CS5242 Final Project: Video Captioning

Shen Yichao
A0225540N
e0576174@u.nus.edu

Xiong Kexin
A0196551X
e0389037@u.nus.edu

Zhang Ziyang
A0225577W
sjtuzzy@gmail.com

***Abstract*—In this project, we performed video visual relationship prediction on a video dataset, which contains sequences of images (also knowns as video clips). By using InceptionResnetV2 as a pre-trained model to extract video features, then passing the features throught a GRU model with attention, we got a score of 0.70 on the Kaggle competition.**

I. Introduction

Video Captioning is a task to generate natural-language utterance (usually a sentence) that describes the visual semantic content of a video. It has great impact in various domains such as video indexing and retrieval [7] [8], visual modality usage in conversational recommender systems (CRSs) [9]–[14], as well as assisting those with visual impairments by converting visual signals into information that can be conveyed using text-to-speech technology.

Computer vision and natural language processing are two key areas of artificial intelligence. Nowadays the most popular research topics to bridge vision and language are visual question answering(VQA) and image/video captioning. VQA follows the same architecture in textual question answering [15], which includes answer retrieval and multi-hop reasoning [17] based on visual content. While captioning is more like a translation from visual content to textual content which requires less retrieval and reasoning. Video captioning is much more challenging than its twin problem ”image captioning”, which has been widely studied [3] [4] [5] [6]. This is not only because substantially frames in videos are more informative than still image, but also it’s difficult to capture the temporal information to understand the video as a dynamic system. Variable length input of frames is another challenge, which can be resolved by related approaches such as holistic video representations, pooling over frames, or sub-sampling on a fixed number of input frames.

In Video Captioning tasks, a video is considered as a series of frames and a caption is considered as a series of words. Data pre-processing typically involves cutting each video into frames and transforming captions into word embedding vectors. Due to the sequential data structure, video captioning usually follows encoder-decoder framework based on sequence to sequence (seq2seq) architecture [1], which has also been widely used in Machine Translation [20] [21], Neural Question Generation [19], Question Answering [15] and Conversational techniques [18]. In video captioning, seq2seq architecture employs an encoder (typically performed by CNNs or RNNs) to analyze and extract useful visual context features from the source video, and a decoder to generate the caption sequentially.

However, one main challenge of video captioning is that natural language results are difficult to evaluate, which leads to another research topic: visual relationship prediction/detection (VRD) [2]. Visual relationship prediction focuses on visual feature extraction, generating triplets of SVO (Subject, Verb, Object) to describe the content in the image/video. Models are trained to generate captions with respect to relational information between objects, which are more dense and informative than arbitrary natural sentences. Researchers only need to measure whether the generated SVO (Subject, Verb, Object) triplets coheres with ground truth, and can ignore other visual and language details.

In this project, each video has been given in a series of 30 frames. Our task is to perform video visual relationship prediction to learn the visual relations of interest in a video. We used a CNN-RNN framework which is trained end-to-end to capture the trajectories of

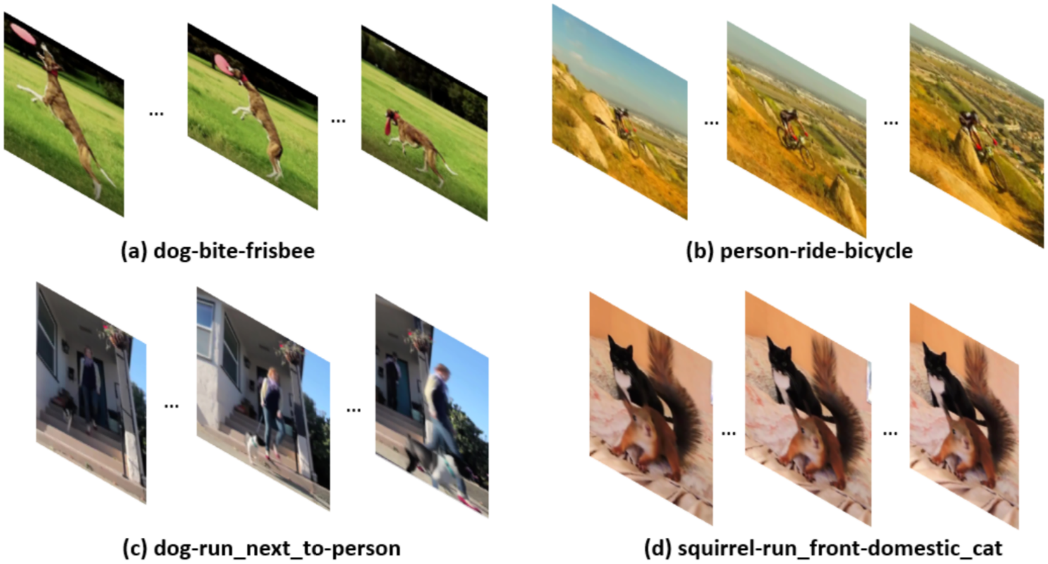
(a) dog-bite-frisbee
(b) person-ride-bicycle
(c) dog-run_next_to-person
(d) squirrel-run_front-domestic_cat

Fig. 1. **Video visual relationship prediction**

the object1 and object2 and relationship bwteen them, which are represented by a triplet in the form of ⟨Object 1⟩ - ⟨Relationship⟩ - ⟨Object 2⟩. Our model get the final score of 0.70 in Kaggle competation.

April 10th 2021

II. Background

A. LSTMs for sequential decoding

One way to handle variable-length input and output is to first encode all the input of frames, one at a time, to represent the video using a latent vector representation, and then decode from that representation to a sentence, one word at a time.

In Long Short Term Memory RNN (LSTM), for an input x_t at time step t, the LSTM computes a hidden/control state h_t and a memory cell state c_t which is an encoding of everything the cell has observed until time t

$$\begin{aligned} i_t &= \sigma\left(W_{xi} x_t + W_{hi} h_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + b_f\right) \\ o_t &= \sigma\left(W_{xo} x_t + W_{ho} h_{t-1} + b_o\right) \\ g_t &= \phi\left(W_{xg} x_t + W_{hg} h_{t-1} + b_g\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \phi\left(c_t\right) \end{aligned}$$

where σ is the sigmoidal non-linearity, ϕ is the hyperbolic tangent non-linearity, $\odot$ represents the element-wise product with the gate value, and the weight matrices denoted by W_{ij} and biases b_j are the trained parameters.

Thus, in the decoding phase, it defines a distribution over the output sequence Y $(y_1, \ldots, y_m)$ given the input sequence X as $p(Y \mid X)$ is

$$p\left(y_1, \ldots, y_m \mid x_1, \ldots, x_n\right) = \prod_{t=1}^{m} p\left(y_t \mid h_{n+t-1}, y_{t-1}\right)$$

where the distribution of $p\left(y_t \mid h_{n+t}\right)$ is given by a softmax over all of the words in the vocabulary.

B. CNN for video encoding

The fundamentals of image understanding lie in identifying basic shapes or geometry of the entities, which requires us extracting enough parts or patterns with meanings. These parts and patterns are defined as image features. Since the deep learning technology rises, it remains its fantacy in image and video feature extraction. Feature extraction field has been fully explored and is widely used for tasks like object recognition, image alignment and stitching, 3D stereo reconstruction, and video captioning.

A Convolutional Neural Network (CNN) is a feed-forward artificial neural network inspired by animal visual cortexes. It is designed for visual imagery, while can also be used in many other fields like language and

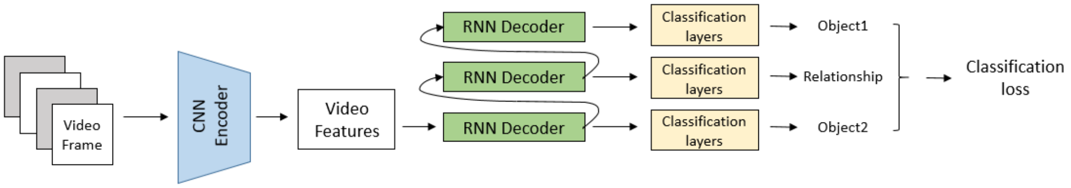

Fig. 2. **Model Structure**

vocal processing. The most significance characteristics of CNNs are weight sharing and hierarchical connections with automatic self-training. There're various types of CNNs which have been proved strenghful in visual feature extraction, such as VGG16, ResNet50, InceptionV3 and so on. In tensorflow, these models are all pre-trained on imagenet and can be used as a feature extractor .

In our project, we follow the CNN-encoder LSTM-decoder architecture in [16]. Each video is considered as sequence of frames, encoded by a pre-trined CNN and flattened, then fed into an RNN decoder (here we use GRU) with BahdanauAttention, to predict a triplet of $\langle$Object 1$\rangle$ - $\langle$Relationship$\rangle$ - $\langle$Object 2$\rangle$ from 35 object names and 82 relationships given. While the final prediction problem can be modeled as sequential labeling task, we choose simple classification model as the benchmark. We use Multilayer Perceptrons (MLPs) as the classifiers, for each decoding time step to choose one of the label for prediction.

III. METHODOLOGY

A. *Data Pre-processing*

In this part, we have completed two parts of work. The first one is transforming the video caption jsons (35 object names and 82 relationships) to 117 labels.

The next part is we have reshaped the images size to match the input size of pre-trained CNN. For InceptionResnetV2 we use, it has an input size of (299,299).

There're still some tricks left for us to try for data preprocessing. One possible work is try to create augmented dataset like rotation the image to multiple degrees or do some flipping for the original image. We haven't do this in our project but it might be a way to improve our final result.

B. *Model Structure*

We have experimented with many different pre-trained models available in Tensorflow Keras, including InceptionV3, Xception, ResNet101 and so on. Out of all of those pre-trained models, we chose InceptionResnetV2 excluding final linear layer as our visual feature extractor to generate representations of each frame.

At the beginning, we planed to use CNN as spatio encoder to extract visual feature within each frame, and use RNN as temporal feature encoder to capture dynamic between different frames. However, after dozens of experiments we found that, the more complex the encoder is, the more easily it gets overfitting. Therefore, we finally simplified our encoder part to a pre-trained CNN + one fully connected layer.

The CNN outputs of each video are flattened into a 2D vector, later fed into an RNN decoder, which generates $\langle$Object 1$\rangle$ - $\langle$Relationship$\rangle$ - $\langle$Object 2$\rangle$ in separate timesteps. The RNN model we use is a single layer Gated Recurrent Unit (GRU) lasting three time steps. The output at each timestep will be passed into a linear classifier to predict object1, relationship and object2 separately.

Besides, we added an Attention mechanism into the decoder that allows the model to focus on important information when choosing word in each timestep. Another important thing is that we removed teacher forcing in the decoder. The reason why we removed it is that, we found the model is easy to get overfitting when trained with teacher forcing, and not general enough on test dataset.

C. *Training strategies*

The parameters of our project is Batch_size = 64, Buffer_Size = 1000 Embedding_dim = 500, Units=512, Vocab_size = 1001, Num_steps = 447.

The way we try to solve vocab size not equal to labels + 1 will explain in the evaluation part.

D. Evaluation

Test prediction is the index of the top 5 probabilities among all the softmax outputs. Because our vocab size is 1001, we need to list the prediction index in decreasing probability order and choose the top 5 valid predictions.

IV. Experiment results

With InceptionResnetV2 pre-trained model, CNN encoder and RNN decoder. We get the final Accuracy of 0.70. Besides, in the training process, we found that early stopping can be used to stop overfitting.

V. Conclusion

In this project, we have experimented with different pre-trained CNN models, different framework structures and different sets of parameters to help with relationship prediction. We have also discussed that the overfitting problem may be due to the small size of dataset (argumented dataset may help).

In summary, we chose InceptionResnetV2 pre-trained model as CNN encoder. With GRU decoder and three linear classifiers for object1, relationship and object2, we got the final test MAP above 0.70. Besides, in the training process, we found that early stopping can effectly prevent overfitting.

VI. Statement of contribution

Shen Yichao: Implementation and training of models.

Xiong Kexin: Implementation and training of models, Report refining.

Zhang Ziyang: Report Writing.

Acknowledgment

In the process of completing this project, the teaching assistant guides and supervises us every week, and provides some learning resources for our reference. The authors of the paper also open source their own code so that we can better understand the content of the paper. We express our sincere gratitude to those who have helped us.

References

[1] Venugopalan S, Rohrbach M, Donahue J, Mooney R, Darrell T, Saenko K. Sequence to sequence-video to text. InProceedings of the IEEE international conference on computer vision pp. 4534-4542. 2015.

[2] Liu, Chenchen, Yang Jin, Kehan Xu, Guoqiang Gong, and Yadong Mu. "Beyond short-term snippet: Video relation detection with spatio-temporal global context." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10840-10849. 2020.

[3] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6077-6086).

[4] Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7008-7024).

[5] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

[6] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence, 39(4), 652-663.

[7] Hu, W., Xie, N., Li, L., Zeng, X., & Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 41(6), 797-819.

[8] Zhang, D., & Nunamaker, J. F. (2004). A natural language approach to content-based video indexing and retrieval for interactive e-learning. IEEE Transactions on multimedia, 6(3), 450-458.

[9] Yu, T., Shen, Y., & Jin, H. (2019, July). A visual dialog augmented interactive recommender system. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 157-165).

[10] Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T. S. (2021). Advances and Challenges in Conversational Recommender Systems: A Survey. arXiv preprint arXiv:2101.09459.

[11] Lei, W., He, X., de Rijke, M., & Chua, T. S. (2020, July). Conversational Recommendation: Formulation, Methods, and Evaluation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2425-2428).

[12] Lei, W., He, X., Miao, Y., Wu, Q., Hong, R., Kan, M. Y., & Chua, T. S. (2020, January). Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In Proceedings of the 13th International Conference on Web Search and Data Mining (pp. 304-312).

[13] Lei, W., Zhang, G., He, X., Miao, Y., Wang, X., Chen, L., & Chua, T. S. (2020, August). Interactive path reasoning on graph for conversational recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2073-2083).

[14] Li, S., Lei, W., Wu, Q., He, X., Jiang, P., & Chua, T. S. (2020). Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users. arXiv preprint arXiv:2005.12979.

[15] Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., & Chua, T. S. (2021). Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. arXiv preprint arXiv:2101.00774.

[16] Venugopalan, Subhashini, et al. ”Sequence to sequence-video to text.” Proceedings of the IEEE international conference on computer vision. 2015.

[17] Zhang, Y., Zhang, X., Wang, J., Liang, H., Jatowt, A., Lei, W., & Yang, Z. (2020). GMH: A General Multi-hop Reasoning Model for KG Completion. arXiv preprint arXiv:2010.07620.

[18] Lei, W., Jin, X., Kan, M. Y., Ren, Z., He, X., & Yin, D. (2018, July). Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1437-1447).

[19] Pan, L., Lei, W., Chua, T. S., & Kan, M. Y. (2019). Recent advances in neural question generation. arXiv preprint arXiv:1905.08949.

[20] Klubička, F., Toral, A., & Sánchez-Cartagena, V. M. (2017). Fine-grained human evaluation of neural versus phrase-based machine translation. arXiv preprint arXiv:1706.04389.

[21] Lei, W., Xu, W., Aw, A., Xiang, Y., & Chua, T. S. (2019, November). Revisit Automatic Error Detection for Wrong and Missing Translation–A Supervised Approach. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 941-951).