AUTOMATED TEST ASSEMBLY WITH MINIMUM REDUNDANT QUESTIONS BASED ON BEE ALGORITHM

BY

MR. VORAPON LUANTANGSRIUK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2017

COPYRIGHT OF THAMMASAT UNIVERSITY
AUTOMATED TEST ASSEMBLY WITH MINIMUM REDUNDANT QUESTIONS BASED ON BEE ALGORITHM

BY

MR. VORAPON LUANTANGSRISUK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2017

COPYRIGHT OF THAMMASAT UNIVERSITY

Ref. code: 25605609035091CUN
THAMMASAT UNIVERSITY
FACULTY OF SCIENCE AND TECHNOLOGY

THESIS

BY

MR. VORAPON LUANTANGSRISUK

ENTITLED

AUTOMATED TEST ASSEMBLY WITH MINIMUM REDUNDANT QUESTIONS BASED ON BEE ALGORITHM

was approved as partial fulfillment of the requirements for
the degree of Master of Science (Computer Science)
on July 19, 2018

Chairman

(Ms. Assistant Professor Nuttawon Hongwarittorn, Ph.D.)

Member and Advisor

(Pokpong Songmuang, Ph.D)

Member

(Tanatorn Tanantong, Ph.D.)

Member

(Asst. Prof. Kreangsak Tamee, Ph.D.)

Dean

(Associate Prof. Somchai Chakhatrakan, Ph.D.)
ABSTRACT

This thesis proposes an automated test assembly based on Bee algorithm that minimizes the redundant questions in a test form. The redundant questions are a question similar to another question. We divide the redundant question into two groups, 1) Similar question pattern, i.e. the questions containing similar textual pattern to another question. The questions are mostly the same in asking style but different subject or focused item. 2) Paraphrase question, i.e. the questions that contain the same meaning or is comparable to another question in the different wordings. The test form is assembled using Bee Algorithm. A neighborhood search in Bee algorithm is improved by using a new technique, called Max-Pair-Similarity (MPS). The MPS is the similarity score of considered question compared to others question in the test form. The MPS of questions are able to indicate the redundant questions in the test form. For experiment, the chosen dataset contains 400 questions collected from Information Technology Professional Examination (ITPE). The dataset has 3,755 redundant question pairs. The experiments compare the redundant question pair with two similarity functions, cosine similarity and, Text Semantic function with path similarity. The result shows that the proposed method can positively lower the number of redundant questions in test assembly.
task. In comparison, using the cosine similarity in MPS shows the better results to minimize the redundant questions.

**Keywords:** Automated Test Assembly, Bee Algorithm, Neighborhood Search, TF-IDF, Cosine Similarity, Text Semantic Similarity
ACKNOWLEDGEMENTS

I would like to express my special thanks to my advisor, Dr. Pokpong Songmuang, for the helpful suggestion for this study. I am grateful to the committee members, Assistant Professor Nattanont Hongwarittorn, Assistant Professor Kreangsak Tamee and Dr. Tanatorn Tanantong, for the valuable comments. I would like to thank my roommate who gave the resource for running the experiments. I would like to thank the department officer who helps for the course suggestion. And I wish to special thank my family for their support.

Finally, I would like to thank the Faculty of Science and Technology who give many supports for the students. The first one is “Scholarship for talent student to study graduate program in Faculty of Science and Technology Thammasat University”. This scholarship gives a budget for the enrollment in the master degree. Moreover, I would like to thank the Faculty of Science and Technology who support for the International and national presentation such as a registration budget and transportation.

Mr. Vorapon Luantangsrisuk
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>6</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Rationales</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>2</td>
</tr>
<tr>
<td>CHAPTER 2 REVIEW OF LITERATURE</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Automated Test Assembly</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Bee Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Redundant Question</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Text Similarity</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 TF-IDF</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2 Cosine Similarity</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Text Semantic Similarity (TSS)</td>
<td>10</td>
</tr>
<tr>
<td>2.4 Related Research</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 3 RESEARCH METHODOLOGY</td>
<td>13</td>
</tr>
</tbody>
</table>

Ref. code: 25605609035091CUN
3.1 Objective Function 13
3.2 Similarity function 14
3.3 Bee Algorithm for Automated Test Assembly 15
3.4 Improved Neighborhood Search for Minimizing Redundant Question 16
  3.4.1 Max-Pair-Similarity (MPS) 16
  3.4.2 Selection 16
  3.4.3 Replacement 17
3.5 Question pool 17
3.6 Experiments 17
  3.6.1 Experiment Setting 1 17
  3.6.2 Experiment Setting 2 18

CHAPTER 4 RESULTS AND DISCUSSION 19

4.1 Result 19
  4.1.1 The experiment result 1 19
  4.1.2 The experiment result 2 21
4.2 Discussion 23

CHAPTER 5 CONCLUSION 27

REFERENCES 29

APPENDIX 31

BIOGRAPHY 32
LIST OF TABLES

<table>
<thead>
<tr>
<th>Tables</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 The average number of similar pattern question pairs with different test length on the new question pool</td>
<td>23</td>
</tr>
<tr>
<td>4.2 The average number of paraphrase question pairs with different test length on the new question pool</td>
<td>23</td>
</tr>
<tr>
<td>A-1 the question pool from ITPE-FE from 2007-2011</td>
<td>31</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figures</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Example of similar pattern question</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Example of paraphrase question</td>
<td>8</td>
</tr>
<tr>
<td>4.1 The result of experiment 1: Average number of similar pattern</td>
<td>19</td>
</tr>
<tr>
<td>question pairs in each iteration of BA</td>
<td></td>
</tr>
<tr>
<td>4.2 The result of experiment 1: Average number of paraphrase question</td>
<td>20</td>
</tr>
<tr>
<td>pairs in each iteration of BA</td>
<td></td>
</tr>
<tr>
<td>4.3 The result of experiment 2: Average number of similar pattern</td>
<td>21</td>
</tr>
<tr>
<td>question pairs in each iteration of BA</td>
<td></td>
</tr>
<tr>
<td>4.4 The result of experiment 2: Average number of paraphrase question</td>
<td>22</td>
</tr>
<tr>
<td>pairs in each iteration of BA</td>
<td></td>
</tr>
<tr>
<td>4.5 The example of the paraphrase question that BA-TSS is failed to</td>
<td>25</td>
</tr>
<tr>
<td>remove.</td>
<td></td>
</tr>
<tr>
<td>4.6 The example of the normal question that has TSS score more than</td>
<td>25</td>
</tr>
<tr>
<td>some paraphrase question</td>
<td></td>
</tr>
<tr>
<td>4.7 The example of the normal question that has TSS score more than</td>
<td>26</td>
</tr>
<tr>
<td>some paraphrase question (Cont.)</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Rationales

In test design, a test assembly is one of the reliable methods to automatically construct a test form from a pool of questions. The method has been researched in many works (1–5) and proved to be reliable and practical. The result of the test assembly is to generate a test form (testlet) by considering an objective function and constraints specified by test administrators. The quality regarding to the test specification and computational complexity issue have been solved with the existing works (1–3).

However, the issue in the generated test form is still remained. The issue is that there is redundancy in the questions. The redundant questions are in a form of similar questions in the same test form. The similarity can be defined into two types. The first one is the similar question pattern of two or more questions. Commonly, the questions with the same pattern are differentiated with the subject of the questions. This issue leads to lower quality to measure tester’s knowledge since it can give an intentional hint by trimming the already chosen choices from the previous question. The second redundancy of the generated question is the similar question concept. The questions with the same topic (paraphrase question or comparable question) create unfairness among test forms since the significant number of topics in the test form is reduced.

The redundancy of the questions generated into a test form is a major issue to reduce quality of the test which is not preferred. In the existing work, the selection of the questions to form a test form does not concern on the redundant issue but computational complexity. To prevent a redundancy of the questions, similarity detection technique to recognize redundant question can be applied. There are two main approaches in similarity detection as term-based similarity and knowledge-based similarity. The term-based approach measures an overlapping term between two texts, namely questions in this task. Calculation methods in this
approach are such as Euclidean distance (6) and Cosine similarity (7). The knowledge-based approach measures a similarity by using information from the semantic network such as WordNet (8). There are various techniques in the knowledge-based approach including Path similarity (9), and Wu and Palmer similarity (10).

In this research, both term-based similarity and knowledge-based similarity are applied to solve a redundancy problem in an automated test assembly. The similarity methods are planned to develop into the assembly process. The redundant questions in terms of similar pattern are expected to be handled with the term-based approach while the redundant questions of similar concepts are to be solved with knowledge-based approach. The study comparing the selected methods is to be conducted to find advantages and disadvantages of the chosen methods.

The rest of this research is organized as follows. Chapter 2 provides background knowledge of the task and a summary of related works. Chapter 3 explains the method to develop automated test assembly that can prevent redundancy of similar questions. Chapter 4 describes results of the proposed method and discussion. Chapter 5 is for conclusion and future work.

1.2 Objectives

- To study question redundancy problem in automated test assembly
- To design and develop a method to minimize similar questions within a test form generated by automated test assembly process without using external corpus

1.3 Scope

The question pool in this thesis is the dataset from the Information Technology Professional Examination (ITPE) (11). The website provides preparation test data in two tracks, Fundamental Information Technology Professional
Examination (FE) and Software Design & Development Professional Examination (SW). This thesis is collected the data from FE track in year 2007-2011. The type of the question in the dataset is a multiple choices. The language of the question is written in English. The dataset consist of 400 questions. The redundant question is manually tagged. There are 3,755 redundant question pairs.
CHAPTER 2
REVIEW OF LITERATURE

This chapter describes four topics. First, the automated test assembly and algorithm for automated test assembly are reviewed. Second, definition of the redundant question and its specification are explained. Third, techniques to detect similarity of texts are described. Last, existing researches related to the topic are reviewed including test assembly, text similarity and redundancy reducing.

2.1 Automated Test Assembly

Test assembly is a task to select questions from a question pool to form a test form. The automated test assembly was developed to assist in generating test forms automatically with a consideration to equally distribute questions in terms of specification such as difficulty and total length of a test form. Test assembly is one of the combinatorial problems which create the combination of the question into a test form. In addition, the assembled test form must be satisfied the constraints and objective function. The constraint is a condition that must be passed before the objective function is called. For example, the common constraint of the test form is a test length. We can define it as follow:

\[
\min \leq \sum_{i=0}^{N} x_i \leq \max
\]

Where, min and max are the range of the number questions in the test form. \(N\) is the number of the question in the question pool. \(x_i \in \{0,1\}\) is the question \(i\) in the question pool. If \(x_i = 1\) represents the question \(i\) is assigned to the test form. Otherwise, \(x_i\) not exists in the test form.

The objective function is a function that measures the quality of the test form. We can formulate objective function as follow:


\[
\begin{align*}
\text{minimize } & F(X) \\
\text{subject to } & \sum_{i=0}^{N} x_i = 10 ; \quad x_i \in \{0,1\}
\end{align*}
\]

Where, the function \( F(X) \) is the objective function. \( X \) is a parameter that represents the test form such as the test information.

So, we can add many constraints and objective function to the test assembly. However, with many constraints, it will eventually increase a computational cost exponentially. This is a trade-off between quality of the test form and computational cost. Moreover, test assembly is not guaranteed the maximum number of test forms. So, the optimum search approach is generally exploited to solve the problem.

The test assembly is NP-hard problem; thus, there are many optimization algorithms to solve the combinatorial problem such as Particle Swarm Optimization (PSO) (12), Ant Colony System (ANTS) (13), Genetic Algorithm (14) and Bee Algorithm (15). Yang (16) reported that Bee Algorithm was the best since it had the lowest computational cost. Moreover, Pham (17) showed that the bee algorithm could solve a complex problem with a good accuracy. Hence, this thesis proposes the automated test assembly with minimum redundant questions based on Bee algorithm.

2.1.1 Bee Algorithm

The Bee Algorithm is an optimization algorithm developed in 2005 (15). The algorithm simulates from foraging activities of honey bees. Foraging activities of the honey bee divide the honey into two groups; scout bee and follower bee. First, the scout bee goes to search for the honey randomly. When scout bees found the honey source, they pass the information to the follower bees in the hive. The follower bees follow to the location of honey that receives from scout bee. Then, follower bees apply a neighborhood search around the location of honey. If they found a new honey source, they pass the information to the other follower bee in the hive.
Accordingly, the Bee Algorithm (BA) also divides bees into two groups. Scout bees randomly create solutions and evaluate the quality of the solutions based on objective function. Then, assign the follower bee solution based on the quality of the previous solutions. Then, apply neighborhood search to the follower bee solution and append a new solution from neighborhood search to the previous solutions. Finally, reassign the follower bee solution until the criterion is met.

Bee Algorithm steps are given below:

1. Initial the solution of scout bees
2. Evaluate the quality of solutions.
3. Continue below steps until the stopping criterion is met
   a. Assign the solution to the follower bees based on the quality of the solution.
   b. Apply the neighborhood search for a new solution
   c. Evaluate the quality of the new solution

2.2 Redundant Question

Redundant question is a question that similar to another question. From observation, we found two types of redundant question including questions with similar pattern and questions with similar concept.

The similar question is commonly found in the test form, especially the ones from automated test assembly since the existing assembly method does not concern on the redundancy issue. The most generally found redundant question is the similar pattern questions in the same test form. The questions contain the same pattern of question text but differentiate by keywords or domains. If the test form has many similar questions, the test-takers could become boring or annoyed from repeatedly same textual expression. In addition, the test may fail to truly evaluate test-takers since this will lead to unfairness and asymmetric difficulty distribution.
The example of the similar questions in the generated test form is given in Figure 2.1:

**Q64.** Which of the following is an appropriate explanation of BPO?

- a) Contracting out all business processes of a company’s specific department, such as an administrative department or call center, to an external service provider, together with the operations of a business system
- b) Operating a system by renting part of the processing power and memory capacity of the server owned by a telecom carrier, etc., instead of taking possession of a company’s own server
- c) Reducing the cost of software development by using a temporary staff agency’s worker whose labor cost is lower than that of a company’s regular employee
- d) Using software functions offered by an external service provider via a network, instead of taking possession of company’s own software

**Q76.** Which of the following is an appropriate explanation of a divisional organization?

- a) A share of profit responsibility is assigned to each of the organizational units formed by product, region, and so on.
- b) Business operations are classified by function, and subordinates are given orders and instructions on the individual functions.
- c) The needs for the business environment are met by actively utilizing external management resources through strategic alliance, joint development, and so on.
- d) The organization is formed for a certain limited period of time to solve a problem and dissolved when the problem is solved.

**Figure 2.1 Example of similar pattern question**

In Figure 2.1, the question Q64 and Q76 similarly ask about the explanation but in a different item subject. The similar questions are commonly found in the test form. In fact, they are allowed to contain similar questions in the test form but the less is the better.

Second, the similar questions in terms of concept are questions that are comparable to one another. The comparable questions mostly are paraphrased...
questions of the same conceptual reference. The paraphrase questions may occur in a question pool, but they are limited to one question of the concept in the test form. The example of the paraphrase question is given in Figure 2.2.

**Q33.** Which of the following is the system configuration with the highest availability? Here, multiple systems connected in parallel are considered operating if at least one system is operating.

a) Four identical systems with each availability of 70% in parallel connection  
b) One system with an availability of 99%  
c) Three identical systems with each availability of 80% in parallel connection  
d) Two identical systems with each availability of 90% in parallel connection

**Q33.** Which of the following is the system configuration with the highest availability? Here, each unit shown by a box has the same availability (less than 1), and multiple units connected in parallel are considered operating if at least one unit is operating.

![Diagram of system configurations](image)

**Figure 2.2 Example of paraphrase question**

In Figure 2.2, both Q33s semantically are the same question with difference of textual expression and image expression. In details, the choices of the second Q33 describe the detail of the figure that shows in the question.

### 2.3 Text Similarity

Text similarity is a task to measure the similarity of two texts. Gomaa and Fahmy mentioned that there are many techniques proposed to measure the text
similarity. Generally, two approaches are suggested; term-based approach and knowledge-based approach.

The term-based approach measures an overlapping term between two texts. The popular method of term-based approach is Euclidean distance and Cosine similarity. To apply the term-based approach, the text needs to transform into the vector using Term-Frequency – Inverse Document Frequency (TF-IDF) (18). The Euclidean distance measures the distance between two vectors. While the cosine similarity measures the angle cosine between two vectors.

The knowledge-based approach measures a similarity by using information from the semantic network such as WordNet. There are many techniques using the knowledge-based approach such as Path similarity and Wu and Palmer similarity. The path similarity is calculated using the inverse of shortest path length from the two given texts. The path length is an edge that connects between two words with hypernym, hyponym or synonym. The Wu and Palmer similarity calculates from the depth of least common subsumer and shortest path of two given texts.

2.3.1 TF-IDF

TF-IDF is a weighted score of text which calculates from two factors TF and IDF that form as $TFIDF = TF \times IDF$. The $TF$ is a frequency of the term that occurs in the document that represents as $TF = \{t_1, t_2, t_3, ..., t_n\}$ where $t_i$ is a frequency of term $i$. The IDF is calculated from the equation below.

$$idf_i = \log_2 \frac{N}{df_i} \tag{2.4}$$

Where $df_i$ is a frequency document that occurs term $i$. $N$ is a number of a document.
2.3.2 Cosine Similarity

Cosine similarity is a technique to measure the similarity of two vectors. The score of cosine similarity calculates from an angle between two vectors which has range [0, 1]. The higher score means two vectors are similar. The cosine similarity can calculate as following equation:

\[
\text{sim}_\text{cosine}(A,B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.5}
\]

Where \(A\) and \(B\) are a vector. \(\|A\|\) and \(\|B\|\) are a length of the vector. \(A_i\) and \(B_i\) are an element of \(A\) and \(B\) respectively.

2.3.3 Text Semantic Similarity (TSS)

There are many techniques to measure similarity of word semantic using knowledge-based such as path similarity and Wu and Palmer similarity. These techniques use path or relation given in WordNet to calculate the word to word semantic similarity score. The path similarity is calculated using the inverse of shortest path length from the two given texts. The Wu and Palmer similarity calculates from the depth of least common subsumer and shortest path of two given texts.

To apply sentence with word semantic similarity, Mihalcea et al. (19) used the TSS function to measure the semantic similarity between two sentences. The TSS function is defined as follow.

\[
\text{sim}_\text{TSS}(T_1, T_2) = \frac{1}{2} \left( \frac{\sum_{w \in T_1} \text{maxsim}(w, T_2) \cdot \text{idf}_w}{\sum_{w \in T_1} \text{idf}_w} + \frac{\sum_{w \in T_2} \text{maxsim}(w, T_1) \cdot \text{idf}_w}{\sum_{w \in T_2} \text{idf}_w} \right) \tag{2.6}
\]
Where $T_1$ and $T_2$ is a text input. $\text{maxsim}(w, T_n)$ is a max similarity between $w$ and text input $n$. The similarity of $\text{maxsim}(w, T_n)$ is semantic similarity concept such as path similarity, or Wu and Palmer similarity.

2.4 Related Research

Angela J. Verschoor (4) proposed a genetic algorithm (GA) for test assembly. He shows that GA can apply for many problems in test assembly. The example of the GA model is applied for the basic model of test assembly that minimizes the maximum ratio of test information function and the target function. The result shows that the solution of GA is acceptable and nearly to the commercial solver solution.

Songmuang P. and Maomi Ueno (5) proposed a Bees algorithm (BA) for multiple test assembly that applies a parallel-computing technique that distributes the computational costs to multiple processors without increasing the differences of quality. The experiments use simulated question pool in the first experiment and actual question pool from the Japan Information Technology Engineers’ Examination and compare with four models, Big Shadow Test (BST), GA₁, GA₂, and BA. The result show that required lower computational time than the other methods while the differences in fitting errors for the constructed test forms is lower or close to that of the other methods.

Gomaa and Fahmy (20) survey the text similarity approach which separates into a string-based approach, corpus-based approach, and knowledge-based approach. The string-based approach operates on string sequences and character composition. The survey shows that string based consists of fourteen algorithms. Seven of them are character based (longest common substring, Dameran-Levenshtein, Jaro, Jaro-Winkler, Needleman-Wunsch, Smith-Waterman, N-gram) and another seven algorithms are term based (Cosine similarity, Dice’s coefficient, Euclidean distance, Jaccard similarity, Matching Coefficient, Overlap). The corpus-based approach is semantic similarity that trained from a large corpus. There are nine algorithms for corpus-based approach, Hyperspace Analogue to Language (HAL),
Latent Semantic Analysis (LSA), Generalized Latent Semantic Analysis (GLSA), Explicit Semantic Analysis (ESA), The cross-language explicit semantic analysis (CL-ESA), Pointwise Mutual Information – Information Retrieval (PMI-IR), Second-order co-occurrence pointwise mutual information (SCO-PMI), Normalized Google Distance (NGD), and Extracting DIStributionally similar words using CO-occurrences (DISCO). The knowledge-based approach measures the similarity between two words using the semantic network such as WordNet. This approach can be divide into two groups, semantic and relatedness. There are six methods for the semantic similarity. Three of them are based on information content (Resnik (res), Lin, Jiang & Conrath (jcn)) and another three methods are based on path length (Leacock & Chodorow (lch), Wu & Palmer (wup), Path similarity (path)). Finally, the relatedness similarity consists of three methods: Hirst and St-Onge’ (HSO), Lesk, and vector pair.

Mihalcea et al. (19) proposed the method to measure the semantic similarity of texts. They use the TSS function to score between two texts. The similarity function that uses in the TSS function can divide into two approaches; corpus-based approach and knowledge-based approach. They use TSS to automatically identify the paraphrase in Microsoft paraphrase corpus. The similarity functions that use to compare the result are PMI-IR, LSA, J&L&C, Lesk, Lin, W&P and Renik. The result is compared with the accuracy, precision, recall, and F-measure. The accuracies are around 69 – 70 percent. And the F-Measure is around 79 – 81 percent.

Ferlež and Gams (21) analyze the semantic measurement of two words with a benchmark set of human similarity judgment based on WordNet 2.0. They compare 8 algorithms, shortest path similarity (SP), Hirst and St-Onge’s semantic relatedness (HSO), Leacock and Chodorow’s semantic similarity (LCH), Resnik’s similarity (RES), Jiang and Conrath’s semantic distance (JCN), Lin’s semantic similarity (LIN), Roget’s Taxonomy Shortest-path semantic distance (RSTP) and Latent Semantic Analysis (LSA). The result shows the comparison that SP has the better result than another method.
CHAPTER 3
RESEARCH METHODOLOGY

The objective of this thesis is to minimize the redundant questions in test form which is defined in chapter 2. The model should minimize the redundant question with two types: 1) same pattern question and 2) paraphrase question. In this section, we will describe the objective function that uses in Bee algorithm. Then, we will describe the similarity function that uses in the experiments. Next, the Bee algorithm for automated test assembly is explained with the proposed method to manage the redundant question. Then, the improved neighborhood search for minimizing redundant question is described. Furthermore, the dataset used in this thesis is clarified. Last, the experiment setting for the test assembly is explained.

3.1 Objective Function

To apply the Bee Algorithm for the automated test assembly, we need to define the objective function for the test form. The main objective of this thesis is to minimize the redundant question in the test form. The solution of the bee algorithm is the test form that contains a low number of similar questions. So, we use the text similarity to measure the redundant questions including cosine similarity and text semantic similarity. As such, if the similarity between two questions is high, the question is a redundant question. So, the objective function to minimize the similarity of all questions can formulate as follow:

$$\text{fitness} = F(Q) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{sim}(Q_i, Q_j)$$  \hspace{1cm} (3.1)

Where, fitness is a result of the objective function. $F(Q)$ is an objective function, $Q$ is the set of question $\{Q_1,\ldots, Q_N\}$ in the test form. $N$ is a test length. $Q$ is a set of question in the test form, which $Q_i$ is a question $i$ in the test form. $\text{sim}(Q_i, Q_j)$ is...
a similarity function that measures the similarity of $Q_i$ and $Q_j$. The $\text{sim}(Q_i, Q_j)$ can replace with cosine similarity and TSS.

### 3.2 Similarity function

There are many techniques for text similarity which can divide into many approaches; term-based approach and knowledge-based approach. We select one technique from each approach to apply in the objective function.

The first technique is the cosine similarity which is the most popular in text similarity. The score of cosine similarity calculates from an angle between two vectors which has in range \( [0, 1] \). The higher score means two vectors are similar. However, the score equal to 1 doesn’t mean two vectors are always identical. It is flexible to find the pattern between texts. So, we use this advantage of cosine similarity to find the similarity pattern between two questions.

The second technique is text semantic similarity that described in section 2.1.3. The paraphrase question is harder to detect because the words or sentence in the question is changed from the original sentence. To detect the paraphrase question, we need to measure the semantic similarity between the questions. So, we use the text semantic similarity function in equation 2.6 to measure the semantic of two texts by applying the path similarity as a similarity function.

Last, we are modified the text semantic similarity function in equation 2.6 to detect both the similarity pattern question and the paraphrase question. In more detail, The IDF of the word is the weight that represents how important of the word. The equation 2.6 is using IDF to measure the similarity of the texts that based on the keyword. However, the similarity of question is not only based on keyword. Therefore, we remove the IDF factor to reduce the difference between two texts that are using the different wording. So, the modified text semantic similarity equation is calculated by following equation.
\[ \text{sim}_{TSS2}(T_1, T_2) = \frac{1}{2} \left( \frac{m}{\sum_{w \in T_1} \maxsim(w, T_2)} + \frac{n}{\sum_{w \in T_2} \maxsim(w, T_1)} \right) \]  

(2.7)

Where \( m \) is a number of words in the text \( T_1 \), \( n \) is a number of words in text \( T_2 \).

### 3.3 Bee Algorithm for Automated Test Assembly

To assemble the test form, let choose the similarity function for the objective function. Then initial the solution of scout bees. The solution is a set of questions id \( Q = \{Q_0, \ldots, Q_n\} \), where \( Q_i \) is the question id that presents in the test form. Each scout bees randomly create the solution which matches the test constraints. Then, they evaluate the quality of solutions. In this step, calculation of fitness value of all solution is performed.

Second, it assigns the solution to each follower bee based on the fitness value of the solution. The probability to choose by follower bee can be calculated by the following equation.

\[ p_x = \frac{1}{\sum_{y=1}^{N} \frac{1}{f_y}} \quad ; x \in S \]  

(3.2)

Where, \( S \) is a set of all solution. \( p_x \) is a probability of solution \( x \). \( f_x \) is a fitness value of the solution \( x \). \( f_y \) is a fitness value of solution \( y \) and \( N \) is a number of solution.

Third, each follower bee applies the neighborhood search for a new solution. Some of the question in the solution is chosen to remove and replace with a new question into the solution.

Finally, the method is to redo the step 2 and step 3 until the criterion is met.
3.4 Improved Neighborhood Search for Minimizing Redundant Question

While follower bee performs a neighborhood search, follower bee selects some of the questions to remove and replaces new questions into the solution. Selection and replacement is effect with how fast to optimize the solution. So, we introduce the function that can detect the redundant question called Max-Pair-Similarity (MPS).

3.4.1 Max-Pair-Similarity (MPS)

The concept of Max-Pair-Similarity (MPS) is to measure the similarity of the question $Q_i$ to entire of the solution $Q$. The higher similarity question should be removed from the solution. The MPS calculates by summation of similarity of given $Q_i$ to the all of other question in $Q$ as the following equation.

$$MPS_{Q_i} = \maxsim(Q_i, Q_j); \quad i \neq j, Q_j \in Q$$

$$\forall \quad i = 1, \ldots, N$$

Where $MPS_{Q_i}$ is the maximum pair similarity score between question $Q_i$ and other $Q_j$ solution $Q$.

3.4.2 Selection

To select the question in neighborhood search step, we need to detect the redundant question in the test form. The redundant question measures by using the score of MPS that define in equation 3.3.

$$p_{Q_i} = \frac{MPS_{Q_i}}{\sum_{j=1}^{N} MPS_{Q_j}} ; \quad Q_i \in Q , i \in \{1, \ldots, N\}$$

Where $p_{Q_i}$ is a probability to select $Q_i$. $MPS_{Q_i}$ is a Max-Pair-Similarity of the question $Q$, $N$ is a test length.
3.4.3 Replacement

After the selection step in section 3.4.2, we need to replace the new question into the solution. Therefore, the question is randomly selected from the item pool. The selected question requires lower MPS score than the remove question. Otherwise, The new question is randomly selected.

3.5 Question pool

The question pool in this study is the dataset from the Information Technology Professional Examination (ITPE) (11). The website provides preparation test data in two tracks, Fundamental Information Technology Professional Examination (FE) and Software Design & Development Professional Examination (SW). We collect the data from FE track in the year 2007-2011. The type of the question in the dataset is a multiple choices. The language of the question is written in English. The dataset consists of 400 questions. The redundant question is manually tagged. There are 3,755 redundant question pairs that consist of 3,749 similar pattern question pairs, and 6 paraphrase question pairs.

In more detail, each question has variously content such as question body, choices, table and, images. We are excludes the content of the image in the question. We analyze only the text in the question. The similar pattern question has increase similarity pattern when the question includes the choices. Therefore, the question in this study means the whole text in the question except the image.

3.6 Experiments

3.6.1 Experiment Setting 1

In this experiment, we use the real question pool that described in section 3.5. The question pool consists of 400 questions. There are 3,755 redundant question pairs that consist of 3,749 similar pattern question pairs and, 6 paraphrase question pairs.
This experiment evaluates the performance of the model to minimize the redundant question which compares three models, Bee algorithm using cosine similarity (BA-cosine), Bee algorithm using TSS (BA-TSS) and, Bee algorithm using TSS2 (BA-TSS2). The test forms of each model are generated 100 times. The parameter of Bee algorithm is set as follow, number of bee = 20, number of iteration = 100, Test length = 50 question.

We evaluate the performance of the model by counting the redundant question pairs that found in the test form. The experiment result is summarized by counting the average number of redundant question in each iteration and the experiment result is summarized in each type of redundant question.

3.6.2 Experiment Setting 2

In the previous experiment, the question pool has low paraphrase question pair. Therefore, This experiment will simulate the paraphrase question pair by using paraphrase sentence in the Microsoft research paraphrase corpus (22).

This experiment evaluates the performance of the model to minimize the redundant question which compares three models, Bee algorithm using cosine similarity (BA-cosine), Bee algorithm using TSS (BA-TSS) and, Bee algorithm using TSS2 (BA-TSS2). The test forms of each model are generated 100 times. The parameter of Bee algorithm is set as follow, number of bee = 20, number of iteration = 100, Test length = 50 question.

We evaluate the performance of the model by counting the redundant question pairs that found in the test form. The experiment result is summarized by counting the average number of redundant question in each iteration and the experiment result is summarized in each type of redundant question.
CHAPTER 4
RESULTS AND DISCUSSION

This section shows the experiment results that compare the performance of the bee algorithm to minimize the redundant question. Then, we discuss the result of the experiment.

4.1 Result

4.1.1 The experiment result 1

According to the experiment setting 3.6.1, the experiment result is shown the average number of the redundant question pair in each iteration of the bee algorithm. The result is divided into two groups, similar pattern question and, paraphrase question. The average number of similar pattern question in each iteration of the bee algorithm is shown as follows:

Figure 4.1 The result of experiment 1: Average number of similar pattern question pairs in each iteration of BA
Figure 4.1 demonstrates the result between the average number of similar pattern question pairs and the number of iteration in BA. The result shows the comparison of the three models, BA-cosine, BA-TSS and, BA-TSS2. The result shows that BA-cosine is the best model to reduce the similar pattern question. While the similar pattern question of BA-TSS and the similar pattern question of BA-TSS2 reduce slower than BA-cosine. The BA-TSS and BA-TSS2 cannot solve the similar pattern question because of the difference of word topic in the question. According to the result, the number of similar pattern question pairs of BA-cosine, BA-TSS and, BA-TSS2 are 8.49, 18.73 and, 20.43 pairs respectively.

Second, the average number of paraphrase question in each iteration of bee algorithm is shown as follows:

Figure 4.2 The result of experiment 1: Average number of paraphrase question pairs in each iteration of BA

Figure 4.2 shows the average number of paraphrase question pairs and the number of iteration in BA. The result shows the comparison of the three models, BA-cosine, BA-TSS and, BA-TSS2. The result shows that all models can
minimize the paraphrase question pair to zero. But speed to minimize paraphrase of all models is not equivalent. As shown in Figure 4.2, the BA-TSS is the fastest model to minimize the paraphrase question pair to zero. However, the number of paraphrase question is quite low in the question pool. The performance of each model does not guarantee the accuracy. So, we construct a new question pool and we will describe the detail in the next experiment.

4.1.2 The experiment result 2

This experiment uses a new question pool to evaluate the performance to minimize the paraphrase question. The number of question in the new question pool is 600 questions. The paraphrase question is collected from Microsoft research paraphrase corpus. Total paraphrase question pair in the new question pool is 100 pairs. The experiment setting is the same as the previous experiment. The experiment result is shown as follows:

![Graph showing the result of experiment 2](image)

**Figure 4.3** The result of experiment 2: Average number of similar pattern question pairs in each iteration of BA
Figure 4.3 illustrates the result between the average number of similar pattern question pairs and the number of iteration in BA. The result shows the comparison of the three models, BA-cosine, BA-TSS and, BA-TSS2. The result same as the previous experiment result. BA-cosine is the best method to minimize the similar pattern question. According to the result, the number of similar pattern question pairs of BA-cosine, BA-TSS and, BA-TSS2 are 0.78, 2.89 and, 2.82 respectively.

Figure 4.4 shows the average number of paraphrase question pairs and the number of iteration in BA. The result shows the comparison of the three models, BA-cosine, BA-TSS and, BA-TSS2. The result shows that BA-cosine and BA-TSS2 can minimize the paraphrase question pair to zero. But BA-TSS stuck around 0.15 to 0.2. According to the result, BA-TSS is inconsistency to the previous experiment result. On the other hand, the BA-cosine the best model to minimize the paraphrase question. Therefore, this issue will discuss in the next section.
4.2 Discussion

In this study, we proposed the technique to minimize the redundant question in the test form. We try to minimize the redundant question in the test form by using the text similarity to detect the redundant question.

We use the question pool from ITPE test since 2007-2011. The number of the redundant question pair in the question pool is around 5 percent. It looks like the redundant question pair is too low. But the number of the redundant question in the test form is increasing exponentially if the test length is increased. So, we summarize the experiment result to show that we can minimize the redundant question in the test form as follows:

Table 4.1 The average number of similar pattern question pairs with different test length on the new question pool

<table>
<thead>
<tr>
<th>Model</th>
<th>10 Questions n (%)</th>
<th>20 Questions n (%)</th>
<th>30 Questions n (%)</th>
<th>40 Questions n (%)</th>
<th>50 Questions n (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.27 (-0%)</td>
<td>2.18 (-0%)</td>
<td>4.68 (-0%)</td>
<td>9.32 (-0%)</td>
<td>15.58 (-0%)</td>
</tr>
<tr>
<td>BA-cosine</td>
<td>0.00 (-100%)</td>
<td>0.02 (-99.08%)</td>
<td>0.03 (-99.36%)</td>
<td>0.43 (-95.39%)</td>
<td>0.78 (-94.99%)</td>
</tr>
<tr>
<td>BA-TSS</td>
<td>0.01 (-96.3%)</td>
<td>0.11 (-94.95%)</td>
<td>0.49 (-89.53%)</td>
<td>2.05 (-78.00%)</td>
<td>2.89 (-81.45%)</td>
</tr>
<tr>
<td>BA-TSS2</td>
<td>0.00 (-100%)</td>
<td>0.10 (-95.41%)</td>
<td>0.43 (-90.81%)</td>
<td>1.10 (-88.20%)</td>
<td>2.82 (-81.90%)</td>
</tr>
</tbody>
</table>

Table 4.2 The average number of paraphrase question pairs with different test length on the new question pool

<table>
<thead>
<tr>
<th>Model</th>
<th>10 Questions n (%)</th>
<th>20 Questions n (%)</th>
<th>30 Questions n (%)</th>
<th>40 Questions n (%)</th>
<th>50 Questions n (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.02 (-0%)</td>
<td>0.14 (-0%)</td>
<td>0.39 (-0%)</td>
<td>0.5 (-0%)</td>
<td>0.99 (-0%)</td>
</tr>
<tr>
<td>BA-cosine</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
</tr>
<tr>
<td>BA-TSS</td>
<td>0.0 (-100%)</td>
<td>0.02 (-85.71%)</td>
<td>0.06 (-84.62%)</td>
<td>0.17 (-66%)</td>
<td>0.15 (-84.84%)</td>
</tr>
<tr>
<td>BA-TSS2</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
<td>0.0 (-100%)</td>
</tr>
</tbody>
</table>
Table 4.1 and Table 4.2 show the average number of redundant question in a different test length. Each cell contains two numbers. The number on the left shows the average number of redundant question pair in the test form. The number in parenthesis is a percent that reduces from the baseline. The baseline is an average number of the redundant question when randomly initiate the test form. The result shows that all of the models can reduce both the similar pattern question pair and the paraphrase question pair.

In the first experiment, the result shows that the BA-cosine is the best model to minimize the similar pattern question pair. While the paraphrase question pair in the question pool is too low. The average number of the paraphrase question pair start at around 0.18 pair. All of the models can minimize the redundant question to zero. We cannot compare the performance of the model to minimize the paraphrase question. Hence, we are mock up the paraphrase question into the question pool and make a new experiment.

In the second experiment, we add the new paraphrase question into the question pool. The result shows that the BA-cosine is the best model to minimize the similar pattern question pair. But the paraphrase question is not the same as the first experiment. BA-cosine is better than BA-TSS and BA-TSS2. So, we found that the new paraphrase question pair of the BA-TSS is not as expected.

The first problem is the new paraphrase question is not in the same domain as ITPE questions. Some of the words in the new question is not occur frequently. Therefore, the TSS score of the paraphrase question is lower than the TSS score of normal question pair. So, BA-TSS fails to remove the redundant question in the test form. For example, the paraphrase question pair is shown as follows
Figure 4.5 The example of the paraphrase question that BA-TSS is failed to remove.

Figure 4.5 shows the example of the paraphrase question pair. The score of cosine similarity, TSS and, TSS2 are 0.511566346, 0.390384658 and, 0.501438492 respectively. The similarity score of TSS is close to the similarity score of the normal question pair which the average score is 0.24083512. While the other similarity functions have more difference score between paraphrase question pair and normal question pair.

Second, TSS raises some score of normal question pair. Some of normal question pairs are not relevant but the semantic of TSS make the score increase. The example of the normal pair is shown as follows.

| id=2494381 | The FTC also asked the judge to suspend his ruling pending its appeal. |
| id=2494531 | The FTC asked the court Wednesday to block the ruling and filed for an appeal. |

Q42. In object orientation, the term “open (white box) reuse” refers to the reusing of base-class data and functions by creating subclasses for the base class. Which of the following is the appropriate description concerning the reuse technology in object orientation of this method?

a) Changes in the base class do not affect the subclasses.

b) Only differences between the data and functions defined by the base class and those of subclasses can be stated in subclasses, so development is highly efficient.

c) Since the data defined in the base class is protected, programs with a high degree of safety can be developed.

d) The base class can be used to develop multiple applications, but its subclasses cannot be re-used.

Figure 4.6 The example of the normal question that has TSS score more than some paraphrase question
**Q64.** Which of the following is an appropriate description concerning the duties of database administrators?

a) To allocate available development staff and resources in an optimal way for the system development using databases, and then manage them so that the system can be developed in an efficient way  
b) To conduct user interviews and create applications based on the external schema of the provided database  
c) To design and maintain databases, monitor their operations, and recover from failures  
d) To perform acceptance inspections to determine whether or not applications satisfy the functions, performance, operability, and other specifications required by users

---

**Figure 4.7** The example of the normal question that has TSS score more than some paraphrase question (Cont.)

Figure 4.6 and Figure 4.7 show the example of the normal question pair that not relevant but the similarity score is higher than paraphrase question from above example. The score of cosine, TSS, TSS2 are 0.160925747, 0.445504516 and, 0.505548838 respectively. In this case, BA-TSS may remove the normal question from the test form before the paraphrase question.

In this study, TSS2 is better to minimize the paraphrase question than TSS function because the TSS2 is not weighted with the IDF. The common word in the question has more effect to the similarity score. If the question is changed some of the words, the similarity changes a little bit. But the best similarity function is cosine similarity because the redundant question wording changes a little bit. There are many unchangeable technical terms. So, the word overlapping is enough to reduce the redundant question in the test form.
CHAPTER 5
CONCLUSION

The test assembly is one of the reliable methods to automatically construct a test form from a pool of questions. The method has been researched in many works (1–5) and proved to be reliable and practical. However, the issue in the generated test form has still remained. The issue is that there is redundancy in the questions. The redundant questions are in a form of similar questions in the same test form. This issue leads to the unfairness to the other test form.

Our main objective is to minimize the redundant question in the test form. The redundant question that should minimize is defined as, 1) Similar pattern question; the question that has a similar to another question. The question is quite same but not in the same domain. 2) Paraphrase question; the question that has the same meaning to another question but not uses the same wording.

The test assembly is NP-hard problem; thus, there are many optimization algorithms to solve the combinatorial problem. Several researches showed that the Bee algorithm could solve a complex problem with a good accuracy and lowest computational cost. Hence, this thesis proposes the automated test assembly with minimum redundant questions based on Bee algorithm.

To minimize the redundant question in the test form, we use the text similarity technique to solve the redundant question issue. So, we defined the objective function to measure a quality of the test form in section 3.1. Hence, we improve the neighborhood search to minimize the redundant question. The technique is called MPS. MPS is a score of question that indicates to the redundant question. Finally, we develop the Bee algorithm for automated test assembly with minimize the redundant question.

According to the experimental results in section 4.1, the results show that we can reduce both similar pattern question and paraphrase question. The best model to minimize the redundant question is BA-cosine. The BA-TSS2 is slower than BA-cosine.
In the real question pool, the question in the item pool come from many sources. The redundant question may occur in the question pool. If we find the redundant question manually, it needs long time to find all redundant questions. So, this thesis proposes the technique to minimize the redundant question automatically. This technique helps the test administrator to reduce the redundant question in the test form. The experiment results show that we can reduce the redundant question in the test form. So, this technique can be applied to another question pool that the question is written in text. But, this technique in this thesis cannot work for some domain such as Math and Art.
REFERENCES


APPENDIX

The question pool in this thesis is summarized as follows:

Table A-1 the question pool from ITPE-FE from 2007-2011

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Question</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Number of similarity pattern question</td>
<td>26</td>
<td>22</td>
<td>20</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>Number of paraphrase question</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Note: The number of redundant question is summarized only the number of question but not show all exist pairs. To make the pair of redundant question, one question may match to many questions in the other year. For example, the number of similar pattern question in the test ITPE 2007 has 26 questions. It doesn't mean the test in ITPE 2007 have 26 similar pattern question pairs. Each question, in 26 questions, can produce many similar pattern question pairs that in the same test and other tests.
BIOGRAPHY

Name Mr. Vorapon Luantangsrisuk
Date of Birth March 25, 1991
Educational Attainment 2012: Bachelor of Science (Computer Science)

Work Position Student

Scholarship 2012 : Scholarship for talent student to study graduate program in Faculty of Science and Technology Thammasat University

Publications