interesting issue; the choice of digital compression or speech coding.

Currently, clean speech recognition systems commonly utilize Linear Predictive Coefficient (LPC) to encode speech signals where as robust speech recognition systems rely on Perceptual Linear Predictive (PLP) approach with, perhaps, a little help from Relative Spectrum (RASTA) for the same purpose (Wang and Pols, 1997). Other better known approaches to speech coding are Fast Fourier Transforms (FFT), Mel Frequency Cepstrum Coefficients (MFCC), Formant Vocoders, and S-Band Coding. All these approaches are, basically, spectral analysis methods of speech signals and deal in the frequency domain of the signal.

Irrespective of the signal analysis and coding method used, the speech signal for representation in the recognition system is captured through an audio acquisition device such as a microphone and relayed to the system to be learned. The format of representation is more important than the coding method because of the emphasis on

learning in this project. Thus the speech signal, as the input of the system, will be captured into frames that represent the individual phonemes of the speech. According to O Shaughnessy (1995), the utterance of a phoneme has the average duration of 80 ms and seldom last beyond 100 ms.

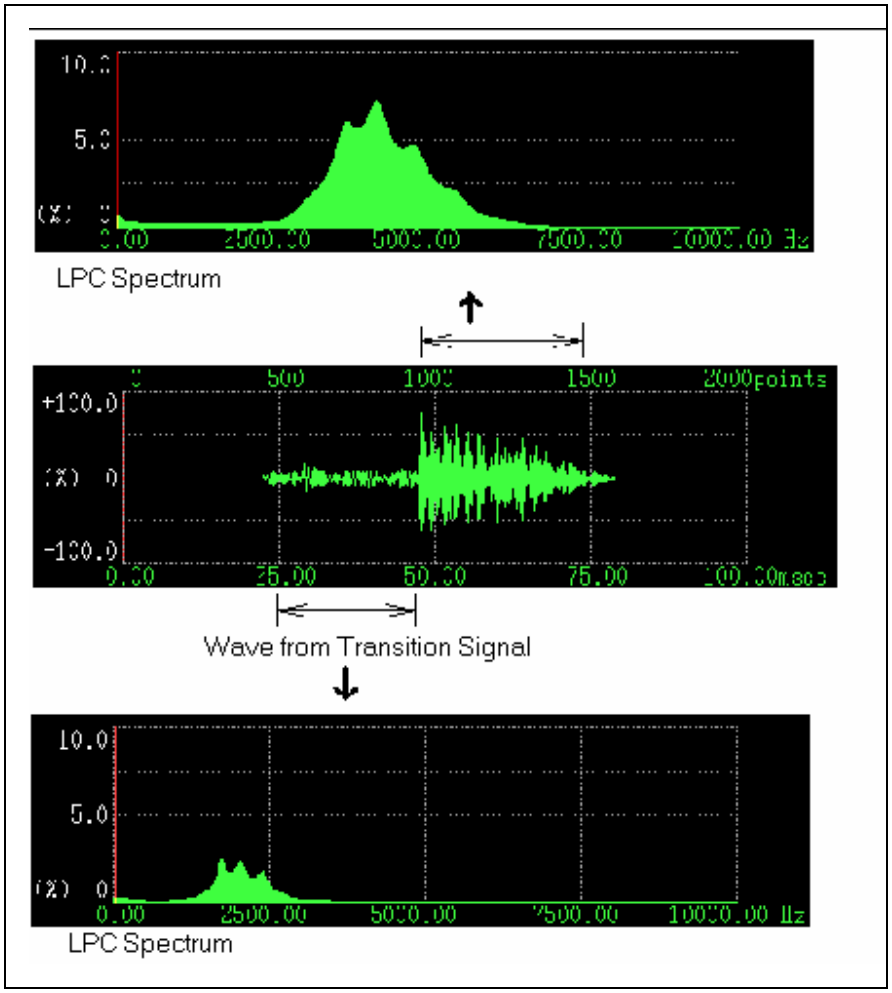


Figure 2.2 :Linear Predictive Coefficients derived from waveform of signal

2.1.3 Dynamic Time Warping Algorithm

Dynamic Time Warping (DTW) combines alignment and distance computation of signals such as speech through a dynamic programming procedure (Itakura, 1975; Sakoe & Chiba, 1978). Its time and space complexity is merely linear in the duration of the

speech sample and the vocabulary size. The algorithm makes a single pass through a matrix of frame scores while computing locally optimized segments of the global alignment path (Tebelskis, 1995). For example, consider two speech patterns **R** and **T** of *R* and *T* frames each; where **R** is the reference pattern and **T** is the test pattern. The DTW finds a warping function $m = w(n)$, which maps the time axis *n* of **T** into the time axis *m* of **R**. Frame by frame, DTW searches for the best frame in **R** against which to compare each test frame

The warping curve derives from the solution of an optimization problem

$$D = \sum_{n=1}^{T} d(T(n), R(w(n))) \tag{2.1}$$

Where each *d* term is a frame distance between the *n*th test frame and the $w(n)$th reference frame. *D* is the minimum distance measure corresponding to the best path $w(n)$ through a grid of *T* x *R* points.

The underlying assumptions of standard DTW are that global variations in speaking rate for someone uttering the same word on different occasions can be handled by linear time normalization, local rate variations within each utterance are small and can be dealt with using distance penalties known as local continuity constraints, each frame of the test utterance contributes equally to recognition, and a single distance measure applied uniformly across all frames is adequate (O Shaughnessy, 1995).

2.1.4 Linear Prediction

According to Coulter (2000), linear prediction is a family of techniques that attempts to remove redundancy in signals by working in the same domain as the original sound production model. It is the basis for virtually all very low bit rate speech coding. The fundamental idea of this technique is that voiced speech can be modeled quite accurately by exciting a filter with impulses at the pitch rate. Linear prediction is the

practical king of speech bit rate reduction so far and has found wide use in applications where low bit rate really matters, such as military secure speech and in internet telephones.

In linear prediction, the poor and synthetic quality of speech coded has led to what is known as the residual excitation approach to speech coding, a concept whereby not all the speech is synthesized, but a small part is transmitted as a coded waveform part of the original envelop, hence the name hybrid. The achieved quality of speech is improved but the penalty is the higher bit rate of transmission required. In LPC, there are many hybrid techniques, among others :- Residual Excited Linear Predictive Coding (RELP). RELP coding system has been developed for low to intermediate bit rate (4.8 to 16kbps) operation. RELP systems employ short-term (and in certain cases, longterm) linear prediction, to formulate a difference signal (residual) in a feed-forward manner. The early systems used baseband coding and transmitted a lowpass version of the residual. The decoder recover an approximation of the full-band residual signal, by employing high frequency regeneration which was subsequently used to synthesize output speech.

- Code Excited Linear Predictive Coding (CELP)
  Code or Codebook excitation linear predictive (CELP) coding employs a vocal tract LP-based model, a codebook based excitation model and an error criterion which serves to select an appropriate excitation sequence using an analysis-by-synthesis (AbS) optimization process. The system selects that excitation sequence which minimizes a perceptually weighted mean square error formed between the input and the locally decoded signals. The vocal tract model utilizes both a short-term filters (STF), which models the spectral envelope of speech, and a long-term filter (LTF), which accounts for pitch periodicity in voiced speech.

- Vector Sum Excited Linear Predictive Coding (VSELP)
  Vector Sum Excited Linear Predictive (VSELP) coding is the same as Regular-pulse excited speech coding but the details of the excitation analysis and bit assignments differ. With a source rate of 7.95kbps, the North American code is

more efficient than the 13kbps code of GSM. On the other hand, the signal processing hardware of the coder is more complex.

The coder derives a new linear prediction every 20ms. This linear predictor is characterized by 10log-area ratios, which are represented in aggregate by 38 bits. The frame energy accounts for 5 bits. The coder obtains long -term predictor coefficients at 5ms intervals, four times per 20ms block. As a form of codebook-excited linear prediction, the VSELP coder computes 14 bit of excitation information in every 5ms sub block. It also computes 8 bits of gain information, which together represent a scaling factor for the long-term predictor and two scaling factors for the vector sum codebook. All this adds up to 159 bit per 20ms or 7.95kbps.

- Regular Pulse Excited-Long Term Prediction Coding (RPE-LTP)
  To know about Regular Pulse Excited-Long Term Prediction Coding (RPELTP) first we must understand what is Multipulse-excitation linear predictive coding (MPE-LPC) systems. This system model the excitation signal as a sequence of irregularly spaced pulses. MPE coders employ a synthesis filter which consists of one or two autoregressive (AR) filters in series. The first filter models the 'smooth' spectral envelope of the signal (short-term filter) while the second (if used) models the harmonic (fine) structure of the spectrum (long-term filter). The parameters of the excitation model (i.e. the positions and amplitudes) and part of the synthesis filter i.e. the long-term filter are determined in a closed-loop optimization process.

  MPE-LPC coders provide near network quality speech (MOS=4) at bit rates in the range of 16 to 8kbps. Their performance deteriorates rapidly, however, at bit rates below 8kbps where acceptable performance can only be achieved by drastically modifying the basic multiple excitation model. The Regular Pulse Excited-Long term. Prediction Coding (RPE-LTP) is a special case of MPE-LPC coding. RPE-

LTP models the excitation signal with a sequence of equally spaced pulses. The performance of RPE systems is similar to that obtained from MPE coders.

2.1.5 Feature Extraction

The input of a speech recognition system is generally speech samples. Regardless of whether the samples are taken straight off an AC/DC source such as a microphone or are preprocessed and categorically represented such as from a wave file, these speech samples form a representation of the input signal that is too unwieldy for use in its raw form. Therefore, features of the input signal are usually extracted from such samples before any further classification or recognition is made. A simple illustration given by Morgan and Scofield states, For example, a one second speech signal may contain three words, each consisting of roughly five characters. At 8 bits/character, the speech could be encoded at 120 bits per second (bps). However, the speech signal is usually sampled at 10 kHz with a 14-bit analog to- digital (A/D) converter. This translates into 140,000 bps, over 1,000 times more information than is contained in the character representation. (Morgan & Scofield, 1991). In other words, certain aspects of a signal must be derived from the raw signal so as the representation of the signal is precise and concise and yet there is sufficient data of the signal so as not to alter the signals content. Thus, it is important that efficient signal processing techniques be employed so that useful features of the input speech samples are extracted. This preprocessing phase helps reduce computation complexity and optimizes the performance of the system. Among the many signal processing techniques, Linear Predictive Coding (LPC) is becoming the de facto signal encoding technique in speech recognition
systems. Therefore, the choice of utilizing LPC in the proposed speech recognition engine only comes naturally.

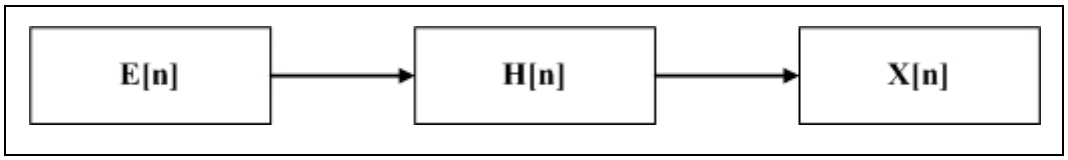


Figure 2.3: Basic source-filter model for speech signals

Before going into the details of LPC analysis, let s take a look at the decomposition of a speech signal *X[n]*. The speech signal is a product of an excitation source *E[n]* that has passed through a filter, in this case the human vocal cords, *H[n]*. Such a source-filter model is depicted in Figure 2.3. The aim of LPC would be to decompose such *X[n]* into some representation that is efficient.

2.1.6 Linear Predictive Coding

Linear Predictive Coding, also known as LPC analysis or auto-regressive (AR) modeling, is a very powerful technique for speech analysis (Huang, *et. al*., 2001).This technique is widely used because it is fast and simple, yet an effective way in estimating the main parameters of speech signals. However, this technique is not as suitable as other techniques such as Fast Fourier Transforms (FFT) when applied to general signal processing because (it) is not appropriate when the input is a harmonic process or a deterministic signal with a flat spectral envelope (Manolakis, *et al*, 2000)

Generally, as stated by Huang, the speech filter *H(z)* is modeled as

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)} \qquad \textbf{(2.2)}$$

where *a* is the predictor coefficients and *P* is the order of the LPC analysis (Huang, *et al*., 2001). The *inverse filter A(z)* is defined as

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \quad \textbf{(2.3)}$$

Taking the inverse *z*-transform in Equation 5.1 results in

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n] \quad \textbf{(2.4)}$$

where *e[n]* is the prediction error of the inversed filter.

Linear Predictive Coding gets its name from the fact that it predicts the current sample as a linear combination of its past *P* samples.

$$\tilde{x}[n] = \sum_{k=1}^{p} a_k x[n-k] \quad \textbf{(2.5)}$$

LPC analysis involves solving for the *ak* predictor coefficients according to a least mean squared error criterion; defined as the prediction error

$$e[n] = x[n] - \tilde{x}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k] \quad \textbf{(2.6)}$$

Thus, to estimate the predictor coefficients from a set of speech samples, short-term analysis technique is used. Consider *xm*[*n*] as a segment of speech in the vicinity of sample *m*.

$$x_m[n] = x[m+n] \quad \textbf{(2.7)}$$

The short-term prediction error for that segment is defined as

$$\begin{aligned} E_m &= \sum_n e_m^2[n] \\ &= \sum_n (x_m[n] - \tilde{x}_m[n])^2 \\ &= \sum_n (x_m[n] - \sum_{j=1}^{p} a_j x_m[n-j])^2 \end{aligned} \quad \textbf{(2.8)}$$

In the absence of the probability distribution of *ai*, a reasonable estimation criterion is the least mean squared error. Thus, given a signal *xm*[*n*], the predictor

coefficients are estimated through the minimization of the total prediction error *Em*. Taking the derivative of Equation 2.8 with respect to the predictor coefficients *aj* and setting it to zero, we have

$$\frac{\delta}{\delta x_m^i} E_m = \sum_n e_m[n] x_m[n-i] = 0, \qquad 1<i<P \qquad \textbf{(2.9)}$$

where *em* and x_m^i is taken to be vectors of samples and their inner product has to be 0. This condition, known as orthogonality principle, says that the predictor coefficients that minimize the prediction error are such that the error must be orthogonal to the past vectors.

Equation 2.9 can be redefined as a set of *P* linear equations

$$\sum_n x_m[n-i] x_m[n] = \sum_{j=1}^{p} a_j \sum_n x_m[n-i] x_m[n-j] \qquad i=1,2,\ldots,p \qquad \textbf{(2.10)}$$

Let $\phi_m[i, j] = \sum_n x_m[n-i] x_m[n-j]$, we obtain the *Yule-Walker* equation

$$\sum_{j=1}^{P} a_j \phi_m[i, j] = \phi_m[i, 0] \qquad i=1,2,\ldots,P \qquad \textbf{(2.11)}$$

Solution of the set of *P* linear equations results in *P* LPC coefficients that minimize the predictor error. With *aj* satisfying Equation 2.11, the total prediction error in Equation 2.8 takes on the following value:

$$\begin{aligned} E_m &= \sum_n x_m^2[n] - \sum_{j=1}^{p} a_j \sum_n x_m[n] x_m \\ &= \phi[0,0] - \sum_{j-1}^{p} a_j \phi[0, j] \end{aligned} \qquad \textbf{(2.12)}$$

It is convenient to define a normalized prediction error $u[n]$ with unity energy

$$\sum_n u_m^2[n] = 1 \qquad \textbf{(2.13)}$$

and a gain *G*, such that

$$e_m[n] = Gu_m[n] \tag{2.14}$$

The gain *G* can be computed from the short-term prediction error

$$E_m = \sum_n e_m^2[n] = G^2 \sum_n u_m^2[n] = G^2 \tag{2.15}$$

The solution to the *Yule-Walker* equation defined in Equation 2.11 can be achieved with matrix inversion. However, due to the nature of the matrix, some efficient algorithms are specifically well-suited to the solution. Among the three different algorithms mentioned by Huang, the *autocorrelation method* is chosen for this project (Huang, *et al*., 2001).

2.1.6.1 Autocorrelation Method

In the definition of [*i, j*] *m* for Equation 2.11, no consideration was given to the range of the sampling of the speech. However, when dealing with windowed speech, the boundary effects need to be taken into account in order to avoid large prediction errors at the edges. Therefore, [*i, j*] *m* can be rewritten as

$$\begin{aligned} \phi_m[i, j] &= \sum_{n=0}^{N+P-1} x_m[n-i]x_m[n-j] \\ &= \sum_{n=0}^{N-1-(i-j)} x_m[n]x_m[n+i-j] \end{aligned} \tag{2.16}$$

Now, [*i, j*] *m* is only dependent on the difference on *i-j* and can be written in terms of the autocorrelation function [*i, j*] *R* [*i j*] *m m* with *Rm*[*k*] being the autocorrelation sequence of *xm*[*n*]

$$R_m[k] = \sum_{k=0}^{N-1-k} x_m[n]x_m[n+k] \tag{2.17}$$

Combining Equation 2.11 and Equation 2.17,

$$\sum_{j=1}^{P} a_j R_m[|\, i-j\,|] = R_m[i] \qquad \textbf{(2.18)}$$

which corresponds to the following matrix equation

(2.19)

This equation forms a *Toeplitz* matrix that can be inverted through Durbin s recursion.

2.1.6.2 Durbin s Recursion Algorithm

a.) Initialization

$$E^0 = R[0] \qquad \textbf{(2.20)}$$

b.) Iteration

For i = 1, , P do the following recursion

$$k_i = \left(R[i] - \sum_{j=1}^{i-1} a_j^{i-1} R[i-j] \right) / E^{i-1} \qquad \textbf{(2.21)}$$

$$a_i^i = k_i \qquad \textbf{(2.22)}$$

$$a_j^{i-1} - k_i a_{i-j,}^{i-1} \qquad \textbf{(2.23)}$$

$$E^i = (1 - k_i^{\,2}) E^{i-1} \qquad \textbf{(2.24)}$$

c.) Final Solution

$$a_j = a_{j,}^P \qquad \textbf{(2.25)}$$

where the coefficients *ki*, called *reflection coefficients*, are bounded between -1 and 1. In the process of computing the predictor coefficients of order *P*, the recursion finds the solution of the predictor coefficients for all order less than *P*.

2.1.7 Phoneme Recognition

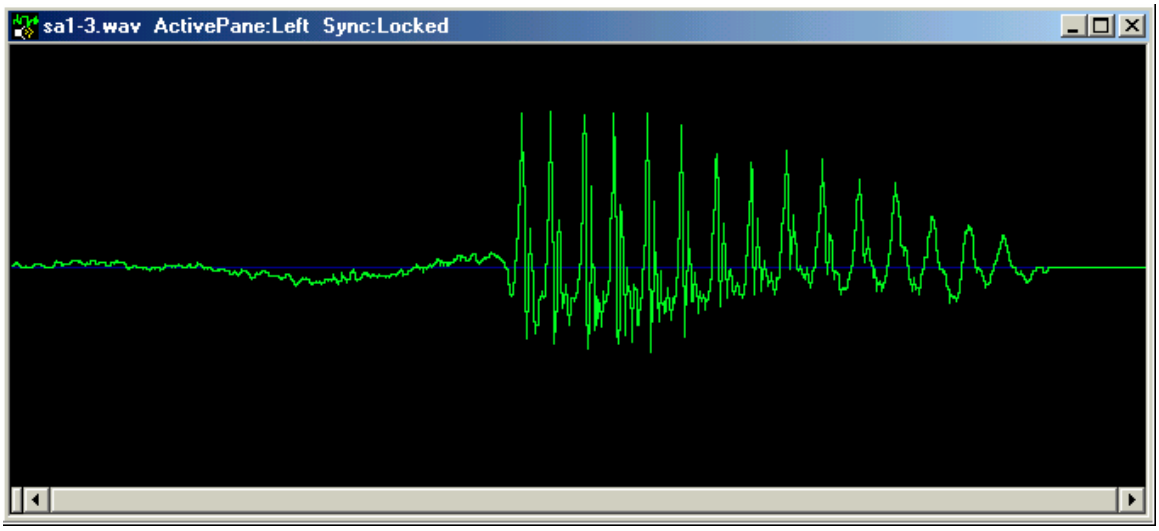


Figure 2.4 Wave form of the syllable ‘sa’

Figure 2.4 gives an idea of the wave form, represented in it s amplitude format, of the syllable sa in the word satu (meaning one in Malay). As can be seen, it is not easy identifying the segments of the wave form that represents the consonant *s* and the vowel *a* and where the starting and end points are. However, as stated in Section 2.2, an interval of 10 20 ms can be used for the simplification and compression of the signal. In the process of experimentation, it is found that an interval of 20 ms is capable of accommodating one utterance of the phoneme. For example, in the syllable sa , which consists of two phones *s* and *a*, the *s* phone only takes up about two utterance or sample frames of the syllable while the rest of the frames segmented with a 20 ms interval was for the phone *a*. That is to say that, out of 14 sample frames, only 2 is for *s* and 12 for *a*. This interesting finding offers the conclusion that for a speech recognition system that attempts at phoneme level recognition, an important feature of the system would be to

determine the variability of the transition of phones within an input signal. It is not just merely classifying the input signal but rather the knowing of where and when the signal changes states between different phones. This is where a new learning mechanism is proposed. It is also the focal point of the research done here. For want of a name, this learning mechanism will be called the Homunculus Network . Though the intentions of proposing this was never to criticize or replace the importance of Hidden Markov Models or its hybrids, the homunculus network is a novel way to look at the training of speech recognition systems, in particular, and as a learning architecture, in general. Further discussions on Homunculus Network will be discussed in Chapter 6.

2.1.8 Fundamentals of Neural Networks

Neural networks, or rather Artificial Neural Networks (ANN), also called Parallel Distributed Processing (PDP) systems and connectionist systems, are intended for modeling the organizational principles of the biological brain (Bose and Liang, 1996). According to Fausett (1994), neural networks have been developed as generalizations of mathematical models of human cognition or neural biology based on these assumptions:

- Information processing occurs at many simple elements called neurons
- Signals are passed between neurons over connection links
- Each connection link has an associated weight, which, in a typical network, is factored into the signal being propagated
- Each neuron applies an activation function to its net input to determine its output signal

A neural network is characterized by its pattern of connections between neurons as well as its method of determining the weights on the connections.

A neural network is made of simple processing elements called neurons, units, cells or simply nodes. These elements are usually grouped together to form

onedimensional layers of elements. As a neural network would have, at minimum, one layer of elements, these elements are connected to each other through links that are usually weighted. A typical neural network would have three layers; i.e. an input layer, a hidden layer, and an output layer. Figure 2.5 shows a network of Perceptrons in multiple layers.

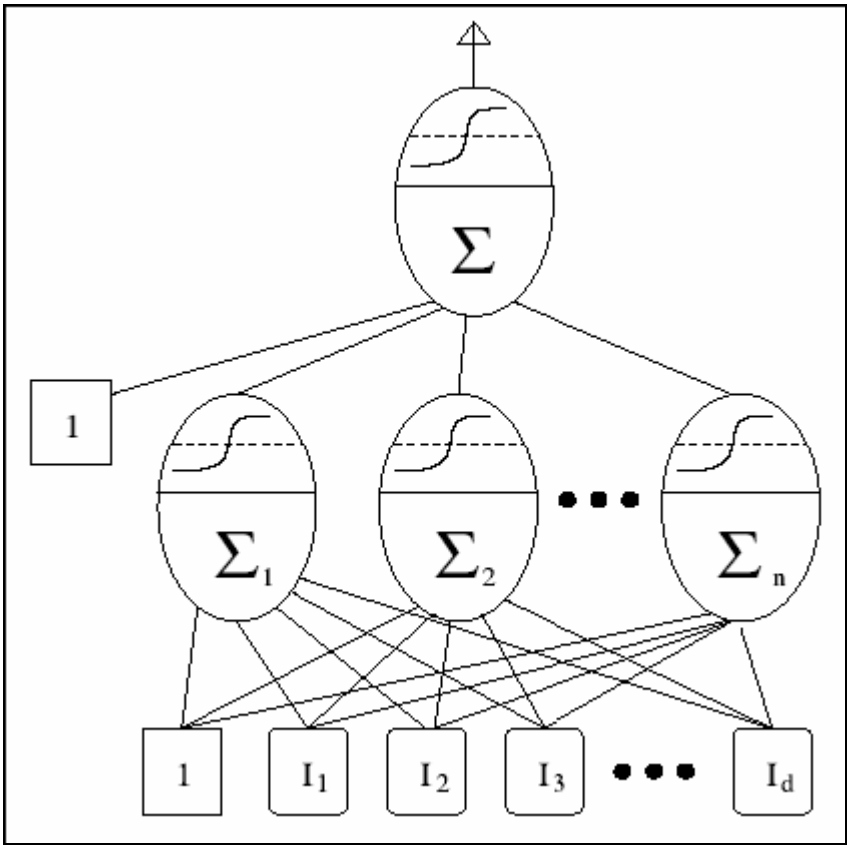


Figure 2.5 Simple multiple layers of Perceptrons

2.1.8.1 The Neuron

At the heart of every artificial neural network is the processing element called neurons, nodes, cells and etc. The neuron is analogous to the neuron nucleus of the biological brain. As is the case of the biological neuron, artificial neurons are themselves simple processing elements. However, when grouped into a network of neurons, the processing power of these neurons are amplified many folds. It is this processing power

that leads to the fertile research work being done in this field as well as the insights and in-roads made in developing Artificial Neural Networks.

2.1.8.1.1 Activation Functions

"*The basic operation of an artificial neuron involves summing its weighted input signal and applying an output, or activation, function*" (Fausett, 1994). For the input layer of elements, the typical function would be an identity function, $f(x) = x$, as illustrated in Figure 2.6a. This function translates the sum of all inputs (as there could be more than one input per element) as the output of the element. Figure 2.6b illustrates the step function. This function is often used to convert the sum of all inputs of the node to an output unit that is a binary (0 or 1) or bipolar (- 1 to 1) signal. The output value is determined through the matching of the input value to a threshold value, , that has been predefined and this threshold value can be altered through the course of training and learning of the neural network.

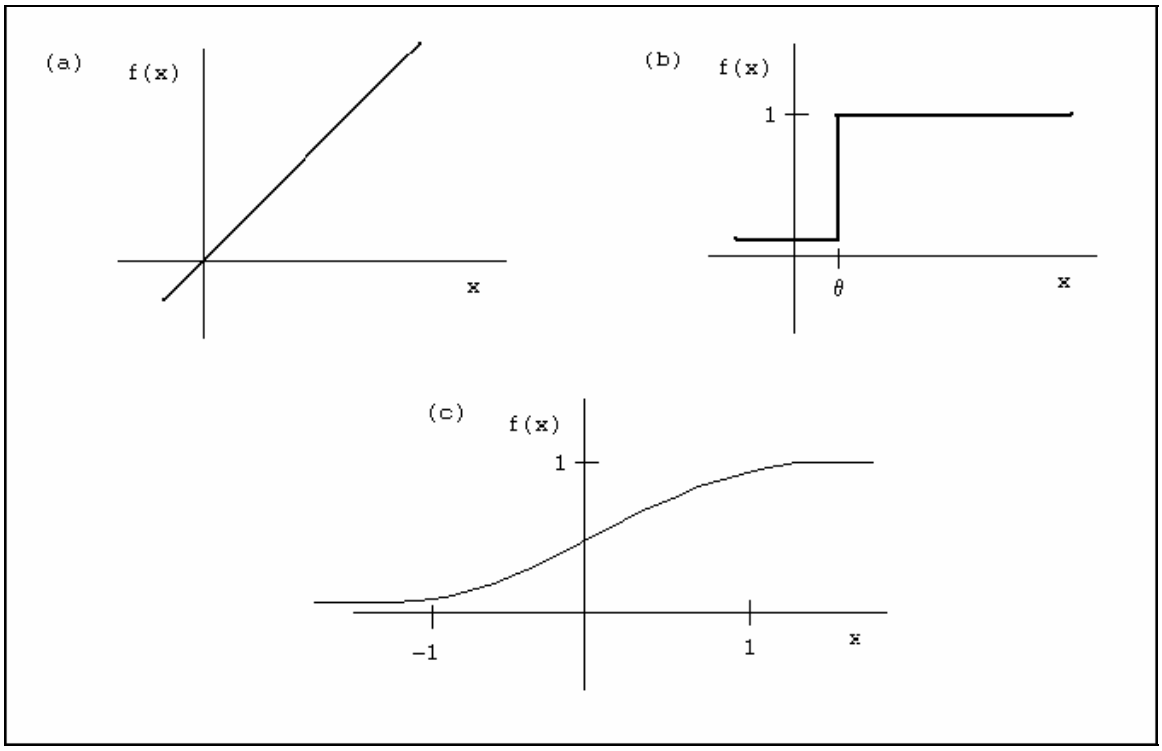


Figure 2.6 (a) identity function (b) step function (c) binary sigmoid function

Of the many types of activation function, by far the most versatile and most used function is the sigmoid function and of the many sigmoid functions, the logistic function and the hyperbolic function are the most common in use. Sigmoid functions are used typically in backpropagation trained networks because of the simple relationship between the value of the function at a point and the value of the derivative at that point reduces the computational burden during training (Fausett, 1994).

In Figure 2.6c, the logistic, or binary sigmoid, function is illustrated. This function is used when the desired output falls within the range of 0 to 1. The steepness of the slope of the function is controlled through the steepness parameter, .

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \qquad \textbf{(2.27)}$$

This binary sigmoid function could be scaled to accommodate other range of output values. If the desired output value should fall within 1 and 1, then a bipolar sigmoid, or hyperbolic tangent, function maybe used

$$f(x) = \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} \quad \textbf{(2.28)}$$

2.1.8.2 Weights

Weights are values associated to the connection or link between neural elements of a network. It is basically used to modify the signal strength being propagated across a network. The values of these weights are usually updated at the learning or training phase of the neural network. There are generally two methods of training a neural network; one is supervised training and the other is unsupervised training.

According to Fausett (1994), many of the task a neural network can perform fall into the areas of mapping, clustering and constrain optimization. On the other hand, pattern classification and pattern association may be considered special forms of the more typical problem of mapping input vectors to a specific output. All these areas of interest require the updating of weight values through proper means of training of the network. At times, training might be so extensive to the point that the number of training epochs may be used as the determining factor in the completion of a training cycle for the neural network.

2.1.8.2.1 Supervised Training

In supervised training, the process of training the network is accomplished by presenting a sequence of training vectors, or patterns, each with an associated target output vector. Apart from that, the weights are adjusted according to the learning algorithm applied to the network. Examples of supervised training networks are the Hebbian network and the Multilayer Perceptron (MLP) network with backpropagation. Therefore, supervised learning is more suited for situations where the set of possible output has been predetermined and the neural network is needed more as a filter than a classifier of problems.

2.1.8.2.2 Unsupervised Training

Unsupervised training, on the other hand, is more concentrated at grouping input vectors together without prior target output being determined. Typically, an unsupervised network modifies the weights to link similar input elements to the particular output elements or clusters of output elements. Examples of unsupervised training networks are Kohonen s Self Organizing Maps and Carpenter s Adaptive Resonance Theory networks. These networks are more suited for problems where the target output couldn t be predefined or that the output required is non-deterministic or has a polynomial level of complexity.

2.1.8.3 Neural Network Topologies

The arrangement of neural elements and their links can have a profound impact on the processing capabilities of the neural networks. In the typical neural network, the elements that receive input from the external environment are referred to as the input nodes. Many neural networks also have one or more layers of hidden elements that receive input only from other elements in the network, and these elements are not strictly

referring to input units alone. A layer of elements receives a vector of data or the output of a previous layer of elements and processes them in parallel. The set of elements that represents the final result of the neural network computation is designated as the output nodes. There are three major connection topologies that define how data flows between the input, hidden, and output nodes. These main categories are described in detail in the next sections.

2.1.8.3.1 Feed-Forward Networks

Feed-forward networks are used in situations when we can bring all of the information to bear on a problem at once, and we can present it to the neural network. In this type of neural network, the data flows through the network in one direction, and the answer is based solely on the current set of inputs.

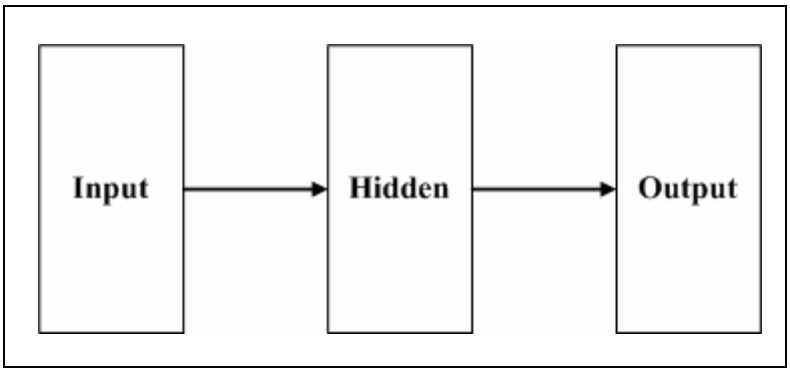


Figure 2.7 Feed-forward neural networks

In Figure 2.7, we see a typical feed-forward neural network topology. Data enters the neural network through the input units on the left. The input values are assigned to the input units as the unit activation values. The output values of the units are modulated by the connection weights, either being magnified if the connection weight is positive and greater than 1.0, or being diminished if the connection weight is between 0.0 and 1.0. If

the connection weight is negative, the signal is magnified or diminished in the opposite direction.

Each processing unit combines all of the input signals corning into the unit along with a threshold value. This total input signal is then passed through an activation function to determine the actual output of the processing unit, which in turn becomes the input to another layer of units in a multi-layer network. The most typical activation function used in neural networks is the S-shaped or sigmoid function. This function converts an input value to an output ranging from 0 to 1. The effect of the threshold weights is to shift the curve right or left, thereby making the output value higher or lower, depending on the sign of the threshold weight. As shown in Figure 2.7, the data flows from the input layer through zero, one, or more succeeding hidden layers and then to the output layer. In most networks, the units from one layer are fully connected to the units in the next layer. However, this is not a requirement of feed-forward neural networks. In some cases, especially when the neural network connections and weights are constructed from a rule or predicate form, there could be less connection weights than in a fully connected network. There are also techniques for pruning unnecessary weights from a neural network after it is trained. In general, less weights translates to a faster network and the network will be able to process data and converge more efficiently. It is important to remember that feed-forward is a definition of connection topology and data flow. It does not imply any specific type of activation function or training paradigm.

2.1.8.3.2 Limited Recurrent Networks

Recurrent networks are used in situations when we have current information to give the network, but the sequence of inputs is important, and we need the neural network to somehow store a record of the prior inputs and factor them in with the current data to produce an answer. In recurrent networks, information about past inputs is fed back into and mixed with the inputs through recurrent or feedback connections for hidden or output

units. In this way, the neural network contains a memory of the past inputs via the activations, as illustrated in Figure 2.8 Figure

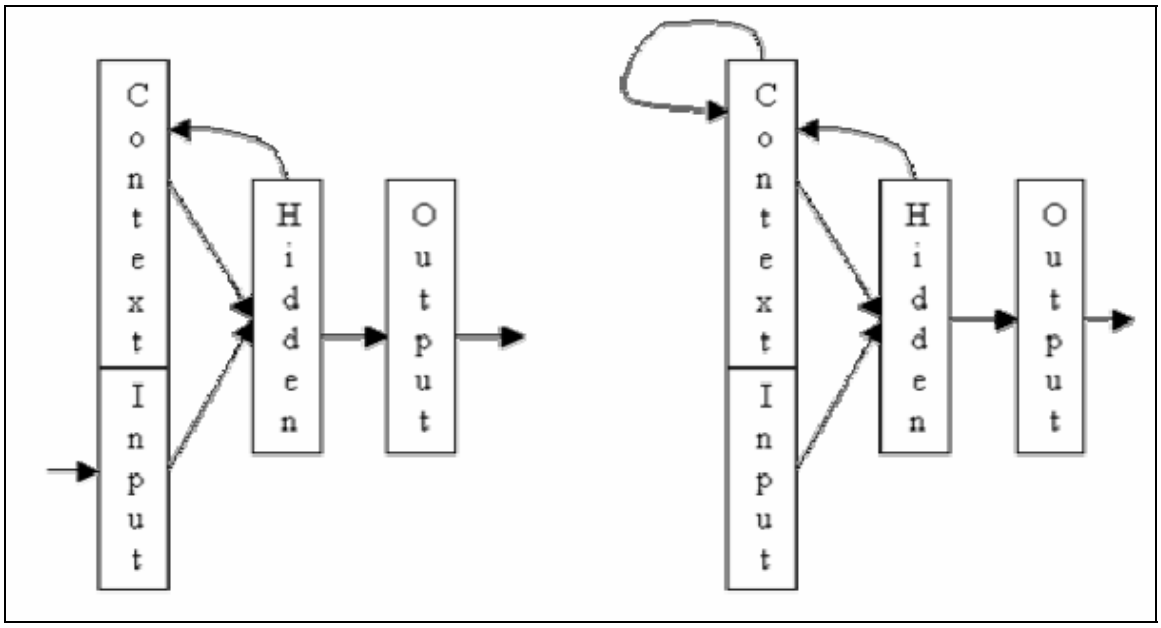


Figure 2.8 Partial recurrent neural networks

Two major architectures for limited recurrent networks are widely used (Zue, *et al*, 1995). The Elman model suggested allowing feedback from the hidden units to a set of additional inputs called context units. Prior to that, the Jordan model described a network with feedback from the output units back to a set of context units. This form of recurrence is a compromise between the simplicity of a feed-forward network and the complexity of a fully recurrent neural network because it still allows the popular back propagation training algorithm to be used.

2.1.8.3.3 Fully Recurrent Networks

Fully recurrent networks, as their name suggests, provide two-way connections between all processors in the neural network. A subset of the units is designated as the input processors, and they are assigned or clamped to the specified input values. The data

then flows to all adjacent connected units and circulates back and forth until the activation of the units stabilizes. Figure 2.9 shows the input units feeding into both the hidden units (if any) and the output units. The activations of the hidden and output units then are recomputed until the neural network stabilizes. At this point, the output values can be read from the output layer of processing units.

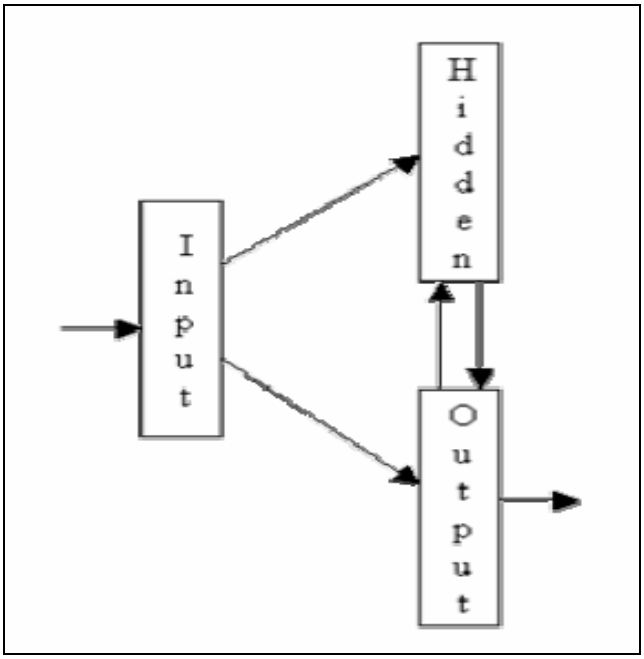


Figure 2.9 Fully recurrent neural networks

Fully recurrent networks are complex, dynamical systems, and they exhibit all of the power and instability associated with limit cycles and chaotic behavior of such systems. Unlike feed-forward network variants, which have a deterministic time to produce an output value (based on the time for the data to flow through the network), fully recurrent networks can take an in-determinate amount of time.

In the best case, the neural network will reverberate a few times and quickly settle into a stable, minimal energy state. At this time, the output values can be read from the output units. In less optimal circumstances, the network might cycle quite a few times

before it settles into an answer. In worst cases, the network will fall into a limit cycle, visiting the same set of answer states over and over without ever settling down. Another possibility is that the network will enter a chaotic pattern and never visit the same output state.

By placing some constraints on the connection weights, we can ensure that the network will enter a stable state. The connections between units must be symmetrical. Fully recurrent networks are used primarily for optimization problems and as associative memories. A nice attribute with optimization problems is that depending on the time available, you can choose to get the recurrent network s current answer or wait a longer

2.1.8.4 Perceptrons

The perceptron was introduced by Frank Rosenblatt at Cornell University in the late 1950 s (Bose and Liang, 1996; Fausett, 1994). The typical perceptron network consist of an input layer connected with fixed weights to associator neurons. Rosenblatt s perceptron evolved from two important neural network concepts of the 1940 s; i.e. McCullochs and Pitts threshold logic model and the Hebbian Learning model. In general, the perceptron neuron is takes in a vector of input and factors in the related weights to the input before summing up the updated input to produce an output. The entire perceptron learning process is illustrated in Figure 2.6.

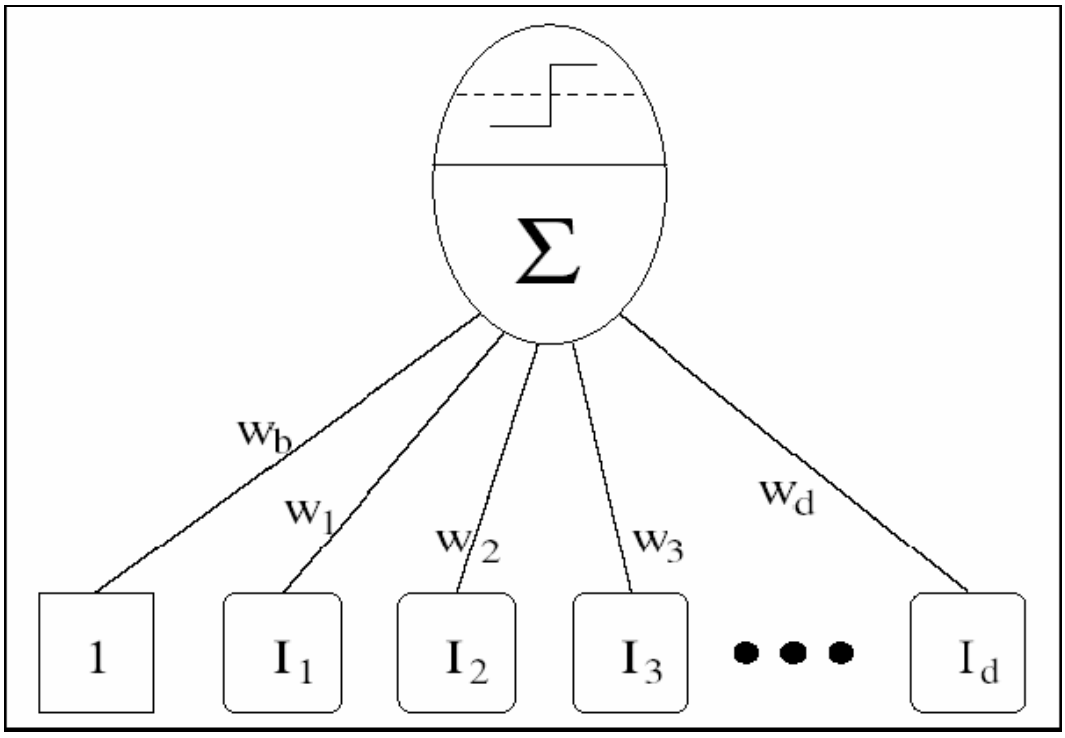


Figure 2.10 Perceptron neuron

The early successes of perceptrons led to many enthusiastic claims. However, Minsky and Papert in their seminal paper Perceptrons , 1969, had demonstrated the limitations of perceptrons and this led to almost a decade of decline in the studies of artificial neural networks (Fausett, 1994). But, even during this decade of relative quietness in the world of neural networks research, we have seen some of the most significant and notable contributors to neural networks emerge with their work. Among these researchers are Kohonen with his Self Organizing Map (SOM), Carpenter with his Adaptive Resonance Theory (ART) and Anderson s Brain-state-ina- box (BSB) (Fausett, 1994).

2.1.9 Summary

Arguably, speech recognition has matured with the introduction of each new phase of technological advancement. From template models that utilize dynamic programming to sophisticated neural network models, from simple monosyllable

phonemes to continuous speech models, speech recognition research is rich and diverse. However, the nature of research in speech recognition is dynamic. As there are still hope to further improve on this field, there will always be new technology and new innovations to accommodate the breakthroughs that happen in speech recognition.

Upon scrutiny, speech recognition research has largely been in the domain of engineering departments of universities the world over. The desire to model speech is often motivated by a desire to produce practical applications. Techniques motivated by knowledge of human processes have therefore been less important than techniques that can be automatically developed or tuned, and broad coverage of a representative sample is more important than coverage of any particular phenomenon. The focus on the exact science of speech recognition has taken away a fundamental aspect of this field; that is, speech is still very much in the domain of human beings. Therefore, it should towards the human psyche that we should be looking into to unlock the secrets of speech recognition.

2.2 Introduction Of Autopoiesis

Autopoiesis is the study into the characteristics of the living. It was developed as an explanation into the workings of life from an organizational perspective of cognition. The theory was introduced by two Chilean biologist; i.e. Humberto R. Maturana and Francisco J. Varela. It was meant to designate the organization of a minimal living system (Varela, 1992). This theory attempts to put answers, beyond the point of philosophical discussions, to the questions of what it is to be alive and what differentiates a living entity from that of automated machinery. However, the focus of this thesis would be upon the cognitive aspect of the theory and the self-organizational prospects that the theory brings to the development of a speech recognition system. After all, Maturana and Varela (1980) did mention that cognition is a biological phenomenon and can only be understood as such; any epistemological insight into the domain of knowledge requires this understanding. Therefore, it is in cognition that the answers to a speech recognition system will be found. Autopoietic systems in general are defined in terms of their organization (Boden, 2000). The main concerns of autopoiesis is on organization and structure because the organization of a

machine (or system) does not specify the properties of the components which realize the machine as a concrete system, it only specifies the relations which these must generate to constitute the machine or system as a unity (Maturana & Varela, 1980). Therefore, the organization of a system is independent of the properties of its components though a given system can be realized by many different structures. However, for a system to have a concrete constitution in any given space, actual components must be defined in that space and have the properties which allow them to generate the relations that had define it, hence the importance of structure.

2.2.1 Complimentary of Structure and Organization

Notions of structure and organization are akin to ideas of the fundamentals of matter. The structure of matter consist of atomic particles, that are, themselves, a constitution of sub-atomic particles. On the other hand, organization refers to the invisible laws of nature that binds all these structure together to form the matter. However in biology, organization refers to the group identity of living entities; i.e. the pattern of relationships among their components which bear resemblance in many of their significant characteristics. On the other hand, structure refers to the particulars within a given entity, i.e. the physical properties of the components and the roles of the components play in the making up of the whole entity.

Therefore, the complimentary between organization and structure is reflected upon the cooperation of both concepts in maintaining a balanced and smooth relationship known as life. This is due to the fact that organization can only exist in terms of relationships amongst structures and structure exists only in filling the roles in those relationships. The simplicity of this relationship of organization and structure highlights the underlying complexity of what would be termed alive, for there must exist an inexplicable mechanism that regulates the workings of this relationship. It is this mechanism that would provide the key to achieving a self-organizing system that we could term alive.

2.2.2 Operational Closure

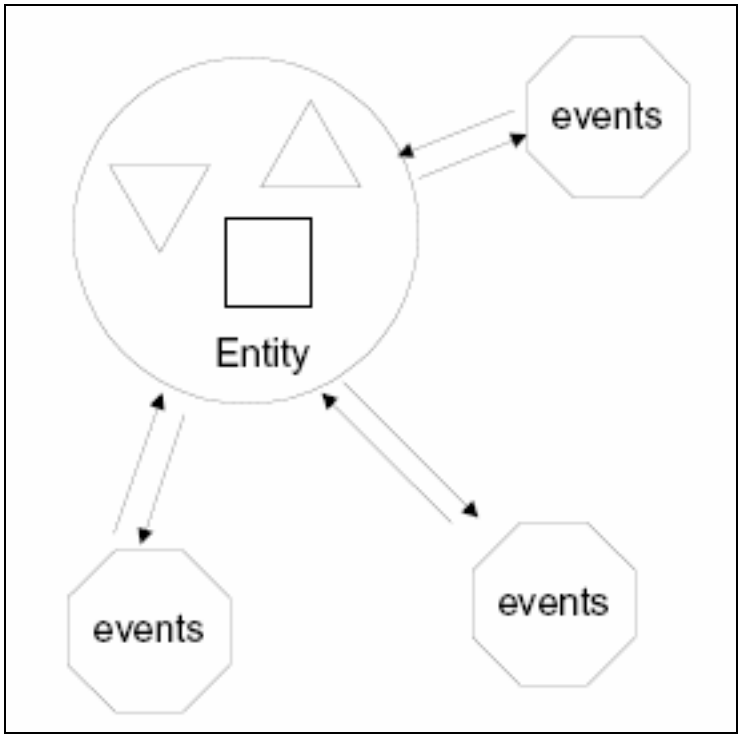


Figure 2.11 Interactions between entity and environment

Closure is a systems notion that refers to the containment of a system's operation within a system s boundary. For example, living entities are open to the environment in terms of energy and material but are operationally close because the physical boundaries with which a living entity is confined to has no correlation between the internal structure of the entity and that of the external view of its environment. However, this phenomenon is a physical representation of the relationship of a living entity with its environment. It doesn t portray the whole picture of what it means to be alive.

2.2.3 Structural Coupling

Structural determination describes the actual course if change in a systemic entity is guided by the entity s own structure rather than influences of the entity's environment. Thus, given this principle, interaction among systems can be summedup as a history of recurrent

interactions leading to the structural congruence between two systems (Maturana & Varela, 1992). Structural coupling is the label for ongoing engagements between systems that positively return a change in structure of each entity. Therefore, structural coupling describes ongoing mutual co-adaptation without hinting of a transfer of some ephemeral force or information across the boundaries that separate the engaged systems.

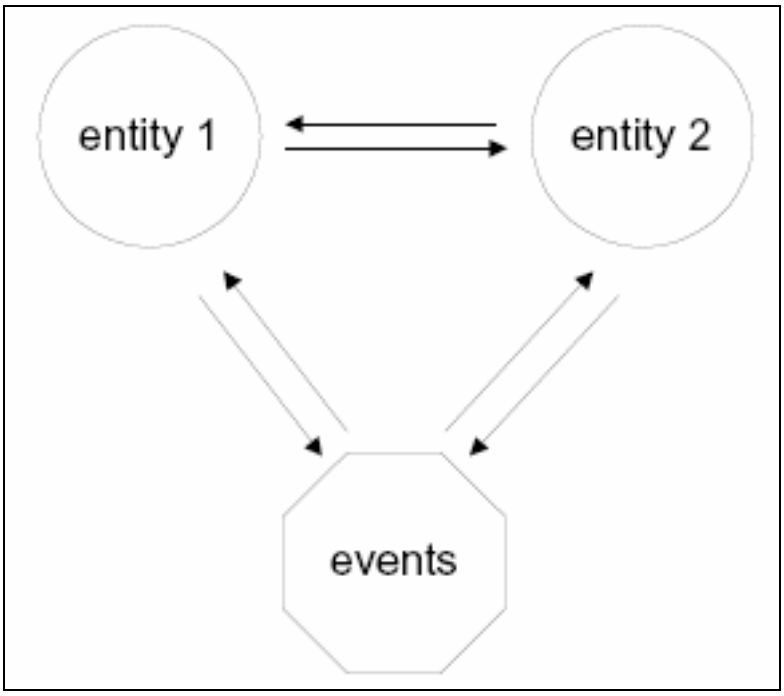


Figure 2.12 Interaction between Entity with other Entities and Environment

2.2.4 Observation and Cognition

From the autopoietic point of view, we, as living entities, are too the observers of our lives. We make these observations in a communicative process and try to explain these observations in a conversational methodology. The reality derived from our observations could be divided into two distinct forms; i.e. an objective reality and a personal reality. Objective reality provides us with a referral point from which we see ourselves from an external point of view; much like as a third person view of ourselves. Personal reality however, refers to the assessments made of the derived objective reality perceived by the observer; i.e. ourselves. Therefore, achieving cognitive status for a living entity is but

accumulating numerous samples of observations that are commonly known as experience. To be endowed with the cognition title, the living entity must benefit from cognition by being capable of change to its awareness in the wake of an ever evolving environment and this can only arrive from a properly constructed internal structure that could facilitate the storage of all foresaid experiences. Thus, cognition is contingent on embodiment because it is truly a consequence of the entity s specific structure.

2.2.5 Calculus of Indication

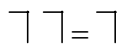
The dynamism in autopoiesis transcends formalism of conventional mathematics, thus rendering the theory with the lack of a formal description of the theory s qualitative phenomena in a quantitative manner without recourse to numeric calculi (Vernon & Furlong, 1992). Therefore, Francisco Varela has adopted and adapted an alternative form of calculus to express the intricate logics of autopoiesis. This formal theory is known as Calculus of Indication and was introduced by Spencer-Brown (1969) in The Laws of Form . In fact, Spencer-Brown had informally presented three distinct logical systems and Varela had adopted the third system in a three-valued logic representation (Turney, 1986). Spencer-Brown's calculus is non-numerical and it is based on the most elementary of all conceivable operations: that of making a distinction of indicating something (Vernon & Furlong, 1992). To truly appreciate this form of logics, Vernon and Furlong has given the example of indicating a chair in a universe or domain in which a chair can be distinguished. The indication of not a chair on the other hand, is not the inverse of the indication of a chair. In fact, the inverse is to indicate the particular indication of the chair, thus making no indication of a chair at all. Therefore according to Vernon and Furlong (1992), compared to other dyadic systems of conventional two-valued logic, this theory is single-valued and offers an excellent alternative to describing the logics needed to carry the abstract concepts of autonomy and self-organization in autopoiesis into the realm of formal mathematical theory and practical applications in the field of engineering and science.

2.2.5.1 Axioms in Calculus of Indications

For a given space (universe) and its distinctions (objects), the parts shaped by the distinctions are the states of the distinctions. The spaces and their states, together, are the forms of the distinctions. States which are distinguished by the distinctions are signified by a mark of distinction.This, then, is the marked state and also referred to as a cross. The state not marked with a mark is called the unmarked state and is signified with a space such as (Vernon & Furlong, 1992). There are two axioms in the calculus; i.e.

2.2.5.1.1 Axiom 1. The law of calling (condensation)

states that the value of a call made again is the value of the call :



This is interpreted as follows: If two crosses are contained in the same space, their value is that of distinguishing twice and, thus, their value is that of the marked state

2.2.5.1.2 Axiom 2. The law of crossing (cancellation)

States that the value of a crossing made again is not the value of the crossing:

This is interpreted as follows: If a cross crosses another, we undo the distinction; hence the value is that of the unmarked state.

2.2.6 Beyond the Axioms

Due to the fact that autopoietic systems exhibit self-referal qualities, it is not enough to merely utilize Spencer-Brown s calculus per se. An indicational expression of the reentry of self must be incorporated into the notations used in the calculus. To facilitate this, Varela adopted a convention which shows the point at which the form that re-enters its own indicational space by extending the cross which contains the whole expression.

For example, f = f can be rewritten as f =

Thus, a self-referential system would be unstable and perpetually alternating its states. Further reference should be made to Vernon and Furlong (1992) for a detailed discussion on the subject matter. Table 2.1 tries to depict the intricate logics derivable from this calculus based on the axioms above.

Table 2.1 Index of the Axioms from a Boolean perspective

i	Boolean $_i$	Formal $_i$
1	T	⌐
2	A or B	A B
3	B imp A	B̄ A
4	A	A
5	A imp B	Ā B
6	B	B
7	A eqv B	Ā B̄ A B
8	A and B	Ā B̄
9	not (A and B)	Ā B̄
10	A xor B	Ā B B̄ A
11	not (B)	B̄
12	not (A imp B)	Ā B
13	not (A)	Ā
14	not (B imp A)	B̄ A
15	not (A or B)	Ā B̄
16	F	

2.2.7 Autopoiesis and Learning

The importance of learning stems from the fact that speech recognition, like any pattern recognition problem, relies on the ability to match given input with the correct output. Such matching is usually based on a best-to-fit situation because the input of the system seldom correspond precisely with the desired output for if there exist one input to one output matching relationship, the matching would then be just a database querying problem. Therefore, the following section will discuss the issue of learning that is involved in the speech recognition model proposed.

2.2.7.1 The Problem of Learning

Let s assume that is the set of all the phonemes in a particular language. Then assume that **A** is a word

$$A = \left\{ a_i : a_i \in [\varepsilon]^k , i, k \in \aleph^+ \right\} \quad \textbf{(3.1)}$$

Suppose that *S* is the global set of speech that will be recognized, therefore

$$S = \{A_i : i \in \aleph^+\} \quad \textbf{(3.2)}$$

If an input speech, $X = [\varepsilon]^k , k \in \aleph^+$

$$X = \{x_i : x_i \in [\varepsilon]^k , i, k \in \aleph^+\} \quad \textbf{(3.3)}$$

Therefore, a learning function $\psi(x)$ would plot input x_i to ouput y_i for all valid words

$$\psi(x_i) = \{y_i, \forall y \in S\}, i \in \aleph^+ \quad \textbf{(3.4)}$$

The validation of this learning would come in the form of

$$\psi'(y_i) = \{a_i, \ni a_i \in S\}, i \in \aleph^+ \quad \textbf{(3.5)}$$

when the y_i in the database of the system could be used to reproduce the correct learned speech inputs.

In order for the function $\psi(x)$ to be successful in plotting the proper input to the proper output, some assumptions are made. In the first place, $\psi(x)$ accepts all speech inputs. The x_i should be filtered through the lexical and syntactic models of the language of x_i. However, the scoping of this research allows for the fact that all x_i are valid or perhaps an indicator function $\phi(x_i)$ could be incorporated into $\psi(x)$ to reject all invalid inputs before the plotting of the output commences. The second assumption is that the length of the inputs and outputs are not taken into consideration. Though the employment of the DTW algorithm and the coding of speech into frames of fixed length do help in maintaining a standard length to

the inputs, however there is a need to consider the inputs as discrete words and not continuous signals; i.e two or more words cannot be combined as one input lest the learning of this inputs considers the two words as one whole word that is definitely invalid syntactically.

Theoretically, *S* represents the lexical model of the target language for the speech recognition system. For a small, specific vocabulary of the recognition system, a limit is set on the accepted number of inputs to be trained or rather a limit is set on the accepted number of inputs that are considered valid. However, if a large, general purpose vocabulary of the recognition system is needed, then the accepted inputs maybe unlimited in number or perhaps the dictionary of words that are accepted would be a complex database of its own. Therefore, the third assumption is that the performance of the recognition system is gauged by the size of *S*.

2.2.7.2 A Statistical Learning Theory

The basic components in learning theory are a set *X*, a σ-algebra of subsets of *X*, a family of probability measures on (X,) and a subset $C \subseteq S$, called the *concept class*, or else a family of measurable functions mapping *X* into [0,1], called the *function class* (Vidyasagar, 1997). This specification of the learning theory by Vidyasagar (1997) clearly distinguishes the difference in learning a concept and learning a function. However, the underlying concepts of both learning theories are the same. Therefore, the discussion of learning would be centered on the learning of concepts rather than functions.

In concept learning, there is a fixed but unknown concept $T \subseteq C$, called the target concept. The objective is to learn the target concept on the basis of observation, consisting of independent and identically distributed (i.i.d.) samples $x_1, , x_m \in X$ drawn in accordance with a fixed probability measure P . In contrast with the target concept *T*, the probability measure *P* maybe either known or unknown. For each sample x_i, an indicator function will determine whether or not x_i *T*. In other words, the function returns the value of $I_T(x_i)$, where

$I_T(\,)$ is the indicator function of T. Thus, after m samples are drawn, Vidyasagar(1997) states that the information produced would consist of labeled multisamples

$$[(x_1, I_T(x_1)), ,(x_m, I_T(x_m))]\ [X \text{ x } \{0,1\}]_m \qquad \textbf{(3.6)}$$

The objective of Equation 3.6 above is merely to construct a suitable approximation to the unknown target concept T on the basis of the labeled multisamples using an approximation algorithm. For simplification of discussion, the algorithm is just an indexed family of maps

$$\{A_m\}_m \text{ where } A_m : [X \text{ x } \{0,1\}]_m \qquad \textbf{(3.7)}$$

If the probability measure P is known, then A_m may depend on P; otherwise A_m must be independent of P, though it can depend on P. Similarly A_m must be independent of T, but it can depend on (Vidyasagar, 1997). Suppose m i.i.d. samples have been drawn and defined, if there exist a function $H_m(T; \mathbf{x})$ where

$$H_m(T; \mathbf{x}) := A_m[(x_1, I_T(x_1)), ,(x_m, I_T(x_m))] \qquad \textbf{(3.8)}$$

Thus $H_m(T; \mathbf{x})$ is the output of the algorithm when the target concept is T and the multisample is $\mathbf{x} = [x_1\ x_m]_t$. It is customary to refer to $H_m(T; \mathbf{x})$ as the hypothesis generated by the algorithm (Vidyasagar, 1997). From Equation 4.7 and 4.8, it is deducible that as m and that the principle difference between T and H_m 0, the target concept is said to be learned. The prove of this is further elaborated by Vidyasagar (1997).

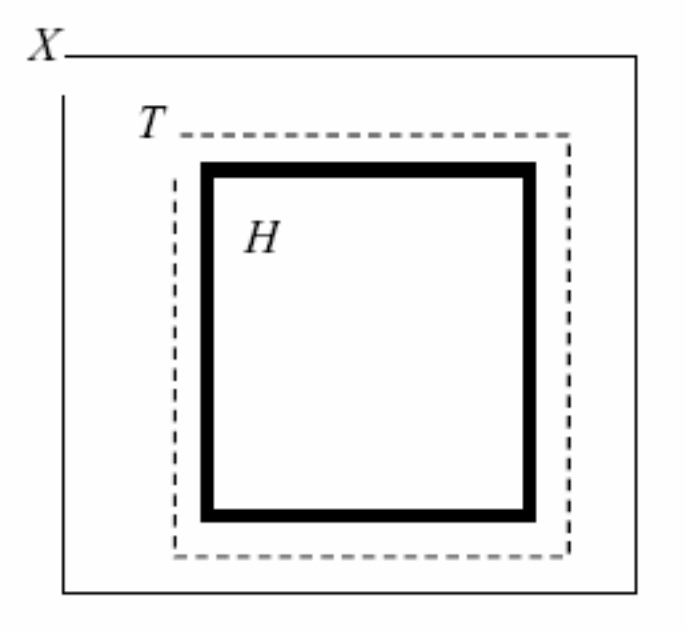


Figure 2.13 Concept Learning

2.2.8 Autopoietic Regulation of Learning

Basically, the proposed speech recognition engine has a dual role to play. It is a learning agent of speech inputs and it is also a practitioner that produces outputs from the inputs. This is due to the fact that the engine must be able to robustly learn as new inputs come along in the process of recognition. Needless to say, the system learns as it recognizes. Though this may seem awkward and contradicting conventional intelligent systems, but it does highlight the point of having selfregulating and self-correcting qualities in the system. However, this approach underlines two main issues which await an autopoietic solution.

The first being the question of the practicality of such approach; after all, the current method of train first, test later has proven to be quite efficient. The second question involves the complexity of the system because the regulation mechanism might add too much overhead to an already difficult problem of recognition. Thus, the introduction of autopoiesis is meant to address these two issues as well as attempt to prove that it is feasible to integrate a self-organizing criterion into speech recognition. Besides, autonomy and self-organization

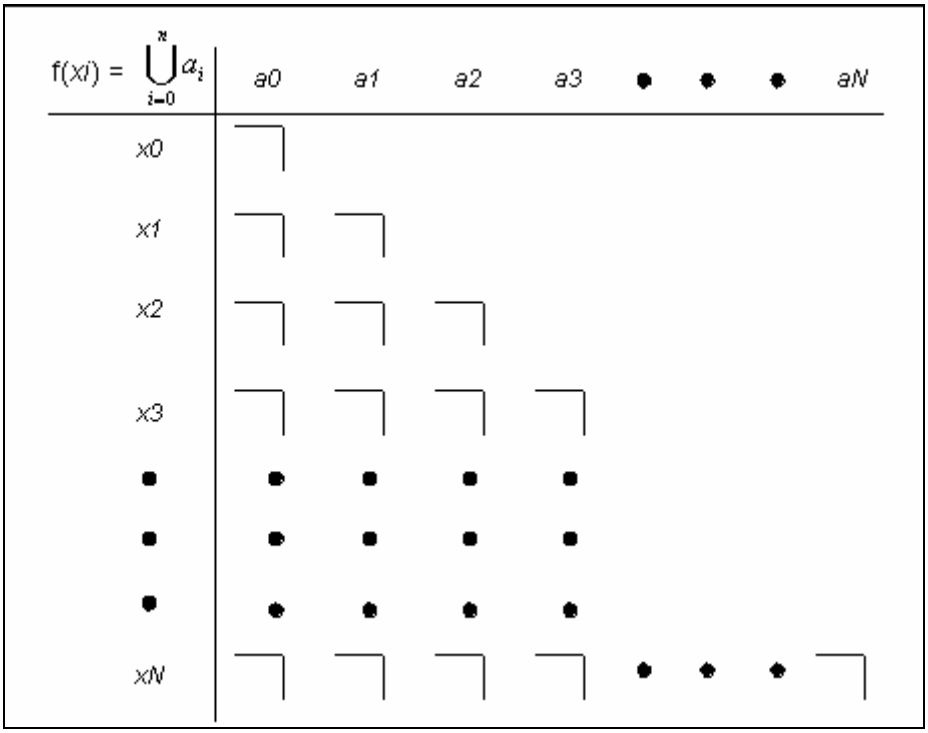
may carry speech recognition to a higher level of achievement. Perhaps such endeavor may lead to better machine vision, image recognition and other fields of machine learning.

2.2.9 Dualism of the System

Imagine the speech recognition system as having two states; i.e. the learner and the practitioner. The current state of system is determined solely on current input. The reason for this is simply because of the first assumption on Section 3.8.1 that states that all input of the system are acceptable. This draws parallels to the human learning process of accepting all given inputs perceivable to their five senses as stimuli of learning. Therefore, all input will either be training data or testing data. The alternating role of the input makes for a need to regulate the system in accordance to perceivable state of the input.

For example, if the x_i in Equation 3.4 is an input that has been learned by the system prior to this particular input session, then x_i will be considered a test data and the system will be geared to a practitioner state, where the matching y_i can be utilized to derive the a_i or correct word of the speech input. If the opposite is true of the input, x_i, and this particular input session introduces the x_i as training data, then a new y_i will be generated and a corresponding a_i will be determined. However, every a_i may have a set of x_i in slight degrees of variation. Therefore, it is undeniable that the system must treat the input as training and testing data at a go; hence a regulatory mechanism is needed. Consider the following problem, if f(x) = y and f(y) = a then f(x) a or in other words, for every x, there exist a unique a, however, every a may have more than one x. Thus, it is assumed that every x may be a current learned a_i or a new a_{i+1}. The general concept of this $x \rightarrow a$ can be defined as

Table 2.2 The truth table through the axioms of Calculus of Indications



From the truth table in Table 2.2, the system treats the x in much of a recurrent or self-referential manner due to the fact that the input of the system will always be considered a training data till being determined otherwise. This only stresses on the point of learning is an ongoing process of the system; even during the testing phase. In addition, this approach to learning requires constant feedback; i.e. the inputs generate outputs that will tune the system or rather that will adjust the state of the system for future inputs.

For example, the learning process, l, receives x. Based on the recurrence of x,

$$l = \overline{x|} \qquad \textbf{(3.9)}$$

2.2.10 Summary

Organization and structure are two concepts that blend in well with the needs of developing an architecture for speech recognition. Autopoiesis is somewhat a paradox at times. At one hand, it is a way of defining the characteristics of life and what it means to be alive. On the other hand, autopoiesis was formulated to bring these characteristics into inanimate objects such as your home PC or your industry standard robot. Does this means that autopoiesis might be the key to future intelligent life forms or is just another theory, sitting on the shelf, waiting for the day another theory comes along and overtakes it in the quest for the definition of life. In the context of speech recognition, autopoiesis promises to be a novel way of approaching the pattern recognition aspect of the problem. With autopoiesis being a theory to rigorously defining intelligence and life, we notice that the inherent need to create more capable recognition architecture can be seriously met by autopoiesis. Thus, it is worth the thought and effort put into developing and autopoietic speech recognition system, even though we have to build this system from scratch.

CHAPTER III

METHODOLOGY

Generally, the research undertaken is aimed at developing a speech recognition system and formulating a mathematical model of the system. Hence, the task of developing a speech recognition system must be approached in two different aspects. The first is research on speech recognition and the possibilities of integrating autopoiesis into the system and the second is the development of an autopoietic speech recognition engine through an object oriented approach. This chapter is on the process of developing the speech recognition system; particularly focusing on feature extraction and word matching. The speech database used will also be discussed here.

3.1 Introduction

Based on the generic components in a Speech Recognition Model, as depicted in Figure 2.1, the overall design of the proposed speech recognition engine focuses on three main modules. These modules are the feature extraction module from the signal analysis phase, the phoneme recognition module of the acoustic modeling phase and the output matching process of the word matching phase. Section 5.2 and Section 5.3 is to document

the overall picture of the development process of the speech recognition system. The subsequent sections in this chapter are devoted to the documentation of the development process of the signal analysis and the word matching phases of the system. The acoustic modeling phase will be discussed in Chapter 6 because it is the main attention of the research carried out.

3.2 Research Methodology

The methodology employed in completing this project is, on the whole, a linear sequential model. Every task required for the success of the research in this project will be executed consecutively in the form of phases that are ordered in accordance to the precedence of completion of the tasks required. Figure 3.1 depicts the phases that are involved in this research of the project. Due to the nature of the research at hand and the lack of similar autopoietic systems in the field, the methodology employed to make the development of an autopoietic speech recognition system a success is to analyze-build-validate the system, with many repetitions of the development process in order to improve the system incrementally.

Basically, there are eight phases to the whole research methodology. The first three phases are research aspect of the project where literature reviews and preliminary analyses are carried out to produce the overall system view of the speech recognition model. The next two phases are the development aspect of the project where a working prototype will be built. The last three phases are the most important of all because these phases are aimed at justifying the importance of the whole project proposed.

3.2.1 Requirements Analysis

This phase handles the problems encountered in the project. This is achieved through the evaluation of all project objectives and scopes as well as the requirements and issues pertaining to the proposed title. The main aim of this phase is to ensure that the

project is manageable and that there is ample time for completion. Apart from that, this phase is meant to give perspective to the issues at hand and give focus to the research carried out. The execution of this phase is mainly based on review of literature on the subjects of digital signal processing, speech technology, pattern recognition, autopoiesis, cybernetics, calculus of indications and other related disciplines in speech recognition and self-organization. The main sources of literature consist of books, journals, online articles and conference proceedings.

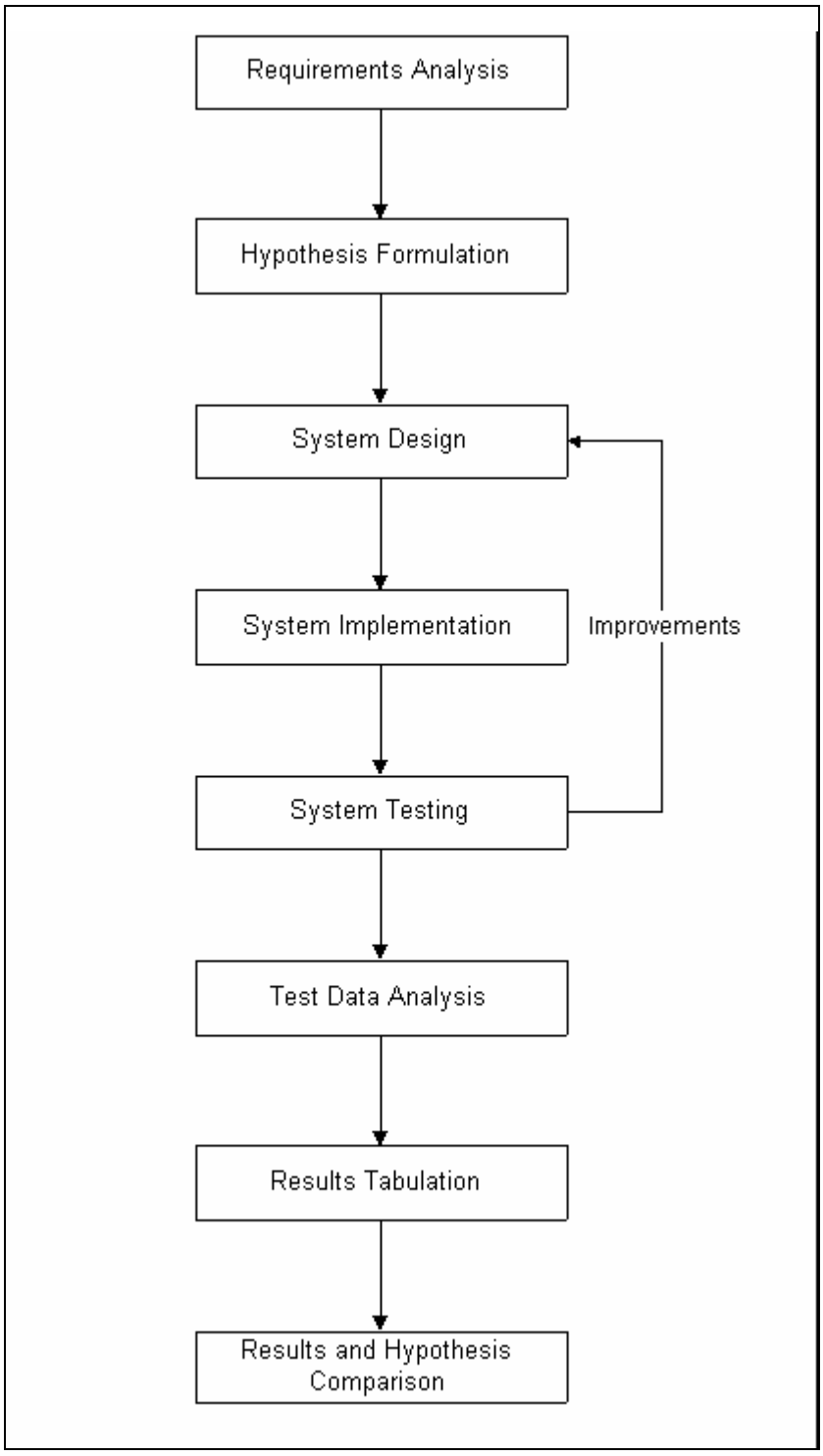


Figure 3.1 Research and Development phases

3.2.2 Hypothesis Formulation

A hypothesis is generated to give meaning and focus to the project undertaken. The method of formulating the hypothesis is mainly about converting the problems and issues identified into questions that need solutions and later, summarizing and analyzing these said questions to produce better and more stringent research guidelines. Further reference to the initial hypothesis of this project is mentioned in Chapter 6. Therefore, the hypothesis is meant to be the guiding light of the research as well as the development carried out in this project.

3.2.3 System Design

The need to validate the research undertaken in this project has lead to the importance of developing a prototype of the system proposed. Due to the lack of an existing model for the system, an evolutionary approach to the development of the foresaid prototype of the system is employed so as to guarantee the end result of the research is certifiably correct and further expansion on the project is foreseeable. Please refer to Section 5.3 for additional discussions of this.

3.2.4 System Implementation

In this phase, the proposed system is implemented through a computer application. Therefore, the process of programming a computer is involved in this phase. The choice of programming language is C++ due to the fact that each initial prototype is developed in an object oriented programming paradigm and the deliverable system is developed with 4GL tools used in Rapid Application Development (RAD) such as Microsoft s Visual C++. Besides, Visual C++ is capable of offering satisfactory graphical user interfacing to the speech engine developed in C++. Such development method is

appropriate in light of the project requiring generated output that meets the design specifications of the system.

3.2.5 System Testing

System testing is the phase in which the implemented prototype from the system implementation phase is subjected to rigorous testing to generate the needed data for analysis. These data will be collected through a systematic sampling method in which the output generated by the prototype from reasonably acceptable input will be considered as data of the testing phase. The data collected will be based on correctness of output, efficiency of output, overall performance of prototype and variability of output. However, due to the nature of this project, all data should be treated as a best-to-fit the problem at hand because seldom does an intelligent system can deal with precise solutions. This phase promises to be the main gauge of the results of the project for when a prototype is deemed unfit to solve the problems
identified by the research into speech recognition and autopoiesis, a new prototype is designed and implemented.

3.2.6 Test Data Analysis

Analysis of the data collected is needed for verification and validation of the developed prototype of the system. Among the many analysis techniques, a statistical analysis will be carried out to determine that the error of the generated data is within given tolerant values. Apart from that, a summary of the data will be produced based on required parameters of the project. Therefore, this phase determines the level of success of the project and ensures that the development process of the project has been carried out properly.

3.2.7 Results Tabulation

The results from the previous phase are now tabulated to produce a summary that is acceptable and simple to study. Such tabulation will be derived from the analysis of the data generated and these data can be differentiated from that of the data hypothesized from the model of the speech recognition system produced through a mathematical algorithm. Therefore, the summary will be presented in a form capable of manipulation that enables a comparison study with the hypothesis of the project to be carried out.

3.2.8 Results and Hypothesis Comparison

In this phase, the summary from the test data and the hypothesis are compared to determine the validation of the project. The importance of such comparison is reflected upon the fact that there is a genuine need to prove that the prototype developed in this project is of use and this project truly contributes to the field of speech recognition and autopoiesis. With a proper comparison study, the process of documenting this project can be carried out accurately. The end result of this study can then be a useful tool to authenticating some of the main objectives of this project. Therefore, this phase must be conducted in the most proper of manner and there must be a certain sense of control over the results achieved.

3.3 Software Development Methodology

The speech recognition engine proposed in this project will be developed in stages more akin to an evolutionary process. Figure 3.2 depicts this process in a spiral model. Basically, the development process involves four major steps; i.e. an objectoriented analysis stage, an object-oriented design stage, an object-oriented programming stage, and an object-oriented testing stage. These four stages form what is generally known as Object-Oriented Software Engineering (Pressman, 1997). The choice of object-oriented

software engineering is encouraged by the fact that the world and, subsequently, all that can be perceived exists in the form of objects.

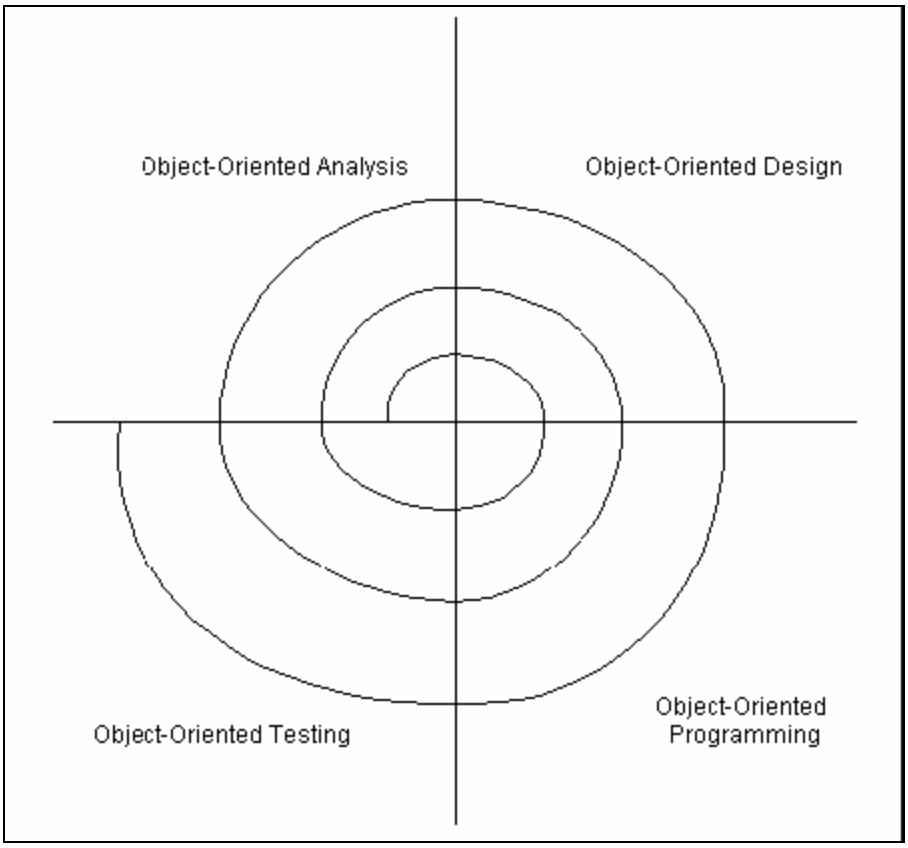


Figure 3.2 Spiral Model

Therefore, it is no surprise that an object-oriented view would be considered for the development of the proposed system. Among the more substantial advantages of object-oriented technology is the fact that object-oriented software is easier to maintain because its structure is inherently decoupled thus leading to fewer side effects when changes have to be made. In addition to that, object-oriented systems are easier to adapt and scale, especially when an evolutionary process of development is employed. Besides, object technologies promote reuse of software components and this reuse of components leads to a better controlled development environment as well as a cleaner, more efficient way to implement subsequent prototypes of the same proposed application model.

3.3.1 Object-Oriented Analysis

The utilization of object-oriented analysis (OOA) in this project is largely based on a combination of the Booch and Rambaugh method where classes and objects of the system would be identified along with all related semantics, relationships and refinements of these classes (Pressman, 1997). An object model will aid in the process of identifying such classes and objects. Following the identification of all related classes and objects, the system flow and function will be describe through a dynamic model in the form of a state diagram and a functional model in the form of a data flow diagram.

In this step of the development process of the speech recognition system, concerns of precision, concision, understandability, and correctness of representation of the problem is important. After all, the aim of analysis is to prevent deviations of the scope and objectives of the project from happening in the course of developing the system. Therefore, practices such as producing static and dynamic views of the system and formulating accurate problem statements are taken to avoid any unforeseen mishaps in the analysis of the proposed system. Thus, analysis of the system is an important step in guiding the project to success.

3.3.2 Object-Oriented Design

Object-oriented design (OOD) of this project aims to translate all the object models generated in the analysis step into one system design that is understandable, decomposable, robust and implement-able. The main interest of the design step is basically transforming an analyzed problem into an implemented system. Therefore, OOD is somewhat the mechanism that launches the ideas put forth in the analysis stage and helps foster a better, more stringent development process of the system. The process of OOD is basically simple. It involves procedures such as partitioning the analysis model into subsystems, identifying concurrency that is dictated by the problem, allocating subsystems to processors and tasks, choosing a

basic strategy for implementing data management and identifying global resources and the control mechanisms required to access foresaid resources (Pressman, 1997). Thus the resultant end of this step is a description of the workings of the proposed system. Such description will be presented in the form of a program design language (PDL), an object relationship model, and a human-computer interface (HCI) model.

3.3.3 Object-Oriented Programming

In this stage of development, the choice of programming language is C++ for reasons mentioned in Section 5.2.4. However, there is a need to stress on the paradigm that is used in object-oriented programming (OOP) so as to clarify the justifications of utilizing C++. As mentioned before, C++ and subsequently Visual C++, offer satisfactory graphical user interface features into the proposed system. This is inline with the need for developing the system for the Windows ME platform that has memory resources management that is satisfactory. The implementation of OOP would depend largely on the design that is produced. With a complex design, the implementation of the system would take a longer and more painstaking route of development. However, if the design is simple, the implementation would be less of a concern then other steps of development of the system. Complex or simple, these two simple approaches to development will be used. First of all, all classes and objects designed will be coded in to modules separately. Such an attempt is to facilitate easier testing as mentioned in Section 5.2.4.

The second approach is the integration approach; where the units of modules coded would be incrementally added on to the system to form the needed speech recognition engine. This is to ensure that the components that form the system are loosely coupled and easily interchangeable as well as reusable. Therefore, OOP stands as the beneficiary of good OOA and OOD practices and thus there is the need to ensure every step of the development process is properly conducted.

3.3.4 Object-Oriented Testing

Testing of the project, in an object-oriented sense, is basically about verification and validation. The process of testing in an object-oriented environment can be a rather tedious affair. The approach to testing in this project can be divided into three aspects. The first is unit testing. Unit testing is about testing the classes or components of the system individually. This testing can be done as each of the components is coded. Therefore unit testing is a somewhat verification process of the individual classes of the design proposed for the system.

The second approach is integration testing. This approach is the method to ensuring that the component that come together to form the system are working in order and that the integration process of the components is correctly executed. Therefore, the stress of integration testing is on the collaboration of the classes as well as the implementation of the logic of the system. So, this testing approach is best done when two or more classes are brought together during coding.

The third approach is the validation testing. Validation testing focuses on the user visible actions and user recognizable outputs of the system (Pressman, 1997). In this project, the method employed to validate the system is the black-box testing method. This method is about utilizing test cases and generating test outputs. These outputs will generally be converted to test data and be analyzed further when the need arises. This approach is important to the review of the system developed as the data generated would evidently be the deciding factor on the success of the project and gauge the requirements of following development efforts.

Therefore, testing in the object-oriented context is rather similar to conventional testing methodologies. The importance of testing should not be underestimated; for if testing of the system was not executed properly, the quality of development process of the system and the resultant product that is created would be of no guarantee in

excellence. In fact, without testing, the system might never be reassuringly correct. Thus, a proper test plan is of need and the formulation of this plan should be in accordance to certain software development metrics. Else, an acceptable system might never be implemented satisfyingly.

3.4 Signal Analysis

Figure 3.3 and Figure 3.4 is the preliminary analysis of the signal analysis component of the speech recognition system. On the whole, this component addresses the problem of formatting and representing the input speech for learning. Figure 3.3a describes the signal analysis component as made up of 3 main objects; namely the signal analysis object, the speech input and the speech frame. The relationship here is that the signal analysis object takes the speech input and converts that into speech frames. Figure 3.3b illustrates the states that the system goes through to generate a speech frame.

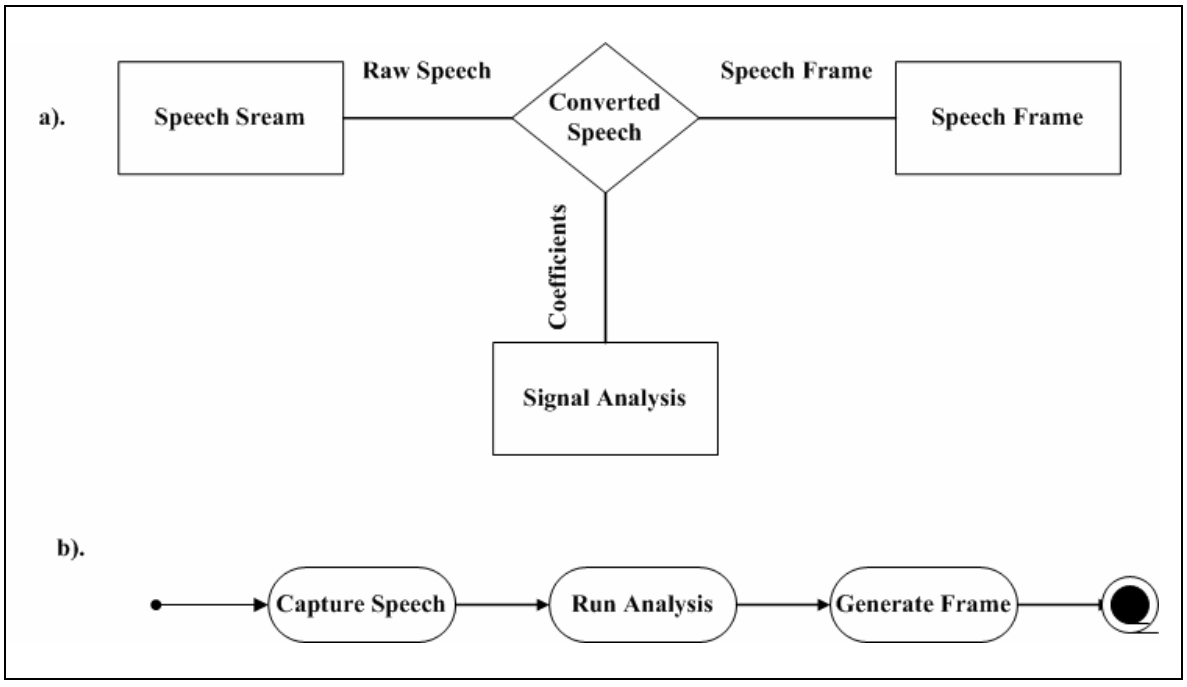


Figure 3.3 a.) Object Model b.) State Diagram

Figure 3.4 is the data flow diagram of the signal analysis component. It depicts the various forms of transformation that happens to the input speech as the raw speech is fed into the system to produce the speech frames needed.

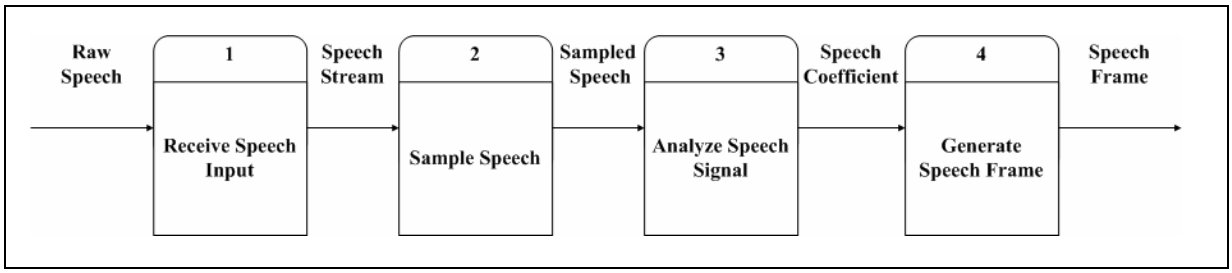


Figure 3.4 Data Flow Diagram of the Signal Analysis Mechanism

3.5 Word Matching

Figure 3.5 and Figure 3.6 is the preliminary analysis of the word matching component of the speech recognition system. On the whole, this component addresses the problem of matching the recognized speech to the words store in the database, as well as producing an output of the system. Figure 3.5a describes the word matching component as made up of 3 main objects; namely the Time Warping Function, the speech signal and the word. The relationship here is that the time warping function attempts to match the speech signal to the word stored in the database. Figure 3.5b illustrates the states that the system goes through to generate an output.

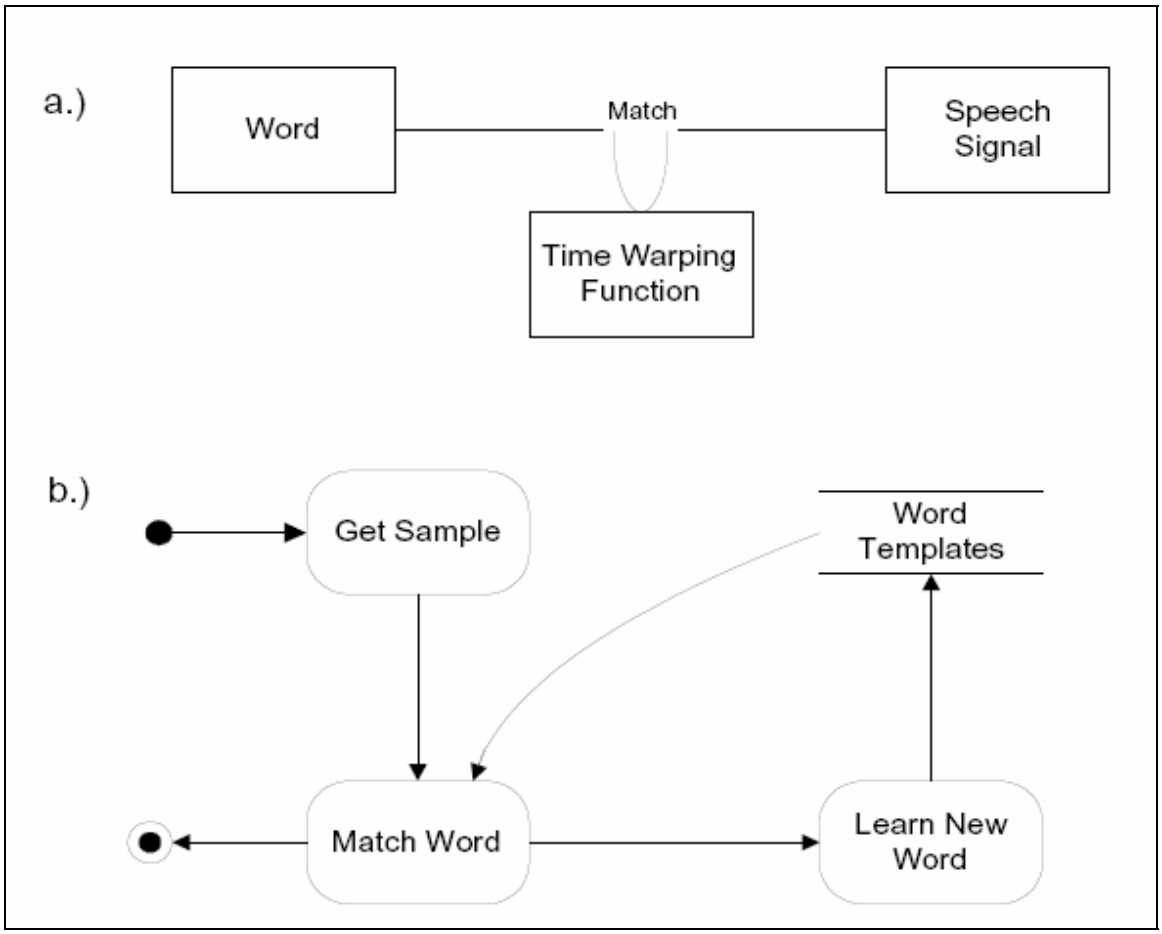


Figure 3.5 a.) Object Model b.) State Diagram

Figure 3.6 is the data flow diagram of the word matching component. It depicts the various forms of transformation that happens to the input speech as the input is fed into the system to produce the output needed. As the matching here relies on the speech database, the entire matching process will work with templates stored in the speech database.

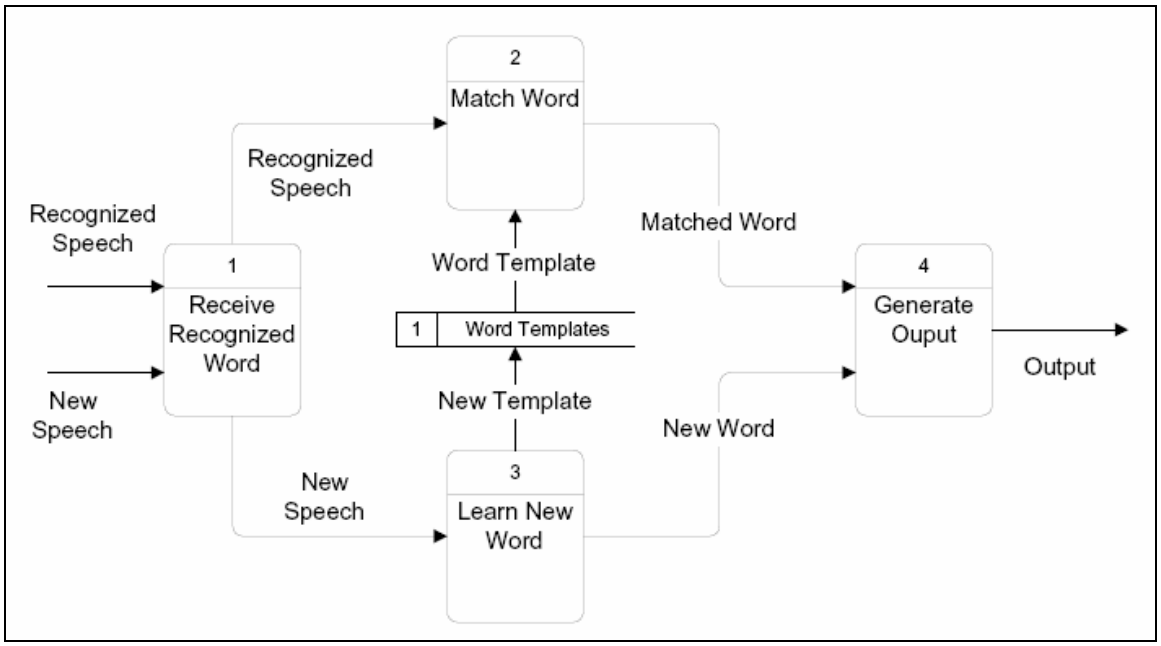


Figure 3.6 Data Flow Diagram of the Word Matching Mechanism

3.6 Output Matching

"The concept of words seems intuitively obvious to most speakers of Indo-European languages. It can be loosely defined as a lexical item, with an agreed-upon meaning in a given speech community, that has the freedom of syntactic combination allowed by its type (noun, verb, etc.)" (Huang, *et al*, 2001).

In a speech recognition system, the thought of utilizing words as the syntactic model of the language may seem feasible, especially when the scope of vocabulary is small. In fact, if the system where to just match 5 to 10 words, and the words are rather distinct, perhaps a word template is sufficient to model the whole recognition system. There would not be a need to construct elaborate learning architectures to accommodate such low dimensionality of the system. However, speech recognition systems of today are very demanding. The systems do not just handle a small vocabulary of words neither are

the systems just targeted at a small group of users, the systems of today have to cope with large vocabularies, multiple users, and in environments that are robust.

Therefore, the generating output for a speech recognition system becomes a non-polynomial (NP) hard problem. For example, if the classification of output is based on the training of inputs in the form of words, we would be faced with an ever increasing word database as new words are introduced to the system. On the hand, if the entire system is modeled on singular phones, the output matching might be a simple phone-for-text match but issues of context-dependency might never be solved. So, a balance must be struck and a basic unit of representation must be set for the building of the language database for the purpose of recognition.

Due to the fact that a phonetic recognition approach is adhered to in this research, the question of selecting the most basic units to represent salient acoustic and phonetic information becomes an important issue. Huang, *et. al.* (2001), states that at a high level, there are a number of issues to consider when designing a workable model to represent the required features. Chief among them are :

- The unit should be accurate; as different contexts bring about acoustic features, so must the chosen unit be able to handle the differences mentioned
- The unit must be trainable; it becomes pointless if the chosen unit stands out as a random variable or that to train with such units would require the processing power that is still not inherent in computers for now or ever.
- The unit must also be generalizable; the chosen unit must be able to represent a flexible, yet definite language database that may grow as the system is utilized extensively.

3.6.1 Syllable Recognition

If words are not trainable and phonemes are not context-dependent, then there has to be some other representation of the language that can be used as unit of the database for training and recognition. Among the more prominent choices are the diphones, the tri-phones and the syllable approach. Of these three approaches, the triphone method is by far the most substantial in providing for efficient performance.However, due to the fact that isolated word training with hand-labeled input speech approach is used in this research, the usage of tri-phones for training becomes rather inappropriate as the input speech has already been segmented into respective syllables. Therefore, the syllable recognition approach will be chosen in this work.

As Bahasa Melayu is highly phonetic, there are not that many phonemes that can form syllables for the training of the system. Compared to the English language, the language provides for the system the capability to match a single representation for each phone. Therefore, just like Kohonen s Self-Organizing Map, we have a phonetic typewriter in our hands. Thus, adopting the syllable approach is the most feasible of choice in this work and perhaps, in future, other approaches can be considered so as to bring improvement into the system as well as provide a basis for comparison.

3.6.2 Building the Language Database

The language database is but a collection of input that has been learned by the system. As the input speech is fed into the system with its accompanying text file, the training of the database is a simple matter of taking the input text and converting it to the relevant entry in the database. To ensure that the database is manageable in size, input that have been learned before will not add to the size of the database. Therefore, database stays manageable even after many epochs of training and that the organization of the entries is entirely independent of prior input.

3.6.3 Matching the Output

The matching of the output from the input will be a simple process. First of all, the respective input will be fed into the matching mechanism. As a list of possible words is being compiled from matched inputs, the system can still train or learn other new inputs. When a result is finally needed, a whole list of possible words is returned. As it is, this list of words might not close to the required output, sometimes, but nevertheless, it serves to highlight the perception of the input by the system and requires great attention in understanding such discrepancies. Figure 3.7 depicts the workings of the speech database.

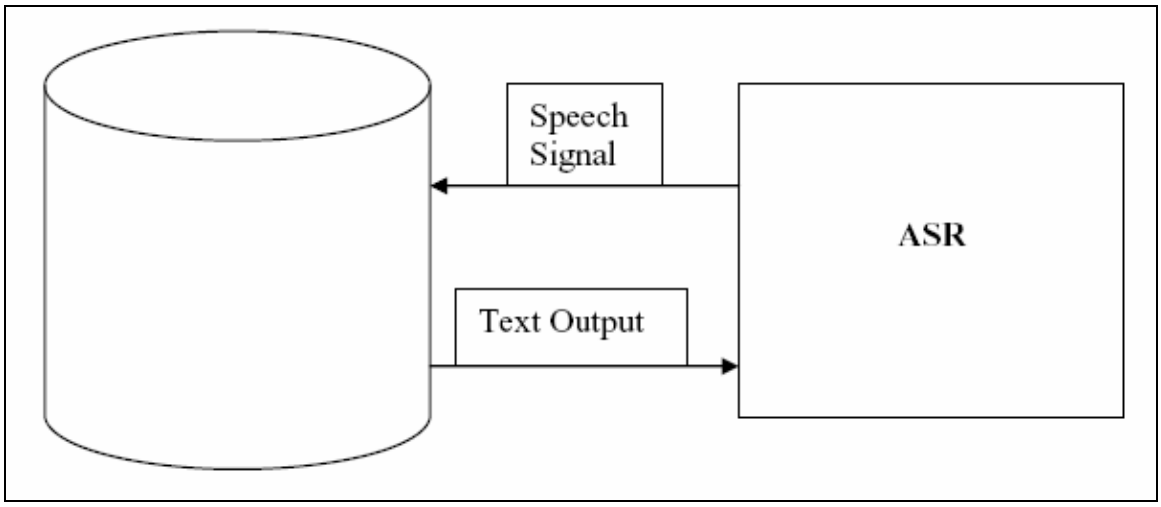


Figure 3.7 Automatic Speech Recognition and the Speech Database

3.7 The Problem of Language

Bahasa Melayu is a highly phonetic language. As such, the twenty six alphabets of the language will generate at least twenty six distinct phonemes for recognition. This estimation of the magnitude of the problem is rather grossly made and an in-depth study of the language used will be carried out before the implementation of the system

commences. However, it is safe to say that all the phonemes of the language can be represented in frames of speech of 80 100 ms in duration and that a sampling rate of 8 kHz to 12 kHz is sufficient.

As the target domain of this system is automotive control, the choice of input speech will be narrowed down to a list of commonly used terms such as digits and simple commands. It must simply be reiterated that the working prototype of the speech recognition engine has to be scoped to a certain limit. Therefore, the choice of speech inputs must reflect a broad range of possible inputs while keeping the complexity of the system within acceptable bounds and achieving the goals of speaker-independent, discrete word recognition.

3.8 Speech Database

On the whole, the system was targeted at the recognition of spoken Bahasa Melayu words at the phoneme level. The domain chosen for the system was automotive control. Therefore, the corpus used in the training and testing of the system consisted of :-

Digits:	satu (one), dua (two), tiga (three), empat (four), lima (five), enam (six), tujuh (seven), lapan (eight), sembilan (nine), sepuluh (ten)
Commands:	kiri (left), kanan (right), atas (up), bawah (down), brek (brake)

Speech samples were taken from 5 speakers, 3 male and 2 female, and were recorded in 3 separate tries. Therefore, the overall corpus had 15 x 5 x 3, or 625, number of words. However, the training and testing of the system does not fully utilize all these words as some of the words are allocated for training and some for testing. Even then, a small portion of the corpus would suffice for the purpose of validating the research carried out.

The samples were captured from a microphone and stored into wave files. The samples were then segregated based on syllables so as the range of phonemes in a single input is but perhaps 2 or 3 phones. Being hand labeled, it was easy to generate the

corresponding input text for the speech samples. These input text became the text input of the system for the learning process of the system.

3.9 Summary

The research carried out is done in a structured, phase-by-phase manner so that the entire process flow of research can be thoroughly executed. Such simplicity is needed so as the work load of the research done is manageable. The development of the speech recognition system, on the other hand, is based on object orientation. The benefits of object orientation allows for an easy-to-handle feel for the development process.

The methodology employed here makes the entire development of the speech recognition system manageable because it treats the issues in speech recognition in the form of objects or modules. *Modularity allows for the problem to be decomposed into simpler components which are easier to build, maintain and understand. In particular, object oriented approach to software engineering has shown that bundling behavior with the state it depends on simplifies both development and maintenance.* (Bryson, 2002).

CHAPTER IV

TESTING AND RESULTS

Testing of the system was done on two levels. The first was the verification and validation level, where the system was tested on its fulfillments of the requirements for the software of the speech recognition system. Testing was conducted for the unit and module levels of the development process with the blackbox testing method used. However, the results of this level of testing will not be discussed here as it is suffice to say that the software is working properly. The results that are discussed here will be about the other type of testing. It is the results for the justifications of the theories proposed in this research as well as the discussions to validate the hypothesis postulated prior to development of the speech recognition system.

4.1 Scope and Domain

The input speech is taken from the speech database and fed into the system in syllable form. Each individual speech input has a corresponding text input so as the learning process can utilize both to create a model of the language being taught to the

system. For the purpose of training, samples from four speakers were used as the fifth one was for testing purpose.

4.2 The Process

The testing process was rather simple. First of all, the required parameters like the *Alpha* value and the success rate were set. Then training of the network would be carried out in accordance to some set number of epochs. Then comes the testing and soon the process is repeated for other values of the parameters of the network. The results of the testing will then be tabulated as shown in Table 4.1.

4.3 Results

Table 4.1 shows the result of the initial training and testing of 36 words of the corpus. The words satu , dua , tiga , empat , lima and enam from speaker 1 (3 tries), speaker 2 (2 tries) and speaker 3(1 try) were used. The results of this testing are tabulated in Table 4.1. A second round of training and testing was carried out with the full list of words from 2 speakers; i.e. speaker 1 (1 try) and speaker 2 (1 try). The results of the training are shown in the following Table 4.2.

Table 4.1 The success rates (%) of the initial Training of the Network

Epochs	Alpha Values							
	0.5	**0.1**	**0.05**	**0.01**	**0.0005**	**0.0001**	**0.00005**	**0.00001**
2	89.0772	89.6045	89.5669	89.4162	4.4444	15.1036	29.4915	75.3672
10	91.2686	90.3236	89.81	89.6353	41.849	8.8752	12.1007	39.7124
25	91.7427	91.3429	90.6562	89.6827	68.2095	12.0382	11.2516	22.2773
50	91.9131	91.7093	91.3593	89.7729	77.3059	40.0864	13.9448	16.9647
100	92.0009	91.898	91.7212	90.2802	81.6267	63.0508	40.4386	16.0944
150	92.0305	91.9617	91.8435	90.8796	83.0336	70.6551	55.704	16.0946
200	92.0454	91.993	91.904	91.1808	83.7378	74.4419	63.3117	16.4747
300	92.0604	92.0258	91.9665	91.483	84.4035	78.2405	70.8779	19.3217
500	92.0724	92.0516	92.016	91.7255	84.8215	81.3304	76.9043	40.7256

Table 4.2 The success rates (%) of training the network with a full list of words

Epoch	**Alpha Values**		
	0.0001	**0.00005**	**0.00001**
2	8.6358	16.2703	59.4493
10	5.7231	7.1908	25.3044
25	5.5021	7.3939	13.5939
50	36.5457	7.21	11.3181

Table 4.3 is a snapshot of the success rate for the testing of the network with a full list of words from the database and at *Alpha* rate of 0.00001 and at 50 epochs of training. The merits of the choice of such parameter values will be discussed in Section 7.4.

Table 4.3 The results at Alpha rate 0.00001 and 50 Epochs

Average Success Rate		
	Word%	Char%
"tu"		
primary	25	31.24555
secondary	4.501916	5.521767
tertiary	4.166667	10.67921
"wa"		
primary	0	0
secondary	0	0
tertiary	0	0
"sa"		
primary	0	0
secondary	0	0
tertiary	0	0
"ti"		
primary	0	11.36364
secondary	11.05477	14.41648
tertiary	0	0
"du"		
primary	6.25	9.52381

secondary	4.640152	2.000721
tertiary	3.125	5.681818
“ga”		
primary	0	0
secondary	0	0
tertiary	0	0
“em”		
primary	0	3.125
secondary	3.095238	3.890562
tertiary	0	0
“pat”		
primary	0	0
secondary	7.407407	9.010417
tertiary	0	0
“li”		
primary	0	9.259259
secondary	3.851541	6.380952
tertiary	0	0
“ma”		
primary	0	0
secondary	3.708791	1.483516
tertiary	0	0
“en”		

primary	12.5	10.9375
secondary	4.545455	5.555556
tertiary	27.5	12.01923
“nam”		
primary	0	0
secondary	6.468254	5.412191
tertiary	0	0
“tu”		
primary	0	0
secondary	4.315606	5.492424
tertiary	0	0
“juh”		
primary	0	0
secondary	0	0
tertiary	0	0
“la”		
primary	3.111111	19.58204
secondary	4.304029	4.871867
tertiary	2.941176	18.29268
“pan”		
primary	0	4.901961
secondary	6.851852	8.630952
tertiary	0	0
“sem”		

primary	0	50
secondary	0	0
tertiary	10	8.333333
“bi”		
primary	0	0
secondary	3.125	3.846154
tertiary	0	0
“lan”		
primary	0	0
secondary	7.216981	11.62935
tertiary	0	0
“sep”		
primary	0	0
secondary	3.333333	4.054054
tertiary	6.25	5
“pu”		
primary	0	0
secondary	3.333333	4.166667
tertiary	0	0
“luh”		
primary	14.28571	34.375
secondary	5.399719	8.802981
tertiary	5.555556	15.90909
“ki”		

primary	0	2.713178
secondary	5.441176	4.547804
tertiary	0	0
“ri”		
primary	0	0
secondary	1.724138	0.724638
tertiary	0	0
“ka”		
primary	0	0
secondary	2	1.694915
tertiary	0	0
“nan”		
primary	0	0
secondary	4.371981	3.621968
tertiary	0	0
“at”		
primary	0	0
secondary	1.631579	1.086957
tertiary	7.142857	2.5
“tas”		
primary	0	16.99029
secondary	14.76608	18.6943
tertiary	0	0
“ba”		

primary	0	17.41294
secondary	0	0
tertiary	2.380952	13.72549
“wah”		
primary	0	0
secondary	0	0
tertiary	0	0
“br”		
primary	6.25	5
secondary	1.612903	1.973684
tertiary	7.142857	3.125
“ek”		
primary	10	7.142857
secondary	3.448276	4.647826
tertiary	8.333333	3.125

4.4 Observations

Table 4.3 attempts to breakdown the results of the testing of the system into three categories; i.e. the primary, the secondary and the tertiary. The primary category is to measure the success of matching the words (in this case, the syllables) that are being tested on the system. The secondary category is to measure the success of the matching of words by the system based on the first character of the word. The tertiary category is just like the secondary category by is based on the second character of the input word. The results shown in the table was generated from *Alpha* value 0.00001 after 50 epochs of training. The values presented in the table are an average of two tries for each word in the system.

The results were also tabulated from a word and character matching rate of success perspective. The word matching rate is determined by the success of matching whole words and the character matching rate is determined by the success of matching the character at the position by which the character should have been. Thus these results are to show the actually success rates derived from the system and not after some reconstitution of the output words have been applied.

4.5 Discussion

From Table 4.1, we observe that the results seem to point to the fact that the speech recognition system trains well with the *Alpha* value at 0.5 or in other words, the system trains well at a 50% rate of change. However, upon scrutiny of the memory dump of the system, it happens to show that the system maps the inputs to the outputs at 3 distinctive values; i.e. 0, 0.5 and 1. In other words, the 50% rate of change causes a fluctuation that converges too fast on these three values. It would seem that this may affect the recognition rate of the system.

On the other hand, the *Alpha* value of 0.00001 or 0.001% rate of change provides for a wider spectrum of mapping of values. The memory dumps from the system too proves this as the memory categorically grows as training continues. However, the system also begins to show signs of convergence to distinctive values after training at epochs of 500 or greater. As it is, these interesting observations seem to provide insight to the natural capabilities of the homunculus network in handling the training and testing of the system.

The chart in the following Figure 4.1 illustrates the points observed from Table 4.1. As can be seen, the top 4 *Alpha* values of 0.5, 0.1, 0.05 and 0.01 seem to point to very good performance levels on the outset. However, it is the bottom 4 values that prove

to be useful at all. Even then, there seems to be a quick convergence of to certain output values after a small amount of training.

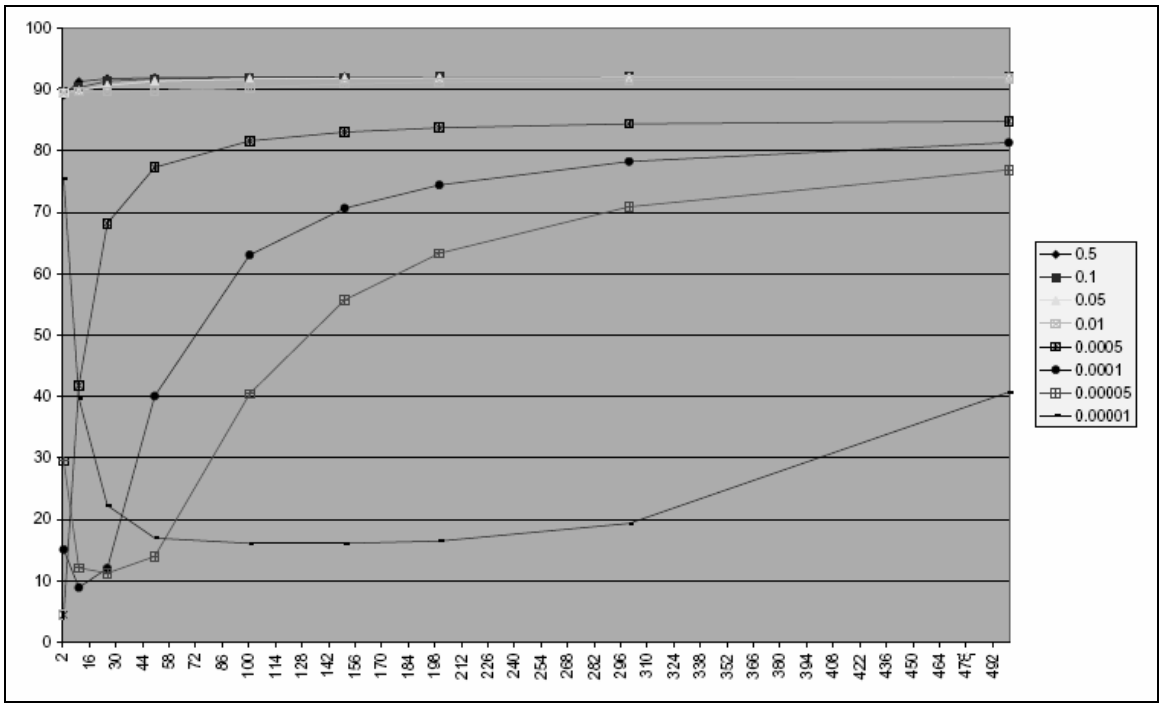


Figure 4.1 The success rates of different *Alpha* values at different epochs

An even more interesting point of observation is this; the so called convergence seems to emerge after the system has accumulated a certain amount of change or fluctuation. With the data from Table 4.1, it seems that the system begins convergence after about accumulating changes that culminates to the value 0.005, or after 0.5% of change has occurred. To prove the point, a second round of training was conducted. The results were tabulated in Table 4.2.

As observed, the convergence phenomenon occurs again after the system was trained for 50 epochs at *Alpha* value 0.0001. As of yet, it cannot be ascertained whether this observation is a constant of the system or that it might just be an emergent property that has associations with the training data utilized. However, an explanation to the reason for such an occurrence will be forwarded. Though it may not be fully conclusive, it may prove to be quite of use in later stages of this research; perhaps even for a further study into the homunculus network.

This 0.5% convergence might have occurred because of the organization of the network. The training of the network is based on a feed-into-system-and-get-an output type of response. Success rates are determined by the addition of the number of correct matches, when the input matches the output of the system, and then dividing this sum with the number of inputs fed into the system. Therefore, a lot depends on the number of successes in the matching of inputs to outputs.

The updating of the network is happens at two fronts; the weights are updated on a winner takes all strategy and the inputs are scaled based on the fluctuation that is observant from the neuron perspective. This means that the network does not just classifies through weight changes but also through neuron states. As more inputs are fed into the system, the neurons in a layer of the network will experience certain changes in states. Some neurons will collapse as the scaling function drives the output of the neuron to approximate 0. Other neurons might grow to overshadow and subsequently dominate the output of the entire layer in which the neuron resides as the scaling function drives the output of the neuron to values that are extremely large.

Therefore, an accumulative effect might have occurred to the system. As the fluctuations gather to the point that the system has changed its state by a portion of 0.5%, output of the system begins to converge to certain local minima or maxima. In the case of *Alpha* values of 0.5 to 0.01, it can be acknowledged that the training of the system would already have accumulated 0.05% change at the first epoch. Therefore, there is no possibility of the system ever being able to map the inputs to the outputs correctly. This also makes for training success rate that is high but a recognition success rate that is low.

Another interesting observation that might back up the previous insight is the training success rate at *Alpha* value 0.00001 or at 0.001% rate of change. Such a small rate requires a more time to gain momentum in training. As can be seen, it only leaves its local maxima after the 3^{rd} or 4^{th} epoch and later regains the maxima after 500^{th} epoch or more. What is interesting to note is that the system is capable of extending itself to a local

minima and then leaving it. This means that the range of change in the system must stay within 0% and 0.05 % in order for the system to be efficient.

However, this observation also suggests that the organization of the network might not be suitable to handle such inputs as the system is not particularly adept at achieving satisfactory results.

More questions have formed than there are answers to counter them. What are the means by which we can use to gain better results? Is an average of 10% the real capability of any learning algorithm and that better results are a product of massive parallelism? Is the convergence phenomenon a reflection of the actual rate of learning that is inherent in all sentient life? Can it be that all that is wrong is the internal representation of the system that does not match the expected outcome of the programmer?

Perhaps the only thing that is of question is the perception by which the system has of the environment. As Maturana and Varela has mentioned, an autopoietic system is operationally closed but organizationally opened. Perhaps too much emphasis has been given to the observer and not enough on the observed. Perhaps it is time to think of ways to redefine the method of learning and not the material for learning. Perhaps all these questions will be answered in time but they will not be answered if we do not carry on with the research

CHAPTER V

CONCLUSION

Initially, the research carried out was to develop a speech recognition system that would be more human like in nature. However, as more work was done into the study of autopoiesis and finding a method to incorporate autopoiesis into a speech recognition system, it became evident that the theory of autopoiesis became an all encompassing technique. It was not enough to formulate a set of rules to regulate the speech recognition process or to add certain modules into a conventional speech recognition system to make it autopoietic-friendly . The entire speech recognition architecture had to be tailored to embrace the theory of autopoiesis. It is this observation that led to the introduction of homunculus network.

Homunculus network is a neural network that attempts to inculcate the concepts that were brought forward by Maturana and Varela and advocated by countless other researchers like Boden, Vernon and Dermot. The network is meant as a way to interpret the formal theory of learning as derived from autopoiesis into a practical and useful technique that could be utilized in day-to-day problems such as speech recognition. This network, though still very much in need of further development, has great promise because it is scalable, adaptable and robust.

Any system built on this network has the ability to have its learning process scaled to a proportion that is manageable. The network is also adaptable in the sense that the same

network architecture can be easily translated into another problem domain without much hassle. The robustness of this network is in the fact that it can take in different types of input vectors due to its real-valued input methods, thus making it easier for anyone to integrate the network into any system with ease.

5.1 Contributions

The first contribution of the research done here is the understanding the intricacies involved in developing a speech recognition system. On top that, the speech recognition system that was built was targeted at the Malay language, Bahasa Melayu. Specific issues that are related to the choice of language and the choice of AI technique were also identified. These issues became the basis for the improvement of speech recognition in Bahasa Melayu.

The most important aspect of this research is the introduction of the Homunculus Network. This neural network was derived largely from the theory of autopoiesis and the learning theory of Piaget. The network was meant as an answer to the issue of learning that was identified through the course of researching a method to develop a speech recognition system for Bahasa Melayu. Apart from these, the research work done here also resulted in a number of papers produced for conferences. A list of papers presented is noted in Appendix B.

5.2 Future Work

As a matter of improving on the homunculus network, a need to review the activation functions and the scaling function of the nodes are in order. I believe a great improvement on the network can be properly derived if the organization of the structure of the network can be reviewed as well. As for the speech recognition system, a method to improve on the results obtained might be to revamp the training methods of the system as well as a new way of acquiring the input speech for the training samples.

REFERENCE

Boden, Margaret A (2000). Autopoiesis and Life. In *Cognitive Science Quarterly (2000) Vol 1*, pp 117 145

Bose, N.K. and Liang, P. (1996). *Neural Network Fundamentals with Graphs, Algorithms, and Applications*. McGraw-Hill International Editions

Bryson, Joanna J. (2002). A practical guide to Behavior-Oriented Design (BOD). In *the Proceedings of Agent Technology and Software Engineering (AgeS 02)*, edited by Jörg P. Müller, Erfurt, Germany

Coulter, D (2000). *Digital Audio Processing*. R&D Books, CMP Media Inc
De Callatay, Armand M. (1992). *Natural and Artificial Intelligence: Misconceptions about Brains and Neural Networks*, Elsevier Science Publishers, North Holland

Fausett, Laurene (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, New Jersey

Flanagan, James L (1972). *Speech Analysis: Synthesis and Perception, 2nd Ed*. Berlin: Springer-Verlag

Haffner, P. and Waibel, A. (1992). Multi-State Time Delay Neural Networks for Continuous Speech Recognition. In *Advances in Neural Information Processing Systems 4(NIPS-4)*, Morgan Kaufmann Pub.

Hild, H. and Waibel, A. (1993) Connected Letter Recognition with a MultiState Time Delay Neural Network. In *Advances in Neural Information Processing Systems5 (NIPS-5)*, Morgan Kaufmann Pub.

Aristotle (1952). *Perception, Change and Truth*, Book Gamma, Aristotle-Metaphysics: translated by Richard Hope, Ann Arbor Paperbacks, University of Michigan.

Huang, Xuedong, Acero, Alex, & Hon, Hsiao-Wuen (2001). *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall, New Jersey.

Hunt, Melvin J. (1995). Signal Representation. In *Survey of the State of the Art in Human Language Technology*, Center for Spoken Language Understanding, Oregon Graduate Institute, pp. 11 16

Itakura, F (1975). Minimum Prediction Residual Principle Applied to Speech Recognition, In *IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol 23*, pp 67 72

Kaplan, Harold M (1960). *Anatomy and Physiology of Speech*, McGraw-Hill Inc.

Kosko, Bart (1996). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, New Jersey

Kohonen, T. (1988). The Neural Phonetic Typewriter. In *IEEE Computer, Vol 21*, pp 11-22

Krogh, Anders and Riis, Soren K. (1999). Hidden Neural Networks. In *Neural Computation, Vol. 11, Issue 2 - February 15, 1999*. MIT Press. pp 541-563

Liew, E. S. and Abdul Manan Ahmad (2000). Developing a Model of Speech Recognition Process from an Autopoietic Approach. In *Proceedings of National Conference on Telecommunication Technology 2000,* J.B., Malaysia

Liew, E. S. (2002). Autopoietic Modeling of Speech Recognition. In *Proceedings of The Annual Workshop, National Science Fellowship 2002*, Kuala Lumpur

Manolakis, Dimitris G., Ingle, Vinay K. & Kogon, Stephen M. (2000).*Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. McGraw-Hill Higher Education, USA.

Markowitz, Judith A (1996). *Using Speech Recognition*, Prentice Hall, New Jersey

Maturana, Humberto R. and Varela, Francisco J (1980). *Autopoiesis and Cognition: The Realisation of the Living,* Reidel Publishing, London

Maturana, Humberto R. and Varela, Francisco J (1992). *The Tree of Knowledge: The Biological Roots of Human Understanding*, Shambala, Boston

Morgan, David P. and Scofield, Christopher L (1991). *Neural Networks and Speech Processing*, Kluwer Academic Publishers, Netherlands.

Nusbaum, Howard C, DeGroot, Jenny, and Lee, Lisa (1995). Using Speech Recognition Systems: Issues in Cognitive Engineering, *Applied Speech Technology*, CRC Press Inc., Florida, pp 127 194

O Shaughnessy, D. (1995). Speech Technology, *Applied Speech Technology*, CRC Press Inc., Florida, pp 47 98

Pais, Abraham (1997). *A Tale of Two Continents: a Physicist's Life in a Turbulent World*, Princeton University Press.

Piaget, Jean (1947). *La Psychologie de l Intelligence*, A. Cohen, Paris.

Pressman, Roger S (1997). *Software Engineering: A Practitioner s Approach*, *4th Ed* McGraw-Hill International Editions

Price, Patti (1995). Spoken Language Understanding, In *Survey of the State of the Art in Human Language Technology*, Center for Spoken Language Understanding, Oregon Graduate Institute, pp 90 -100.

Riis, Soren K. (1998). Hidden Neural Networks: Application to Speech Recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1998*, Seattle Washington, USA.

Sakoe, H. and Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition, In *IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol 26*, pp 43 49

Spencer-Brown, G (1969). *The Laws of Form*, George, Allen, and Unwin Ltd, London

Spina, Michelle S. and Zue, Victor W. (1997) Automatic Transcription of General Audio Data: Effect of Environment Segmentation on Phonetic Recognition. In *Proceedings of Eurospeech 97*, Rhodes, Greece, pp. 1547-1550

Tebelskis, Joe (1995). *Speech Recognition using Neural Networks*. PhD Thesis, Carnegie Mellon Uni.

Thurlin, Leena (1996) The Homunculus in neural networks as a symptom of our dialectic of selfhood. In unpublished article, Faculty of Behavioral Science, Department of Psycology, University of Helsinki, Finland. http://www.helsinki.fi/hum/kognitiotiede/writings/selfhood.h.html

Turney, Peter D (1986). Laws of Form and Finite Automata, *Int. Journal of General Systems, Vol 12*, No. 4, pp 307 318

Varela, Francisco J (1992). Autopoiesis and a Biology of Intentionality, In *Autopoiesis and Perception: Proccedings of a Workshop within ESPRIT BRA 3352*, pp 4 – 14

Vernon, David and Furlong, Dermot (1992). Relativistic Ontologies, Self-Organization, Autopoiesis, and Artificial Life: A Progression in the Science of the Autonomous, In *Autopoiesis and Perception: Proceedings of a Workshop within ESPRIT BRA 3352*, pp 41 64

Vidyasagar, M (1997). *A Theory of Learning and Generalization*, Springer-Verlag, London

Wang, Xue and Pols, Louis C.W (1997). A Preliminary Study about Robust Speech Recognition for a Robotics Application, In *Proceedings of Institute of Phonetic Sciences, Amsterdam, Vol 21*, pp 11 20

Zell, A., Mache, N., Hübner, R., Mamier, G., Vogt, M., Herrmann, K. U., M. Schmalzl, T. Sommer, A. Hatzigeorgiou, S. Döring, D. Posselt, M. Reczko, and M. Riedmiller(1993). SNNS user manual, version 3.0. Technical report, , Fakultät Informatik, Universität Stuttgart.

Zue, V., Cole, R., & Ward, W. (1995). Speech Recognition, In Survey of the State of the Art in Human Language Technology, Center for Spoken Language Understanding, Oregon Graduate Institute, pp 4 10

ACADEMIC CONTRIBUTIONS

1. Developing a Model of Speech Recognition Process from an Autopoietic Approach , National Conference on Telecommunication Technology (NCTT), Johor Bahru, 20-21st November 2000.
2. Developing A Model Of The Autopoietic Theory For Speech Recognition , Student Conference On Research and Development 2001 (SCOReD 2001), Kuala Lumpur, 20-21st February 2001.
3. Genetic Algorithm Shell: A Step Towards Standardization of Genetic Algorithm Systems , Student Conference On Research and Development 2001 (SCOReD 2001), Kuala Lumpur, 20-21st February 2001.
4. Design Of A Speech Recognition Engine Through Autopoiesis , 5th World Multiconference On Systemics, Cybernetics And Informatics (SCI 2001) and 7th International Conference On Information Systems Analysis And Synthesis (ISAS 2001), Orlando, Florida, 22-25th July 2001.
5. Speech Recognition Engine: Design Issues from an Autopoietic Perspective , International Conference on Information Technology and Multimedia (ICIMu2001), Universiti Tenaga Nasional, Kajang, 13-15th August 2001.
6. Autopoiesis and Speech Recognition: A New Perspective , The 6th International Symposium on Signal Processing and Its Application (ISSPA 2001), Kuala Lumpur, 13 16th August 2001.
7. Speech Recognition from an Autopoietic Perspective . 2nd Conference on Information Technology in Asia 2001 (CITA 01), Universiti Malaysia Sarawak, Kuching, Sarawak, 17 19th October 2001.
8. Autopoietic Modeling of Speech Recognition , 6th World Multiconference On

Systemics, Cybernetics And Informatics (SCI 2002), Orlando, Florida, 21 24th July 2002

9. Autopoietic Modeling of Speech Recognition , The Annual Workshop National Science Fellowship 2002 (NSF), Kuala Lumpur, 16 18th December 2002.

APPENDIX A:-FIGURES AND DIAGRAMS

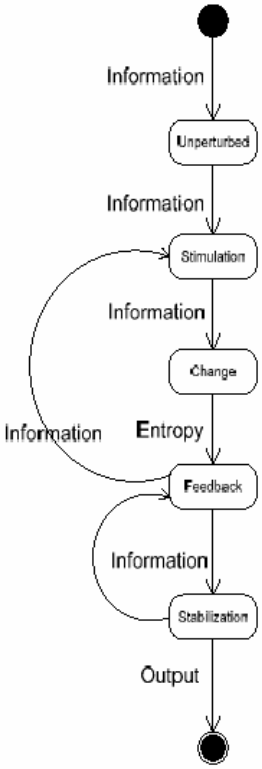


Figure A1 : State Diagram of an Autopoietic Organization

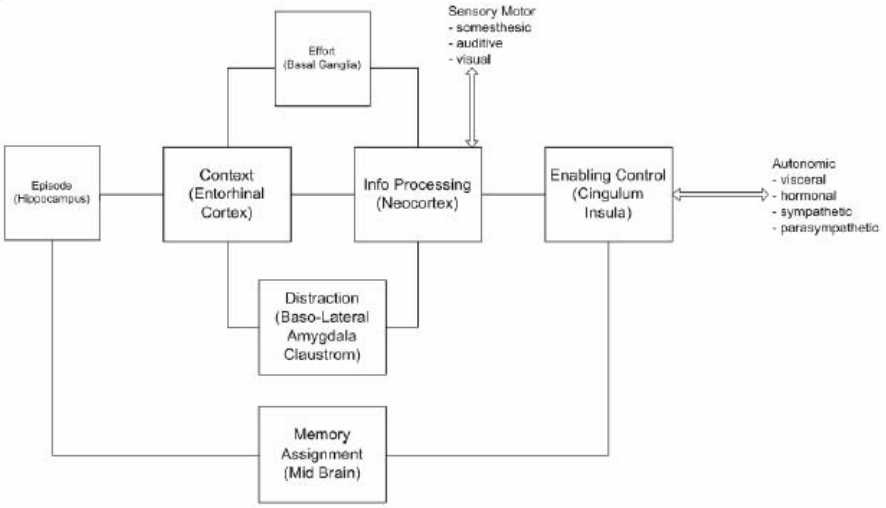


Figure A2 The Main Control Loop of the Brain

Figure A2 : Main Control Loop of The Brain

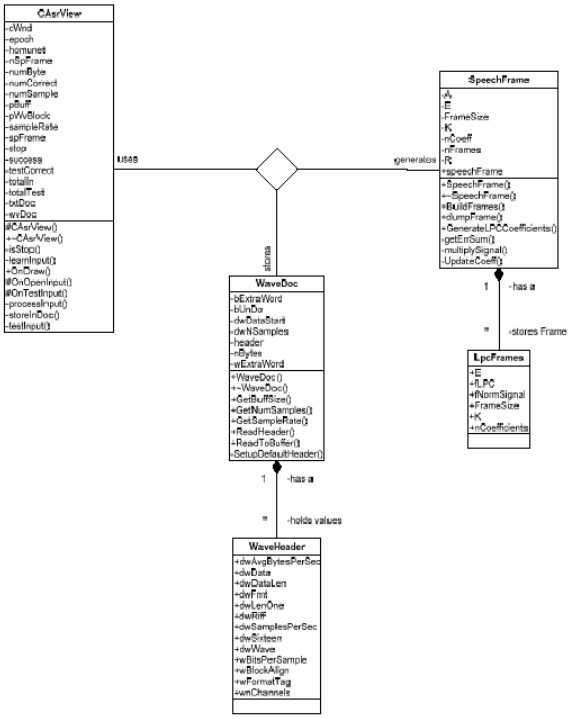


Figure A3 : Object Relationship Model of the Feature Extraction Functionality

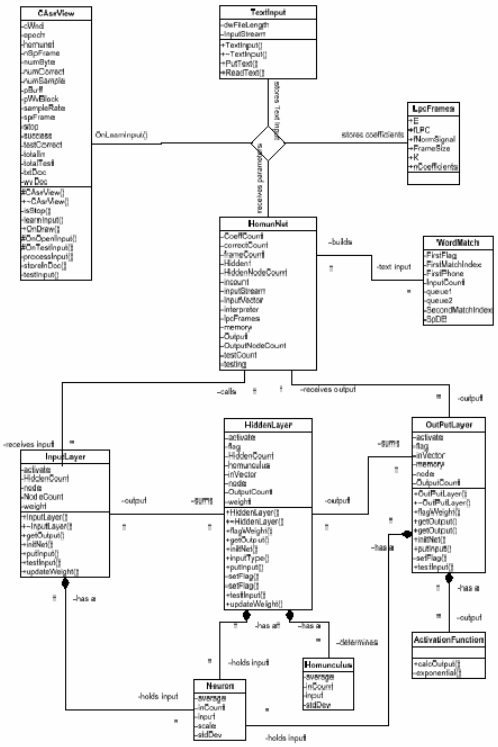


Figure A4 : Object Relationship Model of the Phoneme Recognition Functionality

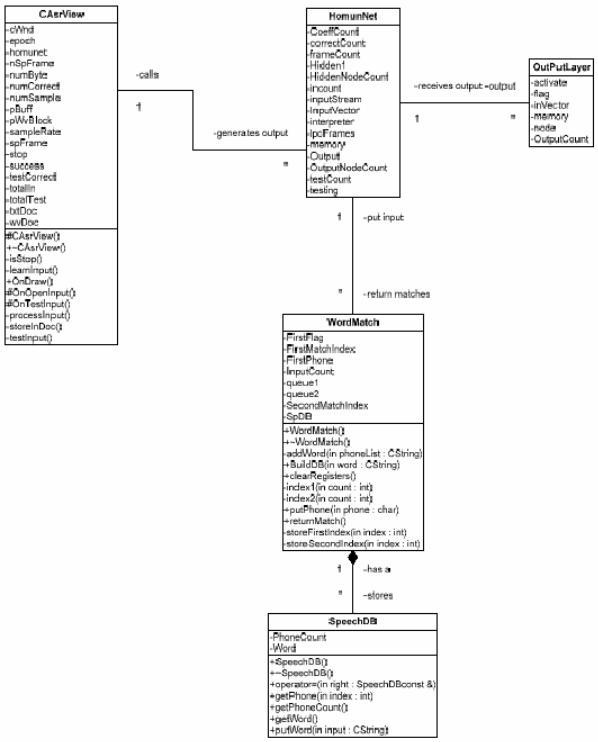


Figure A5 : Object Relationship Model of the Word Matching Functionality

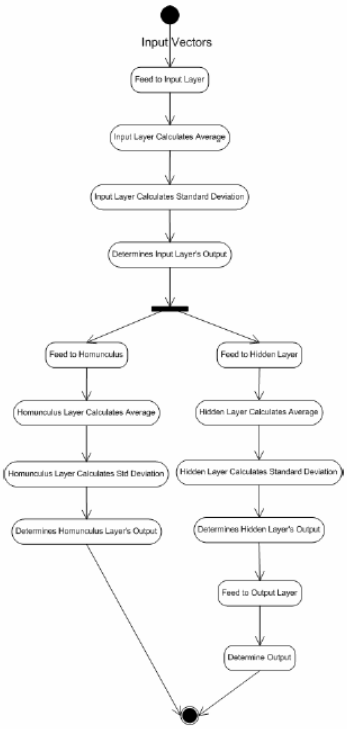


Figure A6 : Activity Diagram for the Training of the Homunculus Network

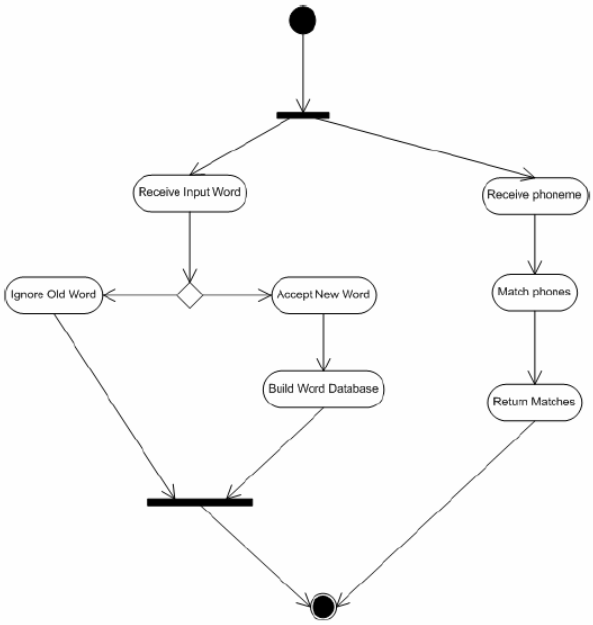


Figure A7 : Activity Diagram for the Building and Matching of Input Words