EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Intent Recognition Enhanced by RAG and Knowledge Graphs for Regulation Execution and O&M Support in Power Engineering Secondary Systems

Qiangchao Xu[1], Kairu Chen[1,*], Zhaolun Deng[1], Jinhua Wu[2], Dan Lin[1], Zhaolin Shi[1]

[1] Guangzhou Power Supply Bureau of Guangdong Power Grid Co., Ltd., Guangzhou, 510000, China
[2] Wuhan Yingrui Electric Power Technology Co., Ltd., Wuhan, 430000, China

Abstract

INTRODUCTION: Secondary systems in power engineering involve complex regulatory documents, operational procedures, dispatching requirements, and maintenance knowledge. Natural-language queries with implicit domain semantics create difficulties for conventional intent recognition in regulation matching and operation-support scenarios.
OBJECTIVES: This study aims to enhance regulatory semantic understanding and intent-recognition accuracy, thereby supporting dispatching, maintenance, regulation execution, and operation and maintenance of power engineering secondary systems.
METHODS: A knowledge-enhanced intent-recognition method combining Retrieval-Augmented Generation (RAG) and Knowledge Graphs (KGs) is proposed. A seven-label regulatory corpus is constructed, and structured semantic triples together with retrieved regulatory contexts are used as composite inputs.
RESULTS: Experimental results show that the proposed method outperforms traditional techniques in terms of Accuracy and Macro-F1 score, improving the recognition of regulation-related, operation-guidance, maintenance-related, and procedure-support intents.
CONCLUSION: The proposed approach strengthens the semantic alignment between technical queries and regulatory knowledge, providing an effective foundation for dispatching support, maintenance guidance, regulation execution, and intelligent operation and maintenance in the power industry.

Keywords: secondary systems in power engineering; intent recognition; knowledge graph; retrieval-augmented generation; dispatching support; operation and maintenance support

Received on 01 April 2026, accepted on 03 May 2026, published on 12 August 2026

doi: 10.4108/ew.14131

1. Introduction

The secondary systems domain within power systems includes relay protection, substation automation, communications, and safety automatic devices, acting as a vital backbone for the secure and stable operation of the grid. But besides being large in number and highly complicated in structure, regulatory documents related to this sphere are frequently inconsistent with one another when it comes to various standards. Such contradictions cause great difficulties to practitioners working hard to master them

*Corresponding author. Kairu Chen, Email: 16676676716@163.com

properly [1][2]. In routine operation, there is always an incidence where technical personnel pose questions using natural language like "How often does this equipment need servicing?" or "Under what regulation does this procedure operate?" Nevertheless, the wide range of variation within such questions renders it difficult to translate these to appropriate regulatory measures [3][4].

Detecting intention has emerged as an important activity in the domain of NLP, receiving considerable focus from researchers in various domains such as open-domain conversation, intelligent customer services, and question answering [5]. Traditional methodologies can be generally classified into two distinct categories: statistical approaches based on rules or features, such as matching keywords or TF-IDF in combination with SVM/Naive Bayes, and semantic modeling techniques driven by deep learning, including CNNs, Transformers, and standard LSTMs [6][7]. Though these methods prove very successful in other areas, there are certain drawbacks associated with their implementation in more specific fields, particularly in second-level systems. The issue gets further complicated during the early stages of training in constructing a deep system because of the small and specialized nature of datasets used. Language used in regulations is very rigid and repetitive; therefore, it is not possible to generalize semantics. Most importantly, the lack of knowledge about structure and domain makes systems misinterpret or ignore rare intentions [8].

To address such issues, the use of retrieval-augmented generation (RAG), as well as knowledge graph (KG) systems, exhibits significant potential in intent recognition for professional settings. The retrieval-augmented generation method bridges the knowledge gap by incorporating related information from external sources along with the user inputs to enhance context understanding [9]. Knowledge graphs model domain knowledge using entities and the relationships between them, thus enhancing semantic constrains and knowledge inference [10]. It has been observed in prior works that using RAG, KG, or both together can enhance the effectiveness and reliability of text understanding in areas such as law, medicine, and education [11].

This paper centers on the semantic comprehension of regulatory documents and the identification of intent within secondary systems. A novel intent recognition approach is introduced, which combines retrieval-augmented generation (RAG) with semantic completion based on knowledge graphs (KG). The proposed method is intended to improve clause matching accuracy and support the interpretation of semantically complex queries. Its main contributions are summarized in three aspects: (1) the construction of a domain-specific regulatory corpus annotated with seven categories of operational intent; (2) the development and validation of an RAG-KG fusion strategy through systematic experiments; and (3) an extensive empirical analysis from multiple perspectives, including baseline comparisons, fusion-model evaluation, and ablation studies, to clarify how each module contributes to accuracy and Macro-F1.

The remainder of the paper is organized as follows. Section 2 reviews related studies on intent recognition, RAG (Retrieval-Augmented Generation), and knowledge graphs. Section 3 presents the proposed method, including data modeling, the intent recognition model, and the RAG-KG integration strategy. Section 4 describes the experimental setup and reports the results. Section 5 concludes the paper and discusses possible directions for future work.

2. Related Work

2.1. Intent Recognition Techniques

Intent recognition is a core task in NLP. It maps natural-language user queries to predefined intent categories. From a technical perspective, given an input utterance in which x_i represents the i-th token, the goal is to construct a mapping function:

$$f: X \to Y, \hat{y} = f(x) \quad (1)$$

Here, $\hat{y}$ denotes the predicted intent category. Conventional approaches first perform feature extraction and then train a classifier, where cross-entropy loss is used to minimize the difference between the predicted labels and the ground-truth labels:

$$L(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} 1(y_i = k)\log p(y_i = k \mid x_i; \theta) \quad (2)$$

where N is the number of samples, θ denotes the model parameters, and $p(y_i = k \mid x_i; \theta)$ is the probability predicted by the model that sample i belongs to class k.

From a technical perspective, intent recognition has developed through three main stages:

(1) Rule-based or statistical feature-based methods: Early studies relied on manually designed keyword rules or shallow textual features, such as TF-IDF and n-grams, together with conventional classifiers, including SVM, Logistic Regression, and Naive Bayes [12]. These methods are computationally efficient, but their robustness is limited when paraphrases or synonymous expressions are involved.

(2) Semantic modeling based on deep learning: As neural-network methods became more widely adopted, researchers began using CNNs and RNNs/LSTMs to encode utterances into vector representations, thereby capturing semantic patterns at both the local and global levels [13]. A common formulation is given below:

$$h = \text{Encoder}(x), \hat{y} = \text{softmax}(Wh + b) \quad (3)$$

where Encoder$(\cdot)$ may be a CNN, LSTM, or bidirectional LSTM; W and b are trainable parameters; and $\hat{y}$ denotes the final predicted intent distribution. Compared with traditional methods, these architectures are able to learn contextual semantic features directly and generally provide more robust intent classification.

(3) Pretrained language modeling methods: In recent years, Transformer-based architectures, including BERT and RoBERTa, have been widely adopted in intent recognition [14]. The basic idea is to first pretrain context-sensitive token and sentence representations using large-scale corpora, and then adapt these representations to downstream tasks:

$$h = \text{BERT}(x), \hat{y} = \text{softmax}(W h_{[CLS]} + b) \quad ()$$

where $h_{[CLS]}$ denotes the sentence-level representation. Pretrained models demonstrate stronger generalization ability in intent recognition, and they are particularly advantageous in low-resource scenarios compared with traditional approaches.

In knowledge-heavy fields like power secondary systems, intent recognition runs into two big challenges. First, there's not enough domain-specific data, and class distributions are imbalanced-this holds back deep model generalization. Second, current methods don't model regulatory clause structures well or encode explicit domain information, which makes it hard to accurately recognize rare intents. These problems hurt real-world reliability, which is why we're looking to combine RAG and KG techniques.

2.2 Retrieval-Augmented Generation (RAG)

Traditional NLP generative models, such as Seq2Seq, BART, or GPT, are widely used for language modeling. A common limitation, however, is that they may produce unsupported content in knowledge-intensive tasks, which can reduce factual reliability. Retrieval-Augmented Generation (RAG) addresses this issue by combining external retrieval and a generation module, enhancing factual accuracy while also improving domain adaptability [15].

RAG begins by retrieving documents relevant to the query from a knowledge base before producing output. These retrieved documents are then combined with the input query before being passed to the generator. The entire framework includes two main stages:

(1) The Retrieval Stage: When a query is provided, the retriever first converts it into a vector by applying techniques such as DPR, BM25, or FAISS, among others. Next, from a pre-encoded set of domain-specific documents (each represented as a vector), it selects the most relevant documents according to similarity scores:

$$D_q = \text{TopK}(\text{sim}(q, d_i)), d_i \in D \quad (5)$$

where $\text{sim}(\cdot,\cdot)$ typically denotes cosine similarity or inner product.

(2) Generation phase: The query is merged with the documents that were obtained and then input into a generative model to generate the final response. The aim here is to maximize the likelihood of producing an accurate output:

$$P(a \mid q) = \sum_{d_i \in D_q} P(a \mid q, d_i) \cdot P(d_i \mid q) \quad (6)$$

where $P(a \mid q, d_i)$is modeled by the generator (e.g., BART, T5), and $P(d_i \mid q)$denotes the document weight estimated by the retriever. This joint formulation gives the system a “read + write” capability: it can produce responses grounded in the retrieved supporting information.

Compared with end-to-end generative models, RAG is more effective at incorporating domain-specific knowledge, which makes it suitable for knowledge-intensive applications such as regulatory document processing and medical-text analysis. It improves factual reliability in applications such as question answering, interactive text generation, and summarization [16]. Recent refinements, including sparse-dense hybrid retrieval, domain-specific vocabulary

adaptation, and sliding context windows, have further improved its efficiency, practicality, and controllability [17]. In regulation-oriented intent recognition, there is often a noticeable gap in phrasing and style between user queries and regulatory clauses. This means that the retriever usually needs additional design choices, such as context filtering or restrictions on document types. To reduce noise, practitioners often select small candidate windows that contain high-confidence clauses. This strategy improves generator accuracy and also makes the decision process easier to interpret.

Overall, RAG provides a systematic and knowledge-enhanced framework for intent recognition. It is especially useful when domain-specific data are limited, and the contextual structure is complex.

2.3. Knowledge Graphs

A knowledge graph (KG) is essentially a graphical representation that illustrates entities and their semantic connections. It's widely used in semantic search, suggestion systems, and NLP question-answering. The primary objective is to transform unstructured content into a semantically rich, layered network by employing structured modeling techniques. This process enhances the comprehension of domain-specific knowledge [18].

In formal terms, a Knowledge Graph (KG) is composed of triples, where h denotes the head entity, t the tail entity, and r the relation between them. Each triple can be viewed as a directed edge. Constructing a knowledge graph usually involves three main stages: named entity recognition (NER), relation extraction (RE), and graph construction.

To improve the usefulness of KGs for downstream applications, graph structure embeddings are often learned. Knowledge-graph embedding maps entities and relations into low-dimensional vectors while maintaining the semantic relations encoded in the graph. For example, TransE assumes that valid triples generally satisfy the following condition:

$$\mathrm{h} + \mathrm{r} \approx \mathrm{t} \tag{7}$$

where $\mathrm{h}, \mathrm{r}, \mathrm{t} \in \mathbb{R}^{\mathrm{d}}$ are the vector embeddings of the head entity, relation, and tail entity, respectively. The training objective is to minimize the distance for positive triples while maximizing it for negative samples:

$$\mathrm{L}_{\mathrm{TransE}} = \sum_{(\mathrm{h},\mathrm{r},\mathrm{t})\in\mathrm{K}} \sum_{(\mathrm{h}',\mathrm{r},\mathrm{t}')\in\mathrm{K}'} [\gamma + \|\mathrm{h} + \mathrm{r} - \mathrm{t}\|_2^2 - \|\mathrm{h}' + \mathrm{r} - \mathrm{t}'\|_2^2]_+ \tag{8}$$

where K' denotes the set of negative samples, γ is a margin hyperparameter, and $[\mathrm{x}]_+ = \max(0, \mathrm{x})$ denotes the hinge loss [19]. This formulation retains semantic information in the graph and maintains good computational efficiency and transferability, which makes it suitable for lightweight downstream applications, including intent recognition.

When it comes to domain-specific modeling-like in power secondary systems-KGs offer structural benefits. Regulatory documents have hierarchical clauses, related terms, and device-function chains, which naturally create graph path semantics. Building a KG also helps abstract processes across various documents and scenarios, bringing together scattered knowledge.

With the introduction of graph neural networks (GNNs), knowledge graphs can be further leveraged for structural enhancement. For example, a graph convolutional network (GCN) updates a node's representation by aggregating information from its neighbors. For any node v, the update rule at layer $\mathrm{l}+1$ can be written as:

$$\mathrm{h}_{\mathrm{v}}^{(\mathrm{l}+1)} = \sigma\left(\sum_{\mathrm{u}\in\mathrm{N}(\mathrm{v})} \frac{1}{\mathrm{c}_{\mathrm{vu}}} \mathrm{W}^{(\mathrm{l})} \mathrm{h}_{\mathrm{u}}^{(\mathrm{l})}\right) \tag{9}$$

where $\mathrm{N}(\mathrm{v})$ denotes the neighbor set of v, $\mathrm{W}^{(\mathrm{l})}$ is the trainable weight matrix at layer l, c_{vu} is a normalization factor, and $\sigma(\cdot)$ is an activation function (e.g., ReLU). Through mechanisms like GCN, node representations can integrate multi-hop semantic path information and achieve deeper graph-level semantic modeling [20].

All in all, KGs offer structured modeling of semantics, and when combined with embedding learning and GNNs, they do a great job supporting complex query understanding. In this study, KG construction and structural embedded elements provide key context and relation inputs for the intent recognition model.

2.4. Fusion Methods for Knowledge Graphs and Semantic Representations

In natural language understanding (NLU) tasks, features based solely on text often face challenges related to semantic variability and ambiguity. This is particularly evident in regulation-style intent recognition, where user expressions

frequently deviate from the formal structure of regulatory clauses. Thus, integrating structured knowledge from knowledge graphs (KG) with semantics of language is crucial for enhancing comprehension [21].

The essence of fusion lies in the joint modeling of textual representations (such as BERT-encoded sentence vectors) and the structural embeddings of KG entities/relations. A commonly used method involves concatenating vectors to create a joint feature representation:

$$z = [e_{text}; e_{kg}] \quad (10)$$

where e_{text} denotes the sentence vector obtained by encoding the input text with a language model (such as BERT), and e_{kg} denotes the embedding representation retrieved from the knowledge graph (e.g., embeddings of related entities and path information). The concatenated vector z is then used as input to a classifier. This straightforward strategy can substantially improve a model's ability to capture structural, regulation-specific semantics.

An alternative fusion path uses an attention mechanism to dynamically modulate graph information with respect to the textual semantics. Suppose the KG yields a set of candidate entity embeddings $\{e_1, e_2, \dots, e_n\}$. We can define an attention distribution based on semantic matching:

$$\alpha_i = \frac{\exp(q^T W_a e_i)}{\sum_{j=1}^{n} \exp(q^T W_a e_j)} \quad (11)$$

where q denotes the text vector and W_a is a learnable parameter matrix; α_i is the attention weight of the i-th KG entity with respect to the text. The final fused representation is a weighted sum:

$$z = q + \sum_{i=1}^{n} \alpha_i e_i \quad (12)$$

This way, semantically relevant KG information gets injected into textual representations, which helps with entity confusion as well as clause matching [22].

To capture the structural relationships in regulatory clauses, path-aware fusion strategies are employed. Multi-hop KG query paths are treated as auxiliary semantic paths, encoded (using methods like LSTM or PathGCN) into path vectors, and then modeled together with text embeddings:

$$z = \mathrm{MLP}([q; p_{max}]) \quad (13)$$

where p_{max} denotes the path vector selected by maximum attention score. This approach captures semantic propagation chains in the graph and helps to uncover structural links between complex intents such as "regulation-content query → reporting-procedure confirmation → operation-norm request" [23].

To sum up, KG-text fusion approaches enhance intent recognition by achieving a balance between graph structure and linguistic context. For power secondary systems, which are characterized by stringent regulations and intricate terminology, structural semantics serves to compensate for the shortcomings of plain text. It offers robust knowledge support for downstream tasks.

3. Methodology

3.1. Overall Framework

This paper tackles the challenges posed by secondary system regulatory texts, including domain-specific terminology, intricate expressions, and implicit structural patterns. It proposes an intent-recognition approach that combines Retrieval-Augmented Generation (RAG) with a lightweight knowledge graph (KG)-based enhancement of structural semantics. Starting with the construction of structured corpora, it incorporates external retrieval contexts and graph-structured semantic completion to improve the comprehension and classification of regulatory inquiries. The framework (see Figure 1) is composed of four core components: data modeling, semantic recognition, knowledge improvement, and fusion inference.

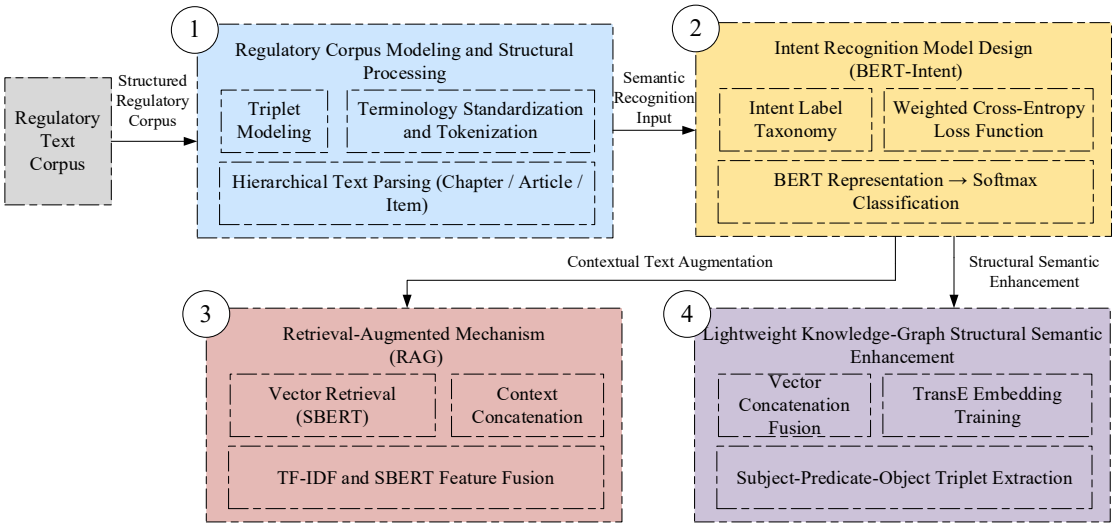

Figure 1. Overall framework of the method

First, we systematically process the regulatory corpus to extract clause-level content from the raw texts, thereby establishing a standardized query input space. We create a taxonomy for domain-specific intent labels and build a multi-class classifier to predict intents. The RAG mechanism integrates contextual details from relevant clauses, providing knowledge that is grounded in regulatory frameworks. At the same time, we utilize large language models in order to extract subject-predicate-object triplets from regulatory texts. These triplets are then used to build a lightweight knowledge graph, which is integrated with queries to capture structural semantics that are difficult to convey through plain text alone. This multimodal information is merged into a unified input representation, which supports reliable identification of semantically complex regulation-related queries. The system offers an efficient solution for regulation-specific NLU applications by retaining engineering efficacy and broadening semantic scope.

3.2. Corpus Modeling and Structural Processing of Power Regulations

Secondary systems in power engineering are responsible for grid safety, regulation, and automated protection functions. Their regulatory materials span multiple technical domains, such as relay protection, substation automation, communication networks, and safety devices. The corpus is composed of technical, operational, and procedural documentation. Although these texts exhibit tightly organized hierarchical structures, formalized wording, and robust internal organization, they still present NLP difficulties, including substantial semantic abstraction and ambiguous entity–scenario relationships. For this reason, raw regulatory texts need to be transformed into structured and model-friendly corpora.

First, clause identifiers, clause bodies, and auxiliary notes are extracted directly from source documents by combining automated parsing procedures with manual annotation. Each fragment is arranged in a three-level hierarchy consisting of Chapter, Article, and Item, with each level assigned a unique path, such as “Chapter 3, Article 5, Item 2.” This hierarchical design enables text localization and supplies prior structural information for constructing hierarchical entity models within the regulatory knowledge graph. Moreover, regulatory provisions are often associated with particular operational scenarios—such as operational procedures, fault response, or remote control—so domain-expert knowledge is incorporated during structural processing so that semantic labels can be assigned. Each regulatory fragment is linked to a triple composed of structural position, textual content, and scenario attributes.

$$S_i = \{(id_i, c_i, t_i)\} \quad (14)$$

where id_i denotes the clause identifier, c_i denotes the original clause text, and t_i denotes the corresponding domain

task semantic label. This representation preserves the original regulatory semantics while embedding task-oriented knowledge, providing a structurally clear and label-controlled training corpus for supervised intent-recognition modeling.

To boost cross-fragment connectivity and feature consistency, we apply unified text standardization during preprocessing: symbol cleaning, unit normalization, terminology standardization (like "protection device" vs. "protective equipment"), and domain terminology alignment. Texts are converted into tokenized sequences to form input feature matrices:

$$X_i = \text{Tokenizer}(c_i) = [w_1, w_2, \ldots, w_n] \quad (15)$$

where $\{w_j\}$ denotes the j-th token of the regulatory text. These structurally organized formats are the primary source of semantic encoding, vector search queries, and intentionality recognition algorithms.

Structural Processing Algorithm of the Regulatory Database includes procedures such as procedure detection, hierarchical classification and enumeration, generation of semantic tags, and converting the texts into standard vectors. This approach ensures the machine readability and semantic accuracy of power regulation documents. Figure 2 presents the key steps together with the corresponding transformation paths.

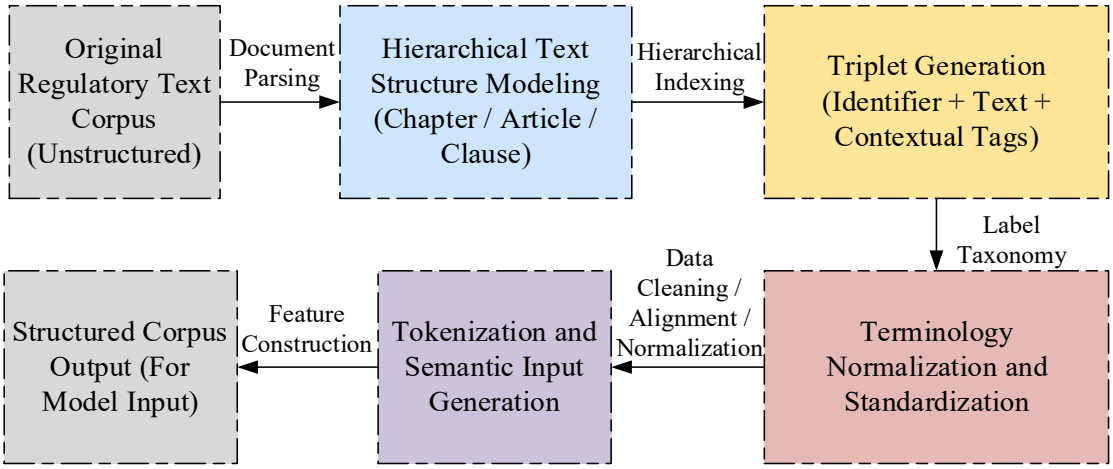

Figure 2. Structural modeling workflow for regulatory corpus

3.3. Intent Label Taxonomy and Recognition Model Design

Model Design. In regulation-oriented question answering for power secondary systems, the natural-language queries raised by technical staff frequently involve several implicit semantic intents. Accurate intent recognition supports effective matching between regulatory resources and the information needs expressed in user questions, thereby helping maintain reliable QA responses. A taxonomy of seven intent categories has been established to reflect typical usage settings as well as recurring query expressions. These categories cover requests related to regulation content, operational guidance, fault causes, maintenance cycles, reporting procedures, training, and device functionality. This taxonomy captures the main needs of regulatory inquiries and provides a basis for accurate intent classification.

To realize automatic recognition of the seven intent types, we formulate this task as a single-label seven-class classification problem. Let the input query be Q_i and its ground-truth intent label be $y_i \in \{1,2,\ldots,7\}$. We build the intent-recognition model on a BERT-based architecture and perform joint semantic encoding with classification. The model architecture can be described as follows: the query Q_i is first processed by the BERT encoder to obtain a semantic vector representation $h_i \in \mathbb{R}^d$; then a fully connected layer maps $h_i$into the output label space, after which a Softmax layer yields the predicted class probability distribution $\hat{p}_i$:

$$h_i = \text{BERT}(Q_i), \hat{p}_i = \text{Softmax}(Wh_i + b) \quad (16)$$

where $W \in \mathbb{R}^{7\times d}$ denotes the weight matrix, and $b \in \mathbb{R}^7$ denotes the bias term; $\hat{p}_i$ represents the predicted probability distribution produced by the model for the seven classes.

The model is trained by minimizing the cross-entropy between the predicted distribution and the ground-truth labels:

$$L_{CE} = -\sum_{i=1}^{N}\sum_{j=1}^{7} \mathbb{I}(y_i = j)\cdot \log(\hat{p}_{i,j}) \quad (17)$$

where N is the number of training samples, $\mathbb{I}(\cdot)$ is the indicator function, and $\hat{p}_{i,j}$ denotes the probability that sample i is predicted as class j.

Given the pronounced class imbalance in the dataset, we further introduce class-weighted cross-entropy to emphasize minority classes. The optimized loss becomes:

$$L_{WCE} = -\sum_{i=1}^{N}\sum_{j=1}^{7} \alpha_j \cdot \mathbb{I}(y_i = j)\cdot \log(\hat{p}_{i,j}) \quad (18)$$

where the class weight α_j is set proportional to $1/\log(1 + n_j)$, and n_j denotes the number of samples in class j. This weighting design increases the model's sensitivity to minority classes throughout training.

The final BERT-Intent model consists of a BERT representation layer and a fully connected classification layer, enabling strong semantic comprehension and classification. It is used for regulation-query intent recognition while also serving as the backbone input module for RAG retrieval and KG semantic augmentation, providing the regulatory QA system with a foundation for semantic perception.

3.4. Retrieval-Augmented Mechanism Design

To strengthen regulatory semantic understanding, a RAG mechanism is introduced for aligning queries with relevant regulatory knowledge via the retrieval of related clauses from a regulatory knowledge base. This mechanism addresses missing semantic information in short user queries and supplies additional contextual information for downstream intent recognition.

From an implementation perspective, RAG consists of a semantic retriever (Retriever) and a context encoder (Reader). The retriever converts queries into vectorized forms and selects the top-K semantically related clauses from the clause collection to form the retrieved context, after which the encoder combines the query with these clauses and represents the combined text for intent classification.

Let the input query be Q_i and the set of regulatory clauses be $C = \{c_1, c_2, \ldots, c_M\}$. Sentence-BERT is used to encode queries as dense vector embeddings $q_i \in \mathbb{R}^d$, while all clauses in the knowledge base are pre-encoded into vectors $\{c_1, c_2, \ldots, c_M\}$. Vector matching is carried out through cosine similarity to quantify relevance:

$$\mathrm{sim}(q_i, c_j) = \frac{q_i \cdot c_j}{\|q_i\| \cdot \|c_j\|} \quad (19)$$

where $q_i \cdot c_j$ represents the inner product, and $\|\cdot\|$ denotes the L2 norm. Using the resulting similarity scores, the Top-K relevant clauses $\{c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(K)}\}$ are retained to form the augmented representation.

Next, a context-extended input Q_i^+ is constructed by semantically concatenating the initial query with the retrieved clauses. This concatenation operates at the text level through sequentially attaching each retrieved clause after the query:

$$Q_i^+ = Q_i \oplus c_i^{(1)} \oplus c_i^{(2)} \oplus \cdots \oplus c_i^{(K)} \quad (20)$$

where $\oplus$ denotes text-level concatenation. The augmented text Q_i^+ serves as the input to the BERT-Intent model, which extracts a global semantic representation h_i. This representation is then processed by a fully connected layer and a Softmax module to obtain intent class scores, and the classification loss is defined by cross-entropy:

$$L = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \quad (21)$$

where $y_{ij} \in \{0,1\}$ is the ground-truth label for sample i on class j, while $\hat{y}_{ij}$ denotes the probability predicted by the model.

To prevent irrelevant or standardized information from interfering with semantic discrimination, domain filtering is incorporated at the retrieval stage. Clauses are pre-labeled with subdomain tags—such as protection, dispatch, or communications—so retrieval is restricted to clauses from subdomains relevant to the query. To obtain more diverse representations, both sparse (TF-IDF) and dense (SBERT) encodings are generated for the retrieved content. These features are further combined into a unified fused representation:

$$x_i^{\text{fused}} = [x_i^{\text{tfidf}}; x_i^{\text{sbert}}] \quad (22)$$

This combined representation works well with traditional classifiers like SVM or logistic regression.

Figure 3 shows the retrieval-augmented mechanism's pipeline: query vectorization, semantic matching, clause concatenation, model encoding, and final prediction.

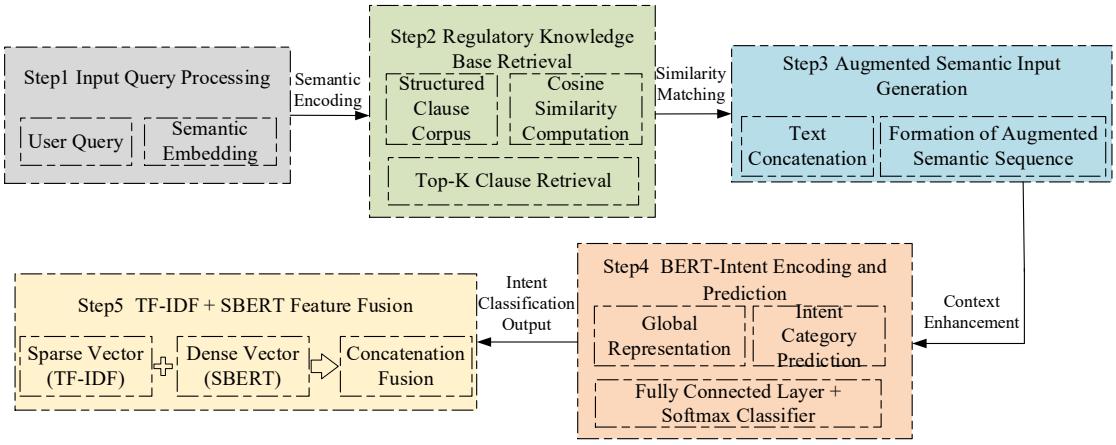

Figure 3. Architecture of the retrieval-augmented mechanism

3.5. Lightweight Knowledge-Graph Construction and Structural Semantic Enhancement Strategy

To further strengthen the intent-recognition model's ability to capture implicit structural semantics in regulatory texts, this study introduces a lightweight knowledge-graph structure as an auxiliary enhancement module. Rather than constructing and storing a complete entity graph in a dedicated graph database, we employ a simplified yet semantically informative KG representation. Triples in the form of $(h, r, t)$are identified from regulatory texts with the assistance of large language models, where h denotes the head entity, r the relation type, and t the tail entity. These triples preserve the structured semantic information embedded in regulatory clauses and represent semantic links among entities in the knowledge graph.

In the course of building the dataset, we apply prompt-based extraction methods to convert both user queries and regulatory clauses into structured triples. Several examples are provided below:

(protection device, has function, self-diagnosis)

(operation ticket, contains, operation steps)

(main transformer, belongs to system, 220 kV grid)

The extracted triples are transformed into vectors through KG embedding. We utilize the TransE translational model, which interprets triples as translations within vector space and optimizes the following objective:

$$L_{KG} = \sum_{(h,r,t)\in G} \|e_h + e_r - e_t\|_2^2 \quad (23)$$

where G denotes the set of valid triples in the graph, and e_h, e_r, e_t are the embedding vectors of the head entity, relation, and tail entity, respectively. This optimization encourages entities and relations to preserve structural consistency in the semantic space, thereby reflecting the logic and hierarchy inherent in the regulations.

The embedded triple information is then fused with the original query text representation. We employ a semantic concatenation fusion strategy: the query encoding q and the graph-structure representation are concatenated to form a joint representation:

$$x_{fused} = [q; g] \quad (24)$$

where [;] denotes vector concatenation. The fused vector x_{fused} can be directly input to downstream intent classifiers. In practice, the structural-vector component is also optimized through forward propagation during training to improve alignment with the semantic representation.

Importantly, even without building a full standard graph database or doing graph-traversal reasoning, this triple-

centered lightweight modeling still explicitly captures key structural semantics in regulatory documents for supervised enhancement. It balances structural expressiveness and engineering cost, avoiding the costs associated with full-scale KG construction-making it well-suited for power-domain regulation-oriented QA tasks.

Figure 4 outlines the workflow of the lightweight KG structural-semantic enhancement strategy, including triple construction, vector embedding, and fusion-enhancement steps.

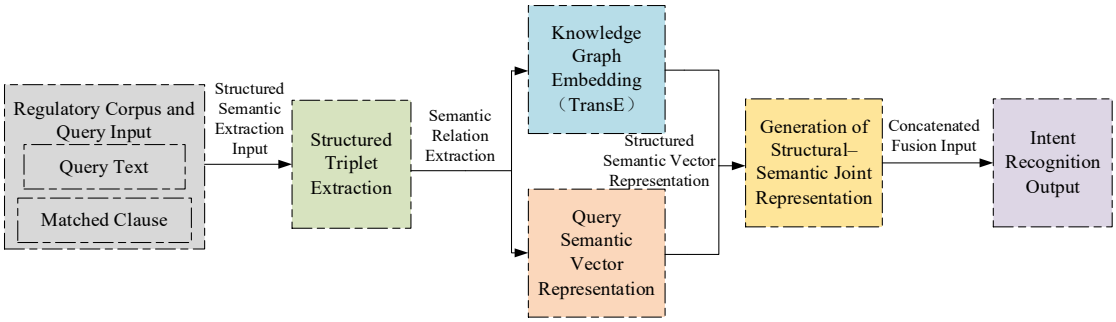

Figure 4. Workflow of the lightweight knowledge-graph semantic-enhancement strategy

4. Experiments

4.1. Experimental Settings

4.1.1. Experiment Overview

To evaluate the proposed RAG-KG fused intent-recognition method for regulatory texts, three comparative experiments were conducted. These experiments examined (1) conventional baseline models, (2) the effectiveness of the fusion strategy, and (3) the contribution of each individual component. This experimental setup analyzes the performance differences among models on power-regulation semantic understanding tasks.

Experiment 1 establishes the benchmark for conventional intent recognition. It uses original user queries together with TF-IDF, Sentence-BERT embeddings, and classifiers such as SVM, Logistic Regression, and Naive Bayes. Experiment 2 extends the baseline features by incorporating clause contexts generated through RAG (Retrieval-Augmented Generation) together with KG-based triples extracted from regulatory texts. By using the query, RAG-derived contexts, and KG triples as multi-source representations, the framework provides a more complete characterization of regulation queries with complex semantic structures. Experiment 3 performs an ablation study on the fusion components under four configurations: (1) query only; (2) query + RAG; (3) query + KG; and (4) query + RAG + KG. The resulting comparisons are used to assess the respective effects of retrieval enhancement and structural semantics. Table 1 summarizes the experimental settings and objectives.

Table 1. Overview of experimental tasks and objectives

Experiment ID	Experiment name	Objective
Experiment 1	Baseline experiment for traditional intent recognition	Provide performance reference for traditional methods and evaluate raw-text features
Experiment 2	Intent Recognition with RAG and Knowledge-Graph Fusion	Validate performance gains from fusing retrieval context and structural-semantic features
Experiment 3	Ablation study	Analyze independent contributions and complementarity of RAG and KG modules

4.1.2. Dataset Description

Our dataset is constructed from regulatory documents in the power engineering secondary-system domain, including procedures, technical specifications, guidance documents, and operating standards related to relay protection, substation automation, communication systems, and safety automatic devices. These materials were organized into an intent-recognition corpus with triplet-based entries: “user_question - intent_label - supporting_knowledge”. Each sample includes a genuine query, its associated intent category, and the applicable regulatory provision, all of which form the

labeled data utilized for training the model. Several examples are provided below:

{
"question": "What should operating personnel do when a relay protection device malfunctions?",
"intent": "operation-guideline request",
"supporting_knowledge": "The operations and maintenance department requires operators to correctly switch protection devices on and off; when a device malfunctions or operates, they should accurately record the operation signals and promptly report to the dispatch authority."
},
{
"question": "Under what circumstances is it necessary to retrofit existing relay protection devices?",
"intent": "regulation-content query",
"supporting_knowledge": "For π-configuration reconstructions of lines at 110 kV and above, if the original protection cannot cooperate with newly installed devices, or if a device has been in service for more than eight years, it should be retrofitted."
}

The dataset consists of 734 annotated queries covering seven intent categories. As reported in Table 2, the distribution is highly uneven: the two dominant categories, "operation-guideline request" and "regulation-content query," account for 76% of all samples, whereas categories including "training/learning related," "maintenance-cycle query," and "fault-cause inquiry" all contain fewer than 20 instances and together account for only about 6% overall. This degree of class imbalance means that the largest category contains 27 times the number of samples in the smallest category. Such a skewed distribution makes the model harder to optimize and its performance harder to assess reliably, which in turn contributes to weaker classification results for minority classes.

Table 2. Class distribution of the intent-recognition dataset (power domain)

Class	Operation-guideline request	Regulation-content query	Reporting-procedure confirmation	Device-function question	Training/learning related	Maintenance-cycle query	Fault-cause inquiry	Total
Sample count	358	203	71	59	16	14	13	734
Proportion (%)	48.8	27.7	9.7	8.0	2.2	1.9	1.8	100

To supply richer semantic information for the model and support a more complete understanding of the data, we used large-scale models to generate two forms of auxiliary information: (1) domain-related knowledge triples used as structured knowledge-graph (KG) information, and (2) context-completion texts generated through Retrieval-Augmented Generation (RAG). These resources were then incorporated into the original dataset to construct multimodal text–feature datasets for the experiments.

4.1.3 Experimental Configuration and Assessment Procedure

Each experiment was carried out on a unified local platform running under Windows 11 and Python 3.13. Scikit-learn was used for classifiers and evaluation, transformers for BERT variants, pandas/numpy for data processing, and joblib for model persistence. The hardware configuration included an Intel Core i7-14650HX CPU and an RTX 4070 GPU, supporting medium-scale training and inference workloads.

Model performance was evaluated on the basis of Accuracy, which indicates classification performance, Macro-F1 (better indicates performance in case of class imbalance) and the confusion matrix, which is used to represent classification bias.

Accuracy is defined by the fraction of correctly predicted labels among all samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP denotes true positives, TN denotes true negatives, and FP and FN denote false positives and false negatives, respectively. Accuracy measures the overall correctness of decisions, but in datasets with highly imbalanced class distributions, it may be heavily influenced by the majority classes.

To better capture classification quality for minority classes, Macro-F1 is computed as:

$$\text{Macro-F1} = \frac{1}{N}\sum_{i=1}^{N} \frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \quad (2)$$

where N denotes the number of classes, while P_i and R_i are the precision and recall for class i, defined by:

$$P_i = \frac{TP_i}{TP_i + FP_i}, R_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

Macro-F1 is obtained by averaging the class-level F1 values and therefore reduces the influence of majority classes while emphasizing balanced behavior across intent categories. It is therefore well suited to imbalanced or skewed datasets. The confusion matrix summarizes classification results for different intent categories and highlights where inter-class confusion occurs. Ground-truth labels correspond to rows, whereas predicted categories correspond to columns. By comparing diagonal and off-diagonal elements, possible bias patterns and class-specific blind spots can be identified.

These metrics describe typical accuracy, neighborhood prediction patterns, and class-level results, which helps discover the place the mannequin need to be refined.

4.2. Experiment 1: Baseline Experiment for Traditional Intent Recognition

Experiment 1 evaluates traditional intent-recognition methods on power-domain regulatory tasks using conventional text representations together with classical classifiers for this corpus. This experiment is intended to serve as a baseline for performance comparison, identify constraints in domain-specific corpora such as power

regulations, and provide a reference for assessing the proposed semantic–structural fusion methods.

The baseline experiment uses the original user queries alone as the sole input, without context expansion or knowledge enhancement, so as to measure baseline performance when only limited semantic support is available. The text representation consists of TF-IDF vectors and SBERT embeddings. Three traditional classifiers were evaluated: (1) Support Vector Machines (SVM), a method that is effective for high-dimensional text features and suitable for medium-sized text collections; (2) Logistic Regression, implemented as a linear discriminative classifier that offers interpretability together with fast optimization; and (3) Naive Bayes, a model that captures feature distribution patterns efficiently and is therefore appropriate for preliminary modeling.

The models were trained and validated with an 8:2 train-test partition for fair comparison. Given the class imbalance, both Accuracy (for overall performance) and Macro-F1 (for class-balanced evaluation) are reported, as described in Section 4.1.3. The test results are presented in Table 3.

Table 3. Performance comparison of traditional classifiers in the baseline experiment

Model	Accuracy	Macro-F1
SVM	0.5918	0.2832
Logistic Regression	0.5714	0.2033
Naive Bayes	0.5102	0.1427

The results show that SVM outperforms the other approaches across both evaluation measures and demonstrates stronger generalization under high-dimensional sparse feature settings. Logistic Regression ranks second, with moderate Accuracy and Macro-F1. Although Naive Bayes involves relatively low computational cost, it is still less effective for fine-grained semantic tasks. As a result, it yields the lowest Macro-F1 scores and shows weak discrimination of minority classes. To better understand class-level prediction behavior, confusion matrices were generated for the three models, as shown in Figures 5 through 7. In this visualization, the horizontal axis represents the predicted categories, while the vertical axis indicates the actual (ground-truth) categories. The intensity of shading in each cell reflects the count of predictions.

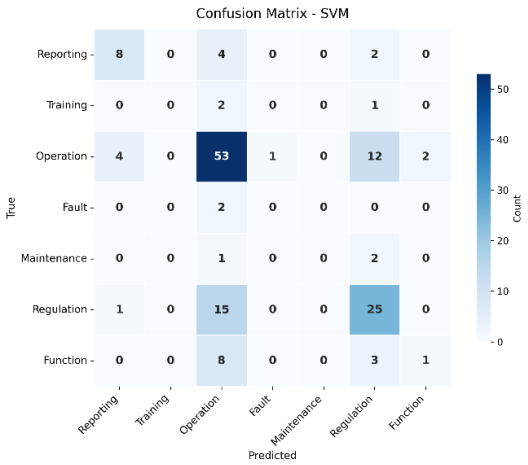

Figure 5. Confusion matrix of the SVM model

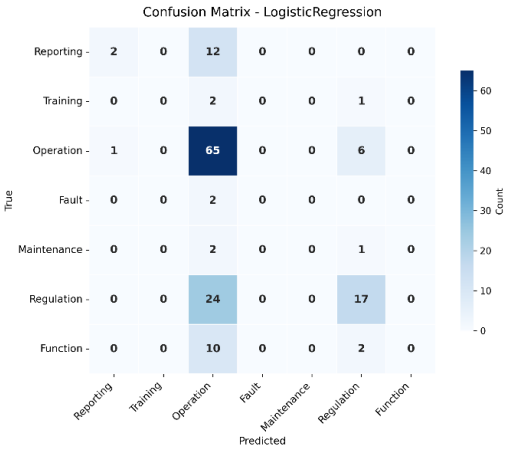

Figure 6. Confusion matrix of the Logistic Regression model

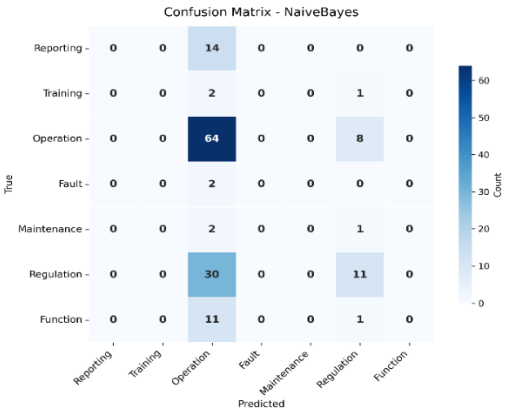

Figure 7. Confusion matrix for Naive Bayes

In the SVM results, the "Operation" class (operation-guideline request) is recognized relatively well, with 53 correct predictions, but it often confuses the "Regulation" class (regulation-content query) with "Operation," making this error 15 times. Classes such as "Training" and "Maintenance" continue to be difficult to recognize, and some receive no predictions at all.

Logistic Regression performs better on the majority "Operation" class, with 65 correct predictions, but shows greater difficulty with minority classes. For example, the term "Reporting" is mapped to the primary class in 12 cases, which increases the possibility of confusion. Naive Bayes shows weak results across all classes, and many minority-class samples are misclassified as dominant categories such as "Operation" or "Regulation," indicating limited semantic separability. Examination of the confusion matrices indicates that these conventional methods remain limited in regulation-domain intent recognition. One contributing factor is that dense semantics, complex structures, and blurred class boundaries make minority classes more difficult to classify. The imbalance in the distribution of classes within the data set makes the algorithm depend on the majority classes while optimizing itself.

The baseline experiment further shows that conventional models have difficulty with domain-specific semantics in the absence of either external support or supplementary contextual information. Although these models are still reasonably effective on majority classes, their overall F1 score remains low and falls short of the precision and coverage requirements of real-world regulatory quality-assurance systems. This is why structural-semantic enhancements, including RAG and knowledge-graph integration, are needed to provide stronger semantic support for classification.

4.3. Experiment 2: Intent Recognition with RAG and Knowledge-Graph Fusion

Experiment 2 further tests the value of feature augmentation for regulation-oriented intent recognition. On the basis of the baseline model in Section 4.2, external domain knowledge is introduced to examine whether it can improve classification under severe class imbalance, particularly for categories with only a small number of samples.

Two additional information sources are incorporated into the original query representation: (1) RAG-derived context from regulatory documents and (2) structured subject-predicate-object triples extracted from a lightweight KG structure. This combined input enriches contextual semantics and broadens domain coverage, helping the classifier identify and separate complex intent categories more effectively. The original query functions as the basis for feature construction. With RAG, we locate semantically related passages from regulatory documents and produce context-completion text through condensation followed by summarization. To express semantic relations more clearly, relevant S-P-O triples obtained from the pre-constructed lightweight power-system knowledge graph are rewritten as structured sentences, which then function as knowledge-enhancement elements.

The fused representation is defined as follows:

Input = original query + retrieval-completion(RAG)+ knowledge-triple expressions(KG)

RAG-generated text offers additional contextual support, whereas KG triples explicitly encode relations among semantic entities. Used jointly, these resources enable the classifier to detect latent intent categories through the joint use of domain knowledge together with supporting document evidence, thereby lessening its dependence on surface-level text. For selecting models, SVM and LR were retained from the baseline setting, and Random Forest (RF) was added to examine the way different architectures process fused features. However, Naive Bayes was left out because it performs poorly in cases where the features are highly dimensional and densely semantically represented, making it unsuitable for fused inputs. The two inputs used during the experiment are: (a) the baseline query input and (b) fused features. The initial question acted as the benchmark. For measuring accuracy and macro F1 score was adopted. The performance variance noted validates the relevance of the feature fusion approach (see Table 4).

Table 4. Performance comparison before and after external-knowledge fusion

Model	Accuracy (baseline)	Macro-F1 (baseline)	Accuracy (fused)	Macro-F1 (fused)	ΔAccuracy	ΔMacro-F1
SVM	0.5918	0.2832	0.6463	0.3929	+0.0545	+0.1097
LogisticRegression	0.5714	0.2033	0.6531	0.4223	+0.0817	+0.2190
NaiveBayes	0.5102	0.1427	N/A	N/A	N/A	N/A
RandomForest	N/A	N/A	0.5102	0.1286	N/A	N/A

A comparison of the results shows that SVM improved from 0.5918 to 0.6463 in Accuracy, while its Macro-F1 score increased by 0.1097. This suggests a notable improvement in performance on long-tail categories. Logistic Regression showed even larger gains: its Accuracy rose from 0.5714 to 0.6531, and its Macro-F1 score increased from 0.2033 to 0.4223, representing a gain of 21 percentage points. This finding suggests that the fusion approach is especially effective in linear-feature learning. In contrast, Random Forest showed weaker performance after fusion (Accuracy = 0.5102, Macro-F1 = 0.1286), indicating susceptibility to feature types and noise, together with weaker structural representation ability than SVM or LR.

The increase in Accuracy and Macro-F1 is not due to mere improvement in fitting to the majority class samples but rather highlights an increased ability to recognize minority classes as a result of the application of feature fusion. This

shows that the use of external context in addition to knowledge bases is more effective.

To examine classification behavior under fused features, confusion matrices were generated for SVM, LR, and RF, including both raw counts and normalized percentages (see Figures 8–13).

The horizontal axis denotes the predicted labels, whereas the vertical axis denotes the ground-truth labels. Each cell reports either the number of samples or the percentage within the corresponding true class. Darker cells indicate higher prediction density, which helps identify bias as well as reveal confusion patterns.

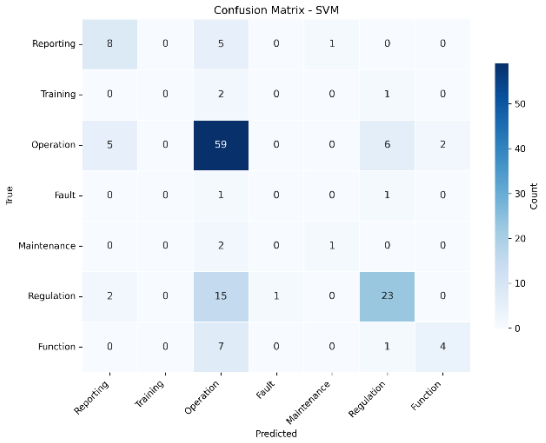

Figure 8. Confusion matrix (absolute counts) of SVM after fusion

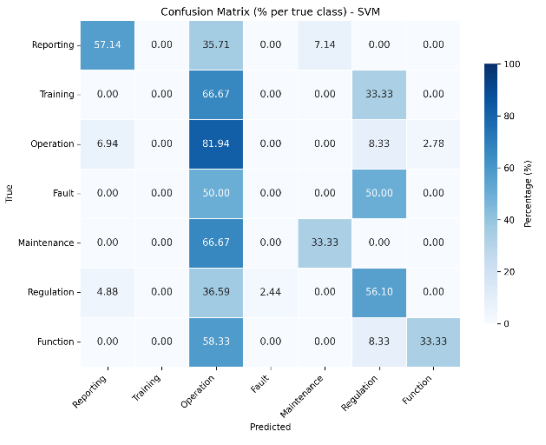

Figure 9. Confusion matrix (percentage, row-normalized) of SVM after fusion

Figure 8 (count-based) and Figure 9 (row-normalized) show the confusion matrix of SVM under the fused-input setting. SVM performs well on the "Operation" category, with 59 correctly classified samples and an accuracy exceeding 80%. Compared with Experiment 1, recognition of the "Regulation" and "Function" categories improves substantially, reaching 23 and 4 correctly identified samples, respectively. Misclassifications are also concentrated in fewer category pairs, suggesting that the added structural information and context-completion text improve the separation of categories with overlapping semantics.

The percentage matrix in Figure 9 further shows that the previously under-recognized classes—"Reporting" (reporting-procedure confirmation) and "Training" (training/learning related)—now reach accuracies of 57.14% and 66.67%, respectively. The separation between rare classes becomes clearer despite the limited number of samples.

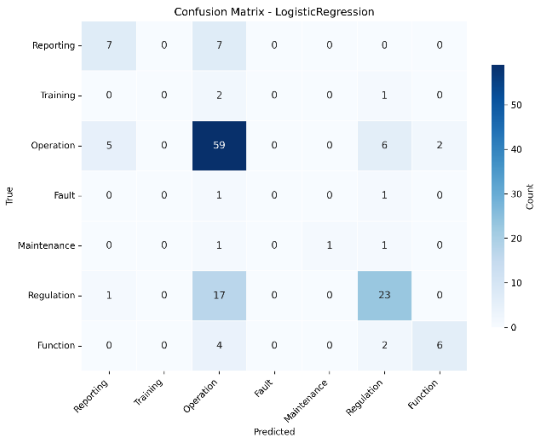

Figure 10. Confusion matrix (absolute counts) of Logistic Regression after fusion

Figure 11. Confusion matrix (percentage, row-normalized) of Logistic Regression after fusion

Figures 10 and 11 summarize the prediction results of Logistic Regression after feature fusion. The "Operation"

category remains the most accurately recognized class, with 59 correctly classified samples, and the "Function" and "Regulation" categories also show improved recognition compared with the baseline. Figure 11 reports an accuracy of 50.00% for "Function" and a recognition rate of 56.10% for "Regulation", accompanied by limited classification confusion. These results indicate that combining linguistic features and structural knowledge improves the modeling of abstract semantic categories, especially for short and implicit queries.

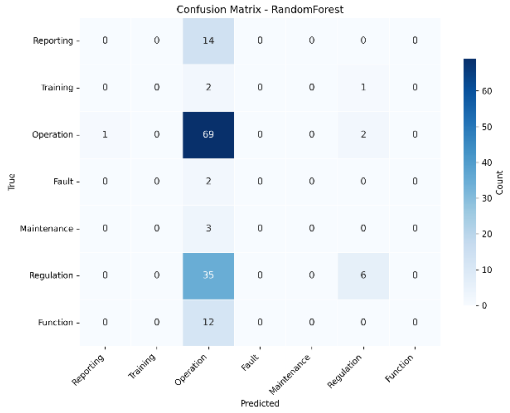

Figure 12. Confusion matrix (absolute counts) of Random Forest after fusion

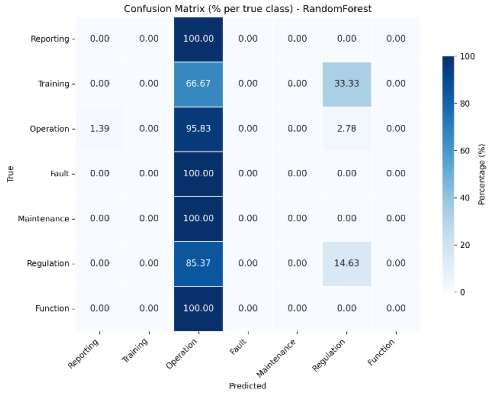

Figure 13. Confusion matrix (percentage by true-class) of Random Forest after fusion

Random Forest, tested solely with fused inputs (see Figures 12-13), performs well with "Operation" (69 correct, surpassing SVM and LR) and "Regulation" (35 correct, achieving an 85.37% recognition rate, the highest among all models).

In the normalized matrix shown in Figure 13, RF achieves more than 80% recognition for "Operation," "Function" (reaching 100%), and "Regulation," but its results are weaker for low-frequency classes such as "Fault" and "Maintenance," and it fails to predict any samples for some of them. This pattern indicates strong sensitivity to severe class imbalance. A joint reading of Table 4 and Figures 8–13 shows that the RAG/KG-enhanced system works well for classifying categories that are structurally complex, difficult to distinguish, or sparsely represented. Both SVM and LR show improved results for small-sample classes such as "Training," "Function," and "Reporting." These findings suggest that fused features help alleviate semantic underfitting under limited-data conditions.

Model architectures respond to fused features in different ways. Logistic Regression delivers the best overall results, whereas Random Forest, although effective on the majority of classes, shows less stable generalization when compared with linear approaches. This difference suggests that differences in classifier structure influence the extent to which integrated information can be utilized.

Overall, combining retrieval-augmented texts with structured knowledge-graph triples through multi-source semantic augmentation improves regulatory intent recognition while keeping the added complexity at a manageable level, which further confirms the value of the proposed feature-fusion mechanism.

4.4. Experiment 3: Ablation Study

To examine the independent contributions and interaction effects of the two enhancement methods, an ablation study was conducted. This ablation study examines the separate effect of each augmentation component. It also examines how performance shifts when components are combined, shedding light on the effects of RAG and KG on the model across various input setups. Using the power-domain dataset, SVM and Logistic Regression were evaluated under four input settings, with query text, RAG context, and KG triples controlled as follows: (1) Q (baseline): original query only; (2) Q+RAG: query + retrieval context; (3) Q+KG: query + knowledge triples, with (4) Q+RAG+KG: fused multimodal input.

To improve result reliability, the models were trained and tested with five random seeds, and the reported metric values were averaged across repeated runs. The evaluation metrics were Accuracy (for overall predictive correctness), together with Macro-F1 (for recognizing imbalanced classes). Results are summarized in Table 5:

Table 5. Model results across input settings

Input configuration	Model	Accuracy (%)	Macro-F1 (%)
A: Q	SVM	59.2	28.3
	Logistic Regression	57.1	20.3
B: Q + RAG	SVM	60.7	42.4
	Logistic Regression	58.9	48.2
C: Q + KG	SVM	57.1	38.4
	Logistic Regression	53.7	39.7
D: Q + KG + RAG	SVM	64.6	39.3
	Logistic Regression	65.3	42.2

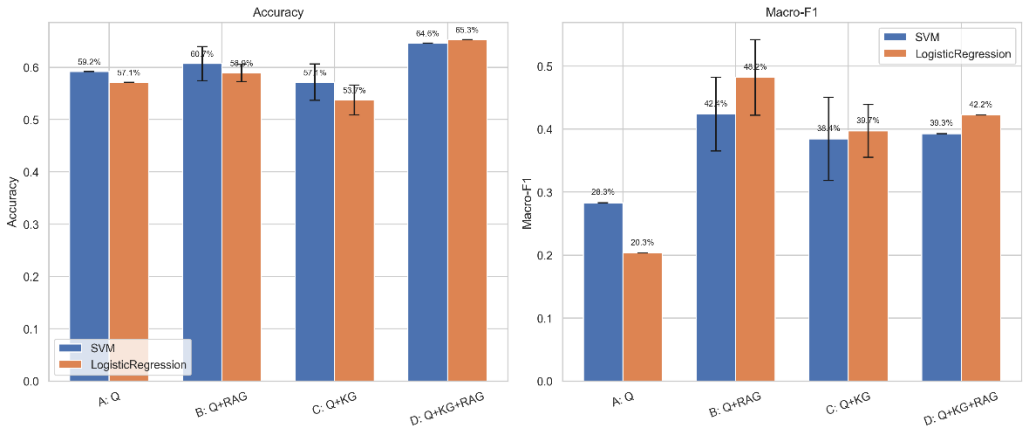

Figure 14. Effects of different enhancement configurations on Accuracy and Macro-F1

Under baseline configuration A (Q only), the two models show weak overall performance, with SVM achieving 59.2% Accuracy and 28.3% Macro-F1, whereas Logistic Regression reaches only 20.3% Macro-F1. This suggests that the raw query text fails to offer enough semantic information for high-precision intent discrimination.

After RAG is incorporated into configuration B, both models register clear gains in Macro-F1. In particular, the Macro-F1 score of Logistic Regression nearly doubles to 48.2%. This indicates that the RAG component helps reduce class-imbalance effects while improving the distinguishability of minority classes. The relatively small increase in Accuracy implies that RAG mainly improves prediction balance among classes.

In configuration C (Q+KG), SVM and Logistic Regression show only a slight improvement in Macro-F1 over the baseline, with less enhancement than RAG. This indicates that KG primarily strengthens the separation between major classes but has difficulties with rare classes due to limited triple coverage.

Configuration D (Q+RAG+KG) has the highest level of precision: Logistic Regression achieves 65.3%, and SVM reaches 64.6%, marking increases of 8.2 and 5.4 percentage points from the baseline, respectively. Although Macro-F1 does not surpass configuration B, performance in general remains robust, demonstrating tangible advantages from integrating information.

4.5. Experimental Results Analysis

Taken together, the experimental results show that adding external semantic information, including RAG texts and KG triples, improves power-regulation intent recognition, with gains in both overall Accuracy and Macro-F1. Baseline experiments indicate that conventional models built solely from the original query text reach a performance bottleneck and show a clear bias toward classes with relatively few samples. Although SVM performs better than the other traditional models, its relatively low overall F1 score still suggests limited performance transfer in the absence of contextual support.

The fusion experiments reveal a more distinct division between primary and secondary intent categories. Following fusion, Logistic Regression shows a Macro-F1 improvement exceeding 21%, indicating that the combined features offer distinguishing information, even for a linear classifier. The performance of SVM also sees a consistent improvement in accuracy of about 5.4%, which is stable even in high-dimensional feature spaces. The RF results indicate that this tree-based nonlinear classifier captures the dominant categories more readily, but its performance degrades on minority categories, where noisy fused features and limited samples weaken the stability of its decision splits.

The ablation results clarify the specific role of each augmentation component. Configuration B (Q+RAG) yields a clear improvement in Macro-F1 over Configuration A (Q), indicating that RAG helps alleviate class-imbalance and increase the separability of minority classes. Compared with the RAG-based setting, Configuration C (Q+KG) brings only marginal improvement in Accuracy and Macro-F1. This result indicates that KG mainly reinforces class separation among majority classes, but provides insufficient semantic support for minority-class recognition. Configuration D (Q+RAG+KG) produces the highest Accuracy (Logistic Regression: 65.3%); however, its Macro-F1 remains lower than that of Configuration B. This result further suggests that, although the inclusion of KG improves the model's fit to the majority classes, it also weakens the benefits previously provided by RAG for long-tail classes, which leads to localized conflicts.

Figure 14 shows a clear variation in metric distributions across configurations and models. The Macro-F1 score of SVM increases from 28.3% to 42.4% after RAG is

introduced, whereas it declines to 39.3% under full fusion (D). This result indicates that the current direct concatenation strategy does not adequately resolve semantic redundancy and conflicts among heterogeneous features, which may cause feature interference as well as lead to overfitting in some models. Therefore, more suitable fusion mechanisms, such as feature-selection modules or attention-based structures, are needed to make the semantic representations of different features more consistent.

The experimental results reveal three main trends: (1) RAG plays a central role in improving class balance; (2) KG enhances the recognition of majority classes but provides limited support for underrepresented categories; (3) fusion-based configurations raise overall Accuracy, although they still leave room for improvement in Macro-F1. These findings provide direction for the continued development of semantic-enhancement methods and point to the importance of class-sensitive modeling, attention-based fusion strategies, and adaptive attribute weighting in future modeling of industrial semantics.

5. Conclusion

This study addresses several challenges in power engineering secondary systems, including the interpretation of regulatory documents, the complexity of intent recognition for this domain, the limited availability of domain-specific corpora, as well as semantic ambiguity. To address these issues, we propose an intent recognition framework that combines Retrieval-Augmented Generation (RAG) with knowledge-graph (KG) structural semantic modeling. The proposed framework improves the model's ability to distinguish query semantics and increases the accuracy of regulation matching, especially in cases where queries involve more intricate semantic conditions.

From a methodological perspective, we designed a composite input mechanism that combines user queries with contextual information retrieved by RAG and structured triples obtained through the KG, with the resulting framework implemented using interpretable traditional classifiers such as SVM and Logistic Regression. Experiments show that the RAG component improves recognition of minority classes, whereas KG triples enhance the stability of majority-class recognition. Their combination improves not only overall Accuracy, but also class-balanced metrics.

This study examines the practical value as well as the limitations of incorporating external information into the intent-recognition task and presents a lightweight, engineering-oriented solution for intelligent power-domain QA systems. Several limitations should still be noted. First, the KG depends on manual extraction and rule-based generation, which limits entity coverage and the level of semantic detail required for more demanding inference scenarios. Second, the static concatenation strategy cannot adaptively assign weights to heterogeneous information, which may cause redundancy or conflict at the semantic level. Third, interpretable classifiers face performance limitations on long-form texts and in multi-hop reasoning settings.

Looking ahead, future research will proceed in three directions. First, attention-based feature selection will be introduced to improve the interaction among heterogeneous semantic features. Second, automatic KG construction and entity alignment will be further developed to improve both coverage and consistency. Third, the study will combine lightweight neural models with the proposed framework to support end-to-end modeling of complex semantics; this is expected to strengthen the model's capacity to generalize across different application settings and support semantic inference.

Acknowledgements

We'd like to thank the project coordinators and all colleagues involved for their valuable discussions and technical support-they were key to getting this research over the finish line.

Author Contributions

Qiangchao Xu led the conceptualization, methodology, investigation, and preparation of the original draft. Kairu Chen contributed to conceptualization, project supervision, administration, and manuscript review and editing. Zhaolun Deng handled data curation, formal analysis, and validation. Jinhua Wu contributed technical resources and consultation. Dan Lin supported investigation and visualization. Zhaolin Shi secured funding, supervised the project, and contributed to manuscript review and editing.

Funding

This work was supported by the Science and Technology Project, "Artificial Intelligence-Based Protection Relay Learning System Built on the OpenHarmony System" (Project No. 030100KC24100057, Science and Technology Code: GDKJXM20240815).

Data Availability Statement

The data supporting the findings of this study are included within the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest

The authors have no conflicts of interest to declare.

References

[1] Shah H, Chakravorty J, Chothani N. Protection challenges on integration of renewable energy sources to power system network: A review[J]. Journal of applied research and technology, 2023, 21(2): 212-226.

[2] Rubio S, Bogarra S, Nunes M, et al. Smart grid protection, automation and control: Challenges and opportunities[J]. Applied Sciences, 2025, 15(6): 3186.
[3] Zhong B, He W, Huang Z, et al. A building regulation question answering system: A deep learning methodology[J]. Advanced Engineering Informatics, 2020, 46: 101195.
[4] Gao M, Li M, Ji T, et al. Key technologies of intelligent question-answering system for power system rules and regulations based on improved BERTserini algorithm[J]. Processes, 2023, 12(1): 58.
[5] WU Y, DAI T, ZHENG Z, et al. Active Discovering New Slots for Task-Oriented Conversation[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2024, 32: 1882-1893.
[6] Song Y, Liu X, Zhou Z. A Comprehensive Review of Text Classification Algorithms[J]. Journal of Electronics and Information Science, 2024, 9(2).
[7] Li D, Wang S, Zhao B, et al. A novel model based on a transformer for intent detection and slot filling[J]. Urban Informatics, 2024, 3(1): 24.
[8] Dash T, Chitlangia S, Ahuja A, et al. A review of some techniques for inclusion of domain-knowledge into deep neural networks[J]. Scientific Reports, 2022, 12(1): 1040.
[9] Klesel M, Wittmann H F. Retrieval-Augmented Generation (RAG)[J]. Business & Information Systems Engineering, 2025, 67: 643-659.
[10] Peng C, Xia F, Naseriparsa M, et al. Knowledge graphs: Opportunities and challenges[J]. Artificial Intelligence Review, 2023, 56: 13071-13102.
[11] Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., & Seidel, J. (2025). Retrieval augmented generation for large language models in healthcare: A systematic review. PLOS Digital Health, 4(1), e0000877.
[12] Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty years of machine-learning-based text classification: A systematic review. Algorithms, 16(5), 236.
[13] Zhao J, Wen Y, Li Q, et al. Deep Learning Approaches for Multimodal Intent Recognition: A Survey[J]. arXiv preprint arXiv:2507.22934, 2025.
[14] Ran Z, Chen Y, Jiang Q, et al. Intent Recognition in Dialogue Systems[C]//Proceedings of the 4th Asia-Pacific Artificial Intelligence and Big Data Forum. 2024: 165-173.
[15] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in neural information processing systems, 2020, 33: 9459-9474.
[16] Yu W. Retrieval-augmented generation across heterogeneous knowledge[C]//Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies: student research workshop. 2022: 52-58.
[17] Gupta S, Ranjan R, Singh S N. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions[J]. arXiv preprint arXiv:2410.12837, 2024.
[18] Hogan A, Blomqvist E, Cochez M, et al. Knowledge graphs[J]. ACM Computing Surveys (Csur), 2021, 54(4): 1-37.
[19] Guo, J., Zhao, M., Zhang, W., Xu, T., Xu, L., Yu, J., Yu, M., & Yu, R. (2024). Hierarchy-Aware Quaternion Embedding for Knowledge Graph Completion. In 2024 International Joint Conference on Neural Networks (IJCNN).
[20] Wu L, Chen Y, Shen K, et al. Graph neural networks for natural language processing: A survey[J]. Foundations and Trends® in Machine Learning, 2023, 16(2): 119-328.
[21] Yasunaga M, Bosselut A, Ren H, et al. Deep bidirectional language-knowledge graph pretraining[J]. Advances in Neural Information Processing Systems, 2022, 35: 37309-37323.
[22] Wang, R., & Wang, Y. (2023). Knowledge Graph Reasoning with Bidirectional Relation-Guided Graph Attention Network. In Proceedings of the 2023 IEEE International Conference on Data Mining (ICDM). IEEE.
[23] Su H, Li H, Li D. Knowledge reasoning with multiple relational paths[J]. Connection Science, 2023, 35(1): 2161480.