An Automated Approach to Checking User Knowledge Levels in Intelligent Tutoring Systems

Wenzu Li
Belarussian State University Informatics and Radioelectronics
Minsk, Belarus
lwzzggml@gmail.com

***Abstract*—This article is dedicated to the issue of automating the implementation of rapid testing of user knowledge in new generation intelligent tutoring systems. A semantic-based approach to automating the entire process from test question generation and test paper generation to the automatic verification of user answers and the automatic scoring of test papers is described in detail in this article.**

***Keywords*—testing user knowledge level, test question generation, user answer verification, intelligent tutoring systems, test paper generation, automatic scoring of test papers**

I. Introduction

Educators have long shared a common desire to use computers to automate teaching and learning services. In recent years, with the development of artificial intelligence technology, this wish is likely to become a reality. The most representative product combining artificial intelligence and education is the intelligent tutoring system (ITS), which can not only improve the learning efficiency of users, but also ensure the fairness and impartiality of the education process [9].

Automatic generation of test questions and automatic verification of user answers are the most basic and important functions of ITS. Using these two functions in combination will enable the entire process from the automatic generation of test questions to the automatic scoring of the user test papers. This will not only greatly reduce the repetitive work of educators, but will also reduce the cost of user learning, thus providing more users with the opportunity to learn various knowledge [1], [2], [7].

Although in recent years, with the development of technologies such as the semantic web, deep learning and natural language processing (NLP), several approaches have been proposed for the automatic generation of test questions and the automatic verification of user answers, these approaches have the following main disadvantages:

- existing approaches to generating test questions allow only the simplest objective questions to be generated;
- some existing approaches (for example, keyword matching and probability statistics) to verifying user answers to subjective questions do not consider the semantic similarity between answers;
- methods that use semantic to verify user answers to subjective questions can only calculate similarity between answers with simple semantic structures [7], [8], [10].

Objective questions usually have a unique standard answer. In this article, objective questions include: multiple-choice questions, fill in the blank questions and judgment questions. Objective questions differ from subjective questions, which have more than one potential correct answer. Subjective questions in this article include: definition explanation questions, proof questions and problem-solving task.

Therefore, based on existing methods and OSTIS Technology, an approach to developing a universal subsystem for automatic generation of test questions and automatic verification of user answers in tutoring systems developed using OSTIS Technology (Open Semantic Technology for Intelligent Systems) is proposed in this article [1], [2], [5]. The universality of the subsystem means that the developed subsystem can be easily transplanted to other ostis-systems (system built using OSTIS Technology). The developed subsystem allows the use of the knowledge bases of the ostis-systems to automatically generate various types of test questions and automatically verify the completeness and correctness of user answers based on the semantic description structures of the knowledge. The discrete mathematics ostis-system will be used as demonstration systems for the developed subsystem.

II. Existing approaches and problems

A. Automatic generation of test questions

Approach to automatic generation of test questions mainly studies how to use electronic documents, text corpus and knowledge bases to automatically generate test questions. Among them, the knowledge base stores highly structured knowledge that has been filtered, and with the development of semantic networks, using the knowledge base to automatically generate test questions has become the most important research direction in the field of automatic generation of test questions. The basics of how to use the knowledge base to automatically generate objective questions are described in detail in the literature [10], [12].

The main problems with the existing approaches to test question generation are as follows:

- the approach of using electronic documents to automatically generate test questions requires a large number of sentence templates;
- the creation of text corpus requires a lot of human resources to collect and process various knowledge;
- existing approaches only allow to generate simple objective questions [11], [13].

B. Automatic verification of user answers

Automatic verification of user answers is divided into verification of answers to objective questions and verification of answers to subjective questions. The basic principle of verification of answers to objective questions is to determine whether the string of standard answers matches the string of user answers. The basic principle of verification of answers to subjective questions is to calculate the similarity between standard answers and user answers, and then to implement automatic verification of user answers based on the calculated similarity and the evaluation strategy of the corresponding test questions [14], [15]. The verification of answers to subjective questions was classified according to the approach to calculating the similarity between answers as follows:

- Based on keyword phrases
  This type of approach first allows to split the sentences into keyword phrases and then calculate the similarity between them according to the matching relationship of keyword phrases between sentences [16].
- Based on vector space model (VSM)
  The basic principle of VSM is to use machine learning algorithms to first convert sentences into vector representations, and then calculate the similarity between them [17].
- Based on deep learning
  This type of approach allows the use of neural network models to calculate the similarity between sentences. Representative neural network models include: Tree-LSTM, Transformer and BERT [18].
- Based on semantic graph
  The basic principle of calculating the similarity between answers (i.e., sentence or short text) using this type of approach is to first convert the answers into a semantic graph representation using natural language processing tools, and then calculate the similarity between them. A semantic graph is a network that represents semantic relationships between concepts. In the ostis-systems, the semantic graph is constructed using SC-code (as a basis for knowledge representation within the OSTIS Technology, a unified coding language for information of any kind based on semantic networks is used, named SC-code) [1]. The main advantage of this type of approach is computing the similarity between answers based on semantics. One of the most representative approaches is SPICE (Semantic Propositional Image Caption Evaluation) [19].

The main disadvantages of the existing methods are as follows:

- keyword phrase and VSM based approaches do not take into account semantic similarity between answers;
- semantic graph-based approaches supporting only the description of simple semantic structure;
- these approaches cannot determine the logical equivalence between answers;
- these approaches are dependent on the corresponding natural language.

In ITS information is described in the form of semantic graphs and stored in the knowledge base. Therefore based on existing approaches and OSTIS Technology an approach to automatically generate test questions using knowledge bases and verify user answers based on the similarity between semantic graphs is proposed in this article.

III. Proposed approach

The subsystem can be divided into two parts according to the functions to be implemented: automatic generation of test questions and automatic verification of user answers. In the following, the functions of these two parts will be introduced separately.

A. Automatic generation of test questions

The basic principle of automatic generation of test questions in the ostis-systems is to first extract the corresponding semantic fragments from the knowledge base using a series of test question generation strategies, then add some test question description information to the extracted semantic fragments, and finally store the semantic fragments describing the complete test questions in the universal subsystem. The subsystem allows a series of test questions to be extracted from the subsystem and formed into test papers according to the user's requirements when test papers need to be generated. Test papers consisting of semantic graphs of test questions are converted to natural language descriptions using a nature language interface. An approach to developing natural language interface using OSTIS Technology is described in the literature [6]. In the following, the basic principles of automatic generation of test questions in the ostis-systems will be introduced using test question generation strategy based on class as examples.

The inclusion relation is one of the most frequently used relations in the knowledge base of the ostis-systems, which is satisfied between many classes (including subclasses), so that the inclusion relation between classes can be used to generate objective questions. The set theory expression form of inclusion relation between classes is as follows: $S_i \subseteq C (i \geq 1)$, ($S$-subclass, i-subclass number, C-parent class) [5], [7]. The following shows a semantic fragment in the knowledge base that satisfies the inclusion relation in SCn-code (one of SC-code external display languages) [1]:

binary tree
⇐ *inclusion**:
directed tree
⇒ *inclusion**:
- *binary sorting tree*
- *brother tree*
- *decision tree*

An example of a judgement question generated using this semantic fragment is shown below in natural language: <<Binary tree contains binary sorting tree, brother tree and decision tree.>>

◯ True ◯ False

Other types of objective questions can be generated using this strategy.

Other strategies used to generate objective questions include:

- Test question generation strategy based on elements;
- Test question generation strategy based on identifiers;
- Test question generation strategy based on axioms;
- Test question generation strategy based on relation attributes;
- Test question generation strategy based on image examples.

The process of generating subjective questions is shown below:

- searching the knowledge base for semantic fragments describing subjective questions using logic rules;
- storing the found semantic fragments in the knowledge base of the subsystem;
- using manual approaches or automatic approaches (such as natural language interfaces) to describe the definition, proof process or solution process of the corresponding test question according to the knowledge representation rules in SCg-code (SCg-code is a graphical version for the external visual representation of SC-code) or SCL-code (a special sub-language of the SC language intended for formalizing logical formulas) [2].

The proposed approach to generating test questions has the following main advantages:

- within the framework of OSTIS Technology, knowledge is described in a uniform form and structure, so that the component developed using the proposed approach to generating test questions can be used in different ostis-systems;
- the semantic models of the test questions are described using SC-code, so that they do not rely on any natural language;
- using the proposed approach, high quality objective and subjective questions can be generated automatically.

B. Automatic verification of user answers

In the ostis-systems, test questions are stored in the knowledge base in the form of semantic graphs, so the most critical step of user answer verification is to calculate the similarity between the semantic graphs of answers, and when the similarity is obtained and combined with the evaluation strategy of the corresponding test questions, the correctness and completeness of user answers can be verified.

Since the knowledge types and knowledge structures used to describe different types of test questions are not the same, answer verification is further divided into: 1. verification of answers to objective questions; 2. verification of answers to subjective questions.

C. Verification of answers to objective question

Semantic graphs of answers to objective questions are described using factual knowledge according to the same knowledge structures, so the similarity between the semantic graphs of answers to different types of objective questions can be calculated using the same approach. Factual knowledge refers to knowledge that does not contain variable types, and this type of knowledge expresses facts. When the user answers to objective questions in natural language are converted into semantic graphs, they are already integrated with the knowledge already in the knowledge base. So the similarity between answers is calculated based on the semantic description structures [19]. The process of calculating the similarity between the semantic graphs of the answers to the objective questions is shown below:

- decomposing the semantic graphs of the answers into sub-structures according to the structure of the knowledge description;
- using formulas (1), (2), and (3) to calculate the precision, recall and similarity between semantic graphs.

$$P_{sc}(u,s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(u)|} \tag{1}$$

$$R_{sc}(u,s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(s)|} \tag{2}$$

$$F_{sc}(u,s) = \frac{2 \cdot P_{sc}(u,s) \cdot R_{sc}(u,s)}{P_{sc}(u,s) + R_{sc}(u,s)} \tag{3}$$

The parameters are defined as shown below:

- $T_{sc}(u)$ — all substructures after the decomposition of the user answers u;
- $T_{sc}(s)$ — all substructures after the decomposition of the standard answers s;
- $\otimes$ — binary matching operator, which represents the number of matching substructures in the set of two substructures.

Once the similarity between the answers is obtained, the correctness and completeness of the user answers can be verified by combining it with the corresponding evaluation strategy. The evaluation strategy of the objective questions is shown below:

- if there is only one correct option for the current test question, only if the standard answer and the user answer match exactly, the user answer is considered correct;
- if the current question has multiple correct options:
  - as long as the user answer contains an incorrect option, the user answer is considered incorrect;
  - if all the options in the user answer are correct, but the number of correct options is less than the number of correct options in the standard answer, the user answer is considered correct but incomplete. At this time, the user answer score is $R_{sc} * Max_{score}$;
  - if all the options in the standard answer match exactly with all the options in the user answer, the user answer is exactly correct.

Fig. 1 shows an example of verification of user answer to judgment question in SCg-code.

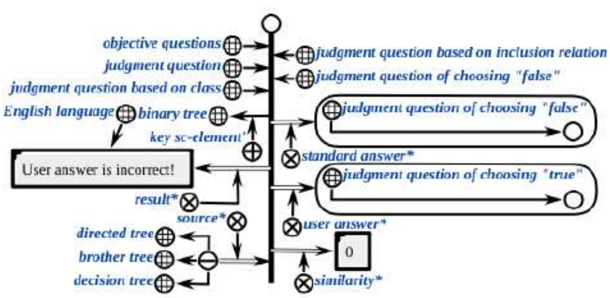

Figure 1. An example of verification of user answer to judgment question.

D. Verification of answers to subjective questions

The approach to calculating the similarity between the semantic graphs of answers to subjective questions, according to the knowledge description structure of the different types of subjective questions, has been divided into: 1. the approach to calculating the similarity between answers to definition explanation questions; 2. the approach to calculating the similarity between answers to proof questions and problem-solving task.

Calculating the similarity between answers to definition explanation questions

The answers to the definition explanation questions are described based on logical formulas (SCL-code). Logic formulas are powerful tools for formal knowledge representation in the framework of OSTIS Technology, which are expanded based on the first-order predicate logic formulas [5]. In the process of calculating the similarity between the semantic graphs of answers to this type of test question, the following tasks need to be solved:

- establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graphs of the answers;
- calculating the similarity between semantic graphs;
- if the similarity between semantic graphs is not equal to 1, they also need to be converted to the prenex normal form (PNF) representation separately, and then the similarity between them is calculated again [23].

The semantic graphs of answers to the definition explanation questions are constructed based on logical formulas, the variables sc-nodes (bound variables) are included in the semantic graphs. In order to calculate the similarity between semantic graphs, mapping relationship of potential equivalent variable sc-node pairs between them needs to be established.

In the ostis-systems, the sc-construction composed of sc-tuple, relation sc-node, role relation sc-node and sc-connector is used to describe logical connectives (such as negation ($\neg$) and implication ($\rightarrow$), etc.) and quantifiers (universal quantifier ($\forall$) and existential quantifier ($\exists$)), atomic logic formula (various sc-constructions) or multiple atomic logic formulas that satisfy conjunctive relation are contained in the sc-structure and connected with the corresponding sc-tuple, and these sc-elements together constitute the semantic graph of answers to the definition explanation questions. Its structure is a tree.

If the standard answer and the user answer are exactly equal, it means that the atomic logic formulas with the same semantics between the answers have the same position in the semantic graph. Thus a mapping relationship between variables sc-nodes can be established by determining the position in the semantic graph of each sc-construction containing the variable sc-nodes and the semantic connotation it expresses [20], [21], [22].

The process of establishing the mapping relationship of the potential equivalent variable sc-node pairs between answers is shown below:

- each sc-tuple and sc-structure in the semantic graph is numbered separately according to the depth-first search strategy (DFS), (for indirectly determining the position of variables sc-nodes in the semantic graph);

- according to the matching relationship of each sc-element between each sc-construction pair with the same number in the semantic graph of the standard answer and the semantic graph of the user answer, the mapping relationships of potential equivalent variable sc-nodes pairs between the semantic graphs are established.

Fig. 2 shows an example of establishing the mapping relationship between semantic graphs in SCg-code.

In Fig. 2, the definition of the partial ordering relation is described. A binary relation R is called a partial ordering, or partial order if and only if it is: reflexive, antisymmetric and transitive.

When the mapping relationship between the potential equivalent variable sc-node pairs between the semantic graphs is established, the similarity between answers can be calculated, and the detailed calculation process is shown below:

- decomposing the semantic graphs of the answers into substructures according to the structure of the knowledge description;
- establishing the mapping relationship of potential equivalent variable sc-node pairs between the semantic graphs;
- using formulas (1), (2) and (3) to calculate the precision, recall and similarity between semantic graphs.

If the similarity between semantic graphs is not equal 1, it is also necessary to determine whether their logical formulas are logically equivalent. Because any predicate logic formula has a PNF equivalent to it. Therefore, based on the approach to convert predicate logic formulas into PNF and characteristics of logic formulas in ostis-systems, an approach to convert logic formulas into unique (deterministic) PNF according to strict restriction rules is proposed in this article [23], [24]. The strict restrictions mainly include the following:

- renaming rule is preferred when converting logical formulas to PNF;
- existential quantifier is moved to the front of the logical formula in preference;
- the logical formula can usually be expressed in the following form: $(Q_1x_1Q_2x_2...Q_nx_n(A \leftrightarrow B))$, where $Q_i(i = 1, ...n)$ is a quantifier. A is used to describe the definition of a concept at a holistic level, and it does not contain any quantifiers. B is used to explain the semantic connotation of a definition at the detail level, and it is usually a logical formula containing quantifiers [8], [24]. Therefore, in order to simplify the knowledge processing, it is only necessary to convert the logical formula B to PNF;

The process of converting the semantic graph constructed based on logic formula into PNF descriptions is shown below:

- if there are multiple sc-structures connected by the same conjunctive connective, the sc-constructions contained in them are merged into the same sc-structure;
- eliminating all the implication connectives;
- moving all negative connectives to the front of the corresponding sc-structure;
- using renaming rules so that all bound variables in the semantic graphs are not the same;
- moving all quantifiers to the front of the logical formula;
- merging again the sc-structures in the semantic graphs that can be merged.

If the calculated similarity between the semantic graphs of PNF representation is not 1, the similarity between the semantic graphs calculated for the first time is used as the final answer similarity.

Fig. 3 shows an example of converting a semantic graph into PNF representation in SCg-code.

In Fig. 3, the definition of the reflexive relation is described. In mathematics, a binary relation R on a set M is reflexive if it relates every element of M to itself.

Calculating the similarity between answers to proof questions and problem-solving task

Both proof questions and problem-solving task follow a common task-solving process:

1) the set (Ω) of conditions consisting of some known conditions;
2) deriving an intermediate conclusion using some of the known conditions in Ω and adding it to Ω. Each element in Ω can be regarded as a solving step;
3) repeat step 2) until the final result is obtained [25], [26].

This task-solving process is abstracted as a directed graph, whose structure is in most cases an inverted tree, and is called a reasoning tree (i. e. the reasoning tree of the standard answer). The automatic verification process of user answers to this type of test questions is the same as the traditional manual answer verification process, i.e., verifying whether the current solving step of the user answer is a valid conclusion of the partial solving step preceding that step. This means whether the solving step in the user answer corresponding to the parent node in the reasoning tree always is located after the solving steps in the user answer corresponding to the child nodes [27].

The semantic graphs of user answers to proof questions and problem-solving task in the ostis-systems are linear structures consisting of some semantic sub-graphs for describing the solving steps and some semantic fragments for describing the logical order and transformation processes between the semantic sub-graphs. The semantic graph of standard answers to this type of test questions is an reasoning tree consisting of a number of search templates (which can be abstracted as the nodes in the tree). Each search template is constructed using SCL-

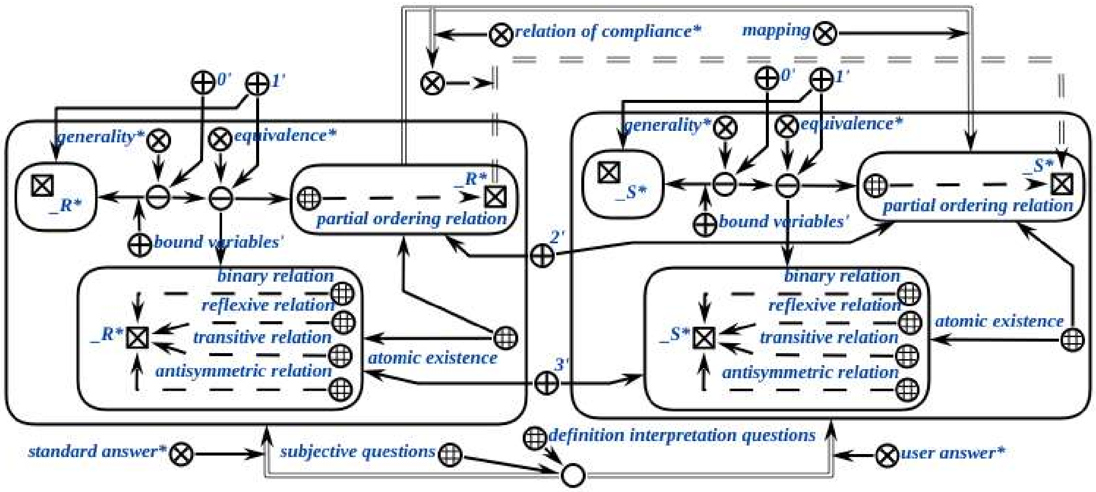

Figure 2. An example of establishing the mapping relationship between semantic graphs.

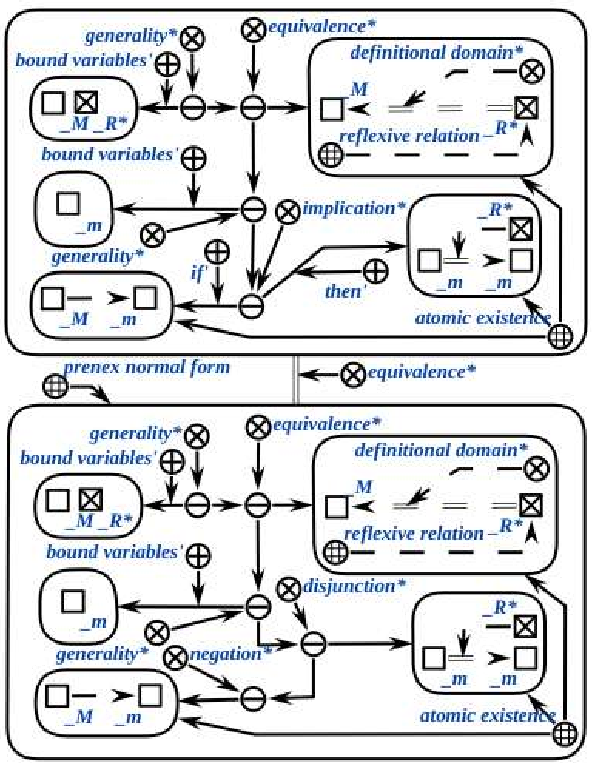

Figure 3. An example of converting a semantic graph into PNF representation.

code in strict accordance with the standard solution steps corresponding to the test question. The search template is used to search in the knowledge base for all semantic fragments corresponding to it [7], [28].

Since the user answers in natural language are converted into semantic graphs they are already integrated with the knowledge already available in the knowledge base. Therefore, when calculating the similarity between the semantic graphs, it is not necessary to consider the differences of the concepts at the natural language level. For example, Segment AB and Segment BA are represented by the same sc-node, they are just two identifiers of the sc-node [6], [7]. An approach to calculate the similarity between the semantic graphs of answers to proof questions and problem-solving task according to the reasoning tree of standard answer (semantic graph of standard answer) is proposed in this article, and the specific calculation process is shown below:

1) numbering each semantic sub-graph in the semantic graph of user answer (the numbering order started from 1);
2) each node in the reasoning tree (search template) is traversed in turn according to the DFS strategy. At the same time, the corresponding semantic sub-graph that is included in the semantic graph of the user answer are searched in the knowledge base using the search template currently being traversed. If such a semantic sub-graph exists, then determine whether the searched semantic sub-graph number is smaller than the semantic sub-graph number corresponding to the search template of the current search template parent node (except for the root node of the reasoning tree), and if so, the searched semantic sub-graph is considered correct;
3) repeat step 2) until all search templates in the reasoning tree have been traversed and the number of correct semantic sub-graphs is counted at the same time;
4) using formulas (1), (2) and (3) to calculate the precision, recall and similarity between answers.

Since this article focuses on the entire process from test question generation to the automatic scoring of test papers and the effective evaluation of subsystems, the basic principles of answer verification to subjective questions are thus briefly presented. For a detailed understanding of the process of constructing semantic models for subjective questions and user answer verification to subjective questions please refer to the literature [3].

Once the similarity between the answers to the subjective questions is obtained, the correctness and completeness of the user answers can be verified combined with the evaluation strategy for the subjective questions. The evaluation strategy for subjective questions includes:

- if the similarity between the answers is equal to 1, the user answer is completely correct;
- if the similarity between the answers is less than 1 and the precision is equal to 1, the user answer is correct but incomplete and the user score is $R_{sc} * Max_{score}$;
- if the similarity between the answers is greater than 0 and less than 1, and the precision is less than 1, then the user answer is partially correct and the user score is $F_{sc} * Max_{score}$;
- if the similarity between the answers is equal to 0, the user answer is wrong.

The proposed approach to automatic verification of user answers has the following advantages:

- verifying the correctness and completeness of user answers based on semantics;
- the logical equivalence between answers can be determined;
- the similarity between any two semantic graphs in the knowledge base can be calculated;
- the developed component using the proposed approach can be easily transplanted to other ostis-systems.

IV. Knowledge base of the subsystem

The knowledge base of subsystem is used to store automatically generated test questions, and it also allows to automatically extract a series of test questions and form test papers according to user requirements. Therefore, in order to improve the efficiency of accessing the knowledge base of the subsystem and the efficiency of extracting the test questions, an approach to construct the knowledge base of the subsystem according to the type of test questions and the generation strategy of the test questions is proposed in this article.

The basis of the knowledge base of any ostis-system (more precisely, the sc-model of the knowledge base) is a hierarchical system of subject domains and their corresponding ontologies [1], [2], [5]. Let's consider the hierarchy of the knowledge base of subsystem in SCn-code:

Section. Subject domain of test questions
⇐ *section decomposition**:
{• *Section. Subject domain of subjective questions*
⇐ *section decomposition**:
{• *Section. Subject domain of definition explanation question*
• *Section. Subject domain of proof question*
• *Section. Subject domain of problem-solving task*
}
• *Section. Subject domain of objective questions*
⇐ *section decomposition**:
{• *Section. Subject domain of multiple-choice question*
• *Section. Subject domain of fill in the blank question*
• *Section. Subject domain of judgment question*
}
}

Objective types of test questions are decomposed into more specific types according to their characteristics and corresponding test question generation strategies. Next, taking the judgment question as an example let us consider its semantic specification in SCn-code:

judgment question
∈ *maximum class of explored objects'*:
Subject domain of judgment question
⇐ *subdividing**:
{• *judgment question based on relation attributes*
• *judgment question based on axioms*
• *judgment question based on image examples*
• *judgment question based on identifiers*
• *judgment question based on elements*
⇐ *subdividing**:
{• *judgment question based on role relation*
• *judgment question based on binary relation*
}
• *multiple-choice question based on classes*
⇐ *subdividing**:
{• *judgment question based on subdividing relation*
• *judgment question based on inclusion relation*
• *judgment question based on strict inclusion relation*
}

}

⇐ *subdividing**:

{• *judgment question of choosing true*

• *judgment questions of choosing false*

}

V. Problem solver

One of the most important components of every intelligent system is the problem solver, which provides the ability to solve a variety of problems. The problem solver of any ostis-system (more precisely, the sc-model of the ostis-system problem solver) is a hierarchical system of knowledge processing agents in semantic memory (sc-agents) that interact only by specifying the actions they perform in the specified memory [1], [4].

Therefore, a problem solver for automatic generation of test questions and automatic verification of user answers has been developed based on the proposed approach, and its hierarchy is shown below in SCn-code:

Problem solver for the automatic generation of test questions and automatic verification of user answers

⇐ *decomposition of an abstract sc-agent**:

{• *Sc-agent for automatic generation of test questions*

⇐ *decomposition of an abstract sc-agent**:

{• *Sc-agent for quick generation of test questions and test papers*

• *Sc-agent for generating single type of test questions*

• *Sc-agent for generating a single test paper*

}

• *Sc-agent for automatic verification of user answers*

⇐ *decomposition of an abstract sc-agent**:

{• *Sc-agent for automatic scoring of test papers*

• *Sc-agent for calculating similarity between answers to objective questions*

• *Sc-agent for calculating the similarity between answers to definition explanation questions*

• *Sc-agent for converting a logical formula into PNF*

• *Sc-agent for calculating the similarity between the answers to proof questions and problem-solving task*

}

}

The function of the sc-agent for quick generation of test questions and test papers is to automate the entire process from test question generation to test paper generation by initiating the corresponding sc-agents (sc-agent for generating single type of test questions and sc-agent for generating a single test paper).

The function of the sc-agent for automatic scoring of test papers is to implement automatic verification of user answers to test questions and automatic scoring of test papers by initiating sc-agents for calculating the similarity between user answers and sc-agents for converting a logical formula into PNF.

VI. Evaluating the effectiveness of the subsystem

The effectiveness of the developed subsystem will be evaluated from the following aspects:

- availability of the generated test questions;
- difficulty level of the generated test papers;
- closeness between automatic scoring and manual scoring of user answers to subjective questions.

In order to evaluate the availability of the automatically generated test questions, 200 automatically generated test questions were randomly sampled from the tutoring system for discrete mathematics and the proportion of test questions that could be used directly was counted (Table I).

Table I
TABLE. RESULTS OF THE EVALUATION OF THE AVAILABILITY OF THE GENERATED TEST QUESTIONS

Availability indicators	Test questions that can be used directly	Test questions that can be used after modification	Unavailable test questions
Number of test questions (total 200)	188	12	0
Proportion	94%	6%	0

It can be seen from Table I that of the 200 automatically generated test questions sampled at random, 94% were able to be used directly and 6% were able to be used after modification.

The difficulty of the test paper is closely related to the user's score. Therefore, 40 second-year students were randomly selected to evaluate the difficulty of the test paper for discrete mathematics, which was automatically generated using the subsystem. 10 multiple-choice questions, 10 fill in the blank questions, 10 judgment questions, 2 definition explanation questions and 2 proof questions are included in this test paper. The maximum score for each objective question is 2 points, the maximum score for each subjective question is 10 points, and the maximum score for the whole test paper is 100 points (Table II).

From the Table II, it can be seen that the students' scores generally follow a normal distribution. Therefore, it can be concluded that the difficulty of the current type

Table II
TABLE. STATISTICAL RESULTS OF STUDENT SCORES

Score	<40	[40-49]	[50-59]	[60-69]	[70-79]	[80-89]	≥90
Total number of students (40)	0	1	4	10	14	8	3
Proportion	0	2.5%	10%	25%	35%	20%	7.5%
Average score	72.85						

of test paper is moderate and that the actual knowledge level of the user can be checked objectively and fairly.

In order to evaluate the closeness between the automatic scoring and manual scoring of user answers to the subjective questions, we decided to first enter the 40 students' answers to the subjective questions into the subsystem, then use the subsystem to automatically verify the students' answers, and finally count the error between the automatic scoring and manual scoring of user answers to the subjective questions (Table III).

Table III
TABLE. RESULTS OF SCORING ERROR STATISTICS FOR USER ANSWERS TO SUBJECTIVE QUESTIONS

Error range (Φ)	Definition explanation question 1	Definition explanation question 2	Proof question 1	Proof question 2	Total	Proportion
$\Phi \leq 1$	35	31	26	28	120	75%
(1-1.5]	2	4	8	8	22	13.75%
(1.5-2]	2	3	5	3	13	8.125%
$\Phi > 2$	1	2	1	1	5	3.125%

The formula for calculating the error Φ is shown below (4):

$$\Phi = |x - y| \quad (4)$$

The parameters are defined as shown below:

- x — is the manual scoring of user answers to the test questions;
- y — is the automatic scoring of user answers to the test questions;

From the Table III, it can be seen that the automatic scoring and manual scoring of user answers to subjective questions in the tutoring system for discrete mathematics generally remained consistent, and that when the maximum score for a subjective question was 10, the sample size with an error $\Phi \leq 1.5$ between scores was over 88%.

The above experimental results show that the developed subsystem can satisfy the conditions for practical applications.

VII. CONCLUSION

An automated approach to checking the knowledge level of users in tutoring systems developed using OSTIS Technology is proposed in this article. Based on the proposed approach, a universal subsystem for automatic generation of test questions and automatic verification of user answers is developed. Using the developed subsystem, the entire process can be automated from test question generation, test paper generation to automatic verification of user answers and automatic scoring of test papers.

Finally the effectiveness of the developed subsystem was evaluated in terms of the availability of the generated test questions, the difficulty of the generated test papers and the closeness between the automatic scoring and the manual scoring of the test questions in the discrete mathematics ostis-system. From the evaluation results, it can be seen that the developed subsystem can meet the conditions for practical application.

ACKNOWLEDGMENT

The work in this article was done with the support of research teams of the Department of Intelligent Information Technologies of Belarusian State University of Informatics and Radioelectronics. Authors would like to thank every researcher in the Department of Intelligent Information Technologies.

REFERENCES

[1] V. V. Golenkov and N. A. Guljakina, "Proekt otkrytoj semanticheskoj tehnologii komponentnogo proektirovanija intellektual'nyh sistem. chast' 1: Principy sozdanija project of open semantic technology for component design of intelligent systems. part 1: Creation principles]," Ontologija proektirovanija [Ontology of design], no. 1, pp. 42–64, 2014.

[2] V. Golenkov, N. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 27–40.

[3] D. Shunkevich, "Metodika komponentnogo proektirovaniya sistem, upravlyaemyh znaniyami," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2015, pp. 93–110.

[4] D. Shunkevich, "Hybrid problem solvers of intelligent computer systems of a new generation," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 119–144.

[5] (2023, Mar) Ims.ostis metasystem. [Online]. Available: https://ims.ostis.net

[6] Longwei. Qian, "Ontological Approach to the development of natural language interface for intelligent computer systems," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 217–238.

[7] Wenzu. Li, "A semantics-based approach to automatic generation of test questions and automatic verification of user answers in the intelligent tutoring systems," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2022, pp. 369–374.

[8] Li. W., Qian L. Semantic Approach to User Answer Verification in Intelligent Tutoring Systems. Communications in Computer and Information Science, 2022, Vol. 1625, pp. 242–266.

[9] Jiang. H., Zhao. Y., Li. X. Intelligent education system: A comprehensive review. IEEE Access, 2020, Vol. 8, pp. 201016-201026.
[10] Li. H. Research on item automatic generation based on DL and domain ontology. Journal of Changchun University of Technology (Natural Science Edition), 2012, Vol. 33(04), pp. 461-464.
[11] Kaur. M., Batra. S. Ontology-based question generation: a review. Journal of Intelligent and Fuzzy Systems, 2020, Vol. 39(01), pp. 49-57.
[12] Andreas. P., Konstantinos. K., Konstantinos. K. Automatic generation of multiple-choice questions from domain ontologies. In: IADIS International Conference e-Learning, 2008, pp. 427-434.
[13] Arjun. S. B., Manas. K., Sujan. K. S. Automatic Generation of Multiple Choice Questions Using Wikipedia. International Conference on Pattern Recognition and Machine Intelligence, 2013, Vol. 8251, pp. 733-738.
[14] Chen. C. H. A comprehensive survey of text similarity measures. Journal of Information Science, 2019, Vol. 45(01), pp. 1-19.
[15] Singh. A., Singh. V. K. Text similarity techniques: a systematic literature review. International Journal of Computer Applications, 2019, Vol. 182(41), pp. 23-30.
[16] Baly. R., Jansen. P., De. Rijke. M. A review of evaluation methodologies in automatic short answer grading. Educational Research Review, 2014, Vol. 13, pp. 20-33.
[17] Kim. Y., Kwon. O. N., Kim. Y. Automatic short answer grading using latent semantic analysis with fuzzy matching. Journal of Educational Technology and Society, 2016, Vol. 19(03), pp. 257-269.
[18] Mingyu. Ji., Xinhai. Zhang. A Short Text Similarity Calculation Method Combining Semantic and Headword Attention Mechanism. Scientific Programming, 2022.
[19] Anderson. P., Fernando. B., Johnson. M., Gould. S. Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision, Springer, 2016, pp. 382-398.
[20] Shu. X., Wang. S., Guo. L., Zhao. Y. Survey on multi-source information fusion: progress and challenges. Information Fusion, 2018, Vol. 42, pp. 146-169.
[21] Wang. X. Y., Hu. Z. W., Bai. R. J., et al. Review on Concepts, Processes, Tools and Methods Ontology Integration. Library and Information Service, 2011, Vol. 55(16), pp. 119-125.
[22] Liu. J., Liu. Y., Wang. X., Wang. B. A survey on ontology integration techniques. Future Generation Computer Systems, 2020, Vol. 105, pp. 17-30.
[23] Fujiwara. M., Kurahashi. T. Prenex normal form theorems in semi-classical arithmetic. The Journal of Symbolic Logic, 2022, pp. 1-31.
[24] Pan. M., Ding. Z. A simple method for solving prenex disjunction (conjunction) normal forms. Computer Engineering and Science, 2008, Vol. 30(10), pp. 82-84.
[25] Ge. L., Li. X. An overview of the methods for automated theorem proving in geometry. Journal of Symbolic Computation, 2016, Vol. 74, pp. 303-335.
[26] Kryakin. A., Nguyen. P. Automated theorem proving in geometry using Gröbner bases. Journal of Automated Reasoning, 2014, Vol. 53(02), pp. 163-189.
[27] Reiter. R. A Logic for Default Reasoning. Artificial Intelligence, 1980, Vol. 13(1-2), pp. 81-132.
[28] Chou. S. C., Gao. X. S., Zhang. J. Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems. World Scientific, 1994.

Автоматизированный подход к проверке уровня знаний пользователей в интеллектуальных обучающих системах

Ли Вэньцзу

Данная работа посвящена проблеме автоматизации реализации быстрого тестирования знаний пользователей в интеллектуальных обучающих системах нового поколения. В данной работе подробно описывается основанный на семантике подход к автоматизации всего процесса от генерации тестовых вопросов и экзаменационных билетов до автоматической проверки ответов пользователей и автоматической оценки экзаменационных билетов.

На протяжении многих лет педагоги активно высказывают желание использовать компьютеры для автоматизации обучения и преподавания. С развитием технологии искусственного интеллекта в последние годы, это желание может наконец-то стать реальностью. Наиболее представительным продуктом, объединяющим искусственный интеллект и образование, являются интеллектуальные обучающие системы (ИОС), которые могут не только значительно повысить эффективность обучения пользователей, но и обеспечить справедливость и беспристрастность образовательного процесса.

Автоматическая генерация тестовых вопросов и автоматическая проверка ответов пользователей являются самыми основными и важными функциями ИОС. Использование этих двух функций в комбинации позволит реализовать весь процесс от автоматической генерации тестовых вопросов до автоматической оценки экзаменационных билетов пользователей. Это не только значительно сократит повторяющуюся работу педагогов, но и снизит стоимость обучения для пользователей, что позволит большему числу людей получить доступ к различным знаниям.

Хотя в последние годы благодаря развитию таких технологий, как семантические сети, глубокое обучение и обработка естественного языка (NLP), было предложено несколько подходов для автоматической генерации тестовых вопросов и проверки ответов пользователей, эти методы все еще имеют следующие основные недостатки:

- существующие подходы к генерации тестовых вопросов позволяют генерировать только самые простые объективные вопросы;
- некоторые из существующих подходов (например, сопоставление ключевых слов и использование статистической вероятности) для проверки ответов пользователей на субъективные вопросы не учитывают семантическое сходство между ответами;
- методы, использующие семантику для проверки ответов пользователей на субъективные вопросы, могут вычислять сходство только между ответами с простыми семантическими структурами;
- и т.д.

Поэтому на основе существующих методов и Технологии OSTIS в данной работе предлагается подход к разработке универсальной подсистемы для автоматической генерации тестовых вопросов и автоматической проверки ответов пользователя в обучающих системах, разработанных с использованием Технологии OSTIS (открытая семантическая технология проектирования интеллектуальных систем).

Received 29.03.2023