Supplementary Material for: Learning Modality Interaction for Temporal Sentence Localization and Event Captioning in Videos

Shaoxiang Chen[1⋆], Wenhao Jiang[2], Wei Liu[2], and Yu-Gang Jiang[1⋆⋆]

[1] Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University
[2] Tencent AI Lab
{sxchen13, ygj}@fudan.edu.cn, cswhjiang@gmail.com, wl2223@columbia.edu

1 Learning Latent Semantic Modality

Apart from the visual, motion, and audio modalities, which can be directly observed (apparent modalities), the latent semantics modality that carries high-level semantic information can be helpful for the language related tasks. We design a lightweight network to perform semantic attributes prediction using the sentence annotation provided by each dataset (either the video captioning or the sentence localization dataset). Note that this is a standalone task and the latent semantics modality is optional for our method.

The input of this network is the concatenation of all apparent modalities $\boldsymbol{X}^A = [\boldsymbol{X}^v||\boldsymbol{X}^m||\boldsymbol{X}^a]$, where $\boldsymbol{X}^A \in \mathbb{R}^{N\times(d_a+d_m+d_v)}$. We simply process $\boldsymbol{X}^A$ using bidirectional LSTMs and concatenate the hidden states of each LSTM:

$$\boldsymbol{X}^l = [\overrightarrow{\text{LSTM}}(\boldsymbol{X}^A)||\overleftarrow{\text{LSTM}}(\boldsymbol{X}^A)], \tag{1}$$

where $\boldsymbol{X}^l \in \mathbb{R}^{N\times 2d_{hid}}$, and $\overrightarrow{\text{LSTM}}(\cdot)$ and $\overleftarrow{\text{LSTM}}(\cdot)$ denote the LSTM networks that have d_{hid} units and process their input sequences in the forward and backward directions, respectively. $\boldsymbol{X}^l$ is then passed through a fully-connected layer with sigmoid activation to predict semantic attribute probabilities:

$$\boldsymbol{P} = \text{sigmoid}(\boldsymbol{X}^l\boldsymbol{W}_c + \boldsymbol{b}_c), \tag{2}$$

where $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are parameters, $\boldsymbol{P} \in \mathbb{R}^{C\times N}$ collects the temporal semantic attributes, and C is the vocabulary size of predefined attributes.

To train this network, we construct labels from the sentence annotations in event captioning or sentence localization datasets. We first process the training sentences of a dataset, select the most frequent C words that are noun or verb, and lemmatize them to form an attribute vocabulary. Then each sentence can

⋆ Part of the work is done when the author was an intern at Tencent AI Lab.
⋆⋆ Corresponding author.

be converted to a one-hot label $\boldsymbol{l} \in \mathbb{R}^C$ according to whether its words are in the vocabulary, where $\boldsymbol{l}_c = 1$ indicates that attribute c is present in the sentence, otherwise $\boldsymbol{l}_c = 0$. The label $\boldsymbol{l}$ is broadcast to the N temporal locations to compute the cross entropy loss at each location:

$$\mathcal{L}_{ce} = -\frac{1}{C}\sum_{c=1}^{C}(\boldsymbol{l}_c \ln \boldsymbol{P}_c + (1-\boldsymbol{l}_c)\ln(1-\boldsymbol{P}_c)), \tag{3}$$

where $\mathcal{L}_{ce} \in \mathbb{R}^N$. For the sentence localization task, the sentence annotations are usually available for temporal segments. To unify the loss representations, we construct a temporal mask $\boldsymbol{M}^{tcp} \in [0,1]^N$ defined as:

$$\boldsymbol{M}_i^{tcp} = \begin{cases} 1 & \text{if } i \in [s,e] \text{ and } rand(0,1) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $[s,e]$ is the temporal segment of the sentence annotation normalized to be in $[0,1]$. In event captioning it is safe to assume $s=0$ and $e=1$ since the videos are relatively short. Randomness is introduced in $\boldsymbol{M}^{tcp}$ to prevent overfitting. The final temporal semantic attributes prediction loss is computed as:

$$\mathcal{L}_{tcp} = \frac{1}{N}\mathcal{L}_{ce} \cdot \boldsymbol{M}^{tcp}, \tag{5}$$

where $\cdot$ is the dot product operator. From the above description, we can see that when the network learns to predict attributes, $\boldsymbol{X}^l$ carries rich information of latent semantics for every temporal location. Thus, it can be used to assist our target tasks through interacting with other modalities.

2 Feed-Forward Network (FFN)

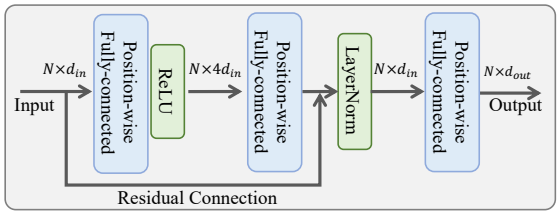

Fig. 1. The structure of the Feed-Forward Network (FNN).

As shown in Fig. 1, FFN is mainly composed of three position-wise fully-connected layers, each of which basically applies a fully-connected layer to each element of the input feature sequence with shared parameters. ReLU activation and layer normalization are applied to the first and second layers, respectively, and the initial input is connected to the second layer's output via a residual connection to encourage gradient flow. The output dimension d_{out} is decided according to the input ($d_{out} \leq d_{in}$).

3 More on Motivation

Our motivation is two-fold (Note that the Equations, Tables, and Figures mentioned in this section are all in the original paper):

(1) It is intuitive that both human and AI models understand events better via a combination of different sensory modalities, but the importances of different modalities vary among videos as well as among the moments inside a video. This motivates us to fuse the modality-interacted tensor by considering both the modality-wise and sequence-wise importances (Eq. (7)).

(2) Neuroscience researches have proven that information processing in one modality can affect another, which means that there are interactions among modalities and complementary information may communicate through such interactions. This motivates us to design sequence- and channel-level interactions for each pair of modalities. In the sequence-level interaction, each element from one sequence interacts with all the elements in the other sequence through the bilinear model (Eq. (2)). This fully-connected information flow between two modalities enables better utilization of complementary information than traditional fusion strategies as shown in Table 1. It is also widely accepted that different feature channels capture different information. Thus the goal of channel-level interaction is to emphasize important channels, which is realized by gating. The gate variable is computed via a channel-to-channel attention mechanism, and sequence-wise mean-pooling (Eq. (4)) is for reducing computation. The gating power is demonstrated by the experiment below in Section 6. The improvement brought by channel-level interaction is not as significant as sequence-level interaction, but it is indeed effective.

Based on the motivation, our goal is finding a better combination of modalities via fine-grained interaction. Attention is the building block we adopted to achieve this goal, because it is easy to understand and implement (also yields a clear framework). Finally, we have proven our modality interaction to be both effective and able to provide explainability (see Figs. 4 and 5).

4 Computational Complexity

Table 1. PMI-CAP's running times on one RTX 2080Ti GPU.

Mode	Memory	Time/batch
Train (batch size=32)	5939MB	0.38s
Infer (batch size=1)	1441MB	0.08s

The major computational cost is from sequence-level interaction, which mainly consists of feature projection and bilinear modeling (Eqs. (2) and (3) in the original paper). Assume that a pair of interacting feature sequences both have dimension $b \times n \times d$, where b and n stand for batch size and sequence length,

Table 2. Performances of PMI combined with other methods of target tasks.

Method	B@4	M	C
Masked Transformer [3]	47.49	32.43	77.35
Masked Transformer [3]+PMI	50.95	35.20	86.61
Method	IoU=0.3	IoU=0.5	IoU=0.7
ABLR [2]	53.55	37.47	16.21
ABLR [2]+PMI	55.26	39.52	16.88

Table 3. Performance comparison on the ActivityNet Captions dataset.

Method	B@4	M	C
vanilla-CAP (IRV2+I3D)	1.75	10.14	40.63
PMI-CAP (IRV2+I3D)	1.99	10.89	43.56
PMI-CAP (IRV2+I3D+A)	2.31	11.00	51.30
PMI-CAP/no-channel (IRV2+I3D)	2.00	10.52	43.06
2019 Rank-1 Intra-Event	3.91	11.96	49.56

respectively. Then the computational complexity is $O(bnd^2 + bn^2d)$. For short videos, since $n \ll d$, the complexity becomes $O(bnd^2)$ (mainly batch matrix multiplication) and is efficient to run on GPU. While for very long videos like TV shows the $O(bn^2d)$ term is dominant and the computational cost would grow quadratically with video length. Nonetheless, reducing the quadratic complexity for very long videos is out of scope of this work and is left for future work. Actual running times of PMI-CAP are shown in Table 1.

5 Compatibility with Other Models

We also test the effectiveness of our proposed PMI when combined with other types of architectures for event captioning or sentence localization. Note that the original methods used concatenated features [3] or single feature [2] as inputs, our implementations concatenate multimodal features for both methods. Results are presented in Table 2. For video captioning, we adopt the Masked-Transformer model [3] which is essentially different from RNN-based caption decoders. We use PMI to encode the multimodal features as its input and a substantial performance improvement over feature concatenation is obtained. We combine PMI with the state-of-the-art RNN-based sentence localization method ABLR [2] by inserting our PMI module between the feature extraction and Bi-LSTM feature encoding of ABLR, and a clear performance gain is also observed.

6 Captioning Performances on ActivityNet Captions

We further evaluate several variants of our PMI-CAP on the ActivityNet Captions and compare them with the 2019 ActivityNet captioning challenge winner [1], which used a more diverse set of features (e.g., objects and contexts) in

addition to the three common modalities. Following the official evaluation protocol, we compare the performances of captioning ground-truth event proposals on the validation set. The results are shown in Table 3. The vanilla-CAP method removes PMI and uses feature concatenation instead. The channel-level interaction and gating are disabled in the "no-channel" setting. As can be observed from the top four rows, our proposed method is consistently effective on ActivityNet. It is notable that our method can also achieve comparable performances with the challenge winner despite using fewer features.

References

1. Chen, S., Song, Y., Zhao, Y., Jin, Q., Zeng, Z., Liu, B., Fu, J., Hauptmann, A.: Activitynet 2019 task 3: Exploring contexts for dense captioning events in videos. arXiv preprint arXiv:1907.05092 (2019)
2. Yuan, Y., Mei, T., Zhu, W.: To find where you talk: Temporal sentence localization in video with attention based location regression. In: AAAI (2019)
3. Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C.: End-to-end dense video captioning with masked transformer. In: CVPR (2018)