

# A SYSTEM FOR THE AUTOMATIC ASSEMBLY OF TEST QUESTIONS USING A NO-SQL DATABASE

Sanggyu Shin and Hiroshi Hashimoto

*Advanced Institute of Industrial Technology*

*1-10-40, Higashiooi, Shinagawa-ku, Tokyo, 140-0011, Japan*

## ABSTRACT

We describe a system that automatically assembles test questions from a set of examples. Our system can create test questions appropriate for each user's level at low cost. In particular, when a user review their lesson, our system provides new test questions which are assembled based on their previous test results and past mistakes, rather than a completely random selection. By providing test question of the appropriate level, we hope to enhance the user's learning. We use MongoDB which is a no-SQL database system because the contents (test questions) are not structured. Most of test contents used in class are not structured, i.e. each question has a different structure and different selection type. Therefore, we used MongoDB system in order to structure the test questions more freely.

## KEYWORDS

e-learning, personalization, automatically assembling, non-structure.

## 1. INTRODUCTION

In a class such as the language course of a university, the level of individual students will vary, even if classes are divided based on student ability. Moreover, even if students begin a course at the same level, differences in level will emerge as the course proceeds as students progress at different rates. However, until now, it has been common to present the same test questions to every student in the class. Even if, ideally, questions should be matched with each student's level, this has been impractical. Some part of this problem can be addressed with the growth of latest e-learning technology (Xia et al, 2014; Kundisch et al, 2013).

First, by using an e-learning system, each student's test result can be evaluated in real time by using the network. Second, the results for each student can be compared immediately. Third, by comparison of a student's test result with those of other students, the level of the student relative to the class can be assessed. In this way, if a study result is known in real time, it is possible to provide the test questions matched with each student at the next step.

Our system separates the example answers from the question. All example answers are managed in a database, with the relation information relevant to the questions. The sentences used for test questions are reconstructed using many example answers managed in the database. The database also has information about the results of each student for each question and example answers.

## 2. OUR SYSTEM FOR AUTOMATICALLY ASSEMBLING TEST QUESTIONS

The goal of our system is the provision of the test questions matched to the level of each student. After the first test about their lesson, we collect the information about their test result. This information is used to restructure the test questions when the student restudies about their past test or lesson. The construction of a system that allows students to improve their learning by solving specially selected new test questions is the concept of our system.

The common test questions have various forms. Some questions are multiple choice and some questions require a written answer. The various types of questions make it difficult when data is stored in a relational database system. Moreover, since the most test questions were created by a program like a Microsoft Word, they cannot be stored in a database as-is. Data in MongoDB has a flexible schema. Collections do not enforce document structure (Kanade et al. 2014; Banker, 2011). This flexibility allows our system to control the document-oriented data.

### 3. SYSTEM ARCHITECTURE

For many, but not all, test questions, example answers can be separated from the question. We call each example answer or question “fragment data”. We consider those questions from which example answers can be separated and focus on how to store the question and example answers from which new questions related to the original question can be formed. An initial set of questions was compiled by foreign language teachers. These questions were categorized by our system according to whether example answers can be separated or not.

A teacher who wants to use a test question can use the question as it is or a new question can be generated using the stored example answers. Our system supports the reassembly of the example answers with their related question automatically if the question has many example answers that can reassemble.

When our system provides the reassembled test question, our system refers to the percentage of correct answers for each test question and example answers and provides the test questions appropriate for the level of each student based on feedback of the student’s previous results.

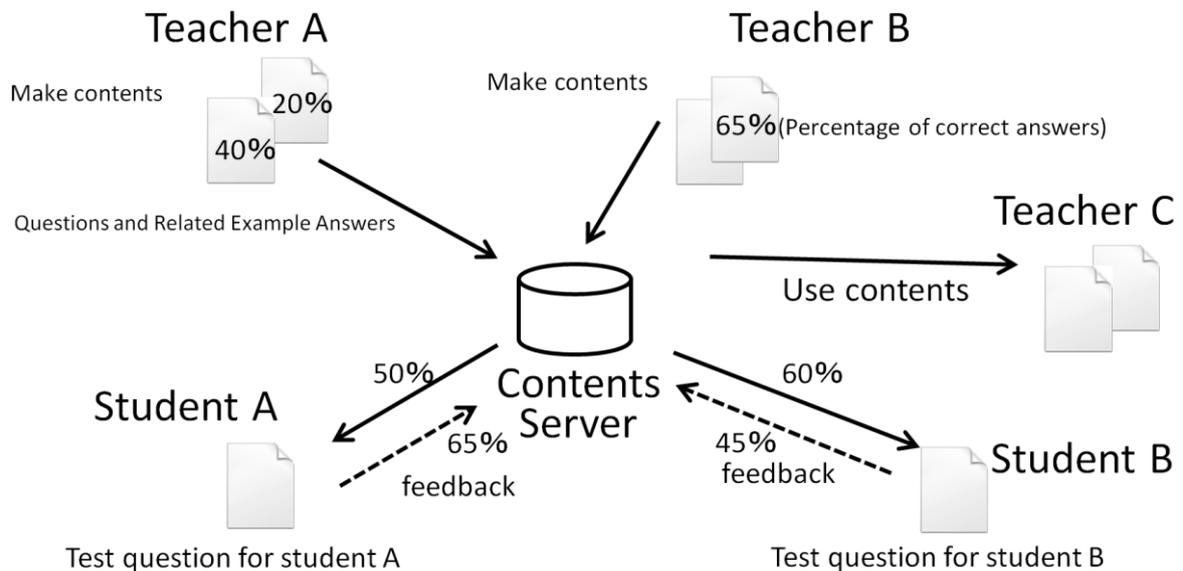


Figure 1. Architecture of System

Figure 2 shows a very simple sample data structure. The provided test question (The data matched to the level of student in Figure 2.) is constructed from the stored example answers (Example answers in Figure 2.). Each example answer has the percentage of correct answers (46%~78%). This percentage is used to generate questions of an appropriate level for each student, based on the student’s previous test results. Of course, this percentage is changed by re-recollection of test results.

Moreover, one test question is stored as one JSON style document, with various data added to each document.

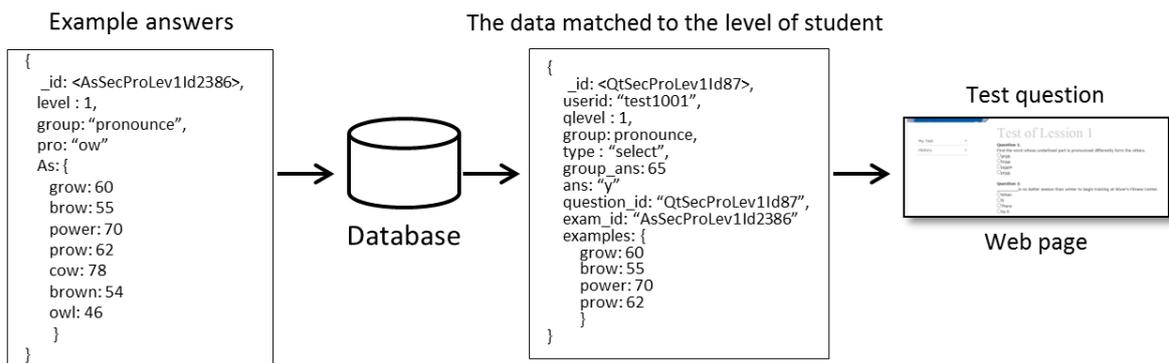


Figure 2. Example of Data Structure

#### 4. CONCLUSION

We proposed a system for automatically assembling test questions. This system can assemble the example answers in the test questions automatically and at low cost. The processing of flexible data became possible by using MongoDB.

#### ACKNOWLEDGMENT

This work was supported in part by JST RISTEX Service Science, Solutions and Foundation Integrated Research Program.

#### REFERENCES

- Xia, H., Zhou, S. and Liu, Y. 2014. Application and evaluation of NoSQL in course group system. *WIT Transactions on Information and Communication Technologies*, Vol. 51, pp.197-203.
- Kundisch, D., Herrmann, P., Whittaker, M., Neumann, J., Magenheimer, J., Reinhardt, W., Beutner, M. and Zoyke, A. 2013. Designing a Web-Based Classroom Response System. *8th International Conference, DESRIST 2013*, pp.425-431.
- Kanade, A., Gopal, A. and Kanade, S. 2014. A study of normalization and embedding in MongoDB. *Advance Computing Conference (IACC), 2014 IEEE International*, pp.416 - 421.
- Banker, K. 2011. *MongoDB in Action*. Manning Publications, U.S.A..