Improving Scene Graph Classification by Exploiting Knowledge from Texts

Sahand Sharifzadeh[1*], **Sina Moayed Baharlou**[1*†], **Martin Schmitt**[2],
Hinrich Schütze[2], **Volker Tresp** [1,3]

[1] Department of Informatics, LMU Munich, Germany
[2] Center for Information and Language Processing (CIS), LMU Munich, Germany
[3] Siemens AG, Munich, Germany
sahand.sharifzadeh@gmail.com, sina.baharlou@gmail.com

Abstract

Training scene graph classification models requires a large amount of annotated image data. Meanwhile, scene graphs represent relational knowledge that can be modeled with symbolic data from texts or knowledge graphs. While image annotation demands extensive labor, collecting textual descriptions of natural scenes requires less effort. In this work, we investigate whether textual scene descriptions can substitute for annotated image data. To this end, we employ a scene graph classification framework that is trained not only from annotated images but also from symbolic data. In our architecture, the symbolic entities are first mapped to their correspondent image-grounded representations and then fed into the relational reasoning pipeline. Even though a structured form of knowledge, such as the form in knowledge graphs, is not always available, we can generate it from unstructured texts using a transformer-based language model. We show that by fine-tuning the classification pipeline with the extracted knowledge from texts, we can achieve ∼8x more accurate results in scene graph classification, ∼3x in object classification, and ∼1.5x in predicate classification, compared to the supervised baselines with only 1% of the annotated images.

Introduction

Relational reasoning is one of the essential components of intelligence; humans explore their environment by grasping the entire context of a scene rather than studying each item in isolation from the others. Furthermore, we expand our understanding of the world by educating ourselves about novel facts through reading or listening. For example, we might have never seen a “cow wearing a dress” but might have read about Hindu traditions of decorating cows. While we already have a robust visual system that can extract basic visual features such as edges and curves from a scene, the description of a “cow wearing a dress” refines our visual understanding of relations on an object level and enables us to recognize a dressed cow when seeing it.

Relational reasoning is gaining growing popularity in the Computer Vision community and especially in the form of scene graph (SG) classification. The goal of SG classification is to classify objects and their relations in an image. One of the challenges in SG classification is collecting annotated image data. Most approaches in this domain rely on thousands of manually labeled and curated images. In this paper, we investigate whether the SG classification models can be fine-tuned from textual scene descriptions (similar to the “dressed cow” example above).

We consider a classification pipeline with two major parts: a feature extraction *backbone*, and a *relational reasoning* component (Figure 1). The backbone is typically a convolutional neural network (CNN) that detects objects and extracts an image-based representation for each. On the other hand, the relational reasoning component can be a variant of a recurrent neural network [Xu et al. 2017, Zellers et al. 2018] or graph convolutional networks [Yang et al. 2018, Sharifzadeh, Baharlou, and Tresp 2021]. This component operates on an object level by taking the latent representations of all the objects in the image and propagating them in the graph.

Note that, unlike the feature extraction backbone that requires images as input, the relational reasoning component operates on graphs with the nodes representing objects and the edges representing relations. The distinction between the input to the backbone (images) and the relational reasoning component (graphs) is often overlooked. Instead, the scene graph classification pipeline is treated as a network that takes only images as inputs. However, one can also train or fine-tune the relational reasoning component directly by injecting it with relational knowledge. For example, Knowledge Graphs (KGs) contain curated facts that indicate the relations between a `head` object and a `tail` object in the form of `(head, predicate, tail)` e.g., `(Person, Rides, Horse)`. The facts in KGs are represented by symbols whereas the inputs to the relational reasoning component are image-based embeddings. In this work, we map the triples to image-grounded embeddings as if they are coming from an image. We then use these embeddings to fine-tune the relational reasoning component through a denoising graph autoencoder scheme.

Note that the factual knowledge is not always available in a well-structured form, specially in domains where the knowledge is not stored in the machine-accessible form of KGs. In fact, most of the collective human knowledge is only

*These authors contributed equally.
†S. M. Baharlou contributed to this project while he was a visiting researcher at the Ludwig Maximilian University of Munich.

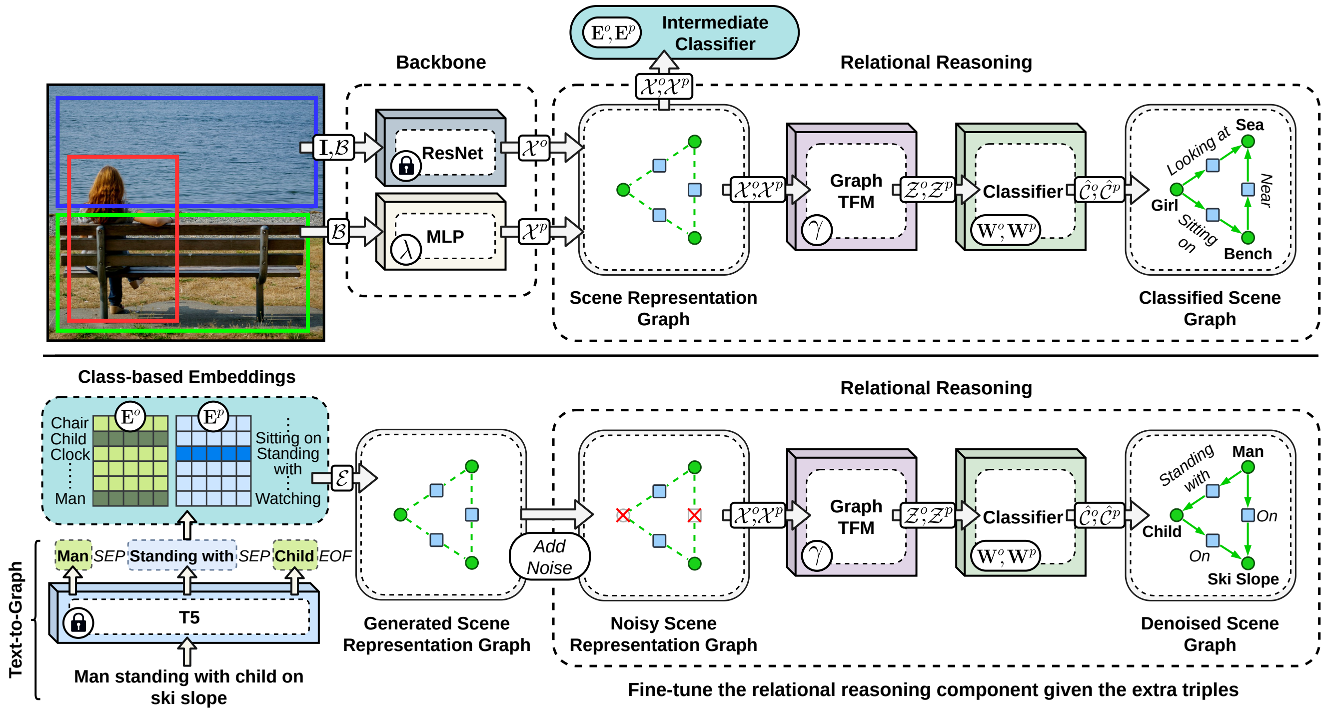

Figure 1: Top: we initially train a scene graph classification pipeline from images and their corresponding SGs. Bottom: we then use a text-to-graph module to extract structured knowledge from unstructured texts. The extracted graph is embedded by image-grounded vectors, masked, and then fed to the relational reasoning module to predict the missing relations and thus, encourage the network to learn the new relations from texts. The *lock* sign indicates pre-trained and frozen parts of the network.

available in the unstructured form of texts and documents. Exploiting this form of knowledge, in addition to structured knowledge, can be significantly beneficial. To this end, we employ a transformer-based model to generate structured graphs from textual input and utilize them to improve the relational reasoning module.

In summary, we propose *Texema*, a scene graph classification pipeline that can be trained from the large corpora of unstructured knowledge. We evaluate our approach on the Visual Genome dataset. In particular, we show that we can fine-tune the reasoning component using textual scene descriptions instead of thousands of images. As a result, when using as little as ∼500 images (1% of the VG training data), we can achieve ∼3x more accurate results in object classification, ∼8x in scene graph classification and ∼1.5x in predicate classification compared to the supervised baselines. Additionally, in our ablation studies, we evaluate the performance of using different rule-based, LSTM-based, and transformed-based text-to-graph models.

Related Works

Scene Graph Classification: There is an extensive body of work on visual reasoning in general that includes different forms of reasoning [Wu, Lenz, and Saxena 2014, Deng et al. 2014, Hu et al. 2016, 2017, Santoro et al. 2017, Zellers et al. 2019]. Here, we mainly review the works that are focused on scene graph classification. Visual Relation Detection (VRD) [Lu et al. 2016] and the Visual Genome [Krishna et al. 2017] are the main datasets for this task. While the original papers on VRD and VG provide the baselines for scene graph classification by treating objects independently, several follow-up works contextualize the entities before classification. Iterative Message Passing (IMP) [Xu et al. 2017], Neural Motifs [Zellers et al. 2018] (NM), Graph R-CNN [Yang et al. 2018], and Schemata [Sharifzadeh, Baharlou, and Tresp 2021] proposed to propagate the image context using basic RNNs, LSTMs, graph convolutions, and graph transformers respectively. On the other hand, authors of VTransE [Zhang et al. 2017] proposed to capture relations by applying TransE [Bordes et al. 2013], a knowledge graph embedding model, on the visual embeddings, Tang et al. [2019] exploited dynamic tree structures to place the object in an image into a visual context. Chen et al. [2019a] proposed a multi-agent policy gradient method that frames objects into cooperative agents and then directly maximizes a graph-level metric as the reward. In tangent to those works, Sharifzadeh et al. [2021] proposed to enrich the input domain in scene graph classification by employing the predicted pseudo depth maps of VG images that were released as an extension called *VG-Depth*.

Commonsense in Scene Understanding: Several recent works have proposed to employ external or internal sources of knowledge to improve visual understanding [Wang, Ye,

Input	man standing with child on ski slope
Reference Graph (RG)	**(child, on, ski slope) (man, on, ski slope) (man, standing with, child)**
$R_{\text{text}\rightarrow\text{graph}}$	*(man, standing, child)*
SSGP	*(standing, with, child) (standing, on, slope)*
CopyNet (1%)	**(man, standing with, child)**
T5 (1%)	**(man, standing with, child)**
CopyNet (10%)	**(man, standing with, child)** *(child, on, slope)*
T5 (10%)	**(man, standing with, child) (child, on, ski slope)**

Table 1: An example of extracted triples from a given text input in VG, using different methods. Bold: correct ($\in$ RG). Italic: incorrect ($\notin$ RG). The results are computed using the respective official code bases of the related works.

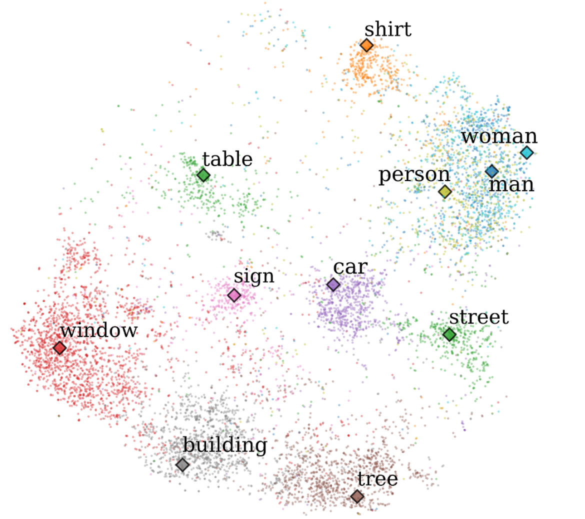

Figure 2: The t-SNE representation of the $\mathbf{e}_i$s (diamonds) and image-based representations $\mathcal{X}$s (dots) where each color represents the ground-truth class of the dot.

and Gupta 2018, Jiang et al. 2018, Singh et al. 2018, Kato, Li, and Gupta 2018]. In the scene graph classification domain, some of the works have proposed to correct the SG prediction errors by merely comparing them to the co-occurrence statistics of internal triples as a form of commonsense knowledge [Chen et al. 2019c,b, Zellers et al. 2018]. Earlier, Baier, Ma, and Tresp [2017, 2018] proposed the first scene graph classification model that employed prior knowledge in the form of Knowledge Graph Embeddings (KGEs) that generalize beyond the given co-occurrence statistics. Zareian, Karaman, and Chang [2020], Zareian et al. [2020] followed this approach by extending it to models that are based on graph convolutional networks. More recently, Sharifzadeh, Baharlou, and Tresp [2021] proposed Schemata as a generalized form of a KGE model that is learned directly from the images rather than triples. In general, scene graph classification methods are closely related to the KGE models. Therefore, we refer the interested readers to [Nickel et al. 2016, Ali et al. 2020a,b] for a review and large-scale study on the KG models, and to [Tresp, Sharifzadeh, and Konopatzki 2019, Tresp et al. 2020] for an extensive investigation of the connection between perception, KG models, and cognition.

Nevertheless, to the best of our knowledge, the described methods have employed curated knowledge in the form of triples, and none of them have directly exploited the textual knowledge. In this direction, the closest work to ours is by Yu et al. [2017], proposing to distill the external language knowledge using a teacher-student model. However, this work does not include a relational reasoning component and only refines the final predictions. Also, as shown in the experiments, our knowledge extraction module performs two times better than the SG Parser used in that work.

Knowledge Extraction from Text: Knowledge extraction from text has been studied for a long time [Chinchor 1991]. Previous work ranges from pattern-based approaches [Hearst 1992] to supervised neural approaches with specialized architectures [Gupta et al. 2019, Yaghoobzadeh, Adel, and Schütze 2017]. Recently, Schmitt et al. [2020] successfully applied a general sequence-to-sequence architecture to graph↔text conversion. With the recent rise of transfer learning in NLP, an increasing number of approaches are based on large language models, pre-trained in a self-supervised manner on massive amounts of texts [Devlin et al. 2019]. Inspired from previous work that explores transfer learning for graph-to-text conversion [Ribeiro et al. 2020], we base our text-to-graph model on a pre-trained T5 model [Raffel et al. 2019].

Methods

In this section, we first describe the backbone and relational reasoning components. We then describe our approach for fine-tuning the network from texts. We have three possible forms of data: Images (**IM**), Scene Graphs (**SG**) and Textual Scene Descriptions (**TXT**). We consider having two sets of data: one is the *parallel* set, which is the set of IM with their corresponding SG and TXT, and another is the *text* set which is a set of additional TXT that come without any images or scene graphs. These two sets have no elements in common.

We initially train our backbone and relational reasoning component from IM and SG, and our text-to-graph model from the TXT and SG in the parallel set. We then show that we can fine-tune the pipeline using the text set and without using any additional images.

Backbone (Algorithm 1.1)

The feature-extraction backbone is a convolutional neural network (ResNet-50) that has been pre-trained in a self-supervised manner [Grill et al. 2020] from unlabeled images of ImageNet [Deng et al. 2009] and Visual Genome [Krishna et al. 2017]. Given an image $\mathbf{I}$ with several objects in bounding boxes $\mathcal{B} = \{\mathbf{b}_i\}_{i=1}^n$, $\mathbf{b}_i = [b_i^x, b_i^y, b_i^w, b_i^h]$, we apply the ResNet-50 to extract pooled object features $\mathcal{X}^o = \{\mathbf{x}_i^o\}_{i=1}^n$, $\mathbf{x}_i^o \in \mathbb{R}^d$. Here $[b_i^x, b_i^y]$ are the coordinates of $\mathbf{b}_i$ and $[b_i^w, b_i^h]$ are its width and height, and d

Algorithm 1: Classify objects/predicates from images

1. **Extract image features (Backbone):**

   Input: Images and object bounding boxes $(\mathbf{I}, \mathcal{B} : \{\mathbf{b_i}\}_{i=1}^{n})$.

   Output: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^n$ and predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^m$.

   Trainable params: λ.

   $\mathcal{X}^o = ResNet50(\mathbf{I}, \mathcal{B})$

   $\mathcal{X}^p = \{MLP_\lambda(t(\mathbf{b}_i, \mathbf{b}_j)) \mid \forall \mathbf{b}_i, \mathbf{b}_j \in \mathcal{B}\}$

2. **Contextualize and Classify (Relational Reasoning):**

   Input: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^n$, Predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^m$ and ground truth classes $\mathcal{C}^o$ and $\mathcal{C}^p$.

   Output: Predicted object class distribution $\hat{\mathcal{C}}^o : \{\hat{\mathbf{c}}_i^o\}_{i=1}^n$ and predicted predicate class distribution $\hat{\mathcal{C}}^p : \{\hat{\mathbf{c}}_i^p\}_{i=1}^m$.

   Trainable params: $\gamma, \mathbf{W}^o, \mathbf{W}^p$.

   $\mathcal{Z}^o, \mathcal{Z}^p = GraphTransformer_\gamma(\mathcal{X}^o, \mathcal{X}^p)$

   $\hat{\mathcal{C}}^o = \{\text{softmax}(\mathbf{W}^o \cdot \mathbf{z}^o) \mid \forall \mathbf{z}^o \in \mathcal{Z}^o\}$

   $\hat{\mathcal{C}}^p = \{\text{softmax}(\mathbf{W}^p \cdot \mathbf{z}^p) \mid \forall \mathbf{z}^p \in \mathcal{Z}^p\}$

3. **Apply Loss (Cross-Entropy):**

   $l_o = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\|\mathbf{c}_i^o\|} \mathbf{c}_{i,j}^o . log(\hat{\mathbf{c}}_{i,j}^o)$

   $l_p = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{\|\mathbf{c}_i^p\|} \mathbf{c}_{i,j}^p . log(\hat{\mathbf{c}}_{i,j}^p)$

are the vector dimensions. Following [Zellers et al. 2018], we define $\mathcal{X}^p = \{\mathbf{x}_i^p\}_{i=1}^m$, $\mathbf{x}_i^p \in \mathbb{R}^d$ as the relational features between each pair of objects. Each $\mathbf{x}_i^p$ is initialized by applying a two layered fully connected network on the relational position vector $\mathbf{t}$ between a head i and a tail j where $\mathbf{t} = [t_x, t_y, t_w, t_h]$, $t_x = (b_i^x - b_j^x)/b_{i\ j}^w, t_y = (b_i^y - b_j^y)/b_j^h, t_w = \log(b_i^w/b_j^w), t_h = \log(b_i^h/b_j^h)$. The implementation and pre-training details of the layers are provided in the Evaluation. $\mathcal{X}^o$ and $\mathcal{X}^p$ form a structured presentation of the objects and predicates in the image also known as **Scene Representation Graph** (SRG) [Sharifzadeh, Baharlou, and Tresp 2021]. SRG is a fully connected graph with each node representing either an object or a predicate, where each object node is a direct neighbor to predicate nodes and each predicate node is a direct neighbor with its head and tail object nodes.

Relational Reasoning (Algorithm 1.2)

The relational reasoning component updates the initial SRG representations through Graph Transformer layers [Koncel-Kedziorski et al. 2019]. The outputs of these layers are $\mathcal{Z}^o = \{\mathbf{z}_i^o\}_{i=1}^n$, $\mathbf{z}_i^o \in \mathbb{R}^d$ and $\mathcal{Z}^p = \{\mathbf{z}_i^p\}_{i=1}^m$, $\mathbf{z}_i^p \in \mathbb{R}^d$ with equal dimensions as $\mathcal{X}$s. From here on, we drop the superscripts of o and p for brevity. We apply a linear classification layer $\mathbf{W}$ to classify the contextualized representations $\mathcal{Z}$ such that $\hat{\mathbf{c}} = \text{softmax}(\mathbf{W} \cdot \mathbf{z}_i)$, with cross-entropy as the loss function.

Fine-tuning from Texts (Algorithm 2)

Let us assume that we have already trained the backbone and relational reasoning components from IM and SG in the *parallel* set. Now, we want to fine-tune the weights in the

Algorithm 2: Fine-tune the relational reasoning component from textual triples using a denoising auto-encoder paradigm

1. **Learn image-grounded representations E for each symbol through classification (without Graph Transformer):**

   Input: Object embeddings $\mathcal{X}^o : \{\mathbf{x}_i^o\}_{i=1}^n$, predicate embeddings $\mathcal{X}^p : \{\mathbf{x}_i^p\}_{i=1}^m$ and their corresponding ground truth classes $\mathcal{C}^o$ and $\mathcal{C}^p$.

   Output: Predicted object class distribution $\hat{\mathcal{C}}^o : \{\hat{\mathbf{c}}_i^o\}_{i=1}^n$ and predicted predicate class distribution $\hat{\mathcal{C}}^p : \{\hat{\mathbf{c}}_i^p\}_{i=1}^m$.

   Trainable params: $\mathbf{E}^o, \mathbf{E}^p$.

   $\hat{\mathcal{C}}^o = \{\text{softmax}(\mathbf{E}^o \cdot \mathbf{x}^o) \mid \forall \mathbf{x}^o \in \mathcal{X}^o\}$

   $\hat{\mathcal{C}}^p = \{\text{softmax}(\mathbf{E}^p \cdot \mathbf{x}^p) \mid \forall \mathbf{x}^p \in \mathcal{X}^p\}$

2. **Apply Loss (Cross Entropy):**

   $l_o = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\|\mathbf{c}_i^o\|} \mathbf{c}_{i,j}^o . log(\hat{\mathbf{c}}_{i,j}^o)$

   $l_p = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{\|\mathbf{c}_i^p\|} \mathbf{c}_{i,j}^p . log(\hat{\mathbf{c}}_{i,j}^p)$

3. **Fine-tune the relational reasoning component given the extra triples (Denoising Graph Autoencoder):**

   Input: Symbolic triples $\mathcal{S} : \{(h_i, p_i, t_i)\}_{i=1}^k$ and canonical object/predicate representations $\mathbf{E}^o$/$\mathbf{E}^p$.

   Output: Embedded representations $\mathcal{E} : \{(\mathbf{e}_i^h, \mathbf{e}_i^p, \mathbf{e}_i^t)\}_{i=1}^k$.

   Trainable params: $\gamma, \mathbf{W}^o, \mathbf{W}^p$.

   - Build $\mathcal{E} : \{(\mathbf{e}_i^h, \mathbf{e}_i^p, \mathbf{e}_i^t)\}_{i=1}^k$ where for each (h_i, p_i, t_i):
     $\mathbf{e}_i^h = onehot(h_i) \cdot \mathbf{E}^o$
     $\mathbf{e}_i^p = onehot(p_i) \cdot \mathbf{E}^p$
     $\mathbf{e}_i^t = onehot(t_i) \cdot \mathbf{E}^o$
   - Randomly set 20% of the nodes and edges in $\mathcal{E}$ to zero.
   - Set $\mathcal{X}^o = \mathcal{E}^h \cup \mathcal{E}^t$ and $\mathcal{X}^p = \mathcal{E}^p$ and run Algorithm 1.2 to fine-tune $\gamma, \mathbf{W}^o, \mathbf{W}^p$, with $\mathcal{E}^h$, $\mathcal{E}^t$ and $\mathcal{E}^p$ as the set of all heads, tails, and predicates in $\mathcal{E}$.

relational reasoning component given the additional *text* set. The relational reasoning component takes graphs as input, therefore, we first need to convert TXT to SG:

Text-to-graph: This model is trained from the SG and TXT in the parallel set, and then used to generate SG from the text set. Let us consider an unstructured text such as "man standing with child on ski slope" (Table 1 - Input). A structured form of this sentence is a graph with unique nodes and edges for each entity or predicate. For example, the reference graph for this sentence contains the triples `(child, on, ski slope)`, `(man, standing with, child)` and `(man, on, ski slope)` (Table 1 - RG).

In order to learn this mapping, we employ a transformer-based [Vaswani et al. 2017] sequence-to-sequence T5_{small} model [Raffel et al. 2019] and adapt it for the task of extracting graphs from texts. T5 consists of an encoder with several layers of self-attention (like BERT, Devlin et al. 2019) and a decoder with autoregressive self-attention (like GPT-3, Brown et al. 2020). In order to use a T5 model with graphs, we need to represent the graphs as a sequence. To this end, we serialize the graphs by writing out their facts separated

Method	Precision		Recall		F1	
	1%	10%	1%	10%	1%	10%
$\mathrm{R}_{\text{text}\rightarrow\text{graph}}$	1.92 ± 0.00	1.86 ± 0.01	1.87 ± 0.00	1.81 ± 0.01	1.89 ± 0.00	1.84 ± 0.01
SSGP	14.86 ± 0.01	14.52 ± 0.02	18.47 ± 0.01	18.05 ± 0.02	16.47 ± 0.01	16.09 ± 0.02
CopyNet	29.20 ± 0.13	30.77 ± 0.49	27.19 ± 0.28	29.79 ± 0.29	28.16 ± 0.21	30.27 ± 0.34
T5	$\mathbf{33.37 \pm 0.11}$	$\mathbf{33.81 \pm 0.08}$	$\mathbf{31.06 \pm 0.18}$	$\mathbf{32.45 \pm 0.33}$	$\mathbf{32.17 \pm 0.13}$	$\mathbf{33.12 \pm 0.16}$

Table 2: The mean and standard deviation of Precision, Recall, and F1 scores of the predicted facts from the texts on four random splits. The results are computed using the respective official code bases of the related works and evaluated on VG.

by end-of-fact symbols (`EOF`), and separate the elements of each fact with `SEP` symbols [Schmitt et al. 2020], e.g. *"child SEP on SEP ski slope EOF"* (Fig. 1). To adapt the multi-task setting from T5's pretraining, we use the task prefix "make graph: " to mark our text-to-graph task. Table 1 shows an example text and the extracted graphs using T5 and other previous methods (see Evaluation for details).

Map to embeddings: Note that the predicted graphs are a sequence of symbols for heads, predicates, and tails where each symbol represents a class $c \in \mathcal{C}$. However, the inputs to the relational reasoning component are image-based vectors $\mathcal{X}$. Thus, before feeding the symbols to the relational reasoning component, we need to map them to a corresponding embedding from the space of $\mathcal{X}$ as if we are feeding it with image-based embeddings. In order to do that, we train a mapping from symbols to $\mathcal{X}$s using the IM and SG of the parallel set. This is simply done by training a linear classification layer $\mathbf{E}$ given $\mathcal{X}$s from the parallel set (Algorithm 2.1). Unlike the classification layer in Algorithm 1, here we classify $\mathcal{X}$s instead of $\mathcal{Z}$s and the goal is *not* to use the classification output but to train image-grounded, canonical representations for each class: each row $\mathbf{e}_i$ in the classification layer becomes a cluster center for $\mathcal{X}$s from class i (Figure 2). Therefore, instead of the extracted symbolic c_i from the text set, we can feed its canonical image-grounded representation $\mathbf{e}_i$ to the graph transformer (Algorithm 2.3).

Denoising Graph Autoencoder: To fine-tune the relational reasoning given this data, we treat the relational reasoning component as a denoising autoencoder where the input is an incomplete (noisy) graph that comes from the text and the output is the denoised graph. If we do not apply a denoising autoencoder paradigm, the function will collapse to an identity map. We create the noisy graph by randomly setting some of the input nodes and edges to zero during the training (Algorithm 2.3). The goal is to encourage the graph transformer to predict the missing links and therefore, learn the relational structure.

Evaluation

We first compare the performance of different rule-based and embedding-based text-to-graphs models on our data. We then evaluate the performance of our entire pipeline in classifying objects and relations in images. In particular, we show that the extracted knowledge from the texts can largely substitute annotated images as well as ground-truth graphs.

Dataset: We use the sanitized version [Xu et al. 2017] of Visual Genome (VG) dataset [Krishna et al. 2017] including images and their annotations, i.e., bounding boxes, scene graphs, and scene descriptions. Our goal is to design an experiment that evaluates whether we can substitute annotated images with textual scene descriptions. Therefore, instead of using external textual datasets with unbounded information, we use Visual Genome itself by dividing it into different splits of *parallel* (with IM, SG and TXT) and *text* data (with only TXT). To this end, we assume only a random proportion (1% or 10%) of training images are annotated (parallel set containing IM with corresponding SG and TXT). We consider the remaining data (99% or 90%) as our text set and discard their IM and SG. We aim to see whether employing TXT from the text set, can substitute the discarded IM and SG from this set. We use four different random splits [Sharifzadeh, Baharlou, and Tresp 2021] to avoid a sampling bias. For more detail on the datasets refer to the supplementary materials.

Note that the scene graphs and the scene descriptions from the VG are collected separately and by crowd-sourcing. Therefore, even though the graphs and the scene descriptions refer to the same image region, they are disjoint and contain complementary knowledge.

Graphs from Texts

The goal of this experiment is to study the effectiveness of the text-to-graph model. We fine-tune the pre-trained T5 model on parallel TXT and SG, and apply it on the text set to predict their corresponding SG. We also implement the following rule-based and embedding-based baselines to compare their performance using our splits: (1) $\mathrm{R}_{\text{text}\rightarrow\text{graph}}$ is a simple rule-based system introduced by Schmitt et al. [2020] for general knowledge graph generation from text. (2) The Stanford Scene Graph Parser (SSGP) [Schuster et al. 2015] is another rule-based approach that is more adapted to the scene graph domain. Even though this approach was not specifically designed to match the scene graphs from the Visual Genome dataset, it was still engineered to cover typical idiosyncrasies of textual image descriptions and corresponding scene graphs. (3) CopyNet [Gu et al. 2016] is an LSTM sequence-to-sequence model with a dedicated copy mechanism, which allows copying text elements directly into the graph output sequence. It was used for unsupervised text-to-graph generation by Schmitt et al. [2020]. However, we train it on the supervised data of our parallel sets. We use a vocabulary of around 70k tokens extracted from the VG-graph-text

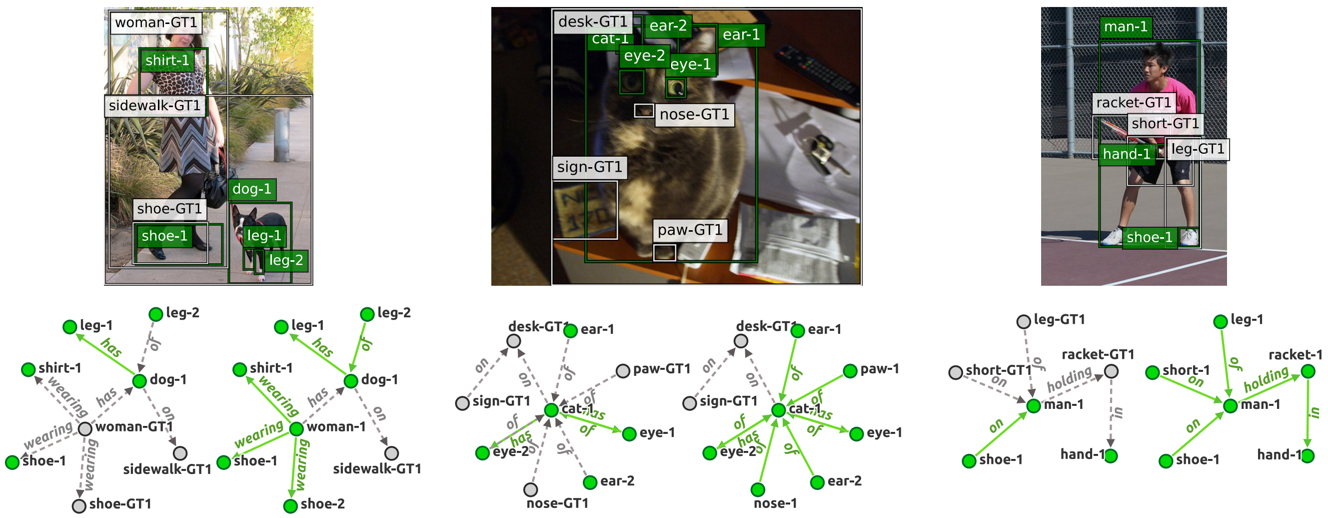

Figure 3: Qualitative examples of improved classification results (Recall@100) before and after (from left to right) fine-tuning the model using the knowledge in texts. Green and gray colors indicate true positives and false negatives concluded by the model.

	Method	R@50		R@100	
		1%	10%	1%	10%
SGCls	$R_{\text{text}\rightarrow\text{graph}}$	10.90 ± 0.12	24.96 ± 0.15	11.80 ± 0.11	26.09 ± 0.15
	SSGP	14.35 ± 0.15	26.11 ± 0.19	15.14 ± 0.17	27.12 ± 0.22
	CopyNet	14.46 ± 0.31	26.05 ± 0.29	15.19 ± 0.24	27.08 ± 0.26
	TXM - T5	$\mathbf{14.53 \pm 0.34}$	$\mathbf{26.16 \pm 0.32}$	$\mathbf{15.28 \pm 0.38}$	$\mathbf{27.22 \pm 0.28}$
	GT	14.72 ± 0.38	26.33 ± 0.45	15.36 ± 0.38	27.37 ± 0.47
PredCls	$R_{\text{text}\rightarrow\text{graph}}$	23.34 ± 0.10	49.99 ± 0.12	26.83 ± 0.15	54.40 ± 0.12
	SSGP	54.65 ± 0.14	55.65 ± 0.15	59.33 ± 0.18	59.67 ± 0.20
	CopyNet	56.24 ± 0.31	59.27 ± 0.28	60.35 ± 0.20	63.28 ± 0.25
	TXM - T5	$\mathbf{58.64 \pm 0.34}$	$\mathbf{59.31 \pm 0.30}$	$\mathbf{63.07 \pm 0.37}$	$\mathbf{63.32 \pm 0.24}$
	GT	62.02 ± 0.10	61.71 ± 0.19	65.68 ± 0.12	65.42 ± 0.19

Table 3: SGCls and PredCls results using different text-to-graph modules. We have substituted the missing 99% and 90% of annotated images with the textual knowledge extracted from their scene descriptions.

benchmark and, otherwise, also adopt the hyperparameters from [Schmitt et al. 2020]. Table 1 shows sample predictions from these models. Table 2 compares precision, recall, and F1 measures. and T5 outperforms others by a large margin.

Graphs from Images

The goal of this experiment is to evaluate scene graph classification after fine-tuning the pipeline using textual knowledge. We evaluate our models for object classification, predicate classification (PredCls - predicting predicate labels given a ground truth set of object boxes and object labels) and scene graph classification (SGCls - predicting object and predicate labels, given the set of object boxes) on the test sets. Our ablation study concerns the following configurations:

- **SPB**: In this setting, both the backbone and the relational reasoning component are trained by *supervised learning* on the IM and SGs (1% or 10%) from the parallel set.
- **SCH**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set.
- **TXM**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set and fine-tuned from the SGs predicted from the text set (99% or 90%) using the text-to-graph module.
- **GT**: Here, the backbone is trained by *self-supervised learning* on all VG images (without labels), and the relational reasoning component is trained on the IM and SGs (1% or 10%) from the parallel set, and fine-tuned from the *ground truth graphs* (99% or 90%), instead of the text-to-graph predictions.

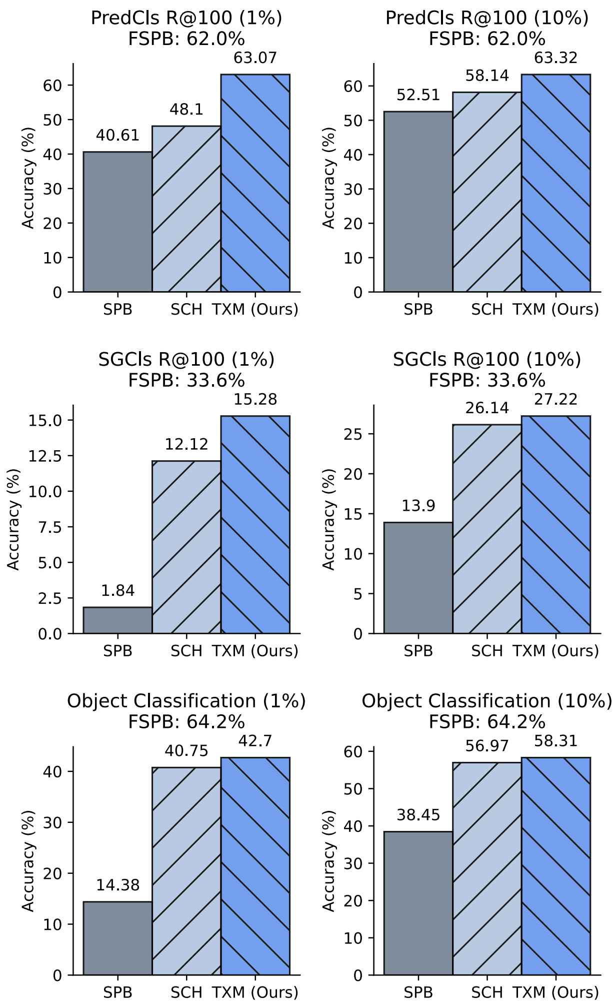

Figure 4: Fine-tuning with the textual knowledge (TXM) significantly improves the results in all settings of PredCls (top), SGCls (middle), and object classification (bottom).

- **FSPB**: Here, both the backbone and the relational reasoning component are trained by *supervised learning* on 100% of the VG annotated images. Meaning that we have redefined the parallel set to include 100% of the VG training data and we do not need to substitute the images with the text set anymore. The goal of this setting is to compute the maximum accuracy that our model achieves, when we have all the annotated images with ground truth SGs, instead of using their textual scene descriptions. The results of this settings are written above each table so that the other bars maintain a meaningful scale.

Figure 4 presents the results of the ablation study. We use the Recall@K (**R@K**) as metric. which computes the mean prediction accuracy in each image given the top K predictions. For more results (mR@K metric and unconstrained setups) refer to the supplementary materials. As shown, fine-tuning with textual scene descriptions (TXM) improves the classification results under all settings, substituting a large proportion of the omitted images. Furthermore, the results even outperform FSPB under PredCls (recall that the scene descriptions are sometimes complementary to image annotations and contain additional information).

Method	SGCls		PredCls	
	R@50	R@100	R@50	R@100
VRD [Lu et al. 2016]	11.8	14.1	27.9	35.0
IMP+ [Xu et al. 2017]	34.6	35.4	59.3	61.3
SMN [Zellers et al. 2018]	35.8	36.5	65.2	67.1
KERN [Chen et al. 2019c]	36.7	37.4	65.8	67.6
VCTree [Tang et al. 2019]	38.1	38.8	66.4	68.1
CMAT [Chen et al. 2019a]	39.0	39.8	66.4	68.1
SIG [Wang et al. 2020]	36.6	37.3	66.3	68.1
GB-Net [Zareian et al. 2020]	38.0	38.8	66.6	68.2
TXM	**39.0**	**39.9**	**66.7**	**68.3**

Table 4: Comparing the general performance of the architecture to some other methods under the VG test set.

Table 3 presents additional results also using different text-to-graph baselines. We can see that fine-tuning with the predicted graphs using T5, is as effective as fine-tuning with the crowd-sourced ground truth graphs (GT), and in some settings even better (object classification with 1%). Notice that compared to the self-supervised baseline, we gained up to $\sim$5% *relative improvement* in object classification, more than $\sim$26% in scene graph classification, and $\sim$31% in predicate prediction accuracy. As expected, the choice of text-to-graph module has a larger effect on the PredCls compared to the SGCls and ObjCls, due to the fact that SGCls and ObjCls rely heavily on the image-based features, whereas PredCls has a strong dependency to relational knowledge. In supplementary materials we also provide additional results on the improvements per object class after fine-tuning the model with the textual knowledge (From SCH to TXM) and show that most improvements occur in under-represented classes. Figure 3 provides some qualitative examples of the predicted scene graphs before and after fine-tuning with the texts. Finally, to provide an intuition on our general performance, Table 4 present the results of our architecture using a VGG-16 [Simonyan and Zisserman 2014] backbone trained with 100% of the annotations, instead of the self-supervised BYOL.

Conclusion

In this work, we proposed the first relational image-based classification pipeline that can be fine-tuned directly from the large corpora of unstructured knowledge available in texts. We generated structured graphs from textual input using different rule-based or embedding-based approaches. We then fine-tuned the relational reasoning component of our classification pipeline by employing the canonical representations of each entity in the generated graphs. We showed that we gain a significant improvement in all settings after employing the inferred knowledge within the classification pipeline. In most cases, the accuracy was similar to when using the ground truth graphs that are manually annotated by crowd-sourcing.

Acknowledgments

We would like to thank Masoud Jalili Sabet for the fruitful discussions, and the anonymous reviewers for their helpful feedback on the manuscript. This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A.

References

Ali, M.; Berrendorf, M.; Hoyt, C. T.; Vermue, L.; Galkin, M.; Sharifzadeh, S.; Fischer, A.; Tresp, V.; and Lehmann, J. 2020a. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *arXiv preprint arXiv:2006.13365*.

Ali, M.; Berrendorf, M.; Hoyt, C. T.; Vermue, L.; Sharifzadeh, S.; Tresp, V.; and Lehmann, J. 2020b. Pykeen 1.0: A python library for training and evaluating knowledge graph emebddings. *arXiv preprint arXiv:2007.14175*.

Baier, S.; Ma, Y.; and Tresp, V. 2017. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*, 53–68. Springer.

Baier, S.; Ma, Y.; and Tresp, V. 2018. Improving information extraction from images with learned semantic models. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 5214–5218. AAAI Press.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *Computing Research Repository*, arXiv:2005.14165.

Chen, L.; Zhang, H.; Xiao, J.; He, X.; Pu, S.; and Chang, S.-F. 2019a. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4613–4623.

Chen, T.; Xu, M.; Hui, X.; Wu, H.; and Lin, L. 2019b. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 522–531.

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019c. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6163–6171.

Chinchor, N. 1991. MUC-3 Linguistic Phenomena Test Experiment. In *Third Message Uunderstanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

Deng, J.; Ding, N.; Jia, Y.; Frome, A.; Murphy, K.; Bengio, S.; Li, Y.; Neven, H.; and Adam, H. 2014. Large-scale object classification using label relation graphs. In *European conference on computer vision*, 48–64. Springer.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640. Berlin, Germany: Association for Computational Linguistics.

Gupta, P.; Rajaram, S.; Schütze, H.; and Runkler, T. A. 2019. Neural Relation Extraction within and across Sentence Boundaries. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6513–6520.

Hearst, M. A. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Hu, H.; Deng, Z.; Zhou, G.-T.; Sha, F.; and Mori, G. 2017. Labelbank: Revisiting global perspectives for semantic segmentation. *arXiv preprint arXiv:1703.09891*.

Hu, H.; Zhou, G.-T.; Deng, Z.; Liao, Z.; and Mori, G. 2016. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2960–2968.

Jiang, C.; Xu, H.; Liang, X.; and Lin, L. 2018. Hybrid knowledge routed modules for large-scale object detection. *arXiv preprint arXiv:1810.12681*.

Kato, K.; Li, Y.; and Gupta, A. 2018. Compositional learning for human object interaction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 234–251.

Koncel-Kedziorski, R.; Bekal, D.; Luan, Y.; Lapata, M.; and Hajishirzi, H. 2019. Text generation from knowledge graphs with graph transformers. *arXiv preprint arXiv:1904.02342*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al.

2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1): 32–73.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 852–869. Springer.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1): 11–33.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Ribeiro, L. F. R.; Schmitt, M.; Schütze, H.; and Gurevych, I. 2020. Investigating Pretrained Language Models for Graph-to-Text Generation. *Computing Research Repository*, arXiv:2007.08426.

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, 4967–4976.

Schmitt, M.; Sharifzadeh, S.; Tresp, V.; and Schütze, H. 2020. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7117–7130.

Schuster, S.; Krishna, R.; Chang, A.; Fei-Fei, L.; and Manning, C. D. 2015. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, 70–80.

Sharifzadeh, S.; Baharlou, S. M.; Berrendorf, M.; Koner, R.; and Tresp, V. 2021. Improving Visual Relation Detection using Depth Maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3597–3604.

Sharifzadeh, S.; Baharlou, S. M.; and Tresp, V. 2021. Classification by Attention: Scene Graph Classification with Prior Knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 5025–5033.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, K. K.; Divvala, S.; Farhadi, A.; and Lee, Y. J. 2018. Dock: Detecting objects by transferring common-sense knowledge. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 492–508.

Tang, K.; Zhang, H.; Wu, B.; Luo, W.; and Liu, W. 2019. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6619–6628.

Tresp, V.; Sharifzadeh, S.; and Konopatzki, D. 2019. A Model for Perception and Memory. *Conference on Cognitive Computational Neuroscience*.

Tresp, V.; Sharifzadeh, S.; Konopatzki, D.; and Ma, Y. 2020. The Tensor Brain: Semantic Decoding for Perception and Memory. *arXiv preprint arXiv:2001.11027*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, W.; Wang, R.; Shan, S.; and Chen, X. 2020. Sketching image gist: Human-mimetic hierarchical scene graph generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, 222–239. Springer.

Wang, X.; Ye, Y.; and Gupta, A. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6857–6866.

Wu, C.; Lenz, I.; and Saxena, A. 2014. Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception. In *Robotics: Science and systems*.

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5419.

Yaghoobzadeh, Y.; Adel, H.; and Schütze, H. 2017. Noise Mitigation for Neural Entity Typing and Relation Extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1183–1194. Valencia, Spain: Association for Computational Linguistics.

Yang, J.; Lu, J.; Lee, S.; Batra, D.; and Parikh, D. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 670–685.

Yu, R.; Li, A.; Morariu, V. I.; and Davis, L. S. 2017. Visual relationship detection with internal and external linguistic knowledge distillation. In *IEEE International Conference on Computer Vision (ICCV)*.

Zareian, A.; Karaman, S.; and Chang, S.-F. 2020. Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision*, 606–623. Springer.

Zareian, A.; You, H.; Wang, Z.; and Chang, S.-F. 2020. Learning Visual Commonsense for Robust Scene Graph Generation. *arXiv preprint arXiv:2006.09623*.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6720–6731.

Zellers, R.; Yatskar, M.; Thomson, S.; and Choi, Y. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5831–5840.

Zhang, H.; Kyaw, Z.; Chang, S.; and Chua, T. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 3107–3115. IEEE Computer Society. ISBN 978-1-5386-0457-1.