ISSN No. 0976-5697

Volume 4, No. 3, March 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Mathematical Analysis of Problem Statements: Artificial Intelligence

Devanshu Paliwal
M.Tech. Scholar, Department of Computer Science and Engineering Geetanjali Institute of Technical Studies
Udaipur, India
devpal24@rediffmail.com

Dipesh Vaya
M.Tech. Scholar, Department of Computer Science and Engineering Geetanjali Institute of Technical Studies
Udaipur, India
dipesh.vaya88@gmail.com

Sarika Khandelwal
Assoc. Prof., Department of Computer Science and Engineering
Geetanjali Institute of Technical Studies Udaipur, India
sarikakhandelwal@gmail.com

Abstract: The calculation of a desired mathematical value is just an arithmetic function with inputs and their respective outputs. Apart from the numbers as inputs and calculating value, the analysis of a language presenting the problem of mathematics can add towards the further advancement of an automated function. This paper discusses the methods to learn and analyze simpler problems of mathematics in algebra and commerce in order to select the desired formula and extraction of the variables as the input parameters.

Keywords: Linguistic knowledge, Information optimization, Information compression, Segregation, Pattern-Recognition.

I. INTRODUCTION

The problems of estimated programming to make computers solve difficult problems can be divided into: Search, Pattern-Recognition, Learning, Planning, and Induction. To consider one aspect of artificial learning we discuss here the scope of solving mathematics language problems by a computer program. Solving word problems is learning goal in the course of study that learners and tutors try to realize in mathematics lessons. The concentration of the ongoing researches is on a better understanding of this question: How do tutors and learners manage a balance between limiting the real life experience to mathematical relevant aspects on the one hand and extending real life by invisible mathematical structures on the other hand? [2] A problem is defined by its objects and their relations. From the observed or given information problems can be processed to generate solutions. A person learning to extract the logic out of a problem has fixed criteria of analysis (Figure 1)

i. Analyze the problem statement
ii. Recognize inputs and applicable function.
iii. Plan the problem
iv. Carry out plan

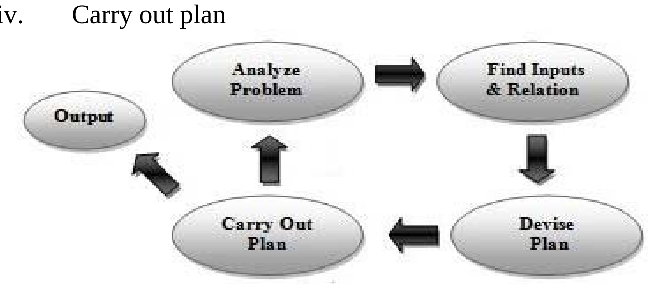

Figure 1. Flow to a problem solution

The selection of function however is sometimes driven by preliminary information [7]. As a human learns to analyze the problems and solve them, a software program can produce equivalent effects when made to remember some hints and calculate the parameter logics before attempting to solve the problem.

A. Knowledge:

The understanding of any text problem is a mental process that utilizes "two sources of information: the statement itself and knowledge – linguistic knowledge as well as knowledge about the world. Knowledge can be divided into two types: Declarative and Procedural Knowledge. Declarative Knowledge means representation of facts or assertions .This tells what about the situation. Procedural Knowledge means representation of actions or consequences and tells how about the situation [5]. The two knowledge types can be further discriminated with the help of a simple example. A trigonometric or logarithmic table is an explicit enumeration of the domain knowledge and can be considered declarative representation. On the other hand, a stored sequence of actions indicating how to compute the logarithm or trigonometric ratio of a number is considered as procedural representation.

In applying natural language to express ideas or formulate decisions, there is a great conciliation of imprecision and uncertainty. Working with some data, we automatically take them as precious [3]. For example "24 students in a class took a computer science test. If 18 students passed the test, what percent do not pass?"

In this question identification of inputs can be done in terms of the numerical values 24 & 18. The selection of applicable function here can be made on the basis of the

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

word percentage. The relation of 24, 18 and percentage can only be stated by"18 is what percentage of 24". Answer to this is given by (18*100) /24. The solution to this problem is said to be complete with the answer of "what percent do not pass", the phrase "do not pass" is the question and input with the verb passed i.e. "18 students passed" gives an idea of the function. So the function can be ultimately modified to (100-(18*100) /24).

According to this example problem solving is supported by sufficient knowledge of grammar of the language in which a problem is written. The standards of a language can be notified in the program memory as done in some word processing software, but when a natural learning is concerned the intelligence of a program is proportional to the richness of the language vocabulary and power to ascertain the problem [6].

B. Learning:

In order to remember the concepts and patterns, an observer must have a conclusion in terms of a concept at the end of a problem solving attempt. Here are few examples when one may learn:

i. Practicing a problem pattern
ii. Storing the experience of a problem each time after an attempt
iii. Taking advices from a smart system, which may be a GUI system operated by human. The user suggests hints in answer to small program questions.

A person learning some language like English once know the key techniques used in mathematical word problems, it becomes easier to create and solve the numerical equations. It is by practice and learning that a human develops vocabulary and thus knowledge [1]. In a similar way the knowledge of a machine can be enhanced by making it to learn the things with practice. A student sharpens the memory by practicing questions on a regular basis, even when not expected. If a program is made to do so along with revisions on schedule intervals, it can enhance the memory and reasoning power day by day. Each time it comes across a problem, it tries to learn the new patterns engaged and the keywords used, so that each time solving a problem adds to the achievement of a machine program as experience. For example, the keywords for mathematical operations are

a. Addition (+): combined, increased, more than, total of, sum, added to, together, plus
b. Subtraction (-): minus, less than, less, fewer than, difference, decreased, take away
c. Multiplication (x): multiplied, product of, times of
d. Division (÷): divided by, into, per, quotient of, percent (divided by 100), out of, ratio of

In addition to learning of words representing algebraic operations, a machine can be made to remember some generally used phrases which help to make the working equation out of a word problem. Some of which can be like

a. "the quotient of x and 4" to x/4
b. "the ratio of 5 more than x" to (x +5) / x
c. "six less than the total of a number and one" to (n + 1) –6 which then simplifies to n – 5

This way the remembering of a particular pattern for each kind of a problem can frame the equations for most of the problems.

The major causes of failing to solve a problem are the linguistic or algebraic capability. Sometimes a human can't solve the problem written in words while when given inputs and formula it becomes far easy. So the failure to understand the problem statement or failure to select the suitable relation results in failure to solve the problem. As in the following word problem:

The product of two consecutive odd integers is 1 less than four times their sum. Find the two integers. A student may know that this is a problem of quadratic equations and the formula to solve it. But the very first step of framing the equation can only be done by understanding of question.

C. Segregation:

It is mostly useful to divide the problem that must be solved into smaller modules separately and then to combine the fractional solution at the end into a complete problem solution. Unless we can do this, the number of combinations of the states of the components of a problem becomes too large to handle in the limited time. The use of a decomposition method is called planning. To elaborate planning of a problem with an example:

"Alfred buys an old scooter for Rs. 4700 and spends Rs. 800 on its repairs. If he sells the scooter for Rs. 5800, what is his gain percent?"

In order to calculate the selling price using cost price and gain percentage, the relation used is sp=((p%/100)*cp)+cp. To apply this relation one needs to identify the exact cost price and the profit percent. The identification of cost price is major part of planning in this problem. A program can discriminate noun Alfred and an old scooter here, whereas still the difference of a doer and an object needs a preliminary knowledge that scooter is a non living thing so it is an object and since Alfred buys it, the scooter has to be an object. Again buying and spending are related to cost price so the numerical value followed by buys i.e. Rs. 4700 and spends followed by Rs. 300, sum together to give cost price of the scooter for Alfred. Here the verb buy can be replaced by its synonyms like purchase and spend with expenditure. It is decided that the solution to this question demands selling price, it is observed by the phrase should he sell, whereas it is assumed that the language is simplest one.

This paper discusses the solving and analysis of problems, as the complexity of a problem can trouble a new learner and so does to a developing program. The developing program here refers to the one in learning phase i.e. making concepts and improving memory. To deal with the complexities of word problems several techniques of differentiating language are engaged.

II. FUZZY LINGUISTIC VARIABLES

The concern of linguistic variables and fuzzy condition statements is an effective modeling language. Basically, the formalism serves as a means of concluding and information optimization through the use of granulation.

CONFERENCE PAPER
II International Conference on
"Advance Computing and Creating Entrepreneurs (ACCE2013)"
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India

The system of linguistic variables and fuzzy if–then rules is centered on the use of information compression [4]. The concept of linguistic variable is applied to deal with situations where efficient and reasonable description in conventional quantitative expressions is desired. For example, '*height*; is a linguistic variable, its values can be very high, high, medium, low, very low etc., These values can also be represented by fuzzy numbers.

III. SCOPE VS. HITCHES

This paper contexts the comparison of a human mathematics learner to a learning program, where the program is a growing mind which uses its previous experiences of problem solving to solve new problems. Whenever a program manages to solve the problem or some human solves the problem in front of it, then it observes the methodology and new words engaged. So the intelligence according to this paper is completely defined on the happened events and remembered linguistic variables, the program do not make self analysis or research of any kind. The future implementations of this kind a program can be improved to make the program innovative and creative, which can be called a process of *thinking*. Thinking in the mathematical perspective is the result of having solved few problems with their proper conceptual and procedural understanding. When one owns the ability to understand something, can teach others too. It is same as a learner grows to become a teacher.

But apart from taking decisions the computer program made to solve statement problems is expected to be realistic, dynamic and reliable. Implementation of the concept of artificial machine, solving language problems demands great precision and consistency. In real time systems the expectations from a program like this rises to higher levels of perfection. A human body is a mind driven machine but is an assembly of a highly sensible nervous system, which takes best possible decisions even at the minimum experience.

IV. CONCLUSION

In this paper we have compared the artificial maths learning of an artificial mind with that of a normal human mind. All the applications yet introduced in the field of artificial intelligence or those using artificial brain somehow relate with the remembrance and decision making. This paper compares the behaviour of such a program with a maths student while learning and solving the problems. The complexities of developing such systems can be limited to storage of highly integrated and related data stored at times of each new experience gained by a program. The program not only stores the data but filters it based on the exhaust time and relevance of data stored. Decision making on the basis of available data is no trouble as many real time systems are already in effect with such functionality. The needs of memory size and the calculation power being dynamic in nature, comes up as a major challenge in development of these kinds of programs.

V. REFERENCES

[1] Children's Difficulties with Mathematical Word Problems. Sara Gooding University of Cambridge, UK

[2] Analysing Processes of Solving Word Problems in Mathematics Lessons Interaction: framings and reference contexts between real world and mathematics Ralph Schwarzkopf University of Dortmund, Germany.

[3] Mathematics Towards Perception of The World, Ivan MeznikBrno University of Technology, Faculty of Business and Management, Czech Republic

[4] Is there a need for fuzzy logic? Lotfi A. Zadeh *,1 Department of EECS, University of California, Berkeley, CA 94720-1776, United StatesReceived 8 February 2008; accepted 25 February 2008

[5] Knowledge Representation and Networked Schemes Dr.habil.sc.ing., professor Janis Grundspenkis, Dr.sc.ing., lecturer AllaAnohina-NaumecaDepartment of Systems Theory and DesignFaculty of Computer Science and Information.

[6] Does Decoding Increase Word Problem Solving Skills? JaLena J. Clement Wolbach, NE Math in the Middle Institute Partnership Action Research Project Report in partial fulfillment of the MAT Degree Department of Mathematics University of Nebraska-Lincoln July 2008

[7] Problem Solving, Search & Control Strategies : AI Course Lecture 7-14, www.myreaders.info/, RC Chakraborty