Automated Generation of Assessment Test Items from Text: Some Quality Aspects

Andrey Kurtasov

Vologda State University, Vologda, Russia
akurtasov@gmail.com

Abstract This paper overviews the problem of automated generation of assessment test items from natural-language text. In a previously published article, an experimental system aimed at generating fill-in-the-blank test items from Russian text was decribed. In this paper, some aspects of the system's quality are analyzed. Main directions for future work are defined, including evaluation of the system and development of methods for filtering text fragments and selecting words to blank out.

Key words: educational assessment, natural language processing, Russian language, test item generation, question generation.

1 Introduction

The teaching process of today widely uses electronic text resources that were not originally intended for use as teaching aids. This is especially true for subjects that deal with rapidly developing domains such as information technology. Teaching these subjects may benefit from use of various articles and technical papers, which do not contain test questions or exercises, as opposed to textbooks. Developing the exercises is a complex task that may require a teacher to spend a significant amount of time on. A promising way to facilitate this task is automated generation of test items from text with the help of Natural Language Processing (NLP).

The general idea is to extract fragments from the source text document and to transform them into questions or test items. This idea has been studied by several researchers, and is commonly considered difficult to implement. For instance, Heilman [1] has discovered numerous challenges in question generation from text. These include linguistic challenges (lexical, syntactic, discourse-related) as well as various challenges related to the application of question generation tools in classrooms (usability, human-computer interaction issues).

Previously, we have described an experimental system for generating fill-in-the-blank test items from Russian-language text, which is designed for use with the e-learning platform Moodle[1] [2]. We have showed that the automated generation of test items is not accomplished easily, but can yield some useful results. In this paper, we are going to review the quality aspects of the approach being studied and consider ways to improve it.

[1] https://moodle.org/

2 Test Item Generation: Approach and Quality Aspects

We present the approach as the sequential application of text processors that perform the following operations on the document:

1. Text preprocessing — to convert a raw text file into a well-defined sequence of linguistically meaningful units (as defined in [3]), or segments
2. Segment filtering — to filter the set of segments so that it contains the most salient segments
3. Test item generation — to transform the text segments into test items

Let us consider each of the operations from the quality perspective.

2.1 Text Preprocessing

This operation consists of two stages: document triage and text segmentation. Document triage is the process of converting a digital file into a well-defined text document. It involves such actions as character encoding identification and text sectioning (identificating the actual content within a file while discarding headers, links, and formatting features). This stage is solely technical and easy to accomplish with available software tools. However, it could crucially affect the results (e.g. improper encoding detection would make the Russian text unreadable), and should be a significant concern to the software developers.

Text segmentation is performed to acquire segments from which to produce test items. Previously we referred to this stage as sentence splitting, because we use sentences as basic segments, while considering a sentence to be a semantically complete portion of text. At first sight, a sentence is a sequence of characters that ends with ".", "!", or "?", but in practice one should keep in mind that these characters can also be used inside one sentence [4]. Today's NLP tools perform sentence splitting with fairly high precision. In preliminary experiments we used a tokenization module provided by the AOT toolkit[2], which recognizes common Russian abbreviations with periods, such as "г." (year), "гг." (years), "и т. д." (etc.), "т. е." (i.e.), "т. н." (so called), as well as special text features such as bulleted lists, sentences enclosed in quotation marks or parentheses, and URLs. The experiments have shown that this step does not introduce a significant number of errors in the resulting test items.

In some cases, it may be reasonable to include more than one sentence in a segment (when multiple sentences are used to express one significant thought). While automatic detection of such sentence groups is a complex semantics-related task, we assume that the user should be given an ability to see the context of the processed sentence at the test item generation step. This ability would allow the user to expand the segment if needed, and should be considered for implementation during the user interface design of the generating software.

[2] `http://www.aot.ru/`

2.2 Segment Filtering

It is obvious that not every text sentence is appropriate for test item generation. We assume that proper filtering of acquired sentences could have a convincing impact on the quality of the resulting test items set, and we propose using extractive text summarization to filter out the unneccessary text portions. In NLP, different methods for scoring sentences by importance are applied (usually in combination) [5]: sentence length cut-off (short sentences are excluded), use of cue phrases (inclusion of sentences with phrases such as "in conclusion"), sentence position in a document/paragraph, occurrence of frequent terms (based on TF-IDF term weighting), and occurrence of title words.

We are planning to leverage an existing summarization toolkit and attempt taming it for our task. For example, MEAD[3] is claimed to be modifiable to support languages other than English. Similarly to the text segmentation, it would be reasonable to show the highest-scoring sentences inline, so that the user could see the discarded portions and use them if they appear to be useful.

The performance of this step is to be evaluated experimentally. We are planning to compare the summarization output with the selection made by human experts and calculate such metrics as precision and recall (commonly used in informational retrieval).

2.3 Test Item Generation

As a starting point of the research, we generate fill-in-the-blank test items ("cloze questions"). To produce a cloze question, we take a sentence and replace some of the words in the sentence with blanks. For additional clarity, we add a hint into the question, explaining what kind of answer is expected. Below is an 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"), replaced it with a blank, and inserted a hint in parentheses: "какой?" ("what?"). Also, the current system is able to add appropriate hints for acronyms, numbers, definitions, sentence subjects, adverbials (more examples were shown in [2]).

[3] `http://www.summarization.com/mead/`

The main problem here is to determine which words should be blanked out to produce a useful question. A good approach could be finding the sentence's focus (in the sense of information structure), which is difficult to do with the state-of-the-art NLP tools. Another idea is based on the assumption that it is more useful to blank out special terms than common words. We could match words of the sentence against either a pre-existing domain-specific bag of words or a bag of words acquired through terminology extraction from the processed text, and blank out the matches.

Another issue, which arises at this step, is that the processed sentences may contain anaphora. Without an implementation of automatic anaphora resolution, the user could resolve the anaphora manually (e.g. to replace pronouns with corresponding nouns) using the in-context display of the processed sentence.

While cloze items are fairly easy to produce from sentences, fill-in-the-blank is a trivial style of test. This concern could be addressed by considering the two ideas: generation of interrogative sentences (it would require text simplification and word reordering [1]) and generation of distracting answers for multiple-choice tests (a possible solution is described in [6]).

3 Conclusion and Future Work

Based on the preceding research, we have considered several quality aspects of the automated generation of assessment test items from natural-language text. We have discovered the following directions for quality improvement of our system:

1. The user interface should display the context of the text excerpt being processed in a user-friendly way for efficient human-computer interaction.
2. We will leverage a summarization toolkit for segment filtering and evaluate it experimentally.
3. Other directions include anaphora resolution, interrogative sentence generation, and distractor generation for multiple-choice tests.

References

1. *Heilman, M.* Automatic Factual Question Generation from Text. Ph.D. Dissertation. — Carnegie Mellon University, Pittsburgh, USA, 2011. 195 p.
2. *Kurtasov, A.* A System for Generating Cloze Test Items from Russian-Language Text / In Proceedings of the Student Research Workshop associated with The 9th International Conference RANLP 2013. P. 107–112. — Hissar, Bulgaria, 2013.
3. *Indurkhya, N.; Damerau, F. J. (eds).* Handbook of Natural Language Processing (Second Edition). — Chapman and Hall/CRC, 2010. 704 p.
4. *Grefenstette, G.; Tapanainen, P.* What is a Word, what is a Sentence? Problems of Tokenisation / In Proceedings of The 3rd Conference on Computational Lexicography and Text Research. — Budapest, Hungary, 1994.
5. *Hynek, J.; Ježek, K.* Practical approach to automatic text summarization. / In Proceedings of the ELPUB 2003 conference. — Guimaraes, Portugal, 2003.
6. *Mitkov, R., Ha, L., Karamanis, N.* A computer-aided environment for generating multiple-choice test items // Natural Language Engineering. 2006. 12(2). P. 1–18.

Автоматизированная генерация тестовых заданий для проверки знаний: некоторые аспекты качества

Андрей Куртасов

Вологодский государственный университет, Вологда, Россия
akurtasov@gmail.com

Аннотация В работе приведен обзор задачи автоматизированной генерации тестовых заданий для проверки знаний из текста на естественном языке. В ранее опубликованной статье была описана экспериментальная система для генерации заданий на заполнение пропусков из русскоязычного текста. В данной работе проанализированы некоторые аспекты качества работы системы. Определены основные направления для дальнейшей работы, включая оценку системы и разработку методов фильтрации текстовых фрагментов и выбора слов для замены на пропуски.

Ключевые слова: оценка знаний в образовании, автоматическая обработка текста, генерация тестовых заданий, генерация вопросов.