An Approach to Automate the Entire Process from the Generation of Test Questions to the Verification of User Answers in Intelligent Tutoring Systems

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

***Abstract*—This article is dedicated to the issues of automatic generation of test questions and automatic verification of user answers in the new generation of intelligent systems based on semantic representation of information. An approach to implement automatic generation of test questions of various types using knowledge bases and automatic verification of user answers according to the semantic representation structure of the knowledge is detailed in this article.**

***Keywords*—test question generation, user answer verification, OSTIS Technology, intelligent systems based on semantic representation of information, semantic structure, knowledge base**

I. Introduction

With the rapid advancement of artificial intelligence technology in areas like natural language processing and image processing, researchers have started integrating artificial intelligence into the field of education. One of the most representative products of the combination of artificial intelligence and education is **Intelligent Tutoring Systems** (ITS). The significance of ITS lies in bringing revolutionary changes to the field of education. By combining artificial intelligence technology with educational theory, ITS can personalize guidance for students, providing tailored educational content and feedback based on each student's learning style, progress, and needs. This personalized teaching approach helps stimulate students' interest in learning, improve learning efficiency, and facilitate knowledge absorption and mastery. In conclusion, the emergence of ITS injects new vitality into education, offering students and teachers more effective learning and teaching tools, thus driving the progress and development of education [1].

In ITS, the automatic generation of test questions and the automatic verification of user answers play a crucial role. By generating test questions automatically, the system can provide personalized tests based on students' learning content and level, aiding them in comprehensive review and reinforcement of knowledge points. On the other hand, the automatic verification of user answers allows the system to promptly check students' answers and provide immediate feedback or guidance on correctness. This helps students correct errors in a timely manner during the learning process, enhancing their understanding and retention of knowledge points. In conclusion, the automatic generation of test questions and the automatic verification of user answers enhance personalization and immediacy in ITS, leading to improved learning outcomes and an increase in teaching quality [2], [3], [4], [5].

In recent years, with the continuous development of artificial intelligence technology, a variety of approaches to automatically generate test questions and automatically verify users' answers have been proposed. However, these approaches still have some limitations, mainly in the following areas:

- current approaches to test question generation only allow for the generation of test questions of simple types;
- some of the existing approaches to verifying user answers (e.g., keyword matching and probabilistic statistics) do not take into account the semantic similarity between answers;
- semantic-based approaches to verifying user answers only allow verification of answers with simple semantic structures [5], [6], [7].

Therefore an approach to develop a subsystem for automatic generation of test questions of various types using knowledge bases and automatic verification of user's answers according to the semantic representation structure of information in intelligent tutoring systems of the new generation based on semantic representation of information is proposed in this article [2], [3], [8]. Using the developed subsystem not only allows to generate test questions of various types using the knowledge bases and to verify the correctness and completeness of the user's answers based on semantics, but also to automate the entire process from the generation of test questions to the grading of the test papers. The ITS for discrete mathematics will be used as a demonstration system for the developed subsystem.

The remainder of this article is organized as follows. Section II presents a review of several existing approaches to generating test questions and verifying user answers. Section III describes our proposed approach to automatically generate test questions and automatically verify user answers. Section IV evaluates the effectiveness of the subsystem developed using our proposed approach. Section V concludes this article.

II. Related works

A. Automatic generation of test questions

Approach to automatic generation of test questions mainly focuses on automatic generation of test questions using electronic documents, text corpora and knowledge bases. Knowledge bases store highly structured knowledge that has been filtered, and with the development of the semantic networks, automatic generation of test questions from knowledge bases has become the most important research direction in this field. The basic principles of automatic generation of objective questions using knowledge bases are described in detail in references [7], [9].

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.

The primary issues with current approaches to generating test questions are as follows:

- using electronic documents to automatically generate test questions necessitates a substantial amount of sentence templates;
- compiling a text corpus demands significant human resources to gather and process diverse knowledge;
- existing approaches only allow to generate simple objective questions [10], [11].

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 [12], [13]. The more similar the standard answer and the user answer are, the higher the similarity between them. The verification of user answers to subjective questions is divided into the following categories according to the approaches used to calculate the similarity between the answers:

- based on natural language;
- based on semantic graph.

The basic principle of approaches to calculate the similarity between answers to subjective questions based on natural language is to convert natural language text into vectors using a series of tools for modelling natural language text (for example, Jaccard similarity, vector space models (TF-IDF, Doc2Vec), deep learning models (Transformer, BERT), and etc.), and then to calculate the similarity between the vectors. Since test questions of various types and their answers are described in the form of semantic graphs in the knowledge base of ITS, this article focuses on approaches to compute the similarity between answers using semantic graphs [14]. A semantic graph is a network that represents semantic relationships between concepts. In the reference [4] we have described in detail the approaches to calculate the similarity between answers to subjective questions based on natural language and compared their advantages and disadvantages.

The basic principle of calculating the similarity between answers to subjective questions (i.e., sentence or short text) based on semantic graphs is that the answers are first converted into semantic graph representations using natural language processing tools (such as syntactic dependency trees and natural language interfaces) and then the similarity between them is calculated (i.e., similarity between answers). 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) [15].

These approaches primarily encounter the following issues:

- the keyword phrase-based approach does not take into account the order between words in a sentence;
- the VSM-based approach leads to the generation of high-dimensional sparse matrices, which increases the complexity of the algorithm;
- semantic graph-based approaches supporting only the description of simple semantic structures;
- these approaches cannot determine whether the sentences are logically equivalent to each other;
- 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 for the above reasons an approach to automatically generate test questions of various types using knowledge bases and to automatically verify user answers according to the similarity between the semantic graphs of the answers in intelligent tutoring systems based on semantic representations of information is proposed in this article.

III. Proposed approach

The main task of this article is to develop a subsystem for automatic generation of test questions and automatic verification of user answers in intelligent tutoring systems of the new generation based on semantic representation of information. To achieve this task OSTIS Technology is proposed to be used [2], [3], [8].

A. Description of the used technology

OSTIS technology is a complex open semantic technology for the design and development of intelligent systems. Intelligent systems developed using the OSTIS Technology are called ostis-systems. Each ostis-system for different application fields includes a platform for interpretation semantic models of ostis-systems, as well as a semantic model of ostis-systems using SC-code (sc-model of ostis-systems). At the same time, the sc-model of the ostis-systems includes the sc-model of the knowledge base, the sc-model of the problem solver and the sc-model of the interface (in particular, the user-oriented intelligent interface). 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. Texts of the SC-code (sc-texts) are unified semantic networks with a basic set-theoretic interpretation. Within the framework of the technology, several universal variants of visualization of the SC-code constructs are also proposed, such as SCg-code (graphic version), SCn-code (non-linear hypertextual version), SCs-code (linear string version). The methods and rules for the detailed design of intelligent systems using OSTIS technology are described in the reference [2].

OSTIS Technology offers the following possibilities:

- unified knowledge description language SC-code;
- models describing knowledge of various types, models and tools for the development of knowledge bases;
- integration of various problem-solving models;
- multi-agent approach to developing problem solvers.

Next we will describe in detail the process of developing a subsystem for the automatic generation of test questions and the automatic verification of user answers in intelligent tutoring systems developed using OSTIS technology.

B. Architecture of the subsystem

The task of this article is to develop a subsystem using the proposed approach for generating test questions and verifying user answers. Fig. 1 shows the organisation of the developed subsystem in the ostis-system.

As can be seen in Fig. 1 in the ostis-system information is passed between components in the form of semantic fragments constructed using SC-code. Natural language interfaces are used to implement the interaction between the intelligent system and the user. An approach to developing natural language interface using OSTIS Technology is described in the reference [16]. In order to facilitate the explanation of the working principle of the subsystem, the illustrations and knowledge base fragments we choose in this article are all in English, but it needs to be emphasized that the subsystem developed does not depend on natural language.

The subsystem is divided into two functional components according to the functions to be realised:

- component for automatic generation of test questions;
- component for automatic verification of user answers.

Fig. 2 shows the complete working process of the component for automatic generation of test questions.

The work corresponding to stages 1 and 5 in Fig. 2 is done using a natural language interface.

Fig. 3 shows the complete working process of the component for automatic verification of user answers.

The work corresponding to stages 1 and 5 in Fig. 3 is done using a natural language interface.

C. Development of formal semantic model of test questions

The formal semantic model of test questions is aimed at defining ways to describe test questions of various types in the form of semantic graphs in the knowledge bases of ostis-systems. The formal semantic model of test questions is defined as shown below:

$$M_{TQ} = \{M_{MCQ}, M_{FBQ}, M_{JQ}, M_{DIQ}, M_{PSQ}\} \quad (1)$$

The parameters are defined as shown below:

- M_{MCQ} — semantic model of multiple-choice question;
- M_{FBQ} — semantic model of fill in the blank question;
- M_{JQ} — semantic model of judgment question;
- M_{DIQ} — semantic model of definition explanation question;
- M_{PSQ} — semantic model of proof question and problem solving task.

The semantic model of multiple-choice question is defined as shown below:

$$M_{MCQ} = \{S_{MBI}, S_{MAI}, R_{MBI}, R_{MAI}\} \quad (2)$$

The parameters are defined as shown below:

- S_{MBI} — a set of concepts that specifies basic information about multiple-choice question, including

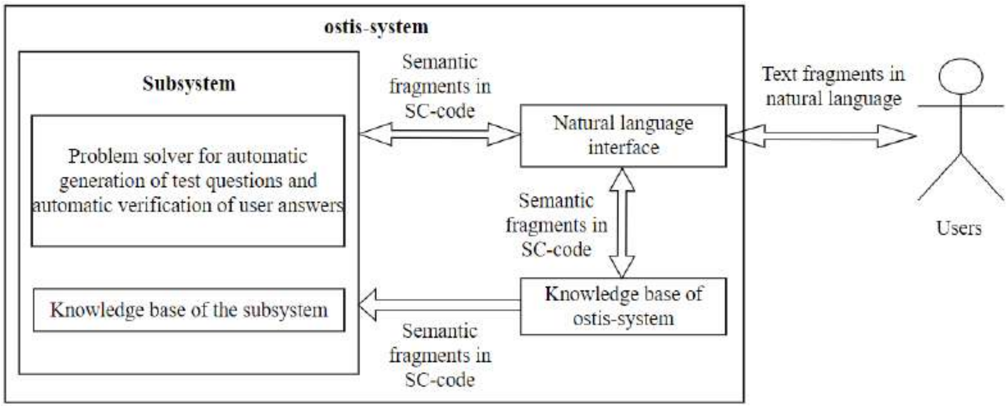

Figure 1. Organisation of the developed subsystem in the ostis-system.

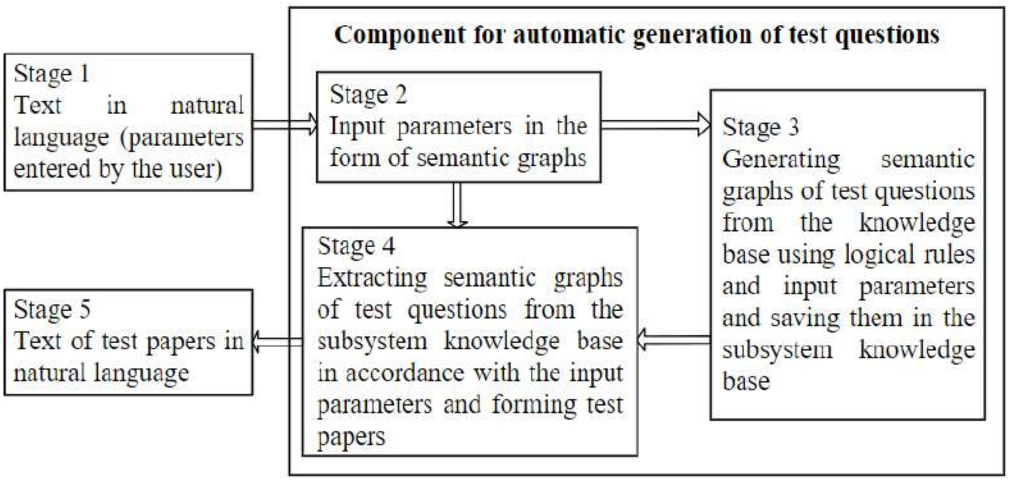

Figure 2. Working process of the component for automatic generation of test questions.

describing its type, key elements, options, and so on;

- S_{MAI} — a set of concepts that specifies the contextual information about the answer to multiple-choice question, including describing the standard answer and the user answer;
- R_{MBI} — a set of relations that specifies concepts from S_{MBI}, including "key sc-element’ ", "standard answer*", etc.;
- R_{MAI} — a set of relations that specifies concepts from S_{MAI}.

Similarly semantic models of test questions of other types can be defined in the way described above.

D. Automatic generation of test questions

The basic principle of automatic generation of test questions of various types (including objective questions and subjective 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 summarized based on the knowledge representation approach and the knowledge description structure in the framework of OSTIS Technology, 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 corresponding section of the subsystem [4]. 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. In the following, the basic principles of automatic generation of objective 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

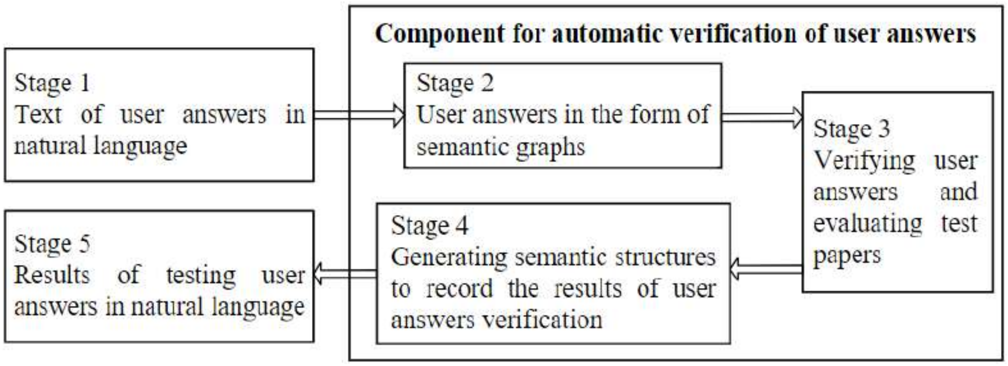

Figure 3. Working process of the component for automatic verification of user answers.

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], [8]. The following shows a semantic fragment in the knowledge base that satisfies the inclusion relation in SCn-code:

binary tree
$\Leftarrow$ *inclusion**:
directed tree
$\Rightarrow$ *inclusion**:
- *binary sorting tree*
- *brother tree*
- *decision tree*

Consider the example of a multiple-choice question generated using this semantic fragment according to the strategy of inclusion relations, which has the natural language form shown below:

<<Binary tree does not include ( )?>>

A. directed tree B. brother tree

C. decision tree D. binary sorting tree

Fig. 4 shows the semantic graph of this generated multiple-choice question that was constructed based on the semantic model of the test questions.

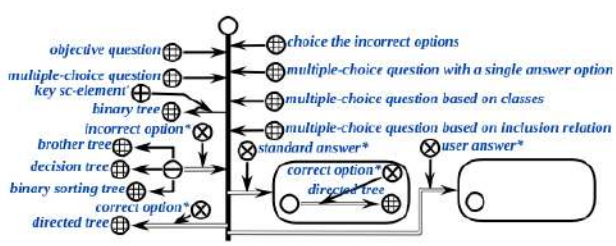

Figure 4. An example of semantic graph of multiple-choice question.

It should be emphasized that the semantic graph of this multiple-choice question was converted into the corresponding natural language description form using a natural language interface.

Fig. 5 shows an example of a logic rule for generating this multiple-choice question constructed based on a strategy of inclusion relation.

Logic rules of test questions in the ostis-systems are constructed using SC-code strictly according to the test question generation strategies. Each logic rule of test questions consists of two main parts: 1. search template; 2. generate template. The search template is used to find all the semantic fragments that satisfy the conditions in the knowledge base. The generation template uses the searched semantic fragments to generate the semantic graph of the test question.

Similarly objective questions of other types can be generated in a similar way 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.

In reference [4] we describe in detail the approach to generating objective questions of various types.

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 or SCL-code (a special sub-language of the SC language intended for formalizing logical formulas), (this part

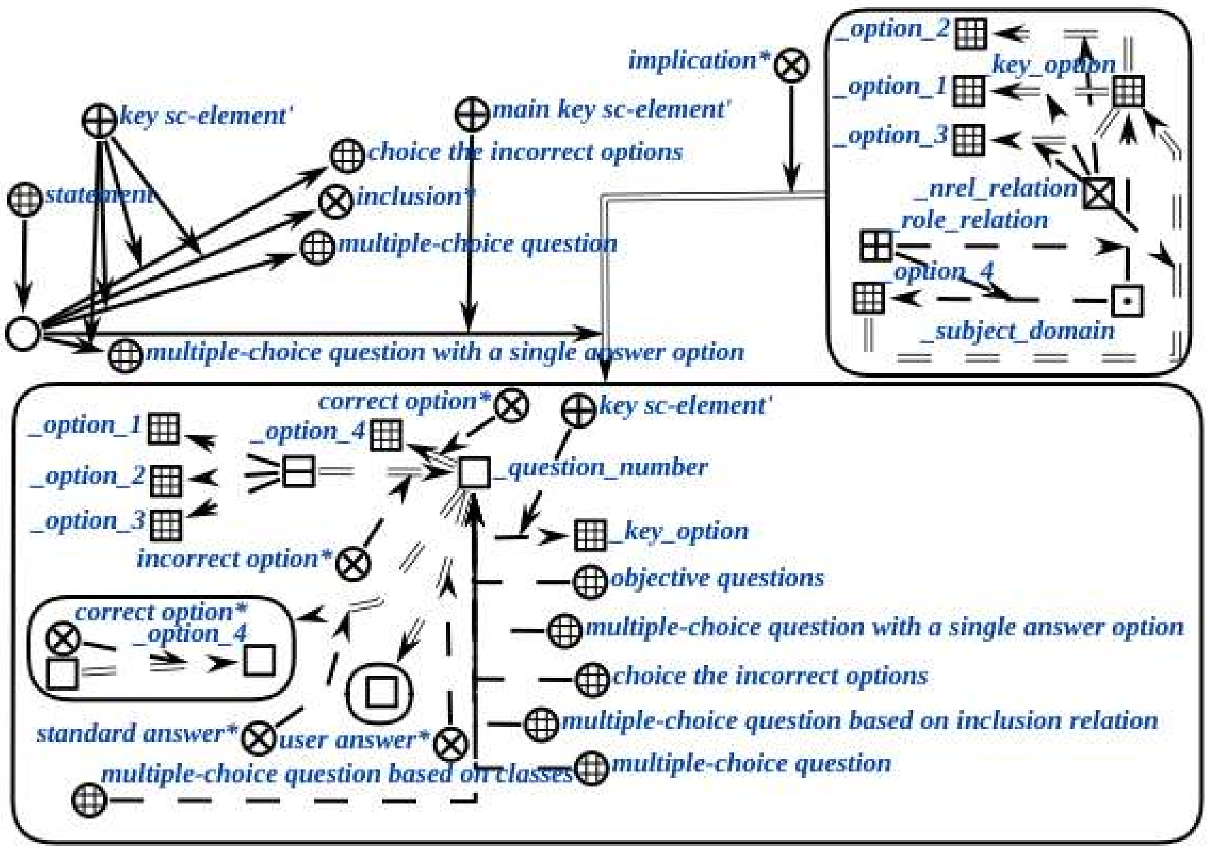

Figure 5. An example of a logic rule for generating multiple-choice question.

of the work can also be done before generating test questions) [8].

The suggested approach to generating test questions offers the following key benefits:

- within the framework of OSTIS Technology, knowledge is described in a unified semantic network language, so that the component developed using the proposed approach to generating test questions can be used in different ostis-systems;
- the developed semantic model of test questions does not rely on natural language, which greatly simplifies the use and processing of semantic graphs of test questions in the knowledge base;
- by utilizing the suggested approach, both objective and subjective questions of high quality can be automatically generated.

E. Automatic verification of user answers

Test questions in the ostis-systems are stored in the knowledge base as semantic graphs. Therefore, the most crucial step in verifying user answers is to calculate the similarity between the semantic graphs of the answers. Once the similarity is determined and combined with the evaluation strategy for the respective 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.

F. Verification of answers to objective question

Semantic graphs of answers to objective questions are described using factual knowledge according to the same knowledge structures. As a result, the similarity between the semantic graphs of answers to objective questions of different types 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 [15]. 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 (3), (4), and (5) to calculate the precision, recall and similarity between semantic graphs.

$$P_{sc}(u,s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(u)|} \tag{3}$$

$$R_{sc}(u,s) = \frac{|T_{sc}(u) \otimes T_{sc}(s)|}{|T_{sc}(s)|} \tag{4}$$

$$F_{sc}(u,s) = \frac{2 \cdot P_{sc}(u,s) \cdot R_{sc}(u,s)}{P_{sc}(u,s) + R_{sc}(u,s)} \tag{5}$$

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. Fig. 6 shows an example of verification of user answer to multiple-choice question in SCg-code.

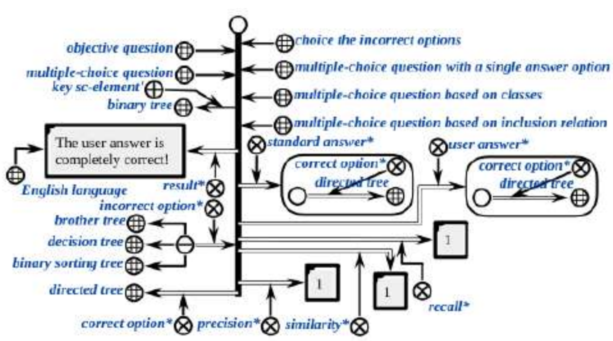

Figure 6. An example of verification of user answer to multiple-choice question.

G. 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 [3], [8]. 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 [17].

The semantic graphs for definition explanation questions are built using logical formulas, incorporating variable sc-nodes (bound variables) within the graphs. To compute the similarity between these semantic graphs, a mapping relationship of potential equivalent variable sc-node pairs between them must 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 [4]. 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 [18]. Fig. 7 shows some sc-constructions used to describe the information in the knowledge base.

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);

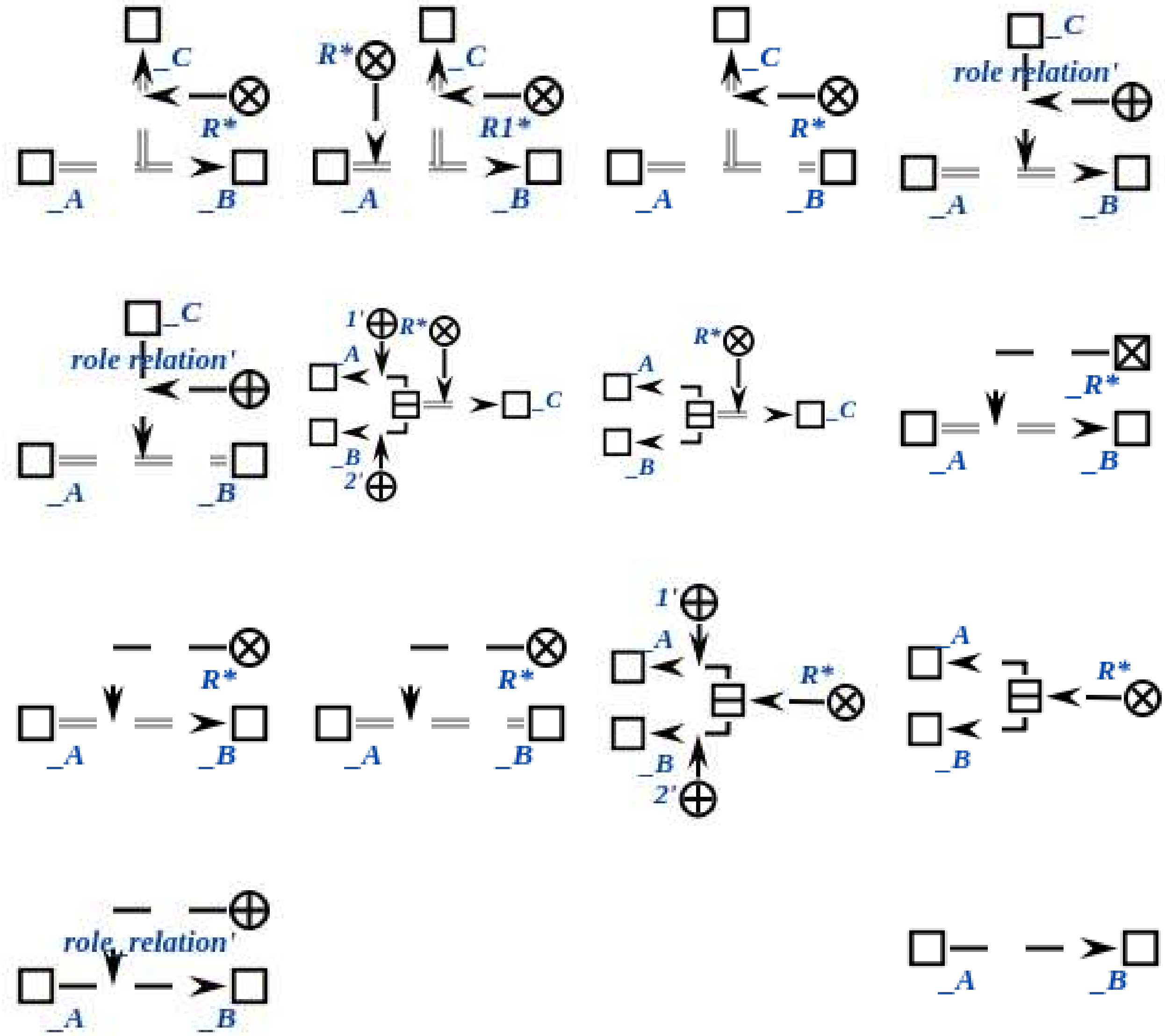

Figure 7. Some examples of sc-constructions.

- 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.

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 (3), (4) and (5) to calculate the precision, recall and similarity between semantic graphs.

Fig. 8 shows an example of calculating the similarity between semantic graphs of answers to a definition explanation question in SCg-code.

In Fig. 8, the definition of the inclusion relation is described $(\forall A \forall B((A \subseteq B) \leftrightarrow (\forall a(a \in A \rightarrow a \in B)))$.

If the similarity between semantic graphs is not equal 1, it is also necessary to determine whether their logical formulas are logically equivalent. Fig. 9 and 10 show examples of logical equivalence between semantic graphs,

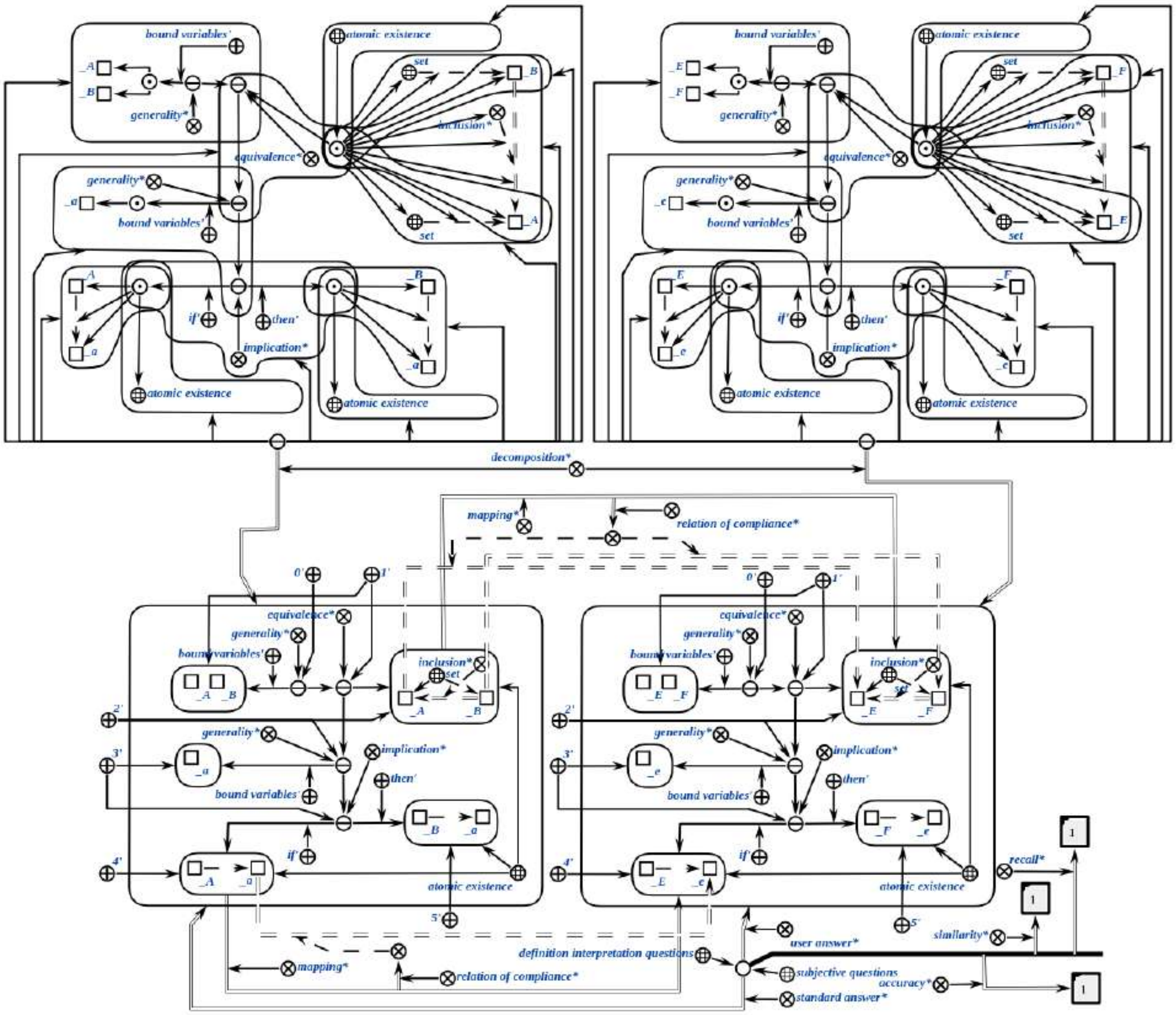

Figure 8. An example of calculating the similarity between semantic graphs of answers to a definition explanation question.

respectively.

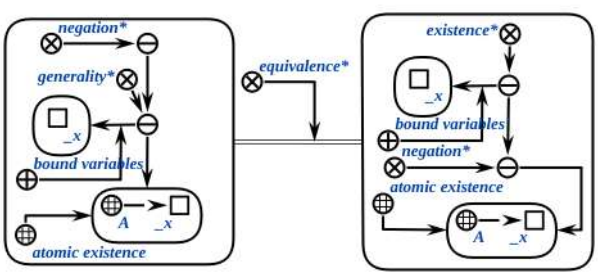

Figure 9. An example of semantic graphs satisfying logical equivalence ($\neg\forall x A(x) \Leftrightarrow \exists x \neg A(x)$).

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

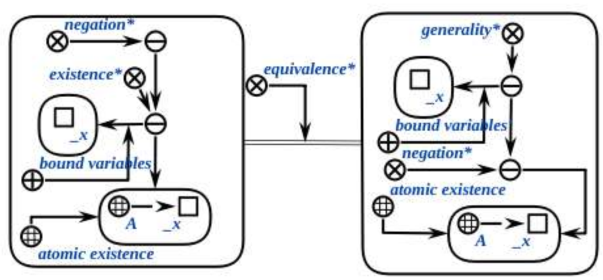

Figure 10. An example of semantic graphs satisfying logical equivalence ($\neg\exists x A(x) \Leftrightarrow \forall x \neg A(x)$).

formulas into unique (deterministic) PNF according to strict restriction rules is proposed in this article [17], [19]. 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 [6], [19]. 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.

Fig. 11 shows an example of converting a semantic graph into PNF representation in SCg-code $(\forall A\forall B((A \subseteq B) \leftrightarrow \forall a(a \in A \rightarrow a \in B)) \Leftrightarrow \forall A\forall B((A \subseteq B) \leftrightarrow \forall a(\neg(a \in A) \vee (a \in B))))$.

It should be emphasized that if the calculated similarity between the semantic graphs of PNF representation is not 1 ($F_{sc} < 1$), the similarity between the semantic graphs calculated for the first time is used as the final answer similarity.

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 [20], [21].

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.,

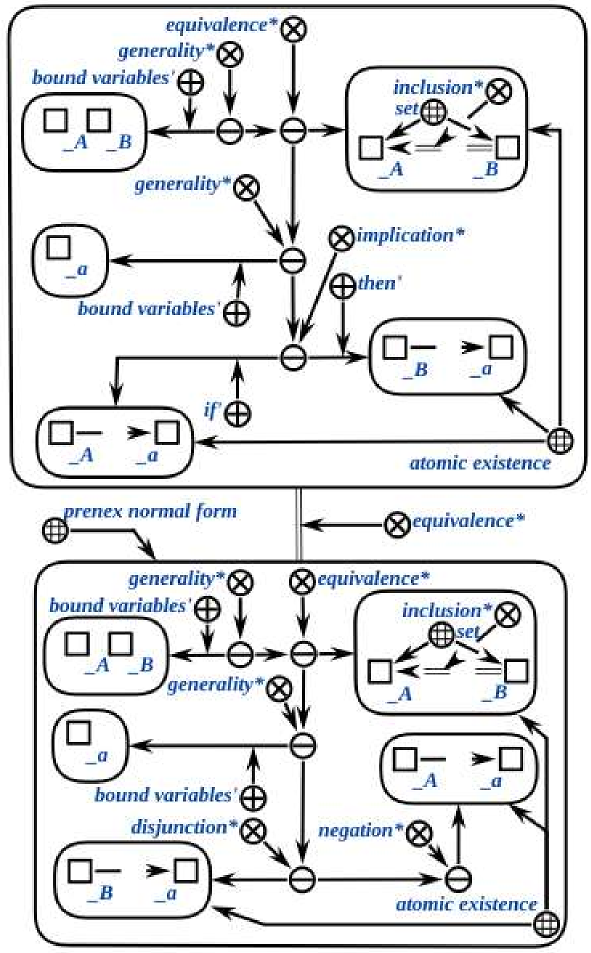

Figure 11. An example of converting a semantic graph into PNF representation.

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 [4].

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-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 [2], [4].

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 [4], [16]. 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 (3), (4) and (5) to calculate the precision, recall and similarity between answers.

Since this article focuses on presenting the fundamentals of the entire process of automatic generation of test questions to verification of users' answers in a holistic manner, it briefly describes the fundamentals of answer verification for subjective questions. In reference [4] we describe in detail the process of verifying user answers to subjective questions in the ostis-systems.

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 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.

H. Development of the subsystem knowledge base

The knowledge base of subsystem is mainly used to store automatically generated test questions of various types. 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 [2], [3], [8]. 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*
}
}

It should be emphasised here that objective questions can be divided into more specific types (e.g., multiple-choice question with multiple answer options and multiple-choice question with a single answer option, etc.) according to their characteristics and the corresponding test question generation strategy. In references [4] and [5] we describe in detail the categorisation of objective questions and the process of construction of their subject domains.

I. Development of 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 [2], [3].

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
$\Leftarrow$ *decomposition of an abstract sc-agent**:
- {• *Sc-agent for automatic generation of test questions*
  - $\Leftarrow$ *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*
  - $\Leftarrow$ *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.

IV. Evaluation of the effectiveness of the developed subsystems

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

- availability of the generated test questions;
- closeness between automatic scoring and manual scoring of user answers to subjective questions;
- reduced time costs due to the use of subsystem.

In order to assess the availability of automatically generated test questions using the knowledge base, 200 automatically generated test questions were taken from the discrete mathematics tutoring system and the euclidean geometry tutoring system, respectively. The percentage of test questions that could be used directly was also manually counted. The evaluation results of the availability of test questions showed that 94 % of the automatically generated test questions in the discrete mathematics tutoring system and 92 % in the euclidean geometry tutoring system could be used directly.

In order to evaluate the closeness between the results of automatic scoring and manual scoring of user answers to subjective questions, the following work was done. Firstly, 40 second year university students were randomly selected and their knowledge was tested using an automatically generated test paper containing 4 subjective questions (the maximum score for each subjective question is 10 grades). The answers of 40 students were then checked using the subsystem and manual methods respectively and the error between the automatic and manual assessment results of these students' answers was calculated (Table I).

Table I
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 (6):

$$\Phi = |x - y| \tag{6}$$

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 I, 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 %.

In order to evaluate the reduced time cost due to the use of the subsystem, a random sample of a class of second-year university students (30 students in total) was selected and the average duration of the process of testing the knowledge level of these students using the subsystem was calculated, as well as the duration of the process of testing the knowledge level of these students using the traditional examination method, respectively. Since it is possible to test the knowledge level of 30 students at the same time using the subsystem, the average duration of the testing process was counted. The evaluation result of reduced time cost showed that the use of the developed subsystem in the process of testing the knowledge level of 30 students can save 89 % time cost.

The duration of the development of the knowledge base should normally also be taken into account when assessing the reduced time cost due to the use of the subsystem. In reference [22], the duration of development a knowledge base for discrete mathematics using the reusable component approach was estimated to be 513 h. Therefore, the relationship between the average duration of generating each test paper using the developed subsystem and the number of test papers generated is evaluated in this article, taking into account the duration of the development of the knowledge base. The experimental result shows that when the number of test papers generated using the developed subsystem is more than 20,000, the impact of the development time of the knowledge base on the average duration of each test paper generated using the subsystem is negligible. It should be emphasised that the use for generating test questions is only one use of the knowledge base, and even if it is only used for generating test questions it can be seen from the evaluation result that it is profitable.

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

V. Conclusion

An approach to automatic generation of test questions using knowledge bases and automatic verification of user answers according to the semantic description structures of the knowledge in an intelligent tutoring system developed using OSTIS Technology is presented in this article. A subsystem for automatic generation of test questions and automatic verification of user answers has been developed in the ostis-systems based on the proposed approach. Using the developed subsystem allows automation of the entire process from generation of test questions to grading of test papers, which greatly improves the learning efficiency of the users.

Finally the effectiveness of the developed subsystem was evaluated in terms of the availability of the generated test questions, the closeness between the automatic scoring and the manual scoring of the test questions in the discrete mathematics ostis-system and reduced time costs due to the use of subsystem. 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] Jiang. H., Zhao. Y., Li. X. Intelligent education system: A comprehensive review. IEEE Access, 2020, Vol. 8, pp. 201016-201026.

[2] 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.

[3] 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.

[4] 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.

[5] Wenzu Li. "An automated approach to checking user knowledge levels in intelligent tutoring systems," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2023, pp. 269–278.

[6] 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.

[7] 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.

[8] (2024, Mar) Ims.ostis metasystem. [Online]. Available: https://ims.ostis.net

[9] 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.

[10] Kaur. M., Batra. S. Ontology-based question generation: a review. Journal of Intelligent and Fuzzy Systems, 2020, Vol. 39(01), pp. 49-57.

[11] 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.

[12] Chen. C. H. A comprehensive survey of text similarity measures. Journal of Information Science, 2019, Vol. 45(01), pp. 1-19.
[13] Singh. A., Singh. V. K. Text similarity techniques: a systematic literature review. International Journal of Computer Applications, 2019, Vol. 182(41), pp. 23-30.
[14] Mingyu. Ji., Xinhai. Zhang. A Short Text Similarity Calculation Method Combining Semantic and Headword Attention Mechanism. Scientific Programming, 2022.
[15] 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.
[16] 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.
[17] Fujiwara. M., Kurahashi. T. Prenex normal form theorems in semiclassical arithmetic. The Journal of Symbolic Logic, 2022, pp. 1-31.
[18] Liu. J., Liu. Y., Wang. X., Wang. B. A survey on ontology integration techniques. Future Generation Computer Systems, 2020, Vol. 105, pp. 17-30.
[19] 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.
[20] 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.
[21] Kryakin. A., Nguyen. P. Automated theorem proving in geometry using Gröbner bases. Journal of Automated Reasoning, 2014, Vol. 53(02), pp. 163-189.
[22] Davydenko, I. "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," in Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems], V. Golenkov, Ed., BSUIR. Minsk, BSUIR, 2018, pp. 99–118.

ПОДХОД К АВТОМАТИЗАЦИИ ВСЕГО ПРОЦЕССА КОНТРОЛЯ ЗНАНИЙ УЧАЩИХСЯ ОТ ГЕНЕРАЦИИ ТЕСТОВЫХ ВОПРОСОВ ДО ПРОВЕРКИ ОТВЕТОВ ПОЛЬЗОВАТЕЛЕЙ В ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

Ли Вэньцзу

Данная статья посвящена вопросам автоматической генерации тестовых вопросов и автоматической проверки ответов пользователей в интеллектуальных системах нового поколения, основанных на смысловом представлении информации. В статье подробно рассмотрен подход к реализации автоматической генерации тестовых вопросов различных типов с использованием баз знаний и автоматической проверки ответов пользователей в соответствии со структурой семантического представления знаний.

Keywords—генерация тестовых вопросов, проверка ответов пользователей, Технология OSTIS, интеллектуальные системы на основе смыслового представления информации, семантическая структура, база знаний

Received 29.03.2024