Situation Recognition with Graph Neural Networks

Ruiyu Li[1], Makarand Tapaswi[2], Renjie Liao[2], Jiaya Jia[1,3], Raquel Urtasun[2,4,5], Sanja Fidler[2,5]

[1]The Chinese University of Hong Kong, [2]University of Toronto, [3]Youtu Lab, Tencent
[4]Uber Advanced Technologies Group, [5]Vector Institute

ryli@cse.cuhk.edu.hk, {makarand,rjliao,urtasun,fidler}@cs.toronto.edu, leojia9@gmail.com

Abstract

We address the problem of recognizing situations in images. Given an image, the task is to predict the most salient verb (action), and fill its semantic roles such as who is performing the action, what is the source and target of the action, etc. *Different verbs have different roles (*e.g. `attacking` *has* `weapon`*), and each role can take on many possible values (nouns). We propose a model based on Graph Neural Networks that allows us to efficiently capture joint dependencies between roles using neural networks defined on a graph. Experiments with different graph connectivities show that our approach that propagates information between roles significantly outperforms existing work, as well as multiple baselines. We obtain roughly 3-5% improvement over previous work in predicting the full situation. We also provide a thorough qualitative analysis of our model and influence of different roles in the verbs.*

1. Introduction

Object [14, 33, 36], action [35, 40], and scene classification [50, 51] have come a long way, with performance in some of these tasks almost reaching human agreement. However, in many real world applications such as robotics we need a much more detailed understanding of the scene. For example, knowing that an image depicts a `repairing` action is not sufficient to understand what is really happening in the scene. We thus need additional information such as the person repairing the house, and the tool that is used.

Several datasets have recently been collected for such detailed understanding [22, 27, 47]. In [22], the Visual Genome dataset was built containing detailed relationships between objects. A subset of the scenes were further annotated with *scene graphs* [17] to capture both unary (*e.g.* attributes) and pairwise (*e.g.* relative spatial info) object relationships. Recently, Yatskar et al. [47] extended this idea to actions by labeling action *frames* where a frame consists of a fixed set of roles that define the action. Fig. 1 shows a frame for action `repairing`. The challenge then consists

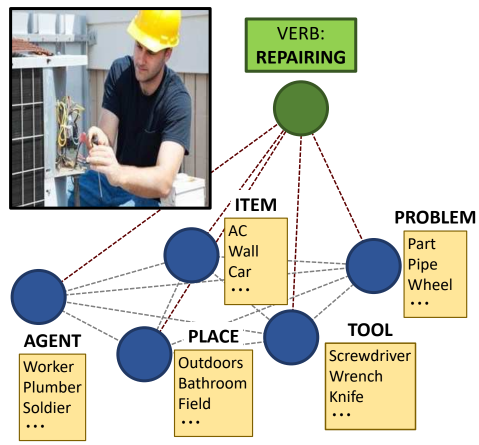

Figure 1. Understanding an image involves more than just predicting the most salient action. We need to know who is performing this action, what tools (s)he may be using, *etc*. Situation recognition is a structured prediction task that aims to predict the verb and its *frame* that consists of multiple role-noun pairs. The figure shows a glimpse of our model that uses a graph to model dependencies between the verb and its roles.

of assigning values (nouns) to these roles based on the image content. The number of different role types, their possible values, as well as the number of actions are very large, making it a very challenging prediction task. As shown in Fig. 2, the same verb can appear in very different image contexts, and nouns that fill the roles are vastly different.

In [47], the authors proposed a Conditional Random Field (CRF) to model dependencies between verb-role-noun pairs. In particular, a neural network was trained in an end-to-end fashion to both, predict the unary potentials for verbs and nouns, and to perform inference in the CRF. While their model captured the dependency between the verb and role-noun pairs, dependencies between the roles were not modeled explicitly.

In this paper, we aim to jointly reason about verbs and their roles using a *Graph Neural Network* (GNN), a generalization of graphical models to neural networks. A GNN defines observation and output at each node in the graph,

RIDING							
ROLE	AGENT	VEHICLE	PLACE	ROLE	AGENT	VEHICLE	PLACE
VALUE	MAN	HORSE	OUTSIDE	VALUE	DOG	SKATEBOARD	SIDEWALK

Figure 2. Images corresponding to the same verb can be quite different in their content involving verb roles. This makes situation recognition difficult.

and propagates messages along the edges in a recurrent manner. In particular, we exploit the GNNs to also model dependencies between roles and predict a consistent structured output. We explore different connectivity structures among the role nodes, and show that our approach significantly improves performance over existing work. In addition, we compare with strong baseline methods using Recurrent Neural Networks (RNNs) that have been shown to work well on joint prediction tasks, such as semantic [49] and object instance [3] segmentation, as well as on group activity recognition [8]. We also visualize the learned models to further investigate dependencies between roles.

2. Related Work

Situation recognition generalizes action recognition to include actors, objects, and location in the activity. There has been work to combine activity recognition with scene or object labels [7, 12, 44, 45]. In [13, 31], visual semantic role labeling tasks were proposed where datasets are built to study action along with localization of people and objects. In another line of work, Yatskar *et al*. [47] created the *imSitu* dataset that uses linguistic resources from FrameNet [10] and WordNet [29] to associate images not only with verbs, but also with specific role-noun pairs that describe the verb with more details. As a baseline approach, in [47], a Conditional Random Field (CRF) jointly models prediction of the verb and verb-role-noun triplets. Further, considering that the large output space and sparse training data could be problematic, a tensor composition function was used [46] to share nouns across different roles. The authors also proposed to augment the training data by searching images using query phrases built from the structured situation.

Different from these methods, our work focuses on explicitly modeling dependencies between roles for each verb through the use of different neural architectures.

Understanding Images. There is a surge of interest in joint vision and language tasks in recent years. Visual Question Answering in images and videos [1, 38] aims to answer questions related to image or video content. In image captioning [19, 39, 42, 26], a natural language sentence is generated to describe the image. Approaches for these tasks often use the CNN-RNN pipelines to provide a caption, or a correct answer to a specific question. Dependencies between verbs and nouns are typically being implicitly learned with the RNN. An alternative is to list all important objects with their attributes and relationships. Johnson *et al*. [17] created *scene graphs*, which are being used for visual relationship detection [27, 30, 48] tasks. In [25], the authors exploit scene graphs to generate image captions.

In Natural Language Processing (NLP), semantic role labeling [11, 18, 20, 32, 43, 52] involves annotating a sentence with thematic or semantic roles. Building upon resources from NLP, and leveraging collections such as FrameNet [10] and WordNet [29], visual semantic role labeling, or situation recognition, aims to interpret details for one particular action with verb-role-noun pairs.

Graph Neural Networks. There are a few different ways for applying neural networks to graph-structured data. We divide them into two categories. The first group defines convolutions on graphs. Approaches like [2, 6, 21] utilized the graph Laplacian and applied CNNs to spectral domain. Differently, Duvenaud et al. [9] designed a special hash function such that a CNN can be used on the original graphs.

The second group applies feed-forward neural networks to every node of the graph recurrently. Information is propagated through the network by dynamically updating the hidden state of each node based on their history and incoming messages from their neighborhood. The Graph Neural Network (GNN) proposed by [34] utilized multi-layer perceptrons (MLP) to update the hidden state. However, their learning algorithm is restrictive due to the contraction map assumption. In the following work, the Gated Graph Neural Network (GGNN) [23] used a recurrent gating function [4] to perform the update, and effectively learned model parameters using back-propagation through time (BPTT).

Other work [24, 37] designed special update functions based on the LSTM [16] cell and applied the model to tree-structured or general graph data. In [28], knowledge graphs and GGNNs are used for image classification. Here we use GGNNs for situation recognition.

3. Graph-based Neural Models for Situation Recognition

Task Definition. Situation recognition as per the *imSitu* dataset [47] assumes a discrete set of verbs $\mathcal{V}$, nouns $\mathcal{N}$, roles $\mathcal{R}$, and frames $\mathcal{F}$. The verb and its corresponding frame that contains roles are obtained from FrameNet [10], while nouns come from WordNet [29]. Each verb $v \in \mathcal{V}$ is associated with a frame $f \in \mathcal{F}$ that contains a set of semantic roles E_f. Each role $e \in E_f$ is paired with a noun value

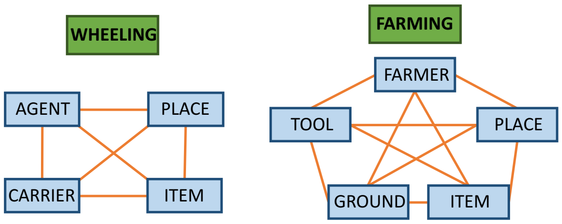

Figure 3. The architecture of fully-connected roles GGNN. The undirected edges between all roles of a verb-frame allows to fully capture the dependencies between them.

$n_e \in N \cup \{\varnothing\}$. Here, $\varnothing$ indicates that the noun is unknown or not applicable. A set of semantic roles and their nouns is called a realized frame, denoted as $R_f = \{(e, n_e) : e \in E_f\}$, where each role is with a noun.

Given an image, the task is to predict the structured situation $S = (v, R_f)$, specified by a verb $v \in \mathcal{V}$ and its corresponding realized frame R_f. For example, as shown on the right of Fig. 2, the verb `riding` is associated with three role-noun pairs, i.e., {`agent:dog`, `vehicle:surfboard`, `place:sidewalk`}.

3.1. Graph Neural Network

The verb and semantic roles of a situation depend on each other. For example, in the verb `carrying`, the roles `agent` and `agent-part` are tightly linked with the `item` being carried. Small items can be carried by hand, while heavy items may be carried on the back. We propose modeling these dependencies through a graph $G = (\mathcal{A}, \mathcal{B})$. The nodes in our graph $a \in \mathcal{A}$ are of two types of *verb* or *role*, and take unique values of $\mathcal{V}$ or $\mathcal{N}$, respectively. Since each image in the dataset is associated with one unique verb, every graph has a single verb node. Edges in the graph $b = (a', a)$ encode dependencies between role-role or verb-role pairs, and can be directed or undirected. Fig. 1 shows an example of such a graph where verb and role nodes are connected to each other.

Background. Modeling structure and learning representation on graphs have prior work. Gated Graph Neural Networks (GGNNs) [23] is one approach that learns the representation of a graph, which is then used to predict node- or graph-level output. Each node of a GGNN is associated with a hidden state vector that is updated in a recurrent fashion. At each time step, the hidden state of a node is updated based on its history and incoming messages from its neighbors. These updates are applied simultaneously to all nodes in the graph at each propagation step. The hidden states after T propagation steps are used to predict the output. In contrast, a standard unrolled RNN only moves information in one direction and updates one "node" per time step.

GGNN for Situation Recognition. We adopt the GGNN framework to recognize situations in images. Each image i is associated with one verb v that corresponds to a frame f with a set of roles E_f. We instantiate a graph $\mathcal{G}_f$ for each image that consists of one verb node, and $|E_f|$ (number of roles associated with the frame) role nodes. To capture the dependency between roles to the full extent, we propose creating undirected edges between all pairs of roles. Fig. 3 shows two example graph structures of this type. We explore other edge configurations in the evaluation.

To initialize the hidden states for each node, we use features derived from the image. In particular, for every image i, we compute representations $\phi_v(i)$ and $\phi_n(i)$ using the penultimate fully-connected layer of two convolutional neural network (CNN) pre-trained to predict verbs and nouns, respectively. We initialize the hidden states $h \in \mathbb{R}^D$ of the verb node a_v and role node a_e as

$$h^0_{a_v} = g(W_{iv}\phi_v(i)) \tag{1}$$
$$h^0_{a_e} = g(W_{in}\phi_n(i) \odot W_e e \odot W_v \hat{v})\,, \tag{2}$$

where $\hat{v} \in \{0,1\}^{|\mathcal{V}|}$ corresponds to a one-hot encoding of the predicted verb and $e \in \{0,1\}^{|\mathcal{R}|}$ is a one-hot encoding of the role that the node a_e corresponds to. $W_v \in \mathbb{R}^{D\times|\mathcal{V}|}$ is the verb embedding matrix, and $W_e \in \mathbb{R}^{D\times|\mathcal{R}|}$ is the role embedding matrix. W_{iv} and W_{in} are parameters that transform image features to the space of hidden representations. $\odot$ corresponds to element-wise multiplication, and $g(\cdot)$ is a non-linear function such as $\tanh(\cdot)$ or ReLU ($g(x) = \max(0, x)$). We normalize the initialized hidden states to unit-norm prior to propagation.

For any node a, at each time step, the aggregation of incoming messages at time t is determined by the hidden states of its neighbors a':

$$x^t_a = \sum_{(a',a)\in\mathcal{B}} W_p h^{t-1}_{a'} + b_p\,. \tag{3}$$

Note that we use a shared linear layer of weights W_p and biases b_p to compute incoming messages across all nodes.

After aggregating the messages, the hidden state of the node is updated through a gating mechanism similar to the Gated Recurrent Unit [4, 23] as follows:

$$\begin{aligned} z^t_a &= \sigma(W_z x^t_a + U_z h^{t-1}_a + b_z)\,, \\ r^t_a &= \sigma(W_r x^t_a + U_r h^{t-1}_a + b_r)\,, \\ \tilde{h}^t_a &= \tanh(W_h x^{(t)}_a + U_h(r^t_a \odot h^{t-1}_a) + b_h)\,, \\ h^t_a &= (1 - z^t_a) \odot h^{t-1}_a + z^t_a \odot \tilde{h}^t_a \quad . \end{aligned} \tag{4}$$

This allows each node to softly combine the influence of the aggregated incoming message and its own memory. W_z, U_z, b_z, W_r, U_r, b_r, W_h, U_h, and b_h are the weights and biases of the update function.

Output and Learning. We run T propagation steps. After propagation, we extract node-level outputs from GGNN

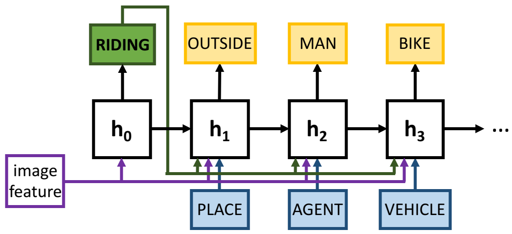

Figure 4. The architecture of chain RNN for verb `riding`. The time-steps at which different roles are predicted needs to be decided manually, and has an influence on the performance.

to predict the verb and nouns. Specifically, for each image, we predict the verb and a set of nouns for each role associated with the verb frame using a softmax layer:

$$p_v = \sigma(W_{hv} h_{a_v} + b_{hv}) \tag{5}$$
$$p_{e:n} = \sigma(W_{hn} h_{a_e} + b_{hn}) \,. \tag{6}$$

Note that the softmax function σ is applied across the class space for verbs $\mathcal{V}$ and nouns $\mathcal{N}$. $p_{e:n}$ can be treated as the probability of assigning noun n to role e.

Each image i in the *imSitu* dataset comes with three sets of annotations (from three annotators) for the nouns. During training, we accumulate the cross-entropy loss at verb and noun nodes for every annotation as

$$L = \sum_i \sum_{j=1}^{3} \Big(y_v \log(p_v) + \frac{1}{|E_f|} \sum_e y_{e:n} \log(p_{e:n}) \Big) \,, \tag{7}$$

where y_v and $y_{e:n}$ correspond to the ground-truth verb for image i and the ground-truth noun for role e of the image, respectively. Different to the Soft-OR loss in [47], we encourage the model to predict all three annotations for each image. We use back-propagation through time (BPTT) [41] to train the model.

Inference. At test time, our approach first predicts the verb $\hat{v} = \arg\max_v p_v$ to choose a corresponding frame f and obtain the set of associated roles E_f. We then propagate information among role nodes and choose the highest scoring noun $\hat{n}_e = \arg\max_n p_{e:n}$ for each role. Thus our predicted situation is

$$\hat{S} = (\hat{v}, \{(e, \hat{n}_e) : e \in E_f\}) \,. \tag{8}$$

To reduce reliance on the quality of verb prediction, we explore beam search over verbs as discussed in Experiments.

3.2. Simpler Graph Architectures

An alternative to model dependencies between nodes is to use recurrent neural networks (RNN). Here, situation recognition can be considered as a sequential prediction problem of choosing the verb and corresponding noun-role pairs. The hidden state of the RNN carries information across the verb and noun-role pairs, and the input at each time-step dictates what the RNN should predict.

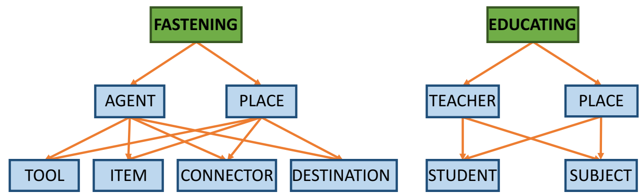

Figure 5. The architecture of tree-structured RNN. Like the Chain RNN, verb prediction is at the root of the tree, and semantic roles `agent`-like and `place` are parents of all other roles.

Chain RNN. An unrolled RNN can be seen as a special case of a GGNN, where nodes form a chain with directed edges between them. However, there are a few notable differences, wherein the nodes receive information only once from their (left) neighbor. In addition, the nodes do not perform T steps of propagation among each other and predict output immediately after the information arrives.

In the standard chain structure of a RNN, we need to manually specify the order of the verb and roles. As the choice of the verb dictates the set of roles in the frame, we predict the verb at the first time step. We observe that the *imSitu* dataset and any verb-frame in general, commonly consist of `place` and `agent`-like roles (*e.g.* semantic role `teacher` can be considered as the `agent` for the verb `teaching`). We thus predict `place` and `agent` roles as the second and third roles in the chain [1]. We make all other roles for the frame to follow subsequently in descending order of the number of times they occur across all verb-frames. Fig. 4 shows an example of such a model.

For a fair comparison to the fully connected roles GGNN, we employ the GRU update in our RNN. The input to the hidden states matches node initialization (Eqs. 1 and 2). We follow the same scheme for predicting the output (linear layer with softmax), and train the model with the same cross-entropy loss.

Tree-structured RNN. As mentioned above, the `place` and `agent` semantic roles occur more frequently. We propose a structure where they have a larger chance to influence prediction of other roles. In particular, we create a tree-structured RNN [37] where the hidden states first predict the verb, followed by `agent` and `place`, and all other roles. Fig. 5 shows examples of resulting structures.

The tree-structured RNN can be deemed as a special case of GGNN, where nodes have the following directed edges:

$$\mathcal{B} = \{(a_v, a') : a' \in \mathcal{Z}\} \cup \{(a', a) : a' \in \mathcal{Z}, a \in E_f \backslash \mathcal{Z}\} \,, \tag{9}$$

where $\mathcal{Z} = \{$`agent`, `place`$\}$, and $E_f \backslash \mathcal{Z}$ represents all roles in that frame other than `agent` and `place`. Similar to the chain RNN, we use GRU update and follow the same learning and inference procedures.

[1] Predicting `place` requires a more global view of the image compared to `agent`. Changing the order to `verb` $\rightarrow$ `agent` $\rightarrow$ `place` $\rightarrow \ldots$ results in 1.9% drop of performance.

	Method	top-1 predicted verb			top-5 predicted verbs			ground truth verbs		
		verb	value	value-all	verb	value	value-all	value	value-all	mean
1	Unaries	36.32	23.74	13.86	61.51	38.57	20.76	58.32	27.57	35.08
2	Unaries, BS=10	36.39	23.74	14.01	61.65	38.64	20.96	58.32	27.57	35.16
3	FC Graph, T=1	36.25	25.99	17.02	61.60	42.91	26.44	64.87	35.52	38.83
4	FC Graph, T=2	36.43	26.08	17.22	61.52	42.86	26.38	65.31	35.86	38.96
5	FC Graph, T=4	36.46	26.26	17.48	61.42	43.06	26.74	65.73	36.43	39.19
6	FC Graph, T=4, BS=10	36.70	26.52	17.70	61.63	43.34	27.09	65.73	36.43	39.39
7	FC Graph, T=4, BS=10, vOH	**36.93**	**27.52**	**19.15**	*61.80*	*45.23*	*29.98*	*68.89*	*41.07*	*41.32*
8	FC Graph, T=4, BS=10, vOH, g=ReLU	36.26	27.22	*19.10*	**62.14**	**45.59**	**30.32**	**69.35**	**41.71**	**41.46**
9	FC Graph, T=4, BS=10, vOH, Soft-OR	*36.75*	*27.33*	18.94	61.69	44.91	29.41	68.29	40.25	40.95

Table 1. Situation prediction results on the development set. We compare several variants of our fully-connected roles model to show the improvements achieved at every step. T refers to the number of **time-steps** of propagation in the fully connected roles GGNN (FC Graph). **BS=10** indicates the use of beam-search with beam-width of 10. **vOH** (verb, one-hot) is included when the embedding of the predicted verb is used to initialize the hidden state of the role nodes. g**=ReLU** refers to the non-linear function used after initialization. All other rows use $g=\tanh(\cdot)$. Finally, **Soft-OR** refers to the loss function used in [47]. Best performance is in **bold** and second-best is *italicized*.

4. Evaluation

We evaluate our methods on the *imSitu* dataset [47] and use the standard splits with 75k, 25k, and 25k images for the *train*, *development*, and *test* subsets, respectively. Each image in *imSitu* is associated with one verb and three annotations for the role-noun pairs.

We follow [46] and report three metrics: (i) *verb*: the verb prediction performance; (ii) *value*: the semantic verb-role-value tuple prediction performance that is considered to be correct if it matches any of the three ground truth annotators; and (iii) *value-all*: the performance when the *entire* situation is correct and all the semantic verb-role-value pairs match at least one ground truth annotation.

4.1. Implementation Details

Image Representations. We adopt two pre-trained VGG-16 CNNs [36] for extracting image features by removing the last fully-connected and softmax layers, and fine-tuning all weights. The first CNN ($\phi_v(i)$) is trained to predict verbs, and second CNN ($\phi_n(i)$) predicts the top K most frequent nouns ($K = 2000$ cover about 95% of nouns) in the dataset.

Unaries. Creating a graph with no edges, or equivalently with $T = 0$ steps of propagation corresponds to using the initialized features to perform prediction. We refer to this approach as *Unaries*, which will be used as the simplest baseline to showcase the benefit of modeling dependencies between the roles.

Learning. We implement the proposed models in Torch [5]. The network is trained using RMSProp [15] with mini-batches of 256 samples. We choose the hidden state dimension $D = 1024$, and train image (W_{iv}, W_{in}), verb (W_v) and role (W_e) embeddings. The image features are extracted before training the GGNN or RNN models.

The initial learning rate is 10^{-3} and starts to decay after 10 epochs by a factor of 0.85. We use dropout with a probability of 0.5 on the output prediction layer (*c.f*. Eqs. 5 and 6) and clip the gradients to range $(-1, 1)$.

Mapping `agent` Roles. The *imSitu* dataset [47] has situations for 504 verbs. Among them, we notice that 19 verbs do not have the semantic role `agent` but instead with roles of similar meaning (*e.g*. verb `educating` has role `teacher`). We map these alternative roles to `agent` when determining their position in the RNN architecture. Such a mapping is not used for the fully connected GGNN model.

Variable Number of Roles. A verb has a maximum of 6 roles associated with it. We implement our proposed model with fixed-size graphs involving 7 nodes. To deal with verbs with less than 6 roles, we zero the hidden states at each time-step of propagation, making them not receive or send any information.

4.2. Results

We first present a quantitative analysis comparing different variants of our proposed model. We then evaluate the performance of different architectures, and compare results with state-of-the-art approaches.

Ablative Analysis A detailed study of the GGNN model with fully connected roles (referred to as *FC Graph*) is shown in Table 1. An important hyper-parameter for the GGNN model is the number of propagation steps T. We found that the performance increases by a small amount when increasing T, and saturates soon (in rows 3, 4, and 5). We believe that this is due to the use of a fully-connected graph, and all nodes sharing most of the information at the first-step propagation. Nevertheless, the propagation is important, as revealed in the comparison between *Unaries* ($T = 0$) from row 1 and $T = 1$ in row 3. We obtain a mean improvement of 3.8% in all metrics.

During test we have the option of using beam search, where we hold B best verb predictions and compute the

		top-1 predicted verb			top-5 predicted verbs			ground truth verbs		
		verb	value	value-all	verb	value	value-all	value	value-all	mean
1	Unaries	*36.39*	23.74	14.01	*61.65*	38.64	20.96	58.32	27.57	35.16
2	Chain RNN	34.62	*24.67*	*17.94*	61.09	*41.67*	*27.80*	*62.58*	*36.57*	*38.36*
3	Tree-structured RNN	34.62	24.24	16.04	58.86	39.15	23.65	60.44	30.91	35.98
4	Chain GGNN, T=8	36.63	27.27	19.03	**61.88**	44.97	29.44	68.20	40.21	40.95
5	Tree-structured GGNN, T=6	*36.78*	*27.48*	**19.54**	61.75	*45.12*	**30.11**	*68.54*	*41.01*	*41.29*
6	Fully-connected GGNN, T=4	**36.93**	**27.52**	*19.15*	*61.80*	**45.23**	*29.98*	**68.89**	**41.07**	**41.32**

Table 2. Situation prediction results on the development set for models with different graph structures. All models use beam search, predicted verb embedding, and $g = \tanh(\cdot)$. Best performance is highlighted in **bold**, and second-best in each table section is *italicized*.

		top-1 predicted verb			top-5 predicted verbs			ground truth verbs		
		verb	value	value-all	verb	value	value-all	value	value-all	mean
dev	CNN+CRF [47]	32.25	24.56	14.28	58.64	42.68	22.75	65.90	29.50	36.32
	Tensor Composition [46]	32.91	25.39	14.87	59.92	44.50	24.04	*69.39*	33.17	38.02
	Tensor Composition + DataAug [46]	34.20	*26.56*	15.61	**62.21**	**46.72**	25.66	**70.80**	34.82	*39.57*
	Chain RNN	*34.62*	24.67	*17.94*	61.09	41.67	*27.80*	62.58	*36.57*	38.36
	Fully-connected Graph	**36.93**	**27.52**	**19.15**	*61.80*	*45.23*	**29.98**	68.89	**41.07**	**41.32**
test	CNN+CRF [47]	32.34	24.64	14.19	58.88	42.76	22.55	65.66	28.96	36.25
	Tensor Composition [46]	32.96	25.32	14.57	60.12	44.64	24.00	*69.20*	32.97	37.97
	Tensor Composition + DataAug [46]	34.12	*26.45*	15.51	**62.59**	**46.88**	25.46	**70.44**	34.38	*39.48*
	Chain RNN	*34.63*	24.65	*17.89*	61.06	41.73	*28.15*	62.94	*37.32*	38.54
	Fully-connected Graph	**36.72**	**27.52**	**19.25**	*61.90*	*45.39*	**29.96**	69.16	**41.36**	**41.40**

Table 3. We compare situation prediction results on the development and test sets against state-of-the-art models. Each model was run on the test set only once. Our model shows significant improvement in the top-1 prediction on all metrics, and performs better than a baseline that uses data augmentation. The performance improvement on the *value-all* metric is important for applications, such as captioning and QA. Best performance is highlighted in **bold**, and second-best is *italicized*.

role-noun predictions for each of the corresponding graphs (frames). Finally, we select the top prediction using the highest log-probability across all B options. We use a beam width of $B = 10$ in our experiments, which yields small improvement. Rows 1 and 2 of Table 1 show the improvement using beam search on a graph without propagation. Rows 5 and 6 show the benefit after multiple steps of propagation.

Rows 6 and 7 of Table 1 demonstrate the impact of using embeddings of the predicted verb (vOH) to initialize the role nodes' hidden states in Eq. (2). Notable improvement is obtained when using the ground-truth verb (3-4%). The *value-all* for the top-1 predicted verb increases from 17.70% to 19.15%. We also tested different non-linear functions for initialization, *i.e.*, $\tanh$ (row 7) or ReLU (row 8), however, the impact is almost negligible. We thus use $\tanh$ for all experiments.

Finally, comparing rows 7 and 9 of Table 1 reveals that our loss function to predict all annotations in Eq. (7) performs slightly better than the Soft-OR loss that aims to fit at least one of the annotations [47].

Baseline RNNs. Table 2 summarizes the results with different structures on the dev set. As expected, *Unaries* perform consistently worse than models with information propagation between nodes on the *value* and *value-all* metrics. The *Tree-structured RNN* provides a 2% boost in *value-all* for top-1 predicted verb, while the *Chain RNN* provides a 3.9% improvement. Owing to the better connectivity between the roles in a *Chain RNN* (especially `place` and `agent`), we observe better performance compared to the

	SPRINKLING				
	AGENT	PLACE	ITEM	SOURCE	DEST.
Unaries	PERSON	KITCHEN	MEAT	HAND	HAND
RNN	PERSON	KITCHEN	FOOD	FINGER	PIZZA
FC Graph	PERSON	KITCHEN	CHEESE	HAND	PIZZA

	FISHING			
	AGENT	PLACE	SOURCE	TOOL
Unaries	MAN	RIVER	-	FISHING
RNN	MAN	OUTDOORS	BODY	FISHING
FC Graph	MAN	RIVER	RIVER	FISHING

Figure 6. Example images with their predictions listed from all methods. Roles are marked with a blue background, and predicted nouns are in green boxes when correct, and red when wrong. Using the *FC Graph* corrects mistakes made by the *Unaries* or *Chain RNN* prediction models.

Tree-structured RNN. Note that as the RNNs are trained jointly to predict both verbs and nouns, and as the noun gradients dominate, the verb prediction takes a hit.

Different Graph Structures. We can also use *chain* or *tree-structured* graphs in GGNN. Along with the FC graph in row 6 of Table 2, rows 4 and 5 present the results for different GGNN structures. They show that connecting roles with each other is critical and sharing information helps. Interestingly, the Chain GGNN needs more propagation steps (T=8), as it takes time for the left-most and right-most nodes to share information. Smaller values of T are possible when nodes are well-connected as in Tree-structured (T=6) or FC Graph (T=4). Fig. 6 presents prediction from all models for two images. The *FC Graph* is able to reason about associating `cheese` and `pizza` rather than `sprinkling meat` or `food` on it.

Comparison with State-of-the-art. We compare the performance of our models against state-of-the-art on both the dev and test sets in Table 3. Our CNN predicts the verb well. Beam search leads to even better performance (2-4% higher) in verb prediction. We note that *Tensor Composition + DataAug* actually uses more data to train models. Nevertheless, we achieve the best performance on all metrics when using the top-1 predicted verb.

Another advantage of our model is in improvement for the *value-all* metric. It yields +8% when using the ground-truth verb, +6% with top-5 predicted verbs, and +4.5% with top-1 predicted verb, compared with the baseline without data augmentation. Interestingly, even with data augmentation, we outperform [46] by 3-4% in *value-all* for top-1 predicted verb. This property attributes to information sharing between role nodes, which helps in correcting errors and better predicts frames. Note that *value-all* is an important metric to measure a full understanding of the image. Models with higher *value-all* will likely lead to better captioning or question-answering results.

4.3. Further Discussion

We delve deeper into our model and discuss why the *FC Graph* outperforms baselines.

Learned Structure. A key emphasis of this model is on information propagation between roles. In Fig. 7, we present the norms of the propagation matrices. Each element in the matrix $P(a', a)$ is the norm of the incoming message from role a' to a averaged across all images (in dev set) at the first time-step, i.e., $\|x^{t=1}_{(a',a)}\|$ regarding Eq. (3). In this example, `tool` is important for the verb `fastening` and influences all other roles, while `agent` and `obstacle` influence roles in `jumping`.

Wrong Verb Predictions. We present a few examples of top scoring results where the verb prediction is wrong in Fig. 8. Note that in fact these predicted verbs are plausible options for the given images. The metric *value* treats them as wrong, and yet we can correctly predict the role-noun pairs. One example is the middle one of `slouching` vs. `sitting`. Fig. 8 (bottom) shows that choosing a different verb might lead to the selection of different roles (`goalitem` vs. `item, destination`). Nevertheless, predicting `book` for `browsing` is a good choice.

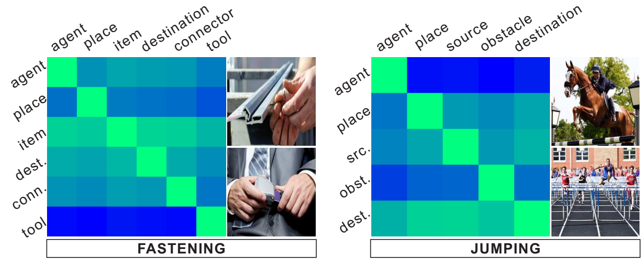

Figure 7. We present the "amount" of information that is propagated between roles for two verbs along with sample images. Blue corresponds to high, and green to zero. Each element of the matrix corresponds to the norm of the incoming message from different roles (normalized column sum to 1). **Left:** verb `fastening` needs to pay attention to the `tool` used. **Right:** important components to describe `jumping` are the `agent` and `obstacles` along the path.

GT: **FISHING**			
AGENT	PLACE	TOOL	SOURCE
MAN	BOAT	FISHING	LAKE
PRED: **CATCHING**			
AGENT	PLACE	TOOL	CAUGHTITEM
MAN	BOAT	BODY	FISHING

GT: **SLOUCHING**		
AGENT	PLACE	CONTACT
WOMAN	OFFICE	CHAIR
PRED: **SITTING**		
AGENT	PLACE	CONTACT
WOMAN	OFFICE	CHAIR

GT: **SHELVING**			
AGENT	PLACE	ITEM	DESTINATION
WOMAN	LIBRARY	BOOK	BOOKSHELF
PRED: **BROWSING**			
AGENT	PLACE	GOALITEM	
WOMAN	LIBRARY	BOOK	

Figure 8. Images with ground-truth and top-1 predictions from the development set. Roles are marked with blue background. Ground-truth (GT) nouns are in yellow and predicted (PRED) nouns with green when correct, or red when wrong. Although the predicted verb is different from the ground-truth, it is very plausible. Some of the verbs refer to the same frame (*e.g.* `sitting` and `slouching`), and contain the same set of roles, which our model is able to correctly infer.

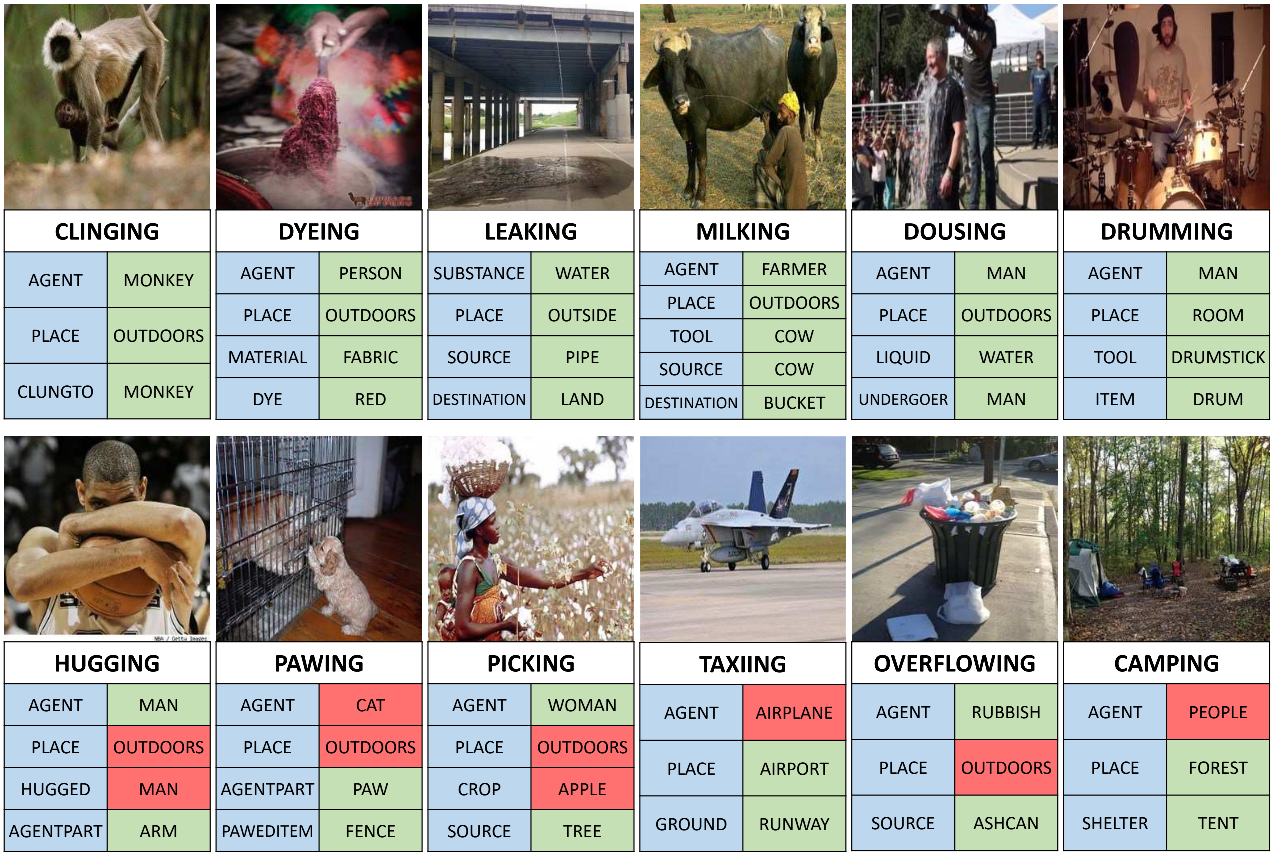

Figure 9. Images with top-1 predictions from the development set. For all samples, the predicted verb is correct, shown below the image in bold. Roles are marked with a blue background, and predicted nouns are in green when correct, and red when wrong. **Top row**: We are able to correctly predict the situation (verb and all role-noun pairs) for all samples. **Bottom row**: The first three samples contain errors in prediction (*e.g.* the `agent` for the verb `pawing` is clearly a `dog`). However, the latter three samples are in fact correct predictions that are not found in the ground-truth annotations (*e.g.* `people` are in fact `camping` in the `forest`).

Predictions with Correct Verb. Fig. 9 shows several examples of prediction obtained by FC Graph, where the predicted verb matches the ground-truth one. The top row corresponds to samples where the metric *value-all* scores correctly as all role-noun pairs are correct. Note that the roles are closely related (*e.g.* `(agent, clungto)` and `(material, dye)`) and help each other choose the correct nouns. In the bottom row, we show some failure cases in predicting role-noun pairs. First, the model favors predicting `place` as `outdoor` (a majority of place is `outdoor` in the training set). Second, for the sample with verb `picking`, we predict the `crop` as `apple`, which appears 79 times in the dataset compared with `cotton` that appears 14 times. Providing more training samples (*e.g.* [46]) could help remedy such issues.

In the latter three samples of the bottom row, although the model makes reasonable predictions, they do not match the ground-truth. For example, the ground-truth annotation for the verb `taxiing` is `agent:jet` and for the verb camping is `agent:persons`. Therefore, even though each image comes with three annotations, synonymous nouns and verbs make the task still challenging.

5. Conclusion

We presented an approach for recognizing situations in images that involves predicting the correct verb along with its corresponding frame consisting of role-noun pairs. Our Graph Neural Network (GNN) approach explicitly models dependencies between verb and roles, allowing nouns to inform each other. On a benchmark dataset *imSitu*, we achieved ∼4.5% accuracy improvement on a metric that evaluates correctness of the entire frame (*value-all*). We presented analysis of our model, demonstrating the need to capture the dependencies between roles, and compared it with RNN models and other related solutions.

6. Acknowledgements

This work is in part supported by a grant from the Research Grants Council of the Hong Kong SAR (project No. 413113). We also acknowledge support from NSERC, and GPU donations from NVIDIA.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 2

[2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *ICLR*, 2014. 2

[3] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, 2017. 2

[4] K. Cho, B. van Merriënboer, C. Gulcehre, B. Dzmitry, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, 2014. 2, 3

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 5

[6] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016. 2

[7] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010. 2

[8] Z. Deng, A. Vahdat, H. Hu, and G. Mori. Structured Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition. In *CVPR*, 2016. 2

[9] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, 2015. 2

[10] C. J. Fillmore, C. R. Johnson, and M. R. L. Petruck. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250, 2003. 2

[11] H. Fürstenau and M. Lapata. Graph Alignment for Semi-Supervised Semantic Role Labeling. In *EMNLP*, 2009. 2

[12] A. Gupta and L. S. Davis. Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. In *ECCV*, 2008. 2

[13] S. Gupta and J. Malik. Visual Semantic Role Labeling. *arXiv:1505.04474*, 2015. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 1

[15] G. Hinton, N. Srivastava, and K. Swersky. Lecture 6a overview of mini-batch gradient descent. 5

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[17] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image Retrieval using Scene Graphs. In *CVPR*, 2015. 1, 2

[18] D. Jurafsky and J. H. Martin. *Speech and Language Processing*, chapter 22. Semantic Role Labeling. 3, draft edition, 2017. 2

[19] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *CVPR*, 2015. 2

[20] P. Kingsbury and M. Palmer. From TreeBank to PropBank. 2002. 2

[21] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017. 2

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *arXiv:1602.07332*, 2016. 1

[23] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated Graph Sequence Neural Networks. In *ICLR*, 2016. 2, 3

[24] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *ECCV*, 2016. 2

[25] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Generating multi-sentence lingual descriptions of indoor scenes. In *BMVC*, 2015. 2

[26] H. Ling and S. Fidler. Teaching machines to describe images via natural language feedback. In *arXiv:1706.00130*, 2017. 2

[27] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual Relationship Detection with Language Priors. In *ECCV*, 2016. 1, 2

[28] K. Marino, R. Salakhutdinov, and A. Gupta. The more you know: Using knowledge graphs for image classification. *CVPR*, 2017. 2

[29] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995. 2

[30] B. A. Plummer, A. Mallya, C. M. Cervantes, and J. Hockenmaier. Phrase Localization and Visual Relationship Detection with Comprehension Linguistic Cues. *arXiv:1611.06641*, 2017. 2

[31] M. R. Ronchi and P. Perona. Describing Common Human Visual Actions in Images. In *BMVC*, 2015. 2

[32] M. Roth and M. Lapata. Neural Semantic Role Labeling with Dependency Path Embeddings. In *ACL*, 2016. 2

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 1

[34] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 2

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 5

[37] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *ACL*, 2015. 2, 4

[38] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *CVPR*, 2016. 2

[39] O. Vinyals, A. Toshev, S. Bengio, and Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. In *CVPR*, 2015. 2

[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1

[41] P. J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4):339–356, 1988. 4

[42] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *JMLR*, 2015. 2

[43] S. Yang, Q. Gao, C. Liu, C. Xiong, S.-C. Zhu, and J. Y. Chai. Grounded Semantic Role Labeling. In *NAACL*, 2016. 2

[44] B. Yao and L. Fei-Fei. Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions. In *CVPR*, 2010. 2

[45] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human Action Recognition by Learning Bases of Action Attributes and Parts. In *ICCV*, 2011. 2

[46] M. Yatskar, V. Ordonez, L. Zettlemoyer, and A. Farhadi. Commonly Uncommon: Semantic Sparsity in Situation Recognition. In *CVPR*, 2017. 2, 5, 6, 7, 8

[47] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. In *CVPR*, 2016. 1, 2, 4, 5, 6

[48] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual Translation Embedding Network for Visual Relation Detection. *arXiv:1702.08319*, 2017. 2

[49] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *ICCV*, 2015. 2

[50] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An Image Database for Deep Scene Understanding. *arXiv:1610.02055*, 2016. 1

[51] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. In *NIPS*, 2014. 1

[52] J. Zhou and W. Xu. End-to-end Learning of Semantic Role Labeling using Recurrent Neural Networks. In *ACL*, 2015. 2