On Possibility of Automatic Generation of Data Files and their Use in Tasks of Descriptive Statistics

Mikuláš Gangur[1] and Václav Sova Martinovský[2]
[1]*Department of Economics and Quantitative Methods, Faculty of Economics, University of West Bohemia, Univerzitní 8, Plzeň, Czech Republic*
[2]*Department of Business Administration and Management, Faculty of Economics, University of West Bohemia, Univerzitnı 8, Plzen, Czech Republic*

Keywords: Quiz, Question with Data File, Descriptive Statistics, Automatic Generation, Moodle XML, LMS Moodle.

Abstract: The contribution deals with the use of automatic generation of parameterized tasks and the inclusion of automatically generated data files as useful feature of generator. This functionality of the generator is used, for example, in tasks of statistical data analysis. The contribution shows usage when generating a descriptive statistics tasks. The basic principles of preparation of statistical data in both sample and population files are described. Methods of integrating these data into the generated task for different output formats are also explained. Selected group of tasks illustrating the outputs of the application of the described methods are presented. The use of a generator for building a Question bank in LMS Moodle is shown as well as the preparation of writing tests. At the same time, various data files storage options and subsequent use with regard to their lifetime are discussed. In the following, a new solution for the implementation of the automatic generator of parameterized tasks in cloud is introduced. This approach will allow the involvement of more users of the automatic generator.

1 INTRODUCTION

Automatic generation of parameterized task can generate a unique problem for each student. Students can practise one task with different input parameters. It helps them to understand the problem from different points of view. In the article (Gangur, 2011a) an automatic generator of questions and answers is described including the architecture of the generator. The universal principle of the automatic parameterized questions generation was explained.

The generator generates variable parameters in the task text. The values of these parameters are generated as a random value according to specified constraints specific to each task. The output of this generator module is a complete definition of the task in universal XML format. Subsequent XSL transforms the task into different output formats, depending on the XSL template used. The most commonly used formats are LaTeX and Moodle XML. The first one allows you to create a PDF file with the generated task texts in one version for students and another version for teachers with tasks answers for correction. The second format is used to import the generated tasks into the Question Bank in LMS Moodle. In this way, a large number of variants can be generated from each task.

Based on this principle, it is possible in LMS Moodle to create a unique task for each student and to assemble a unique quiz from multiple tasks. This approach allows you to practice the problem on different input data. This method helps students understand the principles of problem solving instead of ineffective memorizing the solution process.

A specific approach is applied when generating tasks from the field of statistics and statistical processing as well as an analysis of the data. In this case just the statistical data are the key object. Creating the data according to the required parameters, storing the data in the file and inserting the information about the file must be an integrated part of the generation process, as well as the information about the data themselves and then storing them in the output description of the generated task.

This contribution introduces the using of a new feature of the question that can be processed by the generator, i.e. automatically generated parameterized data, i.e. data file. This feature and usage is described and explained on an example of descriptive statistics tasks in which sample and population data files are

Gangur, M. and Martinovský, V.
On Possibility of Automatic Generation of Data Files and their Use in Tasks of Descriptive Statistics.
DOI: 10.5220/0007742504460453
In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU 2019)*, pages 446-453
ISBN: 978-989-758-367-4

used. The generation of sample data files is described in this article as well as implementing an automatic generator in cloud is explained. In the same described way, it is possible to generate sample data files for other statistical analysis tasks (statistical inferences).

The reminder is structured as following. The section 2 provides an overview of the literature on the problem of automatic generation of tasks. Section 3 describes methods for generating statistical data and their integrating into the generated tasks. In this part the subsection 3.2 explains the process of sample data generation. Section 4 presents the results of the generating process in the form of examples of generated tasks in different output formats. A brief description of the generator implementation in cloud is also part of this section. In section 5 some disadvantages and limitations of proposed solution are discussed. Finally in section 6 we evaluate used methods and we state conclusion.

2 RELATED WORK

The problem of automatic generation of questions has been dealt with in some specific domains. In (Cristea and Tuduce, 2005) the questions were generated in the area of electrical circuit analysis; another study focuses on question generation in the domain of the object-oriented programming (Hsiao et al., 2008). The problem of math/science tasks generation is addressed in (Ugurdag et al., 2009). The authors generate multiple choice questions in the process of image modification by means of a developed graphic tool. In this case new values of parameterized questions are not generated automatically. Authors (Zhang et al., 2017) solve the problem of automatically generated questions with images. They generate visually grounded questions with various types for the same visual input.

Test questions generated automatically directly from chosen text can be seen as a special issue in this direction (Zeng et al., 2013). Contributions dealing with this topic concentrate mainly on generating questions from the English texts (Sung et al., 2007), (Afzal and Mitkov, 2014), (Shah et al., 2017), from knowledge database (Rocha and Zucker, 2018) or from web (Cubric and Tosic, 2011) by semantic web technologies using domain ontologies. Automated creation of adaptive tests with regard to the level of the knowledge of individual students is an independent field in which intensive research is being carried out (Kapusta et al., 2010).

Except for (Zhang et al., 2017), the above mentioned systems generate only text questions in a basic question type with numerical, short or multichoice answers. In the same way, the data types of the input variable parameters are of numeric types or text types. None of these systems provide the possibility of attached data files that are connected with the dynamically generated content of questions. The majority of the above stated tools as well as other examined instruments generate only online web tests.

In this contribution such features of the question generator are introduced. The process of generating dynamical sample data dependent on other generated content of questions is explained as well as the implementation of automatic generator in the cloud environment.

3 METHODOLOGY

The basic tool for generating a large number of tasks is the automatic parameterized task generator developed by the authors of this contribution (Gangur, 2011a). This system allows you to generate tasks with different variables in the input task text, and is able to generate and draw the mathematical text (Gangur, 2011b). In addition to the automatically generated input values, the generator is able to insert images dynamically generated by the parameters in the input text (Plevný and Gangur, 2016) eventually data files that match the task entry text (Gangur, 2018). The generator is used to generate tasks across a range of areas, Financial Mathematics, Statistics, Statistical Data Processing, but also in Management Science (Plevný and Gladavská, 2014).

The process of transforming the generated questions into the demanded output format consists of two basic steps:

1. The questions and correct answers are generated in the selected software (Matlab, Mathematica, etc.). The description of the generated questions and all their parameters including the answers in the universal XML structure are thus obtained.
2. The XML description of questions is used as the input to the XSLT process. In this phase the questions are transformed into the required final format according to the XSL template (Holzner, 2002).

Below we will describe the use of the dynamically generated data files that are integrated as part of the generated task.

3.1 Statistical Data Files

When generating statistical tasks, the generator's features are to create dynamic images and, in particular, statistical data files unique for each student. Next, we distinguish two types of data files:

- a temporary sample statistical file, which is part of each variation of the generated task,
- stable population statistical file that is stored on the available storage in only one version for all the generated tasks.

The difference between the two file types can be identified from two views. From the statistical point of view, this is the difference between the sample data file and the population file, as a key point in the processing of statistical data and the use of statistical inference and hypothesis testing the population parameters. The various generated tasks then show this difference, and the more generated variants with the sample file allow you to calculate the characteristics of the various sample files derived from only one population data file.

In case of the second, rather a technical problem, the population file is one permanent file stored in a dedicated open storage. This one is permanently accessible to all users and is exploited by various generated tasks, and a URL link to the file in the storage is placed in the input task text. In case of a sample file, we generate a temporary data file as part of the task definition. It is necessary to treat access to these data for different output formats of the generated task.

- In case of the MoodleXML (MoodleXML, 2014) output format, a generated task is saved in the Question Bank of LMS Moodle (Moodle, 2010) and the sample data file that is part of the task definition is stored in the internal repository of LMS Moodle. A URL link to this file is generated as part of task text (Gangur, 2018).
- If the output of the generated task is in the format LaTeX (LaTeX, 2013) and then a PDF file, the link to the data file is again a part of the task text, but in this case the URL is a reference to dedicated accessible storage for the data files. These files are again divided into two types with respect to their lifetime. The first ones are the data files that are part of the assignment of, for example, exam tests. The lifetime of such data is only until the test is entered. The second type of the file is the data referenced from the input, for example, sample tests that are part of various syllables, e-courses, etc. The lifetime of such files is much longer, theoretically unlimited.

3.2 Population and Sample Statistical Data File Generation

The process of generating data files is shown on a group of tasks that are based on one selected population file. The source data of the example are taken from a population file with the data from the last population census in the Czech Republic in 2011. The data file contains 5894 records about municipalities in 14 counties. Each county is made up of districts. In the data the information are stored in variables according Table 1. Each municipality is identified by its name, district name and county name. Then the information about population of various groups in municipality is stored. The monitored groups are total population, marital status groups and age groups. The population of each group is stored as total and separately for male and for female.

The automatic generator works with the data structure of the file according to a particular generated task. In case of tasks that work with other population the task solver of the automatic generator will conform to the structure of the used population files due to the implemented task algorithm.

The population data file is too large and it is not attached to every generated task. This particular file is stored in cloud and the generating process creates URL reference to this file in task text. In the generated tasks the calculation of the population parameters is demanded. The average and standard deviation of different population groups (see Table 1) are calculated per municipality or per district in each county. For example the average and standard deviation of male per municipality are calculated, or the average age of single female per district is calculated. These tasks and their solutions are the same for every student.

The calculation of sample data characteristics is more useful for practicing students. In these tasks the same characteristics are calculated as the parameters of the population file (see previous paragraph). The sample file creation is one of the most important things. A unique sample file is generated for every task and each student calculates the demanded statistics for different sample file of one source population file. Many variants of one task allow to recalculate statistics (averages and deviations) for different sample data. This possibility is very useful when tracking iteration of statistics to population parameter values.

The key procedure *select_data()* creates sample data from the population data according to the list of counties, demanded variables and demanded number of records of each county. The procedure randomly selects municipality records using generator of uniformly distributed pseudorandom integers in scale of

Table 1: Variables in population census.

Municipality name	District name	County name
Population		
total	male	female
marital status single	marital status married marital status widowed	marital status divorced
aged 0-14 aged 30-39 aged 60-64	aged 15-19 aged 40-49 aged 65-69 aged 80+	aged 20-29 aged 50-59 aged 70-79

municipality number for every county. After this random selection we need to run the post-process that modifies the sample data file according to the following requirements:

- Each district in each county is represented by at least two municipalities in the sample data.
- The sample data file contains the municipality with the largest and least populous population.

Described in this way, a large number of sample files can be derived from one population file by random selection. As a result, each student solves his/her own assignments, learns the principles of the solution and practices the basic algorithms of descriptive statistics.

The principle used to integrate a data file into a task definition allows you to create, by using built-in functions of the used software, the sample data of the desired statistical distribution and other (for example extreme values and outliers) properties to practice exploration data analysis.

4 RESULTS

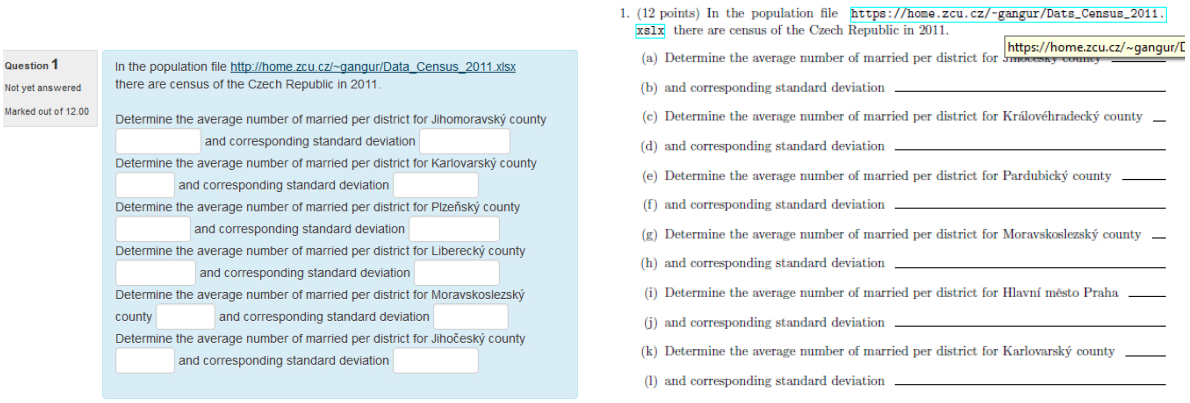
The described features of the automatic generator of parameterized tasks extend the use of the generator in the field of statistical tasks and, in particular, statistical data analysis tasks. For the course Statistical Data Analysis, 19 tasks in the field of descriptive statistics were created and 100 variants were generated for each of these tasks, which were inserted into the LMS Moodle Question Bank. In total, 1900 unique tasks were generated for the 150 students enrolled in the course. Other 19 tasks from the total of 211 tasks in the course were created for the other areas, two sample tests, one sample tests, ANOVA, exploration data analyses, normality and homoscedasticity testing.

The Figure 1 show the generated tasks in which the parameters of population data are calculated. A link to a population census file can be found at the beginning of the task text. In figure 1a, the task is generated in the Moodle XML format and inserted into the Question bank of LMS Moodle. Blank fields allow you to enter the required parameters values. The Figure 1b depicts PDF format of the task generated from the same pattern as the above described task in Moodle. Such a task can be printed as a written test.

The Figure 2 depicts the different output formats of the generated task in which characteristics of the sample require to be calculated. The sample data are randomly generated from the population data file according to the procedure described in subsection 3.2. In both outputs in LMS Moodle or in the PDF file, you can find at the bottom of the task the link to the sample file stored in the Moodle LMS repository or in the reserved temporary storage facility. This differs from the type of tasks in Figure 1. While the population file is located in a single location and the data file is common to all variants generated (the same URL is given in the text of all variants of the given task type), in case of sample file in the task in Figure 2, the generated data files are unique for each task variant, and therefore, a hyperlink that is embedded in the input text is another URL in each task variant.

Links to data files can be used when working online in LMS Moodle to download a file and open it in MS Excel. In case of a PDF file, the data can be obtained directly only when the PDF file is opened online. When writing a workload variation, it is necessary to insert the displayed URL into the web browser. For this reason, in the PDF variant of the URL, the links to files are referenced throughout the full URL specification.

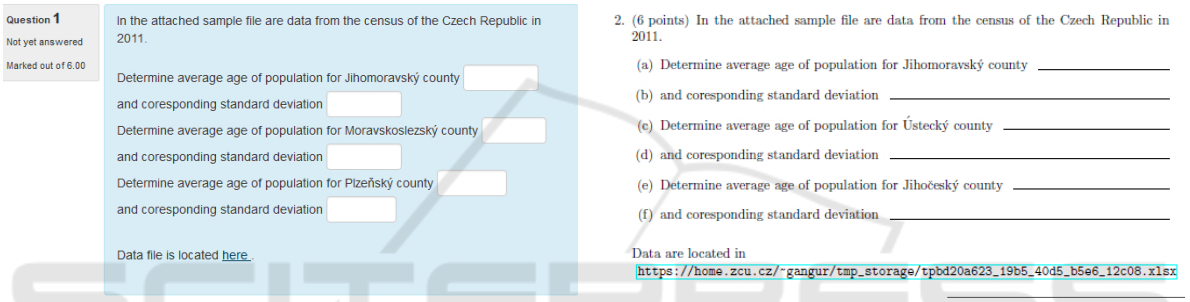
A major innovation of the generator is its implementation in cloud. This solution extends the capability of using it for a larger number of users without the need for a desktop installation of the generator and support software Matlab (MathWorks, 2013).



(a) Output as question in Moodle

(b) Output as question in PDF

Figure 1: Generator outputs of question with population data file.



(a) Output as question in Moodle

(b) Output as question in PDF

Figure 2: Generator outputs of question with sample data file.

4.1 Automatic Generator in Cloud

One of the drawbacks of the existing solution is the dependency on a particular computer where Matlab is installed. That also means that besides the source code of the generator and the source code of all tasks, every user needs a valid Matlab license. If we consider that, this solution is not cost-effective and also very hard to configure and run. Wider usage is therefore very limited.

The proposed cloud solution removes all above mentioned barriers. Since it is available in a software-as-a-service (SaaS) principle, the end user needs only a computer with a regular web browser (Martinovský and Plevný, 2018). No additional software or license is needed. The usage is not limited by hardware (it works on mobile or desktop), software (no special programs needed), user skills (intuitive web GUI) or physical location.

This solution is also cost-effective. It can use all advantages of cloud computing pay-as-you-use model and with flexible scaling, the actual cost may vary according to the anticipated traffic. It also enables the possibility of extended functionality like sharing tests between users, collaborative work etc.

The *backend* of all the application is based on the already proposed generator in Matlab. To achieve independence on a particular computer or licence the existing solution was extended by a set of functions for communication with external programmes and the whole project was translated into an autonomously executable programme by means of Matlab Compiler. It is necessary to have a licence for only one computer on which the translation of the source files is underway (not necessarily in cloud).

The *frontend* is formed by an application created in the PHP script language extended with the Slim framework. The interface is based on a fully responsive template using Bootstrap framework and jQuery library.

Thanks to these components it was possible to create a user friendly interface which can be operated comfortably without any further adjustments on a wide range of devices (see figure 3).

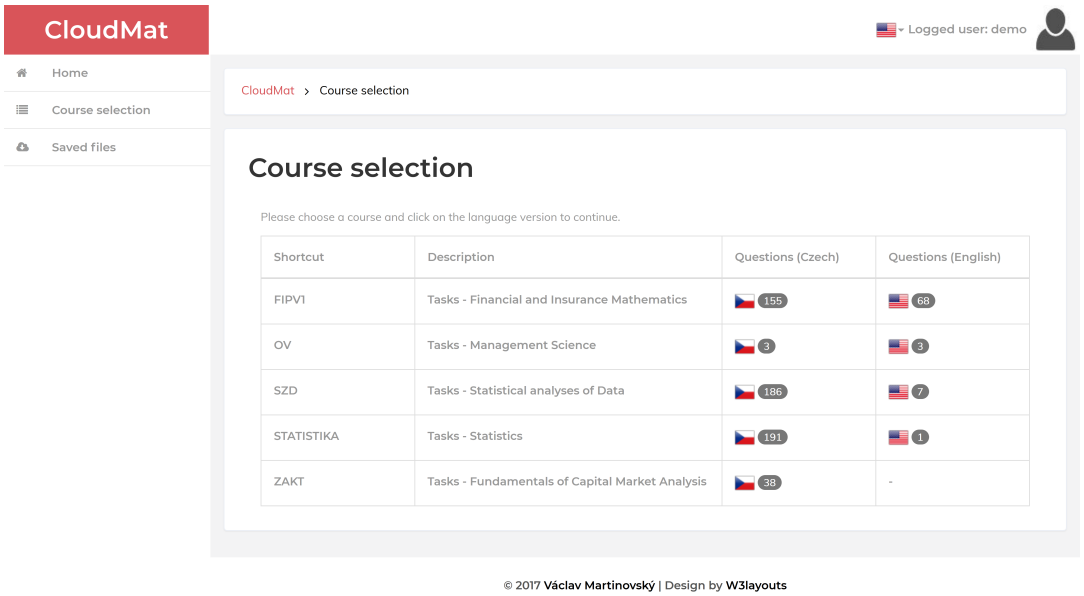


Figure 3: Application interface on a desktop.

5 DISCUSSION

The described solution also has its disadvantages and limitations. The disadvantage of statistical data analysis tasks may be seen in the different iterative algorithms by which various statistical data processing systems solve the given problems. The resulting set values of a task solution that Moodle checks with the specified accuracy may then vary if students use software other than the automatic generator that the Matlab system uses in the described solution. Most often the results differ when calculating the p-value, which is often an important part of solving in statistical data analysis tasks. A partial solution to this problem is the very accurate work with the set tolerance for the entered solution values. Small tolerance leads to different results when the LMS does not accept the values that are calculated by the correct procedure. Big tolerance, on the contrary, allows acceptance of values, that the student has reached in the wrong way, and like this, the correctness of the wrong algorithm is confirmed. Additionally, using such imprecise methods for further calculations can lead to an increase in errors and to the rejection of other task resolving results.

Another limitation when working with the population and sample data file is the size of the source data in the population file. The automatic generator is built on the encapsulated module to generate a single task, and then uses this module to generate various tasks. In case of creating a Question bank in LMS, Moodle the system generates a given number of variants of the same task, and a test of different tasks is created when creating a PDF test. This solution, in which the module always loads the entire population file over the network, is very lengthy and the generation of bigger number of tasks is a time-consuming procedure. For this reason, it is advisable to place the data of the population file in memory once only and to inform the basic module about their loading and location. The system does not have to retrieve population data over the network when it is reused.

6 CONCLUSIONS

The described methods allow practical application in the courses of statistics and statistical data processing. The advantage is the possibility of creating a large number of different tasks that can be stored in the e-learning system repositories along with the generated training data. A unique assignment for each student supports the student's own work and the possibility of different variants of principally the same task requires understanding of the principles of solving the assigned tasks instead of ineffective memorizing procedures.

In the contribution, only generator options in the used area are shown. An example of generating data for statistical analysis describes the use of this generator functionality. However, this functionality is also suitable for tasks from a wide range of areas where the problem is described by a larger amount of input

data that cannot be entered in the task input text and it is useful to place these input data in the attached file.

The paper does not address the evaluation of the described approach used in the teaching process and its effect on the effectiveness of teaching. This is part of the future work where the results and the course of study of two groups of students are compared. In one group, the generated tasks were not used and the exercises were performed on tasks that are the same for all students. In the second group, they were used to practice the generated task unique for each student. Similarly, any impact of the length and course of the problem solution on the student's results is analysed.

The proposed solution for the implementation of the automatic generator in cloud enables to extend its use among more users and it offers its functionality via the web interface. In the future we will also focus on implementing this service as APIs.

ACKNOWLEDGEMENTS

This paper is published thanks to the support of the internal projects SGS-2018-042 of the University of West Bohemia in Pilsen, Czech Republic.

REFERENCES

Afzal, N. and Mitkov, R. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing*, 18(7):1269–1281.

Cristea, P. and Tuduce, R. (2005). Automatic generation of exercises for self-testing in adaptive e-learning environments: Exercises on ac circuits. In *Inter. Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia (Part of WBE)*, pages 1126–1136. WSEAS Publishing.

Cubric, M. and Tosic, M. (2011). Towards automatic generation of e-assessment using semantic web technologies. *International Journal of e-Assessment*, 1(1).

Gangur, M. (2011a). Automatic generation of cloze questions. In *Proceedings of 3rd International Conference on Computer Supported Education Vol.1*, pages 264–269. SciTePress, Portugal.

Gangur, M. (2011b). Automatic generation of mathematic tasks. In *Recent Research in Educational Technologies, Proceedings of the 7th WSEAS/IASME Intl. Conference on Educational Technologies*, pages 129–133. WSEAS Press.

Gangur, M. (2018). Automated generation of statistical tasks. In *Proceedings of DIVAI 2018 - 12th International Scientific Conference on Distance Learning in Applied Informatics*, pages 47–58. Praha: Wolters Kluwer.

Holzner, S. (2002). *Inside XSLT*. New Rider's Publishing.

Hsiao, I., Brusilovsky, P., and Sosnovsky, S. (2008). Web-based parameterized questions for object-oriented programming. In *World Conf. on ELearning in Corporate, Government, Healthcare, and Higher Education*.

Kapusta, J., Munk, M., and Turcani, M. (2010). Evaluation of adaptive techniques dependent on educational content. In *2010 4th International Conference on Application of Information and Communication Technologies*, pages 1–5.

LaTeX (2013). Latex - a document preparation system. Retrieved December 2013, from http://www.latex-project.org/.

Martinovský, V. S. and Plevný, M. (2018). Applying cloud computing for automated generation of parameterized tasks and test. In *Proceedings of DIVAI 2018 - 12th International Scientific Conference on Distance Learning in Applied Informatics*, pages 357–365. Praha: Wolters Kluwer.

MathWorks (2013). Matlab. Retrieved September 2013, from http://www.mathworks.com/.

Moodle (2010). Moodle - a free, open source course management system for online learning. Retrieved January 2011, from http://moodle.org.

MoodleXML (2014). Moodle xml format - moodledocs. Retrieved January 2014, from http://docs.moodle.org/23/en/Moodle_XML_format.

Plevný, M. and Gangur, M. (2016). On the possibility of solving the problem of automatic generation of images. In *Proceedings of DIVAI 2016 - 11th International Scientific Conference on Distance Learning in Applied Informatics*, pages 385–395. Praha: Wolters Kluwer.

Plevný, M. and Gladavská, L. (2014). Problems of automatic generation of questions for the purpose of testing the knowledge in a management science course. In *Proceedings of DIVAI 2014 - 10th International Scientific Conference on Distance Learning in Applied Informatics*, pages 325–335. Praha: Wolters Kluwer.

Rocha, R. O. and Zucker, C. F. (2018). Automatic generation of quizzes from dbpedia according to educational standards. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1035–1041, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Shah, R., Shah, D., and Kurup, L. (2017). Automatic question generation for intelligent tutoring systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 127–132.

Sung, L.-C., Lin, Y.-C., and Chen, M. C. (2007). An automatic quiz generation system for english text. In *7th IEEE International Conference on Advanced Learning Technologies, Proceedings*, pages 196–197.

Ugurdag, H., Argali, E., Eker, O., Basaran, A., Gören, S., and Özcan, H. (2009). Smart question (sq): Tool for generating multiple-choice test questions. In *Proceedings of the 8th WSEAS International Conference on Education and Educational Technology*, pages 173–177.

Zeng, J., Sakai, T., Yin, C., Suzuki, T., and Hirokawa, S. (2013). Automatic generation of tourism quiz using blogs. *Artificial Life and Robotics*, 17:412–416.

Zhang, S., Qu, L., You, S., Yang, Z., and Zhang, J. (2017). Automatic generation of grounded visual questions. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 4235–4243. AAAI Press.