A System for Generating Cloze Test Items from Russian-Language Text

Andrey Kurtasov
Vologda State Technical University
Russia
akurtasov@gmail.com

Abstract

This paper studies the problem of automated educational test generation. We describe a procedure for generating cloze test items from Russian-language text, which consists of three steps: sentence splitting, sentence filtering, and question generation. The sentence filtering issue is discussed as an application of automatic summarization techniques. We describe a simple experimental system which implements cloze question generation and takes into account grammatical features of the Russian language such as gender and number.

1 Introduction

In recent years, e-learning has become a widely used form of post-secondary education in Russia. Highly-developed Learning Management Systems (LMS), such as Moodle[1], have been broadly accepted by Russian colleges and universities. These systems provide rich opportunities for delivering, tracking and managing education, and significantly help to reduce a teacher's workload as well as to establish distance learning. One of the most noticeable functions provided by the LMSs is assessment, which is implemented through automated tests. However, the task of preparing questions for the tests is not yet automated. The teacher has to compose all the test items manually, and this is a time-consuming task.

Moodle allows using different types of test items for student assessment, including calculated questions, multiple-choice questions, matching questions, and questions with embedded answers, also known as *cloze* questions or *fill-in-the-blank* questions.

We are considering the opportunity for automated test generation based on extracting sentences from electronic documents. We find this approach promising, because electronic textbooks are widely used, and many texts with potential educational value are available through the Internet. As a starting point, we aim to study methods for generating cloze questions, because they are obviously the easiest to be produced from sentences. To produce a cloze question, one takes a sentence and replaces some of the words in the sentence with blanks.

Once we have studied how to extract useful sentences from the text and how to select words to blank out, we will continue our research in order to develop methods for generating more complicated types of test items, such as multiple-choice.

2 Related Work

The idea of automating the composition of test items is not new.

For instance, several Russian authors, including Sergushitcheva and Shvetcov (2003) and Kruchinin (2003), suggest using formal grammars (FG) to generate test questions with variable parts. Although the development of FG-based templates is performed manually, this approach allows generating multiple various tests of different types (including multiple-choice) and eliminates students' cheating.

The approach of generating test items by extracting sentences from electronic documents has received significant attention in English-language literature. Several authors have considered different kinds of test items in terms of automation. For instance, cloze questions were studied

[1] Available from: https://moodle.org/

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 107–112,
Hissar, Bulgaria, 9-11 September 2013.

by Mostow et al. (2004) for the purpose of reading comprehension assessment. Mitkov et al. (2006) implemented an environment that allows producing multiple-choice questions with distractors. Heilman (2011) developed a system for generating wh-questions that require an answer to be typed in.

However, only a few authors have considered this approach for Russian. Voronets et al. (2003) published one of the first papers on the topic, in which they proposed applying this approach to instructional texts used in cosmonaut training. Sergushitcheva and Shvetcov (2006) considered using this approach in combination with the FG-based one.

3 Workflow for Computer-Assisted Test Generation

Our idea is to establish a system that delivers computer-assisted test authoring and leverages Natural Language Processing (NLP) techniques to provide the teacher with test items, which are generated automatically from electronic textbooks or similar texts. After the generation the test can be passed to the Moodle LMS and used for student assessment.

Fig. 1 shows the basic workflow of the system, which could be considered as a computer-assisted procedure. The system takes a text as an input and produces test items as the output. The test items are then presented to teachers, who select and edit the ones that they consider useful.

4 Text Processing

The approach is based on sequential application of linguistic processors that perform the following tasks on the text:

- Sentence splitting – to acquire sentences from which the system will produce questions for test items
- Sentence filtering – to filter the set of sentences so that it contains the most salient sentences
- Question generation – to convert the sentences into questions

4.1 Sentence Splitting

At first sight, a sentence is a sequence of characters that ends with ".", "!" or "?". In practice we should keep in mind that these characters can also be used inside one sentence (Grefenstette and Tapanainen, 1994). To address this issue, we initially used a simple tokenization algorithm that had been developed for educational purposes. It took into account abbreviations, proper name initials and other special cases. For instance, the algorithm recognized commonly used Russian abbreviations containing periods, such as "г." (year), "гг." (years), "и т. д." (etc.), "т. е." (i.e.), "т. н." (so called), "напр." (e.g.) and so on.

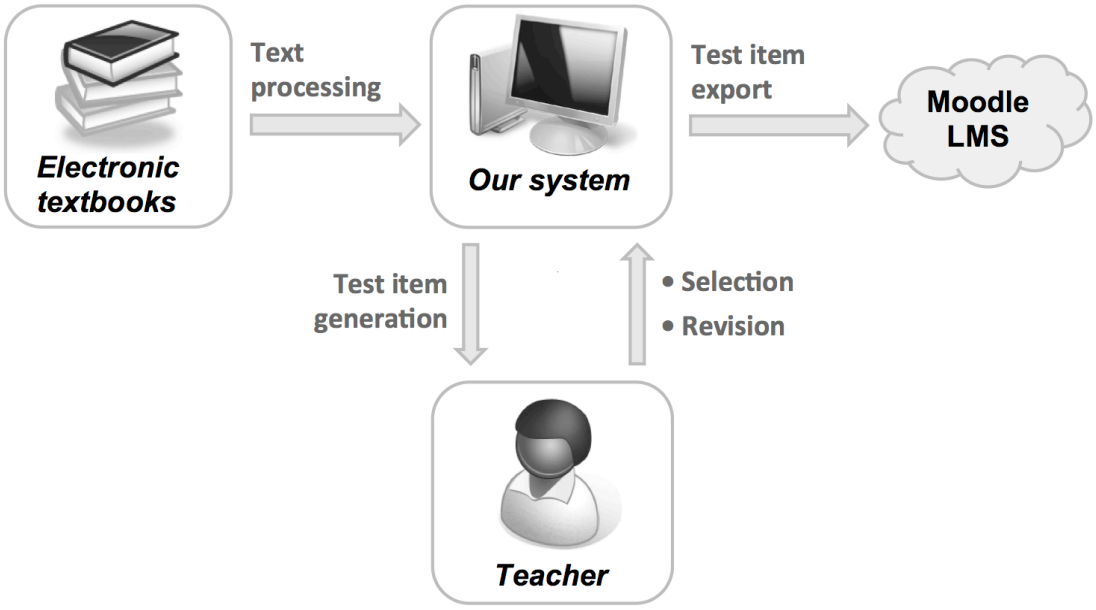

Figure 1: Workflow

In the current system, we use a tokenization module provided by the AOT toolkit[2]. It takes into account more text features including bulleted lists, sentences enclosed in quote marks or parentheses, URLs and mail addresses. In practice, it performs sentence splitting with fairly high precision, therefore this step of text processing does not introduce a significant number of errors in the resulting test items.

4.2 Sentence Filtering

It is obvious that not every sentence acquired from a text is appropriate for question generation. Therefore, we suppose that the sentence set could be filtered in order to provide better results. Reducing a text document in order to retain its most important portions is known as document summarization.

The NLP field studies different techniques for automatic text summarization, with two general approaches: extraction and abstraction. Extractive summaries (extracts) are produced by concatenating several sentences taken exactly as they appear in the materials being summarized. Abstractive summaries (abstracts), are written to convey the main information in the input and may reuse phrases or clauses from it, but the summaries are overall expressed in the words of the summary author (Nenkova and McKeown, 2011). It means that abstracts may contain words not explicitly present in the original.

In our task, the main goal is removing unimportant sentences, therefore we can use extraction-based summarization. Generally, we need to assign an importance score to each sentence and include the highest-scoring sentences in the resulting set. Since 1950s, different methods for scoring sentences have been studied, and they are now usually applied in combination. For example, Hynek and Jezek (2003) listed the following methods: sentence length cut-off (short sentences are excluded), use of cue phrases (inclusion of sentences containing phrases such as "in conclusion", "as a result" etc.), sentence position in a document / paragraph, occurrence of frequent terms (based on TF-IDF term weighting), relative position of frequent terms within a sentence, use of uppercase words, and occurrence of title words.

To date, we have not completed our research in this direction, and we use an unfiltered set of sentences in the current system. However, at the question generation stage we apply some rules that allow selecting sentences of a particular structure, e.g. those containing definitions or acronyms.

4.3 Question Generation

Our current approach uses different algorithms to generate questions for a cloze test. We also take into account the category of the blanked-out word and add a hint into the question, explaining what kind of answer is expected. The algorithms can be divided into two groups depending on how deeply the sentence is analyzed.

The algorithms of the first group simply read the sentence as a sequence of characters looking for acronyms, numbers or definitions. Definitions are recognized based on the common words used to define a term in Russian, such as "является" (is), "представляет собой" (represents) or the combination of a dash and the word "это" (a Russian particle commonly used in definitions; usually preceded by a dash). Below is an example sentence followed by a generated question:

Source: Сеть — это группа из двух или более компьютеров, которые предоставляют совместный доступ к своим аппаратным или программным ресурсам.

Result: .... (определение) — это группа из двух или более компьютеров, которые предоставляют совместный доступ к своим аппаратным или программным ресурсам.

Or, in English:

Source: A network is a group of two or more computers that provide shared access to their hardware or software resources.

Result: ..... (definition) is a group of two or more computers that provide shared access to their hardware or software resources.

The system recognized a sentence containing a definition and replaced the term "Сеть" (network) with a blank. After the blank, it inserted a hint in parentheses: "определение" (definition).

The next example shows how the system can process numbers:

Source: Как известно, классическая концепция экспертных систем сложилась в 1980-х гг.

[2] Available from: http://aot.ru/

Result: Как известно, классическая концепция экспертных систем сложилась в ..... (число)-х гг.

Or, in English:

Source: As is well known, the classical conception of expert systems has developed in 1980s.

Result: As is well known, the classical conception of expert systems has developed in ..... (number)s.

The system recognized a sentence containing a number (1980) and replaced it with a blank. After the blank, it inserted a hint in parentheses: "число" (number). The teacher can edit this question by removing the cue phrase ("Как известно" — "As is well known") and moving the hint to a better position.

The algorithms of the first group are fairly easy to implement and perform relatively fast.

The algorithms of the second group generate questions based on morpho-syntactic analysis of a sentence. They allow producing questions to the sentence's subject ("что?" — "what?"; "кто?" — "who?"), adverbial of place or time ("где?" — "where?"; "когда?" — "when?"), or to adjectives contained in the sentence ("какой?" — "what?"). To perform the morpho-syntactic analysis, we use the AOT toolkit. It helps to define proper hints for the questions, considering the gender and number of the blanked-out word. For example:

Source: В отличие от перцептронов рефлекторный алгоритм напрямую рассчитывает адекватную входным воздействиям реакцию интеллектуальной системы.

Result: В отличие от перцептронов ..... (какой?) алгоритм напрямую рассчитывает адекватную входным воздействиям реакцию интеллектуальной системы.

Or, in English:

Source: In contrast to perceptrons, the reflective algorithm directly calculates the reaction of the intelligent system with respect to input actions.

Result: In contrast to perceptrons, the ..... (what?) algorithm directly calculates the reaction of the intelligent system with respect to input actions.

The system recognized an adjective ("рефлекторный" — "reflective") and replaced it with a blank. After the blank, it inserted a hint in parentheses: "какой?" ("what?").

These algorithms are more complicated than those of the first group and perform slower.

One of the issues, which arise at the question generation stage, is that the current system does not attempt to determine whether blanking out a particular word produces a useful question, which results in a number of superfluous questions that the teacher has to reject manually.

5 Preliminary Experiments

Even though sentence filtering is not yet implemented, our preliminary experiments show that the system may produce relatively fair results with certain text documents. For initial assessment of the system, we tried generating questions for a Russian-language textbook on intelligent information systems. A human judge was asked to classify the resulting questions into 3 categories: *ready to use*, *correctable*, and *useless*.

About 40% of the questions generated with the algorithms of the second group were *ready to use* in a test without modification. It means a teacher would not have to edit the questions by removing superfluous words, replacing pronouns with corresponding nouns etc. About 23% of the questions were *correctable*, i.e. they could be used in a test after some manual correction.

The algorithms of the first group were not as effective (about 15% of generated questions were either *ready to use* or *correctable*), but we expect them to be more effective with texts that contain many explicit definitions (e.g. glossaries) or numbers (e.g. books with history dates).

We also tested the running time of the algorithms on different hardware configurations (from a netbook to a powerful state-of-the-art workstation). The second group algorithms, due to their relative complexity, performed significantly slower than those of the first group, even with short texts. However, it never took more than three minutes to generate questions for an average size textbook (about 250 pages) using any of the algorithms (including sentence splitting time).

6 Conclusions and Future Work

We have done preliminary research regarding two methods for generating test items from electronic documents. We have developed a simple experimental system that allows a teacher to

generate questions for a cloze test. Test authoring in the system is presented as a computer-assisted procedure. The system proposes the generated test items to the teacher who selects and edits the ones that are appropriate for use in the test, and then the test is passed to the Moodle LMS. An advantage of the system is that it is specifically developed for the Russian language and it processes texts with respect to morpho-syntactic features of the language, e.g. it can recognize a sentence's subject.

According to initial experiments, the current system performs fairly well in particular cases. However, we have discovered a number of complex problems that should be assessed and addressed in the near future:

1. It may be difficult for the teacher to select useful items. There are at least two ways to address this issue:
    a. If we implement text summarization, the system will be able to produce test items from the most salient sentences of the text.
    b. We should develop a method for selecting the appropriate words to blank out. One idea is to apply a glossary of domain-specific terms to identify such terms in each sentence. We assume that it is more useful to blank out special terms than common words.
2. In order to reduce the need in manual post-editing of the questions, we should consider the following:
    a. Processed sentences may contain anaphora. If the current system uses such a sentence to generate a test item, the teacher has to resolve the anaphora manually (e.g. to replace pronouns with corresponding nouns). Therefore we should study ways of automatic anaphora resolution, which could be implemented in the system.
    b. It might be useful to remove common cue phrases while performing sentence splitting.
3. Fill-in-the-blank is a trivial style of test. Using this kind of exercise in Moodle may be ineffective, because Moodle will only recognize answers that exactly match the blanked-out word. Therefore, we should consider ways to generate distractors in order to establish multiple choice testing.
4. Comprehensive experiments should be conducted:
    a. We should use a representative selection of text sources to substantially evaluate the portion of useful test items that the system is able to produce.
    b. We should assess how the approach compares against people identifying test items without the system, with respect to consumed time and difficulty of the test items. Our suggestion is to involve a group of human judges to annotate questions as useful or not.

These problems define the main directions for future work.

Acknowledgments

The author thanks his advisor, Prof. Dr.sc. Anatolii N. Shvetcov, for his guidance and support, and the anonymous reviewers for their insightful remarks. This research is still at the preliminary stage, therefore any feedback or advice is much appreciated.

References

Gregory Grefenstette, Pasi Tapanainen. 1994. What is a Word, what is a Sentence? Problems of Tokenisation. *Proceedings of 3rd conference on Computational Lexicography and Text Research*, Budapest, Hungary.

Michael Heilman. 2011. *Automatic Factual Question Generation from Text*. Ph.D. Dissertation, Carnegie Mellon University.

Jiri Hynek, Karel Jezek. 2003. A practical approach to automatic text summarization. *Proceedings of the ELPUB 2003 conference*, Guimaraes, Portugal.

Vladimir V. Kruchinin. 2003. *Generators in Programs for Computer-based Training.* Izdatelstvo Tomskogo Universiteta, Tomsk, Russia (В. В. Кручинин. 2003. *Генераторы в компьютерных учебных программах*. Издательство Томского университета, Томск, Россия) [in Russian].

Ruslan Mitkov, Le An Ha, Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2): 1–18.

Jack Mostow, Joseph Beck, Juliet Bey, Andrew Cuneo, June Sison, Brian Tobin and Joseph Valeri. 2004. Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial

interventions. *Technology, Instruction, Cognition and Learning* (2): 97–134.

Ani Nenkova and Kathleen McKeown. 2011. Automatic Summarization. *Foundations and Trends in Information Retrieval*, 5(2–3): 103–233.

Anna P. Sergushitcheva, Anatolii N. Shvetcov. 2003. Synthesis of Intelligence Tests by means of a Formal Production System. *Mathematics, Computer, Education: Conference Proceedings*, vol. 10 (1): 310–320. R&C Dynamics, Moscow – Izhevsk, Russia (А. П. Сергушичева, А. Н. Швецов. 2003. Синтез интеллектуальных тестов средствами формальной продукционной системы. *Математика, Компьютер, Образование. Сборник научных трудов*, выпуск 10 (1): 310–320. R&C Dynamics, Москва – Ижевск, Россия) [in Russian].

Anna P. Sergushitcheva, Anatolii N. Shvetcov. 2006. The Hybrid Approach to Synthesis of Test Tasks in Testing Systems. *Mathematics, Computer, Education: Conference Proceedings*, vol. 13 (1): 215–228. R&C Dynamics, Moscow – Izhevsk, Russia (А. П. Сергушичева, А. Н. Швецов. 2006. Гибридный подход к синтезу тестовых заданий в тестирующих системах. *Математика, Компьютер, Образование. Сборник научных трудов*, выпуск 13 (1): 215–228. R&C Dynamics, Москва – Ижевск, Россия) [in Russian].

I. V. Voronets, Anatolii N. Shvetcov, Viktor S. Alyoshin. 2003. A Universal Automated System for Knowledge Assessment and Self-teaching based on Analysis of Natural-language Texts of Textbooks. *Proceedings of the 5th International Scientific and Practical Conference "Manned Spaceflight"*, Star City, Moscow Region, Russia (И. В. Воронец, А. Н. Швецов, В. С. Алешин. 2003. Универсальная автоматизированная система тестирования знаний и самообразования, основанная на анализе естественно-языковых текстов учебных пособий. *Сб. докл. Пятой международной научно-практической конференции "Пилотируемые полеты в космос"*, Звездный Городок, Москва, Россия) [in Russian].