A Learning Fuzzy System for Intelligent Conversation

NOUHAD J. RIZK
Computer Science Department
Notre Dame University
P.o.Box 72 Zouk Mickael
LEBANON

Abstract: - It has recently been observed that the computer revolution of educational practice has been anything but swift or dramatic. It can in face, be argued that a revolution that takes 20 years is probably more of an evolution than a revolution. Whether revolution or evolution, it seems clear that the use of computers as an aid to the educational enterprise is here to stay. Indeed, over the past two decades many faculties have been experimenting with computers in the courses they teach, or the information they provide to their students. The results were astonishing even that the computer can almost substitute humans in many fields but still; there exist many disadvantages of using computers instead of human beings.
This study presents learning fuzzy system built in order to perform what a human being is capable of doing. It is a combination of a special algorithm and user interface forms that provide the functionality and ease of using the system. Finally, the study clearly reveals that successfully using the computer, as an aid to teaching and learning is an iterative process involving considerable experimentation, planning, redesign, and persistence.

Key-Words: - Educational Technology, Artificial Intelligence and Software Design.

1 Introduction

Educational technology is one of the most frequent topics of discussion and debate in higher education today. While proponents predict that computers will radically and irrevocably transform education, skeptics compare computers to other technological innovations that have come and gone over the last century. The truth lies most likely somewhere in between. Computers probably will never totally replace teachers, any more than books replaced those centuries ago. But neither will computers ever entirely disappear from the educational scene. The stories of the faculty's success in using the computer as an aid to teaching and learning and the faculty's failures provide a number of interesting insights and an outlook for the future.

A pedagogical function computers seem to have served quite effectively is to help students gain experience and fluency using an essential tool of the trade. That is, in many fields--such as statistics, engineering, and landscape architecture--knowledge of specific computer applications constitutes a type of professional literacy. Hence, most faculties in these fields find it impossible to imagine how they would prepare their students adequately without allowing them an opportunity to acquire those computer-based skills. However, many faculties reported that sometimes technology can unintentionally increase the gap between student and teacher. For example, some were dismayed that their beautifully prepared computer presentations had resulted in students sitting passively in the dark with no real opportunity to interact with the teacher.

Computers may be a new frontier, but they are not an educational panacea. In fact, many teachers express an explicit caution to others that computers are not more or less powerful tools than any other appropriately used media; they will never be able to fix faulty instruction or relieve faculty of the responsibilities to teach. Computer technology is not a black box where dreams come true, but a tool that is only as good as the craftsman using it.

In the end, most advised that it is best to start with a real and vexing pedagogical problem. An unfocused desire to use computers was rarely recommended, and even less rarely netted useful result. Rather, starting with the need to unpack a concept that is frequently misunderstood, to provide students with critical kinds of computer literacy skills, or provide students with access to crucial data or information seemed a surer route to success and time well spent. And although some teachers clearly indicated that more effective learning *can* result when computer applications are used appropriately, they also made it clear that such an outcome could not be assumed. In the end, many noted that enhancing the teaching and learning process with computers is no different from any other aspect of good teaching it requires a good idea, thoughtful planning, intelligent problem solving, and a willingness to try again.

This study describes the application of artificial intelligence techniques and concepts to the design of systems to support learning. This paper discusses how work in artificial intelligence (AI) is contributing new approaches to education and learning by introducing intelligent tutoring system, in section 2. The design approach of the software is presented in section 3. Then, the implementation issue and the special features of the tutor software are discussed in section 4. The tutor control learning issue is explained in section 5. Section 6 enlightens the use of the tutor in classrooms. Finally, the conclusion suggests that the most effective uses of artificial intelligence (or advanced technology in general) in education can be moved into classrooms with a minimum of disruption.

2 Intelligent Tutoring System (ITS)

A British scientist argues that he honestly believes that intelligent machines in this century are going to outstrip humans in many ways and take over from us effectively. But the mid ground, the hybrid ground, part human, part machine, appears to be a possible way forward for humans to evolve [8]. Therefore, this section discusses the development of intelligent tutoring systems for education called A.J. ITS attempts to capture a method of teaching and learning exemplified by a one-on-one human tutoring interaction. For researchers in AI this method of teaching was a natural one to target first for several reasons. Drill-and-practice versions of one-on-one tutoring are relatively well-understood ways of communicating knowledge. This method of learning and teaching is widely accepted both by the educational community and by our culture as a whole. And it has achieved broad popularity for good reason. One-on-one tutoring allows learning to be highly individualized, and consistently yields better outcomes than other methods of teaching [1]. Although many methods have been examined, no other has reliably yielded improvements in student outcomes. Failure to attribute strong outcome improvements to other methods of teaching and learning may be partly a function of inadequate techniques for evaluating novel learning outcomes, as we elaborate below. However, regardless of evaluation problems, it is clear that one-on-one tutoring remains a "gold standard" of learning.

The heart of an ITS is its expert system. The expert system embeds sufficient knowledge of a particular topic area to provide "ideal" answers to questions, correct not only in the final result but in each of the smaller intermediate reasoning steps[5]. The expert system thus allows the ITS to demonstrate or model a correct way of solving the problem. Often, like a human tutors, it can generate many different answer paths or goal structures [5]. The same detailed data structures that expert systems generate in modeling expert reasoning also permit ITS to explain their reasoning at arbitrarily detailed levels. For example, if a student needs an explanation of why or how an information ITS did a step in answering from the knowledge base system developed earlier by other users. If the student requested more justification, it could elaborate by describing the terms that were distributed. Explanations thus turn expert systems from opaque "black box" experts into inspectable "glass boxes" [3].

We have experimented with pedagogical component in our AJ tutor. The pedagogical policy permits high students control. The student decides what to ask, when to request the expert to do and explain, when to provide accurate information for other users.

3 The Design

Computers are developing quickly. Their parameters are currently progressing in the pace determined by so-called Moor's Law that claims that they double every two years (e.g. memory, capacity, speed). Therefore the abilities of computer-based systems improve as well. Our ITS so-called A.J may get closer to the human abilities. Its work is based on the Alan Turing test of computer intelligence. This test is based on an experiment where a chosen person asks questions using keyboard and receives answers on the screen. Whether the answering subject is a machine or another human being the key lies in the accuracy of answers.

The fuzzy system presents the idea of having a computer as an aid to teaching and learning from existing practice. It may reach a point where it will completely take place and provide the efficiency of a human being. The user can ask the system any question, and in return it will provide the answer to that question. If the system does not know the answer, the question will be saved it in its database. Then when a new user signs in, it will ask him the unanswered question and save the answer in a special table. The system then checks the truth level of the answer from asking different trusted users until it reaches a point where it will save the answer permanently.

4 Implementation

This project is divided into many parts. One of the most important parts presents the algorithm upon which the system is build. In addition to all the other elements, every part presents a special feature and has unique roles that are combined together to make this learning fuzzy system which is built to help the students as well as instructors.

4.1 A. J.'s rule based system

A rule based system has five components: the knowledge base, the database, the inference engine, the explanation facilities, and the user interface (Fig. 1).

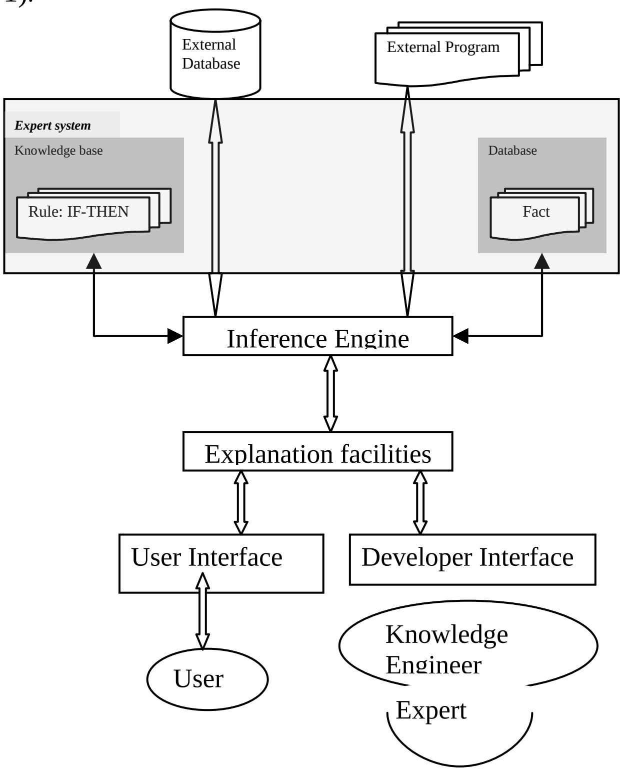

Fig 1: Complete structure of a rule-based expert system

The knowledge base contains the domain knowledge useful for problem solving. In a rule based expert system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure. When the condition part of a rule is satisfied, the rule is said to fire and the action part is executed.The database includes a set of facts used to match against the IF (condition) parts of rules stored in the knowledge base. The inference engine carries out the reasoning whereby the system reaches a solution. It links the rules given in the knowledge base with the facts provided in the database. The explanation facilities enable the user to ask the expert system the how, why, what, where, when, how many and how much questions and the system will provide the most accurate answers through analysis and reasoning. Finally, the user interface is the means of communication between a user seeking a solution to the problem and an expert system. The communication should be as meaningful and friendly as possible.

These five components are essential for any rule-based system. They constitute its core, but there may be a few additional components.

4.2 The Automaton

In order for the algorithm to analyze the type of question asked by a user and whether the input is a word, verb or just a question beginning with the How, What, Where, Why, How much and How many, we designed a non deterministic finite automaton for this purpose. The non deterministic finite automaton has a special feature which is called non-determinism. This is the ability to change states in a way that is only partially determined by the current state and input symbol. The NFA is a severely restricted model of an actual computer known also as finite-state machine.

Here is the full data flow diagram as a finite state machine upon which the algorithm is constructed.

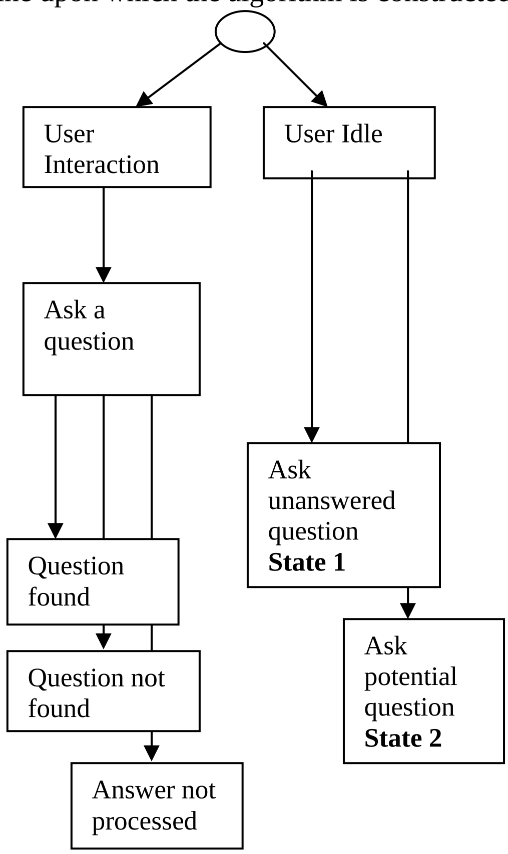

Fig. 2: The data flow diagram

The purpose of each artificial intelligence algorithm is to always enhance itself. Enhancement can be made by specific artificial intelligence language like LISP or by appropriate selection of data and organizing it in a database using a specific algorithm. We opted for the second option in the construction of our system. That is, if the algorithm certifies that the answer to an unknown question is true (Fig. 2) using many variables that changes upon the opinion of chatters and their level of trust the algorithm saves definitely the question and its preceding answer in the database, thus learning the question and its answer. The next time the question is asked, the answer is surely provided.
Many forms are created for the user to access the program and communicate with it whether by asking any information (Fig. 3) or giving appropriate answers to different question asked by the program (Fig. 4).

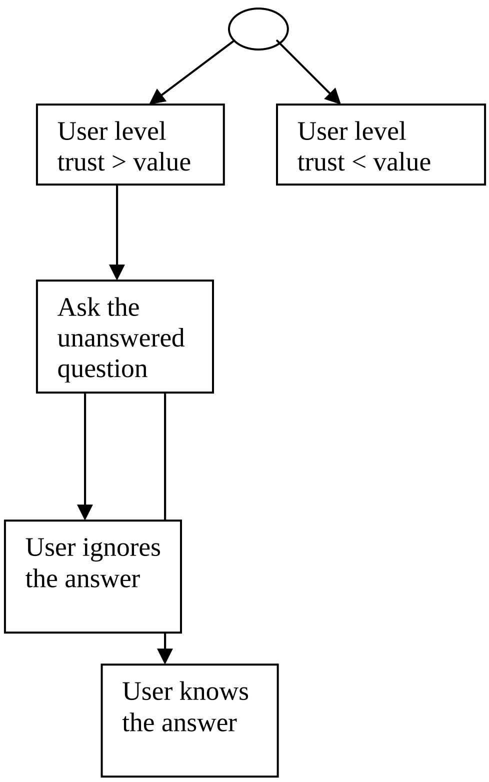

Fig. 3: State 1, the state of asking a question

4.1 A. J.'s Special features

Many special features were added to the system, in order to provide the perfect communication between it and the user. These features make A. J. a thinking, speaking and funny expert system which is able of interacting with the user through typing and speaking:

Indexed tables: In order to provide the maximum speed during the running time of the system and to make this program run swiftly and accurately, we used clustered and non-clustered indexes to provide query speed.
Text to speech conversion: In order to make the communication between the program and the user more real like, we provided a text to speech enhancement by using inherited libraries. That is the program will be able to speak any sentence, piece of information or question that pops on the screen.
Special voice control: We provided a special mouth picture that moves just the words it reads. It contains properties in which the color can be changed, the appearance of the lips, teeth and tongue could be changed by code. This feature makes A.J system looks like a real human being when talking with the users.
Stored procedures: These procedures are queries written in SQL and they provide usability. Since we used the VB.net for writing the code, it was easy using the stored procedures instead of writing SQL statements inside the VB code. This provides speed and dependability.
Error handling: Every time the an error is found, it is caught by the code and thrown away so that the program is always safe from crashes and run time errors
Object oriented: The functions are separately written so that they can be called by the corresponding event for a perfect functionality of the program.

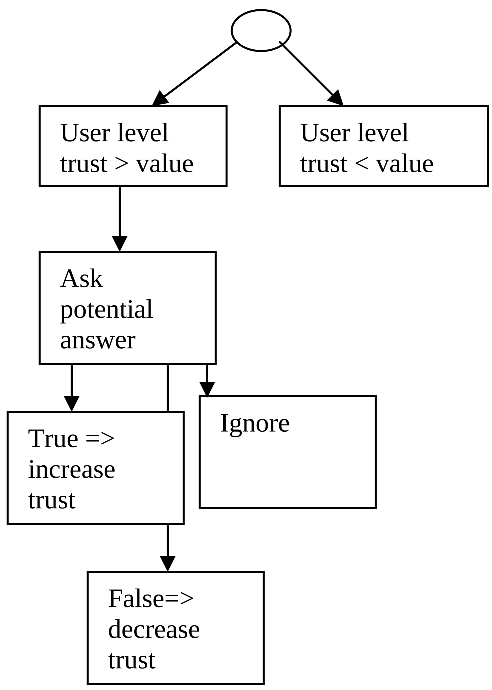

Fig. 4: State 2, the state of answering a question

5 Tutor Control of Learning

Of course no computer could understand anything real -- or even what a number is - if forced to single ways to deal with them. But neither could a child or philosopher. So such concerns are not about computers at all, but about our foolish quest for meanings that stand by themselves, outside any context. Our questions about thinking machines should really be questions about our own minds [6].

Fig 5: The interface

After the login procedure the interface looks like (Fig. 5). As soon as the chatter or user signs in, the program welcomes him by introducing itself in a suitable way so that the user will understand what this program is capable of doing. The form consists of two rich text boxes: One for the program, where it will show the text it is speaking and another one for the user to write in.Although some ITS, such the student-controlled version of our A.J tutor, permit limited student choice, for the most part interactions with ITS are tightly controlled by the software. In most cases, the ITS selects the next question, decides when the student answers and determines the nature of the information the students give. Students may tailor information; for example they may answer wrong answers. But their latitude is usually highly circumscribed. The principle of high tutor control reflects an implicit belief that a competent tutor is usually in a better position to make decisions about what experiences and information students deliver effectively than the students themselves. Of course, this assumes, at a minimum, that the tutor knows the content the students want to learn, and also knows the students' specific knowledge state at any given time. The expert systems and student models of ITS attempt to provide this expertise and to thus meet the demands of high tutor control of learning.

Triggers are used. These are daemon programs running in the background of the expert system in order to keep track of what to do next. There is no interaction of the daemons concerning the user; they are only for programming reasons.

6 A.J fits well into Classrooms

To the extent that ITS appears to contribute to effective learning outcomes, they substantiate the principles of micro-tutoring, high tutor control, impasse-driven coaching and providing rich and immediate feedback. However, although ITS are successful in part because they are consistent with various theoretical principles of learning and teaching, practical reasons may be equally important, if not more so. The simple fact is that ITS actually fit quite well into existing classrooms, programs and integrated learning systems that have enjoyed at least modest success. Easily filling the shoes of earlier Computers Assisted Learning (CAL) systems before them, which have attempted to implement traditional methods of learning and teaching.

7 Conclusion

While ITS have been somewhat successful on a small scale, several problems must be overcome before they have widespread impact. Various authors [8, 9] have discussed a wide range of limitations. Many of these challenges can be predictably factored by ITS component -- limitations associated with the expert system, student model, pedagogical component, and interface.

There is no reason to believe that the most effective uses of AI (or advanced technology in general) in education will happen quickly or without careful policy and planning. In the short-term, technologies resembling many ITS -- that aim at well-defined learning goals and that can be moved into classrooms with a minimum of disruption -- will provide the most statistically significant improvements in student outcomes. Policies that support research based on their ability to generate such results in "horse race" evaluations risk encouraging technology applications that miss longer-term benefits.

On the other hand, policies which simply give researchers free reign to develop software that focuses on new methods of teaching and new learning outcomes also run considerable risks. The problems in developing these systems and moving them into education on a broad scale are not simply technical ones. Our experience provides a case in point. When developing our ITS for education, 75% our effort was spent on technical and research issues, but implementation required most of our time when we integrated A.J into schools. As technology continues to transform the goals for student learning and to enlarge the range of methods for teaching and learning, implementation will require proportionally more effort.

The evolutionary process of technology seeks to improve capabilities in an exponential fashion. Innovators seek to improve things by multiples. Innovation is multiplicative, not additive. Technology, like any evolutionary process, builds on itself. This aspect will continue to accelerate when the technology itself takes full control of its own progression [4].
The success of educational technologies is judged of how well they imitated what good teachers do [2].

References:

[1] B. S. Bloom, "The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring", *Educational Researcher,* Vol. 13, No. 6, 1984.

[2] R. E. Clark, "Confounding in educational computing research", *Journal of Educational Computing Research*, 1985.

[3] C. Foss, "Acquisition of error management skills". *Third International Conference on Artificial Intelligence and Education*, 1987, pp. 27.

[4] Ray Kurzweil, *Th*e Age o*f Spiritual Machines.* Viking, New York, 1998.

[5] D. McArthur, C. Stasz, , J. Hotta, O. Peter and C.Burdorf, "Skill-oriented task sequencing in an intelligent tutor for basic algebra", *Instructional Science,* Vol. 17, 1988, pp. 281-307.

[6] M. Minsky, *AI Magazine*, Vol. 3, No. 4, 1982, pp. 12-14.

[7] J. Psotka, D. Massey and S. Mutter, *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale NJ: Lawrence Erlbaum, 1988.

[8] K. Warwick, *Back to the Future*, BBC interview**,** 1999.

[9] E. Wenger, *Artificial Intelligence and Tutoring Systems.* Los Altos CA: Morgan and Kaufmann, 1987.