Leveraging Generative Language Models for Weakly Supervised Sentence Component Analysis in Video-Language Joint Learning

Supplementary Material

7. Attention Over Similarity Matrix of Text Data For Video-Text Retrieval

We compute the similarity score by incorporating the fine-grained representations of input texts to enhance the models' performance for video-text retrieval. It is a modified form of the one presented in the baseline [32]. Considering the word-level representation for the texts t, p, and n, having dimension $\mathbb{R}^{tok_i \times dim}$, where tok_i denotes the number of words/tokens in any text i; $\{t, p\}$ forms a correct pair and $\{t, n\}$ forms an incorrect pair. For any text pair $\{a, b\}$, either correct or incorrect, a fine-grained similarity matrix, $\mathbf{S}'_{a-b} \in \mathbb{R}^{tok_a \times tok_b}$, can be formulated as:

$$\mathbf{S}'_{a-b} = \mathbf{a}(\mathbf{b})^T. \tag{8}$$

The resulting $\mathbf{S}'_{a-b}$ is a similarity matrix containing similarity scores of every token of a with every token of b. Simply averaging these scores to obtain the final similarity score would be ineffective for learning the relative importance of words and the alignment between them. Hence, we incorporate an attention-based similarity calculation by generating instance-level scores: $\mathbf{S}'_a$ and $\mathbf{S}'_b$, for a and b respectively.

Two stages of attention are applied to generate the instance-level similarity score, $\mathbf{S}'_a$. From the first attention operation, we obtain $\mathbf{S}'_a \in \mathbb{R}^{tok_b}$ by performing a weighted averaging of the similarity scores between each word in the word-level representation of b with all of the words of the sentence embedding a at a time, formulated as:

$$\mathbf{S}'_{\mathbf{a}} = \sum_{i=1}^{tok_a} \frac{\mathbf{e}^{\mathbf{S}'_{a-b(*,j)}/\tau}}{\sum_{j=1}^{tok_a} \mathbf{e}^{\mathbf{S}'_{\mathbf{a}-\mathbf{b}(j,*)}/\tau}} \mathbf{S}'_{a-b(i,*)}. \tag{9}$$

The second attention operation performs weighted averaging on the output from the first attention operation, $\mathbf{S}'_a$ to obtain the instance-level score for embedding $\mathbf{a}$ ($\mathbf{S}'_a \in \mathbb{R}$).

$$\mathbf{S}'_a = \sum_{i=1}^{tok_a} \frac{e^{\mathbf{S}'_{a(1,i)}/\tau}}{\sum_{j=1}^{tok_a} e^{\mathbf{S}'_{a(1,j)}\tau}} S_{a(1,i)}. \tag{10}$$

Similarly, utilizing Equation 9 and 10 we generate the instance-level similarity, $\mathbf{S}'_b$, for text $\mathbf{b}$.

After averaging these two instance-level similarity scores, $\mathbf{S}'_a$ and $\mathbf{S}'_b$, we get the final similarity score, $\mathbf{S}_{a-b}$, of the sentence pair $\{a, b\}$,

$$\mathbf{S}_{a-b} = (\mathbf{S}'_a + \mathbf{S}'_b)/2. \tag{11}$$

The $\texttt{sim}(\cdot, \cdot)$ function in Equation 2 returns this fine-grained similarity score, $\mathbf{S}_{a-b}$, for video-text retrieval tasks.

Additionally, we use an auxiliary loss similar to the self-supervised cross-entropy loss in traditional video-text retrieval tasks. For this, we replace the original input texts to the model with the positive texts generated by the LLM with similar semantics. This loss further contributes to diversifying the video-language joint embedding space.

8. Evaluation Protocols

For moment retrieval, we utilize already established evaluation metrics. These are Recall@1 with 0.5 and 0.7 IoU thresholds, mean average precision (mAP) with 0.5 and 0.75 IoU thresholds along with the average mAP over a series of IoU thresholds (from 0.5 to 0.95 with an increment of 0.05). On the other hand, for video-text retrieval we use the widely used retrieval metrics Recall at Rank K (R@K, higher is better), and Mean Recall (MnR, lower is better) for both video-to-text (V2T) and text-to-video (T2V) retrieval.

9. LLM Prompting

We use one-shot learning approach with the LLM to generate our additional text samples. Prompts used for generating a negative sample, *e.g.* object-changed negative, and a positive sample are shown as Algorithm 1 and Algorithm 2 respectively. We provide the instruction as the **system** role of the LLM. Besides, we also pass an example of the operation in subsequent **user** and **assistant** roles to aid the LLM in understanding the task in the one-shot learning approach. Subsequently, the **user** queries the LLM with the anchor text as input, and the **assistant** outputs the sentence as required.

Algorithm 1 Generation of Object-Changed Negative

1: **system** : Change the object of the sentence
2: **user** : A woman goes for a drive in a Greek island.
3: **assistant** : A woman goes for a drive in Sahara desert.
4: **user** : $\{input\}$
5: **assistant** : $\{output\}$ // output negative for $\{input\}$

Algorithm 2 Generation of Positive Sample

1: **system** : Alter voice of the sentence
2: **user** : The chef cooks a meal.
3: **assistant** : A meal is being cooked by the chef.
4: **user** : $\{input\}$
5: **assistant** : $\{output\}$ // output positive for $\{input\}$