informatics

MDPI

Article

Intelligent Question-Answering System for New Energy Vehicles Integrating Deep Semantic Parsing and Knowledge Graphs

Yaqi Wu [1], Pengcheng Li [1], Tong Geng [1], Yi Wang [1], Haiyu Zhang [1,2,*] and Shixiong Li [3,*]

[1] Yantai Institute, China Agricultural University, Yantai 264670, China
[2] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[3] College of Engineering, China Agricultural University, Beijing 100083, China
* Correspondence: zhanghaiyu@cau.edu.cn (H.Z.); lsx@cau.edu.cn (S.L.)

Abstract

The new energy vehicle (NEV) industry generates massive multi-source heterogeneous data. To overcome traditional database limitations in terminology disambiguation and multi-hop reasoning, this paper proposes a knowledge graph (KG)-based question-answering (QA) architecture. Three primary domain challenges are addressed: First, to tackle the poor semantic extraction of informal diagnostic texts, a deep semantic parsing network (BERT-BiLSTM-CRF) is integrated to extract high-precision knowledge from 150,000 real-world maintenance records. Second, to solve topological redundancy, the Labeled Property Graph (LPG) specification is employed to encapsulate parameters of 2157 vehicle models as internal attributes, significantly streamlining complex multi-hop reasoning. Finally, to enhance limited reasoning capabilities, an intent classification module (TextCNN) automatically translates natural language into graph queries, enabling deep fault tracing across up to five semantic levels. Experimental results demonstrate 98% and 93% accuracy in entity-relation recognition and intent classification, respectively. The resulting KG (8274 nodes, 14,488 edges) establishes a scalable paradigm for intelligent diagnostic reasoning in complex vertical domains.

Keywords: new energy vehicles; knowledge graph; intelligent question-answering system; entity alignment; fault diagnosis

1. Introduction

The development of the new energy vehicle (NEV) industry has generated multi-source, heterogeneous data encompassing vehicle structures, fault diagnostics, and after-sales feedback. In the context of the global energy transition, government subsidy policies and pricing mechanisms have significantly influenced the construction of the electric vehicle (EV) ecosystem [1]. The development of China's NEV industry is deeply correlated with both policy guidance and market performance [2]. With the rapid surge in market vehicle ownership, the complexity of vehicle operation and maintenance data has increased exponentially. Currently, digital twins are used for EV virtual modeling and monitoring [3], while decision mining techniques are used to process operational decision data [4]. Such data exhibit highly fragmented characteristics and contain complex, unstructured semantic associations. Due to rigid schemas, traditional relational databases face limitations when processing component associations with semantic ambiguity and executing multi-hop fault reasoning.

Check for updates

Academic Editor: Alsayed Algergawy

Received: 22 January 2026
Revised: 17 April 2026
Accepted: 22 April 2026
Published: 24 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Knowledge graphs (KGs), which represent entity associations through flexible topological structures, offer an effective approach to processing such data [5–7]. Furthermore, KGs are increasingly foundational in automotive applications, including macro-architecture interoperability [8], multi-source data fusion [9], fault diagnosis [10], and Graph Retrieval-Augmented Generation (Graph RAG) [11]. However, when applied to the high-noise, micro-level environment of real-world NEV after-sales question-answering (QA), these methods face two systemic bottlenecks:

First, inadequate semantic extraction in high-noise environments, with existing frameworks favoring structured or standardized inputs. Static schemas (Resource Description Framework and Web Ontology Language, i.e., RDF/OWL) [8] and Failure Mode and Effects Analysis (FMEA)-dependent Graph RAG architectures [11] struggle with the severe semantic noise, colloquialisms, and ambiguous terminology of frontline maintenance logs. Furthermore, even deep learning fusion strategies [9] relying on conventional Natural Language Processing (NLP) and pure embedding-based alignment (e.g., cosine similarity) fail to disambiguate morphologically similar but functionally distinct industrial terms (e.g., "engine" vs. "generator"), causing severe boundary misalignments.

Second, topological redundancy limiting multi-hop reasoning as traditional paradigms [8,10] rely on "node-expansion" for massive static parameters, which inherently triggers "path explosion" [10]. Attempts to bypass this via algorithmic patches [10] or generative sub-graph retrieval [11] ignore the structural root cause. Feeding large retrieved subgraphs into Large Language Models [11] often causes context overflow and logical hallucinations during deep mechanical fault tracing. Consequently, current vehicle QA systems are mostly restricted to single-hop retrieval, facing insurmountable hurdles when tracing complex 4-to-5 hop logical sequences.

Against this backdrop, our research fundamentally departs from "node-heavy" and "generative" paradigms by proposing a vertical-domain QA architecture that integrates deep semantic parsing with a topologically optimized construction paradigm. The primary scientific contributions of this work are summarized as follows:

(1) Robust Knowledge Extraction via Deep Semantic Parsing: To address terminology ambiguity and complex causal dependencies in noisy texts, this study introduces a fusion of a weighted semantic alignment strategy and a deep sequence labeling network (utilizing the BERT-BiLSTM-CRF architecture). This approach extracts core diagnostic logic from over 150,000 real-world maintenance records, establishing a high-precision factual foundation for the domain-specific KG.

(2) Graph Topological Optimization and Intent Mapping: To overcome path proliferation, the Labeled Property Graph (LPG) specification is applied to encapsulate over 50 technical parameters across 2157 vehicle variants as intrinsic entity attributes. This structural design reduces the average degree of the entire KG to 1.75, effectively mitigating topological redundancy. Furthermore, an intent classification module (TextCNN) is utilized to automatically transform unstructured user queries into structured graph commands, bridging natural language with multi-hop retrieval across up to five semantic levels.

(3) Empirical Evaluation and Architectural Deployment: The proposed methodological framework was rigorously evaluated using a finely annotated benchmark dataset of 3662 entries. Testing results demonstrate that the architecture achieves 98.0% accuracy in joint entity-relation extraction and 93.0% in intent classification, significantly outperforming baseline models. The resulting industrial-grade vertical KG, comprising 8274 nodes and 14,488 edges, offers a scalable and verifiable paradigm for data structuralization and automated diagnostic QA in complex engineering domains.

The remainder of this paper is organized as follows: Section 2 details the methodology for constructing the NEV KG, encompassing multi-source data preprocessing, ontology construction, and knowledge extraction. Section 3 elaborates on the analytical architecture and engineering implementation of the intelligent QA system, as well as the query mapping mechanism. Section 4 presents the experimental verification and result analysis. Finally, Section 5 concludes the study and outlines future research perspectives.

2. Research on the Construction of Knowledge Graphs for New Energy Vehicles

2.1. Theoretical and Logical Framework for Developing Vertical Knowledge Graphs in the New Energy Vehicle Domain

2.1.1. Fundamental Architecture and Intrinsic Value of Knowledge Graphs

Fundamentally, a KG is a semantic network that utilizes graph structures to model entity relationships, facilitating structured knowledge representation through a dual-layer architecture comprising a schema layer and a data layer [12,13]. In contrast to the inherent limitations of traditional relational databases in accommodating multi-source heterogeneous data associations and addressing dynamic knowledge linkage requirements, KGs, with their flexible topological structure, demonstrate superior capability in integrating multidimensional data, including design parameters, fault cases, and policy standards within the NEV domain through the "entity-relation-entity" triplet paradigm, thereby providing robust structured knowledge support for subsequent intelligent QA systems [8,14,15]. Figure 1 illustrates the four-stage construction pipeline: (1) raw data collection and preprocessing; (2) construction of a domain-specific ontology framework for NEVs; (3) execution of knowledge extraction utilizing a weighted entity alignment strategy coupled with the BERT-BiLSTM-CRF model; and (4) importation of the finalized structured data into a Neo4j graph database for storage and query validation.

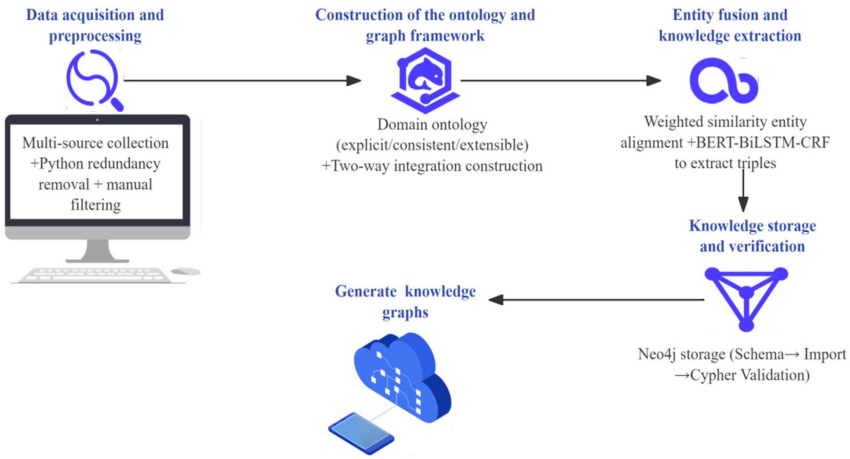


Figure 1. Technical Roadmap for constructing KGs in the NEV Domain.

2.1.2. Ontology Construction in the Field of New Energy Vehicles

An ontology is a formalized, standardized conceptualization of domain-specific knowledge, whose fundamental components include concepts, relations, attributes, axioms, and instances [6,13]. Given the distinctive characteristics of the NEV domain, the ontology construction process strictly adheres to three cardinal principles: conceptual clarity, logical consistency, and structural scalability. Furthermore, it implements a hybrid methodological approach that integrates "optimized reuse with semi-automated construction" strategies. As illustrated in Figure 2, this approach yields a comprehensive hierarchical structure of the NEV body architecture, clearly mapping the core concepts and their semantic relationships.

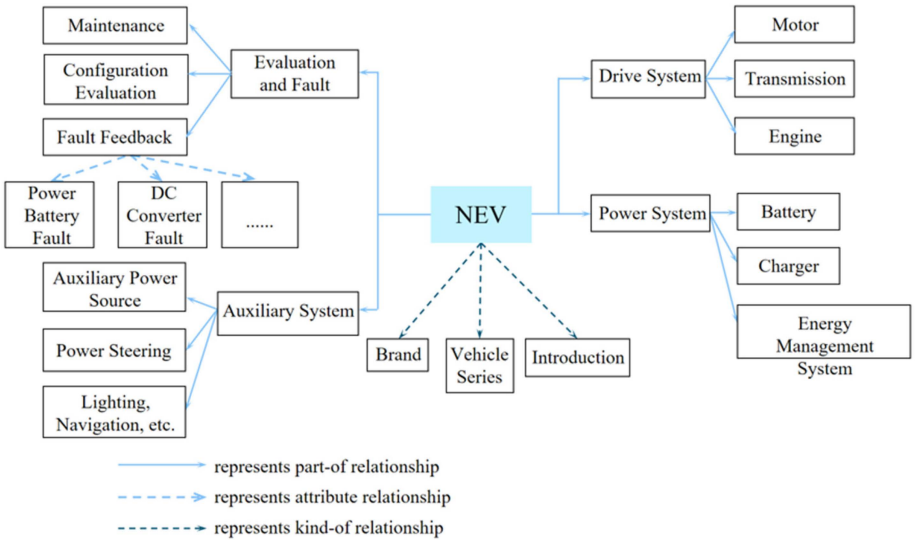


Figure 2. Hierarchical structure of NEV Body Architecture.

2.1.3. Bidirectional Fusion-Based Graph Construction

To mitigate the inherent limitations in unilateral construction methodologies- specifically, the top-down approach's excessive reliance on manual intervention and constrained pattern layer updates, as well as the bottom-up approach's susceptibility to knowledge noise generation—this study introduces a bidirectional fusion construction paradigm that synergistically integrates both top-down and bottom-up methodologies. This innovative scheme preserves the structural rigidity of the top-down approach while simultaneously incorporating the adaptive scalability of the bottom-up approach, thereby significantly improving the efficiency and precision of KG construction [7,13]. The implementation framework encompasses three principal components: top-level architectural guidance, bottom-level knowledge extraction, and dynamic iterative optimization.

2.2. Multi-Source Data Collection and Preprocessing

To cover the entire NEV knowledge domain, source corpora were collected from professional maintenance manuals, automotive portals, and official policy documents. For the semi-structured and unstructured crawled data, a two-stage preprocessing pipeline was employed. First, Python 3.8.10 scripts automatically filtered out invalid HTML tags, garbled characters, and duplicate entries. The sanitized text was then aggregated into an offline CSV benchmark dataset, integrating vehicle technical parameters with real-world fault descriptions. Finally, to ensure the quality of downstream model training, manual verification was introduced to filter out samples with ambiguous semantics or formatting anomalies.

2.3. Entity Fusion and Knowledge Extraction

2.3.1. Entity Alignment: Disambiguation of Polysemous Entities Based on Weighted Fusion Strategy

To mitigate entity redundancy induced by domain-specific synonyms (e.g., "rear bumper" vs. "rear bar," "tensioner" vs. "belt tensioner"), this study employs a weighted-similarity-based entity alignment strategy. To achieve high-precision alignment, a dual-dimensional weighted similarity computation mechanism is introduced to concurrently evaluate lexical morphology and deep semantic congruence.

In the lexical dimension, the Dice coefficient is utilized to quantify the intersection ratio of string sequences, thereby accurately assessing the morphological overlap between entities [16]. Conversely, in the semantic dimension, cosine similarity is employed to

measure the angle between word embedding vectors within a multidimensional space. This effectively circumvents alignment biases arising from entities that are morphologically similar but semantically distinct (e.g., "engine" [Fadongji] and "generator" [Fadianji], which exhibit extreme lexical proximity in Chinese but serve fundamentally divergent physical functions in automotive diagnostics) [17]. The specific formulations are defined as follows:

$$S(A, B) = 0.7 \times S_{\text{Dice}}(A, B) + 0.3 \times S_{\cos}(A, B)$$

Given the domain's fundamental characteristic—a high prevalence of synonymous terms in which morphological variations significantly outweigh semantic discrepancies—the algorithm assigns a predominant weight (0.7) to morphological-feature matching. This configuration is meticulously designed to maximize recall of high-frequency, industry-specific colloquial abbreviations. Concurrently, it integrates the calibration effect of the semantic dimension (weight 0.3). In this context, the deep vector space serves as a semantic "anchor," providing critical corrective constraints when entities exhibit lexical proximity but logical conflict.

To validate this design, we analyzed 150,000 raw user QA records. By employing hard-example mining, 200 pairs of highly challenging entities were extracted—extreme colloquial abbreviations and morphologically similar yet semantically distinct terms—with a 1:1 positive-to-negative sample ratio—to construct a rigorous evaluation benchmark. By isolating the interference from other system modules and executing a targeted ablation study, the quantitative evaluation results are presented in Table 1.

Table 1. Ablation Study of the Weighted Alignment Strategy on a Subset of Highly Confusable Polysemous Entities (N = 200).

Alignment Model	Core Mechanism & Weight Distribution	Accuracy	Failure Analysis
Baseline	Relies solely on the Dice coefficient (S_{Dice})	74.0%	Highly susceptible to misclassifying "engine" and "generator" as synonymous entities (orthographically similar but semantically distinct).
Variant	Relies solely on cosine similarity ($S_{\cos}$)	83.5%	Low recall rate for extreme non-standard colloquial abbreviations such as "rear bumper/rear bar".
Ours	$S = 0.7 \times S_{\text{Dice}} + 0.3 \times S_{\cos}$	92.0%	Effectively circumvents orthographic interference and colloquial variations, significantly reducing alignment bias.

2.3.2. Knowledge Extraction: BIO Annotation Is Implemented in Collaboration with BERT-BiLSTM-CRF

Fault Entity Annotation: In accordance with fault scenarios for NEVs, fault entities are systematically categorized into two primary classifications: subjects (including component units and detection tools) and objects (including fault states and performance characteristics). The BIO tagging methodology (where B denotes the beginning of an entity, I represents the internal portion of an entity, and O indicates non-entity elements) is implemented to precisely delineate entity boundaries, with manual annotation of fault text CSV files being conducted [18]. Illustrative example: "B—detection tool leakage tester I—detection tool O detection B—component unit power battery I—component unit B—fault state leakage I—fault state O" [19].

Pipeline Extraction Implementation: The BERT-BiLSTM-CRF model is used to perform entity and relation extraction. This model effectively resolves term ambiguity through BERT, captures long-range text dependencies via BiLSTM, and optimizes annotation bound-

aries using CRF [8]. For comprehensive details regarding the model's architecture and operational principles, please refer to Section 2.3.3. The formatted results from the preceding entity recognition stage serve as the foundational input for subsequent relation recognition processes.

Data and Results: For knowledge acquisition, a hierarchical strategy is proposed that stratifies entities into Facts and Logic layers. The Facts Layer directly instantiates five static entity categories (series, models, manufacturers, classes, and energy types) from structured tables. This taxonomic design reduces label dimensionality, effectively bridging the extraction targets (4 categories) with the final KG scale (9 categories).

In the Logic Layer, knowledge extraction targets unstructured diagnostic records, aligned with NEV physical topology and diagnostic workflows. Mirroring practical scenarios like "battery leakage" (involving component identification and status evaluation), the model extracts four dynamic entity categories (component units, performance characteristics, fault states, diagnostic tools) and four semantic relations (component-fault, performance-fault, diagnostic-association, physical-composition), as detailed in Table 2.

Table 2. Entity Types and Their Relationship Types.

Entity Type	Object Type	Instances
Component Unit	Various units, parts, and equipment in the manufacturing field	"Fuel pump", "Separator"
Performance Characterization	Characteristics or performance descriptions of components	"Pressure", "Rotational speed", "Temperature"
Fault State	Descriptions of fault states of systems or components	"Oil leakage", "Fracture", "Stuck"
Detection Tool	Specialized instruments for detecting certain faults	"Zero sequence transformer", "Protector", "Leakage current tester"

(a) Entity Types

Subject Type	Object Type	RelationType	Subject Instance	Object Instance
Component Unit	Fault State	Component Failure	Engine Cover	Shaking
Performance Characteristic	Fault State	Performance Failure	Liquid Level	Lowering
Detection Tool	Performance Characteristic	Detection Tool	Detection Tool	Electric Current
Component Unit	Component Unit	Composition	Circuit Breaker	Converter Transformer

(b) Entity relationship types

Trained on 3662 annotated real-world maintenance records, the model's extracted logic relations were integrated with Facts Layer nodes via entity alignment, successfully constructing a foundational KG of 8274 nodes.

2.3.3. BERT-BiLSTM-CRF Model: Entity Recognition in the NEV Context

Traditional sequence labeling models like LSTM-CRF [20] struggle with real-world NEV maintenance records due to prevalent colloquialisms, non-standard terminology, and long-range dependencies between faults and root causes. Consequently, a BERT-BiLSTM-CRF architecture tailored for NEV-specific entity and relation recognition is proposed. As illustrated in Figure 3, the adopted entity recognition model's architectural topology

achieves feature extraction and sequence prediction through the synergistic collaboration of multi-level modules.

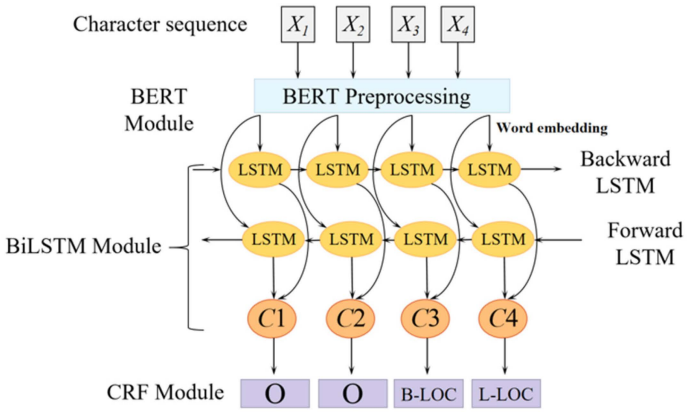


Figure 3. Entity Recognition Model.

(1) BERT Module for Polysemy and Contextual Encoding: NEV corpora abound with non-standard abbreviations and polysemous terms (e.g., "BMS" denoting either high-voltage Battery or low-voltage Body Management Systems), which evade static word vectors. Incorporating BERT's multi-head attention [21,22], the module computes global contextual weights (e.g., capturing co-occurrences of "BMS" with "power battery" versus "window") for dynamic semantic disambiguation.

(2) BiLSTM Module for Colloquial Texts and Sequence Features: Real-world after-sales records are characteristically unstructured and noisy, frequently exhibiting causal inversions (e.g., "starter cranks normally, yet the engine fails to start and the fault light illuminates"). To elucidate how the model captures such long-range semantic dependencies, Figure 4 details the internal architecture of the BiLSTM memory cell. By leveraging precise gating mechanisms (forget, input, and output gates), the bidirectional network effectively mitigates vanishing gradients [23], enabling the robust extraction of complex dependencies—such as "Component (engine)–Symptom (fails to start)–Manifestation (light on)"—from highly colloquial corpora.

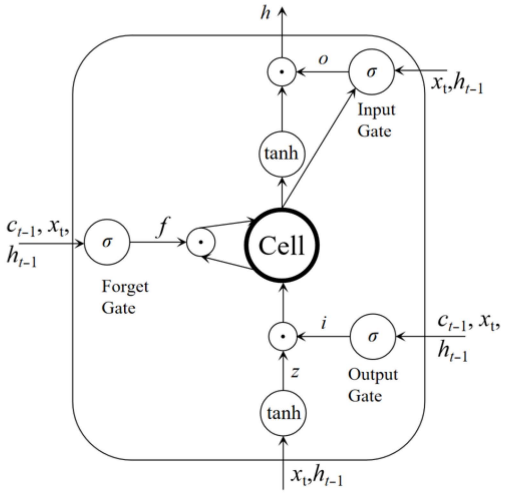


Figure 4. Long Short-Term Memory (LSTM) Model.

(3) CRF Module for Boundary Constraints: Components and fault states frequently form compound expressions (e.g., "left front shock absorber oil leak"), causing BiLSTM boundary misclassifications. The CRF output layer enforces global sequence constraints by learning a label transition probability matrix (e.g., "B-Component" must

be followed by "I-Component" or "O"). This corrects misaligned predictions (e.g., properly segmenting "[Component] rear bumper" from "[State] broken"), effectively minimizing boundary error rates in non-standard texts [22].

2.4. Knowledge Storage and Visualization Infrastructure

2.4.1. Neo4j Graph Database: Advanced Storage Solution for Multi-Source Knowledge Integration and Multi-Hop Reasoning

To address the critical limitations of traditional relational databases in supporting multi-hop reasoning and complex association queries within the NEV domain, this research adopts Neo4j graph database as the foundational platform for knowledge storage [14,24]. As a prominent NoSQL database solution, Neo4j fundamentally differs from conventional databases like MySQL by eliminating the constraints of fixed schemas and complex multi-table relationships. Its graph-based data architecture, built upon the "entity–relationship–attribute" paradigm, inherently provides essential capabilities including flexible data storage, efficient heterogeneous knowledge integration, and seamless algorithm interoperability [7]. Empirical studies conducted by Sandell et al. have substantiated Neo4j's superior performance, demonstrating an average query execution time of 0.0203 s, which represents a 93% improvement over MySQL (0.2803 s) and a 97% enhancement compared to ArangoDB (0.6037 s). Furthermore, the native Cypher query language offers exceptionally concise syntax (e.g., "MATCH (n: Model)-[r: Component Failure]-(m: Failure Status) WHERE n.name = 'Xiaopeng P7' RETURN m"), facilitating rapid structured knowledge retrieval and establishing a robust foundation for the subsequent real-time QA system [7].

2.4.2. Implementation of Knowledge Storage Utilizing Neo4j

To facilitate dynamic storage and enhance retrieval efficiency of multi-source knowledge, the construction and updating of the graph database in Neo4j adhere to the systematic procedure of "Schema Design → Data Import → Storage Verification":

Schema Design: Precisely delineate the entity label system and relationship type set, strategically design the index structure for frequently accessed fields, thereby establishing both semantic and performance foundations for graph construction.

Data Import: Leveraging preprocessed high-quality data, employ Neo4j's LOAD CSV function for batch importing of primary nodes (concurrently establishing index constraints to optimize performance), followed by batch loading of relationships and attributes to achieve comprehensive semantic integration of multi-source knowledge;

Storage Verification: Use Cypher queries to validate node counts, attribute completeness, and relationship coherence; assess index efficiency; and establish a robust data foundation for subsequent visual representation in the question-answering system.

At the storage architecture level, the system adopts the LPG specification. Instead of generating independent topological nodes for over 50 static technical parameters across 2157 vehicle models, they are encapsulated as internal properties within the corresponding vehicle entities. This encapsulation strategy curtails graph inflation, preserving semantic associations while eliminating redundant edges from numerical parameters, thereby significantly reducing the computational overhead of subsequent multi-hop traversals.

2.4.3. Storage Results and Structural Characteristics

The finalized KG comprises 8274 nodes and 14,488 edges. Driven by the property encapsulation design, the average graph degree is meticulously maintained at 1.75. This metric demonstrates that the hierarchical topology effectively mitigates redundancy when integrating static parameters with unstructured maintenance logic, ensuring robust retrieval for complex entity queries. To facilitate further research and verification, the core topological data of this finalized KG is openly accessible at our GitHub repository:

https://github.com/wy1576263-ship-it/NEV-Entity-Alignment-Benchmark (accessed on 21 April 2026).

Furthermore, the architecture resolves naming granularity discrepancies between specific configurations and base vehicle series. It achieves O(1) time complexity for parameter queries and enables multi-hop fault tracing via structural edges such as [: VARIANT_OF] and [:HAS_UNIT]. Table 3 summarizes the distribution of specific entities.

Table 3. Distribution and Technical Definitions of Multi-level Entity Types in the NEV KG.

Entity Label	Count	Proportion (%)	Technical Significance & Data Source Proof
CarModel	5609	67.8%	Core hub of the KG, connecting 5499 HAS_UNIT relationships.
CarVariant	2157	26.1%	Accurately proven by 2157 VARIANT_OF relationships, encapsulating 50+ parameters.
Manufacturer	422	5.1%	2157 PRODUCES edges converge into 422 manufacturer nodes.
Unit	21	0.25%	The core supports 5499 component associations and 176 fault inferences.
Failure	14	0.17%	Carries the normalized results of 176 UNIT_FAILURE relationships.
CarClass	27	0.33%	Corresponds to the classification endpoints of 2157 BELONGS_TO_CLASS relationships.
EnergyType	11	0.13%	Corresponds to the powertrain constraints of 2157 USES_ENERGY relationships.
Feature	9	0.11%	Corresponds to the representation layer of 82 FEATURE_FAILURE relationships.
DetectionTool	4	0.05%	Corresponds to the terminal recommendations of 12 DETECTS edges.
Total	8274	100%	The total number of nodes perfectly matches the total number of relationships (14,488).

3. Analytical Architecture and Engineering Implementation of the Intelligent QA System

3.1. Architectural Rationale and Core Methodological Choices

To operationalize the NEV QA system, a decoupled architecture is proposed. While leveraging standard frameworks (e.g., Flask, D3.js) for operational stability, its methodological core relies on intent-driven reasoning and topological interpretability.

3.1.1. Intent Parsing: TextCNN over Sequential Models

Existing QA systems relying on sequential recurrent networks (RNNs) or generative models frequently underperform on real-world NEV queries, which are characteristically fragmented, noisy, and heavily dependent on local keywords (e.g., component names or fault codes). Consequently, the TextCNN architecture [25,26] is adopted. By utilizing multi-scale convolutional kernels, TextCNN efficiently captures local semantic features irrespective of grammatical noise. This architectural alignment with fragmented NEV texts enables a robust transformation of informal natural language queries into deterministic,

structured representations for graph retrieval [27]. The TextCNN configuration is illustrated in Figure 5.

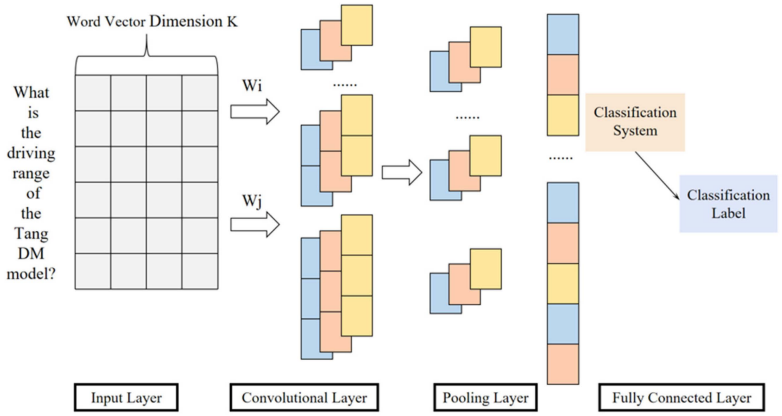


Figure 5. Schematic Diagram of the TextCNN Architecture.

3.1.2. Explainable Fault Tracing and Topological Verification

The backend and presentation layers are specifically designed to enable explainable multi-hop reasoning. Structured upon the MVT (Model–View–Template) paradigm via Flask, the backend operates as a high-concurrency conduit for executing Cypher queries within Neo4j. To overcome the "black-box" limitation of traditional text-only QA systems, a D3.js-driven dynamic topological visualization mechanism is integrated. By dynamically mapping retrieved triple data into a visual sub-graph, the system empowers domain experts to intuitively verify logical reasoning paths across hierarchical levels (e.g., "Vehicle Model → Power System → Battery Component → Fault Type"). This design transforms standard QA outputs into a verifiable and interpretable diagnostic process [28].

3.2. Question Parsing and Querying Framework Utilizing TextCNN

Translating high-noise, colloquial NEV diagnostic queries into precise multi-hop graph logic presents a fundamental scientific challenge. To overcome the semantic-to-topological conversion barrier and enhance recognition accuracy, a deterministic question-parsing framework is proposed [26]. Figure 6 illustrates this reasoning workflow, which constitutes the core methodological component of the QA algorithm.

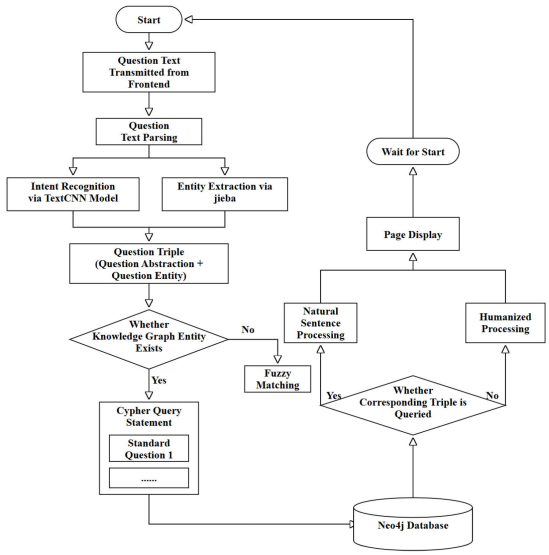


Figure 6. Query processing workflow for QA systems.

3.2.1. Character-Level Intent Parsing for Noisy Short Texts

Unlike general-domain texts, NEV maintenance queries are predominantly fragmented and suffer from severe out-of-vocabulary (OOV) issues, degrading traditional word-level sequential models. Consequently, a TextCNN architecture integrated with character-level Word2Vec embeddings is introduced. This design naturally bypasses segmentation errors, utilizing multi-scale convolutional kernels to efficiently capture salient local n-gram features (e.g., fault codes or component abbreviations) for precise intent and entity extraction.

To ensure computational rigor and reproducibility, the mathematical derivation and parameter configurations are detailed as follows. Let an input query consist of n Chinese characters, each mapped to a k-dimensional vector, yielding a feature matrix $X_{1:n} \in \mathbb{R}^{n \times k}$. Given a weight matrix $W \in \mathbb{R}^{h \times k}$ (where h denotes the convolutional kernel window size), the convolution operation is applied to the sub-matrix $X_{i:i+h-1}$ to generate a feature map $c_i = f(W \cdot X_{i:i+h-1} + b)$, where b is the bias term, and f is the activation function. Subsequently, a max-pooling operation is executed on the feature map to extract the most salient features:

$$c = \max\{c_1, c_2, \ldots, c_{n-h+1}\}$$

3.2.2. Deterministic Semantic-to-Topological Mapping

Generative QA approaches often suffer from logical hallucinations during multi-hop reasoning. To guarantee deterministic fault tracing, a template-based mapping mechanism is engineered. Extracted intent–entity pairs are structured into "question triplets" and aligned with predefined Cypher templates. This pipeline converts unstructured colloquialisms into deterministic Neo4j retrieval instructions, mitigating keyword-based redundancy and effectively bridging natural language with the graph retrieval interface [29].

3.3. Architectural Implementation and Multi-Hop Reasoning Mechanism

3.3.1. Four-Layer Decoupled Architecture Design

To operationalize the KG and bridge the semantic gap between unstructured NEV queries and deterministic graph traversal, a decoupled four-layer architecture is proposed (Figure 7). This architecture is specifically designed to isolate foundational infrastructure from the core reasoning mechanisms, ensuring both operational stability and algorithmic transparency:

(1) Base Infrastructure Layer: Provides foundational computing and storage via standard frameworks (e.g., Python, Neo4j) to sustain high-concurrency operations.
(2) Knowledge Structuralization Layer: Applies entity alignment and relation extraction to multi-source NEV data, institutionalizing the domain KG.
(3) Intent-Driven Reasoning Layer: Transcends traditional keyword matching by classifying intents via TextCNN and mapping them into strict Cypher commands, enabling deterministic multi-hop graph retrieval [30].
(4) Topological Interpretability Layer: Built on Flask and D3.js, this layer transforms backend triplet data into dynamic topologies, aiming to overcome the "black-box" nature of AI diagnostics through visual verification.

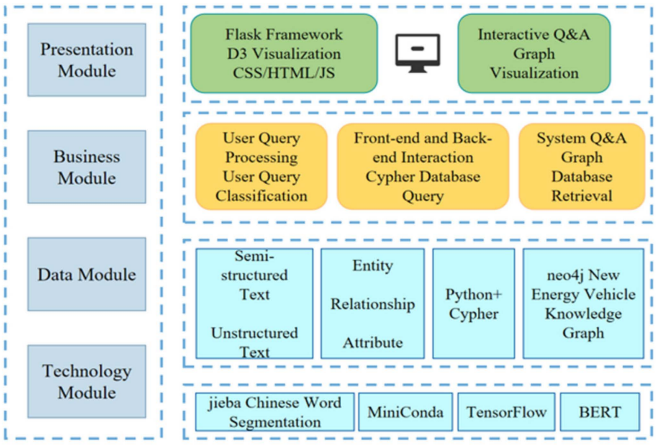


Figure 7. Systematic architecture design.

3.3.2. Methodological Implementations: Multi-Hop Reasoning and Interpretability

Building upon the foundational infrastructure, this section details the system's two primary methodological innovations: the execution of deep multi-hop reasoning and the provision of topological interpretability.

(1) Execution of Multi-Hop Graph Retrieval and Path Pruning

The fundamental scientific challenge in NEV QA scenarios is executing multi-level causal diagnostics (e.g., "vehicle variant–component–fault–detection") without triggering "path explosion." To address this, a multi-hop retrieval logic was engineered. The retrieval engine dynamically generates Cypher graph-traversal statements based on parsed query conditions. Crucially, the system leverages the Labeled Property Graph (LPG) specification to perform prerequisite filtering, significantly pruning the search space.To visually demonstrate this logic, Figure 8 illustrates the topological execution paths for these multi-hop Cypher queries.

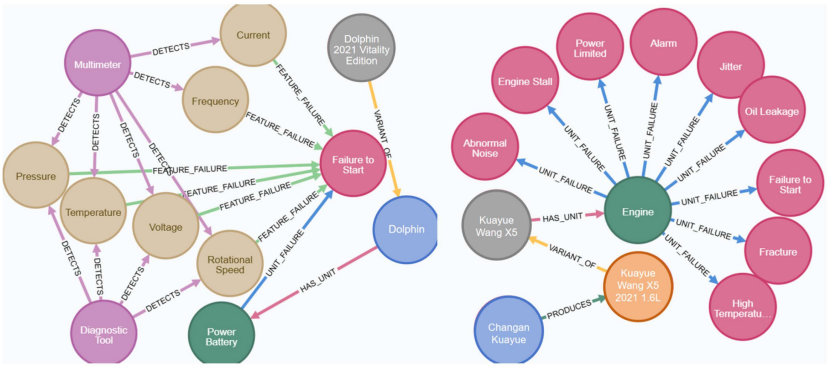


Figure 8. Topology visualization of multi-hop graph retrieval based on Cypher. The left panel illustrates the 5-hop cross-component fault diagnosis path, while the right panel demonstrates the 4-hop brand fault aggregation query path.

Scenario A: Cross-Component Fault Diagnosis (5-Hop Retrieval Path)

This query uses the internal properties of the CarVariant node (e.g., Manufacturer = 'BYD') under the LPG specification to perform prerequisite filtering. Starting from the "power battery" node, it sequentially traverses the vehicle series (CarModel), component unit (Unit), fault state (Failure), and abnormal feature (Feature), ultimately matching the associated detection tool (DetectionTool) node. Spanning 5 topological levels, this path demonstrates that the system robustly supports long-sequence logical association retrieval while effectively utilizing internal properties to curtail redundant node traversals.

Scenario B: Brand Fault Aggregation Query (4-Hop Retrieval Path)

This query originates from the Manufacturer node, traversing specific vehicle variants and core components in sequence to aggregate the corresponding fault-state nodes. Compared to computationally expensive multi-table JOIN operations in traditional relational databases, this graph retrieval achieves data aggregation and structured output of a "Manufacturer–Variant–Component–Fault" chain (e.g., retrieving high-frequency engine faults under the "Changan Kuayue" brand) via a highly efficient 4-hop path.

(2) Result Encapsulation and Topological Interpretability

Upon completing the multi-hop traversal within Neo4j, the system executes a response encapsulation pipeline. The acquired graph entity paths are converted into a standardized JSON format, which then triggers the generation of natural-language descriptions based on predefined semantic templates. This end-to-end execution chain, spanning from initial query parsing to final response delivery, is formalized in the flowchart presented in Figure 9.

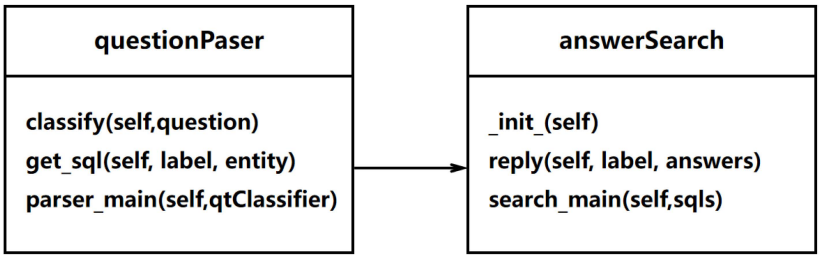


Figure 9. Flowchart of the QA response retrieval and generation process.

Beyond the textual response, the presentation layer is specifically engineered to provide topological interpretability. While generative QA models are often criticized for their "black-box" nature, this architecture integrates a dynamic graph rendering mechanism driven by D3.js. It transforms retrieved Cypher paths into an interactive visual sub-graph (Figure 10), allowing domain experts to intuitively verify the logical consistency of the diagnostic reasoning. Furthermore, a data dashboard is incorporated to visualize macro-level fault distributions (Figure 11), thereby bridging micro-level fault tracing with macro-level analytical insights.

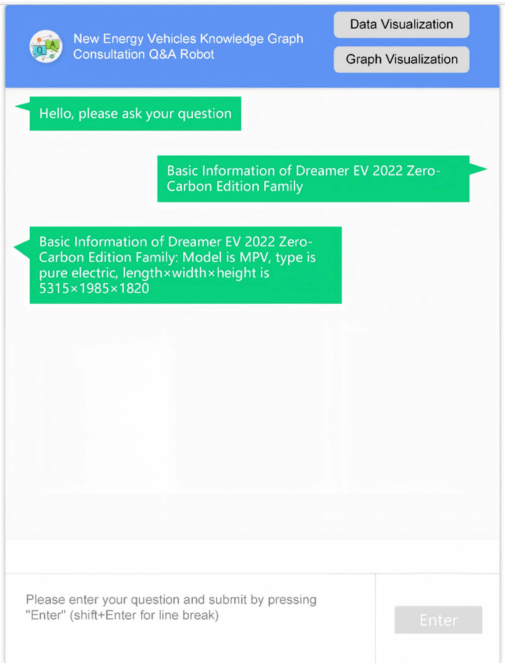
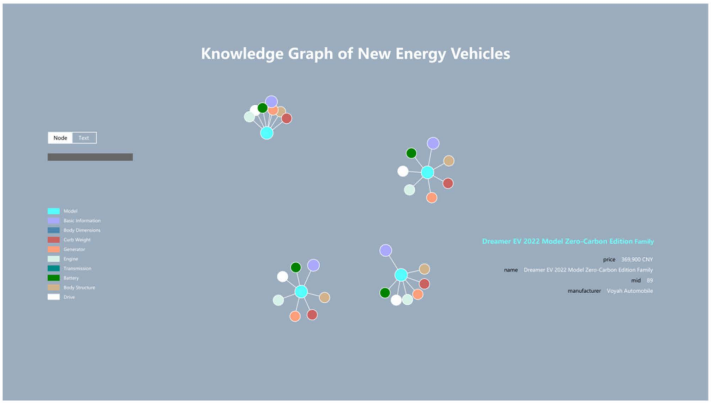


Figure 10. QA display interface.



(**a**) Big data page.



(**b**) Display page.

Figure 11. Big Data and visualization Interface.

4. Experimental Verification and Result Analysis

4.1. Experimental Environment and Dataset Configuration

To ensure computational rigor and full reproducibility, the experimental evaluation was conducted in two distinct computational stages using the TensorFlow 2.6.0 framework.

4.1.1. Hardware and Software Infrastructure

Model Fine-tuning Stage: To accommodate the high computational load of the 12-layer Transformer architecture, the initial fine-tuning of BERT-BiLSTM-CRF and TextCNN was performed on a high-performance server equipped with an NVIDIA RTX 3090 GPU (24 GB VRAM).

Extraction and Deployment Stage: Following training, the models were deployed on a local workstation (Intel i7-11850H CPU, 32 GB RAM, Windows 10 64-bit) to execute the automated knowledge extraction from 150,000 unlabelled records and to sustain the real-time QA service. This hybrid configuration demonstrates the system's capacity for low-cost industrial deployment.

4.1.2. Dataset Statistics and Reproducibility Support

The study utilizes two corpora: (1) a manually annotated benchmark set ($n = 3662$) for training and evaluation; and (2) an unlabeled corpus of 150,000 diagnostic records for large-scale extraction. The resulting KG comprises 8274 nodes and 14,488 edges. To mitigate long-tail distribution bias, denoising and minority-class augmentation were applied across 4 entity types, 4 relations, and 8 intents. A static hold-out split ratio of 8:1:1 was strictly maintained for the 3662 benchmark samples across training, validation, and testing sets.

All reported quantitative metrics reflect single-run performance on the 10% independent blind test set to prevent information leakage.

While the full 150,000 records are protected under Non-Disclosure Agreements (NDAs), a desensitized academic benchmark and reproduction scripts (including inference logits) have been open-sourced at our GitHub repository to facilitate independent verification.

4.1.3. Hyperparameterization

As detailed in Table 4, the training process was optimized for each module. The BERT-BiLSTM-CRF module was fine-tuned with a batch size of 32 and a learning rate of 3×10^{-5}. To ensure maximum feature capture and stable convergence across the complex NEV corpus, the training was executed for a maximum of 240 epochs. As illustrated in the performance curves (Figures 12–14), the models reached a stable convergent state toward the end of the training cycle. The TextCNN module utilized multi-scale kernels (2, 3, 4) with a dropout rate of 0.5. For the entity alignment phase, the weighted fusion strategy was applied with fixed coefficients of 0.7 for morphological similarity (S_{Dice}) and 0.3 for semantic similarity ($S_{\cos}$).

Table 4. Training Hyperparameter Configuration for Core Extraction and Intent Parsing Models.

Hyperparameters	Value	Target Module & Configuration Description
Software Framework	TensorFlow 2.6.0	Primary deep learning engine.
Graph Database	Neo4j 3.5.5	Knowledge graph storage and retrieval.
Transformer Layers	12	Extraction of foundational global semantic representations
Kernel Sizes	(2, 3, 4)	Extraction of multi-scale intent features
Filters	128	Number of channels per convolutional kernel dimension
Dropout Rate	0.5	Prevents overfitting in the intent classification layer
Learning Rate	3×10^{-5}	Prevents gradient oscillation in BERT pre-trained weights
BERT Fine-tuning BS	32	Optimized for VRAM constraints during the GPU-based fine-tuning phase.
TextCNN Batch Size	256	High-concurrency intent parsing during the CPU-based inference phase.
Max Epochs	240	Configured for full convergence; validated by performance curves in Figures 12–14.
Fusion Coefficients	0.7/0.3	Optimized weights for morphological (S_{Dice}) and semantic ($S_{\cos}$) alignment.

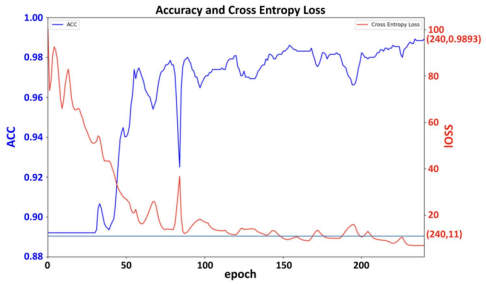


Figure 12. Accuracy versus cross-entropy Loss curve.

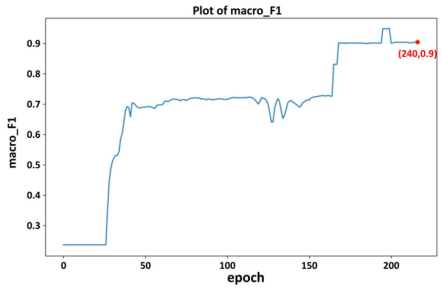


Figure 13. Macro-F1 curve Diagram.

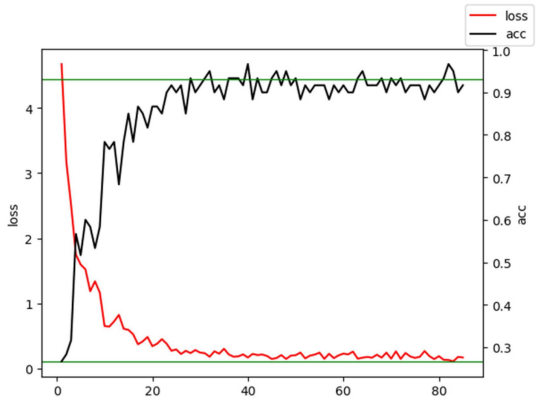


Figure 14. Accuracy and LOSS Rate curves.

4.2. Evaluation Metrics

To quantify model performance, a four-dimensional evaluation framework was established:

Precision (P): Measures the ratio of correctly extracted entities to all predicted entities, ensuring the reliability of the knowledge triplets [30].

Cross-entropy Loss: Quantifies the divergence between predicted probabilities and ground truth, reflecting the network's convergence stability [31].

Macro-F1: Calculates the unweighted mean of F1-scores across all categories. It assigns equal weight to minority classes to impartially assess generalization under long-tail distributions [30,32].

Intent Accuracy (Acc): Evaluates the success rate of intent classification, directly reflecting the efficacy of the natural language parsing module [30].

The computational formulas are as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

$$Macro - F_1 = \frac{1}{n}\sum_{k=1}^{n}(F_1)_k \tag{4}$$

$$Acc = \frac{\text{The number of correctly classified questions}}{\text{Total number of questions}} \tag{5}$$

Variable Definitions:

TP, FP, and R: TP and FP denote the frequency of correctly identified and misjudged samples, respectively; R (Recall) measures the model's coverage of actual positive samples.

Cross-entropy (Equation (2)): N is the batch size. $y_i \in \{0, 1\}$ is the ground truth label for sample i, and $\hat{y}_i \in [0, 1]$ is the predicted probability score.

Macro-metrics (Equation (4)): n denotes the number of intent categories, and $(F1)_k$ is the score for the $k - th$ independent class.

4.3. Performance Evaluation of the Knowledge Extraction Pipeline

Addressing the challenges of term adhesion and high-frequency ambiguity in NEV maintenance corpora, the BERT-BiLSTM-CRF architecture was evaluated. On the benchmark set, the joint extraction of entities and relations achieved 98% precision and a Macro-F1 score of 90% (see convergence curves in Figures 12 and 13).

The structural efficacy of the model is analyzed as follows:

Contextual Embedding: The BERT module generates dynamic word vectors, effectively resolving synonymy and contextual ambiguity.

Feature Extraction: The BiLSTM layer captures long-range temporal dependencies within the "Component-Fault-Manifestation" sequence.

Sequence Decoding: The CRF layer utilizes global transition probabilities to constrain label sequences, ensuring precise entity boundary delimitation.

The 90% Macro-F1 score demonstrates a robust balance between precision and recall, significantly mitigating the mis-extraction of non-standard terminology and providing high-fidelity structural support for the KG construction.

4.4. Robustness Validation of the Intent Recognition Module (TextCNN)

The translation of colloquial, non-standard user queries into structured graph query intents is a critical phase of the QA system. To this end, a comparative analysis between the TextCNN architecture and a Naive Bayes baseline was conducted on an independent blind test set.

Experimental results (Figure 14) demonstrate that the TextCNN model achieves 93% accuracy in multi-class intent classification, a 5% improvement over the baseline. Specifically, when processing noisy samples—such as "fault consultation" containing non-standard terminology or "policy inquiry" with semantic ambiguity—the validation loss converges steadily to approximately 0.1. This performance validates that TextCNN effectively mitigates colloquial noise through local convolutional feature extraction. The resulting intent constraint vectors are directly mapped to structured Cypher queries for the Neo4j database.

The accuracy rate of question classification reaches 93%: The TextCNN model can accurately identify user intentions such as "parameter query", "fault consultation", and "policy understanding", with a recognition rate 5% higher than that of the Naive Bayes model, effectively solving the core problem of difficult question intention recognition.

Furthermore, the model exhibits exceptional stability: the training loss rate consistently remains at approximately 10%. This stability indicates robust parsing capabilities when processing diverse, colloquial queries about NEVs (e.g., "What is the range of BYD Qin?" and "What are the common faults of extended-range vehicles?"). Moreover, its performance will not fluctuate due to differences in question expressions.

4.5. End-to-End Pipeline Testing and Evaluation

4.5.1. Quantitative Performance Assessment (Black-Box Testing)

To evaluate retrieval performance, black-box testing was conducted across four task categories: Parameter Introduction, Specification Query, Policy Interpretation, and Fault Diagnosis. For each category, 50 authentic user queries were randomly sampled (N = 200 in total). By benchmarking system-generated Cypher statements against manually annotated ground-truth queries, we quantified the retrieval accuracy for each module. The results are synthesized in Table 5.

Table 5. Evaluation of Test Results.

Number	Category	Sample	Correct Results	Accuracy Rate
1	Introduction	Introduction of BYD Qin Plus	49	98%
2	Specifications	Motor model of XPeng P7	50	100%
3	Policies	What are the preferential policies for plug-in hybrids?	45	90%
4	Faults	Common faults of range extenders	38	76%

The results reveal a correlation between retrieval accuracy and the topological depth of query tasks. For static fact retrieval (e.g., Specification Query), the system achieved a mean accuracy of 96% (peaking at 100%). This validates the efficacy of our LPG-based attribute design, which encapsulates parameters for 2000+ vehicle variants as internal node properties, effectively minimizing topological redundancy.

Conversely, for Fault Diagnosis tasks involving 4–5 traversal hops (e.g., Feature → Tool → Component → Root Cause), accuracy decreased to 76%. This decline is attributed to semantic ambiguity in natural language and to the accumulation of errors inherent in long-sequence multi-hop retrieval. These findings highlight the challenge of maintaining precision in deep logical inference within unconstrained linguistic contexts.

4.5.2. Error Analysis of Multi-Hop Retrieval

To investigate performance degradation in multi-hop diagnostics, this section conducts a qualitative analysis of representative error cases in which the system failed to return correct results, as detailed in Table 6.

Table 6. Qualitative Attribution Analysis of Representative Diagnostic Failures.

Case ID	Real User Query	Error Category	Actual Performance & Failure Point	Root Cause Analysis
1	"Can I replace the gearbox of my new Tank 300 due to oil leakage and abnormal noise?"	Missing Complex Coordinate Intent	Only retrieved and returned the "oil leakage" node.	The single-hop query template decomposes and executes the dual "AND" logic.
2	"Why does the car 'roar but will not move' when I step on the gas?"	Slang/Jargon Entity Deviation	Failed to link to the target node (power slipping).	Lacked a synonym vector mapping mechanism to align industry slang with standard terminology.
3	"The front chassis keeps making a clicking noise."	Spatial Granularity Mismatch	Unable to drill down and lock onto specific micro-components.	Macro-spatial pronouns ("chassis") cannot be directly mapped downwards to specific micro-components (e.g., shock absorbers) at the bottom layer of the graph.
4	"How is the quality of the Hyundai Elantra? Is it worth buying?"	Intent Out-of-Domain (OOD)	Erroneously switched to the historical fault list of this model.	The graph focuses on after-sales diagnosis; the model lacks a rejection mechanism for out-of-domain intents like "purchasing guide".

Table 6. *Cont.*

Case ID	Real User Query	Error Category	Actual Performance & Failure Point	Root Cause Analysis
5	"What causes the clutch not to disengage, and how to handle it?"	Multi-hop Reasoning Breakage	Only returned the cause, missing the underlying diagnostic tools.	In local sparse subgraphs, explicit relationship edges between performance anomalies and diagnostic tools are missing (KG Sparsity).
6	"The 4S shop found nothing, but it costs hundreds every time."	Implicit Complaint (Entity-less)	Entity extraction failed, triggering an execution error.	Emotional venting expressions lack the core technical entities required to trigger a graph query.
7	"I filed a complaint on 17 June. How long is the processing cycle?"	Missing Non-diagnostic Business Rules	Unable to parse the graph query path.	The underlying graph has not integrated dynamic customer service business rule data (e.g., work order routing).
8	"Are there any recommendations for NEV shock absorbers that do not leak oil?"	Negation Semantic Misjudgment	Reversed all shock absorbers with "oil leakage".	Traditional entity extraction ignores the negation constraint, triggering reverse feature matching.
9	"The tire pressure is only 1.8 now. Will it trigger an alarm?"	Missing Continuous Numerical Reasoning	Failed to trigger the low-pressure alarm threshold in the graph.	Graph nodes are mostly discrete text, lacking the capability to calculate boundaries for continuous numerical values (e.g., IF value < 2.2).
10	"That round part is broken, will it cause oil leakage?"	Coreference Resolution Failure	Unable to lock onto a specific part entity.	Pure-text graph queries cannot combine visual shapes ("round") for commonsense reasoning and pronoun restoration.

Failures in multi-hop reasoning primarily stem from three technical bottlenecks:

Entity Alignment Gaps: Colloquialisms missing from the synonym lexicon cause Entity Linking (EL) failures.

Complex Logic Constraints: The system struggles with negative constraints (e.g., "no leakage") and lacks numerical threshold processing.

KG Sparsity: Missing explicit associative edges between long-tail nodes interrupt reasoning paths.

4.5.3. System Stability and Integration Testing

End-to-end connectivity tests confirm a seamless data flow from natural language input and Cypher execution to D3.js visualization. Throughout multi-round interactions, the system exhibited no interface latency or blockages, demonstrating robust operational stability that meets industrial design requirements.

5. Conclusions and Future Perspectives

To address knowledge fragmentation in the NEV domain, this study developed and implemented a multi-layered KG QA architecture based on topological optimization. By integrating deep semantic parsing for knowledge extraction and intent classification, the system successfully transforms unstructured natural language into structured graph queries. The core methodological value of this framework lies in the parameter-encapsulation strategy under the LPG specification, which fundamentally mitigates topological redundancy

and enables complex multi-hop retrieval across multiple semantic hierarchies. This research offers a robust, scalable paradigm for data structuralization and intelligent diagnostic reasoning across complex industrial vertical domains.

The proposed architecture, synthesized from over 150,000 maintenance records and 2157 vehicle variants, resulted in a vertical KG comprising 8274 nodes and 14,488 edges. By employing a BERT-BiLSTM-CRF model for knowledge extraction and a TextCNN-based intent classification module, the system translates natural language into precise Cypher queries. Empirical validation on a benchmark of 3662 labeled records demonstrated accuracies of 98.0% for Information Extraction (IE) and 93.0% for intent recognition. Performance testing confirms that the attribute-encapsulation strategy maintains high accuracy (>96.0%) for fact-based queries and effectively supports long-sequence retrieval spanning up to five topological layers.

Despite its efficacy in static facts and shallow logical queries, the system encounters specific methodological limitations when faced with extremely complex industrial scenarios. First, the inference attenuation issue: in deep diagnostic traversals exceeding 4–5 hops, retrieval precision declines due to data sparsity in long-tail fault samples and the accumulation of long-sequence reasoning errors. Second is the computational scalability constraint: the response efficiency of long-sequence graph traversals under high-concurrency environments requires further optimization.

Future research will advance along two core trajectories to address these limitations.

Enhanced Representation Learning for Long-Tail Distributions: Future work will explore advanced representation learning techniques to improve the model's boundary recognition and generalization for non-standard colloquialisms and rare long-tail entities, thereby reducing the model's reliance on exhaustive domain-specific manual annotations.

Cognitive Intelligence Evolution via the Graph RAG Paradigm: The framework will delve into the deep coupling of structured KGs with Large Language Models (LLMs). By introducing vertical-domain KGs as external factual constraints to suppress "hallucination" in generative models, this integration aims to comprehensively enhance the reliability, explainability, and underlying cognitive intelligence of autonomous fault diagnosis systems.

Author Contributions: Conceptualization, H.Z. and S.L.; Methodology, Y.W. (Yaqi Wu), P.L., T.G., H.Z. and S.L.; Software, Y.W. (Yaqi Wu) and P.L.; Validation, Y.W. (Yaqi Wu), P.L. and Y.W. (Yi Wang); Formal analysis, H.Z.; Investigation, Y.W. (Yaqi Wu), P.L., T.G., H.Z. and S.L.; Resources, P.L. and S.L.; Writing—original draft, Y.W. (Yaqi Wu), P.L. and T.G.; Writing—review & editing, Y.W. (Yaqi Wu), T.G., H.Z. and S.L.; Visualization, Y.W. (Yaqi Wu), T.G., Y.W. (Yi Wang) and S.L.; Supervision, Y.W. (Yi Wang); Project administration, H.Z.; Funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62573421) and Undergraduate Research Program of China Agricultural University (Project No. U2024091). Additional institutional support was provided by the Yantai Institute of China Agricultural University.

Data Availability Statement: The core topological data (including the developed ontology schemas and nodes), desensitized evaluation benchmark, and verification scripts supporting the findings of this study are openly available on GitHub at: https://github.com/wy1576263-ship-it/NEV-Entity-Alignment-Benchmark (accessed on 21 April 2026). The full raw diagnostic corpus and proprietary model weights are restricted under Non-Disclosure Agreements (NDAs) with industry partners.

Acknowledgments: The authors sincerely acknowledge all members of the research team for their dedicated contributions to data collection, model validation, and system testing. Special thanks are also extended to the anonymous reviewers for their constructive comments, which have substantially enhanced the quality of this manuscript.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Furthermore, all authors declare that there are no potential conflicts of interest with the cooperative enterprises involved in the research, including but not limited to memberships, employment, consultancies, stock or share ownership, honoraria, grants, funding, or any other relevant professional or personal relationships.

References

1. Takiso, T.A.; Yu, J. Research progress on the optimization of thermal management systems for lithium-ion batteries in new energy vehicles. *J. Energy Storage* **2025**, *134*, 118144. [CrossRef]
2. Mohammadzadeh, N.; Zegordi, S.H.; Nikbakhsh, E.; Kashan, A.H. Optimal subsidy and pricing in the electric vehicle ecosystem: A case study on energy pricing policies. *Sustain. Futures* **2025**, *10*, 101298. [CrossRef]
3. Bhatti, G.; Mohan, H.; Singh, R.R. Towards the future of smart electric vehicles: Digital twin technology. *Renew. Sustain. Energy Rev.* **2021**, *141*, 110801. [CrossRef]
4. Li, S.X.; Liu, Y.Q.; Wang, J.Y.; Zhang, L. China's new energy vehicle industry development policy: Based on the market performance. *China Popul. Resour. Environ.* **2016**, *26*, 158–166.
5. Li, T.; Ma, L.; Liu, Z.; Yi, C.; Liang, K. Dual Carbon Goal-Based Quadrilateral Evolutionary Game: Study on the New Energy Vehicle Industry in China. *Int. J. Environ. Res. Public Health* **2023**, *20*, 3217. [CrossRef] [PubMed]
6. Yang, Y.J.; Xu, B.; Hu, J.W. An accurate and efficient domain knowledge graph construction method. *J. Softw.* **2018**, *29*, 2931–2947.
7. Meng, F.Q.; Yang, S.S.; Wang, J.D. Creating knowledge graph of electric power equipment faults based on BERT-BiLSTM-CRF model. *J. Electr. Eng. Technol.* **2022**, *17*, 2507–2516. [CrossRef]
8. Qi, Y.; Mai, G.C.; Zhu, R.; Zhang, M. EVKG: An interlinked and interoperable electric vehicle knowledge graph for smart transportation system. *Trans. GIS* **2023**, *27*, 613–630. [CrossRef]
9. Xie, C.T.; Deng, L.; Tang, Z.T.; He, J. Fusion and construction strategy of knowledge graphs from multi-source data. In Proceedings of the 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkuru, India, 4–5 December 2024; pp. 1–6.
10. Su, C.; Hou, P.; Liu, F.; Yi, X. A review of knowledge graph-based research methods for fault diagnosis of special vehicles. In Proceedings of the 2024 IEEE International Conference, Hangzhou, China, 11–14 October 2024.
11. Ojima, Y.; Sakaji, H.; Nakamura, T.; Sakata, H.; Seki, K.; Teshigawara, Y.; Yamashita, M.; Aoyama, K. Knowledge management for automobile failure analysis using graph RAG. In Proceedings of the 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 15–18 December 2024; pp. 6624–6630.
12. Ma, Z.G.; Ni, R.Y.; Yu, K.H. Recent advances, key techniques and future challenges of knowledge graph. *Chin. J. Eng.* **2020**, *42*, 1254–1266.
13. Huang, H.Q.; Yu, J.; Liao, X.; Xi, Y.J. Review on knowledge graphs. *Comput. Syst. Appl.* **2019**, *28*, 1–12. Available online: http://www.c-s-a.org.cn/1003-3254/6915.html (accessed on 21 April 2026). (In Chinese)
14. Perquku, A.; Minkovska, D.; Stoyanova, L. Modeling and processing big data of power transmission grid substation using Neo4j. *Procedia Comput. Sci.* **2017**, *113*, 9–16. [CrossRef]
15. Liu, Q.; Li, Y.; Duan, H. Knowledge Graph Construction Technology Overview. *J. Comput. Res. Dev.* **2016**, *53*, 582–600.
16. Wahyuningsih, T.; Henderi, H.; Winarno, W. Text mining an automatic short answer grading (ASAG): Comparison of three methods of cosine similarity, jaccard similarity and Dice's coefficient. *J. Appl. Data Sci.* **2021**, *2*, 45–54. [CrossRef]
17. Leewis, S. Improving Operational Decision-Making Through Decision Mining. Ph.D. Thesis, HU University of Applied Sciences Utrecht, Utrecht, The Netherlands, 2025.
18. Reimers, N.; Gurevych, I. Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. *arXiv* **2017**, arXiv:1707.06799. [CrossRef]
19. Jurafsky, D.; Martin, J.H. RNNs and LSTMs. In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed.; Stanford University: Stanford, CA, USA, 2024; Chapter 8.
20. Shen, H.J.; Tian, C.J.; Chen, X.; Ou, J.X.; Hu, X.B.; Han, M. A study on a domain BERT-based named entity recognition method for faulty text. *Data Inf. Comput. Sci.* **2025**, *67*, 88–97.
21. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
22. Chen, S.Y.; Niu, L.Y.; Li, J.N. Structured Element Extraction from Official Documents Based on BERT-CRF and Knowledge Graph-Enhanced Retrieval. *Mathematics* **2025**, *13*, 2779. [CrossRef]

23. Xu, G.X.; Meng, Y.T.; Qiu, X.Y.; Yu, Z.H.; Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access* **2019**, *7*, 51522–51532. [CrossRef]
24. Neo4j Team. Neo4j Graph Database & Analytics: Graph Database Management System. Available online: https://neo4j.com/ (accessed on 26 December 2025).
25. Yan, Y. ERNIE-TextCNN: Research on classification methods of Chinese news headlines in different situations. *Sci. Rep.* **2025**, *15*, 29071. [CrossRef]
26. Jiang, X.; Song, C.; Xu, Y.; Li, Y.; Peng, Y. Research on sentiment classification for netizens based on the BERT-BiLSTM-TextCNN model. *PeerJ Comput. Sci.* **2022**, *8*, e1005. [CrossRef]
27. Zhang, S.; Liu, K.; Xu, Y. TransCNN: A novel architecture combining transformer and TextCNN for detecting N4-acetylcytidine sites in human mRNA. *Anal. Biochem.* **2025**, *703*, 115882. [CrossRef]
28. Ono, K.; Demchak, B.; Ideker, T. Cytoscape tools for the web age: D3.js and Cytoscape.js exporters. *F1000Research* **2014**, *3*, 143. [CrossRef]
29. Tran, Q.B.H.; Waheed, A.A.; Chung, S.T. Robust Text-to-Cypher Using Combination of BERT, GraphSAGE, and Transformer (CoBGT) Model. *Appl. Sci.* **2024**, *14*, 7881. [CrossRef]
30. Sun, X.; Liu, Z.; Huo, X. Six-Granularity Based Chinese Short Text Classification. *IEEE Access* **2023**, *11*, 35841–35852. [CrossRef]
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
32. Lipton, Z.C.; Elkan, C.; Naryanaswamy, B. Optimal thresholding of classifiers to maximize F1 measure. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 225–239.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.