14

POONGKUZHALI - An Intelligent Tamil Chatterbot

T. Kalaiyarasi, Ranjani Parthasarathi, T.V. Geetha

Resource Centre for Indian Language Technology Solutions – Tamil,
School of Computer Science and Engineering, Anna University
Chennai – 600 025, India.

E Mail {kalaivas@yahoo.com, tvgeedir@cs.annauniv.edu, rp@annauniv.edu}

Abstract:

A Chatterbot is a tool for Natural language communication between man and machine. Poongkuzhali is a chatterbot that simulates human conversation through Artificial Intelligence – a program to chat with the system in Tamil. A question or a statement is taken as input from the user. The function of the system is to generate an appropriate response, based on the context of the input. The user can choose any existing topic for conversation and ask in Tamil. Provision for input in transliterated form is available. The system identifies the minimal context of the input as to what the user is trying to ask, no matter in what way the user frames his question. This is done using a set of decomposition rules. The response is then formed using a set of reassembly rules that reside in the knowledge base. This response is then reframed to match the way in which the user had framed his question. Currently there are two domains available. More can be added. It provides a separate interface for adding domains and for updating the knowledge base

1. INTRODUCTION

1.1 What is a Chatterbot?

A robot for chat is called a Chatterbot. It is a software developed for the purpose of allowing the user to chat with the system. A chatterbot is an AI program that simulates human conversation and allows for Natural Language communication between man and machine.

1.2 History of Chatterbot:

A chatterbot is a computer program for simulating conversation between a human and a machine. The user inputs a question or statement of any kind, and the chatterbot replies, just as a person would (using its own version of logic!). Chatterbots try to create the illusion that an authentic exchange is taking place between two person's thinking or between two living entities.

The origins of chatterbots can be traced back to 1950, when the British mathematician Alan Turing (http://www.turing.org.uk/turing) famously asked the question: "Can machines think?" It was a good question, and many people have since spent considerable amount of time and effort in trying to prove that the answer is 'yes'. Researchers in artificial intelligence [5] have devoted much time and effort in trying to understand human cognitive capacities and adapt them to machines. Chatterbots represent just one aspect of this research.

The first chatterbot, named Eliza (http://web.mit.edu/STS001/www/Team7/eliza.html), appeared all the way back in 1966. Eliza was created by Dr. Joseph Weizenbaum of the Massachusetts Institute of Technology, and was intended to resemble a Rogerian psychologist [2]. When a human spoke to Eliza, she returned the sentence in the form of a question, thus inviting the user to give further explanation, ad infinitum. Not exactly a high level of conversation, but nonetheless ingenious and sufficiently 'intelligent' to cause confusion at a time when people were not used to interacting with computers.

1.3. Working of Chatterbots:

Chatterbots, are pieces of software that use natural language processing, neural networks and fuzzy logic to achieve its end. Using natural language processing, the program attempts to identify what a user says - the input - then applies a variety of different methods, from pattern matching and keyword identification to neural networks and fuzzy logic, to produce an appropriate answer. No chatterbot can claim to have human capacities of reasoning and deduction, and is by no means able to reproduce all the complexity of the human brain yet.

How intelligent the intelligence of chatterbots will depend on the technology on which the chatterbot is based , the quality of the knowledge base - the brain - that has been developed by the botmaster, what you expect from them.

Developers have found a number of solutions, which enable chatterbots to imitate human conversational ability, based upon a range of tricks to avoid difficult questions and stick to areas that they can handle, which in some ways is a proof of intelligence in itself. These include, but are not limited to making controversial statements to provoke a response, agreeing with the user, rather than being non-committal, repeating user input in their answers to make it seem as if they are following the conversation, remembering and reusing past topics of conversation, changing the topic when they don't understand, being random and abusive, just as humans are.

1.4 Need for a Chatterbot in Tamil

There isn't much scope for a native user who knows Tamil alone, to use the computer and learn about emerging technologies like the Internet. In such cases, a technical chatterbot such as Poongkuzhali comes in handy. It provides educative entertainment in such a way that it sustains the interest of the user.

This paper presents the details of Poongkuzhali - a chatterbot in Tamil.

This paper is organized as follows: Section 2 discusses about existing chatterbots. Section 3 describes the proposed system and highlights the technical issues. Section 4 discusses about implementation. Section 5 presents the results and performance and Section 6 concludes the paper.

2. EXISTING CHATTERBOTS

A number of chatterbots are available for free download from the net, in a number of languages. Eliza, Parry, Arthur, Simon Laven, MonkeyMind and Agentland [5] are some of the chatterbots that exist in English and German languages. Most of them are self-help programs aimed at creating inner peace for the user. Eliza was originally developed as an expert system in psychotherapy [2].

The Eliza is very useful for psychoanalysis. Responses are triggered with the intention of bringing out the thoughts of the user, right from his subconscious mind. History data is effectively used to analyze the user's mind. However the system is interactive, but not very informative. It is not very good at initiating a dialogue. The main aim is to get as much information about the user as possible.

3. PROPOSED SYSTEM

The witty Tamil chatterbot, Poongkuzhali, is generic and will be able to chat about any technical topic for which a knowledge base is given. The user is free to frame his question in any way - Poongkuzhali will decipher and understand the question, no matter what grammatical form it has. She can handle any question pertaining to a given domain, ranging from simple definitions to complex queries. When the user is at a loss, the system will initiate a dialogue. Although each input question from the user is processed separately, history data is maintained such that references to previous questions can be handled. Also, she will generate a response in the same voice – active or passive – as found in the user's question.

3.1 TECHNICAL ISSUES

The important technical problems involved are:

identification of key words in the question.

discovery of minimal context in the question.

handling Tamil

generic design

construction of Knowledge Base

Of these the first two issues are for the core of the processing involved in Poonguzhali.

3.2 Identification of keywords in the question:

The keywords are identified based on certain priorities that are assigned to the words. The input is split into words, as a word is the smallest unit that is processed. The input sentence is scanned from left to right. Each word is looked up in a dictionary of keywords. Each keyword has an associated weight or priority. A list of all keywords identified in the input is made. This keyword list is sorted on the descending order of weights. The procedure is sensitive to the rank of a keyword in that it will abandon a keyword already found in the left-to-right scan of the text in favor of one having a higher rank.

3.3 Discovery of Minimal Context in the question:

Once the keywords are identified, the next step is to establish the link between them. This is done by choosing the appropriate set of decomposition rules. Each keyword is associated with a set of other keywords linked to it, forming a decomposition rule. Similarly, each decomposition rule is associated with a set of reassembly rules. These reassembly rules provide various options for generating the answer. Keywords and their associated rules are placed on a list. The basic format of a typical key list is the following:

$$
\begin{array}{l}
(K\ ((D_1)\ (R_{1,1})\ (R_{1,2})\ \cdots\ (R_{1,m1})) \\
\quad ((D_2)\ (R_{2,1})\ (R_{2,2})\ \cdots\ (R_{2,m2})) \\
\quad\ \ .\qquad\qquad . \\
\quad\ \ .\qquad\qquad . \\
\quad ((D_n)\ (R_{n,1})\ (R_{n,2})\ \cdots\ (R_{n,mn})))
\end{array}
$$

where K is the keyword, D_i the ith decomposition rule associated with K and $R_{i,j}$ the jth reassembly rule associated with the ith decomposition rule. By applying the appropriate transformation rules to the highest-weighed keyword, we get the minimal context. Once the minimal context is identified, the associated reassembly rules are used to generate the answer.

3.4 Handling Tamil

There are a number of issues related to the inherent nature of the Tamil language that need to be addressed. Tamil is an extensive free word order language. Although the general construct is such that the verb occurs as the last word of a sentence, there is no hard-and-fast rule regarding the syntax in most cases. Unlike English, a single word in Tamil offers a lot of information about the context. For example, nouns give information about cases, plurals, obliques and clitics. Similarly, verbs are often found occurring along with verbal participles, infinitives, auxiliaries, clitics, negatives, tense markers and gender markers. Hence a morphological analyzer is made use of to obtain all the above information.

3.5 Generic Design

The software has been made generic such that it can handle any technical domain without changes in the code. This generic nature is achieved through the file of non-technical terms that contain words common to all technical domains. This file needs to be changed only in exceptional cases where certain general words take special meanings in a particular domain. The same holds good with respect to the priorities of these words. Technical issues pertaining to language handling and response generation remain the same across domains

3.6 Knowledge Base

The discussion being technical in nature, the response generation depends on the Knowledge Base to a great extent. The Knowledge Base consists of 3 files – each containing the Domain Knowledge, technical terms and non-technical terms respectively. The file of non-technical terms is common to all domains. It contains non-technical terms along with their priorities and decomposition rules. The file of technical terms contains the extensive list of technical terms pertaining to the given domain. The file of Domain Knowledge contains the minimal context and their corresponding basic answers as reassembly rules. Creation of the Knowledge Base requires domain expertise.

4. Implementation

The system consists of the following modules, as shown in figure 1:

Pre-processing

Decomposition

Reassembly

Response Generation

Transliteration keyboard driver.

Knowledge Base Creation.

Each of these modules is described below.

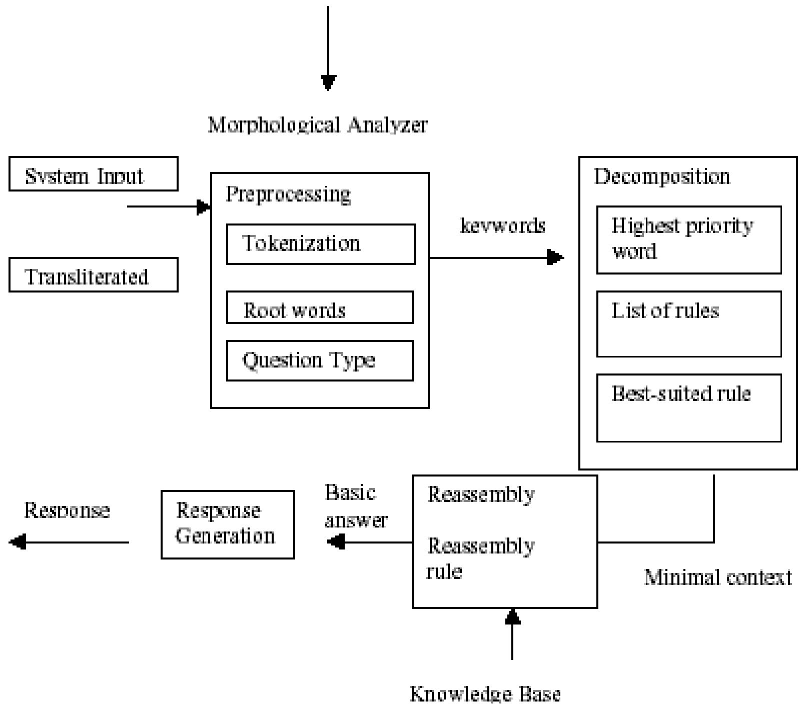


Figure 1. Block Diagram of the System

4.1 Pre-processing

The sentence is broken down into words, separated by spaces and other delimiters. This is referred to as tokenization. These words are stored in an array and all further processing takes place on these words.

The Morphological Analyzer is invoked in order to get the root word from each word in the input sentence. These root words are stored in an array. The result of morphological analysis is used to set the question type. Also, the technical terms are identified by consulting the file of technical terms corresponding to the domain. The history data is set appropriately in the pre-processing phase.

4.2 Decomposition

For each keyword, referring to the file of non-technical terms a list of decomposition rules is created. If more than one rule is available, the best rule corresponding to the highest priority word, that matches the user's input, is identified. Using this rule, the minimal context of the user's question is obtained. If no match is found, the next highest priority keyword is considered for processing.

The decomposition module takes as input, the word that has been analyzed. The flags that give information about the question type are set. The different types of queries such as definition handling, compound technical terms, alternate explanation for the same question, pronoun references, Yes/No type questions and the system-initiated dialogue are handled.

4.2.1.Definition Handling:

If the user asks for a definition of some technical terms, processing requirements are minimal. The term has to be identified and referred to in the knowledge base.

4.2.2.Compound Technical Terms:

Technical terms consisting of more than one word are identified and considered as a single technical term. For example, the term வலைப் பின்னல் விதிமுறை is taken to be a single compound technical term, although it consists of 3 separate words.

4.2.3.Alternate Explanation:

The software offers the facility of an optional alternate explanation to the user, in case he doesn't understand the earlier answer given by the system.

4.2.4.Pronoun References:

The user may use pronouns to refer to previously discussed issues. In such cases, the system uses history data to make such references. For example, the pronoun "இது" in the second question is identified as a reference to the term "இணையம்" in the previous question and the appropriate answer is generated. This is done using the history data that has been stored earlier.

4.2.5.Yes/No Type questions:

The user might ask questions whose immediate answer will be a yes or a no. The system gives this answer first and then comes up with an explanation, if necessary.

4.2.6.System-initiated dialogue:

In the absence of user input, the system initiates a dialogue based on history data.

The technical terms found in the input are identified and stored separately, using the knowledge available in the file of technical terms pertaining to the given domain. Then, a set of decomposition rules is applied and this gives the minimal context as to what the user is asking.

4.3 Reassembly

For the selected decomposition rule, the list of reassembly patterns is created from the Knowledge Base. There may be several reassembly patterns and one of them is selected at random. This gives the basic answer that can be used for response generation.

The reassembly module uses the context information obtained by decomposition and searches for this context in the Knowledge Base. If the context is available, the answer is retrieved and used to generate the response. If the Knowledge Base does not contain this context, the user is prompted for more input.

4.4 Response Generation

A set of post-substitutions based on the question type is applied on the basic answer that has been reassembled earlier. This generates the response, which is displayed. The input information whether the question is in active or passive voice is used in response generation, in addition to the basic answer obtained by reassembly.

In the absence of user input, the system initiates a dialogue based on history data. The technical term used in the previous question-answer pair is utilized for beginning a conversation. In case we get out of focus input, i.e., the user's question is irrelevant to the topic of discussion, the response is such that the user is asked to come back to the topic of conversation.

4.5 Transliterator

The transliteration software is aimed at allowing the user to type out Tamil data using a common keyboard. The symbols are phonetically mapped from English to Tamil. There are a number of users who are required to type out quite an amount of data in Tamil, but are untrained in Tamil typewriting. To fulfill the requirement of such users, we need a Transliterator that takes input in English and gives out its phonetic Tamil equivalent. This generic transliterator can be used in any java application instead of a JTextField, in order to provide for transliterated input. It is currently applied in the Tamil Chatterbot, Tamil word processor, Tamil search engine and other related applications.

Knowledge Base Creation:

A special interface has been provided to add new information to the knowledge base for the existing domains. Additions of new domains are also possible through this interface. The Knowledge Base consists of 3 files – each containing the Domain Knowledge, technical terms and non-technical terms respectively.

Domain Knowledge File Format:

The Knowledge Base has the following general format as shown in the figure.2:

Each entry consists of four fields such as context, number of reassembly patterns

available, reassembly patterns and alternate answer.

$இணையம்$ is the context, and is separated by the delimiter $.

n4 is the number of reassembly patterns available.

The' ~ 'delimiter separates the reassembly patterns.

The' # 'delimiter separates the alternate answer.

```
$இணையம்$$இன்டர்நெட்$n4~இணையம் என்பது கணிப்பொறிகளின் மிகப் பெரிய, மிக
விரிந்த இணைப்பாகும்.~இணையம் என்பது உலகின் எந்தப் பகுதியிலும் உள்ள
விபரங்களை வீட்டிலிருந்தபடியே பெற்றுக்கொள்ள உதவும் சாதனமாகும்.~இணையம்
என்பது ஆயிரக்கணக்கான வலைப்பின்னல்களின் ஒருங்கிணைப்பாகும்.~இணையம்
என்பது கணிப்பொறிகளின் உலகளாவிய வலை போன்ற இணைப்பாகும்.#உலகின் பல
இடங்களிலும் உள்ள கணிப்பொறிகளின் இணைப்பே இணையம் ஆகும்.%
```

Figure 2 A section of the Knowledge Base for the domain 'Internet'

4.6.2 File of Non-Technical Terms File Format:

This file is common to all technical domains. Its general format is shown in the figure 3:

Each entry consists of six fields. They are word number, word and its synonyms, word number (the redundancy is deliberately introduced to minimize file access), word (this word alone is considered even if one of its synonyms occur), priority, and list of links.

1-word number

$பாதுகாப்பு$$பாதுகாப்ப்$-word and its synonyms

n1-Word number given to avoid redundancy wபாதுகாப்பு -Word (this word alone is considered even if one of its synonyms occur)

p5- Priority

p~3,4- list of links

```
1.$பாதுகாப்பு$$பாதுகாப்ப்$n1wபாதுகாப்புp5p~3,4#
2.$சிற$$சிறந்த$$முக்கி$$அத்தியாவசி$n2wசிறந்தp6p~6~1#
```

Figure 3: A section of the file of Non-technical terms.

4.6.3 File of Technical Terms file format:

The file of technical terms contains the set of technical terms pertaining to the given domain. They are separated by delimiters $ as in figure 4.

```
$இணையம்$
$இன்டர்நெட்$
$வலைப்பின்னல்$
$நெட்வொர்க்$
$தனிவலைப்பின்னல்$
```

Figure 4: A section of the file of Technical terms.

5.Results and Performance:

In an application like chat, response time is a very critical factor. To enhance the perfor-mance, file accesses and I/O processes have been reduced drastically. Each file is accessed just once when answering a given question. The time complexity is of the order of n - where n is the number of words in the input.

5.1 Testing:

The software has been carefully tested module by module and as an integral system. The test data has been meticulously chosen to unravel errors and have been tested by many users.

Definition Handling, Compound Technical terms, Alternate Explanation, Pronoun references, Yes/No Type Questions, System-Initiated Dialogue are the cases tested. Figure 5 shows the main interface of Poongkuzhali. and explains the Inaiyam interface with all test cases that have been handled and Figure 6 shows the interface of domain addition and for knowledge base updation.

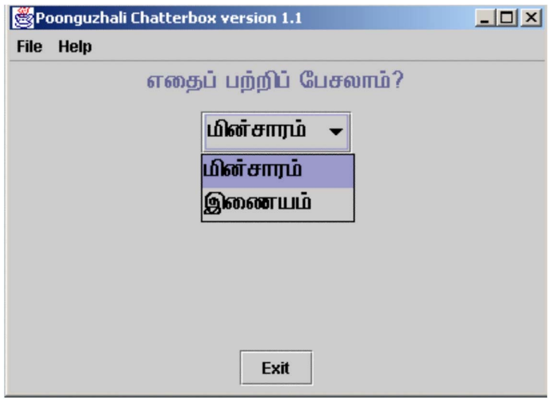
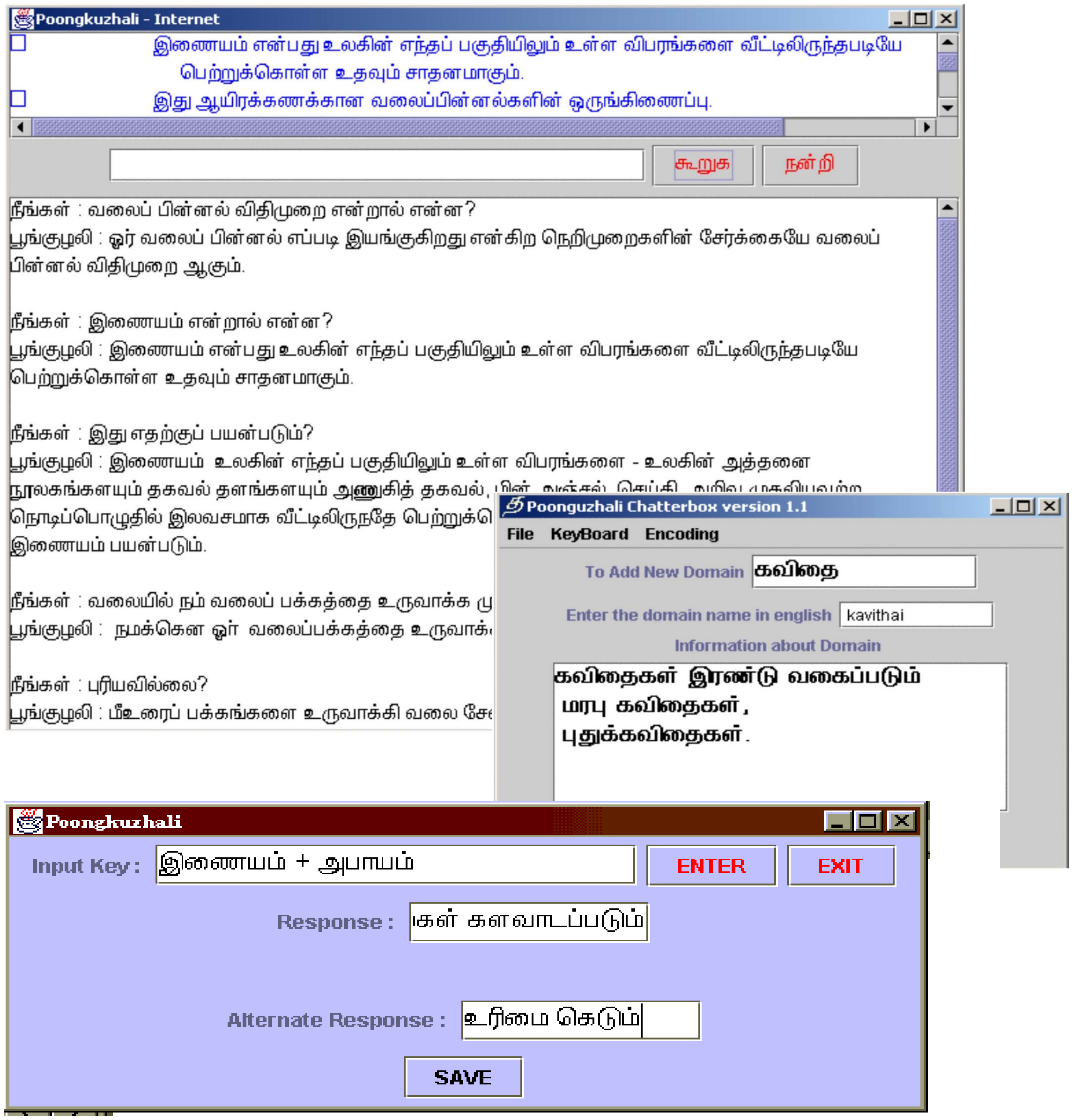


Figure 5 Poongkuzhali interface and Inaiyam interface with queries



இணையம் என்பது உலகின் எந்தப் பகுதியிலும் உள்ள விபரங்களை வீட்டிலிருந்தபடியே பெற்றுக்கொள்ள உதவும் சாதனமாகும்.

இது ஆயிரக்கணக்கான வலைப்பின்னல்களின் ஒருங்கிணைப்பு.

நீங்கள் : வலைப் பின்னல் விதிமுறை என்றால் என்ன?
பூங்குழலி : ஓர் வலைப் பின்னல் எப்படி இயங்குகிறது என்கிற நெறிமுறைகளின் சேர்க்கையே வலைப் பின்னல் விதிமுறை ஆகும்.

நீங்கள் : இணையம் என்றால் என்ன?
பூங்குழலி : இணையம் என்பது உலகின் எந்தப் பகுதியிலும் உள்ள விபரங்களை வீட்டிலிருந்தபடியே பெற்றுக்கொள்ள உதவும் சாதனமாகும்.

நீங்கள் : இது எதற்குப் பயன்படும்?
பூங்குழலி : இணையம் உலகின் எந்தப் பகுதியிலும் உள்ள விபரங்களை - உலகின் அத்தனை நூலகங்களையும் தகவல் தளங்களையும் அணுகித் தகவல்,
நொடிப்பொழுதில் இலவசமாக வீட்டிலிருநதே பெற்றுக்
இணையம் பயன்படும்.

நீங்கள் : வலையில் நம் வலைப் பக்கத்தை உருவாக்க
பூங்குழலி : நமக்கென ஓர் வலைப்பக்கத்தை உருவாக்

நீங்கள் : புரியவில்லை?
பூங்குழலி : மீஉரைப் பக்கங்களை உருவாக்கி வலை சே

To Add New Domain: கவிதை

Enter the domain name in english: kavithai

Information about Domain:
கவிதைகள் இரண்டு வகைப்படும்
மரபு கவிதைகள்,
புதுக்கவிதைகள்.

Input Key : இணையம் + அபாயம்

Response : கள் களவாடப்படும்

Alternate Response : உரிமை கெடும்

6. Conclusion

6.1 Achievements:

A versatile chat software that is generic to any technical domain, has been developed. It enables a native Tamilian to learn about technical topics in his own mother tongue. It makes learning a pleasure, catering to users with varied skill sets. Addition of many domains has been made possible.

6.2 Future Enhancements:

Since the software depends on the Knowledge Base to a great extent, Knowledge Base automation will be a great help to the system. Voice enabling will also make it user-friendly. Advanced Grammar Handling capabilities can be incorporated. Apart from the user asking questions to the system, the system can also be made to ask questions to the user. The system can test the knowledge of the user in a particular domain, by asking questions and evaluating the user’s performance.

7. References

1. Roger S. Pressman (1992) "Software Engineering - A Practitioner's Approach", Third Edition, McGraw Hill International. Joseph Weizenbaum (1966)
2. “ELIZA - A Computer Program For the Study of Natural Language Communication Between Man and Machine”, Massachusetts Institute of Technology.
3. சுஜாதா, (2001), "வீட்டுக்குள் வரும் உலகம் – இன்டர்நெட்", டிஷ்நெட் டிஎஸ்எல் லிமிடெட்
4. மணவை முஸ்தபா, (1999), "கணினி கலைச்சொல் களஞ்சிய அகராதி", மணவை பப்ளிகேஷன்
5. www.botspot.com