An Expert System for Automated Essay Scoring (AES) in Computing using Shallow NLP Techniques for Inferencing

Ade-Ibijola, Abejide Olu
Department of Computer Science,
Federal University of Technology,
Akure, Ondo State, Nigeria

Wakama, Ibiba
Department of Computer Science,
Federal University of Technology,
Akure, Ondo State, Nigeria

Amadi, Juliet Chioma
Department of Computer Science,
Federal Polytechnic, Idah,
Kogi State, Nigeria

ABSTRACT

In this paper, we present the report of the development of an Expert System (ES) that; acquires the knowledge of Subject Matter Experts (SMEs) in a specific computing field, "Software Engineering", uses a built-in Inference Engine designed with Shallow Natural Language Processing Techniques (*Information Extraction* using Tokenization, Statistical Keyword Analysis and Domain-Specific Dictionary) and a Fuzzy-Scoring Model to assess Students' Free-Text Answers to Open-Ended Questions and hence, computes the correctness of students' answers with respect to lecturers' underlying *model answers* or *templates*. The newly developed ES was adapted to an academic course in the University System and its performance was evaluated using certain Statistical Metrics. The results from the evaluation were compared with existing Automated Essay Scoring Systems (AESs) using certain thresholds and conclusions were drawn.

Keywords

Expert System, Automated Essay Scoring (AES), Free Text Answers, Open-Ended Questions, Fuzzy-Scoring Model

1. INTRODUCTION

Using computer programs to score essays (or free text answers) has been studied extensively in recent years. The importance of having effective systems for this purpose cannot be overemphasized. Several models, approaches, solutions, applications and add-ins have hence been developed by researchers and software vendors alike for academic and commercial purposes. Some notable and highly rated works, in terms of performance and efficiency, include; the Project Essay Grader (PEG), Intelligent Essay Assessor (IEA), Electronic Essay Rater (E-Rater) and Intellimetric. Most of these works are underpinned by Natural Language Processing (NLP) techniques or often, a hybrid of variety of NLP techniques such as; the Statistical Keyword Analysis, Surface Linguistic Feature Analysis, Latent Semantic Analysis, Text Categorization, Full Natural Language Processing (NLP), and Information Extraction Algorithms. Correspondingly, the areas of applications of Expert Systems (ES) have enormously grown. An ES is an intelligent computer program designed to simulate the problem-solving behavior of a human who is an expert in a narrow domain or discipline such Medicine, Agriculture, Environmental Management, Personnel Management and Education [2][9][19].

In this paper, we have developed an ES for scoring free-text answers and adapted this ES to the Academic Assessment process in the Nigerian University System, using the Computer Science Programme of The Federal University of Technology Akure (FUTA) as a test case. From a development standpoint, we built a Knowledge Base and populated it with Answer Templates (or Model Answers) from Lecturers on a specific course, designed an Inference Engine using *Information Extraction* (a shallow NLP technique which is an hybrid of Statistical Keyword Analysis technique and Pattern Matching with Domain Specific Dictionary), attached a Fuzzy-Module for correctness evaluation and developed two-web applications; one as a user-interface for lecturers to set their test questions and supply answer templates, and the other for Students' to write open-ended tests online and obtain an instantaneous feedback of their performance. In the concluding section, an attempt is made to compare the performances of existing AES systems with the newly developed ES.

The remaining of this paper is organized as follows: section 2 attempts to pinpoint the essence of this research, section 3 discusses the problem domain, section 4 attempts to justifies this work, section 5 presents the design of the new ES and section 6 its implementation details, section 7 offers few screenshots from the new application, section 8 presents the testing and performance evaluation details, section 9 states the limitations of this research, section 10 offers a conclusion succeeded by an acknowledgement and a short list references.

2. THE PROBLEM

Essays are considered by many researchers as the most useful tool to assess learning outcomes, implying the ability to recall, organize and integrate ideas, the ability to express oneself in writing and the ability to supply merely than identify interpretation and application of data [20]. It is in the measurement of such outcomes, corresponding to the evaluation and synthesis levels of the Bloom's (1956) [6] taxonomy that the essay questions serve their most useful purpose.

One of the difficulties of grading essays is represented by the perceived subjectivity of the grading process. Many researchers claim that the subjective nature of essay assessment leads to variation in grades awarded by different human assessors, which is perceived by students as a great source of unfairness. This issue may be faced through the adoption of automated assessment tools for essays. A system for automated assessment would at least be consistent in the way it scores essays, and enormous cost and time savings could be achieved if the system can be shown to grade essays within the range of those awarded by human assessors and trained with expected students' responses. This problem has been the basis on which researches on Automated Essay Scoring (AES) are motivated and hence conducted.

3. AES AND NLP

An AES system is a computer technology that evaluates and scores written text. AES systems can be very useful because they can provide the student with a score as well as feedback within seconds [15]. The research on AES has revealed that computers have the capacity to function as a more effective cognitive tool [3]. Although AES is a developing technology, the accuracy and reliability of AES systems have been proven

to be high. The search for excellence in machine scoring of essays is continuing and numerous studies are being conducted to improve the effectiveness of these systems.

The importance of having AES systems has been studied. Page (2003) [15] claims that "the automated ratings would surpass the accuracy of the usual two judges". Machine scoring technologies can also increase the practicality in administering large scale assessments of writing ability [5]. A number of studies have equally been conducted to prove the accuracy and reliability of AES systems with respect to the writing assessment and the agreement rate between human raters and AES systems. These correlations have been found to be high [14][18].

The issue of "*what is the most appropriate type of question in testing students' ability*?" has also been studied. There will be no need for AES systems if multiple choice question suffices in a given scenario or domain of discuss. However, the shortcomings of multiple-choice questions have also been addressed by Bloom (1956) [6]. Bloom provided a taxonomy for categorizing the level of abstraction of questions used in the assessment of student work. He identified six different levels: knowledge, understanding, application, analysis, synthesis and evaluation. This taxonomy has been taken as the starting point for analyzing the student's learning competence. Many authors agree that multiple-choice questions only serve to evaluate the lower levels in the taxonomy. When it is necessary to measure the higher levels, open-ended questions should be employed [11][12][16]. The desire to administer open-ended questions to students and the benefits that comes with automated grading of essays justifies the development of any AES system.

Conversely, there have been hard critics about the idea of a computer grading human essays. Nowadays, there are still some skeptical researchers that do not consider that the automatic grading is possible. However, the advances in Natural Language Processing (NLP), Machine Learning and Neural Network techniques, the lack of time to give students instantaneous feedback (despite the general assumption of its importance) and the conviction that multiple-choice questions cannot be the only assessment method are favoring a change in this situation.

The fields of application of AES systems are boundless. Several automated assessment techniques have appeared recently, and some of them are even commercially available. Above and beyond, several traditional tests such as the Graduate Management Admissions Test (GMAT), the Test of English as a Foreign Language (TOEFL) or the Graduate Record Examination (GRE) are including open ended questions with a computer-based delivery, which, to a large extent, supports the use of automated scoring methods. However, every AES system uses some underlying technique or hybrid of techniques for correctness evaluation of essays. These techniques are referred to as Natural Language Processing (NLP) techniques.

NLP is the field of science concerned with techniques, models and algorithms used in processing and interpreting essays. Mitchell et al. (2002) [12] classified the techniques for automatic scoring of free-text responses in three main kinds: Statistical, Information Extraction and Full Natural Language Processing. The Statistical approach, when it is only based on keyword analysis, has usually been considered a poor method, given that it is difficult to tackle problems such as synonymy or polysemy in the student answers, it does not take into account the order of the words and it cannot deal with lexical variability. On the other hand, a full text parsing and semantic analysis is hard to accomplish, and very difficult to port across languages.

Information Extraction (IE), which is the technique used in this work, is in the middle of the Statistical and the full NLP approaches. It only requires shallow NLP without doing an in-depth analysis and it is more robust than ordinary Keyword Analysis. IE techniques pertain to acquiring structured information from free text, e.g. identifying Named Entities in the text and filling in a template. IE may be used to extract dependencies between concepts. Firstly the text is broken into concepts and their relationships, then the dependencies found are compared against the human experts to give the student's score. For example, Automark and ATM are based on this approach. An example of a marking scheme presented by Mitchell et al. in [13] is shown in Fig 1. Pattern-matching is a technique commonly used for IE. It consists in looking for specific information in the student's answer in order to fill in the template that the human experts have previously done. The filled template is compared against the model to calculate the final score.

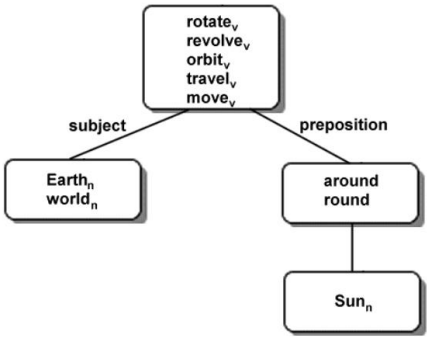


Fig 1: Example of a scheme used in Automark to score the answer to the question like "What movement relates the Earth and the Sun?" (Source: Mitchell et al., 2003)[13].

Full NLP is the application of computational methods to process natural language. Burstein et al. (2001) [7] cited tools such as syntactic parsers to find the linguistics structure of a text [1] and rhetorical parsers to find the discourse structure of a text [10]. In addition, Williams and Dreher (2004) [21] employed electronic thesaurus to extract lexical information and a specifically designed chunking algorithm to extract noun phrases and verb clauses. The C-rater and PS-ME (Paperless School Marking Engine) are also underpinned by these techniques. Their combination improves the use of statistics by involving a deep text parsing and a semantic analysis in order to gather more information to effectively assess the student's answer. On the other hand, it is hard to accomplish and more difficult to port across languages. It is also important to notice that there may be other techniques which are not considered here as the systems that implement them have become commercially available, and thus, their implementation details are not longer published in scientific forums.

4. A NEED FOR AN EXPERT SYSTEM FOR ESSAY SCORING

Perhaps in Academics, the knowledge of Human Experts can be easily converted into texts? The traditional believe in Academics is that knowledge do not only reside in peoples' brains but also in libraries of books, articles and journal papers written by these people. This fact makes the knowledge acquisition process of developing an ES for essay

scoring in an Academic Environment a "piece of cake". Furthermore, most AES systems basically take as input, an essay and an answer model, and outputs a score but having an ES to do this job enables us to have a knowledge base which: can be trained from the start, can grow in amount of knowledge contained, can be easily modified and maintained while the performance of its essay scoring increases as a function of time. Another advantage we stand to gain by having an ES for Essay Scoring is that since ESs can be developed in modular parts (consisting of a separate Knowledge base, separate Inference Engine and a separate Working Memory), one may choose to maintain each of these component parts as a separate application or module and improve on it, independent of the entire ES. For a specific example, the ES developed in this work is currently undergoing a modular upgrade for the Inference Engine Module to be able to support Inferencing using the Rete's Algorithm for pattern matching and hence improve performance.

5. THE DESIGN: ES4ES

The Expert System developed in this work was named "Expert System for Essay Scoring" and abbreviated as "ES4ES". An expert system is typically composed of at least three primary components: the Knowledge base, the Inference Engine, and the working memory. In this section we present the design of these components.

5.1 Knowledge Base (KB): Specification and Design

KB specification entails identifying the; facts (one piece of Information about a subject), subject (the thing that the fact is about), terms (one or more words that, when used together, identify a subject), predicates (the part of a fact that consists of a verb), complements (the part of a fact that gives the value of the predicate), the hierarchy of relationship between the terms of the subject, and a description of the inheritance and multiple inheritance amongst terms.

The knowledge base for ES4ES is based on a course offered in the Department of Computer Science, Federal University of Technology, Akure (FUTA). The description of the course is given in the Tables 1(a) and (b). An enumeration of the few key terms in the subject (Software Engineering) to be housed in the KB is presented in Table 1(c). The extent of the facts contained in the KB designed in this work is restricted to focus on questions and answers arising from the definition, discussion or explanation of the terms listed in Table 1(c).

The design of the KB created in this work uses the Semantic Network approach for knowledge representation. This approach is based on object models. One of the innovative ideas in object modeling is inheritance. Inheritance comes from the recognition of the hierarchy of ideas and concepts, and how this hierarchy/classification involves much inherent reuse of ideas and so on from the higher-level concepts to the lower-level specialization of those concepts. Fig. 2 presents the Semantic Network that identifies the commonalities between the terms and sub-terms of the subject (Software Engineering) defined by the KB for ES4ES. The following prepositions are formulated from Fig. 2; "Software Talks about Testing", "Software Talks about Project", "Software Talks about Metrics", "Software Talks about Systems", "Software Talks about Quality Assurance", and "Risk Talks about Project". Hence, the relationship between the objects of the Semantic Network is the "Talks-About Relationship".

Table 1(a) – course description of the subject

Course Code	Course Title	Course Unit
CSC 409	Software Engineering I	2

Table 1(b) – Course Content

Course Content
Introduction to the techniques and methodologies of software engineering, requirements analysis and definitions, specification, software design, quality assurance, testing, reuse, development tools and environments.

Table 1(c) – Subject terms covered

Software Life Cycle	Acceptance Testing	Delivery	Software Reliability
Software Metrics	Unit Testing	Risk	Maintenance
Product Metrics	System Testing	Risk Identification	Project Management
Process Metrics	Integration Testing	Risk Estimation	Market Analysis
Software Design	Regression Testing	Risk Exposure	Alpha Testing
Implementation	Beta Testing	Risk Mitigation	Quality Assurance

5.1 The Design: ES4ES

The Expert System developed in this work was named "Expert System for Essay Scoring" and abbreviated as "ES4ES". An expert system is typically composed of at least three primary components: the Knowledge base, the Inference Engine, and the working memory. In this section we present the design of these components.

5.2 Working Memory (WM)

Working memory refers to task-specific data for a problem. The content of the working memory changes with each problem situation. Consequently, it is the most kept current. In the implementation logic of this work, the WM was realized with multidimensional arrays, generic lists and data tables which are temporarily kept in the Random Access Memory (RAM) and discarded when the application terminates. In situations where array indexes grew too large or WM data growing beyond certain threshold, other data structures such as XML files were used for storing larger data outside the application and flushed when the application terminates.

5.3 Inference Engine

The inference engine of the new ES was coded as classes in the implementation language (VB.Net/ASP.Net using the IF-THEN clauses) from which instance objects were created and the knowledge base manipulated. These classes and their respective methods are hierarchically structured and the details of the information contents increased downwards. Hence, a method of a class in the hierarchy can call another method of a class at lower level thereby supporting the forward chaining strategy of inference engine. The upward referencing of methods across classes in the hierarchy is also allowable and this supports the backward chaining strategy of inference engine. A textual algorithm describing the Inference Process is stated in Algorithm 1.

Algorithm 1: Expert System Inference Engine Algorithm

For each *Template* in Templates(term)

Initialize: Percent_Match=0.0; Match=0;

For each *keyword* in Input_Text

IF *keyword* is a *substring* in *Template*

Match+=1; Remove *keyword from template;*

ELSEIF *keyword* is a [Domain-Dictionary].*synonym* of a substring in *templates*

Match+=1; Remove *keyword from template*

END IF

Next *Keyword*

Percent_match=(Match/keyword_count)*100;

Next Template

5.4 Domain Specific Dictionary (DSD)

In order to enhance the effectiveness of the Inferencing process of ES4ES and to support shallow NLP, synonyms had to be handled using a Database Dictionary of Software Engineering Terms. We refer to this file as the Domain Specific Dictionary (DSD). The DSD was created with Microsoft's SQL Server 2008 Database Management System using the information mined from Online Glossaries of Software Engineering Terms.

5.5 A Fuzzy Model for Correctness Evaluation

Here we present a fuzzy model for the scoring of students' response in the inference process. The model generates the score of a student in a free-text response given two parameters; the *Percentage Match* of a student's response to the underlying lecturer's model answer (denoted as 'X') and the *Mark* assigned to the question being answered by the student (denoted as 'Y'). The fuzzy function is given in (1) below.

$$Score(X,Y) = \left\{ \begin{matrix} y, if\ x \geq 70 \\ 0.5 \times y, if\ 70 > x \geq 40 \\ 0, if\ x < 40 \end{matrix} \right\} \ldots \qquad (1)$$

The scoring model in (1) above is a fuzzification of the real-life approach within the system of the studied institution; FUTA. Often, in manual marking, awarded score to students' free-text answers to open-ended questions is either full-mark, half-mark or zero.

5.6 NLP Module

As earlier discussed, ES4ES uses some form of NLP (shallow NLP) to aid the inference engine in Pattern Matching from the inference rules to the data (facts) contained in the knowledge base. The NLP module contains: a Lexical Analyzer, a Filter and a Synonyms Handler module. These sub-modules are implemented as VB.Net classes.

The Lexical Analyzer tokenizes the *model answer* supplied by the course lecturer via the ES's training web interfaces or the *test answers* supplied by the students via the Online Test Engine and pass the stream of tokens to the Filter Module. The Filter Module takes the stream of tokens from the Lexical Analyzer and Extracts the Predicates and Complements from the Facts contained.

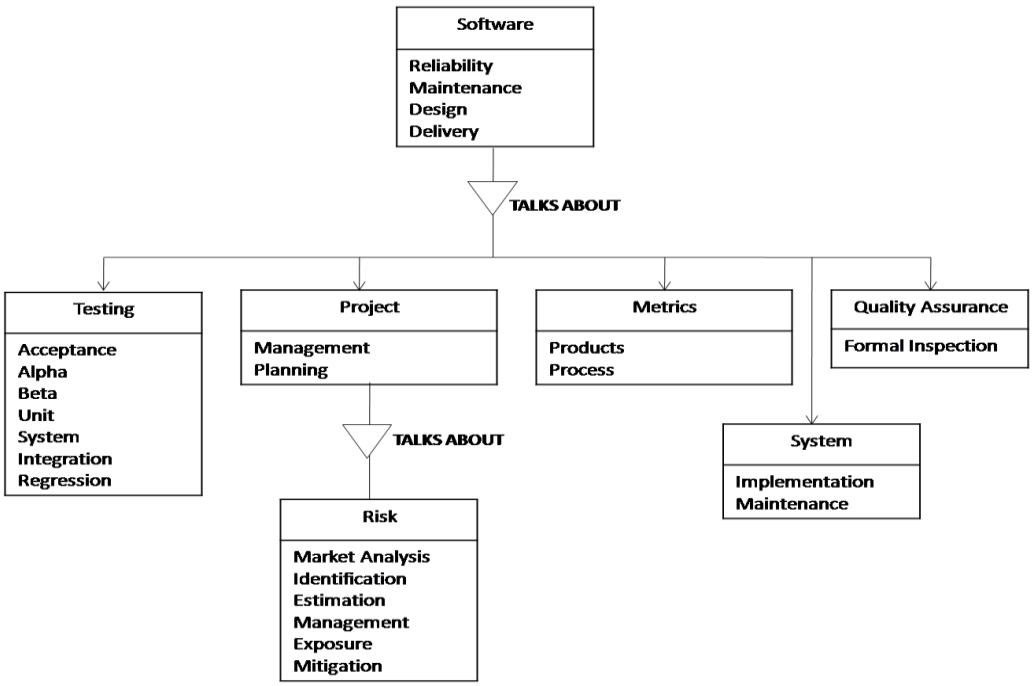


Fig. 2: Semantic Network for the new Knowledge Base

The last Class of the NLP modules, Synonyms Handler, performs an m-to-n mapping of the keywords/tokens contained in the current query string to all its synonyms in the DSD and serializes the produced schema into a XSD (XML Schema Definition) file which is converted into binary data streams and stored in the database for subsequent use in the inference process.

5.7 User-Interface

User-Interface is a mechanism by which users and systems communicate. Bethune et al(2007) [4] described the user-interface as a screen that allows the user to input information in response to questions generated by the system. In most implementation scenarios specific to ES applications, the user interface will also be able to give advice and most importantly, explain why it is giving that advice. In the implementation of ES4ES, the user-interface of the system is broken down into two sub-components, namely; The "ES Training Web Application" and the "Online Test Simulator". The Training Web Application of the ES4ES collects the facts about the terms in the subject domain, structures it using the NLP Modules and stores it in the knowledge base.

The Online Test Simulator on the other hand, is a Web Application that imitates (or simulates) a standardized online test engine, collates students' responses to open-ended examination questions and passes the responses to the *inference engine module* which uses the *fuzzy-scoring module* to assess the responses according to the knowledge contained in the system. It was sole-called "Simulator" because certain system requirements (such as application security) which are of less concern to this research were not considered in its developmental model and as such it is not a full Online Test Engine but certainly suffices to test the newly developed ES.

5.8 ES4ES's Architecture

In this section we describe the full architecture of the new ES, ES4ES, with Fig 3. The ES comprises of the knowledge base specified earlier in section 5.1, a working memory whose definition is stated in section 5.2, an inference engine whose algorithm is described in section 5.3 and stated in Algorithm 1, a Domain Specific Dictionary (DSD) illustrated in section 5.4, a Fuzzy-Model for Correctness Evaluation described in section 5.5 and its mathematical definition stated in Equation 1, an NLP Module whose components are highlighted in section 5.6 and the user interface which comprises of two web applications and described in section 5.7.

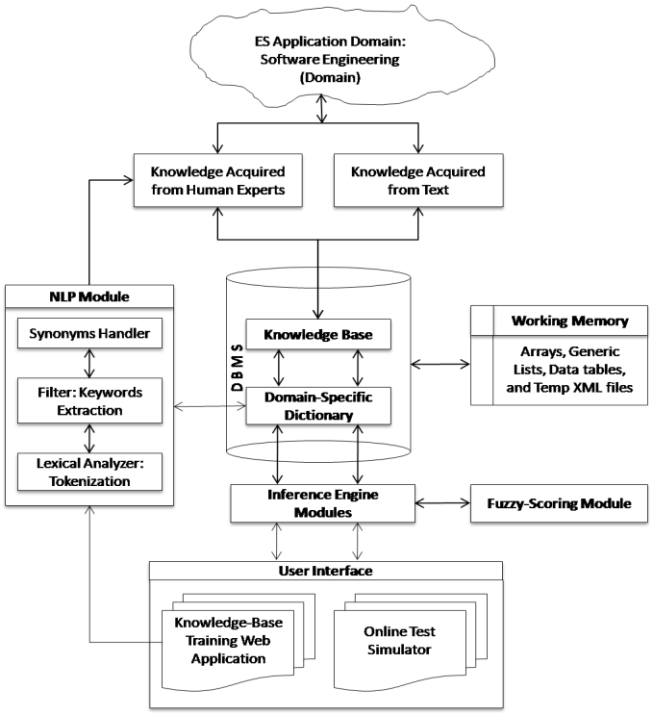


Fig. 3: Architecture of the new ES

6. IMPLEMENTATION DETAILS

ES4ES was programmed with Microsoft's Visual Basic.Net on the .Net Framework 4.0 using Visual Studio Integrated Development Environment 2010 (VS-IDE) for Interface Designs of Active Server Pages (ASP) and Code Editing, Microsoft's SQL Server 2008 for Database Management of the Knowledge Base and Domain Specific Dictionary, Internet Information Service (IIS) for local web hosting and Crystal Reports 2008 for designing the feedback presented to students after writing the online test via the Online Test Simulator.

In implementing ES4ES, certain functional user and system requirements were identified and considered. The main task of the Automated Essay Scoring is, as the name already implicates, the evaluation of the correctness of some students' answers. Hence, the assessment process requires that:

In the Online Test Simulator Web Application:

(i). A student may retrieve some already stored test questions.

(ii). The student is able to state an answer, where the system must uniquely attribute the answer to the student, which requires that students' names and matriculation numbers are remembered by the system.

(iii). A student who has not submitted a test response should be able to modify his/her answers. Once the submit button is clicked, the student should not be able to modify the answer.

(iv). Students should not be able to view the knowledge base of stored answers.

(v). The system must be able to compare the students' answer to the model answer(s), and generate some score value depending on the correctness of the students' answer.

(vi). The system must be able to present the score or some grade to the student as feedback describing students' performance.

(vii). Furthermore, as traceability of the answering and grading process is a desirable property of the system, some time-stamps, comments, or some other logging information should be provided.

In addition, as the grading process should be inspected, modifiable and influenceable by humans, a user who assumes a role similar to a lecturer in this system (Academic Environment) must be able to:

Use the Knowledge Base Training Web Application to:

(i). Create test (and populate it with questions) for students to write,

(ii). Add to, Edit, or Remove from questions of a test,

(iii). Populate the KB with as many model answers as possible for each test question,

(iv). Edit/Remove previously stored knowledge (in the KB) or synonyms (in the DSD),

(v). Retrieve the students' answers for a specified test,

(vi). Change the score and comments provided by the system,

(vii). Allow a student to answer a question again, and

(viii). Perform certain statistical analysis.

Needless to say, as these services could be used to manipulate, or even damage the assessment system, they must only be available to lecturers.

7. SCREENSHOTS

In this section, we present few screenshots from both web applications that represent the user-interface of the developed ES. Fig 4 shows the knowledge acquisition process of the ES, Fig 5 - the KB Update process, Fig 6 – the login page for OTS, Fig 7 – a test in progress with questions 1 and 4 already answered by the student, Fig 8 – test questions and assigned marks for a specific test tagged "SE-" and Fig 9 showing a student's feedback after submitting test responses.

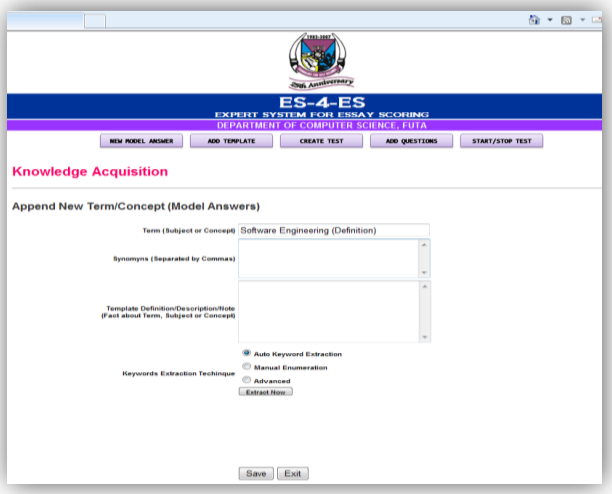


Fig. 4: Knowledge Acquisition Process

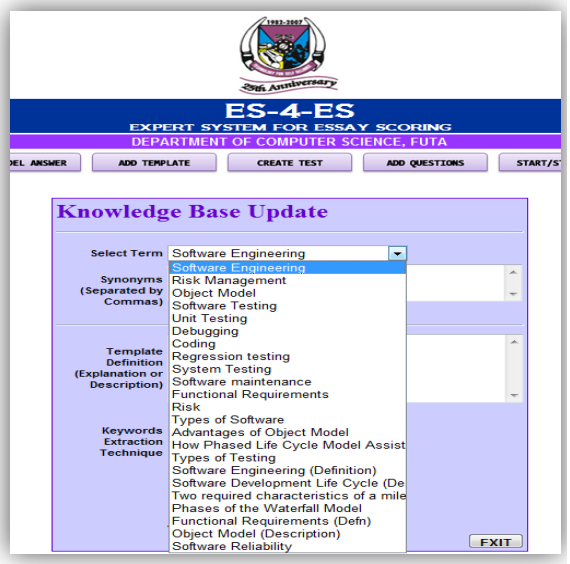


Fig. 5: Knowledge Base Update

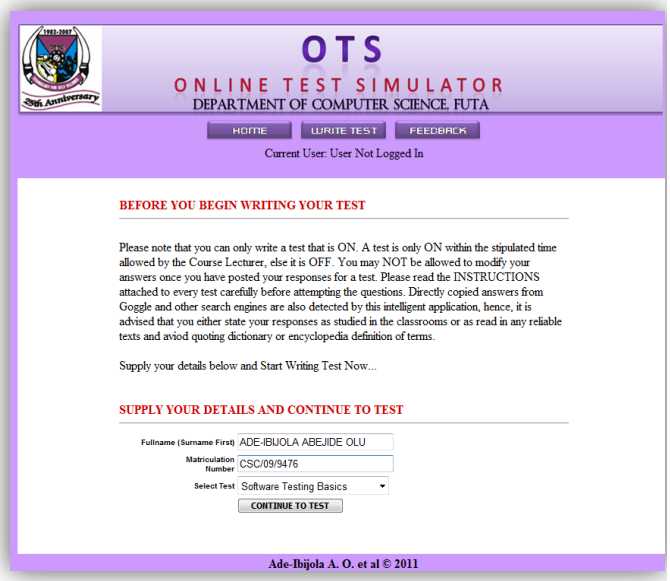


Fig. 6: OTS Login Page

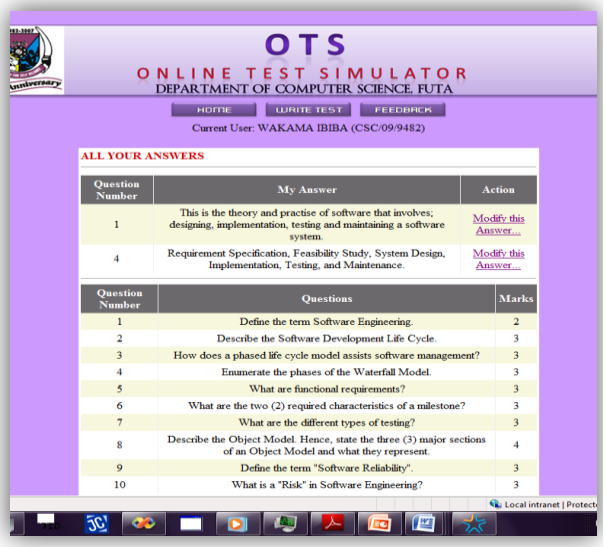
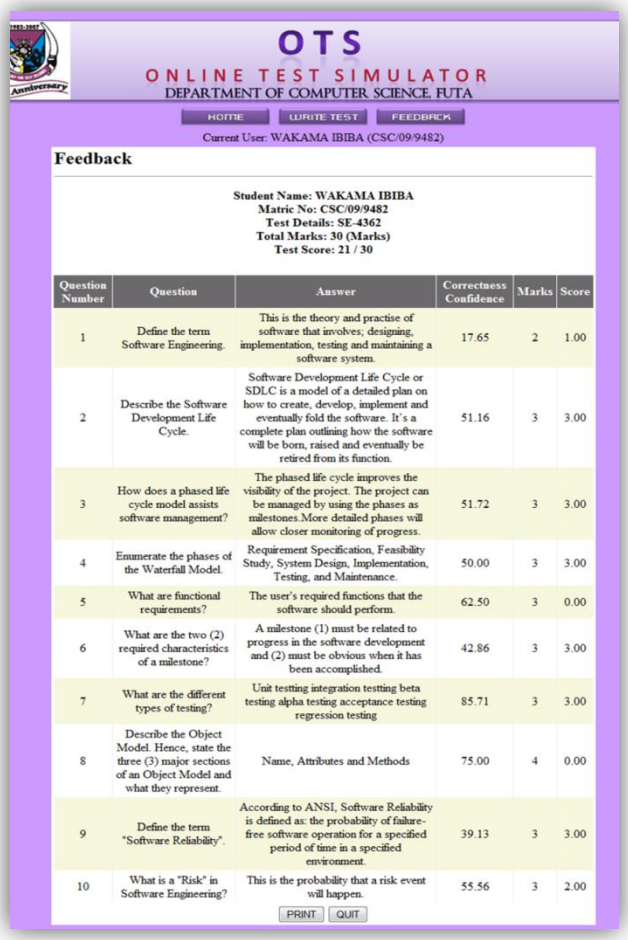


Fig. 7: Test in Progress

Fig. 8: Test Questions and Marks for SE-4362



Question Number	Question	Answer	Correctness Confidence	Marks	Score
1	Define the term Software Engineering.	This is the theory and practise of software that involves; designing, implementation, testing and maintaining a software system.	17.65	2	1.00
2	Describe the Software Development Life Cycle.	Software Development Life Cycle or SDLC is a model of a detailed plan on how to create, develop, implement and eventually fold the software. It's a complete plan outlining how the software will be born, raised and eventually be retired from its function.	51.16	3	3.00
3	How does a phased life cycle model assists software management?	The phased life cycle improves the visibility of the project. The project can be managed by using the phases as milestones.More detailed phases will allow closer monitoring of progress.	51.72	3	3.00
4	Enumerate the phases of the Waterfall Model.	Requirement Specification, Feasibility Study, System Design, Implementation, Testing, and Maintenance.	50.00	3	3.00
5	What are functional requirements?	The user's required functions that the software should perform.	62.50	3	0.00
6	What are the two (2) required characteristics of a milestone?	A milestone (1) must be related to progress in the software development and (2) must be obvious when it has been accomplished.	42.86	3	3.00
7	What are the different types of testing?	Unit testting integration testting beta testing alpha testing acceptance testing regression testing	85.71	3	3.00
8	Describe the Object Model. Hence, state the three (3) major sections of an Object Model and what they represent.	Name, Attributes and Methods	75.00	4	0.00
9	Define the term "Software Reliability".	According to ANSI, Software Reliability is defined as: the probability of failure-free software operation for a specified period of time in a specified environment.	39.13	3	3.00
10	What is a "Risk" in Software Engineering?	This is the probability that a risk event will happen.	55.56	3	2.00

Fig. 9: Test Feedback

8. ES TESTING AND PERFORMANCE EVALUATION

When proposing the adoption of an AES system over an existing one or a manual assessment system, it is often necessary to have a standard 'test set' and evaluation metrics to evaluate the new AES system and its predecessors (if any) in order to allow a reliable comparison of all these systems and to avoid the problem exposed by Whittingdon and Hunt in [22] who warned that, "before admiring the performance of a system, a reflection should be done about the metrics used by the authors". For instance, literature also revealed that "ETS" results could be overvalued since it only scores answers as correct or incorrect and thus, the agreement between the teacher and the system is easier to achieve. However, given that there is no standard test set or metric, this section applies the metrics described in [17] in the evaluation of the new ES, ES4ES. A comparative analysis of the result from both assessments was carried out with the metrics and discussed in section 8.2 below. Finally, the performance of the ES was compared with published results for other AES systems.

8.1 The Experiment

Designing and implementing an ES for Essay Scoring is apparently involving mere software development activities, however, the real value of a program cannot be established without testing it. The testing was performed in a real-world classroom scenario by administering a test of ten (10) questions to five students in Software Engineering – making a totality of 50 test samples. The students' responses were then assessed by the New ES and also by a Subject-Matter Expert (SME) in Software Engineering who was not aware of the ES's awarded scores. The scores obtained from both assessments are presented in Table 2 ('a' to 'e') below.

Table 2: SME's Scores (Subject Matter Expert's Score) and Expert System's Score (ES Score)

SME	ES
1.5	1
2	3
2	3
2.5	3
0	0
2.5	3
2	3
2	0
2.5	3
2	2

(a) Student 1

SME	ES
0	0
2	1
2	1
2.5	3
3	3
3	3
1	3
3	2
2.5	2
2	3

(b) Student 2

SME	ES
1	1
2.5	2
1.5	1
2.5	3
4	3
2	2
3	3
4	4
2.5	2
2	2

(c) Student 3

SME	ES
1.5	1
2.5	2
0	2
0	0
2	2
2.5	3
2	3
4	4
1.5	2
2	2

(d) Student 4

SME	ES
1	1
2.5	2
3	3
2	2
1.5	2
3	2
3	2
3.5	4
2	2
1.5	1

(e) Student 5

8.2 Metrics

Several metrics have been proposed, adopted and/or used for the evaluation of electronic essay assessors. Some notable metrics are: Measure of Exact Agreement, Adjacent Agreement, the Pearson Correlation, Spearman or nonparametric Correlation, Mean and Standard Deviations, Kappa Measure and F-Score. However, since the performance values of most AES system studied in this work is available in Pearson Correlation Coefficient, we have adopted it in measuring the performance of the newly developed expert system in comparative respect to the existing Electronic Essay Assessors.

Pearson Correlation or Inter-rater Reliability: It measures the standard correlation, that is, how much the teachers' scores or true scores (X) are related with the systems' scores (Y). It is calculated by applying Eq. (2) below. It is suitable whenever answers are evaluated with a numerical score. Sometimes the

true scores are the result of the average consensus of several teachers.

$$Correlation\ (X,Y) = r = \frac{Covariance(X,Y)}{StandardDev(X) \times StandardDev(Y)} \quad \ldots (2)$$

8.3 Performance Computation

The function CORREL(array1, array2) on Microsoft Excel Spreadsheet (whose underlying formula is presented in Eq. 2) was used to compute the correlation between the two scores across the 50 test samples in Table 2. The correlation coefficient obtained from the computation is 0.71. A graphical description of this correlation is presented in Fig 10. The overlapping lines on the graph indicate the number of times the teacher's score was exactly the same as the system's score.

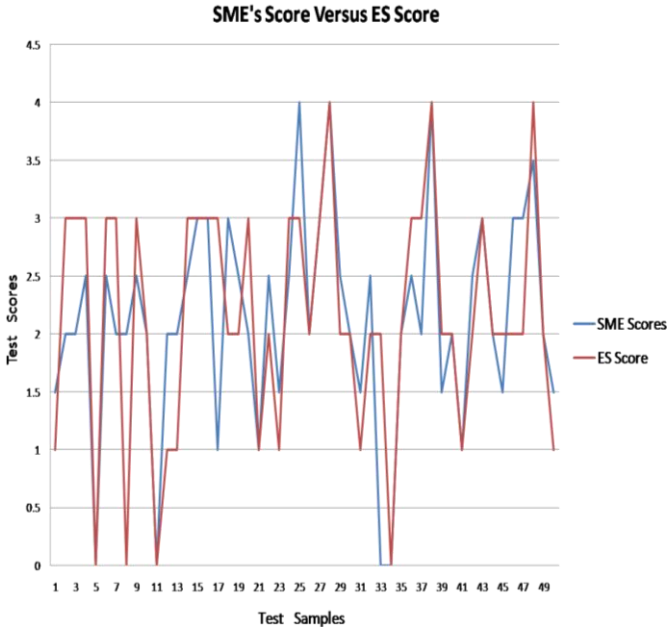


Fig. 10: Correlation between SME's Score and ES's Score Using 50 Test Samples

8.4 Comparative Analysis with Existing Systems

In this section, we present a comparative analysis of the newly developed ES with respect to existing AES systems based on their published correlation coefficients. The table below, obtained from [17], shows 22 different AES Systems, the techniques adopted in the development of these systems, the evaluation and the Language processed by the systems. Included therein and highlighted is our newly developed ES (ES4ES). The computed Correlation for ES4ES is in the same range as (or close to) that of AEA and Jess while IEMS and PEG are a little above it.

Table 3: Overview of the techniques, evaluations and languages of the reviewed AES systems. Possible metrics are: Corr (correlation), Agr (Agreement), EAgr (Exact Agreement), CAcc (Classification accuracy), F-S (f-Score), and, – for not available. In situations when the authors have presented several values for the results, the mean value has been taken. Data Source: Perez (2007) [17].

System	Techniques	Evaluation	Languages
AEA	LSA, PLSA or LDA	Corr: 0.75	Finnish
Apex Assessor	LSA	Corr: 0.59	French
ATM	Pattern Matching	-	English
Automark	Information Extraction	Corr: 0.95	English
Auto-Marking	NLP and pattern matching	EAgr: 0.88	English
BETSY	Bayesian Networks	CAcc: 0.77	English
CarmelTC	ML and Bayesian Networks	F-S: 0.85	English
C-rater	NLP	Agr: 0.83	English
EGAL	NLP and Statistics	-	English
E-Rater	NLP and VSM	Agr: 0.97	English
ES4ES	**Information Extraction**	**Corr. 0.71**	**English**
IEA	LSA	Agr: 0.85	English
IEMS	Pattern Matching	Corr: 0.80	English
IntelliMetric	CogniSearch and Quantum	Agr:0.98	English
Jess	Pattern Matching	Corr: 0.71	Japanese
Larkey's system	TCT	EAgr: 0.55	English
MarkIt	NLP, Pattern Matching & Statistics	Corr: 0.75	English
MRW	Logical Inference	-	German
PEG	Linguistic Features	Corr: 0.87	English
PS-ME	NLP	-	English
RMT	LSA	-	English
SAGrader	QTools	-	English
SEAR	Pattern Matching	Corr: 0.45	English

9. LIMITATIONS

This research is limited by few hitches. Similar to Apex and IEA, ES4ES is not suitable to assess free-text answers where the word order is important. It is also more effective in applications to short text answers rather than bulky texts. In addition, there could be certain overhead, predictably posed by the inference process. However, none of these flaws is a challenge on the assessment credibility of the new system. ES4ES will certainly serve the purpose for which it was designed.

10. CONCLUSIONS AND FUTURE WORK

It is interesting to mention that AES systems have prospered and have not been limited to English texts and academic purposes. In fact, the level of performance achieved by some of these systems has made possible their use as commercial applications. However, most common problems encountered in the research on automated essay grading is the absence both of a good standard to calibrate human marks and of a clear set of rules for specifying teachers' texts. A first conclusion is that in order to really compare the performance of the systems

some sort of unified measure should be defined. Furthermore, the lack of standard data collection is identified. Both these problems represent interesting issues for further research in this field.

11. ACKNOWLEDGEMENT

We sincerely appreciate the effort of Prof. O. S. Adewale, a Professor in the Department of Computer Science, FUTA for playing the role of the Subject Matter Expert (SME) in the field of Software Engineering and taking his time to assess the free-text responses of students in an open-ended test during the Software Testing process of the new Expert System.

12. REFERENCES

[1] Abney S. 1996 . "*Part-of-speech tagging and partial parsing*". Dordrecht: Kluwer. Ade-Ibijola et. al.

[2] Akinyokun O. C. 2002 , "*Neuro-Fuzzy Expert System for Evaluation of Human Resource Performance*". First Bank Endowment Fund Lecture delivered at the Federal University of Technology, Akure.

[3] Attali, Y., & Powers D. 2008 , "*Effect of immediate feedback and revision on psychometric properties of open-ended GRE Subject Test items*" GRE Board Research Rep. No GRE-04-05). Princeton, NJ: ETS.

[4] Bethune .D, Kelly .T, and Liversidge .T. 2007 , "*SCHOLAR Study Guide SQA Higher Information Systems, Unit 3a: Expert Systems*. www.torry.aberdeen.sch.uk/departs/comp/hinfosys3a.pdf

[5] Bereiter, C. 2003 , "*Automated essay scoring: a cross disciplinary approach*". In Mark D. Shermis and Jill C. Burstein Eds. , Foreword (pp. vii- ix), Lawrence Erlbaum Associates: Mahwah, NJ.

[6] Bloom, B. S. 1956 , "*Taxonomy of educational objectives: The classification of educational goals*". Handbook I, Cognitive domain. New York, Toronto: Longmans, Green.

[7] Burstein J., Leacock C., and Swartz R. 2001 , "*Automated evaluation of essays and short answers*. In Proceedings of the 5th International CAA Conference.

[8] Chakraborty R. C. 2010 , *Expert Systems: AI Course Lecture 35-36 Notes*. http://www.myreaders.info/07_Expert_Systems.pdf

[9] Kiong S. W., Latif A. B. Rahman A, Mohd F. Z. and Azwan A. A., 2005 . published on www.generation5.org/content/2005/expert_system.asp

[10] Marcu D. 2000 , "*The theory and practice of discourse parsing and summarization*" The MIT Press.

[11] Mcgrath P. 2003 , "*Assessing Students: Computer Simulation vs MCQs*". In Proceedings of the 7thComputer Assisted Assessment Conference, 2003.

[12] Mitchell T., Russell T., Broomhead P., and Aldridge N. 2002 , "*Towards robust computerised marking of free-text responses*". In Proceedings of the 6th Computer Assisted Assessment Conference, 2002.

[13] Mitchell T., Aldridge N., Williamson W., and Broomhead P. 2003 , "*Computer based testing of medial knowledge*. In Proceedings of the 7th Computer Assisted Assessment Conference, 2003.

[14] Moser J. R. 2009 , "*The Electronic Assessor: Design and Prototype of an Automated Free-Text Assessment System*". A Master's Thesis at Institute for Information Systems and Computer Media (IICM), Graz University of Technology, A-8010 Graz, Austria.

[15] Page, E. B. 2003 , "*Project Essay Grade: PEG*. In M. D. Shermis & J. Burstein (Eds.). *Automated essay scoring: A cross-disciplinary perspective.* Mahwah, NJ: Lawrence Erlbaum Associates

[16] [16]. Palmer K. and Richardson P. 2003 , *"On-line assessment and free-response input - a pedagogic and technical model for squaring the circle*". In Proceedings of the 7th Computer Assisted Assessment Conference, 2003.

[17] Perez D. 2007 , "*Adaptive Computer Assisted Assessment of Free-text students' answers: an approach to automatically generate students' conceptual models*. PhD Thesis, Computer Science Department, Universidad Aut´onoma de Madrid, 2007.

[18] Perez D., Gliozzo A., Strapparava C., Alfonseca E., Rodr´ıguez P. and Magnini B. 2005 , "*Automatic Assessment of Students free-text Answers underpinned by the Combination of a BLEU-inspired algorithm and Latent Semantic Analysis*". Published by the American Association for Artificial Intelligence (AAAI) Press. Presented at the 18th International Conference of the Florida Artificial Intelligence Society (FLAIRS), U.S.A., May 2005.

[19] Rudner, L.M. & Liang, T. 2002 , "*Automated essay scoring using Bayes' Theorem"*. The Journal of Technology, Learning and Assessment, 1(2), 3-21.

[20] Valenti S., Neri F. and Cucchiarelli A. 2003 , "*An Overview of Current Research on Automated Essay Grading*". Journal of Information Technology Education, 2:319–330, 2003.

[21] Williams R. and Dreher H. 2004 , "*Automatically Grading Essays with Markit*. In Proceedings of Informing Science 2004 Conference, Rockhampton, Queensland, Australia.

[22] Whittingdon D. and Hunt H. 1999 , "*Approaches to the computerised assessment of free-text responses*", In Proceedings of the 3rd International Computer Assisted Assessment Conference, 1999.

[23] Wiki 2011, "The Rete Algorithm", Wikipedia Encyclopedia, www.en.wikipedia.org/wiki/rete_algorithm