applied sciences

Article

Domain Knowledge Graph Question Answering Based on Semantic Analysis and Data Augmentation

Shulin Hu, Huajun Zhang * and Wanying Zhang

School of Automation, Wuhan University of Technology, Wuhan 430062, China; bookwoods@whut.edu.cn (S.H.); zhangwanying@whut.edu.cn (W.Z.)
* Correspondence: zhanghj_hust@163.com

Abstract: Information retrieval-based question answering (IRQA) and knowledge-based question answering (KBQA) are the main forms of question answering (QA) systems. The answer generated by the IRQA system is extracted from the relevant text but has a certain degree of randomness, while the KBQA system retrieves the answer from structured data, and its accuracy is relatively high. In the field of policy and regulations such as household registration, the QA system requires precise and rigorous answers. Therefore, we design a QA system based on the household registration knowledge graph, aiming to provide rigorous and accurate answers for relevant household registration inquiries. The QA system uses a semantic analysis-based approach to simplify one question into a simple problem consisting of a single event entity and a single intention relationship, and quickly generates accurate answers by searching in the household registration knowledge graph. Due to the scarcity and imbalance of QA corpus data in the field of household registration, we use GPT3.5 to augment the collected questions dataset and explore the impact of data augmentation on the QA system. The experiment results show that the accuracy rate of the QA system using the augmented dataset reaches 93%, which is 6% higher than before.

Keywords: domain knowledge graph; question answering; data augmentation; large language model

Citation: Hu, S.; Zhang, H.; Zhang, W. Domain Knowledge Graph Question Answering Based on Semantic Analysis and Data Augmentation. *Appl. Sci.* **2023**, *13*, 8838. https://doi.org/10.3390/app13158838

Academic Editor: Christos Bouras

Received: 10 July 2023
Revised: 28 July 2023
Accepted: 28 July 2023
Published: 31 July 2023

1. Introduction

The question answering (QA) system is designed to provide users with personalized information services through human–computer interaction in the form of question-and-answer by analyzing the user's input. As one of the core tasks of artificial intelligence, QA has attracted extensive attention due to its widespread application in natural language processing and information retrieval [1]. Information retrieval-based question answering (IRQA) and knowledge-based question answering (KBQA) are the main forms of QA systems [2]. IRQA is known as open-domain QA: it can answer the questions that come from any domain. This type of QA system retrieves relevant texts from a large amount of passage based on the user's given question using information retrieval methods. Simply using retrieved texts as the answer is not precise enough. Thanks to the breakthroughs in recent years in the large language models (LLM), such as GPT-4 [3] and ChatGLM [4], which can better understand natural language questions, integrating LLM into QA systems can provide users with more comprehensive answers. However, because the answers generated by LLM have a certain degree of randomness, it cannot guarantee that the answers will always be consistent with the retrieved texts. In some special domains (such as medicine, policy, law, etc.), QA systems are required to provide accurate and rigorous answers to ensure that the system can provide reliable and authoritative information. On the other hand, KBQA systems process unstructured or semi-structured passages into structured database storage. They construct query templates through semantic analysis of the questions, and the final answers are also retrieved from these texts. As a result, the answers obtained are consistent with the retrieved texts, providing high accuracy and precision. The household registration domain studied in this paper is a policy subfield, which requires a strict one-to-one correspondence between the answer and the original text to avoid misleading users or producing adverse consequences. Therefore, we adopt the knowledge-based paradigm to construct a household registration QA system to ensure that users obtain accurate and rigorous answers.

The household registration system is one of the most fundamental social management systems in the world, mainly involving matters such as birth, death, migration, marriage, divorce, adoption, and disappearance. For instance, Japan revised the Family Registration Law [5] in 1871, and the Act for Registering Births, Deaths, and Marriages in England [6] has been continuously used in the UK since 1838 and has undergone several revisions. Although the names of household registration laws and regulations in countries around the world are different, their actual content are similar. In recent years, with the development of urbanization and population growth in many developing countries, the handling of household registration in various complex situations has led to continuous updates in related policies with more explicit explanations. For example, the required documents for handling birth registration for newborns may vary depending on the marital status and ethnic composition of the parents. However, due to limitations in users' expression ability and a lack of information query channels, how to accurately and quickly retrieve the household registration information that users need has become an urgent problem to be solved.

The primary objective of this paper is to build a QA system for household registration to provide users with accurate and reliable household policy information, improve the efficiency of household registration services, and speed up the processing time. In household registration text data, the handling information of the same event entity is different under various constraints. Such complex semantic information will lead to multi-constraint query problems in answer retrieval when mapped into a triple form. The corpus data of various consultation questions involved in this study is derived from the records of the consultation system of the government service hall. Since the QA system uses speech recognition technology to convert the user's speech content into a text form of consultation questions, the input of the QA system often has problems with missing sentence components or unclear descriptions. This makes it difficult to correctly identify the event entity and further complicates answer retrieval. The effectiveness of a QA system heavily depends on the quality of its training data. Due to the high cost of sample data acquisition, it is difficult to obtain a sufficient number of training samples.

The main contributions of this paper are summarized as follows:

- This paper uses compound value types (CVT) nodes to store household registration events. Since CVT nodes collect multiple attributes of events and more accurately model complex relationships between entity nodes, this approach simplifies queries with multiple constraints in a knowledge graph (KG) into simple queries;
- This paper comprehensively uses KGs and text similarity technology to improve the accuracy of the QA system. It leverages a corpus of query questions to train a RoBERTa-BiLSTM-MultiHeadAttention (RBMA) model to classify query intent. When the intent is clear, it utilizes the language technology platform (LTP) [7] to extract semantic role subjects from queries, and further retrieves the answer from the KG. When the intent is ambiguous, it uses text similarity techniques to match input queries with a corpus of queries and outputs the most similar answers;
- This paper applies the LLM to enhance the training data to solve the problem of data imbalance and improve the accuracy of intent classification. We use the GPT-3.5-turbo language model to augment the dataset size by replacing synonyms and randomly inserting irrelevant words. The experiment results show that data augmentation techniques greatly improve the performance of QA systems.

The structure of the remaining sections of the paper is as follows: Section 2 discusses related works, Section 3 describes the process of constructing a household registration domain KG, Section 4 details the framework structure of the QA system, Section 5 presents the experiment results and analysis, and Section 6 concludes the paper and discusses future work.

2. Related Works

In natural language processing, a simple question pertains to a single head entity and relation present in the knowledge graph (KG), with its corresponding tail entity acting as the answer [8], and a complex question commonly involves multiple entities and relationships within the KG or obtains the answer through specialized operations. This type of question is also referred to as a multi-constraint question [9].

Template-based methods and semantic parsing-based methods are the two main paradigms in KBQA [2]. Template-based methods use templates or rules to answer questions by mapping the questions to predefined templates [10,11]. Although this approach has higher accuracy, it has lower coverage and recall for various types of domain-specific questions [12]. For instance, H. Bast and E. Haussmann [13] proposed a model called Aqqu that maps the question to three templates, identifies all entities in the KG that match the question, and instantiates the three templates using Aqqu. Based on a ranking model, the best instantiation is selected to query the KG and retrieve the answer. However, these templates provide limited coverage for complex questions. Abujabal et al. [14] introduced an automated template generation model named QUINT, which generates question templates based on the dependency parse of the given question. Then, it queries candidate results based on these question templates, sorts them using a random forest classifier, and outputs the final answer obtained by the query. Semantic parsing-based methods involve constructing a semantic parser to map natural language questions into a semantic representation, logical expression, or query graph [15]. These representations are used to query the knowledge base and retrieve the answer. For instance, Yongrui Chen et al. [16] generated query graphs using a hierarchical self-recursive decoder that outlines the query graph and continually populates it. This end-to-end model enhances the accuracy of answering complex questions but requires manual design of semantic logic representations and query rules. K. Xu et al. [17] introduced a syntactic query graph that represents the intention of input questions based on three types of syntactic information: word order, dependency relations, and constituents. Then, they encoded the syntactic graph using a graph-to-sequence model and decoded the logical form of the question.

In the 1960s, KBQA systems such as BASEBALL [18] and LUNAR [19] had already been developed. BASEBALL was designed to answer questions about American League baseball issues within a one-year cycle, while LUNAR aimed to answer questions related to lunar rock geology analysis based on data collected from the Apollo moon landing missions. These early systems were designed specifically for domain-specific QA through structured data processing. Currently, there are three main ways to store knowledge: The first is RDF storage in the form of triples; the second is storage in traditional relational databases; and the third is storage in graph databases. Graph structures have the natural advantage of exploiting both structural and semantic information to analyze complex relationships [20], so we use a knowledge graph to store knowledge.

After Google proposed the knowledge graph (KG) in 2012, with the emergence of large-scale KGs such as Wikidata [21], Dbpedia [22], and Freebase [23], knowledge-graph-based question answering (KGQA) has gradually become a research hotspot, attracting considerable attention from researchers [24]. This allows us to convert semantic analysis results into structured data and query information in the knowledge base [25]. A KG is a directed graph that uses entities as nodes and entity relations as edges [26]. Essentially, it is a knowledge base represented by a structured semantic network [27,28]. Each directed edge in the graph creates a triplet composed of a head entity, a tail entity, and their relation, forming a directed relationship between entities. The construction of KGs now has a fairly mature workflow, storing domain knowledge in a structured data format and providing data support for fast and efficient QA systems. In recent years, with the concept of the KG spreading to various fields, there has been no shortage of research in finance, medicine, education, e-commerce, and even the military. For example, the medical KGQA system [29], the e-commerce KGQA system [30], and the intelligent travel KGQA system [31], with their greater depth of knowledge, can provide more accurate professional knowledge services to users in their respective fields.

There are two characteristics of knowledge in the household registration domain. Firstly, household registration policies vary significantly by region. Secondly, there are many additional constraint conditions in the supplemental descriptions of household registration events. Template-based methods require manual design of template rules, and the templates have limited applicability. In order to ensure that the QA system has a certain degree of generalization ability and will not change the template due to changes in some household registration policies, we use the semantic parsing-based method to construct the QA system. However, the basic query graph or logical expression in the semantic parsing method is only suitable for simple questions and cannot express multi-constraint questions. Bao et al. [9] constructed multi-constraint query graphs by adding constraints to the basic query graph to limit the set of answers. This method effectively solves the problem of complex question queries by setting more rules to cover multi-constraint problems. Instead of adding rules to achieve multi-constraint queries, we simplified complex questions into simple ones by changing the storage method of the graph. Multiple-constraint event entities were replaced with CVT nodes to participate in the query, and the RBMA model was used to parse the question's intention into a relationship, ultimately obtaining a query representation for simple questions.

3. The Construction of Household Registration Domain Knowledge Graph

The aim of the construction of the household registration domain KG is to process unstructured or semi-structured texts, such as household registration policy regulation files, into structured information and store them in a graph structure to provide data support for subsequent QA systems. At present, there are two main ways to construct a KG: top-down and bottom-up [32]. Top-down construction refers to extracting ontology and schema information from high-quality structured data sources and adding them to the database. It is knowledge-oriented and can ensure that the KG has high standardization and accuracy. Bottom-up construction refers to using certain techniques to extract factual triplets from publicly collected data, processing them, cleaning and summarizing them to obtain ontology, and finally verifying them before adding them to the knowledge library. Considering that the ontology concepts in the household registration field already have standardized definitions and explanations in relevant policies and regulations, we adopted the top-down construction approach. The household registration KG was built based on government-published household registration documents and business question corpus. The process of constructing the KG is shown in Figure 1.

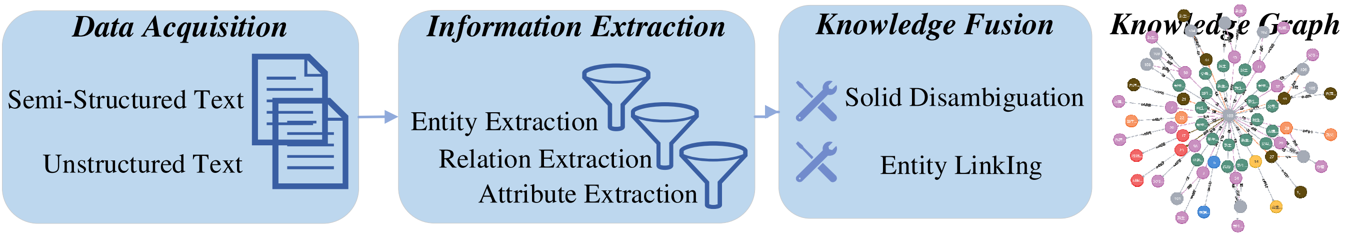


Figure 1. Flowchart of constructing knowledge graph. We preprocessed the collected unstructured documents and semi-structured table data. Through extracting information and knowledge fusion, we improved the graph database, ultimately completing the construction of the knowledge graph.

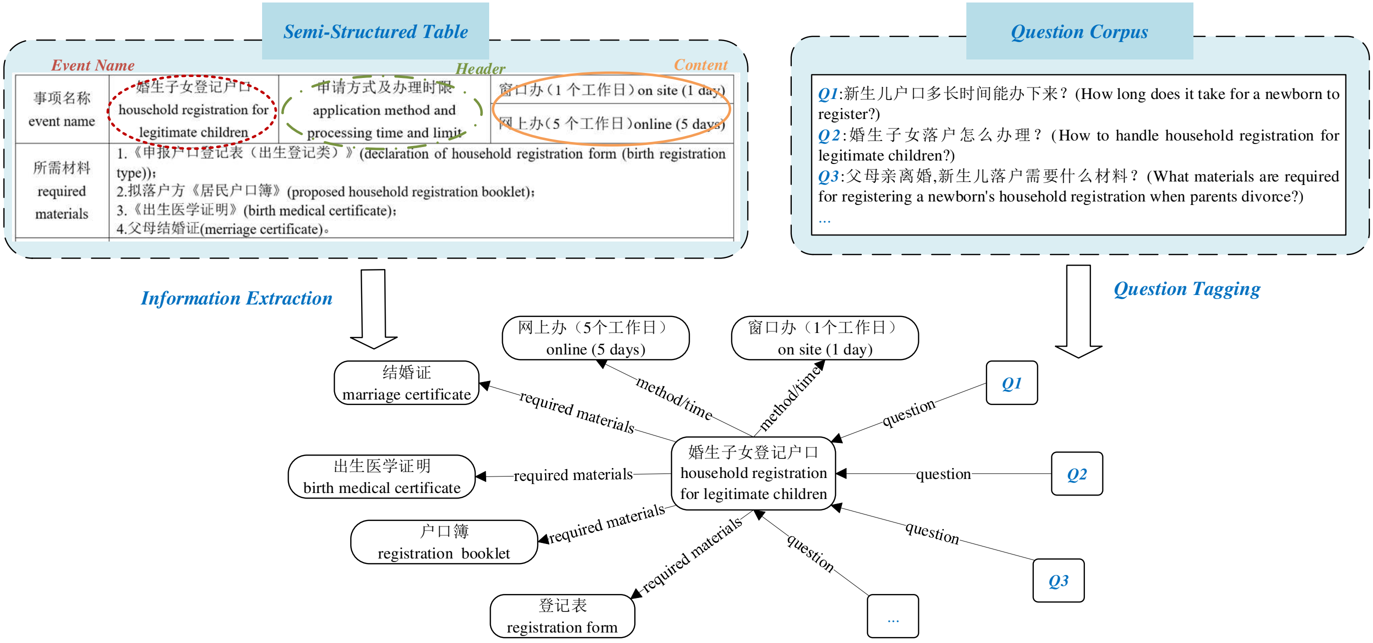
3.1. Data Acquisition

Due to the significant regional differences in China's household registration policy, the requirements for handling the same household registration matter may vary greatly in different cities. Therefore, the knowledge source of the QA system must strictly comply with the household registration policy publicly announced in each specific region. In this paper, the household registration domain KG was constructed using the household registration policy of Wuhan City, Hubei Province, as the data source. The textual data for constructing the household registration KG mainly rely on the Wuhan City Household Registration Business Processing Guidelines [33], (hereinafter referred to as the Guidelines), issued by the Wuhan City Public Security Bureau, and the question corpus dataset compiled by staff of the Household Registration Center as a supplement.

The Guidelines list the handling information for a total of 128 household registration businesses under six major categories and 28 subcategories in a semi-structured data format, totaling 112,400 words. This text summarizes a large number of household registration domain concepts and terminologies, accurately and succinctly describing how to handle various household registration businesses under different constraint conditions. The sentence question corpus dataset consists of 1427 genuine questions recorded by the government service hall inquiry system, and each question corresponds to the household registration business information in the Guidelines. We trained a semantic parsing model using the question corpus dataset to convert the questions into simple queries, which are then used to search for answers in the KG database.

3.2. Information Extraction

We preprocessed the text data and focused on household registration events to extract specific content from semi-structured tables to form event triplets. We then aligned the questions and answers in the question corpus with the household registration item content to obtain the entity relationships in the household registration field. These entities mainly include the business item name, application requirements, required materials, processing procedures, application methods, and processing deadlines. The specific steps for information extraction are as follows, with an illustration shown in Figure 2:



事项名称 event name	婚生子女登记户口 household registration for legitimate children	申请方式及办理时限 application method and processing time and limit	窗口办（1 个工作日）on site (1 day) 网上办（5 个工作日）online (5 days)
所需材料 required materials	1.《申报户口登记表（出生登记类）》(declaration of household registration form (birth registration type)); 2.拟落户方《居民户口簿》(proposed household registration booklet); 3.《出生医学证明》(birth medical certificate); 4.父母结婚证(merriage certificate)。		

Q1:新生儿户口多长时间能办下来？(How long does it take for a newborn to register?)
Q2:婚生子女落户怎么办理？(How to handle household registration for legitimate children?)
Q3:父母亲离婚,新生儿落户需要什么材料？(What materials are required for registering a newborn's household registration when parents divorce?)
...

Figure 2. An example of information extraction. We extracted the contents of the table as entities, and the corresponding table headers as attributes and relationships. For the collected question sentence corpus, we extracted the question sentences as entities and established relationships between the question entity and the event entity involved in the question, resulting in a triple data for the event 'household registration for children born within marriage'.

(1) The Guidelines are semi-structured tables, and the theme tags and headers of the table can be directly extracted as relationships and attributes. The household registration

business event name is extracted as the head entity, while the specific contents in the tables are extracted as tail entities, forming the basic event triplet of the household registration business;

(2) Supplementary explanations for household registration events under different constraint conditions are added as a tail entity to the head entity of the household registration business event name, forming the condition triplet of the household registration event.

(3) The questions from the question corpus were manually labeled, and corresponding relationships were established with the entities in steps (1) and (2), thereby constructing the question triplet of the household registration event.

3.3. Knowledge Representation

In a KG, the triplet is a commonly used representation, denoted as $G = (E, R, F)$, where $E = \{e_1, e_2, \cdots e_m\}$ represents the entity set containing m distinct entities, $R = \{r_1, r_2, \cdots r_n\}$ represents the relation set containing n distinct relationships, and $F = \left\{(e_a, r_b, e_c), \cdots, \left(e_i, r_j, e_k\right)\right\}$ represents the set of fact triplets. In knowledge extracted from household registration business text data, many entities have multiple relationships that cannot be effectively represented using conventional triplet forms. Currently, there are two primary methods for representing multiple relationships: one involves adding the multiple relationships as edge attributes, as exemplified by the knowledge base ConceptNe [34]. However, this method yields redundant relationship data, reducing the efficiency of data retrieval and querying. The second method involves using compound value types (CVT) nodes to represent multiple relationships, as exemplified by the knowledge base Freebase [23]. However, this method requires data to be stored as a graph data structure.

Considering efficiency in querying and the redundancy in relationship data, we used CVT nodes to represent multiple relationships and convert the fact triplets under different circumstances into tuples represented by CVT nodes. CVT nodes are a node type in the Freebase knowledge base that is used to collect multiple attributes of an event and model complex relationships more accurately between entity nodes. We used the CVT nodes to represent household registration events that involve different constraint conditions and complex relationships, as illustrated in Figure 3.

After completing entity relation extraction and deduplication operations, the extracted triplets were reorganized using CVT nodes. The statistics of the quantity and categories of the entity relationships extracted from the document are shown in Table 1. This paper utilized the NEO4J graph database to store multi-tuple data. NEO4J provides Cypher statements to import and query graph data, which is a descriptive graph query language with simple syntax and powerful functions. In this paper, the entity relationships in the Guidelines and the corresponding question corpus were imported into the NEO4J database using the Cypher CREATE statement.

Table 1. Entity relationship data statistics.

Data Sources	Entity Count	Entity Category Count	Relationship Count	Relationship Category Count
Guidelines	622	8	1512	9
Question corpus	1770	12	2564	13
Total	2392	20	4067	22

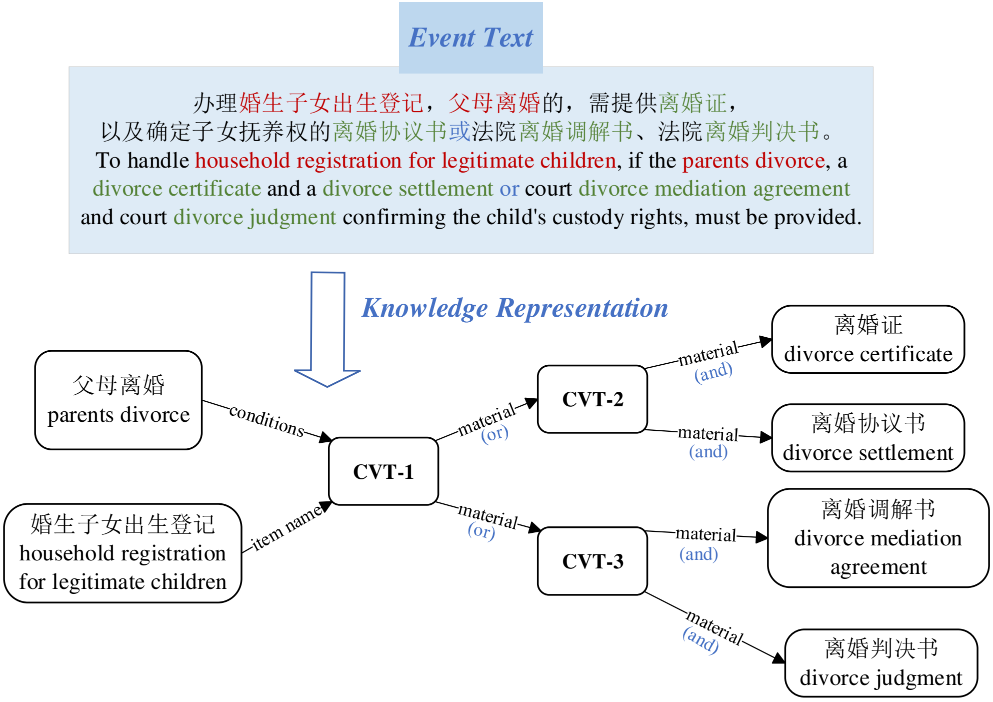

Event Text
办理婚生子女出生登记，父母离婚的，需提供离婚证，
以及确定子女抚养权的离婚协议书或法院离婚调解书、法院离婚判决书。
To handle household registration for legitimate children, if the parents divorce, a divorce certificate and a divorce settlement or court divorce mediation agreement and court divorce judgment confirming the child's custody rights, must be provided.
Knowledge Representation
父母离婚
parents divorce
conditions
婚生子女出生登记
household registration for legitimate children
item name
CVT-1
material (or)
CVT-2
material (or)
CVT-3
material (and)
离婚证
divorce certificate
material (and)
离婚协议书
divorce settlement
material (and)
离婚调解书
divorce mediation agreement
material (and)
离婚判决书
divorce judgment

Figure 3. Example of CVT nodes representation. In the event of household registration for children born within marriage, when the parents divorce, they can choose to provide either one of two sets of documents for registration. One set includes the divorce certificate and divorce settlement, and the other set includes the divorce mediation agreement and divorce judgment. This text involves the use of 'and' and 'or' relationships, which are difficult to express in conventional triple formats, but CVT nodes can handle it effortlessly.

4. Question Answering System

There are presently two methods for retrieving household registration policy information: one is to manually query customer service, which is inefficient and prone to errors, and the other is to search government service websites in various regions (e.g., Hubei Government Services website [35]), which primarily use string-based search methods that cannot accurately interpret user intention. We present a QA system based on a household registration domain KG, which can retrieve information from complex household registration regulations relevant to the user's query, enabling the system to efficiently answer users' questions and improve the efficiency of household registration services.

4.1. System Structure

The QA system consists of two stages: the semantic parsing stage and the answer retrieval stage. In the semantic parsing stage, the QA system parses the question into a simple question composed of a single entity and a single relationship, outputting the household registration event entity and the intention relationship corresponding to the question. We analyzed the household registration event entity relationships in the household registration domain KG and combined them with the question corpus dataset. Overall, there were a total of 622 household registration event entities, and 618 of them had corresponding entries in the question dataset; there were nine types of question intention relationships, and within the question dataset, there were five question intention relationships identified. In addition to these, there were also questions with unclear intentions and those unrelated to household registration affairs, which accounted for a total of seven intention categories.

Since there were a large number of household registration event entities and extremely imbalanced question data, we treated the recognition of household registration event entities in the questions as a text matching task, and used LTP to extract semantic subject roles in the questions to determine the corresponding event entity through the text similarity model. Since there were relatively few question intention relationships, and the imbalance in the dataset was significant, we treated the recognition of the question intention relationships as a text classification task [36] and trained a neural network classifier using the RBMA model. In the answer retrieval stage, we inserted the household registration event entity and the question intention relationship obtained by semantic parsing into a querying statement, and executed the query to obtain the answer. The QA system process is shown in Figure 4.

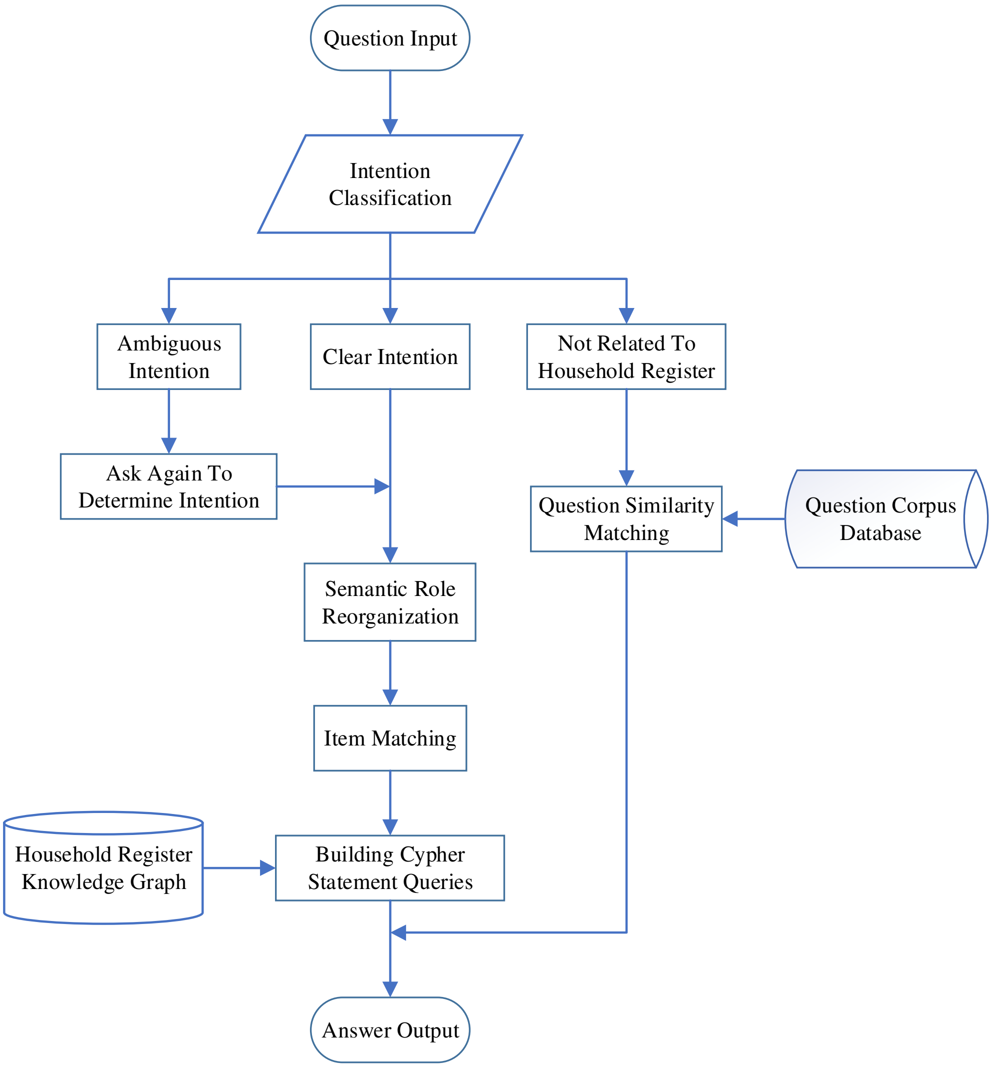

Question Input
Intention Classification
Ambiguous Intention
Clear Intention
Not Related To Household Register
Ask Again To Determine Intention
Question Similarity Matching
Question Corpus Database
Semantic Role Reorganization
Item Matching
Household Register Knowledge Graph
Building Cypher Statement Queries
Answer Output

Figure 4. Flowchart of QA system.

4.2. Semantic Parsing

4.2.1. Question Intention Classification

We conducted a statistical analysis on the question corpus dataset and found that the question intentions in the dataset correspond to five relationships in the KG, as well as some ambiguous and unrelated expressions. Therefore, we classified the question intentions into seven categories, including the five intention categories corresponding to relationships and two special cases. Table 2 lists the seven question intention categories.

Table 2. Question intention category and definition.

Category	Definition
Class 1	Application method and processing time limit
Class 2	Processing location
Class 3	Application conditions
Class 4	Required materials
Class 5	Processing procedures
Class 6	Intention unclear
Class 7	Not related to household registration

In this paper, recognizing the intention of the question was considered a text classification task. We constructed the RBMA model as a classifier that parses and classifies the intention of the question. The model classifies the question process by: encoding input texts semantically using the Robustly Optimized BERT Pretraining Approach (RoBERTa) to obtain the word vector representations of each word; inputting the word vector sequence into a bidirectional long short-term memory (BiLSTM) to capture the contextual semantic information of the sentence; and then using a multi-head attention mechanism (MHA) to extract the essential information from the text [37]. Finally, the classification (CLS) vector representing the entire sentence's semantic is concatenated, and a fully connected layer maps the CLS vector to a predefined set of labels to obtain the text's classification result [38]. The model structure is shown in Figure 5, and the intention category *intention* is ultimately obtained through question parsing.

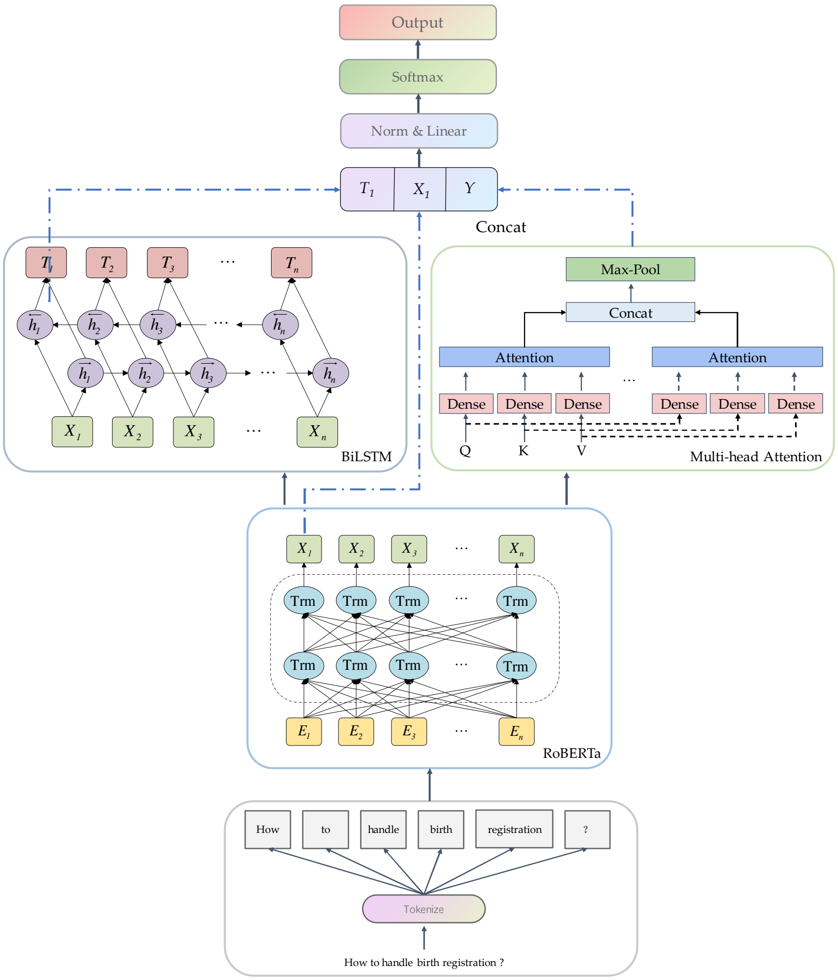


Figure 5. Model structure graph.

(1) RoBERTa Layer

RoBERTa [39] is an improved version of the Bidirectional Encoder Representation from Transformers (BERT) [40] pretraining model, representing a more robust and fine-tuned version of BERT. The RoBERTa model employs a Transformer-based encoder-decoder structure to represent input text as vectors. Specifically, the model converts each input word into its corresponding word vector and feeds them into the encoder in a particular order. The encoder consists of multiple Transformer blocks, each containing multi-headed attention mechanisms and feedforward neural network layers. By iterating through these blocks, the RoBERTa model can gradually convert the input text into a fixed-dimensional vector representation, serving as the output of the model.

The vector representation, $E = \left(E_1^d, E_2^d, \cdots, E_n^d\right)$, of an input text query is generated by a combination of tokenization and encoding procedures. The length of the tokenized query is denoted by n, while the dimensionality of individual word vectors is denoted by d. The extraction of feature vectors through decoder processing, using vector E, yields the output vector $X = (X_1, X_2, \cdots, X_n)$.

(2) BiLSTM Layer

The bidirectional long short-term memory network (BiLSTM) consists of a forward LSTM that processes a sequence and a backward LSTM that processes the sequence in reverse, enabling deep feature extraction of the input context. This allows for effective capture of the position relationships and semantic information between words, which better accommodates longer text sequences while retaining semantic information. Thus, a BiLSTM layer is introduced after the RoBERTa layer to extract features from the context of the input query, in order to more accurately capture global feature information of the input text statement. This compensates for the RoBERTa layer's tendency to forget contextual information.

The BiLSTM model comprises forward and backward LSTM cells, and therefore, the output of the BiLSTM at time t is jointly determined by x_t, $\overrightarrow{h_t}$, and $\overleftarrow{h_t}$. The forward output of the LSTM cell at time t is represented by $\overrightarrow{h_t}$, and the backward output at time t is denoted by $\overleftarrow{h_t}$. The update formula of BiLSTM layer is described below:

$$\overrightarrow{h_t} = \text{LSTM}\left(x_t, \overrightarrow{h_{t-1}}\right) \tag{1}$$

$$\overleftarrow{h_t} = \text{LSTM}\left(x_t, \overleftarrow{h_{t-1}}\right) \tag{2}$$

$$h_t = w_t \cdot \overrightarrow{h_t} + v_t \cdot \overleftarrow{h_t} + b_t \tag{3}$$

In Equations (1)–(3), w_t and v_t represent the weight matrices of the forward and backward LSTM cells, respectively. b_t represents the bias, and h_t denotes the output of the BiLSTM layer at time t. The input feature vector captures the global information of the text sentence through the BiLSTM layer, and the resulting output is represented by $T = (T_1, T_2, \cdots, T_n)$.

(3) Multi-head attention layer

The attention mechanism enables selectively focusing on critical information in text. We employed the multi-head self-attention mechanism (MHA) to extract word dependencies in different semantic spaces [41]. The MHA first conducts multiple linear transformations on the input vectors and then performs attention computation on the transformed vectors to obtain a weighted vector representation. By introducing multiple heads, the MHA can learn more semantic information and improve the model performance [42,43].

The MHA is based on the principle of scaled dot-product attention (SDA), which can be expressed mathematically as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{4}$$

In Equation (4), Q, K, and V represent the query, key, and value matrices used for computing the attention, respectively. The input key and value dimension is denoted by d_k. The attention weight of a value is obtained by calculating the dot product of the query and all keys, dividing the result by $\sqrt{d_k}$, and applying the function denoted by softmax.

For an input vector sequence, multiple linear transformations are first applied to obtain multiple heads (i.e., multiple vector sequences). Then, the attention weight for each head is computed separately, followed by multiplication with the corresponding vectors and summation to obtain the output vector for each head:

$$head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{5}$$

Equation (5) shows that the projection matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ correspond to Q, K, and V respectively. The single-headed output $head_i$ is concatenated to form the final vector representation, and is then dimensionally adjusted by the projection matrix $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ to obtain the multi-headed output:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(head_1, head_2, \cdots, head_n)W^O \tag{6}$$

(4) Linear layer

We concatenate the CLS vector X_1 from the RoBERTa layer, the last sequential output T_1 from the BiLSTM layer, and the pooled output H from the multi-head attention layer, normalize the result, and apply an activation function to obtain vector V_T:

$$V_T = \text{Tanh}(\text{Norm}(\text{Concat}(X_1, T_1, H))) \tag{7}$$

The feature representation, which fuses the semantic information of the textual statements, is input to a fully connected layer. The resulting representation is mapped to the instance label space to obtain the final classification result:

$$Y = \text{Linear}(V_T) \tag{8}$$

4.2.2. Event Matching

After analyzing the questions in the dataset, we found that in actual usage scenarios, sentence components may be missing or unclear due to factors such as limited user expression abilities [44]. As a result, the identification of event entities is frequently less effective; in particular, the input to the QA system consists of spoken audio converted to text, which increases the difficulty of identifying event entities and makes the answer retrieval more difficult.

To address the above issues, we utilized LTP [7] developed by Harbin University of Technology to analyze questions by starting with the sentence structure. The tool performs syntactical analysis on the question, extracts important semantic subject roles. We reorganized them into phrases based on semantic role properties. The resulting phrases were then matched with a predefined statement template. The process partly eliminates noises caused by colloquial language or vague expressions and ultimately identifies the household registration event entity associated with the question.

(1) Semantic Role Analysis

The LTP platform supports user-defined dictionaries. We added commonly used words related to the field of household registration to the LTP dictionary. The question is then segmented, annotated with parts of speech, and labeled with semantic roles. Based on the result of the semantic analysis, the semantic roles were extracted from the clauses and output. Semantic roles and their meanings are listed in Table 3 [45].

Table 3. Semantic role category and definition.

Category	Definition	Category	Definition
A0	causers or experiencers	EXT	extent
A1	patient	FRQ	frequency
A2	semantic role 2	LOC	locative
A3	semantic role 3	MNR	manner
ADV	adverbial	PRP	purpose or reason
BNF	beneficiary	QTY	quantity
CND	condition	TMP	temporal
CRD	coordinated arguments	TPC	topic
DGR	degree	PRD	predicate
DIR	direction	PSR	possessor
DIS	discourse marker	PSE	possessee

The LTP performs semantic role labeling and extracts semantic roles present in interrogative sentences. To illustrate, we take the question ‘Can I apply for my first ID card for non-local household registration in Wuhan?’ as an example. Figure 6 shows the analysis result of the LTP for this question, which provides a list of semantic labeling results:

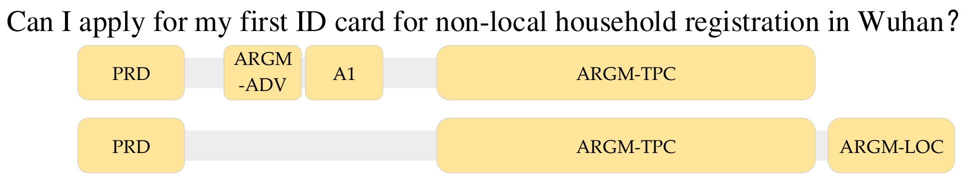

Can I apply for my first ID card for non-local household registration in Wuhan?
PRD
ARGM -ADV
A1
ARGM-TPC
PRD
ARGM-TPC
ARGM-LOC

Figure 6. Example of LTP analysis results. Through LTP, questions are segmented and semantic roles extracted.

[{‘predicate’: ‘apply for’, ‘arguments’: [(‘ARGM-TPC’, ‘non-local household registration’), (‘ARGM-ADV’, ‘first’), (‘A1’, ‘ID card’)]}, {‘predicate’: ‘apply for’, ‘arguments’: [(‘ARGM-TPC’, ‘non-local household registration’), (‘ARGM-LOC’, ‘in Wuhan’)]}].

The resulting list consists of multiple dictionaries where each dictionary contains a predicate ‘predicate’ and a semantic role ‘argument.’ The predicates and semantic roles in each dictionary are then reassembled into a phrase set $P = (p_1, p_2, \cdots, p_m)$, which expresses the key semantic elements of the question, according to the role types, as shown in Table 3. For instance, using semantic role extraction, we obtained the following phrases: ‘non-local household registration apply for first ID card’ and ‘non-local household registration apply for in Wuhan’.

(2) Text similarity matching

The purpose of similarity matching is to calculate the similarity between the phrases extracted from semantic analysis and the event entities stored in the KG. This process can ensure the household registration item names and contextual conditions for the given question, and provides important information to support answer retrieval [46]. The specific process of similarity matching includes the following steps:

- To begin with, the phrases extracted from semantic analysis in the phrase set P are reassembled and concatenated, resulting in the sentence *sent* with extracted semantic role features.
- Search for all CVT nodes that represent household registration events in the KG, extract the corresponding description texts of household registration events for these nodes, and form a candidate set $S = (s_1, s_2, \cdots s_n)$;

- Perform similarity matching between the sentence *sent* and the description texts in the candidate set S to determine the household registration item corresponding to the given question and its CVT node *id* in the KG.

In NLP downstream tasks, determining the similarity between two pieces of text is an essential task called text similarity matching. This task involves transforming input text into vectors to capture semantic information and calculate their similarity. In this paper, we employed pretrained language models to represent the sentence *sent* as a semantic vector e_s, and sentence set S as a semantic vector set $E_S = (e_1, e_2, \cdots, e_n)$. We calculated the cosine similarity between the element vectors in vector set E_S with vector e_s, outputted their maximum result index, which determines the sentence s_k in sentence set S that is most similar to sentence *sent*, along with the corresponding CVT node *id* of s_k. The cosine similarity calculation is provided in (9):

$$cosine(e_1, e_2) = \frac{e_1 \cdot e_2}{\|e_1\|_2 \cdot \|e_2\|_2} \tag{9}$$

The semantic vectors represented by dimensions e_1 and e_2 are identical.

BERT and RoBERTa are highly effective for semantic representation, but these large language models are based on unsupervised learning. To achieve better performance in text similarity subtasks, it is typically necessary to fine-tune the model based on supervised learning. Currently, popular text similarity models, such as Sentence-BERT [47], use average pooling to obtain the mean vector as the sentence vector and have achieved good performance and fast convergence. However, such models have not been optimized for similarity prediction. We used the CoSENT [48] model proposed by Jianlin Su to represent semantic vectors. This model designs a new solution to optimize cosine values and solves the problem of inconsistency between training and prediction.

In text similarity tasks, CoSENT model uses sentence pairs for training, with a denotation that Ω_{pos} refers to the set of all positive sample pairs and Ω_{neg} denotes the set of all negative sample pairs. For any positive sample pair $(i, j) \in \Omega_{pos}$ and negative sample pair $(k, l) \in \Omega_{neg}$, it is desirable to fulfill the following criterion:

$$cosine(ui, uj) > cosine(uk, ul) \tag{10}$$

ui, uj, uk, ul represent the semantic vectors of the respective sentences. In the original cross-entropy loss Equation (11):

$$\log\left(1 + \sum_{i \in \Omega_{neg}, j \in \Omega_{pos}} e^{s_i - s_j}\right) \tag{11}$$

To achieve the prediction target of $s_i < s_j$, $e^{s_i - s_j}$ can be added to log. Therefore, the loss function formula corresponding to (10) can be revised as follows:

$$\log\left(1 + \sum_{(i,j) \in \Omega_{pos}, (k,l) \in \Omega_{neg}} e^{\lambda(\cos(uk,ul) - \cos(ui,uj))}\right) \tag{12}$$

Here, $\lambda > 0$ is a hyperparameter. CoSENT model fine-tunes the model by designing a loss function that optimizes *cosine* value, enabling the model to achieve better convergence speed and final performance in text similarity tasks compared to Sentence-BERT.

The semantic vectors represented by dimensions e_1 and e_2 are identical.

4.3. Answer Retrieval

After intention classification and event matching, we obtain the intention type *intention* and event node ID value *id*. The answer retrieval is divided into two methods by the intention of the question. The flow chart in Figure 7 illustrates the process:

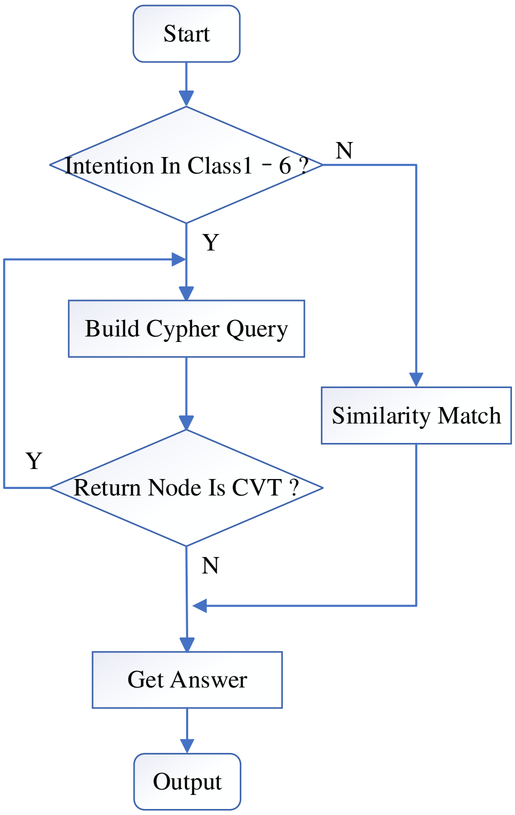


Figure 7. Flowchart of answer query process.

(1) The first method applies to *intention* belonging to class 1 to class 6, which fills in predesigned Cypher query templates with *intention* and *id* of the event node to construct querying statements. The Cypher query template is shown in Figure 8. If the query returns an answer node, the node content will be output as an answer. If a CVT node is returned, the node will be re-inserted into the Cypher query template and the query will continue until an answer node is returned, and use the text content stored in the answer node as the output;

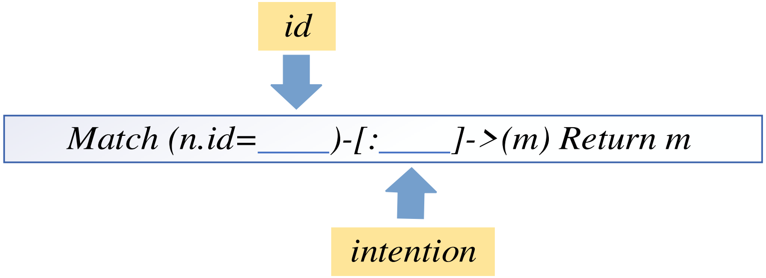


Figure 8. Cypher query statement template.

(2) The second query method is used when *intention* is class 7, which means the question is unrelated to household registration processing. Here, we use the CoSENT model to retrieve the most similar question from the question corpus to generate the answer.

5. Experiment and Analysis

5.1. Data Augmentation

Currently, the training of the model relies heavily on manually annotated corpora, which may not achieve good results when the dataset is small in size [49–51]. Data augmentation refers to the use of various techniques and methods to expand the training dataset, thus improving the model's performance and robustness. Jason Wei et al. [52] proposed four simple operations, including synonym substitution, random deletion, random swapping, and random insertion, to prevent overfitting and enhance model generalization. Ateret et al. [53] used a generative language model GPT-2 for text data augmentation and achieved excellent augmentation results in few-shot scenarios.

As there are currently no open-source question corpus data in the household registration field, all the data used to train the model in this paper come from the question corpus collected from Wuhan Municipal Government Service Center's inquiry system. After removing a small number of invalid and abandoned question corpus due to household registration policy changes, a total of 1427 authentic questions related to household registration were obtained. We labeled all these questions and obtained the question classification dataset required for the experiment. As shown in Fig 10, through visualization and analysis of the dataset, we found that the dataset is extremely unbalanced: the number of data samples for class 5 questions is more than three times that of class 3 questions. Since the smaller class size in the dataset can lead to overfitting during model training, we used the GPT-3.5-turbo generative language model to expand the total amount of the dataset by two methods: synonym substitution and random insertion of irrelevant words.

To enable LLM to rewrite questions according to the requirements, we need to design prompts given to GPT. Currently, there are two main types of prompt templates: cloze prompts [54] and prefix prompts [55]. Since question rewriting belongs to the sentence generation task, the prefix prompt method is often more beneficial because such tasks align well with the model's left-to-right property [56]. We designed prompts based on the CRISPE prompt framework [57] provided by Matt Nigh, and the specific steps are shown in Table 4.

Table 4. Prompts Creation Framework.

Step	Interpretation	Prompt
Capacity and Role	What role (or roles) should ChatGPT act as?	As a user who is going to handle household registration business
Insight	Provides the behind the scenes insight, background, and context to your request.	You have questions about the handling information of some household registration matters
Statement	What you are asking ChatGPT to do.	Please rewrite the given question, provide a similar question
Personality	The style, personality, or manner you want ChatGPT to respond in.	Ensure that the original meaning of the sentence is preserved. The rewriting method includes synonym substitution and random insertion of irrelevant words
Experiment	Asking ChatGPT to provide multiple examples to you.	The sentences that need to be rewritten are:

The final prompt is as follows: 'As a user who is going to handle household registration business, you have questions about the handling information of some household registration matters. Please rewrite the given question, provide a similar question, and ensure that the original meaning of the sentence is preserved. The rewriting method includes synonym substitution and random insertion of irrelevant words. The sentences that need to be rewritten are:'. We rewrote the 1427 questions multiple times and finally obtained a dataset of 7055 questions.

5.2. Question Intention Classification Experiment

5.2.1. Comparison Model

In order to assess the efficacy of the RBMA model in intention classification tasks, this study conducted comparative experiments by comparing the RBMA model against several prevailing text classification models. Furthermore, we conducted ablation experiments to examine the classification performance of RoBERTa, BiLSTM, and MultiHeadAttention. Specifically, we compared the RBMA model against seven other models:

- DPCNN [58]: A deep network-based classification model that extracts text dependency features over long distances;
- TextRCNN [59]: A bidirectional RNN-based model that leverages context information through max-pooling to extract important features for text classification;
- BERT: A model that leverages the BERT architecture to parse the semantics of a sentence and obtain corresponding sentence embeddings for text classification tasks;
- BERT-BiLSTM-MultiHeadAttention: A model that employs BERT to obtain sentence embeddings, BiLSTM to extract contextual information, and MHA to consider multiple aspects of the sentence and perform text classification based on the combination of all information;
- RoBERTa: A model that utilizes RoBERTa pretrained models to obtain semantic embeddings of sentences for text classification tasks;
- RoBERTa-BiLSTM: A model that combines RoBERTa's semantic parsing with BiLSTM's contextual information for text classification;
- RoBERTa-MultiHeadAttention: A model that incorporates MHA with RoBERTa's semantic embeddings to consider multiple aspects of the sentence and performs text classification based on the combination of all information.

For each category's prediction, we calculated TP, TN, FN, and FP, which correspond to the number of true positive, true negative, false negative, and false positive, respectively. We evaluated the performance of the question matching model using three metrics: precision, recall, and F1 score. The formulas for these metrics are shown below:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$\text{f1} - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{15}$$

To ensure optimal generalization abilities of the model during the training process, we saved the model parameter files when the performance of each model on the validation set was at its best, with the premise that the model was not overfitting. Then, the accuracy of the final answer when the QA system used the model for intention classification was tested on the test set. The formula for accuracy calculation is presented below:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{16}$$

The hyperparameter settings for the model experiment are shown in Table 5.

Table 5. Hyperparameter settings.

Parameter	Value
Embedding dim	768
BiLSTM layers num	2
batch_size	64
Epoch	30
Learning rate	5×10^{-4}
Dropout	0.2

5.2.2. Dataset Configuration

For the intention classification model experiments, the data processing methods for the training, validation, and testing sets were as follows. The number of samples in each dataset is presented in Figure 9:

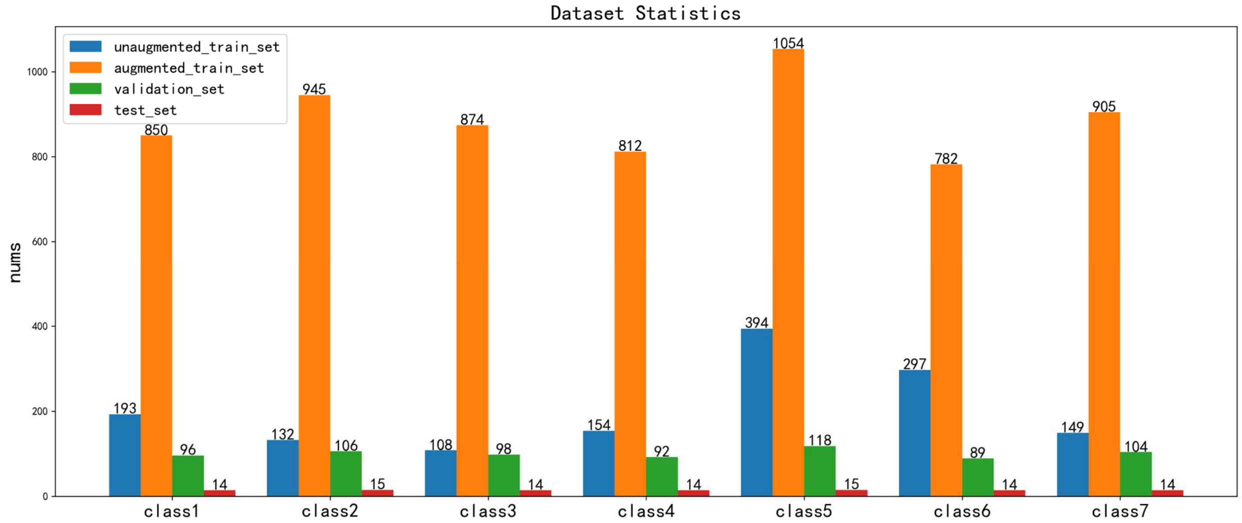


Figure 9. Statistics of each dataset.

- *unaugmented_train_set*: 1427 genuine question collected and labeled from the question corpus dataset;
- *augmented_train_set*: Randomly selecting 90% of the data from the 7055 datasets obtained through data augmentation resulted in a total of 6222 questions;
- *validation_set*: Extracting the remaining 10% of the data from the 7055 datasets obtained through data augmentation resulted in a total of 833 questions;
- *test_set*: 100 genuine questions were randomly selected from the *unaugmented_train_set*, and these questions were manually rewritten to create an additional 100 questions as the test set.

5.2.3. Experimental Results and Analysis

(1) Model Comparison Experiment

To evaluate the performance of the RBMA model in text classification tasks, we conducted comparative experiments between the RBMA model and seven other models on the *augmented_train_set* and *validation_set*. The accuracy, precision, recall, and F1 scores were calculated for each model. The experimental results are presented in Table 6.

Table 6. Model comparison experimental results.

Model	Precision	Recall	F1-Score	Accuracy
DPCNN	0.9456	0.761	0.8236	0.73
TextRCNN	0.951	0.8218	0.8817	0.78
BERT	0.978	0.8762	0.9215	0.85
BERT-BiLSTM-MultiHeadAttention	0.9846	0.9374	0.9591	0.9
Roberta	0.9805	0.9004	0.9371	0.87
RoBERTa-BiLSTM	0.9836	0.9459	0.9628	0.91
RoBERTa-MultiHeadAttention	0.9856	0.9374	0.9595	0.89
RoBERTa-BiLSTM-MultiHeadAttention (trained on unaugmented data)	0.8999	0.8965	0.8969	0.87
RoBERTa-BiLSTM-MultiHeadAttention (trained on augmented data)	0.9949	0.9615	0.9774	0.93

Table 6 shows that the RBMA model outperforms the four baseline models, namely DPCNN, TextRCNN, BERT, and RoBERTa. It can be seen that RBMA compares with RoBERTa, having the most significant improvement in precision, recall, and F1 score by 1.44%, 6.11%, and 4.03%, respectively. This improvement results in a 6% increase in the accuracy of the final answer. The results indicate that the RBMA model performs better in intention classification tasks than the current mainstream text classification models. During the ablation experiment, the RBMA model demonstrated improvements in various evaluation metrics compared to the RoBERTa-BiLSTM and RoBERTa-MultiHeadAttention models. This validates the effectiveness of combining the ability of the BiLSTM layer to extract contextual information from sentences with the multi-level feature representation of the MHA mechanism.

(2) Impact of Text Data Augmentation on the Model

There are seven categories of intention for household-registration-related questions. The specific definitions for each category are presented in Table 2. In order to explore the RBMA model's ability to classify each intention category and the impact of text data augmentation on the model's prediction performance, this study conducted comparative experiments on the *unaugmented_train_set* and *augmented_train_set*. The model's prediction performance was then validated on the *validation_set*, and the evaluation metrics for each category were recorded. The specific data statistics are presented in Table 7. The average values of each evaluation metric and the loss function curve comparison are presented in Figures 10 and 11, respectively.

Table 7. Train dataset comparison experimental results.

Model	Train Dataset	Precision	Recall	F1-Score
Class 1	unaugmented	0.9516	0.9043	0.9271
	augmented	0.9974	0.9893	0.9813
Class 2	unaugmented	0.7953	0.8761	0.8327
	augmented	0.9902	0.9576	0.9271
Class 3	unaugmented	0.8735	0.9858	0.9244
	augmented	0.9911	0.9904	0.9897
Class 4	unaugmented	0.9439	0.9293	0.9354
	augmented	0.9986	0.9973	0.9961
Class 5	unaugmented	0.954	0.9537	0.9537
	augmented	0.9981	0.9952	0.9924
Class 6	unaugmented	0.8738	0.8953	0.8843
	augmented	0.9921	0.9837	0.9754
Class 7	unaugmented	0.9089	0.8372	0.8709
	augmented	0.9965	0.989	0.9816
Average	unaugmented	0.8998	0.8965	0.8969
	augmented	0.9949	0.9615	0.9774

According to the comparison of the evaluation metrics for each category in Table 7, text data augmentation significantly improves the prediction performance of the model for each category, which confirms the significant improvement in the model's training performance through text data augmentation. However, we found that regardless of whether the model was trained on the *unaugmented_train_set* or the *augmented_train_set*, the recognition accuracy for class 2 was lower compared to other categories. Upon analyzing the class 2 questions in the dataset, we found that questions related to processing procedures were highly colloquial and some of them were easily confused with class 3 and class 4 questions. For example, 'What are the procedural requirements for applying for death registration of a household?' and 'What are the procedures and documents required for changing the head of household?' These samples make it difficult for the model to learn, resulting in weak prediction performance for this intention category.

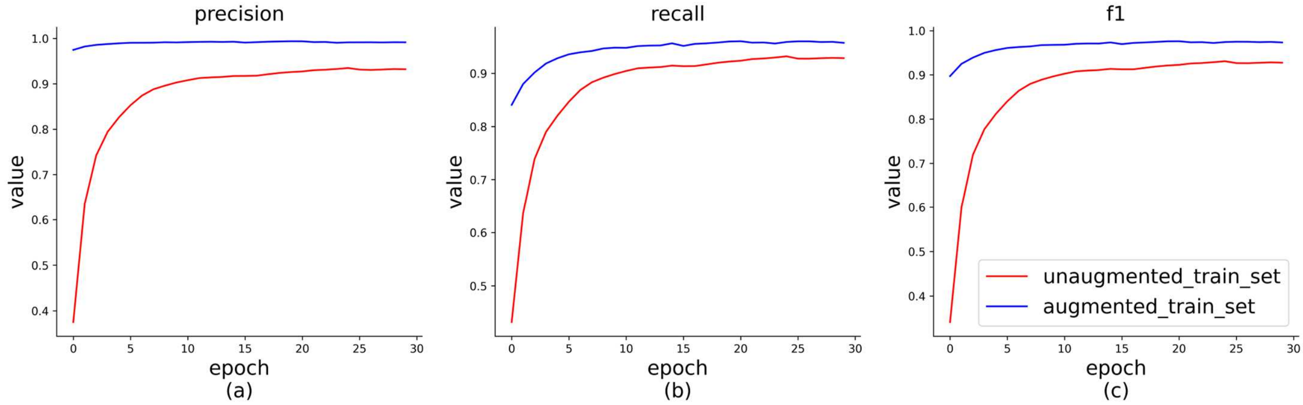


Figure 10. Comparison of various evaluation indicators. (**a**) Comparision of precision curves. (**b**) Comparision of recall curves. (**c**) Comparision of f1 curves.

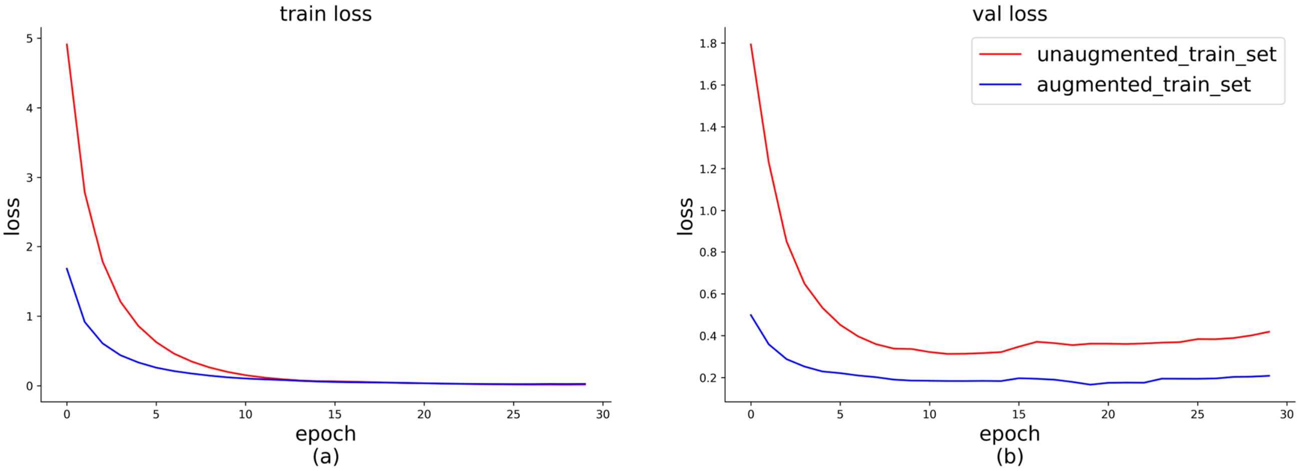


Figure 11. Comparison of loss function curves. (**a**) Comparision of train loss curves. (**b**) Comparision of validation loss curves.

As shown in the evaluation metric average curve comparison in Figure 10, when the model is trained on the *augmented_train_set*, the model's fitting speed increases, and all evaluation metrics improve significantly, resulting in a marked improvement in prediction performance. Similarly, Figure 11 demonstrates that the model's convergence speed accelerates, and the degree of overfitting is reduced, resulting in improved generalization ability.

5.3. Event Entity Matching Experiment

5.3.1. Experimental Method Comparison

After manually annotating the question dataset, we conducted a similarity comparison experiment to verify the effectiveness of using LTP to extract semantic role phrases for similarity matching. The experiment compared two processing methods:

(1) Match the raw question text directly with the corresponding event entity for text similarity without any processing, and record the experimental accuracy and similarity values.

(2) Use LTP for syntactic analysis of the question, extract semantic role phrases, reorganize the phrases into sentences and then conduct text similarity matching with the corresponding event entities, and record the experimental accuracy and similarity values.

We performed text similarity experiments on a dataset of 7055 questions, and calculated similarity values and presented the experimental results using a box plot.

5.3.2. Experimental Results and Analysis

The box plots displaying the results of the two methods are presented in Figure 12. Overall, the use of LTP to extract and reorganize semantic role phrases in method (2) improved the median, mean, maximum, and minimum event entity similarity values compared to method (1). We compared the similarity values of the two methods for the same questions and found that 56.71% of the questions showed improved similarity values with the use of method (2), while 12.74% of the questions showed no change. However, 30.55% of the questions exhibited decreased similarity values after processing with method (2).

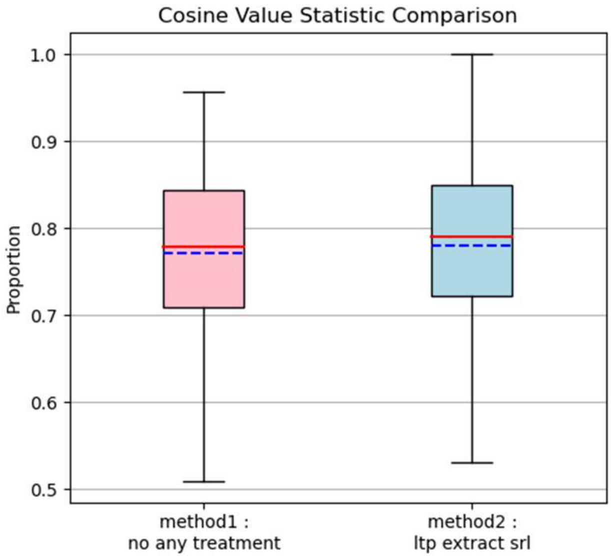


Figure 12. The experimental box plot comparison. The red line represents the median, and the blue line represents mean.

We analyzed the questions that exhibited decreased similarity values and found that their expressions were clearer, and had well-formed syntax and almost no colloquial expression. Therefore, such sentences may have become disordered and repeated when their syntax and semantics were reorganized using LTP, leading to a decrease in similarity values. These observations underscore that while method (2) is effective in enhancing the matching capability of event entities for questions with missing grammatical elements or ambiguous expressions, it may reduce the matching accuracy of properly structured sentences.

6. Conclusions

This paper presents a QA system based on a household registration domain KG using the household registration documents released by Wuhan city as an example. Considering the generality of the QA system, we divide the KG and QA system into two parts for separate construction: The KG processes the collected semi-structured and unstructured documents into structured data to establish a graph database that provides information support for the QA system. The QA system is trained on manually collected and data-augmented question sentence corpus to achieve accurate question parsing results, and then it retrieves answers from the KG. Policy changes and updates can be accommodated by re-establishing the KG to apply new policies and regulations. The QA system can adapt to different regional question expressions by adding question sentence corpus for further training. We validate the effectiveness of the QA system by comparing two semantic parsing steps. Our experimental results demonstrate that 56.71% of questions obtain a significant improvement in similarity calculation after reorganizing the sentence using

LTP. This approach can efficiently resolve problems caused by the lack of grammatical components and unclear expressions in oral language expression. Additionally, the RBMA model outperforms several widely used text classification models with an F1 value of 97.74%, precision of 99.49%, and recall of 96.15% in identifying question intentions. Finally, we apply the popular GPT3.5 generative language model to augment the dataset and reduce the impact of data imbalance. Our QA system achieves an accuracy of 87% in the unaugmented dataset and 93% in the augmented one after training.

Although the QA system based on the household registration domain KG in this paper achieved certain performance results, it still has some limitations: (1) Due to the limited size of the question corpus dataset, despite the use of data augmentation, the amount of data remains insufficient, which limits the performance improvement of the model. We plan to collect more language resources or manually generate more questions to expand the dataset and improve the accuracy of the QA system. (2) In the event entity matching experiment, some questions experienced a decrease in similarity values after processing with LTP. Therefore, we will compare the similarity values before and after processing to ensure that sentences with normal expressions do not lose accuracy due to LTP processing. (3) Currently, for unrelated questions, we retrieve similar question in the question corpus. However, such a small-scale corpus often outputs irrelevant or completely erroneous answers. To address this issue, we plan to use other generative LLM to enhance the system's robustness and improve the performance of answering general questions.

Author Contributions: Conceptualization, S.H. and H.Z.; methodology, S.H.; software, S.H.; validation, S.H., H.Z. and W.Z.; formal analysis, W.Z.; investigation, W.Z.; resources, W.Z.; data curation, S.H.; writing—original draft preparation, S.H.; writing—review and editing, H.Z.; visualization, S.H.; supervision, H.Z.; project administration, H.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work search was funded by the Science and Technology Department of Hubei Province, grant number 2022BAA051.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, W.; Deng, Y.; Liang, Y.; Lei, K. Answer Category-Aware Answer Selection for Question Answering. *IEEE Access* **2020**, *9*, 126357–126365. [CrossRef]
2. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*; Pearson: London, UK, 2014; Volume 3.
3. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774. [CrossRef]
4. Zeng, A.; Liu, X.; Du, Z.; Wang, Z.; Lai, H.; Ding, M.; Yang, Z.; Xu, Y.; Zheng, W.; Xia, X.; et al. Glm-130b: An open bilingual pre-trained model. *arXiv* **2022**, arXiv:2210.02414.
5. Chapman, D. Geographies of self and other: Mapping Japan through the koseki. *Asia Pac. J.* **2011**, *9*, 1–10.
6. Cullen, M.J. The Making of the Civil Registration Act of 1836. *J. Ecclesiastical Hist.* **1974**, *25*, 39–59. [CrossRef]
7. Che, W.; Feng, Y.; Qin, L.; Liu, T. N-LTP: An Open-source Neural Language Technology Platform for Chinese. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (Emnlp 2021): Proceedings of System Demonstrations, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 42–49.
8. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale simple question answering with memory networks. *arXiv* **2015**, arXiv:1506.02075.
9. Bao, J.; Duan, N.; Yan, Z.; Zhou, M.; Zhao, T. Constraint-based question answering with knowledge graph. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2503–2514.
10. Unger, C.; Bühmann, L.; Lehmann, J.; Ngomo, A.-C.N.; Gerber, D.; Cimiano, P. Template-Based Question Answering over RDF Data. In Proceedings of the 21st International Conference on World Wide Web, WWW'12, Lyon, France, 16–20 April 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 639–648. [CrossRef]
11. Zheng, W.; Zou, L.; Lian, X.; Yu, J.X.; Song, S.; Zhao, D. How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD'15, Melbourne, Australia, 31 May–4 June 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1809–1824. [CrossRef]

12. Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.; Wang, W. KBQA: Learning question answering over QA corpora and knowledge bases. *arXiv* **2019**, arXiv:1903.02419. [CrossRef]
13. Bast, H.; Haussmann, E. More accurate question answering on freebase. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 1431–1440.
14. Abujabal, A.; Yahya, M.; Riedewald, M.; Weikum, G. Automated template generation for question answering over knowledge graphs. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1191–1200.
15. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; pp. 2736–2744.
16. Chen, Y.; Li, H.; Qi, G.; Wu, T.; Wang, T. Outlining and Filling: Hierarchical Query Graph Generation for Answering Complex Questions Over Knowledge Graphs. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8343–8357. [CrossRef]
17. Xu, K.; Wu, L.; Wang, Z.; Yu, M.; Chen, L.; Sheinin, V. Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. *arXiv* **2018**, arXiv:1808.07624.
18. Green, B.; Wolf, A.; Chomsky, C.; Laughery, K. BASEBALL: An Automatic Question Answerer. In *Readings in Natural Language Processing*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1986; pp. 545–549.
19. Wa, W. Lunar rocks in natural english: Explorations in natural language question answering. *Fundam. Stud. Computer Sci. Netherl. Da.* **1977**, *5*, 521–569.
20. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
21. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia–A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* **2015**, *6*, 167–195. [CrossRef]
22. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250.
23. Lukovnikov, D.; Fischer, A.; Lehmann, J.; Auer, S. Neural network-based question answering over knowledge graphs on word and character level. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1211–1220.
24. Deng, Y.; Zhang, W.; Xu, W.; Shen, Y.; Lam, W. Nonfactoid Question Answering as Query-Focused Summarization with Graph-Enhanced Multihop Inference. *IEEE Trans. Neural Networks Learn. Syst.* 2023; early access. [CrossRef]
25. Shen, Y.; Ding, N.; Zheng, H.-T.; Li, Y.; Yang, M. Modeling relation paths for knowledge graph completion. *IEEE Trans. Knowl. Data Eng.* **2020**, *33*, 3607–3617. [CrossRef]
26. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
27. Abu-Salih, B. Domain-specific knowledge graphs: A survey. *J. Netw. Comput. Appl.* **2021**, *185*, 103076. [CrossRef]
28. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef]
29. Jiang, Z.; Chi, C.; Zhan, Y. Research on Medical Question Answering System Based on Knowledge Graph. *IEEE Access* **2021**, *9*, 21094–21101. [CrossRef]
30. Du, Z.Y.; Yang, Y.; He, L. Question answering system of electric business field based on Chinese knowledge map. *Comput. Appl. Softw.* **2017**, *34*, 153–159.
31. Aghaei, S.; Raad, E.; Fensel, A. Question answering over knowledge graphs: A case study in tourism. *IEEE Access* **2022**, *10*, 69788–69801. [CrossRef]
32. Liu, Q.; Li, Y.; Duan, H.; Liu, Y.; Qin, Z. Knowledge graph construction techniques. *J. Comput. Res. Dev.* **2016**, *53*, 582–600.
33. Wuhan City Household Registration Business Processing Guidelines. Available online: http://www.wuhan.gov.cn/gfxwj/sbmgfxwj/sgaj_79493/202301/t20230104_2124417.shtml (accessed on 9 June 2023).
34. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
35. Hubei Government Service Network. Available online: http://zwfw.hubei.gov.cn/ (accessed on 9 June 2023).
36. Zhou, W.; Liu, J.; Lei, J.; Yu, L.; Hwang, J.-N. GMNet: Graded-feature multilabel-learning network for RGB-thermal urban scene semantic segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 7790–7802. [CrossRef] [PubMed]
37. Yang, S.; Li, Q.; Li, W.; Li, X.; Liu, A.-A. Dual-level representation enhancement on characteristic and context for image-text retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 8037–8050. [CrossRef]
38. Li, L.; Wang, P.; Zheng, X.; Xie, Q.; Tao, X.; Velásquez, J.D. Dual-interactive fusion for code-mixed deep representation learning in tag recommendation. *Inf. Fusion* **2023**, *99*, 101862. [CrossRef]
39. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
40. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; AN, G.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in neural information processing systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.

42. Lu, S.; Liu, M.; Yin, L.; Yin, Z.; Liu, X.; Zheng, W. The multi-modal fusion in visual question answering: A review of attention mechanisms. *PeerJ Comput. Sci.* **2023**, *9*, e1400. [CrossRef]
43. Lu, S.; Ding, Y.; Yin, Z.; Liu, M.; Liu, X.; Zheng, W.; Yin, L. Improved Blending Attention Mechanism in Visual Question Answering. *Comput. Syst. Sci. Eng.* **2023**, *47*, 1149–1161. [CrossRef]
44. Chen, J.; Wang, Q.; Cheng, H.H.; Peng, W.; Xu, W. A Review of Vision-Based Traffic Semantic Understanding in ITSs. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 19954–19979. [CrossRef]
45. Appendix—LTP4 4.1.4 Documents. Available online: https://ltp.readthedocs.io/zh_CN/latest/appendix.html (accessed on 20 June 2023).
46. Xiong, Z.; Zeng, M.; Zhang, X.; Zhu, S.; Xu, F.; Zhao, X.; Wu, Y.; Li, X. Social Similarity Routing Algorithm based on Socially Aware Networks in the Big Data Environment. *J. Signal Process. Syst.* **2022**, *94*, 1253–1267.
47. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
48. CoSENT: A More Effective Sentence Vector Scheme than Sentence-BERT-Scientific Spaces. Available online: https://spaces.ac.cn/archives/8847 (accessed on 10 June 2023).
49. Liu, X.; Shi, T.; Zhou, G.; Liu, M.; Yin, Z.; Yin, L.; Zheng, W. Emotion classification for short texts: An improved multi-label method. *Humanit. Soc. Sci. Commun.* **2023**, *10*, 1–9. [CrossRef]
50. Cheng, L.; Yin, F.; Theodoridis, S.; Chatzis, S.; Chang, T.-H. Rethinking Bayesian learning for data analysis: The art of prior and inference in sparsity-aware modeling. *IEEE Signal Process. Mag.* **2022**, *39*, 18–52. [CrossRef]
51. Zhang, Y.; Shao, Z.; Zhang, J.; Wu, B.; Zhou, L. The effect of image enhancement on influencer's product recommendation effectiveness: The roles of perceived influencer authenticity and post type. *J. Res. Interact. Mark.* 2023. [CrossRef]
52. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
53. Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; Kantor, A.; Kour, G.; Shlomov, S.; Tepper, N.; Zwerdling, N. Do not have enough data? Deep learning to the rescue! In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 7383–7390.
54. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language models as knowledge bases? *arXiv* **2019**, arXiv:1909.01066.
55. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv* **2021**, arXiv:2101.00190.
56. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* **2023**, *55*, 195:1–195:35. [CrossRef]
57. Nigh, M. ChatGPT3 Prompt Engineering. 24 June 2023. Available online: https://github.com/mattnigh/ChatGPT3-Free-Prompt-List (accessed on 25 June 2023).
58. Johnson, R.; Zhang, T. Deep pyramid convolutional neural networks for text categorization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 562–570.
59. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, Austin, TX, USA, 25–30 January 2015.

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