AUTOMATIC EVALUATION AND ERROR IDENTIFICATION OF SOLUTIONS TO SINGLE-VARIABLE ALGEBRAIC QUESTIONS

Erabadda Lekamalage Buddhiprabha Harshani Erabadda

(158020V)

Thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science (Research) in Computer Science and Engineering

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

February 2017

DECLARATION

“I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

Name: E.L.B.H. Erabadda

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the supervisor: Date:

Name of the supervisor: Dr. Surangika Ranathunga

Signature of the co-supervisor: Date:

Name of the supervisor: Prof. Gihan Dias

Abstract

There are two types of single-variable equation solving questions that are present in the Ordinary Level mathematics curriculum in Sri Lanka: linear equations with fractions and quadratic equations. Answers to these questions are open-ended and multi-step in nature. This thesis describes a mechanism that evaluates answers to these two types of questions and awards full/ partial credit.

It is quite common that students make mistakes in their answers, which results in partial credit. They may repeat the same errors if they do not receive feedback on their mistakes. Therefore feedback in student errors is important for any subject. This thesis introduces a method to automatically identify the errors that the students make in their answers for the aforementioned two types of questions. To the best of our knowledge, this is the first work on automatically identifying student errors in complex multi-step solutions to single-variable equation solving questions.

Our evaluations show that the system we have implemented is capable of awarding full/ partial credit to student answers according to a marking scheme and also to identify errors in student answers with minimal teacher intervention. These evaluations were carried out using student answers from different sources.

Keywords- Computer Aided Assessment, Error Identification, Computer Algebra Systems, Partial Credit, Multi Step

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisors Dr. Surangika Ranathunga and Prof. Gihan Dias, for the continuous support given for the success of this research. This would not have been a success without your tremendous mentorship and advice from the beginning.

This research was supported by LK Domain Registry, University of Moratuwa Senate Research Grant, and DL4D (Digital Learning for Development) Grant. I sincerely thank colleagues from LK Domain Registry and mentors from DL4D committee for the support given.

I would like to thank all staff from the Department of Computer Science and Engineering for their kindness expressed in all occasions.

Special thanks to my loving family: my parents, husband, brother, and sister. Your encouragement always motivated me to do my best. I would also like to thank all my friends who encouraged me all times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

A/L – Advanced Level

ATPM – Approximate Tree Pattern Matching

CAA – Computer Aided Assessment

CAS – Computer Algebra System

CM – Cross Multiplication

GCE – General Certificate of Education

ITS – Intelligent Tutoring System

LAESA – Linear Approximation and Elimination Search Algorithm

LCD – Least Common Denominator

LSA – Latent Semantic Analysis

MCQ – Multiple Choice Questions

MOOC – Massive Open Online Course

O/L – Ordinary Level

1. INTRODUCTION

This thesis addresses the problem of automatic assessment of multi-step answers to algebra questions and identifying errors in student answers. Specifically, this research focuses on two types of single-variable equation solving questions in the Sri Lanka G.C.E. Ordinary Level (O/L) mathematics examination:

- Linear equations with fractions
- Quadratic equations.

Answers to equation solving questions contain large amounts of symbolic expressions. All these questions evaluate student's knowledge in different concepts. For example, in linear equation solving questions with fractions, the concepts that are being evaluated include concepts such as "taking common denominator" and "taking cross multiplication". The students receive marks for the demonstrated ability in applying these concepts correctly in their answers.

When answering equation solving questions, there can be multiple ways of arriving at the final solution. Application of the concepts that are taught in the classroom can be different from one student to the other. For example, if there are three fraction terms in a linear equation solving question, the student can demonstrate the ability to take common denominator correctly using any two terms. Also, one student may simplify the numerator at the same step where she takes the common denominator, while some other student may take few additional steps to simplify the numerator. Therefore, the number of steps in the answer is not fixed. Having different approaches and a variable number of steps makes it difficult to grade student's answer against a single teacher-provided answer.

Students receive full/ partial marks according to a marking scheme depending on the correctness of the application of concepts. If the student answer is wrong or partially correct, it is required to identify the steps where different concepts have been applied to determine whether marks should be awarded. When a student makes a mistake, the type of the mistake can be informed to the student so that she can try to correct and learn from it.

1.1. Problem Definition

Despite the various attempts to make education better in the country, the pass rates of G.C.E. O/L mathematics paper in Sri Lanka has remained around 50% over the past few years [19]. Until 2016, students who fail mathematics were not eligible to sit for the Advanced Level (A/L) examination. Therefore it is quite important to work towards increasing the number of students who pass mathematics subject in G.C.E. O/L examination.

For many students the most common way of preparing for an exam is working on past exam papers. This method is successful only if students get their answers marked by the teacher and get personalized feedback of some form. Considering the number of students assigned to a teacher of a typical school/institute, it is not feasible for the teacher to individually go through all answers attempted by all students.

E-learning has been there for a long time now, which includes using technology for education. With advancements and wide-spread use of technology, e-learning has become popular in many contexts. E-learning allows students to learn at their own pace, which is not possible in a large classroom. In a traditional classroom, if majority of the students understand a particular concept, the teacher moves on to the next concept without having time to concentrate on the students who may require further explanations. This is unavoidable due to the large number of students in a typical classroom. As a result, e-learning has become quite popular around the world.

A potential solution for the low pass rates in mathematics is developing an e-learning platform where students can work on past/ model exam papers and get their answers automatically evaluated with feedback on their work.

1.2. Motivation

Currently there is research done in automatic grading of mathematics answers, but there are many unsolved issues. Questions where the problem has a single mathematically correct answer, where the answer is given as a single line of expression, have been widely explored in related research [1].

There are software systems that are capable of symbolic manipulations. They are known as Computer Algebra Systems (CASs). Many existing studies in the literature confirm that incorporation of CASs for grading answers to mathematics questions in particular has been there for a long time. Most of the existing work looks at using a CAS to evaluate answers that are provided as a single line of expression [22, 23, and 24]. As mentioned earlier, most of this existing research supports only questions that have single mathematically correct answers given as a single line of expression.

Also according to the Sri Lankan O/L evaluation criteria, integration of a marking scheme for the questions is essential. This gives flexibility for the teacher to evaluate different aspects of student answers in different questions. This is required not only for O/L, but also for any exam that intends to measure the student’s knowledge in different concepts. This aspect has not been given focus in the existing research.

Identification of errors in student answers has also gained much interest according to existing research. This is due to two main reasons: it helps teachers to understand the types of mistakes that their students make and it helps students to understand their own mistakes and correct them. Automating error identification is widely present in systems that evaluate software programs. However, much of the research related to mathematics is from pedagogy experts. They focus on identifying the common types of errors that students make and categorize them. The most commonly used technique for this purpose is interviewing students [39, 40]. Therefore, automation of error identification has plenty of room for innovation in the area of Computer Aided Assessment (CAA) for mathematics. According to the best of our knowledge, there are no systems that are capable of automatically locating errors in multi-step open-ended answers to mathematics questions of the types we focus on, without comparing against a teacher-defined answer/ expression.

1.3. Objectives

The objectives of this research are as follows.

- To develop a module that is capable of evaluating linear equation solving questions with fractions.

There are certain concepts that receive marks in this question type. The system should identify the steps where the student has applied the concepts in her answer steps. The system should award full/ partial credit according to a marking scheme, which is defined by a teacher. The system should be able to evaluate one answer at a time and award marks.

- To develop a module that is capable of evaluating answers to quadratic equation solving questions.

This too requires identification of particular steps and awarding full/ partial credit, similar to linear equation solving questions. However, the concepts are different and hence a different module is required.

- To develop a module that is capable of locating errors in student answers.

When a student receives partial credit to the answer, it suggests that she has made a mistake. In such cases the system locates the step where the mistake has been made and suggests the reason for the error. This error location and identification should be done separately for the two types of questions addressed in the research.

1.4. Contributions

To address the aforementioned objectives, a system has been developed with the following capabilities:

- The system evaluates multi- step answers to linear equation solving questions with fractions and quadratic equation solving questions. The system needs teacher involvement only during the question set up stage. The teacher is required to enter the question and specify marking scheme so that the system can use this information for answer assessment. The marking scheme is presented to the system in XML format and the system awards full/ partial credit to the student accordingly.

- If a student fails to obtain full credit, the system identifies the step where the student makes the error and suggests the reason for the error.

1.4.1. Articles

This research has produced the following refereed publication so far:

- Erabadda, B., Ranathunga, S., & Dias, G. (2016). Computer Aided Evaluation of Multi-Step Answers to Algebra Questions. In 16th International Conference on Advanced Learning Technologies (ICALT). 199-201. IEEE.

1.5. Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 reviews related work in the area of CAA: assessment of answers to different types of questions, the state of CAA in mathematics, CASs and how they have been used in CAA systems.

Chapter 3 explains the approach used and the design of the answer grading modules. How the student-provided answers are awarded full/ partial credit according to the teacher-provided marking scheme is described in the chapter. How a CAS is incorporated into the system has also been discussed.

Chapter 4 explains the implementation details of the error identification module. How the system identifies the error in the student answer steps is discussed here.

Chapter 5 evaluates the work presented in the thesis. It contains subsections for evaluating linear equation solving questions with fractions, quadratic equations, and error identification in partially correct answers.

Chapter 6 provides details on contributions of the thesis.

Chapter 7 concludes the thesis with a look into future work.

2. LITERATURE SURVEY

2.1. Overview

This chapter provides details on concepts and techniques of Computer Aided Assessment (CAA) of student answers. The chapter mainly focuses on grading answers to mathematics questions. Other systems that are involved in CAA are also briefly described.

Section 2.2 provides an overview of CAA in general. The types of questions that are evaluated by CAA systems are also mentioned in the section.

Details of existing systems are discussed in the sections 2.3 and 2.4 for the two main types of CAA: intelligent tutoring systems and other computer aided forms of instructions. More focus has been given to systems that grade answers to mathematics questions.

There are systems that are capable of symbolic manipulations, which are known as Computer Algebra Systems (CASs). These systems can be used inside CAA systems that evaluate answers to mathematics questions. Many existing CAA systems that grade mathematics answers make use of CASs. Section 2.5 provides details of CASs.

Some systems give feedback on errors, in addition to the grade. Section 2.6 gives details on error identification in general and types of errors that students make. The section also describes existing work that focuses on error identification.

When awarding marks to student answers, teachers need to decide how exactly marks will be awarded. This should be done before grading is started. Marking schemes with those information, also known as rubrics, can be used for this purpose. A brief introduction to rubrics is provided in section 2.7.

2.2. Computer Aided Assessment (CAA)

There are many systems that evaluate answers in different domains. Most of the existing studies focus on evaluation of answers to Multiple Choice Questions (MCQ) and text-based answers. However, there are systems that evaluate answers to mathematics questions as well.

According to Bennett [1], responses to mathematics questions can fall into the following four categories.

- where the problem has a single mathematically correct answer that can take many different surface forms
- those calling for one or more instances from a potentially open-ended set that meets given conditions
- those requiring the student to show the symbolic work leading to the final answer
- those asking for a short text response

According to the author, only the first type of answers has been widely analysed.

In the context of e-learning, there are two types of grading systems that grade answers in different domains. The two types are:

- Computer Aided Assessment Systems
- Intelligent Tutoring Systems

Sections 2.4 and 2.5 give details on existing systems that fall under these two categories.

2.3. Computer Aided Assessment Systems

Apart from trivial MCQ graders, there are many computer aided evaluation systems that are used for various answer grading tasks. In broad, there are following types.

- Systems that grade summaries/ essays/ texts
- Systems that grade computer programs
- Systems that grade answers to mathematics questions

2.3.1. Systems that Grade Summaries/ Essays/ Texts

Summary and essay writing is available in many English language examinations. There are CAA systems built to grade these summaries and essays automatically in order to ease the tedious work of teachers. Several techniques such as Latent Semantic Analysis (LSA) and n-gram co-occurrence have been used for this.

LSA refers to a technique in natural language processing that extracts and represents the contextual-usage meaning of words [2]. A grader developed by Zipitria et al. [3] makes use of LSA to grade summaries written by students in Basque language. A grade is given to the student based on cohesion, adequacy and use of language.

Summary Street educational software developed by David and Ellen [4] also makes use of LSA. It allows students to get automatic feedback on their summaries so that they can improve the summaries before the teacher reviews them. This has two advantages. From students' perspective, they receive feedback in easy-to-grasp, graphic display so that they can improve their writing skills on their own. From teachers' perspective, it eases the workload of the teacher as the initial feedback is already given by the system and hence student summaries are improved when they reach the teacher.

A technique suggested by Perez et al. [5] makes use of "BLEU" (**Bi**l**i**ngual **E**valuation **U**nderstudy) algorithm to evaluate free-text answers. BLEU was originally suggested to evaluate and rank Machine Translation systems [6]. Perez et al. [5] applied the same method modified to assess short essays written by students. The student's essay is compared with the model essay stored in the system. The authors state that their method is capable of achieving high correlation with human graders.

There are many other systems that are developed for text grading: by Mohler and Mihalcea [7], Pado and Kiefer [8], and Pulman and Sukkarieh [9]. These systems cannot be used to grade answers to mathematics questions due to two main reasons.

- answers to mathematics questions mostly involve symbolic expressions, and not text
- symbolic expressions cannot be compared against a string due to equivalence issues that are present in mathematics

Therefore these systems that are related to essay grading are not explained any further.

2.3.2. Systems that Grade Computer Programs

There are different types of programming assignments that are given to students in Computer Science in order to assess their practical programming capabilities. It is difficult for the teacher if she is going to assess each student's answer individually as the process is time-consuming and cumbersome. There are many systems developed to address this issue and ease up the workload of the teacher.

A survey paper by Ala-Mukta [10] discusses several approaches for assessing programming assignments automatically. He discusses the existing approaches with regard to features that each assignment intends to test: functionality, efficiency, and testing skills.

Functionality

With regard to this feature, Ala-Mukta discusses systems that focus on correcting programs that function according to given requirements. This is usually done by running the program against several test data sets. How well the system assesses functionality depends on the test case design. There are several systems that use this technique such as Try [11], CourseMarker [12], Assyst [13], and Online Judge [14]. These systems assess the functionality of the program by comparing the output, using pattern matching or regular expressions.

There are systems that are capable of assessing functionality of single methods and single expressions. Quiver [15] and Bettini et al. [16] have developed two such systems that are capable of assessing Java functions. WebToTeach [17] and ELP [18] assess individual expressions by combining the student code fragment to an instructor template before compiling.

Efficiency

In order to assess the efficiency of the student's program, it is run against different test cases and the behavior is measured during execution. The results are typically compared against a model solution. Therefore efficiency measurement depends heavily on the test cases and the model solution.

A typical efficiency measure is the running time of the program, which is measured either based on clock-time or CPU-time. Assyst [13] and Online Judge [14] make use of CPU-time to assess efficiency of student programs. However CPU times can be affected by some features of the program. (E.g. implementation of data input/output actions). To avoid these issues Hansen and Ruuska [20] have implemented a system that offers a common input/ output model for use in assignments.

Efficiency can also be measured by different structural behaviors inside the program. For example, Assyst [13] allows calculating how many times certain blocks and statements get executed.

Testing Skills

Although automatic systems that assess student programs depend on test data, it is required that the students correctly develop test cases on their own. Assyst [13] measures this feature by determining how well the student’s test dataset covers all lines in the student’s own program. Chen [21] describes a system whereby the student’s test cases are run against a teacher provided buggy program in order to reveal how many bugs get captured by the test cases.

There are many systems and techniques being developed to assess computer programs in the context of assignments in computer science courses. However, the techniques they use cannot be applied for systems that grade answers to mathematics questions due to the same reasons mentioned in the previous section. Therefore this type of systems will not be further discussed.

2.3.3. Systems that Grade Answers to Mathematics Questions

There exist systems that are capable of grading answers to mathematics questions. However, most of these systems are capable of evaluating answers that are entered as a single line of expression. Some of these systems are capable of awarding partial credit to students' answers, while some systems are limited to awarding either full or zero marks. Following two sections include brief descriptions of existing systems, based on the type of answers they can assess.

2.3.3.1. Systems that Evaluate Structured Answers

In this section, existing systems that evaluate structured answers are considered. The difference between this type and the two types mentioned above (systems that grade essays/ summaries and systems that grade computer programs) is that structured answers have a pre-defined format, entered as a single line. This allows comparison against a teacher-provided answer to a certain extent. For example, in mathematics, an answer by a student can be compared with the teacher's answer for equivalence.

Sangwin et al. [23] discuss the CABLE (Computer Algebra Based Learning and Evaluation) system that evaluates student-provided answers. CABLE makes use of Axiom CAS. The student's answer and teacher provided answer are taken and one is subtracted from the other. The resulting expression is simplified as much as possible and the final value is checked. If the final value results in a zero, it means that the two expressions are algebraically equivalent. The answers can also be evaluated using "answer tests", which are predefined CAS procedures. An example answer test may have options to indicate whether floating-point approximations of coefficients are allowed and if so, the level of accuracy required. In that case, the teacher just has to change the values of the two parameters that are passed to that particular answer test.

This system evaluates answers entered as single algebraic expressions, while multi-step answers have not been discussed. This system specifies that the answer is either correct or wrong. Therefore the system does not award partial marks.

Another research by Sangwin et al. [24] discusses STACK (a System for Teaching and Assessment using a Computer algebra Kernel), which uses the Maxima CAS to evaluate student-provided answers. This system uses some predefined "answer tests", similar to CABLE, which defines properties of an expression (e.g. factored, expanded, etc.).

However, the system takes each line of answer into account to determine the similarity between the student-provided answer and the teacher-provided answer using answer tests. STACK handles three types of questions, which are as follows:

1. The student is asked to manipulate the written syntax of a given algebraic expression

   E.g. rearranging an equation, adding/ subtracting identical terms to/from both sides of an equation

2. The student is asked to perform a calculation

   E.g. adding two rational numbers, solving an equation

3. The student is asked to establish some property of a number or algebraic expression

   E.g. finding the highest common factor of a pair of integers, identifying a part of an expression such as the denominator

STACK has been added as a plugin to Moodle [25], which allows it to be used in an e-learning environment. This enables STACK questions to be added as part of a Moodle quiz. A response processing tree should be defined by the teacher for each question. This tree specifies possible answers at each node (potential responses). To come out with a comprehensive response processing tree, the teacher has to address all possible answers that a student can give.

For better evaluation, the system requires much interaction with the teacher at the time of question creation in order to build a better response processing tree. The teacher has to consider all possible paths a student can take when moving from one step to the other. If the teacher misses a path that is correct, the system has no way to recognize which results in student not being awarded the marks. Hence the question types that can be evaluated with STACK are limited to those that require 2-3 steps at a max. Otherwise it will be cumbersome for the teacher to go through all possible next steps, which can result in a sparse tree. The system relies totally on the response processing tree created by the teacher. Evaluating multi-step open-ended answers is not addressed as a result. However this system can give partial credit to the student answer. That is by specifying the amount of marks to be awarded at each node of the tree.

James [45] has developed m-rater, which is capable of scoring items for which the answer is an equation or a graph. When providing answers as equations, an editor was given to students so that data is properly cleaned before m-rater assesses it. The system has “keys” defined. These keys can be used to provide model answers to the problem. The author has developed generic keys that can be used by most of the mathematics questions. However, m-rater cannot evaluate student solution steps. It is limited to scoring only the final answer.

2.3.3.2. Systems that evaluate free response answers

Unlike the structured answers, free-response answers do not impose restrictions upon what the student can write and the length of what the student writes. This section looks at existing systems that evaluate open-response answers to mathematics questions.

Livne et al. [22] discuss a parser (RUReady) written to grade mathematics answers. This parser is capable of grading mathematics responses constructed by students, as opposed to grading using a predefined set of answers. The response provided by the student is compared with an instructor-provided reference expression. Thus the parser requires two inputs: reference and response.

This parser is capable of assigning partial marks to the answer based on errors in each response. The system does this by first classifying the response’s individual elements into correct, missing, unrecognized, wrong, and redundant. The parser’s output consists of 3 parts as mentioned below.

- Matching: reporting whether the response is a legal expression
- Numerical Evaluation: deciding and reporting whether the response and reference are mathematically equivalent.

- Analysis: error flagging, i.e. classifying individual elements of the response into correct, missing, unrecognized, wrong, and redundant elements.

The parser compares the response syntax tree in relation to the specific reference syntax tree using Approximate Tree Pattern Matching (ATPM). The ATPM approach computes the edit distance, which is the minimum number of edit operations required to transform the response syntax tree to the reference syntax tree. The score for the answer is calculated as a weighted sum of two components: overall correctness and fraction of correct elements. This parser is limited to scoring only final answers. Marking answers with steps is not addressed in this research.

A research by Klai et al. [48] presents AIM (Alice Interactive Mathematics), a web-based system for grading tests that includes mathematical content. They have five types of questions defined: algebraic, matrix, multiple-choice, multiple response, and constant. How they grade answers in the algebraic category is that they subtract the student's answer from the teacher-provided answer and use the CAS called Maple to check whether result is zero. If the result is zero, the student receives full marks, and otherwise student receives a zero. There is also the possibility to incorporate penalty to an incorrect answer, thereby giving a partial grade if students took multiple attempts before writing down the correct answer. However, this system is limited to evaluating answers entered as a single line of expression.

A recent research done by Lan et al. [26] describes a data driven framework for Mathematical Language Processing (MLP). They have enabled identifying errors in single-step or multi-step solutions by clustering the answers. The main idea of this research is to locate the possible errors of an answer. The order of answer steps is not considered when allocating marks and only the presence of steps in an answer is considered. The authors claim that their system is the first to evaluate open-response mathematical questions and to give partial credit. However, each step of a multi-step solution is taken individually and given credit according to where in the cluster it belongs. How a solution can have variable number of steps has not been addressed here.

The authors follow a 3-step approach in their solution:

Step 1 - Convert each solution into a series of numerical features

When extracting features, the authors consider only mathematical expressions and ignore the texts that are there in student answer steps. MLP identifies the unique mathematical expressions contained in the learner's' solutions and uses them as features.

Step 2 - Cluster the features from several solutions

Once the features are extracted, they are clustered using two different approaches: MLP-S (based on a similarity matrix) and MLP-B (a Bayesian non-parametric-based approach).

Step 3 - Grade the solutions

Once the clusters are determined, the teacher should evaluate and award marks to one answer from each cluster. Then the system automatically awards marks to the rest of the answers by considering the distance to teacher-marked answer and the position of the answer in the cluster. Hence the system requires much teacher involvement during marking.

2.4. Intelligent Tutoring Systems

Apart from the computer-aided assessment systems discussed above, there are Intelligent Tutoring Systems (ITSs), which refer to software systems that simulate a teacher [27]. These systems differ from other CAA systems since ITSs learn as they operate. The downside of ITSs is that they can be manipulated by students to provide answers through hints and providing hints can discourage students from exploring more and trying on their own [28].

Koedinger et al. [29] discuss a system that uses a data-driven AI based approach. The system requires high teacher intervention during question creation. The system keeps producing candidate steps until it gets a positive feedback or runs out of options. If it runs out of options, the teacher has to tell the next step so that the system can learn a new rule. This requires excessive involvement by the teacher. Also since the system talks about data-driven approaches, it depends on having a great deal of past data.

In a research by Melis et al. [30], a CAS is employed for evaluations, which is presented to the user for entering data to the system. Each step is evaluated with the correct answer to see if both match. The system tells if the answer is correct or wrong (depending on whether it matches the answer/not). The student is required to tell when he has completed so that the system can match the final step with the expected answer. Then the student is told whether the final answer is correct or not. Evaluating certain concepts taught in the classroom and giving partial credit to answers have not been addressed here.

In a research by Ambekar [31], a method known as AI Grading (or ML Grading) is presented. This is proposed for Massive open online courses (MOOCs), where there is a large number of answers. Due to the large number of answers in MOOC environments, it is not possible for the teacher to go through each and every answer and grade. AI grading is a system that uses machine learning techniques to grade student provided essays. The author specifically looks at EdX [32], a widely used MOOC environment. The idea of AI grading is to use a model trained with few submissions to grade remaining large portion of data.

Some systems first break down questions that require multi-step answers into multiple questions so that the answer for each sub-question will be a single line of expression [22, 48]. But this method does not compensate the process of marking a multi-step answer, where the student gives all steps at once. The complexity arises as the student can follow more than one approach towards the answer with a variable number of steps. Due to this nature of multi-step answers, it is not possible to assess the student’s answer against single reference answer.

2.5. Computer Algebra Systems (CAS)

A Computer Algebra System (CAS) is a software package used for mathematical manipulations. The primary purpose of a CAS is to enable automation of tedious and time-consuming algebraic manipulations. A CAS can be viewed as a calculator. The difference between a calculator and a CAS is that a CAS is capable of symbolic manipulations rather than being limited to numeric manipulations. As mentioned in previous sections, CASs have widely been used in CAA systems.

There are two forms of CASs.

- CASs that can be incorporated into computers
- Handheld devices that are capable of doing symbolic computations (Symbolic calculators/ CAS calculators)

2.5.1. Computer Algebra Systems that Can be Incorporated into Computers

There are many CASs available such as Mathematica [41], Derive [42], and Sympy [43], which can be incorporated into computer systems. Some CASs are commercial while some other CASs are free and open-source.

There are many CASs that work inside other systems. Some of them cost for usage while others are free and open-source. Table 2.1 shows a list of such commercial and free Computer Algebra Systems that are available.

The different types of CASs have been designed using different types of technologies. The offered functionalities of the CASs are more or less common among the CASs. Certain functionalities that are common behave in different ways in the different systems. For example, all CASs have a functionality offered to simplify a given expression. The level of simplification carried out by the CAS is different among the systems. However, most of the systems offer the capability to control the level of simplification.

Table 2.1: Different types of Computer Algebra Systems that can be used inside other systems

Commercial	**Free (for both commercial and non-commercial applications)**
MATLAB	Maxima
Wolfram Alpha	Sympy
SyMAT	Axiom
Maple	Cadabra
Magma	FriCAS
Macsyma	GAP
MathCad	GiNaC
Mathematica	Scilab
Fermat	SageMath

When selecting a CAS for a CAA system, many factors need to be considered. The most important factor is the types of operations expected from the CAS. For example, if matrix operations are needed for the system being built, the developer needs to ensure that the selected CAS supports them. Another important factor is the support by CAS developers. If the developer community is not active, it may suggest that the CAS is at an obsolete condition. In such a situation, it is not advisable to use the CAS in the product.

2.5.2. Handheld Devices that are Capable of Doing Symbolic Computations

There are also handheld devices that include CASs, which can be used without a computer. These devices are a type of graphing calculators. Not all graphing calculators include CASs, but there are several graphing calculator types that include CASs. The graphing calculators that include CASs are known as “symbolic calculators” or “CAS calculators”. Examples of such CAS calculators include HP-50 (by Hewlett-Packard) [44], TI- NSpire CAS (by Texas Instruments) [52], and Casio Classpad (by Casio) [53].

2.5.3. CASs Used in Education

There are many CAAs that make use of CASs. For example, CABLE by Sangwin et al. [23] that evaluates student-provided single-step answers, uses Axiom CAS. They use the "answer tests", which are predefined CAS procedures to assess whether the student provided answer is in the expected form. In another research by the same authors where they developed the system STACK [24], the authors used Maxima CAS. Details of these two systems were given in a previous section.

In the AIM (Alice Interactive Mathematics) system [48], the Maple CAS is used. In the algebraic answer category they use the CAS to evaluate the equivalence between the student-provided answer and the teacher-provided answer.

Lan et al. [26] use Sympy CAS to identify equivalent terms in expressions. In order to identify features, the authors use Sympy to simplify the terms in the expressions so that all equivalent expressions are simplified into the same expression. Since expressions including Pythagorean identities such as $sin^2x + cos^2x$ can be simplified into “1”, it is required to do only arithmetic simplifications of such times. The authors use Sympy’s controlled simplification in such cases.

There are many studies done to validate whether using a CAS in learning can atrophy students' skills. Fifteen studies that incorporate CAS in learning have been reviewed by Heid et al. [51] to answer this question. Their review proves that exposure to a CAS does not adversely affect students' computational or procedural skills.

A research by Driver [46] looks at the impact of using a CAS in the classroom. The experiment was carried out with Grade 10 students using Casio Classpad as the CAS calculator. Ten classrooms were used for the experiment and one of the classrooms was used as the experimental group. This class was given the CAS calculator during learning in the classroom and for take-home assignments. However, the students were not allowed to use the calculator for the final exam. All students in other classes used only pen-and-paper both during learning and during the final exam. Each student in the experimental group was matched with a student in the control group. This pairing was done based on the grade they achieved in the previous year final exam, so that a student in the experimental group was paired with a student who received a grade within 2%. The results for this study showed that there was no significant positive or negative effect on student performance as a result of incorporating a CAS to the learning process.

2.6. Error Identification in Student Answers

The types of errors made by students can be broadly categorized as bugs (systematic errors) [33] and slips (unsystematic errors) [34]. Next two sections give brief descriptions of each type.

2.6.1. Bugs (Systematic Errors)

James et al. [33] describe bugs as flawed procedural knowledge. They call this error category as "bugs" referring to the same notion taken from computer programming, where a bug refers to an algorithm that contains an incorrect operation. A systematic error is produced each time the particular operation is executed. Similarly, some operations can be omitted from the procedure of the computer program that causes the program to fail. This is analogous to a student missing some procedural knowledge. It is required to identify bugs in student answers to understand what conceptual knowledge she lacks.

2.6.2. Slips (Unsystematic Errors)

Norman [34] defines a slip as an error caused by an action that was not intended. According to him, a slip can occur due to one of three reasons: from conflict among several possible actions or thoughts, from intermixing the components of a single action sequence, or from selection of an appropriate act but in some inappropriate way.

According to literature, slips should not be focused as they are not intentional. Any feedback provided on slips will be repetitions of concepts that students already know [35]. For example a student may mistakenly write 2 × 4 = 6 mixing multiplication and addition operations just because he did not read carefully.

2.6.3. Error Identification

It is quite common that students make mistakes when solving mathematics problems. Much research has been done to analyse the reasons behind the student errors in mathematics education. According to Peng and Luo [36], analysis of errors can be categorized into two broad categories depending on the perspective of analysis as follows:

- From the perspective of students - related to the nature of mathematical errors
- From the perspective of teachers - related to the ways that teachers engage in error analysis

Sarwadi and Shahrill [37] state that student errors are a result of failure to make connections with what they already know. Most of the time, the concepts that are taught in the classroom are built on top of what the student was taught in the previous years. If the student did not grasp the earlier concepts clearly, they will try to fit the new concepts according to their knowledge of the previous concepts.

A well-cited research by Brown and Burton [38] describes a system that uses procedural networks to build diagnostic models. Their model captures *a student's common* *misconceptions or faulty behavior as simple changes to a correct model*. The authors have limited their research only to arithmetic skills to focus only on the idea of diagnosis without having to worry about algebraic rules.

That research proved that students are very competent procedure followers, but they often follow "wrong procedures", leading to wrong answers. This is similar to what bugs do according to James et al.'s [33] classification. Following wrong procedures ensure that a student's wrong answers are not random. For example Brown and Burton [38] analysed addition and multiplication of two students and discovered the procedures that the students have followed. Two examples of such procedures are as follows.

E.g. Addition of two 3-digit numbers

$$\begin{array}{r} 365 \\ +574 \\ \hline 819 \end{array} \qquad \begin{array}{r} 679 \\ +794 \\ \hline 111 \end{array} \qquad \begin{array}{r} 923 \\ +481 \\ \hline 114 \end{array} \qquad \begin{array}{r} 27{,}493 \\ +\ 1{,}509 \\ \hline 28{,}991 \end{array} \qquad \begin{array}{r} 797 \\ +48{,}632 \\ \hline 48{,}119 \end{array}$$

Procedure followed by the student: He writes down the carry digit and ignores the units digit.

E.g. Multiplication of two numbers

$$\begin{array}{r} 68 \\ \times 46 \\ \hline 24 \end{array} \qquad \begin{array}{r} 734 \\ \times 37 \\ \hline 792 \end{array} \qquad \begin{array}{r} 543 \\ \times 206 \\ \hline 141 \end{array} \qquad \begin{array}{r} 758 \\ \times 296 \\ \hline 144 \end{array}$$

Procedure followed by the student: He multiplies the two digits in a column, writes down the carry digit and ignores the units digit.

According to the authors, students' work displays near perfection with respect to their own way of doing them. Authors use procedural networks to depict the diagnostic models and a portion of the procedural network for addition is shown in Figure 2.1.

For a particular procedure (e.g. addition), the student uses a set of sub-processes. These sub-processes are depicted in ovals in the procedural network (e.g. sum a column, addition facts table, and carry into the top operand). The students replace these correct sub-processes with buggy ones, which lead them to end up in wrong answers. The authors elaborate on this stating that when buggy procedures are invoked, they replicate the errors made by the student. According to the authors, there can be multiple buggy procedures that explain the reason for a wrong answer, because more than one buggy procedure can generate the same wrong answer. However, the authors only focus on simple addition and subtraction calculations (involving two terms). They do not consider error identification in complex and multi-step answers.

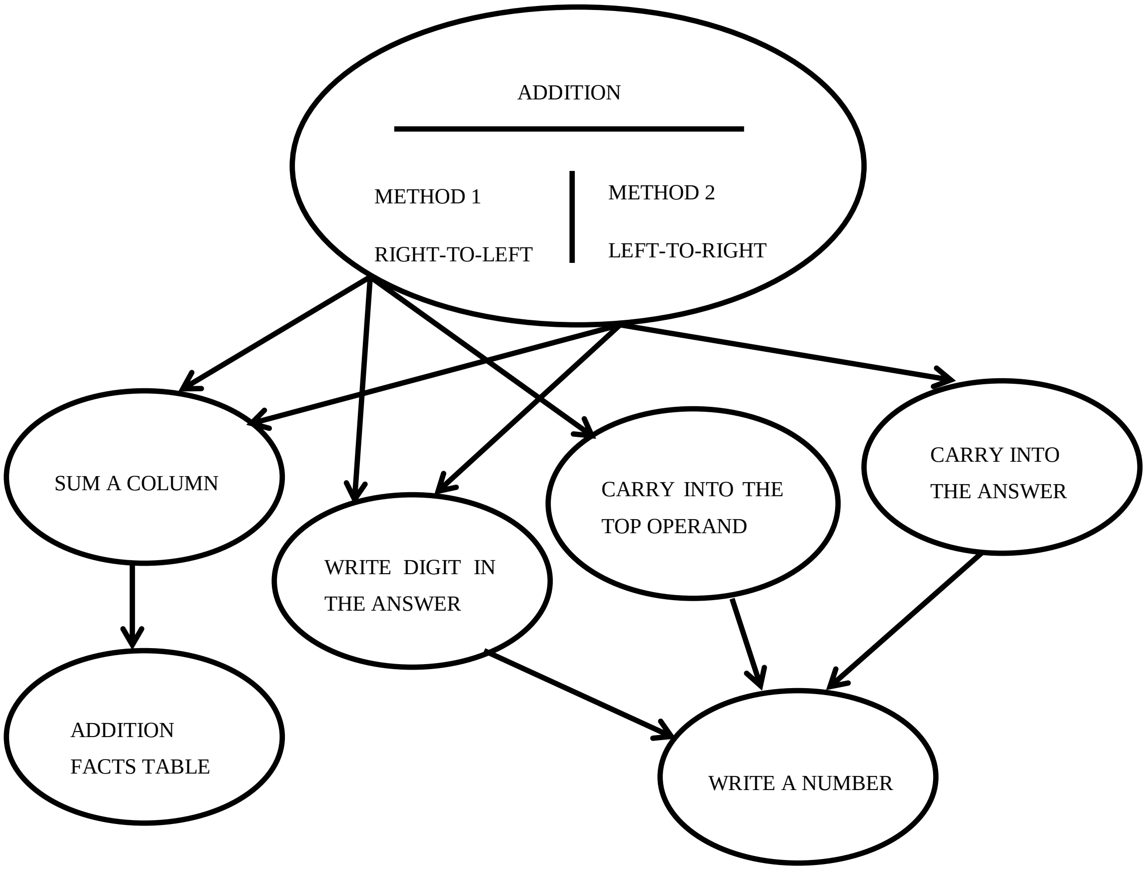

Figure 2.1 A portion of the procedural network for addition [38]

There are studies done to identify errors in various types of questions. In one such research, Dhlamini and Kibirige [39] analyse errors and misconceptions of student answers in fraction adding questions. The author used a sample of 40 Grade 9 students where they were given fraction adding questions to be done in 50 minutes. At the end of the session, the erroneous answers were grouped together based on the similarity of errors. The students were interviewed to identify the processes that led to the wrong answers. The study proved that students make common mistakes as a result of following wrong procedures. Students perceive these wrong procedures to be correct and hence make the same mistake in each question. For example, when adding two fraction terms, some students add numerators and denominators separately. These students think that is the correct procedure to follow and hence keep making the same mistake over and over again.

Makonye and Matuku [40] have identified the common mistakes that students make when solving quadratic equations in mathematics. They have done the study with a sample of 32 students from Grade 11. They focus on identifying errors and analyzing how they emerge as a result of misconceptions. They got the students to provide answers to quadratic equation solving questions and analyzed the erroneous ones. The authors carried out interviews with the students in order to identify what the students were thinking when they wrote the wrong answer. Their study proved that students have common mistakes. The interviews proved that when making mistakes, the students used their own ways of arriving at the solution. The authors produced a list of common mistakes that students make when answering quadratic equation solving question that include errors in distribution law, like and unlike terms, addition, subtraction, substitution, etc. This is a study in pedagogy and hence the authors stop at identifying the types of common mistakes in the said type of questions. They do not consider automating the process of error identification or providing feedback to students.

Gonzalez et al. [35] focus on the identification of errors made by students with Down syndrome. They took a sample of 9 students who were given 15 questions to answer. The questions were simple addition and subtraction tasks. They recorded student activity and used data mining techniques to cluster the answers that had similar errors. With this, they identified the error types that are there in the student answers. Once the error types are identified, they use LEASA (Linear Approximation and Eliminating Search Algorithm) to classify a particular erroneous answer to the most probable error based on the weight. The types of questions are simple due to the sample of students taken for the study included students with Down syndrome. The authors do not talk about complex and multi-step answers.

In the parser created by Livne et al. [22] to award partial credit to student-constructed answers, the partial grade is calculated based on the errors in the answer. The errors are identified by categorizing each element in the answer as correct, missing, unrecognized, wrong, and redundant. This is done by comparing the student-provided answer against the model answer provided by the teacher. For this parser to correctly identify this categorization, the student-constructed answer should have symbols in them. Since the system considers one line of expression, awarding partial marks to multi-step answers is not possible in their system. Also when the student's answer can have multiple approaches, it is impossible to store all variations that can come up.

According to the existing work on error identification, much interest has been from the pedagogy aspects. There are few systems that try to automate error identification, but these systems are limited to identifying errors in answers entered as a single line of expression. Automatically identifying errors in multi-step open-ended answers has not been investigated in the existing research work.

2.7. Rubrics/ Marking Schemes in Grading

When grading answers, the teacher cannot do that arbitrarily. She needs to follow a predefined procedure so that all student answers are evaluated fairly. Rubrics, also known as marking schemes, give details on how a student answer should be assessed. They make assessment quick and efficient, and they help teachers to maintain uniformity when awarding grades to various students, which can be understood by any external party such as parents. Having rubrics enable teachers at all levels of seniority and experience to have a common grading approach for all answers.

The CAAs also make use of the concept of rubrics in various ways. Following is a brief description of such systems.

When considering research that makes use of XML representation of marking scheme in evaluations, Guanyu Li et al. [47] describe MAML (Mathematics Assessment Markup Language), which is used to encode questions or test papers and their description information. Since MAML is defined in XML, it can be easily integrated with any other system. The authors have defined three groups of MAML elements: test paper markup elements (used to describe an examination paper), question markup elements (used to describe a question), and presentation markup elements. Alongside, descriptive markup elements are used to give descriptive information about the contents. Although the authors do not specifically mention about specific marking schemes, the same structure can be followed to generate marking schemes.

Ambekar [31] describes a machine learning technique used to grade MOOC assignments. He describes the rubrics definition, which is used to train his model. Here the author uses XML representations to provide sample answers to the system. Various tags are used to include different types of content, such as question and answer.

Richard [50] has described how a paper-based rubric can be converted into a machine understandable format using XML. The author uses various tags (such as RUBRIC_CRITERIA, RUBRIC_SCORE) and multiple levels (such as multiple criteria levels) to make the rubric more usable. Their model can be extended to any type of questions, with required additional tags as required.

Rubrics are important in tasks where partial credit is possible. This is because it allows specifying which parts of a particular answer will receive partial marks.

2.8. Discussion

Many of the systems that evaluate answers to mathematics questions focus on evaluating answers entered as a single line of expression. Those that evaluate multi-step answers do so by two ways: (i) by breaking up the question to multiple parts and thereby forcing the student to write single step answers to all sub-questions, and (ii) by clustering a large amount of answers and getting a teacher to grade one answer from each cluster and thereby propagating the grades to other answers. To the best of our knowledge, none of the existing systems is capable of evaluating a single student provided answer at a time, where the answer is multi-step and open-ended. Also, for most of the existing systems, adding a new question adds the overhead of adding other information (such as instructions of marking) to the system, even if the question of a same type that was previously entered.

Section 2.6 discussed about error identification in student answers. Pedagogical research has been carried out to identify the common types of errors that students make in their answers. This research work has focused on specific areas of education when identifying errors. For example, two studies exist that identify errors in fraction adding questions and equation solving questions. Only pedagogy related research has been done in the area of error-identification and no automation has been done. There are two types of errors: bugs and slips. Bugs are systematic where student follows a certain procedure when arriving at the wrong answer. These suggest that there are issues in fundamental understanding of concepts. According to literature, only these systematic errors should be further investigated as slips are quite unsystematic and occur unintentionally. A slip can even be made by an expert. Therefore identifying slips does not contribute towards improving student's learning process.

When it comes to algebra, most of the calculations require symbolic manipulations and numeric calculations. There are systems built to automate these tedious and time-consuming tasks. These systems are known as Computer Algebra Systems (CASs). Many of the CAAs that evaluate answers to mathematics questions make use of CASs and their various functionalities.

Grading answers to student answers is not arbitrary and the teachers need to establish a detailed marking scheme (also known as rubrics) prior to grading so that the grades given are justifiable. Rubrics allow teachers to have such pre-defined marking criteria.

This is not specific to any subject area, but equally important to all subjects that require assessment. A brief overview of rubrics was given in section 2.8.

3. AUTOMATIC GRADING OF MULTI-STEP ANSWERS

3.1. Overview

In the previous chapter, research on CAA was discussed with focus on evaluating answers to mathematics questions. Most of the existing CAA systems do the evaluation by comparing the student-provided answer against a teacher-provided answer. Also most of the existing systems evaluate answers entered as a single line of expression.

The grading system presented in this thesis overcomes the issues of existing CAA systems by automatically evaluating multi-step open-ended answers according to a marking scheme. The system is automated such that the teacher does not have to get involved during grading. The teacher is only required to set up the question and to decide which parts of the answer will get partial marks. This is done by looking at the concepts that are required to be present in the answer. If there are mandatory concepts, they are assigned partial marks. A marking scheme is prepared with these details. The system accepts the marking scheme in XML format. The student provided answer is graded according to the marking scheme requirements. G.C.E. O/L examination does not allow follow-through marking (if a student makes a mistake halfway and continues the rest of the answer correctly according to the wrong step, she will not get any further marks).

Considering the factors mentioned in section 2.5.1, Sympy [43] CAS has been incorporated to our system for algebraic manipulations. Sympy has a very active community of developers and new features are added to the CAS as a result. It supports a vast variety of operations including algebraic expression manipulations and matrix manipulations. In addition, Sympy can be used cross-platform.

The methodology used for answer evaluation is shown in Figure 3.1. The rest of this chapter describes the steps of the methodology in detail.

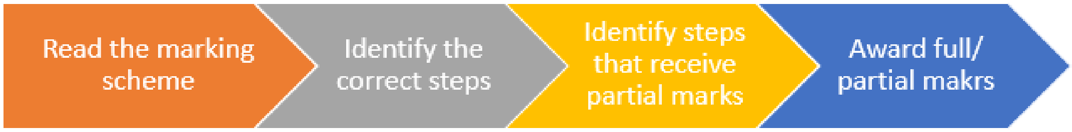

Figure 3.1 Process of awarding full/ partial credit

3.2. Types of Questions Being Evaluated

Two types of answers are evaluated.

- Answers to questions that require solving linear equations with fractions
- Answers to questions that require solving quadratic equations

The G.C.E. O/L examination contains two types of quadratic equation solving questions: taking the perfect square and by substituting to the equation. We focus on the first approach in this research as the latter is quite trivial where student only needs to memorize the equation $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$ and substitute the values for *a*, *b*, and c. Also in most of the past years, the examination either gave the freedom to students to use one of the methods or specifically asked to solve using completing the square method. These suggest that the completing the square method needs more focus.

3.3. Grading Answers

The system takes the student provided answer and evaluates it to award full/ partial credit. The student receives full marks if she demonstrates application of all required concepts in the answer and receives partial credit if she makes a mistake halfway.

For grading the answers, the teacher is not required to provide a sample answer, but required to provide a marking scheme. The interface for entering the marking criteria is out of focus for this research, but it is currently being developed under a separate but related project. Sections 3.3.1 and 3.3.2 explain how student answers are evaluated according to the marking scheme.

3.3.1. Grading Answers to Linear Equation Solving Questions with Fractions

The nature of linear equation solving questions that appear in G.C.E. O/L examination is described in section 3.3.1.1. The section also describes the characteristics of the answer and the marking criteria. How the answers are graded according to the marking scheme is described in section 3.3.1.2.

3.3.1.1. Nature of Question, Answer, and Marking Criteria

In the G.C.E. O/L examination paper in Sri Lanka, linear equation solving questions include two or more fractions. Answer to such a question requires the student to apply knowledge on various concepts taught in the classroom. The total marks for the question are divided among these concepts. Following are the concepts that receive partial marks for linear equation solving questions:

- Taking common denominator
- Taking cross multiplication
- Arriving at the final answer

According to the marking criteria of G.C.E. O/L examination, it is required to show all the expected steps in order to receive full marks. Skipping a step means that marks allocated for that concept are deducted.

An answer consists of multiple steps and students can have many different approaches when arriving at the answer. Figure 3.2 shows an example of such a question and two different approaches used by two students.

Note that both of these solution steps lead to the correct final solution, but the steps are different. Also both students have utilized their knowledge in taking common denominator and cross multiplication and hence should be awarded full marks.

Algorithm 3.1 shows the process used for evaluating answers to linear equation solving questions. The same has been explained in natural language below.

Reading student answer

At present this step is achieved by typing in the student provided answer on the paper into the system. The equation editor for the system is currently being developed under a separate but related project.

$$\frac{2x}{3x-4}-\frac{1}{x}=\frac{2}{3}\ , \qquad Solve\ for\ x$$

$$\frac{2x \times x-(3x-4)}{x(3x-4)}=\frac{2}{3}$$

$$\frac{2x^2-3x+4}{x(3x-4)}=\frac{2}{3}$$

$$3(2x^2-3x+4)=2x(3x-4)$$

$$6x^2-9x+12=6x^2-8x$$

$$x=12$$

Answer 1

$$\frac{2x}{3x-4}=\frac{2}{3}+\frac{1}{x}$$

$$\frac{2x}{3x-4}=\frac{2x+3}{3x}$$

$$2x \times 3x=(2x+3)(3x-4)$$

$$6x^2=6x^2-8x+9x-12$$

$$0=x-12$$

$$x=12$$

Answer 2

Figure 3.2 Answers by two different students for a linear equation solving question

Comparing each step with the correct answer

The solution for a linear equation solving question has only one value (e.g. the answer for the question in Figure 3.2 was 12). If the student goes step by step correctly, when each line is solved for the value of x, it should yield the same value. If the student makes a mistake in a particular step (e.g. if the student does some error in addition), the step where the student makes the mistake does not get evaluated to the correct answer. All lines in the student answer are solved for x and compared against the correct value. The first step that does not evaluate to the correct value is the step where the student makes the first mistake. This property is used to identify the set of correct steps. As mentioned earlier, since G.C.E. O/L does not support follow-through marking we stop at the step where the student makes the first mistake.

Identifying the step where Least Common Denominator (LCD) is taken

This refers to identifying a concept-evaluating step where the student's ability to take LCD correctly is evaluated. In order to evaluate whether the student can calculate LCD correctly, it is required to identify the step where the student calculates LCD. According to the G.C.E. O/L marking criteria, marks are awarded if LCD has been calculated correctly with a wrong numerator. Correctness of the step is evaluated after identifying the step where LCD has been calculated. If the LCD is correct, the designated amount of marks is awarded. Available functions from Sympy CAS are used to identify the step where LCD is taken. For example, Sympy has an in-built functionality to calculate LCD when two terms are given. This functionality was used to generate a list of all possible LCDs so that student's LCD can be compared with the list.

Identifying the step where Cross Multiplication (CM) is taken

Correctly calculating the CM also receives marks in linear equation solving questions. In order to take the CM in an equation, it is required to have only a single fraction term at each side of the equation. Then, the numerator of one side is multiplied with the denominator of the other side and vice versa. Therefore the step which is before where the student takes CM has two fraction terms, one at each side. When the student takes CM, the number of fraction terms becomes zero. These properties are used to identify the CM step. Once the step is identified, the designated amount of marks is awarded depending on the correctness of the step.

Evaluating Final Answer

This is the step where the student indicates that she has arrived at the final solution. Usually this is the last line of the steps, however there are solutions where the last line is not the step where the student denotes the final value for the variable (e.g. when student finishes the solution halfway if she does not know how to proceed). Therefore it is required to confirm that the last step corresponds to the final value. This can be achieved in a simple way - checking whether the particular step comprises of an equation where one side has the variable and other side has the value of the variable (e.g. $x = 12$). Once the step is identified, the marks are awarded depending on the correctness of the answer.

```
<conceptEvaluated, partialMark>: map
answerSteps: list
correctSteps: list
totalMarks: float
function evaluate(answerSteps):
      for step in answerSteps:
            if isCorrect:
                  Step -> correctSteps
```

Algorithm 3.1 Algorithm for evaluating linear equation solving question

3.3.2. Grading Answers to Quadratic Equation Solving Questions

As mentioned in the section 3.2, this research focuses on completing the square method out of the two quadratic equation solving approaches. Further details of this method and marking criteria are described in section 3.3.2.1. How grading is carried out according to the marking criteria is described in section 3.3.2.2.

3.3.2.1. Nature of Question, Answer, and Marking Criteria

Similar to linear equation solving questions, an answer to a quadratic equation solving question requires application of various concepts taught in the classroom. The specific concepts that are typically evaluated are listed below:

- Taking perfect square
- Taking square root correctly indicating that the unknown variable can have 2 values. (+ or -)
- Substituting the given value of square root
- Arriving at the final value(s) for the unknown variable

Similar to linear equation solving questions, students are expected to show the application of all concepts in order to receive marks in full. Moreover, students can have multiple approaches when arriving at the solution, making it impossible for an answer to be graded against a single model answer. Figure 3.3 shows an example of a quadratic equation solving question and answers from two different students. Although the number of steps and the two approaches are not the same, both students receive full marks as they have correctly applied the required concepts.

3.3.2.2. Grading Process

The process for quadratic equation solving questions is similar to that of linear equation solving questions. However, the concepts being evaluated are different. The steps of the process are described below, excluding the steps that were also present in the linear equation solving questions.

Identifying the step where Perfect Square (PS) is calculated

This is the step where the student re-arranges the expression so that it contains a perfect square of the variable. For example, a student may write $x^2 - 8x - 11 = 0$ as $(x - 4)^2 - 16 - 11 = 0$. Once the student takes the perfect square, it allows her to easily find the value for x as now she can move all numeric values to the other side and take the square root. The step where the student takes the PS can be identified by looking at each term: if at a particular step one of the terms is a PS, then that is the step where the student takes the PS. This property is used to identify the step. Then the correctness of the step is evaluated to decide whether to award marks or not.

$\boldsymbol{Solve\ x^2 + 4x - 8 = 0\ taking\ \sqrt{3} = 1.73}$

$$x^2 + 4 = 8$$

$$(x + 2)^2 - 4 = 8$$

$$(x + 2)^2 = 8 + 4$$

$$x + 2 = \pm\sqrt{12}$$

$$x = -2\ \pm 2 \times 1.73$$

$$x = 1.46\ or\ x = -5.46$$

$$x^2 + 4x + 4 - 4 - 8 = 0$$

$$(x + 2)^2 - 12 = 0$$

$$x + 2 = \pm\sqrt{12}$$

$$x + 2 = \pm\sqrt{4 \times 3}$$

$$x = -2 \pm 2\sqrt{3}$$

$$x = -2\ \pm 2 \times 1.73$$

$$x = -2 + 3.46\ or\ x = \ -2 - 3.46$$

$$x = 1.46\ or\ x = -5.46$$

Figure 3.3 Two different approaches to reach a single answer in quadratic equation solving

Identifying the step where square root is taken

The step where the student takes the square root can be identified by evaluating the coefficient of the x^2 term. The step in which the coefficient first becomes zero is the step where the student takes the square root. Since answer to a quadratic equation solving question can have two values, this step is processed further to identify if the student has taken both plus and minus values.

Identifying if the student has correctly substituted the given square root value

In G.C.E. O/L quadratic equation solving questions, as the student goes on solving the equations, she comes to a point where she needs the value of a square root, which is not a perfect square. For example in the sample question explained earlier, the student will then write $(x - 4)^2 = 27$ followed by steps $(x - 4)\ =\ \pm\sqrt{27}$ and $x - 4 = \pm 3\sqrt{3}$. At this point, the student requires the value of $\sqrt{3}$ in order to proceed. In the O/L examination paper, this square root value is given as a part of the question. The student is expected to substitute the given value at the correct time. Simple string matching can be used to identify the step where the student does this.

Identifying if the values for x (unknown variable) are correct

The final answers indicated as values for x are taken and are compared against the correct answers calculated by Sympy. In the case of only positive values are expected as solutions, the teacher can specify that in the marking scheme and the system carries out evaluations accordingly.

Similar to the scenario in linear equation solving questions, functionalities of Sympy are used to identify various steps.

3.3.3. Marking Scheme for Answer Grading

A marking scheme (rubric) was created to evaluate the linear equation solving questions and quadratic equation solving questions. The marks were assigned to the concepts that are usually evaluated in O/L examination as described in the previous sections. These concepts were taken from the marking schemes of the past papers. These were also confirmed with two teachers who are engaged in O/L paper marking. Referring to previous work, the marking scheme was defined as an XML schema and was read into the system.

The schema has concepts to be evaluated and the amount of marks assigned to each concept. The system automatically generates the answers to the questions; therefore the marking scheme does not require the inclusion of answers. The main tags used are described in Table 3.1.

The basic tags that are used in linear and quadratic marking schemes are similar and there exist only minor differences. The main difference in the two marking schemes is the concepts that are being evaluated. Apart from that, quadratic equation solving questions need two more attributes: “positiveOnly” and “decimalPlaces”. In some questions, it is indicated that x (or the variable) can take positive values only and the first attribute caters this purpose. The second attribute is used to indicate the number of decimal places expected in the answer. In the past O/L examination papers, this is given in the question (See Appendices A and B for sample marking schemes in XML).

3.3.4. Awarding Partial Marks

Once the partial marks for application of different concepts are calculated, they sum up to the total marks that should be awarded to the student.

Since total marks are broken down into parts such that various concepts receive portions of total marks, a student can receive partial marks. For example, if the student applies one concept correctly and makes a mistake afterwards, she will be awarded the marks designated for that particular concept only.

Table 3.1 XML schema tags of the marking scheme

Tag	Description
type	Type of question (either “Linear” or “Quadratic”)
question	Includes details about the question, such as id and total marks. The details are given as attributes in the tag.
data	Items of the answer that are expected. These include the amount of partial marks.
concept	The concept to be evaluated (e.g. taking common denominator)

3.4. Discussion

CAA systems have been developed to assess student answers, but there exists no system that evaluates multi-step and open-ended answers according to a marking scheme with zero teacher involvement. As the first step to solve the above issue, a system has been developed that assesses answers to two types of algebra questions: linear equations with fractions and quadratic equation solving questions that use completing the square method.

According to the best of our knowledge, assessing a single student provided answer entered as a set of steps and awarding full/partial marks is not possible in the existing systems. Therefore we claim to have introduced this novel technique whereby partial marks can be awarded to an open-response multi-step answer. The technique has been developed for questions in linear equations with fractions and quadratic equations. The same technique can be extended to any type of simple algebra question that requires application of knowledge in concepts in order to grade according to a marking scheme. The steps where the concepts have been applied can be identified using a CAS.

4. ERROR IDENTIFICATION IN STUDENT ANSWERS

4.1. Overview

Students make mistakes while they attempt exam questions. Some of these errors occur as a result of students having issues in fundamental understanding of concepts taught in the classroom. Identifying errors in student answers is important to both teachers and students: for teachers it allows the identification of areas that students struggle and for students it allows them to identify areas that they need to study again.

According to literature, existing error identification systems look only at identifying errors in answers entered as a single line of expression. Most of the existing research on error identification of multi-step answers is limited to pedagogy and do not automate the error identification process. One study that claims to be the first to evaluate multi-step answers requires a large amount of data to identify errors as it uses clustering techniques to identify groups of answers which are similar to each other [26].

We have developed a technique to identify student errors in open-response and multi-step answers. The system has been developed so that it is capable of identifying errors in answers to linear equation solving questions with fractions and quadratic equation solving questions that use completing the square method. This technique is shown in Figure 4.1 and the rest of the chapter elaborates the same in detail.

Figure 4.1 Process for identifying the error in student’s answer

4.2. Types of Errors

As mentioned in section 2.6, student mistakes in answers to mathematics questions can be broadly categorized into bugs (systematic errors) and slips (unsystematic errors). According to literature, only bugs are of interest to both teachers and students. This is because bugs represent issues in fundamental understanding of concepts while slips represent unintentional mistakes.

A bug in a computer program indicates that there is an incorrect operation in an algorithm [49]. A bug in an answer to a mathematics question is analogous to this, where a student follows an incorrect procedure when arriving at the solution. The student follows the same buggy procedure for all questions in that type as she thinks it is the correct way of arriving at the solution. Therefore if a set of questions is given to a student, she will make the same mistake in all answers.

There is existing research work that identifies errors in answers to fraction addition and quadratic equation solving questions [39, 40]. These are directly related to our study due to the two types of questions that we focus on. Below is a list of systematic errors that they have identified, which are related to our study.

- Mistakes in finding LCD - adding numerator and denominator separately

  e.g. $\frac{2}{x+1} + \frac{1}{x+3} = \frac{3}{2x+4}$
- Applying the distribution law incorrectly

  e.g. $3(x-3) = 3x+9$
- Mixing like and unlike terms

  e.g. $2+3x = 5$
- Mistakes in addition, subtraction, and multiplication of integers

  e.g. $10 - 1 = 11$
- Mistakes in additive inverse

  e.g. $x^2 - 4x + 12 = (x-2)^2 + 4 + 12$

In the existing research, these mistakes have been identified for pedagogical purposes. There is no automation of the error identification.

After exploring the errors identified in the literature, a set of 42 sample student answers was analysed to identify if there are more systematic errors made by students. The following types of errors were discovered in these two question types in addition to the errors found in literature.

For Linear Equation Solving Questions with Fractions

- Adding terms in the numerator when finding LCD

  e.g. $\frac{2}{x+1} + \frac{1}{x+3} = \frac{3}{(x+1)(x+3)}$

- Cancelling terms separated by + or -

  e.g. $\frac{x+3}{(x+1)+(x+3)} = \frac{1}{x+1}$

For Quadratic Equation Solving Questions

- Incorrectly calculating perfect square

  e.g. $x^2 - 4x = (x+2)^2 - 4$

- Not calculating two values for *x*

  e.g. $(x-2)^2 = 24$

  $x - 2 = \sqrt{24}$

- Not substituting the given value for square root correctly

  e.g. $\sqrt{2} \times 3 = 2.4 \times 3$; given $\sqrt{2} = 1.41$

Name codes were assigned to the types of errors that are identified from the system. Table 4.1 shows a list of errors with their assigned feedback text.

Once the common systematic errors were identified, it was required to identify the error in a given answer. As suggested by literature in education, understanding unsystematic errors was not useful from both the student’s and the teacher’s perspectives.

Table 4.1: Error codes and associated feedback

Error Code	Output
`FB_LCD_ADD_TERMS_ERR`	Numerator terms and denominator terms have been added separately when finding LCD
`FB_DIST_LAW_ERR`	Distribution law has been applied incorrectly when removing parentheses.
`FB_MIXED_NUMER_AND_ALGEB`	Algebraic terms and numeric terms have been added/ subtracted together
`FB_ADD_ERR`	A mistake has been done in addition
`FB_SUBT_ERR`	A mistake has been done in subtraction
`FB_MUL_DIV_ERR`	A mistake has been done in multiplication or division
`FB_ADDITIVE_INVRSE_ERR`	Has not subtracted the additional value correctly
`FB_LCD_ADD_TERMS_NUMER_ERR`	Terms of numerators have been added when finding LCD
`FB_CANCEL_TERMS_SEP_PLUS_MINUS`	Terms separated by + or - have been cancelled
`FB_PERF_SQ_ERR`	The perfect square has not been calculated correctly
`FB_SQRT_ERR`	Has not indicated that x can have two values
`FB_SUBST_ERR`	The given value for square root is not substituted correctly

4.3. Error Identification in Answers

4.3.1. Replicating Errors

After coming up with the list of systematic errors that will be handled by the system, a way to generate them automatically was designed. The idea was to apply one of the bugs from Table 4.1 to a given step to predict the student’s next step. In literature, this method has been used to educate teachers regarding the errors that students can make [38].

In our system, various functions were written to replicate the errors. These functions act as buggy procedures that students follow when writing answers. Each function applies one of the identified bugs to the given step and returns a buggy next step. In the next section, how this helps in error identification is discussed.

4.3.2. Identifying the Error Made by the Student

As mentioned earlier, identification of errors in the said type of answers has been limited to pedagogy, where no automation has been discussed. We referred to those studies and came out with the model expressed in the Algorithm 4.1.

```
buggyProcedures : list
possibleBugs : list
function reportError (firstWrong, lastCorrect):
     for (procedure in buggyProcedures):
          apply procedure to lastCorrect - > output
          if output is firstWrong:
               bug of procedure -> possibleBugs
     end
```

Algorithm 4.1 Algorithm to identify errors in student answers

The algorithm is explained in natural language below.

Step 1

Evaluate the student provided answer using the evaluation technique discussed in section 3.3. If there is a mistake in the answer, continue to **Step 2**, if not abort.

Step 2

Identify the step where the mistake has been made.

Step 3

Take the step where the first mistake has been made and the step before that. (The last correct step)

Step 4

Feed the last correct step to each of the buggy procedures.

Step 5

Compare buggy outputs with the student’s wrong step. If a buggy output matches with the student’s answer, it suggests that the student has followed that particular buggy procedure.

Step 6

Indicate the bug that the student has made (This can be a list as it is possible to get the same buggy step from multiple procedures).

Following this method, the error in student’s answer can be identified. Once the list of possible bugs is identified, it can be presented to the student in a way that lets the student to understand where she needs to improve. At present, the list of possible bugs is displayed, together with the step where the error is present. When giving feedback to students, it is required to consider multiple aspects, such as the levels of feedback. Generating proper feedback considering these aspects is out of the scope of this research.

Therefore the feedback given by the system is basic and is limited to indicating the possible errors.

4.4. Discussion

In manual grading, when the teacher goes through the student answer, she can see the error the student has made. But in a CAA system, a mechanism is required to automatically identify the errors that students make. Much research has been done in pedagogy to identify the types of errors that students make and what cause them. However, automation of error identification of multi-step and open-ended answers has not been discussed.

To solve the above issue, a technique that replicates systematic errors has been developed. The results are presented in the next chapter.

5. EVALUATION AND DEMONSTRATION

5.1. Overview

This chapter evaluates the work described in the thesis so far. How the objectives mentioned in section 1.3 have been achieved is described in this chapter with evidence on contributions made by the thesis as mentioned in section 1.4.

In this study three modules were presented for i) evaluating answers to linear equation solving questions with fractions ii) evaluating quadratic equation solving questions iii) error identification in student answers. For demonstration and evaluation of the contributions, answer scripts were collected from various sources. These data sets are described in section 5.2.

Two teachers were consulted to evaluate the system. Section 5.3 gives evaluation details for the answer grading modules for linear equation solving with fractions and quadratic equations. The evaluation results for the error identification module are described in section 5.4.

5.2. Gathering Data for the Study

In order to grade, the student-provided answer is first taken into the system. As mentioned in chapter 1, two types of answers were required.

- Answers to linear equation solving questions with fractions
- Answers to quadratic equation solving questions

As mentioned earlier, the G.C.E. O/L examination supports quadratic equation solving questions of two types: taking the perfect square and substituting to the equation. We focus only on the first approach in this research.

For this research, sample answer scripts were collected from students, which were written on paper. The student answers were then typed on a machine in the exact way written by the students on the paper. Six sets of answers were used for the study. For some of the samples a model paper was given that was prepared from past exam papers. This paper consisted of 5 questions from each of the two types: linear and quadratic.

(i) 40 answer scripts from Grade 11 year-end examination of a government school in Sri Lanka (Government school I)

This was from the year-end examination paper given to school students. The paper included a question that had sub-questions with linear and quadratic equation solving. However, the question was optional, therefore the number of students who answered the question were 34. In this paper, the method to use for quadratic equation solving was not restricted to completing the square method. 22 out of the 34 students who elected the question had used the completing the square method.

(ii) 68 answer scripts from Grade 10 year-end examination of a government school (Government school II)

This was from the year-end examination paper given to students at a government school. The linear equation solving question was mandatory and 68 answers for that question were gathered.

(iii) 42 answer scripts from a government school for a model paper (Government school III)

This class had 42 students who answered the sample paper mentioned above.

(iv) 23 answer scripts from another government school for the model paper (Government school IV)

The same model paper was given to Grade 10 and 11 students from a government school. There were 9 students from Grade 10 and 14 students from Grade 11. Students are taught quadratic equation solving only in Grade 11. Therefore 115 answers for linear equation solving questions and 70 answers for quadratic equation solving questions were gathered from this set of students.

(v) 41 answer scripts from a tuition class for another model paper

This model paper was given to a set of 41 students in a tuition class. This paper had 3 quadratic equation solving questions that required students to use the completing the square method.

The students who were given a model paper were made aware of the reason behind getting their answers. The samples in (iii), (iv) and (v) were informed that the answers they submit will be used to evaluate the accuracy of a system that will help future students. They were also informed that the answers they write do not affect their final exam grades. These two factors motivated the students to work on their own without copying from fellow students.

(vi) 16 answer scripts from university students for the model paper

A set of 16 undergraduate students were given the first model paper explained earlier. They were instructed to show solution steps clearly, and to make random errors in their answers in order to assess the error identification module. This set of answers was used only during the initial development of the system and not for testing system accuracy to obtain final results.

Out of the samples, (i), (iii), and (v) were used as the training set while (ii), (iv), and (vi) were used as the test set. All the students answered the questions on the paper.

Table 5.1 summarizes details of the datasets gathered.

5.3. Evaluation of the Grading Modules

The answer grading has two parts: one for linear equation solving questions with fractions and the other for quadratic equation solving questions. Two marking schemes were defined to evaluate the concepts that were mentioned in previous chapters. The system evaluated the answers according to the marking schemes and awarded full/ partial grade to the student.

The answers were manually evaluated according to the marking schemes, to be compared with the grades given by the system.

Table 5.1 Details of the gathered datasets

Data Set	Source	No. of Answers	Purpose
(i)	Government school I	Linear - 34 Quadratic - 22	Training
(ii)	Government school II	Linear - 68	Testing
(iii)	Government school III	Linear - 210 Quadratic - 210	Training
(iv)	Government school IV	Linear - 115 Quadratic - 70	Testing
(v)	Undergraduates	Linear - 80 Quadratic - 80	Training
(vi)	Tuition Class	Quadratic - 123	Testing

5.3.1. Evaluating Linear Equation Solving Questions with Fractions

According to the sample marking scheme, marks were awarded as follows:

- Correctly calculating the common denominator of two fraction terms - 1 mark
- Correctly calculating the cross multiplication - 1 mark
- Arriving at the final value for x (unknown) - 1 mark

According to the O/L curriculum, students were expected to do the calculations accurately in order to score the respective amount of marks. If the student made a mistake at an intermediate step, no marks were given from that point to conform to the O/L marking scheme where follow-through marks are not allowed. The evaluation results for the module that grades answers are presented in Table 5.2.

Altogether there were 183 answers (68 + 115), that included 7 different questions. The answers are different to each other as they had different number of steps and approaches.

Table 5.2 Evaluation Results for Linear Equation Solving Questions

Data Set	Number of answers	Number of answers correctly evaluated	Accuracy
(ii)	68	68	100%
(iv)	115	115	100%
Total	183	183	100%

The system evaluated the answers and awarded marks to the concepts with 100% accuracy. This proved that the system is capable of evaluating answers that have different approaches, substituting a human grader.

5.3.2. Evaluating Quadratic Equation Solving Questions

Similar to evaluating linear equation solving questions, a marking scheme was defined for quadratic equation solving questions. The concepts that receive marks in the marking scheme are as follows:

- Correctly calculating the Perfect Square – 1 mark
- Indicating that taking the square root of x^2 means x can have two values – 1 mark
- Substituting the given square root value in the answer – 1 mark
- Arriving at the final value/s for x – 2 marks

Similar to the linear equation solving questions, follow-through marks are not allowed. Table 5.3 shows results of the quadratic equation solving module.

Altogether there were 193 answers and the system evaluated them with 100% accuracy. Similar to answers to linear equation solving questions, these were multi-step answers with different approaches.

Table 5.3 Evaluation Results for Quadratic Equation Solving Questions

Data Set	**Number of answers**	**Number of answers correctly evaluated**	**Accuracy**
(v)	70	100%	100%
(vi)	123	100%	100%
Total	193	100%	100%

5.4. Evaluation of the Error Identification Module

Once the marks are awarded to the answers, the next step is to identify errors in partially correct answers. The answers that did not receive full marks were parsed through the error identification module discussed in Chapter 4. As discussed in section 4.2, a list of errors was defined carefully, which are the common mistakes that students make. Slips were not considered as they do not add any value to the student's learning process. The system generated feedback to erroneous answers based on the list of errors.

Two teachers who have experience in O/L teaching and marking were consulted to get advice on error identification. They were separately given the partially correct answers and were asked to give feedback to all of them. The feedback given by the two teachers were compared against the feedback generated by the system. When the feedback by one teacher matched the feedback of the system, while the other teacher gave a general feedback, the system was considered to have generated the correct feedback. For example, there were situations where one teacher and the system indicated that "error in calculating the numerator" for errors in calculating the common denominator, while the other teacher simply mentioned "there is some simplification error".

When teachers gave different specific feedback and the system gave a feedback different from that of teachers, a third teacher was consulted to get advice on the correct feedback.

In this case, the exact wording provided by the teachers was not considered since the wording can be different from one person to the other. For example, what the system identified as “errors in applying the distribution law correctly” was interpreted as “errors in removing parentheses” or “errors in simplifying the brackets” by the teacher.

5.4.1. Error Identification in Erroneous Answers for Linear Equation Solving Questions with Fractions

Table 5.4 shows evaluation details of the module according to the criteria mentioned above.

Table 5.4 Evaluation Results for Error Identification in Linear Equation Solving Questions

Data Set	Number of answers	Number of partially correct answers	Number of partially correct answers that received correct feedback	Accuracy
(iv)	68	51	43	84.31%
(v)	115	46	40	86.96%
Total	183	97	83	85.57%

The evaluation resulted in some interesting observations as listed below:

- The teachers took a considerable amount of time to grade all the answers and to give feedback (around 3 hours by each teacher), while the system graded all answers in less than 5 minutes.
- In some occasions, the teachers evaluated some answers as correct, while they were incorrect. This is because some answers appeared to be correct at the first glance and required careful attention to see the mistake. Our system correctly identified all the wrong answers.

  An example of such an answer is shown in Figure 5.1. In this answer, the student has made a mistake in the last step and wrote down $x = 7$ instead of $x = 9$. But the teacher evaluated the answer to be correct and awarded marks in full.

- For certain errors, the teachers gave a general error description most of the time. For example, in certain occasions, the teachers mentioned that *“the student has made a mistake in calculating the numerator”*, while the system was more specific indicating the exact error: *“there is an error in addition”.*

$$\frac{1}{x-1}-\frac{1}{2}=\frac{3}{1-x}$$

$$\frac{1}{x-1}-\frac{3}{1-x}=\frac{1}{2}$$

$$\frac{1}{x-1}+\frac{3}{x-1}=\frac{1}{2}$$

$$\frac{4}{x-1}=\frac{1}{2}$$

$$\frac{1}{x-1}=\frac{1}{8}$$

$$x-1=8$$

$$x=7$$

Figure 5.1 An erroneous answer marked as correct by the teacher

- For the same mistake, the same teacher gave different feedback for different answers. The wordings used by the teacher were different in different occasions. In contrast, our system was capable of giving the same feedback for all answers, preserving consistency.

The reduced accuracy was due to various reasons. The main reason is the inability of the system to visualize things as a human would do.

For example there was a question as follows, which was given to the students.

$$\frac{1}{x-1}-\frac{1}{2}=\frac{3}{1-x}$$

It is easily understood by the humans that this sum can be solved as follows.

- taking the term in the right hand side to the left hand side
- changing the sign of denominator of the transferred term (so that the denominators of both terms in the left become $x - 1$)
- taking the cross multiplication and simplifying

Although this is quite trivial for a human to understand, it is not easily understood by a computer system. Therefore the human graders can associate the errors students make to the question type like the one mentioned above. The solution to this issue is to incorporate errors resulted from this sum to the system so that it can handle sums that are similar. This is a possible addition to the next version of the system.

Also there were situations where students had combined multiple steps when moving from one step to the next. In such situations, when the student made a mistake, the system was not capable of identifying the error made by the student. Figure 5.2 shows an example to this issue where the student has done two operations: taking the right hand side to the left hand side and interchanging the terms of the denominator.

$$\frac{1}{x-1} - \frac{1}{2} = \frac{3}{1-x}$$

$$\frac{1}{x-1} - \frac{3}{x-1} = \frac{1}{2}$$

$$contd.$$

Figure 5.2 First steps of a sample answer

5.4.2. Error Identification in Partially-Correct Answers for Quadratic Equation Solving Questions

Table 5.5 shows evaluation results of the module according to the criteria mentioned before.

Table 5.5 Evaluation Results for Error Identification in Quadratic Equation Solving Questions

Data Set	**Number of answers**	**Number of partially correct answers**	**Number of partially correct answers that received correct feedback**	**Accuracy**
(iv)	70	34	28	82.35%
(vi)	123	54	47	87.04%
Total	193	88	75	85.23%

The observations noted in the evaluation of linear equation solving questions were present in quadratic equation solving questions as well. The accuracy lies around 85% for both types of questions.

5.4.2.1. Comments by Teachers

Some comments made by the two teachers regarding the system are worth mentioning. One teacher, who has acted as the panel chief of O/L examination grading for more than five years mentioned that *"different teachers carry out marking in different ways. In the panels I headed, there were 14 teachers. There come doubtful situations where you decide whether to give marks or not. Some teachers give marks while some do not, which becomes disadvantageous to some students. Having a computer to do the marking will eliminate such issues".*

The second teacher mentioned *"Marking student answer scripts take a lot of time. It will be great to have some system that does it for you!"*

However, having a computer system to do the marking may raise certain concerns. For example, there was an answer where the student had written down the answer without showing the steps. According to the marking scheme, this resulted in the student receiving marks designated for the final answer. One teacher said that *"Well, sometimes we give the marks to the final answer that does not have steps, and sometimes we do not. It depends on the student's skills. We can look at the other questions that the student attempted to see if she is smart enough to do the calculations in mind and come out with the final answer at once."* Such a situation cannot be accommodated in our system as currently the system does not depend on any other factor when grading a student answer.

In addition to this, one teacher mentioned how they look at other scribble work of the student when they need to give marks to the student. *"We instruct the students to attach the scribbling papers. We sometimes look at them when the student is in the border marks for the next grade."* When answering on the computers, the students would probably do their scribbling work on pen and paper. Therefore it is not possible to look at the student's scribbling work. On the other hand, some teachers may look at the scribble work while some do not. When a computer is used to do the evaluations, it preserves consistency as the computer acts the same in all occasions.

6. DISCUSSION

This chapter discusses the contributions made by the work in this thesis, together with possible improvements. The accuracy and usability of the system are also discussed in the chapter.

6.1. Contributions

In G.C.E. O/L examination of Sri Lanka, pass rates for mathematics have been low in the past years. For students, working on past paper and model paper questions is important when preparing for an examination. This is effective only if students get their work evaluated with feedback on mistakes. This research has been able to develop a mechanism to evaluate answers to two types of equation solving questions: linear equations with fractions and quadratic equation solving questions. As described in previous chapters, the answers to these questions are multi-step and open-ended in nature.

The contributions of this thesis are as follows.

- Evaluating answers to linear equation solving questions with fractions
- Evaluating answers to quadratic equation solving questions
- Locating errors in student answers and giving feedback

Each of the above contributions is described below.

Evaluating answers to linear equation solving questions with fractions

The thesis described an approach to evaluate answers to linear equation solving questions with fractions, according to the G.C.E. O/L marking scheme of Sri Lanka. Answers to questions of this type are open-ended and multi-step. The objective of these question types is to evaluate if the student correctly applied the knowledge of certain concepts taught in the classroom. The O/L marking scheme awards marks for the concepts that are being evaluated. The system is capable of reading a marking scheme that is defined in an XML schema and award full/ partial marks according to that. The teacher involvement is required only to decide on the marking scheme and to set the question. Once a marking scheme is defined, multiple questions of the same type can be evaluated according to the same marking scheme. The system evaluates answers that have a varying number of steps and approaches. Section 3.3.1 explained implementation details of this module. The module reported 100% accuracy, grading all concepts according to the marking scheme.

Evaluating answers to quadratic equation solving questions

Similar to answers for linear equation solving questions, answers to quadratic equation solving questions are also multi-step and open-ended, making it impossible to evaluate against a single teacher-provided answer. Our system is capable of evaluating these answers according to a marking scheme to award full/partial credit. How the system handles this was explained in detail in section 3.3.2. Teacher involvement is required similar to that of linear equation solving questions. For this type also, the system evaluates various concepts taught in the classroom. This module also reported 100% accuracy.

Locating errors in student answers and giving feedback

Students make mistakes in their answers, which result in partial marks. Most of the time, students make these mistakes intentionally, which means that they assume their approach was correct. Therefore, a teacher is expected to tell the student where she made the mistake so that it can be corrected at the next time. Our system identifies the step in which the student makes the mistake in the two modules mentioned previously, and provides feedback. Section 4.3 explained the implementation details. The accuracies reported for error identification in linear and quadratic equations were 85.57 and 85.23, respectively. The reduced accuracies were mainly due to the difficulties encountered by the system in visualizing student behavior as a human grader would do.

6.2. Usability

The system is implemented in Python script and makes use of Sympy CAS for grading and error identification modules. It can run on any machine that has Python and Sympy installed. The system generates a human-readable feedback that includes the amount of marks awarded and the type of error made in partially correct answers. Currently the system has a basic web interface that requires a strict input format, which is the Python format. As mentioned in section 3.3.1.1, the equation editor for the system is currently being developed under a separate but related project.

6.3. Accuracy

As explained in chapter 5, the system is capable of grading student answers to award full/ partial marks with 100% accuracy. It evaluates various approaches to the same question with different numbers of steps. The system can replace the teacher for the grading process. Error identification also produces promising results, being the first system to automatically identify errors in complex algebraic answers.

7. CONCLUSION AND FUTURE WORK

There are many existing systems that evaluate student work. However, most of these systems are either limited to evaluating MCQ type answers or to evaluating answers that can be compared against a single teacher-provided answer or a set of possible answers.

There are few systems that evaluate multi-step mathematics answers, but they rely on one of the two approaches: 1. clustering a large number of answers and getting the teacher to manually award marks to one member of the cluster to propagate the marks to the other cluster members, 2. breaking down the question into multiple sub-questions so that the student is forced to provide single step answers to all the sub-questions. These two approaches create two problems: the need to have a large number of answers and losing the purpose of allowing multi-step answers, respectively.

The system developed by us solves these two problems by allowing the student to provide a multi-step answer and giving the grade as soon as the student submits the answer without having to wait for a large number of students to submit the answers. According to the best of our knowledge, this system is the first to grade multi-step open-ended answers according to a marking scheme to award full/ partial credit. The other most important feature is the requirement of zero teacher-involvement during answer grading.

There is research work done in the area of error identification of multi-step answers, but this research is limited to pedagogy, where no automation has been considered. Our system is capable of automatically identifying the type of error made by the student in partially correct answers. The system does this by replicating errors that are common in these types of questions, and comparing the results with the wrong step entered by the student.

As future work, a performance improvement of the error identification module is required. As mentioned in section 5.4.1, the system should be supported with some intelligence in order to have the required thinking ability. Creating an intelligent agent that is capable of integrating multiple buggy procedures is a possible approach for performance improvement. Also a mechanism can be developed to extract the similar student answers that do not receive a specific feedback. Then we can consult a teacher and get their feedback on what the student has done. That way we can add more buggy procedures to the system.

Currently the G.C.E. O/L marking scheme is strictly followed for grading the student answers. As future work, a practice-mode can be enabled that allows follow-through marking. The student assessment results can be used in all questions to create a student model that gives information of the areas that the student needs to pay more attention.

Currently when the student makes a mistake in the answer, the system identifies the error. This should be presented to the student in the form of feedback. The simplest way to do this is to tell the student that she made a mistake and then to give the correct answer. But the most effective way is to first tell the student the type of the mistake that she made, thus allowing the student to figure out her own mistake. One possible approach will be having multiple feedback levels as follows for partially correct answers:

Level 1: Tell the student that there is a mistake. Ask the student to find the mistake and correct on her own.

Level 2: If the student fails to identify the error, show the step where the mistake has been made. Ask the student to correct on her own.

Level 3: If the student still fails to identify the error, give a sample correct answer to the student. Direct the student to similar questions which will test the area the student lacks knowledge and/ or provide the student with study materials on the particular topic.

REFERENCES

[1] Bennett, R. E., & Bejar, I. I. (1998). Validity and automad scoring: It's not only the scoring. *Educational Measurement: Issues and Practice*, *17*(4), 9-17.

[2] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

[3] Zipitria, I., Elorriaga, J. A., Arruarte, A., & de Ilarraza, A. D. (2004). From human to automatic summary evaluation. In *7th International Conference on Intelligent Tutoring Systems* (pp. 432-442). Springer Berlin Heidelberg.

[4] Wade-Stein, D., & Kintsch, E. (2004). Summary Street: Interactive computer support for writing. *Cognition and instruction*, *22*(3), 333-362.

[5] Pérez, D., Alfonseca, E., & Rodríguez, P. (2004). Application of the BLEU Method for Evaluating Free-text Answers in an E-learning Environment. In *Language Resources and Evaluation Conference*. 1351-1354

[6] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. 311-318. Association for Computational Linguistics.

[7] Mohler, M., & Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* . 567-575. Association for Computational Linguistics.

[8] Pado, U., & Kiefer, C. (2015). Short answer grading: When sorting helps and when it doesn’t. In *Proceedings of the 4th workshop on NLP for Computer Assisted Language Learning at NODALIDA 2015* 114, 42-50. Linköping University Electronic Press.

[9] Pulman, S. G., & Sukkarieh, J. Z. (2005). Automatic short answer marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP*. 9-16. Association for Computational Linguistics.

[10] Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer science education*, *15*(2), 83-102.

[11] Reek, K. A. (1989). The TRY system-or-how to avoid testing student programs. In *ACM SIGCSE Bulletin*. 21(1), 112-116. ACM.

[12] Higgins, C., Symeonidis, P., & Tsintsifas, A. (2002). Diagram-based CBA using DATsys and CourseMaster. In *Proceedings. International Conference on Computers in Education*. 167-172. IEEE.

[13] Jackson, D., & Usher, M. (1997). Grading student programs using ASSYST. In *ACM SIGCSE Bulletin*. 29(1), 335-339. ACM.

[14] Cheang, B., Kurnia, A., Lim, A., & Oon, W. C. (2003). On automated grading of programming assignments in an academic institution. *Computers & Education*, *41*(2), 121-131.

[15] Ellsworth, C. C., Fenwick Jr, J. B., & Kurtz, B. L. (2004). The quiver system. In *ACM SIGCSE Bulletin*. 36(1), 205-209. ACM.

[16] Bettini, L., Crescenzi, P., Innocenti, G., & Loreti, M. (2004). An environment for self-assessing Java programming skills in first programming courses. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*. 161-165. IEEE.

[17] Arnow, D., & Barshay, O. (1999). WebToTeach: an interactive focused programming exercise system. In *29th Frontiers in Education Conference.* 1, 12A9-39. IEEE.

[18] Truong, N., Bancroft, P., & Roe, P. (2003, February). A web based environment for learning to program. In *Proceedings of the 26th Australasian computer science conference*, 16, 255-264. Australian Computer Society, Inc.

[19] Annual Performance Report(2013). (n.d.). Retrieved May 20, 2015, from http://www.moe.gov.lk/english/images/publications/Annual_performance_Report2013/Annual_performance_report_e.pdf

[20] Hansen, H., & Ruuska, M. (2003). Assessing time-efficiency in a course on data structures and algorithms. In *Proceedings of the Third Finnish/Baltic Sea Conference on Computer Science Education*. 93-100.

[21] Chen, P. M. (2004). An automated feedback system for computer organization projects. *IEEE Transactions on Education*, *47*(2), 232-240.

[22] Livne, N. L., Livne, O. E., & Wight, C. A. (2007). Can automated scoring surpass hand grading of students' constructed responses and error patterns in mathematics. *MERLOT Journal of Online Learning and Teaching*, *3*(3), 295-306.

[23] Sangwin, C. J., & Naismith, L. (2008). Implementing Computer Algebra Enabled Questions for the Assessment and Learning of Mathematics. *International Journal for Technology in Mathematics Education*, *15*(1).

[24] Sangwin, C. J., & Grove, M. (2006). STACK: addressing the needs of the neglected learners. In *Proceedings of the 1st Web Advanced Learning Conference and Exhibition* (pp. 81-96).

[25] Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system in *11th World Conference on Educational Media and Technology*.

[26] Lan, A. S., Vats, D., Waters, A. E., & Baraniuk, R. G. (2015). Mathematical language processing: Automatic grading and feedback for open response mathematical

questions. In *Proceedings of the Second ACM Conference on Learning@ Scale*. 167-176. ACM.

[27] Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science (Washington), 228*(4698), 456-462.

[28] Walonoski, J. A., & Heffernan, N. T. (2006). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*. 382-391. Springer Berlin Heidelberg.

[29] Koedinger, K. R., Brunskill, E., Baker, R. S., McLaughlin, E. A., & Stamper, J. (2013). New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine, 34*(3), 27-41.

[30] Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED), 12*, 385-407.

[31] Ambekar, D. (2015). *Evaluation of essays using incremental training for Maximizing Human-Machine agreement* (Doctoral dissertation), Indian Institute of Technology, Bombay.

[32] "edX", *edX*, 2016. [Online]. Available: https://www.edx.org/. [Accessed: 30- Dec-2015]

[33] Pellegrino, J. W., & Goldman, S. R. (1987). Information processing and elementary mathematics. *Journal of Learning Disabilities, 20*(1), 23-32.

[34] Norman, D. A. (1981). Categorization of action slips. *Psychological review, 88*(1), 1-15.

[35] Gonzalez, C. S., Guerra, D., Sanabria, H., Moreno, L., Noda, M. A., & Bruno, A. (2010). Automatic system for the detection and analysis of errors to support the personalized feedback. *Expert Systems with Applications, 37*(1), 140-148.

[36] Peng, Aihui, and Zengru Luo. (2009) A framework for examining mathematics teacher knowledge as used in error analysis. *For the learning of mathematics* 29(3) 22-25.

[37] Sarwadi, H. R. H., & Shahrill, M. (2014). Understanding students’ mathematical errors and misconceptions: The case of year 11 repeating students. *Mathematics Education Trends and Research*, 1-10.

[38] Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive science*, *2*(2), 155-192.

[39] Dhlamini, Z. B., & Kibirige, I. (2014). Grade 9 Learners’ Errors And Misconceptions In Addition Of Fractions. *Mediterranean Journal of Social Sciences*, *5*(8), 236.

[40] Makonye, J. P., & Matuku, O. (2016). Exploring Learner Errors in Solving Quadratic Equations. *International Journal of Science Education.* 12(1), 7-15

[41] Wolfram, S. (1991). *Mathematica: a system for doing mathematics by computer*. Addison Wesley Longman Publishing Co., Inc.

[42] Kutzler, B. (1996). *Improving Mathematics Teaching with DERIVE: a guide for teachers*. Chartwell-Bratt.

[43] Joyner, D., Čertík, O., Meurer, A., & Granger, B. E. (2012). Open source computer algebra systems: SymPy. *ACM Communications in Computer Algebra*, *45*(3/4), 225-234.

[44] "HP 50g CAS Enabled RPN Programmable Engineering Calculator : Educalc.net", HP-50, 2016. [Online]. Available: https://www.educalc.net/1861088.page. [Accessed: 30- Dec- 2015].

[45] Fife, J. H. (2011). Automated scoring of CBAL mathematics tasks with m-rater. *Research Memorandum. 11-12. Princeton, NJ: Educational Testing Service*.

[46] Driver, D. (2008). Pedagogical Use of a CAS. In *Proceedings of 13th Asian Technology Conference in Mathematics: Enhancing understanding and constructing knowledge in mathematics with technology*.

[47] Li, G., Li, L., Su, W., & Zhao, Y. (2006). Design and implementation of MAML. In *Proceedings of 11th Asian Technology Conference in Mathematics*.

[48] Klai, Saliha, Theodore Kolokolnikov, and Norbert Van den Bergh.(2000) Using Maple and the web to grade mathematics tests. In *Proceedings of International Workshop on Advanced Learning Technologies*. IEEE.

[49] Gould, J. D. (1975). Some psychological evidence on how people debug computer programs. *International Journal of Man-Machine Studies*, *7*(2), 151-182.

[50] Boehme, R., Fairweather, P., Farooq, U., Lam, D., & Singley, K. (2003). *U.S. Patent Application No. 10/722,926*.

[51] Heid, M. K., Blume, G. W., Hollebrands, K., & Piez, C. (2002). Computer algebra systems in mathematics instruction: Implications from research. *The Mathematics Teacher, 95*(8), 586-591.

[52] “TI-Nspire”, *TI-Nspire* Available: https://education.ti.com/en/us/products/calculators/graphing-calculators/ti-nspire-cas-with-touchpad/tabs/overview. [Accessed: 30- Dec- 2015]

[53] "ClassPad II", *ClassPad II*, 2016. [Online]. Available: https://edu.casio.com/products/cg/cp2/. [Accessed: 30- Dec- 2015]

Appendix A: Sample Marking Scheme for Linear Equation Solving Question

```
<rubric>
 <type>LinearEquations</type>
 <sectionid>1</sectionid>
 <question id="q01" totalMarks="10">
   <sub_question id="q01-01" totalMarks="3">
       <data required="true" marks="1">
         <concept>
           <name>LCD</name>
         </concept>
       </data>
       <data required="true" isFinal="false" marks="1">
         <concept>
           <name>CM</name>
         </concept>
       </data>
       <data required="true" isFinal="false" marks="1">
         <concept>
           <name>FinalAns</name>
         </concept>
       </data>
   </sub_question>
 </question>
</rubric>
```

Appendix B: Sample Marking Scheme for Quadratic Equation Solving Question

```
<rubric>
 <type>QuadraticEquations</type>
 <sectionid>1</sectionid>
 <question id="q01" totalMarks="10">
   <sub_question id="q01-01" totalMarks="5" positiveOnly="0" decimalPlaces="2">
      <data required="true" marks="1">
        <concept>
          <name>PS</name>
        </concept>
      </data>
      <data required="true" marks="1">
        <concept>
          <name>SQRT</name>
        </concept>
      </data>
       <data required="true" marks="1">
        <concept>
          <name>SUBST</name>
        </concept>
      </data>
      <data required="true" marks="2">
        <concept>
          <name>FinalAns</name>
        </concept>
      </data>
   </sub_question>
 </question>
</rubric>
```