Neural Networks and Deep Learning - Visual Relationship Prediction

Binitha Radhakrishna Shetty
Dolly Agarwal
Simran Aggarwal

April 2021

Abstract

Recent success of activity recognition on video has led to an increased interest in the video captioning problem. The task of video captioning revolves around describing a video clip in an effort to capture its visual semantics. Similar to video captioning, the video visual relationship detection captures a wide variety of possible relationships between pairs of objects in images. Visual relations between objects are captured in the form of relation triplet ⟨subject,relation,object⟩, such as "dog-chasing-frisbee" and "cat-behind-man", providing a more comprehensive visual content understanding beyond objects. In this paper, we outline an end-to-end CNN-LSTM based sequence-to-sequence model for video visual relationship prediction. Along with discussing the architecture, we show that hyperparameter tuning can help improve the performance of a model. With our fine-tuned model, we achieved a MAP of 0.705 on Test set on Kaggle.

1. Introduction

Combining Natural Language Processing with Computer Vision to generate descriptions of visual (image/video) data is gaining traction as it provides a more comprehensive understanding of the image/video. One area of active research in this domain is visual relationship detection which involves detecting pairs of objects in an image and also classifying the relation or interaction between each pair. While objects are the core building blocks of an image, it is often the relationships between objects that determine the holistic interpretation. For example, a video with a person and a dog might involve the dog walking, running, or sitting (Figure 2). Understanding this diversity of relationships is central to a richer semantic understanding of our visual world. In this regard, as compared to still images, videos provide a more natural set of features for generating visual relations, such as the dynamic relations like ⟨A-follow-B⟩ and ⟨A-towards-B⟩, and temporally changing relations like ⟨A-chase-B⟩ followed by ⟨A-hold-B⟩.

In our project, given a set of object categories O and relationship categories R, we generate a relation triplet ⟨object1,relation,object2⟩ O × R × O. The dataset of this project contains 566 videos selected from ILVSRC2016-VID dataset and covers common objects1/objects2 among 35 categories and relationships among 82 categories. The dataset contains a widely-spread and imbalanced distribution of ⟨object1 , relation, object2⟩ triplets. And since relationships are determined between two objects, there is a larger skew of rare relationships as objects infrequently co-occur in images. This poses a challenge in visual relationship detection of having to learn from fewer examples.

To perform the video visual relationship detection, we exploit convolutional neural networks and recurrent neural networks, specifically LSTMs, which have demonstrated state-of-the-art performance in image caption generation. We first extract features from the video frames using a CNN model Resnet-50 which is pre-trained on 1.2M images from imagenet dataset. Then we have an encoder LSTM to extract spatial and temporal features from the 30 frames of a video. Our decoder LSTM model is trained on video-relation triplet pairs and learns to associate a sequence of video frames to a sequence of ⟨object1-relation-object2⟩ triplets in order to generate a description of the event in the video clip.

Our contributions are as follows:

1. Firstly, we implement an end-to-end CNN-LSTM based sequence-to-sequence model to extract features and predict relation triplets. We then define an ensemble model based on raw RGB and optical flow images trained on the same architecture.
2. Then, we demonstrate that data augmentation improves the performance of the model by balancing certain classes of objects and relations.
3. Moreover, we explore various hyper parameter tuning strategies and show that it helped achieve 0.705 MAP on Test set.
4. Finally, we make our code publicly available for reproducibility.

2. Background Information

Visual relationship prediction involves identifying the objects that occur in the video and understanding the interactions between the objects. There have been efforts in capturing human-object interaction [5–7] and action recognition [8] to learn models that distinguish between relationships where the subject is a human. In contrast, Visual relationship prediction is more general as the subject doesn't necessarily have to be a human and the predicate doesn't need to be a verb. Recent research works have focused on visual relationships in images. Some papers explicitly collected relationships in images [9-13] and videos [11, 14, 15] and helped models map these relationships from images to language. Some papers tried to improve the performance by leveraging language prior [16, 17] and regularizing relation embedding space [18, 19]. [20, 21] also studied the visual feature level connection among the components of relation triplet to exploit additional statistical dependency, but required O(NK) parameters for the modeling. Hence, in order to address these problems, [2] proposed a new vision task named VidVRD, which detects all visual relation instances in the form of the relation triplets and object trajectories in videos.
In our project, we refer to [24,25] where image caption generation models generate a fixed length vector representation of an

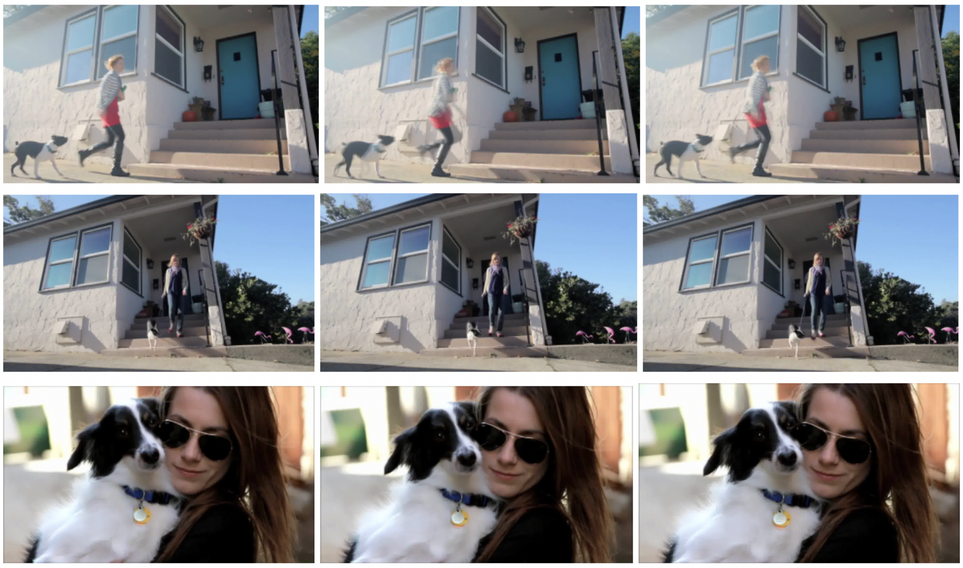

Figure 1: *Dog walking, running and sitting*

image by extracting features from a CNN and decode this vector into a sequence of words composing the description of the image.Since LSTM models are well suited for learning long-range dependencies, they were used as sequence decoders. We use two LSTMs one for encoding the frames of the video to extract temporal and spatial features and another to decode and produce relation triplets. As an improvement, we refer to [4] to build a model as an ensemble of the sequence to sequence models trained on raw images and optical flow images.

3. Dataset

Our dataset includes 566 videos from ILVSRC2016-VID split into 447 training and 119 test instances. As shown in Table 1, the objects and relationships can belong to 35 and 82 categories respectively. Each video clip in the dataset contains 30 frames. As objects infrequently co-occur in images, there is an imbalanced distribution of ⟨object1-relation-object2⟩ triplets. To tackle this, we incorporated informed data augmentation strategies to improve the dataset. We calculated the frequency of occurrence of each category of object and identified 12 objects that occur infrequently (less than 12 times) in the training dataset. We found that the occurrence of relations is very sparse, i.e 47 relations occur less than 4 times in our dataset. Since, it would be difficult to augment so many relations with only limited videos that we have, we focused on augmentation considering objects. We extracted the list of videos where both the object and subject belonged to unpopular object class. We listed 27 such videos and performed multiple data augmentation on each of them to make sure that each object appears at least 12 times in our dataset either in object1 or object2 or both. We found that some videos have image frames of a very low resolution i.e. images are either blurry or indistinct. So, we considered these points while deciding the transformations to apply for data augmentation. We considered rotate, scale out, transform, noise, blur, bright, contrast as our transformations. For every video in the list to augment, we randomly select a combination of a few transformations based on probability and applied those transformations and generated new video. After data augmentation we had 540 videos for our training and validation.

4. Our Approach

Our model uses an ensemble of both raw RGB and optical flow images trained over a CNN-LSTM architecture to generate the output relation triplet. Below we define our model architecture in detail.

4.1. Video and Text Representation

1. **RGB Frames:** Each video is represented as a set of 30 frames. Videos are pre-processed before being passed to ResNet for feature extraction. Every image is first scaled to a size of 224x224 and later normalised around mean and standard deviation.

2. **Optical Flow Frames:** Optical flow, or motion estimation, is a fundamental method of calculating the motion of image intensities, which may be ascribed to the motion of objects in the scene. From 30 frames of each video, 30 optical flow images are generated. Flow values are centered around 128 and multiplied by a scalar to have the flow values fall between 0 and 255. The calculated flow magnitude is added as a third channel to the flow image. These optical flow images are then preprocessed in a similar fashion as RGB images before being passed to the ResNet model.

3. **Relation Triplet (Text Input):** One-hot vector encoding was used to represent the ⟨object1-relation-object2⟩ triplets. We generate a vocabulary of 118 words consisting of 35 objects, 82 relations and a ⟨bos⟩ token which represents the beginning of sentence. Each word was mapped to a 118 dimension vector before being passed to the LSTM.

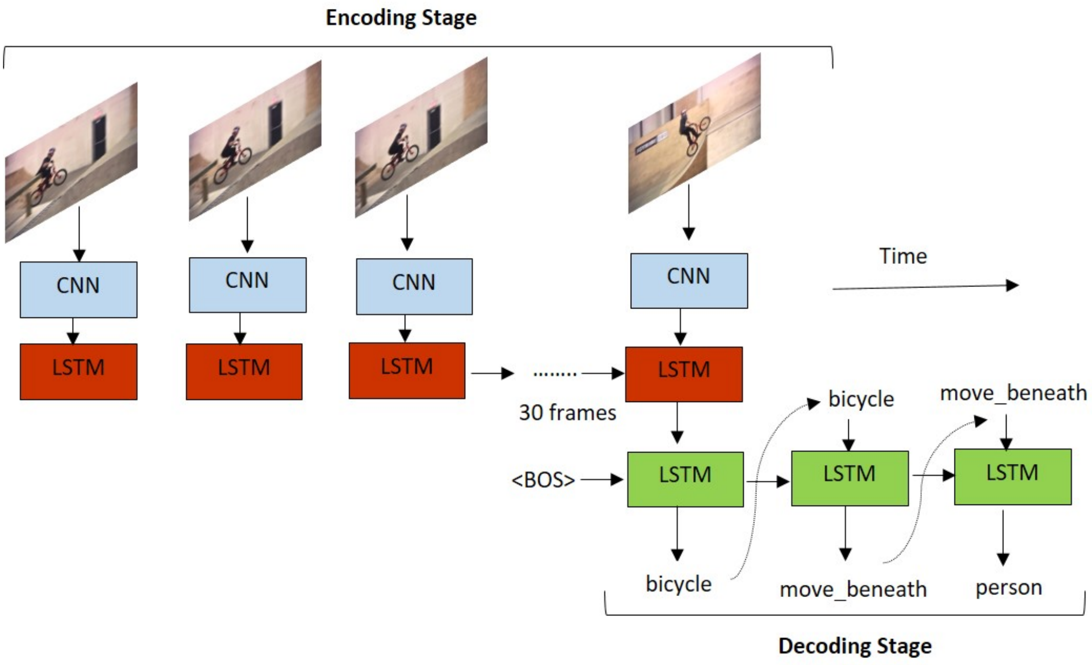

Figure 2: *CNN-LSTM based sequence-to-sequence model*

4.2. Bi-directional LSTMs for sequence modeling

The LSTM (Long Short Term Memory) networks are an extension of the basic RNNs (Recurrent Neural Networks), which are capable of learning the order dependence in sequences by remembering the past data in memory. Bidirectional LSTMs are an extension of traditional LSTMs where two LSTM layers are put together. One layer learns the input sequence in the normal time order and the other layer learns in the reverse time order. This structure allows the network to learn both the forward and backward information about the input sequence at each time step which can improve model performance on sequence classification problems. This may also result in faster and fuller learning of the problem. Hence, we employ bi-directional LSTMs to solve the task at hand.

Equation 1 represents the equations for LSTM RNN, where given an input x_t at timestep t, the LSTM computes a hidden state h_t and a cell state c_t that carries relevant information throughout the processing of the sequence. Here, i_t represents the input gate, f_t represents the forget gate, o_t represents the output gate, σ represents the sigmoid function, tanh is the hyperbolic tangent non-linearity, and the weight matrices denoted by W_{ij} and biases b_j are the trained parameters.

$$\begin{aligned} f_t &= \sigma(W_f^t[h_{t\text{-}1}, x_t] + b_f) \\ o_t &= \sigma(W_o^t[h_{t\text{-}1}, x_t] + b_o) \\ i_t &= \sigma(W_i^t[h_{t\text{-}1}, x_t] + b_i) \\ g_t &= tanh(W_g^t[h_{t\text{-}1}, x_t] + b_g) \\ c_t &= f_t * c_{t\text{-}1} + i_t * g_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{1}$$

In bi-directional RNN, we preprocess the inputs to get input and reversed input. Equation 2 represents the equation for bi-directional RNN, for t=1 to T, where T is the number of elements.

$$\begin{aligned} \overrightarrow{\mathrm{h}} &= tanh(\overrightarrow{\mathrm{U}}^T \overrightarrow{x}_t + \overrightarrow{\mathrm{W}}^T \overrightarrow{\mathrm{h}}_{t\text{-}1} + \overrightarrow{b}) \\ \overleftarrow{\mathrm{h}} &= tanh(\overleftarrow{\mathrm{U}}^T \overleftarrow{\mathrm{x}}_t + \overleftarrow{\mathrm{W}}^T \overleftarrow{\mathrm{h}}_{t\text{-}1} + \overleftarrow{b}) \\ \overleftarrow{\mathrm{h}} &= roll(\overleftarrow{\mathrm{h}}) \\ h_t &= [\overrightarrow{\mathrm{h}}, \overleftarrow{\mathrm{h}}] \end{aligned} \tag{2}$$

4.3. Encoder-Decoder Architecture

4.3.1. Encoder

The encoder network is defined as a combined CNN-RNN architecture, which takes as input a batch of videos and generates a sequential spatial representation for each video. This representation essentially extracts the temporal correlation between all the frames of a video.

Each video containing 30 frames is represented as a sequence of features of dimension 30 x 2048 using the pre-trained Resnet50 model. The fully-connected classification layer of ResNet50 is removed and high-level features are extracted using the last Convolution layer. These features are then passed through a Bi-directional LSTM Layer that encodes the video into a latent vector representation.

In our architecture, we directly pass the 30 x 2048 dimension cnn features through the Bi-directional-LSTM. The LSTM cell takes the feature representation x_t of a frame at time step t, and generates the hidden state h_t and the cell state c_t. These states are then passed to the lstm cell at time step t+1 along with the feature representation x_{t+1} for the next input frame. At time step 30, the lstm cell generates h_{30} and c_{30} which represent the temporal encoding of all the 30 frames that the lstm has observed till time step 30. The hidden and cell state from the last time step are then passed as input to the Decoding stage.

4.3.2. Decoder

In the decoder network we define 3 LSTM Cells, which are used to decode the object1, relation and object2 respectively. We pass the hidden state from one LSTM Cell to the next, in order to retain the entire encoded information at each step. In order to make this information sharing easier, we combine the 35 objects and 82 relations together along with a ⟨bos⟩ (beginning of sentence) token, to form one vocabulary of 118 tokens.

In this network, the first LSTM Cell is given as input the ⟨bos⟩ token, which in essence marks the beginning of the decoding stage. The hidden state and cell state generated in the encoding phase are also passed as the initial state of this cell. It then generates as output the probability distribution over the defined vocabulary for first object prediction.

For the second LSTM Cell, the actual object1 from the training label is passed as input along with the hidden state and cell state generated by the first LSTM Cell. This cell then generates the probability distribution for the relation prediction. Similarly, the probability distribution for object2 is generated from the third and final LSTM cell.

4.4. Ensemble Model

Using the basic architecture defined above we train 2 different models, one using the raw RGB images and other using the optical flow images. Optical Flow captures the motion of objects between the consecutive frames of a video. We use Dense optical flow to compute the optical flow vectors for each pixel of the 30 frames present in the input video. In order to calculate optical flow, we use the Farneback method. In this method, a polynomial expansion transform is used to estimate the image frame windows and these transformations are observed under motion to calculate the dense optical flow.

Finally during the inference phase, we consider only RGB models for object1 and object2 detection but combine the outputs from both RGB and Flow models by averaging the probability for relation prediction. Then, using the probability values, we extract the top 5 predictions for ⟨object1-relation-object2⟩ triplets.

5. Experiments

5.1. Experimental Setup

All experiments for this task were performed using Google Colab GPU.
For the defined encoder-decoder architecture, we performed hyper-parameter tuning for different batch sizes, optimizers and learning rates. Model's performance was analysed for different settings with batch sizes [1, 8, 32], optimizers [Adam, Adamax, AdamW] and learning rate [e-03, e-04]. From the performance analysis we found that our model performs best for a batch size of 8, Adamax optimizer and a learning rate of e-03. For the submission, we have used the same setting to train our final model.
In addition to the hyper-parameter tuning, we applied a set of transformations to each video during the training phase. These transformations were decided based on our study of the training and test videos.
We also split our initial training dataset into a train set and validation set of 0.8 and 0.2 respectively in order to monitor overfitting. We save the model with the lowest validation loss and stop the training process as soon as it reaches a plateau.

5.2. Experimental Results

For our Kaggle submissions, we picked the outputs from the model with the lowest validation loss. Before using a bidirectional LSTM, we also tested the performance with unidirectional LSTM. Below we report the different model settings that were submitted along with their score.

Model Setting	**Score**
Uni-directional LSTM, Adam Optimizer	**0.68394**
Bi-directional LSTM, Adamax Optimizer	**0.69589**
Bi-directional LSTM, Ensemble with Optical Flow	**0.70536**

Table 1: *MAP score on Test Data by different models*

6. Conclusion

In conclusion, we observed that bi-directional LSTMs perform better in sequence prediction tasks as compared to unidirectional LSTMs. The performance of our CNN-LSTM architecture shows that the model was able to perform well for the task of object detection in videos, whereas it lacked in predicting the relationships. One of the reasons behind this is the fact that the relations defined in our problem statement are more fine-grained and the CNN-LSTM architecture is not able to perform well on such a task. Another factor is the imbalance in the given dataset. In the given dataset, we observed that more than 45 of relation classes had less than 4 samples. While we did see an improvement in the model performance after incorporating Optical Flow, the train data was not enough for the model to learn the flow features. Given more time, we believe models like I3D and temporal segmentation could be explored for predicting the fine-grained actions.

7. Statement of Contributions

Binitha, Dolly and Simran contributed equally to the writing of the report. Binitha was involved in cluster setup, data augmentation and optical flow generation. Dolly and Simran were equally involved in data augmentation, optical flow generation, model architecture design and hyper-parameter tuning.

8. References

1. Lu C., Krishna R., Bernstein M., Fei-Fei L. (2016) Visual Relationship Detection with Language Priors. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_51

2. Shang, Xindi & Ren, Tongwei & Guo, Jingfan & Zhang, Hanwang & Chua, Tat-Seng. (2017). Video Visual Relation Detection. 1300-1308. 10.1145/3123266.3123380.

3. Krishnamoorthy, N. & Malkarnenkar, G. & Mooney, R.J. & Saenko, Kate & Guadarrama, Sergio. (2013). Generating natural-language video descriptions using text-mined knowledge. Proceedings of the 27th AAAI Con-

ference on Artificial Intelligence, AAAI 2013. 1. 541-547.

4. S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell and K. Saenko, ”Sequence to Sequence – Video to Text,” 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4534-4542, doi: 10.1109/ICCV.2015.515.
5. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 433–440
6. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: Computer Vision and Pattern Recogni- tion (CVPR), 2010 IEEE Conference on, IEEE (2010) 17–24
7. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed represen- tation of pose and appearance. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3177–3184
8. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(10) (2009) 1775–1789
9. Ramanathan, V., Li, C., Deng, J., Han, W., Li, Z., Gu, K., Song, Y., Bengio, S., Rossenberg, C., Fei-Fei, L.: Learning semantic relationships for better action retrieval in images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1100–1109
10. Guadarrama,S.,Krishnamoorthy,N.,Malkarnenkar,G., Venugopalan,S.,Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 2712–2719
11. Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., Pinkal, M.: Ground- ing action descriptions in videos. Transactions of the Association for Computational Linguistics 1 (2013) 25–36
12. Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., Mooney, R.: Inte- grating language and vision to generate natural language descriptions of videos in the wild. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING), August. (2014)
13. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 702– 709
14. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: Proceedings of the 24th CVPR, Citeseer (2011)
15. Zitnick, C.L., Parikh, D., Vanderwende, L.: Learning the visual interpretation of sentences. In: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE (2013) 1681–1688
16. Xiaodan Liang, Lisa Lee, and Eric P. Xing. 2017. Deep Variation-Structured Reinforcement Learning for Visual Relationship and Attribute Detection. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE.
17. Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In European Conference on Computer Vision. Springer, 852–869.
18. Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual Translation Embedding Network for Visual Relation Detection. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE.
19. Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. 2017. PPR-FCN: Weakly Supervised Visual Relation Detection via Parallel Pairwise R-FCN. In IEEE International Conference on Computer Vision. IEEE.
20. Bo Dai, Yuqi Zhang, and Dahua Lin. 2017. Detecting Visual Relationships With Deep Relational Networks. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE.
21. Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao’ou Tang. 2017. ViP-CNN: Visual Phrase Guided Convolutional Neural Network. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE.
22. J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recur- rent convolutional networks for visual recognition and de- scription. In CVPR, 2015. 1, 2, 3, 4
23. O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. CVPR, 2015. 1, 2, 4
24. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra- manan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- mon objects in context. In ECCV, 2014. 6
25. J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep captioning with multimodal recurrent neural networks (m- rnn). arXiv:1412.6632, 2014. 1
26. N.Srivastava,E.Mansimov,andR.Salakhutdinov.Unsupervised learning of video representations using LSTMs. ICML, 2015.
27. I.Sutskever,O.Vinyals,andQ.V.Le.Sequencetosequence learning with neural networks. In NIPS, 2014
28. J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. CVPR, 2015.
29. A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In CVPR, 2015.