An Ontology-Based Framework as a Foundation of an Information System for Generating Multiple-Choice Questions

Completed Research

Pornpat Sirithumgul  
Department of Computer Engineering  
Rajamangala University of Technology Phra Nakhon  
pornpat.s@rmutp.ac.th

Pimpaka Prasertsilp  
School of Science and Technology  
Sukhothai Thammathirat Open University  
pimpaka.pra@stou.ac.th

Watanyoo Suksa-Ngiam  
Center for Information Systems and Technology  
Claremont Graduate University  
watanyoo.suksa-ngiam@cgu.edu

Lorne Olfman  
Center for Information Systems and Technology  
Claremont Graduate University  
lorne.olfman@cgu.edu

Abstract

In this paper, we propose an ontology-based framework for formulating multiple-choice questions that can assess students’ knowledge in the analytical level. The questions formulated from this framework are aimed at encouraging students to think about applying relative concepts to produce different paradigms for solving the same problem. In the experiment, we demonstrate an implementation method of the framework in which an initial solution of a problem is introduced in the question phrase, and alternate solutions developed from different paradigms are presented as a correct answer choice and distractors. The experimental results show effectiveness of the questions constructed based on this framework in terms of distractors that can reveal students’ misconceptions. Also, difficulty and discrimination indices (P-scores and D-scores) of the questions show that in a range of difficulty, questions generated based on this framework can reveal strength and weaknesses of students in the group of high and low performance.

Keywords

Ontology-based framework, multiple-choice question generation, knowledge measurement, information systems for education.

Introduction

This research is motivated by online learning, e.g., Massive Open Online Course (MOOCS) (“Massive open online course,” n.d.), and Coursera (“Coursera,” n.d.) requiring an automatic knowledge assessment module for generating questions and assessing knowledge of students from their responses to the questions. Objective rather than subjective questions are generally used in online learning because the answers to this type of questions are easily to be graded, and no additional efforts of instructors are required to analyze students’ subjective writings. However, objective questions employed in online learning are doubtful about their effectiveness in assessing students’ analytical and critical skills. The previous research (Kunichika et al. 2004; Brown et al. 2005; Mitkov et al. 2006; Heilman and Noah 2010; Goto et al. 2010; Smith et al. 2010; Agarwal and Mannem 2011; Kumar et al. 2015; Alsabait et al. 2016; Kwankajornkiet et al. 2016)
succeeded in using natural language processing (NLP) algorithms to automatically compose multiple-choice questions, but questions generated from these algorithms were ‘what’ rather than ‘how-and-why’ questions suitable for assessing knowledge in the ‘understanding’ and ‘analysis’ levels (Forehand, 2005).

This current research is aimed to reduce the limitation of research in the same area (Kunichika et al. 2004; Brown et al. 2005; Mitkov et al. 2006; Heilman and Noah 2010; Goto et al. 2010; Smith et al. 2010; Agarwal and Mannem 2011; Kumar et al. 2015; Alsubait et al. 2016; Kwankajornkiet et al. 2016) by proposing an ontology-based framework as an internal mechanism of an information system for automatically generating multiple-choice questions evaluating students’ knowledge in an analytical level.

Different than other question types, the questions generated from this framework require students to apply multiple relative concepts in a knowledge domain as different paradigms for solving the same challenging problem. A result from assessing answers to the questions would allow instructors to acutely know levels of students’ understandings of the concepts they have learned.

**Knowledge Background**

An ontology design, a guideline for formulating questions of Ragonis (2012), and the three measures for evaluating quality of distractors, difficulty and discrimination power of multiple-choice questions are used as a foundation of constructing the framework proposed in this research.

![Figure 1. Four Types of Concepts' Relations](image)

An ontology design is used in this research to specify concepts and concepts’ relations in a knowledge domain. Figure 1(a) presents an association relation between paired concepts A and B. This type of relations is used to describe any two concepts normally occurring together, e.g., candles and a birthday cake (Budanitsky & Hirst, 2001), or used to compare two opposite concepts, e.g., black-box and white-box testing (Sirithumgul & Olfman, 2013). Figure 1(b) presents generalization relations between the generic concept B, and the three specific concepts B_x, B_y and B_z. A generalization relation would show a specific concept having its own specific characteristic(s), and being inherited a common characteristic from a generic concept (Bollegala et al., 2007). Figure 1(c) and (d) show part-and-whole relations. The relations between B and B_x, B_y and B_z in Figure 1(c) are called composition relations, and the relations between B and B_x, B_y and B_z in Figure 1(d) are called aggregation relations. A difference between the two relation types is that composition is used to represent required components of an entire collective piece, e.g., four wheels and an engine are required components of a car, while aggregation is used to represent a property that can be separated independently from a composite unit, e.g., students aggregate into a class.

The question type ‘transforming a solution from one representation to another’ (Ragonis, 2012) is used as a guide in this current research to formulate analytical questions. Generally speaking, the questions in this type are aimed at motivating students to think about different paradigms for solving the same problem. Ragonis (2012) described through examples in the area of computer programming that the ‘different paradigms’ could mean (a) different methods used to solve the same problem, e.g., two different methods for sorting numbers in an array, (b) a method developed by using different programming languages, e.g., Java and C, (c) different programming structures, e.g., if and switch-case statements for solving the same problem, and (d) different algorithmic approaches, e.g., a loop structure and a recursive method applied to solve the same repetitive task.

Quality of multiple-choice questions generated by the framework proposed in this research is considered from occurrences of non-functioning options, difficulty, and discrimination power of the questions.

A non-functioning option is a distractor in a multiple-choice set chosen by fewer than 5 percent of people answering to the question. A question is considered acceptable if at least one distractor in the multiple-choice questions is chosen by fewer than 5 percent of people answering the question.
choice set is not a non-functioning option (Haladyna and Downing 1993; Cizek and O’Day 1994; Sidick et al. 1994; Delgado and Prieto 1998; Roger and Harley 1999; Woodford and Bancroft 2005; Shizuka et al. 2006; Tarrant et al. 2009).

Difficulty of a question is considered from a difficulty index (also called P-scores), and discrimination power of a question is considered from a discrimination index (also called D-scores). The equations (eq.1) and (eq.2) are used to calculate P-scores and D-scores, respectively:

\[ P - scores = \frac{h+l}{n} \times 100 \quad (eq.1) \]

\[ D - scores = \frac{h-l}{n} \times 2 \quad (eq.2) \]

Where  \( h \) is the number of students in the group of high achievers who can correctly answer to the question.

\( l \) is the number of students in the group of low achievers who can correctly answer to the question.

\( n \) is the total number of students in the groups of high and low achievers.

The previous research studies (Pande et al., 2013; Rao et al., 2016) suggest that a question should have P-scores between 30 and 70 meaning that this question is neither too easy nor too difficult, and D-scores should be between 0.20 and 0.29 meaning that this question can be used to distinguish students in the groups of high and low performance.

**Proposed Framework**

The framework proposed in this paper comprises four steps. **Step 1** is about ontology specification in which concepts and concepts’ relations in a knowledge domain are specified. **Steps 2 – 4** are about applying the ontology specified in **Step 1** to formulate question phrases, correct answers and distractors, respectively. Details of the four steps are as follows:

**Step 1 – Ontology Specification:** An ontology design, as illustrated in Figure 1, is applied at this step. Paired concepts having a spatial or a temporal relation, for example, should be specified with an association relation (see Figure 1(a)), a generic concept (e.g., a concept about planets) and its instance concepts (e.g., Saturn, Neptune and Uranus) should be specified with generalization relations (see Figure 1(b)), a whole concept (e.g., cell) and parts being comprised the whole (e.g., nucleus, cytoplasm, plasma, membrane, and cytoplasmic organelles) should be specified with composition relations (see Figure 1(c)).

**Step 2 – Question Phrase Formulation:** A guide of Ragonis (2012) about formulating questions encouraging students to think about multiple solutions of the same problem is applied at this step.

---

**Figure 2. An Employment of Relative Concepts in Formulating Alternate Solutions**

By using an ontology (see an example in Figure 2(a)), instructors would have an idea of formulating an initial solution of a problem, e.g., a solution of repetitive work implemented by a while-loop structure as shown in Figure 2(b). This initial solution should be in the question phase also included a statement encouraging students to think about applying other relative concepts in the same domain, e.g., other loop
structures – ‘do...while’ and ‘for’ on composing alternate solutions of the one introduced in the question phrase.

**Step 3 – Correct Answer Formulation:** A relative concept of the concept applied to compose an initial solution is used to formulate a correct answer at this step. As shown in Figure 2(c), the loop structure ‘for’ – a relative concept of ‘while’ applied in the initial solution is used to formulate an alternate solution in the correct answer.

**Step 4 – Distractor Formulation:** A guide of previous research (Haladyna and Downing 1993; Cizek and O’Day 1994; Sidick et al. 1994; Delgado and Prieto 1998; Roger and Harley 1999; Woodford and Bancroft 2005; Shizuka et al. 2006; Tarrant et al. 2009) about formulating distractors that are not non-functioning options is applied at this step.

To avoid a non-functioning option, distractors should share some kind of similarity with a correct answer in the same multiple-choice set. This framework suggests using relative concepts in the ontology specified at step 1 to formulate distractors as they share similarity through concepts’ relations. As specified at Step 1, paired concepts sharing an association relation, instance concepts sharing a generalization relation with a generic concept, and component concepts sharing a composition/aggregation relation with a whole concept should be used to formulate distractors. As shown in Figure 2(a), for example, the instance concepts – ‘while’, ‘do...while’, and ‘for’ should be used to formulate distractors as they share similarity about loop structure.

**Experiment**

To test the proposed framework, we conducted an experiment by applying the framework on formulating 11 questions about computer programming. These questions were further used to ask 55 first-year college students who were enrolled in the computer programming course. All students participating in this experiment were from the Computer Engineering department in a university. Before taking this course, most students in this group rarely had a background in programming, and only a few were trained programming in high school. The details of formulating questions used in the experiment are as follows:

**Step 1 – Ontology Specification:** An ontology comprising eleven concepts in the domain of computer programming is specified at this step. As presented in Table 1, the eleven concepts are classified into five groups of conceptual ideas. Four groups (Groups 1 – 4) contain relative concepts sharing generalization relations, and one group (Group 5) contains two relative concepts of a composition relation.

<table>
<thead>
<tr>
<th>Group</th>
<th>Conceptual Idea</th>
<th>Relative Concepts</th>
<th>Relation Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conditional structure</td>
<td>‘if’ and ‘switch...case’ statements</td>
<td>Generalization</td>
</tr>
<tr>
<td>2</td>
<td>Sentence construction</td>
<td>String (a variable type) and an array of characters</td>
<td>Generalization</td>
</tr>
<tr>
<td>3</td>
<td>Loop structure</td>
<td>‘for’, ‘while’, and ‘do...while’ structures</td>
<td>Generalization</td>
</tr>
<tr>
<td>4</td>
<td>Methods for calling a function</td>
<td>calling by value and calling by reference</td>
<td>Generalization</td>
</tr>
<tr>
<td>5</td>
<td>Array</td>
<td>pointer and address</td>
<td>Composition</td>
</tr>
</tbody>
</table>

Table 1. Concepts and Concepts’ Relations Specified in the Ontology

**Step 2 – Question Phrase Formulation:** A question phrase containing an initial solution and a statement asking students to find another similar solution is formulated at this step. Figure 3 is an example showing an initial solution in which the loop structure ‘for’ – a concept in Group 3 (see Table 1) is applied. There is also a statement in the question phrase asking students to find code in the multiple choices of which result is similar to the initial solution.
Question 7: Which one in the following choice does give the same answer of the code given in Figure 7?

```
#include <stdio.h>
void main()
{
    int a;
    for(a=10; a<19; a++){
        printf("a = %d \n", a);
    }
    printf("a = %d \n", a);
}
```

Figure 7

Figure 3. An Example of Formulating a Question Phrase

Step 3 – Correct Answer Formulation: A relative concept of the concept applied in the initial solution is used to formulate a correct answer. Figure 4(a) is an example of using a relative concept (i.e., 'do...while') of the concept applied in the initial solution (i.e., 'for') to formulate a correct answer.

```
#include <stdio.h>
void main()
{
    int a = 10;
    do{
        printf("a = %d \n", a);
    } while(a < 20);
    printf("a = %d \n", a);
}
```

(a)  

```
#include <stdio.h>
void main()
{
    int a = 10;
    while(a < 20){
        a = a + 1;
        printf("a = %d \n", a);
    }
}
```

(b)  

Figure 4. (a) Correct Answer and (b) Distractor

Step 4 – Distractor Formulation: Relative concepts in the same group are used to formulate distractors at this step. Figure 4(b) is an example of applying a relative concept (i.e., 'while') of the concept 'for' applied as an initial solution, and the concept 'do...while' applied as a correct answer on formulating a distractor. As seen in Figure 4(a) and (b), in the level of abstraction, relative concepts cause similarity between a correct answer and a distractor. However, in the implementation level, a distractor (see Figure 4 (b)) is different than the correct answer as it is fed with an error.

Table 2 is a summary of concepts used to formulate eleven questions (Q1 – Q11). The concepts used to formulate initial solutions are shown in column ‘Initial Solutions’. Column ‘Correct Answers’ gathers relative concepts of the concepts used in the initial solutions to formulate correct answers of the eleven questions. Column ‘Distractors’ gathers all relative concepts in the same conceptual idea group to formulate distractors.

<table>
<thead>
<tr>
<th>Conceptual Idea</th>
<th>Questions</th>
<th>Initial Solutions</th>
<th>Correct Answers</th>
<th>Distractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditional structure</td>
<td>Q1</td>
<td>switch...case</td>
<td>if</td>
<td>switch...case, if</td>
</tr>
<tr>
<td></td>
<td>Q2</td>
<td>if</td>
<td>switch...case</td>
<td>switch...case, if</td>
</tr>
<tr>
<td>Sentence construction</td>
<td>Q3</td>
<td>String (variable type)</td>
<td>character array</td>
<td>String and character array</td>
</tr>
<tr>
<td></td>
<td>Q4</td>
<td>character array</td>
<td>String (variable type)</td>
<td>String and character array</td>
</tr>
</tbody>
</table>

Table 2. Summary of Concepts Applied to Formulate Questions
An Ontology-Based Framework for Formulating Multiple-Choice Questions

Conceptual Idea | Questions | Initial Solutions | Correct Answers | Distractors |
--- | --- | --- | --- | --- |
Loop structure | Q5 | for | while | for, while, do...while |
| Q6 | while | for | for, while, do...while |
| Q7 | do...while | for | for, while, do...while |
Methods for calling a function | Q8 | calling by value | calling by reference | calling by value, calling by reference |
| Q9 | calling by reference | calling by value | calling by value, calling by reference |
Array | Q10 | pointer | address | pointer, address |
| Q11 | address | point | pointer, address |

Table 2. Summary of Concepts Applied to Formulate Questions (continued)

Experimental Results

The results from using the 11 questions formulated based on the framework to test the 55 students were further used to evaluate quality of the questions in three aspects including (a) quality of distractors considered from an occurrence of non-functioning distractors, (b) difficulty, and (c) discrimination power of the questions.

Quality of distractors in a multiple-choice question is considered from an occurrence of non-functioning options. As suggested by previous research (Haladyna and Downing 1993; Cizek and O’Day 1994; Sidick et al. 1994; Delgado and Prieto 1998; Roger and Harley 1999; Woodford and Bancroft 2005; Shizuka et al. 2006; Tarrant et al. 2009), a non-functioning option is a distractor chosen by fewer than 5 percent of the total number of students answering to a question, and the question is considered reliable if there is at least one out of three distractors not a non-functioning option.

Especially for this experiment, a non-functioning option means a distractor chosen by fewer than 3 students (the number is from 5 percent of the total 55 students answering to a question). The outcome from the test applied on the students shows all 11 multiple-choice questions are reliable – 8 out of 11 questions do not contain any non-functioning option, 2 out of 11 contain one non-functioning option, and 1 out of 11 contains two non-functioning options.

The result of examining the distractors of the eleven questions could be implied that the proposed framework is valid, and can be used to produce a set of effective distractors in a multiple-choice question. Also, the distractors that are not non-functioning options and chosen by a number of students can be used to imply a misconception that the students may have for relative concepts in the domain.

<table>
<thead>
<tr>
<th>Difficulty Index (P-scores)</th>
<th>Difficult ($P \leq 30$)</th>
<th>Moderate ($30 &lt; P &lt; 70$)</th>
<th>Easy ($P \geq 70$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3. The Number of Questions in the Three Difficulty Levels

A difficulty index (or P-scores) is used to evaluate a difficulty level of a question. As presented in equation (eq. 1) in section ‘Knowledge Background’, P-scores are mainly considered from a ratio of the
number of students in the groups of high and low achievers who can correctly answer to a question to the total number of students in the two groups.

In this experiment, 6 out of 55 students who eventually receive As and B’s from the computer programming course are considered as students in the group of high achievers, and 11 out of 55 students who eventually receive Fs from the course are considered in the group of low achievers. The answers to the eleven questions of the 17 students are used to calculate P-scores. The calculation outcomes are shown in Table 3.

According to Pande et al. (2013), difficulty could be divided into three levels – a difficult level would have P-scores equal to or lower than 30, a moderate level would have P-scores between 30 and 70, and an easy level would have P-scores equal to or greater than 70. In this experiment, we found 2 out of 11 questions are in the difficult level, 6 out of 11 are in the moderate level, and 3 out of 11 are in the easy level.

The outcomes from the experiment in this part can be concluded that the six questions in the moderate level are most appropriate to use as standard questions for evaluating knowledge in the area of computer programming, because they are not too easy or too difficult. However, effectiveness of these questions as a knowledge measure would be best if applied with the students sharing learning backgrounds (e.g., academic majors) with the participants of this research.

Questions in the difficult and easy levels are valid as they can be used as a reflection of students’ understandings of the concepts used to compose the questions. Students tend to have good understandings of concepts used to compose easy-level questions, and rarely understand concepts used to compose difficult-level questions (Pande et al., 2013).

A discrimination index (or D-scores) is used to evaluate discrimination power of a question. As presented in equation (eq.2) in section ‘Knowledge Background’, P-scores are mainly considered from a ratio of a difference between the number of students in the groups of high and low achievers correctly answering to the questions to the total number of students in the two groups.

<table>
<thead>
<tr>
<th>Discrimination Index (D-scores)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>(0 &lt; D ≤ 0.19)</td>
</tr>
<tr>
<td>Moderate</td>
</tr>
<tr>
<td>(0.20 ≤ D ≤ 0.29)</td>
</tr>
<tr>
<td>Exceptional</td>
</tr>
<tr>
<td>(D ≥ 0.30)</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Table 4. The Number of Questions in the Three Discrimination Levels

Similar to P-scores calculation, 17 students are used to calculate D-scores. The outcomes from calculating D-scores of the eleven questions are in Table 4.

A reliable question would have D-scores greater than zero because the number of high achievers is more than the number of low achievers who can correctly answer to the question (Pande et al., 2013). On a contrary, D-scores equal to or lower than a zero would reflect an anomaly of the question as the achievers in the two groups can equally answer to the question, or the low achievers can do better than the high achievers in answering to the question (Pande et al., 2013).

D-scores in Table 4 show the eleven questions formulated from the framework are reliable. The D-scores between 0 and 0.19 mean there is a small difference between the number of high and low achievers who can correctly answer to the questions. The D-scores greater than 0.20 are most desirable (Pande et al., 2013); the D-scores between 0.20 and 0.29 show a moderate difference, and D-scores equal to or greater than 0.30 show a high difference between the number of high and low achievers who can correctly answer to the questions.

Discussions and Future Work

The discussions regarding students’ learning performance, generalizability of research findings, and an implication of future research are presented in this section. In the first discussion, learning performance of students would be implied from a matrix of a correlation between difficulty and discrimination indices (as presented in Table 5). The second discussion would be about generalizability of research findings leading to the idea in the last discussion about a design of an automated system for generating knowledge measurement questions.
Table 5 presents a matrix of a correlation between difficulty and discrimination indices. The questions classified in the different levels of the two indices can be used to describe students’ learning performance as follows:

<table>
<thead>
<tr>
<th>Discrimination Index (D-scores)</th>
<th>Difficulty Index (P-scores)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Difficult</td>
</tr>
<tr>
<td>Minimum</td>
<td>Q9</td>
</tr>
<tr>
<td>Moderate</td>
<td>Q1</td>
</tr>
<tr>
<td>Exceptional</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5. Questions Classified by the Levels of Difficulty and Discrimination Indices

1. The questions of which difficulty is in the moderate level, and discrimination power is in the moderate up to exceptional levels should be used as standard questions for measuring knowledge. Questions Q2, Q5 – Q7, and Q11 (see Table 5) are in this group. These questions, from another view, can be used as a tool for distinguishing students in the groups of high and low achievement.

2. Weaknesses of students with high achievement can be observed from difficult questions of which discrimination power is in the minimum and moderate levels. Questions Q1 and Q9 (see Table 5) are in this group. That a difference of discrimination power between high and low achievers is not very high could be implied that students with high achievement may still have an obstacle with the concepts applied in the questions.

3. Weaknesses of students with low achievement can be observed from easy questions of which discrimination power is in the exceptional level. Question Q4 in Table 5 is in this group. That a difference of discrimination power between high and low achievers is very high for an easy question could be implied that students with low achievement may have an obstacle with the specific concept applied in the question.

4. Strength of the majority group of students can be observed from questions of which difficulty is in the easy up to moderate levels, and discrimination power is in the minimum level. Questions Q3, Q8 and Q10 (see Table 5) are in this group. That a difference of discrimination power between high and low achievers is very low for easy/average questions could be implied that most students – both in the groups of high and low achievement tend to have good understandings of the concepts applied in the questions.

Generalizability of Research Findings

The questions Q1 – Q11 would be most effective if used as a knowledge assessment tool for students of whom training background was similar to that of participants in this research. The framework for formulating questions, and the idea of using a correlation matrix for analyzing learning performance, however, could be generalized in a broader context. They can generally be used to formulate questions in any knowledge domains, and assess knowledge of students in any academic majors.

An Implication of Current Research and Future Work

The experimental results can be implied that the framework proposed in this research is worth using by instructors to manually generate questions reflecting students’ knowledge and confusion of relative concepts in the same knowledge domain. Also, the framework can be used by information system developers as an internal mechanism design of a system for automatically formulating multiple-choice questions. Figure 5 presents a high-level architectural design of the system proposed as future work of this research.
The automated system (see Figure 5) comprises three main modules including a repository, a question generation module and a knowledge assessment module. The duties of the three modules are as follows:

**A repository** is used for collecting parts of multiple-choice questions including initial solutions, correct answers, and distractors (or incorrect answers). Essentially, initial solutions and correct answers are the same group of data because they can be used interchangeably. That is, a correct answer can be used as an answer choice of one question, and an initial solution of another.

**A question generation module** would retrieve data from the repository to formulate multiple-choice questions in the way described in the framework of this research. The outcomes of this module are various sets of multiple-choice questions. That parts of multiple-choice questions are separately collected in the repository would be a benefit for constructing a variety set of questions by this module.

**A knowledge assessment module** would analyze answers to the questions generated by the question generation module. The matrix of a correlation between difficulty and discrimination indices (as presented in Table 5) would be an important tool for analyzing students’ learning performance. The outcomes from analyzing students’ learning performance would be used to generate a report sent to an instructor. Finally, a report from the system would allow the instructor to know understanding levels of students with high and low achievement.

**Conclusion**

This research paper proposes a framework for formulating multiple-choice questions. The experiment in this research demonstrates that questions formulated from the framework can be used as a tool for measuring knowledge in a domain, and can also be used to analyze students’ understandings of concepts they have learned. Lastly, this research paper proposes a high-level architectural design of an automated system in which the framework and a knowledge assessment method are suggested as an internal mechanism of the system.

**REFERENCES**


An Ontology-Based Framework for Formulating Multiple-Choice Questions


T. Alsubait, B. Parsia and U. Sattler, Ontology-based multiple choice question generation, Künstliche Intelligenz 30(2) (2016), 183–188.