Automatic Limited-Choice and Completion Test Creation, Assessment and Feedback in modern Learning Processes

Christian Gütl

Institute for Information Systems and Computer Media,
Graz University of Technology, Inffeldgasse 16c, 8010 Graz, Austria
and
Infodelio Information Systems and GÜTL IT Research & Consulting, Austria
cguetl@iicm.edu and cguetl@acm.org

Abstract. The objectives and expectations for learning processes have changed dramatically from remedial repetitive learning to today's learning with understanding to become independent in the learning process. Approaches such as self-directed, explorative or example-based learning not only need advanced e-learning solutions for the knowledge transfer process, but also require enhanced methods for knowledge assessment and feedback. Motivated by the situation stated so far, we have started a research program to support knowledge assessment and feedback by computer-based automatic and peer assessment. In this paper we will focus on the automatic generation of limited-choice and completion questions from learning content, the answer evaluation and the feedback provision.

Keywords: automatic knowledge assessment, automatic feedback, automatic summarization, limited-choice test, natural language processing

1 Introduction

At the beginning of the 21st century our modern life is strongly influenced by effects such as rapidly changing and developing information, technology-enhanced communication and information access, and new forms of production and services in a globalized world. Consequently, according to [1], the objectives and expectations of learning processes have changed dramatically over the last 100 years, from remedial repetitive learning to today's learning with understanding to become independent in the learning process, strengthen metacognitive skills and link knowledge in cultural context. Adopted instructional methods and learning styles need to be applied in such modern learning settings. New research and development strands in e-learning attempt to support these new learning-teaching processes. Approaches such as self-directed, explorative or example-based learning not only need advanced e-learning solutions for the knowledge transfer process, but also require enhanced methods for knowledge assessment and feedback. The later will be used to get a more fine-grained user model, allows instructors to get a better overview over the state of the learning

process, and supports students do improve their learning activities and strategies. To cope with this requirements, an e-learning system must (1) deal with a great variety of learning tasks and diverse learning content for individual learners, and (2) based on that must also support continuous knowledge assessment according to the instructional strategy and (3) provide immediate feedback.

Computer-assisted and computer-based assessment is indeed not a new concept and at least can be traced back to early systems in the 1960s which followed the macro-adaptive approach, such as computer-managed instructions (CMI) systems. Assessment components have also been an essential part of Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS) [2]. To cope with different pedagogical goals, computer-based assessment developments offer different assessment approaches such as limited-choice tests, short natural answer tests, open-ended questions and essays. Automated assessment – in particular of short natural answers and essays - has been an active research topic since years. An overview over research approaches and performance can be found elsewhere in [3]. Related to our implementation discussed in this paper, Mitkov & Ha describe an interesting approach for automatic, or at least computer-aided, generation of multiple-choice tests from digital learning content [4].

Motivated by the situation stated so far, we have started a research program to support knowledge assessment and feedback by computer-based automatic and peer assessment. Our overall architecture is designed (1) to administer different didactic objectives by supporting (semi-)automatic question generation, (2) to support the (semi-)automatic generation and management of various exercise types, (3) to enable the compilation and performance of personalized tests, (4) to execute (semi-) automatic assessment, and (5) to deliver information for updating user models and providing feedback.

In this paper we will focus on the automatic generation of limited-choice and completion questions out of learning content, the answer evaluation and the feedback provision. The remainder of the paper is structured as follows. Application scenarios and high level requirements are given in Section 2. The technical solution of our assessment tool, an overview about the architecture and our first implementation are outlined in Section 3. Finally, Section 4 discusses major findings and room for further improvements.

2 Application Scenarios and Requirements

In the planning stage of the research project, various application scenarios have been identified in which such an assessment tool can support teaching and learning activities in modern learning environments. Interesting application scenarios include but are not limited the following: (1) Frank teaches a course in basic economics at university level and he wants to get support to compile personalized tests referring to the individual assignments of the individual students. (2) Miriam is senior lecturer in the computer science department and she has decided to provide extensive background material for her courses by an e-learning platform. She wants the students to focus exemplarily on selected topics. After completion of the learning task, the

assessment of knowledge acquisition is performed by automatically generated completion tests, feedback is provided to the students and to Miriam as well. (3) Eric prefers to study on his own but he has some weakly developed meta-cognitive capabilities. Thus he needs continuous feedback about his knowledge state in the learning process and based on that some guidance in his learning activities. After finishing learning activities, he initiates the performance of automatically generated completion tests and benefits from the immediate feedback given by the tool.

Based on the literature review, our objectives for a (semi-)automatic tool for exercise generation, assessment performance and feedback outlined above and illustrated by some selected application scenarios, most important requirements on an abstract level can be stated as follows: (1) flexible tool to be applicable in various application scenarios, (2) handling of different types of content for exercise generation from pre-defined learning content to background knowledge of diverse sources to student assignments, (3) designed to be used as module in pre-existing learning environments and as stand alone application, and (4) privacy and security.

3 Technical Solution and Implementation Details

On a glance, our approach uses automatic summarization methods to extract the most important information from content "units" used in learning tasks. Based on this extracted and condensed information, natural language processing methods are applied to (1) identify relevant concepts related to the knowledge domain, and to (2) tag and annotate part of speech. This processed information will be used to generate the limited-choice and completion questions.

For generating completion questions, identified concepts or candidate words in specific relations are used, such as *concept a* "is defined as", *concept b* "comprises" *concept c* and *concept d*, and the like. Some of these identified concepts are omitted to compile completion exercises. For generating limited-choice exercises, for each of the aforementioned concepts synonyms and antonyms are computed. Out of this set of concepts answer alternatives will be processed.

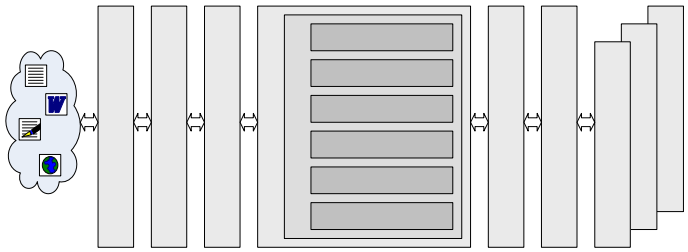

Fig. 1. Architecture overview for (semi-)automatic limited-choice and completion test creation, assessment and feedback

Fig. 1 outlines the architecture overview of our solution, which is entirely implemented in Java. In order to deal with different document types and sources, the *Document Fetcher* supports the access to documents by hypertext transfer protocol

and in the file system. To be flexibly applicable as module in preexisting learning systems, a specific interface call enables other components to send specific content for processing. The *Document Filter* converts various document formats (such as PDF, HTML and Word) into an internal format comprising content and some structure information (headings, title and the like) which will be used for the further processing steps.

The automatic summarization is based on the *Automatic Summarizer* from [5] which *"benefits from a mixture of ideas founded in the early days of automatic summarization."* Unlike other modern approaches, the summarization is simply built by identifying the most important sentences of the content. This is computed by 5 statistical features, namely (1) *word frequency method* (significant content words in neighborhood within sentences), (2) *cue phrase method* (phrases which indicate the importance of a sentence for a summary, such as "the conclusion is"), (3) *location method* (appearance of topic sentences tend to appear at the beginning or the end of sections), (4) *title method* (subject matter is addressed by the title), (5) query method (summary content can be defined by a query which reflects the user's interest). Each of the aforementioned features can be weighed positively or negatively, which provide together with the compression ration (length of the origin text compared to the summary) much room for flexible and tailored summaries. Further details about automatic summarization can be found elsewhere in [6].

In order to identify relevant concepts to be used for completion and limited choice exercises, further natural language processing is performed by the GATE (General Architecture for Text Engineering) tool using appropriate plug-ins [7]. For our first implementation, we focus instead of concepts on nouns, verbs, adjectives and on their combinations for creating exercises. The automatically generated summary is preprocessed by the *Tokeniser* and *Sentence Splitter* and the part of speech is annotated by the *POS Tagger*. Nouns, verbs, adjectives are further processes to find the basic form of the words as well as synonyms and antonyms by applying the *Morphological Analyzer*. In the next step of the processing chain, the *Stopword Finder* eliminates function words and the like. Moreover, predefined words or concepts describing subject domains can also be applied to extract important words from the automatic summary. Based on the aforementioned process steps, a list of candidate words applicable for the exercise creation is processed by the *Test Word Extractor*.

The automatically generated abstract together with the set of candidate words, synonyms and antonyms is delivered to the *Exercise Creator*. It is designed either to create completion exercises or limited choice answers. Parameters not only control the type of exercise but also the number and type of concepts or words to be used for the tests. For completion exercises, the selected concepts and words will be annotated in the abstract for further processing or, to be more concrete, for visualizing the test and for comparing the reference answers with the students' answers. Similarly, limited choice answers are prepared by selecting proper words as well as synonyms and antonyms, which are used to be presented as answer alternatives in the assessment process.

Finally, the *Communication Interface* provides a layer to control the behaviors of the tool and to access the prepared exercises based on a specific content object. The high flexibility supports the tool's applicability in various application scenarios, as

illustrated in the outermost right side of **Fig. 1**. Thus our approach is designed to be used together with a user interface as a stand-alone application or as a module in other systems such as learning management systems or assessment systems.

Our current implementation is designed to be used as a stand-alone tool for experimental purposes and also includes a graphical user interface. It is fully implemented in Java and can be instantiated as an application or as an applet (see **Fig. 2**). In the first tab of the graphical user interface (GUI), the content of documents can be imported or even be pasted into the text field. In the second tab (see A in **Fig. 2**), based on compression rate and summarization method, automatic summarizations can be generated and inspected. Types of words and word lists to be used for the test creation can be configured in the third tab (see B in **Fig. 2**). In the fourth tab, the number of words out of the set of candidate words for any supported word type can be specified to create exercises randomly. Currently, the implementation only supports completion tests (see C in **Fig. 2**). Finally, in the last tab, the solution of created exercises can be inspected (see D in **Fig. 2**).

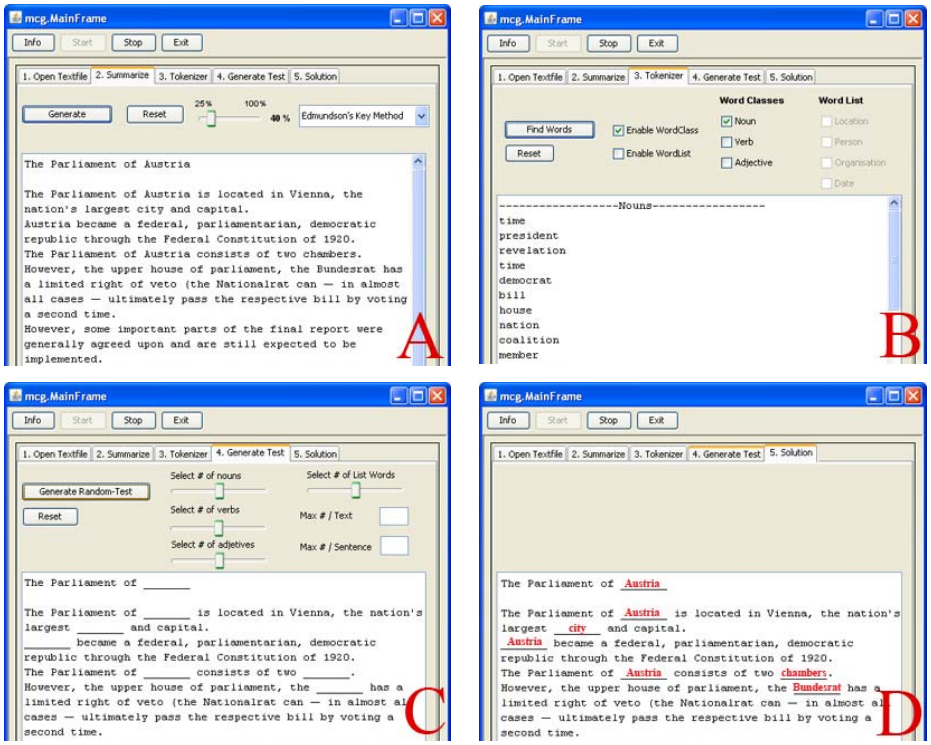

Fig. 2. Screen shot of the assessment prototype: (A) content summarization, (B), candidate concept extraction based on content and predefined lists (C) exercise creation, and (D) exercise solution for self-assessment.

4 Conclusions and Further Work

Modern learning settings require enhanced knowledge assessment procedures and tools which can deal not only with course content but also with related content and background material as well as provide feedback to students immediately. Automatic or at least semi-automatic assessment and feedback approaches can support modern learning environments. First experiences of our tool for automatic limited-choice and completion test creation, assessment and feedback are promising. First tests have also shown that the tools can easily be used as loosely coupled stand-alone applications or as integrative modules in pre-existing learning environments, such as the dotLRN platform.

Although our first promising results, there is much room for improvements. From the solution approach viewpoint we want to apply more advanced summarization techniques and more intelligent methods for the creation of exercises. From the implementation point of view, we want to redesign the tool that any of the modules can be used as plug-ins for the GATE system. Furthermore, for the exchange of exercises or tests we will support the IMS QTI standard.

References

1. Bransford, J.D., Brown, A.L., & Cocking; R.R. (Eds.) (2000). How People Learn: Brain, Mind, Experience, and School. Expanded Edition. Washington DC: National Academies Press.
2. Park, O., & Lee, J. (2003) Adaptive Instructional Systems. Educational Technology Research and Development, 2003(25). p. 651-684, 2003.
3. Gütl, C. (2007). Moving towards a Fully-Automatic Knowledge Assessment Tool. iJET International Journal of Emerging Technologies in Learning, to be published.
4. Mitkov, R., & Ha, L. (2003). Computer-aided generation of multiple-choice tests", Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, 2003, pp. 17-22.
5. Visser, W.T., & Wieling, M.B. Sentence-based Summarization of Scientific Documents. The design and implementation of an online available automatic summarizer. Report, last retrieved Nov. 29th, 2007 form http://home.hccnet.nl/m.b.wieling/files/wielingvisser05automaticsummarization.pdf
6. Mani, I., & Maybury, M.T. (Eds.) (1999). Advances in Automatic Text Summarization. MIT Press, Cambridge: MA.
7. GATE. Overview. Natural Language Processing Research Group, University of Sheffield, UK, last retrieved Nov. 29th, 2007 form http://gate.ac.uk/sale/gate-flyer/2007/gate-flyer-4-page.pdf

Acknowledgement

The author gratefully acknowledges Mario Ouschan and Christian Slamanig, both students at Graz University of Technology, for their development and implementation effort of the automatic assessment system, which contributed valuable input for this paper.