VOL. 11, NO. 5, MARCH 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

NEW APPROACH TO DESIGNING AN EDUCATIONAL AUTOMATED TEST GENERATION SYSTEM BASED ON TEXT ANALYSIS

Anton Valeryevich Arzhakov and Dmitry Sergeevich Silnov
Department of Information Systems and Technologies, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Moscow, Russia
E-Mail: ds@silnov.pro

ABSTRACT

A new approach about automated test generation presented. Original model gives new opportunities to generate well done educational tests. New software developed that gives ability to generate automatic tests just from books and educational manual. It can be made fast and without professional assistance.

Keywords: test generation, study tests, text analysis, tests with open keys.

1. INTRODUCTION

Modern education in many subject domains is automated nowadays. Generation of questions and test items is one of the areas that have not been well automated. Test generation itself is a creative process and often requires quite a lot of time. The quality of manual tests may vary depending on the requirements involved, the amount of time spent on creating the tests and the skill of the person creating them. In today's world, however, high difficulty levels are often not required for tests. But a large number of tests, with acceptable quality and generated in the shortest possible time is required.

Tests come in many varieties and can be applied in a number of educational activities, such as knowledge monitoring, test, exam and others [1]. The nature of tests for humanitarian and technical sciences are different. Therefore, approaches to creating them should also differ [2]. There are a variety of automated testing tools. Automated generation of problems involves generating test items based on certain knowledge. This field is currently poorly explored. Therefore, this paper will focus on this issue.

As shown by analysis of the affected subject domains, this problem belongs to the class of intractable problems. Implementation of the approach described will positively affect the education automation process and will be very effective in many cases. In the ideal case of test 'hominization', teacher's intervention in preparation of sets of tests can be eliminated completely.

2. ANALYSIS OF EXISTING APPROACHES

Despite the fact that the issue of automated test generation is currently not well-explored and the range of problems posed has not been touched at all, there are studies dedicated to solving automated test generation problems. One of these directions is the approach to automated construction of questions to a text. In [3], it is proposed to create meaningful questions to a text, and then use only those that best meet the teacher's criteria. It is intriguing to use this study in the framework of the approach developed. But there are some problems. The most important among the problems is the approach of [3] formulated for English language, when the target language for the approach is Russian. In Russian, automated generation of a question to a text is much more difficult because of the structure of the Russian text unlike the strict structure of the English language.

Another direction is the automated test generation based on a model of the subject domain, similar to the approaches described in [4] and [5]. This approach involves construction of a model based on which the testing system generates tests. A specialist who is knowledgeable in the respective subject area should be the one to construct such a model. The approach calls for detailed domain decomposition, model building, in which the laws of obtaining output data based on input data are determined and in which directories with which tests are built are populated. This approach yields sufficiently high quality tests but requires considerable time and effort, as well as highly-skilled specialists (to create a model of the subject domain), both of which in some cases are lacking.

3. A NEW APPROACH TO SOLVING THE PROBLEM

This approach is primarily aimed at automatic generation of test items based on a text material. The approach can be divided into two parts: text analysis and test generation. Both parts feature a wide range of problems that need to be addressed for the system to function. Let's consider the scheme in Figure-1. A text material is fed to the system at the input; the system examines this material, picking out the knowledge it needs, and then transmits it to a data repository. At the second stage, the system retrieves data from the repository and then generates tests based on predefined models and settings from the configuration file.

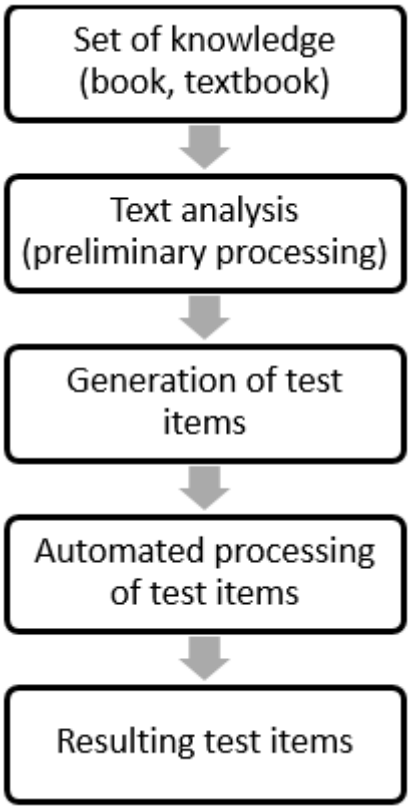

Figure-1. Scheme: system functioning.

Since the approach put forward is aimed at test preparation and not on the testing process, the use of third-party testing systems requires interfaces for interaction between the system described and the testing systems. The problem here lies only with technical implementation, which is not that difficult. This approach has two major advantages: there is no need to design a testing system and there isn't any need to change the testing system for potential users. Let's consider in detail the question of dynamic and static test generation. Each approach has its pros and cons.

Dynamic test generation involves generating the relevant questions on the fly when prompted by the testing system. The advantage of this approach is that a new test item with sufficient degree of uniqueness is created each time a query is made to the system. The disadvantage is obvious - such tests will have lesser quality.

Static tests are tests generated by the system once. The advantage here is that the tests are generated once and there is no need to spend time and computing power on creation of a test item. Another advantage is that, once it is in the system, the test can be manually modified by the author to significantly improve the quality of the tests. One of the methodologies for improving the quality of training materials, which is called crowd forming [6], fits concisely into the test generation concept. The main shortcoming is that the tests are completely static. Consequently, it's easier to find the right answers for the test, which significantly compromises on objectivity. To determine the quality of tests, the criteria described in [7] and [8] are examined.

Using the advantages of both approaches will be the best option among the approaches described, i.e. part of the test at each request is generated dynamically, while the other part is static.

Text analysis

In this particular case, text analysis means information search in the text based on chosen templates. There are two basic kinds of templates: templates for terms and templates for classifications or properties. The term templates can include such templates as "Term is a definition", "Term means definition", "Definition is the term" and other templates describing widely-used formal records of definitions. The classification templates are similar to the term templates. They are constructed under widely-used formats of classifications and descriptions of properties of an object or concept.

On one hand, this analyser is enough for the system to function at the basic level. On the other hand, by adding additional features to the text analyser, one can achieve better results during the second stage - test generation. Such feature may include semantic text analysis [9] and construction of semantic networks. From semantic analysis, we are primarily interested in identifying key words in the text. The purpose of applying these functionalities will be disclosed below. Besides, philological analysis [10] should be carried out to ensure that the system functions properly and that more 'human' tests are generated. So, despite the seeming simplicity, text analysis is not that trivial and there are a number of unresolved problems.

Test generation

Test generation is the second part of the approach. After receiving a glossary of terms and classifications from the text analyser at the output, you can begin to generate tests; the test types are shown in Figure-2.

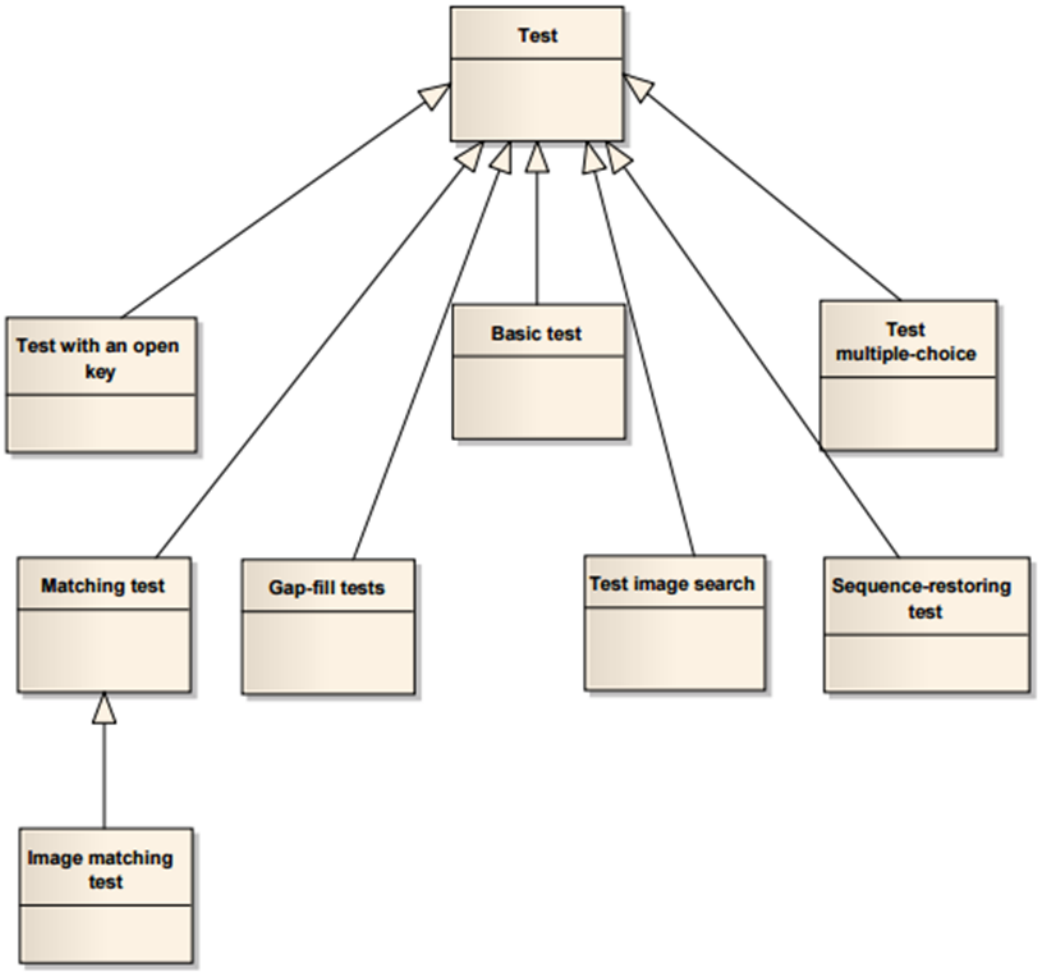

Figure-2. Classification of generated test by type of answers.

Tests with an open key

The easiest terminological test, which can be generated, is the test with an open key. Here, definition is the question, while the term associated with the given definition is the answer. That is, for each term from the glossary, one such question can be constructed.

Tests with one true answer

The next type of tests under construction is the conventional tests with one true answer. The difficulty level of these tests can be varied depending on the question and wrong answer choices generated. Below, we will consider the principles of constructing wrong answer choices. These tests can be generated in two formats: the term serves as the question, while the definition serves as the answer, and the second option will be on the contrary, i.e. the definition stands as the question, while the term acts as the answer. In the first case, when definitions are used as the answer choices, two types of generating answer choices can be considered. The first and more obvious type is when incorrect answer choices are presented as definitions from the same subject domain as the original definition. The second and more complex and interesting type is to change the true definition by replacing the keywords in that subject domain with other words in the same subject domain or adding negation particles that make it hard to find the correct answer. Thus, two test items can be constructed for each term from the glossary. And considering that the complexity of answer choices can be varied, a large number of test items can be generated for each term.

Multiple-choice tests

Multiple-choice tests can be constructed in cases where the glossary contains more than one definitions of the same term. This is the case when definitions are used as answer choices. Wrong answers can be constructed similarly to the tests with one true answer. Another basis for construction of multiple-choice tests is the classification and properties of the object, where wrong answer choices can be generated based on the similar principles described above.

Matching tests

Matching tests are the next type of test that can be generated based on the glossary. Depending on the

configuration, matching tests may contain different number of terms and definitions. This type of test has two columns, the left containing mixed terms and the right column featuring mixed definitions. The test-taker is required to match a term with the appropriate definition.

Gap-fill tests

Test items are constructed by removing the keywords of the subject domain from the definition and offering these removed keywords as answer choices to be inserted in the respective gaps. Such tests can be constructed for each definition. The number of gaps in such a test can be configured via settings.

Text/image matching tests

Text/image matching tests will be interesting for some subject domains. For this type of test, it is necessary to additionally add an image analyser to the text analyser. The image analyser will highlight and store (from the original text) the image and the text describing that image. The test is generated similarly to the text/text matching test. In taking this test, the test taker is required to match the text column with the image column.

Image search tests

In developing the theme of image processing, generation of image search tests can be an intriguing type of test. In such a test, it is necessary to find an area on the image that corresponds to the question posed. For example, having the image of an apple and having a picture of an apple orchard, you can generate a search test asking the test taker to search for an apple on the apple orchard picture. This type of test requires a lot of image processing and pattern recognition. However, it is a very interesting and non-trivial type of test.

Sequence-restoring tests

If succession stages are considered among classifications and properties, we can build another kind of test. In such a test, it is necessary to restore the sequence of actions. Similarly, this type of test can be used to restore the order of words in a sentence. However, application of tests in this way isn't productive.

Partial semantic analysis allows for more extensive sampling on the subject domain, thereby helping to generate more sophisticated tests. This can be achieved by having a common glossary for the entire system, which will help to sample not only the terms of one textbook but all the terms of the subject domain, thus making it possible to generate more interesting tests. That is, each term has a certain cloud tag of keywords, together with which it is used in the subject domain. This approach allows to make the generated test model more sophisticated.

Building a semantic network will enable to build tests of another level that will test students' understanding of which concepts are basic are which are subsidiaries of the basic concepts. Based on such a network, a hierarchy of terms can be built. With such hierarchy, tests are generated. One can safely say that a semantic network significantly diversifies overall test assignments and their 'human' nature. An illustrative example of a semantic web for construction of tests can be shown in the medical field. Having constructed a semantic network, you can generate questions similar to those described in [11], but fully automated - where a set of symptoms serves as questions and the test-taker is required to identify the disease, or the reverse case, where there are multiple-choice tests and the test taker is expected to identify the symptoms characteristic of the disease given. Similar tests can be built in almost any subject field.

Another line of study that can be explored is automated model construction based on examples found during text analysis. In this case, computer-aided learning is considered. The system analyses the text material and finds examples in it. Based on the examples found, the system builds laws for output data retrieval from input data. Then, tests similar to the model tests are generated.

4. CONCLUSIONS

The first tests showed that a system designed, based on the principles described, operates and makes sense. The system selects terms and definitions from the original text, the analyser requires some refinement to screen out debris and search for other formal structures. But in general, the approach allows retrieving the necessary information. Based on the collected glossary and formal model, we were able to generate a variety of tests. For 17 terms, 70 different unique tests were generated for one person (Figure-3). Expanding the set of construction models can increase this number of tests. Construction of tests by properties and classification isn't factored in. The resulting tests are quite adequate and allow to completely replace manual labour for common terminological tests. Test 'hominization' can be extremely diverse. The built system can completely cover generation of tests that focus on checking theoretical knowledge, and some tests that checks application of practical skills.

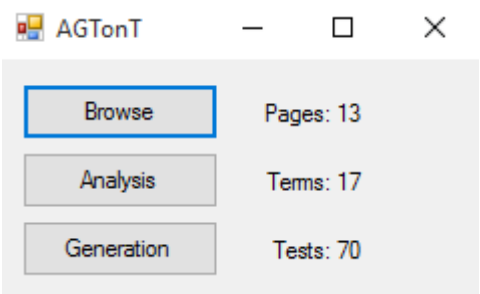

Figure-3. Program prototype interface.

There are presently no ready-to-use automated test generation solutions, whereas there are so many testing systems in the market. Filling up this white spot in the test automation industry is an important task.

REFERENCES

[1] Moshkov I.S. 2012. Sistemnyj analiz tekstovogo predstavleniya taksonomii i razrabotka modelej dlya ocenki professionalnyx znanij s ispolzovaniem testov otkrytogo tipa: avtoreferat dis. kandidata texnicheskix nauk - Samara. p. 22. [in Russian].

[2] Sergushicheva A. P. 2007. Metod i algoritmy avtomatizirovannogo postroeniya kompyuternyx testov kontrolya znanij po texnichiskim disciplinam: avtoref. disc. na soiskanie nauch. stepeni kand. texn. Nauk. p. 18. [in Russian].

[3] Michael Heilman. 2011. Automatic Factual Question Generation from Text. - Language Technologies Institute School of Computer Science Carnegie Mellon University. 195 c.

[4] Fairon C. *et al*. 2002. Automatic item text generation in educational assessment //Proceedings of TALN. pp. 395-401.

[5] Gierl M. J., Haladyna T. M. 2012. Automatic item generation: Theory and practice. - Routledge.

[6] Anton Arzhakov. 2014. Sergey Popov Obrazovanie v epohu informatizacii. - LAP LAMBERT Academic Publishing. p. 169. [in Russian].

[7] Belous V. V., Karpenko A. P., Domnikov A. S. 2011. Testovyj metod kontrolya kachestva obucheniya i kriterii kachestva obrazovatelnyx testov. obzor //nauka i obrazovanie: elektronnoe nauchno-texnicheskoe izdanie. №. 04. [in Russian].

[8] Chelyshkova M. B. Teoriya i praktika konstruirovaniya pedagogicheskix testov. - Logos, 2002. [in Russian].

[9] Yakubovskaya K. A. SEMANTIChESKIJ ANALIZ TEKSTA //Nauchno-texnicheskaya mezhdunarodnaya molodezhnaya konferenciya «Sistemy, metody, texnika i texnologii obrabotki mediakontenta». p. 122. [in Russian].

[10] Bolotnova N. 2015. Filologicheskij analiz teksta: uchebnoe posobie. - Litres. [in Russian].

[11] Gierl M. J., Lai H., Turner S. R. 2012. Using automatic item generation to create multiple-choice test items //Medical education. T. 46. - №. 8. pp. 757-765.