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.

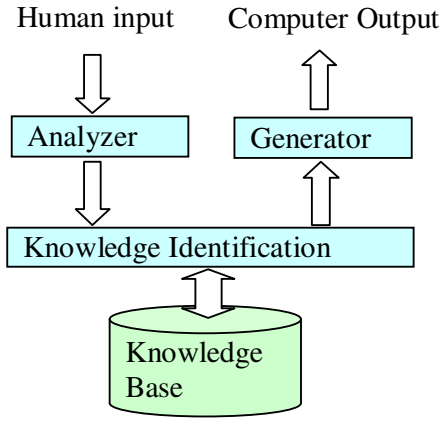

Fig 1: Overview of a Chatbot System

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].

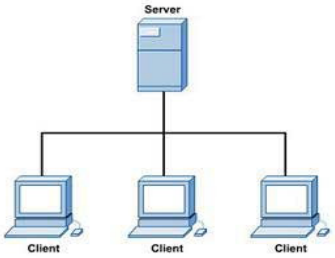

Fig 2: Client-server architecture of the Sinhala chatbot

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.

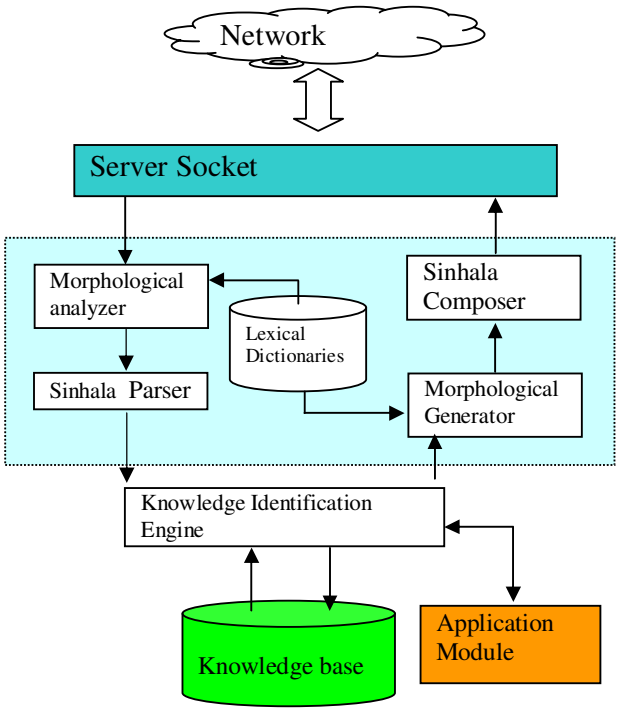

Fig 3: Server side design of the Sinhala chatbot System

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.

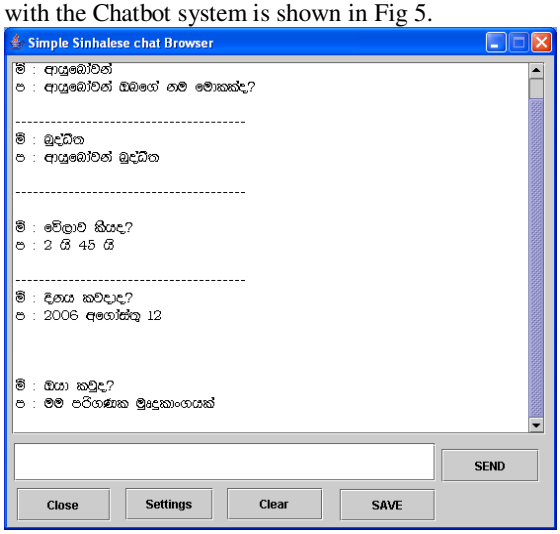

Fig 4: User interface of the chatbot system

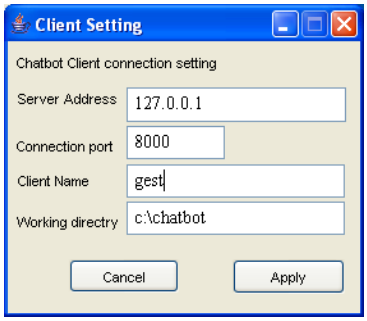

Fig 5: Connection setting of the chatbot system

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

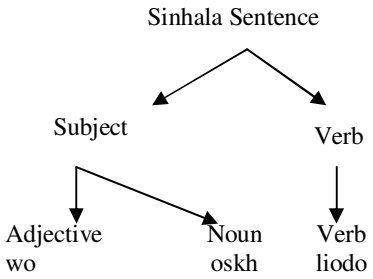

Fig 6: Parser tree for given Sinhala sentence

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