Developing an Intelligent Tutoring System for Assessing Students' Cognition and Evaluating Descriptive Type Answers

Papri Chakraborty
School of Education Technology, Jadavpur University, India

ABSTRACT

An intelligent tutoring system (ITS) is a system based on artificial intelligent factor that performs some specific activities, such as, teaching the students, evaluating their performance with proper feedback, measuring their strength and weakness areas, etc. The paper presents architecture of an intelligent tutoring system which includes a metric to evaluate descriptive type answers. A system is designed to demonstrate the metric and it mainly focuses on delivering content material to the students, organizing test sessions and providing students with suggestive feedback after evaluation. The proposed metric measures the correctness of descriptive type answers with limited spelling relaxation and grammar checking under restricted conditions. The metric works moderately for simple sentence answers and can distinguish the answers as correct, error or elaboration.

Keywords - **Descriptive-type Answer Evaluation in ITS, Intelligent tutoring system, Knowledge Tracing in ITS, Students' Skill Assessment**

I. INTRODUCTION

The concept of using computers to assist learning process has changed the field of learning system surprisingly. At present days the systems not only deliver learning contents but also provide the learners with suggestions and feedback. Students can also ask questions and get hints to complete complex problems. With more research and developments ITS agents may model human behavior and track the students' emotional states. In commonly used Tutoring Systems the learners are generally provided with contents and the test sessions. Apart from these features it is important to determine whether the students are comfortable with the content delivery pattern or not. It is also expected that the systems will select test levels according to the capability of individual student. Moreover the evaluation process should be explicit and suggestions should be given to help the students find out their weaknesses and strengths. Another major problem area is the test sessions only allow objective type questions or MCQ type questions supported by check boxes, radio buttons or OMR sheets. The provision for descriptive type question-answer sessions is very rare. The primal objective of this paper is to deal with the above mentioned challenges. In the paper an ITS is developed through which students can not only experience a sequential user friendly learning process but also obtain feedback through a proper evaluation technique. A metric is proposed for measuring correctness of students' descriptive type answers effectively. The paper is organized into the following sections. Section II illustrates the previous work reviews on this field, section III presents the proposed approach and metric, section IV explains the implementation details of the system, section V analyses the proposed work, and section VI briefs the conclusion, probable future scope and limitations of this paper.

II. PREVIOUS WORK

Many architectures and features have been proposed on ITS. The researches are mainly based on student's knowledge tracing, assessments and evaluation. Bayesian method is followed to identify the most important learning items which can result in the most effective tutoring strategy [2, 11]. Processes like scaffolding, prompting, self-questioning simplifies student's comprehension on subject content [4, 8]. Text-relatedness metrics are used to support the authoring in ITS [5, 7]. Natural language helps to minimize the difference between Human tutors and Computer tutors and author an ITS [9, 12]. Prediction of learning styles from an ITS mark the students as sensing or intuiting, visual or verbal, active or reflective, sequential or global learners [1, 3, 13]. Neural network based methods predict student's mood during learning and test session [10, 14].

III. PROPOSED APPROACH

The paper proposes architecture to serve some basic features of tutoring system and also evaluate the students intelligently. The system focuses on the tutor expert, domain expert and student expert. The system supports two roles: tutor and student. In this work the tutor is given the privileges to access the tutor expert and domain expert. The student has the privileges to access the student model.

3.1 Identification of Roles: Tutor and Student

The tutor and student use the system according to their assigned privileges.

3.2 Content Editing

In the system an authenticated tutor can change or edit the contents easily if required. For this no programming knowledge is required.

3.3 Delivering the Contents and Test Sessions

Content material is delivered to the students. The provided test sessions are pre-knowledge test, level-I test, level-II test, level-III test and general knowledge test. The first four test sessions are objective type test. The last one is descriptive.

3.4 Pre-knowledge Assessment Using Item Effect Model and Evaluation

The pre-knowledge assessment is mandatory for every student before learning the contents. The slip rate and guess rate is derived from this test. Pre-knowledge evaluation shows slip rate and guess rate. The general assessment follows the behavioral model and it only concentrates on the current learning. The post learning evaluation is based on the differential model and epistemic level which compares the student post learning knowledge to the pre-knowledge and determines whether student's knowledge has improved or not.

3.5 Objective Type Test Session and Evaluation

This session comes after the student has completed reading the chapters. The system selects the appropriate test level for the student. The student can also select the test level of his/her own choice. There are three test levels. The sessions are time bound. After completing the tests the students are provided with the correct answers and the score.

3.6 Descriptive Type Test Session and Evaluation

The proposed metric is included in this session. The metric is supported by two sub-modules: spell-checker and grammar-checker. The students submit their answers and go to the evaluation page. The answers are classified into three categories: error, correct, and elaboration. If the answer is completely invalid or contradictory it is an error. If the sentence is syntactically and morphologically same as the required answer it is a correct answer. If the answer is non-contradictory or reasonable with respect to the required answer it is an elaboration.

3.7 Contribution

3.7.1 Spell Checker Module

The spell checker embedded to the system is applied to the descriptive type test session. When the student input an answer the spell checker provides relaxation for minor spelling mistakes. It performs operations like addition, alteration, deletion of a single character and swapping between two consecutive characters at any position of the word. With these combinations it basically searches for all the probable correctly spelled words and forms a suggestion list. The misspelled words are then substituted by the correct words with a priority of taking answer relevant words if found in the suggestion list.

3.7.2 Grammar Checker Module

The grammar checker is also applied to the descriptive type test session. It tags each word of the input answer with its parts of speech. It also defines the relation between different parts of speech and sets them as basic rules for grammar checking [15].

3.7.3 Steps to Evaluate Descriptive Type Answers

Step 1: Start.

Step 2: Form the correct answer and store all the words present in it in a master table.

Step 3: Identify and tag the key words and key verbs. Tag other words as the non-key terms. Put weights to all the words according to their importance in the answer. Calculate the weights for actual correct answer by adding all the assigned scores of all words present in it and store it.

Step 4: Insert all probable negative type words, such as, no, not, never, etc, to another table.

Step 5: Collect and insert all the probable synonyms and antonyms of the words of the correct answer to a synonym table and antonym table.

Step 6: If there exists any verb in the master table or synonym table or antonym table then insert their all possible tenses accordingly to the synonym table and antonym table.

Step 7: Assign a weight to each synonym considering the change of meaning of the sentence due to its presence.

Step 8: Set student's score for the answer to 0. Input student's answer. Split the input into words and store them in a temporary table.

Step 9: Check if the key words or key terms are present in the temporary table. If key words or key terms found put the weight for that word. score = score + new weight. Go to *Step 11.*

Step10: If key word or key verb not found then check if any synonym word of the key terms are found. If synonyms for key terms found put the weight for that word. Score = score+ new weight. Go to *Step 11.*

Step 11: If synonyms for key terms not found then stop further checking and consider the answer as ERROR.

Step 12: Check whether the input answer starts with any no-type word or not. If a no-type word is found at the beginning of the sentence then check whether there is any other no-type word present in the sentence. If found discard the first no-type word. If the number of other no-type words

found is n, score = score * $(-1)^n$. Else if the no-type word at the beginning is the only no-type word in the sentence then the total score of the sentence should be multiplied by -1.

Step 13: Check if any other antonym found in the temporary table. For each antonym the net weight should be multiplied by -1. If number of antonym is n, score = score * $(-1)^n$

Step 14: Check if any antonym is present in the temporary table. For each antonym of the key terms the net weight should be multiplied by -1. If the number of no-type words is n, then score = score * $(-1)^n$

Step 15: Check the position vectors of the nouns and verbs combination in the input answer and compare it to that of the correct answers to verify the dependencies of the nouns and the verbs in the answer.

Step 16: Put the weights of the non-key terms accordingly. Put weight 0 for any unknown word found. Now calculate the net weight.

Step 17: Check if there exists any grammatical error in the sentence. For each grammatical error deduct 2 from the net score.

Step 18: If the net score is negative the answer is an ERROR. If the net weight is positive and in the range of 20% of original score then the answer is CORRECT. If the net weight is positive but does not fall in the range then the answer is ELABORATION.

Step 19: End.

IV. IMPLEMENTATIONS

The proposed architecture and algorithm of the metric is implemented in Microsoft Visual Basic and Oracle Database. There is option for user to select their roles for using the system. Without having a valid authentication code no user can sign in to the system as a tutor. Fig. 1 shows the authentication page.

Fig. 1: Tutor login page

The tutor home page shown in Fig. 2 has the links, such as, read chapters; edit chapters; view current question-answer sets; edit questions; edit answers; discussion page; and database management pages.

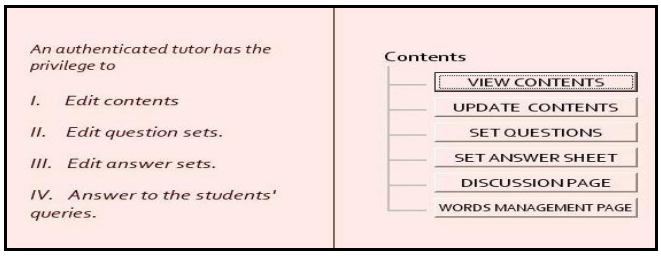

Fig. 2: Tutor home page with necessary links

The tutor can edit the chapter contents and question-answers sets as per requirement and save them accordingly. Fig. 3, 4, 5 show various pages with edit and save options.

Fig. 3: Chapter contents get edited and saved

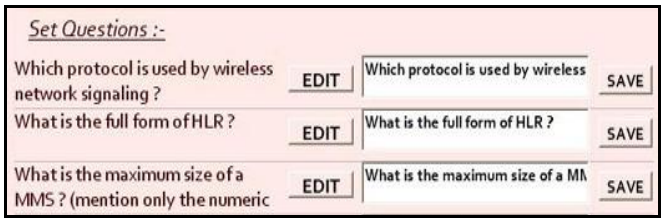

Fig. 4: Question sheet gets edited and saved

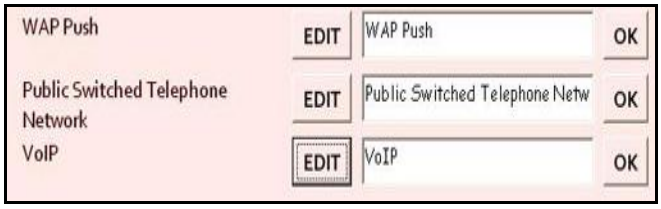

Fig. 5: Answer sheet gets edited and saved

The database management page as shown in Fig. 6 allows the tutor to enter new words, synonyms, antonyms with their weights in the tables used for the descriptive type answer evaluation.

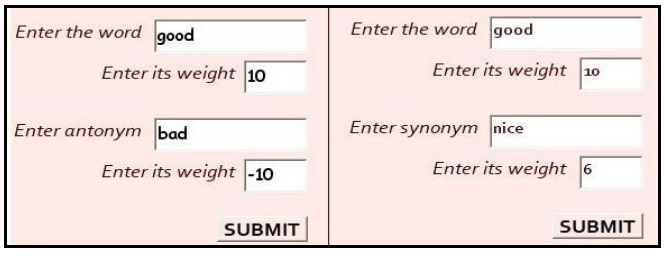

Fig. 6: Synonyms and antonyms inserted into tables

Any user who wants to login as a student will have to create a new account and sign in to the system by giving valid username and password. Fig. 7 and Fig. 8 display the login page and the registration form page respectively.

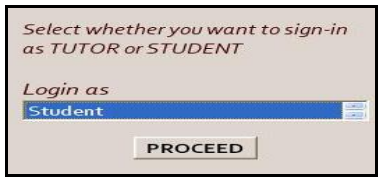

Fig. 7: Student login page

Fig. 8: New student fills up registration form

After signing in the student has to give pre knowledge test. Student has to answer each question on the basis of previous knowledge and enter the expected score for this test. The system calculates the score from the student's given answers and then calculates the slip rate and guess rate. If the student scores less than the expected score then it is depicted that he has misconception about that topic and he has considered some wrong answers as the correct answers. In this case the difference between his score and expected score in pre-knowledge test is termed as the slip rate. If the student scores more than the expected score then it is assumed that he is not confident about all the answers attempted or he has guessed some of the answers which have turned to be the right answers fortunately. In this case the difference between his score and the expected score is termed as the guess rate. Fig. 9 shows the pre knowledge test page.

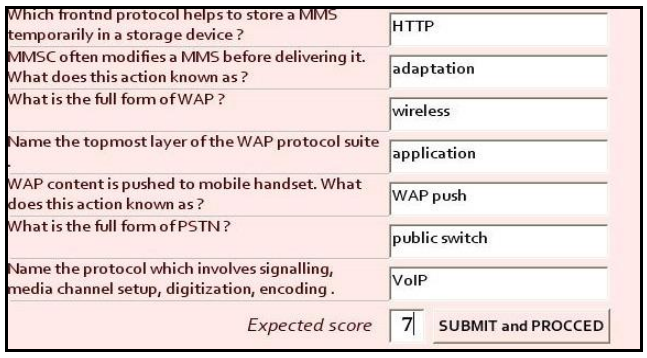

Fig. 9: Pre knowledge test session

The student enters the home page which has the links of reading page as shown in Fig. 10.

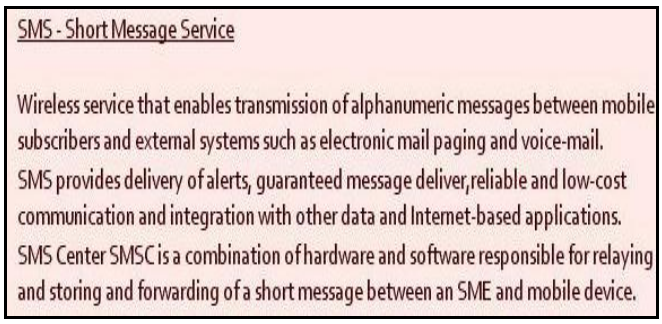

Fig. 10: Student's reading page

Fig. 11 shows that student's activities are traced and suggestions for selecting test levels are provided.

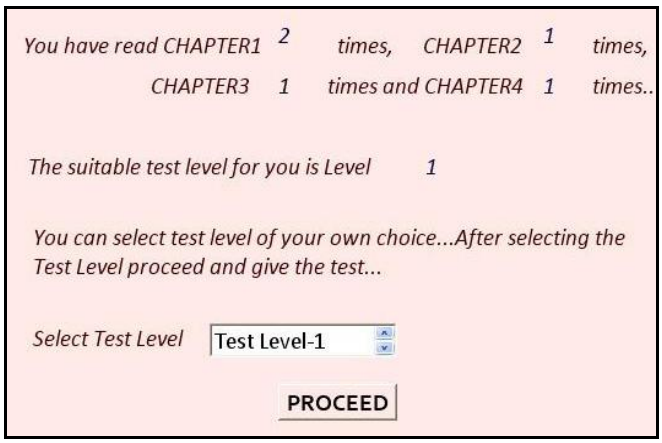

Fig. 11: Suggesting appropriate test level

The student home page has the links for three different test levels. Test sessions are time bound as shown in Fig. 12.

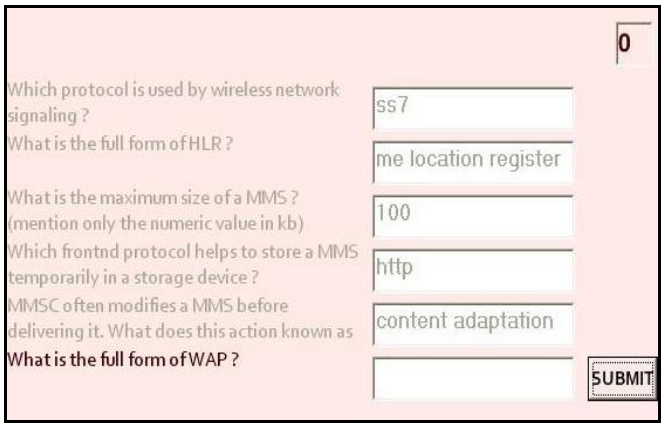

Fig. 12: Test session with timer activated

After submitting their answers the students get the correct answer sheets for the test levels. Fig. 13 shows the correct answer sheet.

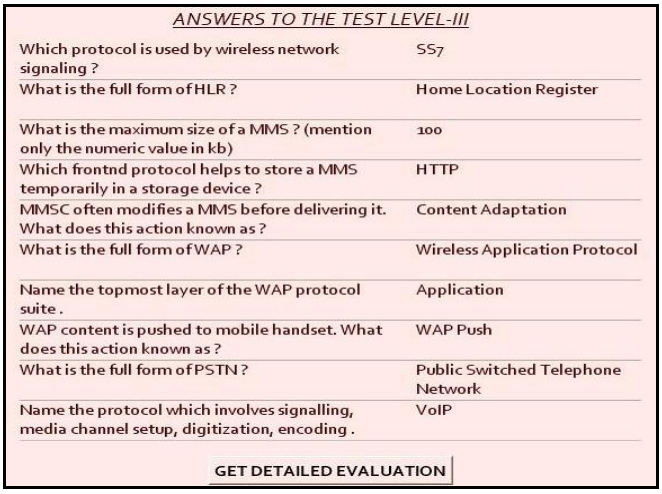

Fig. 13: Answer sheet provided to student after the test.

After tests detailed evaluation on previous knowledge, general overview and post learning is provided. Fig. 14 shows the screenshot of the evaluation page.

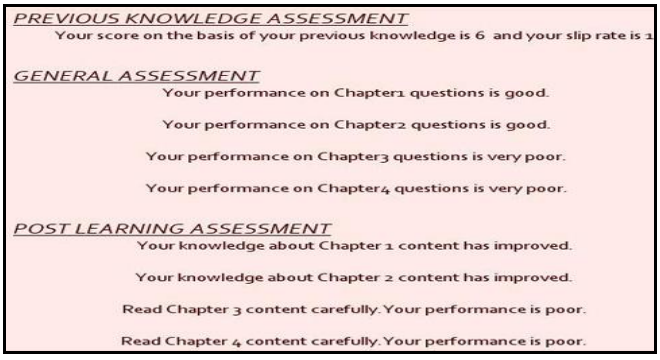

Fig. 14: Assessments on previous knowledge, general evaluation and post learning.

Then descriptive type general knowledge test is attended. Fig. 15 shows the work of the spell checker. Fig 16 and Fig. 17 show screenshots of different answer feedback.

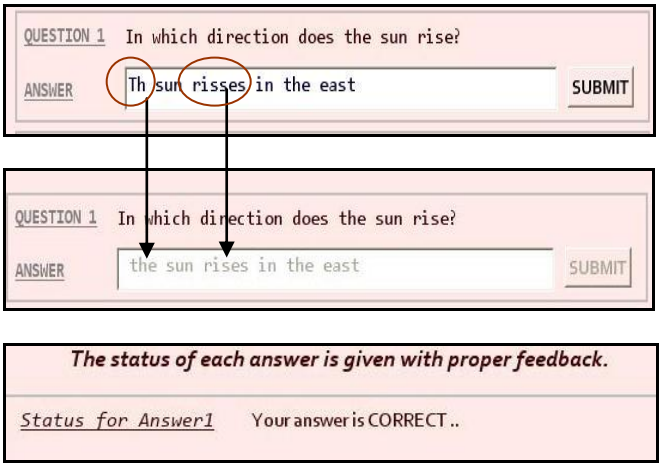

Fig. 15: Spellings get corrected and answer status is shown

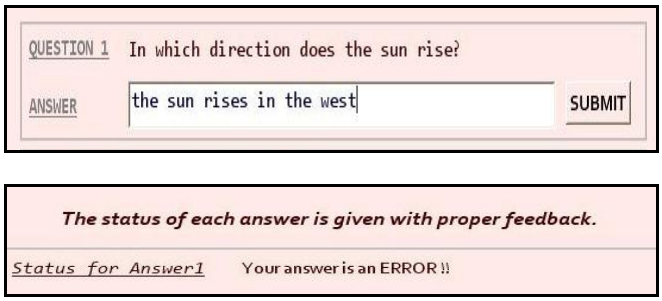

Fig. 16: The wrong answer detected and tagged as an error

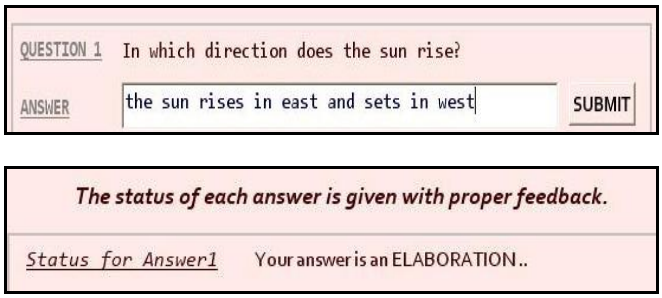

Fig. 17: The answer is an elaboration

V. ANALYSIS

Evaluating descriptive type answers is the main challenge of this paper. The evaluation process basically compares the student input to the correct answer, i.e., it checks the text relatedness between the two answers. The proposed metric minimizes the basic problems of commonly used text relatedness metric and provides moderate results. Unlike the existing intelligent tutoring systems the designed system allows both objective and descriptive type tests. Apart from generating scores the system also provides detailed evaluation based on pre knowledge of the students, general overview on answers given and comparative analysis on pre knowledge and post knowledge. Hence the system can predict how much a student has improved after going through the contents. The system if required can break the contents into sub modules guided with some templates to make the student aware of important keywords. The work has used some of feasible techniques used in previous works [5, 6, 8, 11] and has proposed more accurate assessment metric that allows a limited spelling relaxations and grammar checking under restricted condition and gives better results in evaluating the students' cognition.

VI. CONCLUSION AND FUTURE SCOPE

This paper presents an assembled approach with a number of essential features of an ITS including a metric embedded to it. The metric helps improving the evaluation of descriptive type test sessions. The system demonstrated with the proposed approach can be used as a limited content delivering tutoring system. The contents provided can be

edited by an authenticated tutor. The evaluation is done through various levels of test sessions. Feedbacks given are explicit and suggestive. The metric of evaluating the descriptive answers works best for simple sentences. The provision to enter new words in the databases makes the databases flexible. As it is dependent on spell checking and grammar checking the result may not always be satisfactory. There are definitely scopes for future research work and implementation on this algorithm and architecture. The word dictionary and databases used in the metric can be upgraded for more accurate results. Plug-ins can be used to develop well-formed spell checker and grammar checker to obtain better results. With an improved grammar checker the metric can also support compound and complex answers.

ACKNOWLEDGMENT

I express deepest thanks to my research guide, Smt. Saswati Mukherjee, Lecturer, School of Education Technology, Jadavpur University, India, for her valuable guidance during this research work. I owe a special thanks to my friend, Mr. Bishnu Prasad Das, School of Education Technology, Jadavpur University, India, for his constant support and assistance. Last but not the least I would like to express my gratitude towards my parents for encouraging me with their blessings and best wishes.

REFERENCES

[1] Richard M. Felder, Linda K. Silverman, Learning and Teaching Styles in Engineering Education, *Engineering Education, 78(7),* California, 1988, 674-681.

[2] Hugo Gamboa, Ana Fred, Designing Intelligent Tutoring Systems: A Bayesian Approach, *ICEIS Artificial Intelligence and Decision Support Systems*, 2001, 452-458.

[3] Sherry Y. Chen, Robert D. Macredie, Cognitive styles and hypermedia navigation: Development of a learning model, *Journal of the American Society for Information Science and Technology, 53(1),* 2002, 3-15.

[4] Y H Chan, Randomized Controlled Trials (RCTs) – Essentials, *Basic Statistics for Doctors, 44(2),* Singapore, 2003, 060-063.

[5] Philip M. McCarthy, Vasile Rus, Scott A. Crossley, Arthur C. Graesser & Danielle S. McNamara , Assessing Entailer with a Corpus of Natural Language From an Intelligent Tutoring System, *Proceedings of FLAIRS Conference (2007),* Memphis, USA, 2008, 247-252.

[6] Heinrich Christian Dippel, Volker Neundorf, Vera Yakimchuk, WebSIS – a web based portal with an integrated e-assessment environment, *Conference ICL 2008*, Austria, September 24-26, 2008.

[7] Philip M. McCarthy, Vasile Rus, Scott A. Crossley, Arthur C. Graesser & Danielle S. McNamara , Assessing Forward-, Reverse, and Average-Entailer Indices on Natural Language Input from the Intelligent Tutoring System, iSTART, *Proceedings of the Twenty-First International FLAIRS Conference (2008),* Memphis, USA, 2008, 165-170.

[8] Jack Mostow & Wei Chen, Generating Instruction Automatically for the Reading Strategy of Self-Questioning, *Project LISTEN*, Pittsburgh, USA, 2009.

[9] Sung-Young Jung, Using Natural Language to Represent Knowledge in an Intelligent Tutoring System, *Intelligent Systems Program*, Pittsburgh, Pennsylvania, USA, September 1-4, 2009.

[10] Christos N. Moridis, Anastasios A. Economides , Prediction of student's mood during an online test using formula-based and neural network-based method**,** *Computers and Education*, University of Macedonia, Greece, 2009.

[11] Zachary A. Pardos, Matthew N. Dailey, Neil T. Heffernan, Learning what works in ITS from non-traditional randomized controlled trial data, *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, Pittsburgh, 2010, 1-10.

[12] Sung-Young Jung and Kurt Van Lehn, Developing an Intelligent Tutoring System Using Natural Language for Knowledge Representation, *Intelligent Tutoring Systems. 10th International Conference*, ITS 2010, Pittsburgh, USA, June 14-18, 2010, Proceedings, Part II.

[13] Annabel Latham, Keeley Crockett, David McLean and Bruce Edmonds, Predicting Learning Styles in a conversational Intelligent Tutoring System**,** *Advances in Web-Based Learning - ICWL 2010, Lecture Notes in Computer Science 6483*, Manchester, 2010, 131–140.

[14] John R. Anderson, Shawn Betts, Jennifer L. Ferris, and Jon M. Fincham, Neural imaging to track mental states while using an intelligent tutoring system, *PNAS, 107(15),* Pittsburgh, USA, 25th January, 2010, 7018-7023.

[15] Debela Tesfaye, A rule-based Afan Oromo Grammar Checker, *International Journal of Advanced Computer Science and Applications (IJACSA), 2(8),* Ethiopia, 2011, 126-130.