

Improving Visual Relationship Detection using Semantic Modeling of Scene Descriptions

Stephan Baier¹, Yunpu Ma^{1,2}, and Volker Tresp^{1,2}

¹ Ludwig Maximilian University, 80538 Munich, Germany
stephan.baier@campus.lmu.de

² Siemens AG, Corporate Technology, Munich, Germany
yunpu.ma@siemens.com
volker.tresp@siemens.com

Abstract. Structured scene descriptions of images are useful for the automatic processing and querying of large image databases. We show how the combination of a statistical semantic model and a visual model can improve on the task of mapping images to their associated scene description. In this paper we consider scene descriptions which are represented as a set of triples (*subject*, *predicate*, *object*), where each triple consists of a pair of visual objects, which appear in the image, and the relationship between them (e.g. *man-riding-elephant*, *man-wearing-hat*). We combine a standard visual model for object detection, based on convolutional neural networks, with a latent variable model for link prediction. We apply multiple state-of-the-art link prediction methods and compare their capability for visual relationship detection. One of the main advantages of link prediction methods is that they can also generalize to triples which have never been observed in the training data. Our experimental results on the recently published Stanford Visual Relationship dataset, a challenging real world dataset, show that the integration of a statistical semantic model using link prediction methods can significantly improve visual relationship detection. Our combined approach achieves superior performance compared to the state-of-the-art method from the Stanford computer vision group.

Keywords: Visual Relationship Detection, Knowledge Graph, Link Prediction

1 Introduction

Extracting semantic information from unstructured data, such as images or text, is a key challenge in artificial intelligence. Semantic knowledge in a machine-readable form is crucial for many applications such as search, semantic querying and question answering.

Novel computer vision algorithms, mostly based on convolutional neural networks (CNN), have enormously advanced over the last years. Standard applications are image classification and, more recently, also the detection of objects

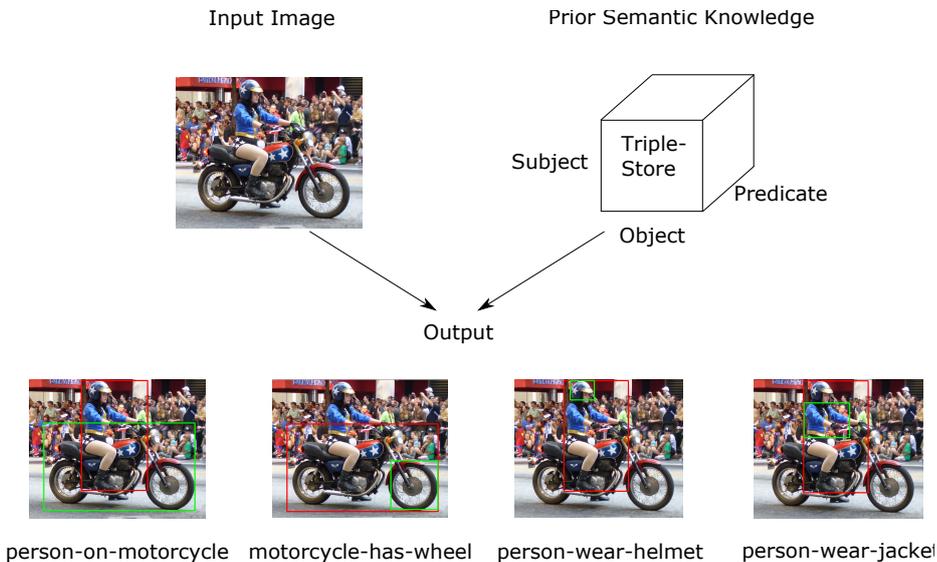


Fig. 1: The input to the model is a raw image. In combination with a semantic prior we generate triples, which describe the scene.

in images. However, the semantic expressiveness of image descriptions that consist simply of a set of objects is rather limited. Semantics is captured in more meaningful ways by the relationships between objects. In particular, visual relationships can be represented by triples of the form $(subject, predicate, object)$, where two entities appearing in an image are linked through a relation (e.g. *man-riding-elephant*, *man-wearing-hat*).

Extracting triples, i.e. visual relationships, from raw images is a challenging task, which has been a focus in the Semantic Web community for some time [2–4, 27, 32, 38] and recently also gained substantial attention in main stream computer vision [6, 7, 18, 25]. First approaches used a single classifier, which takes an image as input and outputs a complete triple [7, 25]. However, these approaches do not scale to datasets with many object types and relationships, due to the exploding combinatorial complexity. Recently, [18] proposed a method which classifies the visual objects and their relationships in independent pre-processing steps, and then derives a prediction score for the entire triple. This approach was applied to the extraction of triples from a large number of potential triples. In the same paper, the first large-scale dataset for visual relationship extraction was published.

The statistical modeling of graph-structured knowledge bases, often referred to as knowledge graphs, has recently gained growing interest. The most popular approaches learn embeddings for the entities and relations in the knowledge graph. Based on the embeddings a likelihood for a triple can be derived. This

approach has mainly been used for link prediction, which tries to predict missing triples in a knowledge graph. A recent review paper can be found in [20].

In the approach described in this paper, statistical knowledge base models, which can infer the likelihood of a triple, are used to support the task of visual link prediction. For example if the visual model detects a motorbike, it is very likely that the triple *motorbike-has_part-wheel* is true, as all motorbikes have wheels. We suggest that integrating such prior knowledge can improve various computer vision tasks. In particular, we propose to combine the likelihood from a statistical semantic model with a visual model to enhance the prediction of image triples.

Figure 1 illustrates our approach. The model takes as input a raw image and combines it with a semantic prior, which is derived from the training data. Both types of information are fused, to predict the output, which consists of relevant bounding boxes and a set of triples describing the scene.

For combining the semantic prior with the visual model we employ a probabilistic approach which can be divided into a semantic part and a visual part. We show how the semantic part of the probabilistic model can be implemented using standard link prediction methods and the visual part using recently developed computer vision algorithms.

We train our semantic model by using absolute frequencies from the training data, describing how often a triple appears in the training data. By applying a latent variable model, we are able to also generalize to unseen or rarely seen triples, which still have a high likelihood of being true, due to their similarity to other likely triples. For example if we frequently observe the triple *person-ride-motorcycle* in the training data we can generalize also to a high likelihood for *person-ride-bike* due to the similarity between *motorcycle* and *bike*, even if the triple *person-ride-bike* has not been observed or just rarely been observed in the training data. The similarity of *motorcycle* and *bike* can be derived from other triples, which describe, for example, that both have a *wheel* and both have a *handlebar*.

We conduct experiments on the Stanford Visual Relationship dataset recently published by [18]. We evaluate different model variants on the task of predicting semantic triples and the corresponding bounding boxes of the subject and object entities detected in the image. Our experiments show, that including the semantic model improves on the state-of-the-art result in the task of mapping images to their associated triples.

The paper is structured as follows. Section 2 gives an overview of the state-of-the-art link prediction models, the employed computer vision techniques and related work. Section 3 describes the semantic and the visual part of our model and how both can be combined in a probabilistic framework. In Section 4 we show a number of different experiments. Finally, we conclude our work with Section 5.

2 Background and Related Work

Our proposed model joins ideas from two areas, computer vision and statistical relational learning for semantic modeling. Both fields have developed rapidly in recent years. In this chapter we discuss relevant work from both areas.

2.1 Statistical Link Prediction

A number of statistical models have been proposed for modeling graph-structured knowledge bases often referred to as knowledge graphs. Most methods are designed for predicting missing links in the knowledge graph. A recent review on link prediction can be found in [20]. A knowledge graph \mathcal{G} consists of a set of triples $\mathcal{G} = \{(s, p, o)_i\}_{i=1}^N \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. The entities $s, o \in \mathcal{E}$ are referred to as *subject* and *object* of the triple, and the relation between the entities $p \in \mathcal{R}$ is referred to as *predicate* of the triple.

Link prediction methods can be described by a function $\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$, which maps a triple (s, p, o) to a real valued score. The score of a triple $\theta(s, p, o)$ represents the likelihood of the triple being true. Most recently developed link prediction models learn a latent representation, also called embedding, for the entities and the relations. In the following we describe the link prediction methods, which are used in paper.

DistMult: DistMult [35] scores a triple by building the tri-linear dot product of the embeddings, such that

$$\theta(s, p, o) = \langle a(s), r(p), a(o) \rangle = \sum_j a(s)_j r(p)_j a(o)_j \quad (1)$$

where $a : \mathcal{E} \rightarrow \mathbb{R}^d$ maps entities to their latent vector representations and similarly $r : \mathcal{R} \rightarrow \mathbb{R}^d$ maps relations to their latent representations. The dimensionality d of the embeddings, also called rank, is a hyperparameter of the model.

ComplEx: ComplEx [33] extends DistMult to complex valued vectors for the embeddings of both, relations and entities. The score function is

$$\begin{aligned} \theta(s, p, o) = \operatorname{Re}(\langle a(s), r(p), \overline{a(o)} \rangle) = & \langle \operatorname{Re}(a(s)), \operatorname{Re}(r(p)), \operatorname{Re}(a(o)) \rangle \\ & + \langle \operatorname{Im}(a(s)), \operatorname{Re}(r(p)), \operatorname{Im}(a(o)) \rangle \\ & + \langle \operatorname{Re}(a(s)), \operatorname{Im}(r(p)), \operatorname{Im}(a(o)) \rangle \\ & - \langle \operatorname{Im}(a(s)), \operatorname{Im}(r(p)), \operatorname{Re}(a(o)) \rangle \end{aligned} \quad (2)$$

where $a : \mathcal{E} \rightarrow \mathbb{C}^d$ and $r : \mathcal{R} \rightarrow \mathbb{C}^d$; $\operatorname{Re}(\cdot)$ and $\operatorname{Im}(\cdot)$ denote the real and imaginary part, respectively, and $\bar{\cdot}$ denotes the complex conjugate.

Multiway NN: The multiway neural network [8, 20] concatenates all embeddings and feeds them to a neural network of the form

$$\theta(s, p, o) = (\beta^T \tanh(A[a(s), r(p), a(o)])) \quad (3)$$

where $[\cdot, \cdot, \cdot]$ denotes the concatenation of the embeddings $a(s), r(p), a(o) \in \mathbb{R}^d$. A is a weight matrix and β a weight vector.

RESCAL: The tensor decomposition RESCAL [21] learns vector embeddings for entities and matrix embeddings for relations. The score function is

$$\theta(s, p, o) = a(s) \cdot R(p) \cdot a(o) \quad (4)$$

with \cdot denoting the dot product, $a : \mathcal{E} \rightarrow \mathbb{R}^d$ and $R : \mathcal{R} \rightarrow \mathbb{R}^{d \times d}$.

Typically, the models are trained using a ranking cost function [20]. For our task of visual relationship detection, we will train them slightly differently using a Poisson cost function for modeling count data, as we will show in section 3.2. Another popular link prediction method is TransE [5], however it is not appropriate for modeling count data; thus we are not considering it in this work.

2.2 Image Classification and Object Detection

Computer vision methods for image classification and object detection have improved enormously over the last years. Convolutional neural networks (CNN), which apply convolutional filters in a hierarchical manner to an image, have become the standard for image classification. In this work we use the following two methods.

VGG: The VGG-network is a convolutional neural network, which has shown state-of-the-art performance at the Imagenet challenge [28]. It exists in two versions, i.e. the VGG-16 with 16 convolutional layers and VGG-19 with 19 convolutional layers.

RCNN: The region convolutional neural network (RCNN) [11] proposes regions, which show some visual objects in the image. It uses a selective search algorithm for getting candidate regions in an image [31]. The RCNN algorithm then rejects most of the regions based on a classification score. As a result, a small set of region proposals is derived. There are two extensions to RCNN, which are mainly faster and slightly more accurate [10, 23]. However, in our experiments we use the original RCNN, for a fair comparison with [18]. Our focus is on improving visual relationship detection through semantic modeling rather than on improving computer vision techniques.

2.3 Visual Relationship Detection

Visual relationship detection is about predicting triples from images, where the triples consist of two visual objects and the relationship between them. This is related to visual caption generation, which recently gained considerable popularity among the deep learning community, where an image caption, consisting of natural text, is generated given an image [16, 17, 34]. However, the output in visual relationship detection is more structured (a set of triples), and thus it is more appropriate for further processing, e.g. semantic querying. Related work on relational reasoning with images can also be found in visual question

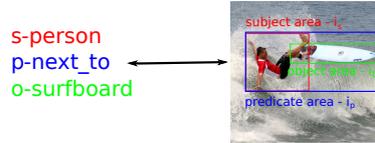


Fig. 2: The subject and object of the triple relate to two regions in the image, and the predicate relates to the union of the two regions.

answering [1, 15, 26, 39] and has also been subject to neural symbolic reasoning [27, 38]. The extraction of semantic triples has also been successfully applied to text documents, e.g. the Google Knowledge Vault project for improving the Google Knowledge Graph [8].

Some earlier work on visual relationship detection was concerned with learning spatial relationships between objects, however with a very limited set of only four spatial relations [9, 12]. Other related work attempted to learn actions and object interactions of humans in videos and images [13, 19, 22, 24, 36, 37]. Full visual relationship detection has been demonstrated in [6, 7, 25], however, also with only small amounts of possible triples. In [6], an ontology over the visual concepts is defined and combined with a neural network approach to maintain semantic consistency.

The Stanford computer vision group proposed a scalable model and applied it to a large-scale dataset, with 700000 possible triples. In their work, entities of the triples were detected separately and a joint score for each triple candidate was computed [18]. The visual module in [18] uses the following computer vision methods, which we will also use in our approach. An RCNN for object detection is used to derive candidate regions. Further, a VGG-16 is applied to the detected regions for obtaining object classification scores for each region. Finally, a second VGG, which classifies relationships, such as *taller-than*, *wears*, etc., is applied to the union of pairs of regions. The model also contains a language prior, which can model semantic relationships to some extent based on word embeddings. The language prior allows the model to generalize to unseen triples. However, our experiments show that integrating state-of-the-art link prediction methods for modeling semantics is more appropriate for improving general prediction and generalization to unseen triples.

3 Modeling Visual Relationships

In the following we describe our approach to jointly modeling images and their corresponding semantics.

3.1 Problem Description

We assume data consisting of images and corresponding triple sets. For each *subject* s of a triple (s, p, o) there exists a corresponding region i_s in the image.



Fig. 3: The pipeline for deriving a ranked list of triples is as follows: The image is passed to a RCNN, which generates region candidates. We build pairs of regions and predict a score for every triple, based on our ranking method. The visual part is similar to [18], however the ranking method is different as it includes a semantic model.

Similarly, each *object* o corresponds to an region i_o , and each *predicate* p to an region i_p , which is the union of the regions i_s and i_o . Thus, one data sample can be represented as a six-tuple of the form (i_s, i_p, i_o, s, p, o) . Figure 2 shows an example of a triple and its corresponding bounding boxes. During training, all triples and their corresponding areas are observed. After model training the task is to predict the most likely tuples (i_s, i_p, i_o, s, p, o) for a given image. Figure 3 shows the processing pipeline of our method, which takes a raw image as an input, and outputs a ranked list of triples and bounding boxes.

3.2 Semantic Model

In contrast to typical knowledge graph modeling, we do not only have one global graph \mathcal{G} , but an instance of a knowledge graph \mathcal{G}_i for every image i . Each triple which appears in a certain image can be described as a tuple (s, p, o, i) . The link prediction model shall reflect the likelihood of a triple to appear in a graph instance, as a prior without seeing the image. By summing over the occurrences in the i -th dimension, we derive the absolute frequency of triples (s, p, o) in the training data, which we denote as $y_{s,p,o}$. We aim to model $y_{s,p,o}$ using the link prediction methods described in Section 2.1. As we are dealing with count data, we assume a Poisson distribution on the model output $\theta(s, p, o)$. The log-likelihood for a triple is

$$\log p(y_{s,p,o} | (s, p, o), \Theta) = y_{s,p,o} \log \eta(\theta(s, p, o)) - \eta(\theta(s, p, o)) - \log(y_{s,p,o}!), \quad (5)$$

where Θ are the model parameters of the link prediction method and η is the parameter for the Poisson distribution, namely

$$\eta(\theta(s, p, o)) = \exp(\theta(s, p, o)). \quad (6)$$

We train the model by minimizing the negative log-likelihood. In the objective function the last term $\log(y_{s,p,o}!)$ can be neglected, as it does not depend on the model parameters. Thus the cost function for the whole training dataset becomes

$$cost = \sum_{(s,p,o)} \eta(\theta(s, p, o)) - y_{s,p,o} \log \eta(\theta(s, p, o)). \quad (7)$$

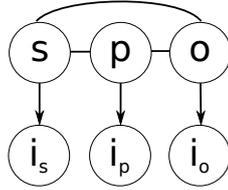


Fig. 4: The probabilistic graphical model describes the interaction between the visual and the semantic part for a given image. We assume the image regions i_s , i_p and i_o to be given by the RCNN and infer the underlying s, p, o triples.

Using this framework, we can train any of the link prediction methods described in Section 2, by plugging the prediction into the cost function and minimizing the cost function using a gradient-descent based optimization algorithm. In this work we use Adam, a recently proposed first-order gradient-based optimization method with adaptive learning rate [14].

3.3 Visual Model

Our visual model is similar to the approach used in [18]. Figure 3 shows the involved steps. An image is first fed to an RCNN, which generates region proposals for a given image. The region proposals are represented as bounding boxes within the image. The visual model further consists of two convolutional neural networks (CNNs). The first CNN which we denote as CNN_e takes as input the subregion of the image defined by a bounding box and classifies entities from the set \mathcal{E} .

The second CNN, which we denote as CNN_r takes the union region of two bounding boxes as an input, and classifies the relationship from the set \mathcal{R} . While training, both CNNs use the regions (bounding boxes) provided in the training data.

For new images, we derive the regions from the RCNN. We build all possible pairs of regions, where each pair consists of a region i_s and i_o . We apply CNN_e to the regions, to derive the classification scores $CNN_e(s|i_s)$ and $CNN_e(o|i_o)$. Then the union of the regions i_s and i_o is fed to CNN_r to derive the score $CNN_r(p|i_p)$, where $i_p = \text{union}(i_s, i_o)$. Figure 2 shows an example of the bounding boxes of the *subject* and the *object*, as well as the union of the bounding boxes, which relates to the *predicate* of the triple.

3.4 Probabilistic Joint Model

In the last step of the pipeline in Figure 3, which we denote as ranking step, we need to combine the scores from the visual model with the scores from the semantic model. For joining both, we propose a probabilistic model for the interaction between the visual and the semantic part. Figure 4 visualizes the joint

model for all variables in a probabilistic graphical model. The joint distribution factors as

$$p(s, p, o, i_s, i_p, i_o) \propto \tilde{p}(s, p, o) \cdot \tilde{p}(i_s|s) \cdot \tilde{p}(i_p|p) \cdot \tilde{p}(i_o|o) \quad (8)$$

with \tilde{p} denoting unnormalized probabilities. We can divide the joint probability of equation 8 into two parts. The first part is $\tilde{p}(s, p, o)$, which models semantic triples. The second part is $\tilde{p}(i_s|s) \cdot \tilde{p}(i_p|p) \cdot \tilde{p}(i_o|o)$, which models the visual part given the semantics.

Following [29, 30] we derive the unnormalized joint probability of the triples $\tilde{p}(s, p, o)$ using a Boltzmann distribution. With the energy function $E(s, p, o) = -\log \eta(\theta(s, p, o))$ the unnormalized probability for the triples becomes

$$\tilde{p}(s, p, o) = \eta(\theta(s, p, o)). \quad (9)$$

The visual modules described in the previous section, model the unnormalized probabilities $\tilde{p}(s|i_s)$, $\tilde{p}(p|i_p)$, and $\tilde{p}(o|i_o)$. By applying Bayes rule to equation 8 and assuming equal probabilities for all image regions we get

$$p(s, p, o, i_s, i_p, i_o) \propto \tilde{p}(s, p, o) \cdot \frac{\tilde{p}(s|i_s) \cdot \tilde{p}(p|i_p) \cdot \tilde{p}(o|i_o)}{\tilde{p}(s) \cdot \tilde{p}(p) \cdot \tilde{p}(o)}. \quad (10)$$

The additional terms of the denominator $\tilde{p}(s)$, $\tilde{p}(p)$, $\tilde{p}(o)$ can be derived through marginalization of $\tilde{p}(s, p, o)$.

For each image, we derive the region candidates i_s , i_p , i_o from the RCNN. We do not have to normalize the probabilities as we are finally interested in a ranking of the most likely six-tuples (i_s, i_p, i_o, s, p, o) for a given image. The final unnormalized probability score on which we rank the tuples is

$$\tilde{p}(s, p, o, i_s, i_p, i_o) = \eta(\theta(s, p, o)) \frac{CNN_e(s|i_s) \cdot CNN_r(p|i_p) \cdot CNN_e(o|i_o)}{\tilde{p}(s) \cdot \tilde{p}(p) \cdot \tilde{p}(o)}. \quad (11)$$

4 Experiments

We evaluate our proposed method on the recently published Stanford Visual Relationship dataset [18]. We compare our proposed method against the state-of-the-art method from [18] in the task of predicting semantic triples from images. As in [18] we will divide the setting into two parts: First an evaluation on how well the methods perform when predicting all possible triples and second only evaluating on triples, which did not occur in the training data. This setting is also referred to as zero-shot learning, as the model has not seen any training images containing the triples which are used for evaluation.

4.1 Dataset

The dataset consists of 5000 images. The semantics are described by triples, consisting of 100 entity types, such as *motorcycle*, *person*, *surfboard*, *watch*, etc. and

Table 1: Results for visual relationship detection. We report Recall at 50 and 100 for four different validation settings.

Task	Phrase Det.		Rel. Det.		Predicate Det.		Triple Det.	
	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50
Lu et al. V [18]	2.61	2.24	1.85	1.58	7.11	7.11	2.68	2.30
Lu et al. full [18]	17.03	16.17	14.70	13.86	47.87	47.87	18.11	17.11
RESCAL	19.17	18.16	16.88	15.88	52.71	52.71	20.23	19.13
MultiwayNN	18.88	17.75	16.65	15.57	51.82	51.82	19.76	18.53
ComplEx	19.36	18.25	17.12	16.03	53.14	53.14	20.23	19.06
DistMult	15.42	14.27	13.64	12.54	42.18	42.18	16.14	14.94

70 relation types, e.g. *next_to*, *taller_than*, *wear*, *on*, etc. The entities correspond to visual objects in the image. For all *subject* and *object* entities the corresponding regions in the image are given. Each image has in average 7.5 triples, which describe the scene. In total there are 37993 triples in the dataset. The dataset is split into 4000 training and 1000 test images. The data split is identical to the split in [18], thus we can directly compare our results. There are 1877 triples, which only occur in images from the test set but not in the training set.

4.2 Visual Relationship Detection

Experimental Setting For doing visual relationship detection, we consider four different types of settings. Three of them are identical to the experimental settings in [18]. We add a fourth setting, which eliminates the evaluation of correctly detecting the bounding boxes, and solely evaluates the predicted triples. The four settings are as follows.

Phrase detection: In phrase detection the task is to give a ranking of likely triples plus the corresponding regions for the *subject* and *object* of the triple. The bounding boxes are derived from the RCNN. Subsequently, we apply our ranking function (see equation 11) to the pairs of objects, as shown in Figure 3. A triple with its corresponding bounding boxes is considered correctly detected, if the triple is similar to the ground truth, and if the union of the bounding boxes has at least 50 percent overlap with the union of the ground truth bounding boxes.

Relationship detection: The second setting, which is also considered in [18] is relationship detection. It is similar to phrase detection, but with the difference that it is not enough when the union of the bounding boxes is overlapping by at least 50 percent. Instead, both the bounding box of the *subject* and the bounding box of the *object* need at least 50 percent of overlap with their ground truth.

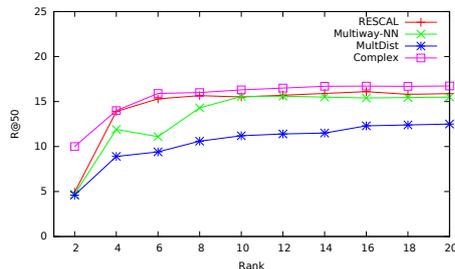


Fig. 5: Recall at 50 as a function of the rank

Triple detection: We add a setting, which we call triple detection, which evaluates only the prediction of the triples. A triple is correct if it corresponds to the ground truth. The position of the predicted bounding boxes is not evaluated.

Predicate detection: In predicate detection, it is assumed that *subject* and *object* are given, and only the correct *predicate* between both needs to be predicted. Therefore, we use the ground truth bounding boxes with the respective labels for the objects instead of the bounding boxes derived by the RCNN. This separates the problem of object detection from the problem of predicting relationships.

For each test image, we create a ranked list of triples. Similar to [18] we report the recall at the top 100 elements of the ranked list and the recall at top 50. Note, that there are 700000 possible triples, out of which the correct triples need to be ranked on top.

When training the semantic model, we hold out 5 percent of the nonzero triples as a validation set. We determine the optimal rank for the link prediction methods based on that hold-out set. For the visual model (RCNN and VGG) we use a pretrained model provided by [18].

Results Table 1 shows the results for visual relationship detection. The first row shows the results, when only the visual part of the model is applied. This model performs poorly, in all four settings. The full model in the second row adds the language prior to it and also some regularization terms during training, which are described in more detail in [18]. This drastically improves the results. As expected the recall at top 100 is better than at top 50, however the difference is rather small, which shows that most of the correctly ranked triples are ranked quite high. The results for predicate detection are much better than for the other settings. This shows that one of the main problems in visual relationship detection is the correct prediction of the entities. In the last four rows we report the results of our method, which adds a link prediction model to the visual model. We compare the results for the integration of the four link prediction methods described in section 2.1. We see that with all link prediction methods the model performs constantly better than the state-of-the-art method proposed by [18],

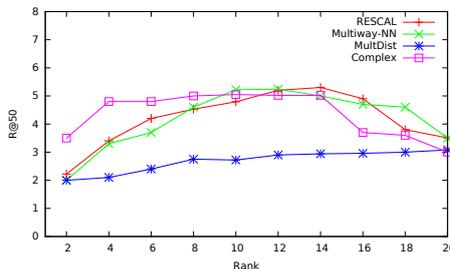


Fig. 6: Recall at 50 as a function of the rank for the zero-shot setting.

except for *DistMult*. For *Relationship detection*, which is the most challenging setting, *Complex* works best, with a recall of 17.12 and 16.03 for the top 100 and top 50 results respectively. *RESCAL* performs slightly better than the *Multiway Neural Network* in all evaluation settings. For the setting of *Triple Detection* the scores are higher for all methods, as expected, as the overlap of the bounding boxes is not taken into account. However, the relative performance between the methods does not vary much.

Figure 5 shows the recall at 50 on the test set for our different variants as a function of the rank. We see that the performances of *Complex* and *RESCAL* converge relatively quickly to a recall of around 16. The *Multiway Neural Network* converges a bit slower, to a slightly smaller maximum. *DistMult* converges slower and to a much smaller maximum recall of 12.5.

4.3 Zero-shot Learning

Experimental Setting We also include an experimental setting, where we only evaluate on triples, which had not been observed in the training data. This setting reveals the generalization ability of the semantic model. The test set contains 1877 of these triples. We evaluate based on the same settings as in the previous section, however for the recall we only count how many of the unseen triples are retrieved.

Results Table 2 shows the results for the zero-shot experiments. This task is much more difficult, which can be seen by the huge drop in recall. However, also in this experiment, including the semantic model significantly improves the prediction. For the first three settings, the best performing method, which is the *Multiway Neural Network*, almost retrieves twice as many correct triples, as the state-of-the-art model of [18]. Especially, for the *Predicate Detection*, which assumes the objects and subjects to be given, a relatively high recall of 16.60 can be reached. In the zero-shot setting for *Predicate Detection* even the integration of the worst performing semantic model *DistMult* shows significantly better performance than the state-of-the-art method. These results clearly show that our model is able to infer also new likely triples, which have not been

Table 2: Results for the zero shot learning experiments. We report Recall at 50 and 100 for four different validation settings.

Task	Phrase Det.		Rel. Det.		Predicate Det.		Triple Det.	
	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50
Lu et al. V [18]	1.12	0.95	0.78	0.67	3.52	3.52	1.20	1.03
Lu et al. full [18]	3.75	3.36	3.52	3.13	8.45	8.45	5.39	4.79
RESCAL	6.59	5.82	6.07	5.30	16.34	16.34	6.07	5.30
MultiwayNN	6.93	5.73	6.24	5.22	16.60	16.60	6.24	5.22
ComplEx	6.50	5.73	5.82	5.05	15.74	15.74	5.82	5.05
DistMult	4.19	3.34	3.85	3.08	12.40	12.40	3.85	3.08

observed in the training data. This is one of the big benefits of the link prediction methods.

Figure 6 shows the recall at 50 on the zero-shot test set as a function of the rank. As expected, the models start to overfit in the zero-shot setting if the rank is too high. With a limited rank the models have less freedom for explaining the variation in the data; this forces them to focus more on the underlying structure, which improves the generalization property. *ComplEx*, which has more parameters due to the complex valued embeddings, performs best with small ranks and reaches the maximum at a rank of around 8. *Multiway Neural Network* reaches the maximum at a rank of 10 and *RESCAL* at a rank of 14. The highest recall is achieved by *RESCAL* at 5.3.

5 Conclusion

We presented a novel approach for including semantic knowledge into visual relationship detection. We combine a state-of-the-art computer vision procedure with latent variable models for link prediction, in order to enhance the modeling of relationships among visual objects. By including a statistical semantic model, the predictive quality can be enhanced significantly. Especially the prediction of triples, which have not been observed in the training data, can be enhanced through the generalization properties of the semantic link prediction methods. The recall of the best performing link-prediction method in the zero-shot setting is almost twice as high as the state-of-the-art method. We proposed a probabilistic framework for integrating both the semantic prior and the computer vision algorithms into a joint model. This paper shows how the interaction of semantic and perceptual models can support each other to derive better predictive accuracies. The developed methods show great potential also for broader application areas, where both semantic and sensory data is observed. For example, in an industrial setting it might be interesting to model sensor measurements from a

plant jointly with a given ontology. The improvement over the state-of-the-art vision model shows that performance improvement does not only rely on better computer vision models but also on improvements in the semantic modeling. As part of future work, we will explore more expressive ontologies, for example by integrating external information from publicly available knowledge graphs, to further improve the results.

References

1. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 39–48 (2016)
2. Bagdanov, A.D., Bertini, M., Del Bimbo, A., Serra, G., Torniai, C.: Semantic annotation and retrieval of video events using multimedia ontologies. In: Semantic Computing, 2007. ICSC 2007. International Conference on. pp. 713–720. IEEE (2007)
3. Bannour, H., Hudelot, C.: Towards ontologies for image interpretation and annotation. In: Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on. pp. 211–216. IEEE (2011)
4. Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., Handschuh, S., Kompatsiaris, Y., Staab, S., Strintzis, M.G.: Semantic annotation of images and videos for multimedia analysis. In: European Semantic Web Conference. pp. 592–607. Springer (2005)
5. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
6. Chen, N., Zhou, Q.Y., Prasanna, V.: Understanding web images by object relation network. In: Proceedings of the 21st international conference on World Wide Web. pp. 291–300. ACM (2012)
7. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3d geometric phrases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 33–40 (2013)
8. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 601–610. ACM (2014)
9. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
10. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
12. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. *International Journal of Computer Vision* 80(3), 300–316 (2008)
13. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(10), 1775–1789 (2009)

14. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123(1), 32–73 (2017)
16. Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12), 2891–2903 (2013)
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* 521(7553), 436–444 (2015)
18. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: *European Conference on Computer Vision*. pp. 852–869. Springer (2016)
19. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. pp. 3177–3184. IEEE (2011)
20. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1), 11–33 (2016)
21. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. pp. 809–816 (2011)
22. Ramanathan, V., Li, C., Deng, J., Han, W., Li, Z., Gu, K., Song, Y., Bengio, S., Rosenberg, C., Fei-Fei, L.: Learning semantic relationships for better action retrieval in images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1100–1109 (2015)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*. pp. 91–99 (2015)
24. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 433–440 (2013)
25. Sadeghi, M.A., Farhadi, A.: Recognition using visual phrases. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. pp. 1745–1752. IEEE (2011)
26. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427 (2017)
27. Serafini, L., Donadello, I., Garcez, A.d.: Learning and reasoning in logic tensor networks: theory and application to semantic image interpretation. In: *Proceedings of the Symposium on Applied Computing*. pp. 125–130. ACM (2017)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Tresp, V., Esteban, C., Yang, Y., Baier, S., Krompaß, D.: Learning with memory embeddings. arXiv preprint arXiv:1511.07972 (2015)
30. Tresp, V., Ma, Y., Baier, S., Yang, Y.: Embedding learning for declarative memories. In: *European Semantic Web Conference*. pp. 202–216. Springer (2017)
31. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *International journal of computer vision* 104(2), 154–171 (2013)

32. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: science, services and agents on the World Wide Web* 4(1), 14–28 (2006)
33. Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: *Proceedings of the 33rd International Conference on Machine Learning* (2016)
34. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: *International Conference on Machine Learning*. pp. 2048–2057 (2015)
35. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014)
36. Yao, B., Fei-Fei, L.: Grouplet: A structured image representation for recognizing human and object interactions. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. pp. 9–16. IEEE (2010)
37. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. pp. 17–24. IEEE (2010)
38. Yilmaz, Ö., Garcez, A.S.d., Silver, D.L.: A proposal for common dataset in neural-symbolic reasoning studies. In: *NeSy@ HLAI* (2016)
39. Zhu, Y., Lim, J.J., Fei-Fei, L.: Knowledge acquisition for visual question answering via iterative querying