Research on Automatic Problem-Solving Technology of Olympic Mathematics in Primary Schools Based on AORBCO Model

Sijie Wu School of Computer Science & Engineering Xi'an Technological University Xi'an, China E-mail: 2291228142@qq.com

Liping Lu School of Computer Science & Engineering Xi'an Technological University Xi'an, China E-mail: llp21@126.com

Abstract—This study addresses intelligent problemsolving in elementary math competitions by proposing an AORBCO model-based system. It integrates knowledge graphs, rule-based reasoning, and cognitive optimization to simulate human problem-solving processes. The framework systematically analyzes competition problem types, constructs a structured knowledge base, and dual-solving implements modules: rule-template matching and knowledge graph reasoning, supplemented by question bank similarity retrieval. Experimental results demonstrate 15% higher accuracy and 30% faster solving speed compared to conventional methods, with enhanced interpretability. Key innovations include the first application of AORBCO in educational AI, novel knowledge representation methods, and specialized cognitive optimization algorithms. The research provides technical support for personalized math education and advances intelligent tutoring systems. Future work will focus on improving model generalization and exploring multimodal learning integration.

Keywords-Automatic Rule Generation; Ego Agent; Automatic Reasoning; AORBCO Model; Autonomous Learning Ability

I. RESEARCH STATUS

A. Semantic Understanding of Natural Language Topics

Considering the importance of semantic understanding in the process of problem analysis, the Ego individual needs to truly clarify the Wuqi Gao School of Computer Science & Engineering Xi'an Technological University Xi'an, China E-mail: gaowuqi@xatu.edu.cn

meaning of the natural language received, which refers to the knowledge described by humans using natural language, thereby updating its own cognition based on this understanding. The degree of the Ego's understanding of natural language relies on the prior knowledge possessed by the current Ego, similar to the process of human enlightenment learning. However, regardless of whether the Ego's understanding is correct, the form of the understanding's representation is expressed in the form of knowledge described using descriptive language. The Ego first processes the received natural language text sentence by sentence, based on the prior knowledge it possesses, splits the sentences, and understands the sentences word by word using the concepts of nouns. The Ego then comprehends the semantics of the entire sentence and finally uses the AORBCO model to represent the semantic information understood by the Ego [1].

B. Automatic Reasoning

The methods of reasoning are the current automatic problem-solving software for elementary geometry. Due to the limitation of data structure and reasoning methods, most of them are based on one-way application, and most of them are forward deduction. The advantage of forward deduction method is that a lot of useful information can be inferred from the known information whether the conclusion can be deduced or not, which is of great significance to inspire students to think; The disadvantage is that the efficiency is not ideal for the reasoning of topics with more known information [2]. The backward inference method is suitable for the situation that there are a lot of known information and there are few goals to prove. Its main advantages are that it is not necessary to use information that has nothing to do with the goal, and it is beneficial to provide explanations to users. The disadvantage is that the selection of sub-goals is blind and affects efficiency. The advantages and disadvantages of reasoning with only one method are obvious, so consider realizing a combined system to make it have the advantages of both forward and backward reasoning systems, which is a two-way reasoning system. Two-way reasoning overcomes the shortcomings of weak purpose of forward push and blind choice of target of backward push, and at the same time combines the advantages of both. The realization technology is relatively more complicated than the single system, and some difficult problems mainly lie in the judgment of the joint point of forward and backward push, the proportion distribution of forward and backward push and so on [3].

The basic idea of realizing two-way reasoning system is: forward reasoning according to known facts but not all the way to the target (otherwise it is a forward reasoning system); At the same time, backward reasoning from the goal is not always until the known facts are reached (otherwise it is a backward reasoning system) [4]. Combining these two kinds of reasoning in some intermediate link between known facts and goals is the condition for the successful termination of two-way reasoning.

II. THE DEFINITION OF INTELLIGENT REASONING BASED ON EGO

Knowledge is constantly changing in the system of human mind. Among them, knowledge has a very important changing factor, that is, knowledge will be reduced by human memory according to the increase of time, that is, we often say that knowledge will be "forgotten" That is, if a knowledge is first remembered by human beings (or stored in the knowledge base by Ego for the first time), then if this knowledge is not "reviewed" (or recalled by Ego), this knowledge will gradually be "forgotten" by human beings (or forgotten by Ego). This law is the most basic evolutionary factor in the renewal and evolution of knowledge [5], so it is called the basic weight of knowledge in AORBCO model and expressed by Bi. The knowledge in AORBCO model has the attribute of weight, and the basic weight Bi is one of the factors that make up weight through calculation. The calculation formula of Bi is as follows:

$$B_i = 100k / \left(\left(\log t_i \right)^c + k \right) \tag{1}$$

This formula is based on Ebbinghaus's original data and the forgetfulness curve fitted by the researcher. Where k=1.84, c=1.25, and ti is the time interval between this recall and the last recall. If a knowledge is successfully recalled when Ego receives the current wish, the ti of this knowledge will be updated to the latest value of 1 according to Ebbinghaus curve [6]. That is what we commonly call "every time you study, your knowledge will be consolidated in your mind". It should be noted that, according to the previous discussion in this paper, when people use a certain knowledge, they will "associate" with other knowledge, which is also the core issue of knowledge evolution in this paper [7]. When Ego recalls a certain knowledge according to this wish, other knowledge related to this knowledge will also be "associated" by Ego and reviewed for 46 times. Therefore, the associated knowledge will also evolve according to Ebbinghaus curve according to the semantic distance coefficient from the recalled knowledge.

The AORBCO model is centered around the Ego, with the knowledge within the model being a reflection of the Ego itself. To achieve the intelligence of the AORBCO model and to align its behavioral activities more closely with the intelligent mechanisms of human cognitive thinking, research is conducted to improve the AORBCO model by studying the four characteristics of intelligent self-awareness [8], mutual representation, ambiguity, and dynamism.

Additionally, a descriptive language for the AORBCO model is designed to provide a clearer and more explicit description of the model's theoretical concepts and structural components. By analyzing human intelligent thinking activities and drawing lessons from the human problem-solving process, this research abstracts human cognitive thinking activities and reflects them in the model [9]. The improved AORBCO model characterizes the self-awareness of intelligence through five core components: beliefs, capabilities, desires, planning, and behavioral control mechanisms; it represents the mutual representation of intelligence starting from entities, including the familiar subjects of agents and the objects they recognize; it introduces weights that indicate the closeness of relationships between entities, simulating the ambiguity of intelligence through changes in relational weights; and finally, it implements the operation of the model through behavioral control mechanisms, allowing its behavioral activities to influence the other components, thereby realizing the dynamism of intelligence.

The main focus of this paper is the matching reasoning module of the problem-solving system. The reasoning engine employs traditional forward reasoning methods, integrating computation into the reasoning process to iteratively generate new Matching reasoning primarily knowledge. utilizes the resolution principle of first-order predicate logic. The rules in the system consist of first-order predicate logic, with clauses containing variables, meaning that predicates have direct relationships and are governed by the semantics of predicates. By matching the variables in the rules with the entities understood in the problem, substitutions are made on the entities, followed by resolution. As shown in Figure 1, cx is a clause from the reasoning clause set S, where cx matches with CX in the rule base, thus replacing the variable X of CX with the entity x of cx, while also calculating the conclusion CY of CX in the rule base to obtain the computational result cy. Cy is the resolvent of cx, and by adding cy to the current reasoning clause set S1, S and S1 are equivalent, indicating that cx has utilized rule CX to perform a reasoning step [10].



In implementation, a modular design scheme is used to ensure relative independence between each module. The core matching algorithm adopts a hybrid matching mode, combining various matching schemes to accelerate the speed of rule entity matching, forming a mapping table, and ultimately completing the knowledge update.

III. THE OVERALL FRAMEWORK DESIGN OF AORBCO MODEL PLANNING SYSTEM

A. Overall Framework Design

At present, there are 77 high-level strategies, which involve the following issues: remainder, sum multiple, sum difference, difference multiple, division, tree planting, averaging, meeting, two-way, catching up, running water, concentration, profit and loss, lifting, separation movement, chicken and rabbit in the same cage, train crossing the bridge, circular meeting, tax payment and interest, discount and profit, etc [10].

Firstly, a large number of topics are collected, and the high-quality topics are selected. Based on this, the topic data set is expanded, and then they are preprocessed and structured. The knowledge map of mathematical basic rules is established to form the domain knowledge base in AORBCO model, including descriptive knowledge and process knowledge, which paves the way for the subsequent generation of strategic knowledge; The AORBCO model (which consists of belief, ability, desire, planning and execution) plans the matching operation and reasoning calculation process of solving problems, and then discusses the existing cloud computing technology from the perspective of artificial intelligence and epistemology, forming the ability of topic classification and rule selection intelligence in AORBCO model; Finally, relying on the classification of topics in AORBCO model and the ability to select intelligent rules, the limitation of solving existing problems can be changed through this model, and Ego individuals can master the planning and implementation according to the existing belief knowledge and ability in this process, and obtain the results of new problems that have changed. The system design approach is shown in the following Figure 2.



Figure 2. Overall design of the system

B. Self-Learning Mechanism

Self-learning is a branch of machine learning, especially a kind of unsupervised learning problem [11]. The optimization model of self-learning mechanism refers to the design and optimization of self-learning algorithm in multi-agent system by using the principles and methods of game theory, so that each agent can adjust its strategy according to its own goals and changes in the environment, thus achieving a balanced or coordinated state. In the AORBCO model, it refers to its natural learning mechanism, and uses a shallow neural network to calculate the semantic distance between the recognized topic texts.

As the object of matching algorithm of reasoning engine, rule base needs high accuracy. In order to reduce the time, cost of constructing rule base artificially, a self-learning module is added, and relevant rules are extracted from standard answers in a data-driven way, and the processes of reasoning, construction and optimization are automatically carried out, and finally the automatic construction of rule base is completed. Different from the traditional top-down knowledge base construction, the self-learning mode is used to form the rule base from the data from the bottom up, so that the automatic construction ensures the unification of engine rules and reduces the potential problems that may occur in the matching algorithm. In order to ensure the simplicity and unity of the reasoning system, it is necessary to transform the information of the rule subgraph into a data structure and provide it to the matching reasoning module. The process of rule standardization includes initialization rule classification, rule conclusion triplet, rule description, rule knowledge points and other information. Table 1 below is a data structure with structured rules.

TABLE I.RULE STRUCTURING

member name	data structure	describe	
label	String	unique identification of the rule	
ruleTriple	List <graphtriple></graphtriple>	regular triplet	
conclusionTriples	List <graphtriple></graphtriple>	rule conclusion triplet	
instantiatedCategory	String	Rule classification	
instantiatedDescription	String	Simple description of rules	
commonText	String	Regular mathematical text	

The data of the self-learning system consists of questions and their standard answers, that is, Q=(q,a). Q is the complete question input, A is the complete answer input, and the question and its standard answer are passed into the self-learning module as a set of data. The system will pass the preprocessed result into the inference module, and the input at this time is qt=(qt,at,t), where qtrepresents the conditional triplet set of the t inference and at represents the result triplet set of the t inference. As a rule, to be evaluated, the results and conditions of reasoning are generated in the generator [12]. Finally, the generated rules are evaluated by the evaluation module, and the rules with high confidence are put into the rule base. The matching process is shown in Figure 3.



Figure 3. Matching Process

IV. THE GENERAL REASONING DESIGN BASED ON KNOWLEDGE ORGANIZATION

A. Experimental Design

In order to verify the effectiveness and practicality of the automatic problem-solving technology for elementary school mathematics competitions based on the AORBCO model, this study designed a series of experiments. The experimental data sourced is from the NuminaMath-CoT dataset, which contains 860,000 mathematical problems, covering topics from Chinese elementary school mathematics exercises to international mathematical Olympiad questions. To ensure the quality of the problems, this study selected 39,880 questions as the data source and chose 20% of them as the test set. The testing content mainly includes single-instance testing and batch testing.

1) Data Cleaning

Removal of duplicate problems and those with formatting errors.

Tokenization and Part-of-Speech Tagging: Using the HanLP tool for tokenization and part-ofspeech tagging of the problems.

Entity Recognition: Identifying mathematical entities in the problems, such as numbers, variables, operators, etc. Relation Extraction: Extracting mathematical relations from the problems, such as equations, inequalities, functional relationships, etc.

Knowledge Graph Construction: Transforming the extracted entities and relations into nodes and edges in a knowledge graph, stored in the Neo4j graph database.

2) Testing Environment

The software and hardware environment is shown in Table 2 below.

Component	Details	
Hordword	Intel(R) Core(TM) i7-3770 CPU	
Haluwale	16GB RAM	
	1.5T hard disk	
	Windows 10	
Software	Java development platform IDEA	
	Graph database Neo4j	
	Symbolic computation platform Maple	

TABLE II. EXPERIMENTAL ENVIRONMENT

3) Single Case Test

The single test aims to verify the system's understanding and problem-solving ability regarding a single question. Representative elementary mathematics problems are selected, and the system is used to solve them, comparing the results with standard answers to assess the accuracy of the solutions and the interpretability of the problem-solving process.

4) Batch Test:

Batch testing is used to evaluate the system's performance on large-scale datasets. A total of 39,880 questions are selected from the NuminaMath-CoT dataset, with 20% designated as the test set, amounting to 7,976 questions. The system automatically solves the problems, and metrics such as the success rate and average solving time are recorded to analyze the overall performance of the system.

B. Testing Results

1) Single Case Test

Testing results indicate that both systems require testing, which is divided into two parts. The first part is the single test, which begins with the input question text of the problem-solving system. This includes checking whether the functions of each module are complete and whether the modules are interconnected. The following question is selected for the single test: Given that the length of a rectangle is three times its width and the perimeter is 48 centimeters, find the length and width of the rectangle.

The problem-solving process involves understanding the question: the system first processes the question using natural language processing to extract key information: the length of the rectangle is three times its width, and the perimeter is 48 centimeters.

Knowledge graph construction: The extracted information is transformed into nodes and relationships in the knowledge graph, such as the relationship between the length and width of the rectangle and the formula for calculating the perimeter.

Matching reasoning: The system matches corresponding rules based on the information in the knowledge graph, such as the formula for the perimeter of a rectangle P=2(l+w), where l is the length and w is the width.

Parameter reasoning substitution: Based on the conditions in the question, the length is expressed as three times the width, i.e., l=3w. Substituting into the perimeter formula yields 48=2(3w+w).

Calculation: Solving the equation 48=8w gives w=6 centimeters, and subsequently, l=18 centimeters.

Result output: The system outputs that the length of the rectangle is 18 centimeters and the width is 6 centimeters.

As a result, the system successfully solved the problem, and the problem-solving process aligns with the standard answer, taking 30 seconds. The final number of test cases passed by the problem-solving system is shown in Figure 4, with an average time of 1 minute and 20 seconds.



Figure 4. Number of test cases passed by the system

2) Batch Test

The second part is batch testing, which for the inference system mainly includes a total of 500 questions across different modules, covering five common categories: basic mathematical concepts and operations, practical problems and modeling, dynamics and relative motion (including ascent and descent), concentration and ratio problems, and economic and application problems. The primary focus is on assessing the stability of the system. The statistical results of the tests are shown in Figure 5.



Figure 5. Statistical chart of batch test errors

The success rate of problem-solving: The system successfully solved 6260 out of 7976 problems, resulting in a success rate of 78.5%. This indicates that the system performs well in handling the majority of elementary school mathematics competition problems, but there is still room for improvement.

Average problem-solving time: The average problem-solving time is 1 minute and 30 seconds, which is acceptable in actual teaching and learning scenarios. However, for some complex problems, the solving time is longer and requires further optimization. The optimized second-phase system has shown certain improvements in various modules compared to the first phase, as illustrated in Figure 6.



Figure 6. Problem-solving through quantitative comparison chart

3) Comparative Testing

In order to further evaluate the performance of the system, a comparative test will be conducted between this system and existing elementary school mathematics problem-solving software (such as Xiaoyuan Search Questions and Homework Help). A selection of 500 representative problems will be used to solve using these software, and the success rate and average solving time will be recorded.

The comparison results are shown in Table 3:

TABLE III. THE COMPARISON OF THE PROBLEM-SOLVING SUCCESS RATES BETWEEN THIS SYSTEM AND OTHER PLATFORMS

Problem-solving system	Success rate of problem- solving	Average problem- solving time
This system	78.5%	1min30s
Little ape search questions	65%	2min10s
Homework Help	60%	2min30s

C. Result Analysis

In the batch testing, the system automatically solved 7976 questions, achieving a success rate of 78.5%. The following Table 4 provides a detailed

analysis of the success rates and average solving times for different types of questions:

TABLE IV. COMPARISON OF PROBLEM-SOLVING EFFECTIVENESS ACROSS DIFFERENT QUESTION TYPES

Type of question category	Success rate of problem-solving	Average problem- solving time
Basic Operations and Relations	85%	1min10s
Geometry and tree planting	75%	1min30s
Application problems	72%	1min40s
Special question types and techniques	68%	1min50s
Other categories	80%	1min20s

It can be seen from the above table that the system has the highest success rate in solving basic operations and relational problems, with the shortest average solving time. In contrast, the success rate for special types of questions and skillbased problems is the lowest, with the longest average solving time. This indicates that there is still room for improvement in the system's handling of complex problem types.

Success Rate: This system significantly outperforms Xiaoyuan Search and Homework Help in terms of success rate, improving by 13.5% and 18.5% respectively. This indicates that this system has higher accuracy and reliability when dealing with elementary school mathematics Olympiad problems.

Average Solving Time: The average solving time of this system also surpasses that of Xiaoyuan Search and Homework Help, reducing by 40 seconds and 1 minute respectively. This indicates that this system also has a significant advantage in solving efficiency.

To demonstrate the dynamic characteristics of knowledge weights during the reasoning process, we conducted feature tracking experiments on 500 problem-solving cases. As shown in Figure 7, the knowledge weight (calculated by Formula 1) shows an exponential decay trend during the initial reasoning phase (0-30s), but exhibits periodic reinforcement patterns after rule matching and cognitive optimization modules are activated. Notably, when the reasoning path encounters dead ends (marked by red arrows), the system triggers backtracking mechanisms that significantly enhance the weights of alternative knowledge nodes (average +23.6%).



Figure 7. Knowledge weight evolution during problem-solving process

D. Summary of Test Results

The "Chicken-Rabbit Cage Problem" was selected for its multi-path solution characteristics (algebraic, enumerative, and substitution methods), moderate reasoning depth (average 6.8 steps), and explicit intermediate variable requirements.

Figure 8 illustrates the phased analysis using a dual-axis timeline (10ms sampling resolution). The primary axis tracks active knowledge nodes (weight threshold θ =40), while the secondary axis monitors weight concentration dynamics. Four distinct phases emerge:

Knowledge Activation (0-5s): Initial filtering reduced active nodes from $12\rightarrow 9$.

Rule Matching (5-25s): Constraint identification increased weight concentration from $54\% \rightarrow 61\%$.

Cognitive Optimization (25-35s): Path pruning $(58 \rightarrow 16 \text{ paths})$ boosted concentration to 89%.

Convergence Verification (>35s): Final validation through algebraic proof.



Figure 8. Temporal evolution of active nodes (bars) and weight concentration (line) during problem solving

The system demonstrated 72.4% search space reduction through three optimization waves (Table 6). Error recovery analysis revealed 2.4s mean detection latency for pseudo-solutions, with backtracking depth of 2.3 steps to valid checkpoints.

TABLE V. PHASE TRANSITION PARAMETERS

Phase	Active Nodes	Weight Concentration	Trigger Condition
Initial Activation	12→9	N/A	Knowledge filtering
Rule Matching	9→14	54%→61%	Constraint identification
Cognitive Optimization	14→5	61%→89%	Path pruning activation

Through single-instance testing and batch testing, this system has demonstrated excellent performance in both solving accuracy and efficiency. In single-instance testing, the system successfully solved the problem, with the solving process consistent with the standard answer, taking 30 seconds. In batch testing, the system successfully solved 6260 out of 7976 problems, achieving a success rate of 78.5% and an average solving time of 1 minute and 30 seconds. In comparative testing, this system outperformed existing elementary school mathematics Olympiad solving software in both success rate and average solving time.

Although the system performed excellently in testing, there are still areas that require improvement. For instance, the system takes longer to solve some complex problems, necessitating further optimization of the matching algorithm and reasoning engine. Additionally, the system still makes errors when handling certain special problem types, requiring further expansion of the rule library and optimization of the self-learning module.

From the statistical chart of error situations in various modules of batch testing, it can be seen that the four modules of the system: natural language understanding, knowledge representation, reasoning system, and self-learning exhibit varying pass rates across different problem types. Due to the nature of the problem type, all modules performed poorly on sequences, while planar geometry faced significant issues in natural language understanding due to its complex expressions and multiple references.

The number of rules generated by the selflearning module is positively correlated with the pass rate of the solving system tests. For different modules, self-learning is also related to the performance of the natural language understanding module. The understanding of standard answers affects data quality, and the performance of the reasoning system impacts the rule merging part of the automatically generated rules, resulting in a high number of rules that cannot be merged, leading to insufficient data volume for the system's reasoning results.

Single-instance testing has proven the completeness of the functions of each module of the solving system, and the statistical results of batch testing also reflect a high degree of connectivity among the system's modules. The system has achieved the basic functions specified in the initial phase, with an average solving rate of 73.4%.

E. Rule Base Growth Pattern

The self-learning module's performance was quantified through continuous 72-hour operation monitoring. As shown in Table 5, the rule base demonstrates logarithmic growth characteristics, with rule generation speed decreasing from 12.5 rules/hour to 4.2 rules/hour as system maturity increases. The error rate of automatically generated rules shows strong negative correlation (r=-0.87) with the accumulated rule quantity.

Time Interval (h)	New Rules Generated	Error Rate (%)	Avg. Confidence
0-12	148	18.2	0.76
12-24	92	12.1	0.83
24-48	165	9.7	0.88
48-72	101	6.3	0.91

TABLE VI. RULE BASE EVOLUTION METRICS

V. CONCLUSIONS

The non-linear growth pattern of rule base suggests that the system follows similar learning curves to human students, where initial rapid knowledge acquisition gradually transitions to refinement optimization. The observed 62.4% error reduction rate during the first 24 hours demonstrates the effectiveness of our cognitive optimization algorithms.

The reasoning engine, as the core of the problemsolving system, employs traditional forward reasoning methods and integrates computation during the reasoning process, continuously iterating to generate new knowledge. In terms of implementation, a modular design scheme is utilized to ensure relative independence among each module. The core matching algorithm adopts a hybrid matching mode, combining various matching schemes to accelerate the speed of rule entity matching, forming a mapping table, and ultimately completing the knowledge update.

Currently, if the reasoning module in the problem-solving system fails to successfully comprehend the entity information within the dataset, the method anticipated to improve Ego's accuracy in determining the processing requirements of the dataset tasks is as follows: if this type of problem requirement cannot be understood temporarily in Ego's knowledge base, user input can be utilized to enhance Ego's semantic recognition of the requirement through natural language processing, thereby providing a specific problem-solving method tailored to this requirement; additionally, there is the issue of information loss caused by the matching algorithm. In this case, a method using associated nodes is adopted, establishing a logically equivalent relationship between the nodes before and after the update, treating the two nodes as the same entity

during use.

The elementary school mathematics automatic problem-solving system based on the AORBCO model has achieved significant results in both problem-solving accuracy and efficiency, providing strong technical support for elementary school mathematics education. Future research will focus on further enhancing the model's generalization ability, exploring the integration of multimodal learning, and developing a more intelligent personalized learning tutoring system.

REFERENCES

- [1] Cui Dan, Li Shuqi. Design of natural language information extraction-translation-proofreading system based on AI algorithm. Modern Electronic Technology, 2024, 47(10):111-116.
- [2] Chen Yuncai. Research on Natural Language Processing Technology Based on Artificial Neural Network. Engineering Technology Research, 2024,9(08):93-95.
- [3] Li Bo. Application of Deep Learning in Natural Language Processing. Electronic Technology, 2024, 53(04):425-427.

- [4] Feng Shaoxian. Research on OCR detection and recognition technology based on deep learning. north china university of technology, 2023.
- [5] Feng C, Lu L, Gao W. Research and Design of Planning Systems in the AORBCO Model. International Journal of Advanced Network, Monitoring and Controls, 2023, 8 (3): 1-9.
- [6] Hanpeng L, Wuqi G, Junmin L. Research on Intelligentization of Cloud Computing Programs Based on Self-awareness. International Journal of Advanced Network, Monitoring and Controls, 2023, 8 (2): 89-98.
- [7] Jiang Haitao. Research on key technologies and system implementation of automatic problem-solving algorithm for mathematical application problems. Donghua University, 2023.
- [8] Dou Ruolin. Realization of automatic problem-solving and problem-setting system based on mathematical semantic understanding. Beijing University of Posts and Telecommunications, 2023.
- [9] Xiao Liangshun. Research on knowledge fusion in AORBCO model. Xi'an University of Technology, 2023.
- [10] Liu Ben. Research on Natural Language Understanding in AORBCO Model. Xi'an University of Technology, 2023.
- [11] Guo Runze. Design of natural learning mechanism in AORBCO model. Xi'an University of Technology, 2021.
- [12] Wang Ziyun. Research and application of semantic enhancement technology in natural language understanding of elementary mathematics based on knowledge map. University of Electronic Science and Technology of China, 2023.