This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082408, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Automatic Arabic Short Answers Scoring Using Longest Common Subsequence and Arabic WordNet

Hikmat A. Abdeljaber[1]

[1]Computer Science Department, Prince Sattam Bin Abdulaziz University, Kharj, B.O. Box 151, 11942 Saudi Arabia

Corresponding author: Hikmat A. Abdeljaber (e-mail: h.abdeljaber@psau.edu.sa).

ABSTRACT The manual process of scoring short answers of Arabic essay questions is exhaustive, susceptible to error and consumes instructor's time and resources. This paper explores longest common subsequence (LCS) algorithm as a string-based text similarity measure for effectively scoring short answers of Arabic essay questions. To achieve this effectiveness, the longest common subsequence is modified by developing weight-based measurement techniques and implemented along with using Arabic WordNet for scoring Arabic short answers. The experiments conducted on a dataset of 330 students' answers reported Root Mean Square Error (RMSE) value of 0.81 and Pearson correlation r value of 0.94. Findings based on experiments have shown improvements in the accuracy of performance of the proposed approach compared to similar studies. Moreover, the statistical analysis has shown that the proposed method scores students' answers similar to that of human estimator.

INDEX TERMS Arabic short answers scoring, Arabic WordNet, string-based text similarity, longest common subsequence, automatic essay scoring.

I. INTRODUCTION

Text classification is a significant field of natural language processing (NLP) [1]. It uses machine learning (ML) techniques and trained data to classify unseen test data [2,3]. Text similarity is an area of text classification that is represented by the distance or the degree of likelihood between two pieces of text [4]. Lexical similarity and semantic similarity are two adopted approaches used for accomplishing this task. Strings are similar lexically if they have a similar character or word sequence; the distance between them is computed based on character or word sequence [5]. On the other hand, strings are similar semantically if they have the same thing, are opposite of each other, used in the same way, used in the same context and one is a type of another; the distance between them is computed based on the likeness of their meaning [6]. In literature, lexical similarity is computed by using string-based measures such as longest common subsequence (LCS), cosine similarity and Jaccard similarity, whereas semantic similarity is computed by using corpus-based measures and knowledge-based measures such as latent semantic analysis (LSA), Disco and lesk [7].

Automatic essay scoring (AES) is one of the main applications of text similarity. AES systems are designed to score and evaluate student answers automatically based on predefined trained set of answer documents and often provide appropriate feedback and corrections for the assessment process [8]. AES systems reduce effort, time, cost of institution resources and achieve fairness in marking student answers compared to manual process. The automatic English short answer scoring has been studied for decades. These studies have produced substantial empirical systems for scoring student answers automatically such as IEA and C-rater [9]. A number of studies have been presented for scoring Arabic essays. Unfortunately, these studies have not produced empirical AES software systems for Arabic language. The research question of this work is how to develop an accurate and effective automated system for scoring short answers of Arabic essay questions. Adopting string-based lexical text similarity approach using longest common subsequence measure along with Arabic WordNet (AWN) is a promising solution for developing such system.

The objectives of this paper are: (1) introduce a model for scoring short answers of Arabic essay questions by incorporating a semantic resource into the syntactic analysis of the compared answers, (2) develop an effective automated system based on the introduced model by employing Arabic WordNet with a variant of longest common subsequence in similarity measurement process, (3) evaluate the effectiveness of the proposed system.

The longest common subsequence of strings is a common method of comparing strings [10]. The LCS of two strings is a subsequence that appears in both strings of maximal length [11]. The LCS has applications in many areas of computing, such as data compression [12], speech and signal processing [13], pattern recognition [14], spell checking [15], bioinformatics and computational biology [16], file comparison [17], computational linguistic analysis [18], combinatorial optimization [19] and text sentiment classification [20,21]. Different variants of LCS algorithm have been introduced in [22]. However, these variants have been used to score English text [23]. This work attempts to introduce an effective variant of LCS algorithm for more accurate measure of similarity on Arabic text.

The WordNet for English (PWN) is defined as a lexical database that contains synsets and relations organized in a hierarchy [24]. A synset is a group of synonym words that represent a sense/meaning. The words grouped in synsets represent one of the four types of parts of speech (POS): noun, verb, adjective, and adverb. Each synset has a gloss that describe its sense, and sometimes also usage examples [25]. For example, the noun "site" has three synsets (senses) in WordNet 3.1: {*site*, *land site*}, {*site*, *situation*}, and {*web site*, *website*, *internet site*, *site*}. The gloss of the first synset is defined as "the piece of land on which something is located (or is to be located)" and the usage example is given as "a good site for the school". The relations in WordNet are established between word forms and between word senses (synsets) [26]. So, two kinds of relations are recognized: lexical and semantic. Lexical relations such as antonymy and synonymy hold between word forms. For example, *hide*/*show* are antonyms and *perform*/*execute* are synonyms. In contrast, semantic relations hold between word senses (synsets). Relations between senses include synonymy, antonymy, meronymy, and taxonomic relations hyponymy and hypernymy [27]. For example, {*tree*} is a hyponym of {*plant*} and {*plant*} is a hypernym of {*tree*}. The PWN is also extended for other languages including Arabic [28]. The Arabic WordNet is used in this study as a standard resource for providing synonyms of student answer words without changing their meanings, hence improves the accuracy of similarity between student answer and model answer.

The contribution of this paper is three fold. First, the improvement of the original longest common subsequence method by developing new weight-based measurement and normalization techniques. Second, the development of an effective automatic essay scoring for Arabic useful for educational sectors. Third, the employment of Arabic WordNet in Arabic NLP applications.

This paper is structured as follows. Section 2 reviews the related work. Section 3 presents Arabic WordNet for improving the accuracy of text similarity. Section 4 explains the proposed approach for measuring the similarity between model answers and student answers. Section 5 highlights on the conducted experiments and their results. Finally, conclusion is given.

II. RELATED WORK

A limited number of surveys are carried out on Arabic text classification techniques compared to English [29]. In addition, few studies have reviewed the approaches used for measuring Arabic text similarity [4,6]. Text similarity approaches have been applied to different applications including AES. State-of-the-art of AES for English is found in [30], but it is not presented for Arabic.

Besides English, many overseas languages have endeavored to develop AES systems including Chinese [31], Punjabi [32], Swedish [33], Bahasa [34], Korean [35], and Arabic [36]. As for Arabic language, diverse approaches and techniques have been examined including a hybrid approach combining string-based, corpus-based and knowledge-based of text similarity measures [37], hybrid approach of LSA and POS tagging of syntactic analysis [38], cosine similarity [39], string-based (N-gram and Damera-levenshtein) and corpus-based (LSA and DISCO2) of text similarity measures [36], vector space model and latent semantic indexing [40], and a combination of LSA, writing style and spelling errors [41].

The study in [38] proposed a hybrid method of syntactic features and LSA for improving the accuracy of Arabic AES. The syntactic features are identified by using POS tagging through transforming term frequency-inverse document frequency (TF-IDF) into term frequency-part of speech (TF-POS) and LSA is used to identify the semantic similarity between student answer and model answer. The authors have constructed a synonym dictionary using Arabic WordNet to replace all words of student answers by their corresponding synonyms. The experiments are performed on the same dataset that introduced by [37]. The authors reported a Root Mean Square Error (RMSE) of 0.268 compared to 0.745 reported by [37].

Shehab et al. [36] experimented two string-based and two corpus-based text similarity measures separately for scoring 210 Arabic students' answers contained in an in-house dataset. The authors reported that character-based N-gram algorithm achieved better results in terms of the correlation r (0.82) than the other three types: Damera-levenshtein, LSA, and DISCO2. However, their work neither combined string and corpus algorithms nor applied semantic analysis through using WordNet for extracting synonyms that could increase the correlation r result.

In a recent work, [41] proposed a hybrid approach to score Arabic essays that combines LSA, writing style, spelling mistakes and some other lexical features. The system was tested using a dataset of 350 Arabic essays collected by school children. The best accuracy of 0.9 and r of 0.756 were reported. The accuracy value is relatively high may be due to that "exact" score and "within range" score of auto-score are considered correct scores. The auto-score is considered “exact” if the difference between auto-score and actual-score ranges from 0 to 0.5 and “within range” if this difference ranges from 0.5 to 2.5.

Longest common subsequence algorithm has extensive applications in diverse areas ranging from computational linguistics to molecular biology. This involved automatic scoring of English short answers. In contrast, the literature is lacking in investigating LCS for scoring Arabic short answers. The researches in [37, 42] are the only works that have applied LCS for automatic scoring of Arabic short answers. These two works are of the same authors and they did not incorporate Arabic WordNet as a standard resource for providing synonyms of student answer words.

The authors in [37] applied a hybrid approach by combining multiple similarity measures including longest common subsequence. The system was tested using a dataset of 61 questions with 10 answers for each, pertaining to an official Egyptian curriculum for a course on environmental science. The answers are translated into English to overcome the lack of text processing resources in Arabic. The authors reported r of 0.83 and RMSE of 0.75. As for LCS metric, the values of r and RMSE were 0.49 and 1.18 respectively.

In [42], a combination of string-based and corpus-based similarity measures is implemented for scoring Arabic short answers with feedback. The system was tested on the same dataset used in [37]. The authors reported r of 0.86 and RMSE of 0.76. As for LCS metric, the values of r and RMSE were 0.53 and 1.22 respectively.

In automated essay scoring systems, when the students answer the questions, they may use words different in forms from the words given by the instructor in the model answer. Though, they are different in forms, they may have exactly the same sense. These words are called synonyms. Arabic WordNet can play a significant role in these cases by replacing the synonyms of student answer words when compared to model answer without changing the meaning of student answer. Synonyms replacement can further increase the similarity between student answer and model answer. So incorporating Arabic WordNet with AES could be a promising direction for improving the performance of sentence similarity measures.

So far, the study presented in [43] was the only work that employed Arabic WordNet in Arabic AES systems. The aim was to enhance the system’s accuracy by replacing the synonyms of student answer words. The authors used f-score to add the selected features in feature space and they applied cosine similarity to measure the similarity between student answer and model answer. An in-house dataset containing 120 questions and three model answers for each question is collected by school children and tested on 30 students. A comparative analysis of the impact of using AWN is carried out. The authors reported r ranges 0.5-1 and Mean Absolute Error (MAE) of 0.117 and they concluded that using AWN increases the accuracy of text similarity.

In literature, no previous work has used a combination of longest common subsequence and Arabic WordNet for scoring Arabic short answers. This limitation is the motivation of this paper. Moreover, the original LCS is modified by developing weight-based measurement and normalization techniques, which are significant to enhance text classification performance [2].

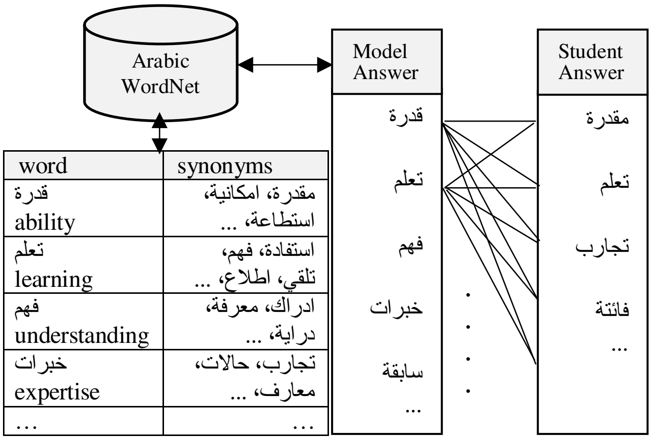
Although the literature in text similarity area suggested various methods for addressing the problem of scoring short answers of Arabic essay questions, longest common subsequence method is adopted in this work as a promising key solution for the following justifications. First, the related work given above points out that the string-based text similarity area for Arabic language has used the original LCS method for addressing the problem of scoring Arabic essays while variations of LCS method are not examined. Investigating effective variants of LCS method can provide robust solution and significant findings as shown by the experimental results (see section V). Second, no existing related work combines both LCS method and Arabic WordNet for scoring Arabic essays. Experiment findings show that incorporating Arabic WordNet with an effective variant of LCS is a new promising direction towards scoring Arabic essays. Third, the time complexity of LCS method is $O(m.n)$ where m and n are the lengths (number of words) of model answer and student answer respectively (see section IV). This moderate polynomial complexity can be improved further to become algorithmic function $O(m.log(n)^2)$ as proposed by [44] where m is the length of largest answer and n is the length of smallest answer between student answer and model answer. This improvement increases the efficiency of LCS method. Fourth, LCS method is simple to implement and efficient compared to many other methods because it does not examine syntactic or semantic analysis of Arabic essays.

III. ARABIC WORDNET

Arabic WordNet is constructed for utilizing it in developing Arabic NLP applications such as Question-Answering, Query Expansion and AES systems. In fact, more than one release of AWN was available. Yet, the gap between AWN and other similar WordNets in terms of the coverage of Arabic language remains one of the limitations of its usage [45]. This limitation was the motivation for [45] to enrich AWN content with more words and synsets. The authors presented the new content added to AWN by using semi-automated techniques and validated by lexicographers. The enriched AWN is transformed into Lexical Markup Framework (LMF) format to make it a standard public available resource.

In this paper, Arabic WordNet plays a vital role for improving the accuracy of system’s scoring result. It is used as a resource for providing the synonyms of model answer word. Matching any of these synonyms against student answer word during the comparison process means that student answer word is a synonym of model answer word. At this point, student answer word is replaced by model answer word without affecting its meaning. Figure 1 depicts the task of mapping model answer word into Arabic WordNet for obtaining its synonyms and the replacement process is taken place when student answer word is one of these synonyms.

Arabic is a complex language in terms of morphology; the study of word structure. Sometimes words consist of solid stem such as the noun "فهم - fahm" (understanding), but more often words are composed of more than one morpheme such as the word "مقدرة - maqdirah" (capability) [46].



word	synonyms
قدرة ability	مقدرة، امكانية، استطاعة، ...
تعلم learning	استفادة، فهم، تلقي، اطلاع، ...
فهم understanding	ادراك، معرفة، دراية، ...
خبرات expertise	تجارب، حالات، معارف، ...
...	...

FIGURE 1. Mapping model answer words into Arabic WordNet for obtaining their synonyms and replacing them for student answer words.

Another issue is that student answer and model answer may contain synonym words in common written in different morphological or broken plural forms. For example, assume that the plural word "تجارب - tajarup" (experiences) appears in student answer and the plural word "خبرات - khibraat" (expertise) appears in model answer as shown in Fig. 1. Since AWN is limited in coverage [45], it may cover the singular form of these two words but not their plural form. Since LCS is string-based measure, the LCS-based scoring system will not consider them synonyms though they are so. Consequently, this failure results in decreasing system's accuracy.

To tackle these issues and many others, stemming is necessary. The stemming process is taken place in the preprocessing phase before applying LCS algorithm. In particular, when comparing student answer words with model answer words. The stemming process occurs when the compared words have different forms and they are not direct synonyms of each other. In this case, stem of student answer word and stem of model answer word are provided. For example, stem of plural word "تجارب" (i.e. "جرب") and stem of the plural word "خبرات" (i.e. "خبر") should be provided to examine whether they are synonyms.

There are two approaches to examine whether the plural words of given stems are synonyms. The first approach is to construct a lookup table involving all model answer words. Each model answer word lists its synonyms and their stems as given in Table. I. If any of these stems for a particular model answer word matches lexically the stem of student answer word, then the plural of model answer word and the plural of student answer word are synonyms. For example, stem "جرب" of the plural word "تجارب" (experiences) matches one of the stems of synonyms of the plural word "خبرات" (expertise). Thus, both plural words are synonyms.

TABLE I
SYNONYMS AND THEIR STEMS OF SOME MODEL ANSWER PLURAL WORDS.

MA Word	Syn.	Stem	Syn.	Stem	Syn.	Stem	...
خبرات	تجارب	جرب	دراية	دري	ممارسة	مرس	...
مسائل	موضوع	وضع	مشكلة	شكل	قضية	قضي	...
...	...	...	...	...	...	...	...

The second approach which is adopted by this work is to obtain the stem of student answer word and the stem of model answer word from AWN. If these two stems have a common AWN synset, then both student answer word and model answer word are synonyms. For example, stem "جرب" of the plural word "تجارب" (experiences) and stem "خبر" of the plural word "خبرات" (expertise) have the common synset {*معرفة*، *تجربة*، *خبرة*}, i.e. {*expertise*, *experience*, *knowledge*}. Thus, both plural words are synonyms.

IV. LONGEST COMMON CONTIGUOUS SUBSEQUENCE

Typically, AES composes of three major components as shown in Fig. 2. Preprocessing component handles text tokenization and removal of special characters and stop words. Text classification component is responsible for implementing the approach used for scoring short answers of Arabic essay questions. Evaluation component is responsible for measuring the accuracy of performance of the system.

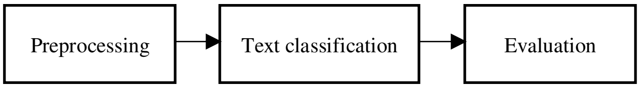


FIGURE 2. The typical components of automatic essay scoring systems.

One of the main contributions of this work is to use Arabic WordNet as a resource for retrieving synonyms of model answer words in order to improve text similarity between student answer and model answer. Note that stemming process is included in synonyms retrieval step. This step is incorporated in preprocessing phase before implementing text classification. Text classification is implemented by applying a suggested variant of LCS algorithm which is LCS-based text classifier enhanced by developing new term weighting measurement and scaling techniques. Figure 3 shows the components of the proposed AES system for Arabic.

Longest common subsequence is a character-based similarity metric that computes the similarity between two strings based on the length of the longest contiguous chain of characters that exists in both strings. Essentially, the LCS method for measuring the similarity between two strings $S1$ and $S2$ (i.e. $Sim_{LCS}(S1, S2)$) is computed by using (1).

$$Sim_{LCS}(S1,S2)=\frac{2\times LCS_{length}(S1,S2)}{|S1|+|S2|} \qquad (1)$$

where $LCS_{length}(S1,S2)$ is the length (number of words) of longest common subsequence of two strings $S1$ and $S2$, $|S1|$ is number of words in $S1$ and $|S2|$ is number of words in $S2$.

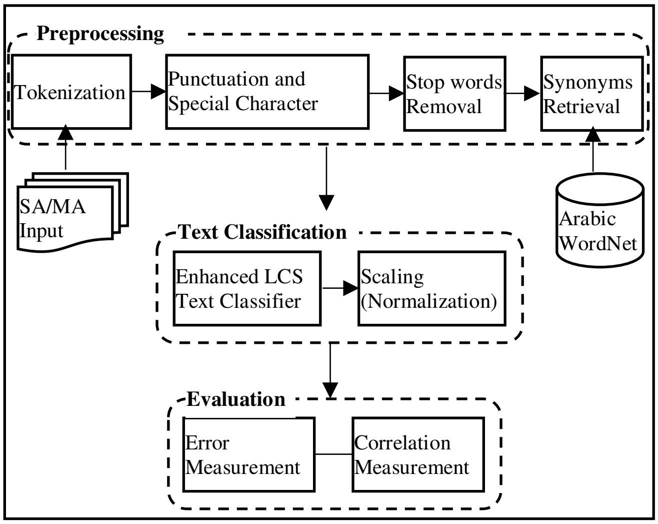


FIGURE 3. The components of the proposed automatic essay scoring system for Arabic.

However, LCS approach has many drawbacks. One of these drawbacks is that the position of an error character typed by mistake affects the computed similarity. For example, suppose the word "القدس" (Jerusalem) is typed by mistake as "الحدس" (intuition), where the character 'ق' is typed by mistake as the character 'ح', then the LCS similarity between "القدس" and "الحدس" becomes less than the LCS similarity between "القدس" and a semantically different word such as "القدر" (destiny).

Another drawback is that the similarity between two strings S1 and S2 may give a value greater than zero, though S1 and S2 contain different words. For example, the two texts S1="القدس عاصمة فلسطين" and S2="القديس جورج من الفلبين" have a similarity of length 4 (computed between the words "القدس" and "القديس") though they are different words and have no common meaning. This work performs LCS to compute the similarity between two texts rather than two character strings. One of these texts is a document represents the model answer of a given essay question and the other text is a document represents the student answer of the given essay question. So, model answer and student answer are documents of a sequence of words. The LCS computes the similarity between student answer and model answer by returning the maximum number of words common to student answer and model answer and in order. The following pseudo code can be defined to calculate length of the LCS between student answer (SA) and model answer (MA):

```
if (SA[i] == MA[j])
        LCS[i,j] = LCS[i-1, j-1] +1
else
        LCS[i,j] = max(LCS[i-1,j], LCS[i,j-1])
```

where i=1,2,...,n, j=1,2,…,m, SA[i] is the ith word of student answer text and MA[j] is the jth word of model answer text. LCS[i,j] is the LCS at the ith word of student answer text and the jth word of model answer text. In other words, LCS[i,j] is the number of common words between student answer and model answer.

The longest common subsequence problem can be viewed as a brute-force search problem in which all student answer subsequences of words from 1 to n must be generated and then each subsequence must be examined against model answer to find the longest common subsequence of both answers. From theory of permutation and combination, the student answer text of length n words has $2^n - 1$ different possible subsequences since the subsequence of length 0 is not considered. In addition, it takes $O(m)$ time to check if a subsequence is common to both answers, where m is the number of words in model answer. Thus, the overall complexity of the brute-force algorithm for solving LCS is $O(m.2^n) \approx O(2^n)$, which is an exponential in term of time complexity.

TABLE II
m×n MATRIX FOR COMPUTING LONGEST COMMON SUBSEQUENCE LENGTH BETWEEN MODEL ANSWER REPRESENTED BY M ROWS AND STUDENT ANSWER REPRESENTED BY N COLUMNS BY USING DYNAMIC PROGRAMMING APPROACH.

	0	1 قدرة	2 تعلم	3 اكتساب	4 معرفة	5 قدرة	6 استنتاج	7 منطقي	8 حل	9 مسائل
0	0	0	0	0	0	0	0	0	0	0
1 قدرة	0	1	1	1	1	1	1	1	1	1
2 تعلم	0	1	2	2	2	2	2	2	2	2
3 فهم	0	1	2	2	2	2	2	2	2	2
4 خبرات	0	1	2	2	2	2	2	2	2	2
5 سابقة	0	1	2	2	2	2	2	2	2	2
6 قدرة	0	1	2	2	2	3	3	3	3	3
7 اكتساب	0	1	2	3	3	3	3	3	3	3
8 معرفة	0	1	2	3	4	4	4	4	4	4
9 احتفاظ	0	1	2	3	4	4	4	4	4	4
10 استخدام	0	1	2	3	4	4	4	4	4	4
11 قدرة	0	1	2	3	4	5	5	5	5	5
12 تفكير	0	1	2	3	4	5	5	5	5	5
13 استنتاج	0	1	2	3	4	5	6	6	6	6
14 منطقي	0	1	2	3	4	5	6	7	7	7
15 حل	0	1	2	3	4	5	6	7	8	8
16 مسائل	0	1	2	3	4	5	6	7	8	9

The longest common subsequence problem can be solved by breaking it down into sub-problems since it has an optimal substructure property. The brute-force algorithm utilizes recursion and dynamic programming approaches for implementing LCS problem. The implementation of LCS using recursive approach has an exponential running time of $O(2^{m+n})$ in worst case. Worst case happens when all words of student answer and model answer are mismatch. The recursive approach takes exponential running time because it runs many overlapping sub-problems that are exhibited by the LCS method. However, time complexity can be improved by using dynamic programming approach which avoids running the overlapping sub-problems. The time complexity of dynamic programming implementation of LCS problem is $O(m.n)$, where m and n are the lengths of words of model answer and student answer respectively. This approach is adopted for solving the proposed method in a polynomial time as depicted in Table II.

Although there exist more efficient solutions for the LCS problem, however, they are approximate solutions, and using them might not give the same results that exact solution used in this paper might give. Notwithstanding, investigating such issues is out the scope of this paper.

Let us give an example to illustrate how LCS works on Arabic text. Suppose the posed question is “عرف الذكاء” (define intelligence). Let the model answer (MA) is given as “قدرة تعلم فهم خبرات سابقة قدرة اكتساب معرفة احتفاظ استخدام قدرة تفكير استنتاج منطقي حل مسائل” and student answer (SA) is given as “قدرة تعلم كسب معرفة مقدرة استنتاج منطقي حل مسائل”. For computing the similarity between student answer and model answer, a matrix of m×n is constructed as shown in Table II. Where m is the length of model answer (number of words in model answer) and n is the length of student answer (number of words in student answer).

Note from Table II that the synonyms retrieval step of preprocessing phase has replaced the student answer word “مقدرة” (capability) by the corresponding model answer word “قدرة” (ability) since they are synonyms and replaced the student answer word “كسب” (acquire) by the corresponding model answer word “اكتساب” (acquisition) since they both have the same stem (root).

Table II shows that there are four common contiguous subsequences of words between student answer and model answer (highlighted with yellow color). These subsequences are listed as follows:

- subsequence 1: “قدرة تعلم”
- subsequence 2: “اكتساب معرفة”
- subsequence 3: “قدرة”
- subsequence 4: “استنتاج منطقي حل مسائل”

It is clear that the subsequence 4 of length 4 is the longest contiguous chain of words that exists in both student answer and model answer. The similarity between student answer (SA) and model answer (MA) (i.e. $Sim_{LCS}(SA, MA)$) is computed by using (1) as follows:

$$Sim_{LCS}\,(SA, MA) = \frac{2 \times LCS_{length}(SA, MA)}{|SA| + |MA|} = \frac{2 \times 4}{9 + 16} = 0.32$$

where $LCS_{length}\ (SA, MA)$ is the length (number of words) of longest common subsequence of SA and MA, $|SA|$ is number of words in SA and $|MA|$ is number of words in MA.

Indeed, (1) is not a reliable or accurate measurement formula for computing the similarity between two texts. For some situations, (1) computes the same LCS similarity score for two different answers, though one answer is syntactically closer to the model answer than the other. For example, consider that the following two student answers are given where SA_1 is the first student answer and SA_2 is the second student answer:

SA_1: “قدرة تعلم اكتساب معرفة قدرة استنتاج منطقي حل مسائل”

SA_2: “قدرة تعلم فهم قدرة اكتساب معرفة استخدام استنتاج منطقي حل مسائل”

then, the LCS similarity between SA_1 and MA is the same as the LCS similarity between SA_2 and MA, which is 4, though SA_2 is syntactically more similar to model answer than SA_1.

To improve the accuracy of similarity between two texts, a new measurement technique based on weights of all common contiguous subsequences is needed. The LCS similarity measure can be modified by taking into account not only contiguity of longest common subsequence of two texts but contiguity of all common subsequences of these texts. This modification changes LCS to become LCCS (longest common contiguous subsequence). The similarity between student answer (SA) and model answer (MA) using LCCS is calculated by using (2).

$$Sim_{LCCS}(SA, MA) = \sum_{i=1}^{k} (w[i] \times log(w[i])) \qquad (2)$$

where $Sim_{LCCS}(SA, MA)$ is the similarity between SA and MA using LCCS (i.e. the similarity between SA and MA using contiguity values of all common subsequences of SA and MA including longest common subsequence), $w[i]$ is the length (number of words) of common contiguous subsequence i of SA and MA and k is the total number of common contiguous subsequences of SA and MA. Applying (2) on the two texts SA_1 and MA and on the two texts SA_2 and MA (SA_1, SA_2 and MA are given above) gives the following results:
The lengths of common contiguous subsequences of SA_1 and MA are [2,2,1,4] and the lengths of common contiguous subsequences of SA_2 and MA are [3,3,1,4], and therefore:

$Sim_{LCCS}(SA_1,MA) = 2\times log(2) + 2\times log(2) + 1\times log(1) + 4\times log(4) = 3.612$

$Sim_{LCCS}(SA_2,MA) = 3\times log(3) + 3\times log(3) + 1\times log(1) + 4\times log(4) = 5.271$

Now, it is obvious that the values obtained from (2) reflect more precisely the text similarity of each student answer to model answer; in this case SA_2 is more similar to model answer than SA_1. However, in order to change or adjust the range of text similarity scores obtained by (2) to be in the range from 0 to 1, scaling (normalization) technique formulated in (3) is applied.

$$Sim_{LCCS(Norm)}(SA,MA)=\frac{|SA\cap MA|}{2|MA|}+\frac{Sim_{LCCS}(SA,MA)}{2|MA|\times\log(|MA|)} \qquad (3)$$

where $Sim_{LCCS(Norm)}(SA,MA)$ is the normalized LCCS similarity value between student answer (SA) and model answer (MA), $|SA\cap MA|$ is the length or total number of common words between SA and MA, $Sim_{LCCS}(SA,MA)$ is the value of LCCS similarity between SA and MA calculated by using (2) and $|MA|$is the length of MA (total number of words in MA). Applying (3) on SA$_1$ and MA and on SA$_2$ and MA (SA$_1$, SA$_2$ and MA are given above) gives the following results:

$$Sim_{LCCS(Norm)}(SA_1,MA)=\frac{9}{2\times16}+\frac{3.612}{2\times16\times\log(16)}=0.375$$

$$Sim_{LCCS(Norm)}(SA_2,MA)=\frac{11}{2\times16}+\frac{5.271}{2\times16\times\log(16)}=0.481$$

For further improvement of text similarity result, another measurement technique based on the normalized LCCS similarity metric is developed. The improvement is achieved by computing the *log* of normalized LCCS similarity value after converting it to a number ranges from 1 to 10. This conversion is done by multiplying normalized LCCS similarity value by 10 as shown in (4). Taking *log* of numbers within the range [1-10] guarantees the results to be within the range [0-1].

$$Sim_{LCCS(Enhanced)}(SA,MA) = \log(10\times Sim_{LCCS(Norm)}(SA,MA)) \qquad (4)$$

where $Sim_{LCCS(Enhanced)}(SA,MA)$ is the final enhanced version of normalized LCCS method for measuring the similarity between SA and MA and $Sim_{LCCS(Norm)}(SA,MA)$ is the value of normalized LCCS similarity between SA and MA calculated by using (3). Now, applying (4) on SA$_1$ and MA and on SA$_2$ and MA, gives the following results:

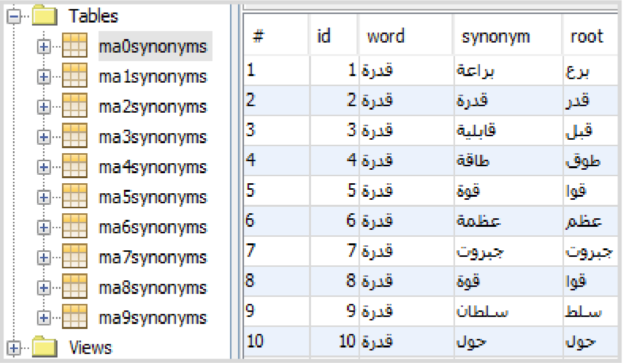
$$Sim_{LCCS(Enhanced)}(SA_1,MA) = \log(10\times0.375)=0.574$$

$$Sim_{LCCS(Enhanced)}(SA_2,MA) = \log(10\times0.481)=0.682$$

As seen, LCS method per se doesn't reflect precisely the extent of text similarity. Adopting Arabic WordNet for capturing synonyms of student answer words and using an enhanced version of LCS can play critical role for narrowing the gap between manual and automatic short answer scoring. Hence improves the accuracy of text similarity.

V. EXPERIMENTAL RESULTS

A new variant of LCS method for scoring Arabic short answers is developed. The development is implemented in Java and MySQL database is created by using JDBC for maintaining synonyms of model answers as shown in Fig 4. The implementation details and experiment results and evaluation are discussed in the following subsections.

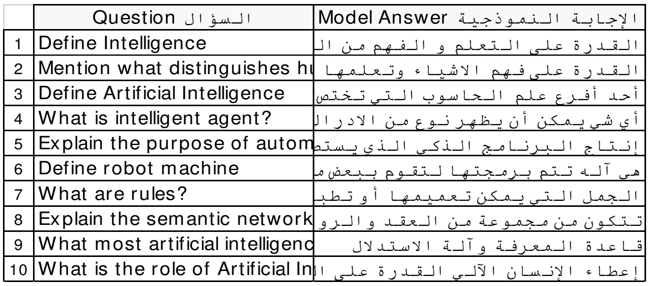


#	id	word	synonym	root
1	1	قدرة	براعة	برع
2	2	قدرة	قدرة	قدر
3	3	قدرة	قابلية	قبل
4	4	قدرة	طاقة	طوق
5	5	قدرة	قوة	قوا
6	6	قدرة	عظمة	عظم
7	7	قدرة	جبروت	جبروت
8	8	قدرة	قوة	قوا
9	9	قدرة	سلطان	سلط
10	10	قدرة	حول	حول

FIGURE 4. MySQL tables for maintaining synonyms of model answers.

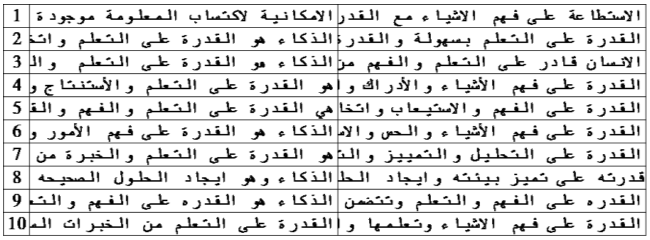
A. DATASET

To evaluate the performance of the proposed system, an Arabic dataset [1]is constructed. A human expert has designed 10 basic questions and their corresponding model answers pertaining to an official curriculum for a course on Artificial Intelligence (AI) as shown in Fig. 5.



	Question السؤال	Model Answer الإجابة النموذجية
1	Define Intelligence	القدرة على التعلم و الفهم من ال
2	Mention what distinguishes hu	القدرة على فهم الاشياء وتعلمها
3	Define Artificial Intelligence	أحد أفرع علم الحاسوب التي تختص
4	What is intelligent agent?	أي شي يمكن أن يظهر نوع من الادرال
5	Explain the purpose of autom	إنتاج البرنامج الذكي الذي يستط
6	Define robot machine	هى آله تتم برمجتها لتقوم ببعض م
7	What are rules?	الجمل التي يمكن تعميمها أو تطب
8	Explain the semantic network	تتكون من مجموعة من العقد و الرو
9	What most artificial intelligenc	قاعدة المعرفة و آلة الاستدلال
10	What is the role of Artificial In	إعطاء الإنسان الآلي القدرة على ا

FIGURE 5. The 10 basic questions of AI course and their model answers.



1	الامكانية لاكتساب المعلومة موجودة	الاستطاعة على فهم الاشياء مع القدر
2	الذكاء هو القدرة على التعلم و اتخ	القدرة على التعلم بسهولة و القدر
3	الذكاء هو القدرة على التعلم و ال	الانسان قادر على التعلم و الفهم من
4	هو القدرة على التعلم و الأستنتاج و	القدرة على فهم الأشياء و الأدراك و ا
5	هي القدرة على التعلم و الفهم و الق	القدرة على الفهم و الاستيعاب و اتخا
6	الذكاء هو القدرة على فهم الأمور و	القدرة على فهم الأشياء و الحس و الا
7	هو القدرة على التعلم و الخبرة من	القدرة على التحليل و التمييز و الت
8	الذكاء وهو ايجاد الحلول الصحيحه	قدرته على تميز بيئته وايجاد الحل
9	الذكاء هو القدره على الفهم و الت	القدره على الفهم و التعلم وتتضمن
10	القدرة على التعلم من الخبرات الم	القدرة على فهم الاشياء وتعلمها و ا

FIGURE 6. The first 2 answers of the first 10 students of the dataset.

The expert is the instructor of AI course at computer science department in the college of computer engineering and science at Prince Sattam Bin Abdulaziz University in KSA. Research data were obtained from a total of 33 volunteer students who were taking AI course at the same department. Each student

[1] Found at https://github.com/hikmatabdeljaber/Arabic-Essay-Scoring

was asked to answer 10 questions. The overall dataset comprises of 330 student answers all are maintained in an Excel sheet where part of it is shown in Fig. 6.

The dataset has been evaluated manually by two human annotators. The annotators are instructors at CS department who are also specialists in AI area. For each annotator, the 10 answers of each student are scored with values range from 0 to 10 and averaged. In this way, two scores are obtained for each student; one score from the first annotator and the other score from the second annotator. These two scores are then averaged to determine the final human score for each student. Table III shows the scores assigned by the two annotators and their average for answers of each student.

The entire dataset of 330 answers is divided into two sets: training dataset of size 0.8 (265 answers) and test dataset of size 0.2 (66 answers). The training dataset is used to prepare the model (i.e. to train it), while test dataset is used to predict the unseen examples. The model is trained by extracting the features of student answers as a bag of words and replacing them by model answer words once they are synonyms.

TABLE III
THE AVERAGE SCORE OF TWO HUMAN ANNOTATORS FOR EACH STUDENT ANSWER.

Student#	1st Annotator Score	2nd Annotator Score	Average
1	7.1	7.0	7.05
2	3.8	4.5	4.15
3	7.4	7.7	7.55
4	5.8	6.9	6.35
5	5.9	6.9	6.40
6	7.7	8.2	7.95
…	…	…	…
32	9.8	8.7	9.25
33	8.9	8.1	8.50

B. ARABIC WORDNET

Arabic WordNet is used in experiments as a thesaurus to find the synonyms of model answers. These synonyms played a significant role in replacement of student answers words by model answer words when student answers words match one of these synonyms. For example, if student answer is given as "قاعدة المعرفة وآلة الاستنباط" (knowledgebase and deduction engine) whereas the model answer is "قاعدة المعرفة وآلة الاستدلال" (knowledgebase and inference engine) and the word "الاستنباط" (deduction) is one of the synonyms of the model answer word "الاستدلال" (inference), then the student answer word "الاستنباط" (deduction) would be replaced by the model answer word "الاستدلال" (inference). This process helps in narrowing the lexical difference between student answers and model answers and hence increasing their text similarity.

Arabic WordNet has lack of finding all direct synonyms of model answer words. For example, AWN does not provide the word "مقدرة" (capability) as a direct synonym of the word "قدرة" (ability), though they are so. This limitation was the motivation for involving the stemming process in implementation. The process is included in synonyms retrieval step of preprocessing phase. For this reason, stems (roots) of synonyms of model answer words are extracted by using Khoja stemmer and they are maintained in MySQL database tables as shown in Fig. 4.

Stemming process is taken place when both compared words are not matched lexically and not direct synonyms of each other. To this end, stems of both words are examined to check whether they are identical. If they are so, then they are synonyms and hence student answer word will be replaced by model answer word. For example, even though they are not captured by AWN as synonyms, student answer word "مقدرة" (capability) is replaced by model answer word "قدرة" (ability) since both words have the same stem ("قدر").

TABLE IV
CONFUSION MATRIX FOR CALCULATING PRECISION, RECALL AND F-SCORE.

		number of student answers (actual)	
		correct	incorrect
number of student answers (predicted)	correct	TP	FP
	incorrect	FN	TN

To examine how Arabic WordNet as a synonym dictionary improves the scoring result, performance of the system without using AWN is compared against performance of the system with using AWN. Precision, recall and f-score are often used as evaluation criteria in measuring system's performance [47]. To get values of precision, recall and f-score, a confusion matrix given in Table IV and (5), (6) and (7) are used.

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

In order to measure precision, recall and f-score, all student answers whether scored by human (actual) or automatically (predicted) need to be classified as correct or incorrect. Student answer is classified correct if the difference between the maximum score of the question and the score assessed by the system is less than or equal to some value ε [48]; otherwise it is classified incorrect. In experiments, the maximum score of each student answer is set to 10 and ε is set 5. This means that the answer is correct if the difference between 10 and the answer score is less than or equal to 5; otherwise it is considered incorrect.

The predicted results which are evaluated by the system using LCCS are obtained by using two means. First, without using AWN as a synonym dictionary and hence replacement of synonyms of student answers is not taken place. In this case,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082408, IEEE Access

student answer which has words different from model answer words is considered incorrect, even if words of both answers are synonyms. Second, with using AWN as a synonym dictionary where replacement of synonyms of student answers is taken place. In this case, student answer which has words different from model answer words is considered correct if words of both answers are synonyms. Figure 7 compares precision, recall and f-score values calculated without using AWN against precision, recall and f-score values calculated with using AWN for all 10 questions. Notice that the implementation of LCCS method with using AWN gives higher precision, recall and f-score than the implementation of LCCS method without using AWN. The obtained results prove that using AWN with LCCS method in scoring process enhance performance of the system.

C. PERFORMANCE EVALUATION

The proposed automatic Arabic essay scoring system consists of three phases: preprocessing, text classification and evaluation, as shown in Fig 3. The operations of preprocessing include tokenization, punctuation and special character removal, stop words removal, lemmatization and synonyms retrieval. All these operations are implemented in Java and applied on both student answers and model answers. The key idea of this phase is to make use of AWN for alleviating the lexical dissimilarity between student answer and its corresponding model answer through replacing model answer words for student answer words.

TABLE V
ACTUAL SCORES VS. PREDICTED SCORES.

Student#	Actual Scores	Predicted Scores
1	7.05	7.03
2	4.15	3.39
3	7.55	7.60
4	6.35	7.05
5	6.40	5.97
6	7.95	7.60
7	4.80	5.29
8	5.15	5.18
…	…	…

The text classification phase of the proposed system performs the enhanced variant of LCS text classifier and uses it as a predictive model. It performs two main operations: the LCCS algorithm and scaling (normalization) of contiguousness values of LCCS. The LCCS is an enhanced variant of LCS algorithm whereas scaling is the task responsible for normalizing or tuning the contiguousness values (text similarity scores) resulted from LCCS to be in the range from 0 to 1. These two operations are implemented in Python. The LCCS algorithm is trained and the final model is used to predict the scores of student answers. Table V lists part of auto generated predicted scores corresponding to their actual scores of student answers. Table V shows that predicted scores are close to actual scores. This closeness is visualized in Fig. 8 for each student.

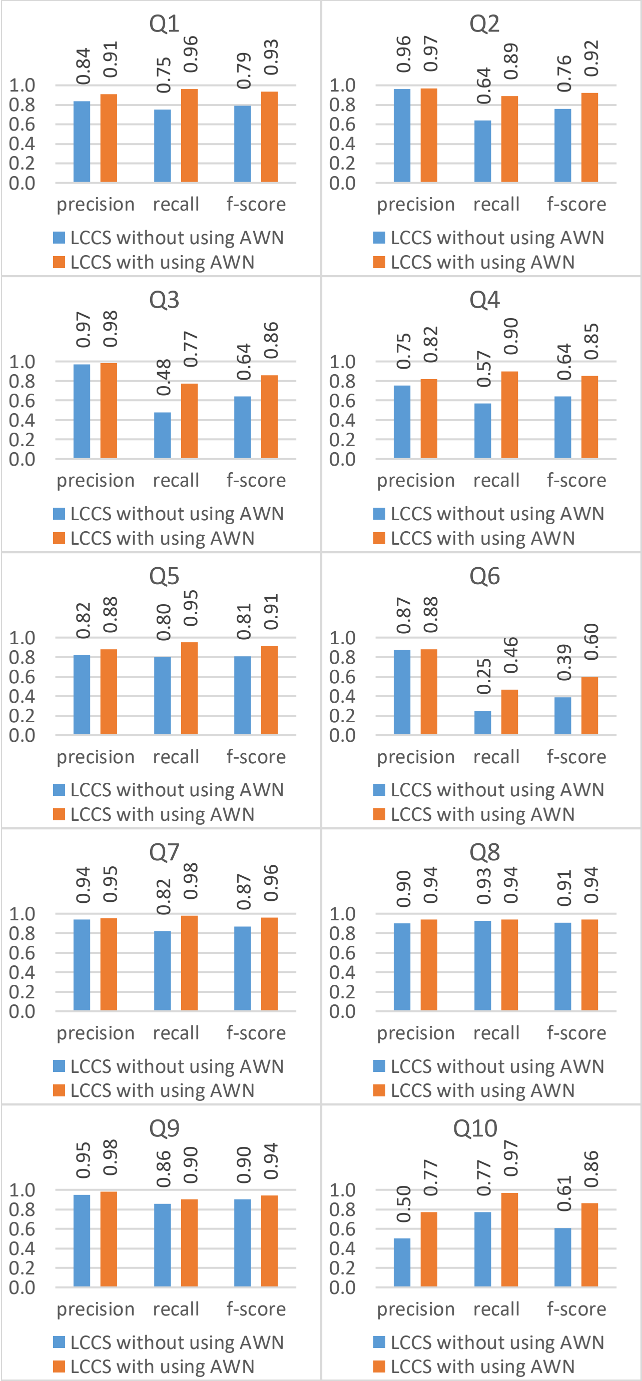


FIGURE 7. **Precision, recall and f-score values calculated without using AWN compared against precision, recall and f-score values calculated with using AWN for all 10 questions.**

The evaluation phase of the proposed system is responsible for evaluating the performance of the model. Often, regression accuracy is used to measure the performance of regression models. For evaluating the accuracy of performance of the proposed model, root mean square error (RMSE) and Pearson correlation coefficient r are selected. The main reason of selecting these two metrics is to compare the performance of the proposed model with the performance of models in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082408, IEEE Access

[34,39]. These two research works are similar to this present work and they have used the same metrics.

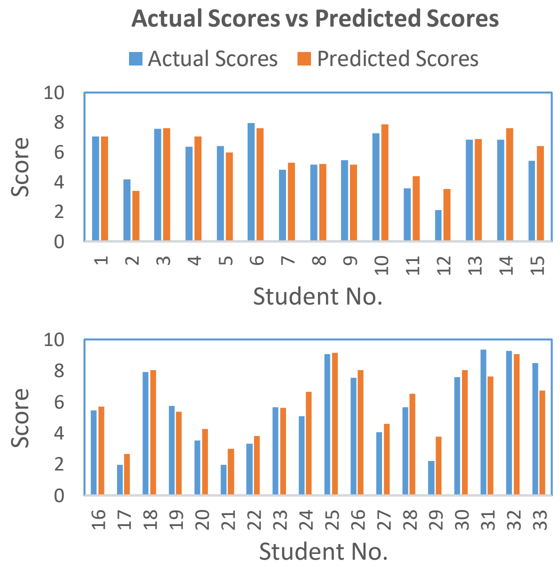


FIGURE 8. Comparison between actual scores and predicted scores for all 33 students.

The RMSE is perhaps the most common error metric used to estimate the performance of regression models. Other error metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). RMSE is a frequently used measure of the difference between actual scores and model predicted scores. It serves to aggregate residuals into a single measure of predictive power. RMSE measures how much error there is between two data sets, actual scores and predicted scores. This error value is always non-negative and values close to 0 are better. However, the error values in the proposed model always range from 0 to 10. RMSE is calculated by using (8).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{8}$$

where y_i is the actual score and $\hat{y}_i$ is the predicted score.

The proposed system is tested on the dataset of 330 student answers and the result reported RMSE value of 0.81. This value shows that the LCCS algorithm presented in this work performs better than the LCS metrics used in [34,39] where their obtained RMSE values were 1.18 and 1,22 respectively. The work in [34] adopted the approach of translating the Arabic text into English whereas the work in [39] benefits from the combination of similarity algorithms. It is believed that text translation may weaken the text meaning and hence influence the accuracy of text similarity. In addition, adopting hybrid approach with the proposed model could present a promising result. However, RMSE of the prediction model is plotted in Fig. 9 which shows the error between actual scores (red line) and predicted scores (blue points). Besides RMSE, the conducted experiments reported MAE value of 0.63, MSE value of 0.65 and MAPE value of 15.52.

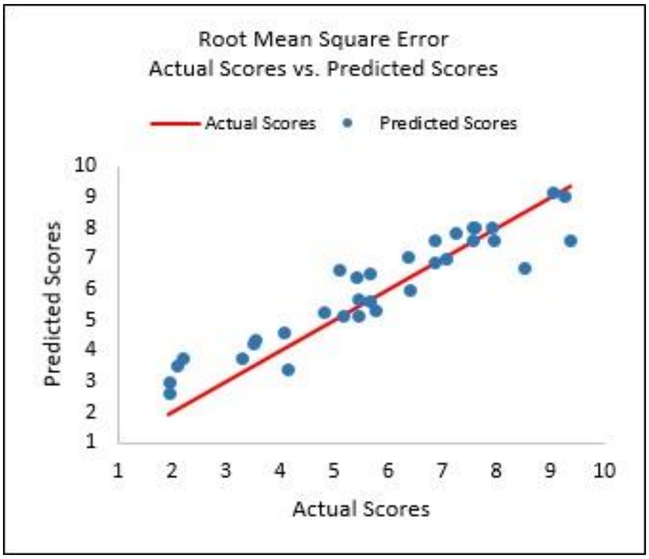


FIGURE 9. Root mean square error of the proposed predictive model.

Pearson correlation coefficient r is another metric used for measuring the accuracy of model performance. Correlation coefficient r is a measurement that tells the strength of the linear relationship between two variables (actual variable x and predicted variable y). Correlation r is a numerical value between -1 and 1. When r is closer to 1 it indicates a strong positive relationship with positive slope. Values close to -1 indicate a strong negative relationship with negative slope. A value of 0 indicates that there is no relationship. Equation (9) is used to calculate the correlation coefficient r.

$$r = \frac{n\sum_{i=1}^{n} x_i\, y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^{\,2} - (\sum_{i=1}^{n} x_i)^2}\ \sqrt{n\sum_{i=1}^{n} y_i^{\,2} - (\sum_{i=1}^{n} y_i)^2}} \tag{9}$$

where n is the total number of samples, x_i (x_1, x_2, …,x_n) are the actual scores and y_i (y_1, y_2, …, y_n) are the predicted scores.

The system is also experimented on the dataset and the result reported correlation r of 0.94. The result indicates that there is a strong positive linear correlation between actual scores and predicted scores as shown in Fig. 10.

The correlation r value obtained by the system is very close to 1 compared to correlation r values reported in [34,39] which were 0.49 and 0.53 respectively. The experiment results show that the model presented in this work outperforms other research works that have adopted approaches using only LCS algorithm for scoring short answers of Arabic questions. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082408, IEEE Access

result is attributed to the employment of AWN as a dictionary for providing word synonyms and to the application of LCCS as an enhanced version of LCS algorithm for tuning text similarity result.

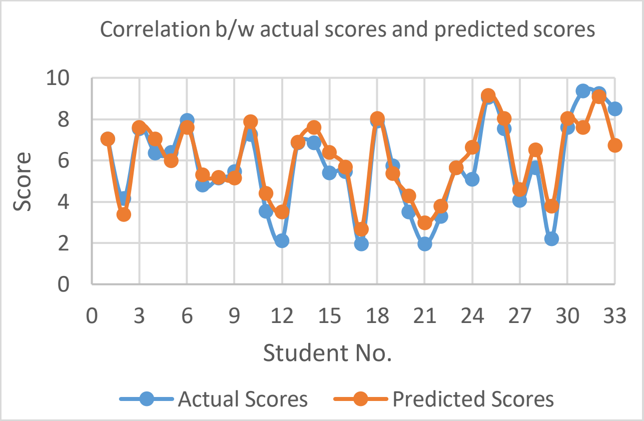


FIGURE 10. **The linear correlation between actual scores and predicted scores.**

D. RESULTS ANS DISCUSSIONS

Besides RMSE and Pearson correlation coefficient r that have been calculated above, several statistical tests are also performed to judge the significance of the proposed method's results. For this purpose, we have calculated average values, standard deviations, minimal and maximal results and ranges of grades of human (actual) raters and automatic (predicted) rater. This descriptive statistical data is provided in Table VI.

TABLE VI
DESCRIPTIVE STATISTICAL DATA.

Rater	Mean	SD	Min	Max	Range
Human1	5.77	2.43	1.20	10.00	8.80
Human 2	5.78	1.94	2.30	8.70	6.40
Automatic	6.05	1.79	2.67	9.16	6.49

As observed from Table VI, the highest mean value is that of the automatic rater (M=6.05) and the lowest mean value is for the first human rater (M=5.77). These arithmetic means are close to 5.5 which is the average value for the 11-point scale (0-10) of grading students' answers. As for standard deviation (SD), the most value is the grade of first human rater (SD=2.43) and the least value is the grade of the automatic rater (SD=1.79). For range, neither automatic nor human raters have value 10 which is the range value for the 11-point scale. However, the closest range value is for first human rater (Range=8.8).

The null hypothesis is defined in terms of statistical significance differences between actual scores estimated by the average human raters and predicted scores that are calculated automatically by the proposed method. In other words, the null hypothesis is that automatic rater and human rater provide similar results when scoring students' answers. Paired-samples t-test is conducted to examine this hypothesis. Paired-samples t-test is selected because two datasets of dependent variables are compared. Table VII shows the statistical results of paired-samples t-test.

TABLE VII
THE STATISTICAL RESULTS OF PAIRED-SAMPLES T-TEST.

Rater	M	SD	Skew	SE_M	Mdiff	SDdiff	SE_{Mdiff}	p-value	t-value	df
Automatic rater	6.048	1.795	-0.20	0.312	0.272	0.771	0.134	.052	2.03	32
Avg. of human raters	5.776	2.153	-0.22	0.375						

In Table VII, the first three columns are arithmetic means (M), standard deviations (SD) and skew values of the two datasets. Skewness is a measure of the symmetry in a distribution. In the next four columns are: standard errors of means (SE_M), difference of means (Mdiff), standard deviation of difference (SDdiff) and standard error of mean difference (SE_{Mdiff}). In last three columns are significance value (p-value), value of t-test (t-value) and degrees of freedom (df).

The results show that the arithmetic mean value of automatic rater (6.048) is little bit higher than that of average human raters (5.776). So, the value of difference between the two arithmetic means (Mdiff) is equal to 0.272 which indicates that automatic rater is slightly moderate than that of average human raters. However, this difference is not statistically significant because p-value (p=.052) is greater than the significance level of α where α=0.05 for df=32. This means that it is fail to reject the null hypothesis and consequently conclude that the average values of raters do not differ sufficiently. Therefore, the average value of automatic rater for scoring students' answers is similar to that of human rater. In addition, the standard deviation value of automatic rater (1.795) is close to that of human rater (2.153) which indicates that the range of scores which are assessed by both raters are close to each other. For instance, with one standard deviation on normal distribution, 68% of students' answers assessed by automatic rater ranges from 4.2-7.8 scores against 3.6-7.9 scores assessed by human rater. This is another indication that support the hypothesis since both raters provide close ranges of scores for the same number of students' answers.

For examining skewness of distributions, two graphs are plotted as shown above, Fig. 11 plots distribution of grades of human rater with skew value -0.22 and Fig. 12 plots distribution of grades of automatic rater with skew value -0.20 (skew values are provided in Table VII). Skew values of both distributions are negative. This means that both distributions have negative asymmetry, i.e. the number of high scores are more than the number of low scores for both raters. So, two inferences can be drawn from this result: (1) both raters are moderate estimators of students' answers; (2) both raters perform similarly in assessing students' answers which confirms the hypothesis.

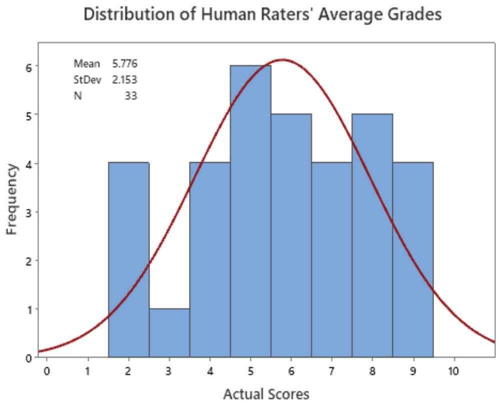


FIGURE 11. Distribution of grades of human raters average.

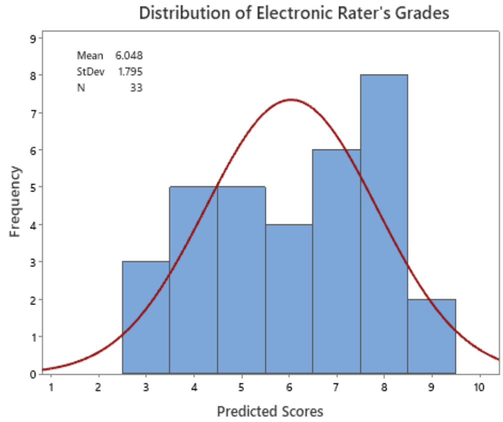


FIGURE 12. Distribution of grades of automatic rater.

To conclude the discussion, the obtained results proved that the proposed method predicts scores of students' answers similar to a high extent to average of human estimators assessing the scores of the same students' answers. Moreover, the distributions of actual and predicted scores show that both human and proposed method raters tend to overestimate students' answers.

VI. CONCLUSION AND FUTURE WORK

Automatic scoring of short answers of Arabic questions is inherently a problematic issue especially when dealing with complex natural language such as Arabic. Though, researchers have adopted several approaches including longest common subsequence measure for tackling this issue, the results were unsatisfied. This study investigated string-based text similarity approach using longest common subsequence measure for handling the issue. However, two key ideas are employed for this investigation. First: Arabic WordNet is used as standard semantic resource for providing synonyms of student answer words. This step narrows the lexical dissimilarity between student answer and model answer. Second: longest common subsequence method is improved by adopting the concepts of contiguousness and scaling. This is a tuning step that adjust the closeness degree of student answer to model answer.

Findings based on experiments conducted on a dataset of 330 students' answers have shown significant improvements in accuracy of model performance. The experiments reported RMSE value of 0.81 and correlation r value of 0.94. In addition, the results from the statistical analysis have shown that the proposed method estimates students' answers similar to that of human estimator. Based on results, it can be concluded that the proposed system outperforms the other systems used the similar method. This conclusion indicates the effectiveness of the proposed method. In this work, the size of the dataset has no influence on the effectiveness of the proposed method because the focus is not on time complexity of the method taken for comparison process, rather the focus is on accuracy of the method in assessing the similarity of texts. Therefore, the contribution of developing an effective automatic essay scoring which is useful for educational sectors is achieved since the effectiveness of the proposed method is verified experimentally on dataset collected from educational institution.

However, the proposed method can be applied in many Arabic applications including: 1) scoring short answers of Arabic essay questions in universities, schools and institutes; 2) detecting plagiarism of Arabic textual assignments in universities and Arabic textual articles in scholar research centers and journalism; 3) compressing Arabic text data files for saving space and time; 4) comparing contents of Arabic text data files for checking file corruption in operating systems; 5) lexicon-based sentiment analysis to identify the orientation of a text document by measuring the semantic orientation of words and sentences based on a dictionary [49]; 6) tracking changes of source code and Arabic documentation files during software development in version control systems.

Although the research work has several advantages and strength aspects, it has some limitations that worth noting. First, this research was conducted on small size of dataset since no gold standard dataset is available for Arabic. Second, linguistic errors which incorporate mistakes in spelling, punctuation and grammar are not considered. Third, semantic analysis of sentences and information such as syntax of sentences which incorporate the word order and structure of sentences [3] are not investigated. Fourth, feedback and correction for the assessment process are not provided.

Arabic WordNet is a standard resource for Arabic text processing. However, it lacks of retrieving roots of some words. For example, the word “ادراك” (realization) is not retrieved from Arabic WordNet as a synonym of the word “الذكاء” (intelligence). It is no doubt that a comprehensive Arabic WordNet would improve the text similarity results of future works.

ACKNOWLEDGMENTS

This work was supported by Prince Sattam Bin Abdulaziz University (PSAU). We thank PSAU for their support.

REFERENCES

[1] D. Sarkar, Natural Language Processing Basics. In: Text Analytics with Python, 2nd ed., Apress, Berkeley, CA, 2019, pp. 1–68.

[2] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," Artif Intell Rev, vol. 52, pp. 52–273, 2019, https://doi-org.sdl.idm.oclc.org/10.1007/s10462-018-09677-1.

[3] A. Onan, S. Korukoğlu and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," Expert Systems with Applications, vol. 57, pp. 232-247, 2016.

[4] M. Farouk, "Measuring Sentences Similarity: A Survey," Indian Journal of Science and Technology, vol. 12, no. 25, 2019, doi: 10.17485/ijst/2019/v12i25/143977.

[5] D. D. Prasetya, A. P. Wibawa, and T. Hirashima, "The performance of text similarity algorithms," International Journal of Advances in Intelligent Informatics, vol. 4, no. 1, pp. 63–69, 2018, doi:org.sdl.idm.oclc.org/10.26555/ijain.v4i1.152.

[6] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," International Journal of Computer Applications, vol. 68, no. 13, pp. 13-18, 2013.

[7] A. Rozeva, S. Zerkova, "Assessing semantic similarity of texts - Methods and algorithms," *AIP Conference Proceedings*, *1910, 060012 (2017)*, vol. 1910, no. 1, 2017, doi.org/10.1063/1.5014006.

[8] Zixuan Ke and Vincent Ng, "Automated essay scoring: a survey of the state of the art," IJCAI'19 Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, pp. 6300-6308, 2019, https://doi.org/10.24963/ijcai.2019/879.

[9] Yusheng Wang, "A Study of applying automated assessment in teaching college English writing based on Juku correction network," International Journal of Emerging Technologies in Learning, vol. 14, no. 11, pp. 19–31, 2019, doi:10.3991/ijet.v14i11.9411.

[10] P. Krusche and A. Tiskin, "Efficient parallel string comparison," Advances in Parallel Computing, IOS Press BV, vol. 15, pp. 193-200, 2008.

[11] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," *Proceedings Seventh International Symposium on String Processing and Information Retrieval SPIRE 2000, String Processing and Information Retrieval, 2000 SPIRE 2000 Proceedings Seventh International Symposium on*, pp. 39-48, 2000.

[12] R. Beal, T. Afrin, A. Farheen, and D. Adjeroh, "A new algorithm for 'the LCS problem' with application in compressing genome resequencing data," BMC Genomics, vol. 17, Suppl 4, pp. 544, 2016.

[13] O. H. Ibarra, T. C. Pong, and S. M. Sohn, "String processing on the hypercube," *IEEE Transactions on Acoustics, Speech, and Signal Processing, Acoustics, Speech and Signal Processing, IEEE Transactions on, IEEE Trans Acoust, Speech, Signal Process*, USA, vol. 38, no. 1, pp. 160-164, 1990, doi:10.1109/29.45630.

[14] G. Nyirarugira, H. Choi, and T. Kim, "Hand gesture recognition using particle swarm movement," Mathematical Problems in Engineering, pp. 1-8, 2016, doi: 10.1155/2016/1919824.

[15] A. Mohaimen and D. Chakraborty, "Bangla OCR Post Processing - Word Based Longest Common Subsequence Technique," B.Sc. thesis, Dept. of CS and Eng. Shahjalal Univ. of Sc. and Tech., Bangladesh, 2017, Accessed on: Nov. 6, 2019. [Online]. Available: https://www.researchgate.net/publication/323105621.

[16] D. Goswami, N. Sultana, and W. Ruheen Bristi (2020) "Algorithms for String Comparison in DNA Sequences," In: Uddin M., Bansal J. (eds) *Proceedings of International Joint Conference on Computational Intelligence*, Algorithms for Intelligent Systems, Springer, Singapore, pp. 327-343, 2019.

[17] J. W. Hunt, M. D. Mcilroy, "An algorithm for differential file comparison," *Computer Science*, 1975.

[18] M. Silfverberg, L. Liu, M. Hulden, "A Computational Model for the Linguistic Notion of Morphological Paradigm," *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1615–1626, 2018.

[19] A. Chaudhuri, "A Dynamic Algorithm for the Longest Common Subsequence Problem using Ant Colony Optimization Technique", *Proceedings of 2nd International Conference on Mathematics*, Cairo, Egypt, 2013.

[20] A. Onan, S. Korukoğlu and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," Expert Systems with Applications, vol. 62, pp. 1-16, 2016.

[21] A. Onan and S. Korukoğlu, "A feature selection model based on genetic rank aggregation for text sentiment classification," Journal of Information Science, vol. 43, no. 1, pp. 25-38, 2017.

[22] Indu and Prerna. "A Comparative study of different longest common subsequence algorithms," International Journal of Recent Research Aspects, vol. 3, no. 2, pp. 65-69, 2016.

[23] K. Chandra, "Variants of longest common subsequence problem," Ph.D. dissertation, Rochester Institute of Technology, 2016.

[24] G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, ACM New York, NY. USA, vol. 38, no. 11, pp. 39-41, 1995.

[25] D. Jurafsky and J. H. Martin, "Word Senses and WordNet," in *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd ed., New Jersey: Prentice Hall, 2019, pp. 354-372. Accessed on: Nov. 18, 2019. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf.

[26] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An On-line Lexical Database," International Journal of Lexicography (special issue), vol. 3, no. 4, pp.235—312, 1990.

[27] J. Priyatno, M. A. Bijaksana, "Clustering Synonym Sets in English WordNet," *2019 7th International Conference on Information and Communication Technology (ICoICT), Information and Communication Technology (ICoICT), 2019 7th International Conference on*, IEEE, Kuala Lumpur, Malaysia, p. 1, 2019, doi: 10.1109/ICoICT.2019.8835313.

[28] S. Elkateb et al., "Building a WordNet for Arabic," in Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), European Language Resources Association (ELRA), Genoa, Italy, 2006.

[29] M. Sayed, R. K. Salem, and A. E. Khder, "A survey of Arabic text classification approaches," International Journal of Computer Applications in Technology, vol. 59, no. 3, 2019.

[30] M. A. Hussein, H. A. Hassan, and M. Nassef, "Automated language essay scoring systems: A literature review," PeerJ Preprints, 2019, https://doi.org/10.7287/peerj.preprints.27715v1.

[31] TH. Chang, and YT. Sung, "Automated Chinese essay scoring based on multi-level linguistic features," In Lu, X. and Chen, B. (eds), *Computational and Corpus Approaches to Chinese Language Learning*. Singapore: Springer, pp. 258–274, 2019.

[32] T. S. Walia, G. S. Josan, and A. Singh, "An efficient automated answer scoring system for Punjabi language," ScienceDirect, Egyptian Informatics Journal, vol. 20, no. 2, 2019.

[33] M. Lilja, "Automatic Essay Scoring of Swedish Essays Using Neural Networks," MSc. Thesis, Department of Statistics, Uppsala Univ., Sweden, 2018, Accessed on: Nov. 24, 2019. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1213688/FULLTEXT01.

[34] R. S. Citawan, V. C. Mawardi, B. Mulyawan, "Automatic Essay Scoring in E-learning System Using LSA Method with N-Gram Feature for Bahasa Indonesia", *MATEC WEB of Conferences,* vol. 164, no. 01037, 2018.

[35] MA. Cheon et al., "Automated scoring system for Korean short-answer questions using predictability and unanimity," Korea Information Processing Society, vol. 5, no. 11, pp. 527-534, 2016.

[36] A. Shehab, M. Faroun, and M. Rashad, "An automatic Arabic essay grading system based on text similarity algorithms," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 3, pp. 263–268, 2018.

[37] W. H. Gomaa, and A. A. Fahmy, “Automatic scoring for answers to Arabic test questions,” Computer Speech & Language, vol. 28, no. 4, pp. 833-857, 2014.

[38] R. Mezher and N. Omar, "A Hybrid method of syntactic feature and latent semantic analysis for automatic Arabic essay scoring," Journal of Applied Sciences, vol. 16, no. 5, pp. 209-215, 2016.

[39] H. Rababah and A. T. Al-Taani, "An automated scoring approach for Arabic short answers essay questions," *2017 8th International Conference on Information Technology (ICIT)*, Amman, 2017, pp. 697-702. doi: 10.1109/ICITECH.2017.8079930

[40] A. R. Abbas and A. S. Al-Qaza, “Automated Arabic essay scoring (AAES) using vector space model (VSM) and latent semantics indexing (LSI),” Engineering and Technology Journal (University of Technology - Iraq), vol. 33, no. 3, pp. 410–426, 2015.

[41] A. M. Azmi, M. F. Al-Jouie, M. Hussain, “AAEE — Automated evaluation of students’ essays in Arabic language,” Information Processing & Management, vol. 56, no. 5, pp. 1736-1752, 2019.

[42] W. H. Gomaa and A. A. Fahmy, “Arabic short answer scoring with effective feedback for students,” *Int. J. Comput. Appl.*, vol. 86, no. 2, pp. 35-41, 2014.

[43] S. A. Al Awaida, B. Al-Shargabi, and T. Al-Rousan, “Automated Arabic essay grading system based on f-score and Arabic WordNet,” Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 170-180, 2019.

[44] R. Bhowmick, M. I. Sadek Bhuiyan, M. Sabir Hossain, M. K. Hossen and A. Sadee Tanim, "An Approach for Improving Complexity of Longest Common Subsequence Problems using Queue and Divide-and-Conquer Method," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, pp. 1-5, 2019, doi: 10.1109/ICASERT.2019.8934638.

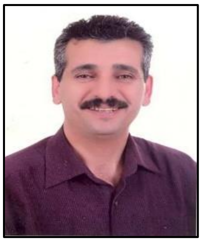
[45] Y. Regragui, L. Abouenour, F. Krieche, K. Bouzoubaa and P. Rosso, "Arabic WordNet: New Content and New Applications," *Proceedings of the 8th Global WordNet Conference*, Bucharest, Romania, pp. 330-338, 2016.

[46] K. Ryding, “Introduction to Arabic morphology,” In *Arabic: A Linguistic Introduction*, Cambridge: Cambridge University Press, pp. 41-54, 2014, doi:10.1017/CBO9781139151016.006.

[47] A. Onan, S. Korukoglu.and H. Bulut, “LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis,” Int. J. Comput. Linguistics Appl., vol. 7, no. 1, pp. 101-119, 2016.

[48] J. Kim, K. Park, J. Chae et al., “Automatic scoring system for short descriptive answer written in Korean using lexico-semantic pattern”, Soft Comput 22, pp. 4241–4249, 2018. https://doi.org/10.1007/s00500-017-2772-7.

[49] A. Onan, “Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks,” Concurrency and Computation: Practice and Experience, p.e5909, 2020.



Hikmat A. Abdeljaber was born in Kuwait, in 1967. He received the Ph.D. degree in information sciences and technology in 2010 from the Universiti Kebangsaan Malaysia, UKM, Malaysia. He currently holds a University Assistant position at the Prince Sattam Bin Abdulaziz University in Saudi Arabia, college of Computer Engineering and Sciences. He lectured in the fields of computer science and information systems for both undergraduate and graduate levels. He has published papers in the area of information retrieval and artificial intelligence. His research interests include information retrieval, semantic web technology, data mining and machine learning.