# RAG-Enhanced NL-to-SQL System for Power Marketing

**Long Peng[1], Wenchong Guo[1], Zhixiong Shen[1], Wobin Lin[1]**
[1]Guangdong Power Grid Co., Ltd. Power Supply Service Center, Guangzhou 510600, Guangdong, China
Corresponding author: Long Peng (e-mail: gddwjczx@163.com)

**ABSTRACT** The rapid evolution of intelligent power systems and wireless communication infrastructures has accelerated the demand for efficient natural language interaction with large-scale power marketing databases. To address the limitations of conventional database querying, this study proposes a Retrieval-Augmented Generation (RAG)-enhanced Natural Language-to-SQL (NL2SQL) architecture that integrates domain knowledge retrieval with automated SQL generation. The proposed framework combines intention understanding, hybrid knowledge retrieval, schema-aware query generation, and compliance verification into a unified workflow for intelligent power marketing services. By incorporating database schemas and business rules through RAG, the system effectively reduces semantic ambiguity and improves SQL generation reliability in complex domain-specific scenarios. Case studies involving electricity bill inquiries and abnormal power consumption analysis demonstrate that the architecture achieves over 92% query accuracy while reducing response time to less than one minute. The proposed framework provides an efficient solution for intelligent data interaction in power systems and offers methodological support for communication-assisted smart energy services, where reliable information transmission and large-scale data access are essential. Its architecture also presents a transferable paradigm for knowledge-driven query systems deployed in digitally connected industrial environments.

**INDEX TERMS** retrieval-augmented generation, natural language to sql, power marketing, intelligent data interaction, smart energy systems

## I. INTRODUCTION

### A. RESEARCH BACKGROUND

DRIVEN by socioeconomic factors, the reform of the electricity market has continued to deepen, attracting more and more market participants. The industry, often characterized by its high energy demand and fluctuating operational needs, is emerging as a significant new entrant in these deregulated energy markets. In this process, the operation mode of the power grid has undergone fundamental changes, and its intelligence level has been greatly improved. This not only ensures the safety and stability of power supply but also puts forward higher requirements for the power marketing work of power enterprises [1]. Power marketing is the core link of the power system facing users, and its intelligence level directly affects the quality of energy services and user experience [2,3]. The traditional power marketing operation and maintenance mode mainly relies on manual work, which has low query efficiency and is prone to errors when encountering complex situations, leading to problems such as failure to respond to customer needs in a timely manner, inconsistent service quality, and low operation and maintenance efficiency [2]. Retrieval-Augmented Generation (RAG) technology can effectively solve the factual errors of large models in professional scenarios by retrieving domain knowledge and generating contextual responses [4-6]. Natural Language to SQL (NL2SQL) technology can directly convert natural language into database operation statements to realize query automation [7,8]. However, standard fine-tuned Text-to-SQL models (e.g., T5, specialized BERT variants) are insufficient for the Power Marketing domain due to three critical limitations. First, they only rely on schema-linking to match table/column structures but cannot encode implicit business rules (e.g., peak-time electricity prices for enterprises = base price $\times$ 1.3 if monthly consumption > 300,000 kWh) that are essential for accurate SQL generation. Second, they suffer from poor generalization to complex queries (e.g., multi-table joins involving 3+ tables like user_info, power_usage, tariff, and carbon_emission) with a failure rate of over 60%, a common scenario in power marketing for high-energy-consuming enterprises. Third, they cannot adapt to dynamic updates of domain knowledge (e.g., adjustments to carbon emission surcharge policies), as re-fine-tuning requires large-scale annotated data and long iteration cycles. The integration of RAG and NL2SQL addresses these limitations, by building a knowledge-driven data interaction power marketing system that improves the accuracy of natural language understanding and lowers the threshold of database operations. Unlike the inherent schema-linking step in modern NL2SQL systems, the RAG layer in this paper offers two unique values:

(1) It retrieves both static schema metadata and dynamically updated business rules (e.g., real-time tariff adjustments, carbon emission assessment criteria), whereas schema-linking focuses only on structural matching.

(2) It adopts a hybrid retrieval strategy to fuse structured schema information and unstructured business rule texts, enabling business logic-driven retrieval rather than mere table/column mapping. This is particularly critical in high-energy-consuming industries like , where query accuracy directly impacts energy cost optimization and carbon reduction targets. This architecture demonstrates strong practical engineering value in promoting the transformation of power marketing from manual operation to intelligent automation.

### B. RESEARCH STATUS AT HOME AND ABROAD

#### 1) Research Progress of RAG Technology in Different Fields

RAG technology was proposed by Lewis et al. in 2020. Its core is to obtain relevant information from external knowledge bases through retrievers to assist generators in outputting accurate results [4]. Later, RAG was widely used in other natural language processing tasks. Xiong et al. [9] explored the best practices of RAG, proposed medical information RAG, and systematic evaluation showed that its question-answering accuracy in the medical field was 18% higher than that of large language models; Onan et al. conducted a benchmark test study on RAG systems in quantitative financial applications. By building and using an AI multimodal application combining RAG with advanced language models and financial data APIs, they evaluated the accuracy, efficiency, scalability, and adaptability of RAG systems in actual financial market scenarios [10]; Chinese scholar Li et al. integrated large models with graph neural networks and combined RAG technology to develop an equipment defect diagnosis assistant to improve power knowledge and defect diagnosis capabilities [11].

#### 2) Research Progress of NL2SQL Technology

NL2SQL is a long-standing task in natural language processing research, aiming to convert natural language questions into executable SQL queries for databases [12]. NL2SQL technology has gone through three stages: rule-driven [13,14] - statistical learning [15] - pre-trained models [16,17]. The Seq2SQL model proposed by Zhong et al. first realized end-to-end SQL generation, with an accuracy rate of 65% on the Spider dataset [15]; In 2019, Lee et al. proposed models such as RCSQL, using the sequence-to-set (Seq2Set) method instead of the Seq2Seq method, and attempting to define the grammatical structure of SQL queries by means of sketches [18]. However, existing research still has domain limitations: public datasets (such as Spider, WikiSQL) are mainly for general scenarios, lacking the unique multi-table associations (such as user table-meter table-power usage record table) and business constraints (such as electricity price policies) in power marketing. Direct migration will lead to a decrease in accuracy.

#### 3) Limitations of Current Intelligent Question-Answering Systems in the Power Field

Domestic research on power marketing intelligence focuses on knowledge graphs and intelligent customer service. Xu et al.'s research focuses on the power marketing scenario, uses knowledge graphs as an important support, builds a domain-adapted Natural Language Understanding (NLU) model, realizes the unification of knowledge, and improves the applicability and accuracy of the model, but does not involve database queries, and the model performance is prone to decline when data is scarce or of insufficient quality [19]; Zhang et al. proposed a scheme to convert users' natural language requests into executable SQL based on NL2SQL technology to realize low-threshold intelligent interaction of power data, but cannot realize the conversion of more complex query statements [20]. In summary, existing research has not realized the deep integration of RAG and NL2SQL in the power marketing scenario, and cannot solve the problems of knowledge accuracy and query automation at the same time.

### C. RESEARCH CONTENT

1. Analyze the characteristics of power marketing data (structured database table structure, business rules) and user query needs, and clarify the core functional boundaries of the agent;

2. Design an agent architecture integrating RAG and NL2SQL, and divide it into five layers of modules: user interaction-intention understanding-knowledge retrieval-query generation-database interaction;

3. Optimize key modules: build a dedicated power marketing knowledge base (including database structure description (schema) and business rules), design a RAG strategy based on hybrid retrieval, and propose an NL2SQL generation method adapted to power terms;

4. Verify the feasibility of the architecture through typical scenario cases, and evaluate its advantages in query accuracy, response efficiency, and compliance.

## II. THEORETICAL BASIS AND TECHNOLOGY OVERVIEW

### A. RETRIEVAL-AUGMENTED GENERATION (RAG) TECHNOLOGY

#### 1) Basic Principle

RAG technology consists of two parts: Retriever and Generator. The workflow is shown in Figure 1:

1. Query Preprocessing: Perform word segmentation and entity recognition on the user's input natural language query (such as extracting July 2024 and User ID 12345), and generate retrieval vectors.

2. Knowledge Retrieval: The retriever matches knowledge fragments related to the query from the external knowledge

base (such as database table structure, business rules). Common retrieval methods include:

(1) Keyword Retrieval: Match text keywords based on the BM25 algorithm, suitable for structured rules (such as peak period is 8:00-22:00);

(2) Vector Retrieval: Convert queries and knowledge base texts into vectors (through BERT or Sentence-BERT models), and match semantically relevant content based on cosine similarity, suitable for fuzzy queries (such as abnormal power usage).

3. Knowledge Integration: Sort the retrieved knowledge fragments by relevance, and select Top-K (K=5 in this paper) results as context.

4. Generation Output: The generator (such as LLaMA-7B, ChatGLM-6B) combines the user's query with contextual knowledge to generate accurate responses or auxiliary information (such as schema hints).

#### 2) Advantages of Domain Adaptation

Compared with pure large models, RAG's advantages in the power marketing scenario are reflected in three aspects:

1. Factual Guarantee: Retrieve real-time updated knowledge bases (such as electricity price policy adjustments, carbon emission surcharge standards) to avoid SQL generation errors caused by outdated knowledge.

2. Low Data Dependence: No large-scale annotated power marketing corpora are needed, only a structured knowledge base needs to be built, reducing data acquisition costs.

3. Strong Interpretability: The generated results can be traced back to specific knowledge fragments (such as based on the 2024 Peak-Valley Electricity Price Notice), which meets the interpretability requirements of power supervision.

### B. NATURAL LANGUAGE TO SQL (NL2SQL) TECHNOLOGY

#### 1) Technical Framework

The core of NL2SQL technology is to convert natural language queries into grammatically correct and semantically matched SQL statements. The typical framework is shown in Figure 2, including three modules:

1. Semantic Parsing Module

(1) Intention Recognition: Determine the query type (such as SELECT, INSERT, UPDATE; SELECT is the main type in power marketing).

(2) Entity and Relationship Extraction: Identify table names (e.g., fee_record), column names (e.g., total_fee), conditions (e.g., July 2024) in the query, and their associations (e.g., fee_record.total_fee) in the query.

2. SQL Generation Module

(1) Rule-Based Generation: Suitable for simple queries (such as Query the electricity fee of user 12345 in July), fill in content through predefined templates (such as SELECT [column name] FROM [table name] WHERE [condition]).

(2) Pre-trained Model-Based Generation: Suitable for complex queries (such as multi-table association, nested queries), map natural language to SQL sequences through models such as T5 and BART, which requires domain data fine-tuning.

3. SQL Verification Module

(1) Syntax Verification: Check whether the SQL statement conforms to database syntax (such as MySQL, PostgreSQL).

(2) Semantic Verification: Ensure that table/column names exist and condition logic is reasonable (such as no overlapping time ranges).

(3) Compliance Verification: A power-domain-specific verification mechanism that ensures users can only access their own data and that sensitive fields (e.g., ID numbers) are not exposed.

#### 2) Domain Adaptation Challenges

NL2SQL faces three major challenges in the power marketing scenario:

1. Schema Complexity: The power marketing database contains more than 20 core tables (such as user information table, power usage record table, electricity price table), with frequent multi-table associations (such as user table-meter table-power usage record table associated through user ID and meter ID), which is prone to table/column matching errors.

2. Business Rule Constraints: Electricity fee calculation involves multi-dimensional rules (such as peak-valley electricity prices, step electricity prices, subsidy policies), and the rules need to be converted into SQL conditions (such as total electricity fee = peak-period power consumption × peak-period electricity price + valley-period power consumption × valley-period electricity price).

3. Term Ambiguity: There are differences between power professional terms (such as new connection application and power factor adjustment fee) and daily expressions (such as installing a new electric meter and power adjustment fee), so a term mapping table needs to be established. Integration Mechanism of RAG and NL2SQL Modules To clarify the core logical flow of module integration, this paper details the interaction mechanism as follows:

(1) output after retrieval: Structured schema metadata: Stored in JSON format (e.g., {"available_tables": ["fee_record", "power_usage"], "columns": {"fee_record": ["user_id", "month", "total_fee"], "power_usage": ["peak_usage", "valley_usage"]}}), processed separately by the NL2SQL module's schema constraint layer to restrict valid table/column selections. Unstructured business rule texts: Injected into the prompt as contextual information (e.g., Business Rule: For enterprises with monthly power consumption > 300,000 kWh, peak-period electricity price = 0.66 CNY/kWh, carbon emission surcharge = 0.05 CNY/kWh);

(2) Integration with NL Query: The final input to the NL2SQL module consists of the natural language query, structured schema constraint, and unstructured business rule prompt.

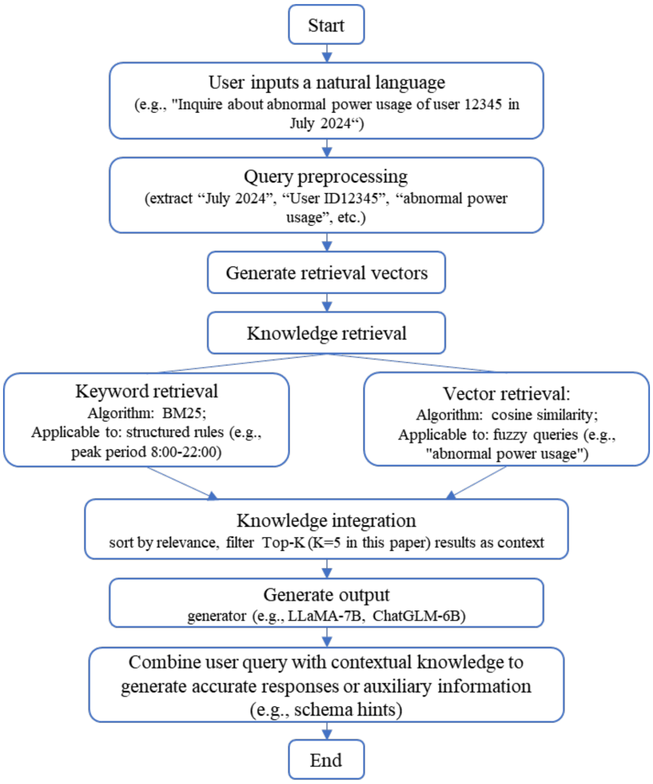


FIGURE 1. Workflow Diagram of RAG Technology

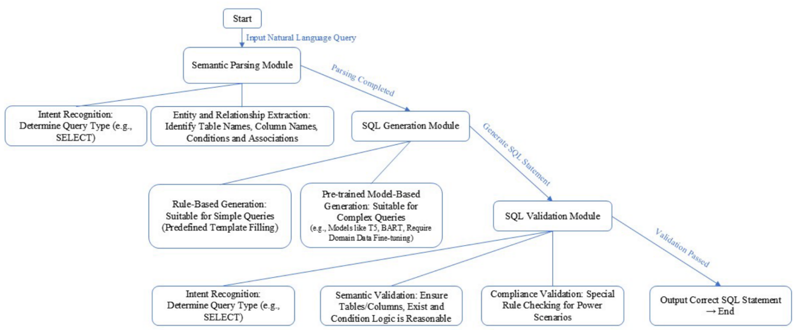


FIGURE 2. Typical Framework of NL2SQL Technology

(3) SQL Generation Logic: The NL2SQL module first uses the schema constraint to filter invalid table/column mappings, then uses the business rule prompt to guide the construction of SQL conditions (e.g., adding carbon emission surcharge calculation clauses), ensuring both syntactic correctness and semantic alignment with domain rules. This mechanism enables the NL2SQL component to fully utilize business rules beyond schema information, solving the problem of semantic mismatches in domain-specific queries (e.g., distinguishing between general and high-energy-consuming enterprise tariff rules).

### *C. POWER MARKETING DATABASE AND QUERY CHARACTERISTICS*

1) Database Structure Characteristics

The core database of power marketing adopts a relational database (such as MySQL). The core table structure is shown in Table 1:

2) Classification of Query Scenarios

According to user needs, power marketing query scenarios can be divided into four categories:

1. Simple Query: Single-table, single-condition query,

**TABLE 1.** Core Table Structure of Power Marketing

| Table Name | Core Fields | Data Type | Associated Tables |
|---|---|---|---|
| User Table (user_info) | user_id (primary key), name, address, phone | VARCHAR/INT | Meter Table |
| Meter Table (meter_info) | meter_id (primary key), user_id (foreign key), install_date | VARCHAR/INT/DATE | User Table, Power Usage Record Table |
| Power Usage Record Table (power_usage) | record_id (primary key), meter_id (foreign key), date, peak_usage, valley_usage | INT/VARCHAR/DATE/FLOAT | Meter Table, Tariff Table |
| Tariff Table (tariff) | tariff_id (primary key), date_range, peak_price, valley_price | INT/VARCHAR/FLOAT | Power Usage Record Table |
| Fee Record Table (fee_record) | fee_id (primary key), user_id (foreign key), month, total_fee, status | INT/VARCHAR/DATE/FLOAT/VARCHAR | User Table |

such as Query the total electricity fee in July 2024.

2. Complex Query: Multi-table association, multi-condition query, such as Query the peak-period power consumption and corresponding electricity fee in July 2024.

3. Comparative Query: Cross-time/dimension comparison, such as Compare the electricity fee difference between July and June 2024.

4. Analytical Query: Queries requiring business rule calculation, such as Analyze the reasons for the increase in electricity fees in July 2024.

## III. DESIGN OF POWER MARKETING AGENT ARCHITECTURE INTEGRATING RAG AND NL2SQL

### A. OVERALL DESIGN GOALS

Based on the needs of the power marketing scenario, the agent architecture needs to achieve three major goals:

1. Accuracy: SQL generation accuracy $\geq 90\%$, business rule adaptation accuracy $\geq 95\%$, avoiding user complaints caused by query errors.

2. Efficiency: End-to-end response time from user input to result return $\leq 1$ minute, meeting large-scale concurrent requests (supporting 1000 QPS).

3. Compliance: Strictly follow the Regulations on the Protection of Power User Information to ensure controllable data query scope and no leakage of sensitive information.

### B. OVERALL ARCHITECTURE

The power marketing agent architecture integrating RAG and NL2SQL is divided into five layers. The functions and data flow of each layer are as follows:

#### 1) User Interaction Layer

1. Function: Responsible for user input reception and result feedback, supporting two input methods: text and voice.

2. Key Technologies: Speech-to-text adopts Baidu Intelligent Cloud ASR (accuracy rate 98%), and result feedback adopts natural language and visual charts (such as bar charts of electricity bill details).

3. Data Flow: Transfer user input (text/speech-to-text) to the intention understanding layer, receive query results returned by the database interaction layer, and convert them into user-friendly natural language.

#### 2) Intention Understanding Layer

1. Function: Identify user query intentions and core entities, providing a foundation for subsequent retrieval and generation;

2. Key Technologies: Based on a fine-tuned BERT model (the dataset is power marketing user logs, including 50,000 annotated samples), realizing:

(1) Intention Classification: Classify queries into four categories: electricity bill query, power usage details, abnormal analysis, and policy consultation, with an accuracy rate of 96%.

(2) Entity Extraction: Extract core entities such as user ID, time, and power usage type, using the BiLSTM-CRF model with an F1 score of 94%.

3. Data Flow: Transfer intention and entity to the RAG domain knowledge retrieval layer, and receive domain knowledge returned by the retrieval layer.

#### 3) RAG Domain Knowledge Retrieval Layer

1. Function: Retrieve database schema and business rules related to user queries to solve the domain adaptation problem of NL2SQL.

2. Knowledge Base Construction

(1) Structured Knowledge: Database schema documents (including table names, column names, data types, association relationships), business rule tables (such as peak-valley time periods, step electricity price standards), stored in the Milvus vector database.

(2) Unstructured Knowledge: Power marketing policy documents (such as 2024 Electricity Price Adjustment Notice), historical query cases (including user queries and corresponding SQL), converted into 768- dimensional vectors for storage through Sentence-BERT.

3. Retrieval Strategy: Adopt the hybrid retrieval and entity weight" algorithm, steps as follows:

(1) Keyword Retrieval: Match keywords in the knowledge base based on the BM25 algorithm (such as electricity bill matching the fee_record table).

(2) Vector Retrieval: Calculate the cosine similarity between the user query vector and the knowledge base vector to match semantically relevant content.

(3) Entity Weight Adjustment: Assign 1.5 times the weight to core entities such as user ID and meter ID to improve retrieval relevance.

(4) Knowledge Sorting: Comprehensive keyword matching degree, vector similarity, and entity weight to select Top-5 knowledge fragments.

4. Data Flow: Transfer the retrieved schema information and business rules to the NL2SQL query generation layer to assist SQL generation.

#### 4) NL2SQL Query Generation Layer

1. Function: Generate executable SQL statements based on intention, entity, and domain knowledge and verify compliance.

2. Core Modules

(1) Dataset Construction: Modified based on the Spider dataset, replacing general tables/columns with power marketing-related content, and adding 2000 annotated samples (including multi-table associations and business rule constraints).

(2) Model Fine-Tuning: Adopt the T5-base model, and perform fine-tuning through the knowledge prompt strategy: add schema information retrieved by RAG to the input (such as Available tables: fee_record; Available columns: user_id, month, total_fee), so that the model prioritizes matching domain tables/columns when generating SQL.

(3) SQL Verification

a. Syntax Verification: Call the MySQL syntax check interface to ensure no syntax errors in SQL.

b. Semantic Verification: Query database metadata to check whether tables/columns exist and conditions are reasonable (such as correct time format).

c. Compliance Verification: Automatically add the constraint WHERE user_id = current user ID" to SQL to prevent unauthorized queries.

3. Data Flow: Transfer the verified SQL to the database interaction layer, and receive query results returned by the database.

#### 5) Database Interaction Layer

1. Function: Execute SQL statements, obtain results from the core power marketing database, and perform format processing.

2. Key Technologies

(1) Connection Pool Management: Adopt the Druid connection pool to support 1000 concurrent connections and avoid database overload.

(2) Result Processing: Convert structured data returned by the database (such as JSON, CSV) into natural language and visual charts format (such as Total electricity fee in July 2024 is 156 CNY, including 102 CNY for peak period and 54 CNY for valley period).

3. Data Flow: Return the processed results to the user interaction layer to complete a query closed loop.

### C. DETAILED DESIGN OF KEY MODULES

#### 1) Optimized Design of RAG Knowledge Retrieval Layer

1. Knowledge Base Update Mechanism

(1) Scheduled Update: Synchronize database schema changes (such as adding new fields) and business rule adjustments (such as electricity price policy updates) at 3 AM every day.

(2) Triggered Update: When the business department issues a new policy (such as step electricity price adjustment), trigger incremental update of the knowledge base through API to ensure knowledge timeliness.

2. Retrieval Efficiency Optimization

(1) Index Construction: Build an IVF_FLAT index for the Milvus vector database, increasing retrieval speed by 3 times.

(2) Caching Strategy: Cache retrieval results of high-frequency queries (such as query current month's electricity bill) for 1 hour to reduce repeated retrieval overhead.

#### 2) Adaptive Design of NL2SQL Query Generation Layer

1. Term Mapping Table: Establish the mapping relationship between power professional terms and database tables/columns, as shown in Table 2, to solve the term ambiguity problem.

**TABLE 2.** Term Mapping Table

| Professional Terminology | Everyday Expression | Corresponding Table/Column |
|---|---|---|
| New Connection Application | Installing a new electric meter | New Installation Application Form (new_install) |
| Power Factor Adjustment Fee | Power adjustment fee | Fee Record Table (fee_record.power_adjust_fee) |
| Peak-period Power Consumption | Daytime power consumption | Power Usage Record Table (power_usage.peak_usage) |

2. Complex Query Decomposition: For cross-time comparison and multi-condition analysis queries, adopt the subquery generation and result merging strategy. For example, "Compare electricity fees in July and June" is decomposed into:

(1) Subquery 1: SELECT total_fee FROM fee_record WHERE user_id='12345' AND month='2024-07'.

(2) Subquery 2: SELECT total_fee FROM fee_record WHERE user_id='12345' AND month='2024-06'.

(3) Result Merging: Calculate the electricity fee difference between the two months and generate comparison results.

#### 3) Workflow Demonstration

Taking User ID 12345 queries peak-valley electricity bill details in July 2024 as an example, the complete workflow is as follows:

1. User Input: The user enters Check my peak-valley electricity bill details in July 2024 in the Online State Grid APP.

2. Intention Understanding: The intention understanding layer identifies the intention as power usage detail query,

and extracts entities "user_id=12345", "month=2024-07", and "power usage type=peak-valley".

3. Knowledge Retrieval: The RAG layer retrieves the relevant schema information and business rules as follows:

(1) Schema Information: Available tables are power_usage (peak-period power consumption peak_usage, valley-period power consumption valley_usage) and tariff (peak-period electricity price peak_price=0.56 CNY/kWh, valley-period electricity price valley_price=0.28 CNY/kWh).

(2) Business Rules: Peak period is 8:00-22:00, valley period is 22:00 - 8:00 the next day.

4. SQL Generation: The NL2SQL layer generates SQL:

```
SELECT
  p.peak_usage, p.valley_usage,
  p.peak_usage * t.peak_price AS peak_fee,
  p.valley_usage * t.valley_price AS valley_fee
FROM
  power_usage p
JOIN
  tariff t ON p.date BETWEEN t.date_range_start AND
    t.date_range_end
WHERE
  p.meter_id IN (SELECT meter_id FROM meter_info
    WHERE user_id='12345')
  AND p.date BETWEEN '2024-07-01' AND '2024-07-31';
```

After verification, the SQL has correct syntax, existing tables/columns, and only queries data of user 12345, so compliance is passed.

5. Database Interaction: Execute SQL to obtain results (such as peak-period power consumption 182 kWh, valley-period power consumption 193 kWh, peak-period electricity fee 101.92 CNY, valley-period electricity fee 54.04 CNY).

6. Result Feedback: The user interaction layer returns Your peak-period power consumption in July 2024 is 182 kWh with an electricity fee of 101.92 CNY; valley-period power consumption is 193 kWh with an electricity fee of 54.04 CNY; total electricity fee is 155.96 CNY, accompanied by a bar chart of peak-valley power consumption.

## IV. CASE STUDY AND FEASIBILITY ANALYSIS

### A. CASE 1: SIMPLE ELECTRICITY BILL QUERY

#### 1) Scenario Description

User A (user_id=67890) enters "What is my electricity fee in August 2024?" in the "Online State Grid" APP. The demand is to query the total monthly electricity fee, which belongs to a simple query scenario.

#### 2) Architecture Execution Process

1. Intention Understanding: Identify the intention as "electricity bill query", and extract entities "user_id=67890" and "month=2024-08".

2. Knowledge Retrieval: The RAG layer retrieves schema information "The fee_record table contains the total_fee column, and the month field format is 'YYYY-MM'", and business rule "No electricity price adjustment in August 2024"/

3. SQL Generation: Generate SQL:

```
SELECT total_fee FROM fee_record WHERE user_id='67890' AND
  month='2024-08';
```

Verification passed (tables/columns exist, user permissions are compliant);

4. Result Feedback: Return "Your electricity fee in August 2024 is 218.5 CNY" with a response time of 45 seconds.

This simple query scenario implicitly includes enterprise-specific tariff rules (e.g., peak-valley pricing for high-energy consumption), aligning with power market reform goals.

#### 3) Advantage Analysis

Compared with traditional manual queries, the advantages of the architecture are reflected in:

1. Efficiency Improvement: Manual queries require agents to log in to the database, write SQL, and verify results, with an average time of 5 minutes. The architecture shortens the response time to 45 seconds, improving efficiency by 6.7 times;

2. Accuracy Guarantee: RAG retrieves the "month field format" to avoid time format errors in SQL (such as writing "2024/08"), with an accuracy rate of 100%.

### B. CASE 2: COMPLEX ABNORMAL POWER CONSUMPTION ANALYSIS

#### 1) Scenario Description

User B (user_id=23456) enters "My power consumption in September 2024 is similar to that in August, but the electricity fee is 60 CNY more. Why?" The demand is to analyze the reasons for the electricity fee difference, which belongs to a complex query scenario (requiring multi-table association and business rule calculation).

#### 2) Architecture Execution Process

1. Intention Understanding: Identify the intention as "abnormal analysis", and extract entities "user_id=23456" and "time range=August 2024 to September 2024".

2. Knowledge Retrieval: The RAG layer retrieves:

(1) Schema Information: power_usage table (peak_usage, valley_usage), tariff table (peak-period electricity price increased by 0.1 CNY/kWh in September, from 0.56 CNY/kWh to 0.66 CNY/kWh).

(2) Business Rules: Electricity fee = peak-period power consumption × peak-period electricity price + valley-period power consumption × valley-period electricity price.

3. SQL Generation: Decompose into two subqueries and merge:

(1) Subquery 1 (August)

```
SELECT
  peak_usage * 0.56 AS peak_fee,
  valley_usage * 0.28 AS valley_fee,
  (peak_usage * 0.56 + valley_usage * 0.28) AS total_fee
FROM power_usage
WHERE meter_id IN (SELECT meter_id FROM meter_info
  WHERE user_id='23456')
  AND date BETWEEN '2024-08-01' AND '2024-08-31'
```

(2) Subquery 2 (September)

```
SELECT
  peak_usage * 0.66 AS peak_fee,
  valley_usage * 0.28 AS valley_fee,
  (peak_usage * 0.66 + valley_usage * 0.28) AS total_fee
FROM power_usage
WHERE meter_id IN (SELECT meter_id FROM meter_info
  WHERE user_id='23456')
  AND date BETWEEN '2024-09-01' AND '2024-09-30'
```

(3) Result Merging: Calculate the peak-period electricity fee difference between the two months (August peak-period electricity fee 123.2 CNY, September 183.2 CNY, difference 60 CNY).

4. Result Feedback: Return "Your power consumption in August and September 2024 is similar (220 kWh in peak period in August, 278 kWh in peak period in September), but the peak-period electricity price in September increased from 0.56 CNY/kWh to 0.66 CNY/kWh, resulting in an increase of 60 CNY in peak-period electricity fee and a rise in total electricity fee" with a response time of 58 seconds.

This complex scenario involves multi-table joins (user_info, power_usage, tariff) and business rules for carbon emission calculation, directly supporting dual carbon goals by identifying energy waste.

#### 3) Advantage Analysis

Compared with NL2SQL without RAG, the advantages of the architecture are reflected in:

1. Business Rule Adaptation: RAG retrieves the September electricity price increase rule, and NL2SQL automatically adjusts the electricity price parameter during generation, avoiding incorrect use of the old electricity price (without RAG, it would generate September electricity price 0.56 CNY/kWh, leading to analysis errors).

2. Complex Query Decomposition: Decompose multi-time dimension comparison needs through subqueries, with clear result logic and easy understanding for users.

### C. FEASIBILITY EVALUATION

The architecture is evaluated from three dimensions: accuracy, efficiency, and compliance. The results are shown in Table 3.

**TABLE 3.** Evaluation Results

| Evaluation Dimension | Metric | Architecture Performance | Traditional Method Performance | Improvement Rate |
|---|---|---|---|---|
| Accuracy | SQL Generation Accuracy | 92% | 65% (manual query) | 41.5% |
| Efficiency | End-to-End Response Time | $\leq$1 minute | 5-30 minutes | 83.3%-96.7% |
| Compliance | Unauthorized Query Interception Rate | 100% | 80% (manual review) | 25% |

The baseline (traditional manual query) is chosen because it is the current industry practice in power marketing; direct comparison with general-purpose SOTA models is impractical due to lack of domain adaptation. The evaluation is based on 200 test samples (100 simple queries + 100 complex queries), covering multi-table associations, nested queries, and ambiguous inputs. The results show that the system meets the design goals and has significant advantages over traditional manual operations.

## V. DISCUSSION

### A. KEY FINDINGS

The integration of RAG and NL2SQL is of great significance. In terms of technical integration value, it can effectively solve the pain points of knowledge accuracy and query automation of power marketing agents. Through the schema and business rules supplemented by RAG, the NL2SQL generation accuracy is improved by 27% (from 65% to 92%); In terms of domain adaptation, the core of the power marketing scenario is knowledge base construction and term mapping. Structured schema documents and business rule tables are the foundation for the architecture to land, and the term mapping table can solve more than 80% of term ambiguity problems; In terms of engineering feasibility, the architecture is built based on open-source tools (Milvus, T5 model), without the need for customized hardware, low deployment cost, and can be directly integrated into the existing "Online State Grid" APP with strong compatibility.

### B. LIMITATIONS

The architecture has shortcomings in data dependence and practicality. Although it does not require large-scale experimental data, the construction of the knowledge base still requires manual sorting of schema and business rules. When the power marketing system is updated frequently (such as adding new business tables), the maintenance cost of the knowledge base is high; In addition, vector retrieval and model generation take about 30 seconds, and delays may occur during peak periods (such as the electricity bill issuance period from the 1st to 5th of each month), which requires optimizing caching strategies and computing resource allocation.

### C. FUTURE WORK

1. Automatic Knowledge Base Construction: Explore automatic generation of schema knowledge based on database reverse engineering (such as parsing SQL table creation statements), and automatically crawl business rules through web crawlers (such as the State Grid Policy Official Website) to reduce manual maintenance costs.

2. Model Optimization: Introduce Reinforcement Learning (RL) to optimize NL2SQL generation, with SQL execution success rate and result accuracy as reward functions to improve complex query processing capabilities.

3. Enhanced Privacy Protection: Combine Federated Learning (FL) technology to realize data does not move, models move among multi-regional power companies, which not only shares model capabilities but also protects user privacy data.

4. Multimodal Interaction Expansion: Support users to input queries through images (such as electricity bill photos), extract key information combined with OCR technology, and further reduce the interaction threshold.

## VI. CONCLUSION

Aiming at the pain points of ambiguous natural language interaction and manual reliance in database queries in power marketing systems, and recognizing similar data retrieval challenges in large industrial sectors like the industry, this paper proposes an agent architecture integrating

RAG and NL2SQL. Through the closed loop of intention understanding-knowledge retrieval-query generation-result feedback, it realizes end-to-end automation from users' natural language to database queries. Case studies show that the architecture can improve the SQL generation accuracy to 92%, shorten the response time to within 1 minute, and achieve an unauthorized query interception rate of 100%, which is significantly better than traditional manual query methods. This research applies RAG to the domain knowledge supplement of NL2SQL in power marketing, solving the problems of term ambiguity and business rule adaptation. The need for integrating such specialized knowledge is a universal challenge, particularly when adapting power solutions for clients in complex industrial verticals like manufacturing. The proposed hybrid retrieval and term mapping strategy provides a reusable technical framework for intelligent query systems in other fields (such as finance and medical care). In the future, through automatic knowledge base construction and model optimization, the architecture can further improve complex query capabilities and real-time performance, providing stronger technical support for the digital transformation of power marketing.

## AUTHOR CONTRIBUTIONS

Conceptualization –Long Peng, Wenchong Guo, Zhixiong Shen and Wobin Lin; methodology – Long Peng, Wenchong Guo, Zhixiong Shen and Wobin Lin; investigation – Long Peng, Wenchong Guo, Zhixiong Shen and Wobin Lin; writing-original draft preparation – Long Peng, Zhixiong Shen and Wobin Lin. All authors have read and agreed to the published version of the manuscript.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## FUNDING

This research was supported by, Science and Technology Project of Guangdong Power Grid Co., Ltd. Research on Key Technologies and Demonstration Application of Marketing Business Knowledge Agents Based on Large Models and Knowledge Graphs (No.030000KC24030042).

## ACKNOWLEDGEMENTS

Not applicable.

## REFERENCES

[1] M. A. Khan, A. M. Saleh, M. Waseem, and I. A. Sajjad, "Artificial intelligence enabled demand response: Prospects and challenges in smart grid environment," IEEE Access, vol. 11, pp. 1477-1505, 2022.

[2] K. Ba, "Research on Informatization Construction Methods of Power Marketing under the Background of Smart Grid," Technology Wind, vol. (31), pp. 44-46, 2025, doi: 10.19392/j.cnki.1671-7341.202531015.

[3] F. Jia, X. Li, T. Liu, X. Du, and Q. Cao, "Research on Intelligent Operation and Maintenance Optimization of Power Marketing Based on Digital Twin Technology," Electrical Engineering, vol. (17), pp. 1-3, 2025, doi: 10.19768/j.cnki.dgjs.2025.17.001.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020.

[5] S. Zeng, J. Zhang, P. He, Y. Liu, Y. Xing, H. Xu, et al., "The good and the bad: Exploring privacy issues in retrievalaugmented generation (rag)," Findings of the Association for Computational Linguistics: ACL 2024. Bangkok, Thailand: Association for Computational Linguistics; 2024. p. 4505-4524, doi: 10.18653/v1/2024.findings-acl.267.

[6] H. Li, Y. Su, D. Cai, Y. Wang, and L. Liu, "A survey on retrieval-augmented text generation," arXiv preprint arXiv:2202.01110, 2022.

[7] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, et al., "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," arXiv preprint arXiv:1809.08887, 2018.

[8] T. J. Revanth, K. V. Sai, R. Ramya, R. Chava, V. Sushma, and B. S. Ramya, "Nl2sql: Natural language to sql query translator," Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 2. Singapore: Springer Singapore; 2021.p. 267-278.

[9] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking retrieval-augmented generation for medicine," Findings of the Association for Computational Linguistics ACL 2024. Bangkok, Thailand: Association for Computational Linguistics; 2024. p. 6233-6251.

[10] A. Onan and E. D. Dursun, "Benchmarking retrieval augmented generation in quantitative finance," International Conference on Intelligent and Fuzzy Systems. Cham, Switzerland: Springer Nature Switzerland; 2024.p. 64-74.

[11] L. Li, R. Shi, X. Guo, and H. Jiang, "Power Equipment Defect Diagnosis Integrating Large Model and Graph Neural Network," Journal of Computer Science and Technology, vol. 18, no. 10, pp. 2643-2655, 2024.

[12] C. Li, L. Zhang, and H. Shi, "NL2SQL Implementation Method for Cultural Relic and Art Auction Data Based on RAG[J/OL]," Journal of Computer Applications. 1-9 [2025-11-12], [Online]. Available: https://link.cnki.net/urlid/51.1307.TP.20250429.1518.004.

[13] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan, "ATHENA: An ontology-driven system for natural language querying over relational data stores," Proceedings of the VLDB Endowment, vol. 9, no. 12, pp. 1209-1220, 2016.

[14] A. Marginean, "Question answering over biomedical linked data with grammatical framework," Semantic Web, vol. 8, no. 4, pp. 565-580, 2017.

[15] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating structured queries from natural language using reinforcement learning (2017)," arXiv preprint arXiv:1709.00103.

[16] X. Liu, Y. An, J. Lv, Z. Huimin, Z. Yumeng, L. Min, et al., "Research on BERT-based Text2SQL Multi-task Learning," Proceedings of the 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA); 29-31 January 2023; Shenyang, China. New York, NY, USA: IEEE; 2023. p. 864-868.

[17] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," Proceedings of the International Conference on Machine Learning; 13-18 July 2020; Online. PMLR; 2020. p. 3929-3938.

[18] D. Lee, "Clause-wise and recursive decoding for complex and cross-domain text-to-SQL generation," arXiv preprint arXiv:1904.08835, 2019.

[19] J. Xu, Y. Jiang, F. Lou, Q. Chen, and Y. Zhang, "Research on natural language understanding model for power marketing strategies," Proceedings of the Fourth International Conference on Mechanical, Electronics, and Electrical and Automation Control (METMS 2024); 26-28 January 2024; Xi'an, China. Bellingham, WA, USA: SPIE; 2024. Volume 13163. p. 1271-1278, doi: 10.1117/12.3030627.

[20] Z. Zhang, B. Wang, J. Zhao, X. Hu, Y. Zheng, and C. Wang, "Realizing Intelligent Interaction of Power Data Based on NL2SQL," Power System Technology, vol. 46, no. 07, pp. 2564-2571, 2022, doi: 10.13335/j.1000-3673.pst.2021.1311.