Less is More: CLIPBERT for Video-and-Language Learning via Sparse Sampling – Supplementary File

Jie Lei*, Linjie Li*, Luowei Zhou2, Zhe Gan2, Tamara L. Berg1, Mohit Bansal1, Jingjing Liu2
1UNC Chapel Hill 2Microsoft Dynamics 365 AI Research
{jielei, tberg, mbansal}@cs.unc.edu
{lindesy.li, luowei.zhou, zhe.gan, jingjl}@microsoft.com

1. Additional Experiments

Visual Question Answering. As CLIPBERT is designed based on 2D CNN, and is pre-trained on image-text corpus, it is also directly applicable to image-text downstream tasks, such as image-based question answering. We show CLIPBERT’s performance on VQA 2.0 dataset [3] in Table 1. The model is finetuned from the image-text pre-trained weights on 8GPUs for 13 epochs, with batch size 32 and learning rate 5e-5. CLIPBERT shows a reasonable performance compared to the strong pre-training baselines. Note that CLIPBERT uses grid features [6, 4] instead of the commonly used region features, which is much more computation efficient, e.g., extracting grid features is around 80× faster than extracting region features according to the computation time reported in [6].

2. Downstream Task Adaptation

Our CLIPBERT is quite generic, once trained, it can be easily adopted and transferred for various downstream tasks. In particular, in this work, we focus on text-to-video retrieval and video question answering.

Text-to-video Retrieval. We use a two-layer MLP with the last layer [CLS] token hidden state for a two way (i.e., matched or not matched) classification for retrieval. We use LogSumExp loss for training. Denote the two-way classification logit output for clip $\tau_i$ from the video associated with the $j$-th example as $g^{(j)}_{\tau_i} \in \mathbb{R}^2$, where $i = 1, \ldots, N_{train}$ for training ($i = 1, \ldots, N_{test}$ for inference; see Section 3 of the main paper). The LogSumExp prediction $p^{(j)} \in \mathbb{R}^2$ is defined as:

$$p^{(j)} = \frac{\sum_{i=1}^{N_{train}} g^{(j)}_{\tau_i}}{\text{sum}(\sum_{i=1}^{N_{train}} e^{g^{(j)}_{\tau_i}})}.$$ (1)

We then use a negative log likelihood loss for training:

$$L = -\frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \log p^{(j)}(y_j),$$ (2)

where $\mathcal{D}$ is the dataset, $y_j$ is the index of the ground-truth answer for the $j$-th example.

We conduct experiments on three popular text-to-video retrieval datasets, MSRVTT [13], DiDeMo [2], and ActivityNet Captions [7]. Table 2 shows the training details for models on each of the datasets.

Video Question Answering. Similar to text-to-video retrieval task, we take the last layer [CLS] token hidden state through a two-layer MLP for classification. We use LogSumExp to aggregate prediction from multiple clips to calculate loss. The formulation of LogSumExp loss is similar to Equation 1 except that the dimension of $g_{\tau}$ equals to the number of answer candidates.

We conduct experiments on three video QA datasets, TGIF-QA [5], MSRVTT-QA [12], and MSRVTT MC Test [14]. For TGIF-QA, we evaluate three sub-tasks,
Table 3: Training details for video question answering tasks. Bsz is short for batch size. Grad-Accu stands for gradient accumulation steps. LR means initial learning rate.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Epochs</th>
<th>Bsz×Grad-Accu</th>
<th>#GPUs</th>
<th>LR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGIF-QA Action</td>
<td>35</td>
<td>32×1×8</td>
<td>1</td>
<td>1e-4</td>
</tr>
<tr>
<td>TGIF-QA Transition</td>
<td>15</td>
<td>32×1×8</td>
<td>1</td>
<td>1e-4</td>
</tr>
<tr>
<td>TGIF-QA FrameQA</td>
<td>15</td>
<td>32×1×8</td>
<td>1</td>
<td>1e-4</td>
</tr>
<tr>
<td>MSRVTT-QA</td>
<td>10</td>
<td>16×1×4</td>
<td>4</td>
<td>5e-5</td>
</tr>
</tbody>
</table>

Table 3: Training details for video question answering tasks. Bsz is short for batch size. Grad-Accu stands for gradient accumulation steps. LR means initial learning rate.

i.e., Action, Transition, and FrameQA. We train a separate model for each of the evaluated TGIF-QA tasks. For MSRVTT MC Test, as it uses the same training set as the MSRVTT retrieval task, we directly use the trained retrieval model to rank the five candidate answers. Table 2 shows the training details for models on TGIF-QA tasks and MSRVTT-QA.

References