Received May 3, 2021, accepted May 19, 2021, date of publication May 25, 2021, date of current version June 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3083540

SCADA-NLI: A Natural Language Query and Control Interface for Distributed Systems

HAO WU[1], CHUNSHAN SHEN[1], ZHUANGZHUANG HE[1], (Student Member, IEEE), YONGMEI WANG[1], AND XINYUAN XU[2]

[1]College of Information and Computer, Anhui Agricultural University, Hefei 230036, China
[2]School of Business, Hunan University, Changsha 410000, China

Corresponding author: Chunshan Shen (csshen@ustc.edu)

This work was supported in part by the Major Research and Development Projects of Anhui Province of China under Grant 1804a07020108.

ABSTRACT Human–Computer natural language interaction is helpful to reduce the operation and maintenance cost of the SCADA system, and it is necessary to solve the complex natural language interface problem that supports data query and real-time control. According to the complexity of natural language instructions, a hierarchical classification of semantic parsing algorithm is adopted. Firstly, the KWECS method is used to classify the intent of natural language instruction, then the TF-IDF keyword extraction algorithm combined with the cosine similarity is used to structure the key-value of the classified natural language instructions which was used into SCADA control intermediate language and then formally converted into actual control or query instruction. If the analysis fails, the complex control and query instruction analysis are carried out according to the classification results, structuring instruction parsing based on dependency parsing and SQL natural language parsing based on deep learning are adopted respectively to implement real-time control interface and database query interface. Our experimental results show that the proposed hierarchical classification of natural language comprehensive query and control interface can better solve the problem of human-computer natural language interaction in the SCADA system, and the accuracy of intent recognition reaches 96.5%. In more detail, the accuracy, precision, recall, and F-score of instruction parsing reach 88.47%, 90.21%, 89.48%, and 89.72% respectively. Especially, it provides more convenient interactive means for industrial and agricultural information management and control.

INDEX TERMS Natural language interface, intent classification, semantic parsing, scada systems, human–computer interaction.

I. INTRODUCTION

Distributed SCADA (Supervisory Control And Data Acquisition, system control, and data acquisition) is usually used for integrated monitoring of large industrial processes and power systems. However, general SCADA system construction and operation and maintenance costs are high, so special operation and maintenance personnel are required and almost all of them rely on the graphical user interface (Graph User Interface, GUI) for human-computer interaction [1]. By observing the chart data, clicking on the button control to set up and deploy the SCADA system, operation and maintenance engineers with special knowledge are needed to complete the limited operation under the fixed program through the Windows interface. This human-computer interaction mode is not friendly and the operation efficiency is low.

To solve this problem, it is also fundamentally necessary to improve the natural properties of human-computer interaction. The natural language interface of the SCADA system is an effective method. Natural Language Interface (NLI) uses human language to interact with machine systems, reducing the complexity of interaction [2], which is particularly important in the manipulation of complex systems and devices, such as large distributed SCADA systems [3], UAVs, autopilot, etc. Natural language interfaces are also being widely used in mobile and Web applications, data management, and other aspects. For example, some smart softwares provide natural language interfaces for mobile Android terminals, such as the voice assistant [4] of Microsoft, Apple, and Baidu; they construct natural language interfaces for the web, realizes the

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

application of natural language operation for web e-mail and other applications, and improves the use efficiency of web applications [5]. And also, the preliminary research on Chinese Text-to-SQL algorithms for Chinese natural language query database provides a new way of thinking [6].

The task of the natural language interface is to convert user-expressed speech or text into an executable computer system call interface. The core issue is to address the following two steps:

(1) Analyze the intention of manipulation and the input/output call relationship between complex manipulation sequences. The core is the formal semantic function description of natural manipulation language;

(2) The mapping between natural language instruction sequences and computer system call interfaces.

The first step is the semantic understanding problem in natural language processing, which is also the essential and fundamental problem to solve the natural language interface. Usually, it needs to go through the process of text disambiguation, lexical annotation, named entity recognition, and syntactic analysis to identify the user manipulation intention through pattern classification and semantic understanding. Regarding the second step, after expressing the manipulation sequence and system call interface based on the first step, it mainly solves the problem of similarity calculation and optimal mapping, and its effect depends largely on the first step.

There are four main categories of research on intent recognition of manipulation: rule-based approach, query-and-click log-based approach, statistical feature-based classification approach, and deep-learning approach. The rule-based approach uses information such as keywords to form rules to detect user intent. Chu and Huang [7] proposed a dialogue-based object query system, which is combined with cosine similarity and TF-IDF to determine user intent. Regarding intent classification based on query click logs, it is mostly used for business scenarios such as search engine, and user intention can be obtained by clicking log. Ye and Ma [8] proposed a web page classification method based on query logs. Aiming at the problem that the sparsity of user click data makes it difficult for web pages to be directly used to construct classification pain points, he explored the semantic relations among different queries through word embedding, and proposed three improved graph structure classification algorithms. Veilumuthu and Ramachandran *et al.* [9] put forward proposed the concept of intention cluster. The members of the intention cluster have the same intention, and use the user session information in the query log and query URL entries to identify the query cluster with the same intention, which solves the problem of synonym query and poly-meaning query in keyword search technology. The method based on statistical feature classification needs to extract the key features of corpus text and then train the classifier to realize the intention classification. Common methods include Naive Bayes, Support Vector Machines and Logical Regression. Setyawan *et al.* [10] proposed a polynomial Naive Bayesian classification method for intent recognition. In this study, chatbots were created who could understand the natural language input of the user and respond according to the user's expectations. Mendoza, Marcelo *et al.* introduced a high-precision query classification method based on informational, navigational, and transactional classification, using a support vector machine as a classifier to identify the intention of the user's query, and pro-posed that time is an important factor to improve the classification accuracy. The usual practice based on deep learning first carries out text vectorization, and the commonly used methods include Word2vec [11], Glove [12] and BERT [13], etc., and then uses models for feature extraction. The commonly used models include LSTM [14], GRU [15] and CNN [16], etc. Finally, the Softmax layer is used to complete intention classification. Zhou *et al.* [17] proposed an improved structure of bilstm-attention for text classification, but it needs further optimization and improvement in aspects such as ''dimensional disaster'', ''semantic gap'' and ''high complexity'' [18], [19]. In general, deep learning requires a large amount of high-quality training data. Theoretically, deep learning model is suitable for instruction intention classification tasks, but the requirements for computational power and data are relatively high.

At present, the formal description and instruction translation of natural language instructions are mainly based on the natural language processing or natural language understanding technology combined with statistical and rule methods [20]. Especially with the development of deep learning theory, distributed feature representation has brought a breakthrough to the development of natural language processing. According to the neural language model proposed in [21], a large-scale corpus is an input as training data to obtain the vector form of words in the corpus, i.e. the distributed representation of words. Then, the NLP downstream tasks such as automatic Q & A, natural language interface, etc. are performed through the obtained word vector. Su *et al.* [5] proposed the first end-to-end framework which constructs NL2API for network application programming interface, i.e. the natural language is converted to the corresponding API, making network data, services, and device access more efficient. Sowmya Kamath *et al.* [22] proposed a framework for web-oriented discovery of composable service sets according to the complex needs of users. The method is based on natural language processing and semantic understanding to resolve the functional semantics of service data sets, and can effectively find relevant services for simple and complex queries. However, when considering the difference between complex query and simple query, only a few keywords are used to distinguish, which is easy to cause ambiguity. Tian *et al.* [23] proposed a structured processing method of pathological reports based on dependent syntax analysis, which converts the statements into the form of a dependent syntax tree, and then iterates through the nodes to obtain keywords in turn. There is a problem that there are too many nodes to extract, which leads to information redundancy. In recent years, the natural language interface for the database has become a research hotspot. Xu *et al.* [24] use the column-attention mechanism to deal with the generic problem of 'sequence-to-collection ', avoiding sequence correlation in the where clause. Yu *et al.* [25] proposed the TYPESQL architecture, like SQLNet, still treats SQL statement generation as a slot fill problem, but used type information to better understand rare entities and data in natural language problems. Hwang *et al.* [26] discussed three variations of the BERT-based architecture and presented how to use word contextualization in semantic analysis tasks.

In conclusion, the design of a natural language interface for human-computer interaction plays an active role in reducing the operation and maintenance cost and service upgrade cost of the large distributed SCADA system. However, some problems need to be studied urgently: lack of small data-driven, strong generalization ability and low calculation amount of intelligent computing model to guide the design of natural language interface, to solve the design problem of natural language interactive interface in the complex uncertain language environment. Based on the design of natural language query and control interface for a SCADA system, combined with semantic analysis and deep learning theory, a scientific, reasonable, and generalization ability of natural language interface for many fields should be studied. Contribution to this document is as follows:

- A natural language query and control interface based on hierarchical classification semantic parsing algorithm is designed, which can effectively identify and parse basic natural language query and control instructions and complex natural language and control instructions.
- Experiments show that the proposed hierarchical and classified natural language integrated control interface can better solve the human-computer natural interaction problem of the SCADA system, and achieve good accuracy and time performance to meet the requirements.

The structure of the paper is as follows: Section 2 discusses the requirements of the SCADA system and four types of natural language instructions. Section 3 describes the module functions and workflows of the SCADA-NLI architecture. Section 4 describes the algorithm for the classification of natural language instructions. Section 5 describes how to embed the key-value parameter fill sketch into the semantic parsing algorithm to solve the basic natural language query and control instruction and complex control instruction, and use the NL-to-SQL model to complete the semantic parsing of complex natural language query instruction. Experimental results show that the method used and the theoretical performance analysis of the proposed technology are given in Section 6, and finally the full-text summary and future work prospects.

II. SCADA SYSTEM REQUIREMENTS

The main functions of the SCADA system include data acquisition, device control, abnormal alarm, and parameter adjustment. Taking SCADA application scenarios such as smart home and smart agriculture as examples, SCADA system instructions are mainly divided into query instructions and control instructions. The query instruction is used for querying parameters such as historical data, charts and operation records, etc. According to the complexity of the query instruction, for example, in technical implementation, it can be realized both by calling the HTML interface linked with HTTP address and by accessing the built-in database of SCADA system through SQL query statement; the control instruction is used for remote control such as operation, parameter modification, etc. such as calling in the Web interface JavaScript methods are used to send commands to the server, which are then further distributed to gateways, nodes, and other field devices. To better describe the natural language query and control interface problem studied in this paper, a specific formal summary of the types of instructions involved in this paper is presented.

Basic control instructions: For example, "请帮我打开大棚的灯光"(“please turn on the lights of the shed”), "请帮我打开田间的传感器"(“please turn on the sensors in the field”), "请帮我升高大棚空调的温度"(“please raise the temperature of the air conditioner in the shed”), etc. These are basic natural language control instructions that do not contain nested relationships or only contain one action.

Basic query instructions: For example, For example, “请问现在大棚的温度是多少”(“What is the current temperature of the greenhouse”), 请帮我查询一下现在田间的二氧化碳浓度”(“Please help me to check the current CO2 concentration in the field”), etc. These are basic natural language query instructions that do not contain sorting, grouping, calculation, or other nested rules.

Complex control instructions: Consider the possibility that a real-world operation might output instructions containing multiple actions at once, such as opening or closing multiple devices at the same time, or opening different devices in multiple locations, and so on. For example, “请分别帮我 打开卧室的台灯和厨房的油烟机” (“Please help me turn on the lamp in the bedroom and the lampblack machine in the kitchen respectively”), “请先升高大棚的温度,然后关闭田间的灯光,最后降低,二氧化碳的浓度”(“Please first raise the temperature of the greenhouse, then turn off the light in the field, and finally reduce the concentration of carbon dioxide”), "请帮我分别打开卧室的台灯和客厅的空调”(“Please help me turn on the lamp in the bedroom and the air conditioner in the living room respectively”) and so on. Therefore, the instruction types similar to those mentioned above are summarized as complex natural language control instructions.

Complex query instructions: Instructions that contain one or more of sorting, grouping, computing, multi-conditional matching queries, etc. with nested rules are called complex query instructions. Examples of instructions are shown in Table 1.

TABLE 1. Types of questions and examples.

Question Types	Examples of complex natural language query instructions
Conditional Query	昨天大棚里温度超过20度的时刻有哪些? (What were the moments in the greenhouse yesterday when the temperature was over 20 degrees?)
Group Query	哪个时刻的卧室的二氧化碳浓度最高? (Which moment has the highest CO2 concentration in the bedroom?)
Calculated Query	昨天机房里最高湿度比最低湿度高多少? (How much higher was the highest humidity than the lowest humidity in the server room yesterday?)
Sort Query	今天客厅温度前3的时刻有哪些? (What are the top 3 living room temperature moments today?)

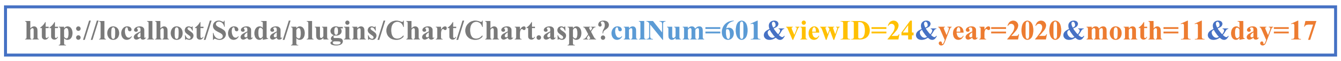

http://localhost/Scada/plugins/Chart/Chart.aspx?cnlNum=601&viewID=24&year=2020&month=11&day=17

FIGURE 1. HTTP address of the shed temperature data.

In the SCADA system, HTTP is divided into three parts: server IP address, view absolute path, and view channel ID. For example, the HTTP address for querying shed temperature data is shown in Figure 1.

Localhost, Scada, plugins, Chart, and Chart.aspx are variables that do not require attention in this paper. The cnlNum corresponds to the channel number of the collected data, and viewID corresponds to the view number, and year, month, and day dates. After determining these three attributes, you can retrieve the values of the variables under the view number and channel number of a certain date. Combined with the usage scenario of the SCADA system, an instruction of intermediate language specification which meets the requirements of the SCADA system is proposed. For example, in a smart agriculture measurement and control system, when input basic natural language query instructions, such as "请问现在大棚的温度是多少?"("What is the temperature of the greenhouse now?"), there are two pieces of information to be extracted: object, location, which are transformed into the intermediate language with a data structure such as { 'object': '温度'('temperature'), 'location': '大棚'('greenhouse')}, SCADA-NLI reads the index values corresponding to the object and location attributes in the above structure and automatically fills the HTTP address to call the HTML data monitoring interface to complete the basic query function. When inputting natural language control instructions, such as "帮我先打开大棚的传感器, 然后打开大田的空调。"("Help me turn on the sensor of the shed first, then turn on the air conditioner of the field."), there are three pieces of information to be extracted: object, location, and action, which are transformed into data structures such as [{'action': '打开'('turn on'), 'location': '大棚'('shed'), 'object': '传感器'('sensor')}, {'action': '打开'('turn on'), 'location': '大田'('field'), 'object': '空调'('air conditioner')}], SCADA-NLI reads the index values corresponding to the action, location and object attributes and sends them to the gateway, node or other field devices to complete complex control functions.

To realize the above process, this paper proposes a sketch of automatic key-value parameter padding, as shown in Figure 2, and describes this type of intermediate language format as a key-value data structure. In Figure2(a), this paper defines the beginning of '$' to indicate the location to be filled, and the variable name after '$' indicates that it is the parameter to be filled. The $viewID is populated with the index value corresponding to the keyword representing the location; $cnlNum is populated with the index value corresponding to the keyword that represents the object attribute; $actionParameters is filled with the index parameter corresponding to the keyword that represents the action property. The dependencies between the parameters are shown in Figure2(b). The keywords in the extracted natural language instructions are used as keys to query the corresponding values, thus implementing the instruction parameter mapping.

For the complex natural language query instructions in Example 4, it is not suitable to parse it into an intermediate language in the form of key-value. A new intermediate language format is needed, and this paper uses SQL statements as the intermediate language format after parsing complex natural language query instructions. The overall flow is shown in Figure 3, including the NL-to-SQL semantic parsing unit and the SQL statement execution unit, which aims to transform the input natural language instructions into SQL statements, and the SQL statement execution unit to complete the query on a given database. In this paper, we mainly study the NL-to-SQL semantic parsing unit and a single table query for a given data table, with SQL statements conforming to the form of Figure 4. Where the fields after the $ symbol indicate the ones to be populated, and * indicates zero or more.

III. SCADA-NLI EQUIVALENT ABSTRACT WORKFLOW

SCADA-NLI is a natural language query and control interface based on a hierarchical classification semantic parsing

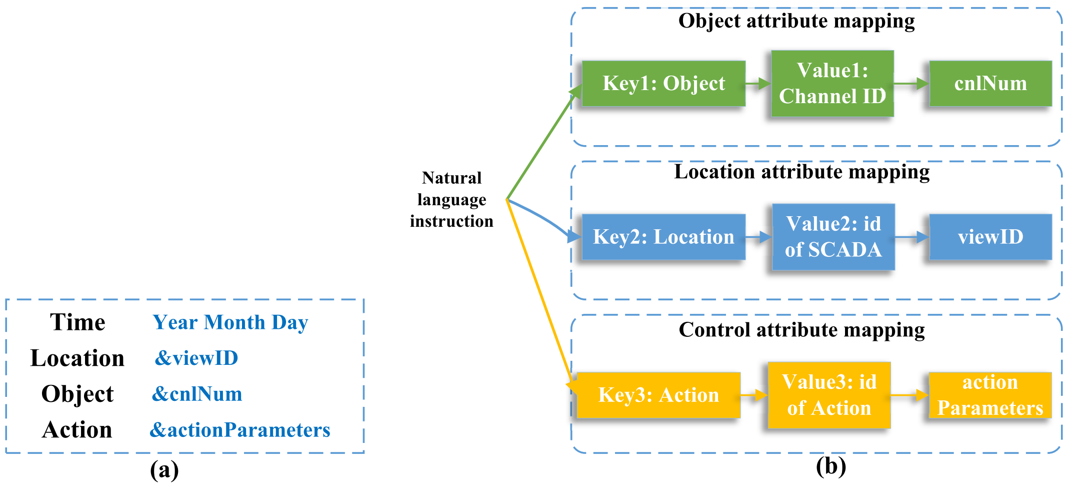


FIGURE 2. Sketch of key-value parameter filling, where (a) denotes the intermediate language data structures and their corresponding parameters, and (b) denotes the parameter dependencies.

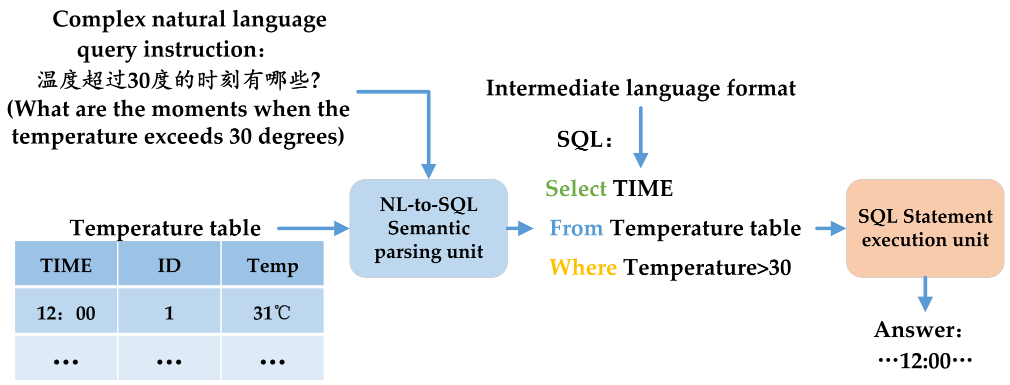

Complex natural language query instruction:
温度超过30度的时刻有哪些?
(What are the moments when the temperature exceeds 30 degrees)
Intermediate language format
SQL:
Temperature table
TIME ID Temp
12: 00 1 31℃
... ...
NL-to-SQL Semantic parsing unit
Select TIME
From Temperature table
Where Temperature>30
SQL Statement execution unit
Answer:
···12:00···

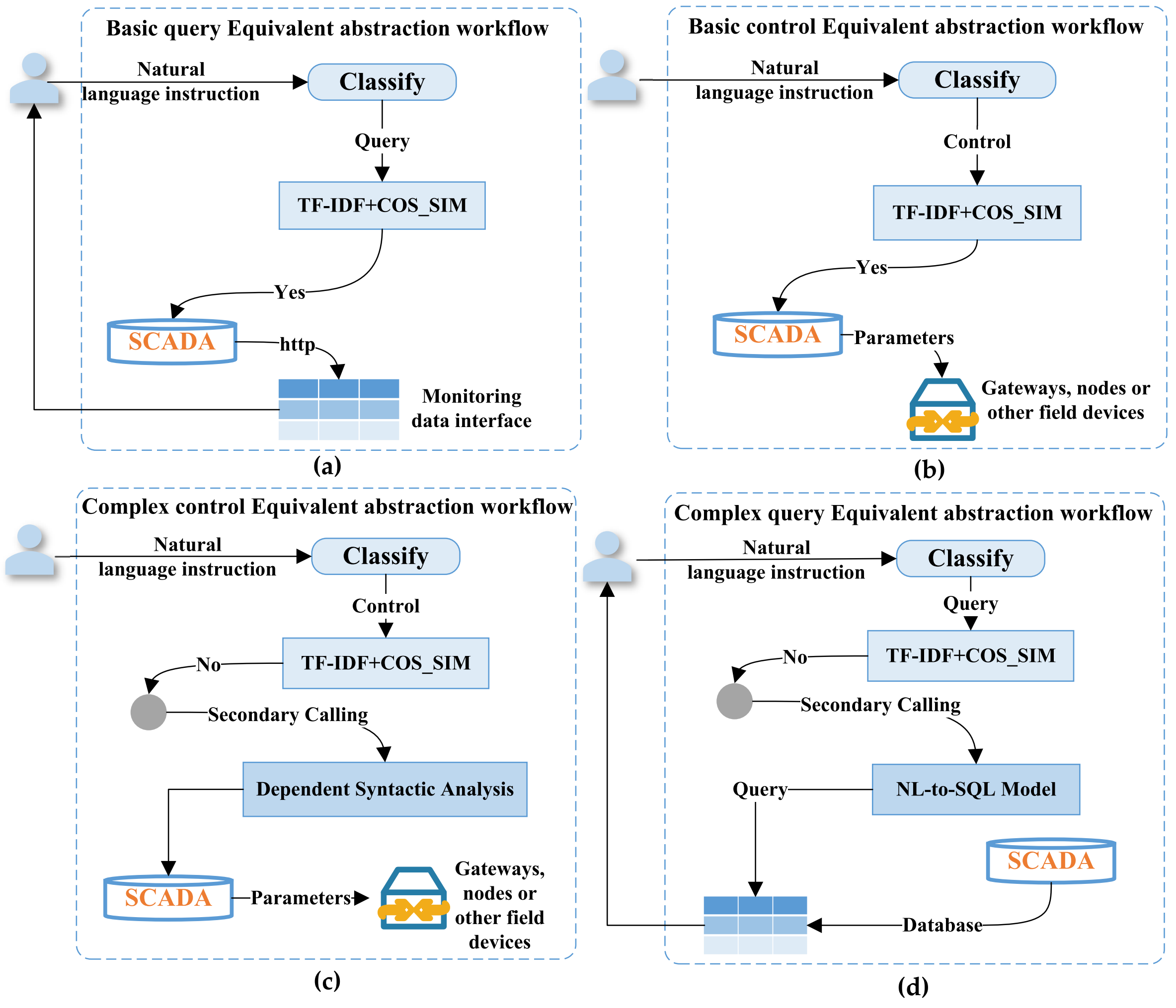
FIGURE 3. Schematic of NL-to-SQL process.

```
SELECT  $agg  $col
FROM T
WHERE $col $op $value(AND $col $op $value )*
```

FIGURE 4. Types of SQL statements studied in this paper.

algorithm. The equivalent abstract workflow of the system is given in this section, as shown in Figure 5.

- Figure 5(a) reflects the equivalent abstraction workflow of basic query instructions. Firstly, natural language instructions are input, and after classification, it is determined to be a query instruction type, however, at this time, the instruction cannot be fully identified as a basic natural language query instruction, and it needs to pass TF-IDF+COS_SIM semantic parsing algorithm, after successful parsing, it can automatically fill HTTP address and successfully call the data monitoring HTML interface to realize the basic data query function.
- Figure 5(b) reflects the equivalent abstract workflow of the basic control instruction. Like the basic query instruction, after the resolution is successful, the parameters are sent to the surrounding gateways, nodes, or other field equipment through the SCADA system to complete the basic control operation.
- Figure 5(c) reflects the equivalent abstraction workflow of complex control instructions, the first step is also through TF-IDF+COS_SIM semantic parsing algorithm, however, the algorithm does not satisfy the complex control instructions, so the parsing fails, at this time SCADA-NLI chooses to call the semantic parsing



Basic query Equivalent abstraction workflow
Natural language instruction
Classify
Query
TF-IDF+COS_SIM
Yes
SCADA
http
Monitoring data interface
(a)
Basic control Equivalent abstraction workflow
Natural language instruction
Classify
Control
TF-IDF+COS_SIM
Yes
SCADA
Parameters
Gateways, nodes or other field devices
(b)
Complex control Equivalent abstraction workflow
Natural language instruction
Classify
Control
TF-IDF+COS_SIM
No
Secondary Calling
Dependent Syntactic Analysis
SCADA
Parameters
Gateways, nodes or other field devices
(c)
Complex query Equivalent abstraction workflow
Natural language instruction
Classify
Query
TF-IDF+COS_SIM
No
Secondary Calling
Query
NL-to-SQL Model
SCADA
Database
(d)

FIGURE 5. Equivalent abstraction workflow of SCADA-NLI. (a) represents the workflow of basic query instructions, (b) represents the work-flow of basic control instructions, (c) represents the workflow of complex control instructions, (d) represents the workflow of complex query instructions.

algorithm based on dependent syntax analysis according to the classification results, and successfully resolves the complex natural language control instructions. And the mapped parameters are sent to the surrounding gateways, nodes, or other field devices through the SCADA system to complete the complex control operation.

- Figure 5(d) reflects the equivalent abstraction workflow of complex query instructions. As in Figure 5c, after the first failed parsing, the NL-to-SQL model will be called according to the classification results, and the complex natural language instructions will be converted into SQL statements to realize the complex query function by accessing the SCADA built in database.

IV. INSTRUCTION INTENT CLASSIFICATION METHOD

Instruction intent classification is the first stage of the SCADA-NLI architecture, Effective intention classification is helpful to reduce information redundancy, improve the retrieval efficiency of subsequent algorithms, and help to rationally select the invocation service. For the four instruction types proposed by the SCADA system requirements, we used the BiLSTM-Attention model for the instruction classification task, due to the lack of sufficient high-quality annotated datasets. The recognition effect is not good, and it takes a lot of time and cost to label a high-quality corpus, so in this paper, we propose an instruction intent classification method KWECS (Keywords extraction and cosine similarity) for measurement and control systems. An example is shown in Figure 6 to illustrate how KWECS works.

The user inputs the instruction "关闭大棚的喷雾器和传感器。" ("Turn off the sprayer and sensor of the shed."), and first goes through the keyword extraction (step 1), using the TF-IDF algorithm, The calculation formula is as follows:

$$TF_{W_i D_i} = \frac{count(w)}{|D_i|} \tag{1}$$

User input	关闭大棚的喷雾器和传感器。 Turn off the sprayers and sensors in the greenhouse.				
Step1:	Keyword extraction	喷雾器 nz (Sprayers nz)	大棚 n (greenhouses n)	传感器 n (sensors n)	关闭 v (Turn off v)
Step2:	Sieve out verbs	关闭 v (Turn off v)			
Step3:	Verb dictionary	Control_dict = ['关闭', '升高', '调低', '监测', '提高', '测量',… '控制'] Control_dict = [' closed ', 'rise', 'down', 'monitoring', 'increase', 'measurement',… 'control']			
Step4:	word embedding	关闭:[2.023750e-01, -3.708000e-03, -7.565430e-01, …-3.586300e-01]			
Step5:	Cosine similarity	$cos < x_i, y_i >= \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2 \times \sum_{i=1}^{n} y_i^2}} = 0.73 > 0.65$			

FIGURE 6. **An example of KWECS.**

where *count(w)* denotes the number of keyword w occurrences and $|D_i|$ is the number of all words in a document D_i; then the keywords of verb lexicality are extracted (step 2); then we construct a good list of common control verbs in the measurement and control field (step 3) for the subsequent cosine similarity calculation; load the trained Word2vec model and perform word embedding operation to transform one keyword into a low-dimensional dense word vector (step 4); finally come to calculate the verb similarity between the verb in the instruction and the verb in the verb word list in turn, where we set the threshold value as 0.65, and once the cosine value exceeds this threshold, the instruction can be considered as a control instruction. For word vectors x_i and y_i, the cosine similarity is calculated as:

$$cos < x_i, y_i >= \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n} x_i^2 \times \sum_{i=1}^{n} y_i^2}} \tag{2}$$

where the cosine similarity takes the value interval [0,1]. The closer the cosine value of the angle between two-word vectors is to 1, the closer the semantics of these two words are.

V. NATURAL LANGUAGE INSTRUCTION PARSING

In this section, details of parsing natural language instructions are presented. section 5.1 and section 5.2 introduce the semantic analysis algorithm based on TF-IDF+COS_SIM and the semantic analysis algorithm based on dependency syntax, respectively, and explain how these two algorithms embed key-value parameters to automatically populate sketches. section 5.3 introduces the NL-to-SQL model based on BERT, describes the specific details of translating natural language problems into SQL statements.

A. TF-IDF+COS_SIM BASED SEMANTIC PARSING ALGORITHM

1) SEMANTIC PARSING FLOW OF TF-IDF+COS_SIM

The overall processing flow of TF-IDF+COS_SIM based semantic parsing algorithm is shown in Figure 7. If the user inputs a basic natural language query instruction, such as "请帮我查看一下卧室的温度是多少?" ("Please help me check what is the temperature of my bedroom?"), it needs to be transformed into the intermediate language {'object': '温度'('temperature'), 'location': '卧室'('bedroom')}. Then fill the sketch with the key-value parameter, complete the parameter mapping and fill it into the parameter list, as shown in the above-structured instruction can be converted to [5,7], as shown in the blue dashed line. And fill the parameters cnlNum and viewID into the HTTP address of the SCADA system, and realize real-time data query by calling the HTML monitoring data interface of address link. If it is a basic natural language control instruction, such as "天气炎热, 请帮我打开客厅的空调。"("It is hot, please help me turn on the air conditioner in the living room."), it also needs to be converted to an intermediate language {'object': '空调'('air conditioner'), 'location': '客厅'('living room'), 'action': '打开'('turn on')}, which has one more control action compared to the query instruction. Still filling the sketch by key-value parameters, mapping the structured intermediate language to parameters, and filling the parameters to the list, the above instruction can be converted to [8,9,1], this process is shown as a yellow dashed line. The parameters are then sent to the corresponding gateways, nodes, or other field devices through the SCADA system to complete the real-time control operation.

2) DETAILS OF THE SEMANTIC PARSING ALGORITHM FOR TF-IDF+COS_SIM

The specific steps of the semantic parsing algorithm of TF-IDF+COS_SIM are shown in Figure 8.

The user entered "太热了,帮我开启客厅的空调。"("It's too hot, help me open the air conditioner in the living room.").Firstly, it is to perform Chinese word segmentation and stop word filtering (Step1). The part-of-speech tagging (Step2) is then performed to improve the keyword extraction accuracy (Step3) of the TF-IDF algorithm. Then, the word vectorization is performed, and each keyword is transformed into a 300-dimensional word vector (Step4). After the

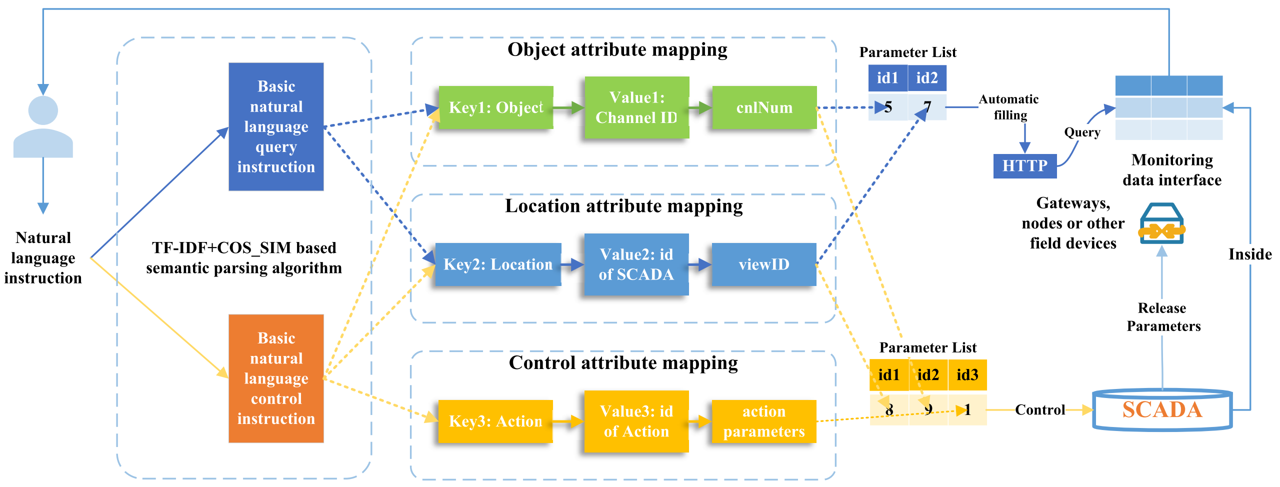

Natural language instruction
TF-IDF+COS_SIM based semantic parsing algorithm
Basic natural language query instruction
Basic natural language control instruction
Object attribute mapping
Key1: Object
Value1: Channel ID
cnlNum
Location attribute mapping
Key2: Location
Value2: id of SCADA
viewID
Control attribute mapping
Key3: Action
Value3: id of Action
action parameters
Parameter List
id1 id2
5 7
Automatic filling
HTTP
Query
Monitoring data interface
Gateways, nodes or other field devices
Inside
Release Parameters
Parameter List
id1 id2 id3
8 9 1
Control
SCADA

FIGURE 7. The flow of semantic parsing algorithm based on TF-IDF+COS_SIM.

	User input	太热了，帮我开启客厅的空调。 (It's too hot. Help me open the air conditioner in the living room.)
Step 1	Words Segmentation	太热(too hot) /开启(open) /客厅(living room) /空调(air conditioner)
Step 2	POS Tagging	太热(too hot)，a/开启(open)，v/客厅(living room)，np/空调(air conditioner)，ns
Step 3	TF-IDF algorithm	开启(open) /客厅(living room) /空调(air conditioner)
Step 4	Word Embedding	开启:[1.123740e-01, -2.345620e-03, -5.566730e-01, …-3.486352e-01]
Step 5	The first match	{‘object’: ’空调’(’ air conditioner’),’location’: ’客厅’(’ living room’), ’action’: ’开启’ (’open’)}
Step 6	Cosine similarity	$cos<x_i,y_i>=\frac{\sum_{i=1}^{n}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{n}x_i^2\times\sum_{i=1}^{n}y_i^2}}=0.85>0.65$
Step 7	The second match	{‘object’: ’空调’(’ air conditioner’),’location’: ’客厅’(’ living room’), ’action’: ’打开’ (’turn on’)}

FIGURE 8. An example of the TF-IDF+COS_SIM algorithm.

conversion is completed, the first match is made, and the extracted keywords are filled into the key-value data structure (Step5). If the filling fails, the second matching (Step6) shall be carried out. Before this, the word similarity shall be calculated. Take the word “开启”(“open”) as an example. If the first matching fails, the keyword with the highest similarity shall be calculated as “打开”(“turn on”). Therefore, the second matching shall be carried out for “打开”(“turn on”).Finally, it is converted into a key-value intermediate language format for reading by the SCADA system.

The pseudo-code implementation of the TF-IDF+COS_SIM semantic parsing algorithm proposed in this paper is shown in Figure 9 (with basic natural language control instructions as an example).

B. SEMANTIC PARSING ALGORITHM BASED ON DEPENDENCY SYNTACTIC ANALYSIS

1) THE SEMANTIC PARSING PROCESS OF DEPENDENT SYNTACTIC ANALYSIS

In this section, we use the Baidu Chinese dependent parsing tool DDParser to analyze the dependent syntax of complex natural language control instructions. By analyzing the dependence between word prototypes in complex control instructions, the problem of semantic understanding of complex control instructions is solved. The overall flow of the algorithm is shown in Figure 10.Firstly, the dependency of natural language instruction is obtained through dependency syntactic analysis. Each node contains four parts, which are word prototype, word character, dependency, and position dependency between words. Then, the dependent syntax tree of the natural language instruction is obtained by using the dependent syntax analysis tool, and the sketch is filled with the key-value parameter to complete the transformation from unstructured natural language to a structured intermediate language. After step 2, the generated structured intermediate language is optimized, and the optimized intermediate language is mapped to corresponding parameters, which are stored in the form of a list. Since complex control instructions usually contain two or more actions, the mapped parameters are usually two or more groups. Finally, it is sent to the SCADA system in turn and then to the corresponding gateway, node, or other field equipment by the SCADA system to complete the real-time control action.

2) DEPENDENT SYNTACTIC ANALYSIS PROCESS

The complex natural language control instruction “先打开大棚的空调 然后关闭田间的灯光。”(“first turn on the air conditioner in the shed and then turn off the lights in the shed.”) in Example 4 generates dependencies

Algorithm 1 Semantic parsing algorithm for TF-IDF + COS_SIM

Input: basic natural language instruction

$StringList = [[Word_1, WordCategory_1], ..., [Word_i, WordCategory_i]]$, $Word_i$ for keywords, $WordCategory_i$ Represents the part of speech of the key word

Output: Structured set of key-value pairs $[\{'key_{action}' :' word_1'\}, \{'key_{location}' :' word_2'\}, \{'key_{object}' :' word_3'\}]$, where the keys represent the intermediate language data structure paradigm defined and the values represent the key metrics for extraction.

```
1: $RESULT = \{\phi\}$ /* Initialize an empty collection */
2: $StringList = TF - IDF\_KeywordExtract(String,'TF')$ /* TF-IDF keyword extraction */
3: $Instruction\_Type = classify(String)$ /* Output 0 is query instruction and output 1 is control instruction */
4: $IFInstruction\_Type == 1$
5: for $List\ INStringList$ /* Traverse the extracted keyword list */ do
6:     if $List[0]\ INlocation\_dict$ or $List[1] ==' ns'$ /* A keyword is considered a location attribute if it is in an established location dictionary or if the part of speech is' ns' */ then
7:         $key_{location} \leftarrow List[0]$
8:     else $COS\_SIM(List[0], Word_{location\_dict}) > =0.65$ /* Keywords with similarity above 0.65 can be considered synonyms */
9:         $key_{location} \leftarrow List[0]$
10:    end if
11:    if $List[0]\ INobject\_dict$ or$List[1] ==' nz'$or$'nc'$ /* If the keyword is in the established object dictionary or the part of the word is' nz' or' nc', it is considered as the object attribute */ then
12:        $key_{location} \leftarrow List[0]$
13:    else $COS\_SIM(List[0], Word_{location\_dict}) > =0.65$
14:        $key_{location} \leftarrow List[0]$
15:    end if
16:    if $List[0]\ INobject\_dict$or$List[1] ==' v'$ then
17:        $key_{location} \leftarrow List[0]$
18:    else $COS\_SIM(List[0], Word_{location\_dict}) > =0.65$
19:        $key_{location} \leftarrow List[0]$
20:    end if
21:    $RESULT \leftarrow [\{'key_{action}' :' word_1'\}, \{'key_{location}' :' word_2'\}, \{'key_{object}' :' word_3'\}]$
22: end for
```

FIGURE 9. **Algorithm 1 Semantic parsing algorithm for TF-IDF + COS_SIM.**

like the list [{'word': ['先'('first'), '打开'('turn on'), '大棚'('shed'), '的'('in'), '空 调'('air conditioner'), '然 后'('then'), '关闭'('turn off'), '田 间'('shed'), '的'('in'), '灯 光'('lights')], 'postag': ['d', 'v', 'n', 'u', 'n', 'c', 'v', 'n', 'u', 'n'], 'head': [2, 0, 5, 3, 2, 7, 2, 10, 8, 7], 'deprel': ['ADV', ' HED', 'ATT', 'MT', 'VOB', 'ADV', 'VV', 'ATT', 'MT', 'VOB']}] are shown. Among them, 'word' indicates the result of the natural language instruction after word division; 'postag' indicates the lexicality of the words; 'head' indicates the position relationship between the semantic dependencies of the words; 'deprel' indicates the semantic relationship; after summarizing the complex natural language control instructions, it is found that there are four main types of semantic dependencies commonly found in the instructions: the core relationship (HED), definite middle (ATT), verb-object (VOB), and concurrent (COO). Table 2 gives examples of common dependencies in natural language.

The dependency tree of the instruction is shown in Figure 11. In natural language control instructions, usually, the verb is the core of the whole sentence and does not depend on any words. As shown in Figure 11(a),the number under '打 开'('turn on') is 0, i.e., it does not form a dependency relationship with any word and is defined as the ROOT node. Meanwhile, the definite relationship (ATT) describes the relationship between the definite word and the core word. In this example, the number corresponding under '大棚'('shed') is 5, i.e., it forms a definite relationship (ATT) with the 5th word '空调'('air conditioner'), considering '大棚'('shed')is the definite article of '空调'('air conditioner'), and similarly, the number under '田 间'('shed') is 10, which is the definite relationship (ATT) with the tenth word '灯光'('lights') constitutes the definite middle relationship (ATT). Based on this property, the word prototype corresponding to the ATT can be selected as the location attribute in the intermediate language data structure pattern. Meanwhile, in the example, the number under '空 调'('air conditioner') is 2, i.e., it forms a verb-object relationship (VOB) with the second word '打开'('turn on'), and the number under '灯光'('lights') is 7, i.e. The number under '灯 光' ('lights')is 7, which is the verb-object relationship (VOB) with the 7th word '关 闭'('turn off'). Based on this property, the word prototype corresponding to the verb-object relation (VOB) can be selected as the object (Object) property in the intermediate language data structure paradigm. However, the natural language control instruction "打开大棚的传感器和灯光。"("Turn on the sensors and lights in the shed.")'传 感 器'('sensors') will constitute a juxtaposition relationship (COO) between '传感器'('sensors') and '灯光'('lights'), i.e., between words of the same type The dependency relationship is shown in the Figure 11(b), so in this case, the word prototypes corresponding to the COO relationship can be extracted and merged into the Object attribute. And for the action (Action) attribute, we can consider the verb-object relationship (VOB), after finding the object, that is, the object attribute, according to the dependency relationship between the verb and the object, extracts the keywords that the object depends on as the action attribute, for example, '关闭灯光'('Turn off the lights'), '关 闭'('turn off') and '灯 光'('lights') form a verb-object relationship, so the action attribute can be determined as '关 闭'('turn off'). So after the above steps, we will get the paradigm of the intermediate language data structure of the complex control language "先打开大棚的空调然后关闭田 间的灯光。"("first turn on the air conditioner in the shed and then turn off the lights in the shed."): that is, [{'action': '打 开'('turn on'), 'location': '大棚'('shed'), 'object':'空 调'('air conditioner')}, {'action':关 闭'('turn off'), 'location': '田 间'('shed'), 'object': 灯光'('lights')}]. Finally, the key-value semantic instruction mapping algorithm is called to complete the parameter transformation.

3) KEY INDICATOR EXTRACTION

The key indicators in the control instruction of complex natural language are extracted through dependency syntactic analysis and part-of-speech features, and filled into the intermediate language data structure defined in Section 2.Key indicator extraction steps are as follows:

- Generating a dependency syntax tree of complex natural language control instructions.

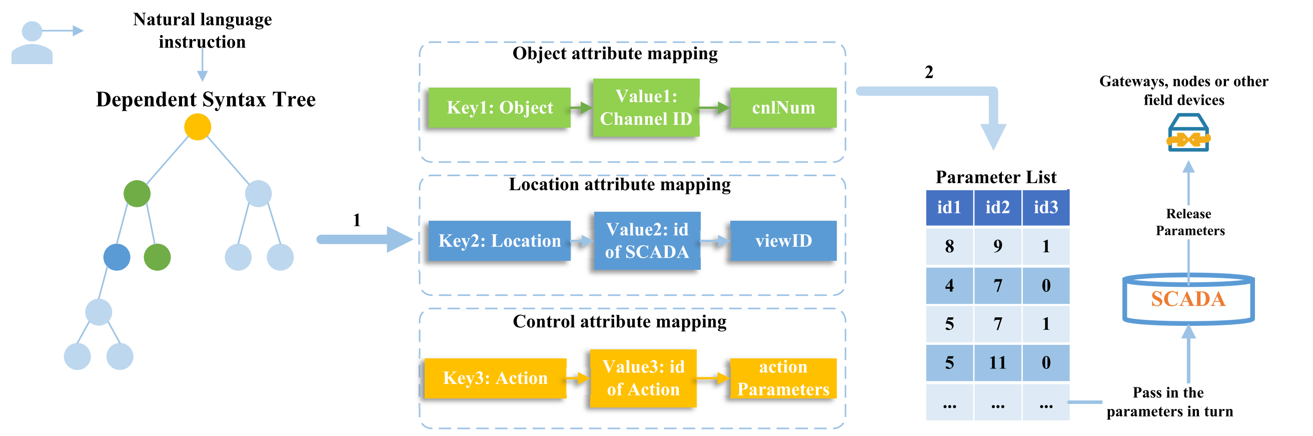


FIGURE 10. The flow of semantic parsing algorithm based on dependency syntactic analysis.

TABLE 2. Dependency relation tags in DuCTB.

Relation	Description	Example
HED	指整个句子的核心 Sentence head and pseudoword	打开卧室的台灯（ROOT,打开，HED） Open the table lamp in the bedroom（ROOT,Open，HED）
ATT	定语与中心词之间的关系 attribute and headword	关闭大棚的传感器（传感器，大棚，ATT） Close the sensor of the shed（sensor，shed,ATT）
VOB	宾语和谓语之间的关系 object and predict	请帮我打开厨房的油烟机（打开，油烟机，VOB） Please help me open the kitchen cooker hood（open，cooker hood，VOB）
COO	同类型词语之间的关系 two coordinate words	打开卧室的空调和台灯（空调，台灯，COO） Open the air conditioner and desk lamp in the bedroom（conditioner，desk lamp，COO）

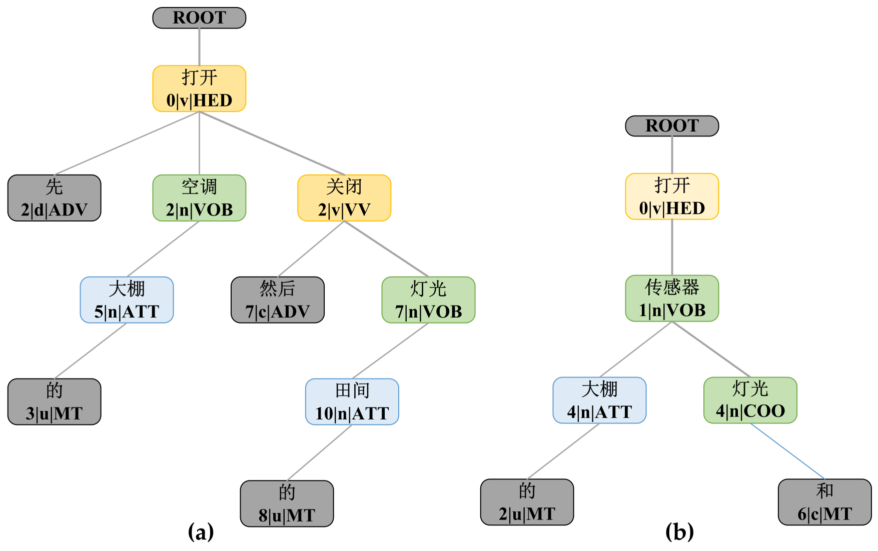
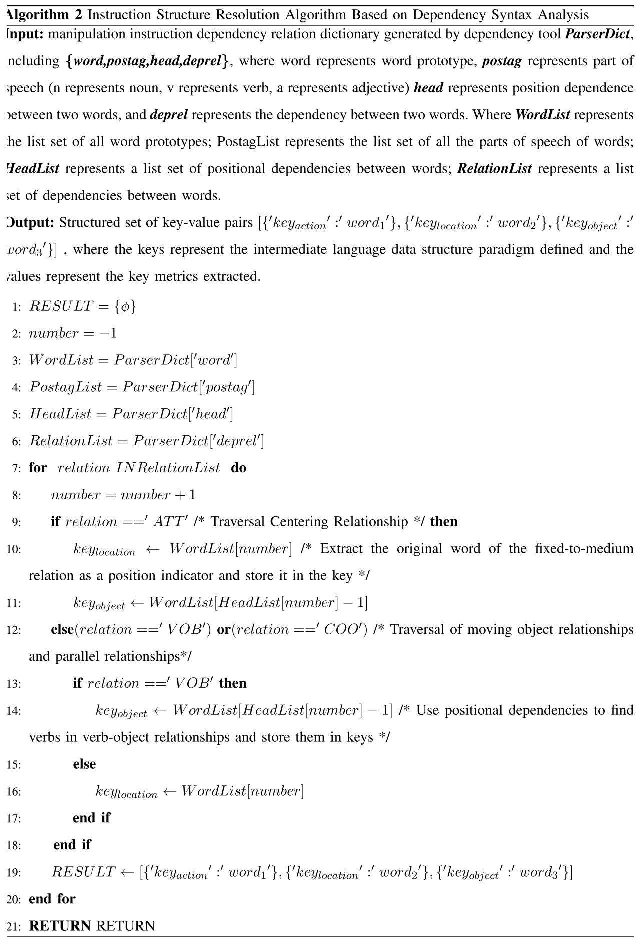


FIGURE 11. Dependency tree.

- After searching dependence analysis results, extracting key indexes according to semantic relationship and part-of-speech characteristics.
- Automatically fill it into the pre-defined intermediate language data structure form.

In this paper, when using semantic features for key indicator extraction, there are three main semantic relations that we are looking for: definite in relation (ATT), verb-object relation (VOB), and concurrent relation (COO). The following rules need to be followed when performing key indicator extraction.

- Rule 1: When there are multiple verbs in the instruction, the verbs will have core relationship (HED) and continuous predicate relationship (VV), which is not conducive



lgorithm 2 Instruction Structure Resolution Algorithm Based on Dependency Syntax Analysis

nput: manipulation instruction dependency relation dictionary generated by dependency tool ***ParserDict***, ncluding {***word,postag,head,deprel***}, where word represents word prototype, ***postag*** represents part of speech (n represents noun, v represents verb, a represents adjective) ***head*** represents position dependence between two words, and ***deprel*** represents the dependency between two words. Where ***WordList*** represents he list set of all word prototypes; PostagList represents the list set of all the parts of speech of words; ***HeadList*** represents a list set of positional dependencies between words; ***RelationList*** represents a list set of dependencies between words.

Output: Structured set of key-value pairs $[\{'key_{action}':'word_1'\},\{'key_{location}':'word_2'\},\{'key_{object}':'word_3'\}]$, where the keys represent the intermediate language data structure paradigm defined and the values represent the key metrics extracted.

```
1:  $RESULT = \{\phi\}$
2:  $number = -1$
3:  $WordList = ParserDict['word']$
4:  $PostagList = ParserDict['postag']$
5:  $HeadList = ParserDict['head']$
6:  $RelationList = ParserDict['deprel']$
7:  for $relation\ IN RelationList$ do
8:      $number = number + 1$
9:      if $relation ==' ATT'$ /* Traversal Centering Relationship */ then
10:         $key_{location} \leftarrow WordList[number]$ /* Extract the original word of the fixed-to-medium relation as a position indicator and store it in the key */
11:         $key_{object} \leftarrow WordList[HeadList[number] - 1]$
12:     else($relation ==' VOB'$) or($relation ==' COO'$) /* Traversal of moving object relationships and parallel relationships*/
13:         if $relation ==' VOB'$ then
14:             $key_{object} \leftarrow WordList[HeadList[number] - 1]$ /* Use positional dependencies to find verbs in verb-object relationships and store them in keys */
15:         else
16:             $key_{location} \leftarrow WordList[number]$
17:         end if
18:     end if
19:     $RESULT \leftarrow [\{'key_{action}':'word_1'\},\{'key_{location}':'word_2'\},\{'key_{object}':'word_3'\}]$
20: end for
21: RETURN RETURN
```

FIGURE 12. Algorithm 2 instruction structure resolution algorithm based on dependency syntax analysis.

to direct extraction, so we first look for the verb-object relationship (VOB) and find the corresponding verb with the help of object according to the language habits and the part-of-speech characteristics.

- Rule 2: Attributive is mainly a modifier. This paper takes into account the semantic parsing of manipulation instructions, and usually the position information that represents the fixed-medium relation (ATT). It can therefore be extracted as a position attribute indicator.
- Rule 3: Co-ordinate relationship (COO) plays the same role in the manipulation instruction, i.e., the action of the control object is the same, and for this kind of relationship so that its control action and location information are consistent, and the word prototypes corresponding to the verb-object relationship (VOB) can be merged.

Based on the above rules, the pseudo-code of the semantic parsing algorithm based on dependent syntactic analysis proposed in this paper is shown in Figure 12.

4) INSTRUCTION PARSING RESULT OPTIMIZATION STRATEGY

This section proposes a set of instruction parsing result optimization algorithms whose main function is to further optimize the instruction structuring results obtained by Algorithm 2. By analyzing the noisy data that often appear in complex controlled natural languages for dependency syntactic analysis, the failed extraction results are automatically eliminated. The Chinese language itself is very complex, and the dependencies formed by the same word in different contexts may be different. For example, "请帮我打开卧室的台灯。"("Please help me turn on the lamp in my bedroom.") and "打开卧室的台灯。"("Turn on the desk lamp in the bedroom."), which are two natural language instructions, express the same meaning, but The dependencies expressed by the keyword '打开'('turn on') are different. In the first instruction, '打开'('turn on') and '帮'('help') form a double object relationship (DOB). In the second instruction '打开'('turn on') and 'ROOT' form a correlation (HED). Similarly, the word prototypes corresponding to the same dependency in the same natural language instruction may not be consistent, for example, '帮'('help') and '台灯'('lamp') in "请帮我分别打开卧室的台灯和客厅的空调。"("Please help me turn on the lamp in the bedroom and the air conditioner in the living room respectively."), both form verb-object relations with other words (VOB). In addition, humans will mix some other words in their speech, for example, "打开卧室的扫地机器人,卫生需要保持干净"("Turn on the bedroom sweeper, hygiene needs to be kept clean"), and the result after Algorithm 2 is: [{'action': '打开'('turn on'), 'location': '卧室'('bedroom'), 'object': '扫地机器人'('sweeper')}, {'action': '打开'('turn on'), 'location': '卧室'('bedroom'), 'object': '保持'('kept')}]. It can be seen that the second dictionary is redundant and mistakes '保持'('kept') for the 'object' property. The main reason is that '保持'('kept') and '需要'('needs') form a verb-object relationship (VOB), and '扫地机器人'('sweeper') and '打开'('turn on') also form a verb-object relationship (VOB). also form a verb-object relationship (VOB). Algorithm 2 will mistakenly identify '保持'('kept') as an object attribute and complete the indicator extraction. Therefore, in the optimization algorithm module, it is necessary to automatically delete the redundant invalid information in the structured results and eliminate the error extraction information by combining the part-of-speech features. Therefore, it can improve the accuracy of instruction parsing and enhance the scientificity, applicability, and expansibility of instruction structure parsing algorithms based on dependent syntactic parsing. The optimization algorithm is shown in Figure 13.

C. NL-TO-SQL MODEL

1) DATA SET AND PROBLEM ANALYSIS

NL2SQL data sets in Chinese are scarce. The training data sets and test data sets used in this paper are from the first Chinese NL2SQL Challenge Competition. The scale is shown in table.3. A total of 41522 natural language problems and corresponding formal SQL statements are included in the training set, and 4086 natural language problems are included

Algorithm 3 Instruction parsing result optimization algorithm

```
Input: $RESULT.$
Output: optimized RESULT
1: $i = 0$
2: $object\_value{=}\phi$
3: if $key_{action}! = \phi$ and $key_{location}! = \phi$ and $key_{object}! = \phi$ /* Any dictionary with an empty index is
   directly removed */ then
4:     for $item\ IN\ RESULT$ do
5:         $object\_value \leftarrow item(key_{value})$
6:     end for
7:     for $string1\ IN\ Wordlist$ /* Traverses the original word, passing the value to */ do
8:         $i = i + 1$
9:         for $string1\ IN\ object\_value$ /* Traverses the list of extracted object properties, passing
   values to string2*/ do
10:            if $string2 == string1$ /* Determine equality and update value */ then
11:                if $(PostagList[i-1]! =' nz')\ and (PostagList[i-1]! =' n')$ /* POS judgment */ then
12:                    $delete(dict \leftarrow key_{object})$ /* The part of an object's property Delete the dictionary where
   the object is located if it is not nz or n*/
13:                end if
14:            end if
15:        end for
16:        $RESULT \leftarrow RESULT$
17:        RETURN RETURN
18:    end for
```

FIGURE 13. Algorithm 3 resulting optimization algorithm for the instruction structure parsing algorithm based on dependent syntactic analysis.

TABLE 3. Data examples.

	Number of samples	Number of data tables
Training set	41522	11199
Test set	4086	1102

in the test set. A sample of data was selected, as shown in Figure 14.

For the sel field, it represents the column selected by SQL, belonging to the multi-label classification model, but since the data table and its meaning of each sample are different, the categories here may change at any time, so it is necessary to dynamically encode each category vector according to the column name of the table. Agg represents the aggregate function corresponding to the selected column, corresponding to sel, and has 6 categories. Cond_conn_op is a single label 4 classification problem. The last field, conds, needs to be classified in combination with the field label, and it needs to determine which column is a condition, the operational relationship of the condition, and the corresponding value of the condition. Details of the model are given in section 5.3.2.

2) MODEL INTRODUCTION

Firstly, marking the input natural language problem and header name. Considering that the column meanings of each table are different, with the help of the characteristics of BERT, We connect the problem sentence with the header of the data table and input it into the BERT model for real-time coding. The coding is as follows:

$$[CLS], w_1, w_2, \ldots, w_n, [SEP] \tag{3}$$

As a result, global attention will be paid to downstream tasks, and these vectors need to be used to predict the number of fields corresponding to different slots. Next, you will see the following information:

1) Represents a global classifier: Used to determine the join between conditions. Where the condition is conn_sql_dict = 0:"",1:" and ",2:" or ", so we define it as a 3 classifier. The corresponding calculation formula is as follows:

$$x = bert\_model([x_1\_in, x_2\_in]) \tag{4}$$
$$x_{conn} = f(x) \tag{5}$$
$$p_{conn} = softmax(x_{conn}) \tag{6}$$

where f means that the vector of questions corresponding to ***[CLS]*** is taken out for classification, and ***num_cond_cpnn_op*** is 3, which means 3 is classified.

2) 7 tags sequence annotation: The main purpose is to mark which columns will be selected and predict aggregation relationships, aggregator dictionary agg_sql_dict = 0:"", 1: "AVG", 2: "MAX", 3: "MIN", 4: "COUNT", 5: "SUM". 6 categories, but we choose to add a category so that if the first 6 categories are selected, it means that this category is selected and also predicted agg. In this way, if the first 6 categories are selected, it means that this category is selected and the agg is predicted, at the same time, and if the 7th category is selected, it means that this category is not selected. The calculation formula is as follows.

$$x_{sel} = f\,(x, h) \tag{7}$$
$$p_{sel} = softmax(num_{agg})(x_{sel}) \tag{8}$$

3) 5 tags sequence annotation: used to mark values and operators as conditions, the operators are op_sql_dict = 0:">", 1:"<", 2:"==", 3:"! =". For the field conds, i.e. WHERE, it is necessary to find the conditional column, the conditional type, and the conditional value. Therefore, for the prediction of conds, we divide it into two steps, the first step predicts the conditional value and the second step predicts the conditional column. Prediction of the conditional value is essentially a sequence labeling problem, where the conditional value corresponds to four operators, consistent with the above approach, adding a new category, that is, converted into a 5 classification problem, the fifth category represents the current field is not labeled, otherwise it can be labeled, so that the conditional value and operators can be predicted. The calculation formula is as follows.

$$p_{op} = softmax(num_{op})(x) \tag{9}$$

num_op is 5, meaning 5 classifications. See Step 4 for the prediction condition column.

4) Combine the columns to do classification, and determine which column the current condition belongs to: predict the condition column corresponding to the condition value, the word vector of the labeled condition value, and each table header for similarity calculation,

```
{     "table_id": "a1b2c3d4", # Corresponding table id
      "question": "世茂茂悦府新盘容积率大于 1, 请问它的套均面积是多少? ", # Natural Language
Question
      "sql":{
            "sel": [7], # Columns selected by SQL
            "agg": [0], # The corresponding aggregation function for the selected column, '0' means no
            "cond_conn_op": 0, # Relationship between conditions
            "conds": [
                  [1, 2, "世茂茂悦府"], # Conditional column, Conditional type, Conditional value,col_1 ==
"世茂茂悦府"
                  [6, 0, "1"]]}}
```

FIGURE 14. An example of training data set.

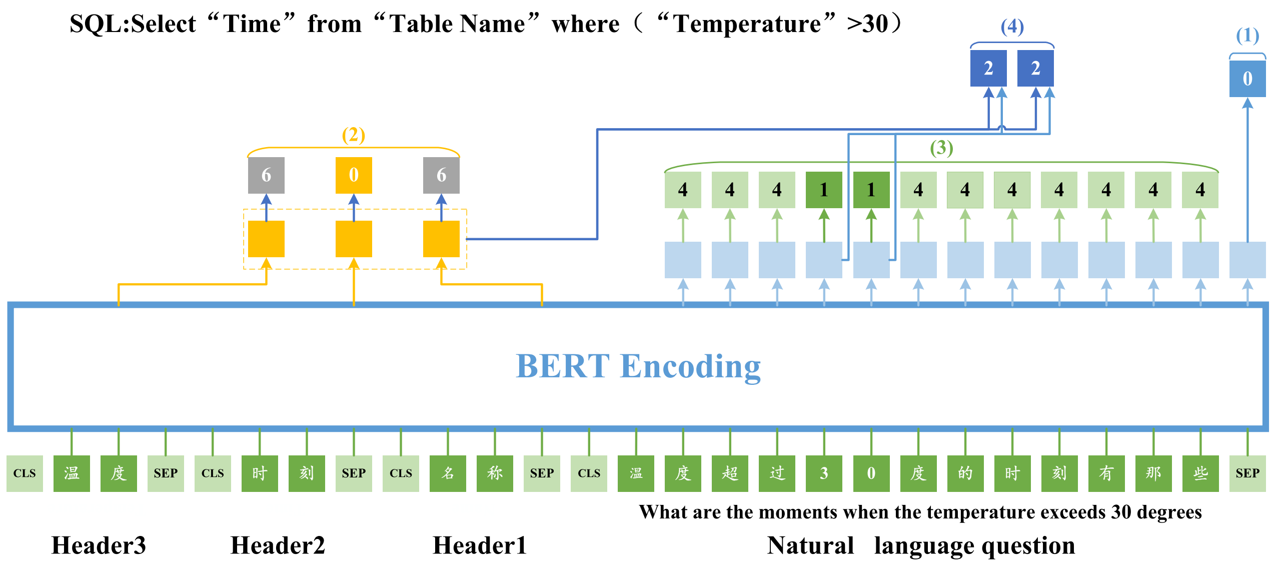

SQL:Select "Time" from "Table Name" where ("Temperature" >30)
(4)
2
2
(1)
0
(2)
6
0
6
(3)
4 4 4 1 1 4 4 4 4 4 4 4
BERT Encoding
CLS 温 度 SEP CLS 时 刻 SEP CLS 名 称 SEP CLS 温 度 超 过 3 0 度 的 时 刻 有 那 些 SEP
What are the moments when the temperature exceeds 30 degrees
Header3
Header2
Header1
Natural language question

FIGURE 15. NL-to-SQL model structure.

here the similarity is calculated by directly concatenating the word vector and the table header vector, and then connect a Dense(1) through the fully connected layer. The calculation formula is as follows:

$$p_{csel} = tanh(f([p_{csel}, p_{csel\_2}])) \tag{10}$$

$$p_{csel} = softmax(p_{csel}) \tag{11}$$

VI. EXPERIMENTS

A. EXPERIMENT PREPARATION

The performance evaluation metrics of the KWECS method are first introduced in Section 6.1.1; then the process of constructing the Chinese natural language query and control instruction test dataset is introduced in Section 6.1.2, and finally, the relevant metrics for evaluating the performance of SCADA-NLI are provided in Section 6.1.3.

1) KWECS EVALUATION METRICS

KWECS method is used to classify the intent of natural language instructions from the point of view of rules and statistics, considering the similarity between key verbs and semantics. An evaluation index, i.e. the percentage of instructions intended to identify success in all tests, is given as follows:

$$ACC = \frac{Correct\ number\ of\ instruction\ intent\ recognition}{Total\ number\ of\ test\ instructions} \tag{12}$$

2) CHINESEQCI-TS DATASET

SCADA-NLI is a natural language interface for specific fields. For the application field proposed in this paper, we build a Chinese query and control instruction test data set: ChinsesQCI -TS. Corpus collection is not a direct integration of dialogue text, but a pre-defined variety of instructions that may appear in the SCADA system, and then spread out into natural and colloquial sentences in the form of staff crowdsourcing. This method can effectively improve the performance of the system. First, we define an instruction {'object': '空 调'('air conditioner'),'location':

Intermediate language		Natural language
{'object'：'空调','location'：'客厅'，'action'：'打开'} {'object'：' air conditioner',' location'：' living room'，' action'：' turn on'}	statement①	帮我打开客厅空调。 Help me turn on the air conditioner in the living room.
	statement②	天气太热，打开客厅的空调。 It's too hot, turn on the air conditioning in the living room.
	statement③	先帮我打开客厅的空调。 First, help me turn on the air conditioner in the living room.

FIGURE 16. Example of corpus.

'客 厅'('living room'),'action': '打 开'('turn on')} that complies with the intermediate language specification mentioned in the SCADA system requirements. The pseudo-language problem description is then rewritten as the natural language problem E ="打开客厅空调。"("Turn on the living room air conditioner.") and statement E is expanded to a more complex and colloquial natural language statement. A case in the corpus is shown in Figure 16.

The final ChineseQCI-TS dataset provides a collection of 600 natural language instructions from different do-mains, such as the agricultural domain (200), the industrial domain (200), and the smart home domain (200). Each domain contains basic natural language query and controls instructions as well as complex natural language query and control instructions. Experimental tests were conducted using various types of data as shown in Figure 17.

3) SCADA-NLI PERFORMANCE EVALUATION METRICS

The evaluation methods of SCADA-NLI performance include four main ones: Accuracy, Precision, Recall, and F-score values, which are obtained by the manual determination in this paper.

The accuracy rate indicates the proportion of the number of instructions executed correctly out of the number of all instructions executed, and in this paper, **N** is used to denote the number of instructions executed correctly and **T** denotes the total number of instructions. As shown in Equation 13.

$$Accuracy = \frac{N}{T} \tag{13}$$

The precision rate indicates the proportion of correctly expressed indicator words and corresponding indicator values in the structured intermediate language results obtained by the corresponding algorithm or model. The formula is calculated as follows, using **S** to denote the number of correctly expressed indicator words and **P** to denote the number of all structured intermediate languages, as shown in Equation 14.

$$Precision = \frac{S}{P} \tag{14}$$

The recall rate indicates the number of correctly expressed indicator words and corresponding indicator values among the structured intermediate language results obtained by the corresponding algorithm or model for all the indicator values contained in the original instruction. **R** is used to denote the number of all indicator values contained in the original instruction. This is shown in Equation 15.

$$Recall = \frac{S}{R} \tag{15}$$

F-score is the summed average of precision rate and recall rate, which can reflect the good or bad performance of SCADA-NLI comprehensively, This is shown in Equation 16.

$$\text{F-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

B. EXPERIMENTAL PROCEDURE AND THEORETICAL ANALYSIS

1) INSTRUCTIONS INTENT CLASSIFICATION EXPERIMENT

Two main works were carried out in this section.

- 200 Instructions were randomly extracted from the ChineseQCI-TS dataset to test the KWECS method. And record the recognition accuracy and the maximum response time, and give several recognition examples.
- The test results were compared with the cosine similarity method and the keyword extraction algorithm, and the results were analyzed.

Table 4 shows the experimental results. For example, for the basic natural language query instructions, 57 instructions were used in the experiments for testing, and the cosine similarity method could recognize 27; the key-word extraction algorithm could recognize 52, and KWECS could recognize 53. The accuracy of the cosine similarity method was 63.5%, the accuracy of the keyword extraction algorithm was 81.0%, and the accuracy of KWECS was 96.5%. The main reason is that KWECS namely considers the importance of words and can extract verbs quickly, and at the same time considers the semantic connection between words. Even if the extracted key verbs cannot be successfully matched with the control dictionary, the most similar words can be identified by calculating the cosine similarity, which greatly improves the accuracy rate. In terms of response time, the cosine similarity response time is the longest, mainly because it treats the importance of each word as the same and needs to calculate the similarity between each word, which consumes more time. the response time of the keyword extraction algorithm, although due to KWECS, has 15.5% less accuracy compared

Field	Total	Type and number	Sample instruction
Agricultural	200	Basic query instruction(50)	大棚的温度是多少？ What's the temperature of the greenhouse
		Basic control instruction(50)	打开大田的灯光。Turn on the field lights.
		Complex query instruction(50)	今天大棚温度大于 20 度的时刻有哪些？ What is The Times when the temperature of the greenhouse is above 20 degrees today?
		Complex control instruction(50)	请帮我分别打开大棚的灯光和空调。Please help me turn on the light and the air conditioning respectively.
Industrial	200	Basic query instruction(50)	机房现在甲醛浓度是多少？ What is the current formaldehyde concentration in the machine room?
		Basic control instruction(50)	关闭机房的吸尘机。Turn off the vacuum cleaner in the machine room.
		Complex query instruction(50)	今天机房的平均温度比昨天高多少？ How much higher is the average temperature in the room today than it was yesterday?
		Complex control instruction(50)	先打开机房的吸尘机再关闭无尘室的中央空调。First, turn on the vacuum cleaner in the machine room and then turn off the central air conditioning in the cleanroom.
Smart home	200	Basic query instructions(50)	卧室的温度是多少？ What's the temperature of the bedroom?
		Basic control instruction(50)	打开厨房的油烟机。Turn on the kitchen hood.
		Complex query instruction(50)	请问今天卧室的平均温度比客厅的平均温度高多少？ How much higher is the average temperature in the bedroom than in the living room today?
		Complex control instruction(50)	先打开卧室的台灯，再关闭卫生间的浴霸。Turn on the lamp in the bedroom, then turn off the bath bully in the bathroom.

FIGURE 17. Example of test instructions for different domains of the ChineseQCI-TS dataset.

TABLE 4. The accuracy of different methods.

No.	Instruction Type	Cosine Similarity	Keyword Extraction	KWECS(ours)
1	Basic Query	27/57	52/57	53/57
2	Complex Query	21/32	18/32	31/32
3	Basic Control	21/34	23/34	32/34
4	Complex Control	58/77	69/77	77/77
	Accuracy	63.5%(127/200)	81.0%(162/200)	96.5%(193/200)
	MAX Time	0.85s	0.72s	0.74s

TABLE 5. Test results of basics natural language query and control instructions.

No.	Number	type	Accuracy	Precision	Recall	F-score	Max Time
1	50	Query	93.47%	92.15%	94.25%	93.19%	0.82s
2	50	Control	94.15%	93.57%	92.14%	92.85%	0.74s
3	100	Query	90.14%	91.24%	91.71%	91.47%	0.67s
4	100	Control	91.25%	92.25%	92.14%	92.19%	0.89s
5	150	Query	88.91%	89.65%	88.54%	89.06%	0.87s
6	150	Control	88.15%	87.58%	89.91%	88.73%	0.73s
	Avg		**90.13%**	**90.37%**	**90.79%**	**90.58%**	**0.79S**

to the latter. the maximum response time of KWECS is 0.74s, which is in line with the intention recognition efficiency.

For clarity and visibility, the type color of the text in the given example is aligned with the label color. The yellow color indicates words or theirs near synonyms that can be matched in the control dictionary. The blue one indicates that the instruction contains a verb, but cannot be matched with the dictionary. Red indicates that the instruction contains no verb, and the default is query instruction. The result is shown in Figure 18.

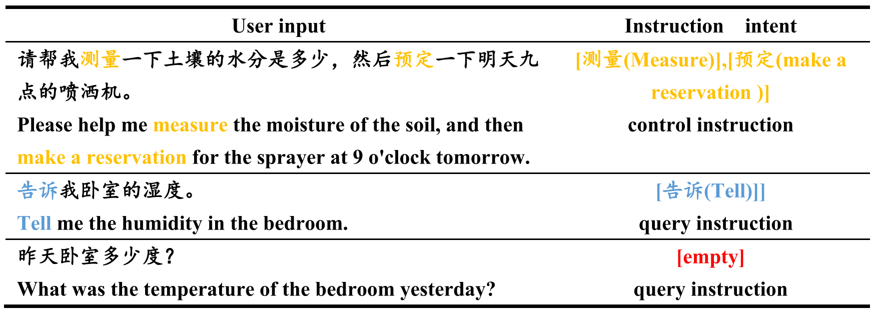
2) TF-IDF+COS_SIM BASED INSTRUCTION SEMANTIC PARSING EXPERIMENT

To evaluate the performance of the TF-IDF+COS_SIM instruction semantic parsing algorithm, the basic query instruction and the basic control instruction in the ChineseQCI-TS data set are used to carry out experiments. Because the quality of the returned results and the time it takes to generate the results may vary depending on the size of the dataset, each experiment is conducted for a different number of instructions. By using the method of random allocation, the data sets of basic natural language instruction in three fields are randomly divided into 100,200 and 300, of which basic query instruction and basic control instruction account for half respectively. It is then inputted into Algorithm 1 to test its performance, and the maximum time of Accuracy, Precision, Recall, F-score and return result is shown in Table 5.

The following insights can be obtained from Table 5.

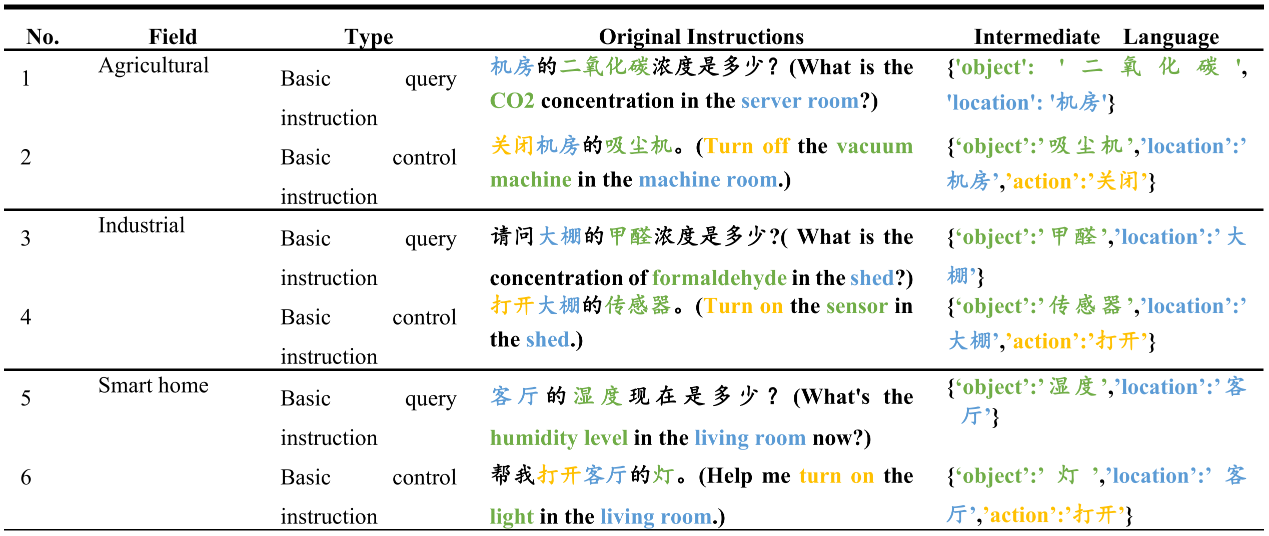
- On the test set, the average values of the four major assessment metrics are 90.13%, 90.37%, 90.79%, and 90.58%, which are all above 90%, and they all achieve good results.
- With the increase in the number of instructions, the evaluation index values all show a decrease, which is between 2% and 6%, which is within an acceptable range.
- The maximum flat response time is about 0.79s, and the response time can satisfy the real-time control and query functions.

Figure 19 gives a summary of the results of this study. In order to correspond to the parameter dependency graph in Section 3, set the yellow font to indicate the action attribute,



User input	Instruction intent
请帮我测量一下土壤的水分是多少，然后预定一下明天九点的喷洒机。 Please help me measure the moisture of the soil, and then make a reservation for the sprayer at 9 o'clock tomorrow.	[测量(Measure)],[预定(make a reservation)] control instruction
告诉我卧室的湿度。 Tell me the humidity in the bedroom.	[告诉(Tell)]] query instruction
昨天卧室多少度？ What was the temperature of the bedroom yesterday?	[empty] query instruction

FIGURE 18. Example of instruction classification.



No.	Field	Type	Original Instructions	Intermediate Language
1	Agricultural	Basic query instruction	机房的二氧化碳浓度是多少？(What is the CO2 concentration in the server room?)	{'object': '二氧化碳', 'location': '机房'}
2		Basic control instruction	关闭机房的吸尘机。(Turn off the vacuum machine in the machine room.)	{'object':'吸尘机','location':'机房','action':'关闭'}
3	Industrial	Basic query instruction	请问大棚的甲醛浓度是多少?(What is the concentration of formaldehyde in the shed?)	{'object':'甲醛','location':'大棚'}
4		Basic control instruction	打开大棚的传感器。(Turn on the sensor in the shed.)	{'object':'传感器','location':'大棚','action':'打开'}
5	Smart home	Basic query instruction	客厅的湿度现在是多少？(What's the humidity level in the living room now?)	{'object':'湿度','location':'客厅'}
6		Basic control instruction	帮我打开客厅的灯。(Help me turn on the light in the living room.)	{'object':'灯','location':'客厅','action':'打开'}

FIGURE 19. Example of structured parsing of instructions.

TABLE 6. Test results of complex natural language control instructions.

No.	Number	type	Accuracy	Precision	Recall	F-score	Max Time
1	50	Algorithm2	87.24%	89.14%	87.64%	88.59%	1.421s
2		**Algorithm2+3**	**92.16%**	**95.10%**	**93.27%**	**94.18%**	**1.468s**
3	100	Algorithm2	81.21%	80.09%	81.36%	80.66%	1.462s
4		**Algorithm2+3**	**90.10%**	**89.81%**	**90.65%**	**90.22%**	**1.475s**
5	150	Algorithm2	77.90%	78.02%	79.37%	78.68%	1.468s
6		**Algorithm2+3**	**88.15%**	**87.58%**	**89.91%**	**88.73%**	**1.479s**
	Avg	Algorithm2	80.56%	80.56%	81.41%	81.49%	1.450s
		Algorithm2+3	**89.47%**	**89.58%**	**90.72%**	**90.15%**	**1.474s**

the blue font to indicate the position attribute, and the green font to indicate the controlled object attribute.

3) EXPERIMENTS ON SEMANTIC PARSING OF INSTRUCTIONS BASED ON DEPENDENT SYNTACTIC ANALYSIS

This section uses the complex control instructions in the ChineseQCI-TS dataset as shown in the table and sets the number of instructions to 50,100,150. Then, it is input into the combined algorithm of algorithm 2 and algorithm 2+3 to test their performance, and the maximum time of Accuracy, Precision, Recall, F-score, and returned results are shown in Table 6.

The following insights can be obtained from Table 6.

- Both Algorithm 1 alone on the complex control instructions test set and Algorithm 1 + Algorithm 2 together on the complex control instruction test set achieved good results, where the average F-score reached 82.64% and 91.04%, respectively.
- As the number of complex control instructions increases, the performance of both Algorithm 2 and Algorithm 2 optimized by Algorithm 3 decreases to some extent. The four evaluation metrics before optimization decreased by an average of 4.67%, 5.56%, 4.14%, and 4.96%, respectively. After optimization, the four evaluated metrics decrease by 2.01%, 2.76%, 2.05%, and 2.73%, respectively. It is clear that the decrease in the values of the indicators is also reduced with the optimized algorithm, which shows that the optimized algorithm plays a significant role in the process of complex control instruction resolution.
- The maximum average response time is about 1.474s, which satisfies the real-time human-computer interaction.

No.	Field	Original Instructions	Intermediate Language
1	Agricultural	先打开温室的加热机,然后关闭大棚的喷雾器。(Turn on the greenhouse heater first, then turn off the sprayer in the greenhouse)	[{'action': '打开', 'location': '温室', 'object': '加热机'}, {'action': '关闭', 'location': '大棚', 'object': '喷雾器'}]
2	Industrial	先打开高温炉的电机,然后关闭热电厂的阀门。(Turn on the motor of the high-temperature furnace first, then close the valve of the thermal power plant.)	[{'action': '打开', 'location': '高温炉', 'object': '电机'}, {'action': '关闭', 'location': '热电厂', 'object': '阀门'}]
3	Smart home	请帮我先打开客厅的电风扇，然后再帮我打开一下卧室的空调，谢谢你了。(Please help me turn on the electric fan in the living room first, and then turn on the air conditioner in the bedroom, thank you very much.)	[{'action': '打开', 'location': '客厅', 'object': '电风扇'}, {'action': '打开', 'location': '卧室', 'object': '空调'}]

FIGURE 20. Example of structured parsing of instructions.

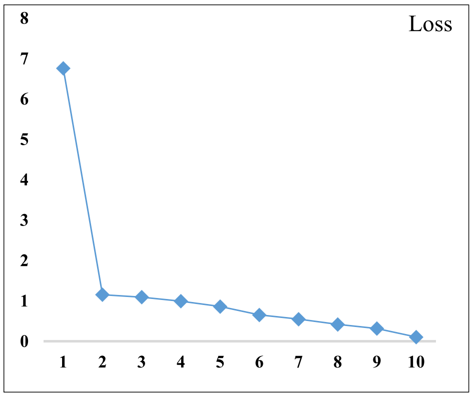


FIGURE 21. Example of structured parsing of instructions.

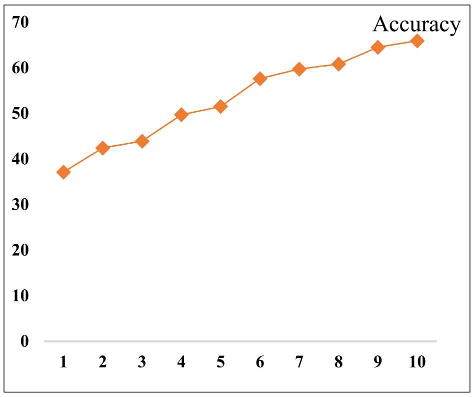


FIGURE 22. Example of structured parsing of instructions.

Figure 20 shows three examples of the actual instruction parsing.

4) NL-TO-SQL MODEL TRAINING EXPERIMENTS

In this experiment, the Bert pretraining model is the Chinese-Bert-WWM version of HINU. The maximum length of one-off coding (maxlen) is 80, the learning rate is 5E-5, the minimum learning rate is 1E-5, the input batch_size of each batch is 16, the epochs = 10, and the sample number of each batch is 32. The Adam algorithm is used for optimization, and the loss function is sparse_categorical_crossentropy. The loss iteration curve and accuracy iteration curve of the test set are shown in Figure 21 and Figure 22 respectively.

The loss value reached convergence after 10 batch iterations, in which the accuracy of the test set reached 65.9%. Considering that the model structure is relatively simple, it basically meets the requirements of the complex query instruction parsing module of SCADA-NLI. Then we selected 150 complex query instructions in the ChineseQCI-TS dataset to test the model, and the test results of each field are shown in Table 7.

TABLE 7. Test results of complex natural language query instructions.

NO.	Number	sel	agg	where	Max time
1	50	71.34%	74.47%	52.48%	1.78s
2	100	66.27%	68.69%	49.35%	1.89s
3	150	61.28%	62.28%	47.5%	1.83s
	Avg	**64.62%**	**66.45%**	**48.95%**	**1.83s**

As the number of instructions increases, the test accuracy of each field decreases, and the average accuracy is only 66.45%. The main reason is that there is no high-quality training data set in a specific field, so the accuracy is low during the test. However, considering that the natural language problems in the human-computer interaction process are mostly control types, and the normal production and life will not query too complex problems, the overall accuracy of 65.9% is also within a reasonable range. Average maximum response time is 1.83s, which satisfies the real-time nature of human-computer interaction.

TABLE 8. Combined service performance test results.

No.	Number	Accuracy	Precision	Recall	F-score	accuracy	MAX Time
1	50	96.27%	98.26%	97.37%	97.81%	96.00%	2.07s
2	100	92.21%	94.62%	93.31%	93.96%	95.00%	2.18s
3	150	89.21%	91.23%	90.39%	90.81%	94.67%	3.29s
4	200	84.10%	85.23%	84.91%	85.07%	93.50%	2.27s
	Avg	**88.47%**	**90.21%**	**89.48%**	**89.72%**	**94.40%**	**2.45s**

TABLE 9. Comparison of experimental results.

No.	Method	Accuracy	Precision	Recall	F-score
1	Seq2Seq	18.72%	19.21%	17.71%	18.43%
2	Word2vec-Seq2Seq	21.21%	22.39%	21.07%	21.71%
3	Word2vec-BiLSTM-Seq2Seq	25.27%	26.67%	27.78%	27.21%
4	Word2vec-BiLSTM-Attention-Seq2Seq	31.33%	33.39%	34.48%	33.93%
5	Template Matching	78.67%	79.97%	80.64%	81.32%
6	**Our Method**	**90.67%**	**91.69%**	**92.21%**	**91.95%**

5) SCADA-NLI COMPOSITE SERVICE PERFORMANCE TEST EXPERIMENT

In this section, we package all the algorithms together for combined service performance testing. First, 200 natural language instructions are randomly selected on the ChineseQCI-TS dataset proposed in this paper, which contains basic natural language query instructions, basic natural language control instructions, complex natural language query instructions, and complex natural language control instructions. Then we input the natural language instructions into SCADA-NLI and record the generated results and the maximum time spent to generate the results. Here, the evaluation metrics Accuracy, Precision, Recall, and F-score are still used to measure the overall performance of SCADA-NLI. The correctness of the intent classification effect of the instructions is also recorded and used to analyze whether the SCADA-NLI equivalent abstraction workflow proposed in Section 4 can complete a reasonable algorithm invocation service efficiently and with high quality. The experimental results are shown in Table 8.

The following insights can be obtained from Table 8.

- Average size of the corresponding evaluation indexes is 88.47%, 90.21%, 89.48% and 89.72% respectively. It is observed that the accuracy of the training data may decrease because the training data is not completely targeted to the application field of this article in the complex query instruction training module. Then, the number of incorrect intermediate language data structures generated is small, and the overall average evaluation index exceeds 88%. Therefore, it can be concluded that the service performance test results of the proposed method are satisfactory.
- The average accuracy of the algorithm calls is 94.40%, and the KWECS algorithm combines the semantics of words and the importance of words to achieve a better result in the classification of instructions on small data sets, which makes the intention recognition get better effect.
- The maximum average response time is 2.45s. Considering that the SCADA-NLI architecture contains multiple algorithm calls, the response time meets the requirements.

6) COMPARATIVE EXPERIMENT AND ANALYSIS

The comparative experimental ideas set up in this paper contain two main categories, the first one is the end-to-end concept, where unstructured natural language instruction is input to automatically generate a structured instruction, and the other one is the template matching approach with manual rules. In the end-to-end concept, this study uses the original seq2seq model, the Seq2Seq model after using word2vec word embedding, the Seq2Seq model after combining word2vec, BiLSTM, and the Seq2Seq model combined with word2vec, BiLSTM, and Attention for comparison experiments, respectively. The dimension of the word vector is 150 dimensions, the number of neurons per layer of LSTM is 256, the batch size (the size of batch gradient descent) is 64, and the epoch is set to 100. 450 items from the ChineseQCI-TS dataset are taken as the training set, and the remaining 150 instructions are used as the test set to obtain the relevant performance evaluation index values. In the manual rule-based template matching method [27], it generally means analyzing the natural language instructions to be structured by keyword information, analyzing the syntactic pattern features in the text by manual reading method, and thus writing the key metric extraction algorithm to transform the unstructured natural language instructions into structured natural language instructions. Finally, for the fairness of the experiment, 150 instructions are used for testing in both template matching and the method proposed in this paper. The final experimental results are shown in Table 9.

The following insights can be obtained from Table 9.

- Experiment numbers 1, 2, 3, and 4 are all end-to-end concepts, and the experimental goal is to generate unstructured natural language instructions, such as “请帮我分别打开卧室的台灯和空调”(“please help me turn on the desk lamp and air conditioner in my bedroom”), into a structured instruction “[打开卧室台灯][打开卧室空调]”(“[Turn on Bedroom Desk Lamp][Turn on Bedroom Air Conditioner]”), but the experimental accuracy, precision, recall, and F-value are not ideal, mainly due to the lack of a large amount of high-quality training corpus, which makes the model appear under fitted and the feature extraction is not obvious.
- The accuracy, precision, recall, and F-value of the template matching-based method reached 78.67%, 79.97%, 80.64%, and 81.32%, respectively. It is significantly ahead of the Seq2Seq model and related variants. The main reason is that it is relatively easy to find rules in a limited field of natural language. However, the artificial rule-based approach requires a lot of human resources to find rules, and if in open domains, there are differences in representations between different instructions, so it is necessary to specify different rules, which shows that the

portability of the approach and its scientific generalization is poor.

- The method proposed in this paper recommends different natural language semantic parsing algorithms according to the different features of instructions. The TF-IDF+COS_SIM algorithm is recommended for basic natural language instructions, and the maximum structured extraction of instructions is accomplished by the idea of keyword extraction and structural similarity optimization. For complex natural language control instructions, the structured extraction of long instructions is accomplished to a greater extent by assembling domain-specific rules based on the results of dependency analysis. And for complex natural language query instructions, it relies on Bert's powerful coding and parallel computing power, combined with NL2SQL rules, to transform natural language into SQL statements, and realize the semantic parsing of complex natural language query instructions. Thus the overall architecture has accuracy, precision, recall, and F-value of 90.67%, 91.69%, 92.21%, and 91.95%, respectively. Finally, the architecture proposed in this paper can recommend different algorithms for different types of instructions and can achieve natural instruction parsing to a large extent.

VII. CONCLUSION

This paper presents a natural language query and control interface (SCADA-NLI) for distributed SCADA systems, The interface understands natural language instructions input by users through a hierarchical classification semantic parsing algorithm. A ChineseQCI-TS dataset is also provided in the form of crowdsourcing to test the performance of the interface. In terms of determining the input instruction intent, the KWECS method is proposed to determine the user intent using the importance of words in the instructions and the semantic connections be-tween words. we conducted several experiments in terms of accuracy and maximum response time, and the KWECS method achieved an accuracy of 96.5% and a maximum response time of only 0.74 s. In the case of basic query and control instructions, using the TF -IDF+COS_SIM algorithm can satisfy the user's request. In the complex control instruction module, the instruction parsing algorithm based on dependent syntactic analysis is used to parse the user input instructions, and finally, in the complex query instruction module, a BERT-based NL-to-SQL model is constructed to complete the complex query instruction parsing. In evaluating the performance of the individual modules and the combined service separately, we conducted several experiments in terms of accuracy, precision, recall, F-value, and maximum response time for both. The interface was evaluated at 88.47%, 90.21%, 89.48%, 89.72%, and 2.45 s. And significantly outperforms the classical deep learning methods during the comparison experiments. The experiments showed that the interface has high accuracy and efficiency in understanding user semantics in human-computer interaction. In the process of instruction parsing in natural language, there is a phenomenon that the instruction cannot be parsed completely. For example, in the basic query and control instruction parsing module, the calculation of semantics depends on the quality of the trained word vector model. In this paper, word2vec is used to train Chinese Wikipedia data. There is a problem that extracting keywords cannot be matched in the word vector model, and word2vec cannot solve the problem of polysemy of a word. In future work, we will study the use of BERT, ELMO, and other pre-training models to encode keywords, and the above problems will be solved. In the complex control instruction parsing module, we only consider active sentence parsing, for passive sentence parsing is not carried out, such instructions as "请帮我把卧室的台灯和空调分别打开。"("please help me turn on the bedroom lamp and air conditioner respectively.") will not be successfully parsed, and in the future, we will consider the form of graph neural network to encode the instructions, to solve such problems. In the complex query instruction parsing module, the disadvantages are obvious. Considering the reasons for the data set and application field, we have not made a comparison with any baseline. In the future, we intend to improve the accuracy of complex query and support higher extensibility by setting up a knowledge map in combination with the neo4j database. In addition, the complexity of the algorithm will be optimized in the future to reduce the service response time of the interface and improve the real-time performance of human-computer interaction. In a word, our method provides more convenient interactive means for industrial and agricultural information management and control.

REFERENCES

[1] F. Ren and Y. Bao, "A review on human-computer interaction and intelligent robots," *Int. J. Inf. Technol. Decis. Making*, vol. 19, no. 1, pp. 5–47, Jan. 2020, doi: 10.1142/S0219622019300052.

[2] J. R. Bellegarda and K. E. A. Silverman, "Natural language spoken interface control using data-driven semantic inference," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 267–277, May 2003.

[3] Z. Zheng, M. Zhai, H. Peng, Y. Meng, Y. Gao, and Q. Wu, "Architecture and key technologies of distributed SCADA system for power dispatching and control," *Dianli Xitong Zidonghua/Automat. Electr. Power Syst.*, vol. 41, pp. 71–77, Mar. 2017.

[4] R. G. Byford, "Voice assistant system," *J. Acoust. Soc. Amer.*, vol. 133, no. 4, p. 2520, 2013.

[5] Y. Su, A. H. Awadallah, M. Khabsa, P. Pantel, M. Gamon, and M. Encarnacion, "Building natural language interfaces to Web APIs," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, Singapore, Nov. 2017, pp. 177–186.

[6] Q. Min, Y. Shi, and Y. Zhang, "A pilot study for Chinese SQL semantic parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, Nov. 2019, pp. 3652–3658.

[7] E. T. H. Chu and Z.-Z. Huang, "DBOS: A dialog-based object query system for hospital nurses," *Sensors*, vol. 20, no. 22, pp. 1–15, 2020, doi: 10.3390/s20226639.

[8] F. Ye and Y. Ma, "Research on Web page classification method based on query log," *J. Shanghai Jiaotong Univ. Sci.*, vol. 23, no. 3, pp. 404–410, Jun. 2018, doi: 10.1007/s12204-017-1899-0.

[9] A. Veilumuthu and P. Ramachandran, "Intent based clustering of search engine query log," in *Proc. IEEE Int. Conf. Automat. Sci. Eng. (CASE)*, Bangalore, India, Aug. 2009, pp. 647–652.

[10] M. Y. H. Setyawan, R. M. Awangga, and S. R. Efendi, "Comparison of multinomial naive Bayes algorithm and logistic regression for intent classification in chatbot," in *Proc. Int. Conf. Appl. Eng. (ICAE)*, Batam, Indonesia, Oct. 2018, pp. 1–5.

[11] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 2931–2939.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language un-derstanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998, doi: 10.1109/5.726791.

[17] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 207–212.

[18] V. Khomenko, O. Shyshkov, O. Radyvonenko, and K. Bokhan, "Accelerating recurrent neural network training using sequence bucketing and multi-GPU data parallelization," in *Proc. IEEE 1st Int. Conf. Data Stream Mining Process. (DSMP)*, Lviv, Ukraine, Aug. 2016, pp. 100–103.

[19] M. Verleysen and D. Francois, "The curse of dimensionality in data mining and time series prediction," in *Proc. 8th Int. Workshop Artif. Neural Netw. (IWANN)*, Vilanova i la Geltrú, Spain, Jun. 2005, pp. 758–770.

[20] B. Goodman and A. Grosz, "Research in knowledge representation for natural language communication and planning assistance," Final Rep., Sep. 1988.

[21] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[22] S. Kamath and V. S. Ananthanarayana, "Discovering composable Web services using functional semantics and service dependencies based on natural language requests," *Inf. Syst. Frontiers*, vol. 21, no. 1, pp. 175–189, Feb. 2019, doi: 10.1007/s10796-017-9738-2.

[23] T. Chiyuan, C. Dehua, W. Mei, and L. Jiajin, "Structured processing for pathological reports based on dependency parsing," *J. Comput. Res. Develop.*, vol. 53, no. 12, pp. 2669–2680, 2016.

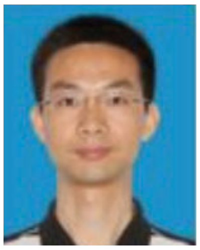
[24] X. Xu, C. Liu, and D. Song, "SQLNet: Generating structured queries from natural language without reinforcement learning," Cornell Univ. Library, Ithaca, NY, USA, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/abs/1711.04436

[25] T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev, "TypeSQL: Knowledge-based type-aware neural text-to-SQL generation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, New Orleans, LA, USA, Jun. 2018, pp. 588–594.

[26] W. Hwang, J. Yim, S. Park, and M. Seo, "A comprehensive exploration on WikiSQL with table-aware word contextualization," Cornell Univ. Library, Ithaca, NY, USA, Tech. Rep., 2019. [Online]. Available: https://arxiv.org/abs/1902.01069

[27] A. Abraham, "Rule-based expert systems," in *Handbook of Measuring System Design*. Hoboken, NJ, USA: Wiley, 2005.

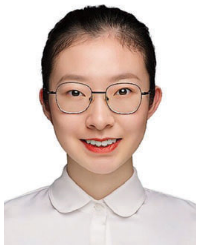
HAO WU was born in Hefei, Anhui. He is currently pursuing the degree in computer science and technology with the School of Information and Computer Science, Anhui Agricultural University. His research interests include machine learning, natural language processing, and human–computer interaction systems.



CHUNSHAN SHEN was born in Anhui, China, in 1980. He received the M.S. and Ph.D. degrees from the University of Chinese Academy of Sciences, in 2005 and 2010, respectively. He is currently a Senior Engineer with the School of Information and Computer, Anhui Agricultural University. His research interests include management and control integration, natural language processing, and so on.

ZHUANGZHUANG HE (Student Member, IEEE) was born in Anhui, China. He is currently pursuing degree in computer science from Anhui Agriculture University. His research interests include deep learning and recommendation systems.

YONGMEI WANG was born in Hefei, Anhui, in 1974. She received the master's degree in computer science and technology from the School of Information and Computer Science, Hefei University of Technology, in 2010. She has been engaged in teaching and research in the fields of big data, agricultural informatization, and agricultural products traceability. She has presided over and participated in more than ten national and provincial educational and scientific research projects. She has published nearly 20 articles. She is editing and participating in the compilation of more than ten provincial and ministerial planning textbooks; three invention patents; and more than ten software copyrights have been approved.



XINYUAN XU was born in Tianjin, in 2001. She is currently pursuing the degree in accounting with Hunan University. Her research interests include statistics and the application of statistics to artificial intelligence.