Local-Global Video-Text Interactions for Temporal Grounding

Jonghwan Mun[1,2] Minsu Cho[1] Bohyung Han[2]
[1]Computer Vision Lab., POSTECH, Korea
[2]Computer Vision Lab., ASRI, Seoul National University, Korea
[1]{jonghwan.mun,mscho}@postech.ac.kr [2]bhhan@snu.ac.kr

Abstract

This paper addresses the problem of text-to-video temporal grounding, which aims to identify the time interval in a video semantically relevant to a text query. We tackle this problem using a novel regression-based model that learns to extract a collection of mid-level features for semantic phrases in a text query, which corresponds to important semantic entities described in the query (e.g., actors, objects, and actions), and reflect bi-modal interactions between the linguistic features of the query and the visual features of the video in multiple levels. The proposed method effectively predicts the target time interval by exploiting contextual information from local to global during bi-modal interactions. Through in-depth ablation studies, we find out that incorporating both local and global context in video and text interactions is crucial to the accurate grounding. Our experiment shows that the proposed method outperforms the state of the arts on Charades-STA and ActivityNet Captions datasets by large margins, 7.44% and 4.61% points at Recall@tIoU=0.5 metric, respectively.

1. Introduction

As the amount of videos in the internet grows explosively, understanding and analyzing video contents (*e.g.*, action classification [3, 7, 28] and detection [18, 22, 26, 27, 31, 34, 36, 38]) becomes increasingly important. Furthermore, with the recent advances of deep learning on top of large-scale datasets [2, 16, 17, 23], research on video content understanding is moving towards multi-modal problems (*e.g.*, video question answering [17, 23], video captioning [16, 24]) involving text, speech, and sound.

This paper addresses the problem of text-to-video temporal grounding, which aims to localize the time interval in a video corresponding to the expression in a text query. Our main idea is to extract multiple *semantic phrases* from the text query and align them with the video using *local and global interactions* between linguistic and visual features. We define the semantic phrase as a sequence of words that may describe a semantic entity such as an actor, an object, an action, a place, etc. Fig. 1(a) shows an example of temporal grounding, where a text query consists of multiple semantic phrases corresponding to actors (*i.e.*, 'the woman') and actions (*i.e.*, 'mixed all ingredients', 'put it in a pan', 'put it in the oven'). This example indicates that a text query can be effectively grounded onto a video by identifying relevant semantic phrases from the query and properly aligning them with corresponding parts of the video.

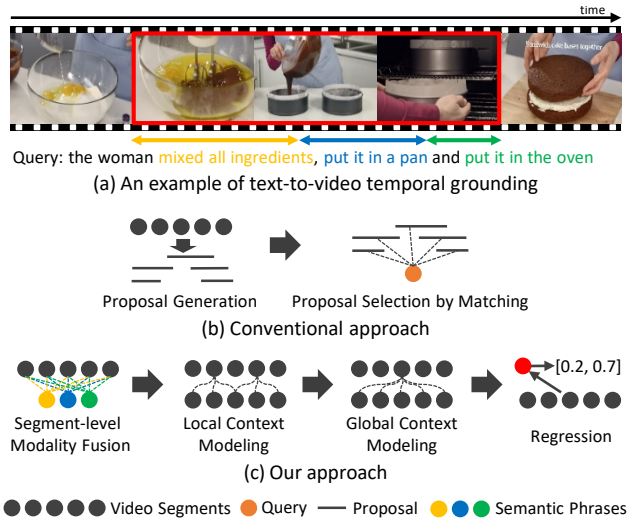


Figure 1. Video-to-text temporal grounding. (a) An example where the target time interval (red box) consists of multiple parts related to semantic phrases in a text query. (b) *Scan-and-localize* framework that localizes the target time interval by comparing individual proposals with the whole semantics of the query. (c) Our method that regresses the target time interval with the bi-modal interactions in three levels between video segments and semantic phrases identified from a query.

Leveraging such semantic phrases of a text query, however, has never been explored in temporal grounding. Most existing methods [1, 4, 5, 8, 9, 20, 32, 37] tackle the problem typically in the *scan-and-localize* framework, which in a nutshell compares a query with all candidate proposals of time intervals and selects the one with the highest matching

score as shown in Fig. 1(b). During the matching procedure, they rely on a single global feature of the query rather than finer-grained features in a phrase level, thus missing important details for localization. Recent work [35] formulates the task as an attentive localization by regression and attempts to extract semantic features from a query through an attention scheme. However, it is still limited to identifying the most discriminative semantic phrase without understanding comprehensive context.

We propose a novel regression-based method for temporal grounding as depicted in Fig. 1(c), which performs local-global video-text interactions for in-depth relationship modeling between semantic phrases and video segments. Contrary to the existing approaches, we first extract linguistic features for semantic phrases in a query using sequential query attention. Then, we perform video-text interaction in three levels to effectively align the semantic phrase features with segment-level visual features of a video: 1) segment-level fusion across the video segment and semantic phrase features, which highlights the segments associated with each semantic phrase, 2) local context modeling, which helps align the phrases with temporal regions of variable lengths, and 3) global context modeling, which captures relations between phrases. Finally, we aggregate the fused segment-level features using temporal attentive pooling and regress the time interval using the aggregated feature.

The main contributions are summarized as follows:

- We introduce a sequential query attention module that extracts representations of multiple and distinct semantic phrases from a text query for the subsequent video-text interaction.
- We present an effective local-global video-text interaction algorithm that models the relationship between video segments and semantic phrases in multiple levels, thus enhancing final localization by regression.
- We conduct extensive experiments to validate the effectiveness of our method and show that it outperforms the state of the arts by a large margin on both Charades-STA and ActivityNet Captions datasets.

2. Related Work

2.1. Temporal Action Detection

Recent temporal action detection methods often rely on the state-of-the-art object detection and segmentation techniques in the image domain, and can be categorized into the following three groups. First, some methods [22, 26] perform frame-level dense prediction and determine time intervals by pruning frames based on their confidence scores and grouping adjacent ones. Second, proposal-based techniques [27, 31, 36, 38] extract all action proposals and refine their boundaries for action detection. Third, there exist some approaches [18, 34] based on single-shot detection like SSD [21] for fast inference. In contrast to the action detection task, which is limited to localizing a single action instance, temporal grounding on a video by a text requires to localize more complex intervals that would involve more than two actions depending on the description in sentence queries.

2.2. Text-to-Video Temporal Grounding

Since the release of two datasets for text-to-video temporal grounding, referred to as DiDeMo and Charades-STA, various algorithms [1, 8, 9, 20, 37] have been proposed within the *scan-and-localize* framework, where candidate clips are obtained by scanning a whole video based on sliding windows and the best matching clip with an input text query is eventually selected. As the sliding window scheme is time-consuming and often contains redundant candidate clips, more effective and efficient methods [4, 5, 32] are proposed as alternatives; a LSTM-based single-stream network [4] is proposed to perform frame-by-word interactions and the clip proposal generation based methods [5, 32] are proposed to reduce the number of redundant candidate clips. Although those methods successfully enhance processing time, they still need to observe full videos, thus, reinforcement learning is introduced to observe only a fraction of frames [29] or a few clips [12] for temporal grounding.

On the other hand, proposal-free algorithms [10, 25, 35] have also been proposed. Inspired by the recent advance in text-based machine comprehension, Ghosh *et al*. [10] propose to directly identify indices of video segments corresponding to start and end positions, and Opazo *et al*. [25] improve the method by adopting a query-guided dynamic filter. Yuan *et al*. [35] present a co-attention based location regression algorithm, where the attention is learned to focus on video segments within ground-truth time intervals.

ABLR [35] is the most similar to our algorithm in the sense that it formulates the task as the attention-based location regression. However, our approach is different from ABLR in the following two aspects. First, ABLR focuses only on the most discriminative semantic phrase in a query to acquire visual information, whereas we consider multiple ones for more comprehensive estimation. Second, ABLR relies on coarse interactions between video and text inputs and often fails to capture fine-grained correlations between video segments and query words. In contrast, we perform a more effective multi-level video-text interaction to model correlations between semantic phrases and video segments.

3. Proposed Method

This section describes our main idea and its implementation using a deep neural network in detail.

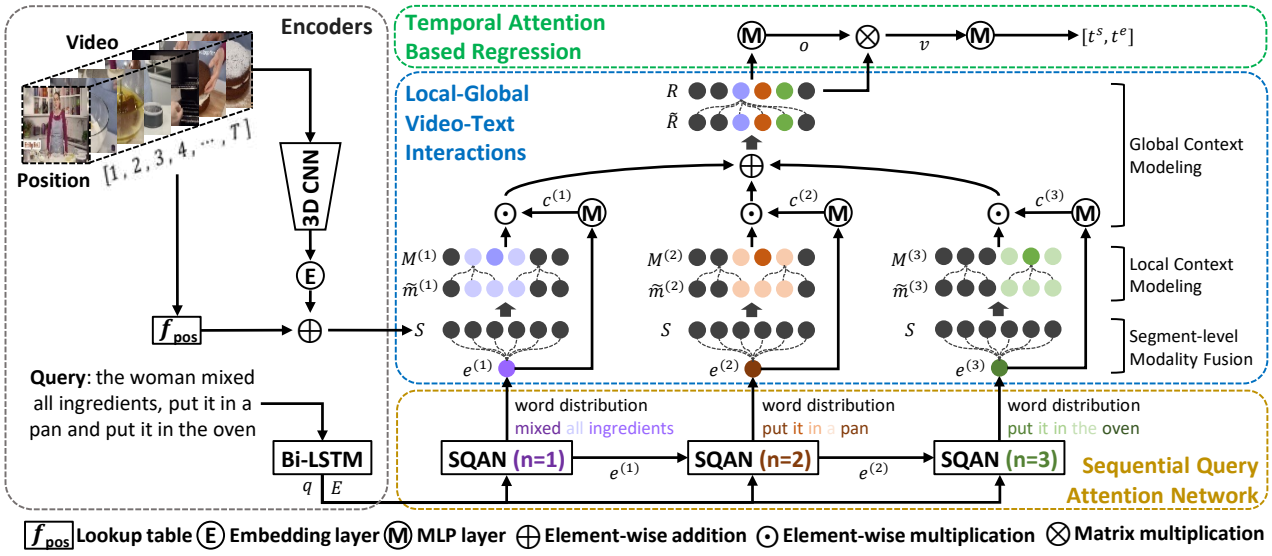


Figure 2. Overall architecture of our algorithm. Given a video and a text query, we encode them to obtain segment-level visual features, word-level and sentence-level textual features (Section 3.2). We extract a set of semantic phrase features from the query using the Sequential Query Attention Network (SQAN) (Section 3.3). Then, we obtain semantics-aware segment features based on the extracted phrase features via local-global video-text interactions (Section 3.4). Finally, we directly predict the time interval from the summarized video features using the temporal attention (Section 3.5). We train the model using the regression loss and two additional attention-related losses (Section 3.6).

3.1. Algorithm Overview

Given an untrimmed video V, a text query Q and a time interval of target region C within V, existing methods typically learn the models parametrized by θ to maximize the following expected log-likelihood:

$$\theta^* = \arg\max_{\theta} \mathbb{E}[\log p_\theta(C|V, Q)]. \quad (1)$$

Note that, in the above objective function, the text query Q often involves multiple semantic phrases as presented in Fig. 1(a), which requires modeling finer-level relations between a query and a video besides global ones to achieve precise localization in temporal grounding. To realize this idea, we introduce a differentiable module f_e to represent a query as a set of semantic phrases and incorporate local-global video-text interactions for in-depth understanding of the phrases within a video, which leads to a new objective as follows:

$$\theta^* = \arg\max_{\theta} \mathbb{E}[\log p_\theta(C|V, f_e(Q))]. \quad (2)$$

Fig. 2 illustrates the overall architecture of the proposed method. We first compute segment-level visual features combined with their embedded time stamps, and then derive word- and sentence-level features based on the query. Next, the Sequential Query Attention Network (SQAN) extracts multiple semantic phrase features from the query by attending over word-level features sequentially. Then, we obtain semantics-aware segment features via multi-level video-text interactions; the segment feature corresponding to each semantic phrase is highlighted through a segment-level modality fusion followed by local context modeling while the relations between phrases are estimated by global context modeling. Finally, the time intervals are predicted using the temporally attended semantics-aware segment features.

3.2. Encoders

Query encoding For a text query with L words, we employ a two-layer bi-directional LSTM to obtain word- and sentence-level representations, where the bi-directional LSTM is applied to the embedded word features. A word-level feature at the l-th position is obtained by the concatenation of hidden states in both directions, which is given by $\mathbf{w}_l = [\vec{\mathbf{h}}_l; \overleftarrow{\mathbf{h}}_l] \in \mathbb{R}^d$, while a sentence-level feature $\mathbf{q}$ is provided by the concatenation of the last hidden states in both the forward and backward LSTMs, *i.e.*, $\mathbf{q} = [\vec{\mathbf{h}}_L, \overleftarrow{\mathbf{h}}_1] \in \mathbb{R}^d$ where d denotes feature dimension.

Video encoding An untrimmed video is divided into a sequence of segments with a fixed length (*e.g.*, 16 frames), where two adjacent segments overlap each other for a half of their lengths. We extract the features from individual segments using a 3D CNN module, denoted by $f_v(\cdot)$, after the uniform sampling of T segments, and feed the features to an embedding layer followed by a ReLU function to match their dimensions with query features. Formally, let $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_T] \in \mathbb{R}^{d \times T}$ be a matrix that stores the T sam-

pled segment features in its columns[1]. If the input videos are short and the number of segments is less than T, the missing parts are filled with zero vectors. We append the temporal position embedding of each segment to the corresponding segment feature vector as done in [6] to improve accuracy in practice. This procedure leads to the following equation for video representation:

$$\mathbf{S} = \text{ReLU}(\mathbf{W}_{\text{seg}} f_{\text{v}}(V)) + f_{\text{pos}}(\mathbf{W}_{\text{pos}}, [1, ..., T]), \quad (3)$$

where $\mathbf{W}_{\text{seg}} \in \mathbb{R}^{d \times d_v}$ denotes a learnable segment feature embedding matrix while $f_{\text{pos}}(\cdot, \cdot)$ is a lookup table defined by an embedding matrix $\mathbf{W}_{\text{pos}} \in \mathbb{R}^{d \times T}$ and a timestamp vector $[1, \ldots, T]$. Note that d_v is the dimension of feature provided by $f_{\text{v}}(\cdot)$. Since we formulate the given task as a location regression problem, the position encoding is a crucial step for identifying semantics at diverse temporal locations in the subsequent procedure.

3.3. Sequential Query Attention Network (SQAN)

SQAN, denoted by $f_{\text{e}}(\cdot)$ in Eq. (2), plays a key role in identifying semantic phrases describing semantic entities (*e.g.*, actors, objects, and actions) that should be observed in videos for precise localization. Since there is no ground-truth for semantic phrases, we learn their representations in an end-to-end manner. To this end, we adopt an attention mechanism with an assumption that semantic phrases are defined by a sequence of words in a query as shown in Fig. 1(a). Those semantic phrases can be extracted independently of each other. Note, however, that since our goal is to obtain *distinct* phrases, we extract them by sequentially conditioning on preceding ones as in [13, 33].

Given L word-level features $\mathbf{E} = [\mathbf{w}_1, ..., \mathbf{w}_L] \in \mathbb{R}^{d \times L}$ and a sentence-level feature $\mathbf{q} \in \mathbb{R}^d$, we extract N semantic phrase features $\{\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(N)}\}$. In each step n, a guidance vector $\mathbf{g}^{(n)} \in \mathbb{R}^d$ is obtained by embedding the vector that concatenates a linearly transformed sentence-level feature and the previous semantic phrase feature $\mathbf{e}^{(n-1)} \in \mathbb{R}^d$, which is given by

$$\mathbf{g}^{(n)} = \text{ReLU}(\mathbf{W}_{\text{g}}([\mathbf{W}_{\text{q}}^{(n)}\mathbf{q}; \mathbf{e}^{(n-1)}])), \quad (4)$$

where $\mathbf{W}_{\text{g}} \in \mathbb{R}^{d \times 2d}$ and $\mathbf{W}_{\text{q}}^{(n)} \in \mathbb{R}^{d \times d}$ are learnable embedding matrices. Note that we use different embedding matrix $\mathbf{W}_{\text{q}}^{(n)}$ at each step to attend more readily to different aspects of the query. Then, we obtain the current semantic phrase feature $\mathbf{e}^{(n)}$ by estimating the attention weight vector $\mathbf{a}^{(n)} \in \mathbb{R}^L$ over word-level features and computing a weighted sum of the word-level features as follows:

$$\alpha_l^{(n)} = \mathbf{W}_{\text{qatt}}(\tanh(\mathbf{W}_{\text{g}\alpha}\mathbf{g}^{(n)} + \mathbf{W}_{\text{w}\alpha}\mathbf{w}_l)), \quad (5)$$

$$\mathbf{a}^{(n)} = \text{softmax}([\alpha_1^{(n)}, ..., \alpha_L^{(n)}]), \quad (6)$$

$$\mathbf{e}^{(n)} = \sum_{l=1}^{L} \mathbf{a}_l^{(n)} \mathbf{w}_l, \quad (7)$$

where $\mathbf{W}_{\text{qatt}} \in \mathbb{R}^{1 \times \frac{d}{2}}$, $\mathbf{W}_{\text{g}\alpha} \in \mathbb{R}^{\frac{d}{2} \times d}$ and $\mathbf{W}_{\text{w}\alpha} \in \mathbb{R}^{\frac{d}{2} \times d}$ are learnable embedding matrices in the query attention layer, and $\alpha_l^{(n)}$ is the confidence value for the l-th word at the n-th step.

3.4. Local-Global Video-Text Interactions

Given the semantic phrase features, we perform video-text interactions in three levels with two objectives: 1) individual semantic phrase understanding, and 2) relation modeling between semantic phrases.

Individual semantic phrase understanding Each semantic phrase feature interacts with individual segment features in two levels: *segment-level modality fusion* and *local context modeling*. During the segment-level modality fusion, we encourage the segment features relevant to the semantic phrase features to be highlighted and the irrelevant ones to be suppressed. However, segment-level interaction is not sufficient to understand long-range semantic entities properly since each segment has a limited field-of-view of 16 frames. We thus introduce the local context modeling that considers neighborhood of individual segments.

With this in consideration, we perform the segment-level modality fusion similar to [14] using the Hadamard product while modeling the local context based on a residual block (ResBlock) that consists of two temporal convolution layers. Note that we use kernels of large bandwidth (*e.g.*, 15) in the ResBlock to cover long-range semantic entities. The whole process is summarized as follows:

$$\tilde{\mathbf{m}}_i^{(n)} = \mathbf{W}_{\text{m}}^{(n)}(\mathbf{W}_{\text{s}}^{(n)}\mathbf{s}_i \odot \mathbf{W}_{\text{e}}^{(n)}\mathbf{e}^{(n)}), \quad (8)$$

$$\mathbf{M}^{(n)} = \text{ResBlock}([\tilde{\mathbf{m}}_1^{(n)}, ..., \tilde{\mathbf{m}}_T^{(n)}]), \quad (9)$$

where $\mathbf{W}_{\text{m}}^{(n)} \in \mathbb{R}^{d \times d}$, $\mathbf{W}_{\text{s}}^{(n)} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_{\text{e}}^{(n)} \in \mathbb{R}^{d \times d}$ are learnable embedding matrices for segment-level fusion, and $\odot$ is the Hadamard product operator. Note that $\tilde{\mathbf{m}}_i^{(n)} \in \mathbb{R}^d$ stands for the i-th bi-modal segment feature after segment-level fusion, and $\mathbf{M}^{(n)} \in \mathbb{R}^{d \times T}$ denotes a semantics-specific segment feature for the n-th semantic phrase feature $\mathbf{e}^{(n)}$.

Relation modeling between semantic phrases After obtaining a set of N semantics-specific segment features, $\{\mathbf{M}^{(1)}, ..., \mathbf{M}^{(N)}\}$, independently, we take contextual and temporal relations between semantic phrases into account.

[1]Although semantic phrases are sometimes associated with spatio-temporal regions in a video, for computational efficiency, we only consider temporal relationship between phrases and a video, and use spatially pooled representation for each segment.

For example, in Fig. 1(a), understanding 'it' in a semantic phrase of 'put it in a pan' requires the context from another phrase of 'mixed all ingredients.' Since such relations can be defined between semantic phrases with a large temporal gap, we perform *global context modeling* by observing all the other segments.

For the purpose, we first aggregate N segment features specific to semantic phrases, $\{\mathbf{M}^{(1)}, ..., \mathbf{M}^{(N)}\}$, using attentive pooling, where the weights are computed based on the corresponding semantic phrase features, as shown in Eq. (10) and (11). Then, we employ Non-Local block [30] (NLBlock) that is widely used to capture global context. The process of global context modeling is given by

$$\mathbf{c} = \text{softmax}(\text{MLP}_{\text{satt}}([\mathbf{e}^{(1)}, ..., \mathbf{e}^{(N)}])), \tag{10}$$

$$\tilde{\mathbf{R}} = \sum_{n=1}^{N} \mathbf{c}^{(n)} \mathbf{M}^{(n)}, \tag{11}$$

$$\begin{aligned} \mathbf{R} &= \text{NLBlock}(\tilde{\mathbf{R}}) \qquad (12) \\ &= \tilde{\mathbf{R}} + (\mathbf{W}_{\text{rv}}\tilde{\mathbf{R}})\ \text{softmax}\left(\frac{(\mathbf{W}_{\text{rq}}\tilde{\mathbf{R}})^{\top}(\mathbf{W}_{\text{rk}}\tilde{\mathbf{R}})}{\sqrt{d}}\right)^{\top}, \end{aligned}$$

where MLP_{satt} denotes a multilayer perceptron (MLP) with a hidden layer of $\frac{d}{2}$-dimension and $\mathbf{c} \in \mathbb{R}^N$ is a weight vector for the N semantics-specific segment features. $\tilde{\mathbf{R}} \in \mathbb{R}^{d \times T}$ is the aggregated feature via attentive pooling, and $\mathbf{R} \in \mathbb{R}^{d \times T}$ is the final semantics-aware segment features using the proposed local-global video-text interactions. Note that $\mathbf{W}_{\text{rq}} \in \mathbb{R}^{d \times d}$, $\mathbf{W}_{\text{rk}} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_{\text{rv}} \in \mathbb{R}^{d \times d}$ are learnable embedding matrices in the NLBlock.

3.5. Temporal Attention based Regression

Once the semantics-aware segment features are obtained, we summarize the information while highlighting important segment features using temporal attention, and finally predict the time interval (t^s, t^e) using an MLP as follows:

$$\mathbf{o} = \text{softmax}(\text{MLP}_{\text{tatt}}(\mathbf{R})), \tag{13}$$

$$\mathbf{v} = \sum_{i=1}^{T} \mathbf{o}_i \mathbf{R}_i, \tag{14}$$

$$t^s, t^e = \text{MLP}_{\text{reg}}(\mathbf{v}), \tag{15}$$

where $\mathbf{o} \in \mathbb{R}^T$ and $\mathbf{v} \in \mathbb{R}^d$ are attention weights for segments and summarized video feature, respectively. Note that MLP_{tatt} and MLP_{reg} have $\frac{d}{2}$- and d-dimensional hidden layers, respectively.

3.6. Training

We train the network using three loss terms—1) location regression loss $\mathcal{L}_{\text{reg}}$, 2) temporal attention guidance loss $\mathcal{L}_{\text{tag}}$, and 3) distinct query attention loss $\mathcal{L}_{\text{dqa}}$, and the total loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{tag}} + \mathcal{L}_{\text{dqa}}. \tag{16}$$

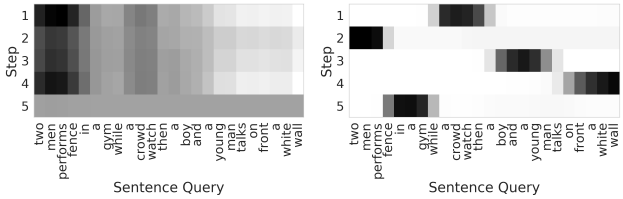


Figure 3. Visualization of query attention weights (left) without the distinct query attention loss and (right) with it. SQAN successfully extracts semantic phrases corresponding to actors and actions across different steps.

Location regression loss Following [35], the regression loss is defined as the sum of smooth L_1 distances between the normalized ground-truth time interval $(\hat{t}^s, \hat{t}^e) \in [0, 1]$ and our prediction (t^s, t^e) as follows:

$$\mathcal{L}_{\text{reg}} = \text{smooth}_{\text{L1}}(\hat{t}^s - t^s) + \text{smooth}_{\text{L1}}(\hat{t}^e - t^e), \tag{17}$$

where $\text{smooth}_{\text{L1}}(x)$ is defined as $0.5x^2$ if $|x| < 1$ and $|x| - 0.5$ otherwise.

Temporal attention guidance loss Since we directly regress the temporal positions from temporally attentive features, the quality of temporal attention is critical. Therefore, we adopt the temporal attention guidance loss proposed in [35], which is given by

$$\mathcal{L}_{\text{tag}} = -\frac{\sum_{i=1}^{T} \hat{\mathbf{o}}_i \log(\mathbf{o}_i)}{\sum_{i=1}^{T} \hat{\mathbf{o}}_i}, \tag{18}$$

where $\hat{\mathbf{o}}_i$ is set to 1 if the i-th segment is located within the ground-truth time interval and 0 otherwise. The attention guidance loss makes the model obtain higher attention weights for the segments related to the text query.

Distinct query attention loss Although SQAN is designed to capture different semantic phrases in a query, we observe that the query attention weights in different steps are often similar as depicted in Fig. 3. Thus, we adopt a regularization term introduced in [19] to enforce query attention weights to be distinct along different steps:

$$\mathcal{L}_{\text{dqa}} = ||(\mathbf{A}^{\top}\mathbf{A}) - \lambda I||_F^2, \tag{19}$$

where $\mathbf{A} \in \mathbb{R}^{L \times N}$ is the concatenated query attention weights across N steps and $||\cdot||_F$ denotes Frobenius norm of a matrix. The loss encourages attention distributions to have less overlap by making the query attention weights at

Table 1. Performance comparison with other algorithms on the Charades-STA dataset. The bold-faced numbers mean the best performance.

Method	R@0.3	R@0.5	R@0.7	mIoU
Random	-	8.51	3.03	-
CTRL [8]	-	21.42	7.15	-
SMRL [29]	-	24.36	9.01	-
SAP [5]	-	27.42	13.36	-
ACL [9]	-	30.48	12.20	-
MLVI [32]	54.70	35.60	15.80	-
TripNet [11]	51.33	36.61	14.50	-
RWM [12]	-	36.70	-	-
ExCL [10]	65.10	44.10	22.60	-
MAN [37]	-	46.53	22.72	-
PfTML-GA [25]	67.53	52.02	33.74	-
Ours	**72.96**	**59.46**	**35.48**	**51.38**

Table 2. Performance comparison with other algorithms on the ActivityNet Captions dataset. The bold-faced numbers denote the best performance.

Method	R@0.3	R@0.5	R@0.7	mIoU
MCN [1]	21.37	9.58	-	15.83
CTRL [8]	28.70	14.00	-	20.54
ACRN [20]	31.29	16.17	-	24.16
MLVI [32]	45.30	27.70	13.60	-
TGN [4]	45.51	28.47	-	-
TripNet [11]	45.42	32.19	13.93	-
PfTML-GA [25]	51.28	33.04	19.26	37.78
ABLR [35]	55.67	36.79	-	36.99
RWM [12]	-	36.90	-	-
Ours	**58.52**	**41.51**	**23.07**	**41.13**

two different steps decorrelated. Note that $\lambda \in [0, 1]$ controls the extent of overlap between query attention distributions; when λ is close to 1, the attention weights are learned to be the one-hot vector. Fig. 3 clearly shows that the regularization term encourages the model to focus on distinct semantic phrases across query attention steps.

4. Experiments

4.1. Datasets

Charades-STA The dataset is collected from the Charades dataset for evaluating text-to-video temporal grounding by [8], which is composed of 12,408 and 3,720 time interval and text query pairs in training and test set, respectively. The videos are 30 seconds long on average and the maximum length of a text query is set to 10.

ActivityNet Captions This dataset, which has originally been constructed for dense video captioning, consists of 20k YouTube videos with an average length of 120 seconds. It is divided into 10,024, 4,926, and 5,044 videos for training, validation, and testing, respectively. The videos contain 3.65 temporally localized time intervals and sentence descriptions on average, where the average length of the descriptions is 13.48 words. Following the previous methods, we report the performance of our algorithm on the validation set (denoted by *val_1* and *val_2*) since annotations of the test split is not publicly available.

4.2. Metrics

Following [8], we adopt two metrics for the performance comparison: 1) Recall at various thresholds of the temporal Intersection over Union (R@tIoU) to measure the percentage of predictions that have tIoU with ground-truth larger than the thresholds, and 2) mean averaged tIoU (mIoU). We use three tIoU threshold values, $\{0.3, 0.5, 0.7\}$.

4.3. Implementation Details

For the 3D CNN modules to extract segment features for Charades-STA and ActivityNet Captions datasets, we employ I3D [3] [2] and C3D [28] [3] networks, respectively, while fixing their parameters during a training step. We uniformly sample T $(= 128)$ segments from each video. For query encoding, we maintain all word tokens after lower-case conversion and tokenization; vocabulary sizes are 1,140 and 11,125 for Charades-STA and ActivityNet Captions datasets, respectively. We truncate all text queries that have maximum 25 words for ActivityNet Captions dataset. For sequential query attention network, we extract 3 and 5 semantic phrases and set λ in Eq. (19) to 0.3 and 0.2 for Charades and ActivityNet Captions datasets, respectively. In all experiments, we use Adam [15] to learn models with a mini-batch of 100 video-query pairs and a fixed learning rate of 0.0004. The feature dimension d is set to 512.

4.4. Comparison with Other Methods

We compare our algorithm with several recent methods, which are divided into two groups: *scan-and-localize* methods, which include MCN [1], CTRL [8], SAP [5], ACL [9], ACRN [20], MLVI [32], TGN [4], MAN [37], TripNet [11], SMRL [29], and RWM [12], and proposal-free algorithms such as ABLR [35], ExCL [10], and PfTML-GA [25].

Table 1 and Table 2 summarize the results on Charades-STA and ActivityNet Captions datasets, respectively, where our algorithm outperforms all competing methods. It is noticeable that the proposed technique surpasses the state-of-the-art performances by 7.44% and 4.61% points in terms of R@0.5 metric, respectively.

4.5. In-Depth Analysis

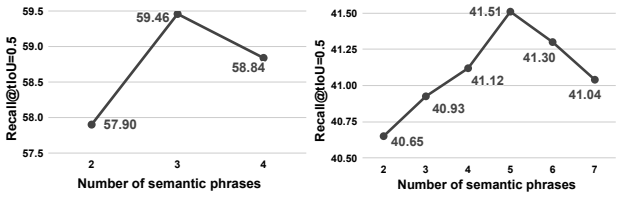
For the better understanding of our algorithm, we analyze the contribution for the individual components.

[2] https://github.com/piergiaj/pytorch-i3d

[3] http://activity-net.org/challenges/2016/download.html#c3d

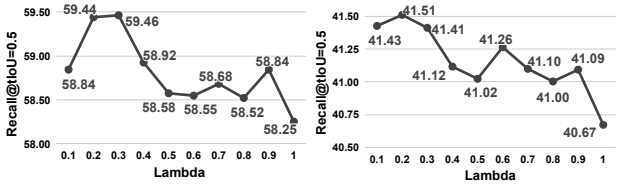
Table 3. Results of main ablation studies on the Charades-STA dataset. The bold-faced numbers means the best performance.

Method	Query Information sentence ($\mathbf{q}$)	Query Information phrase ($\mathbf{e}$)	Loss Terms $+\mathcal{L}_{\text{tag}}$	Loss Terms $+\mathcal{L}_{\text{dqa}}$	R@0.3	R@0.5	R@0.7	mIoU
LGI		√	√	√	**72.96**	**59.46**	**35.48**	**51.38**
LGI w/o $\mathcal{L}_{\text{dqa}}$		√	√		71.42	58.28	34.30	50.24
LGI w/o $\mathcal{L}_{\text{tag}}$		√		√	61.91	47.12	24.62	42.43
LGI–SQAN	√		√		71.02	57.34	33.25	49.52
LGI–SQAN w/o $\mathcal{L}_{\text{tag}}$	√				57.66	43.33	22.74	39.53



(a) Charades-STA (b) ActivityNet Captions

Figure 4. Ablation studies with respect to the number of extracted semantic phrases.



(a) Charades-STA (b) ActivityNet Captions

Figure 5. Ablation studies across with respect to λ values.

Table 4. Performance comparison by varying the combinations of modules in local and global context modeling on the Charades-STA dataset. The bold-faced numbers mean the best performance.

Local Context	Global Context	R@0.5
-	-	40.86
Masked NL (b=1, w=15)	-	42.66
Masked NL (b=4, w=15)	-	45.78
Masked NL (b=4, w=31)	-	47.80
ResBlock (k=3)	-	43.95
ResBlock (k=7)	-	46.24
ResBlock (k=11)	-	49.78
ResBlock (k=15)	-	50.54
-	NLBlock (b=1)	48.12
-	NLBlock (b=2)	48.95
-	NLBlock (b=4)	48.52
Masked NL (b=1, w=15)	NLBlock (b=1)	50.11
Masked NL (b=4, w=15)	NLBlock (b=1)	53.92
Masked NL (b=4, w=31)	NLBlock (b=1)	54.81
ResBlock (k=7)	NLBlock (b=1)	55.00
ResBlock (k=15)	NLBlock (b=1)	**57.34**

4.5.1 Main Ablation Studies

We first investigate the contribution of sequential query attention network (SQAN) and loss terms on the Charades-STA dataset. In this experiment, we train five variants of our model: 1) LGI: our full model performing local-global video-text interactions based on the extracted semantic phrase features by SQAN and being learned using all loss terms, 2) LGI w/o $\mathcal{L}_{\text{dqa}}$: LGI learned without distinct query attention loss $\mathcal{L}_{\text{dqa}}$, 3) LGI w/o $\mathcal{L}_{\text{tag}}$: LGI learned without temporal attention guidance loss $\mathcal{L}_{\text{tag}}$, 4) LGI–SQAN: a model localizing a text query with sentence-level feature $\mathbf{q}$ without SQAN, 5) LGI–SQAN w/o $\mathcal{L}_{\text{tag}}$: LGI–SQAN learned without $\mathcal{L}_{\text{tag}}$. Note that the architecture of LGI–SQAN is depicted in supplementary material.

Table 3 summarizes the results where we observe the followings. First, extracting semantic phrase features from the query (LGI) is more effective for precise localization than simply relying on the sentence-level representation (LGI–SQAN). Second, regularizing the query attention weights for distinctiveness, *i.e.*, using $\mathcal{L}_{\text{dqa}}$, enhances performance by capturing distinct constituent semantic phrases. Third, temporal attention guidance loss $\mathcal{L}_{\text{tag}}$ improves the accuracy of localization by making models focus on segment features within the target time interval. Finally, it is noticeable that LGI–SQAN already outperforms the state-of-the-art method at R@0.5 (*i.e.*, 52.02% vs.57.34%), which shows the superiority of the proposed local-global video-text interactions in modeling relationship between video segments and a query.

We also analyze the impact of two hyper-parameters in SQAN—the number of semantic phrases (N) and controlling value (λ) in $\mathcal{L}_{\text{dqa}}$—on the two datasets. Fig. 4 presents the results across the number of semantic phrases in SQAN, where performances increase until certain numbers (3 and 5 for Charades-STA and ActivityNet Captions datasets, respectively) and decrease afterwards. This is because larger N makes models capture shorter phrases and fail to describe proper semantics As shown in Fig. 5, the controlling value λ of 0.2 and 0.3 generally provides good performances while higher λ provides worse performance by making models focus on one or two words as phrases.

4.5.2 Analysis on Local-Global Video-Text Interaction

We perform in-depth analysis for local-global interaction on the Charades-STA dataset. For this experiment, we employ LGI–SQAN (instead of LGI) as our base algorithm to save training time.

Option	R@0.5
Local-Global-Fusion	46.96
Local-Fusion-Global	53.47
Fusion-Local-Global	57.34

(a) Performance comparison depending on the location of segment-level modality fusion in the video-text interaction.

Option	R@0.5
Addition	46.75
Concatenation	48.15
Hadamard Product	57.34

(b) Performance comparison with respect to fusion methods.

Option	R@0.5
None	45.70
Position Embedding	57.34

(c) Impact of position embedding for video encoding.

Table 5. Ablations on the Charades-STA dataset.

Impact of local and global context modeling We study the impact of local and global context modeling by varying the kernel size (k) in the residual block (ResBlock) and the number of blocks (b) in Non-Local block (NLBlock). For local context modeling, we also adopt an additional module referred to as a masked Non-Local block (Masked NL) in addition to ResBlock; the mask restricts attention region to a local scope with a fixed window size w centered at individual segments in the NLBlock.

Table 4 summarizes the results, which imply the followings. First, the performance of model using only segment-level modality fusion without context modeling is far from the state-of-the-art performance. Second, incorporating local or global context modeling improves performance by enhancing the alignment of semantic phrases with the video. Third, a larger scope of local view in local context modeling further improves performance, where ResBlock is more effective than Masked NL according to our observation. Finally, incorporating both local and global context modeling results in the best performance gain of 16.48% points. Note that while the global context modeling has a capability of local context modeling by itself, it turns out to be difficult to model local context by increasing the number of NLBlocks; a combination of Masked NL and NLBlock outperforms the stacked NLBlocks, showing the importance of explicit local context modeling.

When to perform segment-level modality fusion Table 5(a) presents the results from three different options for the modality fusion phase. This result implies that early fusion is more beneficial for semantics-aware joint video-text understanding and leads to the better accuracy.

Modality fusion method We compare different fusion operations—addition, concatenation, and Hadamard product. For concatenation, we match the output feature dimension with that of the other methods by employing an additional embedding layer. Table 5(b) shows that Hadamard product achieves the best performance while the other two methods perform much worse. We conjecture that this is partly because Hadamard product acts as a gating operation rather than combines two modalities, and thus helps the model distinguish segments relevant to semantic phrases from irrelevant ones.

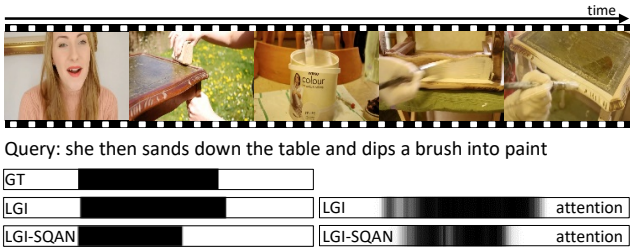


Figure 6. Visualization of predictions of two models (LGI and LGI–SQAN) and their temporal attention weights $\mathbf{o}$ computed before regression.

Impact of position embedding Table 5(c) presents the effectiveness of the position embedding in identifying semantic entities at diverse temporal locations and improving the accuracy of temporal grounding.

4.5.3 Qualitative Results

Fig. 6 illustrates the predictions and the temporal attention weights $\mathbf{o}$ for LGI and LGI–SQAN. Our full model (LGI) provides more accurate locations than LGI–SQAN through query understanding in a semantic phrase level, which makes video-text interaction more effective. More examples with visualization of temporal attention weights, query attention weights $\mathbf{a}$ and predictions are presented in our supplementary material.

5. Conclusion

We have presented a novel local-global video-text interaction algorithm for text-to-video temporal grounding via constituent semantic phrase extraction. The proposed multi-level interaction scheme is effective in capturing relationships of semantic phrases and video segments by modeling local and global contexts. Our algorithm achieves the state-of-the-art performance in both Charades-STA and ActivityNet Captions datasets.

Acknowledgments This work was partly supported by IITP grant funded by the Korea government (MSIT) (2016-0-00563, 2017-0-01780), and Basic Science Research Program (NRF-2017R1E1A1A01077999) through the NRF funded by the Ministry of Science, ICT. We also thank Tackgeun You and Minsoo Kang for valuable discussion.

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video with Natural Language. In *ICCV*, 2017.

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A Large-Scale Video Benchmark for Human Activity Understanding. In *CVPR*, 2015.

[3] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? a New Model and the Kinetics Dataset. In *CVPR*, 2017.

[4] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally Grounding Natural Sentence in Video. In *EMNLP*, 2018.

[5] Shaoxiang Chen and Yu-Gang Jiang. Semantic Proposal for Activity Localization in Videos via Sentence Query. In *AAAI*, 2019.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast Networks for Video Recognition. In *ICCV*, 2019.

[8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal Activity Localization via Language Query. In *CVPR*, 2017.

[9] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. MAC: Mining Activity Concepts for Language-based Temporal Localization. In *WACV*, 2019.

[10] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. ExCL: Extractive Clip Localization Using Natural Language Descriptions. *arXiv preprint arXiv:1904.02755*, 2019.

[11] Meera Hahn, Asim Kadav, James M Rehg, and Hans Peter Graf. Tripping through Time: Efficient Localization of Activities in Videos. *arXiv preprint arXiv:1904.09936*, 2019.

[12] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, Watch, and Move: Reinforcement Learning for Temporally Grounding Natural Language Descriptions in Videos. In *AAAI*, 2019.

[13] Drew A Hudson and Christopher D Manning. Compositional Attention Networks for Machine Reasoning. In *ICLR*, 2018.

[14] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard Product for Low-Rank Bilinear Pooling. In *ICLR*, 2017.

[15] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *ICCV*, 2017.

[17] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*, 2018.

[18] Tianwei Lin, Xu Zhao, and Zheng Shou. Single Shot Temporal Action Detection. In *ACMMM*, 2017.

[19] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A Structured Self-Attentive Sentence Embedding. In *ICLR*, 2017.

[20] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive Moment Retrieval in Videos. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot Multibox Detector. In *ECCV*, 2016.

[22] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks. *arXiv preprint arXiv:1608.08128*, 2016.

[23] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. Marioqa: Answering Questions by Watching Gameplay Videos. In *ICCV*, 2017.

[24] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined Dense Video Captioning. In *CVPR*, 2019.

[25] Cristian Rodriguez Opazo, Edison Marrese-Taylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention. *arXiv preprint arXiv:1908.07236*, 2019.

[26] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*, 2017.

[27] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal Action Localization in Untrimmed Videos via Multi-Stage Cnns. In *CVPR*, 2016.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 2015.

[29] Weining Wang, Yan Huang, and Liang Wang. Language-Driven Temporal Activity Localization: A Semantic Matching Reinforcement Learning Model. In *CVPR*, 2019.

[30] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-Local Neural Networks. In *CVPR*, 2018.

[31] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: Region Convolutional 3d Network for Temporal Activity Detection. In *ICCV*, 2017.

[32] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel Language and Vision Integration for Text-to-Clip Retrieval. In *AAAI*, 2019.

[33] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic Graph Attention for Referring Expression Comprehension. In *ICCV*, 2019.

[34] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-End Learning of Action Detection From Frame Glimpses in Videos. In *CVPR*, 2016.

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

[36] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph Convolutional Networks for Temporal Action Localization. In *ICCV*, 2019.

[37] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment Alignment Network for Natural Language Moment Retrieval via Iterative Graph Adjustment. In *CVPR*, 2019.

[38] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal Action Detection with Structured Segment Networks. In *ICCV*, 2017.

Local-Global Video-Text Interactions for Temporal Grounding

Supplementary Material

Jonghwan Mun[1,2] Minsu Cho[1] Bohyung Han[2]
[1]Computer Vision Lab., POSTECH, Korea
[2]Computer Vision Lab., ASRI, Seoul National University, Korea
[1]{jonghwan.mun,mscho}@postech.ac.kr [2]bhhan@snu.ac.kr

This supplementary document first presents the architecture of our model without semantic phrase extraction (*i.e.*, LGI–SQAN) used for in-depth analysis on the local-global video-text interactions. We also present additional qualitative examples of our algorithm.

1. Architectural Details of LGI–SQAN

Compared to our full model (LGI), LGI–SQAN does not explicitly extract semantic phrases from a query as presented in Fig. A; it performs local-global video-text interactions based on the sentence-level feature representing whole semantics of the query.

In our model, the sentence-level feature ($\mathbf{q}$) is copied to match its dimension with the temporal dimension (T) of segment-level features ($\mathbf{S}$). Then, as done in our full model, we perform local-global video-text interactions—1) segment-level modality fusion, 2) local context modeling, and 3) global context modeling—followed by the temporal attention based regression to predict the time interval $[t^s, t^e]$. Note that we adopt a masked non-local block or a residual block for local context modeling, and a non-local block for global context modeling, respectively.

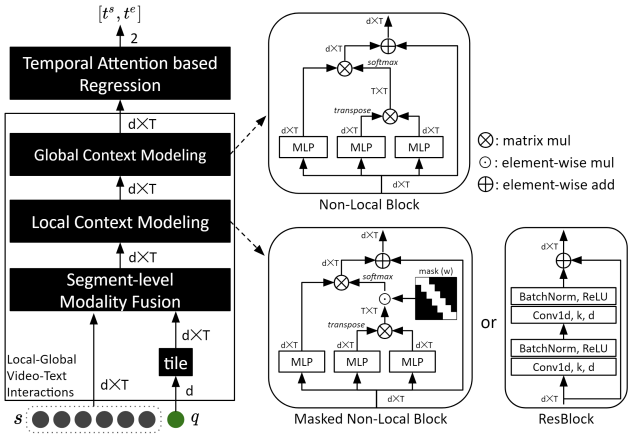


Figure A. Illustration of architecture of LGI–SQAN. In LGI–SQAN, we use sentence-level feature $\mathbf{q}$ to interact with video.

2. Visualization of More Examples

Fig. B and Fig. C illustrate additional qualitative results in the Charades-STA and ActivityNet Captions datasets, respectively; we present two types of attention weights—temporal attention weights $\mathbf{o}$ (T-ATT) and query attention weights $\mathbf{a}$ (Q-ATT)—and predictions (Pred.). T-ATT shows that our algorithm successfully attends to relevant segments to the input query while Q-ATT depicts that our sequential query attention network favorably identifies semantic phrases from the query describing actors, objects, actions, etc. Note that our model often predicts accurate time intervals even from the noisy temporal attention.

Fig. D demonstrates the failure cases of our algorithm. As presented in the first example of Fig. D, our method fails to localize the query on the confusing video, where a man looks like smiling at multiple time intervals. However, note that the temporal attention of our method captures the segments relevant to the query at diverse temporal locations in a video. In addition, as presented in the second example of Fig. D, our model sometimes fails to extract proper semantic phrases; 'wooden' and 'floorboards' are captured at different steps although 'wooden floorboards' is more natural, which results in the inaccurate localization.

Figure B. Qualitative results of our algorithm on the Charades-STA dataset. T-ATT and Q-ATT stand for temporal attention weights and query attention weights, respectively.

Figure C. Qualitative results of our algorithm on the ActivityNet Captions dataset. T-ATT and Q-ATT stand for temporal attention weights and query attention weights, respectively.

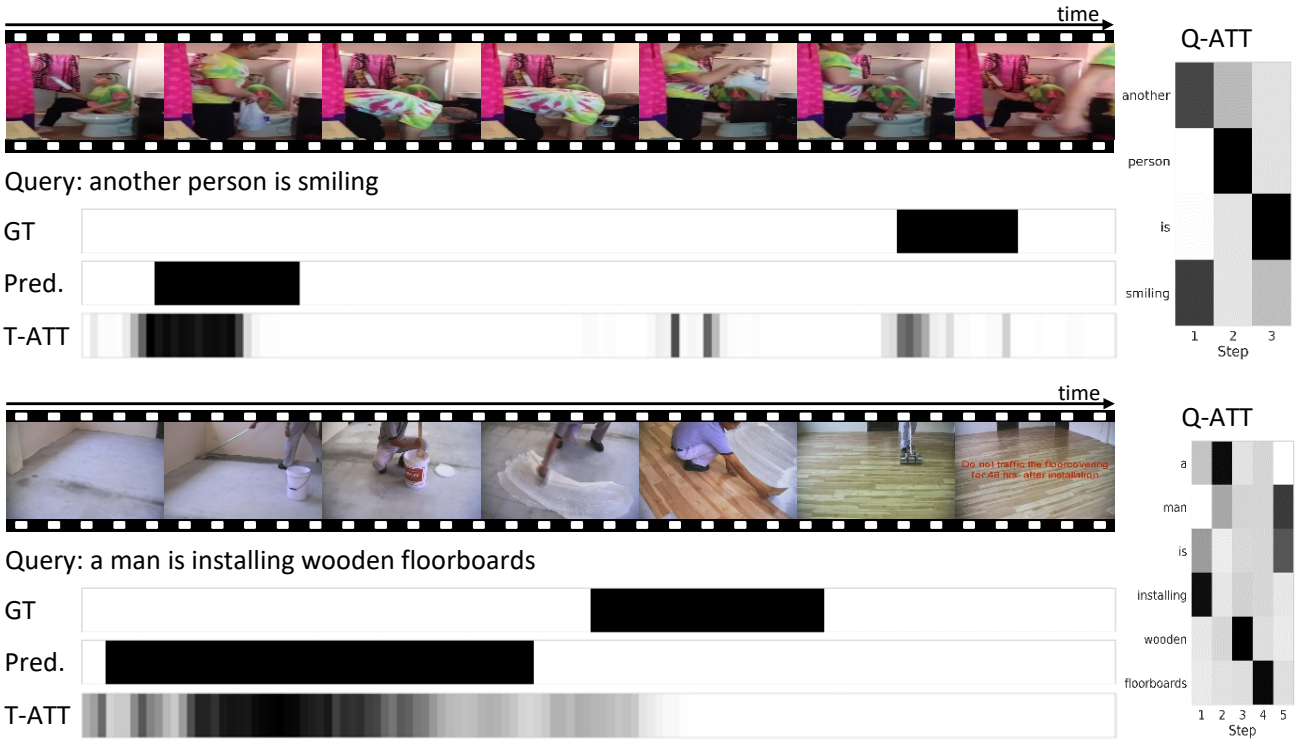


Figure D. Failure case of our algorithm. Examples in the first and second row are obtained from the Charades-STA and Activity Captions datasets, respectively.