Moving towards a Fully Automatic Knowledge Assessment Tool

Christian Gütl
Institute for Information Systems and New Media, Graz University of Technology, Graz, Austria

***Abstract*—Information about a student's level or state of knowledge is a key aspect for efficient, personalized learning activities. E-learning systems gain such information in two ways: directly by examining users' self-assessment and administering predefined tests and indirectly by making inferences on observed user behaviors. However, most of the current solution approaches either demand excessive manpower or lack required reliability. To overcome these problems, we have developed the e-Examiner, an assessment tool that supports the assessment process by creating automatically test items, assessing students' answers and providing feedback. In this paper, we firstly give an overview about a variety of computer-assisted and computer-based assessment systems and methods that support formative assessment activities. Secondly, we introduce the overall concept and architecture of the e-Examiner. Thirdly, we outline implementation details and evaluation results of our prototype implementation. Our solution approach is based on the set of statistical similarity measures defined by the ROUGE toolset for automatic summary evaluation.**

This paper is an extended version of the IMCL 2007 paper.

***Index Terms*—automatic knowledge assessment, formative assessment feedback, computer-based assessment system, ROUGE toolset**

I. INTRODUCTION

Our modern life at the beginning of the 21st century is strongly influenced by effects such as rapidly changing and developing information, technology-enhanced communication and information access, and new forms of production and services in a globalized world. This situation requires individuals to adapt their skills and competencies. Consequently, educational objectives and societal expectations have changed significantly in recent years. According to [1], modern learning environments must consider learning community aspects as well as learner-centered, knowledge-centered and assessment-centered aspects.

By focusing on the assessment, this concept can be further distinguished in (1) summative assessment, performed at the end of a set of learning activities, and (2) formative assessment, which is intended to give continuous feedback to students and teachers and to enable them to revise teaching and learning activities. The latter mentioned formative assessment gives information about the current state of knowledge and/or the degree of knowledge acquisition.

In technology-enhanced learning, formative assessment results can be used by learning management systems to adapt learning activities towards the users [2][3]. Some existing systems try to gain such information by inferring observed user behaviors. An example for a simple and widespread approach in this context is the analysis of the duration and frequency of learning content "consumed" by users. However, frequent and long-lasting visits to learning content do not imply any cognitive process by the user. More sophisticated approaches exploit data from sensors located on the client side. The author in [4] outlines an approach for the extraction of user behavior patterns, such as browsing, searching, and viewing from mouse and keyboard events. Within the AdeLE project, (see for example in [5]) an eye-tracking system was used to identify different gaze patterns for learning content to identify skimming, reading, and memorizing activities. However, all of the above-mentioned approaches do not directly measure the actual knowledge acquisition and are therefore error prone.

In order to measure knowledge acquisition or the knowledge state directly, another approach relies on users' self-assessment. This approach, however, lacks also required reliability. Another commonly used approach is the application of questionnaires, composed by limited choice questions, completion tests, and open-ended questions [6]. However, this approach demands excessive effort and results in high costs to prepare tests for appropriate application in computer-supported testing systems. Additionally, tests must be updated for any changes in the learning content or in the didactical or pedagogic objectives. Moreover, personalized learning content in adaptive e-learning systems also requires a personalized assessment procedure by using a variety of questions which considerably increases the effort for formative assessment procedures. This situation has motivated us to develop the e-Examiner, a computer-based assessment system.

The remainder of the paper is organized as follows: Chapter 2 gives an overview of computer-assisted and computer-based assessment systems, Chapter 3 discusses application scenarios for a modern assessment tool, Chapter 4 outlines the solution approach and an overall architecture of the e-Examiner system, Chapter 5 focuses on the prototype implementation of a ROUGE-based short free-text answer assessment, and Chapter 6 emphasizes conclusions and future work.

II. RELATED WORK

Formative assessment is an important component of modern teaching and learning processes in face-to-face courses as well as in e-learning environments; it provides valuable feedback to teachers and students which allows for the revision and adaptation of teaching and learning activities. Furthermore, assessment activities and results can also be utilized for building and strengthening

metacognitive skills. [1] However, continuous and frequent assessment in learning processes may cause excessive efforts and costs. Therefore, computer-assisted assessment systems (CaAS) and computer-based assessment systems (CbAS) have become of increasing interest over the years. Such systems may support parts or the entire chain of the assessment lifecycle. This lifecycle includes authoring and management of assessment items, compilation of specific tests, performance of assessments, and compilation and management results. Additionally, emerging interest in the sharing and re-use of assessment items or compiled assessment tests and the exchangeability of assessment outcomes has resulted in the IMS Question & Test Interoperability Specification (IMS QTI). [7] [8] [9]

In particular, the research and development of automated assessment tools has caused growing interest over the last years. This trend was induced by increase in teachers' and tutors' workloads, which was caused by intensified formative assessment activities and the need for immediate assessment feedback in e-learning-based teaching and training setups. Interesting approaches can be found for various application domains, such as language training [10], mathematics [11], computer science [12], and in numeric disciplines [13].

The authors in [14] describe an interesting approach for automatic, or at least computer-aided, generation of multiple-choice tests from digital learning content. In general, limited choice questions and completion tests are very popular because of their simple methods for automatically assessing students' answers. These types of tests, however, can not sufficiently assess more abstract educational objectives. [6] [15] To attempt to overcome this problem, other types of assessment are used such as short free-text answers and essays. Consequently, methods and procedures for automated assessment are much more challenging.

By focusing on essay assessment, the complex evaluation process includes both content aspects and style characteristics. In order to gain these aspects automatically, measurements are taken either by indirect characteristics (also termed as "proxes") or by actual dimensions. [9] Automated essay grading (AEG) has been an active research topic since the 1960s and some interesting prototypes and commercial tools have emerged within the last 40 years. [16] The Project Essay Grad (PEG), which began in the early days of AEG research, addresses style aspects by utilizing the concept of proxes and statistical methods (linear regression). The Intelligent Essay Assessor (IEA) focuses on content aspects and applies the latent semantic analysis (LSA) technique. The Bayesian Essay Test Scoring sYstem (BETSY) includes content as well as style aspects in order to separate essays automatically in a four-point nominal scale using Bayesian Text classification and statistical approaches. [9][17]

By focusing further on content aspects of the essay grading process, the author in [18] investigates the applicability for grading on a four-point scale by applying two different text categorization techniques combined with text-complexity features. In the first approach, a set of marked reference essays is used to train Binary Bayesian Independence Classifiers for distinguishing essay candidates on the four-point scale. In the second approach, for each level of the four-point scale a K-Nearest-Neighbor Classifier built on a probabilistic retrieval system is applied. The grading is performed by calculating the mean value given by the grading values of the K-nearest reference essays. Despite the good results reported in this paper, the major drawback of such statistical approaches is that they rely on bag of words but do not include functional relationships between them.

In contrast to the statistic approaches stated before, the authors in [19] describe the CarmelTC approach, which focuses on "correct answer aspects" by using hybrid text classification techniques. It is based on a rule-learning text classification method, and it combines results from syntactic functional analyses of text with bag of words classification. An additional interesting concept, which goes behind statistical analysis of essay answers, is discussed in [20]. The authors borrow the Lexical Conceptual Structure (LCS) approach from the machine translation domain for describing the content in a language-independent internal representation (interlingua) and discuss its application for content-based essay assessment.

Unlike the holistic assessment of content and style aspects for essays, the interest in Short Free-text Answer (SFTA) assessment is focused solely on content aspects. Typically, SFTA are written student responses from specific learning and testing activities, such as end-of-the-chapter review questions, classroom tests, and written homework assignments. [21] As opposed to open ended questions, SFTA are results from factual science questions which can be assessed by objective criteria. [22] In order to perform the challenging assessment task of SFTA, proposed and implemented solutions cover approaches from statistical methods to methods using artificial languages to natural language technologies.

The authors in [23] discuss the applicability of two simple statistic characteristics: (1) the recall for estimating the coverage of test answers compared to the reference answer, and (2) the precision for measuring the conciseness of test answers. For their first experiments, the authors use only the recall measure to assess answers which results in an unexpectedly good performance. However, the recall measure does not consider deeper text structure and meaning. A step towards language understanding and more reliable assessment of short answers could be the usage of a controlled natural language, yet as a drawback, students are restricted to a limited vocabulary and specific grammar for their answers. [24][25]

By focusing on the assessment of natural language answers, the author in [26] describes an interesting approach that uses a semantic network to represent candidate answers, assesses the answers against a model answer, identifies wrong and incomplete answers, and provides feedback in natural language. Another tool, the C-rater, applies a variety of natural language processing methods, such as context-sensitive spelling correction, predicate argument structure and pronominal reference processing, morphological analysis, and synonym expansion. This results in a canonical representation of candidate answers and the reference answer. Finally, a rule-based algorithm processes the assessment derived from the preprocess candidate and reference answers. [21] WebLAS, see [10], and the assessment engine from Intelligent Assessment Technologies, see [27], also apply text pre-processing and tagging and subsequently assess

candidate answers against a number of marked scheme templates.

The authors in [22] report on experiments applying further interesting approaches. Firstly, they adopt an information extraction approach using text pre-processing and tagging and apply a set of both written and hand-adjusted patterns to the tagged and chunked text for the assessment process. Secondly, they utilize three different machine learning methods, inductive logic programming, decision tree learning, and Bayesian learning for the grading process. A possible drawback of these approaches is that a representative set of training data is needed; i.e. a representative number of manually graded answers must be available, which may cause an undesirably high effort for teachers in the initial phase.

III. APPLICATION SCENARIOS FOR A MODERN ASSESSMENT TOOL

In order to gain requirements for our automatic assessment tool e-Examiner, the aim of this section is to outline application scenarios that are based on our former experiences in the field of technology-enhanced learning and on our findings in literature review (see also previous section).

Miriam is a teacher who lectures in history at an undergraduate school. Her didactic goals include the understanding and application of factual knowledge. To gain information about her students' knowledge states, she decides to conduct continuous assessment on both didactic goals supported by the e-Examiner. The tool supports Miriam by automatically creating questions and assessing answers about factual knowledge. Moreover, the e-Examiner supports Miriam in her goal of knowledge understanding and application by assessing essays about specific topics on history.

Alex is a junior lecturer in a department of cognitive science. His active research is in the area of adaptive e-learning. He intends to use an adaptive e-learning system to personalize learning content and activities according to students' knowledge level. Alex's didactic objectives are not only limited to knowledge acquisition, as he is also interested in seeing his students strengthen their metacognitve competences; i.e. that the students can estimate their own knowledge level about a topic and therefore improve self-controlled learning. He decides to integrate the e-Examiner in order to automatically prepare representative questions within the personalized learning sessions. Additionally, the assessment tool requests students' self-assessment about the appropriateness of the given answers. After completing the assessment forms, e-Examiner provides immediate feedback about the correctness of the answers and of the metacognitve state within this domain area. Additionally, the results are used to feed the user modeling system for further personalization activities. Finally, the e-Examiner tool processes anonymized statistics for Alex and helps him prepare specific course content for the face-to-face lectures.

Mona is an undergraduate art history student who must complete the course "Italian Arts in the 16th Century". She has already enrolled the online course weeks before, but since first visiting the online content none of the learning objectives have been completed yet. The interactive digital course assistant has noticed this, and according to an online survey initiated by the system, it has identified low motivation and low interest in the subject. The system suggests an adventure-based game style, where course content-based questions and problems must be solved by reading parts of the learning content and searching for answers on the internet. The e-Examiner tool integrated within this learning environment handles automatic creation of questions and the assessment of Mona's answers. The questions are embedded in the flow of the game story and must be solved in order to complete the game. At the end of the game, Mona has played more than 40 hours, and 25 % of the course has already been completed.

Kevin is preparing himself for a new job in his business unit. He likes to learn on his own, yet he needs continuous feedback about his learning progress. Because of the company-specific and multidisciplinary learning content, appropriate assessment for independent checking is not available. E-examiner supports Kevin by automatically preparing factual knowledge questions on-the-fly based on the selected content. The e-Examiner also assesses Kevin's answers and provides him feedback for his further learning process.

IV. SOLUTION APPROACH AND OVERALL ARCHITECTURE

Unlike most other existing computer-supported and computer-based assessment tools, we focus on an open and flexible approach. Based on the application scenarios outlined in the previous section, the main requirements for our modern assessment tool include:

- Flexible design to be used as a stand-alone service or to be easily integrated in existing e-learning or other systems.
- Standard-conform information exchange and interfaces.
- Assessment support of different didactic objectives by supporting automatic question generation, and assessment by various assessment types, such as multiple-choice tests, short free text answers, and essays.
- Information delivery for updating user models for the purpose of adapting course activities and learning content.
- Providing feedback about the state of user-knowledge and metacognitive skills to learners and teachers.
- Security and Privacy.

The requirements stated so far result in the architectural overview as depicted in Figure 1, which is described briefly as follows. The Assessment Test Management unit is the core module of the e-Examiner. It handles the entire

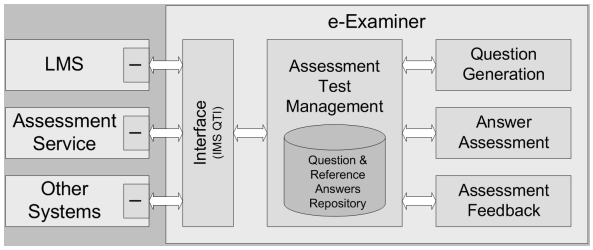

Figure 1. Overall Architecture

lifecycle of the assessment procedures, from question generation to the storage of assessment items (question and reference answers) to the assessment to feedback provision. Furthermore, it controls the communication flow with the other modules. The Question Generation module automatically identifies important concepts from a specified learning content, and based on this identification, it creates questions and reference answers. To illustrate this, consider short free text answer exercises. The module creates simple questions such as 'explain concept X' or 'describe concept X'. Additionally, the reference answer is also extracted from the content for further processing. To give another example, multiple choice exercises can be created by using identified concepts and the surrounding content to ask for the given concept or to select the text as one of the possible answers. Question Generation module can either assist a teacher in the creation of assessment items, or it can be utilized to create the items unsupervised on-the-fly. The Answer Assessment module assesses students' answers according to the type of exercise (multiple choices, short free-text answers or essays). For each type of assessment, the corresponding tool for automated assessment needs to be instantiated. Each of the tools provides assessment results in a standardized format to the Assessment Feedback tool which prepares helpful feedback for students and teachers. Depending on the privacy policy, accessible information for the teacher can be restricted or made anonymous. The Interface module handles the information flow between the e-Assessor system and external systems (learning management systems (LMS), standalone assessment services or other systems) by applying the IMS Question & Test Interoperability Specification [8].

V. ROUGE-BASED SHORT FREE-TEXT ANSWER ASSESSMENT

Moving towards our flexible and open assessment system described in the previous chapter, we have developed a first prototype system. The aim was to provide an assessment application that could enable experiments and allow to gain first experiences for further research and development cycles.

Our prototype was designed to run as a Web-based stand-alone assessment service. It focuses on (1) the management of assessment items, (2) the compilation and performance of student tests supporting short free-text answers, (3) the automated assessment of these answers, and (4) the immediate result and feedback presentation to students and teachers. For automated assessment of students' answers, we have decided to apply a hybrid approach. It is built on a natural language pre-processing chain and on ROUGE characteristics, originated for automated evaluation of text summaries [28]. The remainder of this chapter is partly based on [25].

A. Prototype Architecture and Implementation

Figure 2 depicts the architecture of our Web-based prototype implementation. Teachers and students can access the system and perform their role-specific assessment tasks by a Web Client (see also the following subsection). The Web gateway to the clients is provided by the Apache Tomcat [29], an open source JSP and Servlet Container. Additionally, the Tomcat server together with Apache Struts [30], an open source framework for building Servlet/JSP-based Web applications supporting the Model-View-Controller (MVC) design paradigm, hosts our Web application.

The Control and View component, depicted on the upper left side in the figure, handles the HTTP requests from the client side, delegates information for further processing to the business logic of the application, and compiles and presents results retrieved from the business logic.

The model in the MVC design paradigm represents the business logic. In our application, it manages user data and roles by the User Management component. The component also handles assessment items, entire tests compiled by assessment items, and controls the automated assessment by the Test Management component.

For the purpose of persistent storage of data and states, a Data Storage and Retrieval component based on the open source framework Hibernate [31] was built. The underlying data are managed by the free available database system MySQL [32].

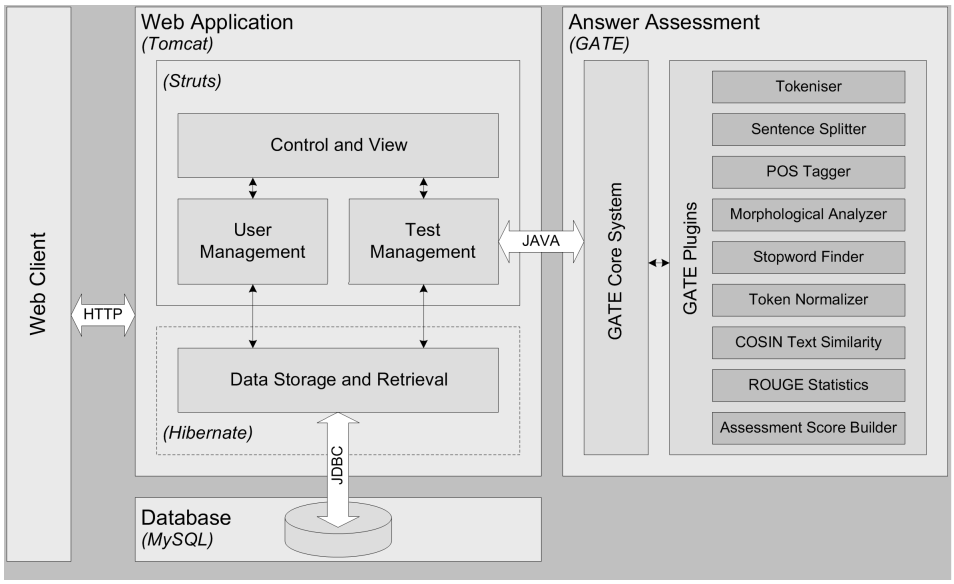

Figure 2. E-Examiner's Prototype Architecture

The automated assessment of short free-text answers is performed by the Answer Assessment component. This component is built on GATE, an open source framework for *"developing and deploying software components that process human language"* [33]. One of the strengths of the GATE system is that it easily enables the import of text-processing functions by plugin, and that it defines processing chains based on these functions. It is important to note that a large number of pre-existing plugins is available. We have applied some of the ANNIE (A Nearly-New IE system) plugins in order to build the natural language pre-processing chain. This pre-processing chain includes the Tokeniser, the Sentences Splitter, the POS Tagger, the Morphological Analyzer, the Stopword Finder, and the Token Normalizer.

For the automated assessment task, we have implemented three plugins: (1) the COSIN Text Similarity plugin estimates the similarity between candidate answers and one or more reference answers based on the vector space model [34]. (2) The ROUGE Statistics plugin computes a variety of statistical characteristics on the level of words as described in the following subsection. (3) The Assessment Score Builder plugin determines the final score by a linear combination of characteristics delivered by the two aforementioned plugins.

B. ROUGE Characteristics for Short Free-Text Answer Assessment

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) defines a set of statistical measures to automatically determine the quality of a summary by comparing it with reference summaries. The original intention was to reduce human efforts in the process of evaluating computer-generated summaries. [28]

We were inspired by the ROUGE idea just described to study its applicability for short free-text answer assessment. From our point of view, this application scenario is somewhat similar to the original one: students' free-text answers related to learning content can be seen as summaries of this learning content, and they are compared with reference answers or reference summaries. The ROUGE metrics applicable for comparing candidate with reference answers are briefly described in the remainder of this subsection. A detailed description and discussion can be found in [28] and [35].

- ROUGE-N defines word n-gram co-occurrence statistics between candidate and reference answers where n stands for the length or number of words to be applied for the co-occurrence statistic. To give an example, let us consider ROUGE-2 (N=2), see also Figure 3. In this example, the number of common occurrences of word bigrams in candidate and reference answers is used to compute the ROUGE-2 measure.
- ROUGE-L compares the longest common subsequence (LCS) of words between candidate and reference answers. It is important to emphasize that this similarity measure does not require consecutive matches as by longest common sequence measures, but it focuses on in-sequence matches; i.e. this procedure searches for common words in a sequence but also allows differences between common word occurrences.
- ROUGE-W stands for weighted longest common subsequence. Unlike ROUGE-L, it differentiates several spatial relations in sequences and favors LCS with consecutive matches.
- ROUGE-S describes the skip-bigram co-occurrence statistic. Skip-bigrams are any pairs of words in sentence order which allow arbitrary gaps between them. This metric measures the overlap ratio of skip-bigrams between candidate and reference answers.
- ROUGE-SU is an extension of the ROUGE-S measure. The ROUGE-S measure described above does not give any credit candidate answers which lack a pair of words in common. To overcome this, ROUGE-SU extends ROUGE-S by adding the count of unigram to this statistical measure.

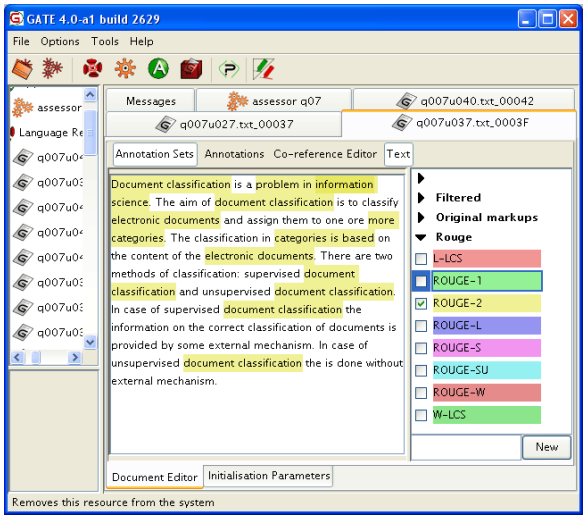

Figure 3. Example of GATE's graphical output for ROUGE-2 characteristic (matching word bigrams) applied on a candidate answer

C. The Assessment Prototype from the User Perspective

The current available implementation of the e-Examiner supports teachers and students over the life-cycle of the assessment process of short free-text answers. The aim of this section is to briefly describe functions for both user groups. User management and login process are the same for both of them and include editing user information and password.

From the teachers' point of view, the first prototype implementation supports them from managing assessment items to performing assessments, to inspecting student results. The following functions can be used by teachers:

- Creation and editing of assessment items include the definition of the question and the assignment of the reference answer.
- Compilation of a set of assessment items for creating student tests.
- Inspection of test results which enables two different viewpoints. Firstly, teachers can review test results for each of the students. The teachers get an overview by providing the grading result and the student's self-assessment in table form for each question (see also the right screen shot in Figure 4). By clicking on a hyperlink, it is also possible to request details for each of the answers; i.e. teachers can review the student answers and correct the final grading. Secondly, teachers can also review the test results for each of the test items and obtain an overview for the performance of all students.

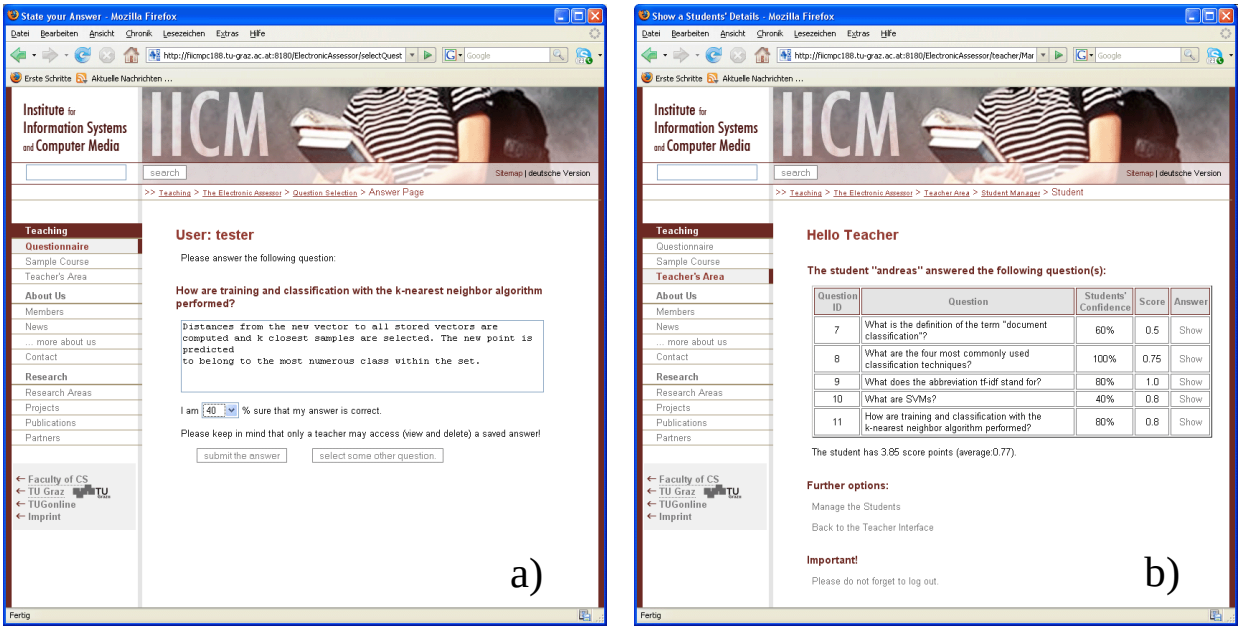

Figure 4. a) Screen shots of student interface for answering short free-text answers and providing self-assessment about given answers
b) Screen shots of o teacher interface for inspecting student's test results

From the students' point of view, the e-Examiner supports them in their completing the examinations and subsequent reviewing of results. The following functions are available for students:

- Completion of student test enables students to obtain an overview about the compilation of test items. Students can freely choose the order in which they wish to answer the questions. For each of the short free-text answer assessment items, students can type in the answer in a text box and can select the percentage of the estimated correctness of their answer (see also the left screen shot in Figure 4).
- Inspection of test results is analogous to the teachers' view except that students will only get their own results in detail; overall results are made anonymous.

D. First Experiment and Evaluation Results

The aim of this section is to report our first findings. For our experiments we have collected a dataset for short free-text answer assessment from two different sources: The Institute for Information Systems and New Media (IICM), Graz University of Technology, Austria, and from Faculty of Engineering, Al-Quds University, Jerusalem (see also TABLE I). The IICM dataset consists of 5 questions related to computer science topics - one reference answer for each of them and 23 sets of student answers. Out of the Al-Quds dataset, we have selected three questions also related to a computer science topic and taken 23 answers for each of these questions. Together, the IICM data and the selected Al-Quds data, are compiled to a dataset which consists of 8 questions, a reference answer for each of them, and 23 sets of answers. All free-text answers in the dataset are manually assessed by a domain expert according to the reference answers and marked by a number between zero (inappropriate) and ten (very good). One reference answer and examples of candidate answers grated by a domain expert are given in TABLE II.

The experiment setup and first results are briefly described in the remainder of this section. Eleven of the 23 sets of answers (88 answers) constitute the set of

TABLE I.
SET OF QUESTIONS FOR EXPERIMENT

	No.	Question Text
IICM dataset	Q1	What is the definition of the term "document classification"
	Q2	What are the four most commonly used classification techniques?
	Q3	What does the abbreviation tf-idf stand for?
	Q4	What are SVMs?
	Q5	How are training and classification with the k-nearest neighbor algorithm performed?
Al-Quds dataset	Q6	What is Artificial Intelligence?
	Q7	Define Computer Virus?
	Q8	What is Multitasking?

TABLE II.
REFERENCE ANSWER AND TWO CANDIDATE ANSWERS GRATED BY A DOMAIN EXPERT FOR QUESTION Q7 "DEFINE COMPUTER VIRUS".

Type	Answer Text	Grading
Reference Answer	A computer program with the ability to modify other programs usually spreads and damages computer systems.	(10.0)
Candidate Answer 1	A computer function often cause damage while spreading to other systems	6.4
Candidate Answer 2	A dangerous computer program with the characteristic feature of being frequently generate generating copies of itself, and change and spoiling software application.	7.2

TABLE III.
OVERVIEW ABOUT EXPERIMENT SETUP AND RESULTS

Experiment Setup		Average Absolute Error			Correlation		
No.	Description of applied Similarity Measures	Training Dataset	Test Dataset	Total Dataset	Training Dataset	Test Dataset	Total Dataset
1	Number of Tokens and Precision Measure of ROUGE-1	1.47	1.51	1.49	0.80	0.80	0.80
2	Number of Tokens, COSIN Similarity and ROUGE Recall and Precision Measures (ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L, ROUGE-S, ROUGE-SU, ROUGE-W)	1.03	2.00	1.54	0.84	0.79	0.81
3	Number of Tokens, Recall of ROUGE-L, Recall and Precision of ROUGE-1	1.32	1.46	1.39	0.82	0,81	0,81

training data. The remaining 12 sets of answers (96 answers) comprise the set of test data. So far, we have focused on three experiment setups (see also TABLE III) which are composed of various linear combinations of selected similarity metrics. For each of them, the parameters for the linear combination are computed by applying linear regression on the results of the training data set.

The three experiment setups are:

- Setup No. 1 simply takes into account the number of tokens and the precision value of word unigrams (ROUGE-1-P).
- Setup No. 2 takes into account a great variety of similarity measures including COSIN similarity, recall and precision for word unigrams and bigrams (ROUGE-1-R, ROUGE-1-P, ROUGE-2-R, ROUGE-2-P) as well as ROUGE-L, ROUGE-S, ROUGE-SU and ROUGE-W.
- Setup No. 3 focuses on the number of tokens,

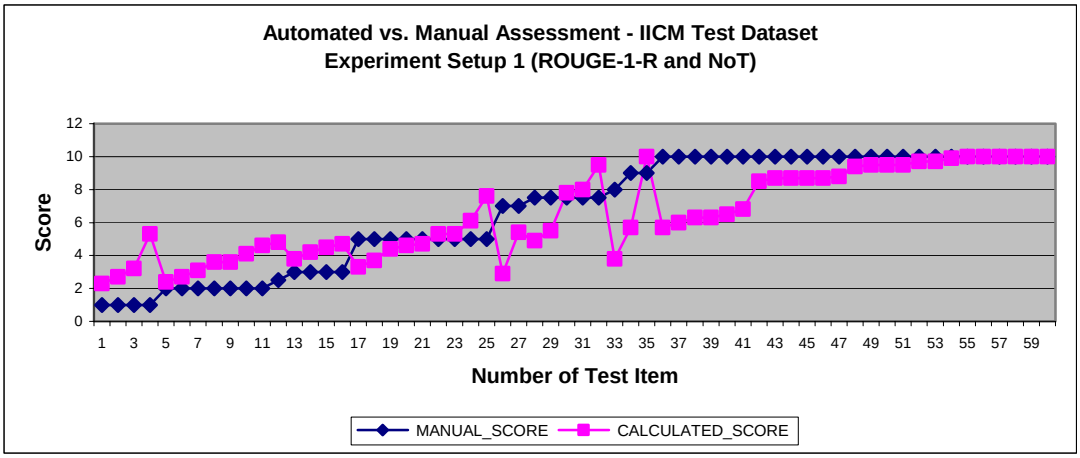

Figure 5. Result plot of experiment setup 1 (ROUGE-1-R and NoT) shows human vs. computer-based scores in ascending order of human scores for the IICM test dataset.

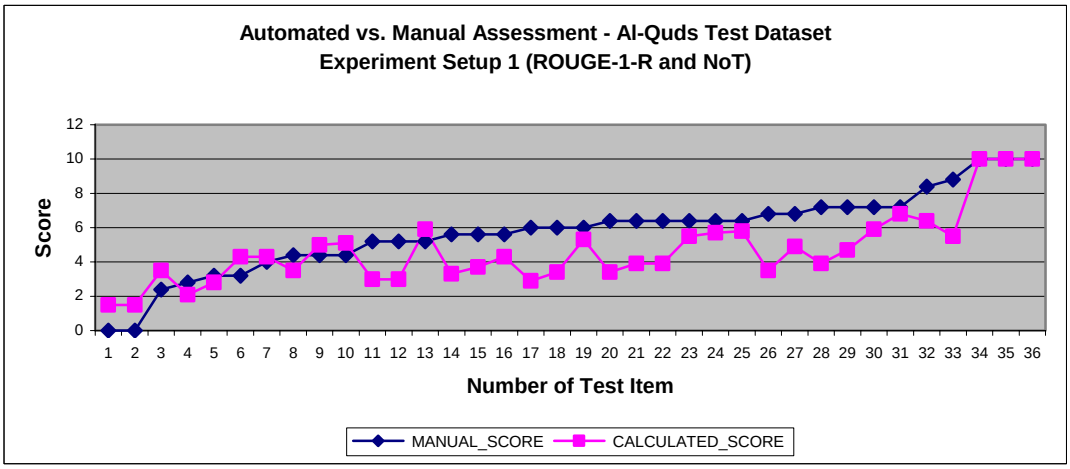

Figure 6. Result plot of experiment setup 1 (ROUGE-1-R and NoT) shows human vs. computer-based scores in ascending order of human scores for the Al-Quds test dataset.

recall and precision of word unigrams (ROUGE-1-R, ROUGE-1-P) as well as Recall of ROUGE-L.

For each of the experiments, the average number of the absolute error and the correlation is computed for the set of test data, training data and the entire set of data. A comprehensive view of the experiments and results is given in Table III. Despite the simple statistical approach, the results are surprisingly good. The correlation factor for the three given experiments calculated for the set of test data runs between 0.79 and 0.81, which is comparable with other systems and approaches. By focusing on the absolute average error for the set of test data, the very simple approach of setup No. 1 performs by a given value of 1.51 unexpectedly good. The enhanced approach, setup No. 3, also takes into account the precession value of word unigrams, and measures of the longest common subsequence. This setup improves the performance marginally. Unexpectedly bad performance compared to the simple approaches has been evident in the more complex setup No. 2 by a given value of 2.00. By taking into account the absolute error values of the training data and the entire data set for setup No. 2, the model seems to be over-trained for the given training data set. Detailed result plots for the IICM test dataset and the Al-Quds test dataset for the experiment setups No. 1 is given in Figure 5 and Figure 6. Of course, despite the surprisingly good results for such a simple statistical approach, there is a great deal of space for further improvements and experiments such as enlarging the data set, conducting further experiments with other combinations of similarity measures, and using more than one reference answer for the assessment process.

First experiences from the architectural point of view of our prototype are also promising. It is evident that the concept easily enables the integration with other systems. Furthermore, a standardized format for the exchangeability of assessment items has been positively evaluated. However, in order to increase the flexibility and the value of our system, standardized interfaces to provide business services and Web 2.0 access must be integrated into our system.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown a variety of computer-assisted and computer-based assessment systems and methods that can support formative assessment activities. These systems may reduce human workload in the assessment process, and they provide prompt feedback to students, which is of particular interest in the e-learning application domain.

Assessment of short free-text answers enables users to evaluate higher educational objectives or skills and competencies. The automated assessment in this context has become an increasingly active research topic. Various solution approaches, from statistic methods to natural language understanding solutions, are present in contemporary research. For the evaluation process, we have applied natural language pre-processing together with statistic ROUGE metrics, originally used for the automatic evaluation of computer-generated summaries. Despite the application of only statistic measures of text similarity, our first experiment results are promising. We believe that such a system can be used in an e-learning environment for providing automatic feedback to students and for updating user-profiling information.

Further experiments which apply a larger dataset and the application of more than one reference answer for each of the test items will be investigated. Furthermore, we also want to adapt and use our approach to evaluate short free-text answers in German language. Form the architectural viewpoint, we want to enable our system for Web 2.0 technologies and to integrate a subsystem for the automatic creation and assessment of multiple choice assessment items.

ACKNOWLEDGMENT

The author grateful acknowledges Josef Moser for contributing valuable input for this paper and for the prototype implementation as part of his master thesis. Furthermore, the author also gratefully acknowledges Labib Arafeh and Yousef Oriquat from Al-Quds University, Jerusalem, for providing a dataset of short free-text answers.

REFERENCES

[1] J.D. Bransford, A.L. Brown, and R.R. Cocking; (Eds.), "How People Learn: Brain, Mind, Experience, and School. Expanded Edition" Washington DC: National Academies Press, 2000.

[2] P. De Bra, G.-J. Houben, and H. Wu, "AHAM: a Dexter-based reference model for adaptive hypermedia", Proceedings of the tenth ACM Conference on Hypertext and Hypermedia, 1999, pp. 147-156.

[3] P. Brusilovsky, "Adaptive hypermedia", User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.), 11 (1/2), 2001,pp. 87-110.

[4] A. Rao, "Recognition of Conative and Affective Behavior in Web Learning using Digital Gestures", Proceedings of NAWeb 2004, New Brunswick, Canada, October 2004.

[5] V.M. García-Barrios, "Real-Time Learner Modeling: Using Gaze-Tracking in Distributed Adaptive E-Learning Environments", Proceedings of MIPRO, Opatija, Croatia, May 22-26, 2006.

[6] F. Mödritscher, and A. Sindler, "Quizzes are not enough to reach high-level learning objectives!", Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005 (ED-MEDIA 2005). Montreal, Canada, June 2005, pp. 3275–3278.

[7] S.Valenti, A.Cucchiarelli, and M. Panti, "Computer Based Assessment Systems Evaluation via the ISO9126 Quality Model", Journal of Information Technology Education, Volume 1, No. 3, 2002, pp. 157-176, and http://jite.org/documents/Vol1/v1n3p157-175.pdf

[8] IMS, "IMS Question and Test Interoperability Overview. Version 2.0 Final Specification.", unpublished, 2005, last retrieved February 18th 2007 from http://www.imsglobal.org/question/qti_v2p0/imsqti_oviewv2p0.html

[9] S. Valenti, F. Neri, and A. Cucchiarelli, "An Overview of Current Research on Automated Essay Grading", Journal of Information Technology Education Volume 2, 2003, pp. 319-330, and http://jite.org/documents/Vol2/v2p319-330-30.pdf

[10] L.F. Bachman, N. Carr, G. Kamei, M. Kim, M.J. Pan, and et al., "A reliable approach to automatic assessment of short answer free responses", Proceedings of the 19th international conference on Computational linguistics, Taipei, Taiwan, 2002, pp. 1-4.

[11] C.E. Beevers, and J.S. Paterson "Automatic Assessment of Problem Solving Skills in Mathematics", Maths CAA Series: July 2001, last retrieved February 18th 2007 from http://ltsn.mathstore.ac.uk/articles/maths-caa-series/july2001/index.shtml

[12] R. Saikkonen, L. Malmi, and A. Korhonen, "Fully Automatic Assessment of Programming Exercises", Proceedings of the 6th annual conference on Innovation and technology in computer science education (ITiCSE '01), 2001, pp. 133-136.

[13] Patel, A., Kinshuk, K., & D. Russell, D. (1998). A computer-based intelligent assessment system for numeric disciplines. Information Service, Vo 18, No 1-2, 53—63.

[14] R. Mitkov, and L. Ha, "Computer-aided generation of multiple-choice tests", Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, 2003, pp. 17-22.

[15] R. Lister, "Objectives and objective assessment in CS1", SIGCSE Bull., Vo 33, No 1, 2001, pp. 292-296.

[16] K. Kukich, "Beyond Automated Essay Scoring", IEEE Intelligent Systems, September/October 2000, pp. 22 – 27.

[17] R. Williams, "Automated essay grading: An evaluation of four conceptual models", Proceedings of the 10th Annual Teaching Learning Forum, February 2001, Perth, Western Australia, Last retrieved February 18th 2007 from http://lsn.curtin.edu.au/tlf/tlf2001/williams.html

[18] L.S. Larkey, "Automatic essay grading using text categorization technique", Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, 1998, pp. 90-95.

[19] C.P. Rosé, A. Roque, , D. Bhembe, and K. VanLehn, "A hybrid approach to content analysis for automatic essay grading", Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada, 2003, pp. 88-90.

[20] D. Whittington, and H. Hunt, "Approaches to the computerized assessment of free text responses", Proceedings of the 3rd Annual CAA Conference, Loughborough, 1999, pp. 207-219.

[21] C. Leacock, and M. Chodorow, "C-rater: Automated Scoring of Short-Answer Questions", Computers and the Humanities, Volume 37, Number 4 / November, 2003, pp. 389-405.

[22] S.G.Pulman, and J.Z. Sukkarieh, "Automatic Short Answer Marking", Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, June 2005, pp. 9–16.

[23] L. Hirschman, E. Breck, M. Light, J.D. Burger, and L. Ferro, "Automated Grading of Short-Answer Tests", IEEE Intelligent Systems, September/October 2000, pp. 22 – 27.

[24] R. Schwitter, "English as a Formal Specification Language", Proceedings of the 13th International Workshop on Database and Expert Systems Applications, Washington, DC, USA, IEEE Computer Society, 2002, pp. 228–232.

[25] J.R. Moser, "The Electronic Assessor. Automatic Knowledge Assessment in the AdeLE E-Learning Environment", Masters Theses, Graz University of Technology, Graz, Austria, in press.

[26] R. Lütticke, "Using Semantic Networks for Assessment of Learners' Answers", Proceedings of ICALT 2006, 2006, pp. 1070-1072.

[27] T. Mitchell, N. Aldridge, and P. Broomhead, "Computerised Marking of Short-Answer Free-Text Responses", Manchester IAEA conference, October 2003.

[28] C.Y Lin, "ROUGE: A Package for Automatic Evaluation of Summaries", Proceedings of Workshop on Text Summarization Branches Out, 2004, pp. 74-81.

[29] APACHE," Apache Tomcat", Official Tomcat Web site, Apache Foundation, unpublished, last retrieved March 3rd 2007 from http://tomcat.apache.org/

[30] APACHE, "Struts Framework", Official Struts Web site, Apache Foundation, unpublished, last retrieved March 3rd 2007 from http://struts.apache.org/

[31] REDHAT, "Hibernate", Official Hibernate Web site, Red Hat Middleware LLC., unpublished, last retrieved March 3rd 2007 form http://www.hibernate.org/

[32] MYSQL, "MySQL Database", Official MySQL Web site, MySQL AB, unpublished, last retrieved March 3rd 2007 form http://www.mysql.com/

[33] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, and et al., "Developing Language Processing Components with GATE - Version 4 (a User Guide)", University of Sheffield, unpublished, last edited February 8 2007, last retrieved March 3rd 2007 form http://gate.ac.uk/sale/tao/index.html

[34] WIKIPEDIA, "Vector space model", Wikipedia, unpublished, last modified February 18th 2007, last retrieved March 3rd 2007 form http://en.wikipedia.org/wiki/Vector_space_model

[35] C.Y. Lin, "Looking for a Few Good Metrics: Automatic Summarization Evaluation – How Many Samples Are Enough?", Proceedings of the NTCIR Workshop 4, Tokyo, Japan, June 2004, last received March 3rd 2007 form http://research.microsoft.com/~cyl/download/papers/NTCIR4.pdf

AUTHOR

Christian Gütl is with the Institute for Information Systems and New Media, Graz University of Technology, Graz, Austria (e-mail: cguetl@iicm.edu). He is also head of Guetl IT R&D and FEO of Infodelio Information Systems.

Manuscript received 28 August 2007. Published as submitted by the author.