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

$\Leftarrow$ *inclusion*:*
directed tree

$\Rightarrow$ *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.

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

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

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

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.

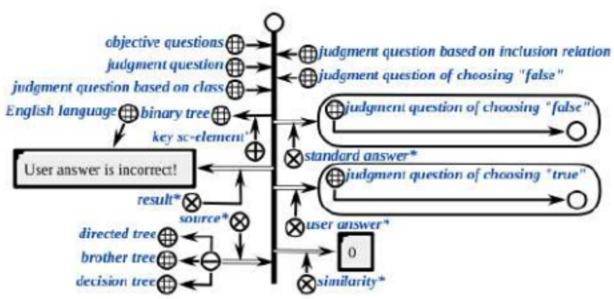

Figure 1: An example of verification of user answer to judgment question.

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 scconnector 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);