Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(1), 41-58
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Dynamic Knowledge Base Updating Mechanism-Based Retrieval-Augmented Generation Framework for Intelligent Question-and-Answer Systems

Yu Li[1,2]

[1]Technical Research Center, China Datang Digital Technology Co. Ltd., Baoding, China
[2]School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China
Email: yli03191990@gmail.com

How to cite this paper: Li, Y. (2025) A Dynamic Knowledge Base Updating Mechanism-Based Retrieval-Augmented Generation Framework for Intelligent Question-and-Answer Systems. *Journal of Computer and Communications*, **13**, 41-58.
https://doi.org/10.4236/jcc.2025.131004

Received: December 6, 2024
Accepted: January 19, 2025
Published: January 22, 2025

Abstract

In the context of power generation companies, vast amounts of specialized data and expert knowledge have been accumulated. However, challenges such as data silos and fragmented knowledge hinder the effective utilization of this information. This study proposes a novel framework for intelligent Question-and-Answer (Q&A) systems based on Retrieval-Augmented Generation (RAG) to address these issues. The system efficiently acquires domain-specific knowledge by leveraging external databases, including Relational Databases (RDBs) and graph databases, without additional fine-tuning for Large Language Models (LLMs). Crucially, the framework integrates a Dynamic Knowledge Base Updating Mechanism (DKBUM) and a Weighted Context-Aware Similarity (WCAS) method to enhance retrieval accuracy and mitigate inherent limitations of LLMs, such as hallucinations and lack of specialization. Additionally, the proposed DKBUM dynamically adjusts knowledge weights within the database, ensuring that the most recent and relevant information is utilized, while WCAS refines the alignment between queries and knowledge items by enhanced context understanding. Experimental validation demonstrates that the system can generate timely, accurate, and context-sensitive responses, making it a robust solution for managing complex business logic in specialized industries.

Keywords

Retrieval-Augmented Generation, Question-and-Answer, Large Language Models, Dynamic Knowledge Base Updating Mechanism, Weighted Context-Aware Similarity

1. Introduction

Power generation companies, as typical asset-intensive, technology-intensive, and knowledge-intensive production enterprises, have accumulated vast amounts of specialized data and rich expert knowledge. However, due to the lack of effective technological tools, challenges such as data silos and fragmented knowledge have emerged, resulting in large volumes of data not being transformed into useful knowledge or insights. This issue hinders the ability of data to effectively support the company's operations and decision-making processes. As the core driving force of the Fourth Industrial Revolution, Artificial Intelligence (AI) plays a crucial role in advancing the national energy transition. To fully leverage industry data, technologies such as Machine Learning (ML), Neural Networks (NN) [1]-[3], and graphs databases [1]-[3] have been applied within the power generation sector.

In November 2022, the release of ChatGPT 3.5 [4] by OpenAI attracted widespread global attention due to its realistic natural language interactions and multimodal content generation capabilities. Since then, as a core technology of Generative Artificial Intelligence (GAI), Large Language Models (LLMs) [5] [6] have brought new insights. When handling expert knowledge, LLMs have demonstrated significant capabilities and potential in several areas:

1) **Versatility.** LLMs acquire knowledge from multiple domains, enabling enterprises to perform tasks such as data analysis, document writing, code generation, and reasoning [7].

2) **Generative Capability.** LLMs can produce logically coherent solutions, providing valuable references and support for decision-makers [8].

3) **Natural Language Understanding (NLU).** LLMs can comprehend complex human language, efficiently processing diverse data and domain-specific information across various industries [9].

These advantages offer feasible solutions for managing complex business logic in specialized industries. However, LLMs also have certain limitations:

1) **Hallucination Issue.** LLMs may generate content that appears coherent in form and grammar but is actually inconsistent with real-world knowledge or facts and may even be entirely fabricated [10].

2) **Lack of Specialization.** LLMs may lack the in-depth knowledge required for highly specialized fields [11].

3) **Lack of Timeliness**. The training data for LLMs is often outdated, which may result in an inability to meet users' demands for the most current information and developments in a field.

To address these challenges, Facebook AI Research (FAIR) [12] introduced Retrieval-Augmented Generation (RAG). RAG combines two key components: information retrieval and text generation. When given a query, RAG first retrieves relevant documents or information from an external knowledge base. Then, it uses this retrieved content to generate a more accurate and contextually relevant response. This approach allows the model to produce better outputs by leveraging up-to-date and specific external knowledge rather than relying solely on its internal training data. It has been widely applied in Question-and-Answer (Q&A) systems [13]-[16]. However, the data in the databases used by RAG, especially regulatory documents in enterprises, is often updated irregularly. These updates may involve a complete revision of a regulatory document or the restatement of specific provisions in other documents. Consequently, issues related to the coordination and conflicts between new and old data in these databases can affect the response accuracy of RAG.

In this study, a time decay model is introduced into the RAG framework to help LLMs prioritize the use of the latest and most relevant information. Additionally, a Dynamic Knowledge Base Updating Mechanism (DKBUM) is proposed for rapidly evolving fields, such as financial markets, medical diagnostics, and legal regulations. This study also leverages RAG technology in conjunction with databases like Knowledge Graphs (KG) [17] and Relational Databases (RDB) to construct an enterprise-level intelligent Q&A system.

The remainder of this study is organized as follows: Section 2 introduces the background knowledge related to Q&A systems. Section 3 details the framework of the RAG-based intelligent Q&A system. In Section 4, the feasibility of the proposed method is demonstrated through various examples. Finally, the concluding section offers prospective remarks.

2. Method Overview

2.1. Transformer

Transformer [18] forms the foundation of LLMs. It is designed for Sequence-to-Sequence (Seq2Seq) tasks and is built around the concept of attention mechanisms, which differs significantly from traditional Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). As illustrated in **Figure 1**, the transformer architecture consists of an encoder-decoder structure. On the left side is the encoder, responsible for processing input sequences, and on the right side is the decoder, which generates output sequences.

Both the encoder and decoder consist of six identical layers ($N_x = N_y = 6$). Each encoder layer includes a multi-head attention mechanism and a feed-forward layer. The decoder contains these layers as well, along with an additional multi-head attention layer to incorporate information from the encoder's output into the decoder's input. Moreover, masked multi-head attention is employed in the decoder to prevent information leakage from future positions.

The multi-head attention mechanism comprises several scaled dot-product attention layers. It maps a query and a set of key-value pairs to a high-dimensional space. The scaled dot-product attention computes the dot product of the query and the key vectors, then scales it by a factor of $\sqrt{d_k}$. Mathematically, the scaled dot-product attention can be represented as:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

where **D, K,** and **V** are the query, key, and value vectors, respectively; and d_k is the dimensionality of the key vector.

The multi-head attention mechanism concatenates the outputs of multiple scaled dot-product attention layers and then processes the concatenated output through a Fully Connected (FC) layer.

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,\cdots,\text{head}_h)W^o \tag{2}$$

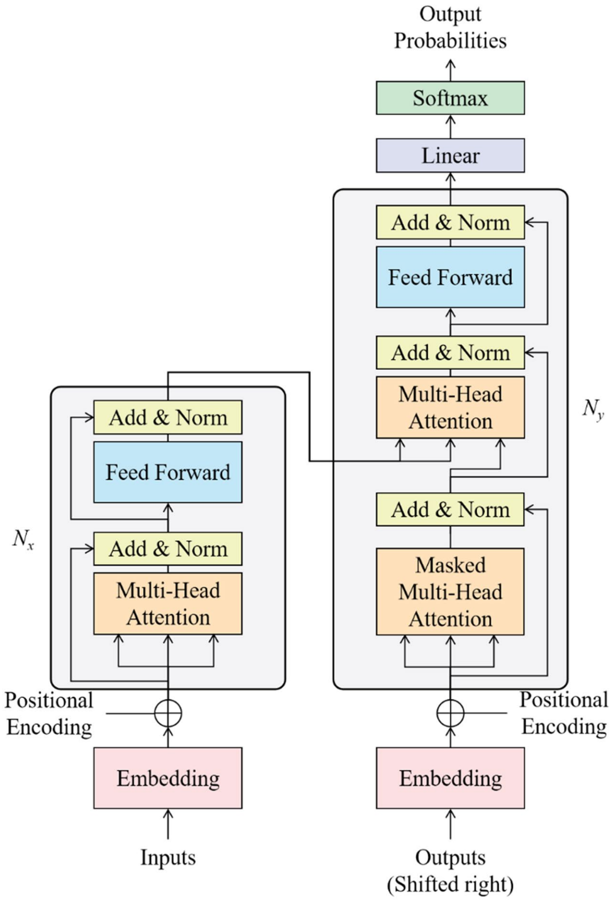

Figure 1. Transformer architecture.

Importantly, like CNNs, the attention mechanism in transformers does not inherently capture the sequential information of input data. Therefore, Positional Encoding (PE) is introduced to represent the positional information, enabling the model to consider the order of the sequence.

$$\begin{cases} \text{PE}_{(\text{pos},2i)} = \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \\ \text{PE}_{(\text{pos},2i+1)} = \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \end{cases} \tag{3}$$

where pos is the position; i is the dimension; and d_{model} is the dimensional of the input embeddings.

In this study, Qwen2.5-72B is employed due to its superior performance in NLU and function calls.

2.2. Graph Database

Graph databases [19] are specialized databases designed to store and query graph data structures. In a graph database, data is stored and managed in the form of a graph, where nodes represent entities and edges represent the relationships between those entities.

Neo4j [20] is a widely used graph database management system that employs a graph data model to represent and store data. Unlike RDBs, which organize data in tables and rows, Neo4j represents data as nodes and relationships, enabling highly efficient querying and traversal of complex networks. Its query language, Cypher, is a descriptive and declarative graph query language designed for graph analysis, and it is renowned for its expressive power, user-friendliness, and ease of use.

In this study, Neo4j is utilized to construct a company-level KG that stores various regulatory documents. As illustrated in Figure 2, 117 regulations are stored across three layers: the first layer represents the company (marked in red), which can further store other categories of knowledge; the second layer consists of regulation names (marked in blue); and the third layer contains institutional provisions (marked in grey). To facilitate retrieval and matching, the nodes in these three layers are labeled as "company", "regulation", and "regulation sub-item", respectively. Statistically, the KG comprises 117 nodes labeled as "regulation", 3,903 nodes labeled as "regulation sub-item", and 6,019 relationships.

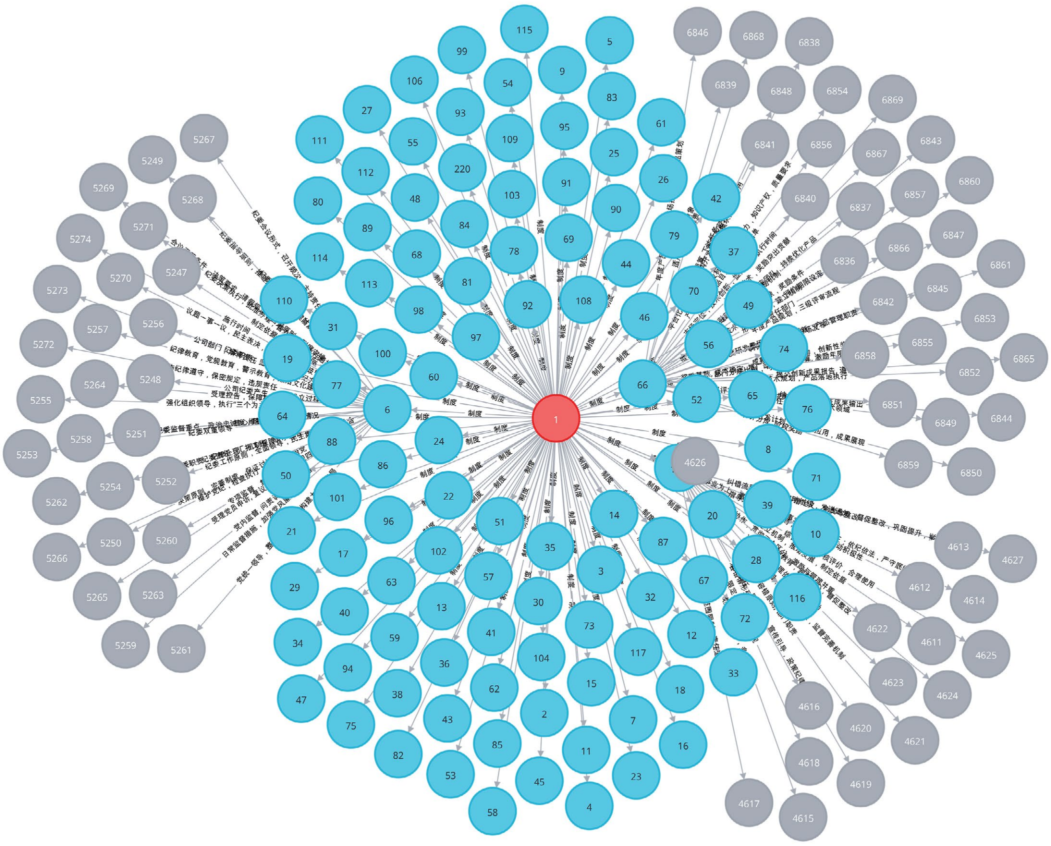

Figure 2. The KG storing regulatory documents.

To ensure data integrity, traditional document-splitting methods based on fixed chunk sizes are not employed. Instead, documents are split according to paragraphs. Additionally, for regulation-related documents, a query may span multiple paragraphs. Consequently, if a paragraph begins with a list marker, it is merged with the preceding paragraph to form a single unit. This approach might result in a node containing a substantial amount of content, potentially exceeding the maximum token length for embedding models. This issue can be effectively managed by defining relationships that summarize the content of the connected node, labeled with the corresponding regulation sub-item. By calculating the similarity between the query and these relationships, the problem of content length can be significantly mitigated.

Furthermore, while the KG primarily stores text data, regulatory documents often contain a significant number of images, tables, and attachments. Fortunately, in Neo4j, each node can be instantiated and can support additional attributes. Leveraging this capability, each regulation sub-item node has been assigned "images" and "appendix" attributes, which store images, tables, templates, and attachments from the documents. When querying regulation-related questions, the corresponding images and appendix can be accessed. This approach effectively addresses the challenge of handling multimodal data in KG-based RAG.

2.3. Relational Database

RBDs [21] are a type of database based on relations (*i.e.*, tables) and utilize mathematical concepts and methods such as relational algebra to process data. They are characterized by a clear structure, support for Structured Query Language (SQL), and consistency and integrity in data management. MySQL [22] is an open-source Relational Database Management System (RDBMS) that operates in a client/server model, providing efficient, reliable, and stable data storage and management services.

Here, MySQL is employed to build an enterprise-level database that stores information such as intellectual property, contracts, projects, employees, and departments. The database is divided into multiple sub-databases. Specifically, three sub-databases are constructed to store intellectual property, projects, and other information separately. Queries are directed to the appropriate sub-database based on the classification of input questions. The table information for sub-databases is detailed in **Tables 1-3**.

Table 1. Table information of database for intellectual properties.

Table name	Field Name	Field type
Patent	Patent_ID	Varchar
	Patent_type	Varchar
	Application_number	Varchar
	Patent_name	Varchar

Continued

Patent	Filing_date	Date
	Public_date	Date
	Patent_status	Varchar
	Patent_owner	Varchar
	Designer	Varchar
Software_copyright	Software_copyright_ID	Varchar
	Software_copyright_name	Varchar
	Certificate_number	Varchar
	Registration_number	Varchar
	Registration_agency	Varchar
	Copyright_owner	Varchar
	Development_completion_date	Date
	First_publication_date	Date
	Obtaining_method	Varchar
	Right_scope	Varchar

Table 2. Table information of database for contracts.

Table name	Field Name	Field type
Contract	Contract_ID	Varchar
	Contract_name	Varchar
	Contract_number	Varchar
	Signing_date	Date
	Effective_date	Date
	Company_ID_of_Party_A	Varchar
	Company_ID_of_Party_B	Varchar
	Contract_type	Varchar
	Contract_amount	Float
	Tax_rate	Float
	End_date	Date
	Payment_method	Varchar
Company	Company_name	Varchar
	Company_ID	Varchar

Table 3. Table information of database for other information.

Table name	Field Name	Field type
Person	Person_ID	Varchar
	Person_name	Varchar

Continued

Person	Gender	Varchar
	Famous_clan	Varchar
	Date_of_birth	Date
	Age	Integer
	Political_outlook	Varchar
	Native_place	Varchar
	ID_number	Varchar
	Job_ID	Varchar
	Phone_number	Varchar
	E_mail	Varchar
	Highest_education_level	Varchar
	Graduation_time	Date
	Graduation_school	Varchar
	Highest_professional_title	Varchar
	Department_ID	Varchar
	Rank	Varchar
	Date_of_employment	Date
	Date_of_confirmation	Date
Department	Department_name	Varchar
	Department_ID	Varchar
Project	Project_name	Varchar
	Project_type	Varchar
	Project_ID	Varchar
	Applied_department_ID	Varchar
	Total_budget	Float
	Application_date	Date
	Project_manager_ID	Varchar
	Project_approval_date	Date
Relationship_between _project_and_person	Relationship_ID	Varchar
	Project_ID	Varchar
	Person_ID	Varchar
	Enter_date	Date
	Project_team	Varchar
Person-responsibilities	Relationship_ID	Varchar
	Duty	Varchar

3. Retrieval-Augmented Generation-Based Q&A System

As shown in **Figure 3**, Qwen2.5-72B [23] is employed in the system.

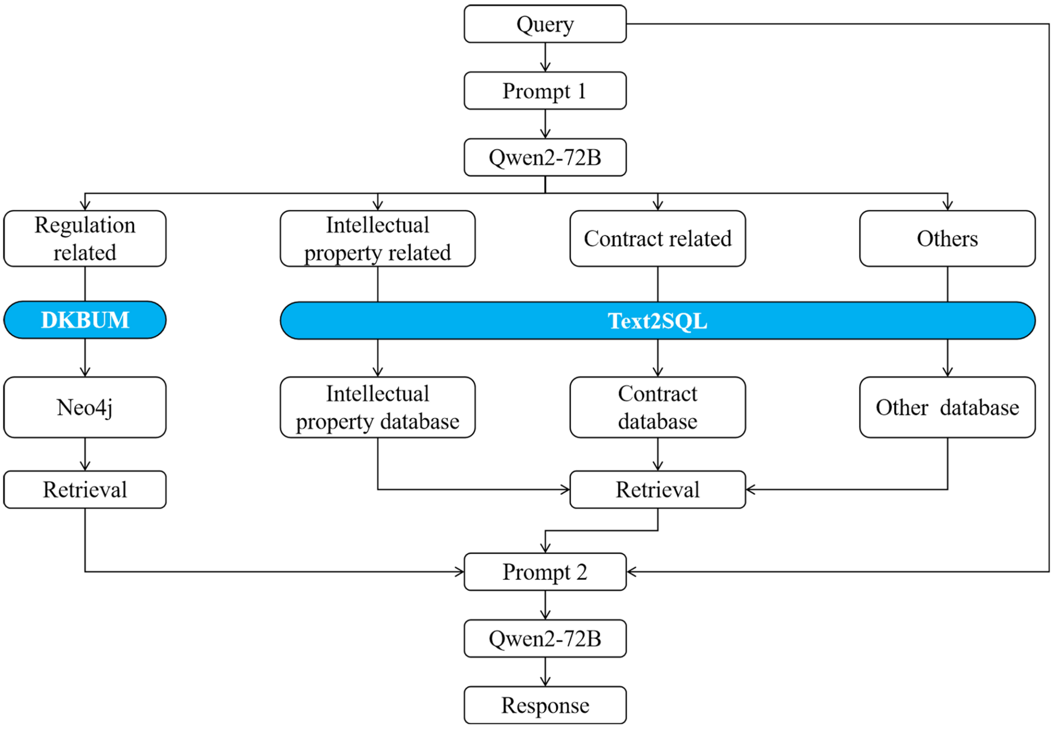

Figure 3. RAG-based system architecture.

Step 1. The input query is classified into various categories: regulation-related, intellectual property-related, contract-related, and others, using Qwen2.5-72B with prompt engineering. The prompt (Prompt 1 as shown in Figure 3) is structured as follows:

…

You are an NLU expert. Analyze the question "{query}" carefully, and categorize it into a specific category. Categories include regulation-related, intellectual property-related, contract-related, and others.

…

Step 2. Relevant knowledge of regulation-related queries is retrieved from Neo4j using the proposed DKBUM, while knowledge for other categories is retrieved from MySQL utilizing Text-to-SQL (Text2SQL).

Step 3. A prompt is generated based on the retrieved knowledge and the input query, and Qwen2.5-72B is used to summarize the corresponding answer.

3.1. Prompt Engineering-Based Text2SQL

Text2SQL is a Natural Language Processing (NLP) technique that facilitates efficient retrieval of information from RDBs by converting natural language queries into SQL statements. Traditional Text2SQL methods often involve training a model to perform the conversion between text and SQL. LLMs, such as Qwen2.5-72B, are trained on extensive datasets that include dialogue data between natural language and SQL, enabling them to accurately map queries to the corresponding SQL statements [24].

Here, prompt engineering is employed for Text2SQL tasks using Qwen2.5-72B. The prompt is structured as follows:

…

You are a database expert skilled in SQL retrieval and queries. Given the input question "{query}", use the following table information to create a syntactically correct SQL query: {table information}.

…

The table information includes detailed field names and field types within MySQL, along with the first three rows of data as illustrative examples.

3.2. Dynamic Knowledge Base Updating Mechanism-Based Neo4j Retrieval

As shown in Figure 3, Neo4j is primarily used to store regulatory documents. Neo4j supports Cypher, a query language similar to SQL. Although the training datasets of LLMs contain natural language and Cypher dialogues, theoretically enabling the implementation of a Text-to-Cypher (Text2Cypher) task, the structure of graph databases is more complex than that of RDBs, with more diverse relationships and attributes. This complexity leads to lower accuracy in Text2Cypher tasks. Therefore, the similarity of natural language word embeddings is employed to retrieve relevant knowledge from Neo4j in response to an input query.

Embedding is an NLP and ML technique that converts high-dimensional data, such as words or phrases, into lower-dimensional vectors. These vectors, often referred to as embeddings, capture the semantic meaning and relationships of the data within a continuous vector space. The primary goal of embeddings is to transform categorical data, which is typically sparse and high-dimensional, into dense, real-valued vectors that can be easily processed by machine learning algorithms.

Moreover, as mentioned in Section 1, issues related to the coordination and conflicts between new and old data in databases affect the response accuracy of RAG. To address this, the DKBUM is introduced. Additionally, a novel method for calculating the similarity of natural language, called Weighted Context-Aware Similarity (WCAS), is proposed to improve context-awareness when evaluating similarities.

3.2.1. Dynamic Knowledge Base Updating Mechanism

The proposed DKBUM continuously adapts and refines a knowledge base in real-time, ensuring that it reflects the most recent and relevant information. Unlike static knowledge bases, which remain fixed once populated, DKBUM enables the dynamic adjustment of knowledge item weights based on the introduction of new data or evolving contexts. This process involves evaluating the relevance of incoming information to existing knowledge items, followed by updating their associated weights to reflect their current importance. By incorporating real-time updates, DKBUM enhances the accuracy and relevance of knowledge retrieval, making it particularly valuable in domains that experience frequent changes, such as finance, healthcare, and legal fields. The mechanism improves decision-making by prioritizing the most up-to-date knowledge, thus providing a more contextually aware and responsive system.

Assume the database K consists of a set of knowledge items $\{k_1, k_2, \cdots, k_n\}$, each associated with a weight w_i. The weight vector at the initial time t_0 is represented as:

$$\mathbf{w}(t_0) = \{w_1(t_0), w_2(t_0), \cdots, w_n(t_0)\} \tag{4}$$

The relevance score of the new data D_{new} to each knowledge item k_i is defined as $\rho_i(D_{new})$, where $\rho_i(D_{new}) \in [0, 1]$ reflects the degree of association between the new data and the knowledge item.

Here, Bayesian inference [25] is employed to update weights. The updated weight $w_i(t_1)$ after receiving new data is given by:

$$w_i(t_1) = \frac{P(D_{new} \mid k_i) \cdot w_i(t_0)}{P(D_{new})} \tag{5}$$

where $P(D_{new}|k_i)$ represents the conditional probability of the new data being associated with the knowledge item k_i, typically estimated based on the relevance score $\rho_i(D_{new})$.

Assuming the new data is independent and identically distributed, Equation (5) can be simplified as:

$$w_i(t_1) = \frac{\rho_i(D_{new}) \cdot w_i(t_0)}{\sum_{j=1}^{n} \rho_j(D_{new}) \cdot w_j(t_0)} \tag{6}$$

Equation (6) indicates that the impact of new data on each knowledge item is relative, with weight adjustments depending on the item's relevance to the new data and its initial weight.

Furthermore, to ensure that the retrieved knowledge base remains relevant and up-to-date, a time decay factor $\lambda(t)$ is employed. This factor allows the weights to naturally decrease over time, reducing the influence of outdated information. The temporal evolution of weights can be expressed as:

$$w_i(t) = w_i(t_1) \cdot \exp(-\lambda(t - t_1)) \tag{7}$$

The optimization goal of the DKBUM is to ensure that the generated content aligns with the latest data while not completely disregarding historical knowledge. The loss function to balance the accuracy and timeliness of generated content can be represented as:

$$L(\mathbf{w}, D_{new}) = \sum_{i=1}^{n} \left[(1 - \rho_i(D_{new})) \cdot w_i(t) \right] + \alpha \cdot \sum_{i=1}^{n} \left(w_i(t) - w_i(t_0) \right)^2 \tag{8}$$

where α is a balancing parameter that controls the proportion of knowledge base updating versus retaining historical knowledge.

By optimizing Equation (8), the dynamically adjusted weight distribution can be obtained, allowing the model to generate responses that are both timely and accurate when encountering new data.

3.2.2. Weighted Context-Aware Similarity

Traditional methods like cosine similarity [26] and Maximal Marginal Relevance (MMR) [27] have been widely used, yet they often fall short in capturing the nuanced, context-dependent semantics inherent in human language. A novel WCAS method is proposed to address this limitation.

WCAS is an advanced metric for measuring semantic similarity between text pairs, which incorporates both contextual information and the relative importance of different components within the texts. Unlike traditional measures, it adapts to the nuanced meaning of words based on their surrounding context, thus capturing the dynamic relationships between terms in varying linguistic environments. By assigning weights to different contextual features based on their relevance, WCAS refines similarity computations, emphasizing more pertinent information while downplaying less relevant elements. This approach enhances the sensitivity of similarity measurements to contextual shifts and domain-specific requirements, making it particularly effective for complex tasks in natural language processing.

WCAS leverages Bidirectional Encoder Representations from Transformers (BERT) [9] to calculate the natural language word embeddings. Given two texts *A* and *B*, each word in them is transformed into a dense vector representation. Let $\mathbf{v}_i^A$ and $\mathbf{v}_j^B$ represent the embeddings of the *i*-th word in text *A* and the *j*-th word in text *B*, respectively.

To capture the fine-grained semantic similarity between the two texts, the similarity ***S*** is calculated.

$$S_{ij} = \frac{\mathbf{v}_i^A \cdot \mathbf{v}_j^B}{\left\|\mathbf{v}_i^A\right\|\left\|\mathbf{v}_j^B\right\|} \tag{9}$$

where S_{ij} is the cosine similarity between $\mathbf{v}_i^A$ and $\mathbf{v}_j^B$. The matrix ***S*** encapsulates the pairwise similarity between every word in text *A* and every word in text *B*, providing a comprehensive comparison of their semantic content.

Importantly, WCAS introduces context-aware weights that modulate the contribution of each word pair in the final similarity score. These weights are derived from the attention mechanisms within the BERT model, which naturally highlight the importance of different words based on their context. The context-aware weights for the words in texts *A* and *B* are defined as α_i^A and β_j^B, respectively. The context-aware weights are computed by:

$$\alpha_i^A = \frac{\exp\left(\gamma \cdot \text{context}_i^A\right)}{\sum_{k=1}^{n} \exp\left(\gamma \cdot \text{context}_k^A\right)}, \beta_j^B = \frac{\exp\left(\gamma \cdot \text{context}_j^B\right)}{\sum_{k=1}^{m} \exp\left(\gamma \cdot \text{context}_k^B\right)} \tag{10}$$

where γ is a hyperparameter that controls the sensitivity to context and context_i^A represents the contextual importance of the *i*-th word in text *A*, as derived from the BERT attention scores.

The WCAS score between texts is computed as the weighted sum of the similarity matrix, modulated by the context-ware weights.

$$WCAS\left(A,B\right)=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^A\cdot\beta_j^B\cdot S_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^A\cdot\beta_j^B} \tag{11}$$

WCAS ensures that the most contextually relevant word pairs contribute more significantly to the overall similarity score, leading to a more nuanced and semantically accurate measure.

Finally, the WCAS similarity score is multiplied by the weight of the corresponding knowledge item. The intuition here is that a high similarity score should be further amplified if the knowledge item is deemed highly relevant (*i.e.*, has a high weight) according to the DKBUM mechanism. Conversely, if a knowledge item has a lower weight due to being less relevant or outdated, its influence will be diminished even if the WCAS score is high.

$$score=WCAS\left(Q,k_i\right)\times w_i\left(t\right) \tag{12}$$

4. Experiments Validation

In this section, queries from various categories are used to test the performance of the proposed RAG-based Q&A system.

4.1. Regulation-Related Evaluation

As previously mentioned, LLMs can perform Text2Cypher tasks. The prompt is defined as follows:

…

You are an expert in knowledge graphs, proficient in retrieving and querying using Neo4j. Given the question "{input query}", create a syntactically correct Cypher query. The Neo4j database includes the following nodes and relationships: {nodes and relationships information}.

…

Node properties (including labels, names, and IDs) and relationships are provided as part of the nodes and relationships information.

For the query "Which department is responsible for the establishment of systems such as ISO9001 and CMMI5?", the generated Cypher query using Text2Cypher is:

…

```
MATCH k=(n1: regulation)-[*]->(n2)
WHERE n2.name CONTAINS 'ISO9001 and CMMI5'
RETURN k
```

…

When this Cypher query is executed in Neo4j, the retrieved knowledge chain is empty. This occurs because the relevant content in the KG is described as "ISO9001 and CMMI 5". The difference in spacing between "CMMI 5" and "CMMI5" causes the failure to match the relevant knowledge. This demonstrates the challenge of achieving high accuracy with Text2Cypher, due to the diversity and complexity of relationships and attributes, as discussed in Section

3.2.

To validate the effectiveness of the proposed DKBUM, a database is constructed using historical regulatory documents within the enterprise. These documents contain revisions, updates, and enhancements to various policies and regulations across different periods. The total number of documents is 132, and the constructed KG contains 4,494 nodes and 6,791 relationships. The static KG, namely the KG without using the DKBUM technique, is introduced as a baseline.

Here, the relevance score measures the proportion of correct retrievals out of the total number of queries. It indicates how often the knowledge item selected by the system is correct or relevant to the user's query. Relevance scores for the DKBUM-based KG and the baseline are 69.4% and 74.6%.

$$s_{relevance} = \frac{n_{correct\_retrievals}}{n_{total}} \tag{13}$$

Self-similarity calculates the correlation between each pair of knowledge entries in the retrieval results. The higher the self-similarity, the more similar content is included in the retrieval results. A smaller self-similarity is preferable, as it reduces the likelihood of retrieving different interpretations or explanations of the same content. As shown in **Table 4**, compared with the baseline, the DKBUM-based KG obtains a smaller self-similarity.

$$\mu = \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} S_{ij},\ \sigma^2 = \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} \left(S_{ij} - \mu\right)^2 \tag{14}$$

where S_{ij} is the cosine similarity calculated by Equation (9).

Table 4. Results of self-similarity.

	Mean	Standard deviation	Maximum
DKBUM-based KG	0.487	0.115	0.601
Baseline	0.590	0.128	0.778

Answer relevance [28] evaluates the quality of the generated answer by calculating the similarity between the answer and the question. It is defined as:

$$\text{AR} = \cos(\mathbf{q}, \mathbf{a}) = \frac{\sum_i (q_i \times a_i)}{\sqrt{\sum_i q_i^2} \times \sqrt{\sum_i a_i^2}} \tag{15}$$

where q_i and a_i are the word vectors representing the question and the answer, respectively.

121 Queries are generated by using an LLM. Their average answer relevance score is 0.893, with a standard deviation of 0.030. This suggests that the generated answers are highly relevant to the source information. Furthermore, the low standard deviation reflects the system's good stability and consistent performance across different queries.

4.2. Person, Project, and Intellectual Property-Related Evaluation

As mentioned in Section 3, knowledge related to persons, projects, and intellectual property is stored in MySQL. Knowledge retrieval is achieved through Text2SQL technology. Six queries are input for evaluation. As shown in Table 5, the system accurately classifies the questions and utilizes the corresponding database to generate SQL queries, ultimately retrieving the correct results. Even for complex queries involving multiple logical conditions, such as “Which male employees have a master’s degree or higher, work in the Technology Research Center, and have participated in projects?”, the system successfully generates a logically coherent SQL query to perform an effective search.

Table 5. Results of Text2SQL.

Query	Generated SQL

What projects has Yu Li participated in?	SELECT p.Project_name
	FROM Project p
	JOIN Relationship_between_project_and_person r ON p.Project_ID = r.Project_ID
	JOIN Person e ON r.Person_ID = e.Person_ID
	WHERE e.Person_name = “Yu Li”
	GROUP BY p.Project_name;

What is Yu Li’s job number?	SELECT Job_ID
	FROM Person
	WHERE Person_name = “Yu Li”;

How many employees have a bachelor’s degree or above?	SELECT COUNT(DISTINCT Person_ID)
	FROM Person
	WHERE Highest_education_level IN (“Bachelor”, “Master”, “Ph.D”);

Who are the employees with a master’s degree or higher and whose hometown is in Changsha?	SELECT *
	FROM Person
	WHERE Highest_education_level IN (“Master”, “Ph.D”)
	AND Native_place LIKE “%Changsha%”;

Which male employees have a master’s degree or higher, work in the Technology Research Center, and have participated in projects?	SELECT DISTINCT s.* FROM Person s JOIN Department a ON s.Department_ID = a.Department_ID JOIN Relationship_between_project_and_person pr ON s.Person_ID = pr.Person_ID WHERE s.Highest_education_level IN (“Master”, “Ph.D”) AND s.Gender = 'Male' AND a_super.Department LIKE “% Technology Research Center %”;

How many software and patents does the company have?	SELECT COUNT(*) AS software_copyright_count,
	(SELECT COUNT(*) FROM Patent) AS patent_count
	FROM Software_copyright;

5. Conclusions and Futher Works

In this study, a novel RAG-based framework for intelligent Q&A systems is proposed. Additionally, a DKBUM-based retrieval method is introduced to address issues of coordination and conflicts between new and old data in databases, which can affect the response accuracy of RAG. The key contributions are summarized as follows:

1) By integrating external databases with RAG, LLMs can effectively acquire domain-specific knowledge. This approach mitigates inherent shortcomings of GAI, such as hallucination tendencies and weak expertise, by providing more evidence-based and fact-reliant information. In this study, an RDB and a graph database are employed to help LLMs acquire domain-specific knowledge without requiring additional fine-tuning.

2) Through prompt engineering, LLMs can effectively support Text2SQL tasks. By leveraging Text2SQL technology, RDBs can significantly enhance LLMs' ability to access and utilize specialized knowledge.

3) The DKBUM and WCAS methods are introduced to improve retrieval accuracy in Neo4j-based knowledge systems. The DKBUM dynamically adjusts the weights of knowledge items in the database, prioritizing the most recent and relevant information while balancing it with historical knowledge. This approach ensures that the generated content remains both timely and accurate. Meanwhile, WCAS computes a nuanced similarity score between natural language queries and knowledge items with an enhanced understanding of context. By integrating these context-aware similarity scores with the dynamically updated weights, the retrieval process achieves a more refined and context-sensitive alignment with the input query.

While the proposed DKBUM and WCAS provide significant improvements, there are some inherent limitations related to their broader applicability. The approach may face challenges when deployed in extremely large-scale knowledge bases, where the computational cost of continuous updates and similarity calculations could increase. Additionally, the effectiveness of the time decay factor in DKBUM may depend on domain-specific characteristics, necessitating careful calibration to ensure optimal knowledge relevance over time.

Further work will be focused on enhancing the scalability of DKBUM by exploring more efficient update algorithms and distributed systems. Simultaneously, fine-tuning models for specific domains will be attempted to improve WCAS's precision in specialized fields.

Acknowledgments

The authors would like to express their sincere gratitude to Datang Xianyi Technology Co. Ltd. for providing the necessary resources, including access to proprietary software and data, which were crucial for conducting this research.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Wang, J., Wang, X., Ma, C. and Kou, L. (2020) A Survey on the Development Status and Application Prospects of Knowledge Graph in Smart Grids. *IET Generation, Transmission & Distribution*, **15**, 383-407. https://doi.org/10.1049/gtd2.12040

[2] Liu, R., Fu, R., Xu, K., Shi, X. and Ren, X. (2023) A Review of Knowledge Graph-Based Reasoning Technology in the Operation of Power Systems. *Applied Sciences*, **13**, Article No. 4357. https://doi.org/10.3390/app13074357

[3] Ding, H., Qiu, Y., Yang, Y., Ma, J., Wang, J. and Hua, L. (2021) A Review of the Construction and Application of Knowledge Graphs in Smart Grid. 2021 *IEEE Sustainable Power and Energy Conference* (*iSPEC*), Nanjing, 23-25 December 2021, 3770-3775. https://doi.org/10.1109/ispec53008.2021.9736038

[4] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K. and Ray, A. (2022) Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems*, Vol. 35, 27730-27744.

[5] Huang, H., Xu, B., Liang, X., Chen, K., Yang, M., Zhao, T., *et al*. (2024) Multi-View Fusion for Instruction Mining of Large Language Model. *Information Fusion*, **110**, Article ID: 102480. https://doi.org/10.1016/j.inffus.2024.102480

[6] Wu, Z. (2024) Large Language Model Based Semantic Parsing for Intelligent Database Query Engine. *Journal of Computer and Communications*, **12**, 1-13. https://doi.org/10.4236/jcc.2024.1210001

[7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., *et al*. (2020) Language Models Are Few-Shot Learners. *Advances in Neural Information Processing Systems*, Vol. 33, 1877-1901.

[8] Müller, M. and Laurent, F. (2022) Cedille: A Large Autoregressive French Language Model.

[9] Devlin, J. (2018) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding.

[10] Bender, E.M., Gebru, T., McMillan-Major, A. and Shmitchell, S. (2021) On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the* 2021 *ACM Conference on Fairness*, *Accountability*, *and Transparency*, 3-10 March 2021, 610-623. https://doi.org/10.1145/3442188.3445922

[11] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., *et al*. (2021) On the Opportunities and Risks of Foundation Models.

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., *et al*. (2020) Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, Vol. 33, 9459-9474.

[13] Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R. and Nanayakkara, S. (2023) Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Transactions of the Association for Computational Linguistics*, **11**, 1-17. https://doi.org/10.1162/tacl_a_00530

[14] Han, R., Zhang, Y., Qi, P., Xu, Y., Wang, J., Liu, L., *et al*. (2024) RAG-QA Arena: Evaluating Domain Robustness for Long-Form Retrieval Augmented Question Answering. *Proceedings of the* 2024 *Conference on Empirical Methods in Natural Language Processing*, Miami, November 2024, 4354-4374. https://doi.org/10.18653/v1/2024.emnlp-main.249

[15] Kim, K. and Lee, J. (2024) RE-RAG: Improving Open-Domain QA Performance and

Interpretability with Relevance Estimator in Retrieval-Augmented Generation. *Proceedings of the* 2024 *Conference on Empirical Methods in Natural Language Processing*, Miami, November 2024, 22149-22161. https://doi.org/10.18653/v1/2024.emnlp-main.1236

[16] Zhong, B., He, W., Huang, Z., Love, P.E.D., Tang, J. and Luo, H. (2020) A Building Regulation Question Answering System: A Deep Learning Methodology. *Advanced Engineering Informatics*, **46**, Article ID: 101195. https://doi.org/10.1016/j.aei.2020.101195

[17] Chen, Z., Wan, Y., Liu, Y. and Valera-Medina, A. (2024) A Knowledge Graph-Supported Information Fusion Approach for Multi-Faceted Conceptual Modelling. *Information Fusion*, **101**, Article ID: 101985. https://doi.org/10.1016/j.inffus.2023.101985

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) Attention Is All You Need. *NIPS'*17: *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.

[19] Pu, H., Hu, T., Song, T., Schonfeld, P., Wan, X., Li, W., *et al.* (2024) Modeling and Application of a Customized Knowledge Graph for Railway Alignment Optimization. *Expert Systems with Applications*, **244**, Article ID: 122999. https://doi.org/10.1016/j.eswa.2023.122999

[20] Fernandes, D. and Bernardino, J. (2018) Graph Databases Comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. *Proceedings of the* 7*th International Conference on Data Science*, *Technology and Applications*, Porto, 26-28 July 2018, 373-380. https://doi.org/10.5220/0006910203730380

[21] Codd, E.F. (2007) Relational Database: A Practical Foundation for Productivity. In: *ACM Turing Award Lectures*, ACM, 1981. https://doi.org/10.1145/1283920.1283937

[22] MySQL, A. (2001) MySQL.

[23] Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., *et al.* (2024) Qwen2 Technical Report.

[24] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., *et al.* (2024) A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science*, **18**, Article ID: 186345. https://doi.org/10.1007/s11704-024-40231-1

[25] Bois, F.Y. (2012) Bayesian Inference. In: Reisfeld, B. and Mayeno, A.N., Eds., *Computational Toxicology*, Humana Press, 597-636. https://doi.org/10.1007/978-1-62703-059-5_25

[26] Thongtan, T. and Phienthrakul, T. (2019) Sentiment Classification Using Document Embeddings Trained with Cosine Similarity. *Proceedings of the* 57*th Annual Meeting of the Association for Computational Linguistics*: *Student Research Workshop*, Florence, July 2019, 407-414. https://doi.org/10.18653/v1/p19-2057

[27] Gunawan, D., Harahap, S.H. and Fadillah Rahmat, R. (2019) Multi-Document Summarization by Using Textrank and Maximal Marginal Relevance for Text in Bahasa Indonesia. 2019 *International Conference on ICT for Smart Society* (*ICISS*), Bandung, 19-20 November 2019, 1-5. https://doi.org/10.1109/iciss48059.2019.8969785

[28] Es, S., James, J., Espinosa-Anke, L. and Schockaert, S. (2023) Ragas: Automated Evaluation of Retrieval Augmented Generation.