




Article

Semantic Scene Graph Generation Using RDF Model and Deep Learning

Seongyong Kim ¹, Tae Hyeon Jeon ², Ilsun Rhiu ³ , Jinhyun Ahn ⁴  and Dong-Hyuk Im ^{5,*} ¹ Department of Big Data and AI, Hoseo University, Asan 31499, Korea; yaba96@hoseo.edu² Department of Computer Engineering, Hoseo University, Asan 31499, Korea; 20161557@vision.hoseo.edu³ Division of Future Convergence (HCI Science Major), Dongduk Women's University, Seoul 02748, Korea; isrhiu@dongduk.ac.kr⁴ Department of Management Information Systems, Jeju National University, Jeju 63243, Korea; jha@jeju.ac.kr⁵ School of Information Convergence, Kwangwoon University, Seoul 01890, Korea

* Correspondence: dhim@kw.ac.kr

Abstract: Over the last several years, in parallel with the general global advancement in mobile technology and a rise in social media network content consumption, multimedia content production and reproduction has increased exponentially. Therefore, enabled by the rapid recent advancements in deep learning technology, research on scene graph generation is being actively conducted to more efficiently search for and classify images desired by users within a large amount of content. This approach lets users accurately find images they are searching for by expressing meaningful information on image content as nodes and edges of a graph. In this study, we propose a scene graph generation method based on using the Resource Description Framework (RDF) model to clarify semantic relations. Furthermore, we also use convolutional neural network (CNN) and recurrent neural network (RNN) deep learning models to generate a scene graph expressed in a controlled vocabulary of the RDF model to understand the relations between image object tags. Finally, we experimentally demonstrate through testing that our proposed technique can express semantic content more effectively than existing approaches.

Keywords: scene graph; RDF model; deep learning; image annotation



Citation: Kim, S.; Jeon, T.H.; Rhiu, I.; Ahn, J.; Im, D.-H. Semantic Scene Graph Generation Using RDF Model and Deep Learning. *Appl. Sci.* **2021**, *11*, 826. <https://doi.org/10.3390/app11020826>

Received: 26 November 2020

Accepted: 14 January 2021

Published: 17 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. 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

The total amount of digital image content has increased exponentially in recent years, owing to the advancement of mobile technology and, simultaneously, content consumption has greatly increased on several social media platforms. Thus, it becomes increasingly more imperative to effectively store and manage these large amounts of images. Hitherto, image annotation techniques able to efficiently search through large amounts of image content have been proposed [1–5]. These techniques represent images in several ways by storing semantic information describing the image content along with the images themselves. This enables users to accurately search for and classify a desired image within a large amount of image data. Recently, scene graph generation methods for detecting objects in images and expressing their relations have been actively studied with advancements in deep learning technology [6,7]. Scene graph generation involves the detection of objects in an image and relations between these objects. Generally, objects are detected first, and the relations between them are then predicted. Relation detection using language modules based on a pretrained word vector [6], and through message interaction between object detection and relation detection [7], have been used for this prediction. Although scene graph generation complements the ambiguity of image captions expressed in natural language, it does not effectively express semantic information describing an image. Therefore, a method of adding semantic information by incorporating the Resource Description Framework (RDF)

model [8] into conventional scene graph generation is proposed in this study. Although several works in the relevant literature [2–4,9,10] also applied the RDF model to image content, these studies did not utilize deep learning technology-based models. The proposed method detects image objects and relations using deep learning and attempts to express them using an RDF model, as shown in Figure 1.

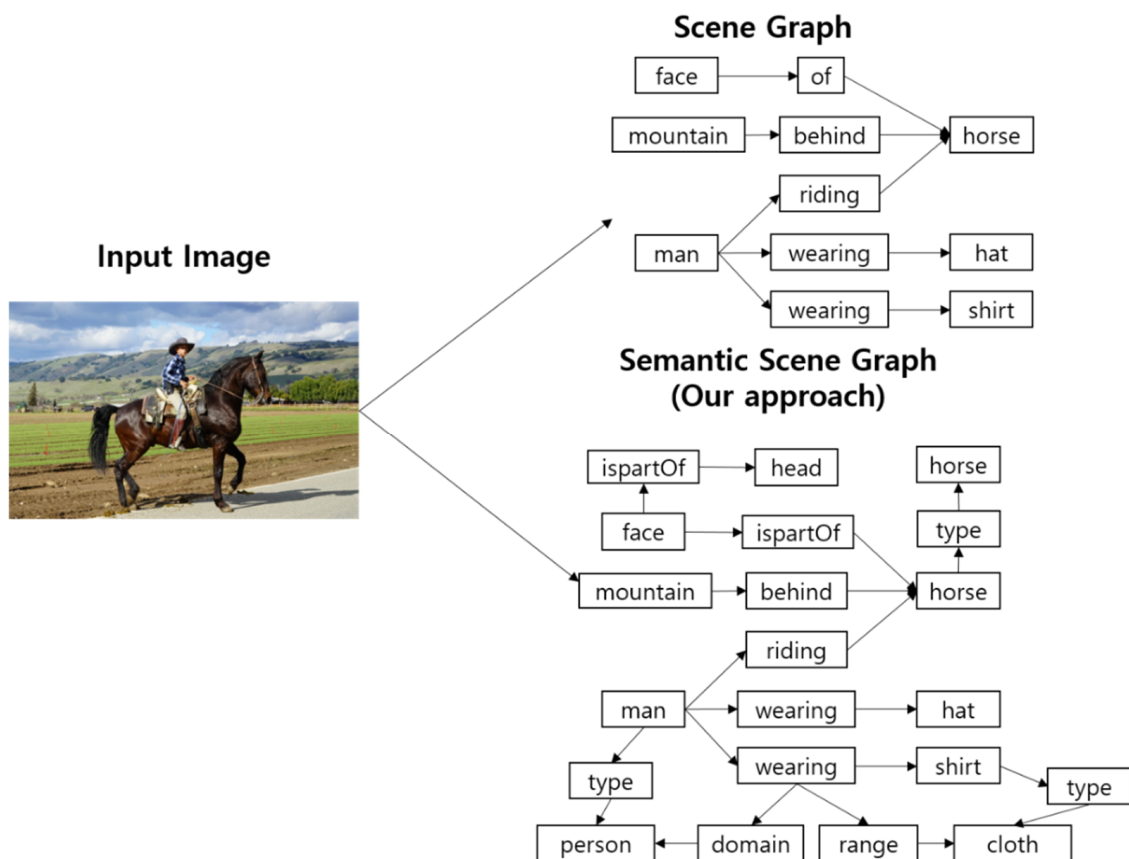


Figure 1. Overview of semantic scene graph generation.

The contributions of the present study are as follows. First, an RDF model for generating a scene graph is proposed. Semantic expression in a controlled vocabulary is possible by expressing the existing scene graph using an RDF model. Specifically, scene graph sentences that are logically contradictory can be filtered out using an RDF schema (RDFS). Furthermore, an image can be queried using SPARQL [11], an RDF query language. Second, deep learning technology was applied to the RDF model-based scene graphs. Although image content was also described in [3,10] using the RDF model, a user had to manually input the relation in [3], and a machine learning method was applied to detect only image tags in [10]. In this study, relations between image tags were detected using a deep learning model during generation of image scene graphs. Furthermore, subjects and objects were detected using a Convolutional Neural Network (CNN) model in the RDF-based scene graph, and a property relation was trained using a Recurrent Neural Network (RNN) model. Third, our results indicate that the expression of scene graphs can be improved significantly using the proposed inference approach with RDF-based scene graphs.

The remainder of this paper proceeds as follows. Various image annotations and image search methods are examined as related work in Section 2, and the proposed methods are described in Section 3. We conclude by presenting test results through a comparison with the conventional method in Section 4.

2. Related Work

Ontology-based image annotation has been studied extensively [1–5,8]. Image annotation using tags is mainly used for image search by identifying rankings and meanings of tags. I-TagRanker [5] is a system that selects the tag that best represents the content of a given image among several tags attached to it. In this system, first, a tag extension step is performed to find images similar to the given image, following which tags are attached to them. The next step is to rank the tags using WordNet (<https://wordnet.princeton.edu/>) in the order that they are judged to be related to the image while representing detailed content. The tag with the highest ranking is the one that best represents the image [12] suggested semiautomated semantic photo annotation. For annotation suggestion, [12] proposed an effective combination of context-based methods. Contextual concepts are organized into ontologies that include locations, events, people, things and time.

An image annotation and search based on an object-relation network was proposed in [13]. Their method entailed finding objects within images based on a probability model by finding parts of images, referred to as segments, and searching images using ontologies to represent the relation between objects. Compared to other approaches, images that are much more semantically similar can be distinguished using this method.

There have also been extensive studies conducted on image processing and tagging in mobile devices [14]. Images were searched using context information provided by smart devices in [14]. The context information used included time and location, as well as social and personal information. Context information was annotated when capturing a camera image; subsequently, the annotated information enables a user to find the desired image through a search. However, only the query words provided by the system can be used because the annotation form is standardized.

A model that shows the semantic relation between tags in an image using a knowledge graph was proposed in [15]. A relational regularized regression CNN (R^3 CNN) model was initialized along with AlexNet pretrained using ImageNet, using five layers of CNN, three pooling layers and two fully connected layers in [15]. First, tags were extracted from an image, then the image tags were vectorized using equations proposed in the study and the distance between vectors was calculated. Subsequently, the vectors were used for training. Although [15] is extremely useful for detecting relations between image tags, their method requires a well-established knowledge base.

3. Proposed System

3.1. Semantic Scene Graph

In this study, we constructed semantic scene graphs using RDF. The RDF was established by the W3C (World Wide Web Consortium) and was designed to be used to describe additional information pertaining to objects to be described, as well as hierarchical and similarity relations between data. In other words, it provides a way to define data and provide a description or relation.

The RDF is usually described using a triple model with a subject, predicate and object. The subject denotes the resource data to be represented, and the predicate denotes characteristics of the subject and indicates a relation between the subject and the object. The object, then, is the content or value of a description. Each value can be described using a uniform resource identifier (URI). The RDF not only describes actual data but also supports the RDFS, a schema that describes the types of terms used in the data and relations between them.

If the scene graph is described using the RDF(S) model, it can be utilized as shown in Figure 2. The upper and lower parts of Figure 2 represent the RDF model and RDFS, respectively. Our proposed scene graph generation method creates a representation of relations between objects in an image and can prevent semantically incorrect scene graphs, such as “Jone wears Banana” or “Jane eats shirt,” from being generated. For instance, RDFS can set “person” as the subject and “fruit” as the object for the predicate “eats.” Likewise, “person” could be used as the subject and “cloth” as the object of the predicate “wears,” to

generate a semantically correct scene graph. Although image annotations were represented using the RDF model in [3,4], in these works the user manually inputs the subject, predicate and object. The uniqueness of the present study lies in our application of the RDF model to scene graph generation using a machine learning algorithm.

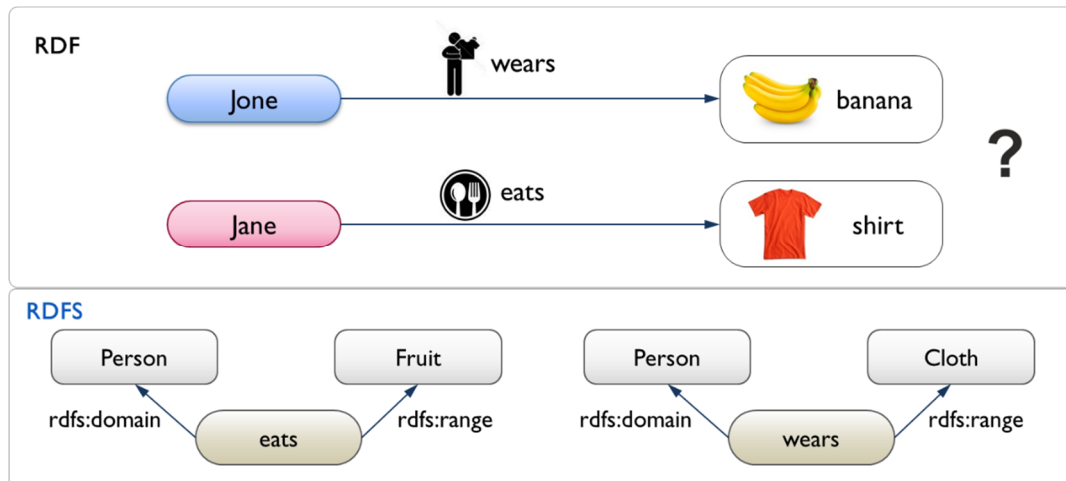


Figure 2. Resource description framework (RDF) and RDF schema for semantics.

3.2. Creating a Deep Learning-Based Scene Graph

Image tagging was performed using a CNN for image annotation in a previous study [10]. However, training images using a CNN makes it impossible to understand the semantics of objects. Furthermore, since the previous method made a direct SPARQL query on DBpedia (<https://wiki.dbpedia.org/>) for detecting relations between images tags, it was impossible to find a relation that was not already stored in DBpedia. Therefore, although we performed image tagging using a CNN, as in the previous study, our approach predicted relations between tags after vectorization and image tagging using a long short-term memory (LSTM) network, as well as gated recurrent units (GRU), an RNN model. The overall process of the proposed method is as shown in Figure 3.

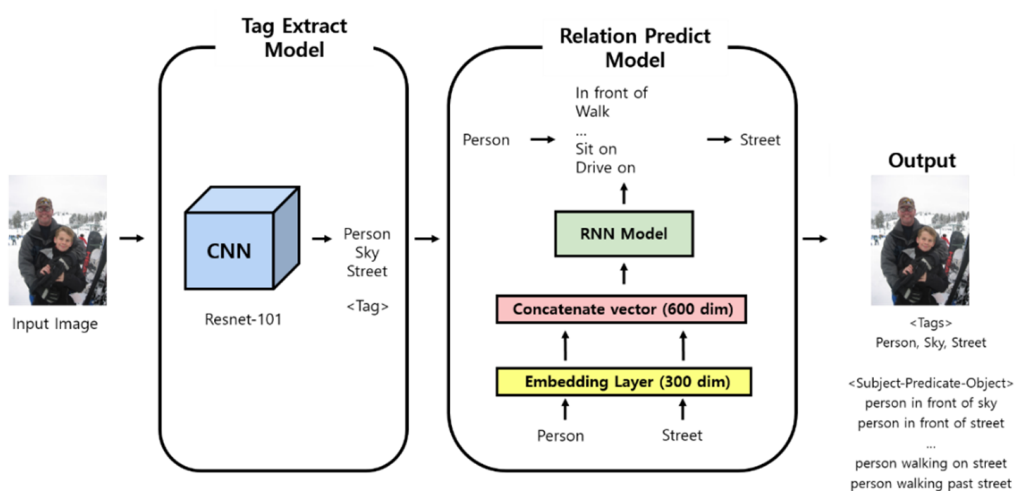


Figure 3. Proposed system.

First, in our proposed method, tags are created for image objects using a CNN model. In particular, we used the ResNet-101 model [16] as the state-of-the-art (SOTA) CNN model for image tagging because it uses skip-connection techniques and has effective performance on ImageNet. The process for image tagging using ResNet-101 is as follows. First, the CNN model extracts the features of images using a filter and then stores them in various

feature maps. The fully connected (FC) layer uses a value projected by the number of each class (tags) from each layer to classify the final image of the extracted feature map. In the output layer, the predicted value of the image tag t_i is output using the Sigmoid activation function, shown in Equation (1) below:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

The Sigmoid function σ is used for outputting values for each class and is defined as follows in Equation (2):

$$t_i = \sigma(Wx_i + b) \tag{2}$$

where W is the FC layer value, and b is the bias term. Then, we adopt a binary cross-entropy loss function given as follows:

$$\mathcal{L}_1 = \sum_{i=1}^N \left(\sum_{j=1}^M y_{ij} \log(t_{ij}) + (1 - y_{ij}) \log(1 - t_{ij}) \right) \tag{3}$$

where x_i is the input image, N is the number of image datasets, M is the number of tags, t_{ij} is a predicted value of a tag j for an image x_i , and y_{ij} is the correct answer for the image x_i . For example, $y_{ij} = 1$ indicates that the image x_i contains tag j . The process of detecting an object in an image using the CNN model is shown in Figure 4.

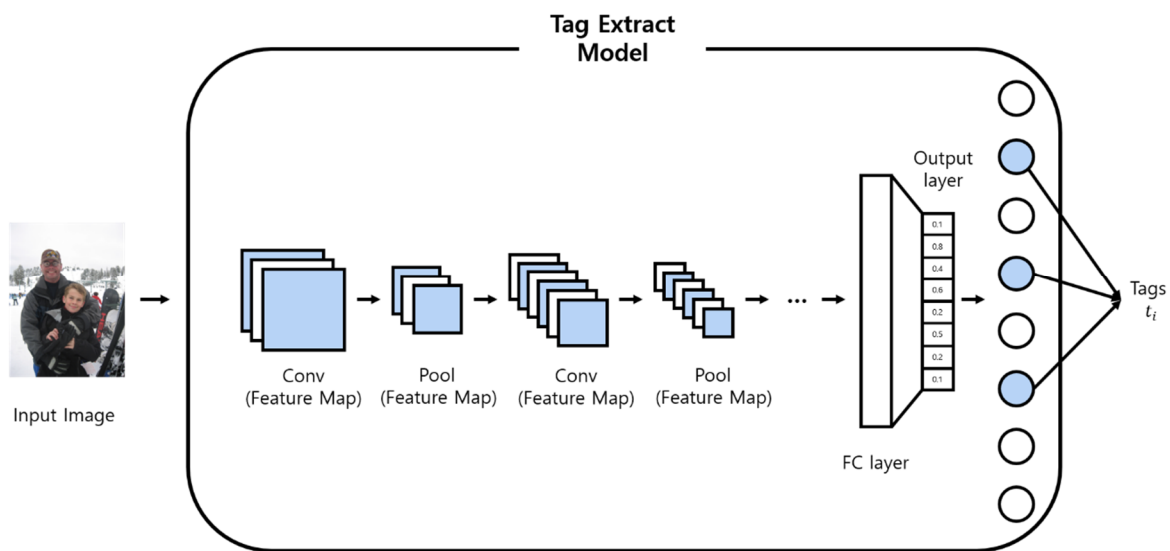


Figure 4. Convolutional neural network (CNN) model for image tagging.

The disadvantage of processing images using a CNN is that it becomes impossible to understand the semantics of the objects (tag words). To understand the meaning of words, techniques such as Word2Vec [17], which expresses words as vectors, have been created. Furthermore, systems such as ConceptNet [18] have become available to help computers understand the relationships between words. ConceptNet is a semantic network designed to enable computers to understand the semantics of words used by people. It provides a 300-dimensional tag representation vector w_i per word embedded. w_i is a 300-dimensional vector that passes through the embedding layer in the image tag t_i , represented as follows:

$$w_i = \text{word2vec}(t_i) \left(w_i \in \mathcal{R}^{300} \right) \tag{4}$$

In this paper, we propose an RDF model-based scene graph describing an image using triple sets of the form <Subject, Predicate, Object> (For convenience, the <Subject-Predicate-Object> triple is denoted as <Subj, Pred, Obj>. The <Subject, Object> pair is expressed

as <Subj, Obj>.). Thus, we predict the predicate p_j using t_{subj} and t_{obj} , selected from the image tag set $\{t_1, t_2, \dots, t_i\}$. We use an RNN model, able to store previous data in a hidden state, to predict the sequence relationship of <Subj, Pred, Obj>. When a value x_t is input at a time step t , h_t is obtained and recorded by calculating with the previous hidden state h_{t-1} . In particular, we used LSTM [19] architecture and GRUs [20], which are RNN models with excellent performance. LSTM learns by controlling memory cells with three gates (input, forget and output) in the hidden state. It also converts backpropagation into addition after multiplication by the RNN. Thus, it alleviates the disadvantages of gradient vanishing and exploding. GRUs modify the three gates of LSTM to two gates (update and reset) and have the advantage of being relatively simpler than LSTM.

In this study, t_{subj} and t_{obj} were not sequenced. For example, we cannot know which t_{obj} comes after the word 'person.' Thus, our method uses the pair <Subj, Obj>, combining t_{subj} and t_{obj} . Because a general RNN model flows in one direction, we have <Subj, Obj> \neq <Obj, Subj>. t_{subj} and t_{obj} are transformed into w_{subj} and w_{obj} respectively. $w_{\langle subj,obj \rangle}$ is defined as follows as Equation (5).

$$w_{\langle subj,obj \rangle} = \text{concat} [w_{subj}, w_{obj}] \left(w_{\langle subj,obj \rangle} \in \mathcal{R}^{600} \right) \tag{5}$$

From Equation (5), we can calculate the predicate p_j as follows:

$$p_j = \sigma \left(\text{LSTM} \left(w_{\langle subj,obj \rangle} \right) \right) \tag{6}$$

Finally, the loss function is defined as:

$$\mathcal{L}_2 = \sum_{\langle subj,obj \rangle} \sum_{j=1}^M y_j \log(p_j) + (1 - y_j) \log(1 - p_j) \tag{7}$$

The relation prediction process is illustrated in Figure 5. We summarize the algorithm of the semantic scene graph, as described in Algorithm 1.

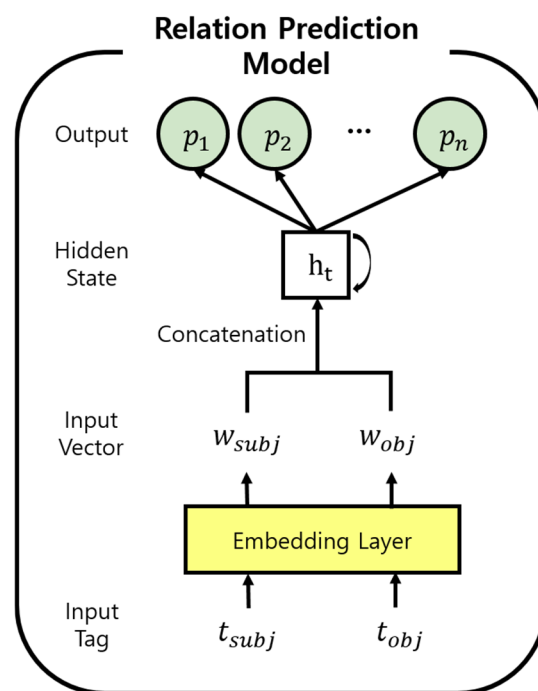


Figure 5. Relation prediction process.

Algorithm 1 Training Algorithm.

Require: image tags $\{t_1, t_2, \dots, t_i\} \in T$

Procedure

- 1: Fine tune image tagging model on images using ResNet (Equations (1)–(3))
- 2: for $i = 1, 2, \dots$ epoch do
- 3: for t_{subj} in T
- 4: for t_{obj} in T
- 5: tags t_{subj}, t_{obj} convert to $w_{\langle subj,obj \rangle}$ (Equations (4) and (5)).
- 6: Predict p_j using $w_{\langle subj,obj \rangle}$ (Equation (6))
- 7: Use stochastic gradient descent to find optimal and backward LSTM using loss function (Equation (7))

End procedure

3.3. Scene Graph Expansion Using Inference

The existing RDF graph introduced in the RDFS has the RDF entailment rule [21] as a constraint. Essentially, it has a structure that infers a sentence from the presence of another sentence. The entailment rule considered in this study was the RDF entailment rule derived from the RDF semantics, as shown in Table 1. For example, Rule 11 shows the subClassOf relation as a transitive closure. If resource U is a subclass of resource V, and resource V is a subclass relation of resource X, automatically resource U has a subclass relation of resource X. Although this extended entailment rule and data type entailment rule are provided, additional entailment rules are not considered in this study.

Table 1. RDFS entailment rules.

Rule	If E Contains	Then Add:
1.	X A Y	A rdf:type rdf:Property
2.	A rdfs:domain X U A Y	U rdf:type X
3.	A rdfs:range X Y A V	V rdf:type X
4.	U A B B A U	U rdf:type rdfs:Resource U rdf:type rdfs:Resource
5.	U rdfs:subPropertyOf V V rdfs:subPropertyOf X	U rdfs:subPropertyOf X
6.	U rdf:type rdf:PropertyOf	U rdfs:subPropertyOf U
7.	A rdf:subPropertyOf B U A Y	U B Y
8.	U rdf:type rdfs:Class	U rdfs:subClassOf rdfs:Resource
9.	U rdf:subClassOf X V rdf:type U	V rdf:type X
10.	U rdf:type rdfs:Class	U rdfs:subClassOf U
11.	U rdfs:subClassOf V V rdfs:subClassOf X	U rdfs:subClassOf X
12.	U rdf:type rdfs:ContainermembershipProperty	U rdfs:subProperty rdfs:member
13.	U rdf:type rdfs:Datatype	U rdfs:subClassOf rdfs:Literal

In this study, we exploited the RDFS entailment rule to generate a semantic scene graph. Additionally, user-defined rules were applied for inference in this study. Although user-defined rules were not defined in the RDFS entailment rules, they enable users to create their own rules and apply them to image annotations. However, inference based on the existing RDFS entailment rules is not well-adapted to context-specific inference because the predefined rules are a limitation. Therefore, the user can infer a triple that represents an image annotation after defining the inference suitable to the situation in advance. For example, if the annotation information on the location of the image taken using a mobile device is “Seoul” and there is a triple <Seoul, isPartOf, Korea>, the location of the image includes “Korea.” Additionally, user-defined rules that can represent general situations were applied in this study, as shown in Table 2. Although [22] also proposed semantic inference rules in the domain of fuel cell microscopy, [22] assists users in constructing rules with domain-specific features, such as colors and shapes.

Table 2. Example of User-Defined Rules.

Rule Type	Example
Season Inference	Between March and May it is spring, while from June to August it is summer, and from September to November it is autumn, and from December to February it is winter
Location Inference	If the location is part of Seoul, the location is also part of South Korea
Time Inference	If the time is eight to twelve o'clock, it is morning

4. System Implementation and Testing

The test dataset used in this study for visual relation detection [6] was composed of 5000 images. The experimental dataset had a total of 37,993 relations, 100 objects and 70 predicates. Duplicate relations with the same subject-predicate-object were removed, regardless of the image. Therefore, 4000 images and 6672 relations were used; among which 1000 images and 2747 relations were used as test data. Eighty-six <Subject, Object> pairs that were not used in training were used to predict the relations.

The test was implemented using Windows 10, with a Nvidia GTX 2060 6 GB GPU, running Python 3.7.4 and PyTorch 1.5.1 to conduct training. All the training processes took place in the same environment.

We combined CNN and RNN models to implement the proposed method. The ResNet-101 architecture using ImageNet was used for the CNN model. The weights were initialized with values pre-trained on ImageNet. The following Table 3 shows the Top-K error index tested using ImageNet.

Table 3. CNN backbone Top-K Error.

Architecture	TOP-1 Error	TOP-5 Error
Ours	22.73	6.44
ResNet-101 [16]	21.75	6.05

We compared our method with [16] as the SOTA image recognition model [16] improved performance by using a skipped-connection technique. In our model, we replaced the last layer of ResNet-101, the FC layer, with the number of t_i and performed fine-tuning. In the case of the ResNet-101 model, the learning rate was 0.001, reduced by 0.1 every 20 epochs. Top-1 error and Top-5 error indexes were used to evaluate image classification performance. Top-1 error is the error rate in which the top selection predicate by the model is not the correct answer. Top-5 error is the error rate without a correct answer among the top five categories predicted by the model. Although our performance was lower than

that of [16] for image recognition, we focused on predicting the relationships between image tags.

The result of predicting the predicate using the <Subject, Object> pair after extracting the tag using the CNN is shown in Figure 6. In the case of the LSTM model, the learning rate was 0.005, the embedding size was 300, the hidden size was 128 and the num layer was set to 1. The test was conducted by setting the prediction of each relation such that it exceeded 0.05 probability when passing through the Sigmoid activation function during relation prediction. Precision, recall and F1-Score were used as evaluation criteria. We found that the proposed LSTM method had higher precision than the GRU method, as shown in Figure 6. Although the recall of the proposed method was lower than that of the GRU, its F1-Score was higher.

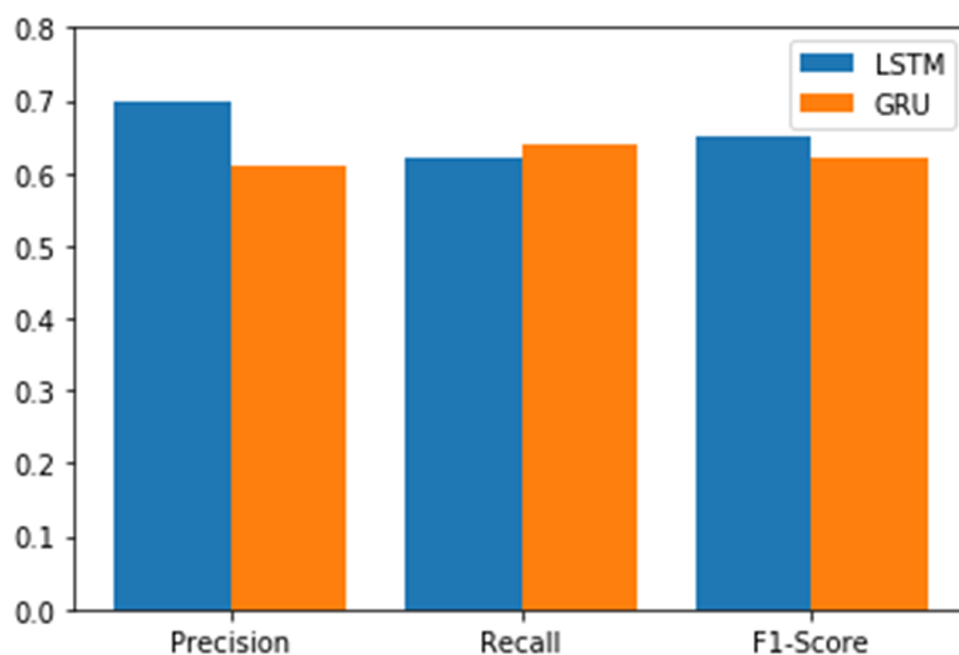


Figure 6. Performance evaluation. Long short-term memory (LSTM) vs. gated recurrent units (GRU).

An example of the proposed method is shown in Figure 7. The first image shows the triple <Subject, Predicate, Object> obtained through the LSTM. However, some cases such as the triple <chair, touch, table> are not semantically correct because the “touch” predicate uses “person” as its domain type. The <person, wear, horse> triple in the third picture is also a semantically incorrect expression because the type range of “wear” is limited to “cloth.” However, a predicate that is not related to an actual image can be created. This is because all possible predicates were created based on <Subject, Object> in the dataset. For example, output tags {person, shirt, hat, horse, phone, street} were created in the third picture. Then, we predicted the predicate for all pairs of output tags, such as <person, shirt>, <person, hat>, . . . , <street, phone>.

Input Image



Output Tags

person, table, shirt, chair

Subject-Predicate-Object

person-on-table
 person-above-table
 person-stand under-table
 person-with-table
 person-on the left of-table
 person-on the right of-table
 person-look-table
 person-stand on-table
 person-sit next to-shirt
 person-in front of-shirt
 person-hold-shirt
 person-by-shirt
 person-on the right of-shirt
 person-sit next to-chair
 person-behind-chair
 person-in front of-chair
 person-touch-chair
 person-rest on-chair
 table-has-person
 table-by-person
 table-on the left of-person
 table-on the right of-person
 table-by-chair
 table-on the left of-chair
 shirt-under-person
 shirt-on-table
 chair-near-person
 chair-below-person
 chair-beside-person
 chair-hold-person
 chair-by-person
 chair-on the left of-person
 chair-carry-person
 chair-under-table
 chair-on the left of-table
 chair-touch-table
 chair-hold-shirt

Input Image



Output Tags

person, shirt, tree, bottle,
 hand, sunglasses

Subject-Predicate-Object

person-sit next to-shirt
 person-in front of-shirt
 person-hold-shirt
 person-by-shirt
 person-on the right of-shirt
 person-in front of-tree
 person-beside-tree
 person-on the right of-tree
 person-wear-bottle
 person-next to-bottle
 person-carry-bottle
 person-in-hand
 person-near-sunglasses
 person-in-sunglasses
 person-with-sunglasses
 shirt-under-person
 shirt-in-hand
 tree-above-person
 tree-in the front of-person
 tree-near-person
 tree-beside-person
 tree-over-person
 tree-on the right of-person
 tree-cover-person
 tree-across-person
 tree-taller than-person
 tree-near-shirt
 tree-below-shirt
 tree-over-shirt
 tree-on the top of-shirt
 tree-cover-shirt
 tree-next to-bottle
 tree-near-bottle
 tree-near-hand
 tree-in-hand
 tree-over-hand
 tree-near-sunglasses
 tree-beside-sunglasses
 tree-over-sunglasses
 bottle-above-person
 bottle-on the left of-person
 bottle-carry-person
 bottle-in-hand
 hand-above-person
 hand-behind-person
 hand-in front of-person
 hand-on the right of-person
 hand-in the front of-tree
 hand-next to-bottle
 sunglasses-above-person
 sunglasses-behind-person
 sunglasses-by-person
 sunglasses-on the left of-person
 sunglasses-on the right of-person
 sunglasses-has-shirt
 sunglasses-on-tree
 sunglasses-next to-tree
 sunglasses-in front of-tree
 sunglasses-next to-bottle
 sunglasses-in-hand

Input Image



Output Tags

person, shirt, hat,
 horse, phone, street

Subject-Predicate-Object

person-sit next to-shirt
 person-in front of-shirt
 person-hold-shirt
 person-by-shirt
 person-on the right of-shirt
 person-below-hat
 person-beneath-hat
 person-touch-hat
 person-wear-horse
 person-has-horse
 person-stand next to-horse
 person-behind-horse
 person-in front of-horse
 person-near-horse
 person-below-horse
 person-hold-horse
 person-by-horse
 person-on the top of-horse
 person-look-horse
 person-touch-horse
 person-adjacent to-horse
 person-taller than-horse
 person-next to-phone
 person-by-phone
 person-with-phone
 person-on the left of-phone
 person-touch-phone
 person-in front of-street
 person-walk-street
 person-walk past-street
 person-by-street
 person-on the right of-street
 person-sit on-street
 person-touch-street
 person-drive on-street
 horse-sit next to-person
 horse-above-person
 horse-with-person
 horse-carry-person
 horse-beneath-hat
 horse-touch-street
 shirt-under-person
 shirt-on-street
 hat-next to-phone
 hat-on the right of-street
 hat-drive on-street
 hat-has-person
 hat-above-person
 phone-has-person
 phone-above-person
 phone-in front of-person
 phone-by-person
 phone-on the right of-person
 phone-lying on-person
 phone-has-shirt
 phone-in front of-shirt
 phone-on the top of-shirt
 phone-beneath-hat
 phone-behind-horse
 phone-by-horse
 phone-on-street
 phone-over-street
 phone-by-street
 phone-on the top of-street
 phone-across-street
 phone-park on-street
 street-has-person
 street-near-person
 street-beside-person
 street-on the right of-person
 street-wear-shirt
 street-has-shirt
 street-has-hat
 street-beneath-hat
 street-near-horse
 street-by-horse
 street-on-phone
 street-next to-phone
 street-near-phone
 street-by-phone

Figure 7. Semantic scene graph.

5. Conclusions

With recent advances in the mobile environment and the exponential growth of image content, semantic image search has become critical. In this study, an RDF model was incorporated into the proposed scene graph generation method, as the semantics of the scene graph could be made more meaningful by describing it using the RDF model. Furthermore, the generated scene graphs could be described logically because they used a controlled vocabulary. CNN and RNN models were also applied to the RDF-based scene graph generation method to learn the semantic information of images. RDF model inference rules, and user-defined rules were used to enrich the annotation information of the scene graph.

In the future, we will conduct research building on the present study to further specify semantic images information using a deep learning model based on image context information in a semantic scene graph in RDF format.

Author Contributions: S.K. and D.-H.I. conceived the problem and supervised the overall research; T.H.J. implemented the algorithm and performed the experiments; I.R. and J.A. clarified some points that helped D.-H.I. write algorithm; S.K., I.R., and D.-H.I. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2021-2018-0-01417) supervised by the Institute for Information & Communications Technology Promotion (IITP). This work was also supported by the Research Resettlement Fund for the new faculty of Kwangwoon University in 2020. Further, this research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1D1A3A03103802).

Institutional Review Board Statement: The study did not require ethical approval.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. Margues, O.; Barman, N. Semi-automatic Semantic Annotation of Images using Machine Learning Techniques. In Proceedings of the ISWC: International Semantic Web Conference, Sanibel Island, FL, USA, 20–23 October 2003.
2. Gayo, J.E.L.; De Pablos, P.O.; Lovelle, J.M.C. WESONet: Applying semantic web technologies and collaborative tagging to multimedia web information systems. *Comput. Hum. Behav.* **2010**, *26*, 205–209. [\[CrossRef\]](#)
3. Im, D.-H.; Park, G.-D. Linked tag: Image annotation using semantic relationships between image tags. *Multimed. Tools Appl.* **2014**, *74*, 2273–2287. [\[CrossRef\]](#)
4. Im, D.-H.; Park, G.-D. STAG: Semantic Image Annotation Using Relationships between Tags. In Proceedings of the 2013 International Conference on Information Science and Applications (ICISA), Pattaya, Thailand, 24–26 June 2013; pp. 1–2.
5. Jeong, J.; Hong, H.; Lee, D. i-TagRanker: An Efficient Tag Ranking System for Image Sharing and Retrieval Using the semantic Relationships between Tags. *Multimed. Tools Appl.* **2013**, *62*, 451–478. [\[CrossRef\]](#)
6. Lu, C.; Krishna, R.; Bernstein, M.; Fei-Fei, L. Visual Relationship Detection with Language Priors. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016.
7. Li, Y.; Ouyang, W.; Zhou, B.; Wang, K.; Wang, X. Scene Graph Generation from Objects, Phrases and Region Captions. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1270–1279.
8. Resource Description Framework(RDF): Concepts and Abstract Syntax. Available online: <http://w3.org/TR/2014/REC-rdf11-concepts-20140225> (accessed on 17 November 2020).
9. Chen, H.; Guo, A.; Ni, W.; Cheng, Y. Improving the representation of image descriptions for semantic image retrieval with RDF. *J. Vis. Commun. Image Represent.* **2020**, *73*, 102934. [\[CrossRef\]](#)
10. Shin, Y.; Seo, K.; Ahn, J.; Im, D.H. Deep-learning-based image tagging for semantic image annotation. In *Advanced in Computer Science and Ubiquitous Computing*; Springer: Berlin/Heidelberg, Germany, 2020.
11. SPARQL Query Language for RDF. Available online: <http://w3c.org/TR/rdf-sparql-query/> (accessed on 17 November 2020).
12. Elliott, B.; Ozsoyoglu, M. A comparison of methods for semantic photo annotation suggestion. In Proceedings of the 22nd International Symposium on Computer and Information Sciences, Ankara, Turkey, 7–9 November 2007.

13. Chen, N.; Zhou, Q.; Prasanna, V. Understanding Web Image by Object Relation Network. In Proceedings of the International Conference on World Wide Web, Raleigh, SC, USA, 26–30 April 2010.
14. Xia, S.; Gong, X.; Wang, W.; Tian, Y.; Yang, X.; Ma, J. Context-Aware Image Annotation and Retrieval on Mobile Device. In Proceedings of the 2010 Second International Conference on Multimedia and Information Technology, Kaifeng, China, 24–25 April 2010.
15. Cui, P.; Liu, S.; Zhu, W. General Knowledge Embedded Image Representation Learning. *IEEE Trans. Multimed.* **2017**, *20*, 198–207. [[CrossRef](#)]
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2015**, arXiv:1512.03385.
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the ICLR, Scottsdale, AZ, USA, 2–4 May 2013.
18. Speer, R.; Chin, J.; Havasi, C. *ConceptNet5.5: An Open Multilingual Graph of General Knowledge*; AAAI: Menlo Park, CA, USA, 2017.
19. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
20. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In Proceedings of the NIPS, Montréal, QC, Canada, 8–13 December 2014.
21. Hayes, P. RDF Model Theory. W3C Working Draft. Available online: <http://www.w3.org/TR/2001/WD-rdf-mt-20010925/> (accessed on 17 November 2020).
22. Hollink, L.; Little, S.; Hunter, J. Evaluating the application of semantic inferencing rules to image annotation. In Proceedings of the 3rd International Conference on High Confidence Networked Systems, Berlin, Germany, 15–17 April 2004; Association for Computing Machinery (ACM): New York, NY, USA, 2005; p. 91.