HOW SAMPLING RATE AFFECTS CROSS-DOMAIN TRANSFER LEARNING FOR VIDEO DESCRIPTION

Yu-Sheng Chou[1], *Pai-Heng Hsiao*[2], *Shou-De Lin*[1], *Hong-Yuan Mark Liao*[3]

[1]Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan
[2]Memorence A.I., Taipei City, Taiwan
[3]Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

Translating video to language is very challenging due to diversified video contents originated from multiple activities and complicated integration of spatio-temporal information. There are two urgent issues associated with the video-to-language translation problem. First, how to transfer knowledge learned from a more general dataset to a specific application domain dataset? Second, how to generate stable video captioning (or description) results under different sampling rates? In this paper, we propose a novel temporal embedding method to better retain temporal representation under different video sampling rates. We present a transfer learning method that combines a stacked LSTM encoder-decoder structure and a temporal embedding learning with soft-attention (TELSA) mechanism. We evaluate the proposed approach on two public datasets, including MSR-VTT and MSVD. The promising experimental results confirm the effectiveness of the proposed approach.

Index Terms— Video Description, Transfer Learning, Sampling Rate, Temporal Representation

1. INTRODUCTION

Using computer to automatically produce video description has many real-world applications and it is indispensable for modern life. To make an appropriate description on a relatively "big" data like video, a good way is to introduce "sampling." The key concept of sampling is to perform processing on a smaller amount of sampled frames, but a close-to-accurate description can still be obtained. Figure 1 shows how sampling rate affect the outcome of video description. In Figure 1 (a), the sampling is sparser. The video description corresponding to this sampling is "a car is crashing." If we take denser sampling as indicated in Figure 1 (b), the resultant video description becomes "a car is dashing with a bus." From the above example, it is apparent that if the sampling interval is too large, some important messages may be missed. How to tolerate the sampling rate change and still generate correct description will be a great challenge for video description researchers.

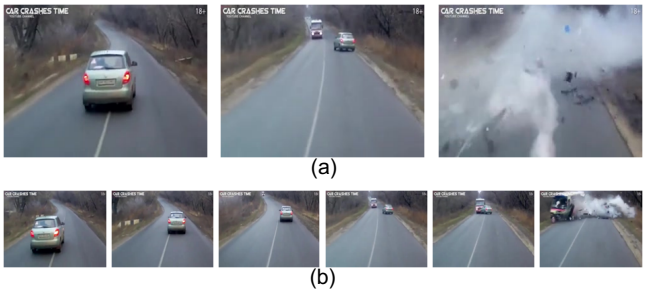

Fig. 1: Examples of automatic video description at different sampling rates on MSR-VTT. (a) set of frames grabbed by sparser sampling rate: "a car is crashing." (b) set of frames grabbed by denser sampling rate: "a car is dashing with a bus."

Recent development of CNN-based video captioning (description) has great impact on automatic expansion of video-to-language resources. The CNN-RNN encoder-decoder structure is composed of a CNN-based extractor and a Recurrent Neural Network (RNN)-based model for mapping the representations between a source sequence and a target sequence. A CNN-RNN framework incorporates both chronological variable-length sequences, which are video frames and words [1, 2, 3, 4, 5, 6, 7, 8], in its main structure. To make global temporal representation of videos, Venugopalan et al. [3] proposed the use of a stacked Long Short Term Memory (LSTM) model to achieve the goal. To use this model, one has to input video frames into the encoder side and then decode the mapped hidden representation as a sentence at the decoder side. In the LSTM model they proposed, all input video clips are all sampled to a fixed 80-frame size during the training process, no matter how long the original is. Therefore, for source videos with different lengths, the corresponding sampling rates may be different since they are all ultimately sampled to a fixed length of 80. This is equivalent to applying different sampling rates for different input videos.

To build a video description (captioning) system that can tolerate different sampling rates, temporal representation invariance among datasets is one of the fundamental problems

needs to be solved. In this paper, we propose a transfer learning technique to solve the above mentioned problem. This transfer learning technique can perform global transfer learning on the temporal representation of a video by a stacked LSTM encoder-decoder architecture. To ensure the proposed system can generate stable temporal representations under different sampling rates, we propose a temporal embedding learning with soft attention (TELSA) mechanism, to learn the temporal embedding and adjust softly the representations for the target domain.

2. THE PROPOSED APPROACH

2.1. Problem Formulation

A video-to-language translation system can be formulated as conditional probability $p(y|x)$ of a predicted sentence $(y_1, ..., y_m)$ given an input sequence $(x_1, ..., x_n)$ extracted from sampling video frames with CNN and it is a variable-length input and output problem. To solve the problem of this sort, the main idea of the encoder-decoder framework is to encode the input sequence $(x_1, ..., x_n)$ as a latent temporal representation z into decoder for sequence modeling. Therefore, the above conditional probability can be defined as the product of conditional probabilities as follows:

$$p(y|x) = \prod_{t=1}^{m} p(y_t|y_{n+t-1}, z) \tag{1}$$

It is reasonable to choose Recurrent Neural Network (RNN) to handle the above mentioned variable-length sequence problem. However, traditional RNN is error-prone when learning sequences that contain long-term dependency. Usually, a sequence with long-term dependency suffers from the vanishing and exploding gradient problem [9]. Fortunately, the Long Short Term Memory (LSTM) [10] unit, which consists of a memory cell c_t and three gates (input i_t, output o_t and forget f_t), is a solution to the above problem. The memory unit through the gates can learn when to forget or write previous hidden states and propagate the explicit information for long-term dynamics.

Consequently, we can re-formulate Equation (1) and rewrite it as follows:

$$p(y|x) = \prod_{t=1}^{m} p(y_t|h_{n+t-1}, y_{n+t-1}, z) \tag{2}$$

In the encoding phase, the LSTM maps the input dynamics as fixed length temporal representation vector z. We then decode the vector at each time step t as a vocabulary-sized vector and output the word distribution y_t by applying $softmax$ function over the training corpus.

2.2. Transfer Learning on Temporal Representation

To transfer the knowledge of source domain to target domain, we make use of the statistics calculated from videos with different lengths and append the soft-attention mechanism. In this way one is able to adaptively learn temporal representation z and decrease domain discrepancy caused by using different sampling rates. Instead of using "attention" to derive the weighted average in encoding phase [11, 4], we take the dynamic $\alpha_i^{(t)}$ to softly adjust the input representation at each time step t such that:

$$z = \prod_{t=1}^{n} p(y_t|h_{t-1}, \alpha_i^{(t)}\widetilde{v_i}) \tag{3}$$

$$\widetilde{v_i} = E(v_i) \tag{4}$$

where v_i represents the element of the set sampled from CNN feature extractor, and it is embedded as a low-dimensional vector $\widetilde{v_i}$ by the learnable embedding function $E(*)$, which is a fully-connected layer.

The dynamic weights $\alpha_i^{(t)}$ are used to adjust the representation of each frame based on new input size (e.g. 40 to 80). The dynamic weights should be normalized whenever the number of frame is changed. This enables the capability of adaptive temporal representation on the target domain. The weights $\alpha_i^{(t)}$ need to be derived by consulting the global observation and the previous hidden state. Thus, the relevance score $e_i^{(t)}$ plays this role and can be computed by:

$$e_i^{(t)} = w^T tanh(W_a h_{t-1} + E(v_i) + b_a) \tag{5}$$

where w, W_a and b_a are the parameters used to reveal the relevance of $\alpha_i^{(t)}$. The relevance scores $e_i^{(t)}$ involved in $\alpha_i^{(t)}$ can summarize previous hidden state h_{t-1} and embed representation $E(v_i)$ to reflect the transfer relevance of the i-th temporal feature. Specifically, we consider the embedding of visual representation into the score, and it can jointly learn the other parameters that are associated with $\alpha_i^{(t)}$.

In a transfer case, the relevance score $e_i^{(t)}$ must fit into the target domain with distinct sampling length n by normalizing them to derive $\alpha_i^{(t)}$:

$$\alpha_i^{(t)} = exp\{e_i^{(t)}\} / \sum_{j=1}^{n} exp\{e_j^{(t)}\} \tag{6}$$

2.3. Network Architecture

We develop the video description architecture, as illustrated in Figure 2, based on the stacked LSTM encoder-decoder framework [3]. The stacked LSTM has 2 layers, each with 256-dimensional hidden unit, and they respectively perform encoder and decoder role for modeling the two variable-length sequences. The function of TELSA is to transfer learned temporal representations. It only operates when training the target domain dataset.

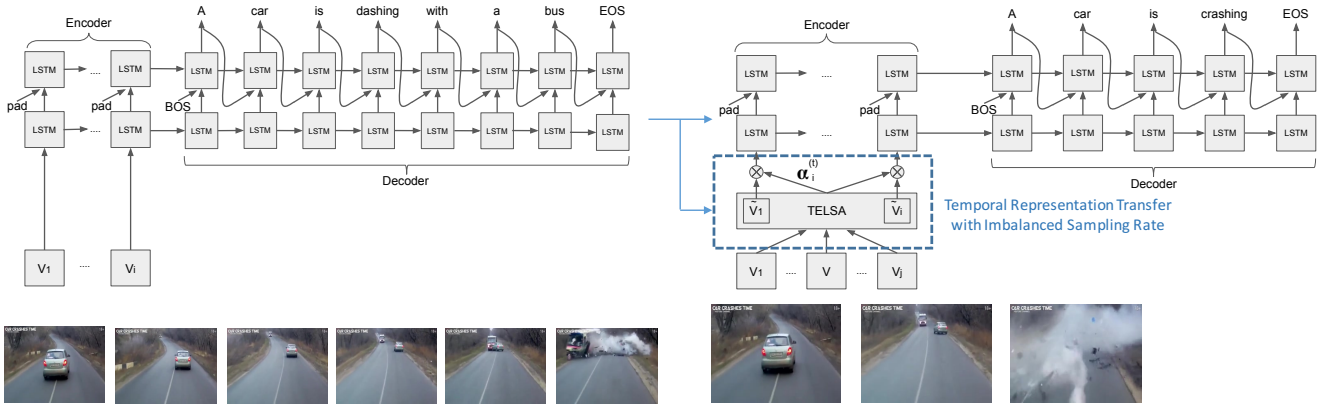

Fig. 2: Architecture for temporal representation transfer: The stacked LSTM encoder-decoder is pre-trained on source domain. To decrease the temporal discrepancy with imbalanced sampling rate (e.g. video frames for source domain is 80 and target domain is 40), TELSA mechanism can transfer temporal embeddings and adjust visual representations in encoding phase (TELSA mechanism in dash line rectangle is only activated when fine-tuning on target domain).

Table 1: Single domain evaluation results on MSR-VTT: We train on the training set and test on the validation set.

train:test samples	METEOR	BLEU @1	@2	@3	@4
80:80	26.10	75.50	60.30	46.70	34.80
40:80	25.40	75.40	58.80	44.20	32.20
40:80+TELSA	26.00	77.40	61.50	47.30	34.60

3. EXPERIMENTS

3.1. Datasets and Metrics

Our experiments were all conducted on two popular datasets, including MSR-Video to Text (MSR-VTT) [12] and Microsoft Video Description corpus (MSVD) [13] datasets. These datasets are all popular for video to language performance evaluation.

For quantitative analysis, we adopt two standard metrics, including METEOR [14] and BLEU@N [15] with N equal to 1, 2, 3, and 4. For qualitative analysis, we show samples of the video clips and the generated descriptions by all baselines and our method.

3.2. Single-domain Analysis: MSR-VTT

In the first set of experiments, we dealt with the imbalanced sampling rate problem within single domain, i.e., the trained source domain and the target doamin were both in MSR-VTT. For every video clip for training and every video clip for validation, we sampled them into 80 frames (we denote this arrangement as 80:80). This sampling is a baseline sampling since the training side and the testing side both sample video clips into the same number of frames. For comparison, we made 40:80 set to show how our TELSA approach can adapt to different sampling rates. In Table 1, row 1 and row 2 show the results obtained by applying 80:80 and 40:80 settings, respectively. It is obvious that the 80:80 setting outperformed the 40:80 setting, since the latter sampled less frames than the former in training the source domain. Intuition tells us sampling only 40 frames (one half of the original sampling rate) means under-sampled and the loss of partial video information for training networks. However, when we added the proposed TELSA mechanism to the 40:80 setting, the performance was significantly improved. In the BLEU@1, @2 and @3 experiments, the performance of the 40:80+TELSA setting was even better than that of the original 80:80 setting. As to the cases of METEOR and BLEU@4, the performances were also very close to that of the 80:80 setting.

3.3. Cross-domain Analysis: MSR-VTT to MSVD

In the second set of experiments, we handled the imbalanced sampling rate problem within cross-domain environment. We used the MSR-VTT and the MSVD datasets as the source domain and the target domain, respectively.

In the training process, the training sets of the two domains were used for training. For evaluation, we used a test set coming from the target domain. For the cross-domain experiments, we denote the number of sampled frames and from which domain these frames were sampled as N_1S:N_2T. For example, 40S:80T means 40 frames from source domain versus 80 frames from target domain.

To analyze how the trained source dataset, the trained target dataset, fine-tuning, and TELSA function during the video description process, we respectively checked their capability on video description. We designed three sets of experiments using different sampling rates to train the source dataset. Thus, the settings of the three experiments were 40S:80T (set A), 80S:80T (set B), and 120S:80T (set C), respectively.

Table 2 illustrates the results obtained by conducting experiments based on 40S:80T, 80S:80T, and 120S:80T set-

Test Video Clip

ground truth: a man scores a goal in soccer.

	A: 40S:80T	B: 80S:80T	C: 120S:80T
source data only:	a man is playing a video game.	a man is playing a football game.	a man is playing a football game.
fine-tuning:	a man is playing football.	a man is playing football.	a football player is running down the field.
fine-tuning + TELSA:	a soccer player is playing the goal.	a man is playing a soccer game.	a football player is playing football.

Fig. 3: Qualitative results on MSR-VTT-to-MSVD transfer.

Table 2: Evaluation results of different settings on MSR-VTT-to-MSVD transfer: All methods are tested on target domain testing data and each column represents one setting.

Method	single domain		transfer learning	
	source	target	fine-tune	ours
A: 40S:80T				
METEOR	26.80	26.70	28.14	**29.19**
BLEU@1	67.90	69.90	72.35	**74.49**
BLEU@2	50.30	54.10	57.25	**59.78**
BLEU@3	38.20	43.70	46.79	**49.26**
BLEU@4	27.10	33.20	36.81	**39.01**
B: 80S:80T				
METEOR	26.90	26.70	27.99	**28.55**
BLEU@1	69.10	69.90	72.38	**73.07**
BLEU@2	52.40	54.10	57.02	**57.49**
BLEU@3	40.60	43.70	46.45	**47.01**
BLEU@4	29.50	33.20	36.53	**36.67**
C: 120S:80T				
METEOR	26.00	26.70	27.30	**28.00**
BLEU@1	67.70	69.90	71.15	**72.08**
BLEU@2	50.60	54.10	56.10	**56.65**
BLEU@3	39.10	43.70	45.65	**46.13**
BLEU@4	27.60	33.20	35.30	**35.75**

tings, respectively. The digits shown in the left two columns of set A are results obtained by testing video description algorithm on the trained source dataset and target dataset, respectively. The right two columns are the video description results obtained by introducing transfer learning across domains. The third column are the results obtained by applying fine-tuning and the results obtained by applying our TELSA method are listed in the fourth column. The single domain results are consistently worse than those obtained by applying the transfer learning based methods. The video description results shown in set B and set C have consistent results.

To demonstrate the effectiveness of our approach qualitatively, we chose one video clip from the test set. Figure 3 shows one set of sampled image frames and the generated video description under the settings of 40S:80T, 80S:80T and 40S:80T. The ground truths are also listed at the bottom for comparison. In Figure 3, our method outperformed the baselines, including the use of source dataset training only and the introduction of fine-tuning. From the generated video description, our method (fine-tuning + TELSA) not only recognized objects in frames and actions across time, but also summarized the video contents in a more satisfactory way.

4. CONCLUSIONS

The contribution of this work is three fold. First, video-to-video transfer learning has been addressed by a number of past works [16, 17, 18]. However, video-to-language transfer learning has not been seriously addressed in the past. In this work, we have defined the problem and proposed an approach to alleviate the temporal domain discrepancy when encountering imbalanced sampling rates of different datasets coming from distinct application domains. Second, we propose TELSA to analyze and solve how varying sampling rates may impact the video-to-language transfer learning results. Third, our TELSA outperforms baseline fine-tuning, as evidenced by MSR-VTT to MSVD transfer learning experiments. Another application with TELSA could be extended to image quality assessment (IQA) for variable-size images, improving the flexibility upon the fixed-size manner [19].

5. REFERENCES

[1] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *ICLR*, 2015.

[2] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko, "Translating videos to natural language using deep recurrent neural networks," in *NAACL HLT*, 2015.

[3] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4534–4542.

[4] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4507–4515.

[5] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[6] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] Yuan Liu and Zhongchao Shi, "Boosting video description generation by explicitly translating from frame-level captions," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 631–634.

[9] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[10] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[12] Jun Xu, Tao Mei, Ting Yao, and Yong Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] David L. Chen and William B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR, June 2011.

[14] Michael Denkowski and Alon Lavie, "Meteor universal: Language specific translation evaluation for any target language," 2014.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," 2002.

[16] Jun Yang, Rong Yan, and Alexander G Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 188–197.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 4489–4497.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[19] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 206–219, 2018.