Computer Vision and Image Understanding 215 (2022) 103333

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

The semantic typology of visually grounded paraphrases

Chenhui Chu [a,*], Vinicius Oliveira [b], Felix Giovanni Virgo [a], Mayu Otani [c], Noa Garcia [d], Yuta Nakashima [d]

[a] Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
[b] Ecole Polytechnique, Route de Saclay, 91120 Palaiseau, France
[c] CyberAgent, Inc., 40-1 Abema Towers, Udagawacho, Shibuya-ku, Tokyo, 150-0042, Japan
[d] Osaka University, 1-1 Yamadaoka, Suita, Osaka, 565-0871, Japan

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
68T45
68T50
68T01

Keywords:
Vision and language
Image interpretation
Visual grounded paraphrases
Semantic typology
Dataset

ABSTRACT

Visually grounded paraphrases (VGPs) are different phrasal expressions describing the same visual concept in an image. Previous studies treat VGP identification as a binary classification task, which ignores various phenomena behind VGPs (i.e., different linguistic interpretation of the same visual concept) such as linguistic paraphrases and VGPs from different aspects. In this paper, we propose semantic typology for VGPs, aiming to elucidate the VGP phenomena and deepen the understanding about how human beings interpret vision with language. We construct a large VGP dataset that annotates the class to which each VGP pair belongs according to our typology. In addition, we present a classification model that fuses language and visual features for VGP classification on our dataset. Experiments indicate that joint language and vision representation learning is important for VGP classification. We further demonstrate that our VGP typology can boost the performance of visually grounded textual entailment.

1. Introduction

A linguistic paraphrase is a restatement of the meaning of a word, phrase, or sentence within the context of a specific language (e.g., "a little girl" and "a small girl" in Fig. 1 are paraphrases) (Bhagat and Hovy, 2013). Linguistic paraphrases have been exploited for natural language understanding, and shown to be very effective for various natural language processing (NLP) tasks, including question answering (Riezler et al., 2007), summarization (Zhou et al., 2006), machine translation (Chu and Kurohashi, 2016), text normalization (Ling et al., 2013), textual entailment recognition (Androutsopoulos and Malakasiotis, 2010), and semantic parsing (Berant and Liang, 2014).

In contrast to linguistic paraphrases, visually grounded paraphrases (VGPs) describe the same visual concept with different phrasal expressions (Chu et al., 2018). For instance, all the phrase pairs in Fig. 1 are VGPs. VGPs can generally include much broader variations of expressions compared to linguistic paraphrases by its definition. Given a concrete concept in an image, any expressions that can identify the same concept can be VGPs, while linguistic paraphrases limit to expressions with the same linguistic meaning. Identifying VGPs has the potential to improve vision and language tasks such as visual question answering (Wu et al., 2017; Samaran et al., 2021) and image captioning (Vinyals et al., 2015) in a way similar to that paraphrases have been applied to natural language processing tasks such as question answering (Riezler et al., 2007) and machine translation (Chu and Kurohashi, 2016).

Previous studies formulate VGP identification as a binary classification task, which classifies a phrase[1] pair into VGPs or non-VGPs (Chu et al., 2018; Otani et al., 2019, 2020). This formulation, however, ignores various phenomena behind VGPs due to the neglect of the fact that we human beings interpret the same visual concept in different ways. For instance, Fig. 1(b) "Chevrolet" and "chevy" are linguistic paraphrases; Fig. 1(i) "competitors" and "a group of bicyclist" describe the same visual concept from different aspects. A VGP pair, by definition, always refers to something in a given image and thus corresponds to a certain concrete concept; however, different expressions may, e.g., explain some different information or emphasize certain aspects that the concept has, as we can see in the above examples. Treating such VGPs equally may spoil semantics enriched by different expressions.

In this paper, we propose semantic typology of VGPs inspired by natural language inference (Maccartney, 2009). We define five classes

* Corresponding author.
E-mail address: chu@i.kyoto-u.ac.jp (C. Chu).

[1] A phrase in the paper is limited to a noun phrase (entity) that corresponds to an object in an image, which is the same phrase definition as in the Flickr30k entities dataset (Plummer et al., 2015).

[2] The dataset is available at: https://github.com/felixgiov/VGP-Typology.

https://doi.org/10.1016/j.cviu.2021.103333
Received 8 January 2021; Received in revised form 12 August 2021; Accepted 30 November 2021
Available online 11 December 2021

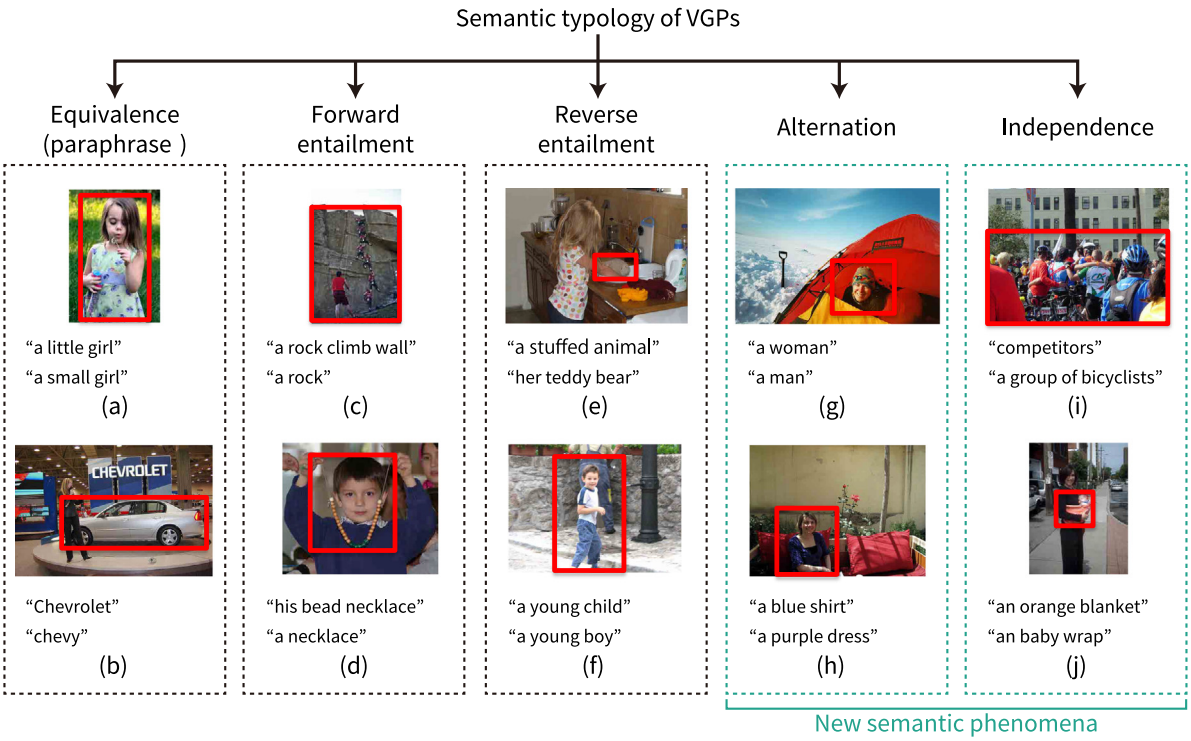

Fig. 1. Semantic VGP typology. We categorize VGPs into 5 semantic relations. *Equivalence* is linguistic paraphrase. In addition, we introduce *forward and reverse entailment*, *alternation* and *independence* as relations for VGPs, which are not linguistic paraphrases but describing the same visual concepts.

of VGPs: *equivalence, forward entailment, reverse entailment, alternation, and independence*. Detailed analyses on the VGP dataset (Chu et al., 2018) identify that they can well describe the possible relations among VGPs. Based on our typology, we construct a large VGP typology dataset of 25k VGP pairs via crowdsourcing with careful quality control.[2] In addition, we propose a VGP classification model that uses both language and visual features in a fusion network. We compare various types of language and visual features including pretrained word embeddings, image/region-level visual features and joint language and visual representations. We also try 4 different fusion methods in our model. Our experiments show that the joint learning of language and visual representations are crucial for VGP classification, which also achieves the best performance.

The creation of the semantic VGP typology will not only elucidate the phenomena behind VGPs and deepen the understanding about how human beings interpret vision with language, but also open up novel ways of utilizing VGPs for various vision and language tasks, which require semantic understanding. For instance, it can significantly boost the performance of textual entailment via visual grounding (Vu et al., 2018b) which is a fundamental but very challenging natural language understanding problem, enhanced by the classification of semantic relations between the VGPs in the premise and hypothesis sentences. Experiments conducted on the visually grounded textual entailment (VGTE) dataset V-SNLI (Vu et al., 2018b) verify the effectiveness our VGP typology for this task.. Understanding the semantic relation via visual grounding is also crucial towards machine reading (Ding et al., 2016), which is the main challenge in the Todai robot project.[3] Therefore, we believe that this work will significantly deepen and promote the research in both vision and language understanding.

[3] https://21robot.org/index-e.html.

2. Related work

2.1. Linguistic paraphrase typology

For linguistic paraphrase typology, Bhagat and Hovy (2013) defined 25 classes of paraphrases based on the kinds of lexical changes within a paraphrase pair. Vila et al. (2014) proposed a hierarchical typology based on lexical, morphological, syntactic, and discourse changes in paraphrases. Vila et al. (2015) annotated a paraphrase typology corpus according to their typology (Vila et al., 2014) on the Microsoft paraphrase corpus (MSPR) (Dolan et al., 2004). Benikova and Zesch (2017) classified paraphrases into word-level, phrase-level, and sentence-level, and analyzed their compositionality. Kovatchev et al. (2018) further annotated non-paraphrases and negations on the MSPR. For corpus-generated paraphrases, Pavlick et al. (2015) automatically added semantic entailment relations to the paraphrase database (PPDB) (Ganitkevitch et al., 2013). Our typology is for VGPs (Chu et al., 2018), which has not been considered in these studies. We adapt the semantic relations as the classes of our typology, which also differs from most linguistic paraphrase typology.

2.2. Visual semantic relations

Visual semantic relations focus on understanding the relationship between two objects, or all the objects represented as a scene graph of an image (Liu et al., 2019). The relationships can be classified into complete similarity, type similarity, hypernym, hyponym, parallel, and unknown (Hong et al., 2015), which are similar to our typology. However, we target on the typology of VGPs describing the same object, which is conceptually different from visual semantic relations.

2.3. Visual grounding

Visual grounding, which aims to find a specific region in an image given a query regarding to an entity, is a fundamental task for enhancing the performance of various joint vision and language tasks (Plummer et al., 2015). Because of the importance of visual grounding, many

research efforts have been dedicated to improve its accuracy (Plummer et al., 2015; Wang et al., 2016b; Fukui et al., 2016; Rohrbach et al., 2016; Wang et al., 2016a; Yeh et al., 2017; Plummer et al., 2017; Chen et al., 2017; Yang et al., 2020b,a; Yu et al., 2018; Dong et al., 2021). VGPs are different expressions of queries that can be grounded to the same entity in an image. We study the semantic typology of VGPs in this paper.

2.4. Visual captioning

Visual captioning aims for interpreting vision with language. Visual captioning has been studied for both images (Hossain et al., 2019) and videos (Aafaq et al., 2020). Image captioning works on either generating a caption for a single image or dense captions for all objects in a single image (Hossain et al., 2019). Video captioning works on either generating a caption for a single video or dense captions for all events in a single video (Aafaq et al., 2020). Both previous image and video captioning work has focused on generating more accurate, diverse, and discriminative captions. However, none previous visual captioning work has studied VGPs and the phenomena behind VGPs.

3. The semantic VGP typology

In natural language inference, seven basic semantic relations between two phrases X and Y have been defined, i.e., *equivalence, forward entailment, reverse entailment, negation, alternation, cover*, and *independence* (Maccartney, 2009). Detailed analyses on the VGP dataset (Chu et al., 2018) identify the five semantic relations of *equivalence, forward and reverse entailment, alternation*, and *independence* that can well describe the possible relations among VGPs. The *negation* (that describes two opposite concepts) and *cover* (that covers all concepts in the world by given two phrases) relations are excluded from our VGP typology definition: this is because they are almost equivalent to *alternation* and *independence*, respectively. The only difference is that, with the former two relationships, concepts described by X or Y should cover all concepts in the world, which does not apply to the later two relations. Because we focus on VGPs that describe the same concrete visual concept, whether X and Y cover all the concepts in the world is not relevant. Note that *forward and reverse entailment*, *alternation*, and *independence* are not semantic relations for linguistic paraphrases, but on the other hand, we observe many VGPs belong to those relations. This is because VGPs are phrases describing the same visual concepts, but they are not necessarily linguistic paraphrases.

Fig. 1 shows some examples of our semantic VGP typology. Fig. 1(a) and (b) are *equivalence*, which also are linguistic paraphrases; Fig. 1(c) and (d) are *forward entailment* VGPs where the first phrase contains more fine object attribute description of the same concept compared to the second phrase. Fig. 1(e) and (f) are *reverse entailment* VGPs where the first phrase contains more general object attribute description of the same concept compared to the second phrase. Fig. 1(g) and (h) use alternate phrases to describe the same visual concept, which may come from the difference in human recognition. The phrases in Fig. 1(i) and (j) are linguistically independent but become VGPs upon grounding. We believe that *alternation* and *independence* are new semantic phenomena specific to VGPs that cannot be explained only with language, and visual context is required to understand the relations between the phrases. Next, we detail the definition of the five VGP classes.

3.1. Alternation

The *alternation* class applies when phrase X and phrase Y are mutually exclusive: the same visual concept cannot have X and Y being true at the same time. For example, some VGPs in alternation may describe the same visual concept with gender difference (e.g., a man/a woman) or age-related difference (e.g., baby/child).

Table 1
Pair of stems used for detecting alternation VGPs.

Stem 1	Stem 2
Man/men	Boy
Baby	Toddler
Woman/women	Girl
Child	Toddler/baby

3.2. Forward and reverse entailment

The *entailment* relationship is categorized into *forward entailment* and *reverse entailment*. *Forward entailment* relationship is present when phrase X is a subtype of phrase Y (e.g., a rock climb wall/a rock). *Reverse entailment* is to be given when Y is a subtype of X (e.g., a stuffed animal/her teddy bear). We treat them as different classes to store the entailment direction.

3.3. Equivalence

The *equivalence* relationship is present when phrases X and Y entail each other in both directions. In particular, they are linguistic paraphrases, such as different expressions with the same meaning or abbreviation of the other phrase (e.g., Chevrolet/chevy).

3.4. Independence

In the *independence* relationship, phrase X and phrase Y describe different attributes of the same visual concept that are true at the same time (e.g., competitors/a group of bicyclist).

4. Dataset construction

In this section, we describe the original VGP dataset, how we produce the VGP typology dataset and ensure its quality. We also present statistics of the VGP typology dataset.

4.1. The VGP dataset

The Flickr30k dataset (Young et al., 2014) is one of standard benchmark datasets for image captioning. In a further step, Plummer et al. (2015) developed the Flickr30k Entities dataset, augmenting the original captions by annotating entities (noun phrases) with corresponding image regions. Chu et al. (2018) used the Flickr30k Entities dataset for the VGP identification task, which treated entities describing the same image region as VGPs. We used the Flickr30k Entities dataset as the starting point for constructing our VGP typology dataset as well.

We follow the steps indicated in Chu et al. (2018) and reproduce the VGP dataset they used in their study with 2.3 M, 80k, and 81k of phrase pairs for training, validation, and testing, respectively. As the main goal of this study is to categorize relations between VGPs into five classes described in Section 3, we extract only phrase pairs labeled as VGPs. At this point, we obtained 257k, 8.8k and 8.6k VGPs for training, validation, and testing, respectively.

4.2. Annotating VGP relations

We annotate the relations between VGP in a two stage process: (1) a first rule-based annotation for the simplest examples, (2) a more detailed human annotation for the more difficult examples.

Table 2
Number of VGPs for each step of the dataset construction.

Split	Rule	Human
Train	1,714	11,662
Val + test	227	11,568

Rule-based annotation. In the first stage, we visually inspect the data and we found that due to the limited vocabulary in the Flickr30K Entity dataset, some *alternation* examples were relatively easy to detect (e.g., age-related differences like girl/woman). We manually define rule-based cases for the those examples and filter phrase pairs according to them, reducing the costs of the manual annotation in the next stage. Table 1 shows the pair of stems used for the rule-based annotation. They are mostly based on the age difference aspect explained and intended to get a small, easy-to-detect *alternation* cases. With this automatic labeling, we annotate 1714, 110, and 117 pair of phrases as *alternation* in the training, validation and test splits, respectively.

Human annotation. In the second stage, we use Amazon Mechanical Turk (AMT)[4] to annotate part of the data that has not been annotated as *alternation* in the first stage. After removing duplicate pairs of phrases, we ask workers to annotate about 11k samples in the test and validation sets as well as 11k samples in the training set. During the annotation process, we show workers the VGP pair, the image[5] and original captions which the phrases belong to.[6] This visual and sentence context helped workers to answer faster and achieve more accurate results. Each HIT (Human Intelligence Task, the basic work unit on AMT) has fifty pairs of phrases, and each phrase pair is annotated by 5 different workers. We use weighted voting based on quality control questions to determine the final label among the labels by the 5 workers, which is explained in detail in the next section. Table 2 represents the different stages of the dataset construction.

4.3. Quality control

To ensure workers carefully read the questions, we used a dummy question. The dummy question specifies a relation label, *e.g.*, by displaying "choose *Alternation*". If a worker fails to select the required relation label, we consider the worker is a bot or answers randomly, thus immediately reject the HITs by the worker.

We employ weighted voting by five workers to determine the final answer. Each answer by a worker is weighted with a reliability score. The reliability score is the correct answer ratio of reliability check questions. For reliability check questions, we use easy examples, which should be correctly answered. To select these easy examples, we collect VGPs for which four or more workers out of five select the same relation label. We insert five reliability check questions at random positions in a HIT. Therefore, the reliability score is 0.8 if the worker submits 4 correct answers for the reliability check questions. We reject HITs by a worker whose reliability score is lower than 0.4. After the annotation, we review 100 random phrase pairs in order to evaluate the quality of annotation. We verify a 96% accuracy score compared to the ground-truth. Fig. 2 shows some mislabeled examples from the 100 VGP random samples. We can see that these examples are really tricky to be annotated. We also calculated the inter-annotator agreement based on Fleiss' kappa (Fleiss et al., 1971), and got a moderate agreement of 0.5308.

[4] https://www.mturk.com/.

[5] Workers do not have access to the entity bounding box, but we believe that for humans it is easy to ground the phrase pair in an image.

[6] The AMT interface is shown in Section 9.

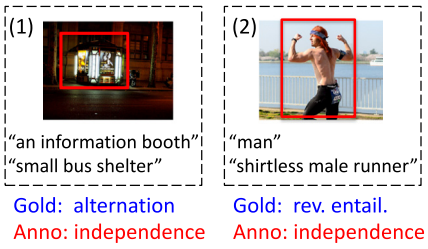

Fig. 2. Mislabeled examples from AMT workers (Anno is short for annotation).

Table 3
Overview of our VGP typology dataset.

	Train	Val	Test	All
# image	9,736	964	964	11,664
# phrase pair	13,376	5,734	6,061	25,171
# unique phrase	14,431	3,991	3,968	18,946
# vocabulary	5,294	2,189	2,170	6,217
Max. phrase len.	11	8	7	11
Avg. phrase len.	2.1	2	1.9	2

4.4. Dataset overview

Table 3 shows the overview of our VGP typology dataset. Our dataset has 25k phrase pairs with 18k unique phrases. Most phrases consist of few words and the average phrase length is 2.

We also present the distribution of the VGP classes for the training, validation, and testing splits in Fig. 3. We can see that in the training split, only a small number (4.8%) of VGPs belong to the paraphrase relationship; a large part of them (43.3%) belong to the entailment relationships; the others belong to the alternation (23.9%) and independence relationship (28.0%). This can somehow indicate the percentage of ways in how human beings interpret the same vision concept with different language expressions. The testing and validation splits have similar distributions of VGP classes. However, different from the other splits, *alternation* VGPs have a significantly higher presence in the training split. The reason for this is that as described in Section 4.2, only a subset of 11k training data of Flickr30k Entities dataset are manually annotated, while 1714 VGP alternation pairs are annotated by rules on the entire training data. Therefore, the proportion of *alternation* is larger in the training split.

5. VGP classification model

Our model considers both language and visual components to identify the classes of VGPs. We introduce a fusion network to fuse both component in different ways. Fig. 4 shows the overview of our model. The model takes an image/region and a phrase pair describing the same visual concept in the image as input. The phrases are encoded into phrase embeddings. For the visual component, we extract visual features from the image or corresponding image region. We explore variations of language and visual feature extraction methods. The fusion network fuses both modalities for VGP classification. The fused feature is fed to an multiple layer perceptron (MLP) network which predicts the probability of each class.

5.1. Language features

5.1.1. Word embedding average

We first preprocess input phrases by lower-casing and stop-word removal. Each word in the phrases is encoded into continuous vector representations by a word embedding model. In our experiment, we explore several word embedding models to exploit knowledge on language, which will be explained in Section 6.2. We obtain a

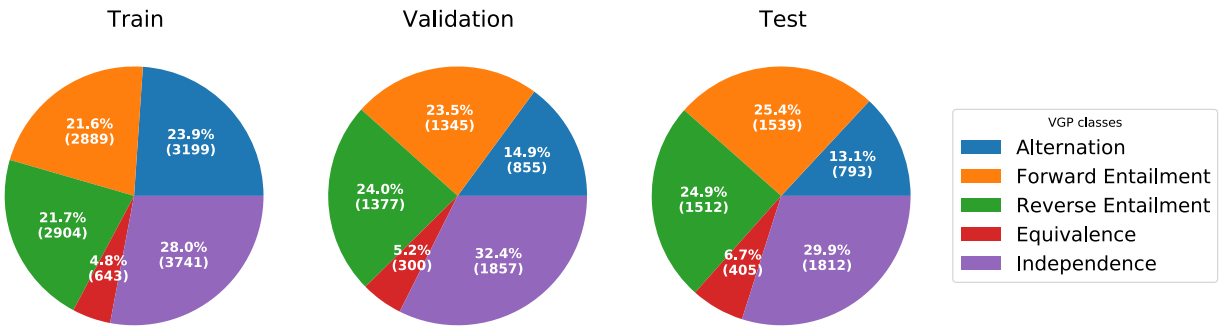

Fig. 3. VGP class distribution according to split. Note that due to rule-based cases in the original training split for the alternation class, we obtain a significant higher amount of data for this class.

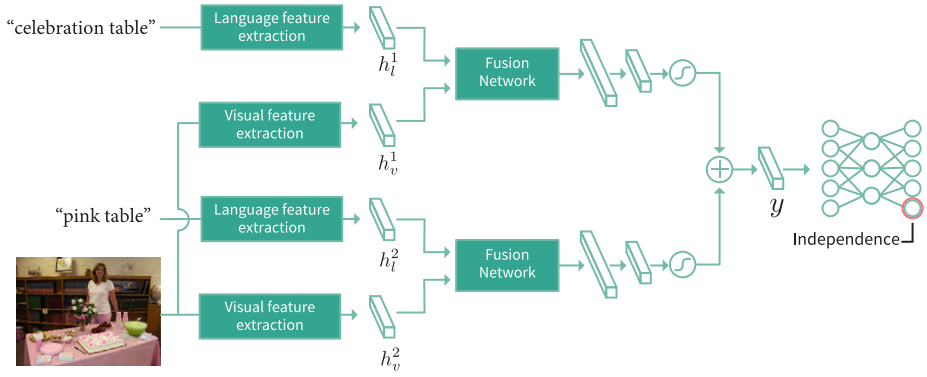

Fig. 4. Illustration of our VGP classification model. Our model fuses language and visual features and classifies the input VGP pair into VGP classes.

phrase embedding by average-pooling the word embeddings. The previous work (Chu et al., 2018) suggested other embedding methods for phrases, but we found that the average pooling performed better than the other methods for our task, such as Fisher vectors. Let $\mathbf{x}_{\mathrm{p}}^1$ and $\mathbf{x}_{\mathrm{p}}^2$ in $\mathbb{R}^{300}$ be phrase embeddings for each phrase in the input phrase pair. These phrase embeddings are transformed using an MLP, Ψ, with two fully-connected layers. Both of these fully-connected layers are followed by batch normalization and ReLU activation. We compute language features $\mathbf{h}_{\mathrm{l}}^1 \in \mathbb{R}^{1,000}$ and $\mathbf{h}_{\mathrm{l}}^2 \in \mathbb{R}^{1,000}$ by:

$$\mathbf{h}_{\mathrm{l}}^1 = \mathrm{ReLU}(\Psi(\mathbf{x}_{\mathrm{p}}^1)), \qquad \mathbf{h}_{\mathrm{l}}^2 = \mathrm{ReLU}(\Psi(\mathbf{x}_{\mathrm{p}}^2)). \tag{1}$$

5.1.2. ViLBERT

ViLBERT (Lu et al., 2019) is a vision and language model pretrained on the large-scale Conceptual Captions image caption dataset (Sharma et al., 2018), which has achieved state-of-the-art performance on vision and language tasks of visual question answering, visual commonsense reasoning, referring expressions, and caption-based image retrieval. Therefore, we use the pretrained language model of ViLBERT[7] to represent each input phrase. Note that similar to BERT (Devlin et al., 2019), the ViLBERT language model is sentence-level and thus we treat each input phrase as a sentence and do not remove stop-word from the phrase. Both phrases are represented as $\mathbb{R}^{1,024}$ phrase embeddings by ViLBERT, and they are further computed with Eq. (1).

5.2. Visual features

We compare two image-level and one region-level visual features as follows:

5.2.1. Image-level

VGG: We extract the fc7 feature vector from the input image using the VGG16 model (Simonyan and Zisserman, 2014). Our VGG16 model is initialized with parameters trained on the ImageNet (Deng et al., 2009) provided by chainercv.[8] The visual feature $\mathbf{x}_{\mathrm{v}} \in \mathbb{R}^{4,096}$ is fed to an MLP, Φ, again with two fully-connected layers followed by batch normalization and ReLU activation. Our VGG visual feature $\mathbf{h}_{\mathrm{v}} \in \mathbb{R}^{1,000}$ is given by:

$$\mathbf{h}_{\mathrm{v}} = \mathrm{ReLU}(\Phi(\mathbf{x}_{\mathrm{v}})). \tag{2}$$

ViLBERT: We use the pretrained visual model of ViLBERT to represent each image. Each image is represented as $\mathbf{x}_{\mathrm{v}} \in \mathbb{R}^{2,048}$ and then computed with Eq. (2).

5.2.2. Region-level

We first use the Faster R-CNN (Ren et al., 2015) in conjunction with the ResNet-101 CNN (He et al., 2016) pretrained on Visual Genome (Krishna et al., 2017) to extract 100 image region features from an image. For these 100 image regions, we calculate the Intersection over Union (IoU) scores against the input image region and select the one that obtains the highest IoU score as the input image region feature. The image region feature $\mathbf{x}_{\mathrm{v}} \in \mathbb{R}^{2,048}$ is again fed to an MLP Φ with two fully-connected layers followed by batch normalization and ReLU activation to obtain our region visual feature $\mathbf{h}_{\mathrm{v}} \in \mathbb{R}^{1,000}$ as in Eq. (2).

5.3. Fusion network

We propose a fusion network to fuse the language and visual features. Our fused features $\mathbf{h}^1 \in \mathbb{R}^{1,000}$ and $\mathbf{h}^2 \in \mathbb{R}^{1,000}$ that fuses two phrase embeddings and a visual features are given by:

$$\mathbf{h}^1 = \mathrm{fuse}(\mathbf{h}_{\mathrm{l}}^1, \mathbf{h}_{\mathrm{v}}), \mathbf{h}^2 = \mathrm{fuse}(\mathbf{h}_{\mathrm{l}}^2, \mathbf{h}_{\mathrm{v}})$$

where fuse is a fusion function and we compare 4 different ways for it:

[7] https://github.com/facebookresearch/vilbert-multi-task.

[8] https://github.com/chainer/chainercv.

Table 4
Accuracy for VGP classification using image-level visual features (VGG for word embedding average based settings, and ViLBERT visual features for the ViLBERT setting). w/sub, w/add, w/mul, and w/con denote fusion with subtraction, addition, element-wise multiplication, and concatenation, respectively. Scores are the average of 10 runs, numbers in parentheses are the standard deviation of 10 runs.

Model	Phrase-only	Visual-only	Phrase+visual			
			w/sub	w/add	w/mul	w/con
Word2vec	0.40 (0.01)	0.27 (0.01)	0.43 (0.01)	0.42 (0.03)	0.43 (0.01)	0.40 (0.06)
Retro_WordNet	0.33 (0.04)		0.21 (0.04)	0.20 (0.04)	0.16 (0.02)	0.28 (0.04)
Retro_FrameNet	0.30 (0.04)		0.21 (0.03)	0.22 (0.04)	0.17 (0.03)	0.22 (0.04)
Retro_PPDB	0.37 (0.01)		0.42 (0.02)	0.45 (0.01)	0.41 (0.03)	0.41 (0.03)
ViLBERT	0.45 (0.00)	0.47 (0.01)	**0.48 (0.01)**	0.47 (0.01)	0.46 (0.02)	**0.48 (0.01)**

Table 5
Accuracy for VGP classification using region-level visual features. w/sub, w/add, w/mul, and w/con denote fusion with subtraction, addition, element-wise multiplication, and concatenation, respectively. Scores are the average of 10 runs, numbers in parentheses are the standard deviation of 10 runs.

Model	Phrase-only	Visual-only	Phrase+visual			
			w/sub	w/add	w/mul	w/con
Word2vec	0.40 (0.01)	0.28 (0.01)	0.42 (0.03)	0.43 (0.01)	0.42 (0.01)	0.42 (0.03)
Retro_WordNet	0.33 (0.04)		0.23 (0.04)	0.22 (0.03)	0.22 (0.02)	0.23 (0.02)
Retro_FrameNet	0.30 (0.04)		0.21 (0.02)	0.24 (0.02)	0.21 (0.02)	0.22 (0.04)
Retro_PPDB	0.37 (0.01)		**0.44 (0.02)**	0.43 (0.01)	0.42 (0.02)	0.42 (0.02)

- Subtraction

$$\text{fuse}(\mathbf{h}_\text{l}, \mathbf{h}_\text{v}) = \text{ReLU}(\Theta(\mathbf{h}_\text{l} - \mathbf{h}_\text{v})),$$

- Addition

$$\text{fuse}(\mathbf{h}_\text{l}, \mathbf{h}_\text{v}) = \text{ReLU}(\Theta(\mathbf{h}_\text{l} + \mathbf{h}_\text{v})),$$

- Element-wise multiplication

$$\text{fuse}(\mathbf{h}_\text{l}, \mathbf{h}_\text{v}) = \text{ReLU}(\Theta(\mathbf{h}_\text{l} \odot \mathbf{h}_\text{v})),$$

- Concatenation

$$\text{fuse}(\mathbf{h}_\text{l}, \mathbf{h}_\text{v}) = \text{ReLU}(\Theta([\mathbf{h}_\text{l}, \mathbf{h}_\text{v}])),$$

where Θ is another MLP with two fully-connected layers followed by batch normalization.[9]

The two fused features are further merged as

$$\mathbf{y} = \tanh(W^1\mathbf{h}^1 + \mathbf{b}^1) + \tanh(W^2\mathbf{h}^2 + \mathbf{b}^2),$$

where $\mathbf{y} \in \mathbb{R}^{300}$. The last part of our model is an MLP with two layers. The output of the first layer in $\mathbb{R}^{300}$, and the second layer outputs five values corresponding to the VGP classes. The probability of the VGP classes are obtained by applying the softmax function to the second layer's output, and we choose the one with the highest probability as the predicted class. We use the cross-entropy loss to train our model.

6. Experimental settings for VGP classification

6.1. Training

For training, we used Adam (Kingma and Ba, 2014) as the optimizer. The mini-batch size was set to 100. The learning rate was initially set to 0.01 and at each epoch was multiplied by a factor of 0.1. Training was terminated after 6 epochs, where we observed the loss converged on the validation set.

[9] A fusion model based on co-attention between the tokens and regions used in Transformer-based vision-and-language pre-training such as Li et al. (2020) might be effective as well, but we leave it as future work.

6.2. Word embeddings

We compared different types of word embeddings to produce the word embedding average language features for our model.

- **Word2vec**: we used the word2vec embeddings trained on the Google News corpus[10] (Mikolov et al., 2013).
- **Retrofitting**: as our typology is based on semantics, we want the word embeddings capture semantic information. To this end, we used the retrofitting method proposed by Faruqui et al. (2015). This method minimizes an objective function so that the retrofitted word embedding will be close to both the original embedding and its neighbors' embedding in semantic lexicon. Following Faruqui et al. (2015), we used the following three semantic lexicons for retrofitting.
  - **Retro_WordNet**: word2vec embeddings retrofitted on the WordNet (Miller, 1995), which is a structured semantic lexical database of English containing synonym, hyponym and hypernym information between word pairs.
  - **Retro_FrameNet**: word2vec embeddings retrofitted on the FrameNet (Baker et al., 1998), which is a large semantic lexical database of English, constructed based on semantic frames.
  - **Retro_PPDB**: word2vec embeddings retrofitted on the PPDB (Ganitkevitch et al., 2013), which is a large paraphrase database of English, created from parallel corpora through bilingual pivoting (Callison-Burch et al., 2006).

6.3. Input modalities

We also conducted experiments to show the effect of the different input modalities by comparing the following three settings:

- **Phrase-only**: this model only takes the language components of the VGPs into account and predicts their class based on the language features. The language features are then fed to the classifier previously described in Section 5.3, which is composed of two layers, with the last layer having five units to generate the prediction based on a softmax function.

[10] https://github.com/mmihaltz/word2vec-GoogleNews-vectors.

- **Visual-only**: this model only takes the visual components of the VGPs into account and predicts their class based on the visual features. The visual features are used in the same way as the phrase-only model.
- **Phrase+visual**: this is the model that we presented in Section 5. We fuse word embedding average language features to either image-level VGG features or region-level features. The fusion of ViLBERT language and vision features is also tested.

7. Results & discussion for VGP classification

Due to the randomness of the initial parameters, we report the average accuracy of 10 runs together with their standard deviation. Tables 4 and 5 show the accuracy of VGP classification using image-level and region-level visual features, respectively. We show the phrase-only accuracy using word embedding average in both tables for easy comparison among different input modalities.

For phrase-only models, we can see that word2vec works better than retrofitting. The reason for this is that although the retrofitting is helpful to improve the accuracy of the entailment and equivalence classes by making the embeddings closer to the semantic lexicons, they perform negatively for the alternation and independence classes due to the fact that they do not exist in the semantic lexicons. ViLBERT works significantly better than other phrase-only models, indicating the importance of vision and language joint representation for our VGP classification task.

The accuracy of visual-only models are worse than phrase-only models but significantly better than random (i.e., 0.2), indicating the importance of visual features for our task. The image-level VGG visual-only model performs slightly worse than the region-level visual-only model. We think the reason for this is that because VGPs are describing a particular image region making region-level features more helpful. The visual features obtained from ViLBERT show a significantly high accuracy compared to VGG and region-level features, indicating the importance of jointly learning language and vision representation for our task again.

For phrase+visual models, we can see that fusing word embedding average language features of Word2Vec or Retro_PPDB with either image-level or region-level visual features with all the four fusion methods improves the accuracy, which indicates the effectiveness of language and visual feature fusion for our task. Fusing Retro_WordNet or Retro_FrameNet with visual features are not helpful, and we think this is again due to the absence of alternation and independence classes in these semantic lexicons. For ViLBERT, phrase+visual performs similarly compared to using a single modality. This is because either ViLBERT phrase-only or visual-only features have already capture both vision and language representations by pretraining, and thus further fusing them does not help more. Regarding the 4 different fusion methods, we did not observe significant difference among them for the best performed models, i.e., phrase+visual models of ViLBERT and Retro_PPDB in both Tables 4 and 5. Subtraction shows slightly better performance in these two models. We think the reason for this might be that the difference between the language and visual features of a VGP pair can be an important clue to classify the relation of this VGP pair.

We further show the class-level performance in Table 6 for the best performed model (i.e., ViLBERT Phrase+visual w/sub in Table 4). We can see that the performance for equivalence is significantly worse than the other classes. This is due to the small number of training data for the equivalence class as shown in Fig. 3. Alternation and independence also show lower performance compared to entailment, we think the reason is that these two classes are new semantic phenomena which requires more sophisticated representation learning to address. Fig. 5 further shows the confusion matrix for the best performed model. For equivalence, we can see that many equivalence VGPs have been incorrectly classified into independence. For alternation, we can see

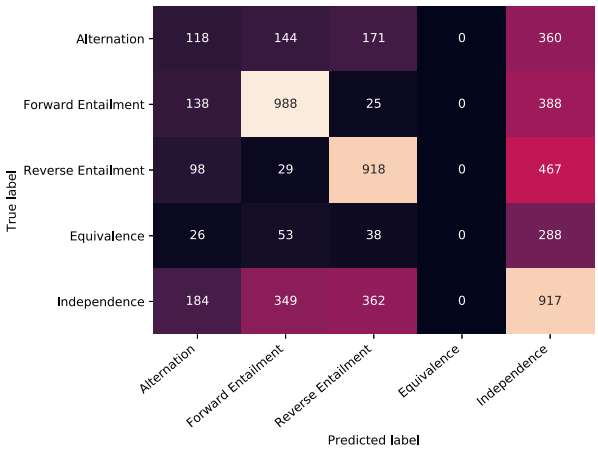

Fig. 5. Confusion matrix for the best model (i.e., ViLBERT Phrase+visual w/sub in Table 4).

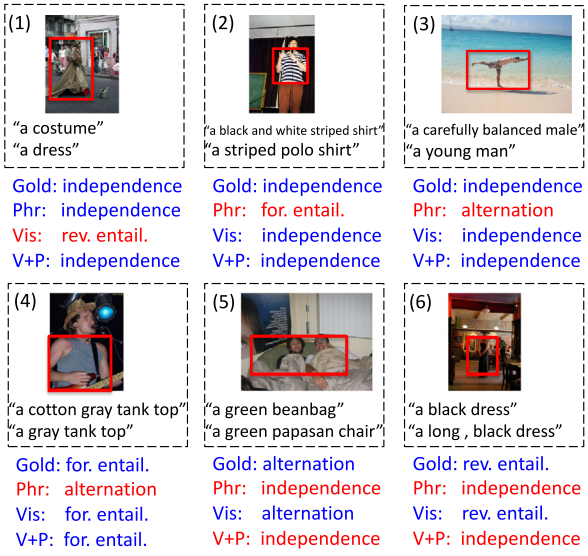

Fig. 6. VGP classification examples for the ViLBERT phrase-only, visual-only and phrase-visual w/sub models. "Gold", "Phr", "Vis", and "V+P" denote for "ground-truth", "phrase-only", "visual-only", and "phrase+visual", respectively. Blue labels are for ground truth and correct prediction, while red labels are for incorrect prediction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Detailed results per class for the best model (i.e., ViLBERT Phrase+visual w/sub in Table 4). The support column represents the absolute quantity of data points for each class in the test split.

VGP class	Precision	Recall	F1	Support
Alternation	0.21	0.15	0.17	793
Forward entailment	0.63	0.64	0.64	1,539
Reverse entailment	0.61	0.61	0.61	1,512
Equivalence	0.00	0.00	0.00	405
Independence	0.38	0.51	0.43	1,812

that they have been incorrectly classified into mainly entailment and independence. Also many independence VGPs have been incorrectly classified into mainly entailment.

In addition, we investigated the results of the ViLBERT phrase-only, visual-only and phrase-visual w/sub models. We found that fusing language and visual features helps the improvement of the independence class most, and it also improves the classification of the entailment classes. Examples (1), (2) and (3) in Fig. 6 are improved examples of

Fig. 7. A screenshot of our AMT user interface.

independence. In example (1), visual-only fails to predict the correct class but phrase-only predicts correctly, and fusing them makes it correct. In both example (2) and (3), the phrase-only model fails to predict them as independence and the visual-only model successfully predicts the correct class; fusing them together remains the correct prediction. Example (4) in Fig. 6 shows an improved example of the forward entailment class by fusion compared to phrase-only, where the visual-only model helps in fusion as it correctly predicts the class. Example (5) and (6) show two examples of side effects by fusion, where the vision-only model predicts the correct class, but fusion leads to the wrong prediction due to the bad effect of the phrase-only model.

8. Experiments on downstream tasks

We investigated the effect of our VGP typology on a downstream task, the VGTE task using the V-SNLI dataset (Vu et al., 2018a). Given a sample consisting of an image and a pair of sentences (a premise P and a hypothesis H) related to the image, the task is to classify the pair into 3 types of relations: P entails H, P contradicts H, or P is unrelated to H. The V-SNLI dataset consists of 545,620, 9842, and 9824 samples for training, validation, and testing, respectively. Because both the V-SNLI and our VGP typology datasets are originally from the Flickr30k dataset, we can find the overlap between these datasets by selecting the V-SNLI sentence pairs that contain our VGP typology pairs. The overlap contains 9995, 180, and 188 samples in the training, validation, and

Table 7
Accuracy for the VGTE task.

	Original	Overlap
Baseline	0.42 (0.01)	0.47 (0.01)
Joint	0.43 (0.00)	0.48 (0.00)

testing splits, respectively. To directly investigate the effectiveness of our VGP typology for the VGTE task, we only used the overlapped 9995 samples for training. We evaluated the performance in two settings:

- Original: Using the original V-SNLI validation and testing sets (9842 and 9824 samples, respectively).
- Overlap: Using the overlapped validation and testing sets (180 and 188 samples, respectively).

For the VGTE task, we compared two models:

- Baseline: We used the same model as our VGP classification model described in Section 5, but changed the output classes to 3 instead of 5 as in our VGP classification task.
- Joint: We used the same model as our VGP classification model described in Section 5, but both VGTE and VGP classification were jointly trained with a loss L_{joint}:

  $$L_{joint} = \lambda L_{vgte} + (1 - \lambda) L_{vgp},$$

  where L_{vgte} is the VGTE loss, L_{vgp} is the VGP classification loss, and λ is a hyperparameter.

For both the Baseline and Joint models, we used the ViLBERT Phrase+visual w/sub setting, because it achieved the best performance in the VGP classification task. We also used the same training settings described in Section 6.1. For the Joint model, we tuned λ from 0.9 to 0.5 with an interval of 0.1.

Same as VGP classification, we report the average accuracy of 10 runs together with their standard deviation for the VGTE task. Table 7 shows the results. We can see that Joint outperforms Baseline in both the Original and Overlap settings, indicating the effectiveness of our VGP typology for the VGTE task. However, the improvement is not big. We suspect the reason for this can be two fold: we simply re-use the VGP classification model for the experiments and a more sophisticated model might be necessary; as shown in Section 7, VGP classification itself is a difficult task, which limits the performance improvement in the VGTE task.

9. Conclusion

In this paper, we presented the semantic typology for VGPs to elucidate the VGP phenomena, caused by different linguistic interpretation of the same visual concept by human beings. We constructed a large VGP typology dataset via crowdsourcing and proposed a multimodal model for VGP classification. Experiments indicated that language and vision representation joint modeling is important for VGP classification. In the future, we plan to improve the language and visual representations by fine-tuning them on our task to further improve the accuracy. One limitation of the current typology is that it focuses more on the linguistic perspective of VGPs. Alternate typology might be created by focusing more on the visual perspective, which is our another future work. We hope that this pioneering work can promote the further research on the understanding and applications of VGPs.

Supplementary material

Fig. 7 shows a screenshot of our AMT user interface for annotating the VGP typology dataset.

CRediT authorship contribution statement

Chenhui Chu: Conceptualization, Software, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition. **Vinicius Oliveira:** Dataset construction, Software, Writing – original draft, Writing – review & editing. **Felix Giovanni Virgo:** Software, Writing – review & editing. **Mayu Otani:** Methodology, Writing – review & editing. **Noa Garcia:** Methodology, Writing – review & editing. **Yuta Nakashima:** Conceptualization, Writing – review editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by ACT-I, JST and JSPS KAKENHI No. 18H03264.

References

Aafaq, N., Gilani, S.Z., Liu, W., Mian, A., 2020. Video description: A survey of methods, datasets and evaluation metrics. ACM Comput. Surv. 52 (6), 1–37.

Androutsopoulos, I., Malakasiotis, P., 2010. A survey of paraphrasing and textual entailment methods. J. Artificial Intelligence Res. 38 (1), 135–187, URL http://dl.acm.org/citation.cfm?id=1892211.1892215.

Baker, C.F., Fillmore, C.J., Lowe, J.B., 1998. The berkeley FrameNet project. In: Proceedings of the 17th International Conference on Computational Linguistics - Volume 1. In: COLING '98, pp. 86–90. http://dx.doi.org/10.3115/980451.980860, URL https://doi.org/10.3115/980451.980860.

Benikova, D., Zesch, T., 2017. Same same, but different: Compositionality of paraphrase granularity levels. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. pp. 90–96. http://dx.doi.org/10.26615/978-954-452-049-6_014.

Berant, J., Liang, P., 2014. Semantic parsing via paraphrasing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Baltimore, Maryland, pp. 1415–1425, URL http://www.aclweb.org/anthology/P14-1133.

Bhagat, R., Hovy, E., 2013. What is a paraphrase? Comput. Linguist. 39 (3), 463–472.

Callison-Burch, C., Koehn, P., Osborne, M., 2006. Improved statistical machine translation using paraphrases. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. pp. 17–24, URL https://www.aclweb.org/anthology/N06-1003.

Chen, K., Kovvuri, R., Nevatia, R., 2017. Query-guided regression network with context policy for phrase grounding. In: ICCV. pp. 824–832.

Chu, C., Kurohashi, S., 2016. Paraphrasing out-of-vocabulary words with word embeddings and semantic lexicons for low resource statistical machine translation. In: LREC. pp. 644–648.

Chu, C., Otani, M., Nakashima, Y., 2018. iParaphrasing: Extracting visually grounded paraphrases via an image. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 3479–3492.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: CVPR09.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.

Ding, N., Goodman, S., Sha, F., Soricut, R., 2016. Understanding image and text simultaneously: a dual vision-language machine comprehension task. CoRR http://arxiv.org/abs/1612.07833.

Dolan, B., Quirk, C., Brockett, C., 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics. pp. 350–356, URL https://www.aclweb.org/anthology/C04-1051.

Dong, W., Otani, M., Garcia, N., Nakashima, Y., Chu, C., 2021. Cross-lingual visual grounding. IEEE Access 9, 349–358. http://dx.doi.org/10.1109/ACCESS.2020.3046719.

Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A., 2015. Retrofitting word vectors to semantic lexicons. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1606–1615. http://dx.doi.org/10.3115/v1/N15-1184, URL https://www.aclweb.org/anthology/N15-1184.

Fleiss, J., et al., 1971. Measuring nominal scale agreement among many raters. Psychol. Bull. 76 (5), 378–382.

Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M., 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In: EMNLP. pp. 457–468.

Ganitkevitch, J., Van Durme, B., Callison-Burch, C., 2013. PPDB: The paraphrase database. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 758–764, URL https://www.aclweb.org/anthology/N13-1092.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: CVPR. pp. 770–778.

Hong, R., Yang, Y., Wang, M., Hua, X., 2015. Learning visual semantic relationships for efficient visual retrieval. IEEE Trans. Big Data 1 (4), 152–161. http://dx.doi.org/10.1109/TBDATA.2016.2515640, URL https://doi.org/10.1109/TBDATA.2016.2515640.

Hossain, M.Z., Sohel, F., Shiratuddin, M.F., Laga, H., 2019. A comprehensive survey of deep learning for image captioning. ACM Comput. Surv. 51 (6), 1–36.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR http://arxiv.org/abs/1412.6980.

Kovatchev, V., Marti, T., Salamo, M., 2018. ETPC - A paraphrase identification corpus annotated with extended paraphrase typology and negation. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L., 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. 123 (1), 32–73. http://dx.doi.org/10.1007/s11263-016-0981-7, URL https://doi.org/10.1007/s11263-016-0981-7.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J., 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV.

Ling, W., Dyer, C., Black, A.W., Trancoso, I., 2013. Paraphrasing 4 microblog normalization. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp. 73–84, URL http://www.aclweb.org/anthology/D13-1008.

Liu, D., Bober, M., Kittler, J., 2019. Visual semantic information pursuit: A survey. CoRR http://arxiv.org/abs/1903.05434.

Lu, J., Batra, D., Parikh, D., Lee, S., 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 32, Curran Associates, Inc., pp. 13–23, URL https://proceedings.neurips.cc/paper/2019/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf.

Maccartney, B., 2009. Natural language inference. (Ph.D. thesis). AAI3364139.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR http://arxiv.org/abs/1301.3781.

Miller, G.A., 1995. Wordnet: A lexical database for english. Commun. ACM 38 (11), 39–41. http://dx.doi.org/10.1145/219717.219748, URL http://doi.acm.org/10.1145/219717.219748.

Otani, M., Chu, C., Nakashima, Y., 2019. Adaptive gating mechanism for identifying visually grounded paraphrases. In: Proceedings of the ICCV 2019 MDALC Workshop.

Otani, M., Chu, C., Nakashima, Y., 2020. Visually grounded paraphrase identification via gating and phrase localization. Neurocomputing 404, 165–172. http://dx.doi.org/10.1016/j.neucom.2020.04.066, URL https://www.sciencedirect.com/science/article/pii/S0925231220306512.

Pavlick, E., Bos, J., Nissim, M., Beller, C., Van Durme, B., Callison-Burch, C., 2015. Adding semantics to data-driven paraphrasing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1512–1522. http://dx.doi.org/10.3115/v1/P15-1146, URL https://www.aclweb.org/anthology/P15-1146.

Plummer, B.A., Mallya, A., Cervantes, C.M., Hockenmaier, J., Lazebnik, S., 2017. Phrase localization and visual relationship detection with comprehensive image-language cues. In: ICCV. pp. 1928–1937.

Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S., 2015. Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2641–2649.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-CNN: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., Liu, Y., 2007. Statistical machine translation for query expansion in answer retrieval. In: ACL. pp. 464–471, URL http://www.aclweb.org/anthology/P07-1059.

Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., Schiele, B., 2016. Grounding of textual phrases in images by reconstruction. In: ECCV. pp. 817–834.

Samaran, J., Garcia, N., Otani, M., Chu, C., Nakashima, Y., 2021. Attending self-attention: a case study of visually grounded supervision in vision-and-language transformers. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop. Association for Computational Linguistics, pp. 81–86. http://dx.doi.org/10.18653/v1/2021.acl-srw.8, URL https://aclanthology.org/2021.acl-srw.8.

Sharma, P., Ding, N., Goodman, S., Soricut, R., 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, pp. 2556–2565. http://dx.doi.org/10.18653/v1/P18-1238, URL https://www.aclweb.org/anthology/P18-1238.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR http://arxiv.org/abs/1409.1556.

Vila, M., Martí, M.A., Rodríguez, H., 2014. Is this a paraphrase? What kind? Paraphrase boundaries and typology. Open J. Modern Linguist. 4, 205–218.

Vila, M., Bertran, M., Martí, M.A., Rodríguez, H., 2015. Corpus annotation with paraphrase types: New annotation scheme and inter-annotator agreement measures. Lang. Resour. Eval. 49 (1), 77–105. http://dx.doi.org/10.1007/s10579-014-9272-5, URL http://dx.doi.org/10.1007/s10579-014-9272-5.

Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2015. Show and tell: A neural image caption generator. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3156–3164.

Vu, H.T., Greco, C., Erofeeva, A., Jafaritazehjan, S., Linders, G., Tanti, M., Testoni, A., Bernardi, R., Gatt, A., 2018a. Grounded textual entailment. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018).

Vu, H.T., Greco, C., Erofeeva, A., Jafaritazehjan, S., Linders, G., Tanti, M., Testoni, A., Bernardi, R., Gatt, A., 2018b. Grounded textual entailment. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2354–2368, URL https://www.aclweb.org/anthology/C18-1199.

Wang, M., Azab, M., Kojima, N., Mihalcea, R., Deng, J., 2016a. Structured matching for phrase localization. In: ECCV. pp. 696–711.

Wang, L., Li, Y., Lazebnik, S., 2016b. Learning deep structure-preserving image-text embeddings. In: CVPR. pp. 5005–5013.

Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., Hengel, A.v.d., 2017. Visual question answering: A survey of methods and datasets. Comput. Vis. Image Underst. 1–20, URL http://dx.doi.org/10.1016/j.cviu.2017.05.001.

Yang, Z., Chen, T., Wang, L., Luo, J., 2020a. Improving one-stage visual grounding by recursive sub-query construction. In: ECCV.

Yang, S., Li, G., Yu, Y., 2020b. Propagating over phrase relations for one-stage visual grounding. In: ECCV.

Yeh, R., Xiong, J., Hwu, W.W., Do, M., Schwing, A.G., 2017. Interpretable and globally optimal prediction for textual grounding using image concepts. In: NIPS. pp. 1909–1919.

Young, P., Lai, A., Hodosh, M., Julia, H., 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Trans. Assoc. Comput. Linguist. 2 (1), 67–78.

Yu, Z., Yu, J., Xiang, C., Zhao, Z., Tian, Q., Tao, D., 2018. Rethinking diversified and discriminative proposal generation for visual grounding. In: IJCAI. pp. 1114–1120. http://dx.doi.org/10.24963/ijcai.2018/155, URL https://doi.org/10.24963/ijcai.2018/155.

Zhou, L., Lin, C.-Y., Munteanu, D.S., Hovy, E., 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Main Conference. Association for Computational Linguistics, New York City, USA, pp. 447–454, URL http://www.aclweb.org/anthology/N/N06/N06-1057.