GNR638: Machine learning for remote sensing II **Jul-Nov 2019**

End-term Project Report : Recurrent models for situation recognition

Student Name: Shubham Sharma & Laveena garg *Roll No: 18i190002 & 185090002*

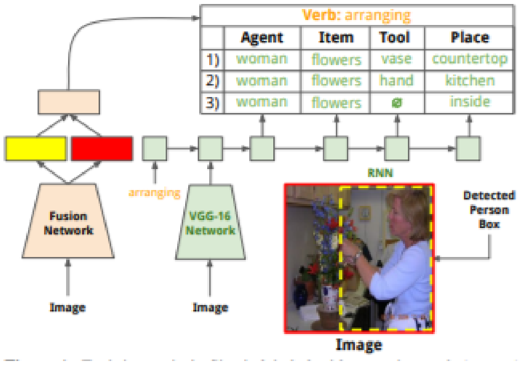
Abstract

This work proposes Recurrent Neural Network (RNN) models to predict structured “mage situations” – actions and noun entities fulfilling semantic roles related to the action. In contrast to prior work relying on Conditional Random Fields (CRFs), we use a specialized action prediction network followed by an RNN for noun prediction.

1 Introduction

Recognition of actions and human-object interactions in still images has been widely studied in computer vision. Earlier the focus was on predicting only the verbs, now the focus has been shifted to more general things like what is being done, who is doing what, where it is being done. The recently introduced imSitu Dataset generalizes the task of action recognition to ‘situation recognition’, the recognition of all entities fulfilling semantic roles in an instance of an action performed by a human or non-human actor.

We can see in figure 1 that each image in imSitu data-set is labeled with an action verb (orange), and each verb is associated with a unique set of semantic roles (bold black) which are fulfilled by noun entities present in the image (green). Each image has multiple annotations to account for the intrinsic ambiguity of the task. Our approach first uses the fusion network to predict the action verb. Then it feeds the verb and a visual feature from a separate network(Noun Prediction network) into an RNN to predict the noun roles in a fixed sequence conditioned on the action.



Verb: arranging

	Agent	Item	Tool	Place
1)	woman	flowers	vase	countertop
2)	woman	flowers	hand	kitchen
3)	woman	flowers	∅	inside

Figure 1: The Overall Architecture

2 Motivation

The use of RNNs for situation prediction is motivated by their popularity for tasks like image caption generation, where they have proven to be successful at capturing grammar and forming coherent sentences

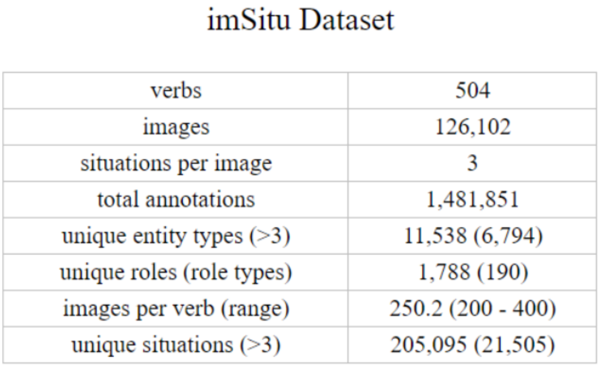
linking multiple concepts. The standard framework for caption generation involves feeding high-level features from a CNN, often trained for image classification on ImageNet, into an RNN that proceeds to generate one word of the caption at a time. Situation recognition involves the prediction of a sequence of noun entities for a particular action, so it can be viewed as a more structured version of the captioning task with a grammar that is fixed given an action.
Captioning task with a grammar that is fixed given an action. Above figure gives an overview of our best proposed system. First, we predict the action verb using the specialized action recognition architecture of, which fuses features from a detected person box with a global representation of the image.

3 Situation recognition task and method

Situations are based on a discrete set of action verbs V , noun entities N, and semantic roles R. Each verb v $\in$ V is paired with a unique frame f $\in$ F derived from FrameNet, a lexicon for semantic role labeling.
A frame is a collection of semantic roles $R_v \subset R$ which are associated with the verb v. For example, the semantic roles Agent, Item, Tool, Place $\subset$ R are associated with the verb arranging. In an instantiation of an action in an image, each semantic role is fulfilled by some noun n $\in N \cup \emptyset$, where $\emptyset$ indicates that the value is either not known or does not apply.The set of nouns N is derived from WordNet.
The authors who introduced situtation prediction also proposed a CRF-based approach for the task.The CRF normalization constant required for computing the loss during training is obtained by predicting the potentials for all valid tuples found in the training set and then summing them.
We take an alternate view of situation prediction by observing that given a verb v, the set of semantic roles R_v associated with it is fixed. For example, given the verb arranging, we know that we have to predict relevant noun entities for the semantic roles of $R_{arranging}$ =Agent, Item, Tool, Place. Conditioned on a given verb, if we assume some arbitrary but fixed ordering over these semantic roles, we can reduce the problem to that of sequential prediction of noun entities corresponding to the semantic roles.

4 Data-set

The data-set[imsitu] that has been used in the research paper and this project is imSitu data-set. The total number of verbs are 504, with the total number of images 126,102. The situation per image are 3. The total number of annotations are 1,481,851. The other parametric information can be seen from figure 2



imSitu Dataset

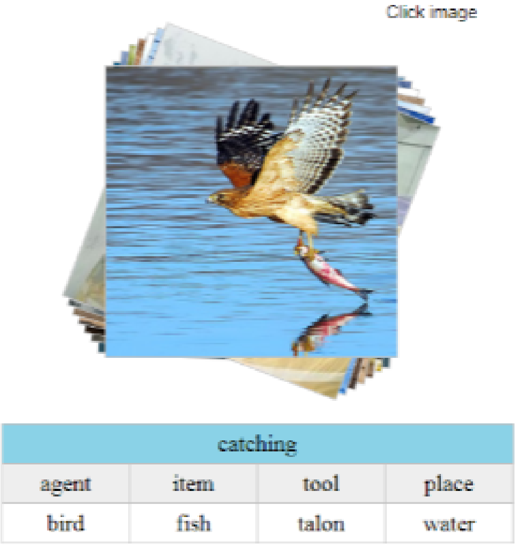
verbs	504
images	126,102
situations per image	3
total annotations	1,481,851
unique entity types (>3)	11,538 (6,794)
unique roles (role types)	1,788 (190)
images per verb (range)	250.2 (200 - 400)
unique situations (>3)	205,095 (21,505)

Figure 2: Parametric information of data-set

As the data-set was taking more than a day to pre-process, asking the instructor, we took a subset of the data-set that has the parametric information as follows:

- Verbs = 50
- Images = 13379
- Situations per image = 1
- Actions = 39
- Nouns = 2310

An example of image from imSitu data-set can be as seen in the figure 3‘ We have converter the extracted data in npy arrays and dictionaries and are saved in the drive. The data-sets can be found here
Link to data files: “https://drive.google.com/open?id=19UPzIyGQMAWGCqoaCyLHowrxm12YFgTQ”
Following are the files and their corresponding properties:



catching			
agent	item	tool	place
bird	fish	talon	water

Figure 3: An example from imSitu data-set

- **list_of_actions**: Contains the list of the actions
- **actions_to_ind**: Can access using actions_to_ind.item(). It contains 39 actions and their corresponding indices corresponding to all the images that we have. Eg. goal , item, tool
- **ind_to_actions**: It contains the index to actions dictionary for the actions
- **dictionary_fifty_verbs**: Contains the dictionary of the random fifty verbs that we selected from the data-set
- **ind_to_verbs**: Contains the index to verbs transformation
- **verbs_to_ind**: Contains the verbs to index transformation

- **list_of_nouns**: It consists of dictionary of all the 2310 nouns corresponding to all the actions in all the images in our data-set

- **nouns_to_ind**: Contains the nouns to index transformation

- **Ind_to_nouns**: Contains the index to nouns transformation

- **X_name**: Contains the names of all the images in a sequential order that is used in making X and corresponding Y

- **X_noun**: Contains the three different sets of possible nouns corresponding to the images we have. As we are working on only one

- **Y_nouns**: It is the uncategorical output for the second architecture to predict nouns in our project. It has a shape of (13379, 39)

- **Y_nouns_categorical**: It is same as above except the categorical version for each noun to corresponding action. It is the end output of our corresponding second network.

- **X**: Corresponding to X_name, it is the matrix of all the images having shape (13379, 256, 256, 3)

- **Y_verbs**: Corresponding to the X or X_name, it is the un_categorical matrix of verbs, which act as output in case of fusion network and one of the inputs in case of other network

5 Architecture

In this section, we'll explain the two architectures that are designed by taking idea from this research paper for action recognition task. The two architectures/networks are fusion network[fusion] and noun prediction network. Fusion network takes as input an image and predict the corresponding verb for the image and noun prediction network takes as input verb and image and predict noun corresponding to each of the 39 actions that are possible. Let us look as both the architectures one by one. The overall architecture of the network can be seen from figure 4

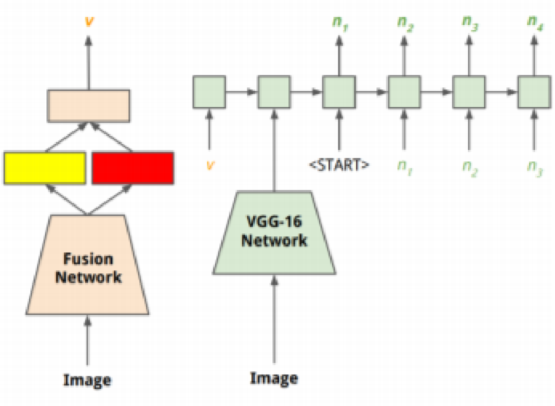


Figure 4: Basic Architecture of Network

5.1 Fusion Network

The fusion network is the network that is used for prediction of verbs from the image. It takes as input as image and has VGG[vgg] like architecture followed by two parallel layers, one from the whole image and other from the region in the bounding box with a RoI pooling layer. Both of these layers are then concatenated and has a softmax activation in the get to get the probability of each verb. The figure 5 shows an architecture for the fusion network.

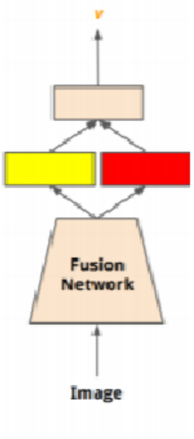


Figure 5: Fusion network

5.2 Noun Prediction Network

The other network is noun prediction network which is taking as input the image and the corresponding verb that we have got from the image. The architecture is an encoder decoder like architecture with the verb following an embedding layer and image following a VGG[vgg] like network and the latent layer is by concatenate both the output from the VGG[vgg] and the embedding layer. the decoder has a LSTM which is predicting the nouns for the verbs by using a certain patterns of actions. Also, we have used a dense layer to get better outputs other than the research papers. The figure 6 shows an example of noun prediction network.

5.3 Implementation Details

This section contains the implementation details of the networks that we have used:

- **Fusion Network**
  - The fusion network was trained for approximately 20 epochs using adam optimizer
  - Learning rate as 1e-4 for the first 15 epochs and 1e-5 for the last 5 epochs
  - Gave an accuracy of around 80% for the test set
- **Noun Prediction Network**
  - The network was trained for 200 epochs using mini-batch SGD with batch size = 32

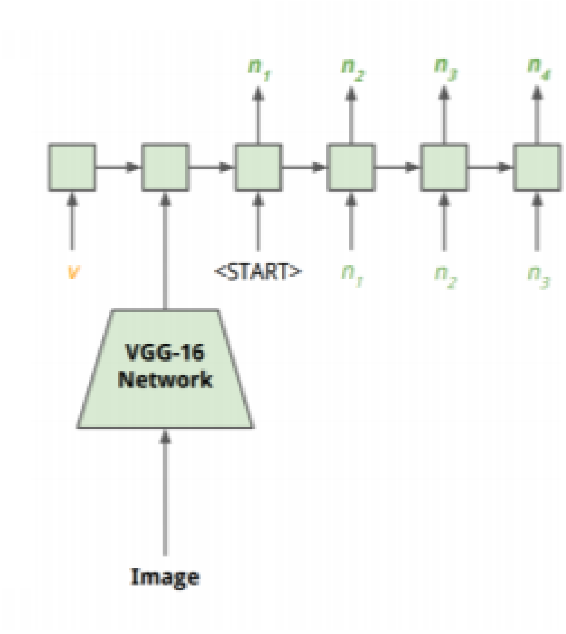


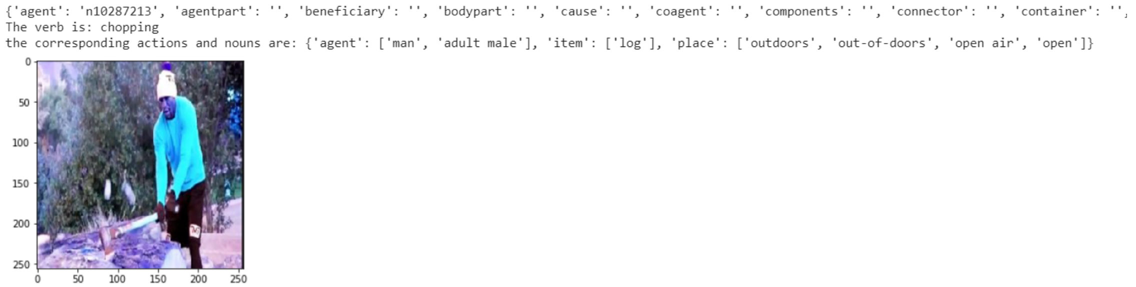
Figure 6: Noun Prediction Network

- Learning rate as 1e-2 for the first 100 epochs and 1e-4 for the next 100 epochs
- Gave an accuracy of around 98.03%

The main issue came with the training of noun prediction network. We tried different number of nodes for the latent layer and used google colab gpu for training. So, a trade-off had to be made on the no of layers and gpu capacity.

6 Result

We are getting an accuracy of 80% on the test data for the fusion network and an accuracy of 98.03% for the noun prediction network. An example from the output is shown in the figure 7



```
{'agent': 'n10287213', 'agentpart': '', 'beneficiary': '', 'bodypart': '', 'cause': '', 'coagent': '', 'components': '', 'connector': '', 'container': '',
The verb is: chopping
the corresponding actions and nouns are: {'agent': ['man', 'adult male'], 'item': ['log'], 'place': ['outdoors', 'out-of-doors', 'open air', 'open']}
```

Figure 7: A Prediction from the Network

7 Conclusions

This project/paper has introduced task of situation recognition as sequential prediction and conducted an extensive evaluation of RNN-based models on the imSitu. We can use recurrent models for situation recognition and are getting good results.

- RNNs-based methods are a straightforward fit for the task and work quite well.
- Using seperate networks for both the tasks gives better results

References

[1] Arun Mallya and Svetlana Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *European Conference on Computer Vision*, pages 414–428. Springer, 2016.

[2] Arun Mallya and Svetlana Lazebnik. Recurrent models for situation recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 455–463, 2017.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.