SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 3rd Annual Sessions

15th September 2006

SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 3rd Annual Sessions

15th September 2006

Sri Lanka Association for Artificial Intelligence – 2006

Proceedings of the 3rd Annual Session
15th September 2006
Moratuwa
Sri Lanka

Editors:

Prof. Asoka S. Karunananda
Mrs. Kumuduni De Silva

Sri Lanka Association for Artificial Intelligence, Department of Mathematics & Computer Science, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.

Table of Contents

	Page
Keynote Address Fuzzy Chained Episodic Reasoning: A Novel Approach for Human Skill Cloning *Dr. Thrishantha Nanayakkara*	1 - 3
First Sinhala Chatbot in action *B. Hettige & Asoka. S. Karunananda*	4 - 10
A Review paper – Recent developments in Bayesian approach in filtering *D. Ihalagedera & U. Ratnayake*	11 - 18
Distributed Artificial Neural Network Training Using an Intelligent Agent *G.R.C De Silva & Asoka S. Karunananda*	19 - 26
Virtual Tour Agent *Rupesh Selvaraj & Asoka S. Karunananda*	27 - 31
Artificial Intelligence Approach to Effective Career Guidance *Chathra Hendahewa, Maheshika Dissanayake, Savindhi Samaraweera, Narmada Wijayawickrama, Anusha Ruwanpathirana and Asoka S. Karunananda*	32 - 42
Widening Scope of Principal Component Analysis: A Fuzzy Approach *D.S. Kalana Mendis, Asoka S. Karunananda and U. Samarathunga*	43 - 50
Qualitative Realization of the Visual World *D. D. M. Ranasinghe and Asoka. S. Karunananda*	51 - 57
Panel of Reviewers	58

Keynote Address

Fuzzy Chained Episodic Reasoning: A Novel Approach for Human Skill Cloning

Dr.Thrishantha Nanayakkara
Department of Mechanical Engineering, University of Moratuwa

The concepts of fuzzy sets and fuzzy rules have found a wide range of industrial applications in the modern industrial world. The original argument for using a fuzzy set was that it would enable us to model imprecise and uncertain linguistic expressions often made by human experts on real world experiences. Therefore, it has been a promising goal to investigate as to how we could exploit this approach to clone human skills so that problems that can not be faithfully solved by conventional crisp mathematics can be addressed effectively.

To this end, there are two main types of structures to construct fuzzy rules:

1. Zadeh-Mamdani type rules: IF x_1 is A_1 and x_2 is A_2 and x_3 is A_3 and … THEN y_1 is B_1 and y_2 is B_2 and and y_3 is B_3 …. where A_1, A_2, A_3, B_1, B_2, B_3 are fuzzy labels represented by their membership functions.

2. Takagi-Sugeno type rules: IF x_1 is A_1 and x_2 is A_2 and x_3 is A_3 and …THEN z is a x_1 + b x_2 + c x_3 …, where A_1, A_2, and A_3 are fuzzy labels and a, b, c are constants.

However, both these two approaches require the human expert to be able to map sensory stimuli to linguistic conclusions at any given time. Yet, humans seem to work in a different manner where they stick to a particular behavioral schema till certain environmental cue prompt the mind to switch to another schema. Once the mind enters a schema, it neglects the intricate details of the incoming sensory stimuli unless an event occurs that could change the track of the mind. The experiments on motor memory consolidation suggest that the learnt internal models of task and limb dynamics guide the motion control tasks when they are performed with previous memories. This causes the mind to work in a subconscious manner while executing an action. In other words, IF-THEN rules are not used at every single moment while implementing an action.

Therefore, the conventional structures of constructing fuzzy rules have kept practitioners from extracting information from real experts but made them use experimental data to train the parameters of fuzzy inference engines. This has also created the problem of the curse of dimensionality because the practitioners have little information to neglect the unnecessary rules.

Let us consider a real world case of an expert driver. Let us take a simple situation where the driver has to accelerate a vehicle. An environmental cue that triggers the accelerating behavior may be an increasing gap to the vehicle in front or a reducing gap to the vehicle behind. A typical rule that might be executed is:

" IF the gap to the vehicle in front increases and the speed of the car is below the allowed maximum speed THEN accelerate the vehicle ".

A Zadeh-Mamdani type rule for accelerating requires the driver to be aware of the ranges of the force exerted on the accelerator so that fuzzy labels can be assigned to each range of forces. A Takagi-Sugeno type rule will require a function that relates the force exerted on the accelerator and the gap between the two cars or some discrete set of forces to be applied on the accelerator given a gap between the two cars. Even if we knew the exact forces or fuzzy labels of the forces that should be applied, they would be outdated with the changes in lubrication, breaks, fuel injection, and air pressure in the wheels. Therefore, Zadeh-Mandani or Takagi-Sugeno type fuzzy rules have to be adapted to suit the process continuously.

Let us now consider the following structure to construct fuzzy rules:

"IF the gap to the vehicle in front increases and the speed of the car is below the allowed maximum speed THEN accelerate the vehicle TILL the situation becomes normal"

This is introduced as fuzzy episodic reasoning. An episodic rule is characterized by two basic properties:

1. **A set of environmental cues that demarcate the entry and exit of a rule in the form IF a set of entry cues THEN start behavior/action TILL a set of exit cues appear:** The driver will feel easier to explain the rough situation that triggers his accelerating behavior and the rough situation that ends that behavior. Therefore, one can improve the contribution from an expert to the construction of fuzzy rules many folds just by identifying the cues that starts and ends an episode of implementing a behavioral schema.

2. **A set of optimality criteria associated with the behavior/action during a given episode**: In the driver's example, the acceleration behavior will be conditioned by a desire to accelerate in a smooth manner to maximize comfort, and to decelerate when the vehicle reaches the normal situation which is the exit of the episode concerned.

Other than the enhanced potential to acquire the true human expertise, fuzzy episodic reasoning requires not adaptation of parameters once the episodic controller is optimized.

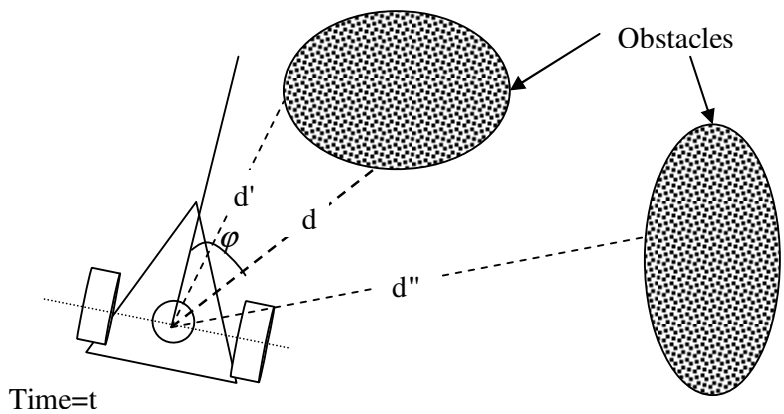

Figure 1: A mobile robot navigating in an environment with obstacles

Let us consider the example given in figure 1, where a non-holonomic mobile robot navigates in an environment with obstacles. In this case, the robot has to take appropriate decisions to navigate from an initial position to a given goal without colliding the obstacles.

The following **Table 1** shows how conditions that demarcate the episodes. The corresponding criteria to evaluate the behavior adopted in an episode are also given.

Episode label	Episodic objective	Entry condition	Exit condition	Evaluation formula
Chase target	To reach the target as close as possible	$\begin{Bmatrix} D > 5 \\ d > 3 \end{Bmatrix}$ or $\begin{Bmatrix} D > 5 \\ d > 1 \\ \lvert\varphi\rvert > \pi/4 \end{Bmatrix}$	$d \le 1$ & $\lvert\varphi\rvert < \pi/4$	$e^{-[D/d]}$
Velocity optimize	Optimize agent speed	$-1 > V > 1$ or	$\lvert V\rvert < 0.7$	$e^{-[\ln(0.5).(0.7-V)^2.5^2]}$
Change azimuth	To direct azimuth towards the target	$\begin{Bmatrix} d < 1 \\ \lvert\varphi\rvert < \pi/4 \end{Bmatrix}$	$d < 1$ & $\lvert\varphi\rvert > \pi/4$	$0.5*[e^{-(\psi-\pi/4)^2}] + 0.25*[e^{-\ln(0.5).(0.02-\tau_l)^2.15^2}] + 0.25*[e^{-\ln(0.5).(0.02-\tau_r)^2.15^2}]$
Avoid obstacle	Avoiding obstacle in a secured and effective way	$d < 2$	$d > 2$	$e^{-\ln(0.5).(0.02-d)^2.10^2}$
Free run	Optimize left write wheel torques	Otherwise*	Until other entry condition	$0.5*[e^{-\ln(0.5).(0.02-\tau_l)^2.15^2}] + 0.5*[e^{-\ln(0.5).(0.02-\tau_r)^2.15^2}]$

Table 1: The entry and exit conditions of episodes and the corresponding evaluation criteria for the mobile robot

First Sinhala Chatbot in action

B. Hettige & Asoka. S. Karunananda

Department of Statistics and Computer Science, Faculty of Applied Science,
University of Sri Jayewardenepura, Sri Lanka.
Faculty of Information Technology, University of Moratuwa, Sri Lanka.

E-mail: budditha@yahoo.com,asoka@itfac.mrt.ac.lk

Abstract

Chatbots are becoming popular as a means for interactive communication between human and machines. Due to their interactivity, chatbots are much better than standard machine translation systems, which may provide unrealistic solutions when the system cannot perform without user intervention. This paper reports on the design and implementation of the Sinhala Chatbot System that can communicate between computer and user, through Sinhala language. This is the first ever Sinhala Chatbot. The current chatbot has been designed to work on Linux and Windows Operating systems. As such the current chatbot can be queried on operating system related concepts such as date, time, and also identify individuals and greet accordingly. This system has been developed as an application of a Sinhala parser that comes under a major component of our project in English to Sinhala machine translation system. Nevertheless, our chatbot is more than a mere application of the said Sinhala paper, but an extension to capture verbal syntax and semantics of Sinhala language into a machine translation. The entire system has been developed using JAVA and SWI-PROLOG that runs on both Linux and Windows. The current chatbot can be used as 'shell' for developing chatbots for any domain.

1. Introduction

Computer-based chat system is one of the most popular communication methods used in the modern world. As such, there are so many chat -systems available world-wide. These chat systems can be broadly classified into two categories, namely, human-human dialog system and human-computer dialog systems. Both systems enable communication using natural languages such as English. The latter systems are generally named as Chatbot.

Developing a human-human dialog system is little challenging. In fact, these systems work only as a mediator between two humans who actually manipulates the respective natural language, but not the machine itself. Stated another way, human-human dialog systems do not need machine level natural language processing abilities. As a result, there are so many human-human chat systems available in the world. Among others, Yahoo Messenger and MSN Messenger are some of the most popular chat systems worldwide. In contrast, developing human-computer dialog system with natural language capabilities is a more challenging task. This has been identified as a more challenging research area in Artificial Intelligence. As an example, Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) [1] is one of major Chatbot system. It is claimed that ALICE has passed the Turing test in two consecutive years [12]. The interest in developments of Computer systems with natural languages capabilities is as old as the field of Artificial Intelligence.

However, at present, majority of these chat systems are available in English language. Therefore people who do not fluent in English Language, cannot use these chat systems due to the obvious reason of the Language barrier. Note that, the language barrier has been an issue not only for communicating with the Chatbot system, but also contributing discovery of knowledge by the persons whose mother tongue is different from English. In this case we are researching to development of a human-computer chatbot system that can be communicated with humans through the Sinhala natural language.

Note that this project has come out as a part of Sinhala on going research in the construction of English to Sinhala parsing systems. As the major components, this system comprises of a Sinhala Morphological analyzer and a Sinhala Language parser [4]. The Sinhala

Morphological analyzer connects with three dictionaries namely, base dictionary, rule dictionary and concept dictionary. Sinhala parsing system receives Sinhala sentence and it returns morphological information of each word in the sentence and syntax information of the Sinhala sentence. Note that English to Sinhala translation system is inherently complex as it requires dealing with two languages. However, since a Chat system generally deals with only one language, chatbots are simpler than translation systems at least in theory.

On the other hand, if the knowledge-base of a chat system deals with English, and the communication is done in Sinhala like language, the same issue of language translations come in chat systems too. It should be noticed that ideally backend of a chatbot could deal with any language other than what is used as the front-end language.

As an application of the Sinhala parsing system, we have developed human-computer dialog system to communicate between computer and a human through Sinhala language. Ideally, this human-computer dialog system is a prototype for Sinhala language interface. In this paper we present design and implementation of a human-computer dialog system (Chatbot).

The rest of this paper is organized as follows. Section 2 describes overview of some existing chatbot systems. Section 3 describes why chatbots are useful. Section 4 reports Sinhala chatbot design and implementation. Finally, Section 5 concludes the paper with a note on further work.

2. Overview of Chatbot System

Alan Turing publishes his paper later called the "Turing Machine" which included the possibility of a machine operating on its own program, modifying or improving it. This is the first idea about intelligent machine. Chatbot is also an intelligent system. Developing an intelligent Chatbot has so many useful applications. This is mainly because; humans want to communicate with various resources through appropriate interfaces. At the outset a chatbot can be considered as an intelligent interface environment. However, developing a chatbot system requires addressing the following issues.

- Computer-based of Natural Languages processing.
- Define and design knowledge base for the chatbot
- Develop suitable algorithms for pattern matching.

Due consideration for the above three factors is crucial for the accuracy and intelligence of a Chatbot system. The first requirement is at the core of any chat system. As per the second point, it is ambitious to develop chatbot to respond in any domain, yet it has its own domain of expertise. The point is also equally challenging. Development of a Chatbot system involves many steps. According to the research in Artificial Intelligence chatbot system, we identify the major steps in a chatbot system (Fig. 1). Brief description of the each step in a Chatbot System is as follows.

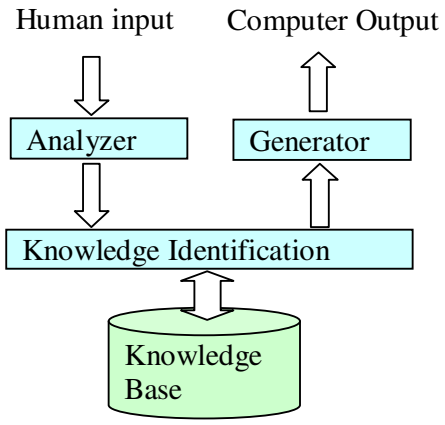

Fig 1: Overview of a Chatbot System

Analyzer reads input sentence from user and analyzes Syntax and Semantic of the sentence. Then knowledge identification engine reads all these information and identifies the suitable answers. This identification is done with the help of knowledge base. Knowledge base is the database or the reservoir of knowledge of the Chatbot system. This is the brain of a Chatbot system. Then knowledge identification engine sends all these information to the generator. Generator generates appropriate grammatically correct sentence to display these information. Note that, from expert system viewpoint, the Chatbot System is an expert system with knowledge identification engine as an inference engine, and generator and analyzer as a user interface. Considering the above generic architecture, mere change of Knowledge base, provides a means for developing chatbot system for any domain.

Brief description of the popular chatbot systems is given below:

A. ELIZA

ELIZA is an early Artificial Intelligent program that was written in the mid 1960s by Joseph Weizenbaum to simulate a non-directive psychotherapist [7]. Also this program operates within the MAC time-sharing system at MIT which makes certain kinds of natural language conversation between man and computer. Input sentences are analyzed on the basis of decomposition rules, which are triggered by key words appearing in the input text. Note that, Weizenbaum wrote ELIZA as an exercise in pattern matching. However, ELIZA had very limited natural language processing capabilities.

B. ALICE.

ALICE (Artificial Linguistic Internet Computer Entity) is a software robot or program that you can chat with using natural language [1]. ALLICE uses AIML (Artificial Intelligence Mark-up Language) files to implement it knowledge [1]. Artificial Intelligence Mark-up Language is a derivative of Extensible Mark-up Language (XML) [11]. It was developed by the Alicebot free software community during 1995-2000 to enable people to input dialogue pattern knowledge into Chatbots based on the ALICE free software technology. ALLICE uses pattern-matching algorithm to identify user input and this algorithm uses depth-first search techniques [2]. ALICE has passed the Turing test in two consecutive years [13].

C. Elizabeth

Elizabeth is another Chatbot system which is an adaptation of Eliza. Elizabeth is used to store knowledge as a script in a text file, where each line is started with a script command notation [3]. These notations are single characters, one for each rule–type. Some commands rule as follows W: Welcome message, Q: quitting message, N: No match, V: Void input etc. Each script command has an index code that is generated automatically [2]. Also Elizabeth has the ability to produce a grammar structure analysis of a sentence using a set of input transformation rules to represent grammar rules. This provides an introduction to some of the major concepts and techniques of natural language processing.

2.1. Why Chatbots

It would be interesting to discuss the importance of chatbots over the standard machine translation systems, in the context of computer-based natural language processing. It is well known that computer-based machine translation is inherently difficult and achieved very little since the inception of AI in 1956. Perhaps, one of the main reasons for drawbacks in machine translation is the restricted involvement of humans in the process of machine translations. This is why concepts like pre-editing and post-editing of documents by humans has done a tremendous impact on development of machine translation systems.

Similarly, Chatbot is yet another approach to bring the human intervention into machine translation. Obviously, in chat systems people always use simple sentences. As such language complexity will not be a serious issue in chatbot systems. On the other hand chatbot systems can always encourage the user to rephrase the question, if the system cannot understand the current phrase. This brings a huge element of interactivity until a reasonable solution is derived. Furthermore, chatbot does not need to bother about complex written grammatical structures of a language, but simple verbal grammar would be sufficient for most of the instance. Therefore, development of chatbot systems would be less time consuming. In addition, due to continuous interaction with the user, chatbot systems can be much easily evolved through the sessions. This is a key difference between traditional expert systems and chatbot systems. The transparency of chatbot systems provides opportunity to detect any anomalies in the answer in early phases of communication. Therefore, we argue that chatbot system is of great importance as a potential solution for making the machine translation a reality.

3. Sinhala Chatbot Design & Implementation

This section describes design and the implementation of the proposed Sinhala Chatbot system. This system designs to answer simple questions. Furthermore, Chatbot system can do the small operations such as, print the current time and date and run a command etc. Note that, this system is a prototype and we do not design this system in a particular domain. This is mainly because, we need only to demonstrate the design and implement a Chatbot system that uses a Sinhala language

parsing system. The entire system has been developed using Java and SWI-Prolog [10] that runs on Linux and Windows environments. Also this system is designed to work on the client server model. This is mainly because, we need to give access to many people to find the information through our Chatbot system. Note that, all the resources and engine modules are available in the main server. Then client can access this information through the network. Fig 2 shows the client-server architecture of the Sinhala Chatbot. And Fig 3 is the server side design of the Sinhala Chatbot. Brief description of the Server side design is given here. Server socket reads data string from client and pass it into Sinhala language parsing system. Sinhala Language parsing system contains Morphological analyzer, Sinhala parser, Sinhala composer and Lexical dictionaries. There are three dictionaries, namely base dictionary, rule dictionary and concept dictionary [4].

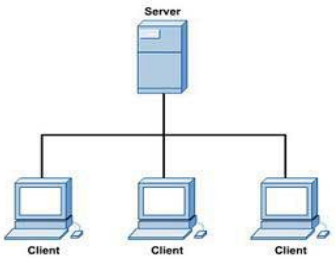

Fig 2: Client-server architecture of the Sinhala chatbot

The base dictionary contains base words of the Sinhala language. In this sense, the base dictionary primarily stores base or root form of each noun and verb (prakurthi) [5]. In addition to base noun and base verbs, irregular nouns, irregular verbs, nipatha are also stored in base dictionary [8].

The rule dictionary stores rules required to generate various word forms [6]. These are the inflection rules for formation of various forms of verbs and nouns from their base words. The concept dictionary contains further information such as synonyms and anonyms for the words given in the base dictionary. The concept dictionary contributes to improve the quality of the morphological analyzer, especially when we are interested in the semantic analysis of words. The Morphological analyzer reads a sentence word by word. For each word, the morphological analyzer identifies grammatical information such as nama (nouns), kriya(verb) and nipatha [6]. This identification is done with the help of three dictionaries mentioned above.

The Sinhala parser receives tokenized words from the morphological analyzer. Also it does analyze the syntax of the Sinhala Sentence. Note that, parser checks correctness of the Sinhala sentence and identified syntax categories.

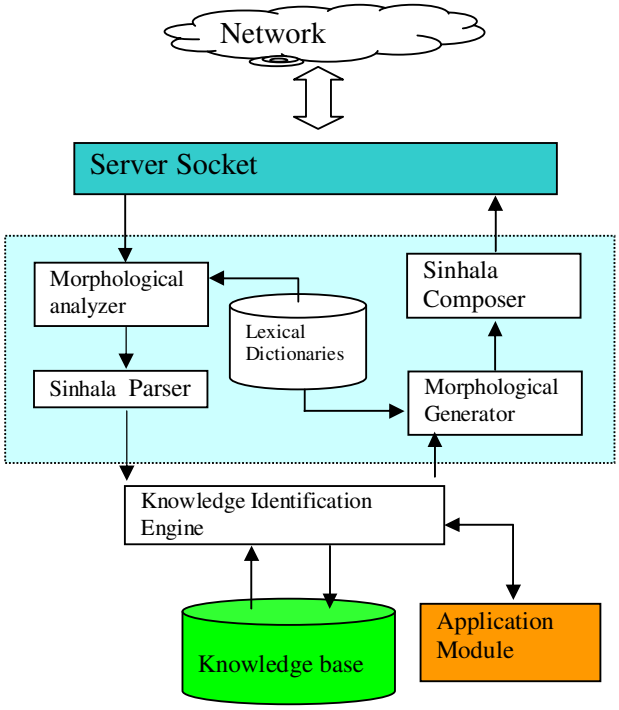

Fig 3: Server side design of the Sinhala chatbot System

In Sinhala Language there are different forms used in written and verbal (talking) forms. These different forms are generated from inflecting a final-verb. As an example in written Sinhala we say 'uu n;a lñ' but in talking form we say 'uu n;a lkjd' according to these two forms we are familiar with talking form. Note that our original parser [4] in the translation system has been designed to handle only the written form of Sinhala sentences. Therefore, here we have done some modifications to the original parser to incorporate the verbal features of parsing in Sinhala language.

Knowledge identification engine reads all the information given from Sinhala language parsing system. Note that, knowledge identification engine is worked as an inference engine. It uses simple pattern matching algorithm to identify user input and find the appropriate solutions from knowledge base. Our chatbot system uses notation (key) for

identifying question patterns. Some of these notations are given bellow.

msg – Message

qyn – Question with yes/no answer form

qni – Question with more answers

qwc – Question with command

qun – Unknown question

qda – Question with direct answer

Large number of question forms can be made available for the chatbot. In this sense these notations are needed to identify answer patterns. According to each pattern, knowledge identification engine generates appropriate answers. This process requires knowledge base. Knowledge base stores all the required knowledge in the Chatbot system. Our knowledge base is implemented using SWI-Prolog data base [9]. Note that, our Chatbot system is designed as an automatically updating system. While users chat through the system, Knowledge base has been updated automatically. Note that, Chatbot system stores user information to improve its performance. Some of the following prolog predicates are used to store knowledge in the knowledge base.

```
user_info(Name).
user_like(Name, Fields).
sysm_command(CommandID, Command).
chat_message(MessasgeID, Message).
chat_question(QuestonID, Question).
```

Brief description of each Prolog predicate is given hear: user_info/1 Prolog predicate stores user name and it is used to identify the users, login to Chatbot system next time. Also system store users interested field by using user_like/2 prolog predicate. Sysm_command/2 predicate stores system command that run on windows and Linux environment. Chat_message/2 and chat_question/2 are the other prolog predicates that used to print messages and ask questions. In addition to these predicates knowledge base stores knowledge about the various domains. This part is not implemented in this version.

The application module can run appropriate commands and read the result. As an example if a user asks the time then system uses this module and runs suitable time function in SWI-Prolog and reads current system time. Note that this module works as a command shell. Further, SWI-Prolog is a powerful tool that can be used for many applications. Our Chatbot system is used with this ability. Then, morphological generator generates appropriate Sinhala words and their grammatical information. All these information read from Sinhala composer and generate suitable grammatically correct sentence. Finally, Server socket reads these sentences and sends it to the particular client.

Our Chatbot system implemented using Java and SWI-Prolog. Java is used to design user interface and the network connection. All other modules are implemented using SWI-Prolog. Fig 4 shows a simple interaction session between a user and the chatbot. At this point, the chatbot uses the Windows Operating systems information as its knowledge The connection settings required to communicate with the Chatbot system is shown in Fig 5.

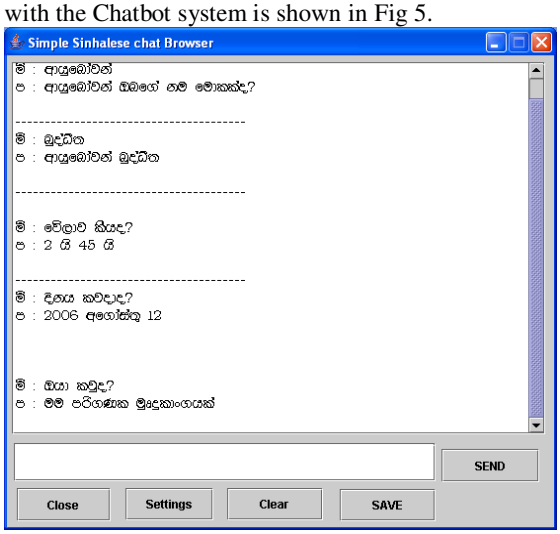

Fig 4: User interface of the chatbot system

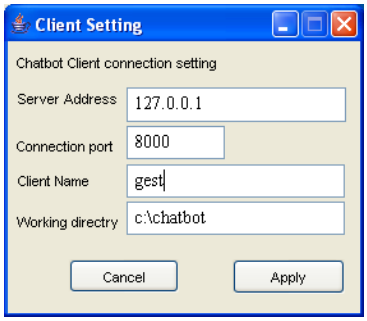

Fig 5: Connection setting of the chatbot system

Now we give an example to describe how system works for the input sentence 'wo oskh ljodo'. Morphological analyzer reads the sentence word-by-word, and identifies 'wo' as an adjective, 'oskh' as a noun, and 'ljodo' as a question verb. Then Sinhala parser reads theses information and identifies syntax

of the sentence such as 'wo oskh' as a subject and 'ljodo' as a final verb. Fig 6 shows parser tree of the analysis of the above

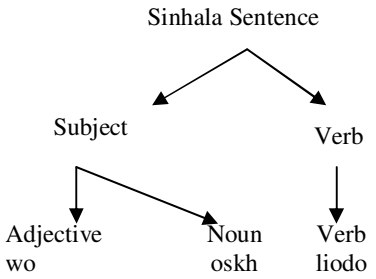

Fig 6: Parser tree for given Sinhala sentence

After that, Knowledge identification engine identifies appropriate pattern to find the suitable answer. The current system identifies only 3 level patterns. These patterns are limited to identify subject, object and verb in a sentence. These patterns are stored by using spattern/3 Prolog predicate. Note that, System uses the standard Prolog matching and unification procedure to find suitable answers. Each pattern contains answer pattern or some task. In this example it identifies spattern(patternID, time, none, tell). Therefore, knowledge identification engine reads system time by using application module. The following code segment shows how Prolog extracts system time into the variable denoted by PD.

```
printtoday(PD):-
        date(A),assert(A),date(Y,M,D),
        retract(A),
        mounth(M,Mo),
        string_concat(Y, ' ', Year),
        string_concat(Mo, ' ', Month),
        string_concat(Year, Month, YM),
        string_concat(YM,' ', PYM),
        string_concat(PYM, D, PD).
```

The values returned after finding an answer will be returned to Sinhala Composer to format the answer to be able to display on the screen.

4. Conclusion & Further Work

This paper presented design and implementation of a Sinhala Chatbot system, which is designed as an application of the developed Sinhala language parsing system. This was designed as an application to demonstrate functionality of Sinhala parser coming in English to Sinhala machine translation system. Note that this version of implementation of the chat system merely use the basic Operating system information as the its knowledge base. However, undoubtedly, the same mechanism can be used by the chatbot to communicate with the user regarding any knowledge base. Ideally the knowledge that is accessed by the chatbot need not be residing on the same machine, and can be anywhere on the internet.

Further work of this project includes extending the chatbot to operate on a more specific domain. As such we would be interested in the construction of knowledge base for chatbot system. Of course, the current version of the chatbot implementation can be used as a 'botshell' for developing any domain specific chatbots. This idea has the similar theme to the use of Expert System Shells to rapid development of expert systems. We also wish to present the Sinhala chatbot as server-side application of client-server architecture. Since the process in a chatbot system is much transparent to the user, it would be easier to incorporate any development in early states of the use of the chatbot system.

5. References

[1] ALICE 2002 A.L.I.C.E AI Foundation, http://www.alicebot.org/, or http://alicebot.franz.com/

[2] Abu Shawar, B. and Atwell, E.: A Comparison Between Alice and Elizabeth Chatbot Systems. School of Computing research report 2002.19, University of Leeds (2002)

[3] Abu Shawar B., Atwell E., "Using dialogue corpora to retrain a chatbot system". In Proceedings of the Corpus Linguistics 2003 conference, Lancaster University, UK, p. 681-690.

[4] Hettige. B, Karunananda A. S. , "A Parser for Sinhala Language – First Step Towards English to Sihala Machine Translation", To appear in the proceedings of International Conference on Industrial and Information Systems, IEEE, August, Sri Lanka, 2006.

[5] Disanayaka, J. B., Basaka Mahima:6 prakurthi, S. Godage & Bros, 661, P. D. S. Kularathna Mawatha, Colombo 10 , Sri Lanka, 2004.

[6] Gunasekara A. M., A Comprehensive Grammar of the Sinhalese Language", Asian Educational Services, New Delhi, Madras, India.1999.

[7] Weizenbaum J, ELIZA--A Computer Program For the Study of Natural Language Communication Between Man and Machine. Communications of the ACM, Vol. 9, No. 1, pp.36-45. 1965.

[8] Karunathilaka W. S., Sinahala Basha Viyakaranaya, M.D. Gunaseena & Company, Clolombo 11, Sri Lanka,2003.

[9] Micheal A. C. Natural Language processing for Prolog Programmers", Prentice Hall, Upper Saddle river, New Jersey. 2002.

[10] http://www.swi-prolog.org

[11] http://www.xml.com/pub/a/98/10/guide0.html

[12] http://en.wikipedia.org/wiki/Loebner_prize

[13] http://en.wikipedia.org/wiki/Turing_test

Recent Developments in Bayesian Approach in Filtering Junk E-mail

D. Ihalagedara & U. Ratnayake

Department of Electrical and Computer Engineering,
The Open University of Sri Lanka

E-mail: dhammikai@pdn.ac.lk

Abstract

Junk mail is one of the main problems in Internet. There are several methods for the automated construction of filters to eliminate such unwanted messages from user's mail system. This paper is mainly concerned about the Bayesian filtering method and its different types of applications in junk e-mail filtering. Bayesian technique is trained automatically to detect spam messages. Several implementations that use Bayesian techniques are available as software. Any user can apply this software in different layers of client side or server side. But spammers are now trying to defeat Bayesian filters by including random dictionary words and/or short stories in their messages. The Bayesian filter can be moderated to block the new spammer techniques. The efficiency of the Baysian filter is greater than the other e-mail filters. If any one wants to filter spam out of email, it is strongly recommended not to automatically delete messages. The same is true for your real email; instead of deleting it, move it to another folder. That way, you'll build a collection of spam and non-spam messages, which will come in handy for training filters.

Key words Bayesian, spam, email filtering

1. Introduction

The number of users connected to the Internet is increasing daily. The electronic mail (E-mail) is quickly becoming the fastest and most economical feature in the Internet users. Every e-mail user can function his/her mail account and mailboxes as he/she needs. Unfortunately some virtues that have made e-mail popular have also enticed flooding of unwanted e-mail.

With the proliferation of direct markets on the Internet and the increased availability of enormous e-mail address mailing lists the volume of junk mail has grown widely in past years. This junk mail often referred colloquially as "Spam". There is no standardized definition for spam, however in generally the word "spam" is used to refer to unwanted "garbage" e-mail massages. Spam constitutes a major problem for both e-mail users and Internet Service Providers (ISP). The spam is costly for both users and ISP.

As a result of this growing situation some automated and manual methods for filtering such a junk from legitimate e-mail are needed. Many of junk mail filtering products are available, [4] which allow users to handcraft a set of logical rules to filter junk mail. The problem with the manual system is construction of rule sets to detect junk mail. It points out the need for adaptive methods for dealing with this problem. The automated learning rules to classify e-mail are introduced in [4]. While such approaches have shown some success for general classification tasks based on the text of message, the average number of spam messages received continues to increase exponentially. Figure 01 shows recent statistics on the number of spam messages received by one e-mail user and taken from [5].

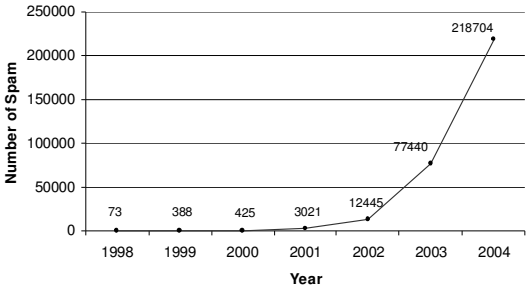

Figure 01: Annual Spam Evolutions

The spam cost to the ISP can be seen at two levels; increase of load of e-mail servers and waste of bandwidth. The slower Internet access is arising according to the bandwidth of the ISPs.

Spam filtering can be applied at the client level or server level. Several solutions and techniques for filtering spam were proposed in the literature. They are based on the header

analysis, address lists, key word lists, digital signatures and content statistical analysis (Bayesian Technique) [6, 7].

The Bayesian technique is the elaborate solution in the spam filter, which constitutes the main core of so many spam filtering software. Generally Baysian technique is used in conjunction with the other techniques in the spam filtering process. They can be applied as several layers. The Baysian filter divides manually the corps of a high number of e-mail messages into two classes; legitimate (ham) or illegitimate (garbage or spam).

2. Anti Spam Technologies

Due to the huge increase of spam in the past years the researches pay more attention for filtering spam. Many researches are presently working in the implementation of new filters that prevent spam from reaching their destination either by blocking it at the sever level or the client level. In January 2003 and 2004, a conference on spam took place at MIT in Cambridge and the Coalition Against Unsolicited Commercial E-mail (CAUCE) [8] was established. While CAUCE is trying to introduce legislations that would make spamming illegal [cauce], some research groups and companies are trying to fight/block spam.

2.1. Centralized filtering server

This architecture used a single anti-spam filter that runs on centralized organization-wide mail server.

2.2. Gateway filtering

All inbound e-mail is routed through a filtering gateway before being delivered to the mail server. Gateway services work well with web based and mobile access to e-mail.

2.3. List-based filtering

This method is richer than the other methods, also it is different to the other methods. This method is operating at the server level. Today, black-listing and white-listing are ineffective, although server-based solutions adopt them as an auxiliary technique often to be integrated with challenge/response. Black-listing resources have become less effective since spammers learned to change their source address to get around the recipient's defenses.

2.4. Rule based filtering

Rule based filters assign a spam score to each e-mail based on whether the e-mail contains features typical of spam massages, such as keywords and HTML formatting like fancy fonts and background colors.

2.5. Heuristic filtering

This method uses baseline artificial intelligence to deliver an automated spam detection process [10]. These automated mechanisms categorize incoming e-mail messages as spam or legitimate based on known spam patterns. Main advantage is that no human actions are needed for filtering here.

2.6. Receipt-time filtering

Once an SMTP server accepts an incoming SMTP connection, it can use a wide variety of techniques to detect and reject spam. An effective heuristic test is to see if the incoming connection has valid reverse DNS (rDNS), giving the sending host's domain name as well as IP address. While there's no technical requirement that all sending hosts have rDNS, many people have noted that most hosts without rDNS are only spam. In mid-2003 AOL started rejecting all mail from hosts without rDNS, which impel the few legitimate senders without working rDNS to get theirs in order.

2.7. Content filtering

Once the SMTP server has decided to accept a message, the sender transfers the entire set of message headers and the message body. (For SMTP purposes, the message headers are just part of the message, and do not affect message delivery.) Many filtering schemes work on the header and body.

2.8. Hybrid filtering

While all of the filtering techniques above can be somewhat effective, a combination of many of them usually works better than any individual one.

Many spam filters can be applied as a series. Typically a mail server will use DNS based blacklists (DNSBLs) to reject some mail, then use body filters on the mail that makes it pass the DNSBLs.
Some other available filtering methods are: Collaborative Spam filtering, Statistical methods, Content-based filtering, Checksum-based filtering, Sender-supported whitelists and tags [22].

3. The Bayesian Method

Bayesian filtering is a statistical approach, which was used by many researchers to build a spam filter. Paul Graham made a significant contribution to this domain in implementing and testing one of the first Bayes spam. [7] Later on, Gary Robinson added some improvement to this filter. He produced a number of alternative approaches to combining and scoring word probabilities. The architecture of the spam Bayesian system has three different parts: Tokenizing, combining and scoring, and testing.

3.1. Mathematical explanation

Bayesian e-mail filters take advantage of Bayesian theorem. Bayesian theorem, in the context of spam, says that the probability that an e-mail is spam, given that it has certain words in it, is equal to the probability of finding those certain words in spam e-mail, times the probability that any e-mail is spam, divided by the probability of finding those words in any e-mail.

Interface of the Baysian uses numerical estimate of the degree of belief in a hypothesis before evidence has been observed. Baysian interface usually relies on degree of belief or subjective probabilities. Baysian theorem adjusts probabilities given new evidence in the following,

This theorem may be summarized as

$$\text{Posterior} = \frac{\text{likelihood x prior}}{\text{normalizing constant}}$$

In words; the posterior probability is proportional to the prior probability times the likelihood. In addition the ratio, Pr(E|Ho) Pr(B) is sometimes called the standardized likelihood, so the theorem may also be

$$P(H_O|E) = \frac{P(E|H_O)\ P(H_O)}{P(E)}$$

where,

- Ho represents a hypothesis, called a null hypothesis that was inferred before new evidence, E become available
- P(Ho) is called the prior probability of Ho
- P(E|Ho) is called the conditional probability of seeing the evidence E given that the hypothesis Ho is true. It is also called the likelihood function when it is expressed as a function of Ho given E
- P(E) is called the marginal probability of E
- P(Ho|E) is called the posterior probability of Ho given E

The factor P(E|H0)/P(E) represents the impact that the evidence has on the belief in the hypothesis. If it is likely that the evidence will be observed when the hypothesis under consideration is true, then this factor will be large. Multiplying the prior probability of the hypothesis by this factor would result in a large posterior probability of the hypothesis given the evidence. Under Bayesian inference, Bayesian theorem therefore measures how much new evidence should alter a belief in a hypothesis.

Multiplying the prior probability $P(H_o)$ by the factor $P(E|H_o)/P(E)$ will never yield a probability that is greater than 1. Since P(E) is at least as great as $P(E \cap H_O)$, which equals $P(E|H_o)$.$P(H_o)$, replacing P(E) with $P(E \cap H_O)$ in the factor $P(E|H_o)/P(E)$ will yield a posterior probability of 1. Therefore the posterior probability could yield a probability greater than 1 only if P(E) were less than $P(E \cap H_O)$, which is never true.

3.2. Alternative forms of Bayes' theorem

Bayes' theorem is often blown up by nothing that

$$P(E) = P(H_O, E) + P(H_O^c, E)$$
$$P(E) = P(E|H_O)P(H_O) + P(E|H_O^c)P(H_O^c)$$

where H_O^c is the complementary event of H_o (Often called "not H_o"). The theorem can be restarted as

$$P(H_O|E) = \frac{P(E|H_O)P(H_O)}{P(E|H_O)P(H_O) + P(E|H_O^c)P(H_O^c)}$$

The Bayes' theorem in terms of odds and likelihood ratio can be explained as follows:

Bayes' theorem can also be written neatly in terms of a likelihood ratio Λ and odds O as

$$O(H_O|E) = O(H_O) \cdot \Lambda(H_O|E)$$

where

$$O(H_O|E) = \frac{P(H_O|E)}{P(H_O^c|E)}$$

are the odds of H_O^c given E

$O(H_O) = \frac{P(H_O)}{P(H_O^c)}$ are the odds of H_O^c by itself and

$$\Lambda(H_O|E) = \frac{L(H_O|E)}{L(H_O^c|E)} = \frac{P(E|H_O)}{P(E|H_O^c)}$$

is the likelihood ratio.

3.3. Process

This section mainly explains the Baysian theorem used in junk mail filtering. According to the section 3.1 and 3.2 we can summarize the Bays' theorem in junk mail detection as follows

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\ \Pr(\text{spam})}{\Pr(\text{words})}$$

Using Bayesian analysis to classify spam and non-spam was suggested by Paul Graham. A Bayesian filter takes each word in a message and looks it up in a database to see how many times that word has appeared in prior spam and non-spam messages. The Bayesian formula then lets it combine those counts into an overall probability estimate to check whether the message is spam or not [7].

Particular words have particular probabilities of occurring in spam e-mail and in legitimate e-mail. For instance, most e-mail users will frequently encounter the word Viagra in spam e-mail, but will seldom see it in other e-mail. The filter does not know these probabilities in advance, and must first be trained so that it can build them up. To train the filter, the user must manually indicate whether a new e-mail is spam or not. For all words in each training e-mail, the filter will adjust the probabilities that each word will appear in spam or legitimate e-mail in its database. For instance, Bayesian spam filters will typically have learned a very high spam probability for the words "Viagra" and "refinance", but a very low spam probability for words seen only in legitimate e-mail, such as the names of friends and family members.

After training, the word probabilities (also known as likelihood functions) are used to compute the probability that an e-mail with a particular set of words in it belongs to which category. Each word in the e-mail contributes to the e-mail's spam probability. This contribution is called the posterior probability and is computed using Bayes' theorem. Then, the e-mail's spam probability is computed over all words in the e-mail, and if the total exceeds a certain threshold (say 95%), the filter will mark the e-mail as spam. E-mail marked as spam can then be automatically moved to a "Junk" e-mail folder, or even deleted outright.

4. Spam filtering mechanism of Baysian technology

There are several algorithms that use various modifications of Bayesian technique.

1. List every word in an incoming mail message
2. Determine the odds of each word appearing in a spam/garbage message, and
3. Use those odds as input to Bayes' Formula to determine if the message is garbage or not.

The first thing needed to do is to teach the Bayesian filter the difference between garbage and non-garbage messages. We can identify the spam or garbage e-mails from the content of e-mails. Most of spam e-mails contain certain key words. Instinctively, people know that a message containing these words or phrases is garbage/spam because of their experience in dealing with junk mail.

The Bayesian filter does not have the benefit of our years of experience, so we have to teach it what spam/garbage messages look like, and how they differ from non-garbage messages. The filter needs to introduce the garbage mail from the mail list. Whenever we show a message to the filter, it finds every word in the message and stores it (along with how many times it occurred) in a database.

Separate databases are kept for garbage and non-garbage mail messages. The filter uses a looser definition of a word than humans do – a

word (more properly called a token) can also be an IP address, a host name, an HTML tag, or a price (such as "Rs100"). However a token cannot be random strings, words less than three characters long, and numbers.
The filter scans through the message, creating a list of every word it knows about (in other words, every word in the message that's also in the token databases). In this example, the words it knows about are "prescription", "when", "today", "visit", and "your". Once the filter has the list of words it knows about, for each word it calculates the probability that the word appears in spam based on the frequency data in the token databases.

This probability value assigned to each word is commonly referred to as spamicity, and ranges from 0.0 to 1.0. A spamicity value greater than 0.5 means that a message containing a particular word is likely to be spam/garbage, while a spamicity value less than 0.5 indicates that a message containing that word is likely to be ham (non garbage/spam). A spamicity value of 0.5 is neutral, meaning that it has no effect on the decision as to whether a message is spam or not.

The current circumvention technique the spam/garbage mail senders (spammers) are trying is to include obviously non-spam/garbage words in their spam/garbage messages. Any spammer had two goals in sending messages,

- The first (and obvious) goal was to see if the spammer could sneak the message past the Bayesian filter by including obviously non-spam/garbage words. In this, he failed.

- The secondary goal was to try to get the filter to start recognizing the words "congresswoman" and "soybean" as words with a high spamicity. If spam/garbage mail senders (spammers) can get the filter to assign a high spamicity to an adequate number of words that commonly appear in non-spam/garbage messages, they can render the filter useless.

This circumvention method also has limited utility for another good old-fashioned marketing reason. If spam/garbage mail senders (spammers) start including a large pile of legitimate text (such as an article from the CNN website)[11] in each message, they would confuse their target audience. A spam/garbage message advertising penile growth supplements that also contains an article about the relative value of the Euro is going to confuse the audience so much they just ignore the message.

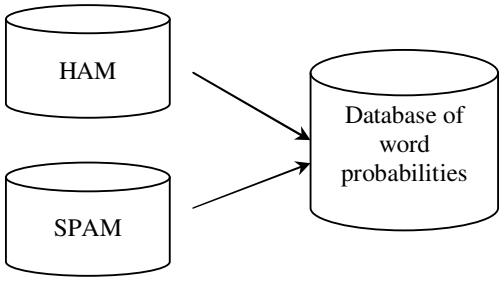

Figure 02: Creating a word database for the filter

Before mail can be filtered using this method, the user needs to generate a database with words and tokens collected from a sample of spam/garbage mail and valid mail (referred to as 'ham').

A probability value is then assigned to each word or token; the probability is based on calculations that take into account how often that word occurs in spam/garbage as opposed to legitimate mail (ham). This is done by analyzing the users' outbound mail and by analyzing known spam/garbage.

When we create the ham database the Baysian method does not require an initial learning period, it has 2 major flaws:

1. The ham data file is publicly available and can thus be hacked by professional spam/garbage mail senders (spammers) and therefore bypassed. If the ham data file is unique to your company/organization, then hacking the ham data file is useless. For example, there are hacks available to bypass the Microsoft Outlook 2003 or Exchange Server spam/garbage filter.
2. Such a ham data file is a general one, and thus not tailored to your company/organization, it cannot be as effective and you will suffer from noticeably higher false positives.

Besides ham mail, the Bayesian filter also relies on a spam/garbage data file. This spam/garbage data file must include a large sample of known spam/garbage and must be constantly updated with the latest spam/garbage by the anti-spam/garbage software. This will ensure that the Bayesian filter is aware of the latest spam/garbage tricks, resulting in a high spam/garbage detection rate

(note: this is achieved once the required initial two-week learning period is over).

When actual filtering is working, once the ham and spam/garbage databases have been created, the word probabilities can be calculated and the filter is ready for use. When a new mail arrives, it is broken down into words and the most relevant words – i.e., those that are most significant in identifying whether the mail is spam/garbage or not – are singled out. From these words, the Bayesian filter calculates the probability of the new message being spam/garbage or not. If the probability is greater than a threshold, say 0.9, then the message is classified as spam/garbage. This Bayesian approach to spam/garbage is highly effective – a May 2003 BBC article reported that spam/garbage detection rates of over 99.7% can be achieved with a very low number of false positives.

5. Applications in E-mail filtering with Baysian technology

The Bayesian technique is widely used in many technological areas. Image processing [12], Microscopic image analyzing, medical research [13], Detecting Speech Recognition Errors [14] are the mostly used areas in this technique. This review paper is trying to concern the Baysian technique in the junk mail filtering area.

5.1. SpamAssassin

SpamAssassin is a rules-based filter written in Perl. It was used for a while but spammers rapidly figured out how to get around each new rule. So it was becoming less and less effective [15]. In version 2.5 the developers added Bayesian learning to address that problem. Besides, since it is still in Perl, it is difficult to maintain and is slow.

5.2. Bogofilter

Bogofilter was one of the first Bayesian filters. Originally by über-hacker Eric S. Raymond, it's written in good old-fashioned C and runs fast [16]. If it has a weakness, it is being little too conservative about rating things as spam.

5.3. Quick Spam Filter (QSF)

QSF is a more recent Bayesian filter. It is also written in C and is even smaller than bogofilter. The scores it generates seem to skew somewhat higher than bogofilter's, to the point where it gives a lot of false positives [17].

5.4. Bayesian Mail Filter (BMF)

BMF is another option. It is very small - only 4600 lines of code, 110 KB. It is quite fast. In addition to SourceForge any person can find it in the FreeBSD ports tree [18].

5.5. iFile

ifile collects statistics on the occurrences of words in mail documents that have been filed/refiled, and uses that to determine a "best guess" of where new mail should best be filed. Some researches have done quite a lot of work to tune it to provide decent performance.

The idea is to collect a dictionary of statistics on the number of occurrences of words in messages filed to each folder. Incoming messages are compared to the dictionary, and are filed to the folder to which they have the highest degree of correspondence. When messages are refiled (due to being misclassified by the filter), the dictionary is revised [23]. Words that are not commonly used are eliminated, so that the dictionary does not get too large.

ifile uses naive Bayesian filtering as a statistical approach to direct messages to MH folders to which they have the highest degree of correspondence.

The "naive" assumption that is made is that correlations need only be done on the basis of individual word occurrences, that is, we count the number of times that the word "stop" is used, and do not consider combinations of words (e.g. "stop it" or "stop and go").

The weakness of this scheme is that while it is wonderful at "discrimination," that is, classifying dissimilar documents into different groups, it has nothing that makes it good at aggregating several kinds of dissimilar documents into a single group.

5.6. dbacl

A digramic Bayesian filter, is not restricted to just spam and non-spam. This mail filter will classify a message into one of many categories [19].

Another junk mail filters are also available like SPASTIC, SpamProbe. These all are based on the Bayesian Technique.

6. The need of Bayesian technique

There are some experiment results of Baysian filters and non-Baysian filters. Each program was installed according to its documentation. For the filters that required training, the training set data was supplied. Each filter was taken in turn and executed once for each e-mail in the spam and legitimate sets and the classification it gives was recorded.

The standard metrics for text classification are recall and precision. Spam classified as non-spam is known as a false negative. Non-spam classified as spam is known as a false positive. Precision is the percentage of messages that were classified as spam that actually are spam. High precision is essential to prevent the messages we want to read being classified as spam. A low precision indicates that there are many false negatives. Recall is the percentage of actual spam messages that were classified as spam messages. High recall is necessary in order to prevent our inbox filling with spam. A low recall indicates that there are many false positives. According to the experiment by using the several types of Bayesian mail filters [25].

According to the experiments Bayesian algorithm is compared with bag-valued features against the RIPPER rule-learning algorithm in different e-mail classification tasks. In learning a user's foldering preferences and learning to detect spam, the Bayesian filter substantially outperformed RIPPER in classification accuracy.

6.1. Advantages of Bayesian filter

The Bayesian method takes the whole message into account. It recognizes keywords that identify spam/garbage, but it also recognizes words that denote valid mail. Bayesian filtering is a much more intelligent approach because it examines all aspects of a message, as opposed to keyword checking that classifies a mail as spam/garbage on the basis of a single word.

A Bayesian filter is constantly self-adapting - By learning from new spam/garbage and new valid outbound mails, the Bayesian filter evolves and adapts to new spam/garbage techniques. For example, when spam/garbage mail senders (spammers) started using "f-r-e-e" instead of "free" they succeeded in evading keyword checking until "f-r-e-e" was also included in the keyword database. On the other hand, the Bayesian filter automatically notices such tactics; in fact if the word "f-r-e-e" is found, it is an even better spam/garbage indicator, since it is unlikely to occur in a ham mail.

The Bayesian technique is sensitive to the user - It learns the e-mail habits of the company/organization and understands that, for example, the word 'mortgage' might indicate spam/garbage if the company/organization running the filter is, say, a car dealership, whereas it would not indicate it as spam/garbage if the company/organization is a financial institution dealing with mortgages.
The Bayesian method is multi-lingual and international - A Bayesian anti-spam/garbage filter, being adaptive, can be used for any language required. Most keyword lists are available in English only and are therefore quite useless in non English-speaking regions.

A Bayesian filter is difficult to fool, as opposed to a keyword filter - An advanced spammer who wants to trick a Bayesian filter can either use fewer words that usually indicate spam/garbage (such as free, Viagra, etc), or more words that generally indicate valid mail (such as a valid contact name, etc).

7. Conclusion

The Bayesian filters, after training, offer better recall than the heuristic filters. Catching a higher proportion of spam is clearly good, since that is the reason people use them. With insufficient training, however, the Bayesian filters perform poorly in comparison with SpamAssassin in terms of recall. But some of Bayesian filters work very poorly (Quick Spam) compared with other Bayesian filters.

The Bayesian filter outperforms by far the keyword-based filter, even with very small training corpora.

As future work, we can plan to implement alternative anti-spam filters, based on other machine learning algorithms. Also the filter can include the foddering mechanism to check the spam if user need it. The server side mail filters are most effective, because many users can be protected from the filters. More than one filter can be activated in a server as a set of layers, and it is more effective than one filter

because if one filter fails another one can be successful.

8. References

[1]. L. Pelletier, J. Almhana, V. Choulakian, Adaptive Filtering of SPAM, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'2004)

[2]. Mingjun Lan, Wanlei Zhou, Spam Filtering based on Preference Ranking, Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05)

[3]. M Sahami, S. Dumais, D. Heckerman, and E. Horvitz, A Bayesian approach to filtering junk e-mail, AAAI'98 Workshop. Learning/or Text Categorization, Madison, WI, July 27, 1998.

[4]. W. W. Cohen, Learning rules that classify e-mail, In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access.

[5]. Spam statistics, http://bloodgate.com/spams/stats.html

[6]. Paul Graham, Better Bayesian filtering, January 2003,

[7]. http://paulgraham.com/better.html

[8]. Paul Graham, A Plan for Spam, August 2002, http://paulgraham.com/spam.html

[9]. The Coalition Against Unsolicted Commercial E-mail, www.cauce.org

[10]. X. Carreras and L. Andm, Boosting trees for anti-spam e-mail filtering, In Proceedings of RANLP-2001, 4th International Conference on Recent Advances in Natural Language Processing., 2001.

[11]. L. Pelletier, J. Almhana, and V. Choulakian, Adaptive Filtering of SPAM, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04), 2004.

[12]. CNN 2005, http://www.cnn.com

[13]. Nobuo Kumagai Masayoshi Aritsugi . On Applying an Image Processing Technique to Detecting Spam/garbages. Department of Computer Science, Faculty of Engineering, Gunma University, 1–5–1 Tenjin-cho, Kiryu 376-8515, Japan. 2005

[14]. S B Tan. Introduction to Bayesian Methods for Medical Research. Division of Clinical Trials and Epidemiological Sciences, National Cancer Centre. 2001

[15]. Lina Zhou1, Jinjuan Feng1, Andrew Sears1, Yongmei Shi2. Applying the Naïve Bayes Classifier to Assist Users in Detecting Speech Recognition Errors. 1Information Systems Department, UMBC, Baltimore, MD 21250, 2Computer Science and Electrical Engineering Department, UMBC, Baltimore, MD 21250. 2005

[16]. SpamAssassin, http://spamassassin.apache.org/

[17]. Bogofilter, http://bogofilter.sourceforge.net/

[18]. Quick Spam Filter (QSF), http://www.ivarch.com/programs/qsf/

[19]. Bayesian Mail Filter (BMF), http://sourceforge.net/projects/bmf/

[20]. dbacl, http://www.lbreyer.com/gpl.html

[21]. SPASTIC, http://spastic.sourceforge.net/index.html

[22]. SpamProbe, http://spamprobe.sourceforge.net/

[23]. http://en.wikipedia.org

[24]. Jason D. M. Rennie, ifile: An Application of Machine Learning to E-mail Filtering, Artificial Intelligence Lab, Massachusetts Institute of Technology

[25]. Spamfilters, http://freshmeat.net/articles/view/964/

[26]. I. Androutsopouios, J. Koutsias, K.V. Chandrinos, and D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Proceedings of the 23rd A CM SIGIR Annual Conference, pages 160-167, 2000.

[27]. Jefferson Provost, Naïve-Bayes vs. Rule-Learning in classification of e-mail, Deprtment of Computer sciences, The University of Texas at Austin

Distributed Artificial Neural Network Training Using an Intelligent Agent

G.R.C De Silva & Asoka S. Karunananda

Informatics Institute of Technology,
In collaboration with Manchester Metropolitan University, UK

Faculty of Information Technology, University of Moratuwa,
Sri Lanka

Abstract

Obtaining the best configuration for an Artificial Neural Network (ANN) has always been problematic due to the non availability of a uniform method of finding the best configuration. The trial-and-error approach is used even today for this purpose regardless of its limitations. This project exploits intelligent Agent technology and distributed computing techniques to automate and streamline the task of ANN training. A static, centralized Agent autonomously generates and trains multiple ANN's in varying configurations for a given dataset. By analyzing the progress of each training session the Agent discovers which ANN configuration is best suited for the training data. The distributed system on which the Agent operates allows multiple ANN's to train concurrently which helps identify the best configuration in a much shorter timeframe. Knowledge on past ANN trainings is gathered by the Agent from which it learns to identify most suitable ANN configurations for subsequent training sessions. The end result is a fully automated system that can identify the most suitable ANN configuration for a given dataset quickly, efficiently and autonomously.

1. Introduction

ANN's have become an attractive Artificial Intelligence (AI) technique for building modern intelligent applications. Image recognition, sound recognition, financial forecasting etc. are popular application areas for ANN's. However training a Neural Network to understand the problem correctly takes considerable manual effort because various configuration parameters of an ANN need to be assigned proper values (i.e. network topology, activation function, learning rate etc.). Since these values differ depending on the problem, it requires a certain degree of experimentation before identifying which values are most suitable for these parameters. This is currently done manually by the ANN developer in an ad-hoc manner. This, together with the most widely used single machine based ANN training method makes it a time consuming process to identify and train a suitable ANN configuration for the problem.

Several techniques have been devised to overcome these limitations in both identifying the proper network parameters and also speeding up the ANN training process. Use of Genetic Algorithms (GA) and Evolutionary Computing to devise network parameters instead of manually identifying them is explained in [17] [18] [19] [20]. Research has also been carried out on developing specialized hardware to train neural networks which is much faster than the software based approach that is widely used [6].

The above mentioned papers propose techniques for addressing the limitations of ANN training. However their main focus is based either on identifying the proper configuration values for the ANN more efficiently or improving the speed of ANN training. This paper focuses on identifying a technique that can address both these issues where the manual nature of ANN training can be reduced to a minimum at the same time reducing the overall time taken to obtain a suitable ANN configuration for any given problem.

1.1. ANN Training: Practices and Issues

ANN training can be categorized into two modes as supervised and unsupervised training. The supervised method is most widely used, where the network is given the inputs as well as the desired outputs for each input. The network tries to produce the desired output for each input during training, where the difference of the desired and actual output

which is also referred to as the 'error' is propagated backwards through the network which modifies its connection weights according to a learning algorithm. Back-propagation learning rule is mostly used for this purpose because it's a general purpose algorithm which is applicable for many real world problems such as image recognition [1]. However it has its limitations such as the tendency for network paralysis and the long periods taken for ANN training [2].

Unsupervised mode is the other alternative, where the network tries to learn patterns in the data on its own without any given desired outputs. Application areas such as data classification, data mining and data compression use unsupervised training. However this mode of training is still not as popular as supervised training. Kohonen self organizing maps are commonly used for this purpose [3].

Identifying the best ANN configuration for the training data is the most challenging aspect when training ANN's in supervised mode. There are many parameters that need consideration such as the learning rule, activation function, network topology, learning rate, momentum, bias etc. Proper values for most of these parameters cannot be determined in advance, because they depend heavily on the nature of the training data. Therefore it requires a certain degree of experimentation to identify which values are most suitable for each parameter. For example, identifying the number of layers and hidden neurons for the network can only be done in an ad-hoc manner by trying out different architectures and observing the training progress of each [2]. This is perhaps the most time consuming task when training ANN's. The ANN developer needs to create the ANN configuration manually by means of a development tool or by directly coding it in a programming language. The created ANN then requires training, where it should be continuously monitored by the developer to decide whether the network is training successfully or not. Mostly, the initial configuration tested will not suffice and alternate configurations needs to be tested which results in repeating the same procedure all over again. Apart from that, each configuration that is tested needs to be kept track of by the developer to avoid re-training any configuration that has already been tested.

Clearly, this approach is inefficient and there is too much manual work involved. Also, since only a single computer is used for training ANN's, each configuration needs to be trained sequentially. As a result, it takes a considerable amount of time to identify the configuration that is most suitable for the network.

The rest of this paper discusses the current approaches and solutions used by ANN developers to overcome the limitations identified above. The focus will then move onto describing how Agent Technology and Distributed Computing can be used to devise a novel solution to address these issues. The remainder of the paper will discuss how this solution has been implemented into a functional prototype followed by the evaluation results of the proposed solution which will confirm the applicability of using Agent Technology and Distributed Computing for ANN training.

2. Current approaches & solutions

Several approaches can be seen as techniques used by ANN developers to improve the task of ANN training. Hierarchical and Modular ANN's [3] are used to break down the training into smaller components and aggregate the final results into a single network. This allows you to obtain the final network quicker than training a very large and complex network which may take longer to train.

Evolutionary computing techniques are used to devise the optimum architecture for a network. Neuro Evolution of Augmenting Topologies (NEAT) is one such method which uses an evolutionary process of obtaining the best ANN configuration [4]. Genetic Algorithms are also used during ANN training to evolve various network parameters such as connection weights [5].

Hardware based approaches are used to improve the speed of ANN training. Digital and analog hardware chips such as NeuraLogix NLX-620 and Micro Devices MD-1220 are some examples [6]. Hardware based techniques are capable of training ANN's much faster than software based implementations.

Among the many software toolkits available for ANN development, NeuroSolutions [7], Statistica Neural Networks [8], Neural Network Toolbox for MATLAB [9], JOONE Distributed Training Environment [10] are popular choices. The graphical user interfaces provided by most of these solutions make ANN development easier and less time consuming than the manual coding method.

Although the approaches discussed contribute to ANN development, they do not fully address the problems related to ANN training. The first problem is that creating different ANN configurations requires manual effort. Although these solutions provide feature rich user interfaces for this purpose, the creation of ANN's is still a manual process that needs to be performed by the user.

The remaining problem is the time factor. Almost all solutions discussed above use a single computer for training ANN's in a sequential manner. This becomes a problem as the number of configurations that require training becomes very large.

Yet another issue is the inability to keep track of ANN configurations that have already been tested. Current approaches tend to re-start the development of ANN's every time a new dataset is trained. There is very little support for re-using the knowledge gained from previous training sessions in current approaches which could contribute immensely to identifying suitable ANN configurations much sooner.

One final issue is the cost factor. Most of the solutions mentioned above are commercial products which come at a cost, and it is difficult to obtain such software for the typical academic student.

3. Approach: Agent Technology & Distributed computing

3.1 Suitability of Agent Technology

Looking at the tasks involved in developing and training ANN's, the concepts of Agent technology were identified as very promising for this purpose. The autonomous, reactive, proactive behavior of Agents [11] together with learning capability blends in well to address the problems faced while training ANN's.

During ANN training, the developer needs to experiment with various ANN configurations by creating, training them and analyzing their outcome. The autonomous features of Agents are ideal for this type of task, where the Agent can perform these operations on its own without user intervention. During training, the reactive nature of Agents becomes useful where the training may need to be halted and re-started (i.e the network becomes paralyzed during training). The learning abilities of Agents can be used to store vast amounts of knowledge gathered during ANN training such as which configurations are best suited for given datasets by classification. This information can be re-used later to identify which configurations are most suited for a particular dataset given.

We have taken a single Agent based approach rather than a multi Agent one, due to the fact that a centrally located Agent can control the entire distributed system at the same time keeping track of the status on remote machines from a single location. Also the need to aggregate results of all trainings for evaluation purposes needs to be done centrally, therefore a single central Agent was chosen for this purpose. Since the system can be controlled from a single location, the Agent need not move across different machines. Therefore a static Agent has been identified as more suitable rather than a mobile Agent.

3.2 Suitability of a Distributed System

In order to speed up the process of identifying the best ANN configuration, we have made use of a distributed system. Conventionally, ANN training is carried out in a sequential manner using a single computer for each configuration tested. Performing this operation on a distributed system enables training of varying ANN configurations concurrently on different machines, thus reducing the overall time taken to identify the best configuration for the training data. The system is scalable, which means new machines can be added dynamically during operation thus increasing the number of ANN configurations that can be concurrently trained. This type of configuration is ideal for a Local Area Network, where machines on the network are mostly sitting idle. The processing power of these machines can be harnessed for the task of ANN training.

3.3 Inputs, process and outputs of the system

The inputs to the system are the initial ANN configuration which includes parameters for Emax, network topology, learning algorithm, learning rate, cycles to train and other learning algorithm related parameters along with the training dataset will be in the form of a text file.

The process of the system will be creating and training various ANN configurations for the

training data and identifying which configuration performs best out of the entire collection of configurations tested.

The output of the system will be a fully trained ANN created to the most suitable configuration which can be readily deployed in a Java application using Java object serialization.

3.4 Features of the system

A host of features will be incorporated onto the system to facilitate the successful training and monitoring of ANN's.

The system will be capable of developing, training and evaluating various ANN configurations autonomously.

The system will be scalable, where client machines can be connected to the centralized Agent server during runtime increasing the processing capabilities of the system.

ANN training on each machine will be remotely administered by the intelligent Agent, and their progress will be displayed on the central server in a graphical format.

Resources on the distributed system will be efficiently utilized. No clients will be sitting idle when there is pending ANN training jobs at the same time, no client will be overloaded with ANN training jobs.

The Agent will be customizable, where the user can change various parameters to alter the behavior of the Agent during operation.

4. Design

The system consists of 3 main modules. Based on the client/server architecture [12], the Agent runs on a centralized server and other machines connect to this server as clients to acquire ANN training jobs.
The concept of ANN training using Agent Technology and Distributed Computing is depicted in Figure 1. The main modules of the system are shown and the communication direction between these modules is depicted using the arrows.

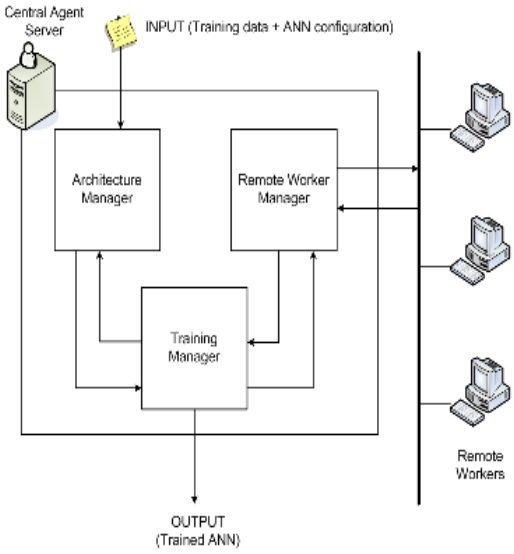

Figure 1: High-level System Design

4.1 Architecture Manager Module

An *ANN Architecture Manager Module* is developed to facilitate the creation of various ANN configurations. The parameters for the ANN such as number of hidden layers, neurons per hidden layer; activation function etc. can be configured by this component to create varying ANN configurations. These ANN's are created in memory as objects which can be transported across the network to remote machines for training.

4.2 Worker Manager Module

A *Worker Manager Module* is developed to manage the resources on the distributed system. A mechanism to keep track of which client machines are available to take ANN training jobs is required, which will be the main purpose of this module. Tasks such as accepting client registrations, de-registering clients from the server, keeping track of client details such as their hostname, IP address, machine type etc. will be maintained by this module.

4.3 Training Manager Module

Finally, a *Training Manager Module* manages all aspects of training the ANN. This is the Agent component. It sends ANN's to train on remote machines, monitors the training conditions, takes decisions based on the conditions and finally brings together the trained ANN's back to the central server. The module acts as the front end to the user and

integrates with the other 2 modules to create the final system.

5. Implementation

Bringing together the 3 main modules, the system has been implemented using Java as the core development language. Java qualifies as the language of choice for this system due to its platform independent nature which is ideal for the distributed system. Together with pure Java, several Java based open-source frameworks have also been used to develop the modules which are discussed below.

5.1 Architecture Manager Implementation

The *Architecture Manager* module is capable of creating ANN objects in varying configurations (i.e topology, activation function etc.) To facilitate this task, JOONE (Java Object Oriented Neural Engine) [10] which is an open-source framework written in Java has been used. It provides the base classes for creating ANN's through Java code. The customization components of the Architecture Manager module use this framework as the base for generating varying ANN configurations.

5.2 Worker Manager Implementation

The infrastructure required by the *Worker Manager* module for communication between the centralized Agent and remote PCs on the distributed system is implemented using Java RMI (Remote Method Invocation) technology. [13][14][15] Remote interfaces at both client and server ends enable communication, transportation and remote method invocation of ANN objects between server and clients. The object serialization capability in Java RMI contributes to passing ANN objects to remote machines and retrieving them as return values.

5.3 Training Manager Implementation

The *Training Manager* module has been developed using JADE (Java Agent Development Environment) [16] which is an open-source Agent platform written in Java. The various behaviors of the Agent are implemented using this framework, and it has been integrated with Worker Manager and Architecture Manager Modules to bring the system together. The Training Manager module also connects to a MS Access database, where it periodically queries and updates records of past ANN training information.

Using the tools and technologies mentioned above, a fully functional prototype has been developed to demonstrate the concept of distributed ANN training using Agent technology.

6. Evaluation

Following the successful implementation of the system, a formal evaluation was carried out to asses the outcome of the proposed system.

6.1 Evaluation criteria

The criteria for the evaluation were mainly the suitability of the approach taken by the authors to address the problems identified during ANN training. The accuracy of the proposed system, features it provides to the users, usability of the system and also the limitations of the system are among the evaluation criteria.

To evaluate the accuracy of ANN training of the prototype, 4 training datasets were used to test the system. 2 datasets were obtained from projects which involved image identification. 2 more datasets were obtained from projects that involved financial forecasting.

The system was tested on an operational Local Area Network (LAN) with a single machine running as the Agent server and 5 other machines on the LAN connected to the Agent server as worker machines to train the ANN's that are generated.

6.2 Evaluation audience and method

Several experts in the field of Artificial Neural Networks were chosen as expert evaluators. They are experienced lecturers on the subject of ANN's for many years at leading educational institutions in Sri Lanka both government and private. 3 subjects were chosen from the field of ANN as evaluators for this purpose. Apart from them, 2 subjects were chosen to evaluate the Distributed Component of the proposed solution. One of them has over 2 years industrial experience in Distributed Computing and the other over 8 years of experience in lecturing on Distributed Systems for undergraduate students.

Apart from the expert group, a selected group of 15 individuals were also selected as standard evaluators. These included both undergraduate and graduate students who have experienced developing and training Neural Networks during their final year research projects.

The evaluation process was carried out using formal interviews, questionnaires and a demonstration of the prototype.

6.3 Evaluation results

Using the training datasets explained above as inputs to the system, the system was tested on an operational LAN using 5 machines connected to the Agent server as workers to train the generated ANN's.

The ANN training results for the 1st dataset from the image recognition problem are given below. The dataset consists of 17 training patterns with 3 input elements and 3 desired output elements in each pattern. 10 different ANN configurations were created and trained using the conventional manual method and the proposed system. Each ANN configuration was trained for 10000 cycles. The dataset used for this training is depicted in Figure 2.

Using the conventional technique (manually creating and training ANN through code)

Total ANN training time (seconds)	1380
Lowest training error achieved	0.1372
Final ANN topology generated	3-8-4-3

Using the proposed system

Total ANN training time (seconds)	270
Lowest training error achieved	0.1301
Final ANN topology generated	3-7-4-3

```
0.02629959514;0.00851724137931035;0.0411666667;0;0;1
0.02354826254;0.00835483870967742;0.037;0;0;1
0.02254406130;0.0100384615384615;0.0435;0;0;1
0.02614574898;0.00851724137931035;0.0352857143;0;0;1
0.02036470588;0.00879310344827586;0.0425;0;0;1
0.02614574898;0.00851724137931035;0.0354257143;0;0;1
0.01963745502;0.00896428571428571;0.0352871429;0;0;1
0.02024014336;0.00962068965517241;0.06975;0;1;0
0.02468339768;0.00925;0.0863333333333333;0;1;0
0.02122180451;0.0102307692307692;0.0665;0;1;0
0.02434732824;0.0100769230769231;0.0655;0;1;0
0.01967669133;0.0095;0.0665;0;1;0
0.02155248387;0.00953846153846154;0.0826666667;0;1;0
0.03531666667;0.0141176470588235;0.024;1;0;0
0.03238618487;0.0132222222222222;0.0238;1;0;0
0.03685331746;0.01576;0.0315;1;0;0
0.0358;0.0153125;0.0222727272727273;1;0;0
```

Figure 2: Training Dataset

The significance when comparing the above results is the time period taken to train 10 different ANN configurations for the training data. As the results show, using an Agent to create the ANN's virtually eliminates the requirement for the user to create each ANN configuration. Because of that, generation of a new ANN configuration is almost instantaneous. This is in contrast to the manual method where it takes at least 20 seconds for the user to modify the previous configuration and create a newer ANN configuration to start training. This also needs to be done keeping in mind the configurations that were tested earlier to avoid re-training a previous configuration. Such tasks are all handled autonomously by the Agent.

The use of a Distributed System has clearly reduced the total training time for all ANN's drastically. As the results show, 5 different ANN configurations could be concurrently trained. This number can be dynamically increased by adding more machines as workers to the system at any point during operation to increase the capacity of training more ANN configurations concurrently.

Apart from the speed in obtaining the final ANN configuration and the autonomy of the system, the graphical reporting capability of the system should also be pointed out where the training error graph for each ANN configuration can be easily viewed using the user interface. Besides from this, the user has the option of saving a chosen number of ANN configurations that produces the lowest training error. Most of these facilities are not available in the conventional training method used.

The data classification ability of the system

should also be noted. The final configuration for the ANN was saved by the Agent to the database. When the system was introduced with the same dataset as a new training session, it was capable of classifying the dataset given using the Kohonen Self Organizing Map and identifying the previously successful ANN configuration of 3-7-4-3 as the first configuration to be trained. This sort of functionality is not available in the conventional ANN training methods discussed.

The overall results of the evaluation revealed that the system scored an eighty percent rating as 'Good', and around a twelve percent rating of 'Excellent' by the evaluators. The remainder fell under the 'Average' rating. The system was mostly appreciated due to the simplicity and ease of use. The scalability of the system and capability to run the client application on any platform was also seen very attractive by the evaluators. The graphical reporting of training progress was also very much appreciated.

Although the evaluators did not find any major drawbacks of the system, some useful additions to the system were suggested which included dynamic worker discovering capability and unsupervised ANN training support which are features that are not currently available in the proposed system.

From the authors' point of view, the system was originally developed to automate and make more efficient the task of ANN training. We feel that this objective has been met where the system is capable of identifying a suitable ANN configuration on its own without the user's intervention.

However it should be pointed out that the solution provided by the system may not be the 'optimum' solution, because there is no method of measuring if a given ANN configuration is the best configuration for the training data. However, considering the vast numbers of ANN configurations that the system is capable of testing, there is much greater capacity to identify a suitable ANN configuration for the training data because testing such large numbers of ANN configurations is not practical in the conventional manual approach of ANN training.

7. Discussion & Conclusion

We have made use of Agent technology and a distributed system as an approach to improve and streamline the process of ANN training. The various features of Agents and how they are incorporated into the ANN training process were discussed throughout this paper. We also pointed out how a distributed system could improve the time taken to identify a suitable ANN configuration. The design and implementation of the proposed system was later discussed pointing out the different modules of the system and the technologies used by each module.

Based on the capabilities of the system we believe that this is a good solution to train ANN's autonomously with minimum user involvement. The main strength of this system lies in its ability to generate ANN's, train and monitor their training and also make decisions during the training period all without the user's intervention. The amount of manual work that is saved by using this system is therefore very high. Also, the distributed system helps identify a suitable ANN configuration much faster than the sequential training methods discussed. As new machines can be dynamically connected to the Agent server to train ANN's, the overall time taken to obtain the final result can be further reduced. The data classification abilities of the system and learning capability allow it to intelligently identify possible ANN configurations that may be best suited for the given dataset. The final output of the system can be readily used as a production Neural Network in any Java application together with the JOONE framework.

Overall we feel that the implemented system is capable of identifying a suitable ANN configuration autonomously in a fast and efficient manner. We also feel that the system has potential to make the task of ANN training much more user friendly than the traditional methods discussed and could be used as a helper tool to aid ANN development.

8. References

[1] Artificial Neural Networks Technology (1992). [Online]. Dave Anderson and George McNeil. http://www.dacs.dtic.mil/techs/neural/neural.title.html

[2] AI-FAQ (2002). [Online]. Warren S. Sarle. ftp://ftp.sas.com/pub/neural/FAQ3.html

[3] generation5 (2004). [Online]. Wan Hussain and Wan Ishak. http://www.generation5.org/articles.asp?Action=List&Topic=Neural%20Networks

[4] Evolutionary Neural Networks: Design Methodologies (2001-2003). [Online]. Rinku Dewri. http://ai-depot.com/Articles/47/EANN.html

[5] Using Genetic Algorithms with Neural Networks (2000). [Online]. James Matthews. http://www.generation5.org/content/2000/nn_ga.asp

[6] REVIEW OF HARDWARE NEURAL NETWORKS: A USER'S PERSPECTIVE (1998). [Online]. Clark S. Lindsey and Thomas Lindblad. http://www.particle.kth.se/~lindsey/elba2html/ba2html.html

[7] NeuroSolutions (2005). [Online]. NeuroDimension, Inc. http://www.neurosolutions.com/

[8] Statistica (2005). [Online]. StatSoft, Inc. http://www.statsoft.com/products/stat_nn.html

[9] The MathWorks: Accelerating the pace of engineering and science (2006). [Online]. The MathWorks, Inc. http://www.mathworks.com/products/neuralnet

[10] Java Object Oriented Neural Engine (2006). [Online]. Paula Marrone. http://www.jooneworld.com/docs/dte.html

[11] Software Agent (2006). In Wikipedia: The free encyclopedia. http://en.wikipedia.org/wiki/Software_agent

[12] Andrew S. Tanenbaum and Maarten van Steen (2002) Distributed Systems: Principles and Paradigm, USA: Prentice-Hall.

[13] Getting started using RMI (1999) [Online]. Sun Microsystems, Inc. http://java.sun.com/j2se/1.4.2/docs/guide/rmi/getstart.doc.html

[14] Kenneth Baclawski (1998) Java RMI Tutorial [Online]. http://www.ccs.neu.edu/home/kenb/com3337/rmi_tut.html

[15] Ann Wollrath and Jim Waldo (2005) Trail:RMI[Online]. http://java.sun.com/docs/books/tutorial/rmi/index.html

[16] Jean Vaucher and Ambroise Ncho (2003). JADE primer and tutorial. [Online]. http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html

[17] Zvi Boger and René Weber. Finding an Optimal Artificial Neural Network Topology in Real-Life Modeling - Two Approaches

[18] Bernhard ¨Omer (1995). Genetic Algorithms for Neural Network Training on Transputers

[19] *Michael Watts and Nikola Kasabov.* GENETIC ALGORITHMS FOR THE DESIGN OF FUZZY NEURAL NETWORKS

[20] Kenneth O. Stanley and Risto Miikkulainen. Efficient Evolution of Neural Network Topologies

Virtual Tour Agent

Sevalaraj Rupesh & Asoka. S. Karunananda

Informatics Institute of Technology, 57, Ramakrishna Road,
Colombo – 04
Faculty of Information Technology, University of Moratuwa,
Sri Lanka.

E-mail: s_rupesh5@yahoo.com, asoka@itfac.mrt.ac.lk

Abstract

Tourists normally have to rely on common tour packages that are designed and prepared by travel agents or they have to prepare their own tour itinerary which may or may not fulfill their expectations. The authors propose that Software Agents technology and other Artificial Intelligence Techniques like Expert System could be utilized to prepare a custom based solution to cater to personalized requirements. The Software Agent Technology is used to filter data from the net; Expert System is used to generate the travel itinerary; and Nearest Neighbor Algorithm is used to find the optimal pathway among locations. It also provides facilities to regenerate a travel itinerary in different circumstances. Apart from that, the travelers are also provided with travel related information plus travel support functions to provide a compressive solution.

1. Introduction

Sri Lanka as a small tropical island relies heavily on tourism as one of the major drivers of its economy. In this context, it is extremely important that the tourists who visit Sri Lanka carry with them a life long experience, which in turn would increase the influx of tourists to the country. Therefore providing a travel plan to fulfill the likes and dislikes of each individual is a key issue that needs to be addressed. Today the potential traveler has access to an unprecedented wealth of travel related information available on the internet. Planning actual trips using only internet-based resources requires a lot of work and sometimes turns to be a

rather difficult task. Therefore, a traveler must devote a lot of effort sifting through many sites of varying quality and data simply to form a coherent picture of his/her intended destinations, choices of hotels restaurants, possible means of transportation, etc. The total amount of data is so large that it is impossible to find all pertinent and important information in a reasonable time. This is vital since the traveler should be aware where he/she is going and what he/she would be doing and such matters would totally depend on the unbiased and true information that he/she has collected.

Travel planning is in itself a complex decision making process that requires undivided attention to detail. Travelers often encounter problems, as they tend to overlook vital matters. Hence, it results in wastage of resources such as money and time. The complexity of travel planning lies with the fact that it must satisfy as many factors as possible. The success of a travel plan greatly relies on its ability to achieve a satisfactory balance among these factors.

The travelers find it difficult to plan the travel itinerary even when the destination is finalized. Most travelers take a long time even to decide on the easiest way of traveling which are the tour packages (also known as all-inclusive packages) because they hardly find a package, which matches most of their requirements. Generating a travel itinerary is one of the most cumbersome and complicated process in the travel and tour domain because of the travelers individual preferences. Common travel packages do not fulfill all the preferences of the traveler, which depend on factors like background characteristic such as age, income, living/social status, location etc. In short, common travel packages cater a general context rather than the individual needs and preferences of the traveler. Apart from this, the burden of searching for information on items listed on the travel plan falls on the traveler.

The main factors which have a direct impact on the unsuitability of common travel packages are; the cost, the need to search for information, unchangeable destinations in travel packages (mainly Pre-Made tour packages), likes and dislikes of both the traveler and travel agent, knowledge level of the person who designs the

travel packages and changes to the current travel plan (Tailor Made tour packages) takes a long time to process. However, the biggest issue with the Tailor-Made travel packaging is that it requires the traveler to have a good knowledge about the arrangement and facilities available for him/her to initially make the selections. In addition, the traveler himself/herself is responsible for becoming aware of changes to his/her travel itinerary on the basis of the current situation pertaining to the tour. This is rather impractical for a busy person.

The aim of the project is to develop and evaluate a system, which will generate and design a travel itinerary using Software Agents Technology and Artificial Intelligence Techniques, for a traveler who is planning to visit Sri Lanka for a holiday.

Travelers have various types of ways to purchase a travel package or a travel itinerary. In the current context, several methods have been identified as ways to generate a travel itinerary. Among these, Manual Planning, Pre-Made Tour Packages and Tailor Made Tour Packages are popular as the common way of traveling. Manual Travel Planning is the most traditional and oldest way of travel planning. In Manual Planning the travelers have to go to the travel agent and select the destination, hotels, etc manually. This is a very long process since both travel agent and the traveler have to come to a conclusion on what the traveler is getting as a travel plan and how the travel agent is going to provide and arrange them.

In contrast, Pre-Made tour packages are set out before hand and travelers can buy those off-the-shelves from travel agents. They are mainly popular among the "Last-Minute Travelers" who have a busy scheduled life and less time for planning any leisure traveling activities. They are mainly sold and brought through the means of the internet. This is comparatively costly than Manual Travel Planning and always targets the upper middle class.

The Tailor Made tour packages are prepared by the travel agents in consultation with the traveler in terms of his preference and requirements. Here the traveler would be providing all the necessary and vital information for travel planning like destination, hotel and all the requirements & facilities that they need etc. The overall role of the travel agent would be to arrange the necessary ground facilities, and make sure what the traveler asked for is provided without any delay or mistakes. This is a product which mainly targets only the high end travelers.

Both Pre-Made and Tailor Made tour packages are more comprehensive and cover a range of requirements including flight reservation, traveling within the destination, hotel reservation, etc. However, the biggest issues with the tour packages are that it requires the traveler to have a good knowledge about the arrangement and facilities available for him/her to initially make the selections.

There are several IT and IS solutions made available targeting both the travel and tour industry. In this sense, the software systems such as eBusinessoft [3], Tour Plan [4] and XYKA [5] etc, have been commonly used in the travel and tour industry. Apart from the above, mentioned there are also Global Distribution Systems (GDS) like SABRE, Galileo, and World Span etc available. The main end users of these systems are hotels, travel agents, tour operators, airlines etc. The main facilities that the software solution provides for travelers are, Online Payments Gateways and to publish a travel itinerary (which is done manually) through the web pages if a web site is available through a dedicated server etc. In the case of GDS it's better to a certain extent as there are general public versions like websites (i.e. easySABRE, Travelocity.com) which are available. But the GDS still operates through agreements, dedicated chain of networks and dedicated means of communication channels which are only open to them, in which overall makes the cost factor increasingly very high.

The above mentioned software solutions and GDS still do not provide a compressive solution for travel planning. But overall the support or features given by both kinds of systems are really low or sometimes non existent. However, it is evident that there are so many powerful software technologies available for supporting the activities in the field of travel planning. Among others the Software Agent Technology is a good candidate for implementing an effective IT solution for travel planning. In fact, the idea of the Agent technology has been inspired by how an experienced travel agent assists a traveler with a little information provided by the traveler to the Agent.

This paper presents our Agent based approach (called Virtual Tour Agent – VTA) to travel planning. The VTA has been designed as a static Agent that runs on a travel planning web sever. VTA is also powered by an Expert Systems (ES) and Nearest Neighbor Algorithms (NNA). VTA is more comprehensive and covers a range of requirements including flight reservation, transportation in the destination, hotel reservation and provides travel support information etc.

2. Design

The design of VTA consists of few major components as shown in Figure 1. The Web Component is the main component that the users directly interact with VTA. The core component of the VTA is the Agent Component. The Agent Component is used in filtering the necessary information from the eXtensible Markup Language (XML) which is hosted in the WWW, and the filtered information would be sent to the database of VTA.

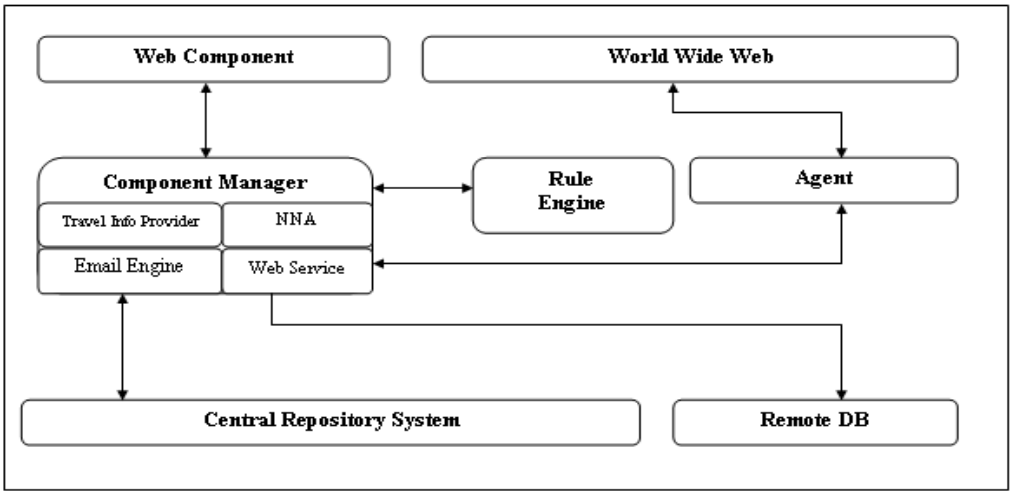

Figure1: The Architecture of the Virtual Tour

The other major component in the VTA is the Component Manager. The Component Manager consists of other minor components like Travel Info Provider, NNA, Email Engine and Web Service. The reason behind Component Manager Module in VTA is to achieve divide and conquer approach. This approach enables the advantages to the VTA service providers where they can come up with their own modules and function which could be pluggable to the current design. The Component Manager gets the information posted by the user such as hotel criteria and user preference. Component manager passes the list of hotels, districts present in the back end with the user preference. Considering the user preferences and the back end information Rule Engine executes the rules and generate the possible cities that can be included in his/her proposed travel itineraries and pass the itineraries to the component manager. The NNA is used to get the best and optimal pathway with the results generated by the Rule Engine. The Web Service Component is used to make reservation to the remote database which is the hotel database. The Email Engine is used to get confirmation to the users personnel email account when a hotel reservation has been made. The email is generated from the particular hotel when a reservation is made using their database which is the remote database.

The other main component of VTA is the Expert System or the Rule Engine. The Rule Engine is used to generate the travel itinerary for the travelers according to their categories, preferences, amenities required, and the price.

It should be noted that all these above mentioned criteria plus sub categories make the total rules of the Expert System. The ES was mainly used to achieve the status of the expert knowledge in travel and tour industry to prepare the travel itinerary. The other main reasons as to why the ES was used: because the problem is well structured, not very much common sense is required, is not based on traditional approaches, and finally the problem domain is well scoped and sized. The Rule Engine works in the Feed Forward Chaining mechanism and the ES is made out of a Rule Engine called Java Expert System Shell (JESS).

The Repository Module is another module which could be divided into major areas such as Central Repository and Remote Database. The Central Repository consists of data of user/travelers personal information, preference information which he/she provides to generate the travel itinerary. The remote database is considered as the hotel database and it is being used for making reservation to the hotels.

3. Implementation of VTA

The Virtual Travel Agent (VTA) has been designed to operate from a web server, which is

accessible by the travelers from anywhere in the world. The VTA is fundamentally designed as an Expert System [1] that uses Nearest Neighbor Algorithm [2] for creating of optimal schedules to meet with dynamically changing requirements. The VTA also provides local news events since most of the travelers do not come across local news media. It also provides travel related information (i.e. weather, transportation, currency rates etc) and travel support functions (reservation and confirmation mail) to travelers as an overall travel planning solution.

The VTA filters the information from the web pages XMLs and stores the data in the database. This would reduce the workload of maintaining the database as the updating is done automatically and periodically. The ES in the VTA will generate the travel itinerary according to the user's preferences and the NNA will generate the optimal pathway between the destinations chosen by the ES earlier. The agent can also filter useful travel information like weather, transportation, exchange rates for travelers whenever required.

The VTA is developed in Java to take advantage of its platform independence, multi threading, robustness, simplicity and other features. The VTA would be filtering travel related information using XML format from the remote web sites and will store data accordingly. The whole project is built in the J2EE - three tier architecture. The JSPs are used to construct the client tier, the EJBs the middle tier and MySQL, the database. The client tier (presentation layer) is done in Model View Controller (MVC) architecture using Struts Framework. The Java Architecture for XML Binding (JAXB), which is a new technology, is used to serialize the XML into an Object with the aid of the JAXB Complier. Java Expert System Shell (JESS) has been used to develop the Expert System to prepare the travel itinerary and Java has been used to develop NNA to generate the optimal pathway.

4. VTA - How IT operates

The user is presented with a web page, where he/she enters how many days he/she will be staying. Then he/she would be selecting his/her preferences, hotel type (star class), amenities, and price range of the hotel. The VTA agent will process the selections and will filter the necessary information from the WWW and store them in the database. The VTA Agent is capable of keeping track of the modifications to remote web pages and will update the database accordingly. The ES in VTA will work on the filtered data and generate the travel itinerary according to the traveler's requirement. The generated itinerary would be worked through NNA to find the optimal pathway between the destinations. The itinerary would then be displayed in the web page of the VTA.

The VTA is capable of catering to a situation where the traveler requires modifications to the itinerary. The user can regenerate the travel itinerary, taking into consideration all the preferences that he/she entered, and taking into account the remaining number of days and the current location. The updated travel itinerary approved by the user would be stored in the database.

If a user logs on the VTA with different start and end dates, and give instructions to generate a new travel itinerary for his second visit, the agent would consider the previous itinerary whilst preparing the new one in order to provide a more personalized itinerary. The agent will learn the particular user's traveling style and adopt accordingly. In any required situation, the agent itself can suggest a new travel itinerary to the traveler by keeping track of destinations and events that he/she missed on his/her previous visit. VTA agent will obtain travel related information (i.e. currency updates etc) as and when required. When a travel itinerary is finalized the traveler can make a reservation in a hotel site using the travel support functions. The personnel details provided by the user can be used to make reservations in the remote database (hotel) through web services and the confirmation details can be emailed to the traveler's personal email account. The travelers are also provided with facilities to alter booking information before it is sent for reservation. VTA system comes along with a state-of-the-art security system which is done using Java Authorization and Authentication Services (JASS). The VTA also provides other travel support function which would enable better travel planning according to situations. VTA has several types of logins which provides different levels of permission rights. This is useful when it comes to maintenance purposes when the system is running live and allows the system do be maintained in the distributed architecture.

5. Evaluation

The main objective of the evaluation was to make sure the prototype is evaluated from the point of effectiveness, efficiency and the proportion by which it meets the requirements of the user. The main evaluation strategies were to

distribute questionnaires, to do live demonstrations and to have interactive interviews. The major areas to be evaluated were problem domain, approach to solving the problem and the prototype itself.

The problem domain was reviewed in order to evaluate the solution provided. The evaluation process was performed under the following criteria: the problem worth solving, current practices and methods in travel planning, accuracy of the travel itinerary generated and amount of support provided for the other travel related activities.

The proposed approach was evaluated independently in order to determine the appropriateness of the methodology proposed for the problem addressed. The technologies used in the system for travel planning and its other related functions were also considered in the evaluation process. The approaches to the problems were evaluated according to the following criteria: applicability of Intelligent Agent to collect information, the use of Expert System technology to generate the travel itinerary and Nearest Neighbor Algorithm to find the optimal pathway.

The evaluation of the prototype is crucial in order to determine the success rate of the core project idea which was designed and implemented with other support functionalities. The prototype was evaluated on the basis of the following categories: accuracy of the travel itinerary created, the support functions provided for travel planning, the editable functionalities for travel planning, user interfaces, user friendliness, hotel reservation and email component, and general opinions and further comments.

6. Conclusion

This paper presented the design and implementation of Virtual Travel Agent as a solution for effective travel planning. We conclude that the Agent-based approach has so many advantages in the travel planning domain due to the following reasons. Firstly, due to its autonomous [6] feature, VTA can operate with very little information from the traveler. Second, the Agent can work proactively, to provide various services related to the destination. Third, the Agent can be reactive on the request of the traveler. Fourth, VTA can maintain a traveler profile and keep contact with the traveler to suggest to the traveler when possible offers are available. Fifth, since VTA is designed to operate on the travel agent web server, this approach does not give a burden on individual travelers to run and install VTA on their machines. From an implementation viewpoint, since the VTA is a static Agent, the security of the Agent is not a serious issue.

This paper is presented for a fully conceptualized VTA system with its design, evaluation and completion. During the evaluation the VTA showed more than 80% success rate which was done with people who are attached to the travel and tour industry, software engineers, normal users and the real travelers. We have presented the use of agent technology for the travel planning domain. We explained the design, implementation and evaluation of the system. We have achieved the efficiency and advantage of the agent system for the approach over the standard learning of travel planning. We conclude that this approach will have many advantages for the planning of a travel itinerary.

7. References

[1] Avelino J Gonzalz and Douglas D Dankel, "TheEngineering of Knowledge-based Systems Theory and Practice" New Jersey, Prentise Hall, 1993

[2] http://theorsociety.com/about/topic/news/article_news_tspjune.htm

[3] http://www.ebusinessoft.com/ebsTA.htm

[4] http://www.tca.co.nz/

[5] http://www.xyka.com/travel.html

[6] H. S. Nwana, "Software Agents: An Overview", Knowledge Engineering Review, British Telecommunications, 1996

Artificial Intelligence Approach to Effective Career Guidance

Chathra Hendahewa, Maheshika Dissanayake, Savindhi Samaraweera, Narmada Wijayawickrama, Anusha Ruwanpathirana, Asoka S. Karunananda

Faculty of Information Technology,
University of Moratuwa
Sri Lanka

Email: {see_chathra, mahe_23diss, sssamaraweera}@yahoo.com
{narmada_w, rppathirana}@yahoo.com, asoka@itfac.mrt.ac.lk

Abstract

With the vast opportunities available for IT education, career guidance has become a crucial theme for deciding the appropriate career path of students. The lack of such expert career guidance systems in this domain inspired us to come up with the initial idea. This paper reports on a Career Advisory Expert System, named iAdvice to guide students engaged in their higher education to determine their career paths and to select their course subjects to be inline with their career goals. Expert System features such as reasoning ability, providing explanations providing alternative solutions, providing uncertainty and probability measures, questioning ability are found in iAdvice. The design of Career Advisory Expert System takes into consideration factors such as past examination performance, student preferences and skills, industry alignment with subjects, which are the main factors also considered by a human expert in providing career guidance. According to the evaluation carried out it was found out that the model has a capability of about 70% accuracy in predicting the performance, about 85% relevance of the advice provided, the self-explanatory nature of the system and about 87% provision of informative and useful advice.

1. Introduction

Career guidance is a domain which involves human experts such as councilors who provide advice to select appropriate career paths for individuals after considering the personnel information and available options. This facility is vital to any higher education institute, because students tend to consult experts in determining their career paths. This activity helps out when students are not confident of their future career and when having problems in determining what is best for them out of the available options.

Further, the universities and other higher education institutes generally provide the flexibility to the students of choosing the subjects of their liking by providing elective subjects in their curriculum.

However, if students are not properly guided or given advice in aligning their decisions with their ultimate career goals, they will not be able to operate in their full potential in the future. More seriously, one may select an inappropriate career path and develop unrest and dissatisfaction in his/her own life, also being a cause for graduate unemployment in the country. Thereby, the objective of higher education will not produce its intended benefits. According to Labor force statistics of Sri Lanka, the unemployment rate of the labor force with the education level higher than G.C.E. Advanced Level in year 2005 amounts to 13.8% indicating its significance. This has pointed out the need for career guidance in higher education [3].

Being one of the oldest Artificial Intelligence techniques, expert systems have the ability to implement the problem solving ability of human experts based on exposure to knowledge in a particular domain. Expert Systems have been proven successful over the last fifty years in areas such as medical diagnosis, fault-diagnosis, interpreting measurement data, configuring systems to mention a few [2]. The major reasons for using expert systems are improving productivity of human experts, economic measures and disparate locations of human experts, applying expertise uniformly, impartially towards preserving experts' knowledge and saving time of human experts.

University Grants Commission of Sri Lanka has promoted career guidance services in universities since 1998 [9]. Yet only a few would actually take advantage of such facilities mainly due to limited availability, anonymity and time constraints. However, it is known that no Sri Lankan institutions have launched a proper computer-based career guidance system [4]. We argue that computer-based

career guidance system with expert system features would be a productive method to be applied in this domain to assist the human career guidance councilors as well as to provide effective advice to students. This paper reports on our Career Advisory expert system, *iAdvice*, as a means for providing effective guidance in the selection of appropriate career paths and elective subjects.

This expert system will pave way for the undergraduates to get advice on their career path that they wish to take. For the students who do not have a clear vision about a career path, *iAdvice* will provide the possible paths that one can take based on their past track records and their preferences. It will also provide alternatives and options students can take in order to mitigate the risk that could prevail in some paths although the student opts for certain subjects. It also expresses certainty levels for each career path proposed for the student and recommends what is best for the student. *iAdvice* provides the users with a clear explanation about the advice provided by means of reports as well as performance, skill and capability evaluation diagrams. *iAdvice* also advices on the subject areas a student should improve if the performance is not up to the required standard. The expert system also has the ability to ask specific questions from the user to determine the user skills and preferences, providing a rich interaction capability along with working on incomplete information.

The rest of the paper is organized as follows. Section 2 describes our approach to the development of *iAdvice*. Section 3 is on the design aspects considered in the development of *iAdvice*. Section 4 reports on the implementation of this system incorporating the expert system features and the knowledgebase. Section 5 describes the actual process that the users have to perform to get the advice from *iAdvice* and also illustrates how it is practically being used as a career advisory system. The evaluation process carried out on this research, the evaluation results and their explanations are described in Section 6. Section 7 concludes the research findings of *iAdvice* and its applicability in the real career guidance process in the higher education system.

2. Approach

The scope of the research was mainly focused at the Faculty of Information Technology of University of Moratuwa. This faculty offers a Bachelor's Degree Course in Information Technology as B.Sc. in Information Technology and B.Sc. in Information Technology (Honors). The undergraduate students who are both eligible and willing to do the specialized degree have the option of selecting from a range of optional subjects offered. In this scenario, the students have to select the subject preferences that they would like to take up based on their future career goals or their preferences. In order to cater to this problem, we came up with *iAdvice* which can provide advice to prospective Level 4 students in determining their subject selection and to provide advice in improving their performance.

iAdvice has been implemented as an expert system, that behaves like a human expert usually in a narrow domain of application, in this case being the career guidance domain.[2] In general, an Expert System is a program that represents and reasons with knowledge of some specialist subject with a view of solving problems or giving advice. The expert system will act as an intermediary between the problem space and the decision maker playing the role of an assistant to the decision maker [5][8]. In this case, the problem is lack of proper career guidance to students where ultimate decision should actually be made by the particular student. *iAdvice* has been implemented as an expert system playing the role of a useful and supporting intermediary between the problem and decision maker by providing career related advice to the students by analyzing a set of factors that affect the future career path determination.

Identifying targeted user groups was one of the important aspects during this research. The sample for our survey comprised of around 100 students who are currently undertaking the B.Sc. in Information Technology Degree Course at the Faculty of Information Technology. Preparing the questionnaires to collect data from specific target groups was the first action that was performed during this research. The intention of the questionnaire was to identify the subject preferences of the students with the career goals that they have in mind, so we could find the correlation with the subjects and the career paths. This was used to capture the requirements for the building of the knowledgebase. Questionnaires were prepared in two different sets to cater to both the old and new revised syllabuses.

From the answers that were given in the questionnaires, the trends that were available in the subject selection of students with regard to their career paths and their past performance were analyzed in detail. This was useful in developing the "Question Bank" with regard to skills and performance in the expert system and for the development of the business logic behind the system. Figure 1 illustrates important factors that were taken into consideration in the approach to the advisory system. Identification of the bylaws applicable in the university and referencing of career paths of some of the leading software companies and their required qualifications and skill levels have been considered.

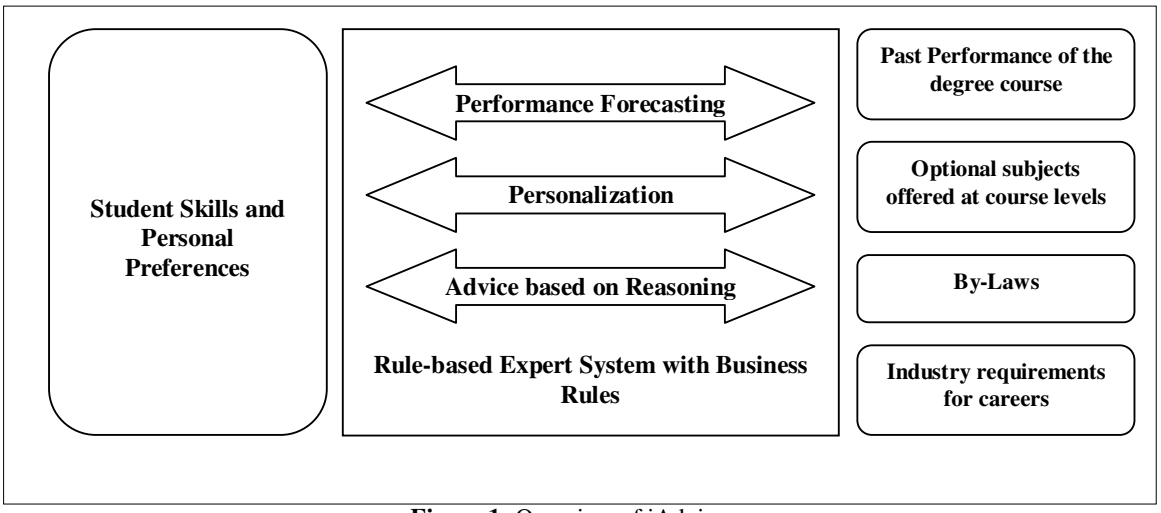

Figure 1: Overview of iAdvice

An expert system should mainly consist of three main modules as a knowledgebase, an inference engine and a user interface [2]. To facilitate this requirement, a literature survey was then carried out to find the expert system features and how expert systems can be implemented. Although there were different approaches to build expert systems in terms of knowledge representation, building the inference engine and the user interfaces, we decided to use the method of rule-based expert system methodology known as *production system* for this purpose. Further, reasoning types such as case based reasoning and rule based reasoning were considered. Most case based reasoning systems make use of general domain knowledge in addition to knowledge represented by cases [1].It was found that, rules were a popular approach to represent knowledge which can be easily expressed using logic where the reasoning process becomes less complicated; secondly easier in handling uncertainty; thirdly ability to incorporate explanations and to find alternative solutions. Rule-based expert systems necessarily operate in forward chaining and backward chaining methods. Then we also considered the possibility to build the expert system from scratch or to use available tools to make the process more productive. In this process, it was found that an existing expert system shell would be more applicable with the support to integrate with other applications and using specific knowledge representation languages to build the knowledgebase. Finally, we used forward chaining, backward chaining, and rule-based inference in designing this expert system with the use of an expert system shell which is more feasible with the time and other constraints and being more applicable to the domain.

3. Design of *iAdvice*

iAdvice has been designed in a modularized manner in order to maintain greater efficiency and maintenance. The modularized components ensure that any change required in terms of the logic being used can be easily incorporated into the system with a minimum effort. We designed the system based on three-tier architecture consisting of a Presentation Layer, Business Layer and Data Access Layer. The system has been divided into two main sub-systems according to the functional requirements.

- *Career Known Subsystem* provides advice when the student has a specific career goal. Backward-chaining is to start from a particular fact or from a query and by means of using deductive reasoning to try to verify that fact or to obtain all possible results of the query. Typically, the reasoner decomposes the fact into simpler facts that can be found in the knowledge base or transforms it into alternative facts that can be proven applying further recursive transformations [7]. Since the ultimate goal is already known in the scenario in concern, Backward Chaining was used as an approach for reasoning to determine the achievability of that goal.
- *Career Unknown Subsystem* provides advice when the student is unclear about career objectives. Forward-chaining is to start from the known facts and to perform the inference in an inductive fashion [7]. In this subsystem, the past performance and skills are identified to derive the possible career paths using the Forward Chaining reasoning.

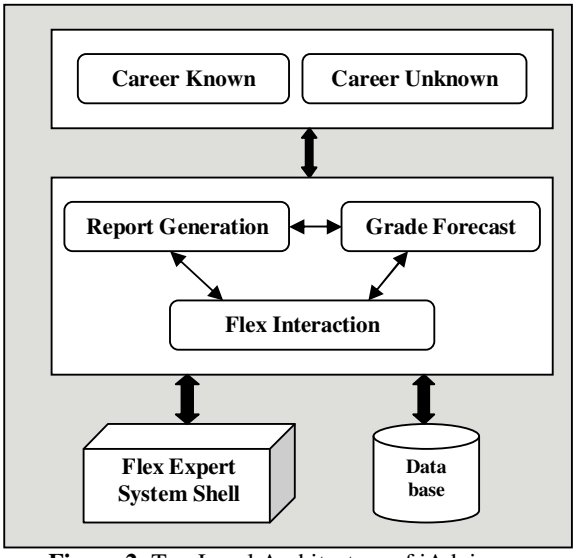

Figure 2: Top Level Architecture of iAdvice

Figure 2 shows the top level architecture of the *iAdvice s*ystem with five modules namely Career Known System, Career Unknown System, Report Generation Module, Grade Forecast Module and Flex Interaction Module which are described in detail in the next sections. From an expert system point of view, the Flex expert system shell's knowledgebase and the database served the purpose of the entire knowledgebase, whereas the inference engine was implemented with both the Flex's inference engine combined with business logic layer and the user interface being provided at the higher level for both Career Known System and Career Unknown System.

3.1. Career Known Subsystem

The main functionality of this subsystem is to indicate whether the user is capable of achieving his/her career goal and to provide advice on the subjects that the student should take up in the next level. It also provides advice on areas to improve to get better performance.

3.1.1. Career Known Module

This is the main module specific to the Career Known Subsystem and the process is described as follows.

- The eligibility criteria for the student to continue into the Honors Degree program is checked, by checking if the student has a GPA (Grade Point Average) of above 2.7. When the career is chosen by the student, this module would first fetch the subject categories which belong to that career
- For each of the subject categories the predicted value for the next level will be determined by using the Grade Forecast Module
- For each of the subject categories the career value for the future level will be determined by using the following formula

  *Category Career value = Predicted Category Value * Category Career Weighting*

- The Benchmark Limit for each of the categories will be also be calculated by considering career weightings applying for each of the categories in order to determine the subject categories the student needs to improve in the future.

  *Benchmark Limit = (Maximum GPA + Minimum GPA to Pass)/2 * (Category Career Weighting)*

- The subject categories which are below the Benchmark Limit will be sent to the Flex Interaction Module in order to retrieve the relevant Category Advice. By adding the individual Category Career Values, the overall Career Value will be calculated.
- The difference between the maximum Career Value which is 4.2 and the actual Career Value will be sent to Flex Interaction Module in order to retrieve the relevant Career Advice. Then it executes the generation of the detailed Career Evaluation Report by sending the datasets on performance, evaluation and advice subjects to the Report Generation Module

3.2. Career Unknown Subsystem

The main functionality of this subsystem is to provide advice on the possible careers for the student by analyzing the past performance as well as the student's interests and skills.

3.2.1. Career Unknown Module

The process of this module has several steps. The major steps are listed below.

- The eligibility criteria for the student to continue into the Honors Degree program is checked, by checking if the student has a GPA of above 2.7
- For each career, the subject categories should be retrieved and for each of the subject categories the predicted value for the next level will be determined by using the Grade Forecast Module
- For each of the subject categories the career value for the future level will be determined by using the same formula used in the Career Known module
  - By adding the individual Category Career Values, the overall Career Value will be calculated for each of the careers

- By using minimum career value of 2.0 the careers which have a Career Value above that limit will be chosen
- For all the chosen careers, selected on performance basis, the specific questions for the career which inquire student's interests as well as other skills required for the career will be retrieved by using the Flex Interaction Module. All the questions will have answers as either "Yes" or "No".
- The Response Ratio for the questions of each of the careers will be calculated as follows.

  Response Ratio =
  No: of positive answers / Total no: of questions

- The Careers which have a Response Ratio of more than 50% will be filtered as the final set of possible careers.
- The generic skills required for the final set of careers will be checked with the user in the form of questions. These questions will be retrieved by using the Flex Interaction Module's Question for Careers sub module. According to the Response Ratio of the Generic skills appropriate advice will be given to the user.

- For each of the chosen careers a Performance Ratio will be calculated using:

  Performance Ratio =
  Career Value / Maximum Career Value

- By considering both the past performance and the qualitative aspects in terms of skills and interests, overall probability of achieving each of the possible careers will be calculated. The careers would be ranked according to the results and advice would also be given accordingly.

  Career Performance Probability =
  Career Performance Ratio / Total Performance Ratio for possible careers

  Career Response Probability =
  Career Response Ratio / Total Response Ratio for possible careers

  Overall Career Probability =
  *((Career Performance Probability + Career Response Probability)/2)*100*

3.2.2. Grade Forecast Module

The module is required by both the career known and career unknown subsystems to calculate the values for the respective subject categories based on the past performance of the students and to determine the most realistic future performance on a linear regression basis. The functions of this module are described in detail as follows.

- To calculate the value for a given category for each of the past levels of the student
- To predict the category value for a future level by analyzing the trends of the past performance for that category
- To save all the category values derived for each level in the database so that a graph could be plotted to visualize the situation
- Determines the current level of the student based on the Category Code and the Student ID and then for each level, the Category Value is calculated by considering the credits and the GPA achieved for each of the subjects in the respective level which belongs to the particular Category.
- Once all the category values are calculated for each of the past levels the average slope is derived, which is then used to predict the slope up to the next level.

3.2.3. Flex Interaction Module

The main responsibility of this module is to interact with the Flex Expert System Shell by providing input and passing the result derived by the inference engine back to the module which requested it. According to the type of advice given the Flex interaction could be further decomposed into three sections as:

- Career Advice

The difference between the maximum value and a student's actual value for a respective career will be calculated and passed on to the Knowledge Specification language file. According to the inference rules, student's performance will be evaluated and as the result the appropriate advice would be sent back to the student from the Flex engine.

- Category Advice

The categories of a particular career which have values less than the benchmark limit defined will be sent as the input and using the inference rules it will advice on the subjects that the student should improve, in order to get better results.

- Questions for Careers

When a single career of a career combination is provided according to the inference rules the system would return the suitable questions for the provided input, in order to conduct a qualitative analysis regarding the skills and interests of the student.

3.2.4. Report Generation Module

This module facilitates the students to get the advisory details and the past performance details in a comprehensive format along with tabular and graphical representations. There are two main reports generated by this module.

- Career Evaluation Report

This report contains a performance evaluation of the student for a chosen career and it is generated by both subsystems. The Career Evaluation Report has been divided in to three sub sections namely Performance Graph, Performance Evaluation and Subject Advice. Performance Graph section includes a line chart indicating the predicted and past category values for each level for the categories belonging to the chosen career. Performance Evaluation section shows the future values derived for each of the categories used in arriving at the final advice including the Performance gap. The section on subject advice is where the student is given advice on the subjects he/she should select in order to be in line with the career goal and also the subjects which the student needs to put more effort in order to improve, in the case of having a value lower than the acceptable performance gap.

- Career Choice Evaluation Report

This report contains a performance evaluation and an evaluation on skills and interests for the possible set of careers for the student. This report provides the rationale behind the selection of possible careers and it is generated by the Career Unknown subsystem. First sub section contains a line chart indicating the Category Values for each level for the categories belonging to the selected set of careers. The section on Performance & Skill Evaluation contains the Performance ratio and the Response Ratio for each of the careers and ranks them according to the derived overall probabilities and also contains the advice on which career is best suited for the student.

4. Implementation of *iAdvice*

The system has been broadly categorized into two sections for the purpose of implementation, being;

- Implementation of the expert features
- Implementation of the business logic

The core functionality of this system is to give customized advice to all its users depending on their performance and career preferences. The knowledgebase and the part of inferencing are implemented through the Flex Intelligent Server. The business logic for the inferencing and the user interfaces of *iAdvice* we used Visual Basic DLLs(Dynamic Link Libraries) and Standard Executables. In addition, the reports are created using Crystal Reports 8.5, which is integrated with the Visual basic DLLs through the references namely Crystal Data Object, Crystal Reports 8.5 ActiveX Designer Runtime Library, Crystal Reports 8.5 Standard Wizard Library, Crystal Report 8,.5 Library and Crystal Report Viewer Control.

4.1. Implementation of the Expert Features using Flex Intelligent Server

iAdvice has to interact with the expert system shell to retrieve appropriate advice to the students regarding various aspects, like the subjects that the student should improve on, whether the student is in-line with his/ her career goals, generating questions on the fly to find the student preferences and skill levels based on a customized manner for each and every student, etc.

Flex supports frame-based reasoning with inheritance, rule-based programming and data driven procedures fully integrated within a logic programming environment, and contains its own English-like Knowledge Specification Language (KSL) [9]. The expert system's knowledgebase was implemented using KSL for defining rules, frames, procedures, etc. The KSL enables developers to write simple and brief statements to handle the expert domain and produce virtually self-documenting knowledge-bases which can be understood and maintained by non-programmers. The knowledge-base comprises of frames, actions, rule-sets, rules and groups. The knowledge-base of our expert system contains three KSL files containing three types of different information namely:

- careeradvice.ksl
- subjectCategoryAdvice.ksl
- questionGroups.ksl

"careeradvice.ksl" contains advice on whether the student is in-line with his/ her career goal. The difference between the calculated final weighted average performance value and the upper limit of the value specified as 4.2 is sent to the expert system, which will in turn make the decision whether the student is in-line with the goal and then accordingly provide an appropriate advice to the student. The action of the knowledge-base will receive the value of the difference, and then using the forward chaining engine the appropriate rules will be fired, and finally a suitable advice will be given to the student. A code snippet of the "careeradvice.ksl" is shown in Figure 3.

```
frame studentAdvice;
default value is 0.0
and default advice is unknown.

rule advice_type_1
  if studentAdvice`s value =< 0.5
  then advice of the studentAdvice becomes 'You
have performed extremely well. Good luck in
achieving your career goal '.
rule advice_type_2
    if studentAdvice`s value =< 1.5 and
studentAdvice`s value > 0.5
    then advice of the studentAdvice becomes
'You have a high probability of achieving your
career goal if you work hard at it '.
rule advice_type_3
    if studentAdvice`s value > 1.5
   then advice of the studentAdvice becomes 'You
need to improve a great deal to be successful in
your career '.
```

Figure 3: Frames and Rules in "careeradvice.ksl"

"questiongroups.ksl" will determine the type of questions that should be asked from the student, in turn to determine the career path of a student, who does not currently have a career of choice. The knowledge-base contains a vast amount of questions, and from these questions the expert system will dynamically decide on what kind of questions should be asked from the particular student, according to his/ her previous answers. The questions are grouped according to career paths and all the possible combinations of career paths and a small section of such implementation is shown in Figure 4.

```
group '1,2,3'
'Are you good in leading teams?',
'Are you comfortable when working in teams
as compared to individual work?',
'Are you willing to share your learnings among
team members?'.

group'1,3,4'
'Can you guide others and give assistance?',
'Are you good in problem solving?',
'Do you have analytical skills?'.
```

Figure 4: Implementation of the Question Bank

"subcategoryadvice.ksl" will determine the subjects that the student should perform well in the following year, in order to be in-line with his/ her career goal. All the subjects are categorized into different groups in the KSL file and then the rules define which subjects to be advised based on the values passed from the business logic layer, which is shown from a part of the KSL file in Figure 5.The expert system advices the students on which subjects that he/ she should concentrate more on the following year. This advice is only given to the students that are not in-line with their career goal.

```
……
group 'MTH'
' Computational Mathematics '.
group 'CSC'
' Embedded Systems ',' High Performance
Computing ',' Human Comuter Interaction '
group 'DOM'
' Bio Informatics 2 '.

group subjectCategory
'MGT','NET','DBS','MUL','AIS','MTH','CSC','DO
M','PRG'.

rule categoryAdvices
if X is categoryAdvice`s name and X is any
subjectCategory
then subjects of the categoryAdvice becomes all
X.

group rules_group
categoryAdvices.

ruleset validate_category_advice
contains rules_group;
update ruleset by removing each selected rule
```

Figure 5: Section of subjectcategoryadvise.ksl

4.2. Implementation of the Business Logic

The business logic of the system is implemented through five classes namely *CareerKnown.cls*, *CareerUnknown.cls*, *FlxInteract.cls*, *GradeForecast.cls* and *Utility.cls*. *AdvisoryBusinessLogic.dll* is the main component of *iAdvice* which takes care of the activation of the initial entry points. In the Career Known system, it executes the *GetCareerAdvice()* method in the *CareerKnown Class*. This method gets all the Subject Categories belonging to the Career chosen by the user and then calculates the predicted value for the next level. The subject categories and the level of the user are retrieved from the database using the *UtilityData class*. The predicted value for each of the subject categories is calculated using the *GradeForecast class*. Then the *FlxInteract class* gets relevant advice for weak subject categories and according to the past performance from the Flex Intelligent Server.

Career Unknown System is implemented in such a way that *GetPossibleCareers()* method in the *CareerUnknown class* of the *AdvisoryBusinessLogic.dll* is used to determine the careers which the user can attempt according to the past performance. This is done by determining the career value for all the careers in the system and selecting the careers which have the career value above a specified limit. The required data such as the careers and the subject categories and their information for each of careers are retrieved from the database using the *UtilityData class* in the *AdvisoryDataAccess.dll*. The predicted value for each of the categories is found using the *GradeForecast class*. The Grade Forecast Module initiates the Report Generation Module to make reports on the fly through the datasets which are passed to the Crystal Report object. The retrieval of data for the creation of reports is implemented in the *ReportData class* in the *AdvisoryDataAccess.dll*. Sub Reports are also attached to the main report as on-demand reports so that the user could drill down to detail level, if required.

5. *iAdvice* in Practice

iAdvice is presented to the user as a stand alone system with simple user interfaces to interact with the system. This provides the users an easily navigable and understandable process which actually hides the complicated logic and expert system functionalities at the back end from the user.

The user's entry point to the system is through a login screen. If the user hasn't registered with the system it provides the facility for the user to register. When the registration is completed a trigger is activated to get student grade information to the local student database from the universities main database where student grades are being stored. Figure 6 is a screen shot of the main user interface that a user sees while accessing the system.

Once logged into the system, it is directed to the next personalized screen which asks the question "Have you decided on a career path?" which determines what subsystem to be implemented. If the user chooses "Yes" option button it executes the Career Known subsystem or else it executes the other subsystem which is the Career Unknown subsystem. If the user opts for the career known option, then a screen will appear showing all the possible career paths available for the user to select. When the "Get Advice" button in the above screen is pressed after the selection of a particular career, the system executes the relevant modules to calculate the predicted value for the next level.

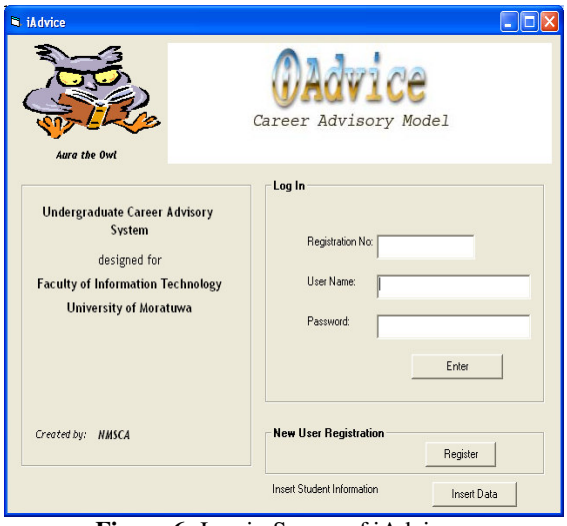

Figure 6: Log in Screen of iAdvice

Using the predicted value for each of the categories a predicted value for the entire career is calculated using a formula specified in the Design Specification, and then it uses Flex interaction module to get the appropriate advice for the user. After all the calculations and appropriate advice is acquired, the final summarized advice is displayed to the user as shown in Figure 7. The advice consists of eligibility of the student to do the special degree, the current SGPA of the user as well as whether the user is in a position to meet the career goal with the current performance level. It also states what subjects to put more effort at the next level to achieve the career goal if the user has under performed in the past on some of the subject categories. By pressing the "View Details" button on the screen the user can get a system generated detailed report on the performance and the justification for the advice.

At the initial point where the system inquires the student as to whether he/she has already determined a career goal, and if the student states "No", then the system proceeds accordingly. *iAdvice* advices on the careers which the user can attempt according to the past performance. This is done by determining the career value for all the careers in the system and selecting the careers which have the career value above the limit. The required data such as the careers and the subject categories and their information for each of the careers are retrieved from the database and are calculated wherever necessary.

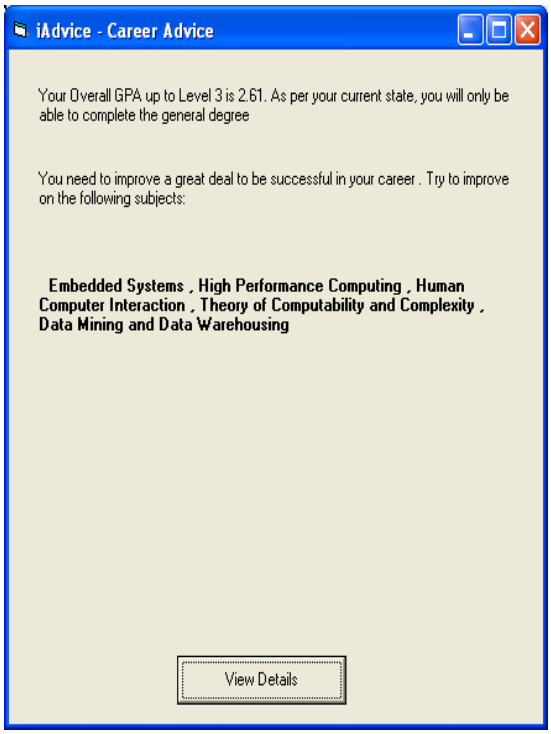

Figure 7: Customized Advice Screen

Once the possible careers are determined the user is prompted with several questions in order to determine user likings and skill level. The expert system will provide the questions relevant for the careers identified by inferring the knowledgebase. The questions will be displayed in the another user interface as shown in Figure 8 and by pressing the next the user could answer all the questions in terms of Yes/No.

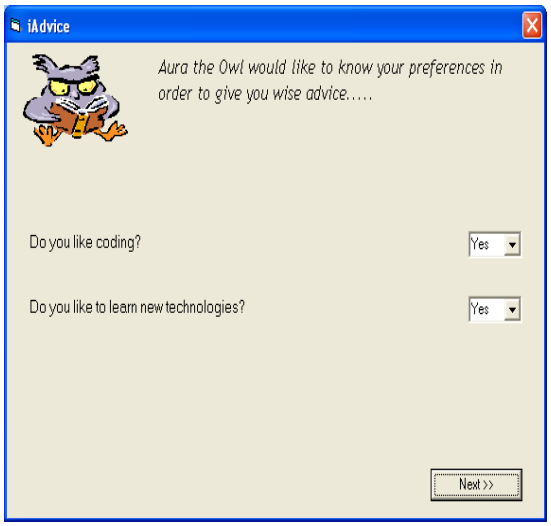

Figure 8: Screen for Questions on Skill Evaluation

According to the responses given, Response ratio is calculated. Using both performance ratio and the response ratio the probability of success is determined. Based on the overall probabilities, the relevant advice is given in a summarized form in the next screen shown in Figure 9. This advice consists of eligibility of the student to do the special degree, the current SGPA of the user and the probabilities of suitability of the possible careers for the user. This interface provides the user to view two sets of reports named as Individual Career Report and Detailed Career Report.

The Individual Career Report would generate a report containing the Performance Evaluation for a chosen career out of the recommended list of careers. This is the same report generated by the Career Known Subsystem which contains the line chart of the user's performance, Performance Evaluation and the Subject Advised along with relevant advice for the user on the considered career. The Detailed Career Options contains a sub report which contains the Performance, skill level of the user for each of the careers and the probability derived accordingly.

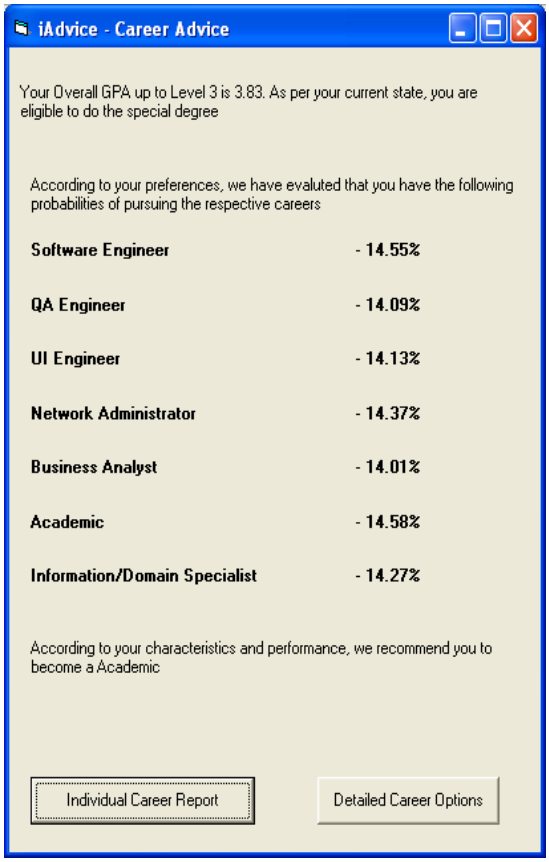

Figure 9: Screen providing Possible Career Options and Advice

Once the user has got all the advice that is required, the user can log out of the system. If the user requires more up to date advice, he/she can reuse the system whenever it is needed and *iAdvice* will always give the most appropriate advice based on the changes being made to the grades or due to the access to more recent student performance.

6. Evaluation

In order to assess the functionality of *iAdvice*, an evaluation was held which considered 30 students as sample data. Accordingly the students were given an evaluation form which covered a range of criteria which the model considered, in order to evaluate whether the advice and the guidance given by the expert system matches the user profile. The criteria considered in the evaluation form are shown below.

- Predicted Performance and actual capability
- Conformity of subjects suggested
- User capability and preference for careers proposed by the system
- Ranking of careers
- Rating of the overall system

According to the results of the evaluation, there was positive feedback for the system from all viewpoints as depicted in Figure 10. 81% students agreed that the predicted performance values reflect a true possibility according to their capability. However, there were also a few students who thought that they are capable of performing better in certain categories than predicted. 18% of the students felt that the system was reflecting over-performance than actually within their capability limit due to the side-effect of using past exam grading, and psychological impact of the users.

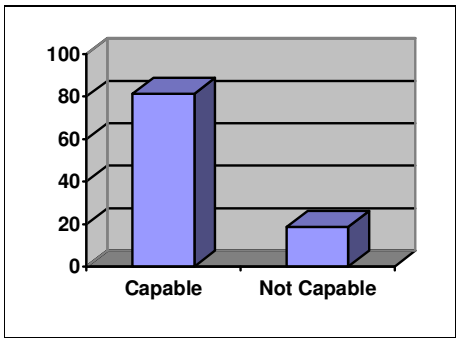

Figure 10: Capability in performing well in all of the proposed careers

When the user did not specify a career path, the system was able to specify a list of probable careers in line with user likings and capabilities every time, and out of which around 70% assured that the order specified were also accurate. The evaluation of the accuracy and compliance of ranking of the possible careers showed that about 60% of the time it is almost truly in line with the student expectations whereas 40% of the time the ranking differed with the same set of careers which is graphically shown in Figure 11.

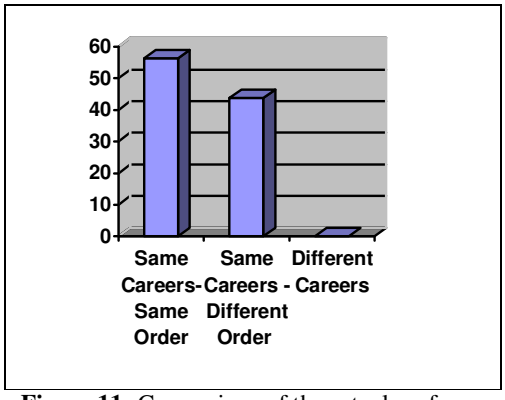

Figure 11: Comparison of the actual preference of careers with the predicted careers

It was evident that most students found this as a useful system to find which career is more suitable as well as subjects that need to be focused to achieve the career goal. Around 85% were impressed with the reporting facility which gave a comprehensive view of once performance as well as a rationale for the advice provided. Table 1 shows the factors given to the students in evaluating the overall system rating which is the final criterion considered in the evaluation form.

Criteria	Good (%)	Medium (%)	Bad (%)
Usefulness - Career	87.5	12.5	0
Usefulness - Subjects	75.0	25.0	0
Relevance of Advice	87.5	12.5	0
Reports	87.5	6.25	6.25
Coverage of Factors	62.5	37.5	0
Overall System	87.5	12.5	0

Table 1: Overall System Evaluation

7. Conclusion

This paper has reported on *iAdvice*, the Career Advisory Expert System as a supportive tool to the students in determining their future career paths and consistently be in line with their career goals.

We have reported on the approach taken for the development of the expert system, the design, implementation and the user process of using *iAdvice*. This system has been developed by targeting the undergraduates of the Faculty of Information Technology of University of Moratuwa. However, *iAdvice* can be used for the purpose of providing effective career advice to any student, with little customization. It can even be useful for the secondary education system, where school students can be guided to be inline with a future career goal from the early stages itself and can also be used as a general career guidance expert system with some modifications to the knowledgebase to suit the particular requirements.

The evaluation of the system has been conducted using a sample of 30 undergraduates and it has been found that the system is quite successful. 81% students agreed that the predicted performance values reflect a true picture of their actual capabilities. Compliance of ranking of the possible careers showed that about 60% of the time it is truly in line with the student expectations whereas 40% of the time the ranking differed with the same set of careers. With regard to the overall system, around 85% provided positive feedback. Therefore, we conclude that *iAdvice* can be effectively used as a career advisory expert system to be used by any student. We wish to point out that this system can also be used by career guidance counselors as a supportive system in providing advice. By doing so, they can have more access to information from various sources such as past examination details and industry requirements while ensuring that the quality of the advice is improved.

Further work of this project includes making it available as a web application, enhancing the system as a customizable solution to suit any educational institute requirement and incorporating Natural Language Processing (NLP) features. By introducing NLP to *iAdvice*, it will increase the usability of the system since it is believed that the effective use of language is intertwined with human general cognitive abilities [6].

8. References

[1]. Aamodt A. and Plaza E. (1994). Case-Based Reasoning:Foundational Issues, Methodological Variations, and System Approaches;AICom - Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59.

[2] Bratko I. Prolog – Programming for Artificial Intelligence, 3rd Edition

[3] Department of Census & Statistics, http://www.statistics.gov.lk/samplesurvey/Bullet_2005_8_spe_English.pdf

[4] Google.2006. http://www.google.lk

[5] Jackson P., Introduction to Expert Systems, 2nd Edition

[6] Jurafsky D. and Martin J.H. Speech and Language Processing; An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition

[7] Ontotext Lab, Sirma Group Corp. Reasoning Strategies, http://www.ontotext.com/inference/reasoning_strategies.html

[8] Russell S.J, Norvig P, Artificial Intelligence – A Modern Approach, 2nd Edition

[9] Vasey P. Flex Expery System Toolkit, version 1.2. Logic Programming Associates Ltd. London, England. 1989.

Widening Scope of Principal Component Analysis: A Fuzzy Approach

D.S. Kalana Mendis, Asoka S. Karunananda, U Samarathunga

Department of Information Technology, Advanced Technological Institute, Labuduwa, Sri Lanka
Faculty of Information Technology, University of Moratuwa, Sri Lanka
Gampaha Wickramarachi Ayurveda Institute, University of Kelaniya, Sri Lanka

E-Mail: kalanaatil@mail.com

Key words: Fuzzy statistics and data analysis; principal component analysis; knowledge modeling; Ayurveda

Abstract

Knowledge modeling is concerned with abstract model mapping using real world domains. Further all knowledge is tacit or rooted in tacit domains. Abstracting is mainly concerned with classification of such knowledge. In this issue statistical techniques can be issued and mainly concerned with multivariate statistical techniques. However using principal component analysis (PCA) as a multivariate technique makes problematic situation due to inability of classifying knowledge. Analysis using PCA is limited up to principal components extracted by PCA. Therefore existing algorithm of PCA to be addressed for modeling knowledge should be modified. This paper presents a novel mechanism for modifying PCA algorithm to address the problem in concerned using Fuzzy Principal component Analysis (FPCA). Here principal components have been used to define intervals for membership function. By doing so, knowledge classification is done effectively by constructing fuzzy memberships functions integrated with PCA. The experimental results using Ayurvedic medicine show that our approach is very promising.

1. Introduction

The existing system of tacit knowledge modeling in Ayurvedic medicine is not consistent [6, 5]. Experts use questionnaires that lead several problems like dependencies among the questions in the questionnaire and analysis of the constituent type. Although Principal Component Analysis (PCA) provides an algorithm to extract principal components from tacit knowledge for removing dependencies among questions in the questionnaire it is meant to address problems such as classification about the underline knowledge.

The ability to model tacit knowledge and represent modeled knowledge is critical to the success of experts, knowledge engineers and general users. It is a problem to experts to take decisions using the modeled tact knowledge as guidelines [9, 13, 15, 22, 25, 26]. It allows decision makers to take the changes into consideration and even take advantage of the changes when they make decisions. Knowing an effective knowledge-modeling algorithm in advance not only allows an expert to provide new guidelines and knowledge engineers to satisfy the requirement of its customers but also allows it to design corrective actions to stop inconsistencies or reduce uncertain decisions.

Existing principal component analysis as a multivariate technique, aimed at producing accurate models of the real world in an effective manner. But all real world domains are considered as tacit or rooted in tacit domains [14, 31, 32]. These are normally represented as questionnaires for human users to better understand the tacit problem domains and for modeling. However, regardless of how accurate a model can predict, it can only model based on the selected features and cannot lead to any knowledge classification because it will no longer be valid otherwise.

In this paper, we study the limitation of modeling tacit knowledge in the context of using principal component analysis (PCA) as a multivariate

technique for classification of knowledge. The problem of PCA above is defined as follows [7]. Let *S* be the set of all questions in the questionnaire and *P* be the set of all extracted principle components. Further, $S = \{S_1, S_2, ..S_{m-1}, S_m\}$, Then I^{th} principle Component is defined as

$$PC_i = a_{1i}S_1 + a_{2i}S_2 + .. + a_{m-1,i}S_{m-1} + a_{m,i}S_m$$

.For n number of extracted principal components, following computation is concluded.

$$X = \sum_{j=1}^{n} PC_j$$

$$\therefore X = \sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}S_i \ .$$

Many researchers have used it extensively and many algorithms have been proposed in PCA based multivariate statistics literature (e.g., [8]).

This problem definition is not developed for classifying knowledge in domains with tacit knowledge. Knowledge modeling without taking the classifications into consideration can result in severe degradation of performance of tacit knowledge. Since it is common that we need to model tacit knowledge based on a questionnaire, the modeling of scope in PCA is an important problem.

This paper presents the problem definition effectively. Furthermore, we generalize the work so that fuzzy PCA integrated system (FPCA) can be used for tackling this problem. Given principle component extracted from PCA, we propose to use fuzzy sets and fuzzy variable to model the knowledge mapped on a questionnaire. Furthermore, we propose to build a fuzzy membership functions to classify knowledge generated as output of PCA. The fuzzy membership functions can then be converted to a set of fuzzy rules.

These fuzzy rules are called *fuzzy multivariate -rules* because they are rules about multivariate technique. The *fuzzy multivariate -rules* can be used for human users to classify and for reasoning of tacit knowledge modeling.

The rest of this paper is organized as follows. In Section 2, we present the problem definition of principal component analysis in classification of knowledge. In Section 3, we describe an approach for classifying principal components.
To evaluate the performance of our approach, we applied it to Ayurvedic domain. The experimental results are provided in Section 4. Testing is described in section 5. In Section 6, we discuss the related work in the literature. Finally, we conclude this paper with a summary in Section 7.

2. The Problem Definition

Let S be the set of all questions in the questionnaire and P be the set of all extracted principal components.

Further, $P = \{PC_1, PC_2, ..PC_{n-1}, PC_n\}$

$$S = \{S_1, S_2, ..S_{m-1}, S_m\}$$

$$PC_i = a_{1i}S_1 + a_{2i}S_2 + .. + a_{m-1,i}S_{m-1} + a_{m,i}S_m \quad (1)$$

Let *M* be the principal components Matrix for filtered tacit knowledge.

$$M = \begin{bmatrix} a_{11}a_{12}..a_{1n} \\ a_{21}a_{22}..a_{2n} \\ . \\ . \\ a_{m1}a_{m2}..a_{mn} \end{bmatrix} \quad (2)$$

$$\therefore PC_1 = a_{11}S_1 + a_{21}S_2 + .. + a_{m1}S_m \quad (3)$$

$$PC_2 = a_{12}S_1 + a_{22}S_2 + .. + a_{m2}S_m \quad (4)$$

$$PC_{n-1} = a_{1n-1}S_1 + a_{2n-1}S_2 + .. + a_{mn-1}S_m \quad (5)$$

$$PC_n = a_{1n}S_1 + a_{2n}S_2 + .. + a_{mn}S_m \quad (6)$$

For n number of extracted principal components, following computation is concluded.

$$X = \sum_{j=1}^{n} PC_j \quad (7)$$

$$\therefore X = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} S_i \quad (8)$$

Further classification of knowledge in the questionnaire cannot be done using above described n number of extracted principle components.

Definition 1. The first principal component of the complex of sample values of the responses $X_1,\ldots X_p$ is the linear compound

$$Y_1 = a_{11} X_1 + \ldots + a_{p1} X_p$$

whose coefficients a_{i1} are the elements of the characteristic vector associated with the greatest characteristic root l_1 of the sample covariance matrix of the responses. The a_{i1} are unique up to multiplication by a scale factor, and if they are scaled so that $a'_1 a_1 = 1$, the characteristic root l_1 is interpretable as the sample variance of Y_1.

Example 1. Following table shows component matrix evaluated using SPSS

Component Matrix

	Component	
	1	2
VAR1	.988	.150
VAR2	.287	.958
VAR3	-.875	.483

Extraction Method: Principal Component Analysis.

2 components extracted.

The first principal component accounts for the three dimensions as follows:

$$Y_1 = 0.988(VAR1) + 0.287(VAR2) - 0.875(VAR3)$$

Definition 2. The jth principal component of the sample of p-variate observations is the linear compound

$$Y_j = a_{1j} X_1 + \ldots + a_{pi} X_p$$

whose coefficients are the elements of the characteristic vector of the sample covariance matrix S corresponding to the jth largest characteristic rot l_j. If $l_i \neq l_j$, the coefficients of the ith and J[th] components are necessary orthogonal; if $l_i = l_j$, the elements can be chosen to be orthogonal, although an infinity number of such orthogonal vectors exists. The sample variance of the jth component is l_j, and the total system variance is thus

$$l_1 + \ldots + l_p = trS$$

The importance of the jth component in a more parsimonious description of the system is measured by

$$\frac{l_j}{trS}$$

3. Classifying Principal components

In the exciting system, tacit knowledge modeling is not consistent [12]. In general, questionnaires are used to capture tacit knowledge, but the questions in such questionnaires can have various dependencies without notice. Although Principal Component Analysis (PCA) provides algorithm to extract principal components from tacit knowledge by removing dependencies among questions in the questionnaire, it is considered to address problems such as classification about the underline knowledge

Fuzzy Logic

Fuzzy logic handles situations, where conclusions do not fall into one extreme. As compared with classical logic, fuzzy logic can handle real world problems, which deal with more than two truth-values. In fuzzy logic, everything is a matter of degree. Therefore fuzzy logic can be used to make decisions in domains with tacit knowledge [13]. Individual classification in Ayurveda is a classic example, where the decision has more than one possible truth-value.

In our research we have used fuzzy logic for addressing the vagueness involved in tacit knowledge. For example, vagueness (dependencies & inconsistencies) involved in Ayurvedic classification of individuals has been manipulated using fuzzy logic.

3.1.1 What is a Fuzzy Set?

A fuzzy set can be simply defined as a set with fuzzy boundaries. Let X be the universe of discourse and its elements be denoted as *x*. Fuzzy set *A* of universe *X* is defined by function $\mu_A(x)$ called membership function of set *A*.

$$\mu_A(x):\ X \longrightarrow [0.1]$$

This degree, a value between 0 and 1, represents the degree of membership, also called membership value, of element *x* in set *A*.

3.1.2 Fuzzy Rules

A fuzzy rule can be defined as a conditional statement in the form:

IF *X*= A THEN *Y*= B

Where *X* and *Y* are linguistic variables; and *A* and *B* are linguistic values determined by fuzzy sets on the universe of discourses *X* and *Y*, respectively.

Building algorithm for Fuzzy Principle Component Analysis (FPCA)

Algorithm for FPCA has been designed by extending the existing mechanism of principle components analysis (PCA). There are 2 no. of stages used for FPCA in addition to PCA, which are described as follows.

3.2.1 Generatingmembership functions(fine–tuning analysis)

Let LS be the Likert scale [14], then

$$LS = [L,..,U] \qquad (9)$$

X_L and X_U values are derived from results of the filtered tacit knowledge. It is computed as given below.

$$\therefore X_L = L \sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij}$$

$$\therefore X_U = \mathbf{U} \sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij} \qquad (11)$$

Let A be fuzzy set defined on a fuzzy concept using the interval of $[X_L,.., X_U]$. Then membership function is as follows.

$$\begin{cases} 0 & X=<X_L \\ A(X) = (X-X_L)/(X_U-X_L) & X_L<X<X_U \qquad (12) \\ 1 & X>=X_U \end{cases}$$

3.4. Fuzzy rule base (reasoning)

Fuzzy rules can be constructed as follows,

Rule 1: If $X <= X_L$ then A (X)=0 %

Rule 2: If $X_L<X<X_U$ then $(X-X_L)/(X_U-X_L)$ %

Rule 3: If $X>=X_U$ then A (X)=100 %

Adding dynamically, in order to operate the reasoning process for answers given by the fuzzy rules, fuzzy rule base can be extended further.

4. Experimental results

Although Ayurvedic practitioners use questionnaire but those are faced with several problems like dependencies among the questions in the questionnaire and classification of constituent type [5,6]. We addressed these problems to solve them in following stages.

4.1 PCA for removing dependencies

It consisted of 72 questions to analyze *vata*, *pita* and *kapha*. We have done a pilot survey for 100 students for statistical modeling using the questionnaire. Principal component analysis has been used to remove dependencies in the questionnaire. Twenty five (25) principal components have being identified using SPSS [17] as shown in matrix given below. Here V1, V2..V24,

K1, k2..K24, P1, P2..P24 denotes question-numbering system in the questionnaire (10)

$$M = \begin{pmatrix} V = \begin{array}{c|c} & \begin{array}{ccccc} 1 & 2 .. & 24 & 25 \end{array} \\ \begin{array}{c} V1 \\ V2 \\ . \\ . \\ V23 \\ V24 \end{array} & \begin{pmatrix} -0.228622 & 0.249362 . & -0.073945 & 0.058179 \\ 0.08431 & 0.20654 . & -0.097192 & -0.112795 \\ & & & \\ -0.645803 & 0.232312 . & 0.0067 & -0.083959 \\ -0.222147 & -0.06453 . & -0.073514 & 0.084404 \end{pmatrix} \end{array} \\ K = \begin{array}{c} K1 \\ K2 \\ . \end{array} \begin{pmatrix} 0.012511 & -0.096332 . & 0.141314 & 0.25113 \\ -0.005642 & 0.268145 . & -0.179992 & 0.111715 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \end{pmatrix}$$

$$V = \begin{pmatrix} -0.228622 & 0.249362. & -0.073945 & 0.058179 \\ 0.08431 & 0.20654. & -0.097192 & -0.112795 \\ & & & \\ -0.645803 & 0.232312. & 0.0067 & -0.083959 \\ -0.222147 & -0.06453. & -0.073514 & 0.084404 \end{pmatrix}_{24*25}$$

4.2 Fuzzy logic for classifying knowledge

Human constituents can be computed in to *vata, pita* and *kapha* in percentages as shown below. Membership functions for *vata, pita* and *kapha* have been constructed using fuzzy logic based on out puts of principal component analysis.

4.21 Membership function for classifying Vata constitution

Boundary values of membership function have been constructed using the output of the principal component analysis.

$$\because X_L = 1 \sum_{i=1}^{25} \sum_{j=1}^{24} a_{ji} = 8.510004 \quad (13)$$

$$\because X_U = 6 \sum_{i=1}^{25} \sum_{j=1}^{24} a_{ji} = 51.06002 \quad (14)$$

Here X_L denotes lower bound value at the minimum level of evaluation scale (Does not apply) in the questionnaire. X_U denotes upper bound value at the maximum level of evaluation scale (Applies most) in the questionnaire

$$V(X) = \begin{cases} 0 & X =< X_L \\ (X-X_L)/(X_U-X_L) & X_L < X < X_U \\ 1 & X => X_U \end{cases}$$

V(x) denotes membership function for classifying *vata* constitution. This has been constructed using Visual Basic

4.2.2 Constructing fuzzy rules

Fuzzy rules can be constructed using the design model for fuzzy rules generator described in section 3.

So following fuzzy rules can be illustrated for classing humeral constitutions in to *vata, kapha* and *pita* in term of percentage values.

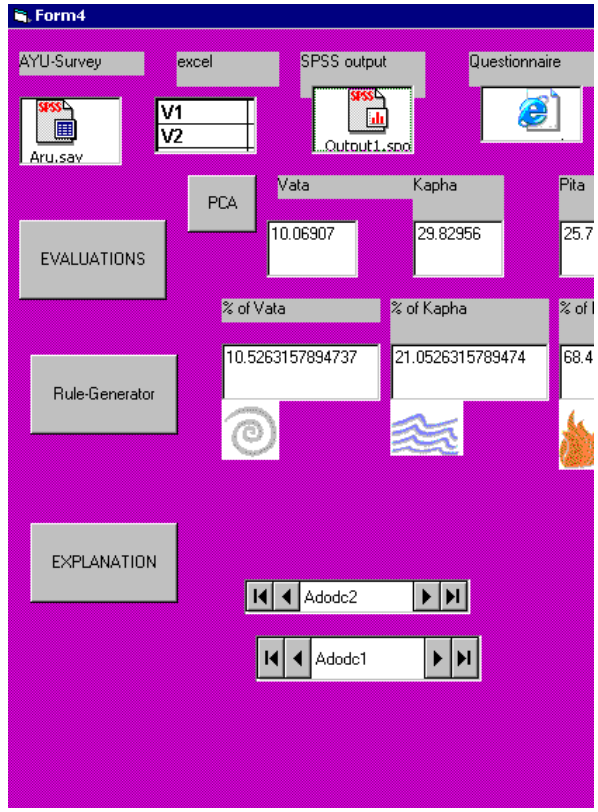

Fig.1. Analysis window

4.2.2.1 For *Vata* constitution

Rule 1: If $X <= X_L$ then $V(X)=0$ %

Rule 2: If $X_L<X<X_U$ then $V(X)= (X-X_L)/(X_U-X_L)$ %

Rule 3: If $X>=X_U$ then $V(X)=100$ %

5. Testing

The expert system developed using this approach was tested with a group of 30 persons of Ayurvedic experts and students (see Table 1).

id	vata	pitta	kapha	Expert_decision
14	25.71	20.71	53.57	KV
15	32.95	23.86	43.18	VP
16	39.88	23.81	36.31	VP
17	27.65	46.1	26.24	KP
22	25.69	29.36	44.95	KV
23	33.58	24.09	42.34	KV
24	25.71	34.28	40	KP
26	32.21	31.54	36.24	KV
27	22.51	29.8	47.68	KP
28	20.37	30.56	49.07	PK
35	30.6	35.52	33.88	PK
46	29.71	17.39	52.9	KV
47	41.07	10.71	48.21	KV
48	34.5	32.16	33.33	KV
49	23.46	28.57	47.96	PK
50	35.27	30.77	33.97	KV
52	42.36	36.11	21.53	VP
56	23.01	35.71	41.27	PK
65	47.94	19.86	32.19	KV
66	14.03	35.96	50	PK
67	19.15	36.88	43.97	PK
68	22.46	25.36	52.17	PK
69	40.47	26.78	32.74	PK
70	30.28	29.58	40.14	KV
71	12.71	44.92	42.37	PK
72	11.18	40	48.82	PK
73	11.24	40.24	48.52	PK
74	23.44	26.9	49.66	PK
93	17.09	36.75	46.15	KV
94	33.09	30.15	36.76	KV

Table 1: System Testing: Expert Vs. System

The evaluation was conducted to see that the answers generated by the system matches with the identification by Ayurvedic experts and the students. Further, the system's ability to fine-tune the answers was also tested. It has been investigated that 23 (77%) of conclusions matches with the system and expert (see Table 2), which leads to determine the accuracy of the system.

id	vata	pitta	kapha	Expert_ decision	conclusion
14	25.71	20.71	53.57	KV	matched
23	33.58	24.09	42.34	KV	matched
24	25.71	34.28	40	KP	matched
26	32.21	31.54	36.24	KV	matched
27	22.51	29.8	47.68	KP	matched
28	20.37	30.56	49.07	PK	matched
35	30.6	35.52	33.88	PK	matched
46	29.71	17.39	52.9	KV	matched
47	41.07	10.71	48.21	KV	matched
48	34.5	32.16	33.33	KV	matched
49	23.46	28.57	47.96	PK	matched
50	35.27	30.77	33.97	KV	matched
56	23.01	35.71	41.27	PK	matched
65	47.94	19.86	32.19	KV	matched
66	14.03	35.96	50	PK	matched
67	19.15	36.88	43.97	PK	matched
68	22.46	25.36	52.17	PK	matched
70	30.28	29.58	40.14	KV	matched
71	12.71	44.92	42.37	PK	matched
72	11.18	40	48.82	PK	matched
73	11.24	40.24	48.52	PK	matched
74	23.44	26.9	49.66	PK	matched
94	33.09	30.15	36.76	KV	matched

Table 2: Comparison of Conclusions: Expert Vs. System

The system facilitated to derive constituents types in percentages while Ayurvedic experts obtain only the constituent type. As recommendation given by the Ayurvedic experts, determining constituent's types in percentages is an important criterion for prescribing drugs for a disease. Further, our system provides as an option to find out possible diseases. In general, the system can be used as a self-assessment for finding constituents. According to Ayurvedic medicine, regiments can be done easily by knowing the constituent type. The human constituents can be computed as a combination. So it would help to find the effectiveness of minimum type in a diagnosis.

5. Discussion

Present mechanism in PCA [7] is not valid in knowledge modeling because it can reduce dependencies but not further classification of knowledge. However by integrating fuzzy logic module, present algorithm of PCA is extended for classification of knowledge.

However FPCA gives novel approach in classifications of Principal Components in principle components analysis.

Construction of Fuzzy membership functions using PCA gives an extensive advantage for tacit knowledge modeling. Here we use a case study of classification of human constitutions in Ayurvedic medicine and gained significant results to reduce existing inconsistencies of diagnosis of human constitutions. In the first instance tacit knowledge involved for classification of human constitutions has been mapped in to a questionnaire. PCA has been used to reduce dependencies among questions in the questionnaire. But existing PCA algorithm could be used up to this stage. However we needed to classify human constitutions in to 3 categories. Existing PCA algorithms has been integrated with fuzzy logic module to classify constitutions in to vata, pita and kapha by giving percentages.

6. Conclusion

In this paper, we present the definition of the problem of classification of principal components in principle component analysis. The proposed approach allows a fuzzy principle component analysis to be used for tackling this problem. In tacit knowledge modeling, it is required to (i) map the tacit knowledge into a questionnaire and reduce dependencies used; (ii) classify the knowledge in the questionnaire using extracted principal components; and (iii) reasoning for knowledge classification using fuzzy rules.
To model tacit knowledge, to do classification and reasoning in a comprehensible fashion, we propose to use fuzzy membership functions. The fuzzy membership functions can then be converted to fuzzy rules. These fuzzy rules are called fuzzy PCA-rules because they are rules about principal components. Furthermore, the discovered fuzzy PCA-rules can be used to suggest decisions using modeled tacit knowledge. To evaluate the performance of our approach, we make use of a domain of tacit knowledge of which is a set of rules collected in analysis of human constitutions in Ayurvedic domain. Tacit knowledge in human constitutions analysis of Ayurvedic domain is mapped in to a questionnaire. Fuzzy membership functions are then constructed in the classification of tacit knowledge modeling and fuzzy rules for reasoning the output of classification of human constitutions analysis in Ayurvedic domain. The experimental results show that our approach is very promising.

7. References

[1] Karunananda A.S., "How To Do Research " Author Publication, 2000.

[2] Jonson L., "Expert system Architectures", Kopan Page Limited, 1988.

[3] George J., "Fuzzy sets and Fuzzy Logic", Prentice Hall of India, 1997.

[4] Noak V., "Discovering the world with Fuzzy logic", A Springer – Verlag Company, 2000, PP. 3 – 50.

[5] Dubey G.P.,"The Physiological concepts in Indian medicine", Science and Philosophy of Indian medicine, 1978.

[6] Tripathi S.N., "Clinical Diagnosis", Science and Philosophy of Indian medicine, 1978.

[7] Chatfied C.," Introduction to Multivariate Analysis", Chapman and Hall, 1996.

[8] Morrison D.F., "Multivariate Statistical Methods", Mc Graw-Hill, Inc, 1990, PP. 312 - 357.

[9] Ross D.,"Developing Knowledge-Based Client Relationship", Butterworth Heinemann, 2001.

[10] Karunananda A.S.,"Pinki – A motivation for Artificial Intelligence projects",Author publication, 2001.

[11] Mendis D.S.K, Karunananda A.S, Samaratunga U., "Reasoning with Uncertainty", Health Informatics Conference, Colombo, 2003.

[12] Mendis D.S.K, Karunananda A.S, Samaratunga U., "Tacit knowledge modeling", Sri Lanka Association for Artificial Intelligence Annual Session and Annual General Meeting, Colombo, 2003.

[13] Bellman R.E, Zadeh L.A., "Decision making in a fuzzy environment", Management Science 17, 1970, pp. 141–164.

[14] Richards D, Bush P., "Measuring, Formalizing and Modeling Tacit Knowledge" IEEE/Web Intelligence Conference (WI-2003) Bejing, 2003.

[15] M. Dzbor., "Explication of Design Requirements Through Reflection on Solutions", In Proc. of 4th IEEE Conference on Knowledge-based Intelligent Engineering Systems. Brighton, UK, 2000,pp. 141-144.

[16] Dave W., Flex reference guide, LPA, U.K, 2000.

[17] Matei Ciobanu Morogan., SPSS for windows, release 8.0.0, DSV KTH/ SU, Sweden, 1997.

[18] Evangelos P., "Mastering Visual Basic 6", BPP Publications, New Delhi

[19] George J.K, Yuan B., Fuzzy sets and Fuzzy logic, prentice hall of India, 1995, pp. 280–300.

[20] XpertRule Knowledge Builder, www.attar.com.

[21] Peter J., Introduction to expert systems, Pearson education Asia, 1995, pp. 182–199.

[22] Karl B, Klaus P, A,Yoichi H, Thomas E. Rothenfluh, Harald L., "Knowledge acquisition in the fuzzy knowledge representation framework of a medical consultation system" International Journal of Artificial Intelligence in medicine, 2004,pp. 1–26.

[23] Rajeev K., "Integrating knowledge bases in expert system shells: an open system approach", International Journal of computer application in technology, 1995, pp. 78–89.

[24] Coppin G, Skrzyniarz A., "Human–centered processes : Individual and distributed decision support", IEEE Intelligent systems, 2003, pages 27–33.

[25] Hayashi Y, Tazaki E, Yoshida K, Dey P., "Medical diagnosis using simplified multi-dimensional fuzzy reasoning", In: Proceedings of IEEE 1988 International Conference on Systems, Man and Cybernetics, Beijing, China, 1988. pp. 58–62.

[26] Kolousek G., ″The system architecture of an integrated medical consultation system and its implementation based on fuzzy technology", Doctoral thesis, Technical University of Vienna, Austria, 1997.

[27] Binaghi E., De Giorgi O., Maggi G., Motta T., Rampini A., "A knowledge acquisition tool for computer assisted diagnosis of postmenopausal osteoporosis using a fuzzy expert system shell", Computers and Biomedical Research, 1993,pp.498-516.

[28] Zadeh L. A., Fuzzy Logic, Computer IEEE, 1998, pp. 83-93.

[29] Senkar J., ″Network and tacit knowledge in innovation″, Economics et societies, 1995, pp. 99-118.

[30] Zimmerman H.J., Zimmerman Fuzzy set theory and its applications, Kluwer–Nijhoff, Boston, 1995.

[31] Ravi V, Reddy P. J. and Zimmermann H.J., Pattern classification with principal component analysis and fuzzy rule bases European Journal of Operational Research, November 2000, pp. 526-533

[32] Paolo G, Henk A. L. Kiers., Principal Component Analysis of symmetric fuzzy data, Computational Statistics & Data Analysis, Volume 45, Issue 3, 10, April 2004, Pages 519-548

Qualitative Realization of the Visual World

D. D. M. Ranasinghe,

Department of Mathematics & Computer Science, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka

E-mail: menaka_dul@yahoo.com

Abstract

Visual capability is one of the strongest senses of perception of human beings. Humans tend to represent the visually perceived world mainly in a qualitative manner and thereby attain considerably high accuracy in reasoning and prediction. Even though this is an innate ability of humans, embedding this feature in the development of cognitive vision systems has been a research challenge. A research has been carried out to develop a system that is capable of learning qualitative rules that underlies in the arrangement of the observed visual scene. A set of symbolic data generated from a dynamic visual scene that comprise object movement is considered as the input to the system and by analyzing object-object qualitative spatial and temporal representation and reasoning mechanisms the system generates the underlying set of rules of the scene. As an application the system produces sketch images of the observed scene. This work has great potential in developing agents that can be used for autonomous learning from visual scenes in a manner closer to human learning from visual scenes.

1. Introduction

Visual ability is one of the richest and most developed avenues of information gathering for humans. Therefore learning from visual scenes is a natural cognitive process and we humans, tend to represent this perceived real world mainly in a qualitative manner rather than in a quantitative manner. Even though quantitative measures yields high precision in representation and reasoning, use of phrases such as *the post office is left to the school* in describing a location, *take the second turn to right side* when describing a direction and *keep the saucer before keeping the cup on top of it* are some common examples for usage of qualitative knowledge in day-to-day life. Further, qualitative representation and reasoning is a good alternative when exact information is missing or when the quantitative manipulations are costly. In addition, the limited mental ability of an average human in performing quantitative manipulations is also another factor for using qualitative knowledge.

Hence, when observing an evolving scene we tend to abstract various qualitative features such as movement, size, brightness, position, and orientation etc of the objects in the scene. It is evident that most of these features are either spatial concepts or temporal concepts. In general much of the knowledge about space and time is qualitative than quantitative [1].

By observing an evolving scene humans tend to develop a conceptual understanding about what is taking place and this understanding is conditioned by exposing to similar situations. Depending on the intention of the observer this abstracted knowledge can be organized to derive new knowledge or update the existing knowledge. In doing so one has to learn the underlying rules of the scene.

Based on the above philosophy we have developed a computer-emulated system that exploit qualitative spatial and temporal representation and reasoning mechanisms on a set of symbolic data obtained from a visual scene and thereby generate rules of the scene using Inductive Logic Programming (ILP).

The rest of the paper is organized as follows. Section 2 carries an overview of cognitive vision systems that exploit qualitative representation and reasoning mechanisms. Section 3 is on the theoretical foundations adopted and section 4 reports the approach taken. Section 5 is on design and implementation while section 6 carries a discussion about the results. Finally section 7 is on conclusion and further work.

2. Qualitative representation and reasoning of visual scenes

The area of cognitive computer vision systems is devoted for learning from cognitively enabled vision and is an active research area in the field of AI [2]. Cognitive computer vision systems deal with vision data with respect to the cognitive process of knowing, understanding, and learning about the things that we happen to see. Hence it has facilities for acquiring data from the outside world through learning or

association and produces a response to appropriate percept. Cognitive vision computer systems that can be fully embedded in the environment need to have facilities for automatically acquiring data, process them and develop conceptual models of the environment in such a manner as humans do.

Badler's work is considered as one of the earliest attempts to learn qualitative models of visual scenes [3]. He conceptualised the observed scene with a hierarchy of motion concepts such as behind, after and so on. Due to the limitations in technology, obtaining these concepts automatically was not a feasible task at that time. Hence these concepts were given beforehand and this reduced the flexibility of the system.

In the views project spatial representations were considered as cells with topological properties, which supports the topological reasoning required by the system [4]. Due to the use of quantitative reasoning mechanisms with a coordinate system lead to hand generation of spatial regions. This has limited the flexibility of the system. Development of spatial regions was automated by using qualitative spatial and spatio-temporal (s-t) reasoning methods in [5]. Here the use of qualitative spatial and s-t regions were limited to conceptual clustering of low level input data and has not gone to the extent of learning any rules to build models of the observed scene.

Description logic was used to learn scene semantics by analysing qualitative spatial and s-t relations in [6]. Due to the pre determination of the learnt relations the work cannot be generalized to learn in another situation. By exploiting spatial and s-t relations context specific rules are learnt in [7],[8]. Even though they have adopted a similar approach as ours the application of spatial and s-t relations was limited to identification of perceptual groups and the notion of presence and absence of objects. Further any qualitative object-object spatial or temporal relations were not analysed. We argue that cognitive systems that can be fully embedded in the environment should posses capabilities in representation and reasoning in such a manner as humans do.

3. Theoretical Foundations

Since spatial reasoning in our everyday interactions with the physical world is generally driven by a qualitative abstraction rather than using exact quantitative measures, there are various theories developed to address qualitative spatial and s-t representation and reasoning [9]. Since we account for the spatial extent of the objects we do not consider any point-set topological theories here. Hence Region Connection Calculus (RCC-8) theory [10] is exploited to determine the type of connection between any two regions and Allen's interval algebra [11] to account for temporal variations in object movement.

A Region Connection calculus (RCC-8)

The theory describes the primary relation of any two spatial entities in the form of connection. Therefore the basic dyadic relation is C(x,y), which reads as x connects with y. Based on the type of connection a set of eight jointly exhaustive and pair wise disjoint (JEPD) relations are derived as shown in fig.1.

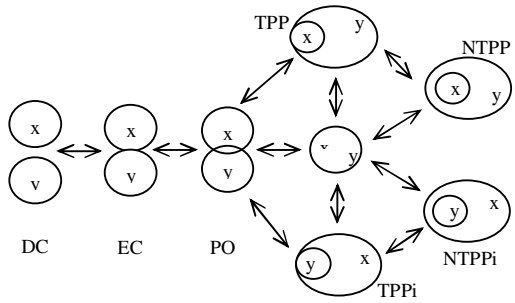

Fig. 1. Relations of RCC-8 calculus and their continuous transitions

The abbreviation DC (x, y) is read as x is disconnected from y. Similarly the other abbreviations are EC-externally connected, PO-partially overlaps, EQ-equal to, TPP-tangential proper part, TPPi-inverse of tangential proper part, NTPP-non tangential proper part and NTTPi- inverse of non tangential proper part.

Further this theory illustrates that when any two spatial entities move they can move sequentially only in adjacent locations. Our adaptation of RCC-8 theory to describe the type of connection between any two moving objects is described in section 5 *c*. To account for movement we employ Allen's interval algebra and an outline of the theory is given in the next section.

B Allen's Interval Calculus

Allen's interval calculus describes the temporal relation between any two moving objects. According to Allen's notion there are 13 JEPD relations, and for any two intervals exactly one of the relations holds as shown in fig.2.

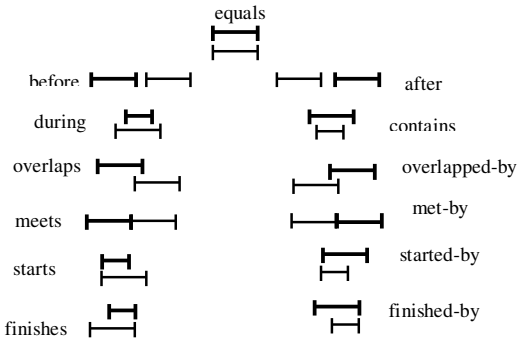

Fig. 2. Temporal Intervals in Allen's calculus

The Allen's relations are defined considering the end points of the intervals and allow movement only in adjacent time stamps. Thus this helps to anticipate the future as well as describing the relations of a third interval with the use of transitivity properties. In next section we will look at the approach taken to implement the proposed framework.

4. Approach

The real world scene is captured by a video capturing process in a frame-by-frame manner and converted to symbolic data using an attention mechanism [12]. We assume this is done before hand.

Initially the symbolic data set is clustered into perceptual groups based on a reference object. An object with salient features is often regarded as a reference object [1]. Some of the salient features are size, brightness, movement, etc.

Since we account only for indoor scenes of object movement the representing world can be regarded as a small-scale space. In small-scale spaces topological and orientation relations provides a restricted form of positional information, which describe the arrangement of the objects [1]. Orientation relations describe where objects are kept relative to one another and can be defined with three basic concepts, the primary object, the reference object and the frame of reference [13]. An intrinsic frame of reference is assumed by considering the characteristic direction of movement of the reference object. The object in which the position has to be determined is called the primary object.

The earlier described RCC-8 theory is used to identify the topological relations of the boundaries of the objects [1]. The theory is adopted in such a way to determine whether objects are apart from each other, any two objects touch each other or one object is on top of another object. Even though TPP, NTPP and their inverses are realistic situations in object movement we treat all of them as EQ. Because in laymen terms, those situations can be considered as situations, where, one object is on top of another object.

In situations where an object A is kept on top of object B a human knows intuitively that object B has to be kept prior to object A because vies versa is an unrealistic situation. The only possible way of learning such concepts for machines is by considering the time factor. Similarly we too account for the order of object placing by considering the Allen's interval calculus for time.

5. Design and Implementation

A dynamic scene of setting covers in a dinner table is used as a prototype model to implement the proposed framework. Fig. 3 is a snap shot of a one such captured scene.

Fig. 3. A Snap shot of the table setting scenario

The symbolic data set contain information about frame number, type of the object, object ID, center coordinates, the bounding box, similarity measure and information regarding whether the object is moving or not.
The design of the system is given in fig. 4. There are two main modules in the system namely Qualitative Knowledge (QK) module and the ILP module.

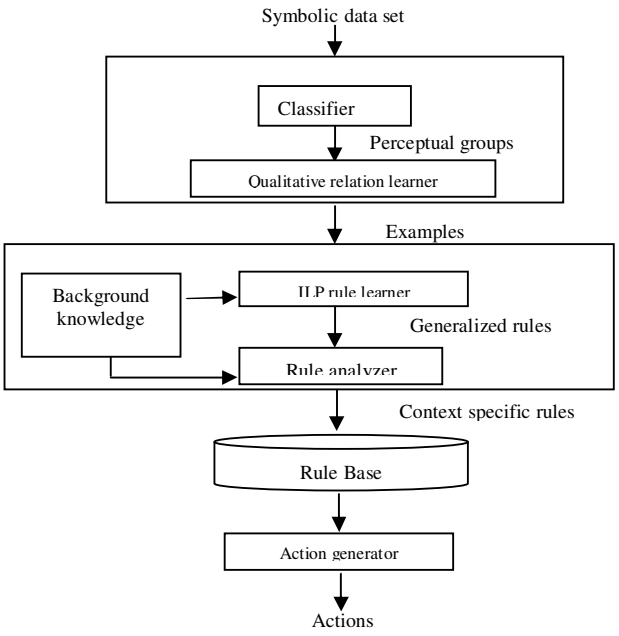

Fig. 4. Overall system design

The two modules are further sub divided into smaller modules with a specific task in hand and are explained below.

5.1. Identifying perceptual groups

To account for the dynamic nature of the scenes we assume non-monotonic reasoning methodology by assuming that new evidence can invalidate the past conclusions [14]. Accordingly the first appearing object is considered as the reference object and when a second object comes, the previously assumed reference object is not used any more. Then the larger object in size is considered as the reference object according to our adaptation. Therefore the reference object is calculated dynamically for each and every frame. The other objects that appear in the frame are clustered, based on a distance constraint of three times the bounding box of the reference object as the limiting distance for a single cluster [1]. The clustering that appears when the scene becomes static is considered as the final grouping of perceptual groups.

5.2. Relative orientation of objects

According to the prototype the largest object, *plate* is considered as the reference object. Based on the reference object eight distinct relations are used: *front* (F), *back* (B), *left* (L), *right*(R), *le*ft-back (LB), right-back (RB), left-front (LF) and right-front (RF) to determine the relative orientation of a primary object. In doing so, we employ a cardinal direction type of a grid system for the layout of the scenario [15]. The reference object is considered to be in the center tile and other primary objects that are in a single cluster fall in surrounding tiles as shown in fig. 5.

The bounding box of the tile that contains the reference object determines the size of the grid. X_{min} and X_{max} are the values of the X-axis of the left most corner and the right most corner of the bounding box respectively. So as Y_{min} and Y_{max}.

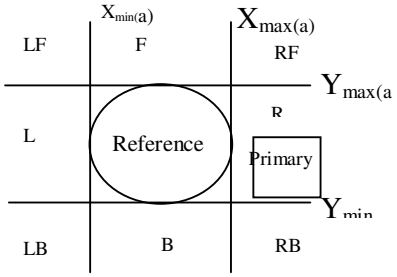

Fig. 5. Boundary line grid system

Based on this notation the orientation of a primary object (∅) can be determined by considering constraints such as:

$Front(\phi) = \{\langle x, y \rangle \mid X_{min}(r) \leq x \leq X_{max}(r) \wedge y \geq Y_{max}(r)\}$

$FrontRight(\phi) = \{\langle x, y \rangle \mid x \geq X_{max}(r) \wedge y \geq Y_{max}(r)\}$

Similarly the boundaries of other tiles too are determined.

It is noted that according to this notation sometimes one or more objects falls into the same tile. For example, according to our prototype model k*nife* and *spoon* both are on the right side of the plate. In such situations it is necessary to determine the relative orientation of the objects that fall in the same tile. Therefore, the same above explained mechanism is carried out for objects that fall with in the same tile and relative orientations are defined again. By doing so more finer orientation relations like *the knife and spoon are both right to the plate and knife is to the left of spoon* can be easily identified.

After deciding the orientation relations the next step is to identify the topological object-object relations. Topological connection between objects are identified by using RCC-8 relations.

The RCC-8 relations are identified by implementing the framework proposed in [16]. A finite set F of

cells are considered to identify the relations and the center coordinates of a primary object is checked against the bounding box values of the reference object. Relations on regions are defined as in fig 6.

DC (P,Q) $P \cap Q = \emptyset \wedge \forall x, y(x \in P \wedge y \in Q \rightarrow \neg A(x, y))$
EC (P,Q) $P \cap Q = \emptyset \wedge x, y(x \in P \wedge y \in Q \wedge A(x, y))$
PO (P,Q) $P \cap Q \neq \emptyset \wedge P \square Q \wedge \square P$
EQ (P,Q) $P = Q$
TPP (P,Q) $P \subseteq Q \wedge \exists x, y(x \in P \wedge y \notin Q \wedge A(x, y))$
NTPP (P,Q) $P \subseteq Q \wedge \forall x, y(x \in P \wedge A (x, y) \rightarrow y \in Q)$
TPP_i (P,Q) $Q \subseteq P \wedge \exists x, y(x \in Q \wedge y \notin P \wedge A (x, y))$
$NTPP_i$ (P,Q) $Q \subseteq P \wedge \forall x, y(x \in Q \wedge A (x, y) \rightarrow y \in P)$

Fig. 6. RCC relations on regions

Even though the resulting algorithm produces all eight relations we are particularly interested in DC, EC, PO and EQ. Because in real life we often come across situations where two objects are apart from each other, externally touching each other, and one object on top of the other object etc. Therefore the objects with an EQ type of connection are regarded as objects that are on top of each other. If the connection is PO then in 3 dimension, one object is being partially occluded by the other.

With the combination of topological and orientation relations the arrangement of the objects in a particular time point can be predicted. Since our prototype evolves in a time interval it is necessary to address the whole time period to determine the order of object placing.

5.3. Object-object time relations

Time factor plays an important role when there is an order restriction of placing some objects. Using Allen's calculus we account for the end points of time intervals to determine situations such as the end time interval of placing saucer has to end before the end of the time interval of placing the cup.

Time points with significant features such as change in object movement are noted. When such significant frames are encountered three frames before and after time point is taken into consideration. Even though we are mainly interested in moving objects we still account for objects that became stationary in the immediate past since the time registry of immediate past situations are more prominent.

Therefore by combining the time variations with positional information object arrangement of the visual scene is learnt. In the qualitative relation-learning module we identify only context specific relations but to reuse the learnt relations in another context we argue that learning relations only, is not sufficient thus we need to learn the rules for the learnt relations. A brief explanation about the ILP rule-learning module is given in the next section.

5.4. ILP module

The qualitative object-object spatial and temporal relations identified in the QK module are considered as input examples for the ILP module. To learn from examples ILP methodology is selected mainly because of the ability to handle symbolic data and due to the more human comprehensible nature of the output [17].

The ILP module comprises of two main components namely ILP rule learner and the rule analyser. The ILP rule leaner is implemented with the use of an off the shelf ILP package called PROGOL [18].

PROGOL generates logic programs in the form of hypotheses/rules in the light of given examples and background knowledge. In brief, rule-generating mechanism of PROGOL is as follows. For each positive example the most specific Horn clause is generated according to the user declared mode declarations. These mode declarations impose restrictions on generalisations. The generated hypothesis is the one that explains the highest number of examples.

The ILP rule learner is supported by relevant background knowledge, which guides the rule learning mechanism. This enhances the efficiency of the system because the rule generating mechanism is guided rather than exploiting a syntactically possible search place. Following is one such given example for learning orientation relation for the knife with respect to the plate.

```
time (t54).
orientation (rightof,[plate2, knife1],t54).
box(plate2,[ 51, 233, 144, 327 ], t54).
center(knife1, [150,280],t54).
move(plate2, 1], [knife1, 1],t54).
```

Bounding box coordinates of the reference object plate, center coordinates of the primary object knife and information about whether the objects are moving or not are given as background knowledge. Likewise rules are found for all orientation, topological and time relations. Since these are context specific generalised rules they can be globally used in any similar situation.

The rule analyser retains the rules that explain the highest number of examples and checks for features such as most frequently occurring rules, rules with any priority factor etc, and store in the rule base for future applications.

In summary, the system works as follows. Initially the symbolic data set is clustered into perceptual groups and qualitative object-object relations are derived in the QK module then these learnt relation examples are passed to the ILP module to generate corresponding rules. The generated rules are further analysed in the rule analyser to find out most appropriate rules that represents the considered world. As an application the system is capable of producing sketch images of arrangement of a given set of objects based on the rules learnt.

The QK module is implemented in Prolog while the ILP module is implemented in PROGOL, which is an ILP tool capable of generating rules that best explain a given set of examples. A detail explanation about the ILP rule-learning module is published in [19].

6. Results and Discussion

In this research we have developed a mechanism to adopt the natural rule learning ability of humans from visual scenes into cognitive vision systems. Thus we employed the hypothesis that in learning rules of visual scenes human tend to exploit object-object robust qualitative spatial and s-t relations. Further we incorporated human ability of learning from examples because observing an evolving scene itself is an example. The proposed framework is more suitable for learning rules from indoor scenarios of object movement.
In the prototype, only the size factor is considered in determining the reference object. Even though this works fine with our prototype scenario sometimes it may be necessary to consider another feature such as brightness.
The output of the QK module is a set of qualitative orientation, connection, and time relations at a particular time point. Following is an orientation relation learnt in the QK module.

orientation (rightof, [plate2, knife1], t54)
(1)

Relation (1) describes that the orientation of knife1 with respect to plate2 is rightof at time t54. Since this relation is highly context specific it cannot be used in another similar situation to determine the orientation relation between the plate and the knife. We overcome this limitation by learning a general rule using PROGOL. Hence the learnt rule is:

orientation(rightof,[plate,knife],A):-
box(plate,[B,C,D,E],A),center(knife,[F,G],A),
move([plate,H],[knife,I],A),F>=D,F>=B,G=<E.
(2)

In (2), orientation relation and the names of the objects are the only context specific information. Therefore unlike (1), the rule (2) can be applied in any similar situation to determine the orientation of knife with respect to plate. Similarly from the ILP rule learner, generalized rules are learnt for all the other context specific qualitative spatial and s-t relations. Therefore these rules impose a set of protocol rules that can be applied in a similar situation hence this improves the flexibility and the applicability of the system.

Here we do not consider time variations for objects with DS type of connection because usually there are no order restrictions in placing discrete objects in small-scale environments. But this may not be the case in large-scale spaces if the involved objects are comparatively different in sizes, then the larger object may have to be placed before the smaller one. If the type of connection is EQ or PO then the time variation has to be considered disregard to scale of the space. Because in 3D situations these connections are depicting objects on top of each other and objects that are occluding each other.
The generated rule set is further analysed in the rule analyser to identify the most appropriate rules and passed on to a rule base for storage for future applications. For evaluation the generated rule set is tested for soundness and completeness against a hand coded example set to test the ability of learning in the presence of noise. It is noted that there is an accuracy of about 70% of generating rules under noise.

7. Conclusion and Further work

We have developed a system that captures the natural rule learning ability of humans from visual scenes by exploiting object-object qualitative spatial and s-t relations. Our system is capable of dynamically identifying reference objects and generates perceptual clusters. Then the system learns qualitative object-object spatial and s-t relations and thereby generates context specific rules that can be applied in any similar situation to learn the arrangement of the objects in the observed scene. According to the 70% of accuracy rate against a perfect scenario we can conclude that learning rules of a visual scene predominantly by analysing qualitative spatial and s-t relations is an effective way of learning rules of a visual scene.

At present we are researching how to incorporate rules learnt from one scenario for reasoning in

another scenario as humans do in real life situations. Our work can have great impact on building autonomous agents that learn from human agents based on visual inputs and can be used as a model for training agents even in a time of scarcity of human experts such as in a disaster.

8. Acknowledgement

The first author gratefully acknowledge the financial assistance under reference LKCN-2003-93 received from Commonwealth Scholarship Commission, United Kingdom, Professor A.G. Cohn for his guidance and supervision during the period at University of Leeds, the guidance and support given by the CogVis team at the University of Leeds and the CogVis team at the University of Hamburg, Germany for providing the data.

9. References

[1] D. Hernandez, Qualitative Representation of Spatial Knowledge, Number 804, In Lecture Notes in Artificial Intelligence, Springer-Verlag 1994.

[2] Technical Annex 1: Cognitive Vision Systems, Description of Work, IST-2000-29375, 2001.

[3] N. I. Badler, Temporal Scene Analysis: Conceptual Descriptions of Object Movements, Technical report no. 80, University of Tronto, Ontario, Canada (1975).

[4] R. J. Howarth, H. Buxton,), "An Analogical Representation of Space and Time," Image and Vision Computing 10(7), 1992, pp. 467-478.

[5] J. Fernyhough, A. G. Cohn, D. C. Hogg, "Constructing Qualitative Event Models Automatically from Video Input," Image and Vision Computing, 18, 2000, pp. 81-103.

[6] B. Neumann, "Conceptual Framework for High Level Vision," Technical report FBI-HH-B-241/02, FB, Informatik, Univeritat Hamburg, 2002, June.

[7] D. Magee, C. Needham, P. Santos, A. G. Cohn, D. Hogg, "Autonomous Learning for a Cognitive Agent using Continuous Models and Inductive Logic Programming from Audio-Visual Inputs," Proceedings of the AAAI workshop on Anchoring Symbols to Sensor Data 2004.

[8] P. Santos, D. Magee, A.G. Cohn, ,"Looking for Logic in Vision, " Proc. Eleventh Workshop on Automated Reasoning, 2004, pp. 61-62.

[9] A. G. Cohn, S. M. Hazarika, "Qualitative Spatial Representation and Reasoning: An Overview," Fundamenta Informatica 46(1-2), 2001, pp. 2-32 IOS Press.

[10] A. G. Cohn, B. Bennett, J. Goday.,N. M. Goots N. M., 1997, "Qualitative Spatial Representation and Reasoning with the Region Connection Calculus," Geoinformatica, 1, Kluwer academic publishers," pp 1- 44.

[11] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," Communications of the ACM 26 (11), 1983, pp. 832-843.

[12] S. Hongeng S., , 'Unsupervised Learning of Multi Object Event Classes, Technical Report FBI-HH-B-257/04, 2004 Informatik, University of Hamburg.

[13] E. Clementini, P. Di. Felice, D. Hernandez, "Qualitative Representation of Positional Information," Artificial Intelligence, 95, pp.317-356.

[14] E. Davis, "Representations of Commonsense Knowledge," editor, Brachman R.J., Morgan Kaufmann publishers, Inc, San Mateo, California, 1990.

[15] A. L. Kor, B. Bennett, "Composition for Cardinal Directions by Decomposing Horizontal and Vertical Constraints," workshop in Foundations and Applications of Spatio-Temporal Reasoning, AAAI Spring Symposium, 2003, pp. 39-45.

[16] A. Galton, "Deriving Qualitative Spatial Information from Low- Level Data" unpublished.

[17] T. M. Mitchell, Machine Learning, Macgraw Hill Company. 1997.

[18] S. Muggleton, J. Firth, "Cprogol4.4; A Tutorial Introduction," In Relational Data Mining, editors, Dzeroski S., Lavarac N., Springer Verlag, 2001, pp. 160-188.

[19] D.D.M. Ranasinghe, A. S. Karunananda, 2006, "Qualitative Knowledge Driven Approach to Inductive Logic Programming," accepted for publication at ICIIS 2006.

Panel of Reviewers

- Dr. Thrishantha Nanayakkara, University of Moratuwa
- Dr. Nalin Wickramaarachchi, University of Moratuwa
- Prof. A.S. Karunananda, University of Moratuwa
- Dr. (Ms.) Uditha Rathnayake, Open University of Sri Lanka

Published by Sri Lanka Association for Artificial Intelligence, Sri Lanka

Printed by Tharanjee Printers, Nawinna, Maharagama. Tel: 2804773