A study of Agent Network Technology to Natural Language User Interface

Khine Thida Myint, Mu Mu Myint
University of Computer Studies, Yangon
khinethidamyint@gmail.com

Abstract

Natural Language Processing (NLP) refers to communicating with a computer in Natural Language (such as English) rather than using special commands. With natural language processing, it is much easier to communicate with computer. Natural Language Interfaces (NLI) are used to simplify and improve the communication between application program and the user. This work is a study of an agent-oriented software toolkit that supports the domain-specific Natural Interaction. In this study, it is focused on how we can develop a Natural Language Interface based on agent technology. Since, an agent can process requests either directly or by combining its processing with results produced by other agents, we use the agents which are wired together in a network (hyperstructure). For case study, we implement the Cyber Home Finding System for Yangon Division as the specific domain.

Keywords: Natural Language Processing (NLP), Natural Language Interface (NLI), Natural Interaction, agent, agent-oriented software

1. Introduction

Language is the fundamental means of communication for human beings. Natural Language Processing (NLP) is one of the most important researches carried out in the world of Artificial Intelligence.[3] NLP is the use of computers to process written and spoken Language for some practical useful purpose such as to translate languages, to get information from the web on the text data banks, to carry on conversation with machines, to improve the communication between the application program and the user. Therefore, the application user can operate the application program with Natural Langauage without the knowledge of special artificial communication language.[3][2]

Agent is a fundamental unit of software and now it have found in many application domains such as Business Organization, Information Retrieval and Management, Distributed sensing, E-commerences and so on.[2] The agent can process requests either directly or by combining its process with results produced by others. And also it can be a network or hyperstructure and can operate by passing requests from agents to agent. For these reasons, we proposed an agent based Natural language user interface for the application user. In this study, the agent can accept the input sentence by the Natural Language.

This paper consists of six sections. Section 2 is the related work of the study domain. Section3 describes about the Natural Language Processing (NLP), which is a wide spread part of Artificial Intelligence, looking for utilization of all in computer both as a communication between humans and computer. The next section, section4 describes the Agent-oriented Technology. The section5 presents the design and implementation of the proposed **Agent** Based Natural Language User Interface for Cyber Home Finding System. And finally, the section6 describes the conclusion, benefits of the system and limitations of the system.

2. Related Works

The work of Burstein, F., Zaslavsky, A. and Arora, N [1] was to the highly dynamic and variable context of healthcare emergency decision-support domain by applying agent technology. They presented an implementation of the proposed agent based architecture, which was based on the specific functional and non-functional application requirements set out based on thorough analysis of literature.

The paper [2] was presented the great value in offering the Dejima Direct Platform for natural interaction solutions to application users. In this paper, they discussed the features, benefits and challenges associated with natural language and present an agent-oriented software toolkit that supports the development of domain-specific natural interaction. Tin Mar New (M.C.Sc.(Thesis), 1998) had presented how to analyse the grammer of simple English sentences by

performing syntatic analysis.[5] Tinzar Gyaw (M.C.Sc.(Thesis), 1999) was developed Natural Language User Interface on a simple well organized Knowledge Base for Myanmar Writers and Journalists.[6] The study of Nant Kyi Let Win (M.C.Sc.(Thesis), 2003) was an implementation Natural Language Processing System with the case study block world system. In her work, the system could accept Natural Language Input Command for Block world retrieval (such as circle, rectangle, square,..) and the output was the action which was equivalent with the command.[7]

The work of [4] had proposed a multi-agent architecture, called ISAME and designed for intelligent information from heterogeneous distributed sources. Their work, ISAME constituted a virtual library that supplies the agents with a simplified access to a set of dynamic information sources available under electronic formats, as well as services for facilitating and optimizing. In this study, we focused on how we can develop Natural Language Interface to integrate with an existing application with the help of agent technology. For case study, we implement a portable Natural Language user interface for Cyber Home Finding System as specific domain.

3. Natural Language Processing

The most common way that people communicate is by speaking or writing in one of the "natural" languages, like English, French or Chinese. On other hand, computer programming languages seem awkward to humans. These "artificial" languages are designed so that the sentences have a rigid format, or syntax, making it easier for compiler to parse a program and convert it into the proper sequence of computer instruction.

Natural Language Processing (NLP) is a wide spread part of Artificial Intelligence, looking for utilization of all in computer both as a communication between humans and computer.[3] Therefore, Natural Language is the fundamental means of communication for human beings and the NLP is the use of computers to process written and spoken language to translate languages and finally try to get information from the data bank.[1]

3.1Natural Interaction

Natural interaction refers to the way, it can communicate with each other. It is the words and the combination of those words into larger meaningful structures i.e. the words, sentences and paragraphs. The language is essentially the same whether it can be typed the words, write them out by hand or said them aloud. Most of the interaction with people involves using natural language (e.g., English).[3]

3.2 Natural Language Interface

It is used to simplify and improve the communication between the application and it allows the user to operate with natural language. Domain-specific Natural Language Interface can provide an impressive degree of accuracy, measured as the percentage of requests that translate into the operation intended by the user. It can also be extended to support additional vocabulary and additional application functions.[1][3]

3.3 Understanding Language: Top Down & Bottom Up

The most common approach to natural interaction is to start at the top: with the language. Natural interaction systems begin with the components of the language -- the words -- and analyze text in terms of the rules of the language: parts of speech, the structure of sentences and so on. [3]A good top down natural interaction system requires a large dictionary, a thesaurus, and a full understanding of the ways people create sentences. It also requires a great deal of general cultural knowledge, including the (often elusive) relationship between the meaning of idiomatic expressions like "he bought the farm" and the literal meanings of the words themselves.

An alternative is to consider language from the bottom up, starting with the tasks to be performed and considering all the different ways potential users might phrase their wishes.[2][8]

4. Agents

The term agent describes a software abstraction, an idea or a concept, similar to OOP terms such as methods, functions and objects. The concept of an agent provides a convenient and powerful way to describe a complex software entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks on behalf of its user. But unlike objects, an agent is defined in terms of its behavior.[7] An agent processes requests either directly or by combining its processing with results produced by other agents.[1] Agents can be wired together in a network or hyperstructure. [2]

4.1 Agent Network (Hyperstructure)

This network structure defines the communication paths between agents, which in turn determines the way agents get requests and provide responses. The hyperstructure can be thought of as a basic tree layout with one enhancement: a node. Nodes can connect to multiple nodes above it in the tree. The agent network operates by passing requests from agent to agent. A request begins at the root of the tree and flows down (downchain) to other agents. Agents examine the request and decide for themselves whether they have anything to contribute. Reponses flow back upchain using the same message paths as the request. The hyperstructure permits an agent to have multiple upchain connections. In such cases the downchain agent will receive the same request from every agent above it. It will only process the request once, however, and will send the same response to all of its upchain agents.[2]

5. The proposed Agent Network Application Architecture

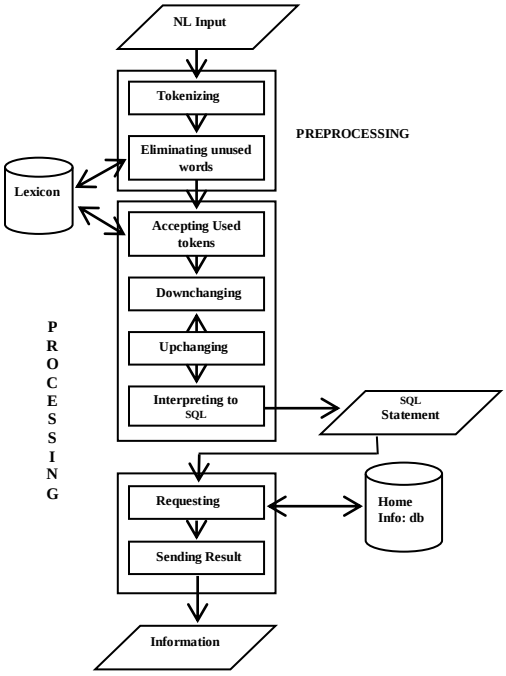


Figure1. Flow Chart of the Proposed System

The proposed system is composed with two main components: preprocessing component and processing component as shown in Figure1. These two components are accomplished by its four participant agents.

The tasks of preprocessing components are tokenizing, eliminating unused tokens and supplying the used tokens to processing phase.

The processing component is begun that used tokens of the user requests arrive at the top level agent as if by magic, that the command generated by the network somehow causes something useful to happen and that the result of that something useful makes its way back to the user. The following Figure2 represents the architecture of a server-side application with a natural interaction interface.

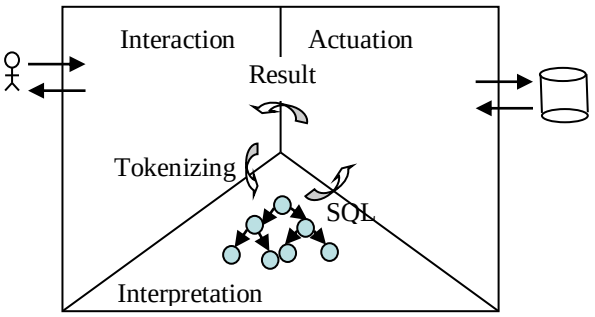


Figure2. The proposed Agent Network
Application Architecture

5.1 The Participant Agents

In this system, there are four participant; interaction agent, user agent, interpretation agent and actuation agent.

Interaction Agent: It can receive the user natural language request as an input and pass the request to user agent and also receive the response from the user agent and then send the information to user as the output.

User Agent: It can receive the input from Interaction Agent and make tokenizing process, can pass the tokens to the root of the interpretation agent, can receive the information from actuation agent.

Interpretation Agent: It can process in two phases.Phase1 is to interpret the received tokens by using downchain and upchain processes and the Phase2 is to generate the SQL command by using QSL converter and then send the SQL to actuation agent.

Actuation Agent: It can receive SQL command and retrieve the required information from the application and then it forwards the information as response to the user agent.

The following Figure3 shows the agent network for the request to find all information for Cyber Home.

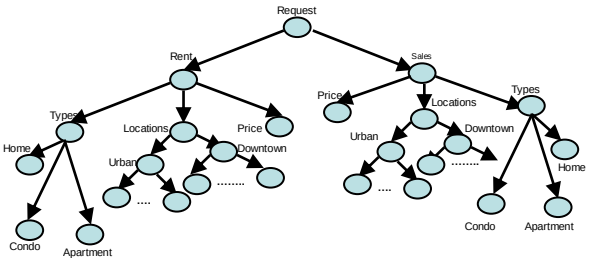


Figure3. An Interpretation agent for the "request" command

Each circle represents an agent; the arrows show the paths a request takes to reach every agent in the network. The tree begins with the Request agent, which receives the request from the user and passes it along to the agents that are downchain.

These second level agents map to the different kinds of requests the network understands:

The Rent and Sales agent understands what purpose the search is to begin.

The Type agent understands the type specifications (e.g condo, home, apartment).

The Location agent understands where the search is.

The Price agent identifies the desired amount of charges (e.g round about kyats 500 Lakhs, between kyats 500 Lakhs and kyats 250 Lakhs).

Agents further down the chain break the problem down further, into specific requirement to be found.

5.2 The Agent Process Flow

In this study, the agent network processes a request in two phases.

Phase one: It is interpretation phase (for determining of the user's intent).

Phase two: It is actuation phase (for generating a command to the application)

5.2.1 Interpretation Phase

This Phase begins when the top level agent receives the request from the outside world. The top level agent passes the request to its downchain agents, which pass it along to their downchain agents, and so on until every agent has seen the request. The leaf nodes then examine the request, each one deciding whether it recognizes anything in the request that it knows how to process. If it sees anything, it makes a claim on whatever part of the request it thinks it understands. The procedure for this work is as follow.

Procedure DownChain ( )
Begin

1. Accept tokens;
2. Find intersection points between input token and knowledge;
   3. Check the intersection points;
   3.1 If (intersection-points == no. of input token) then
      1. Message= "nodes= token";
      2. Call UpChain ( ) to send message;
      3. Exit from DownChain ( );
   3.1 if (0 < intersection-points < no. of input token) then
      1. Message = "nodes=token";
      2. Check node;
      2.1 if (leafnode) then
         1. Call UpChain ( ) to send message
         2. Exit from DownChain ( )
   Else
   ReCall DownChain ( ) to send token
   3.1 If (intersection-point==0) then
      1. Message =" " ;
      2. Check node;
         2.1 if (leafnode) then
         1.Call UpChain ( ) to send message;
         2. Exit from DownChain ( )
   Else
      ReCall DownChain ( );

End

Figure4. DownChain Procedure

5.2.2 Actuation Phase

Procedure UpChain ( )
Begin

1.Accept message;
2.Message = message + newmessage;
3.Check node;
If (! Root) then ReCall UpChain( ) to send message ;
Else return message as a result;

End

Figure.5 UpChain Procedure

In this phase, the network uses its understanding of the request to generate a command to the application. An agent may make multiple messages on multiple parts of the request. An upchain agent waits until it gets a response from every agent down the chain from it. It looks at the claims it receives and combines those claims. It may make its own claim based on the downchain agent claims or send a "no claim" to its upchain agent. In this way claim and "no claim" become messages and response messages travel up the tree until they reach the top level agent. The procedure for this work is as above (Figure5).

When the top level agent has received responses from the rest of the agents, it begins the second phase: the generation of the command to the application. This time, the command is created according to the agents whose messages were sent in phase one. The procedure for this work is as follow.

Procedure SQLConvertor ( )
Begin

1.Accept result from root agent
2. Create SQL statement by setting the results as criteria condition
3.Return SQLStatement

End

Figure.6 SQL Convertor Procedure

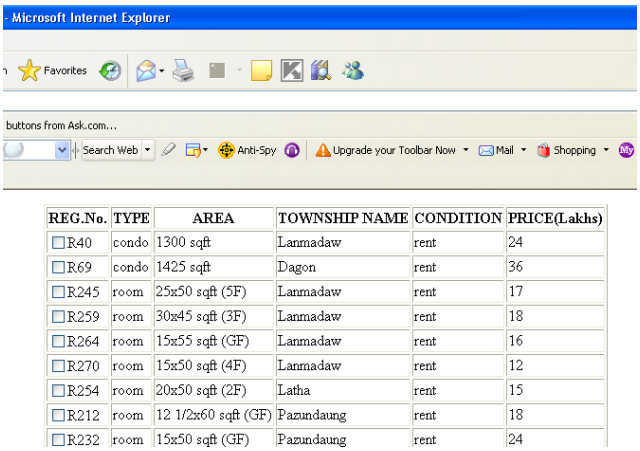
Consider the follwing example user request as the input for our proposed sytem.
" I want to get hired information from downtown not more than Kyats 36 Lakhs. "
After preprocessing, the request becomes as the following two sets.
Unused tokens ={ **I, want, to, get, from, than,.** }
Used tokens ={ **hired, information, downtown, not, more ,Kyats, 36, Lakhs** }
Now, the used tokens are became the request of processing component.

In phase one, the location agent claims the string **"downtown"**, the Price agent claims **"not more Kyats 36 Lakhs"** and the **Rent** agent claims **"hired information"**.

In phase two the Rent agent generates **"the condition is to rent"** (its translation of "hired information ") , Price generates **"<=36"** and Location agent combines and extends the word downtown into **"dagon and latha and lanmadaw and pabedan and botahtaung and pazuntaung"**. This processing continues up the tree until the **Request** agent has a complete command. It passes the command on to a special actuation agent which tells the application what to do as SQL statement:

SELECT * FROM *HomeInfoTbl* **WHERE (**
CON== "*rent*" **AND** PRICE<=*36* **AND** (LOCATION == "*dagon*" **OR** LOCATION == "*latha*" **OR** LOCATION == "*pebedan*" **OR** LOCATION == "*lanmadaw*" **OR** LOCATION == "*botahtaung*" **OR** LOCATION == "*pazuntaung*").

The application sends back the information on the above SQL Statement as an answer, which is fed to the user as shown as Figure 7.



REG.No.	TYPE	AREA	TOWNSHIP NAME	CONDITION	PRICE(Lakhs)
R40	condo	1300 sqft	Lanmadaw	rent	24
R69	condo	1425 sqft	Dagon	rent	36
R245	room	25x50 sqft (5F)	Lanmadaw	rent	17
R259	room	30x45 sqft (3F)	Lanmadaw	rent	18
R264	room	15x55 sqft (GF)	Lanmadaw	rent	16
R270	room	15x50 sqft (4F)	Lanmadaw	rent	12
R254	room	20x50 sqft (2F)	Latha	rent	15
R212	room	12 1/2x60 sqft (GF)	Pazundaung	rent	18
R232	room	15x50 sqft (GF)	Pazundaung	rent	24

Figure 7. Result Screen Shot for User's Request

6. Conclusion

In this study, it is focused on how it can be developed Natural Language Interface and also can be integrated it with an existing application. Therefore, the application user can operate the application program with Natural Language without the knowledge of special artificial communication language. For case study, we implement a portable Natural Language user interface for Cyber Home Finding System as specific domain.

The case study system can provide efficient responses according to the user Natural Language input and access information easily. Also, since the system can identify the online Home Finding System, it can support to reduce the effort, time-consuming and cost for the Internet users who interested searching in Home. Although our NL based system cannot completely solve on some common problems such as many synonyms (outside the regions of the knowledge domain or different subject specialties terms), many meaning (proper meaning of a word "disambiguates") and False Drops (can mimic speech). Also, the system is assumed that the natural language input had

corrected, so it cannot work on 'spelling variations and errors'.

7. References

[1] Burstein, F., Zaslavsky, A., Arora, N. School of Information Management & Systems and School of Computer Science and Software Engineering Monash University, Australia, Context-aware mobile agents for decision-making support in healthcare emergency applications

[2] Dejima Inc.;An Application of Agent Technology to Natural Language User Interface

[3] Efraim Turban, California State University at Long Beach, Expert Systems and Applied Artificial Intelligence, 1992.

[4] Nant Kyi Let Win (M.C.Sc.(Thesis), 2003), UCSY

[5] Sophie-Julie Pelletie, Samuel Pierre and Hai Hoc Hoang, Softimage, Montreal, Canada, Department of Computer Engineering, Ecole Polytechnique Montreal, Canada, Modeling a Multi-Agent System for Retrieving Information from Distributed Sources

[6] Tin Mar Nwe (M.C.Sc.(Thesis), 1998), UCSY

[7] Tinzar Gyaw (M.C.Sc.(Thesis), 1999), UCSY

[8] Woolridge, M. J. (2002). An Introduction to MultiAgent Systems. John Wiley& Sons, New York, NY.