TOWARDS AN ONTOLOGY-BASED FULLY INTEGRATED SYSTEM FOR STUDENT E- ASSESSMENT

SALEH ALMUAYQIL[1], SAMEH ABD EL-GHANY[1,3], ABDULAZIZ SHEHAB[2,3]

[1]Information Systems Department, College of Computers and Information Sciences, Jouf University, Sakaka, KSA

[2]Department of Computer Science, College of Science and Arts, Jouf University, KSA.

[3]Information Systems Department, Faculty of Computers and Information Sciences, Mansoura University, Egypt

E-mail: [1] snmuayqil@ju.edu.sa, [2]saabdelwahab@ju.edu.sa, [3]aishehab@ju.edu.sa,

ABSTRACT

Recently, many higher educational institutions have paid significant attention to distance learning. Although several learning management systems (LMSs), have been widely used in many countries, these systems still require more enhancements, especially in student assessment tasks. Consequently, this paper presents an innovative solution intended to meet the needs of the students easily, while simultaneously eliminating the burden faced by instructors. The proposed approach entails an integrated framework for fully automating the assessment process, starting from the generation of questions from a given corpus. Thereafter, the appropriate answers for each generated question are extracted from different educational resources. The experiments demonstrate the ability of the proposed framework to generate candidate questions, while measuring its difficulty score based on a hybrid technique of semantic and contextual data analyses.

Keywords: *E-assessment, Question generation, Semantic Similarity, Word2Vec.*

1. INTRODUCTION

Currently, student assessment is one of the most challenging tasks in the educational process since it is a time-consuming and tedious effort. From an instructor's perspective, the process of student assessment is time-consuming with respect to generating questions for students' duties and evaluating their achievements .The aim of the assessment is to appraise the students' knowledge, understanding, abilities, or skills using an array of evaluation activities, varying from semester exams to practice exams, quizzes, weekly assignments, and so on. The traditional method of assessment embraces a wide-ranging list of different stages is performed by the instructor. It start from selecting a diverse number of questions that are different in cognitive and difficulty levels, as well as type, after which the answers for each question are extracted, and finally, the answers of the students are graded. With the advances in technological and communication development, most educational institutions are equipped with LMSs to improve the quality of the learning process by availing learners the flexibility to study without the constraints of time and space. Although various LMSs , such as Blackboard, MOOCs, Moodle and eFront are exist in different universities, these systems cannot automate the creation of questions, do not automatically extract the answer, nor do they measure the difficulty of generated questions [1]. However, their role is limited to helping the instructor in storing the question repository, providing online assessment activity, as well as helping to correct types of True/False and multiple choice questions (MCQ) and grading short answers. Thus, there are still deficiencies that require the instructor's intervention in the assessment process. Hence, e-learning systems require more attention and continuous improvement to enhance the assessment process. Therefore, it would be useful to build a system for educational institutions, which would be able to replace the role of the instructor in the assessment process by automating such tasks from start to end.

This paper presents a framework for the assessment process, starting from the generation of questions with diverse cognitive and difficulty levels, as well as diverse question types, extract answer for each question, and finally, the students' answers are automatically graded. The role of the instructor in the proposed framework herein is merely to provide the system with the educational resources used to teach the course.

- Unlike others systems [2-5] that often generate MCQs with the lowest level of cognitive skills, ignoring other questions types, our proposed framework is designed to generate different types of questions with different difficulty levels in order to measure both knowledge and cognitive skills.
- Embedding context similarity fused with semantic similarity which is significantly effect on the classification of question difficulty.
- Dynamic change of question difficulty adaptively through data analytic and crawling course materials and students answers.

The remainder of this paper is organized as follows. In Section 2, we present a literature review respect to the stages of assessment process. A detailed description of the proposed framework is presented in section 3. The experimental results are presented and discussed in section 4, while section 5 provides a conclusion for this research.

2. LITERATURE REVIEW

Manually generating meaningful and relevant questions for student assessment is a time-consuming and challenging task [6]. For example, while evaluating the students on reading comprehension, it is tedious for an instructor to manually generate questions, find answers to those questions, and then evaluate the answers. Automatic transforming of these stages has a positive impact on the educational process, and therefore, many researchers nowadays turn toward automating parts of these stages. The next sub-section of this paper discusses the current work associated with each stage, especially; (2.1) automatic question generation (AQG) stage, (2.2) automatic question answering stage, and (2.3) automatic scoring stage, and indicates the weaknesses and strengths in each case.

2.1 Automatic Question Generation (AQG)

AQG aids in partially solving the challenges that the instructors face in student assessment processes. AQG is concerned with the construction of algorithms for producing questions from learning resources, which can vary from structured (e.g., databases) to unstructured (e.g., textbooks). The generated questions can be simple factual WH-questions (i.e., where the answers are short facts that are explicitly mentioned in the input) or gap-fill questions. Earlier works on question generation can be mainly classified into rule-based systems, ontology-based systems, and neural-based systems

2.1.1 Rule-based systems

When generating questions using rules/ templates, the surface structure of the questions is determined using fixed text and placeholders that are replaced with values to create questions [7]. In the rule-based system, knowledge bases are used to generate the questions, and this requires a deep understanding of the field wherein the questions are created. For example, if the entered text contains a place, then the question is (where) and if it contains time, the question is (when), and so on, while the rules may extend to utilizing syntactic and/or semantic information for generating questions. For instance, Das et al. [8] used different rules of generation and ambiguity to formulate questions from simple and complex sentences, which first begins by identifying sentences and verb phrases, and then, applying a different rule of generation and ambiguity of AQG to those sentences to generate different types of questions, for example, why, what, or how. Similar work was presented by Khullar et al. [9], where a rule-based system was used to generate multiple questions from complex English sentences, relying on the analysis of dependency information from the spaCy parser and exploiting the dependency relationship between relative pronouns and relative adverbs. Agarwal et al. [10], presented a system to select the most informative sentences from a biology textbook to generate gap-fill questions relying on the syntactic and lexical features. Odilinye et al. [11], presented a system that combined the semantic and template–based approaches for question generation. Danon and Last [12], paraphrased sentences by replacing the source verbs, relying on the relationship between deep linguistic analysis (part of speech, recognizing the named entity) and knowledge resources (WordNet, Word2Vec model trained in a field group), and then, applying a set of rules that was propounded by Heilman [13] to create specific questions (usually “what” type). Previous rule- and template-based approaches mainly depend on handcrafted rules and templates, and this results in a set of shortcomings. First, building the rules and templates is laborious and time-consuming [14]. Second, the efficiency of the system depends mainly on the appropriateness of the constructed set of rules and templates. Third, these rules, as well as the templates, are related to a specific domain, and in the event that it changes, the efficiency decreases, and it is difficult to adapt them to other domains. Finally, the questions generated lack diversity due to the limited rules and constructed predefined question templates.

2.1.2 Ontology-based systems

Ontologies have been widely used for automatically generating MCQs. In this regard Leo et al. and Rocha et al. [2, 15] proposed an automatic generation of quizzes containing MCQs, answers, and distractors from domain ontologies. A similar work developed by Bongir et al. [5] aimed at generating complex MCQs from educational ontology, while Bongir et al. [16] used the semantic web technology—DBpedia—to represent the structured information extracted from Wikipedia to generate questions. Diatta et al. [17] proposed a bilingual (English and French) ontology-based automatic question generation system for formulating True/False and MCQ questions with single or multiple answers. With regard to knowledge sources, the most commonly used source for question generation is text. Faizan et al. [18] presented an approach to use semantic annotation to generate a variety of MCQs from slide content. Similarly, Faizan et al. [19] presented the automatic generation of MCQs from slide content using linked data.

2.1.3 Neural based systems

Recent deep learning-based question generation methods have proven to excel and generalize better than rule and ontology based systems. This technique is mainly based on the sequence-to-sequence models that generate questions from a specific sentence or paragraph by providing them with the context and the answer. For instance, Kumar et al. [20] proposed a question generation framework using reinforcement learning. The framework compromises two components, namely, a generator and an evaluator. The former consists of two mechanisms, i.e., copy and coverage mechanisms. The aim of the copy mechanism is to address the rare words problem using the sequence-to-sequence model, while the coverage mechanism addresses the problem of word repetition. Then, the evaluator provides rewards to fine-tune the generator. Liu et al. [21] designed a sequence-to-sequence-based model to identify whether a question word should be copied from the input passage or generated. The model consists of three components: the clue word predictor that is able to predict the distribution of clue word by utilizing the syntactic dependency tree representation of a passage, while the second component is the passage encoder that applies the bidirectional gated recurrent unit to incorporate both the predicted clue word distribution and a variety of other feature embedding of the input words, such as lexical features and answer position indicators. The third component is the decoder, which is responsible for deciding whether to generate or copy words from the passage by applying the gated recurrent unit. Wang et al. [4] present a system to generate questions from a structured knowledge base in Chinese based on the neural generation approach using long short-term memory (LSTM). A problem raised in neural question generation is that many words in the passage are repeated in the question, leading to unintended questions. To solve this problem, Kim et al. [22] used a recurrent neural network (RNN) encoder-decoder architecture to treat the passage and target answer separately by replacing the latter in the original passage with a special token. Wang at al. [23] introduced QG-Net, a recurrent neural network-based model specifically designed for automatically generating quiz questions from educational content, such as textbooks. While popular RNN-based models perform well for short sentences, they are perform poorly with longer text. Song et al. [24] proposed a model that matches the answer with the passage before generating the question, by encoding both the passage and answer using two separate bi-directional LSTMs. Harrison et al. [14] utilized a neural network architecture that uses two source sequence encoders. The first encoder was at the token level, and the second was at the sentence level, while considering the incorporation of linguistic features and an additional sentence embedding to capture the meaning at both sentence and word levels. Liu et al. [25] combined the template-based method with seq2seq learning to overcome problems inherent in both approaches, as well as generate highly fluent and diverse questions. Inversely

Notably, all of the aforementioned work deal with one sentence at a time to generate the question, which leads to the simplicity of the questions. However; Tuan et al. [26] proposed generating questions from several related sentences in one context, while Zhao et al. [27] proposed generating questions on a paragraph level rather than a sentence level using a sequence-to-sequence network.

2.2 Question Answering Systems

Question-answering systems aim to use both information retrieval (IR) and Natural Language Processing (NLP) to automatically answer questions that people ask in their natural languages. Particularly, high-quality and reliable question-answering systems have been greatly beneficial in education field. Choi et al. [28] presented fast model for selecting relevant sentences and used reinforcement learning for answering question from those sentences over long documents. Xiong et al. [29] proposed a question answering system based on a dynamic co-attention neural network architecture, which consists of a co-attention encoder to learn co-dependent representations of the question, and a dynamic decoder, which iteratively estimates the answer span. Wang et al. [30] utilized a bidirectional LSTM model and an RNN-based decoder model to adopt the attention-based sequence-to-sequence architecture that is able to dynamically switch between copying words from the document and generating words from a vocabulary. Dhingra et al. [31] presented a semi-supervised question answering system, which requires feeding a set of base documents and only a small set of question-answer pairs over a subset of these documents. Tatu et al. [32] proposed a semantic question answering system that stores the rich semantic structure identified in unstructured data sources into scalable RDF .

2.3 Automatic Scoring

The manual grading process has many problems, such as being time-consuming, costly, and resource intensive, as well as requiring great effort and placing huge pressure on the instructors. The educational community urgently requires auto-grading systems to address the significant problems associated with manual grading. Shehab et al. [33] proposed an Arabic automated system for essay grading using different text similarity algorithms. This system was based on a new dataset that was prepared in a general sociology course. Corpus-based and string-based algorithms using different preprocessing techniques, such as stem and stop-stem, were utilized. Yang et al. [34] proposed an attention-based neural matching model for ranking short answer text. Liu et al. [35] proposed a two-stage learning framework, Initially, three kinds of scores, semantic score, coherence score, and prompt-relevant score, were utilized to consider deep semantic information and the adversarial samples. Then, the handcrafted features and these scores were fed into a tree model for further training. Yamamoto et al .[36] proposed an automated essay scoring system architecture, which evaluates items based on rubric that were classified into human and automated scoring. Taghipour et al. [37] presented Automatic essay scoring system that utilized RNNs to automatically learns features and the relation between an essay and its assigned score.

3. SYSTEM FRAMEWORK

The proposed framework is directly serve decision makers and e-learning system designers as it could enhance the jobs LMSs in higher education. It comprises five main layers framework, as shown in Figure 1: (3.1) Resource Pre-processing layer (RPL), (3.2) Question Generation Layer (QGL), (3.3) Exam Generation Layer (EGL), (3.4) Automatic Scoring Layer (ASL), and (3.5) Result Analytic Layer (RAL). In the following sub-sections, a detailed explanation of theses layers is presented.

3.1 Resource Pre-processing Layer (RPL)

Determining the specific part of the knowledge, where the questions will be generated, is an important tasks that is inseparable from student assessment. The proposed system presents a new automated approach for identifying the candidate part of knowledge from the various educational sources, around which the question will be created, by representing these sources in an ontological form based on semantic and context analysis. Ontology is one of the main emerging methods for automatically generating questions due to its deductive abilities to represent concepts in unconventional ways. It generate questions about the characteristics of different concepts covering different areas of unstructured types of data, such as textbook or even different educational resources. However, current approaches for ontology-based question generation mainly relies on a manually –human feeded specific domains. Therefore, these approaches lacks the flexibility and adoptability to consider other domains. Hence, the proposed ontology is domain independent that could transforms unstructured learning resources (textbook, slides, etc.), into ontology form that cover different domains moreover, it is extended to focus on single concepts, as well as multi-word concepts by considering taxonomic and non-taxonomic relationships among these concepts, which limits the drawbacks of the systems based on pure handcrafted templates or rules. Based on linguistics approaches and text mining algorithms, the questions will be generated in many forms automatically. The RPL layer comprises a set of main modules, namely syntactic analysis module, information extraction module, and ontology construction module. The next sub-sections present a detailed description of these modules.

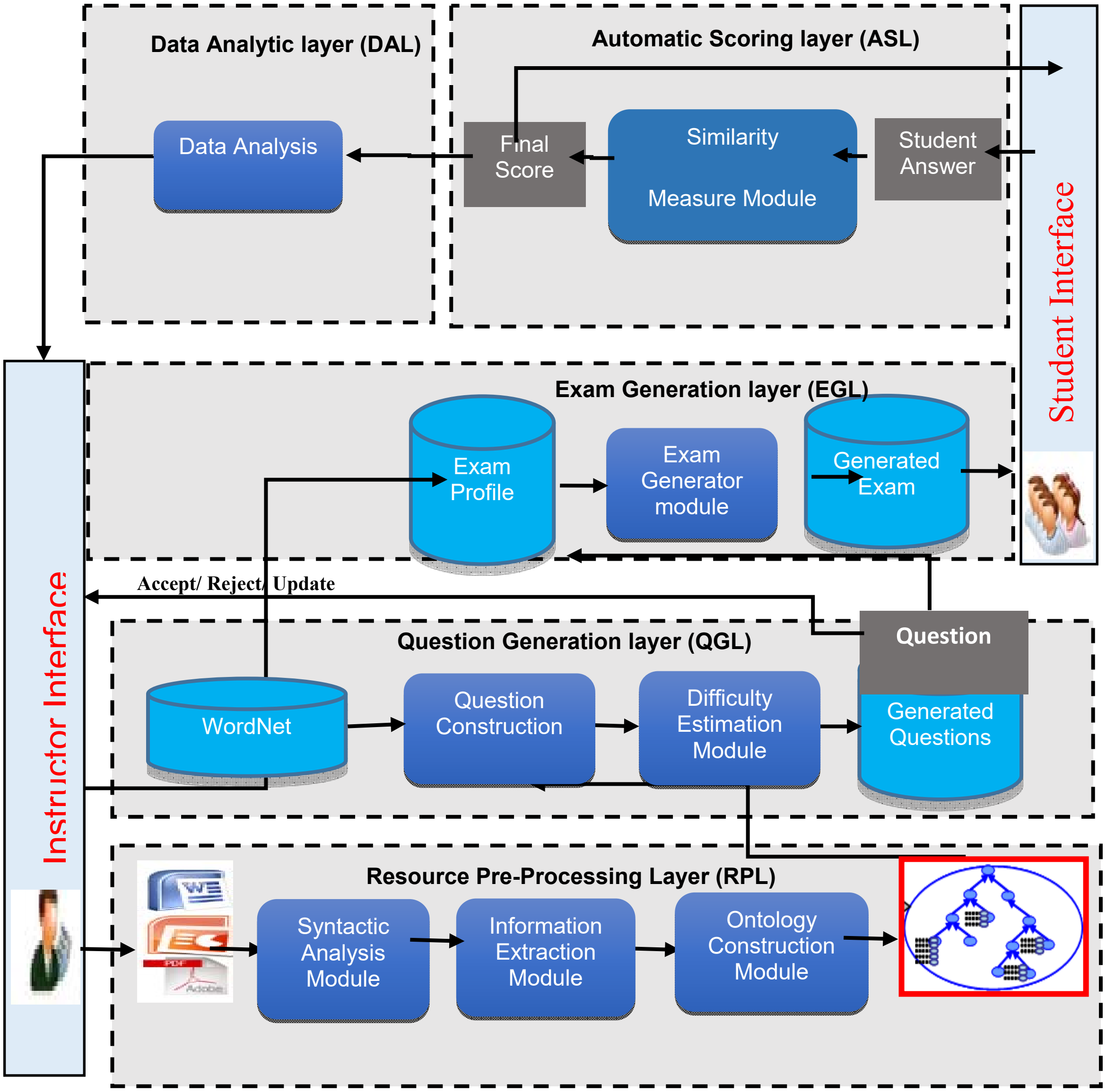
Data Analytic layer (DAL)
Data Analysis
Automatic Scoring layer (ASL)
Final Score
Similarity Measure Module
Student Answer
Student Interface
Exam Generation layer (EGL)
Exam Profile
Exam Generator module
Generated Exam
Accept/ Reject/ Update
Question
Question Generation layer (QGL)
WordNet
Question Construction
Difficulty Estimation Module
Generated Questions
Instructor Interface
Resource Pre-Processing Layer (RPL)
Syntactic Analysis Module
Information Extraction Module
Ontology Construction Module

Figure 1. Block Diagram of the Proposed System

3.1.1 Syntactic Analysis Module

Syntactic analysis module is responsible for dividing the document into a set of tokens (words or sentences), study the linguistic information and structural relationship for each one. It involves the following processes:

- **Tokenization**: It the first obligatory process in syntactic analysis module, as it divides the text into individual tokens, i.e., words, punctuation marks, etc.
- **Part of speech (POS) analysis:** It assigns word class, according to the morphology of each token. According to this analysis, a class is returned for each word, which contains type of word (noun, verb, etc.), gender, grammatical case, tense, etc.
- **Named entity recognition (NER):** It takes the identified proper nouns from the POS process and recognizes their type, according to a specific set of classes such as location, person, time, organization, etc.

- **Dependency parse tree analysis:** It describes the syntactic structure of each sentence.

- **Co-reference resolution analysis:** It identifies which pronouns depends on another noun in previous sentences, then replaces it with its mentioned proper noun.

3.1.2 Information Extraction Module

It extracts important information that will be used later in ontology construction. The extracted information are concepts, properties and relations.

- **Concept extraction:** By analyzing POS, the concepts can be identified by extracting all proper nouns in the sentences. The concept may be a single word or multiple words.

- **Relations extraction:** As the learning resources are in a form of unstructured and diversified data, the relations discovery is more complicated. In the proposed framework, the relations among concepts are classified into two types: The first types is based on the analysis of taxonomic relations. To identify taxonomic relationships, such as hyponym, etc., the external general WorldNet thesaurus will be used to assist in hierarchy construction, while, non-taxonomic relations is verbs-based extraction represented by noun-verbs dependency.

- **Property extraction:** The property of the concepts are those part of the token that describe a specific instance of concept in the sentences. Since the concept is represented by nouns that are extracted using POS, the noun phrases generally are followed by adjectives, and those adjectives are the properties for this concept. Property value may be represented by data values of strings, numbers, or Boolean values, and extracted through the dependency parse tree process.

3.1.3 Ontology Construction Module

It is responsible for building the hierarchal tree for the concepts based on the extracted relationship. The generated discovered relations as mentioned before is classified to two flavors, i.e., taxonomic and non-taxonomic relations. To build Hierarchy for taxonomic relationship, WordNet is looked for two important concepts; hyponyms and hypernyms relations. Hyponyms are subordinate word senses; they are a more specific form of the word sense of the super ordinate word sense of which they are a hyponym. Conversely, calling a word sense a hypernym of another word sense indicates that the first word sense is super ordinate to the other; the first word sense is above the latter in the hierarchy. Many words are both hypernyms of some words and hyponyms of others, of course, so the terms are used depending on which word sense's relations one is currently examining. For Non- Taxonomy Relationship construction, POS is most important in this process. The POS of the sentence is analyzed to represent the sentences in a tree structure.

3.2 Question Generation Layer

In general, the question generation is considered a challenging and time consuming tasks for instructor. However, the proposed framework enhance this process through only feeding the system with the subject materials (Textbook, ppt slides, pdf, etc.). Thereafter, the system has the responsibility to generate question bank by mapping the subject materials to target ontology. The generated question will be passed to the instructor to accept, update, or even reject. Question generation layer comprises two modules, Question construction module and Difficulty level estimation module.

3.2.1 Question construction module

Unlike others systems that often generate MCQs with the lowest level of cognitive skills ignoring other questions types, our proposed framework is designed to generate different types of questions with different difficulty levels in order to measure both knowledge and cognitive skills. The types of questions that can be generated from the constructed ontology are: define, explain, differentiate, and give example for, list components of, fill-in-the-blank, multiple choice, match, and True / False questions.

3.2.2 Difficulty level estimation module

Based on both context and semantic measures, the difficulty level is estimated, as shown in Figure 2. The first measure uses association rule mining to obtain context similarity between words, Association rule mining has gained attention for context similarity. It is intended to capture dependency among terms in the documents. The second measure uses WordNet and word2vec to obtain semantic similarity between words. By integrating such two measures, a highly similar alternative answers will be generated.

Table 1: Description Of Different Relations For Concepts

Relation type	Description
Is_definition_of (X)	It relates a concept and its definition.
Is_example_of (X)	It relates a concept with number (N*)* of instances.
Is_type_of	It indicates the generic inheritance relation in the ontology that forms a hierarchy.
Has_component	It is based on enumerations, it can be extracted from the textbook by following the many expressions, such as “has components,” “includes,” “is composed of.” etc.
Is_ equivalent _to (X)	It relates concept with its word synonyms
Has_Cause (X)	It is used to extract the causes of an action. It can be extracted by expressions, such as “cause”, “effect”, “impact”, etc.
Is_condition_of (X)	It indicates the conditions for performing an action. It can be extracted by expressions, such “constraint”, “condition,” etc.
Is_reason_of (X)	It is used to describe a reason for the existence of (X)
Is_function_of (X)	It links the concept with the function it performs.
Has_characteristic (X)	It indicates the property of concept (X)
Is_value _of (X)	It is used to indicate the data value of property (X)

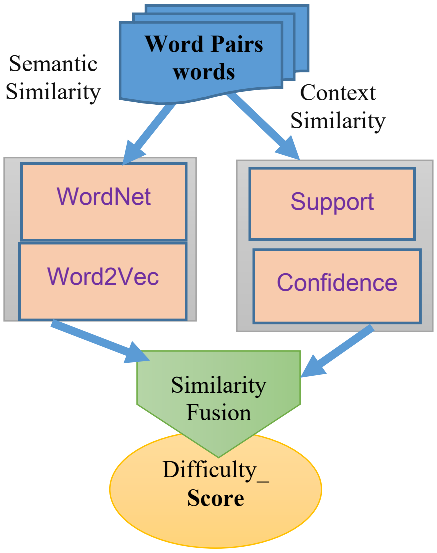

Figure 2. Question Difficulty Level Estimation Module

.Hence the question can be multi-difficulty level, The questions are classified into five categories, very easy, easy, average, hard, and very hard, according to a specific difficulty score, as shown in Table 2.

Table 2: Distribution of difficulty score according to Different question classes

Difficulty_Score	Class
Score ≤ 20	Very easy
20 < score ≤ 40	Easy
40 < score ≤ 60	Average
60 < score ≤ 80	Hard
80 < score	Very hard

3.2.2.1. Context Measure Calculation

The most influential algorithm for association rule mining is Apriori [38], which maps documents into a set of transaction based on terms co-occurrences. Then the support and confidence for the association rules containing these words are obtained. The support of word pairs is calculated as in equation (1) where Ai and A_j is the probability of frequently appearance A_i and A_j in the whole document and *n* indicates the total number of word pairs.

The higher the support, the more frequently the word set occurs.

$$\text{Supp}\ (A_i, A_j) = (A_i \cup A_j\ ) / n \qquad (1)$$

The confidence of a rule (association rule), $A_i \rightarrow A_j$, can be defined as the conditional probability of those word- pairs containing A_j that also contain A_i. Higher the confidence indicates highly related context measure.

Conf (A_i, A_j) = Supp (A_i , A_J) / Supp (A_i) (2)

3.2.2.2 Semantic Measure Calculation

The semantic measure is calculated using two a hybrid of two measures. For the first, WordNet is utilized, which is a lexical ontology wherein the words are connected with each other through linguistic relationships. To obtain the WordNet similarity measure between two word senses, the Wu-Palmer [39] method is used as in Eq. 3. It is based on the depth of the two senses in the taxonomy and the depth of their LCS, which refers to the least common sub-sumer (most specific ancestor node).

W_{sim} (A_i, A_j) = 2*depth (lcs) / depth (A_i) + depth (A_j) (3)

For the second similarity measure, Word2Vec model is utilized. Unlike WordNet; Word2Vec model does not capture linguistic relationships. Word2vec will give a higher similarity if the two words have a similar context. For example, the term”protocol” is highly related to http, TCP, SMTP, etc., as shown in Figure 3. The Word2Vec model is trained using a part of the Google. The Word2Vec similarity Vsim between two vectors of words Ai and Aj is calculated as the cosine similarity between vector representation of Ai and vector of Aj as in Eq. 4.

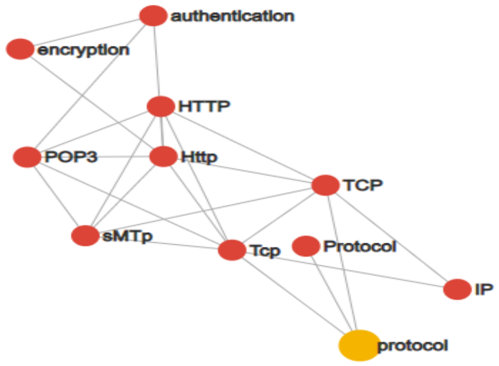

Figure 3. Word2vec Vector Representation Of The Word “Protocol.”

$$V_{sim}(A_i,A_j) = \frac{\sum_{i=1,j=1}^{n} A_i \times A_j}{\sqrt{\sum_{i=1}^{n} A_i^{2}} \times \sqrt{\sum_{i=1}^{n} A_j^{2}}} \quad (4)$$

The final fused semantic similarity (sim) is calculated as the average between Wsim and Vsim measures to the Eq. 5:

$$Sim\,(A_i\ ,A_j) = \frac{Wsim\,(A_i,A_j)+Vsim\,(A_i,A_j)}{2} \quad (5)$$

The final difficulty score of the question is the weighted average between the context measure and semantic measure which is calculated according to the Eq. 6.

Difficulty_Score =

$$\alpha\left(\frac{\sum_{i=1}^{n} Conf(\,C\,,A_i)}{n}\right) + \cdot\beta\left(\frac{\sum_{i=1}^{n} Sim\,(\,C\,,A_i)}{n} + \frac{\sum_{i=1,j=i+1}^{n} Sim\,(A_i,A_j)}{\frac{n(n-1)}{2!}}\right) \quad (6)$$

Where, C denotes concept mentioned in the question, A is the set of distractors, and n is the number of distractors. α and β are weighted values for context and semantic measures respectively, where α+β=1

As mentioned earlier, a question is considered to be very easy if the Difficulty_Score is lower 20%, hard if it is is in the top 80%. The question becomes more difficult if there is a high frequent relationship between the concept and distractors, which is clarified by the context measure. Moreover, whenever the semantic similarity between the distractors is large, the overall Difficulty_Score is become also high. Therefor it is agreed with our assumption that students may become confused and find it hard to answer the question, and hence, it can be concluded that the question is difficult.

3.3 Exam Generation Layer

It contains a module for generating the examination paper. It works according to the pattern defined in the exam profile feeded by instructor. The output of a successful generation can be stored as a pdf for further printing, reprography and storage.

3.4 Automatic Scoring Layer

Based on text mining techniques with the aid of both similarity measure algorithms and NLP, the students' essay answers would be automatically assessed. Thus, the burden of manual grading would be erased from the instructor's duties. As the process of grading becomes automatic, the instructors would only follow the answers of students upload, which refines the learning process. The automatic scoring layer depend on different string-based and corpus-based similarity algorithms. The string based similarity measures are N-gram, matching coefficient, Needleman-Wunsch, Damera-Levenshtein, and Jaccard similarity [33]. The corpus-based similarity measures are extracting distributionally similar words using co-occurrences and Latent Semantic Analysis LSA. This layer consists of two main components, namely the grading engine and fusion engine. The objective of the grading engine is to compute the similarity values between the model answer and student answer using a number of text similarity measuring algorithms (string-based and corpus-based algorithms) separately under different scenarios. After obtaining the similarity scores in the grading engine component, the role of the fusion engine is to combine the different similarity values obtained to enhance the correlation between the model answer and the student answers.

3.5 Data Analytics Layer

The aim of the data analytics layer is two folds. The first is to refine the complexity level of each question based on the analysis of students' responses using descriptive statistics. Accordingly, the system will adaptively change the level of difficulty based on such statistics. The second aim is to inform the instructor with general course statistics like the highest degree, lowest degree, number of students who answer a specific question, etc.

4. EXPERIMENTAL RESULTS

4.1 Dataset

The dataset that is used in the proposed system is the NSC document database [40] obtained from a subset of research reports of the National Science Council, Taiwan, and Republic of China. This dataset consists of 520 documents in computer science related fields, split into 28 categories. The number of documents in each category is between 16 and 25. These documents were selected as learning resources.

4.2 Results and Discussions

To validate our proposed solution for the automatic generation of questions, an experiments has been conducted and analyzed. The confidence measure that is extracted through association rules attains the contextual nature between words by discovering the correlation between terms. Using such dependencies in evaluating the question difficulty significantly increases the effectiveness of the obtained difficulty score of such questions. Table 3, shows the WordNet based semantic similarity matrix between sets of randomly pair words. As it can be noticed, the highest similarity value is 0. 95 that represents the similarity between "windows" and "operating_system" word- pairs. That is according to WordNet hierarchy and using Wu-Palmer method as in eq. 3. Lowest Common Subsumer (LCS) = argmax (depth (subsumer (windows, operating_system))) =10,

depth(windows) = min(depth({tree in software | tree containsLCS}))=11depth(operating_system) = min(depth({tree in operating_system | tree contains LCS })) = 10 Wsim = 2 * depth(LCS) / (depth(software) + depth(operating_system) = 2 * 10 / (11 + 10) = 0.95.

Table 3: Semantic similarity matrix based on WordNet similarity between sets of randomly selected words.

In the following three MCQ questions, we show three candidate questions that are versions of the same question with different distractors.

Q 1: The internet system consists a set of ……….. that are interconnected together to enable data exchange.

Choices: (a) Computers (b) Addresses
(c) Protocols (d) Devices

Q 2: The internet system consists a set of ……….. that are interconnected together to enable data exchange.

Choices: (a) Computers (b) Programs

(c) Browsers (d) Devices

Q 3: The internet system consists a set of ……….. that are interconnected together to enable data exchange.

Choices: (a) Homepages (b) Softwares

(c) Browsers (d) Devices.

Table 3: Semantic similarity matrix based on WordNet similarity between sets of randomly selected words.

Computer	1.00	0.27	0.89	0.44	0.76	0.29	0.67	0.74	0.25	0.76	0.19	1.00
Protocol	0.27	1.00	0.57	0.80	0.40	0.67	0.21	0.43	0.47	0.40	0.42	0.27
devices	0.27	1.00	0.57	0.80	0.40	0.67	0.21	0.43	0.47	0.40	0.42	0.27
address	0.44	0.80	0.77	1	0.63	0.71	0.42	0.50	0.89	0.76	0.80	0.44
windows	0.89	0.57	1.00	0.77	0.84	0.62	0.74	0.82	0.53	0.53	0.20	0.89
database	0.29	0.67	0.62	0.71	0.43	1.00	0.22	0.46	0.50	0.42	0.40	0.29
internet	0.44	0.80	0.77	1	0.63	0.71	0.42	0.50	0.89	0.76	0.80	0.44
network	0.74	0.43	0.82	0.50	0.75	0.46	0.90	1.00	0.40	0.53	0.30	0.74
software	0.76	0.40	0.84	0.63	1.00	0.43	0.64	0.75	0.38	0.53	0.81	0.76
browser	0.25	0.47	0.53	0.89	0.38	0.50	0.20	0.40	1.00	0.86	0.35	0.25
program	0.29	0.67	0.62	0.71	0.43	1.00	0.22	0.46	0.50	0.42	0.40	0.29
operating system	0.19	0.42	0.21	0.80	0.82	0.44	0.18	0.35	0.35	0.95	1.00	0.19
	Computer	Protocol	device	address	windows	database	internet	network	software	browser	program	operating_ system

The Key answer of these three questions is option (d) and other choices are distractors. As mentioned earlier, the RPL layer extracts the concepts through searching the relations summarized in table 1. Hence the extracted concept is the word "internet". The difficulty_ Score of the three questions is calculated using Eq 6. Table 4 summarizes the results. To calculate the difficulty score for this question, we obtain the association rule measure concept (internet) and the alternative choices, as well as calculating the semantic similarity measure between the word, "internet," and every alternative choice word and between the choices together. The semantic similarity is calculated by the average between the WordNet-based similarity and vector-based similarity. As the Difficulty_Score for Q1 is between 0.60 and 0.80, the difficulty level of this question is "hard." It is noticeable that the difficulty score is mainly affected by the question and similar to the key, students may find it difficult to answer the question, and hence, it can be concluded that the question is hard. For Q2, the Difficulty_Score as noticeable in Table 2 is between 40 and 60, so its difficulty level is average.

However, Question 3, as the value of the difficulty of the question is between 0.20 and 0.40, the difficulty level of the question is "easy." The reason for transforming this question to easy is that the low context similarity between question concept and distractors as noticed in Table 2. To study the impact of both context and semantic similarity on the difficulty of the questions, the value of α and β are changed according to table 5. As noticed in table 5, it indicates that the higher the weighted value of the context similarity, the greater of Difficulty_Score, which may sometimes change the state of the question from an average question to a hard one. For question 1, when α =0 and β =1 (i.e. Context similarity is ignored), the Difficulty_Score = 0.39 with difficulty level easy. When α =1 and β =0 (i.e. Semantic similarity is ignored), the difficulty level for this question is transformed to become very hard. This indicates that the impact of context similarity on the question difficulty class is higher than the effect of semantic similarity. Depending on both context and semantic similarity makes the classification of question difficulty is more accurate than depending on only one of them. It is noticeable in Q3 that the difficulty of question does not affect by changing the factors α and β due to low values of context measures between the concept and the distractors. Table 5. Results of changing the impact of context similarity and semantic similarity on the difficulty score.

Table 4 Calculating difficulty score of Question 1, Question 2, and Question 3 with α=0.5 and β=0.5.

	A_i		A_j	**Vector similarity V_{sim} (A_i,A_j)**	**WordNet similarity W_{sim} (A_i,A_j)**	**Context similarity Conf(A_i,A_j)**	**Difficult y_ Score**
QUESTION 1	Concept	Internet	Computers	0.40	0.66	1.00	**0.64**
			Addresses	0.16	0.42	1.00	
			Protocols	0.08	0.21	1.00	
			Devices	0.07	0.73	0.56	
	Distractors	Computers	Addresses	0.13	0.44		
		Computers	Protocols	0.10	0.26		
		Computers	Devices	0.37	0.88		
		Addresses	Protocols	0.15	0.88		
		Addresses	Devices	0.06	0.76		
		Protocols	Devices	0.25	075		
QUESTION 2	Concept	Internet	Computers	0.40	0.66	1.00	**0.55**
			Programs	0.16	0.42	0.61	
			Browsers	0.08	0.22	0.66	
			Devices	0.07	0.73	0.56	
	Distractors	Computers	Programs	0.13	0.19		
		Computers	Browsers	0.27	0.76		
		Computers	Devices	0.37	0.88		
		Programs	Browsers	0.15	0.95		
		Programs	Devices	0.34	0.22		
		Browsers	Devices	0.30	0.61		
QUESTION 3	Concept	Internet	Homepages	0.42	0.17	0.04	**0.29**
			Softwares	0.35	0.20	0.28	
			Browsers	0.35	0.42	0.35	
			Devices	0.07	0.73	0.56	
	Distractors	Homepages	Softwares	0.29	0.38		
		Homepages	Browsers	0.48	0.33		
		Homepages	Devices	0.10	0.30		
		Softwares	Browsers	0.17	0.40		
		Softwares	Devices	0.22	0.58		
		Browsers	Devices	0.30	0.61		

Table 5. Results of changing the impact of context similarity and semantic similarity on the difficulty score.

	α	β	Context similarity	Semantic similarity	Difficulty_ Score	Class
Question 1	0.00	1.00	0.00	0.39	0.39	easy
	1.00	0.00	0.90	0.00	0.90	Very hard
	0.50	0.50	0.44	0.19	0.64	hard
	0.75	0.25	0.66	0.09	0.76	hard
	0.25	0.75	0.22	0.29	0.51	average
	0.40	0.60	0.35	0.23	0.59	average
	0.60	0.40	0.53	0.15	0.69	hard
	0.90	0.10	0.80	0.03	0.84	Very hard
	0.10	0.90	0.09	0.35	0.44	average
	0.00	1.00	0.00	0.39	0.39	easy
	1.00	0.00	0.90	0.00	0.89	Very hard
Question 2	0.00	1.00	0.00	0.38	0.38	easy
	1.00	0.00	0.80	0.00	0.70	hard
	0.5	0.50	0.40	0.19	0.59	average
	0.75	0.25	0.60	0.09	0.69	hard
	0.25	0.75	0.20	0.28	0.48	average
	0.40	0.60	0.32	0.22	0.55	average
	0.60	0.40	0.48	0.15	0.63	hard
	0.90	0.10	0.72	0.03	0.76	hard
	0.10	0.90	0.08	0.34	0.42	average
Question 3	0.00	1.00	0.34	0.00	0.34	easy
	1.00	0.00	0.00	0.23	0.23	easy
	0.5	0.50	0.17	0.11	0.28	easy
	0.75	0.25	0.10	0.16	0.26	easy
	0.25	0.75	0.26	0.05	0.31	easy
	0.40	0.60	0.20	0.09	0.30	easy
	0.60	0.40	0.13	0.13	0.27	easy
	0.90	0.10	0.03	0.20	0.24	easy
	0.10	0.90	0.31	0.02	0.33	easy

5. CONCLUSION

Automatic student assessment technology can be considered to be an effective solution to the challenges associated with manual assessment. Herein, a fully integrated online assessment framework that relies on an ontological learning method to transform the learning topics into an ontological form was presented. The proposed framework can generate candidate questions with different difficulty scales based on a hybrid technique of semantic and context analyses. This research is of great significance to decision makers and designers of e-learning systems as it can serve to enhance the achievements of both students and instructors in higher education. In our future studies, we intend to relate the generated questions to the course learning outcomes using different semantic methods to achieve semantic similarity.

ACKNOWLEDGMENT

We are indebted to the Deanship of Scientific Research at Jouf University for funding this work through General Research Project under grant number (40/254).

ETHICS

This paper is original and contains unpublished material. No ethical issues were involved and the authors have no conflict of interest to disclose.

REFERENCES

[1] A. D. Alrehily, M. A. Siddiqui, and S. M. Buhari, "INTELLIGENT ELECTRONIC ASSESSMENT FOR SUBJECTIVE EXAMS," CSIT, 2018.

[2] J. Leo, G. Kurdi, N. Matentzoglu, B. Parsia, U. Sattler, S. Forge, et al., "Ontology-based generation of medical, multi-term MCQs," International Journal of Artificial Intelligence in Education, vol. 29, pp. 145-188, 2019.

[3] E. Vinu, "A novel approach to generate MCQs from domain ontology: Considering DL semantics and open-world assumption," Journal of Web Semantics, vol. 34, pp. 40-54, 2015.

[4] H. Wang, X. Zhang, and H. Wang, "A Neural Question Generation System Based on Knowledge Base," in CCF International Conference on Natural Language Processing and Chinese Computing, 2018, pp. 133-142.

[5] K. Stasaski and M. A. Hearst, "Multiple choice question generation utilizing an ontology," in Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, 2017, pp. 303-312.

[6] R. Bhirangi and S. Bhoir, "Automated question paper generation system," Computer Engineering Department, Ramrao Adik Institute of Technology, Navi Mumbai, Maharashtra, India, 2016.

[7] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A Systematic Review of Automatic Question Generation for Educational Purposes," International Journal of Artificial Intelligence in Education, pp. 1-84, 2019.

[8] R. Das, A. Ray, S. Mondal, and D. Das, "A rule based question generation framework to deal with simple and complex sentences," in 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 542-548.

[9] P. Khullar, K. Rachna, M. Hase, and M. Shrivastava, "Automatic question generation using relative pronouns and adverbs," in Proceedings of ACL 2018, Student Research Workshop, 2018, pp. 153-158.

[10] M. Agarwal and P. Mannem, "Automatic gap-fill question generation from text books," in Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, 2011, pp. 56-64.

[11] L. Odilinye, F. Popowich, E. Zhang, J. Nesbit, and P. H. Winne, "Aligning automatically generated questions to instructor goals and learner behaviour," in Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), 2015, pp. 216-223.

[12] G. Danon and M. Last, "A syntactic approach to domain-specific automatic question generation," arXiv preprint arXiv:1712.09827, 2017.

[13] M. Heilman, "Automatic factual question generation from text," Language Technologies Institute School of Computer Science Carnegie Mellon University, vol. 195, 2011.

[14] V. Harrison and M. Walker, "Neural generation of diverse questions using answer focus, contextual and linguistic features," arXiv preprint arXiv:1809.02637, 2018.

[15] O. R. Rocha and C. F. Zucker, "Automatic generation of educational quizzes from domain ontologies," 2017.

[16] A. Bongir, V. Attar, and R. Janardhanan, "Automated quiz generator," in The International Symposium on Intelligent Systems Technologies and Applications, 2017, pp. 174-188.

[17] B. Diatta, A. Basse, and S. Ouya, "Bilingual Ontology-Based Automatic Question Generation," in 2019 IEEE Global Engineering Education Conference (EDUCON), 2019, pp. 679-684.

[18] A. Faizan, S. Lohmann, and V. Modi, "Multiple choice question generation for slides," in Computer Science Conference for University of Bonn Students, 2017, pp. 1-6.

[19] A. Faizan and S. Lohmann, "Automatic generation of multiple choice questions from slide content using linked data," in Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, 2018, pp. 1-8.

[20] V. Kumar, G. Ramakrishnan, and Y.-F. Li, "A framework for automatic question generation from text using deep reinforcement learning," arXiv preprint arXiv:1808.04961, 2018.

[21] B. Liu, M. Zhao, D. Niu, K. Lai, Y. He, H. Wei, et al., "Learning to Generate Questions by LearningWhat not to Generate," in The World Wide Web Conference, 2019, pp. 1106-1118.

[22] Y. Kim, H. Lee, J. Shin, and K. Jung, "Improving neural question generation using answer separation," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 6602-6609.

[23] Z. Wang, A. S. Lan, W. Nie, A. E. Waters, P. J. Grimaldi, and R. G. Baraniuk, "QG-net: a data-

driven question generation model for educational content," in Proceedings of the Fifth Annual ACM Conference on Learning at Scale, 2018, pp. 1-10.

[24] L. Song, Z. Wang, W. Hamza, Y. Zhang, and D. Gildea, "Leveraging context information for natural question generation," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 569-574.

[25] T. Liu, B. Wei, B. Chang, and Z. Sui, "Large-scale simple question generation by template-based seq2seq learning," in National CCF Conference on Natural Language Processing and Chinese Computing, 2017, pp. 75-87.

[26] L. A. Tuan, D. J. Shah, and R. Barzilay, "Capturing Greater Context for Question Generation," arXiv preprint arXiv:1910.10274, 2019.

[27] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3901-3910.

[28] E. Choi, D. Hewlett, J. Uszkoreit, I. Polosukhin, A. Lacoste, and J. Berant, "Coarse-to-fine question answering for long documents," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 209-220.

[29] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," arXiv preprint arXiv:1611.01604, 2016.

[30] T. Wang, X. Yuan, and A. Trischler, "A joint model for question answering and question generation," arXiv preprint arXiv:1706.01450, 2017.

[31] B. Dhingra, D. Pruthi, and D. Rajagopal, "Simple and effective semi-supervised question answering," arXiv preprint arXiv:1804.00720, 2018.

[32] M. Tatu, S. Werner, M. Balakrishna, T. Erekhinskaya, and D. Moldovan, "Semantic question answering on big data," in Proceedings of the International Workshop on Semantic Big Data, 2016, pp. 1-6.

[33] A. Shehab, M. Faroun, and M. Rashad, "An automatic Arabic essay grading system based on text similarity Algorithms," Int. J. Adv. Comput. Sci. Appl.(IJACSA), vol. 9, pp. 263-268, 2018.

[34] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "aNMM: Ranking short answer texts with attention-based neural matching model," in Proceedings of the 25th ACM international on conference on information and knowledge management, 2016, pp. 287-296.

[35] J. Liu, Y. Xu, and L. Zhao, "Automated essay scoring based on two-stage learning," arXiv preprint arXiv:1901.07744, 2019.

[36] M. Yamamoto, N. Umemura, and H. Kawano, "Automated essay scoring system based on rubric," in International Conference on Applied Computing and Information Technology, 2017, pp. 177-190.

[37] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 1882-1891.

[38] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in Proceedings of the 1993 ACM SIGMOD international conference on Management of data, 1993, pp. 207-216.

[39] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994, pp. 133-138.

[40] (2020, march 20). A Subset of the Collection of the Research Reports of the National Science Council, Taiwan, Republic of China. Available: http://fuzzylab.et.ntust.edu.tw/NSC_Report_Database/Documents/520documents.html