Chapter 2

Literature Review

2.1 Introduction

Natural language processing (NLP) is a computerized way to analyze text and speech based on a set of methods.

It has various definitions and few of them are listed below.

- It's a field of computer science and linguistics concerned with the interactions between computers and human languages [20].
- Instead of using Boolean logic, the user simply can type in a question as a query. The simplest processing just removes stop words and uses statistical approaches. Natural language processing is the process of using linguistic analysis to infer meaning from human-written text that could not be extracted using the individual word meanings [21].
- It's a simulation of human language processing on the computer by programming the knowledge of human cognitive mechanisms [22].
- It's a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages [8].

2.2 Approaches

There are three main approaches for natural language processing, they are, Symbolic, Statistical and Connectionist. [9]

2.2.1 Symbolic

This method originated in the late 1950's and this tries to model intelligent behaviour using physical symbol systems. Under this approach knowledge about the language is included in to rules or other ways of representation (algorithms, data, etc.), [7]

2.2.1.1 Logic based systems

These systems use mathematical logic, and symbols are taken as logic propositions. Logic programs use a declarative style of programming and computation is done through logic deduction. One of the famous logic based programming languages is Prolog.

2.2.1.2 Rule based systems

These systems use a set of rules, inference engine and working memory. A rule consists of 2 parts, condition (left hand side) and conclusion (right hand side) and it is used to represent knowledge. The condition should be true to fire the conclusion.

If (Condition) then (conclusion).

If (A>B) then (B+1)

The inference engine matches the facts against the rules to select a rule and execute it [9].

2.2.1.3 Semantic networks [9]

This is a way of representing knowledge through interconnected nodes and arcs. There are various types of semantic networks, and graphic representation is the most common feature of them. The most famous classification for semantic networks is John F. Sowa's classification. They are as follows:

- Definitional networks
- Assertional networks
- Implicational networks
- Executable networks
- Learning networks
- Hybrid networks

Definitional Networks

This highlights sub type or is-a relation between a concept and its sub type. The oldest known network of this type is the 'Tree of Porphyry' drawn by Porphyry a Greek philosopher. Figure 2.1 shows the Tree of Porphyry.

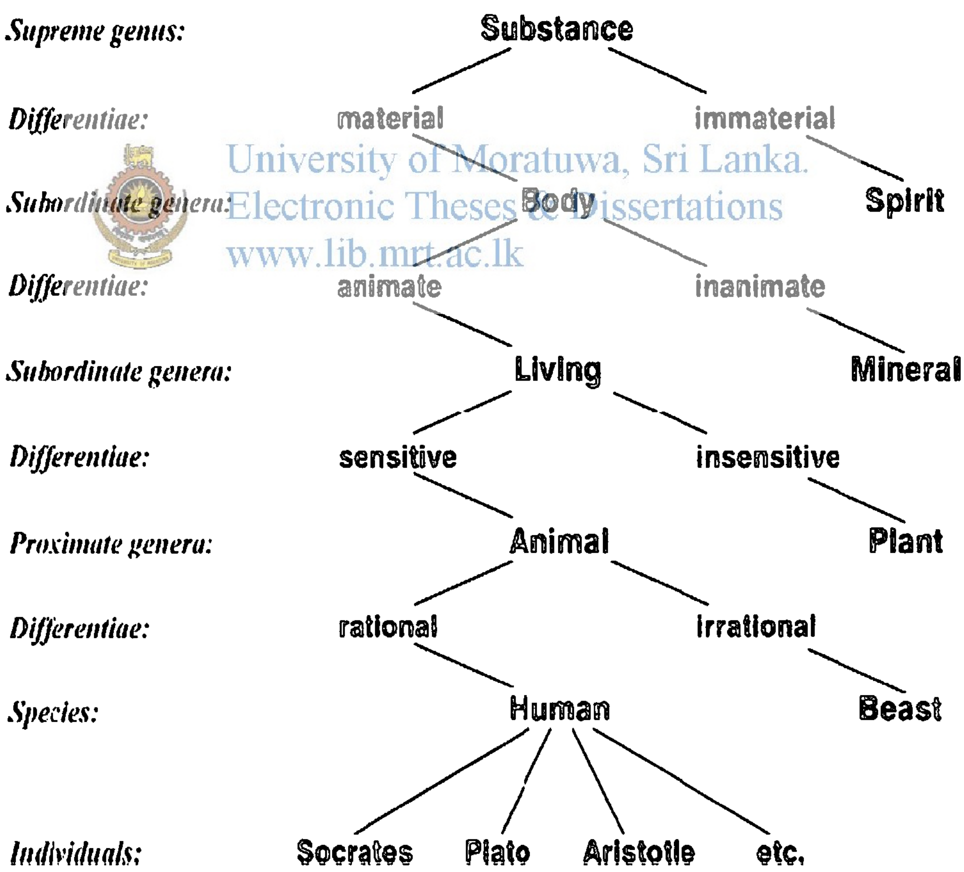

Figure 2.1 - Tree of Porphyry Drawn by Peter of Spain. [6]

Assertional Networks

Assertional networks are a type of a semantic graph, which is used to make assertions about the world and are believed to be true. In these graphs the nodes represent concepts and the edges represent statements, beliefs or assumptions about the concepts. Figure 2.2 shows a sample concept map.

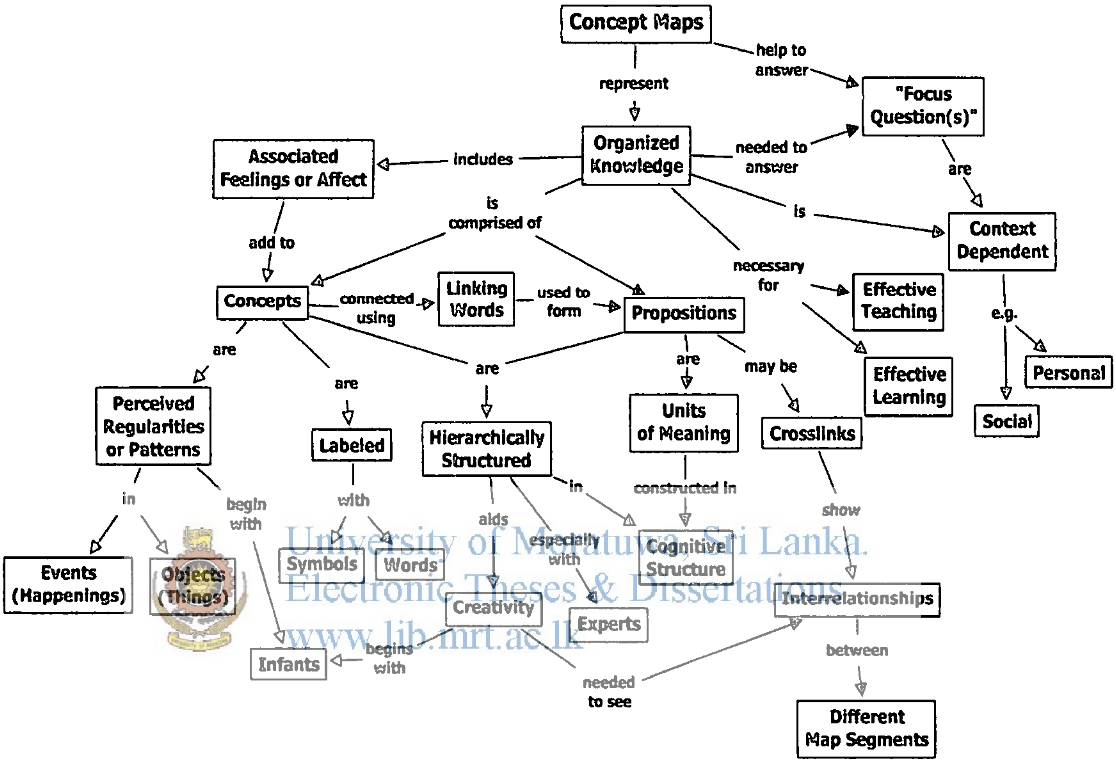

Figure 2.2 - Concept Map [18]

Implicational Networks

This type of semantic networks has propositional nodes and a relationship called implication with an arrow head indicating the next proposition. If the proposition is true then it will imply to the next proposition. If it fails there is no indication about the second proposition. Figure 2.3 shows a sample implicational network.

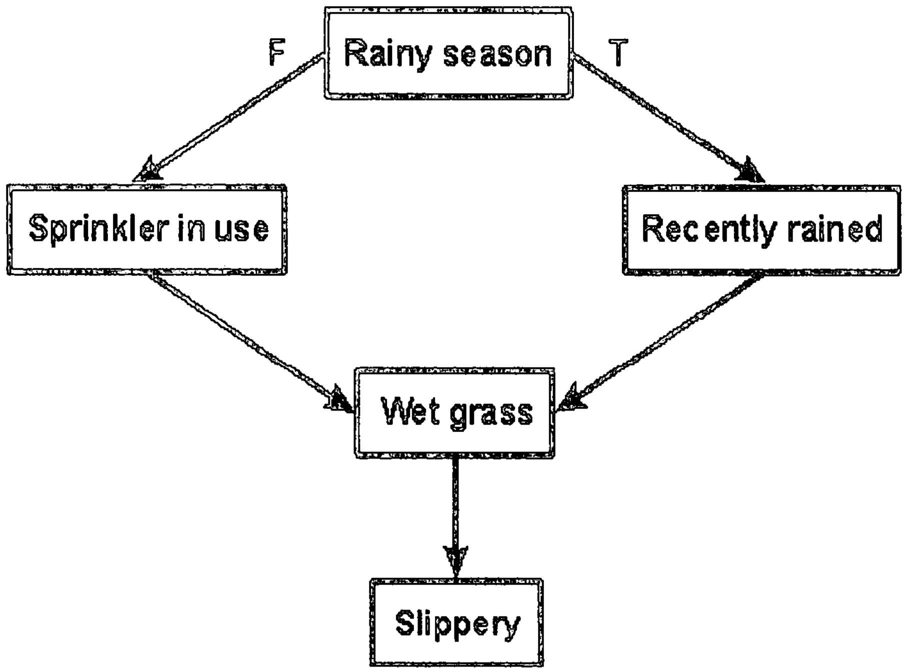

Figure 2.3 - An Implicational Network for Reasoning About Wet Grass. [6]

Executable Networks

These networks have mechanism to make some changes itself. The most common kinds are message passing (Pass data from one node to another, where the data have tokens or triggers for other nodes), attached procedures (A program contain/ in a node performs an action/computation on data that node or nearby node has) and graph transformation (Combine, modify or break graphs in to other graphs by programs external to those graphs triggered).

Figure 2.4 shows a data flow graph with rectangles for passive nodes, which contain data and diamonds for active nodes, which contain functions.

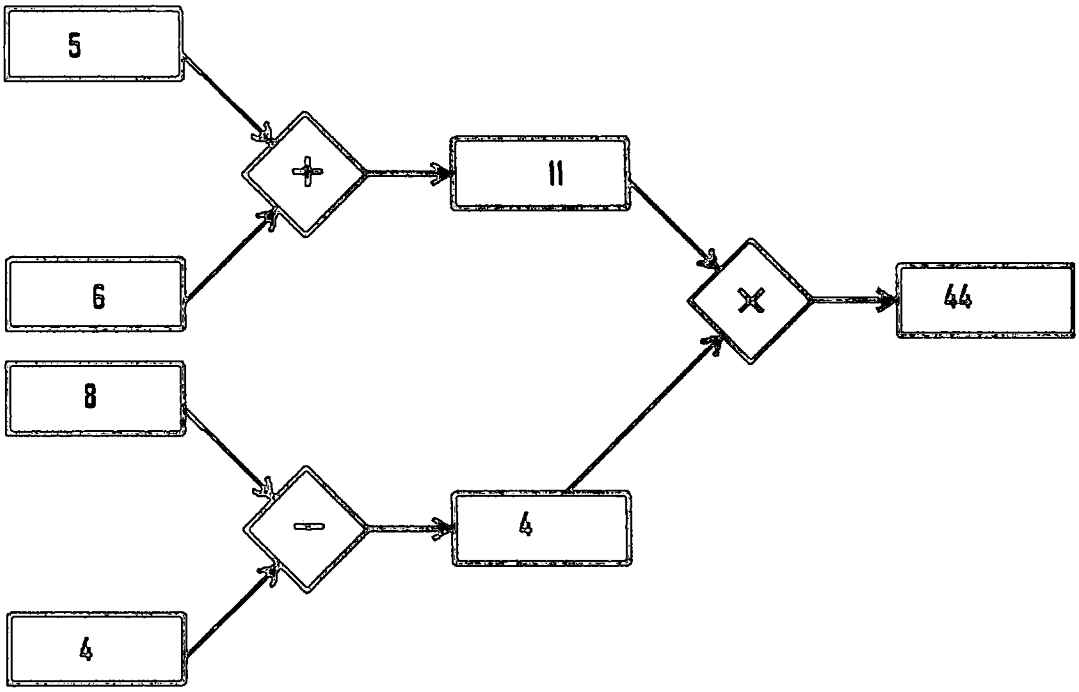

Figure 2.4 - A Data Flow Graph

Learning Networks

When a learning network encounters a new detail/information it will modify itself to handle the new detail/information. There are 3 main ways to achieve this.

1. Convert the new detail/information and add it as a part of the network.

2. If the nodes of the network have weights/probabilities, the network will update the weight/probability according to the new detail/information.

3. Restructure the network according to the new detail/information. This will be the hardest but the most efficient method.

Figure 2.5 shows a sample learning network.

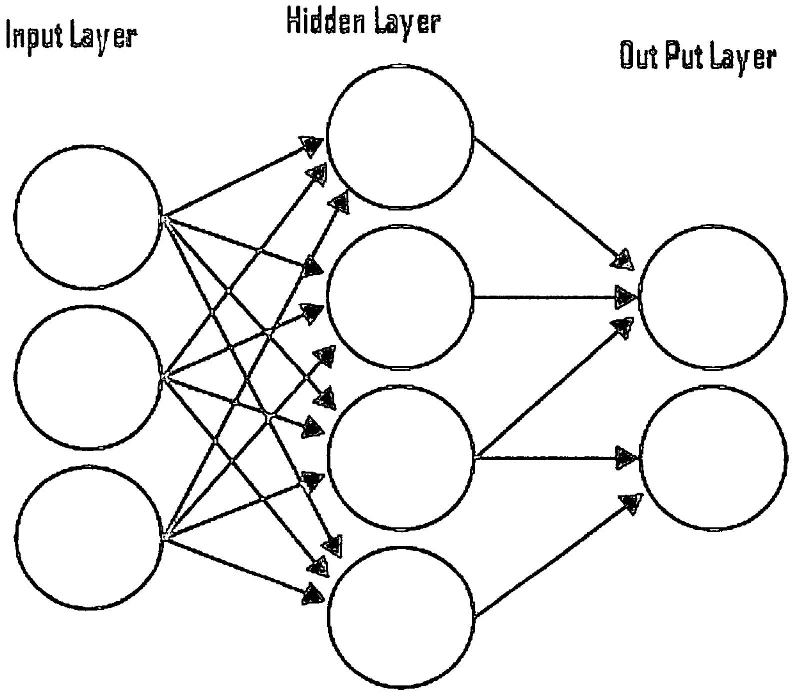

Figure 2.5 - A Learning Network

When a new data/information is available, the weight/probability of the nodes of the input layer and input are combined and will determine the weight/probability of the nodes in the hidden layer. Finally, all the weights/probabilities will decide the weight/probability of the output.

Hybrid Networks

In this type of network, various problems can be handled by various methods. But when one technique is inadequate, researches start developing hybrid networks which have the strengths of several techniques and are better prepared to handle problems more efficiently.

2.2.2 Statistical (Stochastic)

This uses various mathematical techniques with large sets of data (Corpora) to create models. Some of the main areas of statistical techniques are:

- N-Gram Model
- Hidden Markov Models
- Probabilistic Context Free Grammar

2.2.2.1 N-Gram Model

The N-gram (When n = 1 called unigram, when n =2 bigram, n = 3 trigram.) model is a stretch of n words and can be used to predict the next word based on the available n words. This uses a corpus to gather prior information (training data).

Unigram model $P(W1)P(W2)...P(Wn)$

Bigram model $P(W1)P(W2|W1)(W3|W2)...P(Wn|Wn-1)$

Trigram model $P(W1)P(W2|W1)...P(Wn|Wn-2Wn-1)$

N-gram model $P(W1)P(W2|W1)...P(Wn|Wn-n-1...Wn-1)$

2.2.2.2 Markov and Hidden Markov Model

In the Markov model all the states are visible to an observer and the parameters will be the state transitions probabilities. But in the Hidden Markov model, only the outputs which depends on states are visible and not the states. A state has different probabilities for all outputs, and a sequence of outputs will give information about the sequence of states. These models are mainly used for pattern recognition in speech, handwriting, speech tagging, etc.

Markov Model

Figure 2.6 shows a sample Markov model.

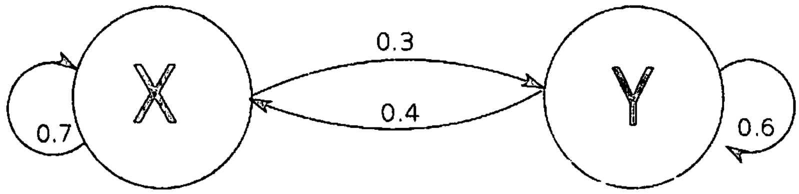

Figure 2.6 - Markov Model

Starting Probability = {'X': 0.6, 'Y': 0.4}

If we want to calculate the probability for the sequence {'Y','Y','X','X'} = P{'X' | 'X'} P{'X' | 'Y'} P{'Y' | 'Y'} P{'Y'} = 0.7*0.4*0.6*0.4

Hidden Markov Model

Figure 2.7 shows a sample Hidden Markov model.

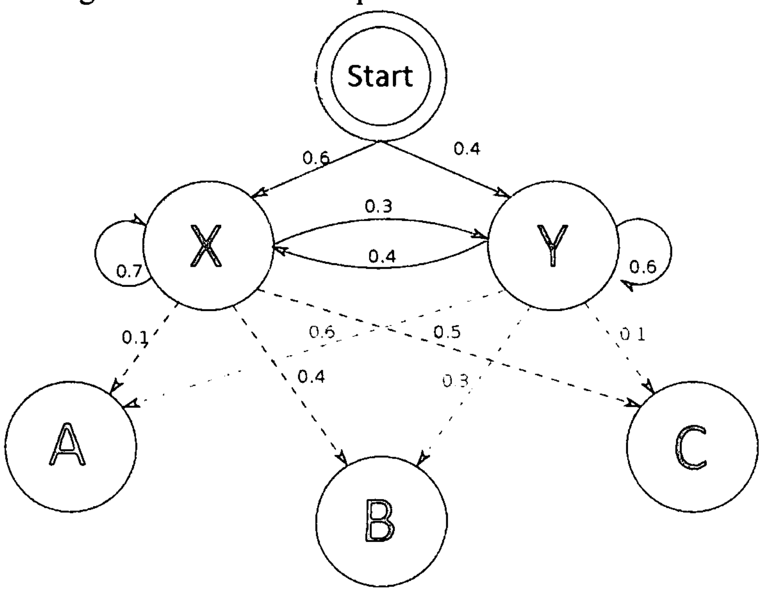

Figure 2.7 - Hidden Markov Model

States = ('X', 'Y')

Observations = ('A', 'B', 'C')

Starting Probability = {'X': 0.6, 'Y': 0.4}

Transition Probability = {'X' : 'X', : 0.7, 'X' : 'Y': 0.3, 'Y' : 'X': 0.4, 'Y' : 'Y': 0.6}

Emission Probability = {'X' : 'A': 0.1, 'X' : 'B': 0.4, 'X' : 'C': 0.5, 'Y' : 'A': 0.6, 'Y' : 'B': 0.3, 'Y' : 'C': 0.1}

If we calculate the probability for {‘C’,’B’}

P{‘C’,’B’} = P({‘C’,’B’},{‘X’,’X’}) + P({‘C’,’B’},{‘X’,’Y’}) + P({‘C’,’B’},{‘Y’,’X’}) + P({‘C’,’B’},{‘Y’,’Y’})

For first term,

P({‘C’,’B’},{‘X’,’X’}) = P({‘C’,’B’} | {‘X’,’X’}) P({‘X’,’X’}) = P({‘C’ | ‘X’}) P({‘B’ | ‘X’}) P({‘X’}) P({‘X’ | ‘X’}) = 0.5*0.4*0.6*0.7

2.2.2.3 Probabilistic Context Free Grammar

In this method probabilities are assigned to each parse tree to resolve syntactic ambiguity and pick the most probable parse tree.

e.g. Book the flight through Houston

According to the probability we can select first parse tree (D1) from Figure 2.8.

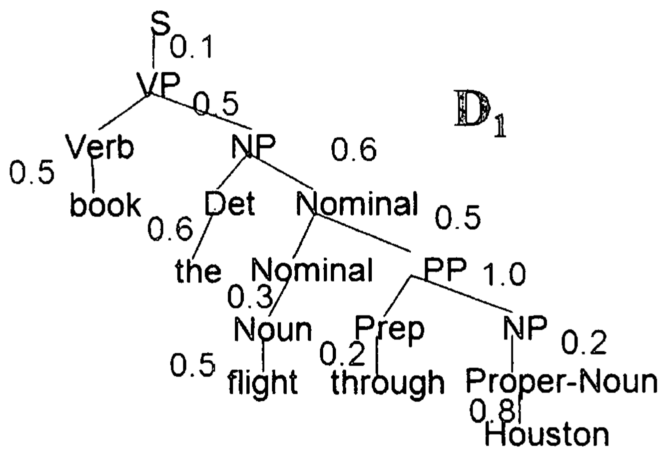

Figure 2.8 - Parse Tree 1 [19]

P(D1) = 0.1*0.5*0.5*0.6*0.6*0.5*0.3*1.0*0.2*0.2*0.5*0.8 = 0.0000216

According to the probability we can select second parse tree (D2) from Figure 2.9.

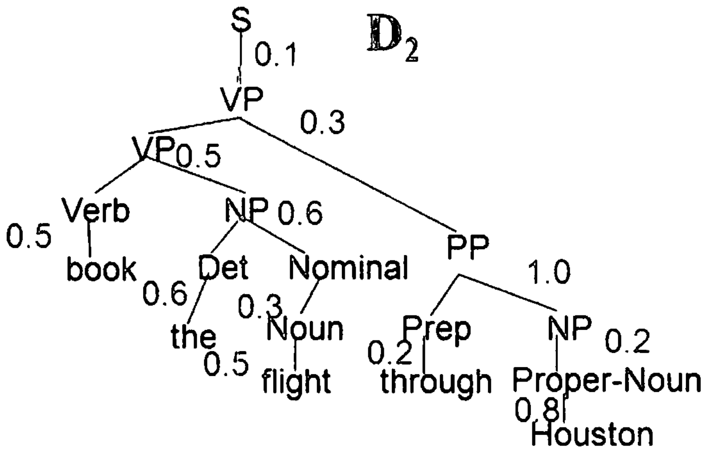

Figure 2.9 - Parse Tree 2 [19]

P(D2) = 0.1*0.3*0.5*0.6*0.5*0.6*0.3*1.0*0.5*0.2*0.2*0.8 = 0.00001296

2.2.3 Connectionist

This is very much similar to statistical methods. The main difference is this connects statistical learning with various theories.

2.3 Existing Systems

2.3.1 ELIZA

Eliza first introduced by Professor Joseph Weizenbaum in 1965 as a computerized conversion simulator. It has the facility to plug in different scripts to represent different characters and the most famous character is "Doctor" which simulates a psychotherapist. Eliza cannot understand anything, and it is designed only to search patterns in the entered text and use that to generate responses. Eliza is written in Basic and it runs on 16 Kb of RAM. [10].

Some of the early computer games based on ELIZA are Ecala, Dungeon, Moria, etc.. Programs like Abuse, ZEBEL, Jesus and I Am Buddha are based on ELIZA but use different programming languages.

2.3.2 Dr. Sbaitso

This was an AI program for DOS based computers in the early 1990's, created by Creative Labs inc. This program tries to simulate a psychologist. The AI engine is similar to the ÈLIZA algorithm and can calculate simple mathematics [11].

2.3.3 PARRY

PARRY is another early chatter bot which attempted to simulate a paranoid schizophrenic (Mental disordered patient). The code compiles and runs using MLISP language (meta-lisp) on the WAITS operating system running on DEC PDP-10. Some parts of the code are written in assembly code [12].

2.3.4 Racter

This is an artificial intelligent computer program, which is used to generate English language prose at random. Written in BASIC on a Z80 with 64k of RAM and run on a CP/M machine and used to compose a book called 'The Policeman's Beard Is Half Constructed'. The macintosh version includes a speech synthesis [13].

2.3.5 MegaHAL

MegaHAL is a computer conversation simulator. When the user types a sentence MegaHAL will answer with a sentence. But MegaHAL doesn't understand the conversation or the sentence structure; it generates its answers based on sequential and mathematical relationships.

The procedure MegaHAL uses is, first it takes the user sentence and breaks it into words (series of alphanumeric characters) and non words (series of other characters) to learn new words. Using two 4th order Markov model (One to predict which symbol follows the sequence of 4 symbols and the other to predict which symbol precedes the sequence of 4 symbols). It tries to generate the replies based on the key words of the input. Frequently occurring words like 'The', 'And' will be discarded and the remaining words are transformed if necessary e.g. my to your [14].

2.3.6 Ultra Hal Assistant

The Ultra Hal Assistant is a chatter bot Loebner price in 2007. Which is developed by Robert Medeksza of Zabaware. Ultra Hal Assistant has a lot of features which other chatter bot's do not have [15]. A few of them are:

1. It has many animated characters to choose from and it generates sound and the user is able to speak to it (Windows version).
2. Ultra Hal can remember important details, dial phone numbers, remind important dates.
3. It can find all the windows programs in the start menu and run them.
4. Helps to browse the internet.

2.3.7 Elbot

Elbot is a chatter bot created by Fred Roberts using artificial solutions technology. It has won some of the artificial intelligence competitions such as Chatterbox Challenge and the Loebner Prize. [17]

2.4 Summary

This chapter covers mainly about the various natural language processing techniques and some of the famous chat bots.

The first section discussed about natural language processing techniques such as symbolic approach, where the intelligent behaviour of they system is modelled using symbols; Statistics techniques, which are the most recent trend of natural language processing, usually based on large corpora performing analysis using text characteristics; and Connectionist techniques, which connects statistical learning with various theories.

The second section discussed some of the successful chat bots which currently exist such as Elisa, Parry and MegaHAL with their details.