Learning Interface for Virtual Education

Shahin MAGHSOUDI
Computer Science Department - The University of Auckland, New Zealand
e-mail: mmag005@ec.auckland.ac.nz

and

Tiru ARTHANARI
Information Systems and Operations Management Department - The University of Auckland, New Zealand
e-mail: t.arthanari@auckland.ac.nz

ABSTRACT

This system is an artificial intelligence entity, with the power of natural language processing, as a virtual teacher for learning and assessment of statistical inference, available through the Internet for student use.
It has a logical reasoning engine and uses two knowledge bases, one for statistical inference and the other for rules of English grammar and general knowledge to facilitate learning. The system is called Learning Interface for Virtual Education (LIVE). It provides lessons in inferential statistics. The user can ask questions in English and have a dialogue with virtual teacher. The system answers the questions or asks for clarification regarding what is required from it. It searches its knowledge bases (KB), uses its first order predicate logic reasoning engine and converses with the student using its Natural Language Processing (NLP) sub-system. It is aimed to provide a multimedia experience.
The system uses the following programming languages for its different functions: AIML for knowledge base, Prolog for reasoning, HTML and Java script for web client, Java for web server and AIML interpreter and Linux shell script for communication.
This system has potential applications outside academic environment such as web-based online marketing and helpdesk diagnostics in different sectors and can be used as assistant lecturer for other courses and subjects by changing its knowledge-base.

Keywords: Education and Information Systems - Computer Assisted Instruction - Artificial Intelligence - Conversational Robots - Natural Language Processing.

1. INTRODUCTION

Imagine that you can talk with your computer in natural language and she understands what you say. We are not referring to voice activated word processors. Goal is something like Mr. DATA in star trek. Imagine for deciding about a subject you consult with your computer and make a brain storming session that is helpful. Or using this program to entertain and inform old people who are sometimes lonely. Talk with a complicated database system and getting results instead of writing SQL code.

Applications of a natural language-processing program that can do logical reasoning are unlimited. This project is a step in this direction.

The post graduate students from the business school doing Research Methods (Quantitative) course have different backgrounds. The students are introduced to multivariate data analysis in this course. It is assumed that the students have considerable exposure to basic univariate statistics, statistical inference and correlation & simple linear regression. However, even the students with exposure to statistics, either because they learnt it long ago or they haven't learnt to apply, find it difficult to do the first quiz on basic concepts. The students were given a two weeks quick revision of these concepts. Still the first test in the course, which is on their basic knowledge in inference, turns out to be tough for them. Also they were suggested to consult certain internet resources. They came back with more questions. However, with a student teacher interactive session, the concepts became clearer to the student.

The idea for this project stems from this experience. Goal was creating a virtual teacher who can understand plain English questions of a student and answer them, explain the concepts, give examples and perform the assessment 24/7 on the internet with a friendly animated human character interface.

2. BASIC APPROACH FOR THE SYSTEM DEVELOPMENT

Observing what takes place during a typical student teacher one to one interaction, to explain a particular idea/concept, revealed that (1) the teacher knows the background and urgency of the student or finds out (2) chooses an appropriate example or uses suitable terms and explains (3) from the students reactions (including emotional and body language inputs from the student) the teacher chooses the next step of showing a graph or a video clip or a formula or a computation to take the student closer to the goal of understanding the concept. And it is an iterative process. All these involve a good amount of natural language processing in the first place. Reasoning abilities and availability of the relevant knowledge bases on general facts, specific to the course and about the student are also required and used. Based on these considerations a scheme for the system development is presented in Diagram -1.

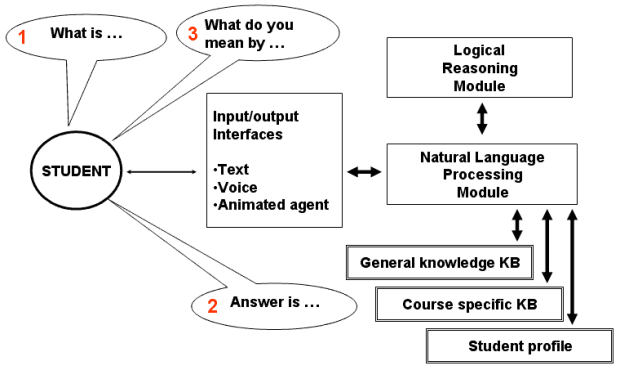

Diagram - 1 Schematic diagram of the Design

2.1. Natural Language Processing

Natural language processing (NLP) is the formulation and investigation of computationally effective mechanisms for communication through natural language. This involves natural language generation and understanding. NLP is a multiple discipline field, which involves general linguistics, cognitive

science, Artificial Intelligence (AI) and computational linguistics [04]. Due to its generic complexity, NLP confronts many problems to solve. Recently, ALICE (**A**rtificial **L**inguistic **I**nternet **C**omputer **E**ntity) which can work as a conversational stand-alone program won Loebner prize, in the competition for AI programs that try to pass a version of Turing test. [2]. ALICE is abbreviation of **A**rtificial **L**inguistic **I**nternet **C**omputer **E**ntity. Alice can talk with the user through a text input/output interface. She has 44000 categories as general knowledge.

2.2. Reasoning

Different reasoning capabilities can be added to the system. In this virtual teacher system deductive reasoning is used to derive new facts from the existing facts in the knowledge bases. Alice does not understand the logical connections of the facts. Since it was decided to use Alice in a major way to implement the scheme, in this project her reasoning ability is improved with Prolog and she has learned some deductive reasoning formulas. Other reasoning abilities like common sense reasoning, deductive reasoning, defeasible reasoning can also be provided, but not attempted at this stage. (See references 3-9)

2.3 Knowledge bases

Two types of knowledge bases and a database are envisaged in the scheme. The general knowledge base of Alice is used for an intelligent conversation with the student. Alice can learn newer facts either supervised or unsupervised. In addition for helping the students with basic statistical inference, her specific knowledge is improved with an additional statistical knowledge base. This knowledge base also contains examples and information on relevant video clips, graphs, formulae and computational programs. To enable the choice of appropriate examples for a particular student based on her background, a database giving student profile is added. This dbase can be built either using the conversation mode or by a form required to be filled by the student at the first interaction with the system.

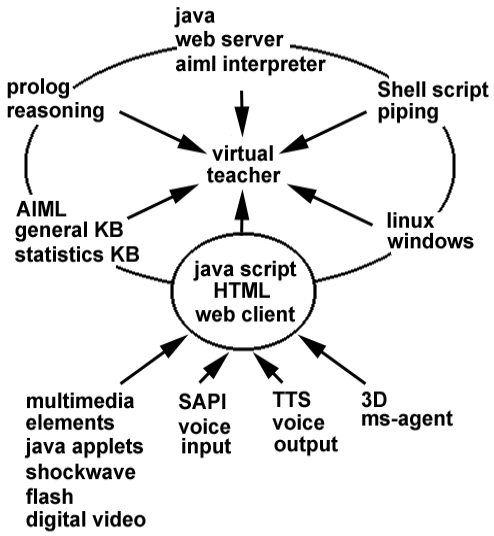

Diagram 2 - Elements of the system

2.4 Input output interface

In the simplest possible case the input from the student like the questions and the other information sought is in the form of text. The system outputs text /graphs /video clips on the screen. It is also possible to have voice input and output and an animated agent interacting with the student.
Diagram 2 gives the elements of the system developed to implement the virtual teacher.

In the next section, how these elements are put together in the system is explained. At the start of the project some integration has already been done. The developments that are specific to this project are highlighted.

3. SYSTEM DESIGN

Diagram 3 shows the new (oval boxes) and the modules that existed (square boxes) prior to the project Logic processor is connected to AIML interpreter through a <system> tag that is for calling external executable programs (in this system, the Prolog logic processor). AIML is the abbreviation for Artificial Intelligence Markup Language. It is an easy to use XML type language for communicating with and teaching the robot. Alice uses AIML for storing and retrieving its KB.

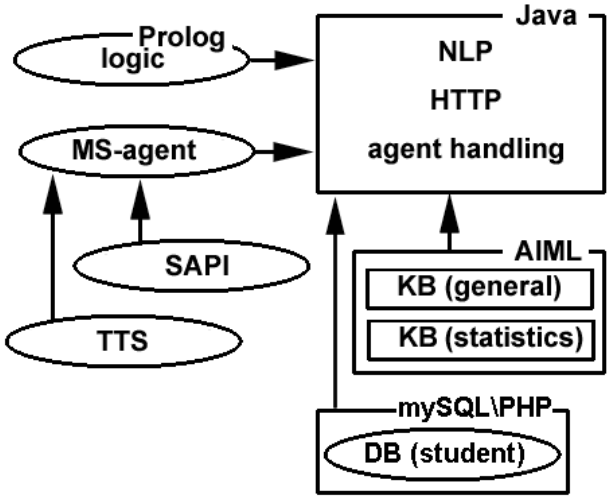

Diagram 3 - System design

NLP = natural language processing module
HTTP= web server
KB = knowledge-base for general conversation and statistics
DB = student profile data base (information and progress report)
SAPI= speech API for voice recognition
TTS = text to speech module for voice synthesis
Logic = logic processing module for common-sense reasoning

3.1. Logic processor sub-system

The user types the question in English. The virtual teacher looks for patterns and goes to the reasoning engine to find the answer. Logic processor is capable of common-sense reasoning over limited categories and uses some of the argumentation theory's logical reasoning rules. System can do logical reasoning and update its knowledge bases according to the new information it collects from conversations. Logic processor is written with Prolog language and is linked to the Java code of the NLP and AIML interpreter with shell scripts under Linux. If question directly uses one of the known common-sense reasoning categories of the system then it draws the answer from the logical module, if it fails then goes back to the AIML module and uses either the general knowledge KB specific KB for answering.

First order predicate logic was selected and within the boundaries of argumentation theory, Modus Tollens, Modus Ponens and Hypothetical Syllogism and within the boundaries of common-sense reasoning predicates “is”, “does” and “has” were incorporated to the program. For example, if I “do” cycling and cycling “is” exercise, Alice+ (enhanced Alice) can conclude that I “do” exercise by combining two predicates and one modus ponens deduction.

There can be virtually unlimited number of predicates that can be programmed to be used for common-sense reasoning. CYC has built a collection of 10,000 predicates. CYC is a program that has a large knowledge-base. It started by Dr. Lenat 18 years ago. It has its own language that is called, CycL. This language is an extension of Lisp. CycL, the CYC® representation language, is a large and extraordinarily flexible knowledge representation language. Recently an interface is built between CYC and ALICE that gives the integrated system beautiful and user-friendly NLP and interface of ALICE with great reasoning abilities of CYC. For more info refer to [1].

Argumentation Theory contains several rules and methods of reasoning. Three fundamental methods were coded to program. Rules are from [10].

- Modus Tollens $(p \Rightarrow q), \neg q \ \backslash \ \neg p$
- Modus Ponens $(p \Rightarrow q), p \ \backslash \ q$
- Hypothetical Syllogism $(p \Rightarrow q), (q \Rightarrow r) \ \backslash \ (p \Rightarrow r)$

Virtual teacher system works based on 1St order predicate logic. The reason for this choice was that computer language PROLOG contains rules of this logic and it was used for programming the project. If other languages such as C++, FORTRAN or Java were going to be used, amount of code could increase enormously. Prolog with logic within, and backtracking mechanism to access the internal database caused a clean understandable and efficient code that can be maintained and updated to incorporate more rules easily in later stages.
A different approach for solving the same problem was taken by creators of CYC system that uses n^{Th} order predicate logic.

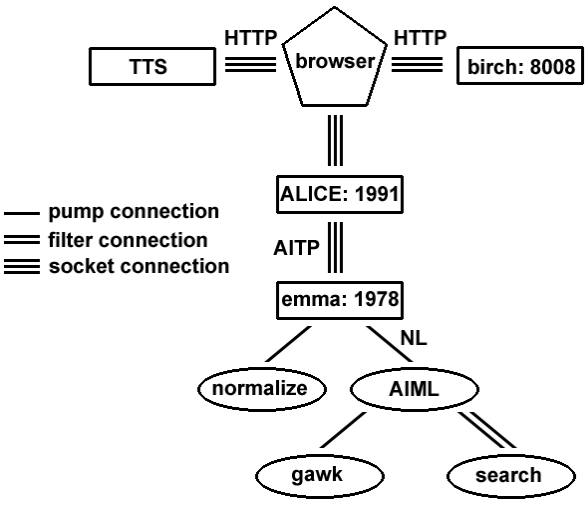

Diagram 4 -Transaction process, constructed from content of [1]

3.2. NLP sub-system

Rules of the English language grammar and natural language processing are built into the system written in Java code. System uses program D of the Alice engine from [1]. It uses pattern matching for NLP. This might result in further dialogue with the student to show him/her graphs, tables or other multimedia learning enhancement objects such as video clips of a human teacher and incrementally introduce the topic, by interactive sessions. Virtual teacher does assessment of the knowledge of the students and presents results and feedback for their course of study.

3.2.1. Transaction Processes

ALICE is a collection of autonomous clients and servers communicating via **TCP/IP**, which includes **AIML** and its interpreter, along with the Artificial Intelligence Transfer Protocol **AITP**. Diagram 4 shows a subset of the processes involved in a typical transaction with ALICE. The client (a human) utilizes a browser to connect to the server installed on the host **Alice: 1991** and transmits a query. The reply contains HTML markup for the video server and for the remote text-to-speech synthesis server. The browser interprets this HTML and initiates connections with the **audio** and **agent** hosts. The process on **Alice: 1991** is a forking server, a protective shield against misbehaved clients. Once the forked child receives the client's input query it initiates an atomic transaction with the server on **Emma: 1978** this transaction consists of two exchanges: first, the faux HTTP server reports the IP address of its client and second, it retransmits the client query in its original HTTP format (specifically, a GET method). The program "normalize" runs as a command line pump. It processes one line of text input and removes all non-meaningful (to AIML) punctuation, converts to upper case, and replaces a number of contractions and acronyms with their expanded forms. The result is one line of normalized text, suitable for matching with AIML patterns. Another line pump is gawk. [1].

3.2.2. Symbolic Reductionism

Symbolic Reduction is a pattern matching process in Alice responsible for the following four tasks [1]:

- Maps multiple patterns to the same response.
- Reduces a complex sentence to a simpler structure.
- Diminishes the need for multiple-wildcard inputs.
- Translates state-dependent inputs into simpler stimuli.

3.3. GUI sub-system

System has a user friendly manner. Students can choose appearance of their teacher. A man, a woman, young or old is available to choose as preferred teacher. Also several options of different male and female voices are available to choose according to preferences of the user.

Graphical user interface presents the 3D animated character as virtual teacher. It uses Microsoft agent technology called by a web page hosting on a web server. Virtual teacher can talk by using Text To Speech (TTS) and Speech Application Programming Interface (SAPI) technologies. TTS is available from Microsoft site to download and install over windows. If user uses Linux as web client virtual teacher can appear as animated GIF. It is also possible to use Flash animated characters. Student can type and receive his answers as text and voice if he has installed TTS over his computer or text only. If user wants he can install speech recognition software and train it with his voice and do a natural conversation with the virtual teacher.

3.4. Knowledge bases representation

The system has three KB that are written in AIML language [1]

- General knowledge KB
- Inferential statistics KB
- Student profiles KB

The First KBs enables the robot to do a meaningful conversation with the student and the other two are used for teaching specific knowledge for the specific student

Automatic detection of new patterns for which the robot does not already have specific answers, is defined as **targeting**. AIML is an XML language. This is an example of an AIML patterns:

Example of an AIML category:
<category>
<pattern>WHAT ARE YOU</pattern>

<template>
<think><set name="topic">Me</set></think>
I am an intelligent computer program.
</template>
</category>

If this category is called, it will produce the response "I am an intelligent computer program.", In addition, it will use the <think> tag and will set the "topic" in its memory to "Me" for further pattern matching.

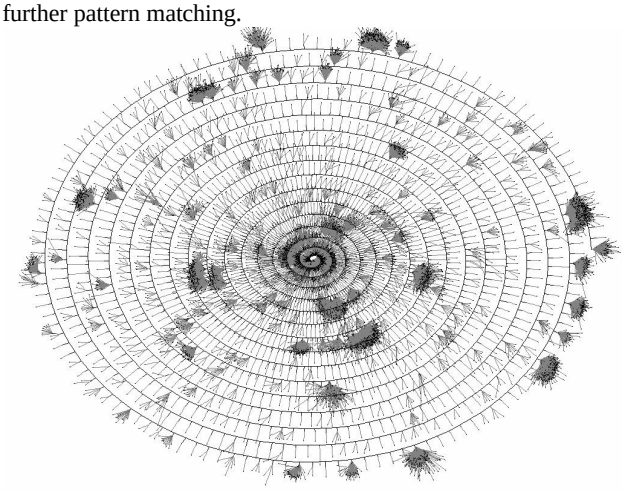

Diagram 5 - Graphmaster output, visualization of categories - from [14]

Alice's brain is shown in Diagram 5 as a spiral plot of 24,637 categories loaded. The spiral itself represents the root. The branches are the patterns recognized in previous training conversations. Lines connected to the spine of the spiral represent different words recognized by Alice and the connection between them.

3.5. Multimedia sub-system

Virtual teacher talks with the students, identifies their questions and depending on the topic, sometimes answers them with text using the GUI or guides them to the pre-made multimedia lessons hosting on the web server when ever additional support material for its lessons are necessary. All multimedia modules are available over the internet using WWW server and digital video streaming. 3D model that acts as virtual teacher sometimes appears presenting a topic in digital video format and sometimes a human teacher is used for lectures. Multimedia modules are:

- Assessments
- Animated graphs, diagrams, pictures and reconstructions of formulas
- Interactive step by step lessons, (Java applets, Shockwave and Flash modules)
- Digital video of the human teacher describing some topics
- Digital video of the 3D model (virtual teacher) describing some topics

3.6. Web server and security

There is a built in web server for the virtual teacher that is written with Java and communicates over the internet with an allocated port. It is possible to use another web server such as Microsoft Internet Information Server (IIS), Apache or Netscape server for hosting multimedia elements and other learning assisting material. For security use of a firewall such as Zone Alarm or Norton Internet security system is recommended.

3.7. Data Flow Diagram

Diagram 6 depicts the flow of data in the system.
1 - Student asks a question or initiates conversation.
2 - System looks at the pattern
- If it is similar to "**is a * a ***" then goes to **y2, y3, y4, y5, y6**
- Else, it goes to **n2**, **n3, n4** and finds a relevant response from **AIML KB**. In logic sub-system it does the deduction and reports it:

- **MT**: Modus tollens
- **MP**: Modus ponens
- **HS**: Hypothetical syllogism
- W: Predicates. IS, DOES, HAS
- **X** W **Y**: Does the query has a pattern similar to **X** W **Y**

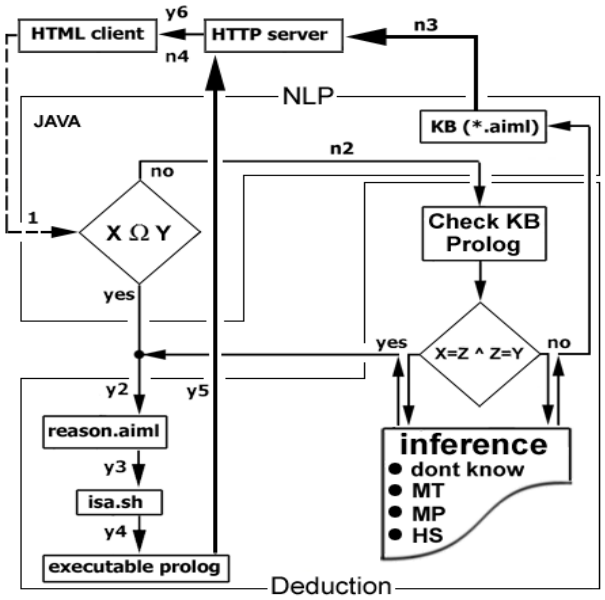

Diagram 6 - Data flow diagram

If the prolog code doesn't find a pattern of (**X=Z ^ Z=Y),** it reports (I do not know) and goes to the path of **n3** and **n4**. If it finds the pattern reports **MT**, **MP** or **HS** and goes to the path of **y2, y3, y4, y5, y6.** Program **reason.aiml** is the link between **executable prolog** program for reasoning and java code for **NLP** and user interface**.**

3.7. Hardware requirements

System uses the Internet for communication; therefore a wide bandwidth (related to estimated traffic) is required. Since All knowledge bases, Statistical KB and general knowledge KB are loaded to RAM in initialization phase, (for quick response) and other services are running on the system simultaneously, (such as web server, firewall and OS itself) then a descent amount of RAM is required. That should not be less than 512 MB. It is also clear that faster CPU clock creates better performance. Multimedia modules are kept as web pages and do not demand RAM. Also a decision should be made about installing a streaming digital video server or downloadable video clips. This decision can affect amount of required RAM. To be in the safe side a computer with 1 GB RAM and 3GHz clock is recommended. Size of the hard disk is related to the amount of the content that is prepared for the course. Digital video clips or streams usually take large hard disk space but multimedia content of the web pages should be designed to be as small as possible for quick web response.

4. LEARNING

4.1. Learning by AIML

By using **AIML** system can learn in two modes. In real-time learning mode system creates a log file from conversations. If it is redirected to its **KB** then it can learn in real time. In this case truthfulness and trustworthiness of the one who is communicating with the system is important. In off-line learning mode if truthfulness of the communicator is not guaranteed then system admin can teach the system off-line. This can be done by editing the log files of the past conversations and adding the useful information back to the **KB**. There are simple AIML editors available in [1] that a bot-master can read and trim these conversation files.

4.2. Avoiding inconsistency of KB

Inconsistency in this case can be defined either as duplication of information with minor differences or opposing information.

With the present structure of the programs there are two knowledge bases. One in the Prolog program and other in the collection of AIML files. Since certain relationships between certain objects need to be taught to the program explicitly by adding it to the KB of Prolog program it is possible to check the AIML collection by a simple search text string program before adding to KB.

This method requires human supervision. To make the process automatic an epistemic logic decision making engine for real-time learning using belief revision axioms can be developed to add facts, remove facts or revise the KB.

4.3. Problems and solution

The positive side of using Prolog in this project is that it contains rules of first order predicate logic and internal backtracking. The negative side is that Prolog reports false if it doesn't have the clause in its predicate list. This is the result of the backtracking mechanism. A programming trick is to check its KB and if a fact is not there, instead of using internal logical inference it says "I do not know." And in the learning version of the program it asks: "I do not know this, Do I learn it or not" Then it is left to the user to decide that new fact should be added to present KB or not.

Learning problem for the system can be defined as: Which type of learning should be used, on-line or off-line? Which sub-system should be enhanced for taking care of learning, AIML interpreter, pattern matching module or another sub-system?
First approach to address this problem was teaching the rules to the system through AIML patterns. This method required a huge amount of RAM.
Second approach was putting the facts and their relationships to Prolog code. In this case Prolog performance will decrease by increasing the number of facts.
Finally the best method used was dealing with new facts that are not in the collection of KB as off-line learning and using Prolog for reasoning and AIML interpreter for pattern matching.

5. RESEARCH METHODOLOGY

It is planned to evaluate, using statistically designed experiments, the effect of the number of hours spent with a virtual teacher in addition to attending the regular lectures by a human teacher. The control groups of students with different backgrounds with respect to prior knowledge of statistical inference have the human teacher only.

The hypotheses, regarding the outcomes and experiential benefits of using a virtual teacher, that are to be tested are as follows:

H1: The performance of the students is not significantly enhanced by the use of a virtual teacher.
H2: Learning experience of the students is not significantly improved by the use of a virtual teacher.
H3: Number of hours spent with a virtual teacher has no correlation with the performance of the students.
H4: Number of hours spent with a virtual teacher has no correlation with the learning experience of the students.
H5: The usefulness of a virtual teacher does not depend on the student's background.

For each student in the study the following variables are recorded:

- Number of hours (on-line learning) with a virtual teacher
- The final grade
- Results from questionnaire
  - Enjoyment of learning
  - Difficulty of course (difficult, medium, easy)
  - Recommend this course to others (Y/N)
  - How was a virtual teacher different from the human teacher?

- background of the student
  - studied statistical inference before (and remembers)
  - studied statistical inference before (and does not remember)
  - did not study

- Since we have more than one response we go for a multivariate analysis of variance (MANOVA). The language data obtained from the questionnaire will be subjected to qualitative data analysis tools.

6. FURTHER DEVELOPMENTS

The proposed system does NLP. It knows some deductive logic formulas but there are other logic formulas to be learnt, therefore, its reasoning is restricted to its present rules. Also similar to human, thinking power of the system (for general conversation) is limited to amount of facts that it knows (44000 at present). For statistical inference course required knowledge is available in its second KB. Ontology is defined as: "An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them [11]." A more complete ontology can increase intelligence of system. A case-based reasoning sub-system can be added to its logic and NLP sub-systems.

As observed earlier the non-verbal inputs during a student-teacher interaction, like body language, moods, and so on are also essential. When a student arrives the teacher recognizes and exchanges pleasantries or introduces herself and the student introduces. [or makes fun of the teacher - how forgetful the teacher is.] Teacher not only answers the questions but also watches the changes in the student's body language / behavior /spark in the eye/ disappointed look / etc. Student not only listens to the teacher but also watches the changes in the teacher's body language / behavior appreciation / irritation/ impatience / etc. Student may not be convinced even with a logical explanation. Teacher may understand this and try another way of putting it. Teacher's knowledge about the students in general may not match the background or cultural aspects of a student, and often gets revised. Student specific profile is dynamically revised by the teacher. She forms opinions and uses in the interaction. In addition to Natural Language Processing additional sub-systems for dynamic revision of student profile, response to emotional changes, revision of cultural predicates, alternative ways of presenting the material, based on student's non-verbal response, may be needed to enhance the virtual teacher-student interactions.

7. IMPLICATIONS

An intelligent robot with reasoning abilities can work in the following jobs: online marketing representatives, smart interfaces, intelligent characters for games, enhanced virtual reality, improved machine translation, improved speech recognition, virtual teacher (specialized knowledge base), Companion for elders (general info), psychologist (expert knowledge base)

8. CONCLUSIONS

By using SAPI, TTS, MS-Agent, [13] and NLP technologies a system was created using AIML, JAVA and Prolog languages that acts as a virtual teacher, available for consultation for students continuously over the Internet, answering their questions patiently and guide them through new course material. This artificial intelligence entity can be replicated to teach different subjects by changing or adding to its specialized **KB**. Although most learners might prefer interacting with an organic teacher, as most essential functions of a human teacher are incorporated in **LIVE**, it can be more useful as a proxy teacher because of its availability, perfect memory, patience and maintenance cost.

9. REFERENCES

[1] AIML - Artificial Intelligence Markup Language - Created by Dr. Richard Wallace and AIML community
http://www.alicebot.org

[2] Huang X, Acero A, and Hon H. W. - Spoken language processing, a guide to theory, algorithm, and system development. - Prentice Hall - 2001

[3] G.R Simari and Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning
and it's Implementation. Artificial Intelligence, 53:125–157, 1992.

[4] Phan M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming and n-person games. Artificial Intelligence, 77:321–357, 1995.

[5] Henry Prakken. Dialectical proof theory for Defeasible argumentation with Defeasible priorities (preliminary report). In Proceedings of the 4th Modelage Workshop on Formal Models of Agents. Springer Lecture Notes on AI. Springer Verlag, 1998.

[6] Alejandro J. Garc´ýa, Guillermo R. Simari, and Carlos I. Ches˜ nevar. An argumentative framework for reasoning with inconsistent and incomplete information. In Workshop on Practical Reasoning and Rationality. 13th biennial European Conference on Artificial Intelligence. August 1998.

[7] Alejandro J. Garcýa. Defeasible Logic Programming: Definition and Implementation (MSc Thesis). Departamento de Cs. de la Computaci´ on, Universidad Nacional del Sur, Bah´ýa Blanca, Argentina. July 1997.

[8] Alejandro J. Garcýa and Guillermo R. Simari. Defeasible logic programming. Technical report, Computer Science Department, Universidad Nacional del Sur, October 1998. Technical Report GIIA-1998-20.

[9] J. Dix, F. Stolzenburg, Automating Defeasible Reasoning with Logic Programming Universit Koblenz Germany, dix@informatik.uni-koblenz.de

[10] Introduction to Logic - Roderic A. Girle - Prentice Hall - 1999

[11] On-Line Dictionary of Computing:
http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=ontology

[12] Stephen L. Reed and Douglas B. Lenat - Mapping Ontologies into Cyc - www.cyc.org

[13] SAPI, Speech Application Programming Interface, and TTS, Text to Speech are Microsoft products.

[14] http://www.alicebot.org/documentation/gallery