International Journal of Engineering & Technology, 7 (3.33) (2018) 257-262

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Generation System of Multiple Choice Questions Using Idioms with Multiple Component Keywords

Jae-Young Lee[1]*

[1]*Department of Computer Engineering and Smart Computing Lab, Hallym University*
Chuncheon-si, Gangwon-do, 24252, Korea
**Corresponding author E-mail: jylee@hallym.ac.kr*

Abstract

In order to alleviate the burden for the time-consuming and tedious tasks to make multiple choice questions, we proposed the system that generates multiple choice questions from the sentence with multiple component keywords and then relocates the questions selected by an array with random numbers instead of random functions in order to reduce the relocation time, after the system searches for the group of informative sentence with multiple component keywords by using special idioms. In this paper, the idiom is the *CRm* type idiom that has several components at the right side of this idiom including in a main informative sentence. The next sentences consist of other informative sentences including the components keywords. To make multiple choice questions, the system randomly selects an informative sentence including a component keyword and it also converts the informative sentence into a question. The selected component keyword is used as the correct answer and the three other component keywords are used as distractors. To produce many different questions about the same contents with different positions of the question and items, the system uses a random number array to reduce the relocation time.

Keywords: *automatic question generation system; dynamic question generation system; multiple choice questions; natural language processing; question generation; semantic role labels*

1. Introduction

To evaluate the applicants' abilities, there are many questions, such as true-false questions, cloze questions, and multiple choice questions. The tasks to generate questions from a text file are time-consuming and tedious tasks for evaluators who want to check applicants' abilities. In order to alleviate the burden for the time-consuming and tedious tasks, question generation systems have been studied in various kinds of questions, such as true-false questions, cloze questions, multiple choice questions, and so on. When ordinary people is evaluated by simple tests, such as simple questionnaires, the system of generating true-false questions first makes an informative sentence false by replacing key word with antonym. After doing this, the system uses the informative sentence as a true sentence and it also uses the modified sentence as a false sentence [1]. The cloze questions, known as fill-in-the-blank questions, are useful to test the applicant's ability that finds the better word to fit a sentence[2]. The automatic question generating system make cloze question by replacing one of multiple component keywords by blank[3]. On the other hand, a test set of a multiple choice question consists of a question, a correct answer, and some distractors. This type of test has proved to be an efficient tool in order to measure applicants' achievements. It is also widely used for evaluation and diagnostics of applicant's mental. For many tests of medical knowledge, multiple choice items are created using a methodology for developing multiple items based on automatic item generation[4]. The generating multiple choice questions is an approach to automatically generate multiple choice questions from OWL ontologies with potential educational value[5]. To evaluate the understanding of the semantic role labels and named entities in a test, automatic multiple choice question system selects the informative sentence and keyword in the sentence and distractors are chosen based on a similarity measure between sentences[6]. To generate multiple choice questions automatically using a sports domain text as input, the system selects informative sentences by using topic modeling and parse structure similarity which is computed between the parse structure of an input sentence and a set of reference parse structure[7,8].

The various automatic question generation systems will be able to play an import role in learning and evaluating on distance education. On the distance education, it has the advantage that complements space problem and solves educational quality problems that occur in a traditional education. On the other hand, it has the disadvantage that is difficult to keep applicant's evaluation on fair when applicants want to be evaluated their abilities anytime and anywhere. It is the reason why the way to evaluate using questions with the same contents and the positions of answer at the same questions may overestimate the applicant who has a good memory. To overcome the disadvantage, the system offers applicants different question tests with answers to be changed their locations, even when it evaluates them using the same question. Shuffling system and a dynamic relocation system of question tests extract a set of a question, a correct answer, and several distractors from database that has been stored an automatic question generation system in advance and then relocate the questions, a correct answer, and distractors in random order using random functions in the multiple choices questions[9,10]. The system has the advantage that offers different question tests but the disadvantage of the system is that it slows down due to the numerous calls made by random function. Instead of random function, the array which consists of random numbers made by random function in advance is used for reduction of the relocation time. Multiple choice question generation system had used this ready-made array to relocate

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the questions, so it had resulted in that the speed of relocation by using the array is faster than that by using random functions.
In this paper, we proposed the automatic question generation system that reads each line from text, searches for idiom with multiple component keywords in the line, makes next lines including multiple component keywords and keyword as questions and choices. In the next procedure, it selects the set of questions and keywords to make multiple choice questions, and then it relocates the selected questions and choices to generate many different question sets. In addition, it uses an array with random numbers instead of random functions in order to reduce the relocation time.

2. A generation system of multiple choice questions using idioms

There are two systems to generate multiple choice question using idioms. One is the system that automatically generate multiple choice questions from text using idioms and the other is the system that relocate the questions with choices to make various test sets.

2.1. A system of automatic question generation using idioms

In an automatic question generation system using idioms, it consists of clients, a server, and a database. The client is applicant to be evaluated through a client computer and the database stores raw data for making question, complete questions, and so on. The server has an automatic question generation system using some kinds of idioms and a dynamic relocation system with random numbers, as shown in Figure 1.

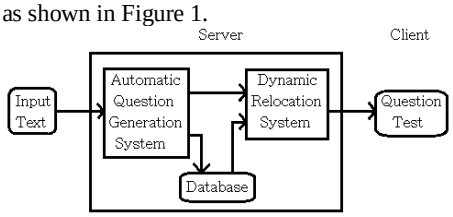

Fig. 1: A system of automatic question generation

An automatic question generation system accepts an input text contents from database or Internet site and then it makes test questions to evaluate each applicant's learning ability or to help each learner to practice. At this time, the system offers the applicants or leaners lots of different question tests that meet some criteria. It consists of automatic question generation system and dynamic relocation system.
The first system reads each sentence from a text input and it converts the sentences into corresponding suitable questions, such as multiple choice questions, true-false questions, cloze questions, and so on. The outputs of the system may be sent to the second system or be saved into database to relocate questions at the second system.
The next system reads outputs of the previous system or database, and selects some questions to meet criteria for both the evaluation of every applicant's ability and the assistance of every learner's practice. And then it relocates the selected questions in random order for the generation of test in different position. When the first system selects some questions, it may or may not select the questions which meet some criteria, such as, the area to evaluate, levels, personal abilities, and so on. When each applicant repeatedly takes the same test in distance education especially, the system has the disadvantage that overestimates the applicant's ability, because the applicant memorizes the position of the same question and the same choices. To overcome the disadvantage, it needs to relocate both the positions of questions and positions of multiple choices in random order.

2.2. Dynamic relocation system of multiple choice questions

In a dynamic relocation system of multiple choice questions which are made using multiple component keywords in an informative, the system also consists of clients, a server, and a database. The server has question element generation system and a dynamic relocation system, as shown in Figure 2.
In this paper, we proposed the system that selects multiple choice questions from the database which store questions generated by the idioms to find multiple component keywords and then relocates the selected questions and choices of their questions in random order. When relocating in addition, it uses a random number array instead of random functions to reduce the relocation time.

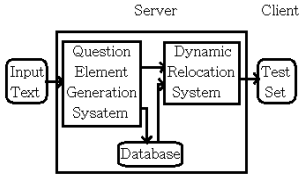

Fig. 2: A dynamic relocation system of multiple choice questions

3. Algorithms to generate multiple choice questions

The algorithms to generate multiple choice questions first needs idioms that are able to find informative sentences from text and then the it is made by using the idioms. There are four types of idioms in the finding informative sentences from text and three algorithms in the procedures of generating multiple choice questions from the informative.

3.1. Classifications of idioms to search for keyword of question generation

In the previous question generation system using idioms, idioms are used to search for keyword in each sentence which accepts from an input text. The sentence including the very important keyword is useful for making various types of questions, such as true-false questions, cloze questions, and multiple choice questions. For example for the simple case, the automatic cloze question system accepts input file from database and then it searches for an informative sentence including the keyword using idioms. Finally, it replaces the keyword in the informative sentence with blank to make a cloze question and then the keyword is used as the correct answer. In the case of multiple choice questions, the additional procedure is to make some distractors with the different meaning of the correction answer.
The idioms used in this system are classified as four types of idioms according to the position of the important keyword, supplementary informative sentences, the number of the multiple informative sentences, and the number of multiple keywords. In the case of multiple sentences, additional sentences are needed when one sentence does not have full meaning about the keyword. The example of the sentence including a keyword only is "This is called volatile memory". In order to explain thoroughly, one or more supplementary sentences should be added to the sentence. Thus supplementary sentences are able to make the perfect question. In the case of the sentence with multiple keywords, the multiple keywords are occurred in the idioms, such as, "consist of", "be composed of", and so on. For example, "A data communications system consists of sender, receiver, transmission line, message, and protocol". The sentence has a main keyword at the left side of the idiom and five component keywords at the right side, where the main keyword can be data communications system and five component keywords can be sender, receiver, transmission line, message, and protocol. It is easy to make distractors using multiple component keywords except for the component keyword,

known as correct answer, in order to make multiple choice questions. When such idioms are classified as four types, the notation ***C*** stands for a current sentence. ***L*** and ***R*** mean that the position of the keyword is at left side and right side of the verb, respectively. And the notation for a numerical number ***m*** is the number of keywords or sentences. The four types of idioms which are represented by the notation are classified as the followings:

(1) ***CL1*** Type: In this of idioms, the leftmost character ***C*** of the type means that the important keyword is included at the current sentence only and it needs neither the previous sentence nor the subsequent sentence to have full meaning of the keyword. On the other hand, ***L*** means that the keyword is located at the left of this type idioms and the number ***1*** also means that the number of keywords is one. The examples of the idioms are "be defined as", "be referred to as", "be used as", "consist of", "be made up of", and so on, where "be" can be one of "am", "are", "is", "was", and "were". In order to process the verb "be", the system needs extra table about be verb in addition to an idiom table. The verb table can have an infinitive verb and irregular verbs.

(2) ***CR1*** Type: The meaning of ***C*** is the same as that of the first type, while the rightmost two characters ***R1*** of the type mean that the keyword is located at the right of the idiom and the number of keyword is one. The examples are "be known as", "called", "be said to be", and so on. In this type, the extra verb table is also needed to process verb "be" beside an idiom table.

(3) ***CR1P1*** Type: The ***CR1P1*** type idiom that has a keyword at the right side of the idiom. The leftmost character ***C*** has the same meaning of the first type. The second character R also represents the right direction of a keyword against the idiom and the third number is the same as the number of keywords at the first type. The rightmost two characters ***P1*** mean that the previous sentences need to be added as the supplementary informative sentences to have full meaning of the keyword, since the current sentence does not have sufficient meaning of the keyword. For example, "This memory is known as auxiliary memory" does not sufficiently offer information of the keyword to us. Using only one sentence is not sufficient to make a perfect question. In order to have the full meaning of the question, more previous sentences should be added at the question. The examples are "be known as", "called" and so on.

(4) ***CRm*** Type: The character ***C*** has the same meaning as the other types. The rightmost two characters ***Rm*** mean that the number of multiple component keywords is m keywords located at the right of the idiom. The examples are "consist of", "be composed of", "be classified as", and so on. Naturally, the keyword at the left side of the idiom is a main keyword. In the case of multiple component keywords, the system needs to have extra process to choose one from multiple component keywords in order to make a correct answer and distractors. The other extra table is also needed to process verb "be". In this type, the system has a big advantage that can produce distractors from these multiple component keywords without external data.

In the above four types, it especially notices that the ***CRm*** type have properties of both the first type ***CL1*** and the last type ***CRm***. In other words, ***CRm*** have both the main keyword at the left side of the idiom and multiple keywords at the right side of the idiom. A group of sentences for this type is suitable for making multiple choice questions since there are many keywords in this type, so it is easy to make distractors using many keywords.

3.2. Algorithm for making multiple choice questions using idioms

The task that makes questions to evaluate learners' abilities is a tedious and time consuming job. In particular, it is more tedious to make more similar questions from limited information. To reduce the burden of the tedious and time consuming job, many similar multiple choice questions can be made by group of informative sentences relative to the fourth type ***CRm*** of idioms, in which the number of their questions increases in proportion to the number of multiple component keywords. For example, a group of informative sentences relative to the type ***CRm*** is "A data communication system consists of message, sender, receiver, transmission media, and protocol. The message is the information to be communicated. The sender is the device that sends the message. The receiver is the device that receives the message. The transmission medium is the physical path by which a message travels from sender to receiver. A protocol is a set of rules that governs data communications." The idiom "consists of" in the first sentence has five component keywords at the right side of the idiom and each sentence for each component follows after this sentence. So the system saves five sentences and their five components into database for making multiple choice questions. In the procedure to make up a multiple choice question, it selects each sentence at the database in ascending order and replaces the subject of the sentence with a question pronoun 'what' in order to make question. And then the subject is used as a correct answer which is located at the first and three of other components are used as distractors, as shown in Figure 3. At the next dynamic relocation procedure, questions are relocated using random numbers and then the correct answer and three distractors are also relocated using different random numbers to make many question test sets with different positions using a dynamic relocation.

No	Multiple Choice Questions
1	What is the information to be communicated? 1) message 2) sender 3) receiver 4) transmission medium
2	What is the device that sends the message? 1) sender 2) receiver 3) transmission medium 4) protocol
3	What is the device that receives the message? 1) receiver 2) transmission medium 3) protocol 4) message
4	What is the physical path by which a message travels from sender to receiver? 1) transmission medium 2) protocol 3) message 4) sender
5	What is a set of rules that governs data communications? 1) protocol 2) message 3) sender 4) receiver

Fig. 3: Five question using a group of five sentences and five keywords

3.3. Algorithm for dynamic relocation system of multiple choice questions with random number array

In the client-server system to generate multiple choice question, many different questions which are made by the same contents are needed to evaluate lots of applicants with questions at the similar level. At first, the system selects as many questions as it needs from a group of questions and keywords. To make the large number of different question sets, the dynamic relocation system at the server randomly relocates a set of the multiple choice questions that are made by the system. The delay in the system is proportion to relocation time which is mainly influenced by the number of calls of random functions when lots of applicants take tests. The way to reduce the relocation time can be achieved by using a random number array instead of the random function. Before the relocation of the multiple choice questions, the system initializes an n x n question array which is filled by n^2 random numbers and a 4 x 5 choice array which is filled by 4^2 random numbers in the case of each question with 4 choices. The last column of the last choice array will be stored the position of a correct answer which points the column with the value 0 after choices are relocated. And then the system gets the random numbers from the array instead of

random function when in needed. Figure 4 shows an example of an initialized 10x10 question array with 100 random numbers.

After selecting a set of multiple choice questions, the system dynamically relocates the questions using the initialized question array which have been made in advance. To relocate the 10 selected multiple choice questions, the system needs to choose a row of the question array using random number. Assuming the 4-th row is selected, the row is a tuple (5,1,4,9,8,7,3,2,6,0), as shown in Figure 5.

1	4	2	3	0	5	8	7	9	6
2	0	1	4	3	8	9	6	5	7
7	5	9	3	1	4	6	8	0	2
5	1	4	9	8	7	3	2	6	0
0	2	4	1	3	8	6	9	7	5
8	6	9	0	2	4	7	5	1	3
0	8	7	6	9	2	3	1	4	5
6	7	5	4	3	0	8	9	2	1
4	3	0	2	1	6	7	5	8	9
5	8	7	1	4	2	9	6	3	0

Fig. 4: An example of an initialized 10x10 question array

4-th row(k=3)

i=0	1	2	3	4	5	6	7	8	9
5	1	4	9	8	7	3	2	6	0

Fig. 5: The 4-th row of the initialized 10x10 question array

Figure 6 shows the procedure that relocates the 10 selected multiple choice questions using the 4-th row of the initialized 10x10 question array. The dynamic relocation system selects ***n*** sets of questions and their choices and saves the selected questions and choices into an input question buffer ***BuffInQ[]*** and an input choice buffer ***BuffInC[],*** respectively. On the other hand, the dynamic relocation system selects each question and a corresponding choice in ascending order. And then it puts the selected question and choices into an output question buffer ***BuffOutQ[]***and an output choice buffer ***BuffOutC[]*** according to the position pointed by the index of the 4-th row of the question array, respectively. For example, the system loads the random number 5 which is equal to *array[0] at index* i=0, and it gets ***MC Question 0*** from the input question buffer ***BuffInQ*** *[0]*. Next, it puts ***MC Question 0*** into the output question buffer ***BuffOutQ*** *[5]*. For i=1, it loads 1 which is equal to *array[1]*. It gets ***MC Question 1*** from ***BuffInQ****[1]* and puts ***MC Question 1*** into ***BuffOutQ****[1]*, and so on.

i	MC Question[i]	j	MC Question[j]
0	MC Question 0	0	MC Question 9
1	MC Question 1	1	MC Question 1
2	MC Question 2	2	MC Question 7
3	MC Question 3	3	MC Question 6
4	MC Question 4	4	MC Question 2
5	MC Question 5	5	MC Question 0
6	MC Question 6	6	MC Question 8
7	MC Question 7	7	MC Question 5
8	MC Question 8	8	MC Question 4
9	MC Question 9	9	MC Question 3

Fig. 6: A relocation of multiple choice questions using the 4-th row of the 10x10 array

Figure 7 shows a random number array with 4x5 in which the random number is initialized in advance except for the last column. The leftmost four columns are random numbers and the last column is the changed position of a correct answer which is equal to the position of the index 0. The numbers to generate using the array randomizes less than ones using random function but the former has an advantage with high speed and is suitable for relocation of questions because applicants must solve these questions within limited time.

To relocate the 4 choices for each question, the system should first choices a row, for example the 3rd row, from the choice. Figure 8 shows the selected row used to relocate four choices using random numbers in it. For example, the relocation system loads random number 2 at index i=0 in the 3rd row and then moves ***Choice[0]*** into the position with index = 2 in a new choice array. And then it loads 3 at index 1 and moves ***Choice[1]*** into the position with index 3 in the new choice array, and so on, as shown in Figure 9. On the other hand, the value 3 at the last index 4 in the row is the answer for this multiple choice question. This answer is moved into a correct answer array for checking applicant's answers. For all of the questions, the output choice buffer can be filled with relocated choices by using both the input answer buffer and the array including their correct answers.

0	2	1	3	1
1	0	3	2	2
2	3	0	1	3
3	1	2	0	4

Fig. 7: An example of an initialized 4x5 choice array

3-rd row(k=2)

i=0	1	2	3	4
2	3	0	1	3

Fig. 8: The 3rd row of the initialized choice array

3-rd row(k=2)

i=0	1	2	3	4
2	3	0	1	3

i	Choice[i]	j	Choice[j]
0	Choice 0	0	Choice 2
1	Choice 1	1	Choice 3
2	Choice 2	2	Choice 0
3	Choice 3	3	Choice 1

Fig. 9: A procedure to relocate four choices using the 3rd row of the 4x5 array

Algorithm for producing the elements of multiple choice questions using idioms with multiple component keywords is the following:

Step 1: Read a line of text from database and find idioms with multiple component keywords.

Step 2: If it is not found, go to **Step 1**.

Step 3: Divide the next sentences including each component keywords into modified questions and keywords.

Step 4: Replace keyword with interrogative pronouns in the sentence to make an interrogative sentence and save the interrogative sentence into the input question buffer ***BuffInQ[]*** of database.

Step 5: If the generation of elements for questions is not finished, go to **Step 1**.

Algorithm for the dynamic relocation of multiple choice questions is the following:

Step 1: In order to relocate questions and choices of each question, initialize an ***n*** **x** ***n*** question array ***A*** with n^2 random numbers and also initialize a ***4x5*** choice array with random numbers and correct answers, in which the random numbers are generated by random function.

Step 2: Extract each set of a question, its correct answer, and multiple distractors from the database randomly up to the predefined number.

Step 3: If it checks the duplicate, then repeat **Step 2**.

Step 4: Save the questions into the input question buffer ***BuffInQ[]*** with one dimension and also save both an answer and multiple distractors into the input choice buffer ***BuffInC[][]*** with two dimensions.

Step 5: Select a row of the question array using a random number as index ***k*** to relocate questions.

Step 6: Select a random numbers ***j*** at index ***i*** = 0, … , (x-1) from the ***k***-th row.

Step 7: Load ***Multiple Choice Question[i]*** from the input question buffer ***BuffInQ[i]*** and load ***Choices[i][y]*** for ***y = 0…4*** from the input choice buffer ***BuffInC[i][y]***.

Step 8: Put ***Question[i]*** into an output question buffer ***BuffOutQ[j]*** and put ***Choice[i][j]*** into an output choice buffer ***BuffOutC[j][y]***.

Step 9: If the relocation of questions is not finished, go to **Step 6**.

Step 10: Repeat the following steps for every question in regular order, ***h*** = 0, … , (x-1).

Step 11: Select a row of the choice array using a random number as index ***y*** to relocate multiple choices including an answer.

Step 12: Select a random numbers ***j*** at index ***i*** = 0, … , (y-1) from the ***k***-th row.

Step 13: Load ***Choices[h][i]*** from the input choice buffer ***BuffInC[][]***.

Step 14: Put ***Choice[h][j]*** into an output choice buffer ***BuffOutC[h][j]***.

Step 15: If the relocation of choices for the h-th question except the answer of the h-th question is not finished, go to **Step 12**.

Step 16: Save the last value ***Choice[h][y]*** into ***CorrectAns[h]*** to score applicants' answers later.

Step 17: If the relocation of choices for all of the questions is not finished, go to **Step 10**.

Step 18: Send the multiple choice question test set for the applicant and save the corresponding answer set to score the applicant's answers.

4. Implementations and discussions

The system for generating multiple questions using idioms with multiple component keywords accepts a set of multiple questions about topic on data communications and then selects some suitable questions. It finally relocates the questions and choices of each question to produce many different test sets for lots of applicants simultaneously. And the system for the dynamic relocation of the questions and the corresponding choices is implemented by a random number array to reduce the relocation time. The system is implemented on environment of JSP, apache server, file system and mysql. For example, the system repeatedly reads a line from the text about data communications and searches idiom with multiple component keywords. Each sentence including component keyword is transformed into question and all component keywords are used as choices for making multiple choice questions into the chapter. Whenever lots of applicants at their client sides click this chapter at a distance, the server randomly selects 10 multiple choice questions and then relocates the selected questions using the random number array instead of random function to make lots of test sets. Figure 10 shows the screen of 10 multiple choice questions before an applicant fills in the blanks. The big advantage of the dynamic relocation system can offer different multiple choice question sets whenever every applicant tries to take a test or refreshes the test for practice.

To compare relocation times of random function and a random number for the multiple choice questions, each relocation time is measured after the relocation of 10 questions and four choices for each question is executed 100times and each result is divided by 100, respectively, because the relocation time is too small. Unit of the time is Nano seconds in the experiments. The relocation time of an array is 0.175 times relocation time of random function for 10 questions and the relocation time of an array is also 0.665 times relocation time of random function for 4 choices of 10 questions,

No	Multiple Choice Questions	Answer
1	What are the devices by which the computer sends out the symbolic results?	
	(1) control unit	
	(2) CPU	☐
	(3) output devices	
	(4) memory	
2	What is a topology in which each device has a dedicated point-to-point connection with only the two devices on either side of it?	
	(1) Ring topology	
	(2) Mesh topology	☐
	(3) Star topology	
	(4) Bus topology	

No	Multiple Choice Questions	Answer
3	What is forun which focuses on long-term research topics related to Internet protocols, applications, architecture, and technology?	
	(1) IAB(Internet Architecture Board)	
	(2) IRTF(Internet Research Task Force)	☐
	(3) IETF(Internet Engineering Task Force)	
	(4) ISOC(Internet Society)	
4	What is the technical advisor to the ISOC?	
	(1) IAB(Internet Architecture Board)	
	(2) ISOC(Internet Society)	☐
	(3) IETF(Internet Engineering Task Force)	
	(4) IRTF(Internet Research Task Force)	

as shown in Table 1. The more the number of objects, the better the improvement of relocation time of the system.

Fig. 10: A procedure to relocate four choices using the 3rd row of the 4x5 array

No	Relocation Time (Random Function)		Relocation Time (Random Number Array)	
	10 Questions	4 Choices	10 Questions	4 Choices
1	2,131	1,756	431	1,319
2	2,035	1,636	293	870
3	2,070	1,621	322	973
4	2,345	1,832	193	568
5	1,724	1,510	375	1,096
6	1,856	1,559	422	1,237
7	2,407	1,835	378	1,128
8	1,976	1,659	351	1,123
9	1,117	900	372	1,143
10	1,319	1,083	184	785
Total	18,980	15,391	3,321	10242
Ratio	1	1	0.175	0.665

Table 1: Comparison of relocation times of random functions and an array

5. Conclusion

We proposed the system that chooses a set of multiple choice questions, and relocates the chosen questions and their choices using a random number array instead of random function to improve the speed of the server, after generating multiple choice questions using idioms with multiple component keywords in an informative sentence that comes from the text on a special topic. In order to evaluate lots of applicants using the important sentence with multiple component keywords online system, the system can fairly help every applicant to be fairly evaluated as they take many different multiple choice questions with the same contents. The process of relocation using an array is faster than that using random function.

Acknowledgement

The research was supported by Hallym University Research Fund, 2017(HRF-201712-007).

References

[1] Lee JY (2016), Dynamic relocation of true-false questions using ready-made arrays with random numbers. *International Journal of Software Engineering and Its Applications*, Vol.10, No.8, 91-100.

[2] Correia R, Baptista J, Eskenazi M & Mamede N (2012), Automatic generation of cloze question systems. *In Computational Proceeding of the Portuguese Language, Springer-Verlag Berlin Heidelberg*, 168-178.

[3] Lee J & Kim B(2017), A dynamic group relocation of the cloze questions with multiple component keywords. *Advanced Science Letters*, Vol.3, No.10, 10284-10288.

[4] Gierl MJ & Lai H (2012), Using automatic item generation to create multiple choice test items. *Medical Education. Blackewll Publishing*, Vol.46, 757-765.

[5] Alsubait T, Parsia B & Sattler U (2014), Generating multiple choice questions from ontologies:lessons learnt. *The Scientific World Journal*, 1-11.

[6] Fattoh IE (2014), Automatic multiple choice question generation system for semantic attributes using string similarity measures. *Computer Engineering and Intelligent Systems*, Vol.5, No.8, 66-73.

[7] Majumder M & Saha SK (2015), A system multiple choice questions: with a novel approach for sentence selection. *Proceedings of the 2nd Workshop on Natural Language Proceeding Techniques for Educational Applications*, 64-72.

[8] Majumder M & Saha SK (2015), Automatic selection of informative sentences: The sentences that can generate multiple choice questions. *Knowledge Management & E-Learning: An International Journal*, 2015;Vol.6, No.4, 377-391.

[9] Dryver A (2009), The Enhancement of Testing Materials for Applied Statstics Courses by Combining Random Number Generation and Portable Document Format Files via LATE. *Journal of Statistical Software*, Vol.31, No.3, 1-9.

[10] Jamail N & Sultan A (2010), Shuffling Algorithms for Automatic Generator Question Paper System. *Computer and Information Science*, Vol.3, No.2, 244-248.