Retrieving Chinese Questions and Answers Based on Deep-Learning Algorithm

Huan Wang 1, Jian Li 1 and Jiapeng Wang 2,*

1 Beijing Modern Manufacturing Industry Development Research Base, College of Economics and Management, Beijing University of Technology, Beijing 100124, China; wanghai0828@emails.bjut.edu.cn (H.W.); lijiansem@bjut.edu.cn (J.L.)
2 Information Technology Department, Xiaomi Inc., Beijing 100085, China
* Correspondence: jiapengwang@xiaomi.com

Abstract: Chinese open-domain reading comprehension question answering is a task in the field of natural language processing. Traditional neural network-based methods lack interpretability in answer reasoning when addressing open-domain reading comprehension questions. This research is grounded in cognitive science’s dual-process theory, where System One performs question reading and System Two handles reasoning, resulting in a novel Chinese open-domain question-answering retrieval algorithm. The experiment employs the publicly available WebQA dataset and is compared against other reading comprehension methods, with the F1-score reaching 78.66%, confirming the effectiveness of the proposed approach. Therefore, adopting a reading comprehension question-answering model based on cognitive graphs can effectively address Chinese reading comprehension questions.

Keywords: dual-process theory; cognitive graph; Chinese open-domain reading comprehension question answering

MSC: 68T50

1. Introduction
1.1. The Core Concept of Cognitive Graph

The cognitive graph is inspired by the dual-process theory of human cognitive processes [1]. This theory regards human reading comprehension as comprising two distinct cognitive processes: “quickly focusing attention on relevant entities” and “analyzing sentence semantics for inference”. In cognitive science, the well-known “dual-process theory” posits that human cognition operates through two systems. System One functions as an intuitive, unconscious thinking system, relying on experiences and associations. In contrast, System Two represents the unique logical reasoning ability of humans, utilizing knowledge stored in working memory to perform slower but reliable logical reasoning. System Two is the unique logical reasoning ability of humans, relying on knowledge in working memory to perform slow but reliable logical reasoning. System Two is explicit, requiring conscious control, and represents the manifestation of human higher intelligence. The cognitive graph leverages System One to query states and construct the graph through relevant entity recognition models. Subsequently, System Two learns hidden representations of contextual information on graph nodes and performs interpretable relationship reasoning.

The essence of the cognitive graph lies in minimizing information loss during graph construction while retaining the graph structure for interpretable relationship reasoning. Simultaneously, it transfers the burden of information processing to retrieval and natural language understanding algorithms.
1.2. Basic Concept of Chinese Reading Comprehension Question Answering

With the development of information technology, available data resources have experienced explosive growth, and users require powerful retrieval tools to find desired information from large datasets. Various data retrieval systems, represented by search engines, have had a significant impact and provided great convenience to users. However, they also exhibit several drawbacks: the system can return to users a batch of sorted document links, and users themselves need to browse through them to locate genuinely useful information in order to find answers. Consequently, the quality of the query terms constructed by users profoundly affects the efficiency and performance of the retrieval system.

Open-domain question answering and reading comprehension [2], also known as OpenQA, aims to provide accurate answers to natural language questions without being limited to a specific domain. Its distinctive feature is that users can express their queries in natural language, and the system automatically retrieves precise answers from various data resources. The scope of user questions is not confined to a specific application or domain. In contrast to traditional machine reading comprehension (MRC) tasks [3], where given a question, the system provides answers from a single passage or document, Open-QA requires searching for the answer within a collection of documents or the entire web.

Early open-domain question-answering systems utilized non-parametric models, such as TF-IDF or BM25 [4], to retrieve answers from a fixed set of documents, with the answer scope extracted using neural reading comprehension models [5]. These methods performed well on single-hop questions, where they could answer questions based on individual paragraphs. However, they often struggled to retrieve evidence required for answering multi-hop questions. Multi-hop question answering typically involves finding multiple supporting paragraphs, where one supporting paragraph might have little lexical overlap or semantic relationship with the original question. Subsequent open-domain QA approaches employed end-to-end models to jointly retrieve and read documents. These methods trained and unified various modules in the neural network to retrieve answers from given documents [6]. However, these approaches compressed necessary information into the embedding space, leading to the lack of capturing semantic information about entities’ vocabulary or terminology. Consequently, challenges persisted in dealing with entity-centered question-answering tasks [7], and these model-based methods still faced issues concerning answer interpretability.

In recent years, with the development of knowledge graphs [8], some methods have attempted to utilize existing facts or relationships within the knowledge graph to infer new relationships and obtain answers, thereby addressing interpretability concerns. Currently, knowledge graph reasoning can be broadly divided into two categories: methods based on logical symbols [9,10] (ontology axioms or symbolic rules) and methods based on representation learning [11]. While traditional methods based on logical symbols offer interpretability, they struggle to handle implicit and uncertain knowledge. On the other hand, representation learning-based methods can capture implicit knowledge, significantly improving reasoning efficiency, making them the mainstream technique for knowledge graph reasoning. However, when using knowledge graphs for knowledge base question answering (KBQA) [12], it is often assumed that there are enough triple instances of entities or relationships in the existing knowledge graph to train vector representations. In open-domain question answering, the existing knowledge graph may not contain entities or relationships present in the questions, leading to a lack of corresponding training instances. This presents a challenge for knowledge graph reading comprehension question-answering methods.

Chinese reading comprehension question answering aims to find answers to questions from a large collection of documents. In current methods, although end-to-end approaches can achieve satisfactory results, they lack reasoning with answer paths. On the other hand, using knowledge graphs with reasoning paths for question answering requires a knowledge graph with reasoning paths, which is not available in existing open-domain large-scale knowledge graphs for assisted reasoning.
Cognitive graph is inspired by human cognitive processes [7], which categorize question answering into two different thinking processes: “quickly focusing attention on relevant entity information” and “analyzing sentence semantics for inference”. In cognitive science, the well-known “dual-process theory” posits that human cognition consists of two systems. System 1 is an intuition-based, unconscious thinking system that relies on experience and associations. On the other hand, System 2 is the unique logical reasoning ability of humans, utilizing knowledge from working memory for slow but reliable logical inference. System 2 is explicit and requires conscious control, representing the manifestation of human higher intelligence.

In response to the limitations of previous Chinese reading comprehension methods, this paper proposes a cognitive graph-based Chinese reading comprehension retrieval approach. This approach utilizes Wikipedia as a source to find evidence documents as reasoning paths for answering complex questions. Subsequently, existing reading comprehension models are employed to answer questions given the identified reasoning paths. The strong interaction between the retrieval of reasoning paths and the reading of answers within these paths enables robust pipeline processing. Figure 1 provides an overview of the proposed cognitive graph-based reading comprehension retrieval model (QARCG) in this paper.

![QARCG model framework.](image)

In the experimental section, we selected MemN2N [13], LSTM [14], DrQA [15], BIDA [16], R-net [17], Bert [18], SRQA [19], and Attentive and Impatient Reader [20] as the comparison methods. We conducted experiments using Baidu’s open-source Chinese reading comprehension question-answering dataset, WebQA [21]. The evaluation was performed against both single-fact-answer and complete-answer methods, and the model’s effectiveness was validated using strict matching and fuzzy matching approaches. The results analysis indicates that the proposed model outperforms existing methods, achieving better performance in terms of accuracy and efficiency.

1.3. Triple Extraction

Extracting relationship triples from unstructured natural language texts is a extensively studied topic in the field of information extraction [22], and it constitutes a fundamental basis for various artificial intelligence applications, including information retrieval, intelligent question answering, and dialog systems. The key components of knowledge graphs are factual relationships, with a significant portion represented by relationship triples. A triple is composed of two entities connected by a semantic relationship; these facts take the form of (subject, relationship, object) or (s, r, o), commonly referred to as relationship triples. Extracting relationship triples from natural language texts is a crucial step in constructing cognitive graphs for retrieval, which is a focal point of this study.

Early efforts in relationship triple extraction processed this task in a pipeline manner [22–25]. They extracted relationship triples through two distinct steps: firstly, performing named entity recognition on the text to identify all entities, and then classifying the identified entities for relationships. While this segregated framework simplifies task handling, it overlooks the interdependency between the two subtasks; errors
in entity recognition might impact relationship classification, often leading to issues like error propagation.

Presently, this paper effectively addresses the challenge of triple overlap. The method employed for triple extraction in this study is derived from this paper and is implemented using the bert4keras framework.

1.4. Main Contribution

This paper’s key contribution is the introduction of the QARCG model, a cognitive graph-based approach for enhancing reading comprehension. The model seamlessly merges retrieval and reasoning systems, facilitating efficient inference and interpretability in open-domain question answering. The QARCG model partitions the open-domain reading comprehension task into two systems: retrieval and reasoning, utilizing a cognitive graph representation to ensure effective information interaction.

In the retrieval system, the model employs triplet extraction to establish reasoning pathways, enhancing coherence between paragraphs and enabling the flow of information. The reasoning system employs an RNN to capture dynamic interactions among paragraphs, enabling reordering and scoring of reasoning pathways to generate answers. The model’s efficacy is substantiated through WebQA dataset evaluations, demonstrating superior performance in entity-level and comprehensive answers. Moreover, the paper conducts experiments and analyses, shedding light on component functions and offering insights for model optimization. Future prospects include exploring cognitive science theories, integrating memory mechanisms, and incorporating external feedback, paving innovative directions in cognitive graph representation research.

2. Model Definition Core Concepts

Definition 1 (Cognitive Graph). In the QARCG model, the cognitive graph is defined as $G = [P_1, P_2, \ldots, P_k]$, where each node in $G$ corresponds to a paragraph $P_i$. The retrieval system 1 reads each paragraph $P_i$ and extracts triples from the paragraph as the next hop paragraphs. These new nodes are then used to expand $G$, providing an explicit structure for the reasoning module system 2.

Definition 2 (Bert-wwm Embedding [8]). As System 2 requires transforming the paragraph $P_i$ into vector representations when learning reasoning paths, wwm stands for whole-word masking, which is an improvement over Bert’s masking technique. It replaces a complete word with a mask label instead of subword tokens. In contrast to English, where the smallest token is a word, in Chinese, the smallest token is a character, and words are composed of one or more characters with no obvious delimiters. Words contain more information, and whole-word masking involves masking the entire word. In the model, the input for Bert-wwm is as follows: [CLS] Question[SEP] clues[p, G][SEP] Para[x].

Here, clues $[p, G]$ denote the passage $p$ propagated from the preceding node in the cognitive graph $G$. During the first hop, Bert-wwm encodes the concatenation of the question and the paragraph. The output vector representation of Bert is denoted as $T \in \mathbb{R}^{L \times H}$, where $L$ is the length of the input sequence, and $H$ is the dimensionality of the hidden representation.

3. Model Framework

The model consists of two systems: the retrieval reasoning system and the reading comprehension system. System 1 is responsible for constructing the inference path of the paragraph graph, while System 2 involves scoring the reasoning paths and extracting answers from the highest-scoring paragraphs. The two systems are jointly trained and integrated to provide the final objective function.
3.1. Retrieval Reasoning Paths

System 1 of the cognitive graph method requires constructing and retrieving reasoning paths. For some complex questions, the evidence paragraphs may not directly have lexical relevance to the question, and independent retrieval of a given document list may not be sufficient for inferring the answer. However, it is highly likely to find the answer through text related to the answer (as shown in Figure 2). To perform such multi-hop reasoning, it is necessary to build a paragraph graph covering Wikipedia paragraphs relevant to the question. The Wikipedia graph is defined as $G$, where each node $P_i$ represents an individual paragraph.

$q$: When was the basketball team that Yao Ming played center for in the United States formed?

Paragraph 1: Yao Ming in the 2002 NBA Rookie Draft. Selected first overall by the Houston Rockets who believed Yao could fill the void left by their star center, Akeem Olajuwon, a year earlier.

Paragraph 2: Houston Rockets is an NBA basketball team located in Houston, Texas, United States, belongs to the Southwest Division of the West, the home court for the Toyota Center Arena team was established in 1967 in San Diego, formerly known as the San Diego Rockets.

Figure 2. Examples of problematic cases.

3.1.1. Paragraph Graph Construction

In Wikipedia, there is a wealth of entity entry information, which can be regarded as a knowledge resource for constructing multiple linguistic corpora. QARCG uses Wikipedia entries to construct direct edges in $G$, enabling the navigation from one paragraph to another. Based on the given question, the model initially retrieves the top $F$ highest-scoring paragraphs using the TF-IDF method as the initial nodes. Then, starting from these $F$ paragraphs, it utilizes the extracted triples to retrieve and direct to other paragraphs in Wikipedia, continuously expanding the tree structure. The iteration stops when the tree depth reaches the upper limit or no more triples can be found in the paragraphs. Once the paragraph graph construction is complete, the next step is to model the reasoning paths based on this paragraph graph.

3.1.2. Reasoning Path Modeling

In the constructed paragraph graph, its sequential nature plays a crucial role in text localization, while RNN (recurrent neural network) is proficient in learning hidden state connections in relationships, enabling the learning of reasoning paths. Therefore, utilizing RNN for reasoning path modeling can achieve a more effective representation [26].

In the reasoning path, [EOE] is used as the end-of-path control symbol for the next-hop paragraph $p_i$ related to the question. Research has shown that when there is explicit interaction between paragraphs and questions, it is possible to compress the necessary information of paragraphs that are irrelevant to the question but can lead to the answer into vectors, resulting in improved performance on entity-centered questions. Therefore, the model adopts the concatenation of paragraph $p_i$ and question $q$, followed by encoding using Bert (bert-wwm). The [CLS] token is used to independently encode the paragraph. BERT wwm (whole-word mask) is a variant of BERT that masks the entire word instead of subword-level tokens.

$$w_i = BERT_{[CLS]}(q, p_i) \in \mathbb{R}^d$$ (1)

At the t-th time step, the model selects a paragraph $p_i$ from the candidate set $C$ based on the current hidden state $h_t$ of the RNN. The initial hidden state $h_1$ is independent of any
question or paragraph and is based on a parameterized vector. The probability score of selecting $p_i$ as the reasoning path is calculated using $P(p_i|h_1)$:

$$P(p_i|h_1) = \sigma(w_i h_1 + b)$$

(2)

where $b$ is a bias term, and its range is represented as $b \in \mathbb{R}^d$.

The RNN selection process captures the relationships between paragraphs in the reasoning path by conditioning on the history of selections. For each time step, the top $K$ paragraphs with the highest probabilities are selected as candidate paragraphs. The inference path retrieval process stops when the end-of-path symbol [EOE] is selected, capturing reasoning paths of arbitrary lengths for each given question. The resulting form of the reasoning paths is shown in Equation (4).

$$
\alpha_{t+1} = W_r [h_t; w_i] + b_r \hspace{1cm} (3)
$$

$$
\alpha_{t+1} = \frac{\alpha}{|\alpha_{t+1}|} \alpha_{t+1} \hspace{1cm} (4)
$$

$$
Path = [p_1, p_2, \ldots, [EOE]] \hspace{1cm} (5)
$$

Finally, the score of each reasoning path is defined as the product of the probability scores of all paragraphs along that path. The top $B$ reasoning paths with the highest scores are then selected and passed to the reasoning module of System 2 for answer prediction.

3.1.3. Model Optimization

In the main text, to improve the retrieval of reasoning paths, the model is optimized as follows:

(1) The computation cost of determining whether a paragraph can be included in the reasoning path in Wikipedia is high, prior to outputting the reasoning path results to System 2. The score of a reasoning path is calculated as the product of the probability scores of all paragraphs along that path [27]. Beam search [28] is then utilized for pruning to select the top $B$ reasoning paths.

In the specific approach, to construct an effective cognitive graph, the TF-IDF retrieval method is initially used to initialize the candidate paragraphs and guide their search on Wikipedia. The top $F$ paragraphs with the highest TF-IDF scores relevant to the question are selected as the initial candidate set $C_1$. The candidate paragraph set $C_t (t \geq 2)$ is then expanded using the end-of-path marker [EOE]. The time complexity of processing the candidate set is given by $O(|C_1| + B \sum_{t \geq 2} |C_t|)$, where $B$ is the beam size and $|C_t|$ is the average size of the candidate set $C_t (t \geq 2)$.

Beam search is used to retrieve the reasoning paths in the cognitive graph. For a reasoning path $E$, the probabilities of selecting paragraphs are multiplied together:

$$E = [p_1, \ldots, p_k] \hspace{1cm} (6)$$

$$P(p_1|h_1) \ldots P(p_k|h_{|E|}) \hspace{1cm} (7)$$

By using the beam search method, the top $B$ reasoning paths with the highest scores are selected from the set of reasoning paths $E$ and passed to System 2, the reader model.

$$E = [E_1, \ldots, E_B] \hspace{1cm} (8)$$

$$S(q, E, a) = S_2(q, E, a), E \in E \hspace{1cm} (9)$$

(2) To effectively control the depth of retrieval when searching existing paragraphs in Wikipedia, the approach adopts the extraction of triples from paragraphs. The subject and object of the extracted triples are then used as keywords for retrieval. Subsequently, the relations matching the subject and object entities in the retrieved Wikipedia entries are considered as the next hop in the retrieval process. Experimental results demonstrate that this triple extraction approach effectively reduces the retrieval depth.
(3) The training is performed using negative sampling. During the training of the retrieval reasoning path model, it is necessary to distinguish between relevant and irrelevant paragraphs. Therefore, the model uses the “no_answer” paragraphs from the WebQA dataset [21], which cannot derive the answer, as negative examples for training. Since the number of paragraphs provided for each question is limited, there is no specific threshold set for the quantity of negative examples. The training loss function is represented by formula (17).

The specific algorithmic procedure is as follows Algorithm 1:

Algorithm 1: Cognitive Graph-Based Question Answering with Reasoning

1. Input: System 1 model, System 2 model, question Q, predicted value F, Wikipedia database W
2. Use Q and the given paragraphs to select the top K paragraphs P using the TF-IDF algorithm to initialize the cognitive graph G
3. Repeat
   4. Pop the outermost paragraph p from graph G as the node x
   5. Extract the preceding paragraphs of node x as reasoning path clues clues[x, G]
   6. If p is not None: continue retrieving paragraphs from Wikipedia W
   7. If x is a reasoning path node then
      8. Use triple extraction of to find a new paragraph p as y in Wikipedia
      9. For y as a reasoning path node do
         10. If y ∉ G and y belongs to Wikipedia W then
            11. Create a new node for y in the cognitive graph;
            12. If y ∈ G then
               13. Pass
               14. End
      15. Until there are no boundary nodes in G, or the threshold is reached
16. Return Path

The algorithm’s space complexity is $O(d|V| + d|W|N)$, where $V$ represents the number of supporting evidence documents in the dataset, $W$ represents the number of paragraphs retrieved from each Wiki search, and $N$ represents the retrieval depth. As the sentences in the model are encoded by BERT into d-dimensional vectors, the required storage space is $d|V| + d|W| \times N$. The algorithm’s time complexity is $O(tV(K + WN))$, where $t$ denotes the number of model training iterations and $K$ is the number of negative sampling iterations.

3.2. Reading and Answering Based on Reasoning Paths

System 2 of the cognitive graph needs to read and answer the already constructed reasoning paths. The model first scores the reasoning paths from E, selecting the highest-scoring path. Then, from the most reasonable path, the model extracts the answer span from the paragraphs of the path.

(1) Scoring the Reasoning Paths

Answer extraction heavily relies on the paragraphs within the reasoning paths. Therefore, the model’s initial task is to re-rank the reasoning paths in the path set based on their relevance to the question. Both the paragraphs and the question in the reasoning paths are encoded using the BERT method, as shown below:

$$u_E = \text{BERT}_{CLS}(q, E) \in R^D$$  \hspace{1cm} (10)

$$P(E|q) = \delta(w_n, u_E)$$  \hspace{1cm} (11)

Consequently, in the inference set, the highest-scoring inference path $E_{best}$ is selected as the final basis for deriving the answer, where $E_{best} \in E$ and $W_n \in R^D$ is a weight vector:
$E_{\text{best}} = \arg\max_{E \in E} P(E|q)$

(2) Extracting Answers from the Highest-Scoring Paragraph

After obtaining the highest-scoring reasoning path from the candidate set of reasoning paths, the next step is to extract the answer span from this path to predict the answer. Similarly, the reasoning path is encoded using BERT. For the BERT output, a linear layer and softmax function are applied to compute the probabilities of each position in the paragraph as the start of the answer. The top $K$ positions with the highest probabilities are selected. Subsequently, the model continues to find the end position of the answer by considering each position in the paragraph and computing the probabilities of being the end.

$$p_{\text{start}}^i = \frac{e^{W_{\text{ans}}^i}}{\sum_m e^{W_{\text{ans}}^m}}$$

$$p_{\text{start}}^j = \frac{e^{W_{\text{ans}}^j}}{\sum_m e^{W_{\text{ans}}^m}}$$

$$\text{end}_k = \arg\max_{\text{start} \leq j \leq \text{maxL}} P_{\text{end}}^j$$

where $W_{\text{ans}}$ is the parameter during training, $p_{\text{start}}^i$ represents the probability of the $i$-th position in the paragraph as the start of the answer span, $end_p$ represents the probability of the $j$-th position in the paragraph as the end of the answer span, and $\text{maxL}$ is the maximum initial answer span length set at the beginning. In this way, $K$ combinations of start and end predictions are found. $E_{\text{best}}$ contains the $i$-th and $j$-th tokens representing the probabilities of the start and end positions of the answer span, respectively. The final answer is selected by taking the maximum of the product of probabilities among the $K$ combinations. The calculation is as follows, where $S_2$ represents the reasoning system 2:

$$S_2 = \arg\max_{i,j,i \leq j} P_{\text{start}}^i P_{\text{end}}^j$$

(3) Model Optimization

To better distinguish paragraphs that do not contain answers, a negative sampling strategy is introduced during training. If a paragraph can directly lead to an answer, it is labeled as $P_r$; otherwise, it is labeled as $1 - P_r$. The definition of the loss function is as shown in Formula (18).

3.3 Joint Training

QARCG uses Wikipedia for open-domain paragraph retrieval, where each article is divided into multiple paragraphs, resulting in millions of paragraphs. Each paragraph $p$ is treated as the retrieval target for the retriever. Given a question $q$, the QARCG framework aims to derive the answer through retrieval and reading of reasoning paths, as shown in Equation (5). Each reasoning path is represented by a sequence of paragraphs. QARCG formulates the task by decomposing the objective into the retriever objective $S_1 = (q, E)$, which selects reasoning paths $E$ relevant to the question, and the reader objective $S_2 = (q, E, a)$, which finds the answer $a$ within $E$:

$$\arg\max S(q, E, a)s.t.S(q, E, a) = S_1(q, E) + S_2(q, E, a)$$

$$L_1(p|t) = -\log P(p|t) - \sum_{\tilde{p} \in \tilde{C}_t} \log(1 - \tilde{p}|\tilde{C})$$

$$L_2 = L_{\text{answer}} + L_{\text{no}} = ( -\log p_{\text{start}}^\ast - \log p_{\text{end}}^\ast ) - \log P_r$$
After encoding paragraphs and questions, the retriever model of QARCG captures paragraph interactions through the [CLS] representation of BERT, learning interactions among paragraphs to enhance the credibility of predicting reasoning paths. Additionally, the reordering process reduces uncertainty during reasoning path selection, making the model framework more robust.

4. Experimental Analysis

This section validates and evaluates the effectiveness of the proposed model using a Chinese reading comprehension question-answering dataset, which consists of questions with one or multiple documents as context. To ensure the authenticity of the experiments, the Baidu WebQA dataset, an open-source dataset, is used for experimentation and testing. The experimental results of QARCG and the compared methods are presented in a unified environment. For single-fact answer questions, the strict matching F1 score on the validation set increases from 74.28% to 75.99%, and on the test set, it increases from 73.53% to 74.98%. Furthermore, for the complete dataset, both the strict matching and fuzzy matching scores for annotated and retrieved types also show improvement, confirming the effectiveness of the proposed model.

4.1. Experimental Configuration

In this section, the model experiments were conducted in a unified experimental environment with a GeForce RTX 2080Ti GPU and Ubuntu 16.04.6 operating system. The hyperparameters were set following the principle of controlling variables uniformly, and benchmark parameters were defined to compare and analyze the model’s accuracy and time efficiency. In the experimental analysis section, we will further elaborate on the model’s performance and the impact of its hyperparameters.

4.2. Dataset

To ensure the authenticity and effectiveness of the experiments, we selected the Baidu WebQA dataset [10] for experimentation. This dataset comprises questions, annotated evidence, retrieved evidence, and answers. Unlike SQuAD [29], the questions in WebQA are derived from user queries in search engines, while the provided passages are extracted from web pages. WebQA provides document passages for each question retrieved from the evidence. Therefore, we used this dataset to evaluate the model’s answer-locating ability in reading comprehension tasks.

The statistical information of the WebQA dataset is presented in Table 1. The experiments were conducted separately on annotated and retrieved evidence for model training and evaluation. In the “Annotated” setting, each question is provided with one evidence passage, while in the “Retrieved” setting, multiple evidence passages are provided for each question. The data distribution of the validation set (validation) is closer to the training set (train). Generally, the validation set is used to assess the model’s accuracy, while the test set (test) is used to evaluate its transferability. In this section, we primarily focus on evaluating the model’s accuracy.

Table 1. Statistical information of the WebQA dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Question</th>
<th>Annotated Evidence</th>
<th>Retrieved Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Word</td>
<td># Word</td>
<td># Word</td>
</tr>
<tr>
<td>Train</td>
<td>36,145</td>
<td>374,500</td>
<td>140,897</td>
</tr>
<tr>
<td>Validation</td>
<td>3018</td>
<td>36,666</td>
<td>5412</td>
</tr>
<tr>
<td>Test</td>
<td>3024</td>
<td>36,815</td>
<td>5445</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>#</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>10,757,652</td>
<td>171,838</td>
<td>7,233,543</td>
</tr>
<tr>
<td>Validation</td>
<td>233,911</td>
<td>60,351</td>
<td>3,633,540</td>
</tr>
<tr>
<td>Test</td>
<td>234,258</td>
<td>60,645</td>
<td>3,620,391</td>
</tr>
</tbody>
</table>

4.3. Comparative Algorithms

In this section, we compare our model with other algorithms from two dimensions:
(1) For questions with fact-based entity-type answers, we compare our model with the following end-to-end approaches: MemN2N, LSTM with attention mechanism, and end-to-end sequence-based baseline model.

MemN2N [13] is one of the implementations of end-to-end models with memory networks [30]. It uses the bag-of-words method to encode the question and evidence, and stores the representation of evidence in an external memory. The recurrent attention model is used to retrieve relevant information from memory to answer questions.

Attentive and Impatient Readers [20] use bidirectional LSTM [14] to encode the question and evidence and employ a model that classifies the vocabulary based on these two encodings. It is a classic application of a simple fine-grained attention mechanism in machine reading comprehension tasks. The simpler “attention reader” computes the attention to evidence documents in an attention-based manner, while more complex readers calculate attention after processing each query word.

The end-to-end sequence-based baseline model [21], proposed by Baidu, uses sequence labeling to generate answers by classifying a huge number of vocabularies. However, this approach incurs high computational cost, and it is difficult for it to handle unseen words. This model utilizes an end-to-end trainable sequence labeling technique to process question answering, ultimately predicting answers, and is evaluated on the WebQA test dataset.

(2) For all types of answers in the WebQA dataset, we conduct experiments and compare our model with the following methods: LSTM [21], DrQA [15], BIDAF [16], R-net [17], BERT [18], and SRQA [19].

LSTM is a baseline model that uses sequence labeling to label answers. The simple usage of LSTM leads to limited text representation capacity, and its accuracy in sequence labeling is low, resulting in a fuzzy matching answer score below 70% for datasets with single texts.

The rest of the models are based on attention mechanisms and pointer networks. DrQA, BIDAF, and R-Net propose innovative attention methods. DrQA simply uses a bilinear term to calculate attention weights to obtain word-level question-merged paragraph representations, capturing the similarity between paragraphs and questions, and computing the probability of each word being the start or end position of the answer.

BIDAF introduces a memory attention mechanism to generate bidirectional attention flow and obtains word-level representations through multi-stage hierarchy for different granularities of context. It still uses span range probability prediction for answer extraction.

R-Net extends self-matching attention to fully exploit information from the paragraph itself to distinguish different meanings of the same word, thereby enhancing the context information. It ultimately extracts answers by predicting the positions of the answers.

BERT adopts the Transformer as the attention mechanism model, and is trained on a large-scale corpus. It uses span probability prediction as the final answer extraction.

SRQA utilizes a multi-layer attention network to learn better representations. The multi-layer attention network learns the interaction between questions and documents in each layer, and each layer’s document representation corresponds to the needs of the question. It conducts experiments using three different approaches: multi-layer attention (MA), cross evidence (CE), and adversarial training (AT).

4.4. Evaluation Metrics

The evaluation of open-domain reading comprehension question answering is of paramount importance [31]. In the WebQA dataset, the majority of answers are entity names, such as names of people, locations, and time. Therefore, it is appropriate to directly measure the accuracy of predicted answers as the evaluation criterion, instead of using approximate evaluation methods such as Bleu [32] and Rouge [33]. In Section 4.6, the model performance is evaluated using three metrics: precision (P), recall (R), and F1-score, which are defined as follows in Formula (20):
where \( |C| \) denotes true positive, \( |A| \) denotes false positive, and \( |Q| \) denotes false negative. These metrics are used to compare the predicted answers with the given answers in the dataset and assess the model’s performance.

Because the WebQA dataset is collected from the internet, answers with the same meaning in the questions may appear in different forms, such as “Beijing” and “Beijing city”. To properly evaluate the correctness of these cases, two methods are used in the experiments to calculate the correct answers: strict matching and fuzzy matching. Strict matching refers to considering the model’s predicted answer as correct only if it is exactly the same as the given correct answer in the dataset, while fuzzy matching refers to considering the model’s predicted answer as correct if it contains the correct answer or is a synonym of it.

Moreover, the WebQA dataset includes questions with a single evidence document (annotated) and questions with multiple evidence documents (retrieved). To fully consider these scenarios, the model is evaluated separately on the Annotated and Retrieved subsets of the dataset.

In the algorithm performance analysis in Section 5, the exact match (EM) is used to calculate whether the predicted results match the standard answers exactly, and the F1 score is used to measure the word-level matching between the predicted results and the standard answers. EM is a common evaluation criterion in question-answering systems, used to assess the percentage of correctly matched answers in the predictions. The calculation is represented by Formula (21), where \( C \) represents the number of correct answers predicted, and \( A \) represents the total number of answers.

\[
EM = \frac{C}{A}
\]

4.5. Parameter Settings

In the construction of the cognitive graph, QARCG adopts the TF-IDF algorithm [34] to select initial nodes. The number of TF-IDF is set to 3, and the maximum depth for retrieving from Wikipedia is set to 5. The pretrained Bert-wwm model is used, and both the retrieval system and the reasoning system use the same wwm configuration (\( d = 1024 \)). For the other comparative methods, the best parameter settings provided in the original papers are used for comparison. To explore the impact of different dimensional parameters on model performance, various parameter settings are compared within the QARCG model.

4.6. Algorithm Comparison

Experiments are conducted on the entity-answer data in the WebQA dataset, and the results are shown in Table 2. Experiments are also conducted on all data in the WebQA dataset, and the results are presented in Table 3.

Entity-Answer Experiment: End-to-end models are more suitable for entity-centered questions. Therefore, in the WebQA dataset, a comparative experiment is conducted on questions with entity-type answers. The validation set, which includes both annotated evidence and retrieved evidence, is combined for validation. Likewise, the test set, which includes both annotated evidence and retrieved evidence, is combined for testing. The experiments are conducted using strict matching. The results in Table 2 show that QARCG consistently outperforms MemN2N, the Attentive Reader, the Impatient Reader, and the baseline model, demonstrating the effectiveness of the proposed model.
Table 2. Comparison results for entity answers.

<table>
<thead>
<tr>
<th>System</th>
<th>Validation (Strict)</th>
<th>Test (Strict)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>MemN2N</td>
<td>52.61</td>
<td>52.61</td>
</tr>
<tr>
<td>Impatient Reader</td>
<td>63.05</td>
<td>63.05</td>
</tr>
<tr>
<td>Base</td>
<td>64.46</td>
<td>87.62</td>
</tr>
<tr>
<td>QARCG</td>
<td>76.62</td>
<td>75.13</td>
</tr>
</tbody>
</table>

The font is bolded to highlight the score results of the algorithm in this paper.

Table 3. Comparison results for all answers.

<table>
<thead>
<tr>
<th>Model</th>
<th>Annotated Evidence</th>
<th>Retrieved Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Strict (Val.) P</td>
<td>R</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>LSTM+softmax</td>
<td>59.74</td>
<td>69.11</td>
</tr>
<tr>
<td>LSTM+softmax(k-1)</td>
<td>59.84</td>
<td>67.51</td>
</tr>
<tr>
<td>LSTM+CRF</td>
<td>64.42</td>
<td>75.84</td>
</tr>
<tr>
<td>DrQA</td>
<td>69.62</td>
<td>69.62</td>
</tr>
<tr>
<td>BIDAF</td>
<td>70.04</td>
<td>70.04</td>
</tr>
<tr>
<td>R-net</td>
<td>70.48</td>
<td>70.48</td>
</tr>
<tr>
<td>Bert</td>
<td>71.36</td>
<td>71.36</td>
</tr>
<tr>
<td>MA</td>
<td>71.03</td>
<td>71.03</td>
</tr>
<tr>
<td>SRQA</td>
<td>71.28</td>
<td>71.28</td>
</tr>
<tr>
<td>MA+RN</td>
<td>72.51</td>
<td>72.51</td>
</tr>
<tr>
<td>MA+AT</td>
<td>72.51</td>
<td>72.51</td>
</tr>
<tr>
<td>QARCG</td>
<td>79.73</td>
<td>72.66</td>
</tr>
</tbody>
</table>

The font is bolded to highlight the score results of the algorithm in this paper.

WebQA Dataset Experiment: Experiments are conducted on both annotated evidence and retrieved evidence subsets of the WebQA dataset. For the annotated evidence validation set, strict matching is used for the experiments, while for the test sets of both annotated and retrieved evidence, experiments are conducted using both strict matching and fuzzy matching. The results in Table 3 show that QARCG outperforms different models, such as LSTM, DrQA, BIDAF, R-net, Bert, and SRQA, achieving the best F1 values under various conditions, further demonstrating the effectiveness of the proposed model.

In summary, there exist three key distinctions between Tables 2 and 3. Firstly, Table 2 utilizes a subset of the WebQA dataset, specifically focusing on instances where answers are expressed as entities. In contrast, Table 3 encompasses the entirety of the WebQA dataset, comprising a broader scope. Therefore, the dataset employed in Table 2 can be regarded as a subset of the dataset utilized in Table 3.

Secondly, the dataset utilized for Table 2 is characterized by answers in the form of entities, denoted as “retrieved”. In contrast, Table 3 introduces complex representations, encompassing both “retrieved” and “annotated”. In this context, “retrieved” signifies cases wherein definitive answers are present, while “annotated” encapsulates scenarios where multiple answer candidates are identified.

Thirdly, our experimental methodology encompasses two distinct approaches: “Strict” and “Fuzzy”. The “Strict” approach signifies instances where the generated answers exhibit precise alignment with the dataset answers, thus reflecting a complete match. On the other hand, the “Fuzzy” approach pertains to situations where the generated answers demonstrate partial correspondence with the dataset answers, indicating a nuanced level of alignment. The distinctions presented highlight the intricate dataset compositions, as well as the evaluation methodologies employed, contributing to a deeper understanding of the experimental framework.
5. Algorithm Performance Analysis

5.1. Ablation Experiments

Ablation experiments are particularly important for studying the role of each module in the model [35]. As shown in Table 4, removing certain components from the model significantly decreases its performance. We analyze the components used for retrieval in System 1 and those used for reading comprehension in System 2. In the experiments, EM values indicate whether the predicted answers match the correct answers exactly, and F1 values represent the overlap between predicted answers and correct answers.

Table 4. Rigorous experimentation: evaluating different variants of the model on the WebQA dataset.

<table>
<thead>
<tr>
<th>Setting</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieval System I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No RNN</td>
<td>56.7</td>
<td>52.9</td>
</tr>
<tr>
<td>CNN</td>
<td>72.6</td>
<td>69.3</td>
</tr>
<tr>
<td>GNN</td>
<td>74.5</td>
<td>72.6</td>
</tr>
<tr>
<td>No Ternary</td>
<td>70.8</td>
<td>68.6</td>
</tr>
<tr>
<td>No beam search</td>
<td>74.2</td>
<td>70.7</td>
</tr>
<tr>
<td>No negative sampling</td>
<td>69.3</td>
<td>66.2</td>
</tr>
<tr>
<td>No Rearrangement Path</td>
<td>75.8</td>
<td>72.2</td>
</tr>
<tr>
<td>No negative sampling</td>
<td>62.3</td>
<td>60.6</td>
</tr>
<tr>
<td>Reading System I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Rearrangement Path</td>
<td>75.8</td>
<td>72.2</td>
</tr>
<tr>
<td>No negative sampling</td>
<td>62.3</td>
<td>60.6</td>
</tr>
</tbody>
</table>

In System 1, QARCG constructs reasoning paths using the RNN module. As shown in Figure 1 of the model diagram, the retrieval process in the construction of cognitive graph often depends on the information mentioned in the previous paragraph. Therefore, if the model does not condition the next paragraph on the previous one, it will fail to retrieve the connection between paragraphs. Additionally, following the approach of the CogQA model [7], experiments were conducted using CNN and GNN to replace the RNN module. The triple extraction module provides evidence paragraphs more efficiently for the reasoning paths. When pure entity recognition is used for paragraph retrieval, a drop in performance is observed, indicating the importance of the triple extraction module for reasoning paths. The beam search module is designed to facilitate the search for better reasoning paths. Replacing it with the greedy module for search leads to a performance drop of approximately 4 points, demonstrating that the graph structure is more conducive to finding the optimal reasoning path. The negative sampling training is also an essential module that cannot be neglected. Experimental results show that without negative sampling as part of the training, the QARCG model is easily misled by reasoning paths without the correct answer, resulting in the retrieval of supporting paragraphs with incorrect answers.

In System 2, the module for reordering reasoning paths demonstrates the importance of finding the best reasoning path among multiple ones to identify the answer span. The use of negative sampling for answer prediction [36] also indicates that negative sampling is beneficial to prevent the model from overly relying on selecting answers related to the question.

5.2. Parameter Experiments

5.2.1. Retrieval Length

The maximum allowed retrieval length in Wikipedia during the construction of the cognitive graph in System 1 can have an impact on the model’s performance. Due to limitations in machine performance in the experiments, we only evaluated the QARCG model by fixing the reasoning path length to 1, 2, 3, 4. As shown in Figure 3, although the retrieval performance with a reasoning path length of 2 is slightly lower than that with a length of 3 in the WebQA dataset, the reasoning time is significantly reduced. Considering the trade-off between retrieval efficiency and performance, we set the reasoning path length to 2 as it provides the best retrieval performance.
5.2.2. Pretrained Models

Considering that different pretrained models can also have an impact on the model's performance, several variants based on BERT have been proposed in recent years. Therefore, we conducted experiments to encode the QARCG model using different pretrained models for comparison, including Albert [37], Roberta [38], and GPT [39]. As shown in Figure 4, the influence of different pretrained models on the model's performance is negligible in the same batch training. Therefore, in the QARCG model, we continue to use Bert-wwm as the pretrained model.

5.2.3. Top-K Inference Paths

In the model, we set three top thresholds for inference paths. Firstly, we obtain the top-K inference paths from RNN, then the top B paths selected by beam search are passed to System Two for reading comprehension, and finally, the top inference path is determined in System Two based on its relevance to the question. The top thresholds are set as commonly
used empirical values, representing a gradually narrowing scope: \( K = 5, B = 3, \) and finally, 1.

5.3. Data Augmentation Experiment

To validate the QARCG model, we conducted additional experiments using the Sogou question-answering competition dataset. Since the Sogou competition dataset is not publicly available, we performed data augmentation experiments by mixing the training data from the WebQA dataset with that from SogouQA. We then tested the QARCG model on the WebQA’s Annotated and Retrieved datasets (as shown in Table 3). The experimental results, as depicted in Figure 5, demonstrate that the F1 score of the model improved after data augmentation, showcasing the universality and scalability of the model in open-domain question answering.

![Figure 5. Comparison of F1 scores with data augmentation.](image)

6. Conclusions and Future Work

6.1. Conclusions

In light of the current state of open-domain reading comprehension question-answering methods, this study proposed the QARCG method based on the dual-process theory of cognitive science. The QARCG approach views open-domain question answering as a combination of retrieval and reasoning systems. System 1, responsible for retrieval, extracts triples from given supporting text and iteratively retrieves information from Wikipedia, constructing a cognitive graph with reasoning paths. System 2, responsible for reasoning, learns the interaction information between paragraphs using RNN based on the built cognitive graph. It reorders and scores different reasoning paths, and predicts the answer’s span based on the highest-scoring reasoning path’s paragraphs. The integration of retrieval and reasoning reduces the loss in graph construction and maintains graph structure, thereby enhancing interpretability and overcoming the lack of reasoning interpretability in traditional end-to-end reading comprehension methods. Additionally, it addresses the requirement for existing large-scale knowledge graphs in knowledge graph question answering.

6.2. Future Work

The open-domain reading comprehension question-answering method based on cognitive graphs integrates two systems to attain precise answers while maintaining interpretability. Nevertheless, there exists considerable potential for advancing and refining this approach.
(1) Dual-process theory: Currently, the cognitive graph is based on the dual-process theory of cognitive science, dividing reading comprehension into System 1 for retrieval and System 2 for reasoning. However, there may be other relevant theories in cognitive science that could provide support. Exploring how to construct a novel learning architecture that combines symbolic reasoning and deep learning is an important future task.

(2) Memory mechanisms in cognitive graphs: The retrieval system in the cognitive graph question-answering method simulates memory models in reading comprehension. However, human memory mechanisms encompass both long-term and short-term memory, operating with distinct modes and mechanisms. Considering how to build a memory model that reflects long-term memory storage is a challenge that needs to be addressed.

(3) Integration of cognitive graphs and external feedback: While the cognitive graph question-answering method achieves answer extraction through the tight integration of retrieval and reasoning systems, incorporating human cognitive processes may benefit from reinforcement learning to learn feedback and interact with the external world. Thus, the integration of cognitive graphs and external feedback is a topic worth exploring.

In conclusion, the research and application of cognitive graphs offer great potential for future exploration. Further work will focus on how to perform text reasoning based on complex knowledge, as there is still considerable room for improvement in addressing reading comprehension question-answering challenges.

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

Funding: This work was supported by the National Natural Science Foundation of China [grant numbers 71932002], and by the Youth Beijing Scholars Program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We also sincerely appreciate the comments and suggestions given by the editor-in-chief, the review team and the editors to improve the quality of this paper.

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

References


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.