Keyframe Segmentation and Positional Encoding for Video-guided Machine Translation Challenge 2020

Tosho Hirasawa [*]
Tokyo Metropolitan University
hirasawa-tosho@ed.tmu.ac.jp

Zhishen Yang [*]
Tokyo Institute of Technology
zhishen.yang@nlp.c.titech.ac.jp

Mamoru Komachi
Tokyo Metropolitan University
komachi@tmu.ac.jp

Naoaki Okazaki
Tokyo Institute of Technology
okazaki@c.titech.ac.jp

Abstract

Video-guided machine translation as one of multimodal neural machine translation tasks targeting on generating high-quality text translation by tangibly engaging both video and text. In this work, we presented our video-guided machine translation system in approaching the Video-guided Machine Translation Challenge 2020. This system employs keyframe-based video feature extractions along with the video feature positional encoding. In the evaluation phase, our system scored 36.60 corpus-level BLEU-4 and achieved the 1st place on the Video-guided Machine Translation Challenge 2020.

1 Introduction

In multimodal machine translation (MMT), a target sentence is translated from a source sentence together with related nonlinguistic information such as images (Specia et al., 2016) and videos (Wang et al., 2019). The goal of Video-guided Machine Translation (VMT) Challenge 2020 is to generate target-language video descriptions given both the videos and its description in the source languages.

Videos preserve rich visual information that guides textual translation. Since video descriptions illustrate visual objects, scenes and actions in the videos, we hypothesize that obtaining appearance features (objects and scenes) and action features from videos will contribute to the quality of translation. Additionally, keyframes store entire images in the video, helping us extract high-quality video features.

A video consists of an ordered sequence of frames, while features extracted from them often do not preserve such order information. We hypothesize that incorporating such order information with visual feature facilities our model in improving translation quality.

Based on the above hypotheses, in the proposed video-guided machine translation system, we introduce a keyframe-based approach for video feature extraction, along with positional encoding to inject order information into video features. The core model in our system is a modified hierarchical attention model (Libovický and Helcl, 2017) with encoder and decoder architecture.

2 Hierarchical Attention with Positional Encoding

Our video-guided machine translation system is an extension of the hierarchical attention model (Libovický and Helcl, 2017). The underlying model has a simple encoder and a modified decoder from Bahdanau et al. (2015) that uses two individual attention mechanisms to compute the textual context vector and the auxiliary context vector (in our case, the context vector over sequential video representations). However, the model is assumed to incorporate with spatial image features (e.g., region of Interest feature from Faster-RCNN models) and cannot leverage order information, which is a distinguishing property of video features (e.g. I3D).

To address this problem, we add positional encodings (Vaswani et al., 2017) to the video representations at the beginning of the attention to make the model use the order of the representations.

Encoder We first encode the N-tokens input sentence $\mathbf{x} = (x_1, \cdots, x_N)$ into encoder states $\mathbf{h} = (h_1, \cdots, h_N)$ by a bidirectional GRU, where each h_* is a vector with d dimension. The T-elements video representations $\mathbf{z} = (z_1, \cdots, z_T)$ is extracted from a video $\boldsymbol{v}$ using either video or imagery encoder described in Section 3.

Additionally, we add positional encoding to video representations $\mathbf{z}$ to obtain position-aware video representations $\hat{\mathbf{z}} = (\hat{z}_1, \cdots, \hat{z}_T)$ at each

[*]Equal contribution

timestep $pos \in (1, \cdots, T)$:

$$\begin{aligned} \hat{z}_{pos} &= z_{pos} + PE_{pos} && (1)\\ PE_{(pos,2i)} &= sin(pos/10000^{2i/d}) && (2)\\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d}) && (3) \end{aligned}$$

where i is the dimension.

Decoder In each position j while decoding, we first compute the decoder state proposal $\boldsymbol{s}_j$ from previous word embedding w_{j-1} and previous decoder state $\hat{\boldsymbol{s}}_{j-1}$,

$$\boldsymbol{s}_j = \mathrm{GRU}(w_{j-1}, \hat{\boldsymbol{s}}_{j-1}) \quad (4)$$

Afterward, the textual context vector $\boldsymbol{c}_j^{(t)}$ and the video context vector $\boldsymbol{c}_j^{(\boldsymbol{z})}$ are computed using two separate attention mechanisms $\mathrm{att}_{\boldsymbol{t}}$ and $\mathrm{att}_{\boldsymbol{z}}$.

$$\begin{aligned} \boldsymbol{c}_j^{(\boldsymbol{t})} &= \mathrm{att}_{\boldsymbol{t}}(\boldsymbol{s}_j, \mathbf{h}) && (5)\\ \boldsymbol{c}_j^{(\boldsymbol{z})} &= \mathrm{att}_{\boldsymbol{z}}(\boldsymbol{s}_j, \hat{\mathbf{z}}) && (6) \end{aligned}$$

The final context vector $\boldsymbol{c}_j$ is computed using another attention over modalities $m \in \{\boldsymbol{t}, \boldsymbol{z}\}$.

$$\begin{aligned} \boldsymbol{e}_j^{(m)} &= o^T \tanh(\boldsymbol{W}_1 s_j + \boldsymbol{U}^{(m)} \boldsymbol{c}_j^{(m)}) && (7)\\ \alpha_j^{(m)} &= \frac{\exp(\boldsymbol{e}_j^{(m)})}{\sum\limits_{m' \in \{t,z\}} \boldsymbol{e}_j^{(m')}} && (8)\\ \boldsymbol{c}_j &= \sum_{m \in \{t,z\}} \alpha_j^{(m)} \boldsymbol{Q}^{(m)} \boldsymbol{c}_j^{(m)} && (9) \end{aligned}$$

where o^T and $\boldsymbol{W}_1$ are model parameters and shared among all modalities, while $\boldsymbol{U}^{(m)}$ and $\boldsymbol{Q}^{(m)}$ are dedicated model parameters for each modality. $\boldsymbol{U}^{(m)}$ and $\boldsymbol{Q}^{(m)}$ are a projection matrices that map each single-modality context vector into a common space. o is the weight vector with the same dimensions of the common space.

The final context vector $\boldsymbol{c}_j$ is fed into the second GRU along with the decoder state proposal $\boldsymbol{s}_j$ to generate the final decoder state $\hat{\boldsymbol{s}}_j$ and output distribution $p(y_j|y_{<j})$

$$\begin{aligned} \hat{\boldsymbol{s}}_j &= \mathrm{GRU}(\boldsymbol{c}_j, \boldsymbol{s}_j) && (10)\\ p(y_j|y_{<j}) &= \mathrm{softmax}(\boldsymbol{W}_2 \hat{\boldsymbol{s}}_j + b) && (11) \end{aligned}$$

where $\boldsymbol{W}_2$ and b are model parameters.

Multiple Video Feature Integration Integrating various types of video features into a video-guided machine translation system represents a potential way of improving translation quality (Wang et al., 2018b). In our system, we decided to ensemble models trained on different types of video features. In section 5, we detail choices on ensemble models.

3 Video and Imagery Encoders

Videos, as another input in our system, often possess visual clues that guide the translation, such as actions, objects and scenes. Encoding video to acquire information-rich video features acts as visual-guidance to text translation.

We classified two types of video features: action features derived from actions, and appearance features from visual objects and scenes. Keyframes in videos store whole images, which often provide good visual representations of objects and scenes. We used keyframes for appearance feature extraction and as a basis to segment videos for obtaining motion features.

Our video-guided machine translation system consists of a video encoder that outputs action features, and an imagery encoder generates appearance features.

Video Encoder We first segmented a video based on keyframes [2] to build a segment list, and each segment contains a keyframe and 31 consecutive frames after it. We then feed the video segment list to obtain the action feature matrix from a non-local neural network (Wang et al., 2018a) with Res-Net 101 (He et al., 2016) as the backbone pre-trained on ImageNet and fine-tuned on Kinetics400 dataset [3]. Each feature vector in the matrix is in chronological order of appearance of its video segment by the time. The action feature matrix $\mathbf{M} \in \mathbf{R}^{T \times d}$ for a video $\boldsymbol{v}$ is:

$$\begin{aligned} \mathbf{M} &= \text{Video Encoder}(S) && (12)\\ S &= \text{Segmentation}(\boldsymbol{v}) && (13) \end{aligned}$$

where S is the list of T keyframe video segments with chronological order.

Imagery Encoder Keyframes in the video store complete imagery information that suites our need to extracting high-quality appearance features.

[2] https://github.com/dmlc/decord

[3] https://gluon-cv.mxnet.io/model_zoo/action_recognition.html#id113

Model	Validation Set	Public Test Set
Wang et al. (2019)	-	29.12
(1) Text-only	35.10	-
Official I3D features [1]		
(2) with positional encoding	35.28	35.26
(3) without positional encoding	35.02	-
Keyframe-based video feature extraction		
(3) Action features	**35.42**	**35.35**
(4) Object features (Res-Net 152, ImageNet)	35.29	-
(5) Scene features (Res-Net 50, Place365)	35.14	-
Ensemble Model		
3 Action features (3)	36.20	-
3 Object features (4) + 3 Scene features (5)	36.38	-
3 Action features (3) + 3 Object features (4) + 3 Scene features (5)	**36.48**	**36.60**

Table 1: Corpus-level BLEU on validation and public_test sets.

Frames in the video often involve visual objects and visual scenes. Therefore we obtained these two types of appearance features from keyframes. We employed a object-recognition system that is a Res-Net 152 model pre-trained with ImageNet and a scene-recognition system that is a Res-Net 50 model pre-trained with Place365 (Zhou et al., 2018) [4]. The input to our imagery encoder is the keyframes from the video.

4 Experiment setup

Model The encoders of our model have one layer with 512 hidden dimensions, and therefore the bidirectional GRU has a dimension of 1024. The decoder state has a dimension of 512. The input word embedding size and output vector size are 1024.

During training, we used Adam optimizer with a learning rate of 0.001, clipping gradient norm to 1.0, the dropout rate of 0.5, batch size of 512, and early stopping patience of 10. The loss function was cross entropy. In the evaluation phase, we performed a beam search with a size of 5.

Preprocess We preprocessed both English and Chinese sentences in the same manner as in the starter code [5], where English sentences are lowercased, and Chinese sentences are split into sequences of characters.

The vocabulary of either English or Chinese contains tokens that occur at least five times in the training set, giving 7,947 types for English and 2,655 types for Chinese.

5 Experimental Result

The official evaluation metrics for VMT challenge 2020 is BLEU (Papineni et al., 2002). Table 1 shows the corpus-level BLEU-4 scores of each model on the validation set and public test set.

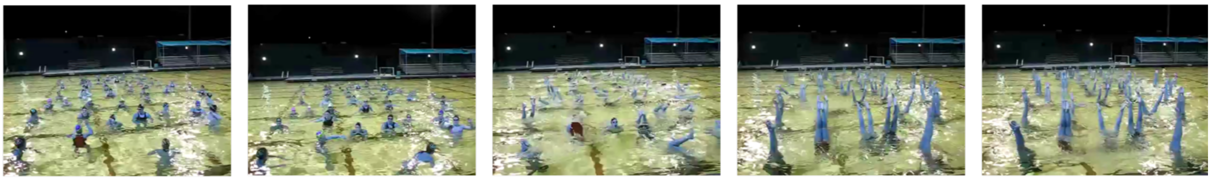
Our model using textual feature achieved a score of 35.10 on validation set, which makes it serve as a baseline (text-only baseline) for the following experiments with inputs of both textual and video features. Engaging official video features, we observed slight performance deterioration (-0.08 BLEU) without positional encoding, while a score improvement (+0.18 BLEU) with it. Based on above results, we decided to keep positional encoding in our system.

We extracted two types of video features based on keyframes: action features and appearance features, in which appearance features consist of two types of features: object features, and scene features. To evaluate how each type of video feature helps in improving the translation quality, we trained our system using each type of video feature with textual features.

On evaluation set, training our system using object features ((4) in the table 1) and textual features improves 0.19 BLEU scores over text-only baseline, and 0.01 BLEU compared to system trained on official I3D features ((2) in table 1). Training on scene features ((5) in table 1) and textual features give the system a slight 0.04 BLEU score

[4] http://places2.csail.mit.edu/

[5] https://github.com/eric-xw/Video-guided-Machine-Translation



English (Source): a swimming group practicing to do there under water dance trick .
Chinese (Target): 有一组游泳队正在练习在水下舞蹈的动作。

Ensemble model (Final submitted system): 一群游泳运动员正在练习在水下舞蹈技巧。
Trained on action features: 一群人正在一个游泳池里练习着游泳。
Text-only: 一个穿着黑色衣服的男人正在一个游泳池里练习游泳。

Figure 1: Example translations from the validation set generated from three of our system variants. Only our system with ensemble model (final submission) could correctly translate the group people as 游泳运动员 (swimming athletes), and their actions as 练习水下舞蹈技巧 (practicing underwater dance tricks).

improvements over text-only baseline. Compared to the appearance features, our system trained on action features and textual features achieved the highest BLEU scores, particularly, 0.32 BLEU over text-only baseline and 0.14 BLEU over our system trained on official I3D features. Based on above results, all types of video features improve our system performance compared to using textual feature only.

Compared to our system trained on single types of video feature and textual feature, our system ensembling models trained on different types of video features and textual features give another raise in the BLEU score. On evaluation set, compared to best preform single video feature model (3 in table 1), ensemble three models of (3) improves 0.78 BLEU score, while ensemble 3 models of (4) and 3 models of (5) get 0.96 BLEU score boost. An ensemble of three different models (3), (4), and (5) achieves the best BLEU score on validation set, this ensemble model is also our final submission, which obtains 36.60 BLEU score in the public test set an ranks the first place. Figure **??** shows example translations generated from three of our system variants.

6 Conclusion

In the Video-guided Machine Translation Challenge 2020, we revealed that keyframe-based video feature extraction and positional encoding jointly enhance the translation quality by showing a substantial improvement from the text-only baseline.

We also demonstrated that the ensemble of multiple models trained on different types of video features brought further performance improvements. In the future, we will explore the best integration of different features to improve translation quality under the video-guidance.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *ACL*, pages 196–202.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *MT*, pages 543–553.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018a. Non-local neural networks. In *CVPR*, pages 7794–7803.

Xin Wang, Yuan-Fang Wang, and William Yang Wang. 2018b. Watch, listen, and describe: Globally and locally aligned cross-modal attentions for video captioning. In *NAACL*, pages 795–801.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, pages 4581–4591.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464.