Final Report on Intelligent Tutoring System

Rajat Agrawal, Shubham Agarwal, Yash R. Singh

November, 2021

Abstract

This report focuses on the designing of an Intelligent Tutoring System (ITS) targeted at teaching algebraic equation to middle school students through real-life-story based problem formulation while also preaching social messages through them. Earlier, we conducted a literature review of the various ITSs in this domain of mathematics teaching to come out with a novel way of tutoring the target group through automated AI based solution. Now, in this report, more emphasis has been laid on discussing about the prototype built and its implementation details.

1 Problem Statement

This section describes the problem that we have chosen for solving. The literature referred to has been a source of ideation as well as motivation for this section. We define our problem in line with the works that try to solve Algebraic Linear Equations (Single and Double variable) problems. Specifically, we have targeted the medium grade (Grade 5 to Grade 8) learners, hailing from the rural parts of India.Most of the works as mentioned previously focuses on teaching algebra and equation solving using the traditional tutoring pedagogy for automating the tutoring of school mathematics.

We identified that this method has three primary issues:-

1. Using variables that have little or no physical connection leads to confusion and learners start hating the subject
2. Teaching of the concepts abstractly which lacks real world connection, thereby demotivating the learners
3. It loses out on addressing the social and emotional connected topics that could be conveyed otherwise

Hence, the problem statement formulated is:-
Tutoring the middle grade students to solve linear equations in single and double variables by using real world situation-based word problems by establishing physical connection between the variables and real objects, while also emphasising on teaching social and ethical values through the same.

2 Design of the software

Below Flow Chart[1] explains the working of a the ITS. Initially, a user is asked if they possess an ID. If they don't, then a new profile is created for them and an ID is provided to them. Then users are presented with three options, i.e. to start practicing questions, view their progress report and logout from the ITS. Once user selects the option of start practicing, system checks the academic level and social level of the user, if available to select the appropriate question from domain model. To make the learning process more comprehensive, user is initially asked to find the number and semantics of the variables followed by formation of linear equation and then the process of solving equation begins. Here, users are presented with five options once they are read the word problem, which are to enter answer in input dialog box, to ask for hint, to ask for solution of current step, or to ask the system

[1]For the sake of simplicity, the flowchart has been compressed to the main components only. Also, the last part of value based problem component has been limited to view to simplify the block diagram

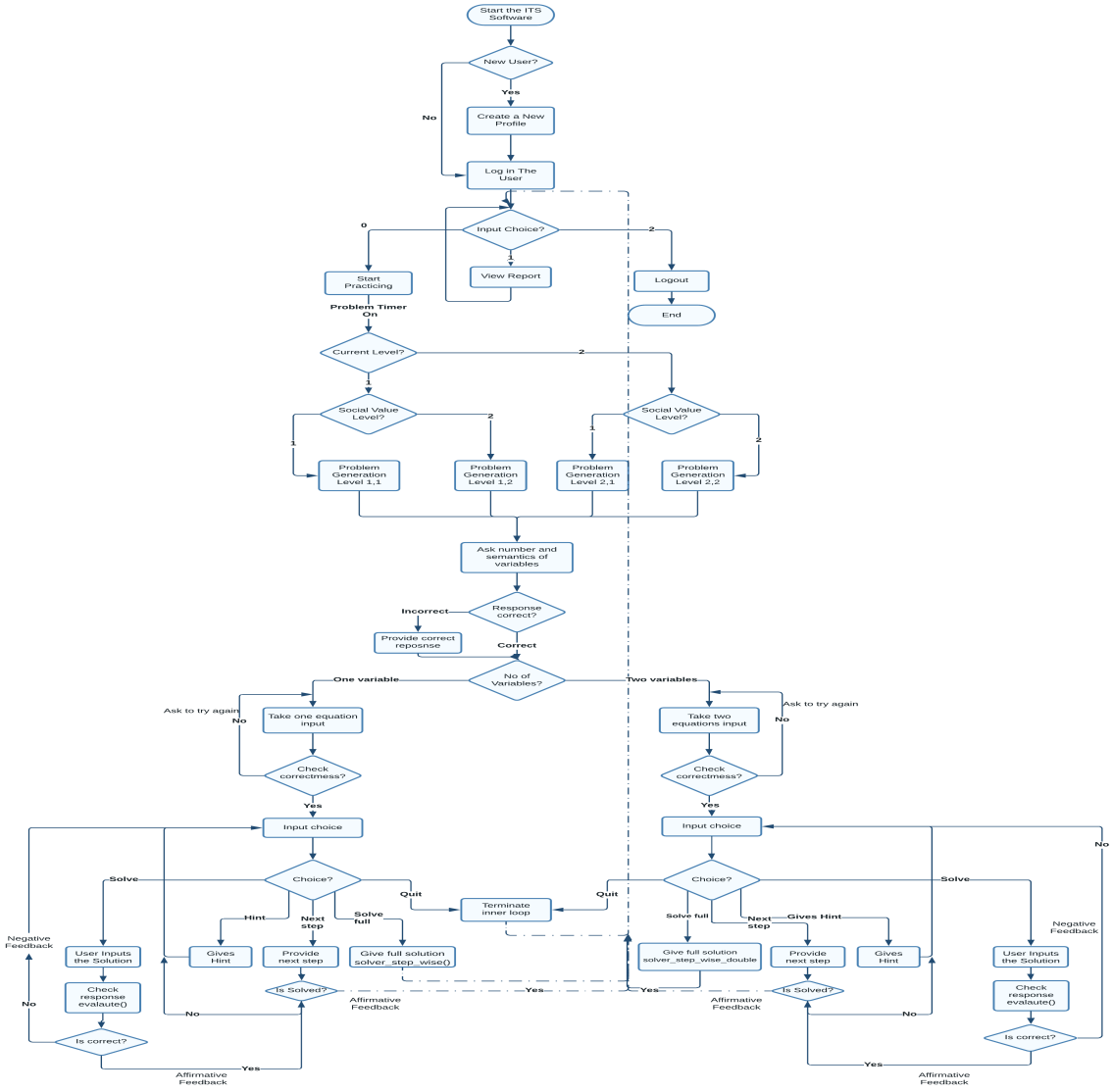

Figure 1: Block diagram to show the design of your software

to solve problem completely, or to quit this question. This process continues unless the question is solved. After this,the score of user is changed on the basis of total time spent in solution, number of hints asked and number of steps taken to solve the question. This also ensures that the tutor rewards the user who solves in less steps, and penalise those solving in more than usual steps. Also a social question is asked to user after this to gauge the understanding of social issue. This ends the process for one question after which user is again presented with options to continue solving, to view report and to quit.

3 Process to execute the program(s) with required tools

This section mentions the sequence and description of the flow to be used for execution.

- **Step 1:** Unzip the folder Group_14_AI. The folder has three program files: functions.py, requirements.txt, ITS.ipynb along with the report pdf, demo video files and flow chart png.
- **Step 2:** You can either use any local terminal or Google Colaboratory(preferred) to run the python notebook. Upload or use the functions.py, requirements.txt files accordingly.

- **Step 3:** Next, use requirements.txt to download all the required dependencies by running the first cell of the noteboook: **`!pip install -r requirements.txt`**.
- **Step 4:**Run the import command to import all the functions from **functions.py**.
- **Step 5:**Run the next cell to make a call to the Tutor function **tutor()** and follow the steps that follow.

4 Novelty of idea

This section describes similar tools and/or software that is already available, and how our proposed idea is different.

4.1 Domain Model

1. Most of the tools available, like ActiveMath offers only multiple-choice questions and few interactive exercises, hence limiting the usability and scope of tutoring.
2. Another tool, Equation Guru supports various forms of linear equations. But, its scope is limited to solving equations with one unknown only and is not extensible to many real-world algebra problems involving two or more variables.
3. Other tools like Equation Solving Tutor (EST) or E-Sit limits itself to solving only certain forms linear equation in one variable problems and lacks the depth in terms of problem space and the knowledge being imparted.
4. Our proposed solution's novelty lies in addressing all these concerns as it supports solving multiple forms of linear equations is both single and double variable problems and gauges to solve real-world problems by unifying equation solving with storytelling. Particularly, the aim is to make the random variables denote some existing physical objects already known to the learner. This will also help in improvising the linear equation formation capability of the learner which has been ignored in most of the previous works.
5. Also, we enable the learning of ethical and practically significant knowledge of the problem solving techniques. For evaluating based on social awareness, system asks the learner a MCQ at the end to gauge the understanding of the issue.
6. In previous works like E-Sit, the next expected action from the student is specific. This may lead to wrong pedagogical decisions and strategies. In some MCQ based agents like MathFluency, the final answer along with the steps are pre-encoded into the domain model and presented to the learners as it is, leaving very little room for multiple solution strategies.
7. In the proposed domain model design supports both word problem to equation formation and equation solving. Domain model contains rules to verify the student's solution as well as provide the ideal solution when asked by the student. The model knowledge supports step-by-step solutions to the equations, and also supports non-similar but correct responses to be evaluated, thereby being flexible in the number of steps taken to solve the equation.

4.2 Student Model

1. Another important aspect of any ITS is the way it teaches and how the student's progress gets tracked. From the ideas gathered from the literature, most of the student models lacked an important feature of including the background of a student.
2. In tutors like PAT2Math, the level of the student is not taken into account, and problems are presented in a fixed sequence as decided by the Professor
3. Our proposed model obtains ideas about the learner's state of thinking and social awareness level to help target the areas that need to be worked upon.

4. The student model's dynamic attributes tracked include the student's progress track using the responses and timing for each step of solution, the feedback and help requested or delivered, and the correctness of each step.

5. We also check regularity of a learner by keeping track of the frequency and gaps of tutoring timings.

4.3 Tutor Model

1. Some of the prominent ITSs prevalent like Math Fluency uses MCQs which has a drawback as it leaves a room for guesswork and restricts students from trying again. Here the system lays down a blind eye to the steps taken to derive the solution, which is a critical aspect as the solution can be correct even if the method is wrong or lengthy.

2. Cognitive tutor generates strong hint messages sequentially, by exactly telling what operation to perform at end of sequence. But this approach contradicts with the principles of effective teaching. That is, the tutor should not provide strong hints for the solution of equations when students do not need them and the students are not allowed to learn to reason for themselves.

3. Both ActiveMath and Mathesis lack any support mentoring techniques to accompany students in the solution modeling steps, making them less useful as compared to others.

4. The tutoring model acts as a driving engine for our entire system. It focuses on deciding facts like choosing appropriate materials, evaluating the performance, and providing guidance and feedback to the student as in existing works.

5. The proposed ITS design is clean and easily adaptable. For any learner, the tutor model provides questions based on the student model attributes, and guides throughout the step-by-step solution providing custom feedback and evaluation.

6. The tutor model checks the solution if the current step equation provided is consistent with the previous step equation. If it preserves the solution, but makes the equation more complex, a warning is issued and the student is allowed to proceed further. But if it doesn't, then the student is penalised for making the mistake and asked to try again.

7. The Time to Respond to the question, number of wrong attempts, number of times hints are asked etc. are used to score the student, and the tutor model decides the next problem to be presented to the student based on the score and level of the student.

8. Also, the model incorporates a "level-based" strategy to gamify the tutoring process by providing scores after solving each question and setting milestones for each level at a certain score. Penalising the student for being irregular is also used to inculcate consistency in learners.

5 Result and Analysis

The proposed idea was implemented using python and tested over a small group of audience. Actual stakeholders could not be involved due to time and resource constraints. The audience used for evaluation and testing involved 2018 Batch students of BITS Pilani. In all, a group of 10 students were used independently, and their feedback and observations were used for the purpose of evaluation of the model.

1. Out of 10 students, all of them liked the concept of the ITS, 7 liked the functionality, and 4 liked the simplicity of the model.

2. Out of 10 students, 2 complained about the usability and wanted to extend the problem space, 4 of the students wanted to use a better interface(NOTE: front-end has been omitted for the sake of time constraint).

3. Out of 10 students, all of them liked the domain model the best, 6 of them liked the student dynamic modelling attributes, and 4 of them liked the tutoring feedback mechanism the best.

From the analysis of the team and the test-user audience involved, following conclusions were drawn:-

1. Our proposed model tries to take in the pros of the prior design models and address their cons by enabling better functionality, design and layout for the domain, student and the tutor models.
2. Few of the important aspect to highlight are the questioning, feedback and evaluation components of the domain model. The inherent flexibility of solution checking mechanism, and soft feedback mechanism along with the problem delivery methodology stands out and makes it a win-win situation for the proposed model.
3. Apart from the domain model, the student model takes into account various attributes to model the student behaviour and track its progress minutely. It uses the same to provide a detailed personalised report for each user. It also tracks some previously unaddressed parameters like tracking regularity, student moral and knowledge levels.
4. Although the tutor model lacks the UI front-end part(due to time constraints), it still is able to show up for the entire ITS, like trying to act like a human monitoring tutor. It well inculcates the basic values of an ITS by efficiently utilising the domain and the student model. Apart from that, the tutor models feedback mechanism, solution checking and evaluation processes tries to outperform the prior works by working on their potential limitations.

6 Other discussions

We describe some of the implementation related aspects and the potential future prospects in this section.

1. The implementation was limited to the building of the functionality components only, i.e. the domain, student and tutor models.
2. Abiding by the intention behind this project and time constraints, the UI/UX front-end section was limited to ideation only.
3. The prototype built is designed as an exhaustive ITS that covers only the essential functionalities of back-end development, with any additional tweaks left to the ideation part only.
4. The database of the question bank has been designed to only cover certain aspects, and can be easily expanded as per the requirements of the user.
5. All the models' designs are prototype-ready for functionality coverage of the ITS, but need to be expanded comprehensively for any product-based usage.

Team Information

This project report is a part of the course Artificial Intelligence(CS F407) for the Semester-I 2021-22 under the guidance of Dr.Vishal Gupta

Student group Memebers:

1. Agrawal Rajat Ramesh - 2018A7PS0182P
2. Shubham Agarwal - 2018A7PS0301P
3. Yash Raj Singh - 2018A7PS0214P