Supplementary Materials: Towards Bridging Event Captioner and Sentence Localizer for Weakly Supervised Dense Event Captioning

Shaoxiang Chen and Yu-Gang Jiang*
Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University
{sxchen13, ygj}@fudan.edu.cn

Algorithm 1: Steps In One Training Iteration

Input: Video $\boldsymbol{V}$, Annotated Caption $\boldsymbol{C}$ (can be batched).
Inherit the Event Captioner $\boldsymbol{\mathcal{C}}$, Sentence Localizer $\boldsymbol{\mathcal{L}}$, and Induced Set Attention Block (ISAB) $\boldsymbol{\mathcal{I}}$ from previous iteration;
Load a predefined set of event proposals $\boldsymbol{G}$, $Q \leftarrow \emptyset$;
for $G_i \in \boldsymbol{G}$ **do**
  $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}(\boldsymbol{V}), G_i) \rightarrow (\boldsymbol{C}_i, \boldsymbol{S}_i)$; `// Output captions and features.`
  `// ISAB is abstracted as a feature transformer for clarity.`
  Q.append($(G_i, \boldsymbol{C}_i, \boldsymbol{S}_i)$);
Select the best $(\hat{G}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{S}})$ from Q based on caption confidences;
$\boldsymbol{\mathcal{L}}(\boldsymbol{V}, \hat{\boldsymbol{S}}) \rightarrow (\boldsymbol{W}^s, \boldsymbol{W}^v)$; `// Localize the sentence.`
`// Losses that connect` $\boldsymbol{\mathcal{C}}$ `and` $\boldsymbol{\mathcal{L}}$`.`
For $\boldsymbol{\mathcal{C}}$, compute captioning loss with $(\hat{\boldsymbol{C}}, \boldsymbol{C})$ guided by $\boldsymbol{W}^s$;
For $\boldsymbol{\mathcal{L}}$, compute localization loss between $\hat{G}$ and $\boldsymbol{W}^v$;
`// Losses within` $\boldsymbol{\mathcal{L}}$`.`
For $\boldsymbol{\mathcal{L}}$, compute contrastive loss and MIL loss within $\boldsymbol{\mathcal{L}}$;
Perform gradient-based parameter updates for $\{\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{I}}\}$;

Algorithm 2: The Inference Process for One Video

Input: Video $\boldsymbol{V}$.
Output: Segment predictions with captions.
Load trained parameters for $\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{I}}$;
Initialize the proposals $\boldsymbol{G}$, $Q \leftarrow \emptyset$;
for *iter* $\leftarrow 1$ **to** *max_iter* **do** `// max_iter can be 1.`
  for $G_i \in \boldsymbol{G}$ **do**
    $\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{I}}(\boldsymbol{V}), G_i) \rightarrow (\boldsymbol{C}_i, \boldsymbol{S}_i)$; `// Proposal to sentence.`
    Q.append($(G_i, \boldsymbol{C}_i, \boldsymbol{S}_i)$);
  Select the best $(\hat{G}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{S}})$ from Q; `// Similar to training.`
  $\boldsymbol{\mathcal{L}}(\boldsymbol{V}, \hat{\boldsymbol{S}}) \rightarrow (\boldsymbol{W}^s, \boldsymbol{W}^v)$; `// Localize the sentence.`
  $\boldsymbol{G} \leftarrow \text{TAG}(\boldsymbol{W}^v)$; `// Generate new proposals with TAG, and refine` $\boldsymbol{G}$ `for the next iteration.`
return top K of $\boldsymbol{G}$, captioned and sorted by caption confidences.

1. Algorithm Description

We show the simplified sequence of steps during training and inference in Algorithms 1 and 2.

2. More Model Details

Masked Temporal Attention. The masked temporal attention (adapted from [1]) in our event captioner (Eq. (16)) is formulated as

$$\boldsymbol{h}_t^{(1)} = \texttt{LSTM}^{(1)}(\texttt{att}(\widetilde{\boldsymbol{V}}, M(:, G_i), \boldsymbol{h}_{t-1}^{(1)}), \boldsymbol{h}_{t-1}^{(1)}),$$
$$\boldsymbol{h}_t^{(2)} = \texttt{LSTM}^{(2)}([\texttt{embed}(w_{t-1}), \boldsymbol{h}_t^{(1)}], \boldsymbol{h}_{t-1}^{(2)}),$$
$$\texttt{att}(\widetilde{\boldsymbol{V}}, M(:, G_i), \boldsymbol{h}_{t-1}^{(1)}) = \sum_{l=1}^{L} \beta_l \widetilde{\boldsymbol{V}}_l,$$
$$\text{where} \quad \beta_l = \frac{M(l, G_i)\exp(r_l)}{\sum_l M(l, G_i)\exp(r_l)},$$
$$r_l = W_r(\tanh(W_{r_V}\widetilde{\boldsymbol{V}}_l + W_{r_h}\boldsymbol{h}_{t-1}^{(1)} + b_r)),$$

where W_r, W_{r_V}, W_{r_h} and b_r are trainable parameters. Masked Temporal Attention is mainly for temporally aggregating video features within the segment indicated by the mask. The value of $M(l, G_i)$ is 1 if $s_i \leq l \leq e_i$, and otherwise is 0.

More Implementation Details. The video and sentence features are linearly transformed with a ReLU activation in

*Corresponding author.

	Value	M	C	R	B@4
Concept number	1024	7.37	19.86	12.82	1.25
	2186	7.49	21.21	13.02	1.33
	All (5305)	7.45	20.73	12.96	1.29
Concept type	Verb (615)	7.18	19.61	12.79	1.23
	Noun (1571)	7.34	19.92	12.85	1.26
	Verb+Noun (2186)	7.49	21.21	13.02	1.33

Table 1: Study on the number and type of concepts.

Eq. (6). The number of heads in the $\mathrm{MAB}(\cdot)$ function is generally set to 4. The $\mathtt{FFN}(\cdot)$ has two fully connected layers with ReLU activation and a residual connection from its inputs to the outputs. The γ in Eq. (18) is set to 0.1. A dropout layer with ratio 0.5 is applied to each of the LSTM outputs (Eq. (16)). As in most video captioning methods, we generate captions using beam search and the beam size is set to 5.

3. More Ablation Experiments

Table 1 above shows the effects of the number and type of concepts. We can conclude that (1) as the number increases, there is a point (2186, our final choice) where we can get more information from more concepts to enhance the features without having too much noise and making the concepts too long-tailed; (2) it is better to include verbs and nouns since they are both important aspects of an event. But noun concepts are the majority and yield better performances than verb concepts.

4. Visualization of the Learned Space

To show how our method models the video-sentence interaction and learn their joint space, we visualize the learned features with t-SNE[1] in Figure 1. We can see that the video and sentence subspace are initially separated, and the localization loss we optimize will pull them close and make them overlap as the training proceeds. More specifically, we plot the features of multiple clips from a video, and we can clearly observe that the sentence and video features are gradually pulled close. Note that although we did not explicitly enforce the similar videos' (or sentences') features to be close to each other, the features from the same video's clips and sentences appear to be clustered, which may be because the underlying concepts of the video are captured and can make it easier to cluster similar clips.

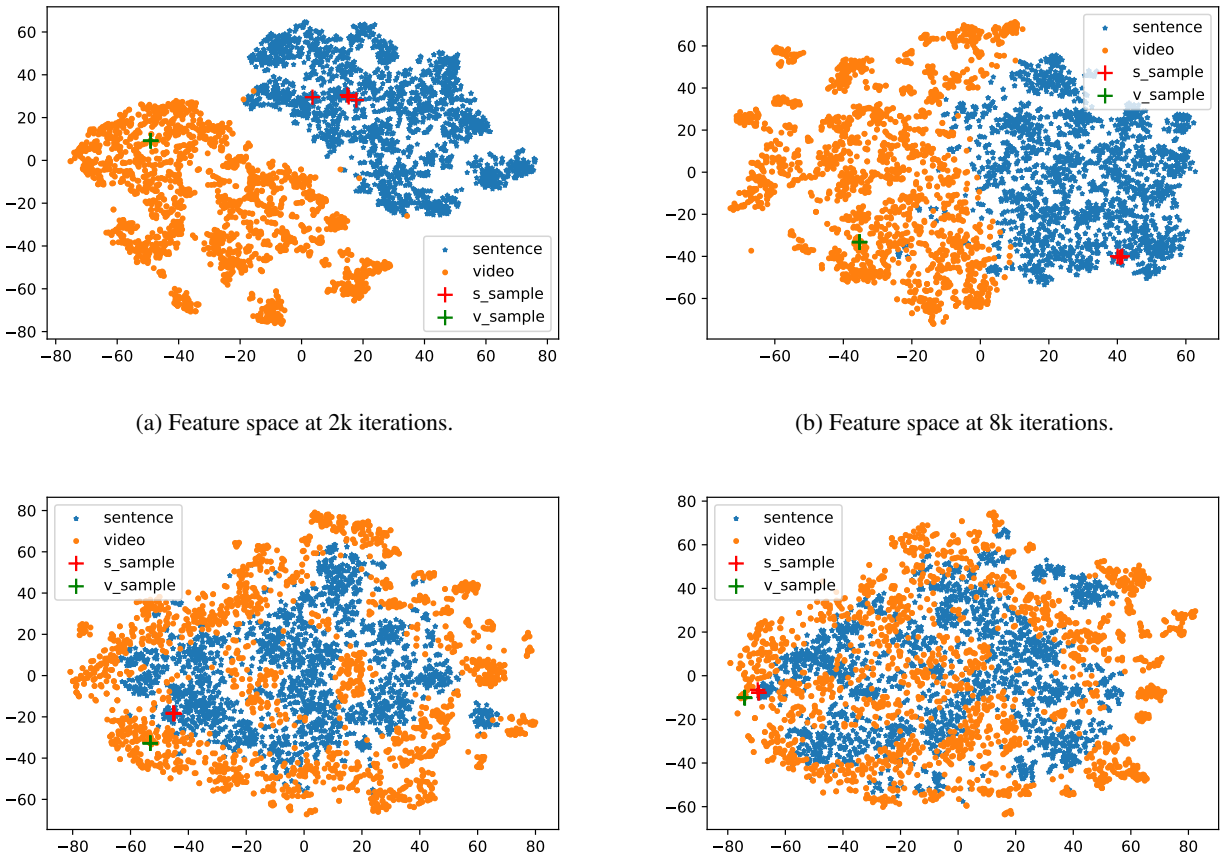
5. More Qualitative Results

In Figures 2 and 3, we provide more qualitative results generated by our method on the testing set. Each figure contains three examples. Figure 2 shows high-quality examples, where our method can correctly localize and describe the key events, e.g. 'a game of cricket' and 'volleyball'. From the examples in Figure 3, we can see that one weakness of the method is that it can generate repetitive captions for multiple segments. This is due to that inter-segment relation is not modeled and the captioning process of each segment is independent.

[1] We also note that t-SNE is sensitive to its hyperparameter settings.

References

[1] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015. 1



(a) Feature space at 2k iterations.

(b) Feature space at 8k iterations.

(c) Feature space at 14k iterations.

(d) Feature space at 20k iterations.

Figure 1: t-SNE visualization of the joint feature space (around 2,000 sentence-clip pairs shown here) learned at different training stages. The '+' markers denote samples from a video `bjtjeUcoxkg.mp4`.

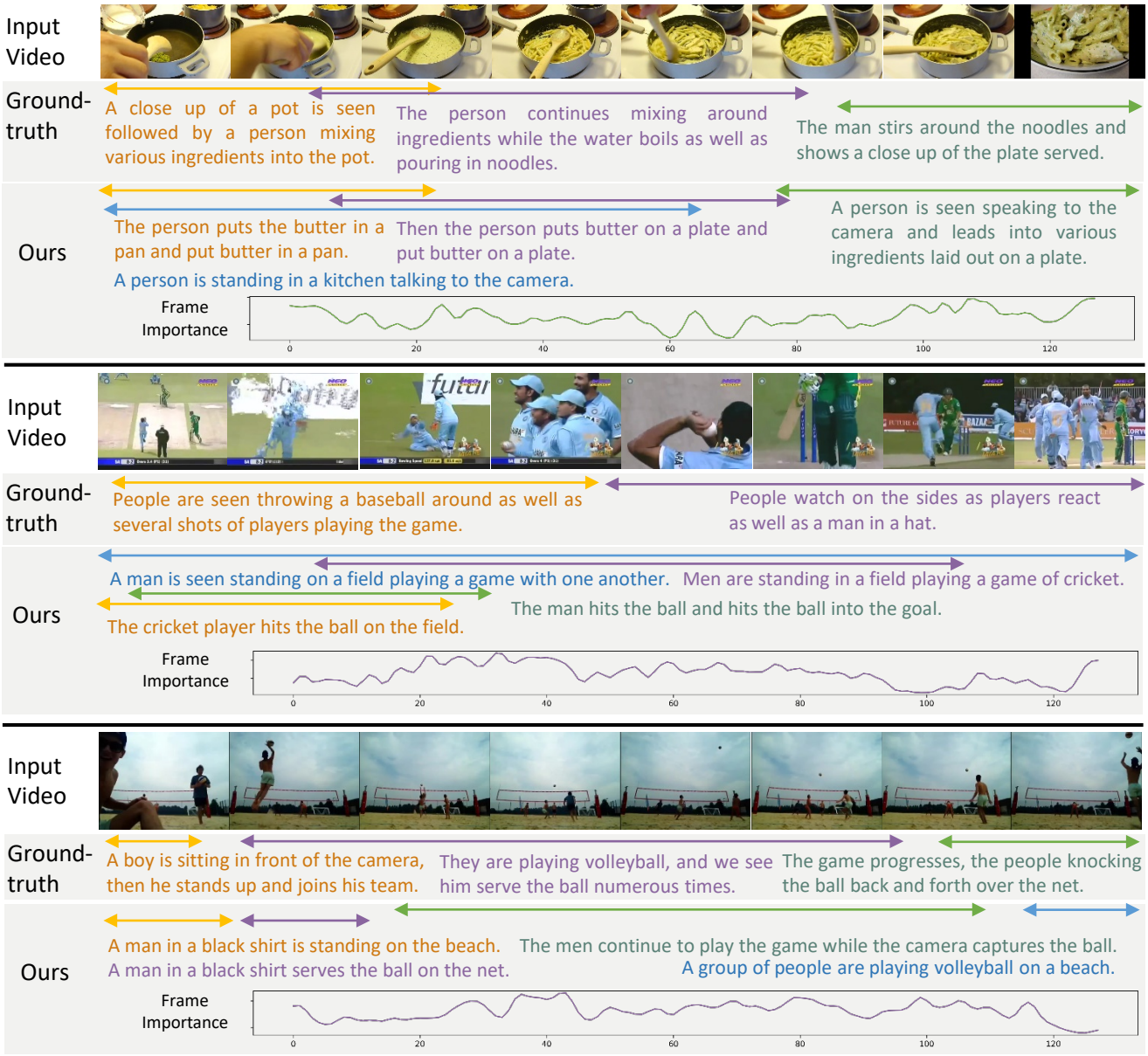


Figure 2: More qualitative examples.

Figure 3: More qualitative examples.