Performance Evaluation of a New Approach for Automatic Question Production

Mile Jovanov and Marjan Gusev

Institute of Informatics, FNSM, Gazi Baba b.b., 1000 Skopje
{mile,marjan}@ii.edu.mk

Abstract. Nowadays, e – testing is an often used method for evaluation in the process of learning. In this paper, we discuss the e – testing problem of creating large question set that will reflect the knowledge of some domain. A new model of E – testing is introduced with a proposal of a new solution to the problem of creation a large question set for a given domain. Then, we present a methodology for comparison of the results and the contribution of the new model and realization on the automated creation of large number of questions, and we evaluate the quality and the vulnerability of the question set, as well. It is shown that the new model increases the speed of question production by more 10 times.

Keywords: Semantic Web, semantic web technologies, ontology, OWL, e - testing, question set.

1 Introduction

Semantic Web is an evolving extension of WWW, in which the meaning of information and services on web are defined in such way to allow computers to understand and satisfy human requests using the web content. The goal is to develop standards and technologies designed to help machines understand more information on the web so that they can support richer discovery, data integration, navigation, and automation of tasks.

Semantic Web is an attempt to address the initial goal of the web enabling automation. Short term goal of the Semantic Web is interoperability, and long term goal is to make computers work on our behalf instead of using them like tools [1].

OWL, as one of the developed technologies, is the language for description of ontologies. OWL document describes an existing ontology.

Semantic Web technologies can be employed in many areas of computer science. In this paper we use OWL documents in area of e-learning, particularly e-testing.

E-learning is a process of education in electronic form through Internet network or the Intranet with the use of management system for education. Evaluation is important step in learning and e-learning process.

The process of electronic evaluation of students is referred to as e-testing, web testing, online quiz, etc.

An e-test consists of set of questions that could be: multiple choice, true/false, ordering, matching, drag and drop, essay, etc.

The test could have a time limit or not, even more, every question could be time limited with different time. The question set could be predetermined or the questions could be given depending on the previous answers of the student.

Advantages of e-testing over regular testing are numerous. For example, among the possibilities offered by “Moodle” platform are the following [2]:

- Teachers can define a database of questions for re-use in different quizzes;
- Questions can be stored in categories for easy access, and these categories can be "published" to make them accessible from any course on the site;
- Quizzes are automatically graded, and can be re-graded if questions are modified;
- Quizzes can have a limited time window outside of which they are not available;
- At the teacher's option, quizzes can be attempted multiple times, and can show feedback and/or correct answers;
- Quiz questions and quiz answers can be shuffled (randomized) to reduce cheating;
- Questions allow HTML and images;
- Questions can be imported from external text files;
- Quizzes can be attempted multiple times, if desired;
- Attempts can be cumulative, if desired, and finished over several sessions.

E-testing allows evaluation of large number of students which can be very helpful in institutions where student-teacher ratio is high.

Additional features offered by e-testing provide learning manager (i.e. teacher) a tool for student self evaluation in the process of learning. Also, large set of different type of questions permit more accurate evaluation of the student. The possibility of re-grading the quizzes after modification of some question(s) offers flexibility and quick recovery if some mistake or inaccuracy in given questions is noticed.

2 Weaknesses of e-Testing

E-testing as well as regular testing has more weaknesses. One of major ones is collecting (printing, saving, etc…) questions by the students and sharing the copies among them. This can happen when the test is set to be taken by the student in unattended (and unsecured) environment, and also when the testing is performed in classroom where students are proctored by someone.

If the environment (web or application) of the test is not secure enough, possibility of cheating through going forward and backward, delaying the time, accessing other recourses is also present.

But, most important issue is question database. The questions included in e-tests can be taken from question database. With every test a part of the database is exposed. If students can save this questions they can quickly have the question database (or main part of it) so after that results from the testing will not illustrate the knowledge of the student on the subject, but just on the database.

When dealing with students that have more computer skills (IT students) one should be aware that they could try to attack the database directly using SQL injection, URL manipulation, buffer overflow, remote command execution, weak authentication and authorization, etc. [4]

When students have the questions in electronic format then if access to other applications and processes on the computer where the e-testing occurs isn't protected, students may simply search thru the list of questions (as simple as option "Find"), and just see the right answer of the given question.

Feeling comfortable about test security usually comes down to feeling comfortable that (a) the person whose name is associated with the test is indeed the person who took the test and (b) the students were not exposed to the test items before taking the test. If that comfort isn't provided through an honor code, it has to be established through the testing procedures. [5]

But, the main question is if there is a way to discourage students to make a collection of the questions from the set of questions that the teachers have. One solution already implemented in some e-testing environments is randomizing the question order and the order of answers (for example, [3]). It makes the printouts a lot less useful.

Creating larger question banks and giving tests with random subsets is also an effective strategy. If students can only print a small number of questions at a time, they will need to view the test again and again, and then sort the questions to eliminate duplicates. In this way, memorizing the questions will be rather difficult.

Very clear observation made by many researchers (for example, [7]) is that creating a question database is time-consuming. This is the task that nowadays should be done by teachers. Creating only a minimal set of questions could take more than 10 hours work per week. [6]

The question that remains open is how to create a large set of questions. This is the question of interest in this paper.

3 How to Create a Large Set of Questions for e-Testing?

One direction in which one could look for the solution is the existence of large community of teachers that can use same standard for produced questions. For example, Advanced Distributed Learning (ADL) has offered Sharable Content Object Reference Model (SCORM) which integrates a set of related technical standards, specifications, and guidelines designed to meet SCORM's high-level requirements — accessible, reusable, interoperable, and durable content and systems. SCORM content can be delivered to learners via any SCORM-compliant Learning Management System (LMS) using the same version of SCORM. [6]

In this way, large sets could be easily created but only in languages that are massively spoken and only on more common topics. Additionally, great effort should be put in division of questions in categories and subcategories.

The other direction that we propose is use of software for automatic creation (generation) of the questions. The proposed software should be able to produce a large set of questions using files that contain knowledge of a certain domain. These files should contain knowledge in "non-linear" way, difficult to be memorized by the students. The application should offer different structures of questions and possibility to change the fixed text of the question.

4 A Model for Automatic Question Production for e-Testing Systems

The model that we propose is given on Figure 1.

Semantic web technology, OWL (web ontology language) in particular, offers a way of “non-linear” description of knowledge. Nowadays, OWL files describing ontologies are produced every day for many specific domains. These files are used as sources for the produced software built on the model. The software extracts the knowledge from the file by parsing and then produces a large number of questions concerning the described domain. The questions can be of different type, but more preferably multi-choice and true-false questions, easy for computer grading.

Produced questions can be used in two ways. First option is, an other part of the software to generate the test by choosing a random subset of the questions. The test can be used to grade a student (or more students). Second option is to export this

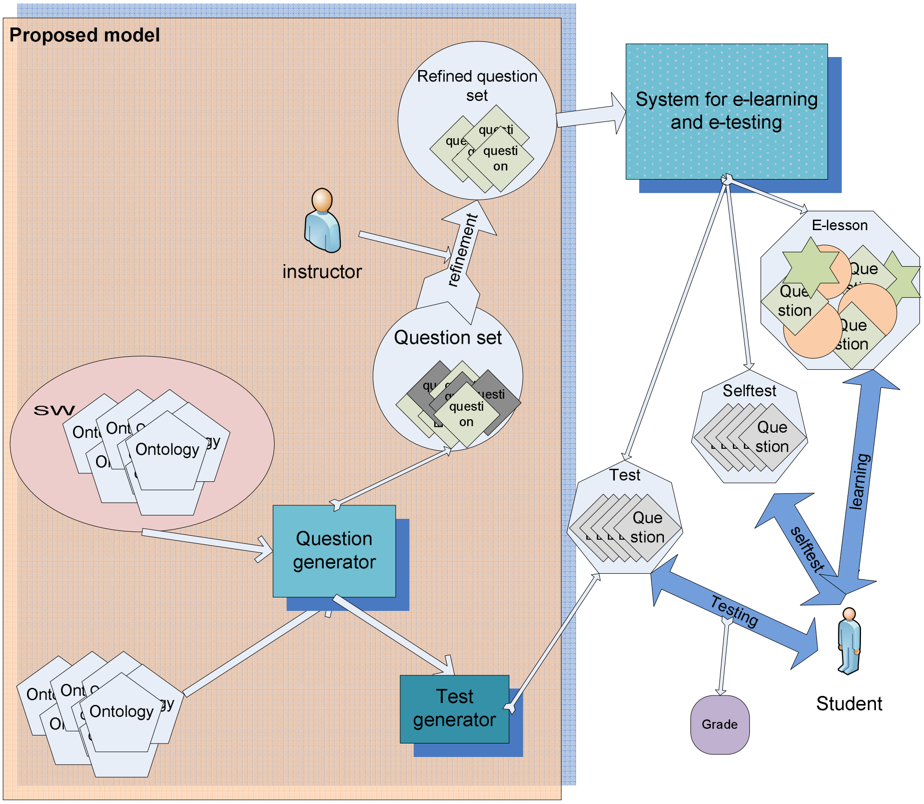

Fig. 1. A model for automatic question production for e-testing systems

questions in some format (preferably XML) and to store them. This option gives additional possibility for the set to be checked by qualified instructor in order to make corrections to some of the questions (syntax and/or semantic) or to completely reject some. Such refined question set can be used in any Learning Content Management System that allows e-testing, self-testing and/or e-lessons.

The process of question generation consists of phase in which the knowledge is extracted from the input ontology, and the phase of question generation.

In the first phase, the document is parsed, and the data structures containing detected concepts (classes, properties) are created.

In the second phase, using the elements of the mentioned structures, different form of questions are created. In the software that we produced based on the proposed model there are 27 different types of multi-choice and true-false questions (such as, questions about relations between classes, properties, characteristics of classes and properties). Different type of sub algorithm decides on the false answers that will be offered in the multi-choice questions, to mach the question itself. With exhaustive search every possible question is created.

5 Methodology to Evaluate the Model

Proposed model tries to solve the problem of the question set vulnerability. Therefore, the following characteristics are evaluated:

- Question production speed,
- Good question formulation,
- Solvability of the questions.

Question production speed is key criteria for measuring the quality of the proposed solution, as the main goal of the solution is fast production of questions. It is measured through the time interval for creating a question (or fixed number of questions), the time interval for checking a question (or fixed number of questions), which sums up to the time for producing a question. The result is compared to the time for manual production of a question.

Good question formulation as a quality is measured by counting the rejected and fixed questions in the process of question checking (refinement phase) in both ways of production.

Solvability of the question represents "the possibility" for the question to be solved by the student. In reality, there are questions that can be solved by almost anyone, and as opposite, questions solvable by very small number of students. Coefficient of question solvability is calculated for every question using:

$$k_1 = \frac{\sum_{i=1}^{t} \frac{1}{t} p_i - \sum_{i=1}^{f} \frac{1}{2f} q_i}{N} \tag{1}$$

where t represents the number of true options, f – number of false options, p_i – number of students that have chosen the i-th true option, q_i – number of students that have chosen the i-th false option, and N – total number of students that had the possibility to answer the question.

A coefficient of answering the question is also calculated by:

$$k_2 = \frac{n}{N} \tag{2}$$

where n – is the number of students that have tried to answer the question, and N – total number of students that had the possibility to answer the question. It should be stated that every inaccurately answered question gives negative points to the final score of the student, so some of them decide not to answer some question.

6 Comparative Analysis of the Results

The software that we use in testing the model performance is "OWL_Question_generator". It is produced, as visual application, based on the model in Microsoft Visual Studio C++ 2005 Express Edition. It parses the OWL document on input and stores the extracted knowledge in various data structures. Then, using different algorithms generates different forms of multi choice questions. Questions are exported in suitable XML format.

We compare the performance of this model to the existing solution of manual production of questions. The results for the creation and checking (refinement) of the questions are gained through experiments done by 8 qualified instructors on the topic of Object and Visual Programming. The result about solvability of the questions are calculated from the results of the exam given to the students taking the course Object and Visual Programming.

Table 1 shows the results for the manual production of questions. Given that average time for production of question, the calculated question production speed is 0,1814 questions/min.

Table 1. Estimated time in the process of manual question production

	Average time in minutes	**Standard deviation**
Question creation	4,131	1,086
Question checking	1,381	1,068
Total time for question production:	**5,512**	

Table 2 shows the results for the automatic creation and manual checking of questions. Given that average time for production of question it is calculated that question production speed is 1,944 questions/min.

Table 2. Estimated time in the process of automatic question production

	Average time in minutes	**Standard deviation**
Question creation	0,0004	~0
Question checking	0,514	0,292
Total time for question production:	**0,5144**	

According to the previous result, we may conclude that even when the process of manual refinement of question set is included in the question production, the new model offers ***almost 11 times faster production***.

If we consider ***the good question formulation*** according to the results in the process of manual production of questions 39,88% of the questions were repaired (changed) and 7,14% were rejected. On the other hand, in the process of automatic production 2,44% were repaired and 0,35% rejected. So, in both cases (repairing or rejecting) process of automatic production shows ***over 16 times better results***.

Solvability of questions is calculated on every question in both sets by giving the questions to large number of students. The gained interval for the coefficient of solvability in both cases is [-0.3, 1]. Table 3 defines the boundaries of "classes" of solvability.

Table 3. Defined boundaries of the classes of solvability

Class	**Interval of coefficient k1**
1. "very hard to solve" question	[-0,3; -0,04]
2. "hard to solve" question	(0,04; 0,22]
3. "standard solvable" question	(0,22; 0,48]
4. "easy to solve" question	(0,48; 0,74]
5. "very easy to solve" question	(0,74; 1]

Figure 2 and Figure 3 show that produced questions in both ways are distributed in the 5 classes of solvability with no significant differences.

In the case of automatic production of questions slightly greater solvability, but more important in both cases there is non-uniform but good distribution among classes.

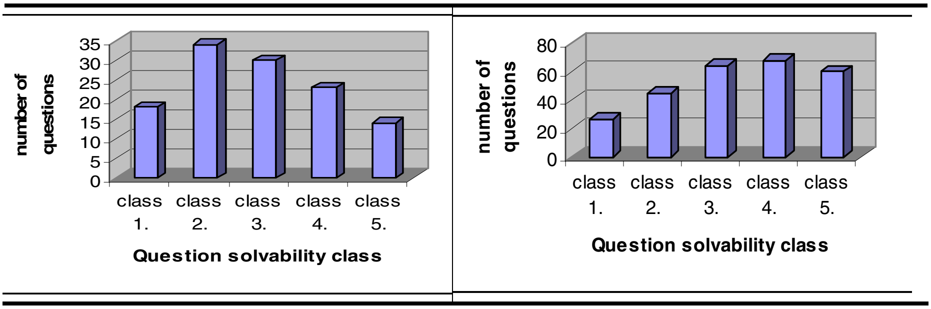

Fig. 2. Column charts showing the number of questions per class of solvability, produced manually (on left) and automatically (on right)

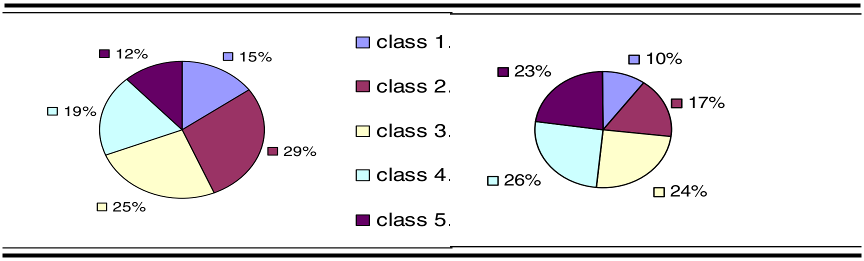

Fig. 3. Bar charts showing the percentage of questions per class of solvability, produced manually (on left) and automatically (on right)

Figure 4 represents coefficient of answering the question by classes. Here, the coefficient is in the interval [0; 1], and the five presented classes are [0; 0,2], (0,2; 0,4], (0,4; 0,6], (0,6; 0,8], (0,8; 1]. It can be concluded that students more bravely were answering the automatically produced questions. This is, probably, due to the fact that for automatically produced questions there is finite number of formulations of the questions.

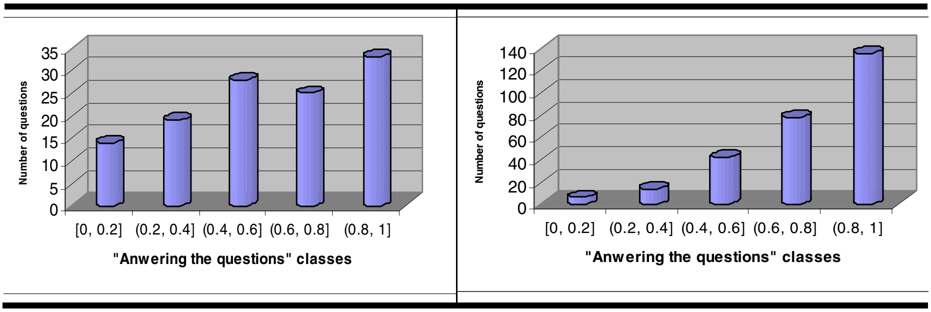

Fig. 4. Column charts showing the number of questions per class of answering the question, produced manually (on left) and automatically (on right)

Additional qualities offered by the new model are:

- Creating the questions unmistakably
- Form of question storage

Creating the questions unmistakably is an important quality which can be offered by any (well done) software in a process versus a process done, or partly done by human. In our case this quality depends on the produced software based on the model and on the ontology used as input in the software.

Form of question storage can have a great effect on the vulnerability of question set. Software based on presented model, can test the student even without a stored question set, because the questions can be produced in the same moment. In this case the advantage of this model is obvious, because in this way the knowledge is coded in the ontology, not in the question set. So, someone who tries to game the system can only get the ontology, but if she learns all the concepts and relations in it, she will have the necessary knowledge.

However, if we decide to use this approach we well have to sacrifice the possibility to store the questions in a database. In this case the testator will not be able to check the created questions and to select just part of them as a pool for testing.

On the other hand, even if we decide that we need to store the questions (to have possibility to check them) there is still an advantage because the main goal of a large question set is achieved.

7 Conclusion

The problem of vulnerability of the question set in e-testing systems motivated our research. In this paper we presented a performance evaluation of a new model for automatic question production that uses Semantic Web ontology (OWL document) as input. The model allows very fast production of large question set. Even with the additional checking of produced questions the production speed is 10 times bigger than in the process of manual production. Good results of the model are also shown on "good question formulation" quality. We showed that the set of automatically produced questions doesn't significantly defer from the set of manually produced ones, in the sense of question solvability. So, the presented model could be used in the process of creation of questions for e – testing purposes.

References

1. Lasila, O.: Towards the semantic web. Presentation. W3C Semantic Tour, London (2003)
2. Moodle, Features, Quiz module, http://docs.moodle.org/en/Features#Quiz_Module (last accessed July 2009)
3. Gusev, M., Armenski, G.: E-Learning realized by E-Testing. In: Proceedings of the 2nd Conference on Informatics and Information Technology, pp. 181–188. Institute of Informatics, PMF Skopje (2002)
4. Lim, C.C., Jin, J.S.: A Study on Applying Software Security to Information Systems: E-Learning Portals. IJCSNS International Journal of Computer Science and Network Security 6(3B), 161–166 (2006)
5. Rocklin, T.: Computers and testing. The National Teaching and Learning Forum 8(5), 1–4 (1999)
6. Advanced distributed learning, http://www.adlnet.gov/scorm/index.aspx
7. Pain, D., Le Heron, J.: WebCT and Online Assessment: The best thing since SOAP? Educational Technology & Society 6(2), 62–71 (2003)