Knowledge Graph Path-Enhanced RAG for Intelligent Residency Q&A

Jian Zhu, Huajun Zhang*, Jianpeng Da, Hanbing Huang, Chongxin Luo, Xu Peng
School of Automation, Wuhan University of Technology, Wuhan 430070, China

***Abstract*—As the demand for efficient information retrieval in specialized domains continues to rise, vertical domain question-answering systems play an increasingly important role in addressing domain-specific knowledge needs. This paper proposes a retrieval-augmented generation method that integrates path search in knowledge graphs to enhance intelligent question-answering systems for professional information retrieval. The proposed approach leverages fine-tuned large language models to identify entities and extract relations from user queries, combining pruned marker method with a shortest path generation tree algorithm to efficiently retrieve relevant information. The retrieval results are then integrated with user queries using prompt engineering to generate precise and contextually relevant answers. To validate the practicality of the proposed method, this paper develops a knowledge graph encompassing policies, regulations, and social services within the household registration vertical domain. The experimental results within this vertical domain reveal that the proposed method significantly outperforms existing methods in terms of evaluation metrics such as BLEU, ROUGE, and METEOR, achieving improvements exceeding 3%. Furthermore, ablation experiments validate the importance of combining path search algorithms with fine-tuning techniques in enhancing the question-answering performance.**

***Keywords*—Retrieval-augmented generation; path search; knowledge graph; household registration policy vertical field**

I. Introduction

With the rapid development of natural language processing (NLP) technology, intelligent question-answering systems have been widely applied in the field of policy interpretation and are gradually becoming important tools for obtaining information in medical, legal, educational, and other professional fields. These systems understand user questions through semantic analysis and information extraction technology, and generate relevant answers from knowledge bases or domain-specific databases [1]. Compared with traditional keyword matching, intelligent question-answering systems can more accurately capture the semantic intent of the user, thereby improving the accuracy of the answers and user experience [2]. However, intelligent question-answering systems still face notable limitations. For instance, although pre-trained language models exhibit strong performance across various tasks, they remain less effective in addressing queries that demand a high level of domain-specific expertise, such as medical diagnoses or legal code interpretation [3]. Additionally, due to the system's dependence on static knowledge bases, it is difficult to update domain knowledge in real-time, which may lead to delays in answers [4].

The research on intelligent question-answering systems typically unfolds along multiple technical routes, which mainly include rule-based question-answering systems, information retrieval-based question-answering systems, and generation model-based question-answering systems. Rule-based question-answering systems rely on pre-defined rules and templates, matching answers through the parsing of keywords and syntactic structures. They perform well in structured, fixed-pattern questions, but they have limitations in addressing diverse needs and complex issues [5]. Information retrieval-based question-answering systems find answers by searching for relevant documents, utilizing keyword matching and semantic retrieval techniques to quickly locate content related to the user's question [6]. Their advantage lies in their ability to handle large document repositories and providing diverse sources of answers. However, they often struggle to meet the needs for questions that require complex reasoning or the generation of new answers. In comparison, generation model-based systems directly apply natural language generation technology to generate answers, performing even better in handling complex issues [7].

In recent years, Retrieval-Augmented Generation (RAG) technology has attracted widespread attention due to its combination of the advantages of information retrieval and generation models. RAG first finds relevant documents through information retrieval and then generates more accurate and semantically rich answers based on these documents citeyang2019xlnet. Compared to traditional generation models, RAG maintains the flexibility and generation capability of the latter while ensuring the contextual relevance of the answers [8]. RAG models combine a retrieval module based on BERT with a GPT generation module, demonstrating outstanding performance in knowledge-intensive tasks [9]. Research shows that RAG not only improves the accuracy of answers but also effectively reduces the lag caused by outdated information by retrieving the latest documents [10]. RAG has shown significant effects in open-domain question-answering tasks, especially in areas such as law and policy where knowledge updates are frequent, ensuring timeliness of answers through real-time retrieval [11]. Additionally, multimodal RAG systems that combine image and text data further expand the application of the technology and enhance its ability to handle complex tasks[12]. RAG models have significant advantages in handling complex policy-related questions, especially in cross-domain and multi-dimensional issues, significantly enhancing the relevance and depth of the answers [13].

In the context of the current complex policy system and the growing demand for public services, how to efficiently acquire and utilize knowledge of the household registration policy has become one of the core issues of concern for policymakers and public service institutions [14] [15] [16]. As an important part of social governance, the household registration policy have a

wide-ranging impact that covers multiple areas such as social security, education, and employment[17]. Although RAG technology has made significant progress in question-answering systems, it still faces some critical challenges. RAG technology has a high dependency on the quality of the retrieval stage; if the retrieval results include irrelevant documents, the generated answers are often not accurate; in multi-hop reasoning tasks, insufficient information is more likely to lead to incomplete answers [18]. Additionally, when handling long texts, RAG technology often suffers from context loss, leading to answers that lack coherence [19]. Furthermore, RAG technology has a high computational cost, and the system response speed is slower when processing large document libraries, making it difficult to meet the needs of real-time applications[20].

The above challenges have prompted the development of a retrieval-enhanced generation method that incorporates knowledge graph path search, which has been applied to the household registration vertical field. This method combines knowledge graph path search technology with entity recognition performed by a fine-tuned large model, mapping user queries to corresponding entities and relationships in the graph and using path algorithms for inference to precisely extract relevant knowledge information. Finally, the large model is used for natural language generation to provide answers that are contextually appropriate. In addition, in the subsequent application phase, this paper will construct a knowledge graph that covers policies, regulations, and social services in the household registration vertical field, and further discuss its specific implementation and application effects in vertical field question-answering systems. The primary contributions of this paper are as follows.

1) Proposal of an MST-based algorithm for knowledge extraction: This paper presents a novel Minimum Spanning Tree (MST) algorithm based on the shortest-path methodology to facilitate efficient knowledge extraction from graphs. When integrated with a large language model for natural language generation, this approach enables the system to deliver accurate, contextually relevant policy responses to user inquiries. This enhancement significantly improves the intelligence and responsiveness of the question-answering system.

2) Fine-tuned model for entity recognition and relationship extraction: For tasks involving entity recognition and relationship extraction within the household registration domain, this study employs fine-tuning techniques on large models, coupled with the knowledge graph, to achieve robust entity recognition and relationship inference. This method effectively processes complex entities and multi-layered relationships within the policy domain, ensuring that user queries are accurately aligned with pertinent knowledge points in the graph.

3) Development of a specialized knowledge graph for household registration: A comprehensive knowledge graph tailored to the household registration domain has been developed, incorporating policy content, legal regulations, and related social services. The graph systematically organizes the interconnections among various policy clauses and entities, providing a structured foundation for addressing complex policy queries. This framework significantly improves the system's accuracy in addressing issues related to household registration.

The rest of this paper is organized as follows. Section II describes the related work of this paper. Section III introduces the retrieval-enhanced generation technique, utilizing knowledge graph path search to improve query handling. Section IV details the construction of a knowledge graph within the civil affairs domain and demonstrates the application of this technique to enhance information retrieval and response generation within civil affairs contexts. As for Section V, we conduct comprehensive experiments and sufficient analysis. Finally, Section VI provides a summary of the entire paper.

II. Related Work

Knowledge graphs can better capture semantic relationships in complex domains by structuring the representation of entities and their relationships, thus providing more accurate retrieval cues and context information for RAG technology [21]. In the field of knowledge graphs, early research focused mainly on how to effectively extract entities and relationships from text data to build a structured knowledge base. For example, methods based on statistical models can automatically identify and associate relevant entities from multiple data sources, laying the foundation for the construction of knowledge graphs [22]. However, these traditional methods often rely on predefined rules and templates, and their ability to handle complex semantic relationships is relatively limited. To overcome this limitation, research on knowledge graph completion techniques based on deep learning has gained widespread attention in recent years. Graph neural networks (GNNs), in particular, have been widely applied due to their superior performance in handling complex entity relationship graph structures [23]. In addition, multi-task learning techniques have also begun to be introduced into the construction of knowledge graphs to handle entity recognition and relation extraction simultaneously. Compared to single-task learning, multi-task learning can better utilize the correlations between different tasks, thereby improving the overall performance of the model. For example, through a multi-task learning framework, joint training of entity recognition and relation classification has been achieved, effectively improving the accuracy and efficiency of knowledge graph construction [24]. However, since knowledge graphs are a semantic network that dynamically updates, how to achieve efficient knowledge updating and completion remains an important research challenge [25]. At the same time, with the rapid development of large-scale pre-trained language models (LLMs), the integration of knowledge graphs with LLMs has become a new research direction. Knowledge graphs can provide structured knowledge support for LLMs, thereby improving accuracy in specific domain question-answering systems [26]. For example, by combining knowledge graphs with LLMs, the model's explainability and personalized recommendation capabilities can be enhanced [27]. However, since LLMs are essentially a "black box" model, the issue of how to utilize its powerful capabilities while maintaining the transparency and explainability of the system remains a pressing problem [28].

The Retrieval-Augmented Generation (RAG) technique, which integrates knowledge graphs with pre-trained language models, significantly improves the accuracy and consistency of generated content by introducing structured knowledge into the generative process. For example, Yasunaga et al. proposed the QA-GNN model, a framework that combines

language models with knowledge graphs to handle complex question-answering tasks. This model leverages path reasoning within the knowledge graph, thereby enhancing the logical consistency of the generated responses [29]. Similarly, Hu et al. developed the OREO-LM model, which connects pre-trained language models with knowledge graph reasoning modules via a knowledge interaction layer, resulting in improved performance on question-answering tasks [30]. In dialogue generation, advancements have been made by incorporating structured knowledge from graphs. Xu et al., for instance, introduced the DMKCM model, which fuses structured knowledge graphs with unstructured textual data to produce dialogue with greater depth and coherence [31]. In a related development, Kang et al. proposed the SURGE framework, which retrieves relevant subgraphs and employs contrastive learning during generation, enhancing the quality and knowledge consistency of the generated dialogues [32].

Significant strides in dialogue generation have been achieved by integrating knowledge graphs with pre-trained language models. Peng et al. introduced the GODEL model, which incorporates external knowledge during pre-training to improve performance across task-oriented dialogue, conversational question-answering, and open-domain dialogue. This model outperforms existing pre-trained dialogue models, particularly in few-shot fine-tuning scenarios [33]. Yang et al. further advanced the field with the Graph Dialog model, which embeds structural knowledge from graphs into an end-to-end task-oriented dialogue system. By utilizing graph convolutional networks (GCNs), Graph Dialog captures structural information from both dialogue history and knowledge bases, thereby generating more precise responses [34].

III. Retrieval-Augmented Generation Based on Knowledge Graph Path Search

The retrieval-enhanced generation method based on knowledge graph path search includes three modules: entity recognition, path search, and answer generation. In the entity recognition module, a large language model is used to analyze the user's question and extract a set of entities similar to the knowledge graph nodes. In the path search module, the identified entities are linked to knowledge graph nodes, and the shortest path is found using a 2-hop coverage query, generating a tree structure that covers all entity nodes. Finally, in the answer generation module, the results of the path search are input into a large language model along with the user's question. Through prompt engineering, the model generates and returns the final responses. The overall workflow diagram of the retrieval-enhanced generation method based on knowledge graph path search is illustrated in Fig. 1.

A. Entity Recognition Module

In this paper, a large language model is employed to conduct an in-depth analysis of user input queries, extracting a set of entities related to nodes within a knowledge graph. This process leverages advanced natural language processing techniques and harnesses the semantic understanding capabilities of the model, enabling accurate identification of essential entities within user queries. These entities, typically exhibiting strong correlations with knowledge graph nodes, form a crucial foundation for subsequent retrieval and reasoning tasks.

The user query Q is initially received as input, upon which the model conducts multi-level semantic parsing and entity mapping to identify a potential set of entities $E = [E_1, E_2, \ldots, E_n]$. This set is derived from relevant nodes within the knowledge graph, guided by the context of the user query and its implicit semantic cues. Through this entity recognition method, the model not only extracts entities directly pertinent to the user query but also captures potential implicit information, enhancing the alignment between the query and the knowledge graph. This process can be represented as follows:

$$E = LLM\,(Q) \tag{1}$$

Prompt Template for Household Registration Information Extractor:

[Prompt]: You are an expert in household registration information extraction, skilled in analyzing provided sentences and extracting specified household registration items. Please note that the output should only include the household registration items mentioned in the text, and they should be numbered sequentially. The input sentence may correspond to multiple household registration items; do not output any extraneous information. The household registration items include: 1) Proof of legal and stable residence. 2) Correction of place of birth.

[Input Text]: I want to register my parents' household registration under my name. What materials are required?

[Output Keywords]: 1) Parental relocation to child's household registration

Through the entity recognition and retrieval stage, the model effectively minimizes the semantic gap between the user's query and the nodes within the knowledge graph, thereby improving the accuracy of subsequent knowledge graph queries and information reasoning. This process deepens the system's comprehension of the user's query and establishes a more precise entity foundation for path search and information integration during the generation stage. Consequently, it enhances both the quality of the system's responses and the overall user experience.

B. Path Search Module

In this paper, a knowledge graph is constructed for a specialized domain, transforming unstructured or semi-structured texts related to domain knowledge into structured information. This information is stored within a graph structure to provide reliable data support for the subsequent question-answering system. The NEO4J graph database is utilized to store multi-relational data, leveraging Cypher queries for data import and graph data retrieval. Cypher, a descriptive graph query language, is recognized for its simplicity and robust functionality. Furthermore, NEO4J offers the neo4j-import tool, which facilitates the rapid import of large-scale data.

Given the large number of nodes and relationships in the constructed household registration policy knowledge graph, directly executing path searches on the graph proves time-consuming, which adversely affects user experience. Additionally, storing the distances and paths between each pair of nodes offline demands excessive memory space. To address this issue,

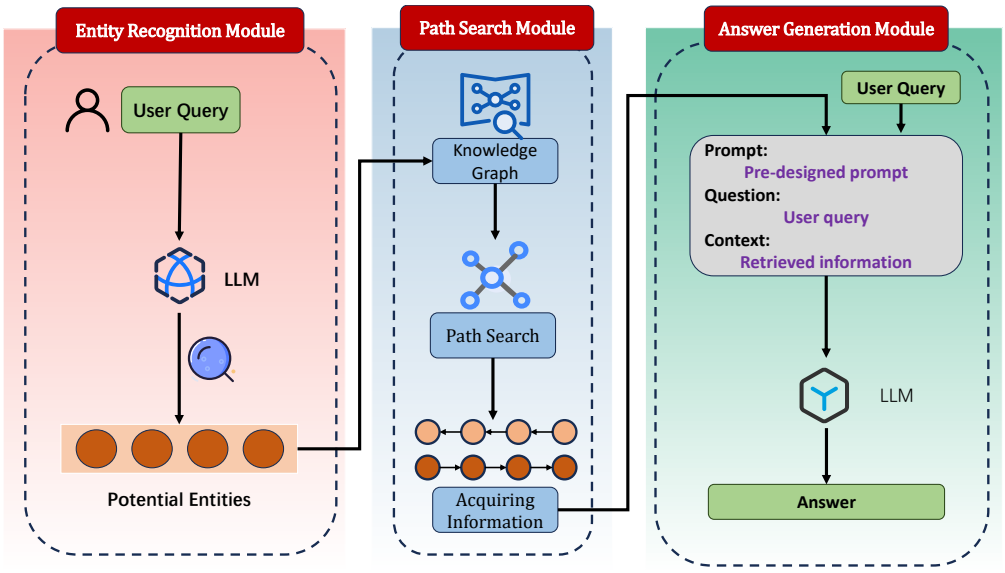

Fig. 1. System workflow diagram.

this paper implements a pruning labeling strategy. During the preprocessing stage, labels are assigned to nodes within the graph, allowing these labels to be utilized at query time to swiftly determine the shortest path between node pairs. Since relationships in the knowledge graph are unweighted, an in-depth analysis was performed on an undirected, unweighted graph. Table I provides explanations of the commonly used symbols in this paper.

1) Definition of shortest distance and shortest path queries: In the knowledge graph, a set of composite labels, denoted as $L(u)$, must be precomputed for each node u. The label set $L(u)$ comprises multiple label pairs, each represented as a triplet (v, d_{uv}, p_{uv}), where v is a node, d_{uv} denotes the shortest distance from node u to node v, and p_{uv} represents the set of all nodes along the shortest path between u and v.The relationships in the knowledge graph are not assigned weights, so the shortest distance between two nodes is represented by the number of nodes in the shortest path between them.

When querying the shortest path between nodes s and t, the shortest distance between s and t is first calculated. To facilitate this, a function $QD(s,t,L)$ is defined, with its specific form as Eq. (2). u represents a common node between s and t. For each common node u, the distance from node s to node t is calculated as $d_{ut} + d_{us}$, with the minimum value among all possible distances considered as the shortest distance.If no common nodes exist between $L(s)$ and$L(t)$, $QD(s,t,L)$ is defined as $+\infty$.

In addition to calculating the shortest distance between nodes s and t,, it is essential to record the path information. To accomplish this, a function $QP(s,t,L)$ is defined to retrieve the shortest path, with its specific definition as Eq. (3). $QP(s,t,L)$returns the path $p_{us} \cup p_{ut}$, representing the shortest path between nodes s and t. This path is constructed by combining the optimal subpaths from node s to the common node u and from u to node t.

For any pair of nodes s and t, the shortest distance between them can be determined by calculating $QD(s,t,L)$, from which the shortest path can then be derived.

2) Generation of composite labels: A composite label generation algorithm based on pruned breadth-first search (BFS) is proposed to optimize the computational efficiency of traditional BFS. Similar to the naive approach, this algorithm performs searches sequentially in the order of vertices $v_1, v_2, \ldots, v_n$. The process begins with an empty index L'_0, and after each pruned BFS iteration from vertex v_k, it updates the index L'_{k-1} to a new index L'_k based on the information gathered. Algorithm 1 outlines the specific steps involved in this composite label generation algorithm.

Algorithm 1 Composite Label Construction Algorithm

$L_k(v) \leftarrow \emptyset$ for all $v \in V$
for $i \leftarrow 1$ **to** n **do**
  $L_i(v) \leftarrow L_{i-1}(v)$ for all $v \in V$
  $distance(v_i) \leftarrow 0$; $distance(v) \leftarrow \infty$ for all $v \in V \setminus \{v_i\}$
  $path(v_i) \leftarrow [v_i]$; $path(v) \leftarrow \emptyset$ for all $v \in V \setminus \{v_i\}$
  $Q \leftarrow$ a queue with only one element v_i
  while Q is not empty **do**
    $u \leftarrow Q$.pull()
    if $d_G(u) < QD(v_i, u, L_{i-1}(v))$ **then**
      $L_i(u) \leftarrow L_{i-1}(v) \cup \{v_i, distance(u), path(u)\}$
      for $w \in$ Neighbors(u) **do**
        $distance(w) \leftarrow distance(u) + 1$
        $path(w) \leftarrow path(u) \cup \{w\}$
        Q.put(w)
      end for
    end if
  end while
end for
return L_n

$$QD(s,t,L) = \min\{d_{ut} + d_{us} | (u, d_{us}, p_{us}) \in L(s), (u, d_{ut}, p_{ut}) \in L(t)\} \quad (2)$$

$$QP(s,t,L) = \operatorname{argmin}\{d_{ut} + d_{us} | (u, d_{us}, p_{us}) \in L(s), (u, d_{ut}, p_{ut}) \in L(t)\} \quad (3)$$

TABLE I. EXPLANATION OF KEY SYMBOLS

Symbol	Description
$G = <V, E>$	Knowledge graph
n	Number of vertices in the knowledge graph
m	Number of edges in the knowledge graph
$N_G(v)$	Neighboring nodes of node v in the knowledge graph
$d_G\ (u, v)$	Shortest distance between nodes u and v in the knowledge graph
$P_G\ (u, v)$	Set of all nodes on the shortest path between nodes u and v in the knowledge graph

The composite label generation algorithm proceeds as follows: Given an input knowledge graph $G =< V, E >$, the algorithm outputs composite labels comprising distance-path pairs for each node in the graph. Initially, the label $L_k(v)$ for each node $v \in V$ is set as an empty set. The algorithm begins at node $v_i$and performs BFS across all nodes. For the starting node v_i, the distance is initialized to 0, and the path path(v_i) is set to $[v_i]$. For all other nodes, distances are initialized to infinity, and their paths are represented as empty sets. The queue Q is initialized to contain only v_i.

During the BFS process, when visiting a vertex u with distance δ and path P, the algorithm checks if the query result $QD\left(v_k, u,\ L'_{k-1}\right) \leq \delta$. If this condition is met, vertex u is pruned, meaning that the label (v_k,δ ,P) is not added to $L'_k(u)$, and traversal of $u\prime s$ adjacent edges is terminated. Conversely, if $QD\left(v_k, u,\ L'_{k-1}\right) > \delta$, the label $L'_k(u)$ is updated to $L'_{k-1}(u) \cup \{(v_k, \delta, P)\}$, and the algorithm proceeds to traverse all adjacent edges of vertex u. For each neighboring vertex w of u, the algorithm updates $w\prime s$ distance and path accordingly and enqueues w into Q.

Fig. 2 presents examples of pruned BFS traversals. Yellow vertices denote the roots, blue vertices denote those which we visited and labeled, green vertices denote those which we visited but pruned, and gray vertices denote those which are already used as roots. The initial pruned BFS, originating from vertex 1, successfully visits all vertices, as shown in Fig. 2a. In the subsequent BFS traversal starting from vertex 2 (Fig. 2b), upon reaching vertex 6, we observe that $QD(2, 6, L'_1) = d_G(2, 1) + d_G(1, 6) = 3 = d_G(2, 6)$, allowing us to prune vertex 6 and avoid traversing any of its outgoing edges. Similarly, vertices 1 and 12 are pruned in this traversal. As additional BFS traversals are performed, the search space progressively diminishes in size, as illustrated in Fig. 2c and Fig. 2d.

The algorithm enhances computational efficiency by employing pruning techniques to minimize redundant traversal, thereby optimizing the process. The final output is the updated label set L_n, which constructs composite labels for each node, embedding both shortest path and corresponding distance information. These composite labels enable rapid determination of the shortest path between any two connected nodes in the graph, eliminating the need for exhaustive graph traversal or complex path-finding operations. This composite labeling mechanism thus provides an efficient approach for querying shortest paths, significantly improving query speed and conserving computational resources.

3) Minimum spanning tree query algorithm based on shortest path.: Given that user queries frequently involve multiple entities, represented as a set of nodes within the graph, the task of identifying the minimum-cost tree that connects these nodes is defined as the minimum spanning tree problem. Algorithm 2 leverages the pre-established composite labels to compute the shortest paths between nodes, thus facilitating the construction of an optimal tree that spans all nodes within the specified set.

Algorithm 2 Minimum Spanning Tree Query Algorithm Based on Shortest Path

```
tree ← ∅
distance(k) ← 0, path_i ← {N_k}
for i ← 1 to k − 1 do
    for j ← i + 1 to k do
        d_G(N_i, N_j) ← QD(N_i, N_j, L_n)
        PG(N_i, N_j) ← QP(N_i, N_j, L_n)
    end for
    distance(i) ← min{d_G(N_i, N_j) | j = 1, 2, ..., k}
    path(i) ← arg min PG(N_i, N_j)   for j = 1, 2, ..., k
end for
Q ← {1, 2, ..., k}
while Q is not empty do
    y ← arg min (distance(m) | m ∈ Q)
    if y ∈ Q then
        tree ← tree ∪ path(y)
        Q.remove(y)
    end if
end while
return tree
```

The minimum spanning tree query algorithm based on shortest path operates as follows: Given input parameters consisting of composite labels L_n and a set of k nodes $N_1, N_2 \ldots N_k$, the algorithm outputs the minimum spanning tree that connects these nodes. Initialization starts by setting the minimum spanning tree, denoted as tree, to an empty set. The initial path distance $distance\,(k)$ for node N_k is set to 0, with the path $path\,(k)$ initialized to N_k. For the remaining nodes $N_1, N_2 \ldots N_{k-1}$, distances are initially set to infinity, and paths are initialized as empty sets.

The algorithm begins with the first node N_1 and performs

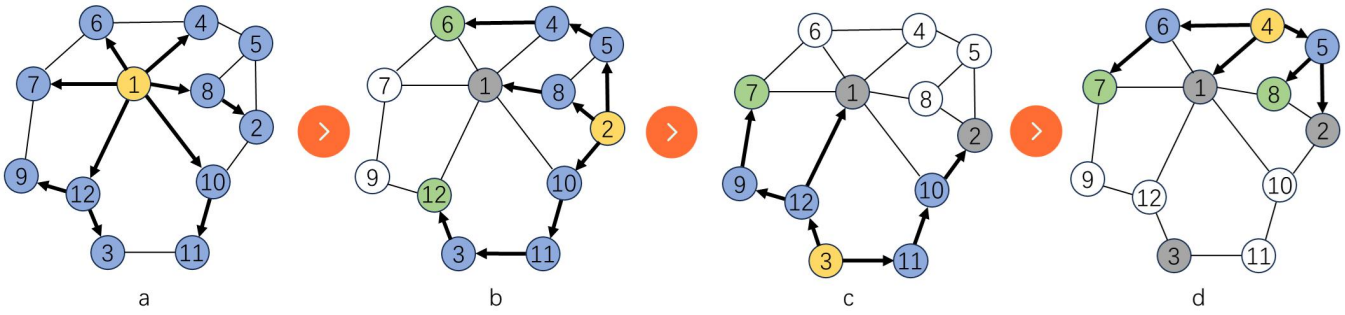

Fig. 2. Examples of pruned BFS traversals.

shortest path queries for each node pair N_i and N_j. Specifically, it calculates the shortest distance $d_G(N_i, N_j)$ and the corresponding path $P_G(N_i, N_j)$ between nodes N_i and N_j using the composite labels L_n. Among all query results, the shortest distance is selected, and the associated shortest path is assigned to $path(i)$. Subsequently, a greedy approach constructs the minimum spanning tree by enqueueing all nodes into Q. In each iteration, the algorithm selects the node y with the smallest path distance from Q, and appends the shortest path $path(y)$ to the minimum spanning tree tree. The node y is then removed from Q, and this process continues until all nodes have been processed.

The algorithm utilizes composite labels to pre-store and compress distance and path information between nodes, enabling rapid retrieval of shortest paths and their corresponding distances through label queries. This approach markedly enhances path computation efficiency by eliminating the need for complex graph traversal each time a shortest path calculation is required. Additionally, the algorithm adopts a greedy strategy to incrementally construct the minimum spanning tree, selecting the next path based on the minimum distance between nodes and adding it to the spanning tree. This greedy selection ensures that each chosen path is optimal at its respective step, thus guaranteeing the overall optimality of the final spanning tree. This method not only simplifies the process of constructing the spanning tree but also maximizes the algorithm's efficiency.

The algorithm guarantees the accuracy of query results while maintaining low time complexity. By minimizing redundant path calculations and optimizing the construction process, it effectively processes large-scale graph data in a short time frame, making it especially suitable for constructing minimum spanning trees in complex networks with numerous nodes and edges. By integrating the efficient query mechanism of composite labels with the benefits of a greedy strategy, the algorithm offers a rapid and precise solution for minimum spanning tree construction.

4) Approximation Ratio Analysis: As the Steiner tree problem is a classic NP-hard problem in graph theory and combinatorial optimization, obtaining an optimal solution for large graphs with extensive terminal node sets demands considerable computational resources and may be infeasible within a reasonable timeframe. To assess and compare the effectiveness of approximation algorithms, the approximation ratio is commonly employed as a metric. This ratio quantifies the discrepancy between the solution produced by an approximation algorithm and the true optimal solution. It is typically defined as Eq. (4).

$$r = \frac{w(T_{appopt})}{w(T_{opt})} \tag{4}$$

In this context, T_{opt} represents the optimal solution to the Steiner tree problem, T_{appopt} denotes the spanning tree obtained using the minimum spanning tree query algorithm based on the shortest path, and $w(T)$ indicates the total weight of tree T.

For a query on the node set $N = \{N_1, N_2 \dots N_K\}$, the algorithm constructs the tree by iteratively adding the shortest paths between nodes. In each iteration, a shortest path is appended to the spanning tree. Suppose that, in each greedy selection, the node $pair(N_i, N_j)$ is chosen, and the shortest path $P_G(N_i, N_j)$ with length $d_G(N_i, N_j)$ is identified between them. Consequently, the weight of the spanning tree generated by the approximation algorithm can be expressed as the sum of all shortest path lengths.

$$w(T_{appopt}) = \sum_{edges} d_G(N_i, N_j) \tag{5}$$

The optimal Steiner tree T_{opt} is the spanning tree with the minimum total weight that connects the specified node set N. This tree may incorporate non-terminal nodes to reduce the connection costs between terminal nodes, and therefore, its total weight $w(T_{opt})$ represents the minimum achievable weight for a spanning tree in the graph.

$$w(T_{opt}) = \min_{tree T \supseteq N} w(T) \tag{6}$$

The upper bound of the approximation ratio for the classical greedy approximation algorithm for the Steiner tree problem is $2\left(1 - \frac{1}{k}\right)$. When k is small, the shortest paths between terminal node pairs closely align with the optimal solution, causing the approximation ratio to approach 1. However, as the number of terminal nodes k and the number of paths between terminal nodes increase, the algorithm increasingly overlooks

the introduction of Steiner points, leading to a gradual rise in the approximation ratio. Nonetheless, the upper bound remains capped at 2, as Eq. (7).

$$r = \frac{w\left(T_{appopt}\right)}{w\left(\ T_{opt}\right)} \leq 2 \tag{7}$$

The above analysis of the approximation ratio suggests that for smaller node sets, the approximation algorithm can closely approach the optimal solution, offering reliable performance guarantees. Consequently, the algorithm demonstrates a degree of effectiveness in addressing large-scale Steiner tree problems.

5) Acquiring graph information via the minimum spanning tree: In this paper, acquiring neighborhood information from the minimum spanning tree aims to establish foundational data support for an advanced question-answering system, thereby enhancing the system's capacity to address complex queries. Specifically, a first-order neighborhood query is performed for each node in the minimum spanning tree, retrieving all directly connected neighboring nodes and their respective relationships to construct each node's local subgraph. Through this approach, the minimum spanning tree provides a streamlined representation of the graph, maintaining essential shortest connections between nodes while eliminating redundant information. By systematically extracting neighborhood information for each node within the minimum spanning tree, the algorithm comprehensively captures the graph's local structural characteristics. This approach ensures the acquisition of complete local information for each node, supplying the necessary contextual and semantic relationship data to support the question-answering system.

C. Answer Generation Module

Given that path algorithms often yield highly redundant information, directly presenting this data to the user can result in information overload and unstructured content, diminishing the overall user experience. To mitigate this issue, we incorporate prompt engineering using a large language model. By designing tailored prompts, we integrate the user's query with relevant information retrieved from external knowledge bases, thereby generating responses that are more targeted, accurate, and contextually relevant.

We have developed a method to convert triples from a knowledge graph (i.e. "entity-relation-entity") into a textual format that is suitable for input into large language models. For each triple, we generate a complete sentence by incorporating connecting words such as "of" and "is". For example, the triple (A, relation, B) is transformed into the sentence "The relation of A is B". This conversion technique not only preserves the structured information from the graph but also facilitates its seamless integration into the LLM input, thereby guiding the model to generate more accurate and coherent responses.

Specifically, the large language model leverages prompt engineering alongside its robust pre-trained generation capabilities to guide the response generation process. Initially, we combine the user's input query Q with a set of relevant retrieved information $[I_1, I_1, \ldots, I_n]$ to construct a prompt, denoted as $Prompt(Q, [I_1, I_1, \ldots, I_n]$. This prompt is then input into the generative model (LLM), which, utilizing both the prompt and the external information, produces a contextually appropriate and precise answer Answer. This process can be represented as Eq. (8).The specific prompts are as follows:

Prompt Template for Response Generator:

[Prompt]: You are a text summarization expert. Please provide answers based solely on the content of the text and conversation records, without fabrication. Pay attention to the distinctions between "approval department", "acceptance department", and "review department", and avoid unnecessary information.

[Question]: What is the application method for replacing (or reissuing) a resident household register?

[Text information]: The acceptance department for replacing (or reissuing) a resident household register is the local police station. The application methods and processing timelines for replacing (or reissuing) a resident household register are: in-person at the service window (1 working day) or online (3 working days). The application conditions for replacing (or reissuing) a resident household register are as follows: if the "Resident Household Register" is damaged or lost and needs to be replaced or reissued, the head of the household can apply for a reissued register for the entire household, while household members can apply for a register that includes only the cover page and their own page.

$$Answer = LLM\left(Prompt\left(Q, [I_1, I_2, \ldots, I_n]\right)\right) \tag{8}$$

Through effective prompt design, the large language model is guided to integrate internal knowledge with external information sources. By leveraging the model's inherent language generation capabilities alongside external knowledge bases, the generated content aligns more closely with the context of the user's query while filtering out redundant information and retaining essential content. This approach ensures both contextual coherence and factual accuracy in the responses, thereby enhancing the user experience.

IV. Construction of the Knowledge Graph in the Household Registration Vertical Domain

This paper presents the construction of a knowledge graph specific to the household registration domain. It transforms unstructured or semi-structured texts, such as policies and regulations related to household registration, into structured information stored in a graph format. This structured representation offers reliable data support for the subsequent question-answering system.

The construction process of the household registration policy knowledge graph encompasses several key steps, including knowledge acquisition, knowledge integration, and knowledge import. The knowledge acquisition phase is primarily based on the Wuhan Household Registration Business Processing Guide(hereafter referred to as Guide) and a question-answer corpus. Through text data preprocessing and information extraction, event triplets reflecting household registration business conditions, procedures, and required materials are generated. In the knowledge integration phase, synonym resolution

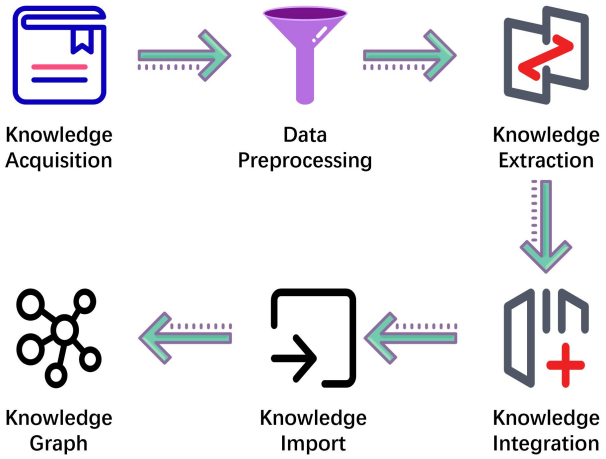

Fig. 3. Flowchart of knowledge graph.

and duplicate entity handling are conducted to ensure data accuracy and consistency. Finally, during the knowledge import phase, the NEO4J graph database is employed to efficiently import the triplet data using Cypher queries and the neo4j-import tool, successfully constructing the knowledge graph that underpins the household registration policy question-answering system. The construction process of the household registration policy knowledge graph is illustrated in Fig. 3.

A. Knowledge Acquisition and Information Extraction

Due to significant regional differences in household registration policies across China, the requirements for handling the same household registration matter can vary considerably from city to city. Therefore, the knowledge sources for the question-answering system must strictly adhere to local household registration policies, and the knowledge graph for the household registration domain must be closed and independent. This paper utilizes the household registration policies of Wuhan, Hubei Province, as the data source to construct the knowledge graph for the household registration domain. The text data primarily derives from the Guide, published by the Wuhan Public Security Bureau, along with a question-answer corpus compiled by household registration center staff. The Guide adopts a semi-structured data format and provides detailed information on 127 household registration services, categorized into six major types and 28 subtypes, covering the conditions for processing these services in Wuhan. It also summarizes numerous domain-specific terms related to household registration and concisely explains how to handle various household registration matters across different scenarios. The question-answer corpus consists of 1,603 question-answer pairs, with all questions being real queries collected manually, closely corresponding to the household registration services described in the Guide.

Based on the textual characteristics of the aforementioned knowledge sources, this paper preprocesses the text data with a focus on household registration events. Specific content for each household registration matter is extracted from the semi-structured tables, resulting in the formation of event triplets. Subsequently, the question-answer texts in the corpus are aligned with the household registration event content, facilitating the construction of entity relationships within the household registration domain. These relationships primarily include item names, application conditions, required materials, processing procedures, application methods, and processing time limits. The specific steps for information extraction are outlined below, with an extraction example illustrated in Fig. 4.

- Given that the Guide is presented in a semi-structured table format, the subject labels and headers of the table can be directly extracted as relationships and attributes. The household registration service item names are identified as head entities, while specific content such as processing locations, procedures, required materials, requirements, application methods, and time limits are extracted as tail entities. Together, these elements constitute the basic event triplets for household registration services.
- To provide supplementary explanations for household registration events under different contextual conditions, the household registration service item names are utilized as head entities, with the supplementary explanations serving as tail entities, thus forming contextual condition triplets for household registration events.
- The questions from the question-answer corpus are manually annotated and mapped to the entities established in steps (1) and (2), resulting in the formation of question triplets for household registration events.

B. Knowledge Integration and Knowledge Import

Merging the data from the two knowledge sources generates a substantial dataset for the knowledge graph; however, numerous duplicate and synonymous entities exist within the two databases. Directly importing these entities would result in redundancy within the knowledge graph. To ensure the accuracy of the question-answering system, this paper integrates the knowledge graphs from both sources by calculating the similarity of entity names. Following this knowledge integration, a total of 1,566 entities across 15 categories and 1,662 relationships across 16 categories were extracted from the documents. The various household registration service contents from the Guide were then imported into the NEO4J database using Cypher queries and the neo4j-import tool.

V. Construction of the Knowledge Graph in the Household Registration Vertical Domain

A. Entity Recognition Experiment

Large language models are trained on extensive text corpora, typically comprising billions of parameters. However, directly applying a large language model to entity recognition tasks often results in suboptimal performance, and conventional fine-tuning methods may induce catastrophic forgetting. The LoRA (Low-Rank Adaptation) efficient fine-tuning technique addresses this issue by employing low-rank decomposition to reduce the number of parameters during training. This

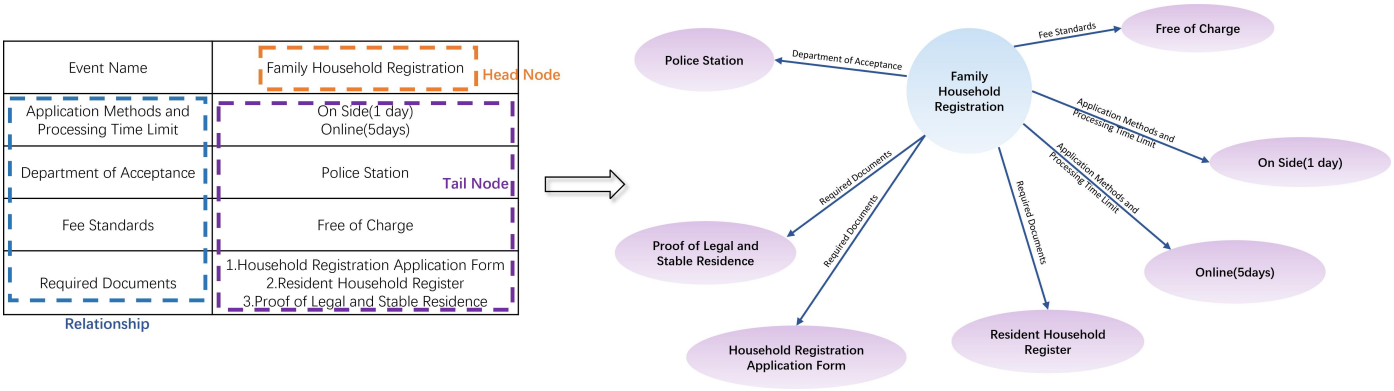

Fig. 4. Example diagram for information extraction.

TABLE II. TEMPLATE OF QUESTIONS FOR VARIOUS CONSULTATION ITEMS

Consultation Items	Template of Questions
Application Methods and Processing Procedures	What application methods are available for processing {item}?
Application Methods and Processing Procedures	What are the procedures for processing {item}?
Precautions	What precautions should be taken when processing {item}?
Application Requirements	What requirements do I need to meet to handle {item}?
Required Documents	What documents are needed to process {item}?
Accepting Department	Where is the accepting department for {item}?
Review Department	Which department handles the review of {item}?
Approval Department	Which department is responsible for approving {item}?
Fee Standards	Does processing {item} incur a charge?
Situation Explanation	When handling {item}, what should be done in the case of {situation}?
Document Explanation	What {materials} are required when handling {item}?

TABLE III. STATISTICS ON DATASET QUANTITIES

Question Types	Quantity
Template Questions	1143
QA Pair Questions	824
Multi-node Questions	611

approach necessitates learning only small parameter matrices, which approximate the updates to the model's weight matrix.

To comprehensively evaluate the interaction capability between the large language model and the knowledge graph, we designed three types of questions. Template-based questions were generated using fixed question templates specifically tailored to consultation items, as outlined in Table II. The question-answer pairs were extracted from the question-answer corpus, while multi-node questions, which involved multiple consultation items, were manually generated. The dataset statistics are presented in Table III. A total of 2,578 question data points were collected, and the dataset was randomly divided into training, validation, and test sets in an 8:1:1 ratio.

In this paper, the $LoRA$ technique was employed to fine-tune the general large model $Llama3-8B$. $LoRA$ enhances fine-tuning efficiency by introducing trainable low-rank matrices atop the model's original weights while preserving overall performance. Tailored to the specific requirements of the household registration domain, the model underwent careful tuning using a specialized fine-tuning dataset to improve its adaptation to entity recognition and relation extraction tasks. During the fine-tuning process, the model progressively learned the specialized terminology and intricate entity relationships present in household registration services, resulting in a significant increase in accuracy within this domain. Ultimately, this fine-tuning process elevated the entity recognition accuracy from its initial level to 96.1%, demonstrating the effectiveness and high adaptability of the $LoRA$ technique when combined with a custom-built dataset.

B. Parameter Settings

In this paper, $Llama3-8B-Chinese$ model was chosen as the core model for answer generation. $Llama3-8B-Chinese$ is a deep learning-based language model renowned for its robust language understanding and generation capabilities, enabling it to handle long text inputs and produce high-quality text outputs. To ensure the accuracy and reproducibility of the experimental results, the parameters of the generation model were meticulously configured and adjusted. The specific parameter settings for the generation model experiments are presented in Table IV.

TABLE IV. EXPERIMENTAL PARAMETER SETTINGS

Parameter	Value
Maximum Input Length	8192
Maximum Output Length	8192
Temperature Coefficient	0
Top_p	1
Top_k	1

Retrieval-augmented generation technology based on knowledge graph path search effectively addresses the complex and varied query demands within household registration

services. Fig. 5 illustrates an example of the household registration intelligent question-answering system.

C. Evaluation Metrics

1) $BLEU$ Evaluation metric: $BLEU$ (Bilingual Evaluation Understudy) is an automatic evaluation method based on N-gram matching, employed to assess the similarity between generated answers and reference answers. By taking into account the matching precision of different N-grams along with a length penalty factor, BLEU produces a score ranging from 0 to 1. Scores closer to 1 indicate a higher similarity between the generated text and the reference text, reflecting better evaluation results. The formula for calculating $BLEU$ is as Eq. (9).

$$BLEU = BP \times \exp\left(\sum_{n-1}^{N} w_n log P_n\right) \tag{9}$$

p_n represents the precision of the $N-gram$ match between the generated text and the reference text, while w_n denotes the weight assigned to the N-gram. The term BP refers to the brevity penalty, which is applied to prevent the generated text from being excessively short. The formula is as Eq. (10). c is the length of the generated text, and r is the length of the reference text.

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{\left(1-\frac{r}{c}\right)}, & \text{if } c \leq r \end{cases} \tag{10}$$

2) $ROUGE$ Evaluation metrics: The core of the $ROUGE$ evaluation metric lies in calculating the recall and precision of the generated text by matching N-grams between the generated answer and the reference answer, resulting in a final score. Specifically, $ROUGE$ assesses the quality of the generated text by evaluating the number of matching N-grams between the generated text and the reference text. This metric is widely applicable across various natural language processing tasks, including summarization and machine translation, where the effectiveness of the generated text often depends on its ability to capture key information from the reference text. $ROUGE$ offers a comprehensive performance analysis through multi-dimensional evaluation and demonstrates exceptional effectiveness in large-scale text processing scenarios.

3) $METEOR$ Evaluation metrics: The $METEOR$ evaluation metric was developed to address the limitations of $BLEU$ in accounting for sentence syntax and word order. It combines word-level and phrase-level alignment methods while incorporating strategies such as stemming and synonym replacement, thereby enhancing its accuracy in evaluating the semantic similarity between generated text and reference text. $METEOR$ offers a more nuanced assessment of translation quality by considering factors such as lexical precision, stemming, synonyms, and word order. The metric initially calculates precision and recall, subsequently balancing these two metrics through an F-score. Furthermore, $METEOR$ introduces a word order penalty to more accurately reflect the significance of word sequence in translation.

4) $Perplexity$ Evaluation metrics: $Perplexity$ is a crucial evaluation metric for measuring the predictive capability of a language model, assessing how effectively the model adapts to a given text sequence. A lower perplexity value indicates a stronger ability of the model to predict new text, signifying that the generated output aligns more closely with the probability distribution of the language model, resulting in more fluent sentences. Conversely, a higher perplexity value suggests a weaker predictive ability, which can lead to lower quality and consistency in the generated text. For a high-quality language model, the perplexity value should be minimized to ensure that the generated text meets the expected semantics and adheres to appropriate language structure. The formula for calculating perplexity is as Eq. (11). $P\left(w_i|w_{i-1}\right)$ represents the model's predicted probability of the current word w_i given the previous word w_{i-1}, and N is the total length of the text.

$$Perplexity = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} logP\left(w_i|w_{i-1}\right)\right) \tag{11}$$

D. Experimental Subjects

In this paper, we conducted experimental comparisons of the proposed household registration policy intelligent question-answering system with three baseline methods: $LLM-only$, $RAG-Embedding$, and $RAG-Graph$.

1) $LLM-only$: This method relies exclusively on the internal knowledge of the large language model (LLM) for reasoning, without incorporating any external knowledge. Techniques such as Chain of Thought (COT) enhance the model's capacity for complex reasoning by generating a series of intermediate reasoning steps [35]. COT facilitates the model's ability to decompose multi-step problems, thereby improving reasoning accuracy.

2) $RAG-Embedding$: This method is based on RAG technology, retrieving relevant information from external knowledge bases using embedding vectors and combining it with LLM to generate answers. Embedding vectors effectively capture semantic similarity in text, improving retrieval accuracy and making the generated answers more relevant and precise. $LangchangChatChat$ retrieves relevant information from external knowledge bases using embedding vectors, combines it with LLM to generate answers, and integrates this information to enhance the quality and consistency of responses.

3) $RAG-Graph$: This method combines RAG technology with knowledge graphs to leverage the structured knowledge in knowledge graphs to enhance the contextual understanding and answer accuracy of the generative model, thereby improving the quality and relevance of the generated content. Graph RAG applies community detection algorithms to partition the knowledge graph into multiple sub-communities, generates local summaries for each community, and integrates these summaries to form a global answer [36].

VI. Experimental Results

A. Comparative Experiment

Table V presents the comparative experimental results across various models, underscoring the superior perfor-

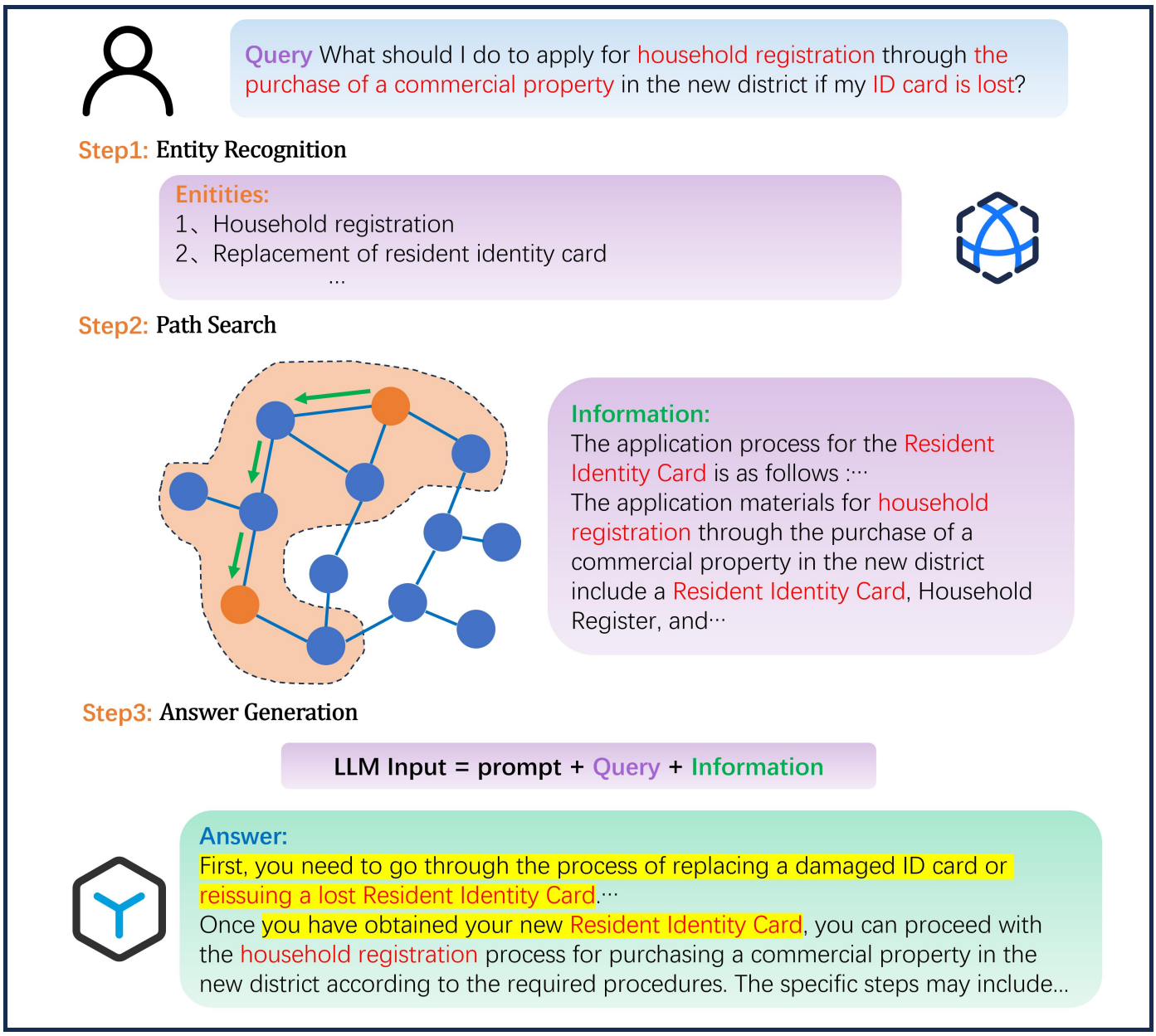

Fig. 5. Example of the household registration question-answering system.

mance of the proposed household registration policy intelligent question-answering system. The results indicate that our model ($Ours$) significantly outperforms other models across all evaluation metrics. Specifically, $ours$ achieved $BLEU$, $ROUGE-1$, $ROUGE-L$, and $METEOR$ scores of 48.19%, 58.14%, 52.39%, and 55.48%, respectively, outperforming both the COT model and the $Graph-RAG$ model. These metrics highlight the model's effectiveness in generating content with enhanced fluency, comprehensive information coverage, and improved semantic relevance. Furthermore, $ours$ recorded a Perplexity score of 42.48, lower than that of other models, reflecting greater coherence and reduced language model ambiguity. Collectively, these findings underscore the exceptional performance of the proposed model in intelligent question-answering tasks.

B. Ablation Study

To investigate the impact of model fine-tuning and the path algorithm on the experimental results, ablation studies were conducted on the dataset. To ensure the objectivity and accuracy of the experiments, we designed the following sets of control experiments:

1) $KG(Baseline)$: This experiment directly uses the pre-trained model without fine-tuning for entity extraction and performs entity linking on the knowledge graph. It extracts relevant information using first-order neighborhood queries and generates answers. The purpose of this experiment is

TABLE V. EXPERIMENTAL COMPARISON RESULTS

Model	$BLEU \uparrow$	$ROUGE-1 \uparrow$	$ROUGE-L \uparrow$	$METEOR \uparrow$	$Perplexity \downarrow$
COT	22.59%	33.03%	25.03%	27.78%	53.79
$GraphRAG$	15.15%	21.47%	11.56%	26.32%	65.56
$Langchain$	41.65%	52.97%	47.41%	49.98%	43.38
$Ours$	48.19%	58.14%	52.39%	55.48%	42.48

to evaluate the performance of the base pre-trained model, serving as a baseline for subsequent experiments.

2) $KG+PathAlgorithm$*:* While keeping the basic model parameters unchanged, the path algorithm is applied to process the data to quantify its contribution to the experimental results.

3) $KG+Fine-tuning$*:* In this experiment, the pre-trained model is fine-tuned without applying the path algorithm. The aim of this experiment is to independently assess the effect of fine-tuning on improving the model's performance.

4) $KG + Fine-tuning + PathAlgorithm(ours)$*:* This experiment combines both model fine-tuning and the path algorithm to explore whether their combined effect can significantly enhance the overall model performance.

Through this experimental design, we aim to comprehensively analyze the individual and combined contributions of fine-tuning and the path algorithm to the model's performance. The results of these experiments are presented in Table VI.

The experimental results show significant differences in performance across different model combinations on various evaluation metrics. The Ours system achieved the best results on metrics such as BLEU, ROUGE, and METEOR. The BLEU score of this system reached 48.19%, significantly higher than other combinations, indicating that the optimized model performs better in terms of similarity between the generated text and the reference text. The ROUGE-1 scores was 58.14%, respectively, further demonstrating the model's superiority in precise matching and coverage at both the word and phrase levels. The METEOR score, at 55.48%, also indicates stronger semantic consistency in the generated content. Although the Perplexity score for the Ours system was slightly higher than that of the KG+Fine-tuning system, the overall score still suggests that the model fine-tuned with the path algorithm exhibits high-quality text generation. Therefore, these results demonstrate that the integration of the path algorithm and fine-tuning significantly improves the model's accuracy, consistency, and coverage in language generation.

VII. DISCUSSION

This research proposes a Retrieval-Augmented Generation (RAG) method based on knowledge graph path search and successfully applies it to an intelligent question-answering system in the field of household registration policy. Experimental results demonstrate that the integration of path search algorithms with model fine-tuning significantly enhances the system's efficiency and accuracy in multi-node query processing, entity recognition, and information extraction. However, the research process also reveals several issues that require further exploration.

Firstly, although knowledge graph-based path search significantly improves retrieval effectiveness, the efficiency of path search still faces bottlenecks as the scale of the knowledge graph continues to expand. Particularly with the rapid increase in the number of nodes and relationships, path computation and multi-node queries still consume substantial time and computational resources. Although this study employs pruning strategies to reduce the search space, further optimization of path search algorithms, especially in real-time query scenarios, remains an important direction for future research.

Secondly, while fine-tuning methods (such as the LoRA method) have effectively improved the accuracy of entity recognition, the model may still encounter misidentification and erroneous inferences when dealing with complex and hierarchical domain knowledge. Although fine-tuning methods have significantly enhanced the model's performance in the field of household registration policy, balancing the model's specificity and generalization capabilities to avoid losing its ability to handle problems in other domains while overfitting to domain-specific terminology remains a significant challenge for future research.

Lastly, the knowledge graph construction in this study is based on household registration policy data from Wuhan City. While this domain-specific data source provides efficient service support for the system, the construction and updating mechanisms of the knowledge graph still face considerable challenges in cross-regional or broader application scenarios. Efficiently integrating policy data from different regions into a unified knowledge graph and ensuring its timeliness and accuracy will be crucial for enhancing the system's generalization capabilities.

VIII. FUTURE WORK

In future research, we aim to further optimize and expand the outcomes of this study in the following key areas:

1) Automatic update mechanism for domain knowledge: The knowledge graph developed in this study is based on household registration policy data from Wuhan City. However, policies and regulations are subject to continuous changes in real-world scenarios. To ensure the intelligent question-answering system can provide up-to-date policy information in a timely manner, future research will focus on the development of an automatic update mechanism for the knowledge graph. Specifically, we will explore the integration of real-time data crawling techniques with knowledge graph update strategies, enabling the system to rapidly adapt to policy changes and thereby enhancing its real-time performance and accuracy.

2) Integration and fusion of multi-domain knowledge graphs: While this study primarily addresses household registration policy, real-world applications often require knowledge from multiple domains. Future research will focus on

TABLE VI. ABLATION STUDY RESULTS

Model	$BLEU \uparrow$	$ROUGE-1 \uparrow$	$ROUGE-L \uparrow$	$METEOR \uparrow$	$Perplexity \downarrow$
KG	28.48%	38.18%	34.70%	32.86%	46.93
KG+PathAlgorithm	36.13%	44.14%	38.94%	41.40%	47.95
KG+Fine-tuning	40.56%	52.59%	49.26%	47.32%	40.99
Ours	48.19%	58.14%	52.39%	55.48%	42.48

the construction and integration of multi-domain knowledge graphs. For example, household registration policy intersects with domains such as social security, taxation, and education. Efficiently fusing knowledge from these diverse domains and enabling cross-domain reasoning within the intelligent question-answering system will be a critical direction for further investigation.

3) User interaction and optimization of personalized question-answering systems: The current research predominantly addresses standardized question-answering tasks. However, user needs in practical applications are often complex and varied. Future research will explore the development of personalized question-answering systems by incorporating users' historical queries and preferences. By analyzing user behavioral patterns and interests, and leveraging recommendation system technologies, we aim to enhance the system's intelligence and enable it to deliver more personalized responses tailored to users' specific needs.

4) Cross-language and cross-cultural expansion: Although this study focuses on Chinese household registration policy, future work will extend this methodology to multilingual and cross-cultural contexts. With the increasing trend of globalization, developing a question-answering system that can operate effectively across different languages and cultures is of significant practical importance. Future efforts will concentrate on designing knowledge graphs and generation models that are adaptable to multilingual environments, ensuring consistency and accuracy across various languages, and facilitating the system's broader application on a global scale.

IX. SUMMARY

This paper presents a retrieval-enhanced generation approach that leverages knowledge graph path search, specifically applied to the household registration domain. Comprehensive experiments validate the effectiveness of this method. By constructing a knowledge graph tailored to this vertical field and employing a pruning marker technique alongside the shortest path generation tree algorithm, the method ensures efficient path querying. Additionally, fine-tuning a large language model enhances the accuracy of entity recognition and information extraction. The knowledge graph developed for the household registration field encompasses policies, regulations, and social services, creating a robust resource for relevant information retrieval. Experimental results, using the household registration field as a case study, reveal that the proposed method significantly outperforms traditional approaches across BLEU, ROUGE, and METEOR metrics, achieving substantial reductions in perplexity. This improvement demonstrates the model's coherence and accuracy in text generation. The specific conclusions are as follows.

1) The knowledge graph developed in this paper encompasses entities and relationships pertinent to household registration policies. By integrating fine-tuning strategies for large language models, the system effectively provides accurate answers to complex, multi-node queries. Experimental results demonstrate that the combination of pruned marking and shortest path generation tree algorithms substantially enhances the efficiency of multi-node path queries, ensuring the system's high performance and accuracy in handling complex questions.
2) Ablation experiments confirm the performance improvements attributed to model fine-tuning and the path algorithms. Specifically, in knowledge graph path searches and information retrieval tasks, the path algorithm ensures information relevance and precision, while model fine-tuning further enhances the accuracy of answer generation.
3) Effective prompt engineering played a critical role in answer generation, enabling the system to filter out extraneous information by combining retrieved knowledge with user queries, thus producing answers that are both contextually relevant and targeted. The reduction in perplexity corroborates the system's improved ability to generate high-quality responses, enhancing overall system performance and user experience.

This paper presents an efficient and precise solution for question-answering in complex vertical domains, highlighting the substantial potential of combining large language models with knowledge graphs. The findings offer valuable technical insights for advancing question-answering systems in specialized fields.

ACKNOWLEDGMENT

This work was partially supported by the Department of Science and Technology of Hubei Province, China, for the Hubei Provincial Key Research and Development Program project (2022BAA051).

REFERENCES

[1] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1. Minneapolis, Minnesota, 2019, p. 2.

[2] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard *et al.*, "Kilt: a benchmark for knowledge intensive language tasks," *arXiv preprint arXiv:2009.02252*, 2020.

[3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

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

[5] M. A. Hearst, "Untangling text data mining," in *Proceedings of the 37th Annual meeting of the Association for Computational Linguistics*, 1999, pp. 3–10.

[6] X. Yao and B. Van Durme, "Information extraction over structured data: Question answering with freebase," in *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers)*, 2014, pp. 956–966.

[7] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[8] Y. Luan, J. Eisenstein, K. Toutanova, and M. Collins, "Sparse, dense, and attentional representations for text retrieval," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 329–345, 2021.

[9] D. Chen, "Reading wikipedia to answer open-domain questions," *arXiv preprint arXiv:1704.00051*, 2017.

[10] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee *et al.*, "Natural questions: a benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[11] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[12] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," *arXiv preprint arXiv:2007.01282*, 2020.

[13] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[14] H. Paulheim, "Automatic knowledge graph refinement: A survey of approaches and evaluation methods (2015)."

[15] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE transactions on knowledge and data engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[16] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier *et al.*, "Knowledge graphs," *ACM Computing Surveys (Csur)*, vol. 54, no. 4, pp. 1–37, 2021.

[17] K. W. Chan and W. Buckingham, "Is china abolishing the hukou system?" *The China Quarterly*, vol. 195, pp. 582–606, 2008.

[18] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," *arXiv preprint arXiv:2007.00808*, 2020.

[19] K. Lee, M.-W. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," *arXiv preprint arXiv:1906.00300*, 2019.

[20] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[21] W. Ding, J. Li, L. Luo, and Y. Qu, "Enhancing complex question answering over knowledge graphs through evidence pattern retrieval," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2106–2115.

[22] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[24] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[25] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.

[26] L. Yao, C. Mao, and Y. Luo, "Kg-bert: Bert for knowledge graph completion," *arXiv preprint arXiv:1909.03193*, 2019.

[27] J. Gao, X. Wang, Y. Wang, and X. Xie, "Explainable recommendation through attentive multi-view learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3622–3629.

[28] E. M. Bender and A. Koller, "Climbing towards nlu: On meaning, form, and understanding in the age of data," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 5185–5198.

[29] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," *arXiv preprint arXiv:2104.06378*, 2021.

[30] Z. Hu, Y. Xu, W. Yu, S. Wang, Z. Yang, C. Zhu, K.-W. Chang, and Y. Sun, "Empowering language models with knowledge graph reasoning for question answering," *arXiv preprint arXiv:2211.08380*, 2022.

[31] F. Xu, S. Zhou, Y. Ma, X. Wang, W. Zhang, and Z. Li, "Open-domain dialogue generation grounded with dynamic multi-form knowledge fusion," in *International Conference on Database Systems for Advanced Applications*. Springer, 2022, pp. 101–116.

[32] M. Kang, J. M. Kwak, J. Baek, and S. J. Hwang, "Knowledge graph-augmented language models for knowledge-grounded dialogue generation," *arXiv preprint arXiv:2305.18846*, 2023.

[33] B. Peng, M. Galley, P. He, C. Brockett, L. Liden, E. Nouri, Z. Yu, B. Dolan, and J. Gao, "Godel: Large-scale pre-training for goal-directed dialog," *arXiv preprint arXiv:2206.11309*, 2022.

[34] S. Yang, R. Zhang, and S. Erfani, "Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems," *arXiv preprint arXiv:2010.01447*, 2020.

[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[36] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson, "From local to global: A graph rag approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, 2024.