

Article

Symmetric Graph-Based Visual Question Answering Using Neuro-Symbolic Approach

Jiyoun Moon 

Department of Electronics Engineering, Chosun University, Gwangju 61452, Republic of Korea;
jymoon@chosun.ac.kr

Abstract: As the applications of robots expand across a wide variety of areas, high-level task planning considering human–robot interactions is emerging as a critical issue. Various elements that facilitate flexible responses to humans in an ever-changing environment, such as scene understanding, natural language processing, and task planning, are thus being researched extensively. In this study, a visual question answering (VQA) task was examined in detail from among an array of technologies. By further developing conventional neuro-symbolic approaches, environmental information is stored and utilized in a symmetric graph format, which enables more flexible and complex high-level task planning. We construct a symmetric graph composed of information such as color, size, and position for the objects constituting the environmental scene. VQA, using graphs, largely consists of a part expressing a scene as a graph, a part converting a question into SPARQL, and a part reasoning the answer. The proposed method was verified using a public dataset, CLEVR, with which it successfully performed VQA. We were able to directly confirm the process of inferring answers using SPARQL queries converted from the original queries and environmental symmetric graph information, which is distinct from existing methods that make it difficult to trace the path to finding answers.

Keywords: high-level task planning; visual question answering; neuro-symbolic; symmetric graph; graph database; SPARQL; CLEVR dataset



Citation: Moon, J. Symmetric Graph-Based Visual Question Answering Using Neuro-Symbolic Approach. *Symmetry* **2023**, *15*, 1713. <https://doi.org/10.3390/sym15091713>

Academic Editors: João Ruivo Paulo, Cristina P. Santos and Gabriel Pires

Received: 18 August 2023

Revised: 31 August 2023

Accepted: 4 September 2023

Published: 7 September 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robots are widely utilized across various services, including education, medical, and logistics, mostly owing to the advancement of technology [1]. Educational robots are commonly used to trigger interest in students and heighten their learning motivation, while chatbots contribute to an improved learning efficiency by interacting with students [2]. In the medical field, robots demonstrate high precision and stability, thus providing various services, including surgical assistance for surgeons, rehabilitation therapy, and data analysis [3]. In logistics centers, robots are applied in sorting and transporting products, thus optimizing the overall logistics process [4]. Accordingly, service robots interact with humans in diverse settings and are applied in a wide range of fields. Numerous researchers are conducting studies on simultaneous localization and mapping (SLAM), robot vision, task planning [5], and human–robot interaction (HRI) for the development of service robots. However, relatively fewer number of studies are being conducted on task planning compared with other fields. In particular, little attention is being paid to research HRI-based task planning. Robots must be capable of executing task planning to perform high-level missions in real life, rather than performing just simple repetitive work. Furthermore, HRI-based task planning is especially important for successfully achieving the goal of a mission in an eco-system in which humans, robots, and the environment interact with each other [6].

Nonetheless, research has been conducted on task planning using robots in extreme environments, such as deep sea or space, and in varied environments including markets and factories. Cashmore et al. [7] introduced the ROSPlan framework, which is a

system for planning tasks in water using Girona 500 AUV. Crosby et al. [8] proposed a robot with the ability of transporting boxes in a factory through six-step task planning. Galindo et al. [9] suggested a method for task planning using semantic maps consisting of hierarchical spatial information and semantic knowledge in a home environment. Similarly, the majority of task planning algorithms do not take the human factor into consideration. However, human intervention is necessary for task planning when performing high-level missions [10]. Regarding HRI-based task planning, Crosby et al. [10] developed a human-aware task planning system. In this particular system, human constraints were taken into account in planning to enable interactions between humans and the robot. Alami et al. [11] proposed a robot control architecture in which humans and robots share the burden of work to perform certain tasks. Especially, HRI-oriented task planning becomes a serious issue when unexpected external events occur in extreme situations in which direct interaction with humans is challenging.

To handle unexpected external events in extreme situations, robots must be able to obtain and understand information about their surrounding environments, as humans cannot always directly acquire such information. Furthermore, robots must be able to communicate with humans through scene understanding. Thus, visual question answering (VQA), i.e., the capability of question answering in the form of a natural language based on image information as input, is a significant problem in HRI-based task planning. VQA is largely studied from a connectionist approach [12], symbolic approach, and neuro-symbolic approach. Yang et al. [13] proposed stacked attention networks (SANs), in which natural language questions are answered based on images. Marino et al. [14] introduced Knowledge Reasoning with Implicit and Symbolic representations (KRISP), which is capable of executing VQA in open-domain. Mao et al. [15] designed a neuro-symbolic concept learner (NS-CL) capable of executing VQA without explicit supervision.

The connectionist approach has the advantage of carrying out VQA for various unexpected environments and questions, but the disadvantage is that it is difficult to explain the process of deducing answers. The symbolic approach can explain the process of inferring answers but entails the limitation of being unable to be used in dynamic environments. The neuro-symbolic approach combines the advantages of all methods in which answers can be deduced for various VQA, and the relevant process can also be explained [16]. However, existing neuro-symbolic-based VQA methods mostly use table databases, which causes difficulties with expanding object information flexibly and performing high-level inference. Therefore, scene information is expressed and saved as a symmetric graph in this study, which can be flexibly extended. Applying the advantages of a graph database enables the simple inference of relation information as well as high-level inference. The proposed framework is shown in Figure 1. In the scene understanding part, objects are first detected using an object detector, and then the objects' type, color, and position information is extracted through an object feature extractor. In the question understanding part, natural language statements are converted to SPARQL to be applicable to a graph database. In the answer reasoning part, answers are inferred from a graph database through a SPARQL query. The proposed method is verified using a public dataset, CLEVR dataset [17], which is most widely used in neuro-symbolic-based VQA. The results show that answers can be successfully found for various types of queries.

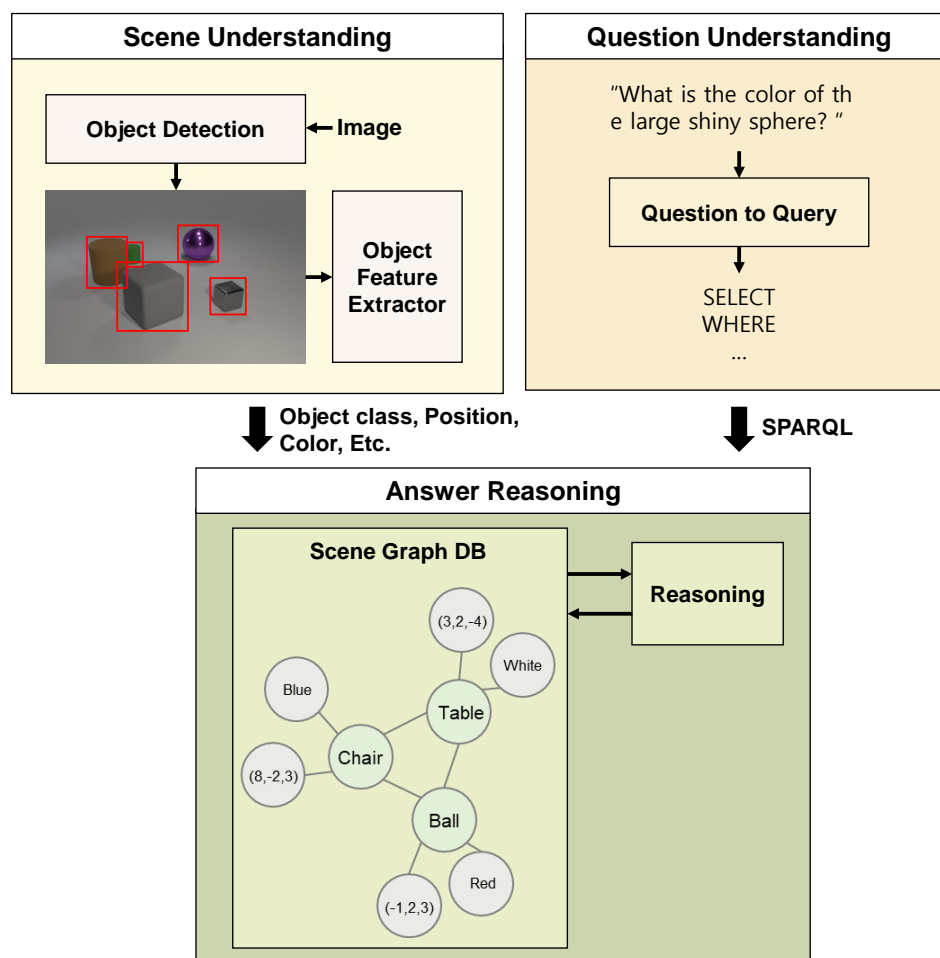


Figure 1. A symmetric graph-based visual question answering framework consisting of scene understanding, question understanding, and answer reasoning parts. In scene understanding, object information is extracted and saved in a graph database in a symmetric graph form. In question understanding, natural language queries are converted to SPARQL form. In answer reasoning, answers are inferred using a symmetric graph and SPARQL.

2. Related Work

VQA involves understanding the situation in given images and finding answers to questions. To solve VQA problems, many researchers use VQA datasets consisting of images and complicated natural language questions and answers related to the images. Johnson et al. [17] thus proposed a dataset called CLEVR. CLEVR has details on information, attributes, and position relations of objects constituted of each scene added to the dataset, and thus can be utilized in various VQA algorithms. In this chapter, different methods for solving VQA problems using various VQA datasets are introduced based on the connectionist approach, symbolic approach, and neuro-symbolic approach.

2.1. Connectionist Models for VQA

A large number of VQA algorithms [18–20] which train networks using massive datasets have been developed. Li et al. [21] enabled zero-shot transfer learning through a downstream task using architecture in which an image is learned directly from raw text. Yang et al. [22] solved VQA problems by proposing an auto-parsing network (APN) in which a probabilistic graphical model (PGM) is applied to the self-attention layer. Nam et al. [23] proposed dual attention networks (DANs) for elaborate interactions between vision and language. Noh et al. [24] introduced more efficient VQA methods through a method that initializes weighted values by using pre-trained convolutional neural net-

works (CNN) and gated recurrent units (GRU). The neural network-based connectionist approach has the advantage of executing VQA by robustly extracting features of images and natural language statements, but the disadvantage of being unable to execute complex inference and explain the inference process.

2.2. Symbolic Models for VQA

Diverse methods for solving VQA problems have been proposed whereby symbolic techniques have been used to infer answers from the available knowledge by using inductive and deductive reasoning. Lei et al. [25] proposed a new method capable of continual learning, in which all questions related to images can be answered. Han and Wang [26] introduced a symbolic approach-based graph matching-based reasoner (GMR) that can infer answers by automatically comparing graphs, and evaluated the algorithm's performance using the bAbI dataset. Malinowski and Fritz [27] applied discrete reasoning with uncertain prediction to automatically answering questions related to images. Symbolic models for VQA are capable of inferring answers but are vulnerable to external changes and difficult to respond to if proper actions are not taken for such changes.

2.3. Neuro-Symbolic Models for VQA

In recent years, several VQA algorithms utilizing the neuro-symbolic approach, which combines the strengths of the connectionist and symbolic approaches, have been actively developed. Amizadeh et al. [28] applied the neuro-symbolic of a logic framework first to infer queries on scenes that are incompletely recognized. Vedantam et al. [29] proposed a probabilistic neural-symbolic model that can respond to queries on the environment through a small learning model. Bosselut et al. [30] developed an algorithm that executes VQA and learns zero-shot, while dynamically generating commonsense knowledge graphs. VQA methods that employ the neuro-symbolic approach can explain the process of inferring answers and respond to queries related to various images and natural language processing. A new algorithm that utilizes the neuro-symbolic architecture is thus proposed in this study. Information on the objects constituting a scene can be extended flexibly by using scenes expressed in a symmetric graph form, and natural language statements can be converted directly into SPARQL for efficiently finding answers and reviewing the process.

3. Methods

In this study, a graph database with objects and object characteristics as nodes is generated from an image, and natural language queries are converted to SPARQL using bidirectional encoder representations from transformers (BERT) models with the transformer architecture. Object class, position, shape, and color can be used as scene information, and the respective symmetric graph can be extended by obtaining related data. Answers are then inferred through a graph database built using environment information and queries converted to SPARQL. For the inference process, Prolog, which uses the first-order logic, is utilized. The overall process of the proposed method is shown in Figure 2.

3.1. Scene Graph Generation

Symmetric graphs are generated using objects consisting of an image and any relation information between those objects. One symmetric graph is generated from one scene, and each graph contains various types of information including the color, size, shape, and texture of each object constituting the scene and the position relations between objects. Since the input of the graph node is text data, even if the property of an object such as color, shape, and other parameters is more than one, all data can be managed by extending the text. Therefore, the entire graph maintains a symmetric graph form. If the number of objects constituting one scene increases, one small symmetric graph composed of the object information is added to the entire graph. Each scene is saved and utilized using Protégé, which is an ontology editor developed at Standard University. An ontology is generated and checked by Protégé, and a plug-in capable of mapping, merging, and applying Pellet

inference to graph data is provided [31]. Previous studies saved and used scene information in table databases. Therefore, there is a limitation in terms of freely utilizing the relations of each object [32]. In contrast, a graph database allows for relatively easier access among objects, wherein the nodes of each graph are connected, thus allowing for fast searches for relations. Compared with a table database, a graph database demonstrates the extendibility and stability, which are both beneficial for inference [33,34]. In this study, an ontology was designed for saving the scene information. The design details are provided in the Appendix A.

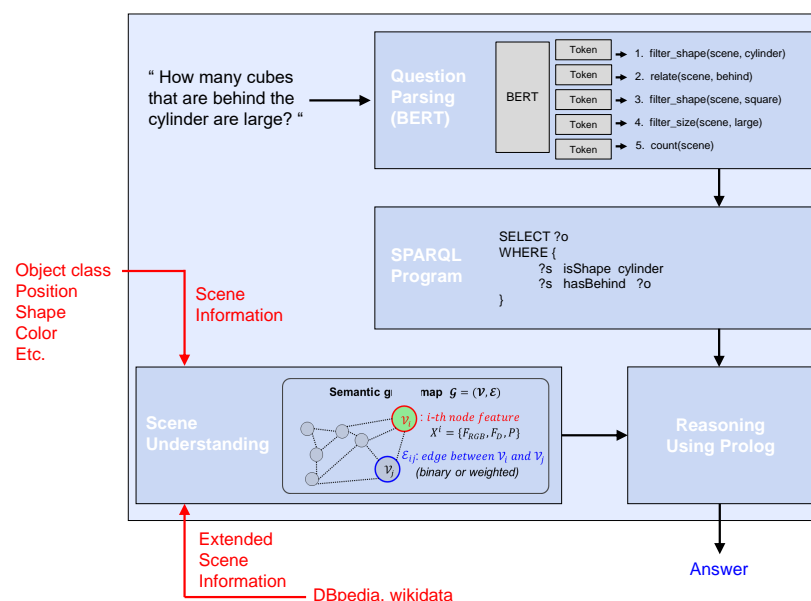


Figure 2. Graph-based visual question-answering process in which scene information is expressed as a semantic graph map to be saved or utilized. Natural language queries are converted to SPARQL using bidirectional encoder representations from transformers, and lastly, Prolog is used to infer answers.

3.2. Question Understanding

A transformer-based BERT model is used to convert natural language into SPARQL grammar. The transformer architecture comprises an encoder consisting of multi-head attention, feed forward, add/norm layers, and a decoder, in which a masked multi-head attention layer is added to the encoder. BERT is a model built by stacking several layers of encoders in the transformer architecture. The name and performance of the BERT model vary depending on the number of encoders, embedding vectors, and self-attention heads; in this study, in particular, the BERT-Small model is used [35]. To apply our method, it is essential to construct SPARQL datasets corresponding to natural language queries. The BERT-Small model is leveraged for effective learning even in situations with limited data resources. This part encompasses the transformation of natural language queries into SPARQL, a process in which the BERT-Small model's capability to comprehend context and meaning from smaller datasets proves valuable. As a result, it aids in generating precise and relevant SPARQL queries.

3.3. Answer Reasoning

Answers to the questions are found through natural language statements that have been converted to SPARQL. Natural language statements are converted to sequential SPARQL queries through a query analysis. A graph corresponding to the query is returned as each query is carried out, and ultimately, the results are output in the form of a string. Consequently, the process of finding answers to queries can be demonstrated. For questions requiring additional inference, Prolog [36], which is a logic programming language that supports knowledge expression and inference, is used.

4. Experiment Results

In this study, experiments are conducted using the CLEVR dataset. The CLEVR dataset is related to VQA and consists of images, detailed information on a scene, and queries/responses. Among the detailed information of a scene, information related to objects typically includes colors (gray, blue, brown, yellow, red, green, purple, and cyan), size (large, small), shape (cylinder, sphere, and cube), three-dimensional volume, texture (rubber and metal), three-dimensional center point, and angle of rotation. In addition, position relations (front, back, left, and right) between objects constituting one scene is provided. An example of the dataset is shown in Figure 3.

The graph generated for Figure 3 is shown in Figure 4. A knowledge symmetric graph of each object is expressed as follows. A node of a graph is composed of the index of the image. A symmetric graph's properties are expressed in relations through domain and range, and the object information is input. Position relation between objects is relative and can thus be set bidirectionally. The sequential SPARQL queries are then sequentially applied to the environment graph stored in the graph database. This process enables the retrieval of answers to complex queries related to environmental information. It allows for the verification and explanation of the process of finding answers to intricate queries concerning the environment.

A searchable database needs to be generated in order to execute a SPARQL query using the CLEVR dataset and the BERT model. The relevant process is as follows. First, the data being used for conversion are extracted by analyzing the existing dataset. Then, the extracted image information is converted to graph data, and the statement information is converted to SPARQL. The converted data need to be verified in terms of whether they function identically to the existing dataset. The converted dataset is used to implement a translation model between the natural language and SPARQL, and then tested to confirm whether it has been properly converted. A logical error test is performed to verify the results of the natural language-SPARQL conversion. The Pellet inference feature of Protégé, which is a knowledge graph visualization tool, is utilized as a test tool. The detailed process is provided in Appendix B.

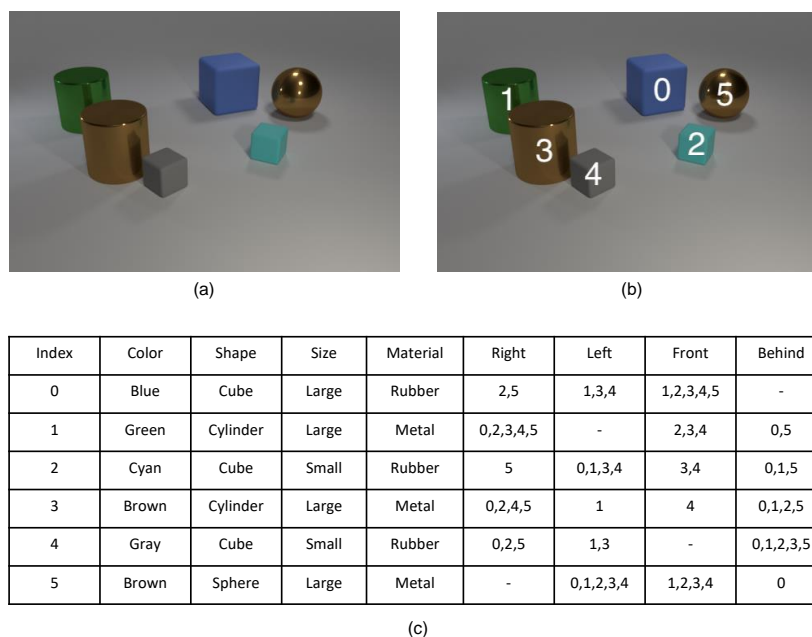


Figure 3. Dataset example; (a) original image, (b) index assigned to each object in the image, and (c) table of information on each object organized through scene understanding.

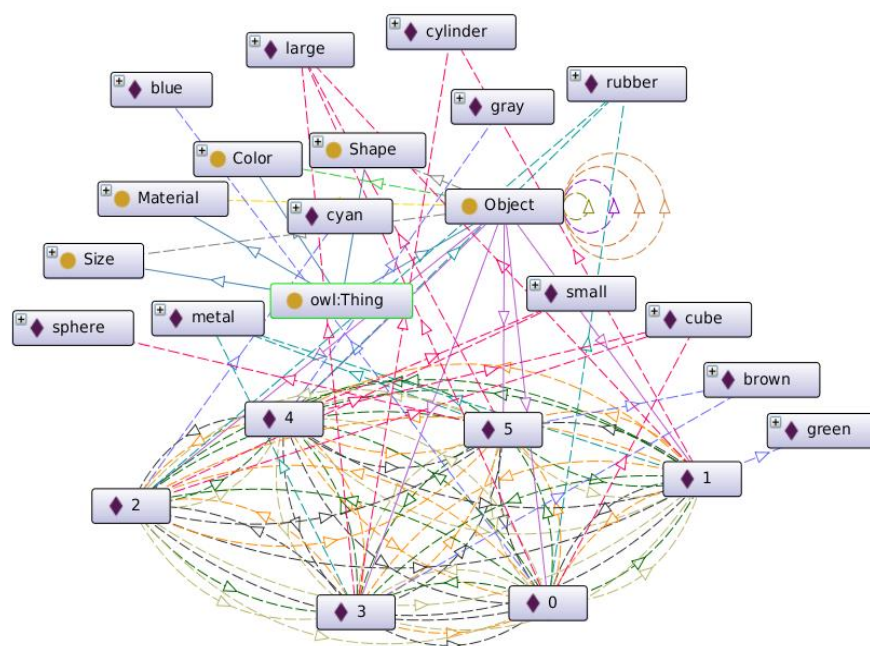


Figure 4. Graphical representation of an example image in Figure 3.

In this study, the BERT-Small model is used from among various BERT models to convert natural language queries into SPARQL, which is designed by applying the transformer architecture. Natural language query was set as the input and SPARQL query was set as the output of the BERT model for training the network. Figure 5 illustrates the result of converting the given queries into SPARQL queries. As shown in the experimental results, natural language queries have been successfully converted into sequential SPARQL query statements.

Input (Question)	Output (SPARQL)
How many other objects are there of the same material as the small cylinder?	<pre> SELECT * FROM <graph> WHERE { ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isSize :small . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isShape :cylinder . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o } SELECT ?s ?p ?o FROM <graph1> FROM <graph> WHERE { GRAPH <graph1> { ?x :isMaterial ?y } GRAPH <graph> { ?s ?z ?y . ?s ?p ?o } FILTER (?x != ?s) } SELECT (COUNT(?s) AS ?s) FROM <graph1> WHERE { ?s a :Object } </pre>
Are there any other things that are the same shape as the small red shiny object?	<pre> SELECT * FROM <graph> WHERE { ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isSize :small . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isColor :red . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isMaterial :metal . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o } SELECT ?s ?p ?o FROM <graph1> FROM <graph> WHERE { GRAPH <graph1> { ?x :isShape ?y } GRAPH <graph> { ?s ?z ?y . ?s ?p ?o } FILTER (?x != ?s) } SELECT ?s FROM <graph1> WHERE { (SELECT (COUNT(*) AS ?s) WHERE { ?s a :Object }) BIND(IF(?s > 0, "yes", "no") as ?s) } </pre>
How many gray objects have the same size as the blue rubber thing?	<pre> SELECT * FROM <graph> WHERE { ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isColor :blue . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s :isMaterial :rubber . ?s ?p ?o } SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o } SELECT ?s ?p ?o FROM <graph1> FROM <graph> WHERE { GRAPH <graph1> { ?x :isSize ?y } GRAPH <graph> { ?s ?z ?y . ?s ?p ?o } FILTER (?x != ?s) } SELECT * FROM <graph1> WHERE { ?s :isColor :gray . ?s ?p ?o } SELECT (COUNT(?s) AS ?s) FROM <graph1> WHERE { ?s a :Object } </pre>

Figure 5. Result of converting natural language queries into sequential SPARQL query statements using the BERT-Small model.

The final experimental results are shown in Figure 6. Figure 6a shows the original image and the index image, while Figure 6b shows an overview of the object information. Figure 6c illustrates the process of converting the query “Do the small shiny object and the matte cylinder have the same color?” into SPARQL and sequentially inferring the answer. Additional experimental results are provided in Appendix C. As shown in Figure 6, the

proposed method successfully inferred the answer. The method proposed in this study expressed scene information in the form of a graph, converted natural language queries into sequential SPARQL, and inferred answers for high-level queries, and its effectiveness was verified using the CLEVR dataset. Through experiments, the proposed algorithm found answers to 500 untrained test questions with 95.3% accuracy. The approach we propose stands apart from conventional methods in that it enables direct querying to retrieve answers from a graph database. This process allows for a step-by-step verification of the query, offering distinct advantages. Furthermore, leveraging the characteristics of the graph, it becomes feasible to easily expand the semantic information of environmental objects.



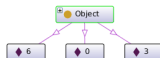



(a)

Index	Color	Shape	Size	Material	Right	Left	Front	Behind
0	Yellow	Sphere	Small	Rubber	1,2,5	3,4,6	-	1,2,3,4,5,6
1	Purple	Sphere	Large	Metal	2,5	0,3,4,6	0,2,3,4,5,6	-
2	Blue	Cylinder	Large	Metal	-	0,1,3,4,5,6	0,3,4,5,6	1
3	Gray	Cube	Large	Rubber	0,1,2,4,5,6	-	0	1,2,4,5,6
4	Brown	Cylinder	Small	Metal	0,1,2,5,6	3	0,3	1,2,5,6
5	Purple	Sphere	Large	Metal	2	0,1,3,4,6	0,3,4	1,2,6
6	Brown	Cylinder	Small	Rubber	0,1,2,5	3,4	0,3,4,5	1,2

(b)

Question				
Do the small shiny object and the matte cylinder have the same color?				
No	SPARQL	Graph	Type	Result
1	SELECT * FROM <graph> WHERE { ?s ?p ?o }		Graph	0,1,2,3,4,5
2	SELECT * FROM <graph1> WHERE { ?s :isSize :small . ?s ?p ?o }		Graph	0,4,6
3	SELECT * FROM <graph1> WHERE { ?s :isMaterial :metal . ?s ?p ?o }		Graph	4
4	SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o }		Graph	4
5	SELECT * FROM <graph> WHERE { ?s ?p ?o }		Graph	0,1,2,3,4,5

Figure 6. Cont.

6	SELECT * FROM <graph1> WHERE { ?s :isMaterial :rubber. ?s ?p ?o }		Graph	0,3,6
7	SELECT * FROM <graph1> WHERE { ?s :isShape :cylinder. ?s ?p ?o }		Graph	6
8	SELECT * FROM <graph1> WHERE { ?s a :Object. ?s ?p ?o }		Graph	6
9	SELECT ?z FROM <graph1> FROM <graph2> WHERE { GRAPH <graph1> { [] :isColor ?x } GRAPH <graph2> { [] :isColor ?y } BIND(IF(?x = ?y, "yes", "no") AS ?z) }		String	yes

(c)

Figure 6. Experimental result: (a) input image and object indexed image; (b) table presenting an overview of the object information; (c) process of converting queries into SPARQL and sequentially inferring the answer.

5. Discussion and Conclusions

The development of task planning methods capable of performing HRI for high-level task planning is inevitable. The method of VQA is thus introduced in this study for task planning, which is an important element in interactions between humans and robots. Contrary to previous methods, the proposed method efficiently expressed the relation of objects constituting a scene using a graph database, and the answer was inferred using natural language queries that have been converted to sequential SPARQL queries. The proposed method was experimentally verified and successfully executed by VQA. The proposed method can employ extended scene information, thus being utilizable with web data having an extensive range of data. The performance can be further enhanced by designing various forms of ontology structures. Through future research, we plan to expand the semantic information of each object that makes up the environment to be able to answer more complex and difficult questions. Furthermore, we aim to enable high-level task planning by allowing robots and humans to communicate with each other through inquiries.

Funding: This work was funded by the National Research Foundation of Korea grant number 2021R1G1A1007097.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the National Research Foundation of Korea grant number 2021R1G1A1007097.

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

Abbreviations

The following abbreviations are used in this manuscript:

SLAM	Simultaneous localization and mapping
HRI	Human–robot interaction
VQA	Visual question answering
NS-CL	Neuro-symbolic concept learner
APN	Auto-parsing network
DANs	Dual attention networks
GMR	Graph matching based reasoner
BERT	Bidirectional encoder representations from transformers

Appendix A

A class diagram expressing object information is shown in Figure A1. Object information inheriting four items, including the objects' color, size, shape, and texture is shown in Figure A1a; specifically, inheritance information concerning shape is shown in Figure A1b.

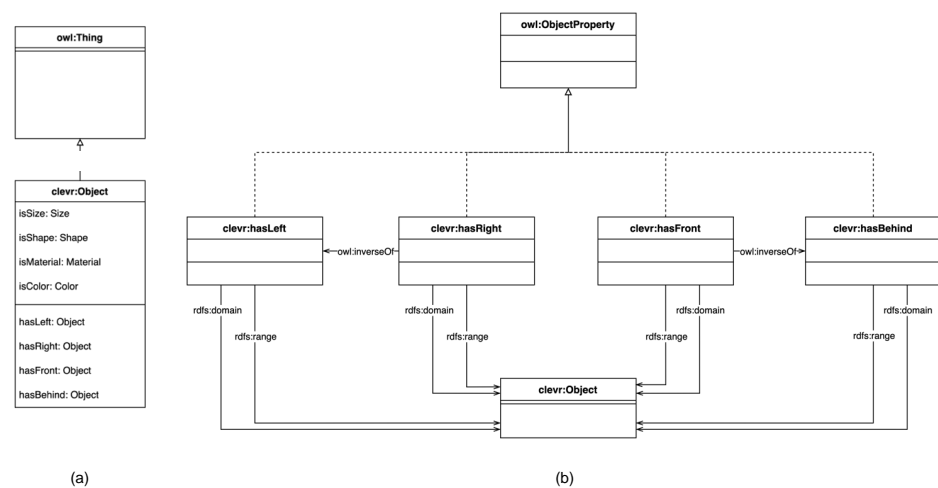


Figure A1. Class diagram; (a) class diagram inheriting shape information and (b) class diagram representing object property.

Appendix B

The process of generating a sequential SPARQL query to enable inference according to natural language statements is demonstrated in Figure A2. In this study, the program function used for finding the answer using the converted query statements is provided by the CLEVR dataset.

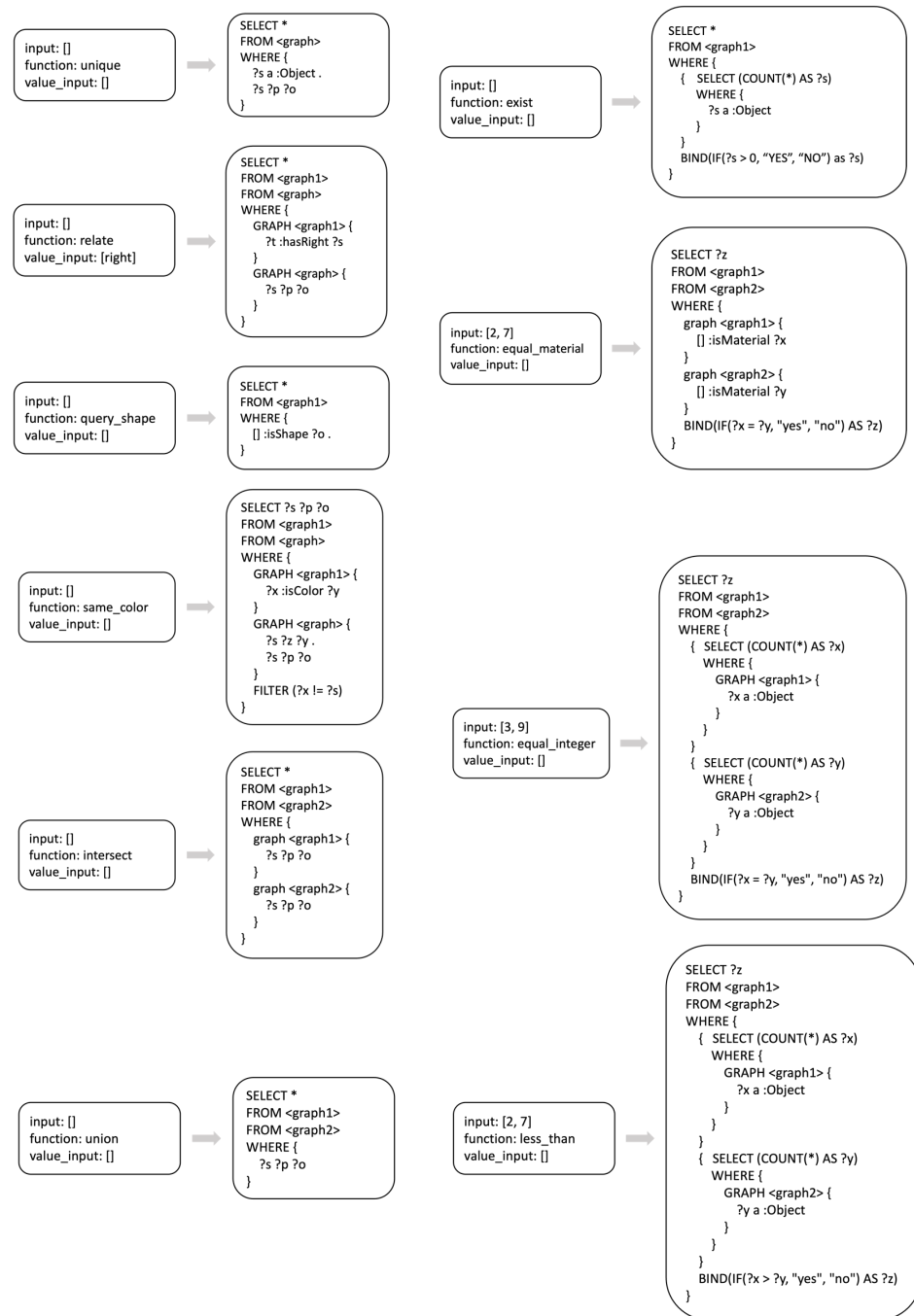
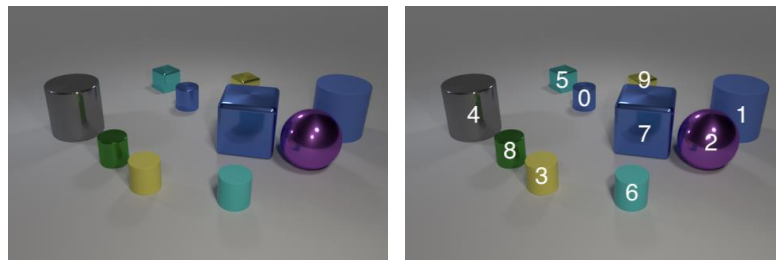


Figure A2. Generating a sequential SPARQL query using CLEVR dataset.

Appendix C

Figure A3 shows the process of finding the answer to different queries for the same scene. Figure A3a shows the image information, while Figure A3b shows detailed information of each object. Figure A3c shows the process of finding the answer to a query about color, Figure A3d shows the process of finding a yes or no answer, and Figure A3e shows the process of finding the answer to a query related to a count.







(a)

Index	Color	Shape	Size	Material	Right	Left	Front	Behind
0	Blue	Cylinder	Small	Metal	1,2,3,6,7	4,5,8,9	1,2,4,6,7,8,9	3,5
1	Blue	Cylinder	Large	Rubber	-	0,2,3,4,5,6,7,8,9	2,6,7,8,9	0,3,4,5
2	Purple	Sphere	Large	Metal	1	0,3,4,5,6,7,8,9	6,9	0,1,3,4,5,7,8
3	Yellow	Cube	Small	Metal	1,2	0,4,5,6,7,8,9	0,1,2,4,6,7,8,9	5
4	Gray	Cylinder	Large	Metal	0,1,2,3,5,6,7,8,9	-	1,2,6,7,8,9	0,3,5
5	Cyan	Cube	Small	Metal	0,1,2,3,6,7	4,8,9	0,1,2,3,4,6,7,8,9	-
6	Cyan	Cylinder	Small	Rubber	1,2,3,7	0,4,5,8,9	-	0,1,2,3,4,5,7,8,9
7	Blue	Cube	Large	Metal	1,2,3	0,4,5,6,8,9	2,6,8,9	0,1,3,4,5
8	Green	Cylinder	Small	Metal	0,1,2,3,5,6,7,9	4	2,6,9	0,1,3,4,5,7
9	Yellow	Cylinder	Small	Rubber	0,1,2,3,5,6,7	4,8	6	0,1,2,3,4,5,7,8

(b)

Question				
The other small shiny thing that is the same shape as the tiny yellow shiny object is what color?				
No	SPARQL	Graph	Type	Result
1	SELECT * FROM <graph> WHERE { ?s ?p ?o }		Graph	0,1,2,3,4,5,6,7,8,9
2	SELECT * FROM <graph1> WHERE { ?s :isSize :small . ?s ?p ?o }		Graph	0,3,5,6,8,9

Figure A3. Cont.

3	<pre>SELECT * FROM <graph1> WHERE { ?s :isColor :yellow . ?s ?p ?o }</pre>		Graph	3,9
4	<pre>SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o }</pre>		Graph	3
5	<pre>SELECT ?s ?p ?o FROM <graph1> FROM <graph> WHERE { GRAPH <graph1> { ?x :isShape ?y } GRAPH <graph> { ?s ?z ?y . ?s ?p ?o } FILTER (?x != ?s) }</pre>		Graph	5, 7
6	<pre>SELECT * FROM <graph1> WHERE { ?s :isSize :small . ?s ?p ?o }</pre>		Graph	5
7	<pre>SELECT * FROM <graph1> WHERE { ?s :isMaterial :metal . ?s ?p ?o }</pre>		Graph	5
8	<pre>SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o }</pre>		Graph	5
9	<pre>SELECT * FROM <graph1> WHERE { [] :isColor ?o }</pre>		String	cyan

(c)

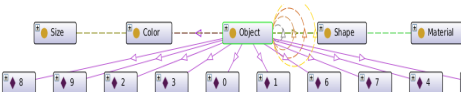
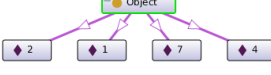
Question				
Is there anything else that has the same color as the large shiny cube?				
No	SPARQL	Graph	Type	Result
1	<pre>SELECT * FROM <graph> WHERE { ?s ?p ?o }</pre>		Graph	0,1,2,3,4,5,6,7,8,9
2	<pre>SELECT * FROM <graph1> WHERE { ?s :isSize :large . ?s ?p ?o }</pre>		Graph	1,2,4,7

Figure A3. Cont.

3	<pre>SELECT * FROM <grpah1> WHERE { ?s :isMaterial :metal . ?s ?p ?o }</pre>		Graph	2,4,7
4	<pre>SELECT * FROM <grpah1> WHERE { ?s :isShape :cube . ?s ?p ?o }</pre>		Graph	7
5	<pre>SELECT * FROM <graph1> WHERE { ?s a :Object . ?s ?p ?o }</pre>		Graph	7
6	<pre>SELECT ?s ?p ?o FROM <graph1> FROM <graph> WHERE { GRAPH <graph1> { ?x :isColor ?y } GRAPH <graph> { ?s ?z ?y . ?s ?p ?o } FILTER (?x != ?s) }</pre>		Graph	0,1
7	<pre>SELECT ?s FROM <graph1> WHERE { { SELECT (COUNT(*) AS ?s) WHERE { ?s a :Object } } BIND(IF(?s > 0, "yes", "no") AS ?s) }</pre>		String	yes

(d)

Question				
How many metallic objects are big blue cubes or blue objects?				
No	SPARQL	Graph	Type	Result
1	<pre>SELECT * FROM <graph> WHERE { ?s ?p ?o }</pre>		Graph	0,1,2,3,4,5,6,7,8,9
2	<pre>SELECT * FROM <graph1> WHERE { ?s :isSize :large . ?s ?p ?o }</pre>		Graph	1,2,4,7

Figure A3. Cont.

3	SELECT * FROM <grpah1> WHERE { ?s :isMaterial :metal . ?s ?p ?o }		Graph	1,7
4	SELECT * FROM <grpah1> WHERE { ?s :isShape :cube . ?s ?p ?o }		Graph	7
5	SELECT * FROM <graph> WHERE { ?s ?p ?o }		Graph	0,1,2,3, 4,5,6,7, 8,9
6	SELECT * FROM <grpah1> WHERE { ?s :isColor :blue . ?s ?p ?o }		Graph	0,1,7
7	SELECT * FROM <graph1> FROM <graph2> WHERE { ?s ?p ?o }		Graph	0,1,7
8	SELECT * FROM <grpah1> WHERE { ?s :isMaterial :metal . ?s ?p ?o }		Graph	0,7
9	SELECT (COUNT(?s) as ?s) FROM <graph1> WHERE { ?s a :Object }		int	2

(e)

Figure A3. Experimental results for various queries: (a) original image and indexed image; (b) table showing detailed information on objects; (c) answer inference process for finding object color; (d) inference process for query with a yes or no answer; (e) inference process for query about the number of objects.

References

- Gonzalez-Aguirre, J.A.; Osorio-Oliveros, R.; Rodríguez-Hernández, K.L.; Lizárraga-Iturralde, J.; Morales Menendez, R.; Ramírez-Mendoza, R.A.; Ramírez-Moreno, M.A.; Lozoya-Santos, J.D.J. Service robots: Trends and technology. *Appl. Sci.* **2021**, *11*, 10702. [\[CrossRef\]](#)
- Mubin, O.; Stevens, C.J.; Shahid, S.; Al Mahmud, A.; Dong, J.J. A review of the applicability of robots in education. *J. Technol. Educ. Learn.* **2013**, *1*, 13. [\[CrossRef\]](#)
- Holland, J.; Kingston, L.; McCarthy, C.; Armstrong, E.; O'Dwyer, P.; Merz, F.; McConnell, M. Service robots in the healthcare sector. *Robotics* **2021**, *10*, 47. [\[CrossRef\]](#)
- Echelmeyer, W.; Kirchheim, A.; Wellbrock, E. Robotics-logistics: Challenges for automation of logistic processes. In Proceedings of the 2008 IEEE International Conference on Automation and Logistics, Qingdao, China, 1–3 September 2008; pp. 2099–2103.
- Antonyshyn, L.; Silveira, J.; Givigi, S.; Marshall, J. Multiple mobile robot task and motion planning: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–35. [\[CrossRef\]](#)
- Tsarouchi, P.; Makris, S.; Chrysosolouris, G. Human–robot interaction review and challenges on task planning and programming. *Int. J. Comput. Integr. Manuf.* **2016**, *29*, 916–931. [\[CrossRef\]](#)
- Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; Ridder, B.; Carreras, A.; Palomeras, N.; Hurtos, N.; Carreras, M. Rosplan: Planning in the robot operating system. In Proceedings of the International Conference on Automated Planning and Scheduling, Jerusalem, Israel, 7–11 June 2015; Volume 25, pp. 333–341.

8. Crosby, M.; Petrick, R.; Rovida, F.; Krueger, V. Integrating mission and task planning in an industrial robotics framework. In Proceedings of the International Conference on Automated Planning and Scheduling, Pittsburgh, PA, USA, 18–23 June 2017; Volume 27, pp. 471–479.
9. Galindo, C.; Fernández-Madrigal, J.A.; González, J.; Saffiotti, A. Robot task planning using semantic maps. *Robot. Auton. Syst.* **2008**, *56*, 955–966. [\[CrossRef\]](#)
10. Cirillo, M.; Karlsson, L.; Saffiotti, A. Human-aware task planning: An application to mobile robots. *ACM Trans. Intell. Syst. Technol.* **2010**, *1*, 1–26. [\[CrossRef\]](#)
11. Alami, R.; Clodic, A.; Montreuil, V.; Sisbot, E.A.; Chatila, R. Task planning for human-robot interaction. In Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies, Grenoble, France, 12–14 October 2005; pp. 81–85.
12. Srivastava, Y.; Murali, V.; Dubey, S.R.; Mukherjee, S. Visual question answering using deep learning: A survey and performance analysis. In Proceedings of the Computer Vision and Image Processing: 5th International Conference (CVIP 2020), Prayagraj, India, 4–6 December 2020; pp. 75–86.
13. Yang, Z.; He, X.; Gao, J.; Deng, L.; Smola, A. Stacked attention networks for image question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 21–29.
14. Marino, K.; Chen, X.; Parikh, D.; Gupta, A.; Rohrbach, M. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14111–14121.
15. Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J.B.; Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv* **2019**, arXiv:1904.12584.
16. Yi, K.; Wu, J.; Gan, C.; Torralba, A.; Kohli, P.; Tenenbaum, J. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
17. Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2901–2910.
18. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
19. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv* **2016**, arXiv:1606.01847.
20. Wu, Q.; Wang, P.; Shen, C.; Dick, A.; Van Den Hengel, A. Ask me anything: Free-form visual question answering based on knowledge from external sources. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4622–4630.
21. Li, M.; Xu, R.; Wang, S.; Zhou, L.; Lin, X.; Zhu, C.; Zeng, M.; Ji, H.; Chang, S.F. Clip-event: Connecting text and images with event structures. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16420–16429.
22. Yang, X.; Gao, C.; Zhang, H.; Cai, J. Auto-parsing network for image captioning and visual question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2197–2207.
23. Nam, H.; Ha, J.W.; Kim, J. Dual attention networks for multimodal reasoning and matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 299–307.
24. Noh, H.; Seo, P.H.; Han, B. Image question answering using convolutional neural network with dynamic parameter prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 30–38.
25. Lei, S.W.; Gao, D.; Wu, J.Z.; Wang, Y.; Liu, W.; Zhang, M.; Shou, M.Z. Symbolic replay: Scene graph as prompt for continual learning on vqa task. *arXiv* **2022**, arXiv:2208.12037.
26. Han, J.; Wang, H. Graph matching based reasoner: A symbolic approach to question answering. *Eng. Appl. Artif. Intell.* **2021**, *105*, 104425. [\[CrossRef\]](#)
27. Malinowski, M.; Fritz, M. A multi-world approach to question answering about real-world scenes based on uncertain input. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1682–1690.
28. Amizadeh, S.; Palangi, H.; Polozov, A.; Huang, Y.; Koishida, K. Neuro-symbolic visual reasoning: Disentangling. In Proceedings of the International Conference on Machine Learning, Virtual Event, 3–18 July 2020; pp. 279–290.
29. Vedantam, R.; Desai, K.; Lee, S.; Rohrbach, M.; Batra, D.; Parikh, D. Probabilistic neural symbolic models for interpretable visual question answering. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6428–6437.
30. Bosselut, A.; Le Bras, R.; Choi, Y. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, No. 6, pp. 4923–4931.
31. Banu, A. Ontologies for Knowledge Representation: Tools and Techniques for Building Ontologies. In *Semantic Web Technologies*; CRC Press: Boca Raton, FL, USA, 2017; pp. 223–244.

32. Lawson, C.; Larson, K.; Van Erdewyk, J.; Smith, C.; Rizzo, A.; Ross, L.; Rendell, M. A Facilitated Interface to Generate a Combined Textual and Graphical Database System Using Widely Available Software. *J. Softw. Eng. Appl.* **2012**, *5*, 789. [\[CrossRef\]](#)
33. Yoo, K. Knowledge graph-based knowledge map for efficient expression and inference of associated knowledge. *J. Intell. Inf. Syst.* **2021**, *27*, 49–71.
34. Chai, X. Diagnosis method of thyroid disease combining knowledge graph and deep learning. *IEEE Access* **2020**, *8*, 149787–149795. [\[CrossRef\]](#)
35. Tsai, H.; Riesa, J.; Johnson, M.; Arivazhagan, N.; Li, X.; Archer, A. Small and practical BERT models for sequence labeling. *arXiv* **2019**, arXiv:1909.00100.
36. Szymanski, B.K. A simple solution to Lamport's concurrent programming problem with linear wait. In Proceedings of the 2nd International Conference on Supercomputing, Saint Malo, France, 4–8 July 1988; pp. 621–626.

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.