Journal of Engineering Science and Technology
Vol. 14, No. 2 (2019) 935 - 950

A COMBINATION METHOD OF SYNTACTIC AND SEMANTIC APPROACHES FOR CLASSIFYING EXAMINATION QUESTIONS INTO BLOOM'S TAXONOMY COGNITIVE

OMAR J. MOHAMED[1,*], NAWAF A. ZAKAR[2], BASEL ALSHAIKHDEEB[3]

[1,2]Department of Computer Science, University of Al-Hamadaniyah, Iraq
[3]Department of Strategic Information Systems, Faculty of Information Science and Technology, National University of Malaysia
[*]Corresponding Author: omar_jamal_m@yahoo.com

Abstract

Bloom's taxonomy has been proposed for categorizing examination questions in accordance with the student's cognitive ability. Recently, researchers tend to utilize machine learning techniques in order to classify the questions. However, there is still a remarkable limitation, which can be represented by the ambiguity lies on the question. Due to the short length of the question, it is difficult to identify the contextual information of the words. This means that a single word could yield multiple meanings. This would significantly affect the process of classification especially for the verbs that are usually located within the question such as 'define' or 'write'. The ambiguity of such verbs would mislead the classification process regarding Bloom's cognitive levels. Therefore, this study aims to propose a combination method of semantic and syntactic approaches in order to overcome such drawback. The semantic approach aims to utilize an external knowledge source in order to retrieve semantic correspondences. Whereas, the syntactic approach aims to determine the syntactic tag of the terms to address the significant verbs and nouns. Finally, three machine learning techniques will be used including Support Vector Machine, J48 and Naïve Bayes classifiers to classify the questions. In order to assess the effectiveness of the proposed combination method, the classifiers have been applied with the proposed combination and without it. Results revealed that the classifiers with the combination method have outperformed the traditional ones. This implies the significance of using the proposed semantic and syntactic approaches.

Keywords: Bloom's taxonomy, J48, Lesk algorithm, Naive bayes, POS tagging, Question classification, Support vector machine, WordNet.

1. Introduction

In the process of forming examination questions, it is necessary to assess these questions in order to be more suitable for the student's cognitive ability [1]. Bloom's taxonomy as one of the famous exam questions categorization framework where the question is associated with six classes in terms of cognitive levels [2].

These classes, which are depicted in Fig. 1, consist of Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. Knowledge class aims to examine the student's ability in terms of recognition, recalling and defining a concept using keywords such as, "define, list, label, identify, etc." [3].

Comprehension class is considered to be an upper class of knowledge in which, a higher complicated ability is being addressed such as explaining, illustrating and describing more details of a concept. The keywords that associated with comprehension class consist of "elaborate, discuss, explain, etc.".

Application class, in the same manner, requires the higher ability of comprehension in which, the student is asked to formulate an application based on his illustration. The keywords that associated with this class consist of "apply, carry out, demonstrate, etc.".

Analysis class is similarly addressing higher ability where the analysis capability of the student is being examined by analysing the impact of an application and its consequences. The keywords associated with class consist of "analyse, examine, distinguish, etc."

Synthesis class examines the higher level of cognitive in which, the student is required to combine multiple and existing components in order to form a new approach. The key distinguish between Synthesis and Application classes is that Synthesis required a new application rather than applying an existing one. The keywords associated with the synthesis class consist of "create, initiate, develop, etc.".

Finally, Evaluation class is considered to be the highest level of Bloom's taxonomy where the student is required to provide an evaluation of the new approach that was produced by the synthesis phase. Keywords associated with this class consist of "justify, evaluate, assess, etc.".

Recently, several research efforts have been conducted to make the process of classifying the examination questions in an automatic manner [4]. In this vein, the machine learning techniques have been utilized for such purpose in which, a set of labelled questions as historical data is being used for training [5]. During the training, machine-learning techniques would have the ability to generate a statistical model that estimates the probability of the question's class label.

Different machine learning techniques have been proposed for the automatic classification of exam questions [6-11]. Some of these studies have improved the classification performance by combining two or more classifiers. Others have enhanced the classification by utilizing some statistical features. However, there is a significant limitation still face the field of question classification into Blooms.

Obviously, most of the questions contain verbs that play an essential role in terms of identifying the cognitive level of such a question. For example, the verbs 'list', 'define', 'identify' and 'label' are strongly related to the class 'Knowledge' in the cognitive level of Bloom's taxonomy. However, due to the uniqueness of each domain of interest, there would be ambiguous associations with these verbs. For instance, in the field of computer science, the question 'write a code to reverse string' contains a verb that could be related to 'Application' or 'Synthesis' Bloom's classes. Whereas, in the domain of economy, the question 'write an essay to describe inflation' contains the same verb 'write' but it is related to a different class, which is 'comprehension'.

Such ambiguity could be eliminated using an external knowledge source that has the ability to identify semantic correspondences [12]. Therefore, this study aims to propose a combination method of syntactic and semantic approaches in order to classify domain-dependent and domain-independent question into Bloom's taxonomy.

The paper has been organized as; Section 2 analyses the related work, Section 3 discusses the proposed method and its components, Section 4 highlights the results obtained by the proposed method and provides a comparison against the state of the art.

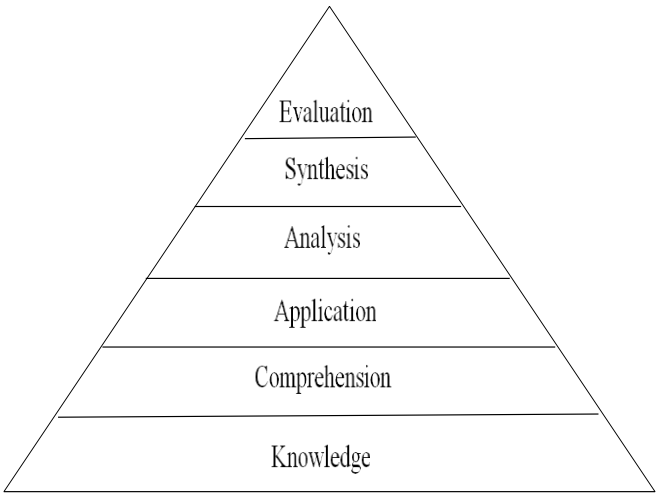

Fig. 1. Bloom's hierarchy [13].

2. Related Work

Zhang and Lee [6] have proposed a method based on machine learning techniques in order to categorize questions into Bloom's cognitive levels in which, several research studies have been presented in the field of exam question classification. For example, the authors have utilized the *n*-gram features in order to address each term located in the question separately.

In addition, Chang and Chung [7] proposed a combination of machine learning and keyword-based approach for classifying questions based on Bloom's cognitive level. Such an approach utilizes the keywords that

distinguish the level of the question, for instance, the keyword 'define' is related to remembering level.

Furthermore, Yusof and Hui [8] proposed a machine learning technique with a statistical feature called Category Frequency-Inverse Document Frequency (CF-IDF). The proposed method aims to utilize the frequency of each class label of Bloom in order to provide a probability mechanism for new questions.

Moreover, Haris and Omar [9] proposed a rule-based approach for classifying questions that are related to the computer science domain. Such rule-based approach is based on Natural Language Processing (NLP) techniques such as normalization (i.e., eliminate the stopwords) stemming (i.e., retrieving the root of each word), Part-Of-Speech (POS) Tagging (i.e., provide the syntactic class for each word).

However, this approach is not dynamic as the machine learning techniques and produced poorly results of classification. Yahya et al. [10] have addressed the use of machine learning techniques in classifying the questions into the same cognitive level of Bloom. The authors have collected questions and annotate them with their actual classes manually.

Consequentially, a comparison has been performed between three classifiers Naïve Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) in which, these classifiers have been trained and tested on the annotated data.

Results showed that SVM has superior performance compared to the other classifiers. Abduljabbar and Omar [11] have proposed a combination classification using voting technique among multiple classifiers including NB, SVM, and KNN. In fact, the proposed method has been created to classify programming questions. In this manner, multiple feature extraction techniques have been used including chi-square, mutual information, and odds ratio.

Reviewing the literature, one can notice that most of the related works have been concentrated on enhancing the classifiers' effectiveness whether by proposing a combination of two classifiers or by adding statistical features. Yet, the semantic aspect has not been addressed adequately.

3. Proposed Method

The methodology of this study consists of five main phases as depicted in Fig. 2. The first phase concentrates on the data used in the experiment, which indeed would contain annotated questions in accordance with Bloom's classes.

The second phase focuses on the preparation tasks that would be carried out in order to turn the data into an appropriate form. The third phase is associated with the combination method of semantic and syntactic approaches. The fifth phase aims to represent the data produced by the previous two phases.

The final phase is associated with the classification process where three classifiers are being used including SVM, J48 and NB, which aim to categorize the questions based on the *N*-gram representation.

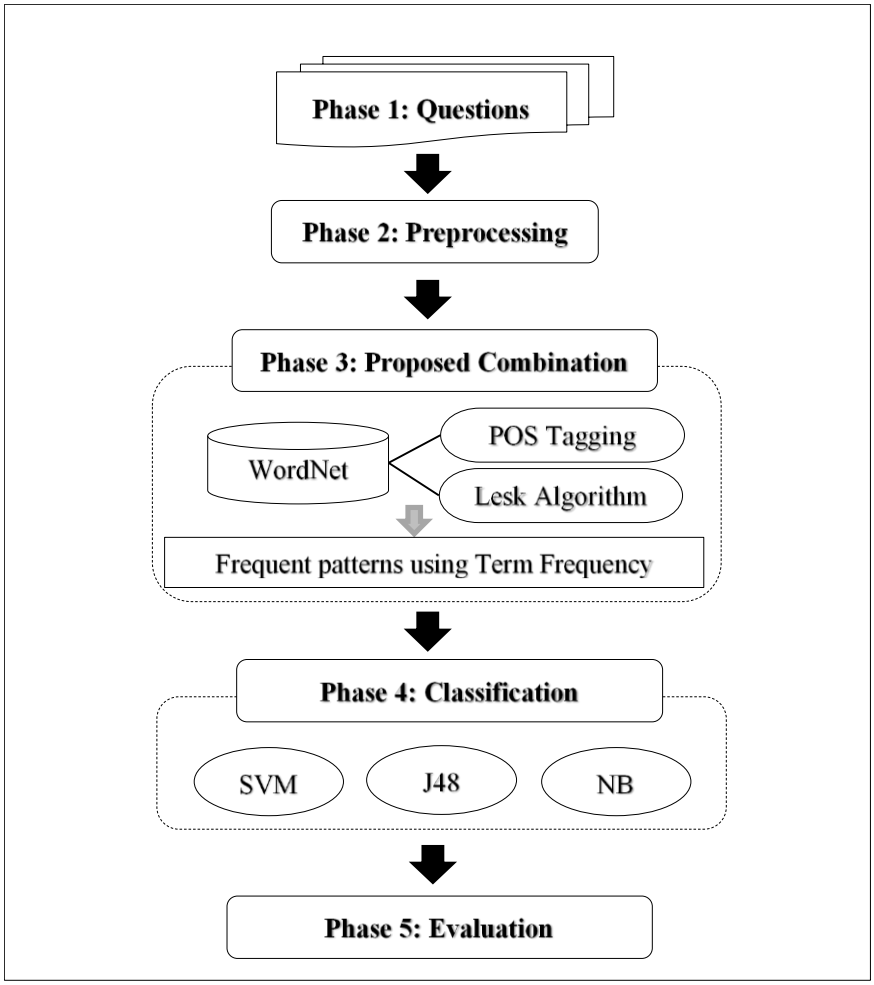

Fig. 2. Proposed method.

3.1. Questions

Regarding the scope of this study, which represented as handling domain-dependant and domain-independent questions, this study aims to use two datasets. Firstly, a dataset introduced by Yahya and Osman [14] was used. This dataset composed of domain-independent questions. Secondly, a dataset introduced by Haris and Omar [9], was used too in this study.

This dataset composed of domain-dependant questions related to Computer Programming. In fact, both datasets have been collected manually and have been labelled by experts. The reason behind selecting such datasets lies on the possibility of comparison in which, the proposed method can be assessed in comparison with the other studies. Table 1 depicts a sample from both datasets, while Table 2 shows the details of them.

Table 1. Sample of both datasets.

Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
Dataset 1 (Domain-independent) [14]					
Define compound interest	Compare historical events to contemporary situations	Apply laws of statistics to evaluate the reliability of a written test	Analyse safe and dangerous aspects of these features	Apply and integrate several different strategies to solve a mathematical problem (not according to one formula)	Appraise data in support of a hypothesis
Dataset 2 (Domain-dependent) [9]					
Define inheritance concept	Write a program that prompts the user to input the masses of the bodies and the distance between the bodies. The program then outputs the force between the bodies	Draw a flowchart to accept a number and output its factorial. 1: input 5 - > 5! = 5×4×3×2×1 -> output 120	Consider the following statement in C++: string firstStr = "What does a programmer do?"; string secondStr = "Programmer solves problems"	Consider the following statement in C++: string firstStr = "What does a programmer do?"; string secondStr = "Programmer solves problems"	Consider the following statement in C++: string firstStr = "What does a programmer do?"; string secondStr = "Programmer solves problems"

Table 2. Dataset details.

Dataset	Number of question
Dataset 1	600
Dataset 2	137

3.2. Preprocessing

In order to get rid of the noisy and unwanted data, which facilitates the process of classification, several tasks are being carried out including normalization, tokenization and stemming. Normalization aims to get rid of the stop words due to their insignificant impact on the context of the question. A list of English stop words has been used to remove such words. Tokenization aims to turn the text into a series of tokens. Finally, stemming aims to retrieve the root of words by eliminating the derivations. A light stemming of Porter [15] has been used in this study.

3.3. Proposed combination method

In fact, Pal et al. [16] addressed in different studies in which, utilizing the semantic aspect in terms of classifying questions. Therefore, this study has been motivated by such a study where the semantic and syntactic aspects have been employed in the task of classifying questions into Bloom taxonomy. In this phase, the proposed combination method is being carried out in which, multiple tasks are being conducted. These tasks are described as follows:

3.3.1. Part-of-speech tagging

In this task, a syntactic tool of POS tagging is being used in order to determine the syntactic tag of each word such as verb, noun or other. The key characteristic behind POS tagging lies on the demand of identifying the syntactic class of the word in order to clarify the meaning of a specific word. The most important tags are the verbs and nouns. Verbs are associated with the classes of the question (e.g.,

the verb define related to 'Knowledge' class). Whereas, nouns are associated with the domain of the question (e.g., the noun 'data' is related to computer science domain). Note that, the POS tagging that has been used in this study is the Stanford NLP [17].

3.3.2. WordNet

In this task, the question will be expanded in order to eliminate the ambiguity. This can be conducted by utilizing an external knowledge source (i.e., WordNet) in order to get more semantic correspondences for each word within the question. Basically, in WordNet dictionary, contains multiple forms for a single word whether verb, noun or other. Therefore, the use of POS tagging in the previous phase is crucial in order to determine the required information from WordNet. However, even for the same syntactic tag, there are multiple meanings and definitions for the single word. For example, the word 'protocol' is associated with different meanings in the WordNet. First meaning expressed as "form of a ceremony organized by diplomats", second meaning expressed as "code of correct conduct", and third meaning expressed as "standard rule for computer communications". In this vein, it is necessary to determine the relevant sense in order to correctly classify the question. For this purpose, this study adopts the Lesk algorithm to do such a task, which can be illustrated in the next sub-section.

3.3.3. Lesk algorithm

Lesk [18] introduced this algorithm as a tool that is able to eliminate the ambiguity within multiple senses. The main hypothesis lies behind the Lesk algorithm can be represented as "the similar words in the meaning contain similar contexts". This means that the words that have the same meaning occurred in similar contexts in which, the neighbouring words would frequently occur in both contexts. Let *A* and *B* are two words that shared the same meaning. Assume that the word A is being occurred in three contextsC_{a1}, C_{a2} and C_{a3}, similarly, *B* is being occurred in three contexts C_{b1}, C_{b2} and C_{b3}, the Lesk among *A* and *B* can be calculated as in Eq. (1).

$$Lesk\ (A,B) = Max\ C_{ai} \cap\ C_{bj} \tag{1}$$

where Max is the maximum intersection between two contexts. This means that the two contexts that have the maximum similarity or matches in terms of the words will be identified as similar contexts. Assume a question *q*, which can be expressed as "Write a Java program to show the overload concept". In this manner, each word will be processed under WordNet in order to retrieve semantic correspondences. Note that the stop words will be discarded. Hence, we will have six words where *A* = "write", *B* = "Java', *C* = "program", *D* = "show", *E* = "overload", and *F* = "concept". Table 3 depicts the retrieval senses using the example's words. As shown in Table 3, every word has been supplemented with multiple meanings. This can be an overload in which, irrelevant information could be obtained. Therefore, Lesk algorithm has been used in order to determine the required meaning.

It is obvious that there are common words shared by multiple meanings, for instance, sense number *A3* (i.e., Write) has mutual words with the sense number *B2* such as 'computer', 'code' and 'program'. Similarly, for senses *C4*, *D2* and *E3*, the words 'computer', 'science' and 'program' have also occurred commonly. Hence,

Lesk algorithm will utilize such lexical similarity in order to identify the accurate senses. Note that, the last word *F1* 'Concept' has only one sense thus, it would be retrieved even if does not contain mutual words. Table 4 depicts the similar senses extracted by Lesk algorithm.

The pseudo code of the Lesk algorithm can be illustrated as shown in Fig. 3.

Table 3. Expanding question's words using WordNet [19].

ID	Word	Senses brought from WordNet
***A*1**	Write	*S*: (v) **write**, compose, pen, indite (produce a literary work) "She composed a poem"; "He wrote four novels"
***A*2**	Write	*S*: (v) **write**, drop a line (communicate (with) in writing) "Write her soon, please!"
***A*3**	Write	*S*: (v) **write** (create code, write a computer program) "She writes code faster than anybody else"
***B*1**	Java	*S*: (n) **Java** (an island in Indonesia to the south of Borneo; one of the world's most densely populated regions)
***B*2**	Java	*S*: (n) **code**, computer code (computer science) the symbolic arrangement of data or instructions in a computer program or the set of such instructions)
***B*3**	Java	*S*: (n) coffee, **java** (a beverage consisting of an infusion of ground coffee beans) "he ordered a cup of coffee"
***C*1**	Program	*S*: (n) plan, **program**, programme (a series of steps to be carried out or goals to be accomplished) "they drew up a six-step plan"; "they discussed plans for a new bond issue"
***C*2**	Program	*S*: (n) broadcast, **program**, programme (a radio or television show) "did you see his program last night?"
***C*3**	Program	*S*: (n) **program**, programme (an announcement of the events that will occur as part of a theatrical or sporting event) "you can't tell the players without a program"
***C*4**	Program	*S*: (n) **program**, programme, computer program, computer programme (computer science) a sequence of instructions that a computer can interpret and execute) "the program required several hundred lines of code"
***D*1**	Show	*S*: (n) **show** (the act of publicly exhibiting or entertaining) "a remarkable show of skill"
***D*2**	Show	*S*: (n) display, **show** (something intended to communicate a particular impression) "made a display of strength"; " show the results of a computer program"
***D*3**	Show	*S*: (n) **show** a social event involving a public performance or entertainment) "they wanted to see some of the shows on Broadway"
***E*1**	Overload	*S*: (v) **overload** (become overloaded) "The aerator overloaded"
***E*2**	Overload	*S*: (v) clog, **overload** (fill to excess so that function is impaired) "Fear clogged her mind"; "The story was clogged with too many details"
***E*3**	Overload	*S*: (n) **overload** (an electrical load that exceeds the available electrical power) "the computer has been overloaded with too many programs installed"
***F*1**	Concept	*S*: (n) **concept**, conception, construct (an abstract or general idea inferred or derived from specific instances)

Table 4. Accurate contextual senses using Lesk algorithm.

ID	Word	Senses brought from WordNet
***A*3**	Write	*S*: (v) **write** (create code, write a computer program) "She writes code faster than anybody else"
***B*2**	Java	*S*: (n) **code**, computer code ((computer science) the symbolic arrangement of data or instructions in a computer program or the set of such instructions)
***C*4**	Program	*S*: (n) **program**, programme, computer program, computer programme (computer science) a sequence of instructions that a computer can interpret and execute) "the program required several hundred lines of code"
***D*2**	Show	*S*: (n) display, **show** (something intended to communicate a particular impression) "made a display of strength"; " show the results of a computer program"
***E*3**	Overload	*S*: (n) **overload** (an electrical load that exceeds the available electrical power) "the computer has been overloaded with too many programs installed"
***F*1**	Concept	*S*: (n) **concept**, conception, construct (an abstract or general idea inferred or derived from specific instances)

Input: words
Output: relevant senses

1. **For each** word *W1* → *W*m
2. Retrieve senses of *W* from an external knowledge
3. **For each** senses *S*1 → *Sn*
4. Compute the Max (*W1 (S*1 → Sn) ∪ *W*m (*S*1 → *Sn*)) where max is the highest number of matching words among the senses
5. **End for**
6. **End for**
7. **Return the max**

Fig. 3. Algorithm 1. Lesk.

3.3.4. Frequent patterns

After acquiring the relevant senses from the Lesk algorithm as shown in Table 4, the frequent patterns of verbs and nouns will be brought. These patterns consist of the verbs and nouns that have occurrence count more than one. In fact, obtaining such patterns relies on the Term Frequency, which can be computed as in Eq. (2).

$$TF(t) = \sum_{i=0}^{n} S_i(t) \quad (2)$$

where TF is the term frequency for a term *t*, and $S_i(t)$ is the number of time the term *t* has been occurred in the sense S_i. This will be applied until sense S_n.

The obtained patterns will be formed using *N*-gram representation specifically Unigram where every single term (whether frequent noun or frequent term) will be sorted in columns separately.

Several research efforts have recommended the use of *N*-gram method in terms of an appropriate representation for machine learning techniques [20, 21]. Note that, some authors have used such representation as One-Hot encoding [22]. In this manner, the *N*-gram representation can be depicted as in Table 5.

As shown in Table 5, the representation will be based on the term occurrence in which, the presence of a specific word (whether verb or noun) will be represented

by '1' and the absence of such word will be represented as '0' in accordance to the corresponding question.

The pseudo code of the proposed combination of syntactic and semantic method will be illustrated as in Fig. 4.

Table 5. Representation of data.

Semantic and Syntactic modification of the question	Sample of frequent patterns brought from Table 4 (Unigram Terms)						Class
	Computer	Program	Science	Write	Show	Display	
Question 1	1	0	1	1	1		Application
Question 2	0	1	1	0	1		Analysis
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-
Question *n*	1	1	1	0	0		Knowledge

Input: Question *q*
Output: frequent verbs and nouns in an N-gram representation

1. **For each** word *W1* → *W*m within the question
2. Normalize (remove the word if it is stop word)
3. Apply POS tagging *W*
4. Retrieve senses of *W* from an WordNet
5. **End for**
6. Compare senses of all words within the question
7. Retrieve the most relevant sense using Lesk algorithm
8. **For each** word *W'1* → *W'*n from the relevant senses
9. Apply Term Frequency TF
10. **For each** word *W'* that has a tag of Noun or Verb && TF*(W')* > *1*
11. Store the word *W'* as a pattern *P*
12. **End for**
13. **End for**
14. Represent the patterns using N-gram
15. **End**

Fig. 4. Algorithm 2: Combination of syic and semantic method.

3.4. Classification

To classify the questions based on their actual class of Bloom, three machine learning techniques have been used including SVM, J48 and NB. The reason behind selecting such classifiers lies in their popularity and substantial performances. The classifiers have been trained on 80% percent of the data where the class label was existed and tested on the remaining 20% portion of the data in which, the class label was required to be attained by the classifiers. Basically, all the questions in the two datasets are being mapped with a class label. Therefore, to test the classifiers, the class labels have been removed from 20% of the total questions. The reason behind selecting such portion of training and testing is to facilitate the comparison with the state of the art who used the same division [9-11]. The description of each classifier can be depicted in the following sub-sections.

3.4.1. Support vector machine (SVM)

Since every question is mapped with one of the six classes of Bloom. In this vein, SVM will classify the questions based on a linear mechanism where the data is being divided using a hyperplane into two parts; questions related to class *i*, and questions that irrelevant to class *i*. This process will be carried out for each class label until all the data is classified. In this manner, the evaluation will be based on the testing portion in which, the three classifiers.

The following pseudo code will illustrate the SVM classification algorithm as shown in Fig. 5.

Input Data *D* the classes represented by the frequent pattern *N*-gram

1. **Divide** the data into 80% training and 20% testing
   For each training data
2. Identify the hyperplane between the Knowledge and not Knowledge
3. Identify the hyperplane between the Comprehension and not Comprehension
4. Identify the hyperplane between the Application and not Application
5. Identify the hyperplane between the Analysis and not Analysis
6. Identify the hyperplane between the Synthesis and not Synthesis
7. Identify the hyperplane between the Evaluation and not Evaluation

End For
For each testing data
8. Classify the instances based on the hyperplanes

End For

Fig. 5. Algorithm 3: Support vector machine.

3.4.2. J48

This classifier is also known as Decision Tree, which aims to classify the instances based on a tree that simulates the features that contribute toward classifying the instance into a class label. This can be performed by depicting the features on the branches of the tree, which lead to a class label that represented in the leaves. The following pseudo code will illustrate the J48 classification algorithm as in Fig. 6.

Input Data *D* with the classes represented by the frequent pattern *N*-gram

1. Divide the data into 80% training and 20% testing
   For each training data
2. Split the instance into a tree where the branches mimic the patterns *Pi* (i.e., features) and leaves represent the class label
3. Compute the target value for each pattern that corresponds to class Knowledge
4. Compute the target value for each pattern that corresponds to class Comprehension
5. Compute the target value for each pattern that corresponds to class Application
6. Compute the target value for each pattern that corresponds to class Analysis
7. Compute the target value for each pattern that corresponds to class Synthesis
8. Compute the target value for each pattern that corresponds to class Evaluation

End For
For each testing data
9. Classify the instances based on the target values calculated in the training

End For

Fig. 6. Algorithm 4: J48.

3.4.3. Naive bayes (NB)

This classifier uses the probabilities of each feature associated with every data instance in order to predict the class label. This can be conducted by analysing the training data and compute probabilities for each feature in accordance with every class label. In other words, the number of times feature *f* has led to a class label *C*. The following pseudo code will illustrate the NB classification algorithm as shown in Fig. 7.

1. **Divide** the data into 80% training and 20% testing
   For each training data
2. Calculate the probability for each pattern *pi* in accordance to the class Knowledge
3. Calculate the probability for each pattern *pi* in accordance to the class Comprehension
4. Calculate the probability for each pattern *pi* in accordance to the class Application
5. Calculate the probability for each pattern *pi* in accordance to the class Analysis
6. Calculate the probability for each pattern *pi* in accordance to the class Synthesis
7. Calculate the probability for each pattern *pi* in accordance to the class Evaluation

End For
For each testing data
8. Classify the instances based on the probabilities calculated in the training

End For

Fig. 7. Algorithm 5: Naive Bayes input data *D* with classes represented by the frequent pattern *N*-gram

3.5. Evaluation

Basically, the evaluation will be conducted upon the testing portion that was being classified by the three classifiers. The evaluation has been performed using common information retrieval metrics precision, recall, and f-measure. These evaluation metrics have been widely used in terms of evaluating machine learning techniques. Firstly, precision can be calculated as in Eq. (3).

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

where TP is the number of correctly classified questions, and FP is the number of incorrectly classified questions. Secondly, recall can be calculated as in Eq. (4).

$$ecall = \frac{TP}{TP+FN} \tag{4}$$

where FN is the number of questions that that was not being classified. Finally, f-measure can be calculated as in Eq. (5).

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

Table 6 shows the confusion matrix.

Table 6. Confusion matrix.

		Predicted Class	
Actual class		**Positive**	**Negative**
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

4. Experimental Results

In order to address the effectiveness of the proposed combination method, the three classifiers will be applied with the combination and without the combination. In the case of the application without the combination, the classifiers are being applied on the N-gram of the question without applying the semantic and syntactic approaches. While in the case of the application with the proposed combination, the classifiers are being applied on the *N*-gram that generated by the combination of semantic and syntactic approaches. In addition, both datasets have been involved in the examination in order to determine the effectiveness of the proposed combination method to handle both domain-independent and domain-dependent datasets. Table 7 shows such results.

First of all, considering the results in Table 7, one could notice that all the classifiers have been improved when using the proposed combination. This can be represented where NB with the combination has achieved 69% compared to 51% for the NB without the combination based on the domain-independent dataset, as well as, for the domain-independent dataset where results were 78% compared to 55%. The same applied for SVM where results of the combination for the first dataset was 80% compared to 58% without the combination, as well as, for the second dataset where results were 82% against 66%. Finally, for the J48 results of the combination was outperforming too as 76% against 40% for the first dataset, and 75% against 65%.

On the other hand, SVM has outperformed the other classifiers for both datasets, and for both with the proposed combination and without the combination. This is because our study has utilized a unigram representation thus; the number of features would be large. According to Altawaier and Tiun [23], SVM has superior performance compared to other classifiers when handling a large number of features.

Table 7. Experimental results.

Domain-independent dataset		
Classifier	**F-measure (with combination)**	**F-measure (without combination)**
NB	0.69	0.51
SVM	0.80	0.58
J48	0.76	0.40
Domain-dependent dataset		
Classifier	**F-measure (with combination)**	**F-measure (with combination)**
NB	0.78	0.55
SVM	0.82	0.66
J48	0.75	0.65

5. Discussion

From the previous section, apparently, SVM with the proposed combination has obtained the highest f-measure. Therefore, this section aims to compare such results against the state of the art. Firstly, Yahya et al. [10] proposed an SVM for classifying questions based on Bloom's taxonomy and achieved an f-measure of 72%. Secondly, Abduljabbar and Omar [11] obtained an *f*-measure of 75%. On the other hand, Yusof and Hui [8] have gained an f-measure of 62% using an artificial neural network. In addition, Haris and Omar [9] have achieved an *f*-measure of 77% using a rule-based.

It is obvious that our proposed method has outperformed the other related work especially for Haris and Omar [9] and Abduljabbar and Omar [11] whose approaches have been carried out on the same domain-dependent dataset that was used in this study. As well as, for Yahya et al. [10] who used the same domain-independent dataset that has been used in this study.

In particular, Abduljabbar and Omar [11] and Yahya et al. [10] have used the same classifier that has been used in this study, which is SVM. However, the SVM in our study has been combined with the proposed combination method. This is the reason for the superiority of our SVM's results. This can demonstrate the effectiveness of the proposed combination method in terms of attaining the contextual information that would improve the classification results. Table 8 summarizes the comparison with the state of the art.

Table 8. Comparison against the state of the art.

Author	Method	Dataset	F-measure
Haris and Omar [9]	Rule-based	Dependent (dataset 2)	77%
Yahya et al. [10]	SVM	Independent (dataset 1)	72%
Abduljabbar and Omar [11]	SVM	Dependent (dataset 2)	75%
Proposed Method	SVM with the proposed combination	Independent (dataset 1)	80%
		Dependent (dataset 2)	82%

6. Conclusions

This paper articulates a combination method for identifying Bloom's categories for questions. The proposed combination method has utilized a semantic and syntactic approach. The semantic approach can be represented by using an external knowledge of the WordNet dictionary with the Lesk algorithm. While the syntactic approach can be represented by using the POS tagging in order to retrieve frequent patterns of terms. Finally, three classifiers have been applied in order to classify the questions. In order to examine the effectiveness of the proposed method, the three classifiers have been carried out twice; first with the proposed combination method, and second without the proposed combination. Results revealed that the three classifiers with the proposed combination method shown better performance rather than without using the combination. For future directions, addressing large-scale dataset of questions would be an interesting effort. In addition, examining recent technologies such as word embedding would yield promising results.

Nomenclatures

C_i	Context
q_i	Question
S_i	Sense
W_i	Word

Abbreviations

J48	Decision Tree
NLP	Natural Language Processing
NB	Naive Bayes
POS	Part-of-Speech
SVM	Support Vector Machine
TF	Term Frequency

References

1. Adams, N.E. (2015). Bloom's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association*, 103(3), 152-153.
2. Kaya, M.; and Karamustafaoglu, O. (2015). Analysis of TSKT questions on science teaching in 2013 PPSE according to reconstructing of Bloom taxonomy. *Eurasian Journal of Physics and Chemistry Education*, 7(1), 29-36.
3. Tarman, B.J.; and Kuran, B.T. (2015). Examination of the cognitive level of questions in social studies textbooks and the views of teachers based on Bloom taxonomy. *Educational Sciences: Theory and Practice*, 15(1), 213-222.
4. Omar, N.; Haris, S.S.; Hassan, R.; Arshad, H.; Rahmat, M.; Zainal, N.F.A.; and Zulkifli, R. (2012). Automated analysis of exam questions according to Bloom's taxonomy. *Procedia-Social and Behavioral Sciences*, 59, 297-303.
5. Loni, B. (2011). A survey of state-of-the-art methods on question classification. TU Delft Repository, TU Delft University of Technology, 54(3), 1-40.
6. Zhang, D.; and Lee, W.S. (2003). Question classification using support vector machines. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. Toronto, Canada, 26-32.
7. Chang, W.-C.; and Chung, M.-S. (2009). Automatic applying Bloom's taxonomy to classify and analysis the cognition level of English question items. *Joint Conferences on Pervasive Computing (JCPC)*. Tamsui, Taipei, Taiwan, 727-734.
8. Yusof, N.; and Hui, C.J.(2010). Determination of Bloom's cognitive level of question items using artificial neural network. *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA)*. Cairo, Egypt, 866-870.
9. Haris, S.S.; and Omar, N. (2012). A rule-based approach in Bloom's Taxonomy question classification through natural language processing. *Proceedings of the 7th International Conference on Computing and Convergence Technology (ICCCT)*. Seoul, South Korea, 410-414.
10. Yahya, A.A.; Osman, A.; Taleb, A.; and Alattab, A.A. (2013). Analyzing the cognitive level of classroom questions using machine learning techniques. *Procedia-Social and Behavioral Sciences*, 97, 587-595.
11. Abduljabbar, D.A.; and Omar, N. (2015). Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination. *Journal of Theoretical and Applied Information Technology*, 78(3), 447-455.
12. Alshaikhdeeb, B.; and Ahmad, K. (2015). Integrating correlation clustering and agglomerative hierarchical clustering for holistic schema matching. *Journal of Computer Science*, 11(3), 484-489.
13. Forehand, M. (2010). Bloom's taxonomy. *Emerging Perspectives on Learning, Teaching, and Technology*, 41-47.
14. Yahya, A.A.; and Osman, A. (2011). Automatic classification of questions into Bloom's cognitive levels using support vector machines. *Proceedings of the International Arab Conference on Information Technology*. Riyadh, Saudi Arabia, 335-342.

15. Porter, M.F. (2001). Snowball: A language for stemming algorithms.
16. Pal, D.; Mitra, M.; and Datta, K. (2014). Improving query expansion using WordNet. *Journal of the Association for Information Science and Technology*, 65(12), 2469-2478.
17. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System demonstrations*. Baltimore, Maryland, United States of America, 55-60.
18. Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *Proceedings of the 5th Annual International Conference on Systems Documentation*. Toronto, Ontario, Canada, 24-26.
19. Miller, G.A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.
20. Alshaikhdeeb, B.; and Ahmad, K. (2016). Biomedical named entity recognition: A review. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 889-895.
21. Alshaikhdeeb, B.; and Ahmad, K. (2017). Feature selection for chemical compound extraction using wrapper approach with Naive Bayes classifier. *Proceedings of the 6th International Conference on Electrical Engineering and Informatics (ICEEI)*. Langkawi, Malaysia, 1-6.
22. Vinyals, O.; Jia, Y.; Deng, L.; and Darrell, T.(2012). Learning with recursive perceptual representations. *Advances in Neural Information Processing Systems*, 4, 2825-2833.
23. Altawaier, M.M.; and Tiun, S. (2016). Comparison of machine learning approaches on arabic twitter sentiment analysis. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 1067-1073.