Phrase-level Prediction for Video Temporal Localization

Sizhe Li
Wangxuan Institute of Computer Technology,
Peking University
Beijing, China
lisizhe@pku.edu.cn

Chang Li
Wangxuan Institute of Computer Technology,
Peking University
Beijing, China
1900012977@pku.edu.cn

Minghang Zheng
Wangxuan Institute of Computer Technology,
Peking University
Beijing, China
minghang@pku.edu.cn

Yang Liu*
Wangxuan Institute of Computer Technology,
Peking University
Beijing Institute for General Artificial Intelligence
Beijing, China
yangliu@pku.edu.cn

ABSTRACT

Video temporal localization aims to locate a period that semantically matches a natural language query in a given untrimmed video. We empirically observe that although existing approaches gain steady progress on sentence localization, the performance of phrase localization is far from satisfactory. In principle, the phrase should be easier to localize as fewer combinations of visual concepts need to be considered; such incapability indicates that the existing models only capture the sentence annotation bias in the benchmark but lack sufficient understanding of the intrinsic relationship between simple visual and language concepts, thus the model generalization and interpretability is questioned. This paper proposes a unified framework that can deal with both sentence and phrase-level localization, namely Phrase Level Prediction Net (PLPNet). Specifically, based on the hypothesis that similar phrases tend to focus on similar video cues, while dissimilar ones should not, we build a contrastive mechanism to restrain phrase-level localization without fine-grained phrase boundary annotation required in training. Moreover, considering the sentence's flexibility and wide discrepancy among phrases, we propose a clustering-based batch sampler to ensure that contrastive learning can be conducted efficiently. Extensive experiments demonstrate that our method surpasses state-of-the-art methods of phrase-level temporal localization while maintaining high performance in sentence localization and boosting the model's interpretability and generalization capability. Our code is available at https://github.com/sizhelee/PLPNet.

CCS CONCEPTS

• **Computing methodologies** → **Visual content-based indexing and retrieval**; *Activity recognition and understanding*.

*Corresponding author

Conference ICMR '22, June 27–30, 2022, Newark, NJ

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/3512527.3531382

KEYWORDS

Video temporal localization, Contrastive Learning, Natural language query, Phrase Localisation, Sentence Localisation

ACM Reference Format:
Sizhe Li, Chang Li, Minghang Zheng, and Yang Liu. 2022. Phrase-level Prediction for Video Temporal Localization. In *Proceedings of ACM International Conference on Multimedia Retrieval (Conference ICMR '22).* ACM, New York, NY, USA, 9 pages. https://doi.org/3512527.3531382

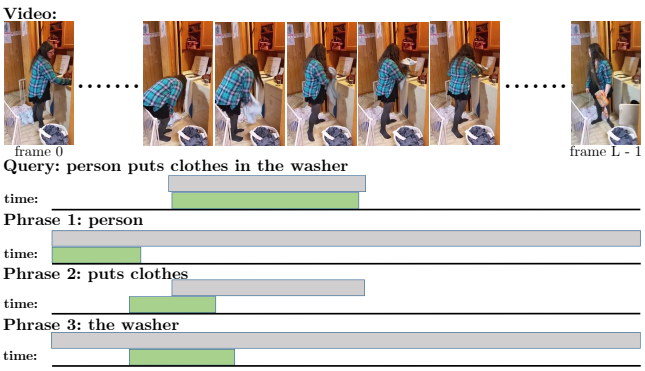

Figure 1: An illustrative example in Charades-STA dataset on video temporal localization task, where the gray and green blocks represent the ground truth and model prediction. Although existing approaches gain a reasonable performance on sentence localization, the performance of phrase localisation is far from satisfactory.

1 INTRODUCTION

As humanity continues to produce videos at an ever-increasing scale, understanding the contents of the videos has drawn growing research interests. The goal of temporal localization is to locate temporally video moments-of-interest (segment) of a specific activity described by a natural language query in an untrimmed long video. Automatic temporal grounding enables us to find the moment of interest from the whole video effectively and efficiently, which has

broad application potential in video surveillance [3, 4, 21], visual question answering [1, 2, 22], etc.

In recent decades, fully supervised methods [5, 8, 17, 18, 20, 23, 24, 28, 31–33, 35, 37] have achieved remarkable prediction accuracy in sentence-level prediction settings. However, we notice that these models, trained with sentence-level annotation, have unsatisfying performance in phrase-level prediction settings. As illustrated in fig. 1, the overall sentence-level prediction is satisfying while phrase-level predictions of, for example, 'person' and 'washer' (shown in grey blocks) are far from accurate. We empirically observe that in the Charades-STA[8] dataset, when we replace the original sentence-level query with the verb-object phrase-level queries (e.g. open door), the IoU@0.5 score of [17] is dropped from 59.17% to 32.08% (as shown in Tab.1).

The phrase in our work is defined as one or several consecutive words in the original sentence query and is a more fundamental component in a semantic context than the whole sentence. In principle, phrase-level queries are easier to deal with than sentence-level queries since less semantic information and fewer concepts combination scenarios need to be considered. The inability to perform phrase-level prediction tasks indicates that (1)Although existing temporal localization models achieve decent results, they may not grasp the intrinsic relationship between visual and semantic information, but overfit the dataset annotation biases instead. (2) The generalization ability of existing models is questionable, as without explicit understanding of more straightforward phrase-level concepts, localizing their combinations is not trivial. Empirically, when testing on a new test split of Charades-STA with a different combination of known phrases, IoU=0.5 accuracy score drops from 59.17% to 56.07% as shown in Tab. 3). The understanding and revelation of the cross-modal correlation of a single simple concept is the essence of solving the challenging cross-modal matching problem due to the corresponding relationship between various visual entities and textual words. (3) Existing models lack interpretability and reliability, raising practical problems when applied in real-world scenarios that are far more complicated than ideal dataset settings.

Motivated by the above observation, we make the first attempt to take phrase localization into account. Note that the simplest solution is to collect temporal boundaries for all phrases and retrain the model, but it needs laborious manual annotation, thus limiting scalability and practicability in real-world scenarios. In this paper, we propose a method considering phrase localization without temporal annotation required, namely Phrase-level Prediction Net (PLPNet). Specifically, we hypothesize that similar phrases tend to focus on similar video cues, while dissimilar ones not. Instead of directly regressing phrase predictions' boundary timestamps, we rely on this hypothesis and build a contrastive mechanism to restrain phrase localization. Moreover, considering the flexibility of the sentence and wide discrepancy among phrases, sampling randomly when forming batches may lead to missing similar phrases and not provide enough supervision signal. We propose a batch sampling mechanism using sentence clustering to ensure contrastive learning to be conducted efficiently in a batch-wise manner.

Our main contributions are in three folds:

1. We are the first to study the phrase-level localization problem and propose the Phrase-Level Prediction Network, which can be trained end-to-end without phrase-level temporal annotation.

2. By taking advantage of the inherent sentence relationships (via clustering), we propose a new sampling mechanism for benefiting contrastive learning, which assumes similar phrases tend to focus on similar video cues, while the dissimilar ones should not.

3. We perform experiments on two widely used datasets Charades-STA [8] and ActivityNet Captions [13]. Our experiments prove that our method improves the model's phrase localization accuracy and generalization capability.

2 RELATED WORK

The task, temporal localization, proposed by TALL[8], has drawn interest of various researchers. There are now two different tracks for this task, fully-supervised and weakly-supervised. In fully-supervised scenario, model can obtain the accurate timestamps for sentence, while in weakly-supervised scenario, only the video and corresponding sentence are available.

2.1 Fully-supervised Temporal Localization

Recently, fully-supervised works tend to consider fine-grained information. Some methods focus on video details, like Dori [20] and HTVG [5], which consider the features of objects in the video to improve the model's performance. MSA[32] produces stage-aggregated features for prediction, to identify different stages of the required time section. It makes an unprecedented move when training the model to identify different stages of the wanted period. 2D-TAN[33] considers frames' relation with each other with an adjacent temporal network to obtain better segment visual features. Following it, models like DPIN[23] and SMIN[24] dive into exploring generating both frame and segment features for interaction and attention, considering both local and global video features.

Other works notice the effect of sentence. DeNet[37] uses Gaussian distribution to deal with the uncertainty of part of the query sentence, built on the hypothesis that verbs and nouns are relatively specific and others may have variances. It proposes a solution to a hardly-noticed but crucial problem in the temporal localization field: annotation bias. In LGI[17], the features of subqueries generated via attention pooling are fused with video features. It is a novel designation for LGI to interact subquery feature with video feature to generate fused feature for prediction. However, the guidance of the generation of subquery-level information is not fully guaranteed. MMN [26] trains the model to distinguish matched and unmatched video-sentence pairs collected from intra-video and inter-video in order to find the relationship between positive and negative sentences. It also ignores the relationship between phrases and video.

Although supervised models perform well in this task, experiment results show that their performances in phrase-level predictions are unsatisfying. Different from single words that can apply to various scenarios, yet different from a whole sentence's competency, phrases are components that have meanings individually, presenting linguistic logic. As a result, the unsatisfying performances indicate that these models may not truly capture the correlation between the visual and textual information of a simple concept, limiting their generalization performance and interpretability. To

solve the above problems, our PLPNet generate phrases from sentence and use the temporal and semantic relationship between the video and text to gain fine-tuned fused features.

2.2 Weakly Supervised Temporal Localization

For Weakly-Supervised Video Localization [7, 25], due to the lack of accurate timestamp annotations, researchers try to promote models in learning the cross-modal correlation without the supervision of time stamps. To overcome this difficulty, many weakly supervised models use contrastive learning to use the similarity information between sentences and make full use of the combination information between video and text. For example, TGA [16] train the model to distinguish matched video-query pairs and unmatched ones collected from other training samples; CPL [36] utilize Gaussian distribution to generate positive and negative video proposals for contrastive learning. Under these circumstances, the model does not have access to accurate timestamps, but only takes the whole video as positive sample. They still consider the sentence query as a fundamental semantic element without exploiting phrases that represent information in a more fine-grained level. Inspired by the ideas used in weakly supervised scenario, we also use contrastive learning in our method. What's more, we compare of similarity between phrases and corresponding video clips and understand the sentence in phrase-level, to improve the interpretability of our mudel.

3 METHOD

3.1 Overview

Given an untrimmed video $v = \{v_1, v_2, ..., v_T\}$ composed of T frames and natural language query $Q = \{w_1, w_2, ..., w_m\}$ of m words, the objective of temporal localization is to identify the temporal boundary of a target moment $(\hat{t}_s, \hat{t}_e)$ in v, so that the video segments $\{v_t\}_{t=\hat{t}_s}^{t=\hat{t}_e}$ matches the query.

Figure 2 illustrates the architecture of our PLPNet model. Given a video and sentence query, the video encoder extracts position-aware video feature $\mathbf{V}$ while the text encoder extracts word features and uses a bi-directional LSTM to generate sentence feature q. Then the phrase generation part parses the sentence into phrases and provides phrase features $\tilde{\mathbf{q}}$. In the interaction module, video feature meets each phrase to get fused features and global fused feature , which are then used to gain frame attention weight. Finally in prediction module, boundaries for both sentence and phrases are generated. Moreover, based on the hypothesize that similar phrases tend to focus on similar video cues, while the dissimilar ones should not, we propose feature loss $\mathcal{L}_{feature}$ to boost the phrase-level performance without phrase annotation. More details about the model are shown in section 3.2. To make it more efficient and reliable, we propose a new sampling method to form batches and ensure that each batch contains both similar and dissimilar phrases, as shown in section 3.3. The overall training and inference paradigm is discussed in section 3.4.

3.2 Model Architecture

3.2.1 Video Encoder: We first extract the high-quality visual features $\mathbf{V}$ from an untrimmed video v. The video v is divided into several segments with a fixed length (*e.g.*, 16 frames), and the segment features $\mathbf{f} = [f_1, f_2, ..., f_L] \in \mathbb{R}^{L \times d_v}$ are extracted with a 3D CNN model, where L is the number of video segments and d_v is the feature dimension. Since the temporal localization regression is related to the position inside a video, we aggregate $\mathbf{f}$ with a learnable positional encoding to get the final visual features $\mathbf{V} \in \mathbb{R}^{L \times d}$:

$$\mathbf{V} = \text{ReLU}(\mathbf{W}_{seg} \cdot \mathbf{f}) + \mathbf{f}_{pos}, \tag{1}$$

where $\mathbf{W}_{seg} \in \mathbb{R}^{d \times d_v}$ is a learnable matrix for segment features and $\mathbf{f}_{pos}$ is learnable positional encoding obtained by a lookup table [6].

3.2.2 Text Encoder: We then extract both word features $\mathbf{w}$ and sentence feature q using Bi-LSTM, and send them into a phrase generation module to generate phrases and extract fine-grained phrase features. Specifically, given a sentence query Q containing m words, we first extract the word features $\mathbf{w} = [w_1, ..., w_m] \in \mathbb{R}^{m \times d_0}$ with pretrained word2vec model. Then we use a two-layer Bi-LSTM to obtain context-awared word and sentence features. The i-th word feature is described as $\tilde{w}_i = [\overrightarrow{w_i}; \overleftarrow{w_i}] \in \mathbb{R}^d$, where $\overrightarrow{w_i}$ and $\overleftarrow{w_i}$ are the hidden states in the forward and backward LSTMs, and $[\cdot;\cdot]$ represents concatenation operation. Similarly, the sentence features are obtained by the concatenation of the last hidden states of the Bi-LSTM: $q = [\overrightarrow{w_m}; \overleftarrow{w_1}] \in \mathbb{R}^d$.

To automatically determine the meaningful phrases in the sentence, we use a phrase generation module to identifies different phrases via attention mechanism, and generates phrase-level features following [10, 29]. Following [17], we extract the phrases iteratively. To obtain the n-th phrase feature $\tilde{q}^{(n)} \in \mathbb{R}^d$, we feed the sentence-level feature q and the $(n-1)$-th phrase feature $\tilde{q}^{(n-1)} \in \mathbb{R}^d$ into embedding to get the guiding vector $g^{(n)} \in \mathbb{R}^d$:

$$g^{(n)} = \text{ReLU}(\mathbf{W}_g[\mathbf{W}_q^{(n)} q; \tilde{q}^{(n-1)}]) \tag{2}$$

where N is a hyper-parameter representing the quantity of phrases in a query, $\mathbf{W}_g \in \mathbb{R}^{d \times 2d}$, $\mathbf{W}_q^{(n)} \in \mathbb{R}^{d \times d}$, are learnable embedding matrices. Then we gain the word attention weights $\alpha^{(n)} \in \mathbb{R}^m$ of the n-th phrase through attention mechanism:

$$\begin{aligned} \tilde{\alpha}_k^{(n)} &= \mathbf{W}_{qatt}(\tanh(\mathbf{W}_{g\alpha} g^{(n)} + \mathbf{W}_{\mathbf{w}\alpha} \tilde{w}_k)) \\ & k = 1, 2, ..., m \\ \alpha^{(n)} &= \text{softmax}([\tilde{\alpha}_1^{(n)}, ..., \tilde{\alpha}_m^{(n)}]), \end{aligned} \tag{3}$$

where N is a hyper-parameter representing the quantity of phrases in a query, $\mathbf{W}_{qatt} \in \mathbb{R}^{1 \times \frac{d}{2}}$, $\mathbf{W}_{g\alpha} \in \mathbb{R}^{\frac{d}{2} \times d}$, and $\mathbf{W}_{w\alpha} \in \mathbb{R}^{\frac{d}{2} \times d}$ are learnable embedding matrices. Then we compute a weighted sum of $\tilde{w}$ to obtain phrase features $\tilde{q}^{(n)}$ as follows:

$$\tilde{q}^{(n)} = \sum_{k=1}^{m} \alpha_k^{(n)} \cdot \tilde{w}_k, n = 1, 2, ..., N \tag{4}$$

To encourage the phrase attentions corresponding to different phrases to be as distinct as possible, we apply the phrase loss $\mathcal{L}_{phrase}$ following [14]:

$$\mathcal{L}_{phrase} = ||(\mathbf{A}^T \mathbf{A}) - \lambda \mathbf{I}||_F^2, \tag{5}$$

where $\mathbf{A} = [\alpha^{(1)}, \ldots, \alpha^{(N)}] \in \mathbb{R}^{m \times N}$, represents the query attention weights across N steps, $||\cdot||_F$ denotes the Frobenius matrix

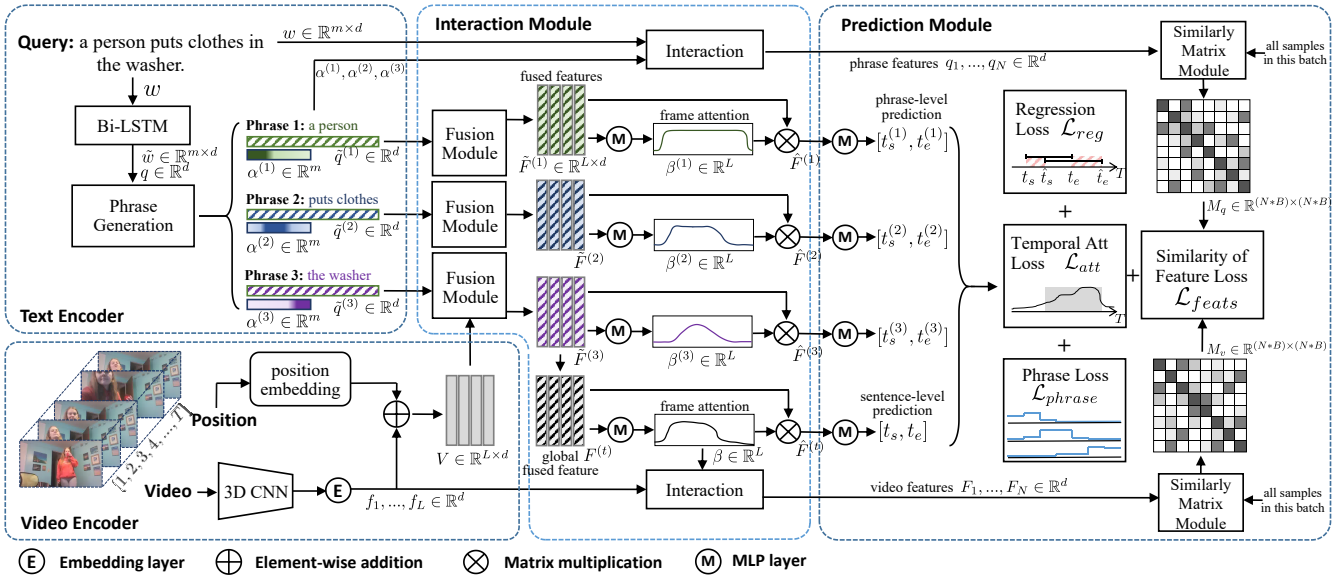

Figure 2: An overview of our proposed PLPNet method. The model contains four modules, a text encoder to extract sentence features and generate N phrases, a video encoder to extract visual features with position embedding, an interaction module where both sentence and phrase features interact with video features, harvesting fused features used in prediction part, and a prediction module to give both sentence-level and phrase-level predictions. We apply the similarity of feature loss aiming at shorten the distance between corresponding text and visual features, the overall regression loss to supervise the prediction timestamps, temporal attention loss to supervise the frame-level attention score, and phrase loss to generate different phrases in training.

norm[14], and λ is a hyperparameter in range [0, 1] wich controls the extent of overlap between different phrases.

3.2.3 Video-Text Interaction: To highlight the video features relevant to the phrases and suppress the irrelevant ones, we apply a fusion module to generate phrase-aware visual feature $\tilde{\mathbf{F}}^{(n)}$ combining video features $\mathbf{V}$ and the n-th phrase feature $\tilde{q}^{(n)}$. Then, all phrase-aware visual features should be aggregated cross phrases to generate global visual features $\mathbf{F}^{(t)}$. Finally, for each phrase-aware visual feature $\tilde{\mathbf{F}}^{(n)}$ and the global visual features $\mathbf{F}^{(t)}$, we summarize the information temporally and highlight important frames to get aggregated feature $\hat{\mathbf{F}}^{(n)}$ and $\hat{\mathbf{F}}^{(t)}$ respectively. The aggregated features can be directly used to predict the temporal boundaries of phrases and sentence.

In detail, To highlight the video features relevant to the phrases, we use Hadamard product [11] to fuse video features $\mathbf{V} = [v_1, ...v_L]$ and phrase features $\tilde{\mathbf{q}} = [\tilde{q}^{(1)}, ..., \tilde{q}^{(N)}]$:

$$\tilde{\mathbf{m}}_i^{(n)} = \mathbf{W}_m^{(n)}(\mathbf{W}_v^{(n)} v_i \odot \mathbf{W}_q^{(n)} \tilde{q}^{(n)}) \quad i = 1, 2, ..., L; \quad n = 1, 2, ..., N \tag{6}$$

where $\tilde{\mathbf{m}}_i^{(n)} \in \mathbb{R}^d$ stands for the i-th fused feature with the n-th phrase, $\mathbf{W}_m^{(n)} \in \mathbb{R}^{d\times d}$, $\mathbf{W}_v^{(n)} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_q^{(n)} \in \mathbb{R}^{d\times d}$ are learnable embedding matrices, and $\odot$ represents the Hadamard product operator[11]. Then, we apply a residual connection block that consists of two temporal convolution layers on $\tilde{\mathbf{m}}_i^{(n)}$ to generate the phrase-aware fused feature $\tilde{\mathbf{F}}^{(n)} = \text{ResBlock}([\tilde{\mathbf{m}}_1^{(n)}, ..., \tilde{\mathbf{m}}_L^{(n)}]) \in \mathbb{R}^{L\times d}$. Meanwhile, the phrase features $[\tilde{q}^{(1)}, ..., \tilde{q}^{(N)}]$ are sent into another MLP layer to generate the attention weight $\gamma \in \mathbb{R}^N$ of each phrase:

$$\gamma = \text{softmax}(\text{MLP}_{satt}([\tilde{q}^{(1)}...\tilde{q}^{(n)}]) \tag{7}$$

where MLP_{satt} is a fully connected layer. γ represents the importance of each phrase, and is used to aggregate with the phrase features to generate the sentence-aware feature $\tilde{\mathbf{F}}^{(t)} \in \mathbb{R}^{L\times d}$:

$$\tilde{\mathbf{F}}^{(t)} = \sum_{n=1}^{N} \gamma^{(n)} \tilde{\mathbf{F}}^{(n)} \tag{8}$$

Finally, the final global visual feature $\mathbf{F}^{(t)} \in \mathbb{R}^{L\times d}$ is generated with a global context modeling process, which aims to integrate the semantics of the context in case that some references in one phrase are defined in other phrase:

$$\mathbf{F}^{(t)} = \tilde{\mathbf{F}}^{(t)} + (\mathbf{W}_{rv}\tilde{\mathbf{F}}^{(t)})\text{softmax}(\frac{(\mathbf{W}_{rq}\tilde{\mathbf{F}}^{(t)})^T \cdot (\mathbf{W}_{rk}\tilde{\mathbf{F}}^{(t)})}{\sqrt{d}})^T \tag{9}$$

where $\mathbf{W}_{rv} \in \mathbb{R}^{d\times d}$, $\mathbf{W}_{rq} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_{rk} \in \mathbb{R}^{d\times d}$ are learnable embedding matrices.

Then we calculate the frame attention weight $\beta^{(n)} \in \mathbb{R}^L$ for the n-th phrase-aware visual feature and $\beta^{(t)} \in \mathbb{R}^L$ for the sentence-aware visual feature with MLP layers:

$$\beta^{(n)} = \text{softmax}(\text{MLP}_{fatt}(\mathbf{F}^{(n)})), n = 1, 2, ..., N \tag{10}$$

$$\beta^{(t)} = \text{softmax}(\text{MLP}_{fatt}(\mathbf{F}^{(t)})) \tag{11}$$

After that we apply the frame attention weight onto visual features to gain the final aggregated features $\hat{F}$ as shown below:

$$\hat{F}^{(n)} = \sum_{i=1}^{L} \beta_i^{(n)} \tilde{\mathbf{F}}_i^{(n)}, \hat{F}^{(t)} = \sum_{i=1}^{L} \beta_i^{(t)} \tilde{\mathbf{F}}_i^{(t)} \tag{12}$$

3.2.4 Prediction Module: In prediction module, the model uses fused phrase-aware features $\hat{F}^{(1)}, ..., \hat{F}^{(N)}$ and the sentence-aware feature $\hat{F}^{(t)}$ as inputs, and outputs $N + 1$ pairs of prediction, including N phrase-level predictions $[t_s^{(n)}, t_e^{(n)}], n \in [1, ..., N]$ and a sentence-level prediction $[t_s, t_e]$:

$$[t_s, t_e] = \text{MLP}_{reg}(\hat{F}^{(t)}), [t_s^{(n)}, t_e^{(n)}] = \text{MLP}_{reg}(\hat{F}^{(n)}), \tag{13}$$

where MLP_{reg} is a fully connected layer. We utilize the result of sentence-level prediction $[t_s, t_e]$ to calculates the smoothL1 distance with the ground truth $[\hat{t}_s, \hat{t}_e]$ which serves as the regression loss $\mathcal{L}_{reg}$:

$$\mathcal{L}_{reg} = \text{smoothL1}(\hat{t}_s - t_s) + \text{smoothL1}(\hat{t}_e - t_e), \tag{14}$$

Due to the lack of ground-truth supervision for phrase-level predictions, we design a self-supervised feature loss $\mathcal{L}_{feature}$ by hypothesizing that similar sentences and phrases naturally tend to focus on videos with similar contents in the ideal scenario. We first use $\mathbf{M}_q$ to represent the similarity of any two phrases in a batch. Specifically, we aggregate the word embeddings $\mathbf{W}$ by the n-th phrase attention weights $alpha^{(n)}$ obtained by eq. (3):

$$q^{(n)} = \sum_{k=1}^{m} \alpha_k^{(n)} \cdot w_k \tag{15}$$

Then, we collect all the phrase features in a batch and normalize them into unit vectors, and they make up a collection of phrase features $\mathbb{Q} \in \mathbb{R}^{BN \times d}$, where B is the batch size. For any two phrases in a batch, we calculate their cosine similarity and obtain the similarity matrix $\mathbf{M}_q$ for phrases, as shown in fig. 3:

$$\mathbf{M}_q = \mathbb{Q}\mathbb{Q}^T \in \mathbb{R}^{BN \times BN} \tag{16}$$

We can use the similar method to obtain the matrix $\mathbf{M}_v$, which represents the similarity of the visual features attended by any two phrases. With the attention weights $\beta^{(1)}, ..., \beta^{(N)}$ obtained in eq. (10), we harvest phrase and sentence attention features $F^{(n)} \in \mathbb{R}^d$ via computing the weighted sum as follows:

$$F^{(n)} = \sum_{i=1}^{L} \beta_i^{(n)} f_i \tag{17}$$

We normalize the visual features, collect them in a batch into a collection $\mathbb{F}$, and compute the similarity matrix $\mathbf{M}_v$:

$$\mathbf{M}_v = \mathbb{F}\mathbb{F}^T \in \mathbb{R}^{BN \times BN} \tag{18}$$

As we assume that similar sentences and phrases naturally tend to focus on videos with similar contents in the ideal scenario, the distance of $\mathbf{M}_v$ and $\mathbf{M}_q$ should be controlled in a margin, and we design our similarity of feature loss $\mathcal{L}_{feature}$ as:

$$\mathcal{L}_{feature} = \sum_{i,j} \max\{0, (|\mathbf{M}_{v_{i,j}} - \mathbf{M}_{q_{i,j}}| - m_0)\} \tag{19}$$

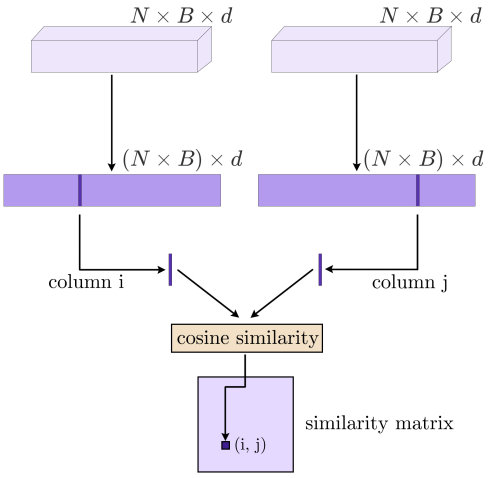

Figure 3: An illustration of the similarity matrix module. It takes weighted features from one batch as input. After concatenating all features in a batch, we compute the cosine similarity between every two columns. N, B, d denote the number of phrases, the batch size, and the dimension of one phrase-level feature, respectively.

where m_0 is a hyperparameter denoting the margin. We expect that $\mathbf{M}_v$ and $\mathbf{M}_q$'s distance falls within the scope of $[-m_0, m_0]$.

To encourage the model to gain attention weight β with higher quality and increase the weights where the video contents match the query, we apply the temporal attention loss as in [30].

$$\mathcal{L}_{attention} = \text{BinaryCrossEntropy}(\beta, \text{tag}) \tag{20}$$

where tag$\in \mathbb{R}^L$ is a vector containing only 0 and 1. If frame i locates in the ground truth period, then $\text{tag}_i = 1$. Otherwise, $\text{tag}_i = 0$.

3.3 Data Sampling Method

In order to maximize the effectiveness of $\mathcal{L}_{feature}$, it is expected that there are phrases with high similarity in a batch, also a batch should contain some contrastive phrases. As a result, we develop a dataset sampling method considering the likelihood of sentence queries and the potential combination of phrases based on the hypothesis that similar sentences are more likely to contain similar phrases. We first cluster all sentences into several classes according to their similarity and select sentences from different classes to form batch.

Upon harvesting sentence features from pretrained sentence-transformer[19], which maps sentences or paragraphs to a 768 dimensional dense vector space and can be used in many cases. We divide sentences into different clusters using K-Means based on their level of similarity, trying to gather sentences with high similarity in the same cluster. We shuffle these clusters of sentences before forming batches while the inner-category sequence remains unchanged. This sampling method aims to provide batches that train the model with similar phrases rather consecutively, hypothesizing that similar phrases have an inclination of focusing on similar videos, and sentences with a higher level of resemblance have a

Method	feature	sentence prediction				phrase prediction			
		IoU=0.3	IoU=0.5	IoU=0.7	mIoU	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
LGI [17]	I3D	72.18	59.17	35.32	50.93	60.62	32.08	12.44	36.71
MIGCN [34]		71.69	57.10	33.25	49.35	59.06	37.45	17.73	38.26
RaNet [9]		72.50	**60.46**	**38.33**	**52.03**	56.49	36.76	18.70	37.54
Ours[1]	VGG	57.82	41.88	20.56	39.12	46.24	22.94	7.69	28.46
Ours[2]	I3D	**73.49**	58.95	35.48	51.53	**63.49**	**40.52**	**19.27**	**40.76**

Table 1: Sentence-level and Phrase-level prediction accuracy on original test split of Charades-STA.

higher possibility of containing similar phrases. To raise the effectiveness of the $\mathcal{L}_{feats}$ and provide randomness, we choose queries from several batches.

3.4 Training and Inference

The total loss of our model is as follows.

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda_{att}\mathcal{L}_{attention} + \lambda_{phr}\mathcal{L}_{phrase} + \lambda_{feats}\mathcal{L}_{feature}$$

where λ is for balancing the losses.

In inference condition, after generating the phrases, we only apply sentence-level and phrase-level predictions to generate our output, without going through the similarity matrix modules. In detail, word attentions $\alpha^{(1)}, ..., \alpha^{(N)}$ are only used to calculate phrase features $\tilde{q}^{(1)}, ..., \tilde{q}^{(N)}$, without being sent into interaction module and generate matrix $\mathbf{M}_q$. Similarity, frame attentions $\beta^{(1)}, ..., \beta^{(L)}$ will not be sent in interaction module and generate matrix $\mathbf{M}_v$.

4 EXPERIMENTS

4.1 Datasets

Charades-STA. Charades-STA[8] is formed from the original Charades dataset. It includes 9848 videos shot indoors, 12408 sentence queries with annotated ground truth for training, and 3720 for testing. Non-complex and complex sentences contain 6.3 words and 12.4 words on average, respectively. It also provides annotations of combinations of 33 verbs and 38 objects, which can be used for phrase-level evaluation.

ActivityNet Captions. ActivityNet Captions [13] originates from the original ActivityNet dataset. ActivityNet Captions contains 20k videos from YouTube and 37417, 17505, and 17031 sentence queries with annotated ground truth for training, validation, and testing. Descriptions of the videos in the dataset have an average length of 13.48 words. It provides 7654 annotations of verbs and objects.

4.2 Experiment Settings

Metrics. We apply R@n, IoU = m and mIoU, which calculates the IoU between the top n retrieved video segments and the ground truth, to measure the accuracy of both sentence-level and phrase level prediction following [8]. We set n to 1 and use three threshold values, $m = \{0.3, 0.5, 0.7\}$ to evaluate the performance.

Implementation Details. For video encoder, we employ I3D[1] and C3D[2] networks to extract features for Charades-STA and ActivityNet Captions separately and sample $L = 128$ segments from each video. For query encoder, we use pretrained word2vec model[15] to extract word features of each sentence and unify the length of each sentence to $m = 10$ for Charades-STA and $m = 25$ for ActivityNet Captions. For Charades-STA, we extract $N = 3$ phrases from each query and set the parameter λ_{phrase} to 0.3 in $\mathcal{L}_{phrase}$, while for Activitynet Captions, we set N to 5 and λ_{phrase} to 0.2. We use Adam optimizer[12] to learn the parameters with a batch of 100 pairs of video and query in training. For sampling, we employ K-Means to classify all queries in training set to $K = 1000$ and $K = 3000$ classes for Charades-STA and ActivityNet Captions separately. The dimension d of features is 512 and the learning rate is 0.0004. Hyper-parameters λ_{reg}, λ_{att}, λ_{phr} and λ_{feats} are all set to 1.0.

Combinational Generazation. To demonstrate that learning more fine-grained phrase-level predictions is beneficial to improve model's generalization ability to new combinations of seen phrases (combinational generazation), we put forward a new dataset split on Charades-STA. Inspired by data splitting methods proposed in some weakly supervised settings[7, 25], we aim to test the model's performance in this scenario: the data distributions of training set and testing set are different. We split the Charades-STA dataset as below to maximize the variance of phrases in the training section. In practise, we gather frequent combinations of nouns and verbs in our new training set and the remaining sentence-video pairs forms the new testing set. We make sure that the new training set contains all verbs and nouns. The new split dataset contains 14059 and 2069 sentences in training and testing set separately.

Due to the lack of annotation focused on phrases, we split our dataset based on previous annotation focusing on verbs, objects and the combination of verbs and objects in the video. First, we choose the most frequent combination of verb and object and add all videos that contain this specific combination into the new training set. Then we check the current training set and obtain a set of all combinations of verb and object in current training set. After that we renew the training set by adding all videos that contain the combinations in the above set, and repeat this operation until the number of videos in the new training set reaches the limit we have set. The remaining videos form the new testing set. We aim to cover

[1]https://github.com/piergiaj/pytorch-i3d
[2]http://activity-net.org/challenges/2016/download.html#c3d

Method	sentence prediction				phrase prediction			
	IoU=0.3	IoU=0.5	IoU=0.7	mIoU	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
2D-TAN[33]	59.45	44.51	27.38	—	51.71	42.19	32.22	—
LGI[17]	58.48	41.65	24.10	41.48	35.39	21.07	9.76	25.14
MMN[27]	**65.05**	**48.59**	**29.26**	—	**51.91**	**42.27**	**32.88**	—
RaNet[9]	60.96	45.59	28.67	**44.82**	47.44	37.51	27.58	**38.45**
MIGCN[34]	60.03	44.94	27.85	43.59	42.25	33.75	16.37	30.90
Ours	56.92	39.20	20.91	39.53	50.10	38.12	25.24	37.96

Table 2: Sentence-level and phrase-level prediction accuracy on ActivityNet Captions.

Method	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
VSLNet[31]	70.16	53.74	33.92	**49.65**
LGI[17]	69.41	56.07	31.66	48.38
Ours	**70.76**	**57.27**	**34.07**	49.58

Table 3: Performance when testing the combinational generazation on Charades-STA

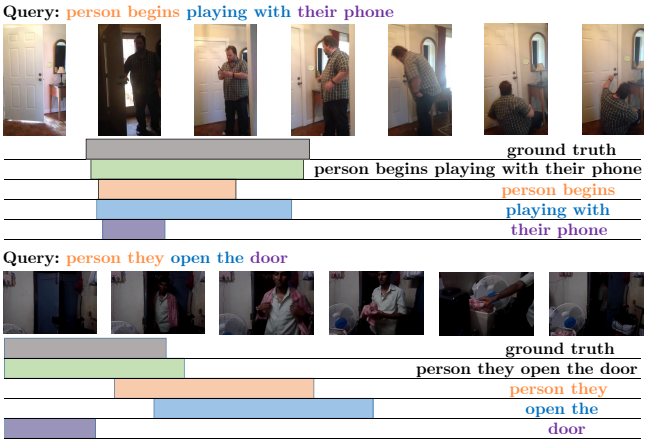

Figure 4: An illustration of our model's performance on Charades-STA. In both examples, our model achieve satisfying performance on both phrase-level and sentence-level predictions.

all objects and verbs in training set for deriving the model's ability to understand and deal with unseen combinations.

The split avoids this scenario: data distribution is inappropriate for training and testing by stop adding videos into the training set upon reaching a manually set limit.

4.3 Performance of Prediction

We compare existing approaches[17, 31] and our model's ability to deal with sentence-level and phrase-level prediction. We utilize the annotations of combinations of verb and object on Charades-STA and annotations of verbs on ActivityNet Captions as ground truth provided by the datesets. Since most existing models[8, 31] do not tackle phrase-level prediction, we generate phrase-level input and feed it directly into the whole-sentence prediction network, to report their phrase-level prediction accuracy.

As shown in table 1, PLPNet surpasses many competing existing methods[17, 31] on Charades-STA. It performs better than LGI[17], which notices phrases in training. It surpasses LGI by 2.87%, 8.44% and 6.83%, in terms of R@1 IoU={0.3,0.5,0.7} of phrases. Our model addresses the importance of understanding simple, fine-grained concepts, leading to a better performance than previous works in phrase-level prediction, demonstrating our improvement in the model's interpretability and generalization performance. We visualize two examples on Charades-STA as shown in fig. 4.

Results of experiments on ActivityNet-Captions are shown in table 2. Although we do not achieve best performance on ActivityNet-Captions, we still outperforms LGI (which serves as our baseline). It surpasses LGI by 14.71%, 17.05%, and 15.48%, in terms of IoU={0.3,0.5,0.7]} for phrase-level prediction, which demonstrates the effectiveness of our approach. We hypothesize that the reason why our method's performance is less satisfying is that, proposal-based approaches like [33] degrade less when dealing with phrases. What's more, LGI does not preform well when dealing with ActivityNet Captions as a baseline. For our sampling method and similarity matrix module are dismountable, we can migrate them to stronger baselines to gain better performance in our future work.

4.4 Combinational Generazation

We conduct a study on the combinational generazation of different models with our dataset splitting method described in section 4.2 on Charades-STA, where novel combination of seen phrases are tested to prove models' combinational generazation. We trained methods on our defined training set and used our defined testing set to get the sentence-level prediction accuracy.

As shown in table 3, our method achieves a better prediction accuracy than all other models[17, 31] in all metrics above. These results demonstrate that our method is more competent in dealing with unseen combinations of seen phrases, indicating a better generalization performance. It also proves that learning phrase-level prediction helps to understand sentences in a fine-grained way, rather than capturing sentence annotation bias in the dataset.

5 ABLATION STUDIES

In this section, we empirically investigate how the performance of the proposed method is affected by different model settings

Methods		Sentence prediction				Phrase prediction			
		IoU=0.3	IoU=0.5	IoU=0.7	mIoU	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
Number N	PLPNet(N=1)	72.90	59.46	34.33	50.85	58.25	34.01	15.63	37.15
	PLPNet(N=2)	72.02	58.74	35.48	50.75	**64.80**	**38.37**	**17.87**	**40.60**
	PLPNet(N=4)	70.97	59.01	34.84	50.09	60.18	34.58	15.55	37.78
Phrase generate	PLPNet(average)	66.83	52.93	29.57	46.12	59.30	35.79	16.37	38.19
Loss function	PLPNet w/o $\mathcal{L}_{attention}$	60.83	47.12	24.03	41.81	49.22	23.36	7.61	30.28
	PLPNet w/o $\mathcal{L}_{phrase}$	72.12	57.82	31.83	49.72	61.06	36.79	16.61	38.85
	PLPNet w/o $\mathcal{L}_{feature}$	73.28	59.25	34.97	51.33	62.94	36.20	16.21	39.14
Full Model	PLPNet(N=3, full)	**73.49**	**58.95**	**35.48**	51.53	**63.49**	**40.52**	19.27	**40.76**

Table 4: Ablation Study of number of phrases N, methods to generate phrases and each component in loss function $\mathcal{L}$ using I3D features on original test split of Charades-STA.

IoU=0.5	k=500	k=1000	k=2000	random
B=50	34.54	<u>36.53</u>	35.32	35.74
B=100	33.30	**40.52**	35.23	36.95
B=200	36.00	34.67	<u>37.83</u>	37.63

Table 5: Phrase-level performance of different parameters in sampling method on original test split of Charades-STA.

on Charades-STA dataset. We study mainly in two aspects: the contribution of network component and the sampling mechanism.

5.1 Network Components

We conduct detailed ablation study by examining the effectiveness of each proposed component in our model as shown in table 4.

We firstly evaluate the effect of the number of phrases N from a sentence. The results as shown in the first three rows in table 4 show that setting $N = 3$ achieves the best sentence-level prediction accuracy, and setting $N = 2$ achieves the best phrase-level performance. We used $N = 3$ for Charades-STA dataset throughout the paper unless otherwise specified.

We secondly verify the effectiveness of the proposed phrase generation component. We report another baseline in fourth row PLPNet(average), where the phrases are obtained by dividing the sentence query into three parts in the pre-process step. Specifically, in PLPNet(average), $\mathcal{L}_{phrase}$ is removed from the loss function when training. Our proposed approach (last row) outperforms this baseline by a large margin, which indicates that effectiveness of the learnable phrase generation component.

Finally, We conduct a detailed ablation study by examining the effectiveness of each proposed loss terms in our network structure, including $\mathcal{L}_{attention}$, $\mathcal{L}_{phrase}$ and $\mathcal{L}_{feature}$. We remove each of them and compare the performance with the full model. Results are shown in the last three rows in table 4, which imply that all components of our loss terms contribute to the performance.

5.2 Sampling Method

We compare our sampling method using K-Means with shuffling method while forming batches and the effect of batch size and the number of clusters in K-means on sentence-level prediction. B, K denote batch size of the training dataloader and the number of clusters in K-means, respectively. The last column named random shows the results only using random sampling, which can be assumed that K is taken as infinity and is a model without our sampling method. Performance is evaluated in IoU=0.5 in table 5.

From the results we observe that when selecting proper K, no matter what B is, it is effective to use our sampling method. Generally speaking, when we have larger batches, each batch tends to contain more similar and dissimilar pairs of phrases, which benefits the loss $\mathcal{L}_{feature}$ and have better performance.

6 CONCLUSION

In this work, we address the importance of phrase-level prediction. Increasing phrase-level prediction accuracy improves the model's interpretability and generalization performance. We present the PLPNet considering the phrase-level prediction to tackle the low-performance of phrase-level prediction. Our method integrates the information of the predictions of phrases and the whole query sentence, improving the model's ability to accommodate different application scenarios. We also define a data sampling for datasets used, considering the potential effect of the combination of phrases. Our experiments on Charades-STA and ActivityNet Captions show that the model improve the performance of the baseline model in both sentence-level and phrase-level prediction, demonstrating the interpretability of our model. The performance comparison of models trained using our sampling method and the original method demonstrates that our sampling increases the model's ability.

Limitation and Future Work: Although we have improved the performance on phrase-level prediction, the connection of phrase and the whole query is still complicated. We will look further at the relationship between phrases and sentence to accurately ground the sentences with more words or more complex structure.

7 ACKNOWLEDGMENTS

This work is supported by Zhejiang Lab (NO. 2022NB0AB05), State Key Laboratory of Media Convergence Production Technology and Systems, National Engineering Laboratory for Big Data Analysis and Applications Technology.

REFERENCES

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. VQA: Visual Question Answering. International Journal of Computer Vision 123 (2015), 4–31.

[2] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. VQA: Visual Question Answering. International Journal of Computer Vision 123 (2015), 4–31.

[3] Pol Albacar, Oscar Lorente, Eduard Mainou, and Ian Riera. 2021. Video Surveillance for Road Traffic Monitoring. ArXiv abs/2105.04908 (2021).

[4] Jianguo Chen, Kenli Li, Qingying Deng, Keqin Li, and Philip S. Yu. 2019. Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing. ArXiv abs/1904.06400 (2019).

[5] Shaoxiang Chen and Yu-Gang Jiang. 2020. Hierarchical Visual-Textual Graph for Temporal Activity Localization via Language. In European Conference on Computer Vision. Springer, 601–618.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL.

[7] Zhiyuan Fang, Shu Kong, Zhe Wang, Charless Fowlkes, and Yezhou Yang. 2020. Weak Supervision and Referring Attention for Temporal-Textual Association Learning. arXiv preprint arXiv:2006.11747 (2020).

[8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In Proceedings of the IEEE international conference on computer vision. 5267–5275.

[9] Jialin Gao, Xin Sun, Mengmeng Xu, Xi Zhou, and Bernard Ghanem. 2021. Relation-aware Video Reading Comprehension for Temporal Language Grounding. ArXiv abs/2110.05717 (2021).

[10] Drew A. Hudson and Christopher D. Manning. 2018. Compositional Attention Networks for Machine Reasoning. ArXiv abs/1803.03067 (2018).

[11] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv:1610.04325 (2016).

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980 (2015).

[13] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In International Conference on Computer Vision (ICCV).

[14] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. ArXiv abs/1703.03130 (2017).

[15] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In ICLR.

[16] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. 2019. Weakly supervised video moment retrieval from text queries. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 11592–11601.

[17] Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10810–10819.

[18] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Fine-grained iterative attention network for temporal language localization in videos. In Proceedings of the 28th ACM International Conference on Multimedia. 4280–4288.

[19] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. ArXiv abs/1908.10084 (2019).

[20] Cristian Rodriguez-Opazo, Edison Marrese-Taylor, Basura Fernando, Hongdong Li, and Stephen Gould. 2021. DORi: Discovering Object Relationships for Moment Localization of a Natural Language Query in a Video. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 1079–1088.

[21] P. Shah, Arpit Garg, and Vandit Gajjar. 2021. PeR-ViS: Person Retrieval in Video Surveillance using Semantic Description. 2021 IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW) (2021), 41–50.

[22] Nobuyuki Shimizu, Na Rong, and Takashi Miyazaki. 2018. Visual Question Answering Dataset for Bilingual Image Understanding: A Study of Cross-Lingual Transfer Using Attention Maps. In Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1918–1928. https://aclanthology.org/C18-1163

[23] Hao Wang, Zheng-Jun Zha, Xuejin Chen, Zhiwei Xiong, and Jiebo Luo. 2020. Dual path interaction network for video moment localization. In Proceedings of the 28th ACM International Conference on Multimedia. 4116–4124.

[24] Hao Wang, Zheng-Jun Zha, Liang Li, Dong Liu, and Jiebo Luo. 2021. Structured Multi-Level Interaction Network for Video Moment Localization via Language Query. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7026–7035.

[25] Zheng Wang, Jingjing Chen, and Yu-Gang Jiang. 2021. Visual Co-Occurrence Alignment Learning for Weakly-Supervised Video Moment Retrieval. In Proceedings of the 29th ACM International Conference on Multimedia. 1459–1468.

[26] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. 2021. Negative Sample Matters: A Renaissance of Metric Learning for Temporal Grounding. CoRR abs/2109.04872 (2021).

[27] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. 2021. Negative Sample Matters: A Renaissance of Metric Learning for Temporal Grounding. ArXiv abs/2109.04872 (2021).

[28] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. 2021. Boundary Proposal Network for Two-Stage Natural Language Video Localization. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 2986–2994.

[29] Sibei Yang, Guanbin Li, and Yizhou Yu. 2019. Dynamic Graph Attention for Referring Expression Comprehension. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019), 4643–4652.

[30] Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To Find Where You Talk: Temporal Sentence Localization in Video with Attention Based Location Regression. In AAAI.

[31] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020. Span-based localizing network for natural language video localization. arXiv preprint arXiv:2004.13931 (2020).

[32] Mingxing Zhang, Yang Yang, Xinghan Chen, Yanli Ji, Xing Xu, Jingjing Li, and Heng Tao Shen. 2021. Multi-Stage Aggregated Transformer Network for Temporal Language Localization in Videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 12669–12678.

[33] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020. Learning 2d temporal adjacent networks for moment localization with natural language. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. 12870–12877.

[34] Zongmeng Zhang, Xianjing Han, Xuemeng Song, Yan Yan, and Liqiang Nie. 2021. Multi-Modal Interaction Graph Convolutional Network for Temporal Language Localization in Videos. IEEE Transactions on Image Processing 30 (2021), 8265–8277.

[35] Yang Zhao, Zhou Zhao, Zhu Zhang, and Zhijie Lin. 2021. Cascaded Prediction Network via Segment Tree for Temporal Video Grounding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4197–4206.

[36] Minghang Zheng, Yanjie Huang, Qingchao Chen, Yuxin Peng, and Yang Liu. 2022. Weakly Supervised Temporal Sentence Grounding with Gaussian-based Contrastive Proposal Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[37] Hao Zhou, Chongyang Zhang, Yan Luo, Yanjun Chen, and Chuanping Hu. 2021. Embracing Uncertainty: Decoupling and De-bias for Robust Temporal Grounding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8445–8454.