TranstextNet: Transducing Text for Recognizing Unseen Visual Relationships

Gal S.Kenigsfield
sgalk87@campus.technion.ac.il

Ran El-Yaniv
rani@cs.technion.ac.il

Abstract

An important challenge in visual scene understanding *is the recognition of interactions between objects in an image. This task – often called visual relationship detection (VRD) – must be solved to enable higher understanding of the semantic content in images. VRD can become particularly hard where there is severe statistical sparsity of some potentially involved objects, and the number of many relationships in standard training sets is limited. In this paper we show how to transduce auxiliary text so as to enable recognition of relationships absent in the visual training data. This transduction is performed by learning a shared relationship representation for both the textual and visual information. The proposed approach is* model-agnostic *and can be used as a plug-in module in existing VRD and* scene graph generation *(SGG) recognition systems to improve their performance and extend their capabilities. We consider the application of our technique using three widely accepted SGG models [20, 24, 16], and different auxiliary text sources: image captions, text generated by a deep text generation model (GPT-2), and ebooks from the Gutenberg Project. We conduct an extensive empirical study of both the VRD and SGG tasks over large-scale benchmark datasets. Our method is the first to enable recognition of visual relationships missing in the visual training data and appearing only in the auxiliary text. We conclusively show that text ingestion enables recognition of unseen visual relationships, and moreover, advances the state-of-the-art in all SGG tasks.*

1. Introduction

Scene graph generation (SGG) [20, 24] is the task of inferring a graph given an image. The SGG task, which relies on both computer vision and natural language understanding, belongs to a family of tasks that requires abstract capabilities that are deeper and much more challenging than standard image classification or tracking/detection tasks [5, 6]. A *scene graph* (SG) is a topological structure of a scene where the nodes represent the objects and the edges represent relationships between pairs of objects. Inferring an SG allows the extraction of information from the image (e.g., regional descriptions, global descriptions, labels etc.). For example, in Figure 1 we see the SG of an image containing four relationships among five objects. A straightforward approach to generate an SG is to decompose the task into subtasks such that the SG is assembled from a set of inferred relationships between all object pairs in the image. This subtask of inferring the interaction or relationship between a pair of given objects is called *visual relationship detection* (VRD). To solve SGG/VRD one must surmount three challenges: (1) sparse relationships between objects; given N objects in an image, there are $N \times (N-1)$ possible (non-symmetric) relationships; (2) detrimental training bias, where more frequent relationships dominate others, e.g., `on` dominates `above`, and `standing on`; and (3) models are less relevant in real-world scenarios because the systems are trained on a small relationship sets.

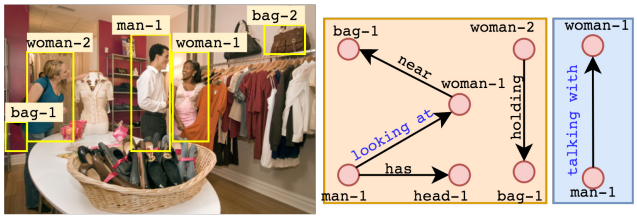

Figure 1: Left: An image from VG. Right: A corresponding scene graph. The relationship recognized between `man-1` and `woman-1` is `looking at`. The more informative relationship we would like to infer, `talking with`, is absent from the training data.

In Figure 1, we demonstrate problems (1), (2) and (3). The localized objects are `man-1`, `woman-1`, `woman-2`, `bag-1`, and `bag-2`. A system designed to perform VRD would note all 20 possible interactions even though it is obvious that `man-1` and `bag-1` do not interact. When observing the objects `man-1` and `woman-1`, the relation-

ships `talking with` and `looking at` are both reasonable predictions, but `looking at` exists in the visual training data while `talking with` does not, making it impossible for a standard SGG system to recognize it.

To address these issues, in this paper we introduce a method that utilizes auxiliary text and enables: (1) recognition of unseen visual relationships, and (2) better recognition of less frequent relationships. We show, for the first time, how to successfully ingest auxiliary text for improving SGG and VRD. We also identify three points that are key to ensuring successful utilization of auxiliary text for VRD and SGG: (1) the text should be related to visual descriptions, such as image captions or prose, (2) the fusion of the text should be implicit and not rely solely on conditional statistics extracted from the text, and (3) the text must be utilized to learn representations that are vital for relationship recognition. We note that the utilization of auxiliary text for VRD was already considered by [23] who used subject-relationship-object statistics of text parsed from Wikipedia, as well as a teacher-student architecture to distill knowledge from the text. Their attempt, however, was not successful and their work indicated that parsing text from Wikipedia *does not improve the results on VRD*. The authors suspected that this failure was due to noisiness of the data acquired from Wikipedia.

The first step in our pipeline is to distill text describing images or visual scenes, and parse it into a subject-relationship-object representation. In the second step, we assemble an object-relationship mapping from the parsed text, where we define an object-relationship mapping as a function from a pair of objects to a set of predicates. Finally, in the third step, we employ a neural *fusion mechanism* that combines the information from the parsed text with the visual features. By utilizing the parsed text, we enable SGG models to recognize relationships even if they appear less frequently or are completely absent from the training data.

We present an extensive empirical study to evaluate our model and conclude that the infusion of auxiliary text enables: (1) recognition of relationships that were absent from the relationship training set, thus enabling SGG systems to recognize larger sets of relationships, (2) better recognition of less frequent relationships, and (3) better recognition of subject-relationship-object triplets unseen during training. To conduct our study, we introduce a new visual relationship recognition task, which we call *recognition of unseen visual relationships (ROUVR)*. We support our claims by applying our text transduction model on three well-known and successful SGG models [20, 24, 16]. In all three cases, we demonstrate consistently good performance, indicating that the proposed model is effective, generic and model-agnostic.

2. Related Work

Visual Relationship Detection. Early studies in VRD tended to rely on data statistics [10], or adopt a joint model for subject-relationship-object triplets. For example, [9] showed how to tackle VRD using a relationship embedding space from the subject and object appearance model for VRD. [27] and [26] used visual embedding networks, which embed objects in a low-dimensional space and integrate them as context for VRD. [8] proposed a deep structural model integrating multiple cues to predict the relationships. All these models advocated a two-stage approach for VRD: first working on objects, and then working on relationships. [25] tackled an interesting problem of predicting undermined relationships, i.e., unlabeled positive relationships, by utilizing external linguistic features from Wikipedia. All these recent attempts demonstrated satisfactory success when confronting small relationship sets, e.g., 50/70 relationships in the VG/VRD datasets, respectively. In contrast, here we introduce a novel approach to integrating data from various sources and enable scaling VRD onto larger relationship sets without additional training.

Scene Graph Generation. Scene graphs were first introduced by [5], who utilized them for image retrieval. An SG is a topological description of a scene with the nodes corresponding to objects and the (directed) edges corresponding to the relationship between objects. An earlier approach was to detect all the objects in the scene and then utilize object appearances to detect relationships between objects [9]. [20] used graph-based inference to propagate information in both directions, between objects and relationships. [24] investigated recurring structures in VG-SG, and employed a global context network to predict the graphs. They also introduced a strong frequency baseline based on VG statistics. [4] proposed a permutation-invariant prediction model, and [2] proposed combining dataset statistics with a knowledge-embedded routing network. They also addressed the problem of biased models and proposed the *mean recall at K* (mR@k) metric to overcome this issue. [16] suggested a tree-LSTM architecture and hybrid reinforcement learning, which currently achieves the state-of-the-art (SOTA) results on SGG. [15] tackled the issue of unbiasing SGG models by proposing a causal graph approach and achieved impressive mR@K results, while compromising the R@K metric.

Fusing Text and of Visuals The many attempts to integrate language into vision have produced impressive

results [3, 21, 23]. [3] were the first to utilize modern language models for vision tasks. [21] used an encoder-decoder style architecture that utilized a CNN as the encoder and an LSTM with attention as a decoder for image captioning. [11] introduced feature-wise linear modulation as a mechanism that combines vision and language features for visual question answering (VQA). Presently, when considering both VRD and SGG, the most common approach to fusing visual and lingual features is simply to concatenate them [22, 23].

3. Problem Formulation

Following [24] and [2], we define an SG for a given image I as a directed graph $G_I \triangleq (O, R, B)$, where $O \triangleq \{o_1, o_2, \ldots, o_n\}$ is a set of (visual) objects appearing in I, $R \triangleq \{r_{1\to2}, r_{1\to3}, \ldots, r_{(n-1)\to n}\}$ is a set of directed edges representing (non-symmetric) relationships, potentially between all object pairs, and $B \triangleq \{b_1, b_2, \ldots, b_n\}$ is a set of bounding boxes, where $b_i \triangleq (x, y, w, h)$ is the bounding box of object o_i. The standard bounding box definition has (x, y) as the center coordinates of the box, and w, h its width and height, respectively. Setting $p(G|I) \triangleq p(B, O, R|I)$, we decompose the probability distribution $p(G_I|I)$ of the graph G_I into three components: $p(G_I|I) = p(B|I)p(O|B, I)p(R|O, B, I)$. This decomposition and the three components motivate three computation steps that are sufficient for assembling the SG. The first component, $p(B|I) = \Pi_{i=1}^{N} p(b_i|I)$, corresponds to the first step whereby the bounding boxes in the image are identified. Given these bounding boxes, the second component, $p(O|B, I) = \Pi_{i,j=1}^{N} p(o_i|b_i)$, corresponds to predicting class labels for the objects (within their bounding boxes). The third component, $p(R|O, B, I) = \Pi_{i=1}^{N} p(r_{i\to j}|o_i, o_j)$, corresponds to the last step where relationships are predicted for object pairs (namely, VRD). Following [20], [24], and [2], we consider a supervised structure learning approach to generating SGs. Given a set of training examples, $S_m \triangleq \{(I^{(i)}, SG^{(i)}), i = 1, \ldots, m\}$, where $I^{(i)}$ is an image and $SG^{(i)}$ is its corresponding SG, the goal is to train a model to predict SGs for unseen images. The common performance measure for both the VRD and SGG tasks is *recall at K* (R@K) [9], which computes the fraction of correctly predicted object-relationship-object triplets among the top-K confident predictions. To further emphasize how the fusion auxiliary text facilitates recognition of relationships in the long tail, we follow [2], [15] and adopt mean Recall@K (mR@K).

Recognition of Unseen Relationships. We define the new task of *recognition of unseen visual relationships* (acronymed ROUVR), and then describe the evaluation metrics adopted for this task. ROUVR differs from the previously defined zero-shot VRD [9] in that the unseen relationships are not included among the relationships in the training set. Given a set of training examples S_m, containing a set of training relationships R_{train}, and an auxiliary text corpus T containing a set of relationships R_T, we follow the same training protocol as for any SGG; the only difference is that now we use T to facilitate detection of unseen relationships not appearing in R_{train}. We name this subset R_{unseen}, and by definition, $R_{unseen} \cap R_{train} = \emptyset$. To test our model's effectiveness on the task of recognizing unseen visual relationships, we must use a specialized dataset. In this paper we propose the VRD dataset, which relies on a relationship set R containing relationships unseen in R_{train}. At test time we aim to assign the correct relationship r^* to a pair of objects even though r^* is not necessarily contained in R_{train}.

4. TranstextNet

Our method, called TranstextNet, provides an effective way to utilize text for SGG and VRD. TranstextNet comprises three components: (1) an SGG backbone, (2) a mapping between subject-object (s-o) pairs to sets of relationships based on the text, and (3) fusion mechanisms that combine together visual features with textual features. We first describe a general SGG backbone, and then how the s-o pairs of relationship sets are acquired. Thereafter we offer an overview of the proposed fusion mechanisms, and describe how to modify any SGG backbone so as to successfully utilize textual information.

4.1. SGG Backbone

A general SGG backbone [24, 20, 16] comprises three main components: (1) an object detector (OD), (2) a context layer (CL), and (3) a relationship recognition layer (RRL), as schematically illustrated at Figure 2. The first step in SGG is object detection, where Faster R-CNN [13] is usually used as the OD. The OD detects object class candidates, estimates their bounding boxes, and also provides feature extraction for computing regions of interest (ROI). The task of the CL is to contextualize the ROI features such that s-o connections are formed between them. As illustrated in Figure 2, the CL is a recurrent neural network. A general RRL takes the contextualized feature vectors and feature vectors that represent unions of bounding box pairs, and performs a multi-class classification task, which we term *relationship recognition* (RR). The simplest RRL is achieved by concatenating the contextual fea-

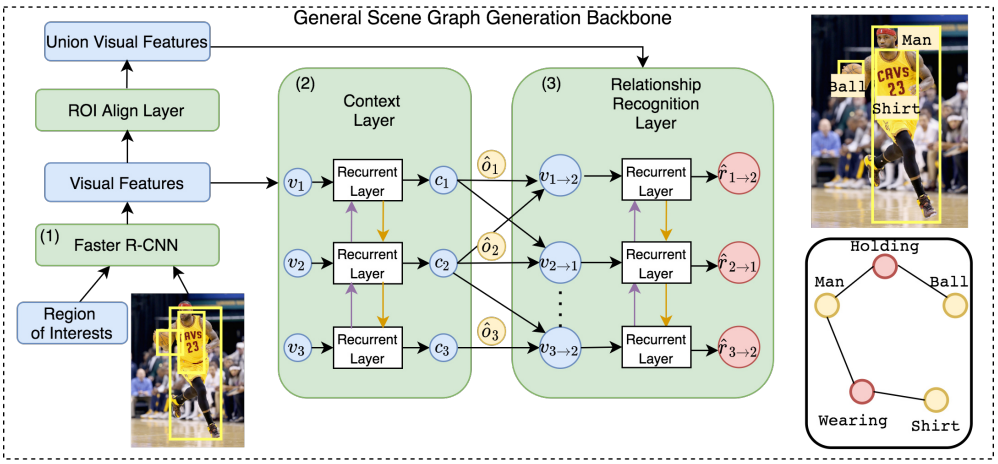

Figure 2: Schematic view of a general SGG backbone. Green rectangles are neural networks modules, blue rectangles and circles are features, yellow circles are object class predictions, and red circles are the relationship class predictions. The module has three main components: (1) an object detector (Faster R-CNN), (2) a context layer, and (3) a relationship recognition layer. The region of interest input exists only in Pred-Cls and SG-Cls setups.

ture vectors with the union ROI feature vectors and feeding the result to a fully connected layer for the final classification. We call this the context head, and denote by R_c its relationship candidates. For all other RRLs used in [24], [20, 16], e.g., Highway LSTM layer [24], we call the relation head $\hat{R}$.

4.2. Subject-Relationships-Object Acquisition

We now describe the subject-relationship-object (`<s, r, o>`) acquisition from the text process. Similarly to [23], we parse the text to `<s, r, o>` triplets, using a scene graph parser [14]. We collect data from three different sources: (1) sentences from image captioning datasets, (2) Gutenberg ebooks [7], and (3) sentences generated through a natural language generation (NLG) model [12]. The statistics we collect is a straightforward counting statistics, i.e., $P(r_{i\to j}|o_i,o_j) = \frac{Count(r_{i\to j},o_i,o_j)}{Count(o_i,o_j)}$. We only keep a relationship if it is in the GloVE vocabulary, and if $P(r_{i\to j}|o_i,o_j) > 10^{-3}$ (in which case we say the relationship is *valid*). To fine-tune the NLG model so as to generate `<s, r, o>`, we tried two strategies. The strategy first is to use caption sentences from image captioning datasets, and the second is parsing the captions using the scene graph parser and training on parsed `<s, r, o>` from an image captions datasets. We used [19] to fine-tune the NLG model. To compare the information contained in the different text sources, we use two measures: (1) coverage, i.e., the % of object pairs that have one or more valid relationships, and (2) the number of valid relationships in the resulting relationship set, i.e., $|R_{orm}|$ (see a detailed comparison in Section 3 of the appendix). A performance comparison of the three text sources appears in the appendix Section 8.

4.3. Fusion Mechanisms

Fusion mechanisms are the building block we use to combine visual features with textual features. In our setting, the input to a fusion mechanism is always visual features that represent relationships between objects, and textual features that represent the relationships from the text. The desired outcome is a visual representation, which is enriched by the text. We use two fusion mechanisms that are common in vision and language models: (1) attention [1], and (2) feature-wise linear modulation [11]. We later demonstrate the effectiveness of fusion mechanisms in an ablation study where we compare models trained with and without fusion mechanisms (see Table 4, where we examine more primitive fusion approaches). Moreover, by utilizing distributed word representations as inputs instead of the conditional statistics (as in [23]), we fuse real-world knowledge without explicitly conditioning it on the statistics of the text.

Attention Mechanism. We now describe how we utilized attention to fuse visual features with features from the text. Denote the visual features (called *query*) by q, and the textual features (called *context*) by $T = \{t_i\}_{i=1}^k$. First, we obtain the attention coefficient, a_i: $a_i = T \cdot q$, $a_i \in \mathbb{R}$. Next, we calculate the attention weights, w_i: $w_i = \text{Softmax}(a_i)$, $w_i \in \mathbb{R}$. The attention vector, $v = \sum_{i=1}^k w_i t_i$, is the weighted sum of the context vector and the attention weights. The final vector, q', is a concatenation of the query and attention vectors that we fuse using a linear layer, $q' = W_{att} \cdot [q, v]$, $q' \in \mathbb{R}^n$. Throughout the paper we denote this attention

procedure by $q' = A(q, T)$.

Feature-Wise Linear Modulation. Feature-wise linear modulation (Film), introduced by [11], and has been utilized for combining visual and textual features for several visual reasoning tasks. Early forms of feature fusion were basic algebraic operations such as summation and multiplications. Film combines both operations, and there is a strong motivation to use it for SGG and VRD because feature multiplications have demonstrated effectiveness in recognizing relationships between objects [24]; moreover, summation has been shown to work well in representation-based tasks. We now describe a Film block, and formulate a general Film procedure that meets our needs. A Film block contains two components: a generator $\mathcal{G}$ and a Film layer. $\mathcal{G}$ can be any neural layer, where for our purposes, we utilize a long-short term memory (LSTM) unit as it fits our needs for different lengths of sequences. A Film layer performs an affine transformation between two sets of features, and its procedure operates as follows. Let $q \in \mathbb{R}^n$ be an image feature and $T = [t_1, \ldots, t_k]$, $T \in \mathbb{R}^{k \times n}$ be the features from the text. First, we obtain the modulator $\hat{T} = \mathcal{G}(T) = [\gamma(T), \beta(T)]$ where $\hat{T} \in \mathbb{R}^{2n}$, and $\gamma(T)$ and $\beta(T)$ are the first and last n components of $\hat{T}$, respectively. Then we perform the modulation by utilizing the Film layer to obtain $q' = \gamma(T) \bigodot q + \beta(T)$, where $\bigodot$ stands for feature-wise multiplication. We denote this feature-wise modulation procedure by $q' = Film(q, T)$.

4.4. Transducing Relationships

TranstextNet is schematically illustrated in Figure 3. We now describe an `ORM` layer, and how we plug it into SGG backbones. A general `ORM` layer consists of three main blocks: a fusion mechanism, a pre-trained GloVE layer, and a mapping between s-o pairs to relationship sets. The `ORM` layer takes s-o class pair predictions detected by the OD, namely $\hat{O}$, and the visual features describing them, V (the visual features are projected to the same dimension as the textual features prior to the `ORM` layer). This procedure, applied on a single s-o pair, $\{o_i, o_j\}$, is exemplified and illustrated in Figure 3 (B); here, the s-o mapping returns a set of relationships $R_{i \to j} = \{r^k_{i \to j}\}_{k=1}^M$, and we randomly sample a subset of $R_{i \to j}$. In our case, the result is $R^{'}_{i \to j} = \{\texttt{with}, \texttt{holding}, \texttt{wearing}\}$. Next we extract T, the relationships' embeddings of $R^{'}_{i \to j}$. Then, V and T are fed to the fusion mechanism. The output $V' = ORM(\hat{O}, V)$ of the `ORM` layer is thus a feature vector enriched with information parsed from the text. We plug the `ORM` layer into an SGG backbone by connecting it in two different locations: (1) the output of CL, and (2) the output of *RRL* (outputs of layers 2 and 3 in Figure 2, respectively). Consider Figure 3 that illustrates TranstextNet . The first plug-in is at the output of CL, denoted $\hat{C}$. The `ORM` outputs are $\hat{C}' = ORM(\hat{O}, \hat{C})$ such that the context head predictions are $\hat{R}_c = \mathrm{W}_\mathrm{c}^\mathrm{T}\mathrm{C}' + \mathrm{b_c}$. $\hat{C}'$ and the union's visual features, $V_{i \to j}$, are fed into the *RRL*. The outputs of *RRL* are $V'_{i \to j}$, a combination of the $\hat{C}'$ and $V'_{i \to j}$, and $\hat{R}$ relationship class candidates. The second plug-in is at the output of the *RRL* such that $V''_{i \to j} = ORM(\hat{O}, V'_{i \to j})$ and the relationship predictions are $\hat{R}_r = W_r^T V''_{i \to j} + b_r$. Additionally we utilize $V''_{i \to j}$ to learn relationship representations, namely, $\hat{V}_r$. The final prediction is a combination of the three $\hat{R} = \mathrm{Softmax}(\hat{R}_c + \hat{R}_r + \hat{R} + \hat{V}_r)$. To predict the unseen relationships we utilize cosine similarity. Two vectors are considered close if their cosine similarity is close to one. For each ROI, the model produces a representation $\hat{v}_r$, and we compute $\hat{r} \triangleq \mathrm{Softmax}(\cos(\hat{v}_r, V_R^{GT}))$, where $V_R^{GT} \in \mathbb{R}^{e \times |R|}$ are the GloVE representations of the relationships in our test set.

4.5. Loss Function

Our loss functions is constructed to achieve three goals: (1) optimize object detection, (2) optimize seen relationship detection, and (3) enable unseen relationship recognition. For (1) and (2), we use the standard cross-entropy (CE) loss function. To enable unseen relationship recognition, we want to learn a relationship embedding space that would enable our model to recognize unseen relationships. To this end, we utilize the *cosine* loss function, which was selected from various options (see Section 5). Our final loss function is:

$$\begin{aligned} L = CE(\hat{O}, O) + CE(\hat{R}_c, R) + \\ CE(\hat{R}_r, R) + CE(\hat{R}, R) + cosine(\hat{V}_r, V_r), \end{aligned} \tag{1}$$

where V_r are the pre-trained GloVE word embeddings of R. We experiment with alternative loss functions that we believe would enable recognition of unseen relationships and compare them with the *cosine* loss function.

5. Empirical Study

In all our experiments we consider the application of TranstextNet (our method) on three best-known SGG models that are used as backbones: IMP [20], Motifs [24], and VCTree [16]. We demonstrate performance of these baselines with and without TranstextNet. Both Motifs and VCTree were trained using training scripts provided by their authors, and IMP was trained using the script provided by the authors of Motifs (see training protocols details in the appendix, Section 7). The three TranstextNet-extended models were trained using the same (respective) scripts.

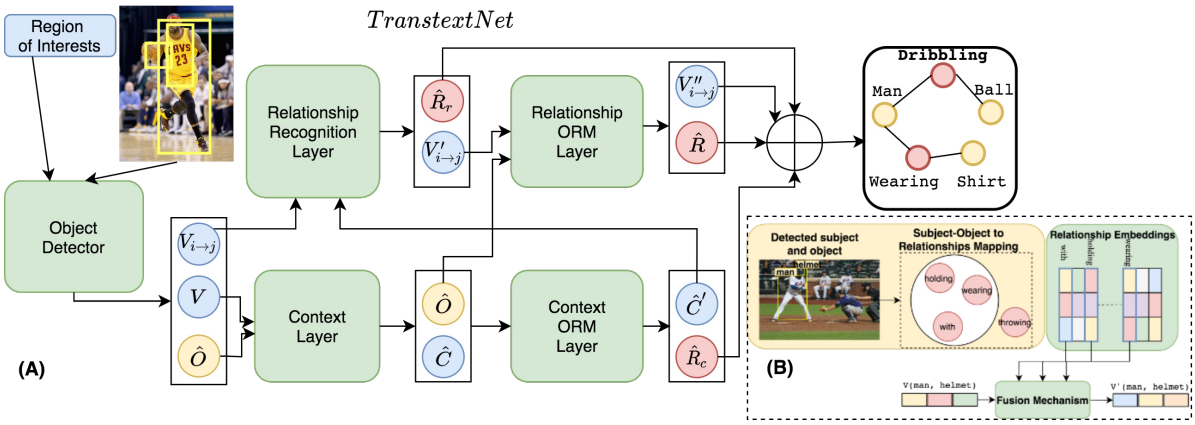

Figure 3: **General Notations**: Same as in Figure 2.(**A**) A schematic overview of TranstextNet . The `ORM` layer is plugged-in at two intersections, the output of the context layer and the output of the relationship recognition layer. The final relationship recognition is based on three prediction heads: the context head, the relationship recognition head, and the relationship recognition `ORM` layer. The predicted scene graph is illustrated on the right, and the unseen relationship recognized by TranstextNet is in bold. (**B**) An overview of an `ORM` layer. The layer receives s-o pairs and visual features $\mathbf{V}_{i\to j}$, and it maps the s-o pairs to relationship sets, extracts relationship embeddings for each, $\mathbf{T}$ and fuses the visual features with the relationship embeddings.

Datasets. We consider two datasets, VG-200 and VRD dataset. Introduced by [20], VG-200 is a filtered version of VG containing the most frequent 150 objects and most frequent 50 relationships. The VRD dataset (VRDD) was introduced by [9] and contains 5,000 training images and 1,000 test images. It has 100 object classes and 70 relationship classes, 57 of which are unseen during training. We further demonstrate TranstextNet's effectiveness on unseen relationship recognition by evaluating on VRDD without additional training.

Tasks. We consider three tasks, two standard tasks commonly used for evaluating SGG and VRD models, and a new task designed for testing recognition of unseen relationships.

Recognition of Unseen Relationships ROUVR tests the model's ability to recognize relationships that were absent from the relationship training set. To evaluate ROUVR, we use VRDD. Similar to SGG, we compare results across models and fusion mechanisms. We focus on predicting visual relationships; thus, at test time we provide ground truth boxes and object labels. We use the VRDD test set instances without additional training. **Scene Graph Generation** For VG-200, we use the same evaluation protocol used by [24], [2] who considered three tasks in two setups. In the first task, denoted *Pred-Cls*, the goal is to predict relationship labels, given the correct labels for subjects and objects. The objective in the second and harder task, denoted *SG-Cls*, is to predict subject and object labels, given their correct bounding boxes as input. In addition, correct relationships must be predicted. The last task is called *SG-Det*. Here the input is an image and the output is a prediction of the object boxes and labels, and the relationships. The first setup considered is graph constrained evaluations, which allows only one relationship per object pair, while the second omitted this constraint. In our study we consider both versions and refer to them as constrained and unconstrained.

Zero-Shot Scene Graph Generation. This task was introduced by [9], and first evaluated on VG by [15]. We consider *zero-shot scene graph generation* (ZS-SGG) whereby a triplet, `<s, r, o>` is absent during training but appears during testing. This task differs from ROUVR where all test relationships are introduced during training. The evaluation protocol is the same as in SGG.

5.1. Results

Scene Graph Generation. In Table 1 we present our results. The table is divided into two sections, *constrained* and *unconstrained*, and the metrics reported are R@K and mR@K. The results in the table clearly indicate that text ingestion enhances SGG, and that auxiliary text reduces the training bias and improves recognition of relationships in the long tail (see elaboration on mR@K in the appendix, Section 5). The results also indicate that attention mechanisms slightly outperform feature-wise linear modulations on SGG. For example, consider the last row of the constrained

Setup	Model	SG-Det				SG-Cls				Pred-Cls			
		R@50	R@100	mR@50	mR@100	R@50	R@100	mR@50	mR@100	R@50	R@100	mR@50	mR@100
Constrained	IMP	20.7	24.5	3.8	4.8	34.6	35.4	5.8	6	59.3	61.3	9.8	10.5
	IMP + TranstextNet$_F$	21.9	25.4	4.4	5.5	35.1	35.9	6.3	6.5	60.5	62.6	10.4	11.6
	IMP + TranstextNet$_A$	**22.4**	**25.8**	**4.6**	**5.9**	**35.8**	**36.3**	**6.4**	**6.8**	**60.8**	**62.9**	**10.9**	**12.3**
	Motifs	27.2	30.3	5.3	6.1	35.8	36.5	7.1	7.6	65.2	67.1	13.3	14.4
	Motifs + TranstextNet$_F$	28	31.1	5.6	6.8	36.3	37.1	**8.4**	**9.1**	66.9	68.3	**16.8**	**18.4**
	Motifs + TranstextNet$_A$	**28.2**	**31.3**	**5.9**	**7**	**36.5**	**37.3**	8.2	9	**67**	**68.5**	16.1	18.3
	VCTree	27.7	31.1	6.9	8	37.9	38.6	10.1	10.8	66.1	67.4	17.9	19.4
	VCTree + TranstextNet$_F$	27.9	31.6	7.2	8.6	38.3	39.2	10.5	11.4	66.8	68.5	18.3	20.3
	VCTree + TranstextNet$_A$	**28.1**	**31.7**	**7.4**	**8.9**	**38.3**	**39.3**	**10.6**	**11.7**	**66.9**	**68.7**	**18.5**	**20.6**
Unconstrained	IMP	22	27.4	5.4	8	43.4	47.2	12.1	16.9	75.2	88.3	20.3	28.9
	IMP + TranstextNet$_F$	**22.2**	27.6	5.9	8.3	44.6	47.9	12.8	17.7	79.7	85.3	24.6	33.3
	IMP + TranstextNet$_A$	22.1	**27.6**	**6**	**8.5**	**44.7**	**48.1**	**12.8**	**17.8**	**79.9**	**85.4**	**24.7**	**33.4**
	Motifs	30.5	35.8	9.3	12.9	44.5	47.7	15.4	20.6	81.1	88.3	27.5	37.9
	Motifs + TranstextNet$_F$	30.8	36	9.7	13.3	46.1	48.3	15.8	21.3	83.3	90.1	31.5	45.3
	Motifs + TranstextNet$_A$	**31**	**36.2**	**9.9**	**13.5**	**46.4**	**48.8**	**16.1**	**21.5**	**83.7**	**90.2**	**32.6**	**46.2**
	VCTree	31.3	36.9	11.8	16.2	47.1	49.2	20.1	26.5	82.3	89.4	36.8	49.2
	VCTree + TranstextNet$_F$	31.5	37.2	11.9	16.4	48.6	51.1	20.6	27.1	84.1	90.3	37.8	52.3
	VCTree + TranstextNet$_A$	**31.6**	**37.4**	**12**	**16.45**	**48.8**	**51.2**	**20.8**	**27.3**	**84.4**	**90.6**	**38.9**	**53.4**

Table 1: Recall@K and mean Recall@K SGG results on VG-200. Top: Constrained setup. Bottom: Unconstrained setup. TranstextNet$_A$ denotes the use of our attention mechanism, TranstextNet$_F$ denotes the use of Film.

setup, i.e., VCTree +TranstextNet$_A$. The results show a consistent improvement across all metrics and tasks when compared with the VCTree baseline. For qualitative results, consider Figure 4 that illustrates different SGs generated by the different models. We compare results across fusion mechanisms.

Zero-Shot Scene Graph Generation. In Table 3 we present our results for ZS-SGG. The results clearly show that TranstextNet outperforms all baseline models on this task by a huge margin. We also compare our results to the results reported in [15]. We outperform the results on the same baselines while using an inferior object detector (they used Mask-RCNN). These results emphasize the importance of text ingestion in ZS-SGG.

Recognition of Unseen Relationships. Our results for all models appear in Table 2. The results clearly support our hypothesis that transduction of auxiliary text facilitates ROUVR, as models that utilize auxiliary text outperform their baseline models by an extreme margin. We also note that models that utilized auxiliary text were able to recognize 40 out of 57 unseen relationships (see our elaboration on the ROUVR results in the appendix, Sections 6) The results of both fusion mechanisms are similar where feature-wise linear modulation outperforms attention mechanisms by a small margin.

Model	R@5	R@10
IMP	2.2	4.6
IMP + TranstextNet$_F$	14.3	19.9
IMP + TranstextNet$_A$	**15.04**	**23.1**
Motifs	8.5	13.3
Motifs + TranstextNet$_F$	16.1	20.3
Motifs + TranstextNet$_A$	**16.8**	**23.7**
VCTree	6.3	9.2
VCTree + TranstextNet$_F$	15.8	21.6
VCTree + TranstextNet$_A$	**16.65**	**23.3**

Table 2: Results of ROUVR in VRDD

Task	Pred-Cls		SG-Cls		SG-Det	
Model	R@50	R@100	R@50	R@100	R@50	R@100
IMP	15.7	17.9	2.4	3.8	0.13	0.25
IMP + TranstextNet$_F$	25.3	30.3	3.6	5.3	0.19	0.37
IMP + TranstextNet$_A$	**23.8**	**30.5**	**3.9**	**5.7**	**0.2**	**0.37**
Motifs	10.9	14.5	2.2	3	0.1	0.2
Motifs + TranstextNet$_F$	25.4	31.9	4.4	6.3	0.17	0.3
Motifs + TranstextNet$_A$	**26.7**	**32.1**	**4.5**	**6.7**	**0.19**	**0.33**
VCTree	10.8	14.3	2.5	3.3	0.2	0.24
VCTree + TranstextNet$_F$	23.7	29.9	4.1	6	0.16	0.33
VCTree + TranstextNet$_A$	**24.98**	**30.2**	**4.3**	**6.2**	**0.17**	**0.33**

Table 3: Results of ZS-SGG in VG-200.

	SGG				ZS-SGG		ROUVR	
	Pred-Cls							
Loss Function	R@100	R@50	mR@100	mR@50	R@100	R@50	R@10	R@5
Loss Functions								
LMCL	71.5	66.7	12.4	9.7	31.3	24.6	14.8	9.9
LMGM	68.8	67.2	22.6	18.3	31.3	25.4	19.1	13.6
Without fusion Mechanisms								
Sum	67.6	65.8	14.8	12.9	19.5	16.7	6.9	16.5
Concat	67.7	65.9	15.7	13.6	19.8	17	7.4	17.1

Table 4: Comparing loss functions for representation learning (rows 5 and 6). TranstextNet without fusion mechanisms (rows 8 and 9). All results obtained with Motifs backbone.

5.2. Ablation Studies

Loss Function. To demonstrate the effectiveness of the *cosine* loss, we investigate two additional loss functions. The first is *Large Margin Cosine Loss* (LMCL) [18], which was employed for face recognition. The authors presented an interesting result where the face embeddings space held similar properties as word em-

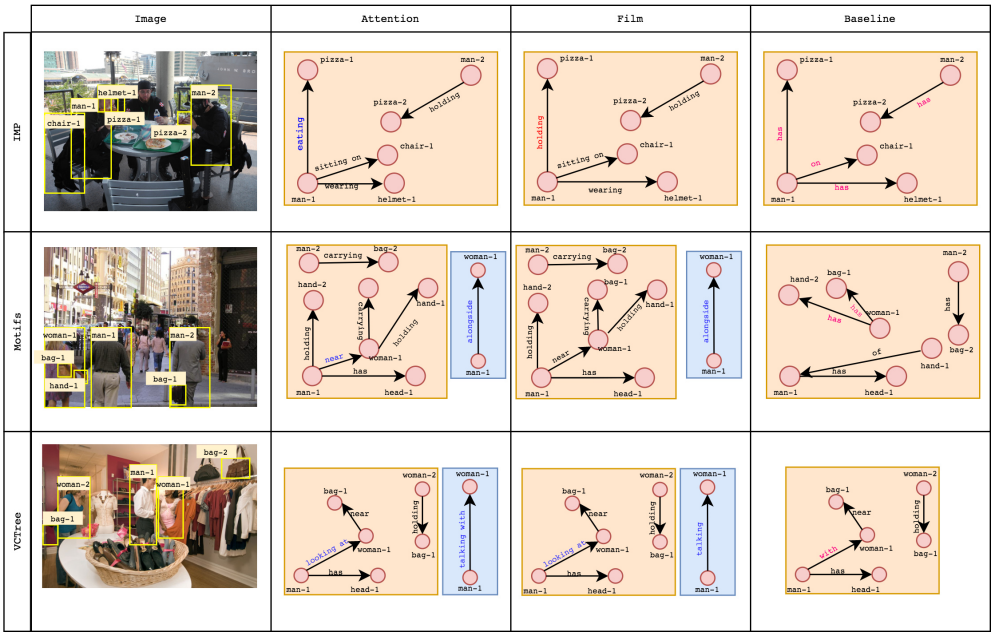

Figure 4: Qualitative results of three baseline models. Images appear (in column 1) with detected objects and bounding boxes, the respective SGG results (in columns 2-4, in yellow rectangles), and recognized unseen relationships (in the blue rectangles). The different colors of relationships signify the different detection results across fusion mechanisms.

bedding spaces, where faces belonging to the same face were inside the same sphere. The other loss function we considered is the *Large Margin Gaussian Mixture Loss* (LMGM) of [17], which was utilized for image classification, and also demonstrated a representation with similar properties to LMCL. We experimented with all the SGG and ROUVR tasks, specifically in a Pred-Cls setup, to demonstrate their effectiveness on relationships. The quantitative results are in Table 4 in rows 5 and 6. The results indicate that LMCL works very well on SGG and ZS-SGG. In fact, TranstextNet$_A$ applied with the Motifs backbone and LMCL, achieves SOTA performance on SGG and ZS-SGG (in a Pred-Cls setup) by a wide margin. Nevertheless, w.r.t. the mR@K metric, LMCL achieves results inferior to other models. LGML performs very well on SGG and ZS-SGG, and is also marginally better on mR@K, and is able to recognize infrequent relationships very well. It is also clear that both loss functions fail to transduce unseen relationships, which is our main focus here. These results supports our choice of *cosine loss* for learning relationship representations. **With vs. Without Fusion Mechanisms.** To show the positive effects of utilizing fusion mechanisms, we train Motifs with auxiliary text without our fusion mechanism, and to combine visual and textual features we take the weighted average of $V_R = \sum_{t=1^M} p^t_{i\to j} v^t_{i\to j}$, and try to fuse it in two simple ways: (1) summation, and (2) concatenation. To test the effectiveness on recognition of relationships, we compare those approaches on SGG, ZS-SGG, and ROUVR in the Pred-Cls setup. The results are shown in Table 4 (two bottom rows), and clearly indicate that fusion mechanisms benefit SGG, ZS-SGG, and ROUVR. Omission of fusion mechanisms degrades performance on all tasks.

6. Concluding Remarks

TranstextNet is a novel model for ingesting auxiliary text, which can easily extend to existing SGG backbones, leading to improved SOTA on all SGG tasks (previous SOTA achieved by [16]). Importantly, in this paper we introduced the task of recognizing unseen relationships and demonstrated how TranstextNet impressively transduces relationship knowledge from text to images by effectively fusing textual and visual features. Another distinct benefit achieved by TranstextNet is the reduction of the training bias due to imbalances in the training data.

7. Acknowledgments

This research was supported by The Israel Science Foundation, grant No. 710/18.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. *arXiv preprint arXiv:1903.03326*, 2019.

[3] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.

[4] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *arXiv preprint arXiv:1802.05451*, 2018.

[5] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015.

[6] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[7] Marie Lebert. *Le Projet Gutenberg (1971-2008)*. Project Gutenberg, 2008.

[8] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual relationship detection with deep structural ranking. 2018.

[9] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016.

[10] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014.

[11] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[14] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80, 2015.

[15] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. *arXiv preprint arXiv:2002.11949*, 2020.

[16] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019.

[17] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9117–9126, 2018.

[18] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[20] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[21] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[22] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. *arXiv preprint arXiv:1809.07041*, 2018.

[23] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. *arXiv preprint arXiv:1707.09423*, 2017.

[24] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[25] Yibing Zhan, Jun Yu, Ting Yu, and Dacheng Tao. On exploring undetermined relationships for visual relationship detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5128–5137, 2019.

[26] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5532–5540, 2017.

[27] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 589–598, 2017.