Zero-shot Object Classification with Large-scale Knowledge Graph

Kohei Shiba, Yusuke Mukuta, Tatsuya Harada
The University of Tokyo
{shiba, mukuta, harada}@mi.t.u-tokyo.ac.jp

Abstract

Zero-shot learning is research for predicting unseen categories, and can solve problems such as dealing with unseen categories that were not anticipated at the time of training and the lack of labeled datasets. One of the methods for zero-shot object classification is using a knowledge graph, which is a set of explicit knowledge. Since recognition is limited to the categories contained in the knowledge graph and the relationships among categories are expected to be quantitatively and qualitatively richer depending on the graph size, it is desirable to handle a large-scale knowledge graph that contains as many categories as possible. We use a knowledge graph that contains about seven times as many categories as the knowledge graphs used mainly in existing research to enable classification of a larger number of categories and to achieve more accurate recognition. When using large-scale knowledge graph, it is expected that the number of noisy nodes and edges will increase. Therefore we propose a method to extract useful information from entire graph by positional relationships between categories and the types of edges in the knowledge graph. We classify images that were unclassifiable in existing research and show that the proposed data extraction method improves performance compared to using entire graph.

1. Introduction

In recent years, research on deep learning enables object classification with very high performance. [6, 7, 17, 21]. Usually, in object classification, a model is trained using labeled image dataset, and the target category are predicted from categories that appeared in the training phase. Therefore, to obtain a model with high performance, it is necessary to train the model on a large number of labeled image dataset. In addition, to classify new categories, it is necessary to collect the corresponding dataset and retrain the model. To solve such problems, zero-shot object classification, which aims to predict categories not included in the training dataset, has been widely studied [1].

In order to perform zero-shot object classification, knowledge obtained from categories in the training dataset should be used for unseen categories not existing at the time of training. One approach is to use knowledge graphs such as WordNet [12] and ConceptNet [16]. A knowledge graph is a representation of knowledge using nodes corresponding to words and concepts, and edges corresponding to relationships of nodes. The semantic distance between nodes in the knowledge graph is expressed in terms of hops. Nodes directly connected by an edge are 1 hop away from each other, and nodes that have a relationship with a node in between are 2 hops away from each other. In knowledge graphs used in zero-shot object classification, there are also nodes that not indicate nouns but other words such as verbs and adjectives, or concepts such as four legs, and edges not only connect nodes, but may also have information such as the type and strength of the relationship. When using the knowledge graph for zero-shot object classification, nodes in the neighborhood of each other have semantic similarity, and thus tend to have similar features in the image. Therefore, it is possible to infer which node corresponds to an unseen category image by using image features of the nodes in the neighborhood on the knowledge graph.

The advantage of zero-shot object classification is being performed without training dataset, but it is limited to categories contained in the knowledge graph. In addition, relationships between categories are expected to become quantitatively and qualitatively richer depending on the graph size. Therefore, it is desirable to use large-scale knowledge graphs containing as many categories as possible. In existing researches [8, 13, 19], WordNet or a part of it is mainly used as a knowledge graph. However, ConceptNet is a knowledge graph that includes a much larger nodes. Therefore, this study aims to propose a scalable zero-shot object classification method that can be applied to ConceptNet to enable classification of a wider range of categories and to achieve higher performance. However, using a large knowledge graph is expected to significantly increase the amount of noisy information, such as nodes that do not convey information between categories and edges that do not represent similarity. Therefore, we do not apply the large-scale knowledge graph to existing methods as is, but extract

information from the knowledge graph that is effective for zero-shot object classification.

Our contributions are:

- In zero-shot object classification, we applied a large-scale knowledge graph to classify categories that had not been the target of recognition by existing researches.
- In applying large-scale knowledge graph to zero-shot object classification, we proposed methods for extracting effective information from a knowledge graph based on the semantic connections between categories and the types of relationships.
- We show that the proposed data extraction methods improves the performance of zero-shot object classification in experiments.

2. Related Work

Zero-shot Object Classification. Methods for zero-shot object classification include using category attributes [3, 10], using semantic features of words and captions [5,11,15, 24], and using image generation models to generate training data for unseen categories [4, 20], and so on. Among them, Using information from the knowledge graph [8, 13, 19] is effective because a knowledge graph itself is a database containing a lot of categories and at the same time provides information on the relationships between categories that is useful for generalization. Actually, existing researches using knowledge graphs have shown particularly high performance.

Knowledge Graphs. In WordNet [12], each node is a synset, a set of synonymous words or phrases, and relationships in WordNet are shown with a clear hierarchical structure using 6 kinds of directed edges, such as Synonymy, Hyponymy, etc. In ConceptNet [16], each node is a word or phrase of natural language and may have multiple meanings. 36 types of edges representing various concepts, such as IsA, UsedFor, CapableOf, etc., are used to indicate relationships in ConceptNet. The number of nodes and edges in ConceptNet is considerably larger than that in WordNet.

Methods Using Knowledge Graphs. Wang et al. [19] proposed a method to construct a model for zero-shot object classification by learning the output of a graph convolutional network to regress to the final layer of a pre-trained image classifier, extracting information from two information representations: word embeddings, which are implicit feature representations, and knowledge graphs, which are explicit feature representations. However, their method uses only a part of WordNet because it requires the entire graph structure during training, and it cannot be applied to large graphs.

Kampffmeyer et al. [8] argued that GCNs with shallow layers should be used in zero-shot object classification with knowledge graphs, and proposed a connection scheme that incorporates distant information by directly connecting to descendant and ancestor nodes using a weighting method that considers the distance between nodes. This method showed even higher zero-shot object classification performance, but it assumes a clear hierarchical structure of WordNet and is not applicable to ConceptNet, which contains a wide range of concepts.

Nayak et al. [13] proposed a method to overcome the limitation of the knowledge graph and apply the information of large-scale and extensive concepts to zero-shot object classification. By simulating random walk on the nodes of a knowledge graph and selecting nodes with the highest hit probability for sampling, They avoids referring to the entire graph and can perform learning and inference even on large-scale graphs. The experimental setup for the ImageNet dataset in their study uses ConceptNet to train GNNs, but relies on WordNet for classifiable categories.

3. Method

As in previous studies using knowledge graphs, this study uses a method that takes as input information from embedding semantic features and constructs an object classification model that can classify zero-shot categories by graph convolutional network(GCN) [9] of a knowledge graph. In order to use a large knowledge graph, the model construction is based on the framework of Wang et al [13]. However, categories included in ConceptNet are the target of recognition, including categories that could not be classified by the WordNet-based method. In addition, to deal with noisy nodes and edges, which are expected to increase due to the large-scale concepts in ConceptNet, we use graph information such as the positional relationships among categories and types of edges in ConceptNet to extract only the information that is effective for zero-shot object classification from the entire graph.

3.1. Overall Pipeline

Figure 1 shows the overall pipeline. First, GCN based on a knowledge graph extracted from ConceptNet is trained using the weights of the final layer of pre-trained ResNet50 [6] as the training data. The input to the GCN is a word feature embedding vector corresponding to each node obtained by GloVe [14], and a feature vector corresponding to the weights of the final layer of ResNet50 is output from each node corresponding to the pre-trained category of ResNet50. In this training, L2 distance is minimized. The learned GCN outputs weights of a predictive classifier for the target categories of zero-shot object classification in addition to the training categories. The predictive classifier is replaced with the final layer of ResNet50. Next, with the

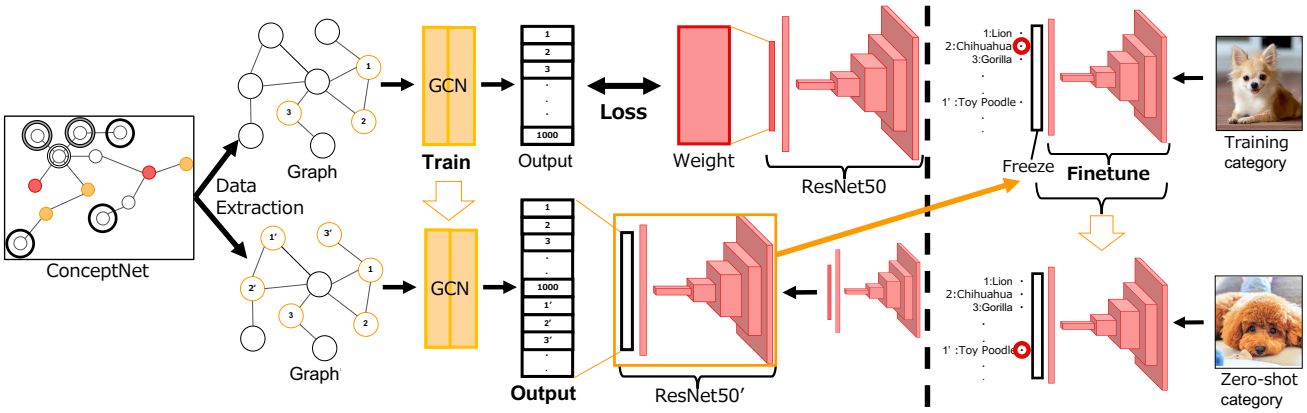

Figure 1. Overall pipeline. **Left:**GCN of a graph extracted from ConceptNet is trained using the final layer of ResNet50 as the training data, and outputs a predictive classifier for zero-shot categories in addition to the training categories. **Right:**Image classifier with a predictive classifier is finetuned with training categories to classify zero-shot categories.

predictive classifier weights freezed, the original classifier part which extracts the image feature is finetuned by learning a classification task for the training category. With the above training and finetuning, it is possible to classify the target category for zero-shot object classification.

Graph Convolutional NetWork. GCN proposed by Kiph et al. [9] can be described as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1}$$

Here, $H^{(l)}$ represents the features of each node in the l layer and $H^{(0)}$ represents the matrix of initial features of all nodes, $\tilde{A} = A + I$ (A is the adjacency matrix, I is the identity matrix), $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, and $W^{(l)}$ is the weight matrix in the l layer. Also, $\sigma(\cdot)$ represents the activation function. In this way, the node features are updated at each layer, and the weight matrix of each layer is learned by performing back propagation based on the loss between the output of the final layer and the training data.

According to Xu et al. [22], a graph convolutional neural network that aggregates feature information in the local neighborhood of a node can be described, using a function AGGREGATE to aggregate information from neighboring nodes and a function COMBINE to update node features from the aggregated nodes, as follows:

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)}, u \in \mathcal{N}(v)\right\}\right) \tag{2}$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right) \tag{3}$$

where $h_v^{(k)}$ represents the features of node v in the k-layer and $h_v^{(0)}$ is the initial features of node v. $a_v^{(k)}$ is the aggregated node information and $\mathcal{N}(v)$ is the set of nodes adjacent to node v.

Therefore, the GCN used in our model can be described as follows:

$$a_v^{(l)} = \text{Mean}\left(\left\{h_u^{(k-1)}, u \in \mathcal{S}(v)\right\}\right) \tag{4}$$

$$h_v^{(k)} = \sigma\left(W^{(k)}a_v^{(k)}\right) \tag{5}$$

The Mean is the process of taking the average of the nodes, and $\mathcal{S}(v)$ is the sampled set from nodes adjacent to v. The σ represents activation by LeakyReLU. The $W^{(k)}$ is the weight of the k-layer. Sampling is performed using a random-walk [23]. We simulate random walk with node v as the start point, moving according to edges. The number of visits by the random walk is counted at each neighboring node, and the number of visits for all neighboring nodes is normalized to obtain a hit rate that takes into account the importance of the node in the graph structure. When sampling, the nodes are selected in order of their hit rate.

We follow Nayak et al. [13] and use the 2-layer GCN in our experiments. In practice, it is difficult to increase the number of layers without reducing the number of samples, since the computational complexity of graph convolution is proportional to the product of the number of sampled objects in each layer.

3.2. Data Extraction from ConceptNet

3.2.1 2-Hop Node Choice

In the operation of obtaining a predictive classifier from a knowledge graph by performing graph convolution from neighboring nodes, the predictive classifier reflects the node information within the same hop count as the number of convolution layers. Therefore, when considering a 2-layer GCN, the predictive classifier obtained from a category is constructed based on the features of categories that exist

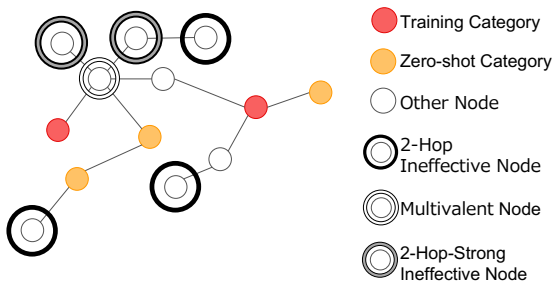

Figure 2. Example of ineffective Nodes. **2-Hop Ineffective Node** represents not existing within 2 hops of either the training category or the zero-shot candidate category, or both. **2-Hop-Strong Ineffective Node** represents existing within 2 hops of both categories only by being adjacent to a multivalent node.

within 2 hops. Therefore, only information from nodes that exist within 2 hops of any training category is used for training, and only information from nodes that exist within 2 hops of any zero-shot object classification candidate category is used for classifier prediction. From this, we hypothesize that only nodes that are within 2 hops of both any of the training categories and any of the candidate categories contribute to learning for zero-shot object classification.

Figure 2 explains that. The red nodes are the training categories and the yellow nodes are the zero-shot candidate categories. The nodes surrounded by black circles do not exist within 2 hops of either the training category or the candidate category, or both. These nodes may have a negative effect on the output of the categories, so removing them is expected to improve the performance of zero-shot object classification.

In addition, even if a node exists within 2 hops of both the training and candidate categories, it is possible that it does not contribute to the classifier's predictions. Large-scale knowledge graphs contain very multivalent nodes that are connected to many nodes and edges. Specifically, the most multi-valued node in our experimental setting has more than 10,000 edges. Therefore, we consider nodes such as the node surrounded by the gray circle in Figure 2, which exists within 2 hops of both categories only by being adjacent to a multivalent node that is adjacent to both the training category and the candidate category, have the effect of diluting the information of the multivalent node. Therefore, removing such nodes may also improve the performance of zero-shot object classification.

These methods remove nodes such as */c/en/giant_cockroach/n/wn/animal*, which is a noise node that has edges only between it and a training category */c/en/cockroach*.

Although this method is set up with explicit zero-shot candidate categories in spite of the zero-shot learning, since ConceptNet contains many words other than nouns and concepts, it is natural to exclude such nodes from the candidate categories and to target only in categories where visually recognizable objects exist, so this method is considered reasonable.

Graph	Nodes	Edges
WordNet	$82,115$	$75,850$
ConceptNet	$559,928$	$1,380,131$

Table 1. The number of nodes and edges in WordNet and ConceptNet.

3.2.2 Labeled Edge Choice

In ConceptNet, edges have labels that indicate the type of relationship. Many relationship labels, such as *red wolf SimilerTo eastern wolf*, directly indicate the similarity of image features of a category, while others, such as *suit AtLocation closet*, indirectly indicate the similarity of image features of a category, since the relationship between suit and other costumes can be traced through closet.

On the other hand, there are some labels that may impair the similarity of image features. For example, an edge labeled Antonym connects nodes with opposite meanings, such as *drop Antonym pickup*, so that no similarity of image features can be expected between categories indirectly connected through the edge. Therefore, by removing such edges from the graph, we can expect to improve the performance of zero-shot object classification by selecting more relevant categories for training and classifier prediction.

In this research, edges with labels, *Antonym, DistinctFrom, NotCapableof, NotDesires, NotHasProperty* are removed, taking into account the meaning of the label and the connection target.

3.2.3 Data Removal or Priority Change

When extracting datas using the method proposed above, there are two possible methods: one is to remove unnecessary nodes from the knowledge graph, and the other is to leave the nodes in the knowledge graph, but prioritize the information of important nodes for graph convolution. In this research, the nodes used in the graph convolution are determined by sampling based on the hit rate defined by random walk. Therefore, by reducing the hit rate of nodes that are considered ineffective, important nodes can be sampled with priority.

Layer	Samples	Input	Output
First Layer	200	300	2,048
Second Layer	100	2,048	2,049

Table 2. Details of the GCN used in the experiments.

4. Experimental Setup

4.1. Settings

4.1.1 Datasets

In this work, we use the same knowledge graphs used by Nayak et al [13]. The graph data of ConceptNet [16] is a graph in which each category in WordNet is mapped to a node in ConceptNet 5.7, and the nodes within 2hops of each category are collected. The nodes that exist within 2hops of each category on ConceptNet are collected. All non-English nodes and their edges are removed. All edges are made bidirectional. Next, for nodes that share the same noun prefix and are considered to represent the same category, only one node with the sum of their edges is used. For example, although both */c/en/lawyer* and */c/en/lawyer/n* exist in Conceptnet nodes, the edges corresponding to them are summed. The graph data for WordNet [12] is taken from the code obtained from Kampffmeyer et al. [8]. The number of edges is less than the number of nodes because the data is obtained by first specifying the categories and then obtaining the edges between the corresponding nodes. Table 1 shows details of the graph. As features for each node of the graph, word embedding from the pre-trained 300-dimensional GloVe 840B [14] is used. For categories such as idioms, embeddings for each word in the category are averaged.

For object classification, we mainly use ImageNet [2], which is a large object classification dataset containing more than 20,000 categories, with over 14 million images. For classification of additional categories, we use images out of a large dataset of natural images, iNat2021 [18].

4.1.2 Hyperparameters

The details of the GCN used in the experiments are shown in Table 2. The input is a 300-dimensional feature vector, the middle layer has 2,048 dimensions, and the output has 2,049 dimensions. In ResNet50, the classifier used in the experiment, the input to the final layer is a 2,048-dimensional vector, and the 2,049-dimensional output of the GCN corresponds to a 1-dimensional bias in addition to the 2,048-dimensional weights. For sampling, the top 100 nodes selected based on random walks among the nodes adjacent to the target node are used in the convolution of the second layer, and the top 200 nodes adjacent to the 100 nodes selected in the second layer are used in the convolution of the first layer.

In training of the GCN, Adam was used to update the weights. We trained 3,000 epochs with a learning rate of 0.001 and a weight decay of 0.0005. When finetuning ResNet50 with the predictive classifier obtained using the trained GCN as the final layer, we used SGD for updating the weights and trained 20 epochs with a learning rate of 0.0001 and a momentum of 0.9.

4.2. Benchmarks for Zero-shot Object Classification

In this experiment, we use Top-K performance as a benchmark. It is a measure of the presence of correct outputs in the top K predicted categories. For each of Top-1, Top-2, Top-5, Top-10, and Top-20 criterion, we measure the average of the percentage of correct outputs for each category. Since we use ConceptNet for the predictive classifier, we do not use the benchmarks based on the number of hops in Wordnet [5]. Instead, we take the correspondence between ConceptNet and ImageNet 21k by their category names, and use the results for the 13,791 categories.

4.2.1 Applying ConceptNet for Zero-shot Framework

In the experiment to obtain a predictive classifier using ConceptNet, we compare a model that uses ConceptNet to train the GCN and WordNet to obtain the predictive classifier, which is similar to Nayak et al. [13], with a model that uses ConceptNet to both train the GCN and obtain the predictive classifier.

We also use ConceptNet to construct a model that extends the classifiable categories by obtaining predictive classifiers for additional categories that do not exist in WordNet. As additional categories, we use 588 categories from iNat2021 [18], which are detailed species in the natural world. We measure the performance of zero-shot object classification for each of the additional categories, the existing categories, and the whole, and compare it with the classifier without the additional categories.

4.2.2 Effectiveness of data extraction methods

We conduct experiments to investigate the change in performance of zero-shot object classification by applying the proposed method of node and edge extraction.

- Baseline: ConceptNet same as Nayak et al. [13]
- 2-Hop: Baseline knowledge graph with **2-Hop Ineffective Nodes** in Figure 2 removed.
- 2-Hop-Strong: Baseline knowledge graph with **2-Hop Ineffective Nodes** and **2-Hop-Strong Ineffective Nodes** in Figure 2 removed.

Method	Nodes	Edges
Baseline	559,928	1,380,131
2-Hop	517,322	1,322,606
2-Hop-Strong	213,558	843,182
Labeled	559,928	1,365,844
2-Hop-Labeled	517,322	1,308,490
2-Hop-S-Labeled	213,558	830,918

Table 3. The number of nodes and edges in each knowledge graph.

	Top-K				
Graph	1	2	5	10	20
WordNet	**2.04**	**3.71**	**7.41**	**11.56**	**17.35**
ConceptNet	1.41	2.55	4.98	8.15	12.59

Table 4. The performance for zero-shot object classification of the predictive classifier using ConceptNet compared with that of the predictive classifier using WordNet.

- Labeled: Baseline knowledge graph with edges negatively labeled removed.
- 2-Hop-Labeled: 2-Hop knowledge graph with edges negatively labeled removed.
- 2-Hop-S-Labeled: 2-Hop-Strong knowledge graph with edges negatively labeled removed.

We measure the performance of zero-shot object classification for models constructed using knowledge graphs with nodes and edges removed using the above six proposed methods in both in training GCN and obtaining predictive classifier, respectively. Details of each knowledge graph are shown in Table 3.

In addition, instead of removing nodes and edges, We constructed graphs by multiplying the priority(hit ratio) of ineffective nodes by 0.01, so that other nodes and edges are preferentially selected. We measured the performance of zero-shot object classification for each of these models: 2-Hop-N, 2-Hop-Strong-N, Labeled-N, 2-Hop-Labeled-N, 2-Hop-S-Labeled-N.

5. Results

5.1. Applying ConceptNet for Zero-shot Framework

Table 4 shows the performance of the ConceptNet-based predictive classifier is significantly lower for all top criterion, indicating that the number of edges and nodes that are useless noise for inference increases significantly as the knowledge graph becomes larger.

Table 5 indicates whether extending target categories is possible or not. Since the number of candidate categories increases as the number of categories is added, the performance decreases accordingly. However, the decrease in performance for the existing categories or all categories is smaller than the increase in the number of categories, so we can say that we have succeeded in extending the target of zero-shot object classification. The reason for the low performance for the additional categories can be attributed to the fact that the dataset used in this work targets detailed species in the natural world, and the presence of many similar species may have caused many patterns in which the answers could not be narrowed down, as **(C)** in Figure 3.

In any case, since it was confirmed that the use of large-scale graphs can extend the categories that can be recognized, it can be concluded that although there are obstacles in applying large-scale graphs to zero-shot object classification, the value of achieving this goal is great.

		Top-K				
Category	number	1	2	5	10	20
Additional	588	1.36	1.7	3.74	5.78	8.16
Default	13,791	1.4	2.5	**5.01**	8.09	12.38
All	14,379	1.4	2.46	4.95	7.99	12.21
Baseline	13,791	**1.41**	**2.55**	4.98	**8.15**	**12.59**

Table 5. The performance for zero-shot object classification of a model that extends the classifiable categories by obtaining additional predictive classifier for the additional, for each of the additional categories, the existing categories and the whole, comparing with the performance of a classifier without additional categories

	Top-K				
Method	1	2	5	10	20
Baseline	1.41	2.55	4.98	8.15	12.59
2-Hop	1.32	2.42	4.92	7.99	12.39
2-Hop-Strong	1.27	2.35	4.79	7.78	12.03
Labeled	**1.46**	2.61	**5.25**	**8.35**	**12.86**
2-Hop-Labeled	1.45	**2.65**	5.15	8.2	12.37
2-Hop-S-Labeled	1.27	2.32	4.75	7.64	11.9

Table 6. The performance of zero-shot object classification for the knowledge graphs to which the proposed method of node and edge removal is applied.

5.2. Effectiveness of data extraction methods

Table 6 and 7 show that both Labeled and Labeled-N outperform Baseline, indicating that edge extraction focusing on labels is effective. Since Labeled has a higher performance than Labeled-N, it can be said that the existence of edges with negative relationships itself has a negative effect on the performance of zero-shot object classification, and that removing edges is effective.

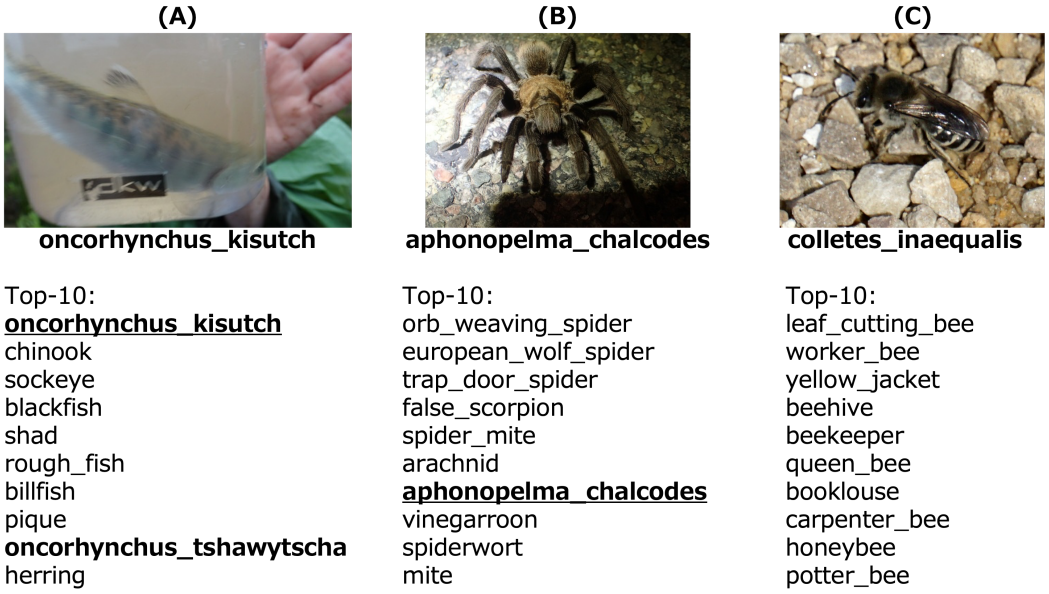

Figure 3. Example of results for zero-shot object classification of additional categories. Additional categories are shown in bold, and correct outputs in Top-10 are underlined.

	Top-K				
Method	1	2	5	10	20
Baseline	1.41	2.55	4.98	8.15	12.59
2-Hop-N	**1.48**	**2.67**	5.18	8.25	12.66
2-Hop-Strong-N	1.43	2.59	5.13	8.2	12.49
Labeled-N	1.41	2.59	5.18	8.29	12.73
2-Hop-Labeled-N	1.36	2.6	5.1	8.23	12.64
2-Hop-S-Labeled-N	1.44	2.66	**5.27**	**8.36**	**12.71**

Table 7. The performance for zero-shot object classification of the knowledge graphs to which the convolution priority is varied based on the proposed method of node and edge extraction.

However, in 2-Hop-Strong where many nodes are removed, 2-Hop-Strong-Labeled shows lower performance, while 2-Hop-Strong-Labeled-N, where edges are left with lower priority, shows better performance than 2-Hop-Strong-N. In other words, it is possible that some of the edges removed by this method contain effective information.

The 2-Hop and 2-Hop-Strong methods are below Baseline. In other words, the hypothesis that only the nodes within 2 hops of both the training and candidate categories contribute to learning for zero-shot object classification is incorrect, and it is highly likely that the entire structure of the knowledge graph is relevant for generalization in zero-shot object classification using the knowledge graph.

However, when reducing the convolution priority instead of removing data, node extraction methods such as 2-Hop-N outperform Baseline, confirming that the node extraction methods proposed in this work are useful for improving the performance of zero-shot object classification. This suggests that when performing sampling in GCN, there are many cases where the number of neighboring nodes after node removal or all neighboring nodes is less than the number of sampling nodes.

In the end, however, none of the proposed methods has achieved the same performance as the predictive classifiers using WordNet shown in Table 4, and it can be said that they have not sufficiently removed the noisy information.

6. Conclusion

In this work, we confirmed that the number of noisy edges and nodes in zero-shot object classification is likely to increase significantly as the knowledge graph becomes larger, and that the categories that can be recognized can be expanded by using a large-scale graph. We have also shown that removing edges from the knowledge graph by focusing on their labels and changing the convolution priority by selecting nodes in the knowledge graph based on the graph structure are effective in improving the performance of zero-shot object classification. However, there is the limitation of this research that the proposed method does not sufficiently remove the noisy information.

References

[1] Jiaoyan Chen, Yuxia Geng, Zhuo Chen, Ian Horrocks, Jeff Z Pan, and Huajun Chen. Knowledge-aware zero-shot learning: Survey and perspective. In *IJCAI*, pages 4366–4373, 2021. 1

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[3] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009. 2

[4] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, pages 21–37, 2018. 2

[5] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 2, 5

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[8] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR*, pages 11487–11496, 2019. 1, 2, 5

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2, 3

[10] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3):453–465, 2013. 2

[11] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *CVPR*, pages 4247–4255, 2015. 2

[12] George A Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995. 1, 2, 5

[13] Nihal V Nayak and Stephen H Bach. Zero-shot learning with common sense knowledge graphs. *TMLR*, 2020. 1, 2, 3, 5

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 2, 5

[15] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NeurIPS*, 2013. 2

[16] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017. 1, 2, 5

[17] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 1

[18] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *CVPR*, pages 12884–12893, 2021. 5

[19] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, pages 6857–6866, 2018. 1, 2

[20] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, pages 10275–10284, 2019. 2

[21] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 1

[22] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019. 3

[23] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*, pages 974–983, 2018. 3

[24] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 2021–2030, 2017. 2