INTELLIGENT ELECTRONIC ASSESSMENT FOR SUBJECTIVE EXAMS

Alla Defallah Alrehily, Muazzam Ahmed Siddiqui, Seyed M Buhari

Faculty of computing and information technology, King Abdulaziz University, Saudi Arabia, Jeddah

ABSTRACT

In education, the use of electronic (E) examination systems is not a novel idea, as E-examination systems have been used to conduct objective assessments for the last few years. This research deals with randomly designed E-examinations and proposes an E-assessment system that can be used for subjective questions. This system assesses answers to subjective questions by finding a matching ratio for the keywords in instructor and student answers. The matching ratio is achieved based on semantic and document similarity. The assessment system is composed of four modules: preprocessing, keyword expansion, matching, and grading. A survey and case study were used in the research design to validate the proposed system. The examination assessment system will help instructors to save time, costs, and resources, while increasing efficiency and improving the productivity of exam setting and assessments.

KEYWORDS

Subjective Assessments, E-Examination, WordNet, semantic similarity.

1. INTRODUCTION

With the rapid growth of modern education, the idea of E-learning system has been implemented to enhance the teaching of online courses, allowing instructors to offer online examinations through virtual classrooms. Electronic-learning overcomes many problems faced by students, such as the expense of traditional academic courses. Exams are an essential activity for students' learning as they assess the students' knowledge and level of understanding about a given subject. Therefore, the key aspects of an examination system are preparing a new paper for each student and conducting follow-up assessments.

In universities, a faculty member needs to set a minimum of three assessments per semester for a course (i.e., mid-term I, mid-term II, and final examination). Each faculty member generally teaches three courses per semester. Examination paper setting, and assessment are time- and labor-intensive, requiring many resources and placing immense pressure on course instructors. So, E-examination systems are importance in universities and institutions because it presents them electronic exams as a function open to all students in various places. For example, universities such as MIT, Berkeley, and Stanford have prepared electronic exams for massive open online courses (MOOCs) [1]. E-examination systems have the ability to check and set exam papers electronically, setting grades and assessing answers efficiently and yielding results quickly. These systems utilize fewer resources and minimal effort on behalf of the users. In contrast, traditional examination systems require physical resources such as pens and paper, greater user efforts, and more time.

Dhinaharan Nagamalai et al. (Eds) : ACSIT, ICITE, SIPM - 2018
pp. 47–63, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80804

Existing electronic-examination systems only evaluate exams with objective questions. But recently, researchers have identified the need to assess subjective questions using this tool [2]. Therefore, universities are in search of improved examination setting and assessment methods aside from the currently used manual method [3]. Therefore, there is a need for automatic examination and assessment systems in this context.

To tailor the existing assessment process in which examinations are set manually, this research aimed to develop an electronic assessment system for subjective examinations to assist instructors with exam setting and the assessment process. A new design is proposed for an electronic-examination assessment algorithm, which is achieved using the concept of semantic and document similarity to find a matching ratio between instructor and student answers to each question. The electronic system randomly generates exam papers, including both objective and subjective questions. A survey and case study are used in the research design to validate the electronic-examination system. In the case study,10 students in King Abdul-Aziz University (KAU) were tested.

The rest of this paper is structured as follows. Section 2 discusses related research in the literature. Section 3 presents the problem statement. Section 4 explains the proposed system. Sections 5 and 6 describe the exam paper and the proposed assessment algorithm, respectively. In Section 7, the output of the examination assessment is presented, and in Section 8, the system is evaluated. Section 9 provides concluding remarks.

2. LITERATURE REVIEW

Xinming and Huosong [4] present an automated system that addresses the following problems with assessing subjective questions: synonymy, polysemy, and trickiness. Latent semantic analysis (LSA) and the ontology of a subject are introduced to solve the problems of synonymy and polysemy. A reference unit vector is introduced to reduce the problem of trickiness. The system consists of two databases: a science knowledge library and a question- and reference-answer library. The science knowledge library stores the ontology of a subject as text documents. The question- and reference-answer library stores questions as text documents and reference answers as a text document matrix. When a teacher adds new questions, a system using this science knowledge library will search for related points of knowledge and keywords and give them to the teacher. Then, the teacher will submit the reference answer to the system. It will process the reference answer using Chinese automatic segmentation, which produces text-document vectors and sends them to the teacher. Then, the teacher detects the terms and their weights for each vector and sends them back to the system. Weights of the terms in the reference answer are computed using the term-frequency and inverse-document-frequency functions. In the questions and reference answers, the library will save the vector of the reference answers and questions as text documents. To compute the similarity between a student's answer and the reference answer, the former is sent to the system, which assesses the answer using Chinese automatic segmentation and produces a text vector projected into k-dimensional LSA space. This LSA is formed by a vector using the mathematical technique of singular value decomposition (SVD), which represents terms and documents that are correlated with each other. The system computes the cosine similarity of student and reference answer vectors projected into k-dimensional LSA space in the reference unit vector.

In [5], machine learning techniques with and without ontology are presented to evaluate subjective answers. The techniques without ontology include LSA [4], generalized latent semantic analysis (GLSA), bilingual evaluation understudy (BLEU), and maximum entropy (MAXENT). Using ontology to evaluate subjective answers, student answers to questions and concept details are fetched from the ontology based on the type of question. If short questions are answered, only a few details are extracted from the ontology. And if longer questions answered, details extracted from ontology are more and the similarity score among concepts extracted. After the information extracted from ontology, be configured a Multi-Hash map that used for evaluating answers. This Multi-Hash map collected all the words symmetrical for the same concept. If the concepts have a track among each other, then the length of such the track is computed. The authors combined ontology with machine learning techniques. The input of all machine learning techniques is the model answer and students' answers, a multi-hash map of Ontology concepts and distance among concepts. The method of combine ontology with machine learning techniques is constructing Ontology concepts of the sentences in the model answer and using the machine learning technique, merging concepts with the Ontology map. Using same machine learning technique, finding a correlation between every concept and students' answer in the multi-Hash map. To compute the final score of the similarity between students' answers with the model answer, the distance among the main concept and current concept is multiplied by the whole number of concepts having a positive correlation with students' answers. Then, this estimate is divided by a whole number of concepts in multi-Hash Map to construct final score. The most technique merged with Ontology is the word-weight technique. In this technique, the words are extracted from ontology and then words in the model answer are associated with ontology concepts. Finally, the weight of every keyword is computed.

Using the machine learning techniques without ontology, they take keywords of the model answer and student answer as input. The output is a similarity measure in the range between 0 and 1 where a value of 0 indicates no similarity and 1 indicates the high similarity. Before applying the machine learning techniques, pre-processing of words is tokenization, stop word removal, synonym search and stemming performed for the input.

Maram et al. [6] introduces an Automatic evaluation of an essay (AEE) system which is written in Arabic. The system presents a hybrid approach which integrates the LSA [4] and rhetorical structure theory (RST) algorithm. LSA method supports the semantic analysis of the essay, and the RST to evaluate the writing method and the cohesion of the essay. The LSA method finds the similarity ratio among two texts even if they do not include similar words. The system processes input essay into two phases is a training phase and testing phase. The training phase is made up of three parts: calculating the average of words per essay, calculating the most ten visible words on a given topic and applying LSA algorithm. The testing phase passes through a number of processes:1) calculating LSA distance.2) calculating the number of a vernacular.3) calculating a number of repeated sentences.4) calculating the length of the essay.5) calculating number of spelling mistakes.6) applying RST algorithm.7) checking cohesion of essay related to the topic. Then applying two phases, the system computes the final score based on the cosine distance of LSA between the input essays and the training essays. The system graded school children essays based on three criteria which are 40% of the total score for writing method, 50% for the cohesion of the essay and 10% for spelling and grammar mistakes.

Anirudh et al. [7] propose an automated evaluation system for descriptive English answers that contains multiple sentences. The system evaluates the student's answer with an answer-key for questions of professional courses. It depends on a group of algorithms for natural language processing which are Wu and Palmer, Longest Common Substring (LCS), LSA [4], Cosine Similarity and Pure PMI-IR. The algorithms analyze the student's answer with an answer-key for finding the similarity score between them. Then, similarity scores extracted from algorithms are merged using the logistic regression machine learning to produce a score that is recommended by instructor. Wu-Palmer technique compares the word in the student's answer with each word in answer-key. If both words are present in the English dictionary, Wu-Palmer technique computes a similarity score for both words. Otherwise, if both words are not present in the dictionary, then the comparison is done using edit distance. LCS used to compare both sentences of the student's answer and answer-key. Then, the similarity score of LCS combined with a similarity score of

Wu-Palmer technique using the similarity matrix method. LSA uses SVD [4] on the similarity matrix that formed of both sentences. SVD produces two vectors representing two sentences. The similarity between two sentences is computed using cosine similarity. Pure PMI-IR combines all similarity scores of word pairs among sentences in one value using the similarity matrix method. The multi-class Logistic Regressors technique combines results of all five techniques to produce a score for the answer.

Ishioka and Kameda [8] propose an automated Japanese essay scoring system named jess. The system uses to mark essays in Japan for the University Entrance Exams. It assesses the essay from three metrics: rhetoric, organization, and content. Rhetoric means a syntactic variety that measures the details which are the ease of reading, diversity of vocabulary, percentage of big words and percentage of passive sentences. Organization means presenting and relating ideas in the essay. For organization assessment, jess examines the logical structure of the document and attempts to determine the occurrence of definite conjunctive expressions. Content means relevant information such as the precise information provided, and the vocabulary employed to the topic. For content assessment, jess applies a technique named LSA [4] which be applied to examine if the contents of a written essay react well with the essay prompt. Jess uses learning models which are editorials and columns extracted from the Mainichi Daily News newspaper.

In [9] proposes an approach of evaluation of online descriptive type students' answers using Hyperspace Analog to Language (HAL) procedure and Self-Organizing Map (SOM) method. To evaluate students' answer, the student writes the answer and sent as input to HAL. HAL constructs a high dimensional semantic matrix from a collection of an n-word vocabulary. Method for construct matrix through motivation a window of length "1" by the corpus through one-word increment. HAL ignores sentence boundaries, punctuation and converts each word to numeric vectors expressing information on its meanings for words. Inside window computes the distance between two words is "d", then computes "(l-d+1)" which denotes the weight of an association among two words. This matrix presents words by the analysis of lexical co-occurrence. Every word represents in the row vector based on the co-occurrence data for the words preceding this word and every word represents in the column vector based on the co-occurrence data for words following it. The matrix converts into a singular value by using SVD function [4]. Vector produced by HAL enters as an input to the Self-Organizing Map (SOM). It clusters words based on finding Euclidean distances denote the document similarity. SOM is neural technique. SOM takes vectors and produces a document map. Then, neurons are nearby will include the similar document. The authors compared SOM results with other clustering methods like Farthest First, Expectation Maximization (EM), Fuzzy c-Means, k-Means and Hierarchical. They concluded that SOM awards better performance.

kumaran and Sankar[10] propose a technique of an automated system for assessing the short answers using ontology mapping. Three stages of assessing the short answers are RDF sentence builder, ontology construction, and ontology mapping. In the first stage, the system constructs the RDF sentence for every sentence in student answer and model answer after reading the model answer and student answer as input in plaintext form. The system parses each sentence and builds the grammatical relationships each sentence. It uses Stanford typed dependency parser to represent dependency relationships. In the second stage, the RDF sentences are as input to ontology constructor to construct an ontology for them. The authors use sequential and coordinate links to construct RDF graph for the RDF sentences. The sequential link means that object or predicate is mutual among two RDF sentences. The coordinate link means that subject is same of two RDF sentences. Each link in ontology has the weight in the range of 0 to 100 based on the level of significance of that sentence in the answer and the whole weight of all links will be 100. In the third stage, the output of the previous stage is the model answer ontology and student answer ontology that use them the ontology mapping to perform matching operation. Output for this stage is the mark for the student answer depend on the weight age and the similarity score.

The method of the ontology mapping is the first finding the matching between the edges of the model answer ontology and student answer ontology are using the Cartesian product. The second, finding the similarity between two vertices of two ontologies using wordnet based similarity measure.

Raheel and Christopher [11] propose a system that provides a novel approach for automated marking of short answer questions. To compute the grade for the student's answer, authors introduce the architecture for the system that is composed of three phases to address the student's answer. Three phases are 1) spell checking and correction that is implemented by an Open Source spell checker like JOrtho.2) parsing the student's answer using the Stanford Parser. This statistical parser can be creating parses with high accuracy. The parser offers the following results which are the part of speech tagged text and design dependency grammatical relations among singular words. 3)The Third phase of the processing answer is a comparison between the tagged text with syntactical structures specified by authors in Question and Answer Language. This phase addressed by syntax analyzer. Also, architecture contains analyzer of grammatical relation that compares between the grammatical relations in student answer with the grammatical relations specified by the examiner. The last task in the comparison phase is passing the results summarized from the syntax analyzer and the grammatical relation analyzer to the marker that calculates the final grade of the answer.

The Automatic marking system for a student's answer examination of the short essay was introduced by Mohd et al. [12]. The system applied to sentences were written using the Malay language that requires technique to process it. The technique mentioned in [11] which is the syntactic annotation and the dependency group to represent the Grammatical Relations(GR) from Malay sentences. To process the sentences from the marking scheme and the students' answers, all entries to the Computational Linguistic System (CLS) for linguistic processing like tokenizing, recognizing, collocating and extracting the GRs. The system contains a database for a table of Malay words and their Part of Speech (POS) to assist the CLS. To compute the mark for the student's answer, compare the GR extracted from the students' answers with the GR for the marking scheme. In other words, comparison components of the sentences as follows: subject to the subject, verb to the verb, object to object and phrase. The authors did the test of the system to view how the system gives marks compared to the marks awarded by a human. They selected lecturers have experienced in marking the scheme from Malaysia to set the mark for each question. The test presents which the system can give similar marks as marks awarded by the lecturers.

A new automated assessment algorithm for assessing the Chinese subjective answers was proposed by Runhua et al. [13]. The algorithm called Automated Word and Sentence Scoring (AWSS) assesses the student answers for the level of word and sentence. From fundamental problems of the Chinese, Natural Language Processing is the word segmentation, but this problem solved by the Institute of Computing Technology, the Chinese Lexical Analysis System (ICTCLAS). It assesses the student's answer to the standard answer in two phases as follows:1) compute similarities between two words depend on How-Net. In this phase, they check keywords weight and phenomena of the synonym. The authors' present results of How-Net is satisfied. To compute the similarity between student answer with the standard answer for the level of the sentence, the authors' divide sentence to a series of words. Then computing the best matching pair of every word in the sentence and computing the sentence similarity as functions mentioned in [13]. 2) compute the similarity of sentences depending on dependency structure among words of a sentence. This phase parses the sentence by the language technology platform (LTP) to find out the dependency structure of the sentence. The method of computing dependency structure is finding a valid pair which is a noun, verb or subjective linking to the head of the sentence. Then, computing the sentence similarity based on dependency structure as functions mentioned in [13].

Xia et al. [14] design automatic scoring algorithm for a subjective question. They use the idea of a one-way approach degree depending on the closeness theory of fuzzy mathematics. The authors are calculating the closeness of two fuzzy sets which are set "A" denoted by the standard answer string and set "B" denoted by the student answer string. A fuzzy set is an ordered collection from a single character that decomposed from a string. To compute a one-way approach degree between two fuzzy sets "A" and "B", "B" contain n characters and one-way approach degree denoted by δ (B, A)=m/n whereas m denotes by the effective sum number of the set B in each element in the set A. δ (B, A) introduce B close to A unidirectional closeness. The introductory algorithm provides the aim of the system.

Zhenming et al. [15] propose a novel web-based online objective examination system for computer science education. This system conducts the examination and auto-marking of objective questions and operating questions. The system transmits answers and questions into the bit stream after encoding to ensure security and intrusion. It is the password protected system and camera are used to monitor the activities of students. The auto-grading system can automatically grade the answers, that are collected from the examination system. The objective questions can be graded effectively via fuzzy matching. But operating questions is difficult to grade by simple Matching technologies. Thus, researchers propose a universalized grading system that is achieved on the foundation of a database for key knowledge. The system does the following: first, they elicit all likely knowledge points and store them in a triple form (key, value, location). Then they make the question file via labelling the question point directly on it. After that, the system will add the identical question key to the standard key library. The last process of the system is comparing the answer file with the standard key library.

Our study is similar from previous studies for using the concept of semantic similarity and document similarity to find the matching ratio between instructor answer with student answer.

3. PROBLEM STATEMENT

In the education field, universities are currently setting and assessing the examination papers manually. Therefore, it is in need of automatic examination and assessment systems. Due to the manual exam setting and assessment for university faculties, they are facing following the main problems:

1) It is a tedious process to set exam papers and quizzes in every semester.

2) It needs a lot of time, more effort on instructors and consumes more resources to set and assess the examination papers especially if a number of students in the class are greater than thirty.

3) The paper-based examinations are currently scanned to convert them electronically for the review of The Accreditation Board for Engineering and Technology (ABET). This requires extra time, cost and resources.

To cater three issues, this research aimed to develop an electronic objective and subjective examination an assessment system to address the problems of universities and it will be helpful for the other universities inside and outside of the Kingdom of Saudi Arabia. It is anticipated that the proposed system will help instructors in the exam setting and its assessment. The proposed system will save time, cost, resources, increase efficiency and improve the productivity of exam setting and assessments.

4. THE SYSTEM USERS

The electronic system consists of two concepts which are the examinations setting and assessment system. Figure 1 below shows the electronic system users. The system has two courses which are System Analysis and Design and Software Engineering. The questions and answers of exams in two courses collected and questions composed of equally distributed simple, average and difficult questions.

The main users of a system which are an instructor, head of the track, head of department, student and system administrator. Each of users has own screen to log in with the user name and password. The users have specific functions applied to them in the system. The instructor can create the objective and subjective questions and select course type and write the grade for each question. He can approve grades for students on the main screen. The head of the track can modify and approve all the objective and subjective questions which are created by the instructor. The student selects a course to start the exam and solve the questions. A student can view final grade. After approving the instructor for final grades, head of the department can approve and publish it to students.

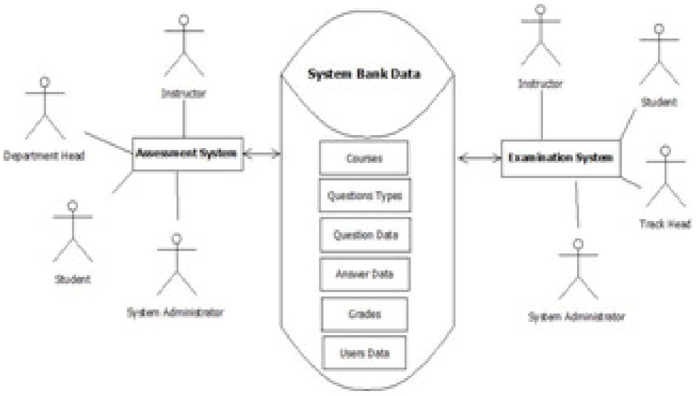


Figure 1. System users

5. EXAM PAPER DESIGN

The Exam paper had consisted of two sections. The first section was objective questions of multiple choices. The second section was subjective/ descriptive questions which are a short essay, definitions, and lists. Questions were coming one after the other -in both parts. The system selected questions randomly of the database using function “RAND”. In the system, two courses were System Analysis and Design, Software Engineering. The student selected the course to start the exam. Then, the system selected randomly five objective questions and five subjective questions. The system had a specific time for the exam. It set 25 minutes for solving objective section and 55 minutes for solving subjective section.

6. THE PROPOSED ASSESSMENT ALGORITHM

The Assessment system architecture consists of different modules which assess student answers with reference answers. The modules are a pre-processing module, Keyword Expansion module, matching module and Grading Module. Figure 2 bellow presents subjective examinations assessment algorithm. The details were explained in next section.

6.1. Preprocessing module

The inputs to the module were the reference answers provided by the instructor and student answers. Two answers converted to lowercase using “lower ()”. Then, the module removes stop words, punctuations, and prepositions from converted answers. The output of the module is the cleaned answer.

For processing punctuations of text, the module was called a string containing all characters considered punctuation. This is the string """!"#$%&'() *+, -./:;<=>?@[\]^_`{|}~""" . The module has converted the string of punctuations to set using “set ()”. If the characters of the answer were not in punctuations set, the module has joined the characters with an empty space using “join ()”. The sentence can be split into words using the method “word tokenize ()”. Tokenizers can be used to find the words in a string. The module imports natural language toolkit(NLTK) can provide “word tokenize ()” and other methods for processing texts. The output of the method is a list of words named keywords.

For processing stop words of text, the module was imported stop words of the NLTK package can provide a list of stop words. If keywords were not in a set of English stop words, the keywords remain in the list.

For processing prepositions of text, the module was classified keywords into their parts of speech is known as part-of-speech tagging (POS-tagging). Parts of speech are also known as lexical categories. The module was imported POS-tagging of the NLTK package can attach a part-of-speech tag to each word. It was set POS-tagging using the method “pos tag ()”. If POS-tagging of keywords were not in prepositions list, the keywords remain in the list. After applying all processing texts methods, the module returned all keywords of answer joined with an empty space using “join ()”.

As for the objective answers are being processed by fetching the reference answers from Objective Questions Approved store and also fetches the student answers from Exam Session store. If student answer matches with reference answer, then answer is correct. Otherwise, the answer is not correct.

The inputs to the pre-processing module:
Reference answer="The Cost required to deVelop the System"
Student answer = "The scheme NEED to involve by estimate the cost”
The outputs of the pre-processing module:
Cleaned Reference answer =” cost required develop system”
Cleaned Student answer =” scheme need involve estimate cost”

6.2. Keyword Expansion module

After cleaning student's answer and reference answer in the Pre-processing module. The cleaned reference answer split to a list of keywords using “split ()”. The Keyword Expansion module achieved two tasks were got synonyms and apply synonyms. The first task was got synonyms set each keyword of wordnet. The module was imported wordnet of the NLTK package can provide synonyms set of English words. Look up a keyword using function “synsets()”. The output of this process is synonyms set of keywords. It was accessed synonyms set and get the synonyms each keyword using function “lemma names ()”. The synonyms contained within synonyms set are called lemmas. The module was imported chain from iterator tools (itertools) module makes an iterator that returns the synonyms from the many literators until it is exhausted. Thus, synonyms presented as the chain using the function “chain. From iterable()”. Then, the module was put synonyms in the set using “set ()”. The output of the first task is synonyms of the keyword. Then, it was presented each keyword and its synonyms as a dictionary using the method “Dict ()”.

After finding synonyms of keywords of reference answer. The second task received two inputs were cleaned student's answer and dictionary of keywords in reference answer. The cleaned student's answer split to a list contained in words named text. The module was generated empty list named words list. Each word in the text added to words list using the method “append ()”. The module was called items of keywords and synonyms of reference answer using the method “items ()”. Thus, this method returned (keyword, synonym) tuple pairs. If the word in the student's answer matched with the synonym as presented in tuple pair. Then, this module has deleted this word from words list using delete operator. And it was replaced its place the basic keyword as presented in tuple pair using “append ()”. The basic keyword is from the reference answer. The output of the second task is the basic keywords and words have not the synonym. All these words joined with an empty space using “join ()”. Thus, this is the new student's answer after converting synonyms to the basic keywords. This task applied also to cleaned reference answer.

The inputs to the Keyword Expansion module:
Cleaned Reference answer=” cost required develop system”
Cleaned Student answer=” scheme need involve estimate cost”

a. get synonyms of keywords of cleaned reference answer
input to task a:
Cleaned Reference answer=” cost required develop system”
Output of task a:
'system': {'system', 'organization', 'arrangement', 'scheme', 'organisation', 'system_of_rules'} , 'develop': {'germinate', 'grow', 'explicate', 'acquire', 'rise', 'get', 'modernize', 'break', 'evolve', 'make_grow', 'recrudesce', 'develop', 'originate', 'uprise', 'modernise', 'build_up', 'produce', 'educate', 'arise', 'prepare', 'train', 'formulate', 'spring_up'}, 'cost': {'price', 'be','monetary_value', 'cost', 'toll'}, 'required': {'requisite', 'mandatory', 'need', 'take', 'necessitate', 'require', 'involve', 'ask', 'command', 'needful', 'call_for', 'want', 'required', 'postulate', 'demand', 'expect', 'compulsory', 'needed'}

b. applies synonyms of cleaned reference answer and cleaned student answer
The inputs to task b:
1.Cleaned Reference answer=” cost required develop system”
2.Cleaned Student answer=” scheme need involve estimate cost”
3.dictionary of task a
The outputs of task b:
1. Cleaned Reference answer
“cost” =” cost”
“required” = “required”
“develop” = “develop”
“system” =” system”
2. Cleaned Student answer
“scheme” =” system”
“need” =” required”
“involve” = “required”
“estimate” did not match any synonym
“cost” =” cost”
The outputs of the Keyword Expansion module:
New Reference answer=” cost required develop system”
New Student answer=” system required required estimate cost”

6.3. Matching module

This module achieved two tasks: the first task was converted text to vector and the second task was computed cosine similarity between two vectors. The first task received two inputs were the new student's answer and new reference answer from the previous module. It displays the textual representation of two answers into Vector Space Model (VSM). VSM represents answers as vectors in n-dimensional space where n is the total number of keywords in all the answers.
The first task works as the following:

1. import the module regular expressions defined as re. This module provides an interface to the regular expression engine. It can compile the regular expression to pattern objects which have methods for pattern matches.

2. construct the regular expression as pattern object named WORD= re.compile(r'\w+').The regular expression r'\w+' passed to the object as a string. It used to match words in the target answer.

3. The module was applied directly the method “findall” on the regular expression object “WORD”. The method received the target answer to finding matches in.

4. The output of step 3 is a list contains on matched keywords named keywords.

5. To compute how many frequencies of matched keywords and not matched keywords each answer. The module was imported Counter from collections module and call the method “counter ()” which receives keywords list of step 4. It counts of keywords per the answer. The output of this process is keyword frequency vectors are created each student's answer and reference answer. In keyword frequency vector, the keyword was indicated key and frequency number was indicated value. The mathematical expression of step 5 represents as follows whereas Keyword (K), Keyword Frequency (KF), frequency (fr) and Answer Vector(AV):

$$KF(K,A) = \sum_{x\epsilon A} fr(x,K)\ where\ fr(x,K) = 1\ if\ x = K\ otherwise\ fr(x,K) = 0$$

$$AV = (KF(K1,A), KF(K2,A)..., KF(Kn,A) where\ n\ is\ \text{number of keywords} \qquad (1)$$

The basic trend in the research is finding matched keywords between the student's answer vector and reference answer vector to assess the student's answer is correct or incorrect. This is the second task received two inputs were the student's answer vector and reference answer vector. It computed similarity ratio between two vectors using the similarity method is known document similarity. It used measure named cosine which computes the distance angle between the student's answer vector and reference answer vector. The mathematical expression of the task cosine similarity is as follows. Whereas it gives two vectors are Reference Answer Vector(RAV) and Student Answer Vector(SAV):

$$\boldsymbol{similarity\ ratio = \cos(\theta)} = \frac{\sum_{i=1}^{n} RAVi \times SAVi}{\sqrt{\sum_{i=1}^{n} RAVi^2 \times \sum_{i=1}^{n} SAVi^2}} \qquad (2)$$

The programmatic representation of the function “cosine similarity” is as the following:

1.To represent numerator as in the mathematical expression, extracting all the keywords of each vector using the method “keys” which returned keywords list of RAV and SAV. Then, putting keywords list of each vector as a set. Finding the intersection between two sets to extract matched keywords. The intersection represented as in the programmatic representation set (RAV. keys) & set (SAV. Keys). Each value of RAV in the intersection multiplied by each value of SAV. Then, summation all results using the method “sum”.

2. To represent denominator as in the mathematical expression, square each value in RAV then summation all values. And also, the module was applied to SAV values. The result for RAV named “sum 1” and for SAV named “sum 2”. After that, the module was extracted sqrt of “sum 1” and “sum 2”. The result of sqrt 1 multiplied by the result of sqrt 2.

3.If the result of numerator equals to zero, similarity ratio was zero. Otherwise, similarity ratio was the result of divide numerator to the denominator.

The similarity ratio of function “cosine similarity” represented the cosine of the angle was between 0 to 1. The similarity ratio was 1 means student answer is matched with reference answer. If the similarity ratio is the approximate number closer to 1 means that student answer is more similar for reference answer. Otherwise, if the approximate number closer to 0 means student answer is less similar for reference answer. Similarity Percentage (SP)is calculated as follows:

$$\mathbf{SP\%} = \boldsymbol{similarity\, ratio} \times \mathbf{100} \tag{3}$$

The inputs to the matching module:
New Reference answer=” cost required develop system”
New Student answer=” system required required estimate cost”
a. convert text to vector
reference answer vector ({'system': 1, 'develop': 1, 'cost': 1, 'required': 1})
student answer vector ({'required': 2, 'cost': 1, 'system': 1, 'estimate': 1})
b. compute cosine similarity
cosine similarity ratio between RAV and SAV =0.75
The outputs of the matching module:
SP=75%

6.4. Grading Module

The module computes the full mark based on the similarity percentage. The full mark of the exam paper is ten out of ten. The number of exam questions is ten questions, and each question is of one mark. After processing answers using previous modules, this module gets the percentage of similarity of each question. If the similarity percentage between the student's answer and reference answer are getting between 70% and 100%, the student's answer is correct and full mark awarded. Otherwise, it is incorrect. Then the module computes the final grade of the exam by collecting grades of all questions. And grades store in Exam Session store and Grades Not Approved store.

Figure 2. subjective examinations assessment algorithm

Input 1: student answer
Input 2: reference answer

The output of all modules: similarity ratio

1. Pre-processing module

a. removes punctuations

```
exclude = call all characters considered punctuation
for a character in two answers
     if character not in exclude
       sentence = join all characters with an empty space
      End if
tokens= split words in a sentence into tokens
End for
```

b. removes stop words

```
Stop words = call list of English stop words
for word in tokens
     if word not in Stop words
       tokens = put a word in the list.
      End if
End for
```

c. removes prepositions

```
tagged = call parts of speech tagging and define it to tokens
for a tag in tagged
     if tag not in prepositions list
          keywords = put tokens in the list
      End if
End for
Output 1:  cleaned student answer
Output 2:  cleaned reference answer
```

2. Keyword Expansion module

Input 1: cleaned student answer

Input 2: cleaned reference answer

a. get synonyms of keywords of cleaned reference answer

```
synonyms set = call synonyms set of keywords
for a keyword in synonyms set
     synonyms = get synonyms each keyword as chain
     keywords and synonyms of reference answer = present each keyword and its synonyms as a
dictionary
End for
```

b. applies synonyms

```
text = split cleaned student answer to list
words list = generate an empty list
 Tuple pairs (keyword, synonym) = call items of keywords and synonyms of reference answer
for word in the text
```

```
   add a word to words list
    for keyword and synonym in tuple pairs
           if the word in the synonym
                delete word of words list
                add a keyword to words list
           End if
      End for
    End for
Output 1: new student answer after applying synonyms
Output 2: new reference answer after applying synonyms
```

3.Matching module

```
Input 1: new student answer
Input 2: new reference answer
```

a. convert text to vector

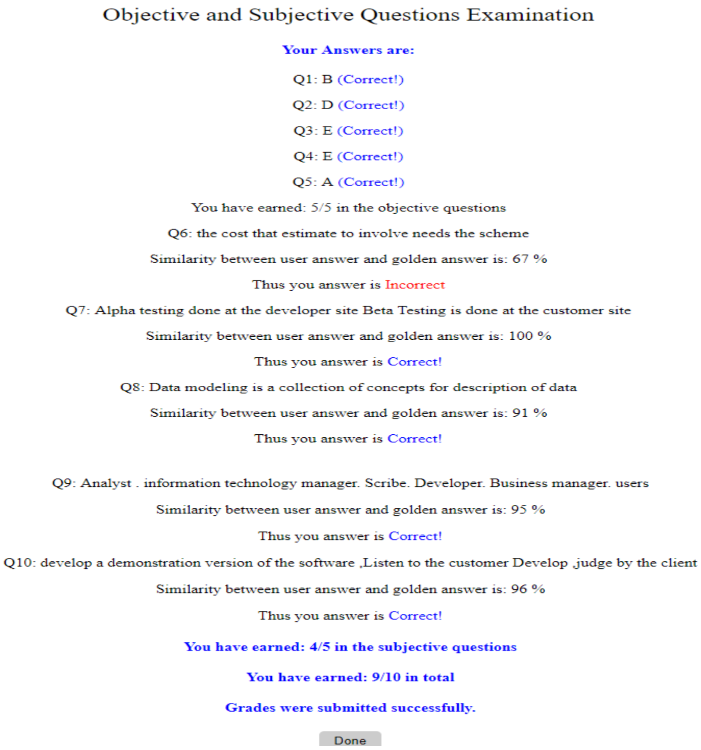
```
words = find keywords in new student answer
SAV= count words
RAV= count words
```

b. compute cosine similarity

```
SAV keywords = call keys of SAV
SAV set = put SAV keywords in the set
RAV keywords = call keys of RAV
RAV set = put RAV keywords in the set
intersection = Find the intersection between two sets
for x in the intersection
      value 1 = get x value of SAV
      value 2 = get x value of RAV
      values = value 1 * value 2
      numerator = summation all values
End for
for x in SAV keywords
    square 1 = square x value of SAV
    sum 1 = summation all square 1
End for
for x in RAV keywords
    square 2 = square x value of RAV
    sum 2 = summation all square 2
End for
Sqrt 1 = sqrt of sum 1
Sqrt 2 = sqrt of sum 2
denominator = Sqrt 1 * Sqrt 2
if not denominator
   return 0
else
   similarity ratio = numerator / denominator
return similarity ratio
End if
```

7. OUTPUT OF EXAMINATION ASSESSMENT

The electronic system interfaces of users and exam paper implemented using PHP and java script languages. The system database implemented using my SQL. The assessment algorithm for exam paper implements using python program. After the student has solved exam, student answers with reference answers evaluated using assessment algorithm. Figure 3 shows bellow assessment results of the exam paper.



Objective and Subjective Questions Examination

Your Answers are:

Q1: B (Correct!)

Q2: D (Correct!)

Q3: E (Correct!)

Q4: E (Correct!)

Q5: A (Correct!)

You have earned: 5/5 in the objective questions

Q6: the cost that estimate to involve needs the scheme

Similarity between user answer and golden answer is: 67 %

Thus you answer is Incorrect

Q7: Alpha testing done at the developer site Beta Testing is done at the customer site

Similarity between user answer and golden answer is: 100 %

Thus you answer is Correct!

Q8: Data modeling is a collection of concepts for description of data

Similarity between user answer and golden answer is: 91 %

Thus you answer is Correct!

Q9: Analyst . information technology manager. Scribe. Developer. Business manager. users

Similarity between user answer and golden answer is: 95 %

Thus you answer is Correct!

Q10: develop a demonstration version of the software ,Listen to the customer Develop ,judge by the client

Similarity between user answer and golden answer is: 96 %

Thus you answer is Correct!

You have earned: 4/5 in the subjective questions

You have earned: 9/10 in total

Grades were submitted successfully.

Figure 3. Assessment Output of the Exam Paper.

8. EVALUATION

Electronic system evaluated in several respects. First, in terms, the quality evaluation of the system will be using a survey. Second, in terms the performance evaluation using popular evaluation measures for electronic assessment and traditional assessment. And, evaluation using Spearman correlation for electronic assessment with traditional assessment.

8.1. The system quality Evaluation

The survey will be used research design to validate the quality of the electronic system. The survey distributed on different categories of KAU and other universities. The categories are the users of the system. The sample size is fifty user which tested the quality of the system. The fifty users are twenty-one students, six heads of departments, eight heads of tracks and three administrators of systems. Most users are female, and others are male. When analysing the survey results, 86% of users strongly agree and 14% of users agree which the system gives correct results when they used it. And also, same previous percentages, they take less time for accomplishing any task into the system. 78% of users strongly agree and 22% of users agree which the efficiency of the system is fast to accomplish their tasks. 98% of users strongly agree and 2% of users agree which the system is easy to accomplish their tasks when they used for the first time. 96% of users strongly agree and 4% of users agree which functions as users of the system is completed. All users of the system are 100% strongly agree for consuming less cost and resources at use the system and accomplishing tasks. And they are satisfied with the whole system. Thus, the system achieves the main criteria of the system evaluation that it consumes a less of time and fewer resources. And, it reduces the effort on the users of the system and functions are high quality.

8.2. Comparison Electronic Assessment with Traditional Assessment Result

The electronic system has experimented on ten students of KAU. Data sets are 100 questions composed of 50 objective questions and 50 subjective questions of different courses. Experimental results of answers grades are conducted and analyzed using excel program. The study was used commonly the evaluation measures which are recall, precision, accuracy and F measure for measuring retrieval effectiveness of electronic assessment and traditional assessment. Traditional assessment is using instructors approach to assessing exam papers manually. The instructor's approach counts the correct words in the student answer and divides them by the total number of words in instructor answer. Then multiply the result at 100 to extract the similarity percentage between student answer with instructor answer manually.

The table following shows the mathematical expression of previously defined measures whereas True Positive (TP) is the number of answers correctly labeled as positives, True Negative (TN) is the number of answers correctly labeled as negatives, False Positive (FP) is the number of answers incorrectly labeled as positives and False Negative (FN) is the number of answers incorrectly labeled as negatives.

Table 1. Mathematical Expressions of Evaluation Measures

Measure	Mathematical Expression
Recall	TP / (TP + FN)
Precision	TP / (TP + FP)
Accuracy	(TP + TN) / (TP+TN+FP+FN)
F measure	(2×Recall × Precision) /(Recall + Precision)

Figure 4 shows the Performance comparison for electronic and traditional assessments using common evaluation measures. The recall is called True Positives Rate in the electronic assessment was 0.9778 and was 0.9655 in traditional assessment. Precision is called Positives Predictive Value was 0.9072 in electronic assessment and was 0.9882 in traditional assessment. Accuracy is called true results was 0.8900 in electronic assessment and was 0.9600 in traditional assessment. Finally, F measure is the geometric mean of recall and precision. It was 0.9412 in electronic assessment and was 0.9767 in traditional assessment.

The study was concluded recall rate for electronic assessment is so much closer to traditional assessment. When compared precision, accuracy and F measure for electronic assessment with traditional assessment, the study was concluded these measures for traditional assessment is higher than electronic assessment.

For evaluation purpose, the Spearman correlation is very important to note the extent of correlation between electronic assessments with traditional assessments. The Spearman correlation computed using excel program. The results of Spearman correlation were 0.83. Thus, the correlation represented the strength of the relationship between grades computed by electronic assessment and traditional assessment.

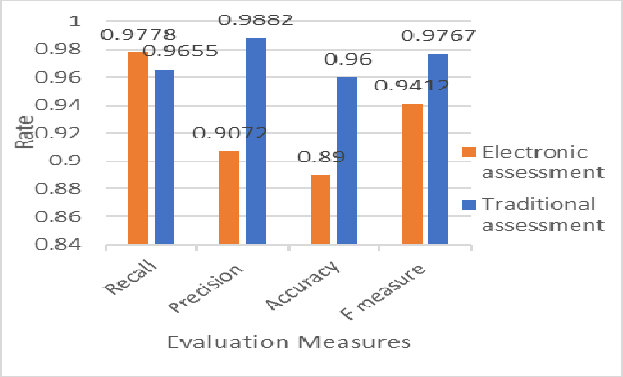


Figure 4. The Performance comparison for electronic and traditional assessments.

9. CONCLUSIONS

In the education domain, electronic examination systems are used to deal with objective assessments. Now, our need electronic examination systems to assess subjective questions in exams. There are several problems associated with the manual examination and assessment processes such as time-consuming, costly, enormous resources, a lot of efforts and huge pressure on instructors.

The paper was introduced a new design for an electronic examinations assessment system which achieves using the concept of semantic similarity and document similarity to find matching between instructor answer with student answer for each question. Then the system extracts the grade based on a percentage of similarity. The electronic grades correlate with instructor grades using Spearman’s correlation. The accuracy of assessment using the electronic system is high. Thus, the proposed system will be beneficial for the faculties of other universities inside and outside of Kingdom of Saudi Arabia. The electronic system will help instructors in the exam setting and its assessment. It will save time, cost, resources, increase efficiency and improve the productivity of exam setting and assessments. Future work will develop assessment algorithm to address syntax errors of keywords and investigate high equality and performance for assessing them.

REFERENCES

[1] J. Dreier, R. Giustolisi, A. Kassem, P. Lafourcade, G. Lenzini, and P. Y. A. Ryan. Formal analysis of electronic exams. In SECRYPT’14. SciTePress, 2014.

[2] P. Kudi, A. Manekar, K. Daware, and T. Dhatrak, “Online Examination with short text matching,” in Wireless Computing and Networking (GCWCN), 2014 IEEE Global Conference on, 2014, pp. 56–60.

[3] K. Woodford and P. Bancroft, “Multiple choice questions not considered harmful,” in Proceedings of the 7th Australasian conference on Computing education-Volume 42, 2005, pp. 109–116.

[4] X. Hu, and H. Xia," Automated Assessment System for Subjective Questions Based on LSI," Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, China, pp. 250-254, April 2010.

[5] M. S.Devi and H.Mittal,"Machine Learning Techniques With Ontology for Subjective Answer Evaluation," International Journal on Natural Language Computing, Vol. 5, No.2, April 2016.

[6] M.F. Al-Jouie, A.M. Azmi, "Automated Evaluation of School Children Essays in Arabic," 3rd International Conference on Arabic Computational Linguistics, 2017, vol.117, pp.19-22.

[7] A.Kashi, S.Shastri and A. R.Deshpande, "A Score Recommendation System Towards Automating Assessment In Professional Courses," 2016 IEEE Eighth International Conference on Technology for Education,2016,pp.140-143.

[8] T. Ishioka and M. Kameda, "Automated Japanese essay scoring system: Jess," in Proc. of the 15th Int'l Workshop on Database and Expert Systems Applications, 2004, pp. 4-8.

[9] K.Meena and R.Lawrance, "Evaluation of the Descriptive type answers using Hyperspace Analog to Language and Self-organizing Map", Proc. IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp.558-562.

[10] Vkumaran and A Sankar," Towards an automated system for short-answer assessment using ontology mapping," International Arab Journal of e-Technology, Vol. 4, No. 1, January 2015.

[11] R. Siddiqi and C. J. Harrison, "A systematic approach to the automated marking of short-answer questions," Proceedings of the 12th IEEE International Multi topic Conference (IEEE INMIC 2008), Karachi, Pakistan, pp. 329-332, 2008.

[12] M. J. A. Aziz, F. D. Ahmad, A. A. A. Ghani, and R. Mahmod, "Automated Marking System for Short Answer examination (AMSSAE)," in Industrial Electronics & Applications, 2009. ISIEA 2009. IEEE Symposium on, 2009, pp. 47-51.

[13] R. Li, Y.Zhu and Z.Wu,"A new algorithm to the automated assessment of the Chinese subjective answer," IEEE International Conference on Information Technology and Applications, pp.228 – 231, Chengdu, China, 16-17 Nov. 2013.

[14] X.Yaowen, L.Zhiping, L.Saidong and T.Guohua," The Design and Implementation of Subjective Questions Automatic Scoring Algorithm in Intelligent Tutoring System," 2nd International Symposium on Computer, Communication, Control and Automation, Vols. 347-350, pp. 2647-2650, 2013.

[15] Y. Zhenming, Z. Liang, and Z. Guohua, "A novel web-based online examination system for computer science education," in 2013 IEEE Frontiers in Education Conference (FIE), 2003, vol. 3, pp. S3F7–10.

AUTHORS

Alla Alrehily is a student in King Abdul-Aziz University, Saudi Arabia. She teaches master of Faculty of Computing and Information Technology. Her research interests intelligent information retrieval.

Muazzam Siddiqui is an Associate Professor at the Faculty of Computing and Information Technology, King Abdul-Aziz University. He received his BE in Electrical Engineering from NED University of Engineering and Technology, Pakistan, and MS in Computer Science and PhD in Modelling and Simulation from the University of Central Florida, USA. His research interests include sentiment analysis, named entity recognition, and keyword and relationship extraction.

Seyed Buhari is an Associate Professor in Faculty of Computing and Information Technology, King Abdul-Aziz University, Saudi Arabia. His current research interests are in the areas of cognitive radio networks, grid computing, IPv6 performance testing and high-performance computing.