Visual Objects As Context: Exploiting Visual Objects for Lexical Entailment

Masayasu Muraoka mmuraoka@jp.ibm.com
Tetsuya Nasukawa nasukawa@jp.ibm.com
Bishwaranjan Bhattacharjee bhatta@us.ibm.com

IBM Research

Abstract

We propose a new word representation method derived from visual objects in associated images to tackle the lexical entailment task. Although it has been shown that the *Distributional Informativeness Hypothesis* (DIH) holds on text, in which the DIH assumes that a context surrounding a hyponym is more informative than that of a hypernym, it has never been tested on visual objects. Since our perception is tightly associated with language, it is meaningful to explore whether the DIH holds on visual objects. To this end, we consider visual objects as the context of a word and represent a word as a bag of visual objects found in images associated with the word. This allows us to test the feasibility of the visual DIH. To better distinguish word pairs in a hypernym relation from other relations such as co-hypernyms, we also propose a new measurable function that takes into account both the difference in the generality of meaning and similarity of meaning between words. Our experimental results show that the DIH holds on visual objects and that the proposed method combined with the proposed function outperforms existing unsupervised representation methods.

1 Introduction

Recognizing lexical entailment (LE) is a fundamental component in Natural Language Processing (NLP), helping with many tasks such as textual entailment recognition (Garrette et al., 2011; Dagan et al., 2013), taxonomy creation (Snow et al., 2006; Navigli et al., 2011), and metaphor detection (Mohler et al., 2013). Lexical entailment defines an asymmetric relation between two terms, where one term can be inferred by the other, but not vice versa. For example, *dog* entails *animal* but not vice versa because *animal* does not always mean *dog*. To recognize LE, it is required (1) to construct a good representation that captures the generality of the meaning of a term, and (2) to define a measure to jointly calculate the asymmetric difference in the generality of meaning and the similarity of meaning between two given terms.

An increasing number of representation methods and measures to compute hypernymy have been proposed to date (Weeds and Weir, 2003; Clarke, 2009; Kotlerman et al., 2009; Lenci and Benotto, 2012). Especially, Santus et al. (2014) and Rimell (2014) proposed unsupervised methods that follow the *Distributional Informativeness Hypothesis* (DIH). However, they have not used visual information in their methods and instead, required a large amount of textual data to construct the representations. In the field of computer vision, Deselaers and Ferrari (2011) have shown that terms at higher levels in the hierarchy of WordNet (Miller, 1995) tend to correspond to a greater variety of images than terms at lower levels. Kiela et al. (2015a) have focused on this tendency and used a set of images obtained through image search to construct a word representation for the LE task, where image features were extracted from a Convolutional Neural Network (CNN) (Jia et al., 2014). However, no work has directly studied whether the DIH holds on visual objects.

To this end, we propose an unsupervised method to construct word representations for the LE task by using a set of images associated with each word. More specifically, we define a representation of a word as a bag of visual objects (labels) found in the associated images. Thus, our method allows us to directly evaluate the feasibility of the DIH on visual objects. Moreover, our method has two advantages over the previous methods. Firstly, unlike previous text-based approaches, our method does not require a huge amount of text corpora to construct representations. Secondly, due to the discrete nature of object labels, our representation

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2723–2735

is expected to be more discriminative than others constructed from a (middle layer of) CNN. It is a desirable property to distinguish the generality of the meaning of a word in the LE task. In addition to our representation method, we also propose a new function to jointly measure the difference in the generality of meaning and similarity of meaning between two terms to distinguish word pairs in a hypernym relation from others.

We evaluate our representation method and function on different types of LE datasets. We experimentally show that the combination of our representation and function outperforms the DIH-based method (Santus et al., 2014), word embeddings trained on a large text corpus, and the CNN-derived visual representation method (Kiela et al., 2015a), revealing that the DIH holds on visual objects. We also analyze the number of images as well as the number of unique objects to study how they affect the quality of our representations.

In summary, our contributions are three-fold:

- Propose a new unsupervised representation method that constructs representations from visual objects to solve the LE task,
- Propose a new function to distinguish word pairs in a hypernym relation from others, and
- Experimentally show that the DIH holds on visual objects.

2 Related Work

2.1 DIH-based representation methods

The idea of the *Distributional Informativeness Hypothesis* (DIH) was originally proposed by Santus et al. (2014) and Rimell (2014). On the basis of the hypothesis, Santus et al. (2014) measured the informativeness of a context with the median entropy of associated context words. Rimell (2014) used the ratio of change in topic coherence as a hypernymy measure. They experimentally showed that the DIH holds with such measures.

Other similar approaches have been proposed based on another hypothesis called the *Distributional Inclusion Hypothesis* (Geffet and Dagan, 2005) that states that contexts of a hyponym are expected to be the subset of contexts of a hypernym. Also, several asymmetric measures based on this hypothesis have been proposed (Weeds and Weir, 2003; Clarke, 2009; Kotlerman et al., 2009; Lenci and Benotto, 2012) so far, and each measure has focused on different linguistic aspects.

Recently, Shwartz et al. (2017) conducted an exhaustive study regarding measures for lexical entailment including not only the DIH-based methods but also *Distributional Hypothesis* [1]-based methods. While these investigated methods were constructed from text corpora, we construct our representations from visual objects to investigate whether the DIH holds on visual objects instead of text.

2.2 Visually-derived representation methods

A number of studies have shown the effectiveness of visual representations for different NLP tasks (Kiela and Bottou, 2014; Kiela et al., 2016, 2015b; Hartmann and Søgaard, 2018; Hewitt et al., 2018). As the most relevant work to ours, Kiela et al. (2015a) proposed a multi-modal representation method for the LE task. They represented a word as a combination of visual and textual features. They first collected a set of images associated with a word through image search. The visual feature was then extracted from the image set by taking the middle layer of a pre-trained CNN model (Jia et al., 2014), while the textual feature was obtained from text data. They showed that their method outperformed text-based representations.

While they also proposed new hypernymy measures, however, these measures did not directly test the feasibility of the DIH on visual objects because they were not based on the hypothesis. Therefore, to test it, we use the hypernymy measure that follows the DIH combined with our representations based on visual objects.

2.3 Supervised representation methods

The recent trend of learning efficient representations for lexical entailment has moved to supervised learning. In particular, pre-trained word embeddings are retrained to distinguish a hypernymy relation from other relations (Vulić and Mrkšić, 2018; Nguyen et al., 2017; Alsuhaibani et al., 2019). Hierarchical structures defined in taxonomies and ontologies (e.g., WordNet (Miller, 1995)) are commonly used for the retraining (Nguyen et al., 2017; Alsuhaibani et al., 2019). Also, several hypernymy measures and hypernym detection/directionality functions have been proposed and incorporated into

[1] The hypothesis is that words that share similar contexts tend to have similar meanings (Harris, 1954). The two DIHs were derived from this hypothesis.

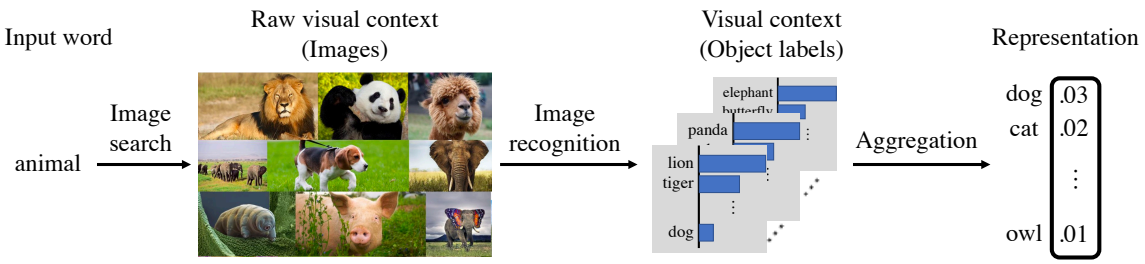

Figure 1: Overview of our representation construction.

(a) animal

(b) carnivore

(c) tiger

Figure 2: Images returned by Google Image Search.

the loss functions. However, these measures and functions no longer follow the principle of the DIH. We apply DIH-based measures and functions to our representation method.

3 Methodology

We construct our representations from visual objects. We illustrate an overview of our representation construction in Figure 1. Based on the representations, we introduce hypernymy measures to measure the generality of word meanings. We then explain how the LE task is solved.

3.1 Object-based representation

We follow the procedure described in the work by Kiela et al. (2016). We represent a word w as a vector $\boldsymbol{w} \in \mathbb{R}^D$, where D is a dimensionality of a vector. We construct a vector from a set of images associated with the word. We extract a feature that includes object labels from an image. The vector $\boldsymbol{w}$ is constructed by aggregating a matrix $\boldsymbol{W} \in \mathbb{R}^{D \times L}$ by using an aggregation function g (See Section 4.1.2), in which each column in $\boldsymbol{W}$ corresponds to a feature extracted from an image. We describe how to construct our representation step by step in the following.

We first collect images relevant to a word as a (visual) context. We use image search as our image source to collect the L most relevant images $V = \{v_i | i = 1 \ldots L\}$ for a word. An image search returns images for a textual query based on the relevancy. Kiela et al. (2016) and Kastner et al. (2019) have shown that publicly-available image searches such as Google or Bing Image Search can return images so that the images associated with a more general word have a greater visual variability than a more specific word. Figure 2 shows example images retrieved by queries *animal*, *carnivore*, and *tiger* through Google Image Search[2]. We can see that the variability of visual objects actually decreases as we see narrower concepts, as in *carnivore* or *tiger*.

Next, we extract visual object labels as a discrete feature by using image recognition. We can use any recognizers that generate a list of object labels with confidence scores such as CNNs or image recognition systems provided by vendors (e.g., Google Cloud Vision[3]). We represent a feature extracted from the i-th image in V as an n-hot vector $\boldsymbol{v}_i \in \mathbb{R}^D$, in which each dimension represents a visual object, and confidence scores obtained by a recognizer are stored in the corresponding dimensions. By concatenating L vectors, we obtain $\boldsymbol{W} \in \mathbb{R}^{D \times L}$:

$$\boldsymbol{W} = [\boldsymbol{v}_1; \ldots \boldsymbol{v}_L], \quad (1)$$

where $[;]$ denotes a concatenation of vectors. A representation for a word is obtained by a row-wise aggregation function g: $\boldsymbol{w} = g(\boldsymbol{W})$.

Since current image recognizers can achieve comparable accuracy with humans (He et al., 2016), we can expect to obtain reasonably accurate labels. The main reason for using object labels is

[2] https://images.google.com/
[3] https://cloud.google.com/vision

because we consider that object labels are more discriminative than continuous, more abstract features brought from the middle layers of neural networks. For example, when we have two images that show a dog and cat, respectively, the continuous features are likely close to each other, while the discrete features represented by the object labels are treated differently from one another. Still, the similarity of the discrete features between dog and cat could be higher than one between more dissimilar concepts such as dog and table. This can be explained as follows. An image recognizer often generates more general object labels (e.g., carnivore or animal) in addition to specific labels such as dog and cat to the objects shown in the dog and cat images. The recognizer also generates labels for co-occurring objects (e.g., grass or tree) because similar concepts tend to share these labels in their discrete features while dissimilar concepts do not. This results in a moderately higher similarity between similar concepts.

3.2 Hypernymy measures

We use a measure to quantify the extent of the hypernymy of a word w and call it the *hypernymy measure*. To validate whether the DIH holds on visual objects, we adopt a measure based on the informativeness of the contexts of a word. The measure was originally introduced by Santus et al. (2014). It has been obtained by the median entropy of the n most associated contexts of the word, and the association strength has been calculated with *Local Mutual Information* (LMI) (Evert, 2005). However, because the original measure highly depends on the amount of a textual corpus used[4], we use a modified version proposed by Shwartz et al. (2017):

$$E(\boldsymbol{w}) = -\Sigma_{i=1}^{n} p(w_i) \log_2 p(w_i). \qquad (2)$$

We obtain $p(w_i)$ with $\frac{w_i}{||\boldsymbol{w}||}$, where w_i indicates the i-th element of $\boldsymbol{w}$ and $||\boldsymbol{w}||$ is the vector length. We consider only the positive values in $\boldsymbol{w}$ in the computation. We call this measure *entropy* (ent).

From the definition, the entropy increases as the vector $\boldsymbol{w}$ forms closer to a uniform distribution, which means that different labels uniformly appear in an image set V for a word. We can see this tendency in Figure 2. Consequently, a broader word is likely less informative (i.e., higher entropy).

[4]Particularly, the calculation of association strength requires the total number of occurrences of words in the corpus.

3.3 Hypernym detection

3.3.1 Detection of hypernym

Based on hypernymy measures, we measure the difference in the generality of meaning between two words. Santus et al. (2014) used the ratio of the informativeness of a word x to the other y:

$$\text{diff}(x, y) = 1 - \frac{E(\boldsymbol{w}_x)}{E(\boldsymbol{w}_y)}, \qquad (3)$$

in which $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ are representations of x and y, respectively. The above function returns a positive value if y is a hypernym of x.

3.3.2 Detection of hypernym relation

In addition to detecting hypernyms, we have to detect pairs in hypernym-hyponym relations from other relations. Similarity functions such as cosine similarity or Jensen-Shannon (JS) divergence have been used to distinguish the pairs from others to date. However, such functions cannot distinguish well hypernym relations from certain relations, such as co-hyponyms[5]. Therefore, we propose a new function to distinguish pairs in hypernym relations from others:

$$\text{hrel}(x, y) = \text{sim}(x, y) \cdot \text{diff}(x, y), \qquad (4)$$

where $\text{sim}(x, y)$ measures the similarity of the meaning between two words. We can use cosine similarity and JS divergence as $\text{sim}(x, y)$. The proposed function $\text{hrel}(x, y)$ has a larger value if and only if two words are in a hypernym relation (i.e., similar in meaning but dissimilar in the generality of meaning) and conversely, a smaller value if and only if two words are in a reversed hypernym relation, in which x should be a hypernym of y. For this generalized function, we can use any combination of $\text{sim}(x, y)$ and $\text{diff}(x, y)$ unless the value of $\text{sim}(x, y)$ becomes larger when the two words are closer in meaning, and the value of $\text{diff}(x, y)$ becomes larger when the two words are different in their generalities of meaning. When we detect word pairs in both hypernym and reversed hypernym relations, we take the absolute value of $\text{diff}(x, y)$: $\text{sim}(x, y)|\text{diff}(x, y)|$. In our experiment (Section 4.1), we tested as $\text{hrel}(x, y)$ cosine similarity (cos), JS divergence (JS), cos · diff, JS · diff, cos|diff|, and JS|diff|.

[5]The co-hypernym relation is defined for word pairs where both words have the same hypernym, such as (dog, cat) and (bike, car).

3.3.3 Classification

We introduce two thresholds, α_{rel} and α_{hyp}, to detect word pairs in a hypernym relation and hypernyms in the detected word pairs. We regard a word pair (x, y) such that hrel$(x, y) \geq \alpha_{rel}$ is in a hypernym relation. Likewise, we consider a word y in a word pair (x, y) in hypernym relation such that diff$(x, y) \geq \alpha_{hyp}$ is a hypernym of x. Otherwise, x is marked as a hypernym of y. We explain how to optimize these thresholds in Section 4.1.2.

4 Experiment

We conducted lexical entailment experiments by using different types of datasets to evaluate the capability of our object-based representation method.

4.1 Classification task

4.1.1 Task overview

We first evaluated our method on three different tasks with three datasets that measure different aspects of lexical entailment. We used the datasets compiled by Kiela et al. (2015a). The datasets consisted of animate and inanimate concepts in English (e.g., animals, plants, and vehicles).

The first task is referred to as the **directionality** task, which is a binary classification task. Given two words (x, y), the goal of this task is to predict the hypernym that entails the other. We used the BLESS dataset to evaluate this task. The dataset consisted of 1,337 word pairs that were all in a hypernym relation, as in (*tiger*, *animal*) and (*tiger*, *carnivore*), where the latter was always a hypernym of the former. Our method had to assign positive scores based on Equation (3), which means the former word is more informative, i.e., a hyponym.

The second task was the **detection** task. This is also a binary classification. In this task, our method aimed to distinguish word pairs in hypernym relations from the others, namely, holonymy-meronymy (*tiger*, *jaw*), co-hyponymy (*tiger*, *bull*), reversed hypernym (*vertebrate*, *tiger*), or no relation (*tiger*, *maneuver*). The corresponding dataset was WBLESS, which included 1,668 word pairs.

The third one was a combination of **directionality** and **detection** tasks. Our method had to detect hypernym-hyponym pairs from the others and then predict the hypernym in the detected pairs. We used the BIBLESS dataset that had the same word pairs as WBLESS, but word pairs in a reversed hypernym relation were marked as another category. Thus, this is a three-class classification task.

We used the two thresholds introduced in Section 3.3.3 when evaluating on the WBLESS and BIBLESS datasets. Following Vulić and Mrkšić (2018) and Nguyen et al. (2017), we tuned the thresholds with 2% randomly chosen from the datasets and evaluated our method on the remaining 98%. We repeated this procedure 1,000 times and report the average accuracy.

4.1.2 Experimental setup

Because our method consists of multiple elements as shown in Figure 1, we investigated several options for each element. This contributes to excluding the possibility that our method outperforms methods for comparison described below by using parameters favorable for our method by chance. See Appendix A for a detailed description of the other experimental setup.

Image search engines and image sources. These engines and sources probably make a significant impact on the representation quality. We considered two image engines and two image sources.

imgsrc: {Google Image Search, Bing Image Search[6], ImageNet, Flickr[7]}.

Both Google and Bing Image Search return images relevant to a query word from the Web. ImageNet (Russakovsky et al., 2015) is a hierarchical image database whose structure is brought from WordNet. Flickr is an image hosting service that accommodates tens of thousands of photos. With each image search and source, we collected $L = 50$ images for each word.

Image recognition models. We used publicly-available CNN models pre-trained on a 1k-class image recognition dataset (Russakovsky et al., 2015). We tested three models: AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2015), and DenseNet (Huang et al., 2017). In practice, we used the pre-trained CNN models provided by the torchvision package[8].

Recently, some vendors have been providing their own image recognition systems, which can recognize more than 1k classes. These systems predict a list of object labels with confidence probabilities for an image. We can utilize the output of such systems to construct our representations. In this work, we examined two image recogni-

[6] https://www.bing.com/
[7] https://www.flickr.com/
[8] https://pytorch.org/docs/stable/torchvision/index.html

Representation method	Accuracy	Optimal setting	
		model, imgsrc if any, hyp_func	agg if any, norm
Text-based DIH	60.51	plmi, ent	L2/min-max
Word embedding	71.35	SGNS, ent	min-max
Visual representation	90.95	DenseNet, Google, cos-all	all aggs, zscore
Object-based DIH (Ours)	**94.39**	WVR, Bing, ent	avg, all norms

Table 1: Results on directionality task (BLESSE dataset). "model" denotes model names or values specific to each method, and "hyp_func" represents hypernymy measure. See Sections 3.3.2 and 4.1.2 for other notations.

Representation method	Accuracy	Optimal setting	
		model, imgsrc if any, hyp_func, hrel	agg if any, norm
Text-based DIH	55.35	ppmi, JS·diff	L2
Word embedding	54.09	fastText, cos	zscore
Visual representation	76.11	DenseNet, Google/Bing, cos-all, cos·diff	avg, L2
Object-based DIH (Ours)	**79.73**	WVR, Bing, ent, JS·diff	avg, zscore

Table 2: Results on detection task (WBLESS dataset). For notations in Optimal setting, see caption for Table 1 and Sections 3.3.2 and 4.1.2.

tion systems: IBM Watson Visual Recognition[9] (WVR) and Google Cloud Vision[10] (GCV). We found WVR and GCV could predict more than 13k and 8k unique objects, respectively.

imgreco: {AlexNet, VGGNet, DenseNet, WVR, GCV}.

Aggregation functions. We considered three aggregation functions as g described in Section 3.1: **agg**: {avg, max-pool, mean-std}.

Average (avg) aggregation calculated the row-wise average in $\boldsymbol{W}$. Max-pooling (max-pool) took the maximum value in each dimension in $\boldsymbol{W}$. Mean and standard deviation (mean-std) aggregation computed the mean and standard deviation for each row and then concatenated them; thus, the resulting vector was double in size.

Normalizations. We assumed that different representation methods would prefer different normalization methods. Kiela et al. (2015a) adopted L2 normalization, while Santus et al. (2014) used min-max normalization. We thus analyzed which normalization methods best matched our method among three:

norm: {L2, mim-max, zscore},

which respectively indicate L2 norm, min-max (Priddy and Keller, 2005), and z-score normalization (Jayalakshmi and Santhakumaran, 2011).

[9] https://www.ibm.com/cloud/watson-visual-recognition

[10] https://cloud.google.com/vision

4.1.3 Methods for comparison

We compared three unsupervised representation methods with our method.

Text-based DIH. We constructed text-based DIH representations (Santus et al., 2014) from the Reuters corpus (RCV1)[11] (Lewis et al., 2004), which included 806,791 English documents. We applied spaCy[12] (Honnibal and Montani, 2017) to the Reuters corpus for tokenization and PoS tagging. We obtained 90,043,588 tokens as a result. To construct the representations, we used the scripts provided by Shwartz et al. (2017)[13], where we set the minimum frequency to 100 and the context window size to 5. As for the values in a representation, we tested the raw frequency (freq), positive local mutual information (plmi), and positive pointwise mutual information (ppmi). Each representation formed 4,346-dimensional vectors.

Word embeddings. We also investigated three well-known word embeddings that were all pre-trained with a large amount of textual corpora: skip-gram with negative sampling (SGNS)[14] (Mikolov et al., 2013), GloVe[15] (Pennington et al., 2014),

[11] https://trec.nist.gov/data/reuters/reuters.html

[12] https://spacy.io/

[13] https://github.com/vered1986/UnsupervisedHypernymy

[14] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

[15] http://nlp.stanford.edu/data/glove.840B.300d.zip

Representation method	Accuracy	Optimal setting	
		model, imgsrc if any, hyp_func, hrel	agg if any, norm
Text-based DIH	49.25	plmi, ent, JS·diff	L2
Word embedding	51.32	fastText, ent, cos	min-max
Visual representation	63.05	DenseNet, Google, cos-all, JS	avg, L2
Object-based DIH (Ours)	**63.35**	WVR, Bing, ent, JS|diff|	avg, zscore

Table 3: Results on detection and directionality task (BIBLESS dataset). For notations in Optimal setting, see caption for Table 1 and Sections 3.3.2 and 4.1.2.

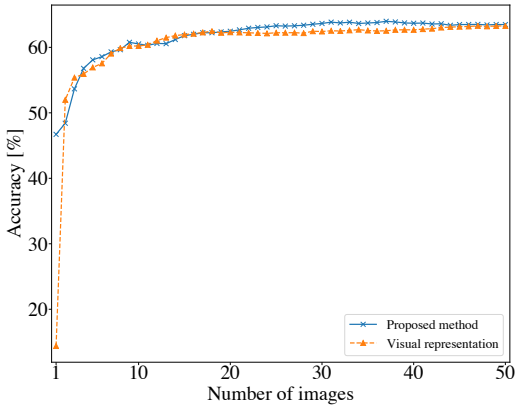

Figure 3: Effect of number of images.

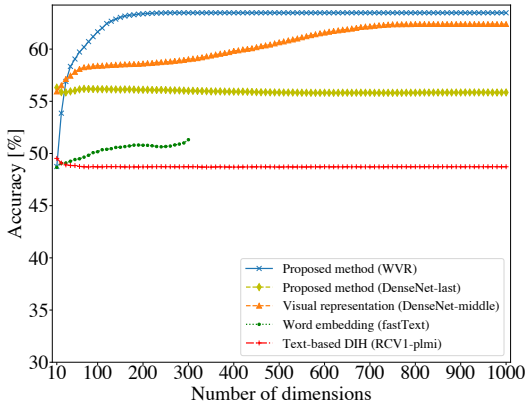

Figure 4: Effect of number of dimensions.

and fastText[16] (Bojanowski et al., 2017). The dimensionality of all the embeddings was 300.

CNN-based visual representations. We constructed visual representations proposed by Kiela et al. (2015a). We investigated the same options described in Section 4.1.2 except for the image recognition models, where we used image features $\boldsymbol{v}$ extracted from the middle layer of the CNNs. Specifically, we extracted the final fully-connected activation layers in the CNNs as image features. The resulting representations formed 4,096-dimensional vectors for AlexNet and VGGNet, and 2,208 for DenseNet. Since we cannot obtain the intermediate features from WVR and GCV, we omitted them.

We also tested two hypernymy measures used in their work (Kiela et al., 2015a): cos-all and cos-cen. The measure cos-all calculates an average cosine distance between *all* pairs of visual representations in $\boldsymbol{W}$ while cos-cen computes an average cosine distance of visual representations to the *centroid* $\mu = \frac{1}{L}\Sigma_i^L \boldsymbol{v}_i$.

4.1.4 Results

We present our experimental results in Tables 1 through 3. We report the best accuracy that each method achieved with the optimal setting. Any of the methods could have reached the best accuracy in multiple combinations in their settings. For example, in Table 1, Text-based DIH achieved an accuracy of 60.51% with the configuration that used plmi as the value in the representation, entropy (ent) for the hypernymy measure, and L2 norm and min-max normalization.

Our method outperformed all the methods for comparison in all the tasks. This indicates that representations based on visual objects are useful for the LE task and implies that the DIH holds on visual objects. Also, we can see that our proposed function (Equation (4)) worked well with not only our method but also other methods. In particular, multiplication of $\text{sim}(x, y)$ and $\text{diff}(x, y)$ was appropriate in the detection task (Table 2) because it was required to distinguish hypernym-hyponym pairs from the other relations. In addition, since only the degree of the multiplied value was important to discriminate pairs in hypernym and reversed hypernym relations from the others in the directionality and detection task (Table 3), $\text{sim}(x, y)|\text{diff}(x, y)|$ was effective. The main reason that the proposed method outperformed the visual representation method is probably because the discrete nature is more suitable for the LE task.

[16] https://fasttext.cc/docs/en/crawl-vectors.html

The disappointing results for Text-based DIH were possibly caused by noise in the corpus. Since the representations were very sparse, they were sensitive to the noise. Word embeddings, which have been shown to be more robust to noise than Text-based DIH, slightly outperformed Text-based DIH in Tables 1 and 3. However, they were not comparable with visually-derived representations because these embeddings were not trained specially for the LE task. Since the visual representation method was constructed based on the generality of images reflected in the image search result, it reasonably solved the tasks.

As for the optimal settings, each method had its own preference. For example, our method consistently best matched WVR as the image recognition model and Bing Image Search as the image source. Particularly, the performances of Bing and Google Image Search with the visual representations and our method were consistently better than those of ImageNet and Flickr even if we fixed the other hyperparameters. We suppose that this was caused by the coverage of the words because ImageNet and Flickr could not return any images for some words, and thus, the corresponding representations became zero vectors.

4.1.5 Analysis

We inspected two factors that may potentially affect the performance of our method: the number of images and the context size, i.e., the dimensions in a representation. In the analyses below, we used the BIBLESS dataset and the representations with the optimal settings achieved in Table 3.

Number of images. We assumed that more images could yield better performance. To confirm this, we conducted an experiment changing the number of images used in the representations. Figure 3 compares the results between our object-based representations and the CNN-derived representations. We found that both representations tended to be saturated with a relatively smaller number of images (i.e., around 10-20) in contrast to our expectations. This indicates that around 20 images are enough to construct our representation of high quality. These results are consistent with those reported by Kiela et al. (2016), though they tested this in another task.

Number of dimensions. Next, we investigated how many different objects, i.e., dimensions, we should take into account for obtaining our representations with the optimal performance. This investigation directly tests whether our method's ability to outperform others relies on the number of objects that an image recognizer can recognize. To this end, we made a comparison experiment where we restricted the number of dimensions of the representations when calculating the entropy (Equation (2)). In addition to the optimal setting, we also included our representations constructed from the last layer of the CNNs to compare the effectiveness of the number of object labels that the typical CNNs can predict.

Method	All	Nouns	Verbs
Text-based DIH	0.176	0.195	0.026
Word embedding	0.180	0.196	0.043
Visual representation	0.250	0.274	0.085
Object-based DIH	**0.266**	**0.289**	**0.107**

Table 4: Spearman's ρ on *HyperLex* dataset.

Figure 4 illustrates that our representation (WVR) achieved the best performance with 250 dimensions, which is much smaller than the visual representations derived from the middle layers of the CNNs (DenseNet-middle). This reveals that our representation had a strong capability in the LE task even if the number of unique objects was small. Moreover, the difference in performance between WVR and DenseNet-last in our method implies that a larger number of unique objects that an image recognizer can predict would lead to further improvement.

4.2 Graded lexical entailment task

4.2.1 Task overview

For a more fine-grained evaluation, we conducted another experiment for the LE task on *HyperLex* dataset (Vulić et al., 2017). It measures the correlation between scores by a method and ones rated by humans. The dataset is composed of 2,616 word pairs, which also contains verb pairs (453 out of 2,616) unlike the previous datasets. Seven different relations are defined in it: synonym, antonym, meronym-holonym, co-hypernym, hypernym, reversed hypernym, and no relation. The scores rated by humans range from 0 to 10, which indicate "to what degree is the former word a type of the latter word." A higher score is assigned to a word pair in a hypernym relation (e.g., 9.85 for *girl - person*).

Method	1	2	3	4 $\leq$
Text-based DIH	53.89	47.28	57.96	53.05
Word embedding	20.00	24.83	29.30	24.05
Visual representation	55.83	**68.71**	64.65	77.10
Object-based DIH	**58.61**	63.61	**70.70**	**77.48**

Table 5: Accuracy by WordNet shortest path.

4.2.2 Results

We calculated hypernymy measures for each method based on the best configurations obtained in Table 3. Using these measures, we then computed Spearman's ρ with the human-rated scores in *HyperLex*. We report our results in Table 4.

Similar to the previous evaluation, our method outperformed all the comparison methods in all combined datasets (All). It is notable that our method further improved on the Verb portion.

4.2.3 Analysis

To take a close look at our results, we conducted a quantitative analysis of hypernymy measures and the level of hypernymy. We assumed that our method (and other methods) could distinguish hypernyms from hyponyms more easily as two words become conceptually dissimilar. To examine this assumption, we first classified hypernym and reversed hypernym pairs in all combined datasets in terms of lengths of the shortest path between two words in WordNet. We then calculated accuracy based on Equation (3).

We show the results in Table 5. As expected, the accuracies tended to increase with larger path lengths. This shows that it is easier to measure the difference in the generality of meaning between more dissimilar concepts. This tendency is consistent with results reported by Kiela et al. (2015a) and Vulić et al. (2017). In addition, our method outperformed other methods all but one category (i.e., the path length is 2).

5 Conclusion

We proposed a new word representation method based on discrete visual objects in images associated with each word for the LE task. Our method outperformed both traditional unsupervised CNN-based representations and text-based DIH representations on different types of lexical entailment datasets. We also experimentally confirmed that the *Distributional Informativeness Hypothesis* holds on visual objects. In addition, we revealed that our method got rapidly saturated at around 10-20 images and 200 dimensions (i.e., the context size). This suggests that our representations can achieve sufficient informativeness even with a smaller number of images and contexts. One of our future research directions is to examine the capability of our object-based representations in other tasks, such as lexical induction or word similarity tasks.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful comments. We also thank Issei Yoshida for the constructive comments that greatly improved our manuscript. We appreciate a number of helpful discussions with our colleagues. IBM Watson, Google, and Bing are registered trademarks of International Business Machines Corporation, Google LLC, and Microsoft Corporation, respectively, in the United States, other countries, or both.

References

Mohammed Alsuhaibani, Takanori Maehara, and Danushka Bollegala. 2019. Joint learning of hierarchical word embeddings from a corpus and a taxonomy. In *Automated Knowledge Base Construction (AKBC)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, Athens, Greece. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–222.

Thomas Deselaers and Vittorio Ferrari. 2011. Visual and semantic similarity in imagenet. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, page 1777–1784, USA. IEEE Computer Society.

Stefan Evert. 2005. *The Statistics of Word Cooccurrences: Word Pairs and Collocations.* Ph.D. thesis, Stuttgart University.

Dan Garrette, Katrin Erk, and Raymond Mooney. 2011. Integrating logical representations with probabilistic information using Markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Zellig S. Harris. 1954. Distributional structure. *WORD*, pages 146–162.

Mareike Hartmann and Anders Søgaard. 2018. Limitations of cross-lingual learning from image search. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 159–163, Melbourne, Australia. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

John Hewitt, Daphne Ippolito, Brendan Callahan, Reno Kriz, Derry Tanti Wijaya, and Chris Callison-Burch. 2018. Learning translations via images with a massively multilingual image dataset. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2566–2576, Melbourne, Australia. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

T Jayalakshmi and A Santhakumaran. 2011. Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3:89–93.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678.

Marc A Kastner, Ichiro Ide, Yasutomo Kawanishi, Takatsugu Hirayama, Daisuke Deguchi, and Hiroshi Murase. 2019. Estimating the visual variety of concepts by referring to web popularity. *Multimedia Tools and Applications*, 78(7):9463–9488.

Douwe Kiela. 2016. MMFeat: A toolkit for extracting multi-modal features. In *Proceedings of ACL-2016 System Demonstrations*, pages 55–60, Berlin, Germany. Association for Computational Linguistics.

Douwe Kiela and Léon Bottou. 2014. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45, Doha, Qatar. Association for Computational Linguistics.

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015a. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 119–124, Beijing, China. Association for Computational Linguistics.

Douwe Kiela, Anita Lilla Verő, and Stephen Clark. 2016. Comparing data sources and architectures for deep visual representation learning in semantics. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 447–456, Austin, Texas. Association for Computational Linguistics.

Douwe Kiela, Ivan Vulić, and Stephen Clark. 2015b. Visual bilingual lexicon induction with transferred ConvNet features. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 148–158, Lisbon, Portugal. Association for Computational Linguistics.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2009. Directional distributional similarity for lexical expansion. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 69–72, Suntec, Singapore. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. Semantic signatures for example-based linguistic metaphor detection. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 27–35, Atlanta, Georgia. Association for Computational Linguistics.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1872–1877.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Kevin L Priddy and Paul E Keller. 2005. *Artificial Neural Networks: An Introduction (SPIE Tutorial Texts in Optical Engineering, Vol. TT68)*. SPIE press.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 511–519, Gothenburg, Sweden. Association for Computational Linguistics.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International journal of computer vision*, 115(3):211–252.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden. Association for Computational Linguistics.

Vered Shwartz, Enrico Santus, and Dominik Schlechtweg. 2017. Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1134–1145, New Orleans, Louisiana. Association for Computational Linguistics.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

A Appendix

Here, we describe the detailed settings in our experiment (Section 4.1).

A.1 Data collection

We collected all the images from July 10th to November 15th, 2018 using the MMFEAT toolkit (Kiela, 2016)[17]. At the same time, we also collected image features from the CNNs as well as visual object labels from IBM Watson Visual Recognition and Google Cloud Vision. As for the CNNs, especially VGGNet and DenseNet, we used 19-layers VGGNet with batch normalization and 161-layers DenseNet, respectively, among the same network architectures according to the lowest error rates on the ImageNet image classification task[18]. For a fair comparison between our representation method and Kiela's visual representation method, we used the exact same image sets for all the words contained in the datasets when constructing both representations. Since all the methods tested in our experiments are fully unsupervised, we do not have either training or validation data.

A.2 Implementation of methods and experimental environment

We used Python to implement our representation method as well as methods for comparison. We conducted a series of our experiments on a server running with twelve processors (6 cores, 3.33 GHz, Intel Xeon W3680) and 24 GB main memory. We computed the accuracy scores by using the scikit-learn library (Pedregosa et al., 2011)[19].

A.3 Hyperparameters and runtimes

Tables 5 through 8 show the hyperparameter search space and average runtime for each method. We used **grid search** to test all possible combinations across the hyperparameters and find the best accuracy for each method. The best assignments of hyperparameters for each method are reported in Tables 1 to 3.

[17] https://github.com/douwekiela/mmfeat

[18] https://pytorch.org/docs/stable/torchvision/models.html

[19] https://scikit-learn.org/stable/index.html

Number of search trials	540 (BLESS), 3,240 (WBLESS, BIBLESS)*
Hyperparameter	**Search space**
image source (imgsrc)	{Google, Bing, ImageNet, Flickr}
number of images L	50
image recognizer (imgreco)	{AlexNet, VGGNet, DenseNet, WVR, GCV}
hypernym measure (hyp_func)	{ent, cos-all, cos-cen}
hrel(x, y)	{cos, JS, cos · diff, JS · diff, cos|diff|, JS|diff|}
aggregation (agg)	{avg, max-pool, mean-std}
normalization (norm)	{L2, min-max, zscore}
Evaluation runtime	4.4 minutes

Table 6: Hyperparameter search space and average runtime for our Object-based DIH method.

Number of search trials	324 (BLESS), 1,944 (WBLESS, BIBLESS)*
Hyperparameter	**Search space**
image source (imgsrc)	{Google, Bing, ImageNet, Flickr}
number of images L	50
image recognizer (imgreco)	{AlexNet, VGGNet, DenseNet}
hypernym measure (hyp_func)	{ent, cos-all, cos-cen}
hrel(x, y)	{cos, JS, cos · diff, JS · diff, cos|diff|, JS|diff|}
aggregation (agg)	{avg, max-pool, mean-std}
normalization (norm)	{L2, min-max, zscore}
Evaluation runtime	2.8 minutes

Table 7: Hyperparameter search space and average runtime for Kiela's visual representation.

Number of search trials	9 (BLESS), 54 (WBLESS, BIBLESS)*
Hyperparameter	**Search space**
model	{SGNS, Glove, fastText}
hypernym measure (hyp_func)	ent
hrel(x, y)	{cos, JS, cos · diff, JS · diff, cos|diff|, JS|diff|}
normalization (norm)	{L2, min-max, zscore}
Evaluation runtime	0.9 minutes

Table 8: Hyperparameter search space and average runtime for word embeddings.

Number of search trials	9 (BLESS), 54 (WBLESS, BIBLESS)*
Hyperparameter	**Search space**
value	{freq, plmi, ppmi}
hypernym measure (hyp_func)	ent
hrel(x, y)	{cos, JS, cos · diff, JS · diff, cos|diff|, JS|diff|}
normalization (norm)	{L2, min-max, zscore}
Evaluation runtime	2.8 minutes

Table 9: Hyperparameter search space and average runtime for text-based DIH method.

*Note that the number of search trials differs among the datasets because hrel(x, y), the function for detecting word pairs in a hypernym relation, is applied only to the WBLESS and BIBLESS datasets.