IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 15, 2021, accepted June 29, 2021, date of publication July 1, 2021, date of current version July 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094115

Arabic Questions Classification Using Modified TF-IDF

ALI SALEH ALAMMARY
College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia
e-mail: a.alammary@seu.edu.sa

ABSTRACT Classifying the cognitive levels of assessment questions according to Bloom's taxonomy can help instructors design effective assessments that are well aligned with the intended learning outcomes. However, the classification process is time consuming and requires experience. Many studies have attempted to automate the process by utilizing different machine learning and text mining approaches, but none has examined the classification of Arabic questions. The purpose of this study is to examine this research gap and to introduce a new feature extraction method that would better suit Arabic questions and their unique characteristics. It also aims to provide Arab instructors with a tool that can help them automatically classify their assessment questions. To accomplish this purpose, the study developed a dataset of more than 600 Arabic assessment questions. It then proposed a modified term frequency- inverse document frequency (TF-IDF) method for extracting features from Arabic questions. Unlike the traditional TF-IDF, the proposed method was designed to take the nature of assessment questions into consideration. It was evaluated by comparing it to two methods that have been used for classifying English questions, i.e., the traditional TF-IDF and a modified TF-IDF method called term frequency part-of-speech- inverse document frequency (TFPOS-IDF). A t-test was utilized to examine whether the difference in performance between the three methods was statistically significant. The proposed method outperformed the two other methods. The overall accuracy, precision, and recall scored by the proposed method were significantly higher than those scored by the traditional TF-IDF and TFPOS-IDF methods. The evaluation results indicate the promising potential of the proposed method, which can be extended to other languages.

INDEX TERMS Arabic text classification, feature extraction, learning analytics, machine learning, TF-IDF.

I. INTRODUCTION

Bloom's revised taxonomy is a widely accepted framework for classifying learning outcomes and assessment questions [1], [2]. The taxonomy has six levels of cognitive skills that move in a hierarchal order of complexity. At the bottom are remembering and understanding, which are referred to as lower-order thinking skills (LOTS). The top four levels are applying, analyzing, evaluating, and creating, which are referred to as higher-order thinking skills (HOTS) [3]. While LOTS can be valuable, purely memorizing and understanding knowledge is insufficient in modern-day higher education. University students are expected to develop HOTS; these skills have been identified as an essential attribute for these students [4], [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiwei Gao.

Assessments drive students' learning and affect the skills that students acquire [6]. Instructors can foster student learning by aligning assessment questions with the intended learning outcomes [7], [8]. Thus, it is extremely important to design effective assessments that cover the different cognitive levels, assess diverse students' skills and are well aligned with the intended learning outcomes. However, Momsen, *et al.* [9], El-Gohary [10] and Jones, *et al.* [11] examined assessment questions of different courses and found that the majority of these questions assessed LOTS, and only a small percentage of questions targeted HOTS. Lightfoot and Schwager [12] and Martin [13] also studied several undergraduate courses and found that the majority of assessment questions are not well aligned with the learning outcomes and that misalignment is a common problem in undergraduate courses.

Designing effective assessments seems to be very challenging, especially for instructors who lack theoretical knowledge

and practical experience with instructional design, which is the case for many instructors in postsecondary institutions [10], [14]. A major challenge is to identify the cognitive level of assessment questions and use that to ensure accurate alignment with the learning outcomes and an appropriate balance of LOTS and HOTS questions. While instructors normally look at the verb or the interrogative word to determine the cognitive level of questions, there are verbs and interrogative words that belong to more than one cognitive level of Bloom's revised taxonomy [11], [15]. For example, ''Explain what do we mean by the term democracy.'' is a LOTS question, while ''Explain the reason why link state routing is preferable to distance vector style routing'' would be a HOTS question. In such cases, the whole context of the question should be taken into consideration, and instructors need to rely on their own experience to decide the right cognitive level [11]. This task can be even more challenging considering the many courses that instructors teach each semester and the large number of questions that they need to produce for each course [16].

To facilitate the process of classifying assessment questions according to Bloom's revised taxonomy, many studies have tried to automate the process by using different machine learning approaches [17]–[20]. However, none of these studies has looked at classifying Arabic questions. Furthermore, the feature extraction approaches that have been used in these studies are not tailored toward assessment questions. They were developed for other text classification tasks and, therefore, do not take the nature of assessment questions into consideration.

This study focuses on classifying Arabic assessment questions according to Bloom's taxonomy. It introduces a new modified term frequency-inverse document frequency (TF-IDF) for feature extraction. It develops the first dataset for Arabic question classification. To the best of our knowledge, this is the first study that applies machine learning to the classification of Arabic questions. The proposed modified TF-IDF approach considers the nature of assessment questions and assigns weights to the different words in questions based on their importance to the classification task.

The remainder of the paper is organized into six sections. Section II provides background information about Bloom's taxonomy, machine learning, text mining, and feature extraction approaches. It also discusses related work. Section III explains the proposed method and how it was designed. Section IV describes how the method was evaluated by combining it with five different classifiers. Section V presents the evaluation results. Section VI provides a brief discussion. Section VII concludes the paper and suggests some future work.

II. BACKGROUND

Before demonstrating the proposed modified TF-IDF method, this section presents brief information about Bloom's taxonomy and the different machine learning approaches that have been used in this study. It also discusses related work.

A. BLOOM'S TAXONOMY

Bloom's taxonomy was first introduced in 1956. It aimed to help instructors describe and classify educational goals and assessment items [21]. The taxonomy has three overlapping domains: cognitive, psychomotor, and effective. The cognitive domain attracted the most attention and became the most widely utilized domain in higher education [22]. It organizes the learning process into six levels, in increasing order of complexity, with achievement at lower levels being required to move up to the higher levels [2].

The taxonomy has been revised several times. The most notable revision was that of Anderson and Krathwohl [3] in 2001. In the revised taxonomy, the number of cognitive levels was retained, but the order of the two upper levels was interchanged, and the names of all six levels were changed to verb forms. These levels are (from low to high level):

1) Remember: recall basic knowledge from memory.
2) Understand: construct meaning from previous learning materials.
3) Apply: use a procedure in a given situation.
4) Analyze: break learning materials into smaller parts and determine how these parts are related to each other or an overall structure.
5) Evaluate: judge a situation based on given criteria.
6) Create: put a group of elements together to form a functioning whole.

Bloom's taxonomy has proven to be very helpful in writing effective assessments. It can help instructors to not only understand the cognitive levels of both the learning outcomes and the assessment questions but also to use them to ensure accurate alignment between the two [11]. Instructors can also use the taxonomy to maintain an appropriate balance between HOTS and LOTS questions [11] and to perform a cross-check to ensure that no learning outcome is overtested at the expense of others [23].

B. MACHINE LEARNING

Machine learning can be defined as ''a subfield of artificial intelligence that includes software able to recognize patterns, make predictions, and apply newly discovered patterns to situations that were not included or covered by their initial design'' [24]. Machine learning has become crucial for solving complex problems and has been successfully applied in a wide range of areas, including the stock market, games, medical diagnostics, robotics, information security, and education [25]. In education, it has been used for a variety of purposes, such as predicting student dropout and retention [26], admission decisions and course scheduling [27], profiling students and modeling their learning behavior [28], and evaluating student engagement and academic integrity [29].

Machine learning is broadly categorized into two main categories: supervised and unsupervised machine learning. Supervised machine learning, which is the focus of this paper, refers to algorithms that use labeled datasets to find patterns that can be applied to make predictions [30]. A successful

supervised learning model should (1) predict a target value of a training dataset to a satisfactory degree of accuracy and (2) be able to be successfully generalized to other datasets [31].

Many machine learning algorithms have been developed over the years, such as linear regression, logistic regression, decision tree, support vector machine, K-nearest neighbor and naive Bayes. Each algorithm has its strengths and can be used to solve a variety of problems. Therefore, data scientists normally try more than one algorithm to find the one that is most suitable for their application [32].

C. TEXT MINING

Text mining can be defined as the process of extracting new meaningful information and knowledge from unstructured text [33]. Text mining is a variation of another field called data mining. While data mining works on structured data, text mining can handle unstructured or semistructured data such as assessment questions [34].

To enhance the extraction process of text mining, a technique called natural language processing (NLP) is normally used [35]. NLP performs a special kind of linguistic analysis on text, such as [34]:

1) Tokenization: splitting a piece of text into words or phrases.
2) Stop word elimination: removing words such as articles and prepositions that do not add much value to the meaning of the text.
3) Stemming and lemmatization: identifying the root/stem of words in the text.
4) Feature extraction: identifying a meaningful set of features in the text.

A wide variety of methods can be used for feature extraction. The bag-of-words model is a simple method in which documents are represented by word occurrences while ignoring the positions of the words [36]. TF-IDF is another classic method that is simple but very efficient. It measures the importance of a word to a document in a corpus. It is computed by multiplying the term frequency (TF) by the inverse document frequency (IDF). TF is the number of times a term appears in a document, while IDF is a measurement of how significant that term is relative to the whole corpus [37].

Another method that has received increasing attention is word embedding. In this method, each word in the corpus is represented by a vector of real numbers. This representation facilitates language understanding by mathematical operations. It allows capturing the similarity between individual word vectors, thus providing information on the underlying word meanings [38]. Several pretrained word embedding models have been developed, such as word2vec, global vectors for word representation (GloVe) and FastText [39].

D. RELATED WORK

Several studies have applied machine learning and NLP to classify assessment questions according to Bloom's taxonomy. Jayakodi, *et al.* [40] generated a dataset of 88 questions taken from several information technology courses. They used WordNet and cosine similarity to develop a rule-based classifier, which was able to achieve 70% accuracy. Similarly, Aninditya, *et al.* [19] developed a dataset of 300 mid-term and final exam questions taken from different information technology courses. They then used TF-IDF for feature extraction and naive Bayes for classification. Their classifier achieved a precision of 85% and recall of 80%.

Osadi, *et al.* [41] developed an ensemble classifier by combining four classification algorithms: support vector machine, naive Bayes, k-nearest neighbor, and rule-based classifier. They used a word vector method for feature extraction. They tested their classifier on a dataset of 100 questions taken from a programming course. Their ensemble classifier outperformed the individual classifiers and scored an overall accuracy of 82%. Mohammed and Omar [17] used word2vec and a modified TF-IDF called term frequency part-of-speech-inverse document frequency (TFPOS-IDF) to extract features from exam questions. The extracted features were fed into three different classification algorithms: support vector machine, K-nearest neighbor and logistic regression. In their experiments, they used two datasets. The first set contained 600 questions, while the second had 141 questions. The support vector machine performed the best and was able to achieve an F1-measure of 89.7% on the first dataset and 83.7% on the second dataset.

All of the previous works targeted English questions, and none examined Arabic question classification. In addition, the feature extraction approaches that were utilized in these studies were not developed to extract features from assessment questions. Even in the study of Mohammed and Omar [17], the modified TF-IDF that they used was initially developed to extract features from information retrieval queries. They used it in their study as it does not take the nature of assessment questions into consideration. Despite this, the modified method performed better than the traditional TF-IDF. In fact, the same result has been reported in many studies where modified TF-IDF methods were utilized.

Roul, *et al.* [42] developed a modified version of the traditional TF-IDF to extract features from news articles. Their modified method takes the frequency distribution and document length into consideration. They tested it on three different datasets and achieved better or comparable results than the traditional TF-IDF method. Similarly, Kim and Lee [43] proposed a modified TF-IDF for classifying health symptoms. Their proposed TF-IDF assigns higher weight when a disease and a symptom are mentioned together in many academic articles or when a symptom that is mentioned with a disease is rarely mentioned with other symptoms. The modified TF-IDF achieved better performance than the traditional TF-IDF. Zhu, *et al.* [44] also proposed a modified TF-IDF method that they called TA TF-IDF. Their method is intended to detect and classify hot

news topics. It incorporates time distribution information and user attention. Their experiment showed that the modified method could reach an average accuracy of 78.36%

III. THE PROPOSED METHOD

To understand how the proposed method works, it is important to first explain the traditional TF-IDF and how it is calculated. In the traditional TF-IDF, the TF of a term t in document d is computed as follows [37]:

$$tf\ (t, d) = \frac{c_{t,d}}{\sum_k c_{t,d}}, \tag{1}$$

where $c_{t,d}$ indicates the number of times t appears in d and $\sum_k c_{t,d}$ denotes the total number of terms in d. The more times t appears in d, the more significant it becomes to that document. The TF is normally referred to as the local term weight.

The IDF is calculated as follows:

$$idf\ (t, D) = log\frac{D}{d_t} + 1, \tag{2}$$

where D denotes the total number of documents in the corpus and d_t indicates the number of documents that contain the term t. The fewer times t appears in a corpus, the more significant it becomes in identifying the documents in that corpus. The IDF is normally referred to as the global term weight.

Traditional TF-IDF is calculated by multiplying Equation (1) by Equation (2).

$$TF - IDF\ (t, d, D) = tf\ (t, d) \times idf\ (t, D)\,. \tag{3}$$

Traditional TF-IDF treats all terms in a document the same. It assumes that all terms have the same level of importance to the classification task and, therefore, calculates their TF-IDF in the same way. Unlike other types of data, assessment questions have characteristics that need to be taken into consideration when classifying them according to Bloom's taxonomy. There are certain terms, i.e., question verbs and interrogative words, that carry greater importance in the classification task. It is true that relying only on a predefined list of verbs and interrogative words to decide the Bloom's level would not yield good performance, as shown in the study of Wen-Chih and Ming-Shun [45]. However, these verbs and interrogative words should be assigned higher weights than other terms that are of less importance to the classification task.

When looking at a question such as "اشرح معنى الاشتراكية الديمقراطية", which can be translated as "Explain the meaning of social democracy.", it can be seen that the verb اشرح "explain" has the greatest importance in deciding the Bloom's level of the question as compared to the other terms. By considering this verb, the classifier can deduce that there is very little possibility for the question to belong to the Remember, Apply, Evaluate or Create level of the taxonomy. The verb اشرح "explain" has no or few occurrences in questions that belong to these four Bloom's levels. It frequently occurs in questions that belong to the Understand and Analyze levels. By eliminating 4 out of 6 classification possibilities, we could say that the verb has helped in achieving more than 60% of the classification work. This must be reflected in the weight that is assigned to the verb when calculating its TF-IDF score. This is what our modified TF-IDF does. It calculates the TF score of question verbs and interrogative words as follows:

$$tf_{ar}\ (t, d) = \frac{c_{t,d} + (\sum_k c_{t,d} \div 2)}{\sum_k c_{t,d} + (\sum_k c_{t,d} \div 2)}. \tag{4}$$

The equation increases the term frequency of question verbs and interrogative words by a number equal to half of the question length. Thus, it guarantees that question verbs and interrogative words have TF scores larger than any other word in the question no matter the length of the question.

Because our modified TF-IDF is designed to work with Arabic questions, it is important to note that there are two types of Arabic interrogative words. Type 1 includes words that have meaning by themselves such as لماذا "Why", متى "When" and كيف "How". Type 2 includes words that do not have meaning by themselves, and one should look at the words that follow these interrogative words to understand their meaning, such as ما and من. For example, let us look at the interrogative word من in the following two questions: "من أي مركب يودي الحصول على الكيميائية التالية الصيغ" and "من هو أول رئيس للولايات المتحدة". In both questions, the word does not have complete meaning by itself. However, if we combine it with the word that follows, the meaning becomes "Which" in the first question and "Who" in the second question. By considering this, our modified TF-IDF will divide the TF score in Equation (4) between Type 2 interrogative word and the word that follows it. The TF score for each of the two words will be calculated as follows:

$$tf_{\ ar}\ (t, d) = \frac{c_{t,d} + (\sum_k c_{t,d} \div 2)}{\sum_k c_{t,d} + (\sum_k c_{t,d} \div 2)} \div 2. \tag{5}$$

For the remaining words in the Arabic question, the TF score will be calculated as follows:

$$tf_{\ ar}\ (t, d) = \frac{c_{t,d}}{\sum_k c_{t,d} + (\sum_k c_{t,d} \div 2)}. \tag{6}$$

No change has been made to the way the IDF score is calculated in our modified TF-IDF. This is because IDF is a global term weight and is calculated at the corpus level, not at the question level;

$$idf_{\ ar}\ (t, D) = log\frac{D}{d_t} + 1. \tag{7}$$

Therefore, the TF-IDF score in our modified method is calculated as follows:

$$TF - IDF_{ar}\ (t, d, D) = tf_{\ ar}\ (t, d) * idf_{\ ar}\ (t, D)\,. \tag{8}$$

The whole calculation can be illustrated as follows:

$$TF - IDF_{ar}(t, d, D) = \begin{cases} \frac{c_{t,d}+(\sum_k c_{t,d}\div 2)}{\sum_k c_{t,d}+(\sum_k c_{t,d}\div 2)} & \text{if question verb or Type 1 interrogative word} \\ \frac{c_{t,d}+(\sum_k c_{t,d}\div 2)}{\sum_k c_{t,d}+(\sum_k c_{t,d}\div 2)} \div 2 & \text{if Type 2 interrogative word or word that follows} \\ \frac{c_{t,d}}{\sum_k c_{t,d}+(\sum_k c_{t,d}\div 2)} & \text{otherwise} \end{cases} * log\frac{D}{d_t} + 1. \quad (9)$$

IV. EVALUATION

To evaluate the proposed method, a process of five steps was performed (see Fig. 1).

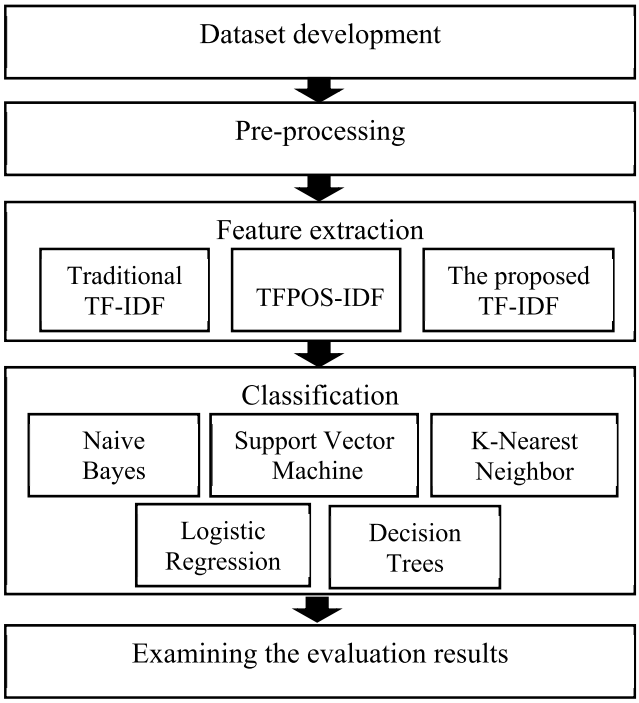

FIGURE 1. The process for evaluating the proposed method.

A. STEP 1: DATASET DEVELOPMENT

As no previous dataset has been developed for classifying Arabic assessment questions according to Bloom's taxonomy, the first step was to develop a dataset. The dataset contains 610 questions. All questions are short answer and essay questions. They were collected from different academic resources [46], [47] and by translating questions from the dataset developed by Yahya and Osman [48]. The questions belong to different academic disciplines, including computer science, engineering, mathematics, physics, history, economics, business, and art.

The questions were of various lengths. The longest question had 36 words, while the smallest question had 3 words. The average question length was approximately 9 words. Approximately 18% of the questions (106) had interrogative words, while the rest (72%) had Bloom's verbs. As shown in Fig. 2, the dataset was developed so that it contains an approximately equal number of questions for each cognitive level.

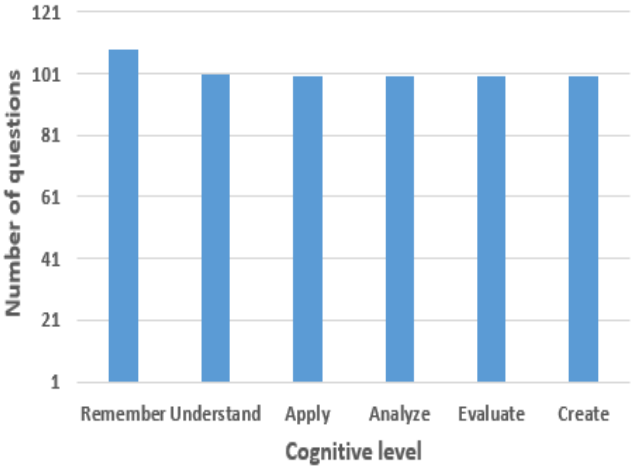

FIGURE 2. Distribution of questions per cognitive level.

B. STEP 2: PRE-PROCESSING

In the preprocessing step, the input text is structured and cleaned to prepare it for further analysis [49]. The first step of the preprocessing procedure was the removal of punctuation and extra spaces. Then, non-Arabic letters and numbers were eliminated. The last step was to remove stop-words. However, not all stop-words were removed because many interrogative words and question verbs were included in the predefined stop-words list. These words, as discussed in the previous section, are very important for the classification task and, therefore, were kept in the questions.

C. STEP 3: FEATURE EXTRACTION

As the aim of this study is to modify the traditional TF-IDF method to be suitable for extracting features from Arabic questions, it was important to compare the proposed method with the traditional TF-IDF as well as with any modified TF-IDF that has been proven suitable for extracting features from assessment questions. Thus, the proposed method would yield better results. As explained in the related work, the majority of previous studies used TF-IDF. Only Mohammed and Omar [17] used a modified TF-IDF method called TFPOS-IDF in their study.

TFPOS-IDF was not developed for assessment questions but rather to extract features from information retrieval queries. It assigns a weight to each term in a document according to its type, i.e., verb, noun, adjective, or adverb. Verbs are given the highest weight of 5, nouns and adverbs are given 3, while the other terms are given a weight of 1. These weights were selected to accommodate the nature of information retrieval queries.

In this study, our proposed method was compared with both the traditional TF-IDF and TFPOS-IDF methods. Each method was used and tested separately.

D. STEP 4: CLASSIFICATION

Five popular machine learning classifiers were used in this study: naive Bayes, support vector machines, k-nearest neighbors, logistic regression and decision trees. The selection of these five classifiers was for several reasons. First, the five classifiers were previously used for classifying assessment questions, and they all performed very well in the classification task [17], [19], [41]. These five approaches were also used for classifying Arabic text, and their performance was very good [50]–[52]. In addition, because this is the first study to explore the classification of Arabic assessment questions, it was important to examine multiple classifiers to determine which classifier would perform the best with Arabic questions. Furthermore, it was vital to test the different feature extraction methods with different classifiers to see if the performance of these methods will differ with the type of classifier.

E. STEP 5: EXAMINING THE EVALUATION RESULTS

The last step was to examine the evaluation results that were achieved by the five classifiers. To do that, several experiments were conducted. For each method, the extracted features were fed into the five classifiers. Then, the experiment was repeated 20 times. For each run, the training and testing sets were randomly selected, i.e., 80% to the training set and 20% to the testing set. Three evaluation metrics were computed and recorded: overall accuracy, recall, and precision.

Overall accuracy measures the goodness of classification as a ratio of correctly predicted instances to the total number of cases [53].

If $c_{\text{instances}}$ is the total number of predicted cases, $\hat{x}_i$ is the predicted value of the i-th instance and x_i is the corresponding true value, then the overall accuracy of the classification is calculated as follows [54]:

$$Overall\ accuracy\left(x, \hat{x}\right) = \frac{1}{c_{\text{instances}}} \sum_{i=0}^{c_{\text{instances}}-1} 1(\hat{x}_i = x_i). \quad (10)$$

Recall measures the classification perfection as the fraction of all correct results returned by the classifier. It is calculated as follows [53]:

$$recall = \frac{TP}{TP + FN}, \quad (11)$$

where true positive (TP) is the number of instances that the classifier has correctly classified and false negative (FN) is the number of instances that the classifier has not classified [54].

Precision measures the classification fineness as a ratio of correctly predicted positive instances to the total predicted positive instances. It is calculated as follows:

$$precision = \frac{TP}{TP + FP}, \quad (12)$$

where false positive (FP) is the number of instances that the classifier has incorrectly classified.

To determine if the difference in overall accuracy scores between the proposed method and the other two methods is significant, a two-sample t-test of the null hypothesis was utilized. The scores achieved by the proposed method were tested against the scores achieved by each of the two methods [55]. The hypotheses for this test were as follows:

- The Null Hypothesis (Ho): There is no significant difference between the overall accuracy scores that have been achieved by the two feature extraction methods.
- The Research Hypothesis (Ha): There is a significant difference between the overall accuracy scores achieved by the two methods.

A confidence level of 95% was employed in this study. If the p-value < 0.05, then the Ho of no significant difference in scores between the two methods will be rejected, and Ha will be accepted. If the p-value > 0.05, then Ha will be rejected, and Ho will be accepted.

To compare the performance of the five classifiers, the one-way ANOVA test was conducted with the following hypotheses:

- The Null Hypothesis (Ho): There is no significant difference between the performances of the five classifiers.
- The Research Hypothesis (Ha): There is a significant difference between the performances of the five classifiers.

However, ANOVA test results do not map out which classifier performances are different from others. As a result, a post hoc Bonferroni correction test was conducted to determine which classifiers differ significantly in performance [56].

V. RESULTS

The performance of the five classifiers (i.e., naive Bayes, support vector machine, k-nearest neighbor, logistic regression and decision trees) changed depending on the feature extraction method that was used (i.e., the proposed method, TFPOS-IDF or traditional TF-IDF).

A. NAIVE BAYES

Fig. 3 shows the overall accuracy scores that were achieved by the naive Bayes classifier when it was used with the three feature extraction methods. Over the 20 runs, naive Bayes performed the best when it was combined with the proposed method. The average accuracy of the proposed method (0.787295), as seen in Table 1, was higher than the average accuracy of the other two methods (TFOPS-IDF = 0.727782 and traditional TF-IDF = 0.727049). Similarly, the proposed method helped the classifier achieve better recall and better precision than the other methods. The complete results are shown in Appendix A.

When conducting the t-test to determine if the difference in overall accuracy scores between the proposed method and the TFPOS-IDF method is significant, the test yielded a p-value of 2.89×10^{-8}, indicating that there is a statistically

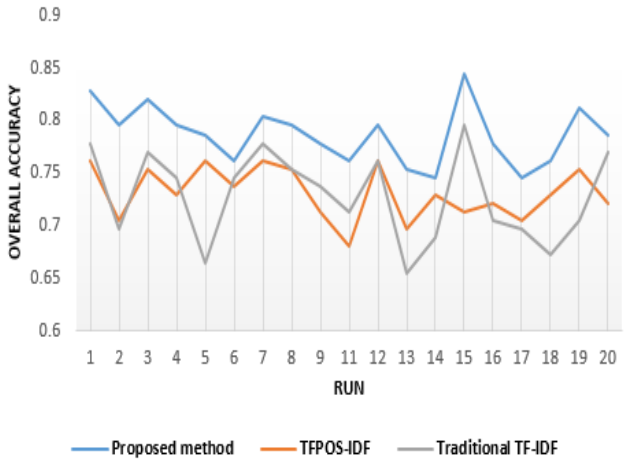

FIGURE 3. The overall accuracy of naive Bayes when it was used with each of the three feature extraction methods.

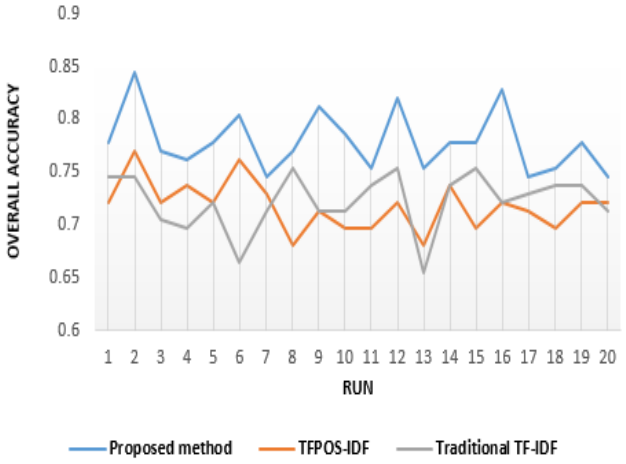

FIGURE 4. The overall accuracy of the logistic regression when it was used with each of the three feature extraction methods.

TABLE 1. Performance of the naive bayes classifier when used with each of the three feature extraction methods.

	Average accuracy	Average recall	Average precision
Proposed Method	0.787295	0.789585	0.791332
TFPOS-IDF	0.727782	0.730801	0.734149
Traditional TF-IDF	0.727049	0.729994	0.731195

TABLE 2. Performance of the logistic regression when used with each of the three feature extraction methods.

	Average accuracy	Average recall	Average precision
Proposed Method	0.779508	0.783896	0.788825
TFPOS-IDF	0.718033	0.725218	0.728673
Traditional TF-IDF	0.722541	0.722135	0.722154

significant difference between the overall accuracy scores achieved by the two methods.

Similarly, the t-test yielded a p-value of 6.01×10^{-6} for the difference in overall accuracy scores between the proposed method and the traditional TF-IDF method. This also indicates that there is a statistically significant difference between the overall accuracy scores achieved by the two methods.

B. LOGISTIC REGRESSION

Logistic regression was tested with different combinations of parameter values to find the combination that performed the best for Arabic question classification. The algorithm performed the best with the following parameter values:

- Inverse of regularization strength (C) = 3.35
- Algorithm to use in the optimization problem (solver) = 'lbfgs'
- Penalty = 'l2'

The overall accuracy scores that were achieved by the logistic regression classifier were quite similar to those achieved by the naive Bayes classifier (see Fig. 4). The support vector machine also performed the best when it was combined with the proposed method. The average accuracy of the proposed method (0.779508), as seen in Table 2, was higher than the average accuracy of the other two methods (TFPOS-IDF = 0.718033 and traditional TF-IDF = 0.722541). The proposed method was also able to help the logistic regression achieve better recall and better precision than the other methods.

The t-test indicates that the difference in overall accuracy scores between the proposed method and the two other methods is statistically significant. The t-test yielded a p-value of 8.09×10^{-9} for the difference in overall accuracy scores between the proposed method and the TFPOS-IDF method. It also yielded a p-value of 1.49×10^{-7} for the difference in overall accuracy scores between the proposed method and the traditional TF-IDF method. Both values are less than 0.05; therefore, the null hypothesis was rejected, and the research hypothesis was accepted for both comparisons.

C. SUPPORT VECTOR MACHINE

A support vector machine was also tested with different combinations of parameter values, and it performed the best with the following settings:

- Regularization parameter (C) = 10
- Kernel type = 'rbf'
- Kernel coefficient (gamma) = 0.1

The performance of the support vector machine classifier was quite similar to the performance of the naive Bayes and logistic regression classifiers (see Fig. 5). In addition, the support vector machine also performed the best when it was combined with the proposed method. The proposed method

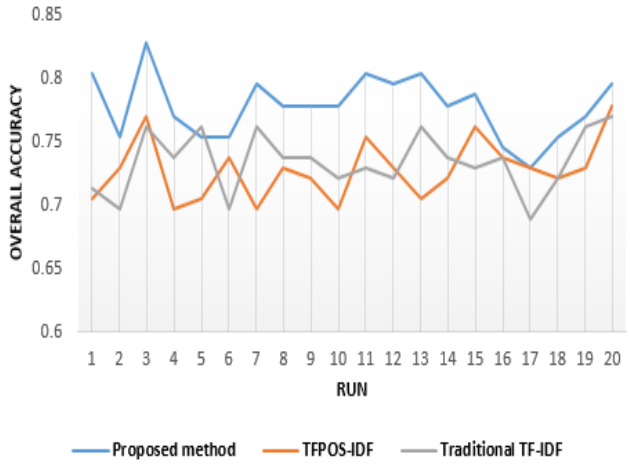

FIGURE 5. The overall accuracy of the support vector machine when it was used with each of the three feature extraction methods.

helped the support vector machine achieve an average accuracy of 0.777869 compared to 0.727869 for TFPOS-IDF and 0.734426 for traditional TF-IDF (see Table 3). The proposed method also helped the support vector machine achieve better recall and better precision than the other feature extraction methods.

TABLE 3. Performance of the support vector machine when used with each of the three feature extraction methods.

	AVERAGE ACCURACY	Average recall	Average precision
Proposed Method	0.777869	0.779845	0.780158
TFPOS-IDF	0.727869	0.726345	0.735457
Traditional TF-IDF	0.734426	0.735135	0.738378

The t-test yielded a p-value of 1.05×10^{-7} for the difference in overall accuracy scores between the proposed method and the TFPOS-IDF method. It also yielded a p-value of 1.64×10^{-6} for the difference in overall accuracy scores between the proposed method and the traditional TF-IDF method. Both values are less than 0.05, which indicates that the difference in overall accuracy scores between the proposed method and the two other methods is statistically significant.

D. K-NEAREST NEIGHBOR

Similar to the logistic regression and support vector machine, K-nearest neighbor was tested with different combinations of parameter values. It performed the best with the following settings:

- Number of neighbors = 10
- Leaf size = 1
- Power parameter (p) = 2

The performance of K-nearest neighbor was lower than the performance of the three previous classifiers (see Fig. 6). There was a slight decrease in the overall accuracy of the proposed method, but the decrease in overall accuracy of the other two feature extraction methods was comparatively larger. Overall, the proposed methods outperformed the other methods by a large margin. As seen in Table 4, the proposed method was also helpful for the K-nearest neighbor to achieve the highest precision and recall.

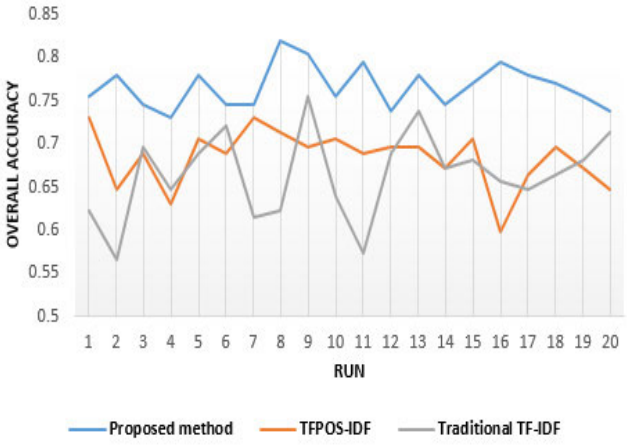

FIGURE 6. The overall accuracy of k-nearest neighbor when it was used with each of the three feature extraction methods.

TABLE 4. Performance of the k-nearest neighbor when used with each of the three feature extraction methods.

	Average accuracy	Average recall	Average precision
Proposed Method	0.765984	0.766462	0.774511
TFPOS-IDF	0.683607	0.683432	0.698041
Traditional TF-IDF	0.664344	0.665725	0.685768

Again, a t-test was conducted to determine if the difference in overall accuracy scores between the proposed method and the TFPOS-IDF method was significant. The test yielded a p-value of 1.27×10^{-10}, indicating that there is a statistically significant difference between the overall accuracy scores achieved by the two methods. The t-test also yielded a p-value of 6.8×10^{-9} for the difference in overall accuracy scores between the proposed method and the traditional TF-IDF method. This indicates that the difference between the overall accuracy scores achieved by the two methods is statistically significant.

E. DECISION TREES

Decision trees performed the best with the following parameter values:

- Maximum depth of the tree = 65

- Minimum number of samples required to split an internal node = 4
- Randomness of the estimator = 0

The performance of the decision tree classifier was the lowest of all five classifiers (see Fig. 7). However, similar to the other classifiers, the decision trees performed the best when it was combined with the proposed method. The proposed method helped the decision trees achieve an average accuracy of 0.597131 compared to 0.500863 for TFPOS-IDF and 0.508197 for traditional TF-IDF (see Table 5). The proposed method was also the best in terms of recall and precision compared to the other two methods. The complete results are shown in Appendix E.

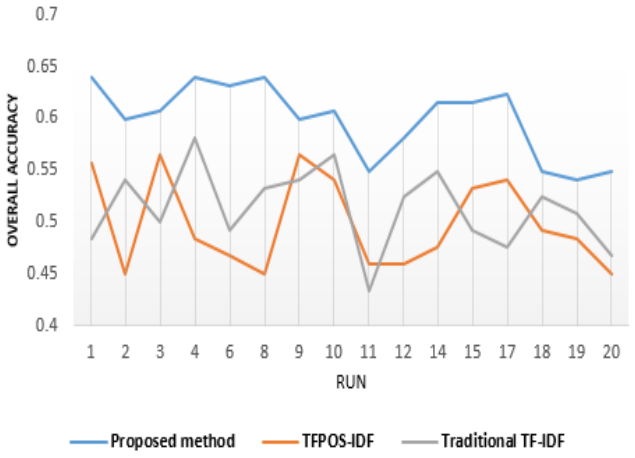

FIGURE 7. The overall accuracy of the decision trees when it was used with each of the three feature extraction methods.

TABLE 5. Performance of the decision trees when used with each of the three feature extraction methods.

	Average accuracy	Average recall	Average precision
Proposed Method	0.597131	0.59654	0.636493
TFPOS-IDF	0.500863	0.496981	0.547922
Traditional TF-IDF	0.508197	0.505037	0.580381

The difference in overall accuracy scores between the proposed method and the two other methods is statistically significant. The t-test yielded a p-value of 1.33×10^{-7} for the difference in overall accuracy scores between the proposed method and the TFPOS-IDF method. It also yielded a p-value of 1.50×10^{-8} for the difference in overall accuracy scores between the proposed method and the traditional TF-IDF method.

F. COMPARING THE PERFORMANCE OF THE FIVE CLASSIFIERS

An ANOVA test was used to examine whether the difference in performances between the five classifiers was statistically significant. The focus was on comparing the performance of the proposed method across the five classifiers as it achieved the highest performance.

As seen in Table 6, the ANOVA test yielded a p-value less than 0.05, which indicates that there is a significant difference between the performances of the five classifiers.

TABLE 6. ANOVA test results.

Source of Variation	SS	df	MS	F	P-value	*F crit*
Between Groups	0.52613	4	0.13153	163.42	1.09×10^{-41}	2.4675
Within Groups	0.07646	95	0.00081			
Total	0.60259	99				

The post hoc test shows that there is no significant difference in performance between naive Bayes and support vector machines or between naive Bayes and logistic regression. Meanwhile, there is a statistically significant difference between naive Bayes and k-nearest neighbors and between naive Bayes and decision trees (see Table 7).

TABLE 7. Post Hoc bonferroni correction results.

Comparison	p-value (t-test)	Significant?
Naive Bayes vs Support Vector Machine	0.53552	No
Naive Bayes vs Logistic Regression	0.29309	No
Naive Bayes vs K-Nearest Neighbor	0.00311	Yes
Naive Bayes vs Decision Trees	3.07×10^{-19}	Yes

VI. DISCUSSION

This study acknowledges the importance of automating the process of classifying the cognitive levels of Arabic questions according to Bloom's taxonomy. As the first study to examine this text classification task, it started by developing a dataset of more than 600 questions. This is larger than the size of any of the other datasets that have been used in previous studies to classify English questions [17], [19], [41]. The dataset includes questions from different academic disciplines. The questions were of various lengths. Some of them start with Bloom's verbs, while others include interrogative words. The aim was to adequately cover the space of possible inputs and help the classifier generalize better, as recommended by [57].

Most importantly, the study introduces a modified TF-IDF method to extract features from Arabic assessment questions. This method outperformed the traditional TF-IDF and

a modified TF-IDF TFPOS-IDF that has been used with English questions. A t-test has shown that the difference in performance between the proposed method and the other two methods is statistically significant. This agrees with the findings of other studies [17], [37], [42], [44] that while the traditional TF-IDF can be very useful for feature extraction, using it 'as is' might not yield the best result. Careful consideration of the data being classified and modification of the traditional TF-IDF accordingly could have a significant positive impact on the quality of classification.

The proposed method performed the best when it was combined with the naive Bayes algorithm. However, the difference in performance between naive Bayes and support vector machines and between naive Bayes and logistic regression was not found to be statistically significant. This agrees with the findings of Al-Saqqa, *et al.* [58], Nabil, *et al.* [59], El-Masri, *et al.* [60] and Nieuwenhuis and Wilkens [61], which indicate that naive Bayes, support vector machine and logistic regression can outperform other classifiers at classifying short Arabic text.

In terms of complexity, the proposed method does not add much to the computational complexity already encountered in the traditional TF-IDF and TFPOS-IDF methods. It does not require any additional parameters to be calculated. It uses the same parameters that the traditional TF-IDF and TFPOS-IDF methods use, i.e., the number of times a term appears in a document and the total number of terms in that document. Its computation includes only some additional divisions and additions but does not need any iteration to be performed. Thus, it can be said that the three feature extraction methods that have been discussed in this paper do not differ in terms of complexity, but our proposed method outperforms the other two methods in terms of performance.

Overall, the contributions of this paper can be summarized as follows:

1) It developed the first dataset for classifying Arabic assessment questions according to Bloom's Taxonomy. This dataset would be publicly available for researchers who want to further investigate the discussed classification task.
2) It introduced a modified TF-IDF method to extract features from Arabic assessment questions. The proposed method has proven to be superior to the traditional TF-IDF and TFPOS-IDF methods. As it has the unique advantage of taking the nature of questions into consideration, the proposed method can be extended to other languages and has the potential to yield better results with these languages.
3) It provided evidence that when extracting features from a corpus, the nature of that corpus needs to be taken into consideration, and the feature extraction method should be modified accordingly. While a general method can be used as-is and still produces a satisfactory result, a modified method can yield superior results.

Although very valuable, this study still has some limitations. The developed dataset contains only short answer and essay questions. Other types of questions, such as multiple-choice, problem-based, and true/false, were not included. While this is the norm for similar datasets that have been developed for English questions [17], [19], [41], [48], the absence of such questions would limit the application of the proposed method.

VII. CONCLUSION

This study looked at automating the process of classifying Arabic assessment questions according to Bloom's taxonomy. It developed a new dataset for this classification task. It also proposed a modified TF-IDF method for extracting features from Arabic questions. The proposed method outperformed the traditional TF-IDF as well as a modified one called TFPOS-IDF that was used for classifying English questions. The overall accuracy, precision, and recall scored by the proposed method were significantly higher than those scored by the other two methods. This indicates the promising potential of this method.

This study has opened up new research areas for further improvements and future work. The work presented in this paper could be extended in different ways:

1) By including additional types of questions in the training dataset, i.e., multiple-choice, problem-based, and true/false questions. This will widen the applicability of the proposed work.
2) By combining the proposed method with a word embedding method. This combination of methods would be able to extract more features from questions, i.e., semantic features. This could yield better results, as has been shown in previous studies.
3) Most importantly, more recent methods, such as extreme learning machines (ELMs) or long short-term memory (LSTM) with autoencoders, can be used to compare the performances and prove the performance of our fault diagnosis.

APPENDIX A

See Table 8.

APPENDIX B

See Table 9.

APPENDIX C

See Table 10.

APPENDIX D

See Table 11.

APPENDIX E

See Table 12.

TABLE 8. Performance of naïve Bayes when used with each of the three feature extraction methods.

	Propose Method			TFPOS-IDF			Traditional TF-IDF		
Run	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision
1	0.827869	0.822337	0.82795	0.762295	0.750017	0.750148	0.778689	0.783946	0.792582
2	0.795082	0.81193	0.799123	0.704918	0.714862	0.719788	0.696721	0.689963	0.697175
3	0.819672	0.820546	0.820573	0.754098	0.760984	0.761147	0.770492	0.7679	0.782631
4	0.795082	0.798754	0.792437	0.729508	0.745458	0.736964	0.745902	0.754304	0.747964
5	0.786885	0.7846	0.810864	0.762295	0.763043	0.778297	0.663934	0.660417	0.663429
6	0.762295	0.764895	0.773296	0.737705	0.749345	0.759649	0.745902	0.747924	0.753829
7	0.803279	0.807021	0.803741	0.762295	0.763828	0.769577	0.778689	0.781096	0.779514
8	0.795082	0.801712	0.797664	0.754098	0.76327	0.75955	0.754098	0.754046	0.762146
9	0.778689	0.787132	0.789083	0.713115	0.725884	0.722092	0.737705	0.75169	0.745827
10	0.795082	0.792543	0.792776	0.696721	0.692186	0.702423	0.704918	0.706547	0.706207
11	0.762295	0.748476	0.754453	0.680328	0.673903	0.690091	0.713115	0.71171	0.712628
12	0.795082	0.801347	0.806347	0.762295	0.765248	0.760907	0.762295	0.758354	0.756782
13	0.754098	0.738194	0.744505	0.696721	0.686591	0.700216	0.655738	0.653844	0.650023
14	0.745902	0.740797	0.747437	0.729508	0.738239	0.737751	0.688525	0.695557	0.692649
15	0.844262	0.852695	0.846515	0.713115	0.699596	0.708878	0.795082	0.817564	0.790476
16	0.778689	0.771132	0.772622	0.721311	0.714542	0.729682	0.704918	0.69927	0.724372
17	0.745902	0.776388	0.768687	0.704918	0.699378	0.707258	0.696721	0.705403	0.719018
18	0.762295	0.769209	0.774832	0.729508	0.721072	0.714758	0.672131	0.687365	0.673597
19	0.811475	0.814776	0.819214	0.754098	0.760106	0.75726	0.704918	0.698191	0.702796
20	0.786885	0.787217	0.784525	0.721311	0.747693	0.732549	0.770492	0.774785	0.770247
Avg	**0.787295**	**0.789585**	**0.791332**	**0.727782**	**0.730801**	**0.734149**	**0.727049**	**0.729994**	**0.731195**

TABLE 9. Performance of logistic regression when used with each of the three feature extraction methods.

	Propose Method			TFPOS-IDF			Traditional TF-IDF		
Run	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision
1	0.7787	0.777935	0.80811	0.7213	0.72599	0.734652	0.7459	0.762573	0.753951
2	0.8443	0.844217	0.851783	0.7705	0.793497	0.77828	0.7459	0.744279	0.752809
3	0.7705	0.794674	0.792577	0.7213	0.721907	0.729574	0.7049	0.70754	0.705611
4	0.7623	0.755934	0.764669	0.7377	0.739844	0.73828	0.6967	0.702983	0.683977
5	0.7787	0.789801	0.792098	0.7213	0.719112	0.738686	0.7213	0.719663	0.714266
6	0.8033	0.812472	0.813026	0.7623	0.774473	0.765545	0.6639	0.654348	0.650643
7	0.7459	0.744333	0.758537	0.7295	0.717501	0.726919	0.7131	0.711615	0.713374
8	0.7705	0.769216	0.777492	0.6803	0.701587	0.698204	0.7541	0.75508	0.754916
9	0.8115	0.818866	0.817723	0.7131	0.723358	0.723092	0.7131	0.694132	0.706394
10	0.7869	0.781692	0.786377	0.6967	0.70874	0.709062	0.7131	0.705501	0.716033
11	0.7541	0.757989	0.761749	0.6967	0.729282	0.707298	0.7377	0.728935	0.734573
12	0.8197	0.822004	0.822015	0.7213	0.734823	0.736478	0.7541	0.761288	0.763704
13	0.7541	0.764479	0.756949	0.6803	0.677721	0.700015	0.6557	0.660075	0.663005
14	0.7787	0.787507	0.79438	0.7377	0.713825	0.739189	0.7377	0.737376	0.735376
15	0.7787	0.788447	0.793056	0.6967	0.719192	0.713464	0.7541	0.75129	0.75002
16	0.8279	0.835202	0.830184	0.721311	0.720302	0.733194	0.7213	0.720311	0.723306
17	0.7459	0.748912	0.759496	0.7131	0.723358	0.723092	0.7295	0.727052	0.734762
18	0.7541	0.770978	0.753803	0.6967	0.697412	0.70725	0.7377	0.748615	0.741152
19	0.7787	0.772908	0.796067	0.721311	0.736448	0.736541	0.7377	0.734106	0.742905
20	0.7459	0.740345	0.746409	0.7213	0.72599	0.734652	0.7131	0.715937	0.70231
Avg	**0.779508**	**0.783896**	**0.788825**	**0.718033**	**0.725218**	**0.728673**	**0.722541**	**0.722135**	**0.722154**

TABLE 10. Performance of support vector machine when used with each of the three feature extraction methods.

	Propose Method			TFPOS-IDF			Traditional TF-IDF		
Run	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision
1	0.8033	0.804116	0.805687	0.7049	0.709353	0.71068	0.7131	0.713819	0.711524
2	0.7541	0.756347	0.761153	0.7295	0.709986	0.744846	0.6967	0.695887	0.701611
3	0.8279	0.841168	0.82915	0.7705	0.785156	0.78991	0.7623	0.769435	0.775119
4	0.7705	0.757037	0.757125	0.6967	0.703398	0.715297	0.7377	0.735772	0.731273
5	0.7541	0.757144	0.755372	0.7049	0.704873	0.714972	0.7623	0.745657	0.753038
6	0.7541	0.757684	0.758914	0.7377	0.72815	0.73782	0.6967	0.726045	0.700789
7	0.7951	0.803212	0.793866	0.6967	0.696014	0.700078	0.7623	0.769435	0.775119
8	0.7787	0.782644	0.785086	0.7295	0.709986	0.744846	0.7377	0.732331	0.734089
9	0.7787	0.78405	0.780672	0.7213	0.703219	0.704316	0.7377	0.73643	0.740521
10	0.7787	0.770538	0.772522	0.6967	0.703925	0.708213	0.7213	0.720221	0.735434
11	0.8033	0.799141	0.801763	0.7541	0.749874	0.750451	0.7295	0.728325	0.733321
12	0.7951	0.795342	0.798462	0.7295	0.724124	0.721019	0.7213	0.742008	0.730759
13	0.8033	0.804116	0.805687	0.7049	0.716255	0.724713	0.7623	0.731481	0.787045
14	0.7787	0.798512	0.783038	0.7213	0.723609	0.720104	0.7377	0.737401	0.742439
15	0.7869	0.768991	0.778439	0.7623	0.758887	0.762734	0.7295	0.727907	0.725841
16	0.7459	0.748709	0.750887	0.7377	0.762005	0.755411	0.7377	0.732331	0.734089
17	0.7295	0.745651	0.734784	0.7295	0.709986	0.744846	0.6885	0.704449	0.688504
18	0.7541	0.754951	0.778302	0.7213	0.724114	0.742408	0.7213	0.720221	0.735434
19	0.7705	0.772205	0.773797	0.7295	0.723676	0.735809	0.7623	0.756825	0.758594
20	0.7951	0.795342	0.798462	0.7787	0.780303	0.780676	0.7705	0.776728	0.773025
Avg	**0.777869**	**0.779845**	**0.780158**	**0.727869**	**0.726345**	**0.735457**	**0.734426**	**0.735135**	**0.738378**

TABLE 11. Performance of k-nearest neighbor when used with each of the three feature extraction methods.

	Propose Method			TFPOS-IDF			Traditional TF-IDF		
Run	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision
1	0.754098	0.752815	0.794598	0.7295	0.72868	0.747211	0.623	0.627595	0.662983
2	0.778689	0.786695	0.783539	0.6475	0.663645	0.66528	0.5656	0.567742	0.606948
3	0.745902	0.732584	0.741129	0.6885	0.68112	0.678305	0.6967	0.673061	0.722018
4	0.729508	0.724978	0.726223	0.6311	0.619655	0.616068	0.6475	0.674432	0.667073
5	0.778689	0.77788	0.774897	0.7049	0.713594	0.728479	0.6885	0.690626	0.723682
6	0.745902	0.743218	0.779006	0.6885	0.691745	0.688788	0.7213	0.723467	0.737178
7	0.745902	0.740163	0.770208	0.7295	0.728032	0.75607	0.6148	0.616994	0.640956
8	0.819672	0.831861	0.829805	0.7131	0.732051	0.741839	0.623	0.621487	0.629513
9	0.803279	0.797096	0.808642	0.6967	0.680489	0.746126	0.7541	0.74804	0.761693
10	0.754098	0.747725	0.770394	0.7049	0.713151	0.702355	0.6393	0.632526	0.670407
11	0.795082	0.813036	0.800258	0.6885	0.681558	0.682835	0.5738	0.584147	0.634347
12	0.737705	0.739493	0.745064	0.6967	0.675011	0.675917	0.6885	0.678701	0.72709
13	0.778689	0.77207	0.777511	0.6967	0.688667	0.725224	0.7377	0.748122	0.740898
14	0.745902	0.757578	0.751755	0.6721	0.694444	0.712443	0.6721	0.656398	0.676497
15	0.770492	0.769501	0.777347	0.7049	0.721506	0.714131	0.6803	0.688718	0.720668
16	0.795082	0.813036	0.800258	0.5984	0.586291	0.657057	0.6557	0.650306	0.663377
17	0.778689	0.782914	0.807089	0.6639	0.66354	0.667704	0.6475	0.654826	0.639596
18	0.770492	0.767324	0.764058	0.6967	0.669172	0.696315	0.6639	0.658667	0.675387
19	0.754098	0.749032	0.760202	0.6721	0.698024	0.695327	0.6803	0.696714	0.687351
20	0.737705	0.730237	0.728242	0.6475	0.638266	0.663345	0.7131	0.721936	0.727695
Avg	**0.765984**	**0.766462**	**0.774511**	**0.683607**	**0.683432**	**0.698041**	**0.664344**	**0.665725**	**0.685768**

TABLE 12. Performance of decision trees when used with each of the three feature extraction methods.

	Propose Method			TFPOS-IDF			Traditional TF-IDF		
Run	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision	Overall Accuracy	Recall	Precision
1	0.6393	0.611396	0.645507	0.5574	0.571859	0.608193	0.4836	0.442249	0.576964
2	0.5984	0.587208	0.617555	0.4508	0.47289	0.499724	0.541	0.534448	0.61138
3	0.6066	0.605548	0.684259	0.5656	0.579189	0.604955	0.5	0.460673	0.504714
4	0.6393	0.64799	0.695587	0.4836	0.506215	0.529734	0.582	0.585313	0.621494
5	0.5738	0.546094	0.602097	0.5082	0.491853	0.539667	0.5082	0.502273	0.549026
6	0.6311	0.615659	0.676848	0.4672	0.458049	0.490281	0.4918	0.502662	0.564587
7	0.6475	0.655689	0.684962	0.6393	0.619253	0.665273	0.4098	0.402491	0.495077
8	0.6393	0.638297	0.660218	0.4508	0.441455	0.521251	0.5328	0.540854	0.61462
9	0.5984	0.601393	0.645543	0.5656	0.558286	0.626195	0.541	0.535045	0.635952
10	0.6066	0.613332	0.698551	0.541	0.526257	0.571516	0.5656	0.546512	0.645698
11	0.5492	0.550736	0.611291	0.459	0.442698	0.478278	0.4344	0.4418	0.524566
12	0.582	0.584537	0.604109	0.459	0.461182	0.571911	0.5246	0.525947	0.632579
13	0.541	0.546088	0.563355	0.4508	0.419818	0.501157	0.5	0.527566	0.586664
14	0.6148	0.627322	0.657086	0.4754	0.444725	0.512813	0.5492	0.553761	0.699068
15	0.6148	0.598611	0.626226	0.5328	0.530308	0.601957	0.4918	0.502249	0.526773
16	0.5984	0.603567	0.610648	0.5	0.488545	0.541052	0.5328	0.516534	0.541879
17	0.623	0.623948	0.670449	0.541	0.52457	0.558589	0.4754	0.46293	0.520476
18	0.5492	0.54862	0.589885	0.4918	0.518048	0.546145	0.5246	0.518344	0.639198
19	0.541	0.528218	0.549185	0.4836	0.507734	0.553771	0.5082	0.526166	0.606824
20	0.5492	0.571724	0.594281	0.4508	0.451557	0.496251	0.4672	0.472928	0.510072
Avg	**0.597131**	**0.59654**	**0.636493**	**0.500863**	**0.496981**	**0.547922**	**0.508197**	**0.505037**	**0.580381**

REFERENCES

[1] B. Tarman and B. Kuran, "Examination of the cognitive level of questions in social studies textbooks and the views of teachers based on Bloom taxonomy," *Educ. Sci., Theory Pract.*, vol. 15, no. 1, pp. 213–222, 2015.

[2] M. T. Chandio, S. M. Pandhiani, and S. Iqbal, "Bloom's taxonomy: Improving assessment and teaching-learning process," *J. Educ. Educ. Develop.*, vol. 3, no. 2, p. 19, Dec. 2016.

[3] L. W. Anderson and D. R. Krathwohl, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York, NY, USA: Longman, 2001.

[4] F. King, L. Goodson, and F. Rohani, *Higher Order Thinking Skills: Definition, Teaching Strategies, Assessment*. Tallahassee, FL, USA: Center for Advancement of Learning and Assessment, 2013.

[5] B. Tanujaya, J. Mumu, and G. Margono, "The relationship between higher order thinking skills and academic performance of student in mathematics instruction," *Int. Educ. Stud.*, vol. 10, no. 11, pp. 78–85, 2017.

[6] S. DeDecker, R. Clemmer, K. Gordon, and J. Vale, "How do engineering students react to memorization vs. Problem analysis questions on exams?" in *Proc. Can. Eng. Educ. Assoc. (CEEA)*, 2020, pp. 1–6.

[7] A. Martone and S. G. Sireci, "Evaluating alignment between curriculum, assessment, and instruction," *Rev. Educ. Res.*, vol. 79, no. 4, pp. 1332–1361, Dec. 2009.

[8] J. Biggs, "Enhancing teaching through constructive alignment," *Higher Educ.*, vol. 32, no. 3, pp. 347–364, Oct. 1996.

[9] J. L. Momsen, T. M. Long, S. A. Wyse, and D. Ebert-May, "Just the facts? Introductory undergraduate biology courses focus on low-level cognitive skills," *CBE—Life Sci. Educ.*, vol. 9, no. 4, pp. 435–440, 2010.

[10] T. M. El-Gohary, "Multiple choice questions exams: Guiding principles to develop exam designing skills among novice& inexperienced academics," *Int. J. Therapies Rehabil. Res.*, vol. 6, no. 3, pp. 42–48, Feb. 2017.

[11] K. O. Jones, J. Harland, J. M. V. Reid, and R. Bartlett, "Relationship between examination questions and Bloom's taxonomy," in *Proc. 39th IEEE Frontiers Educ. Conf.*, San Antonio, TX, USA, Oct. 2009, pp. 1–6.

[12] K. Lightfoot and D. Schwager, "Alignment of course objectives and assessment items: A case study," in *Cases on Assessment and Evaluation in Education Hershey*. Hershey, PA, USA: IGI Global, 2013, pp. 1–17.

[13] L. M. Martin, "Using assessment of student learning outcomes to measure university performance: Towards a viable model," Melbourne Graduate School Educ., Univ. Melbourne, Melbourne, VIC, Australia, 2016.

[14] L. Meda and A. J. Swart, "Analysing learning outcomes in an electrical engineering curriculum using illustrative verbs derived from Bloom's taxonomy," *Eur. J. Eng. Educ.*, vol. 43, no. 3, pp. 399–412, May 2018.

[15] P. W. Airasian, *Classroom Assessment: Concepts and Applications*. Stockholm, Swedish: ERIC, 2001.

[16] K. Exley. (2010). *Writing Good Exam Questions*. Accessed: Jun. 8, 2020. [Online]. Available: https://www.bezaspeaks.com/eassessmentafrica/writinggoodexamquestions.pdf

[17] M. Mohammed and N. Omar, "Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec," *PLoS ONE*, vol. 15, no. 3, pp. 1–21, Mar. 2020.

[18] S. Sulaiman, R. A. Wahid, A. H. Ariffin, and C. Z. Zulkifli, "Question classification based on cognitive levels using linear SVC," *Test Eng. Manag.*, vol. 83, pp. 6463–6470, Mar. 2020.

[19] A. Aninditya, M. A. Hasibuan, and E. Sutoyo, "Text mining approach using TF-IDF and naive Bayes for classification of exam questions based on cognitive level of Bloom's taxonomy," in *Proc. IEEE Int. Conf. Internet Things Intell. Syst. (IoTaIS)*, Bali, Indonesia, Nov. 2019, pp. 112–117.

[20] A. Osman and A. A. Yahya, "Classifications of exam questions using natural language syntatic features: A case study based on Bloom's taxonomy," in Proc. 6th Int. Arab Conf. Quality Assurance Higher Educ., 2016, pp. 1–8.

[21] L. W. Anderson and L. A. Sosniak, *Bloom's Taxonomy*. Chicago, IL, USA: Univ. Chicago Press 1994.

[22] M. J. Pickard, "The new Bloom's taxonomy: An overview for family and consumer sciences," *J. Family Consum. Sci. Educ.*, vol. 25, no. 1, pp. 1–11, 2007.

[23] (2020). *The Quality Assurance Agency for Higher Education (QAA)*. [Online]. Available: https://www.qaa.ac.uk/

[24] S. A. D. Popenici and S. Kerr, "Exploring the impact of artificial intelligence on teaching and learning in higher education," *Res. Pract. Technol. Enhanced Learn.*, vol. 12, no. 1, p. 22, Dec. 2017.

[25] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: An overview," *J. Phys., Conf. Ser.*, vol. 1142, no. 1, 2018, Art. no. 012012.

[26] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West, "Predicting student dropout in higher education," 2016, *arXiv:1606.06364*. [Online]. Available: http://arxiv.org/abs/1606.06364

[27] A. A. Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani, "Prediction of student course selection in online higher education institutes using neural network," *Comput. Educ.*, vol. 65, pp. 1–11, Jul. 2013.

[28] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi, "Student engagement predictions in an e-learning system and their impact on student course assessment scores," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–21, Oct. 2018.
[29] A. Amigud, J. Arnedo-Moreno, T. Daradoumis, and A.-E. Guerrero-Roldan, "Using learning analytics for preserving academic integrity," *Int. Rev. Res. Open Distrib. Learn.*, vol. 18, no. 5, pp. 192–210, Aug. 2017.
[30] Z. Omary and F. Mtenzi, "Machine learning approach to identifying the dataset threshold for the performance estimators in supervised learning," *Int. J. Infonomics*, vol. 3, no. 3, pp. 314–325, 2010.
[31] K. K. Hyde, M. N. Novack, N. LaHaye, C. Parlett-Pelleriti, R. Anden, D. R. Dixon, and E. Linstead, "Applications of supervised machine learning in autism spectrum disorder research: A review," *Rev. J. Autism Develop. Disorders*, vol. 6, no. 2, pp. 128–146, Jun. 2019.
[32] B. Baharudin, L. H. Lee, and K. Khan, "A review of machine learning algorithms for text-documents classification," *J. Adv. Inf. Technol.*, vol. 1, no. 1, pp. 4–20, Feb. 2010.
[33] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *J. Emerg. Technol. Web Intell.*, vol. 1, no. 1, pp. 60–76, Aug. 2009.
[34] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining—An overview," *Int. J. Comput. Sci. Commun. Netw.*, vol. 5, no. 1, pp. 7–16, 2015.
[35] M. Rajman and R. Besançon, "Text mining: Natural language techniques and text mining applications," in *Data Mining and Reverse Engineering*. Boston, MA, USA: Springer, 1998, pp. 50–64.
[36] A. Onan and M. A. Toçoğlu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021.
[37] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, 2003, vol. 242, no. 1, pp. 29–48.
[38] Y. Goldberg and O. Levy, "Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*. [Online]. Available: http://arxiv.org/abs/1402.3722
[39] L. Akhtyamova, P. Martinez, K. Verspoor, and J. Cardiff, "Testing contextualized word embeddings to improve NER in Spanish clinical case narratives," *IEEE Access*, vol. 8, pp. 164717–164726, 2020.
[40] K. Jayakodi, M. Bandara, I. Perera, and D. Meedeniya, "WordNet and cosine similarity based classifier of exam questions using Bloom's taxonomy," *Int. J. Emerg. Technol. Learn.*, vol. 11, no. 4, p. 142, Apr. 2016.
[41] K. Osadi, M. Fernando, and W. Welgama, "Ensemble classifier based approach for classification of examination questions into Bloom's taxonomy cognitive levels," *Int. J. Comput. Appl.*, vol. 162, no. 4, pp. 76–92, Mar. 2017.
[42] R. K. Roul, J. K. Sahoo, and K. Arora, "Modified TF-IDF term weighting strategies for text categorization," in *Proc. 14th IEEE India Council Int. Conf. (INDICON)*, Dec. 2017, pp. 1–6.
[43] G.-W. Kim and D.-H. Lee, "Intelligent health diagnosis technique exploiting automatic ontology generation and Web-based personal health record services," *IEEE Access*, vol. 7, pp. 9419–9444, 2019.
[44] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, "Hot topic detection based on a refined TF-IDF algorithm," *IEEE Access*, vol. 7, pp. 26996–27007, 2019.
[45] W.-C. Chang and M.-S. Chung, "Automatic applying Bloom's taxonomy to classify and analysis the cognition level of English question items," in *Proc. Joint Conf. Pervas. Comput. (JCPC)*, Tamsui, Taiwan, Dec. 2009, pp. 727–734.
[46] A. A. Abual-Hamael. *Classroom Questions*. Accessed: Mar. 6, 2021. [Online]. Available: https://www.kau.edu.sa/search.aspx?Site_ID=0&lng=EN&q=%d8%a7%d9%84%d8%a7%d8%b3%d8%a6%d9%84%d8%a9%20%d8%a7%d9%84%d8%b5%d9%81%d9%8a%d8%a9
[47] I. S. Al-Mzoughi, "An evaluation of final examinations questions according to Bloom's classification of cognitive aims," *Sabratha Univ. Sci. J.*, vol. 3, no. 1, pp. 92–107, 2018.
[48] A. A. Yahya and A. Osman, "Automatic classification of questions into Bloom's cognitive levels using support vector machines," in *Proc. Int. Arab Conf. Inf. Technol. (ACIT)*, Riyadh, Saudi Arabia, 2011, pp. 335–342.
[49] D. Meyer, K. Hornik, and I. Feinerer, "Text mining infrastructure in R," *J. Stat. Softw.*, vol. 25, no. 5, pp. 1–54, Mar. 2008.
[50] A. H. Mohammad, T. Alwada'n, and O. Al-Momani, "Arabic text categorization using support vector machine, Naïve Bayes and neural network," *GSTF J. Comput.*, vol. 5, no. 1, pp. 108–115, Sep. 2016.
[51] A. M. El-Halees, "A comparative study on Arabic text classification," *Egyptian Comput. Sci. J.*, vol. 30, no. 2, pp. 1–11, 2008.
[52] R. Alhutaish and N. Omar, "Arabic text classification using k-nearest neighbour algorithm," *Int. Arab J. Inf. Technol.*, vol. 12, no. 2, pp. 190–195, 2015.
[53] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine Learning, Neural and Statistical Classification* (Artificial Intelligence). Cambridge, U.K.: Ellis Horwood, 1994, p. 289.
[54] scikit-learn developers. (Mar. 2, 2021). *Metrics and Scoring: Quantifying the Quality of Predictions*. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html
[55] T. K. Kim, "T test as a parametric statistic," *Korean J. Anesthesiol.*, vol. 68, no. 6, p. 540, 2015.
[56] R. A. Armstrong, "When to use the B onferroni correction," *Ophthalmic Physiol. Opt.*, vol. 34, no. 5, pp. 502–508, 2014.
[57] Google Developers. (Mar. 4, 2021). *Machine Learning Guides-Text classification*. [Online]. Available: https://developers.google.com/machine-learning/guides/text-classification
[58] S. Al-Saqqa, N. Obeid, and A. Awajan, "Sentiment analysis for Arabic text using ensemble learning," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. 1–7.
[59] M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2515–2519.
[60] M. El-Masri, N. Altrabsheh, H. Mansour, and A. Ramsay, "A Web-based tool for Arabic sentiment analysis," *Procedia Comput. Sci.*, vol. 117, pp. 38–45, Jan. 2017.
[61] M. Nieuwenhuis and J. Wilkens, "Twitter text and image gender classification with a logistic regression n-gram model," in *Proc. 9th Int. Conf. CLEF Assoc. (CLEF)*, 2018, pp. 1–9.

ALI SALEH ALAMMARY received the master's and Ph.D. degrees from Monash University, Australia, in 2011 and 2016, respectively. He is currently an Associate Professor with the College of Computing and Informatics, Saudi Electronic University. He is also the General Supervisor of the University Branch, Jeddah. His current research interests include blockchain technology, educational technologies, machine learning, text mining, and software security.