Natural Language Video Moment Localization Through Query-Controlled Temporal Convolution

Lingyu Zhang
Rensselaer Polytechnic Institute
aileenlingyu@gmail.com

Richard J. Radke
Rensselaer Polytechnic Institute
rjradke@ecse.rpi.edu

Abstract

The goal of natural language video moment localization is to locate a short segment of a long, untrimmed video that corresponds to a description presented as natural text. The description may contain several pieces of key information, including subjects/objects, sequential actions, and locations. Here, we propose a novel video moment localization framework based on the convolutional response between multimodal signals, i.e., the video sequence, the text query, and subtitles for the video if they are available. We emphasize the effect of the language sequence as a query about the video content, by converting the query sentence into a boundary detector with a filter kernel size and stride. We convolve the video sequence with the query detector to locate the start and end boundaries of the target video segment. When subtitles are available, we blend the boundary heatmaps from the visual and subtitle branches together using an LSTM to capture asynchronous dependencies across two modalities in the video. We perform extensive experiments on the TVR, Charades-STA, and TACoS benchmark datasets, demonstrating that our model achieves state-of-the-art results on all three.

1. Introduction

Long, untrimmed videos are common, such as surveillance recordings, sports broadcasts, or conference streams. In many situations, we are only interested in particular short segments of the original video, e.g., suspicious behaviors in airports, goal replays in soccer games, or summary segments of discussion panels. This is the task of automatic video moment localization.

We are specifically interested in finding specific segments via text queries that express complex relationships between people in the videos, their activities, and their environments. For example, we would like to be able to distinguish "The janitor cleans windows after eating lunch" vs. "The husband cleans windows before going out". This is a harder problem than simply finding video segments containing "cleaning windows", which is the focus of older benchmarking datasets [12, 1].

As shown in Figure 1, given a natural sentence as a query, our objective is to find the temporal segment of a longer video that corresponds to the query (annotated as the blue area). This task requires cross-modal processing that interprets subjects, objects, and their actions and interactions in the query sentence and maps them to appearance and motion in the visual data.

Existing approaches to the problem can be divided into proposal-oriented and boundary-oriented methods. The former include top-down approaches in which multiple candidate proposals are generated at different scales [6, 11, 7] and a matching metric (e.g., a cosine function or multi-layer perceptron model) is designed to assess the semantic similarity between the current proposal and the given query. The latter are bottom-up approaches in which the start and end points of the target temporal segment are directly estimated via a regression or classification model [8] after fusing query text and video features together.

Since proposal-based methods may generate redundant negative samples that increase the computational cost for the classifier and also may have limited performance due to the hyperparameters of the proposal generation process, here we develop a boundary-based scheme. Following the pipeline in [14], the objective is to detect the possible start boundary F_{st} and end boundary F_{ed} of a target video segment from the whole video timeline. Each possible detection is associated with a confidence score p_{st} or p_{ed}. The final prediction is generated by grouping all possible F_{st} and F_{ed} detections to maximize the joint confidence score $(p_{st} \cdot p_{ed})$ under the constraint that the start boundary must occur before the end boundary $(F_{st} < F_{ed})$.

We focus on several challenges that remain unsolved in the existing literature:

- **CG1:** Frame-wise matching techniques may fail if a non-boundary frame also contains partial key actions from the query. Figure 1 illustrates a query about two people walking through a door and one putting his hands on his hips. The target temporal segment is annotated in blue. The later frame in yellow also contains

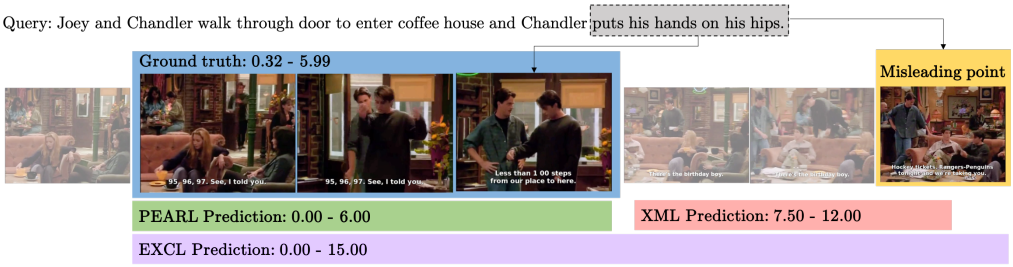

Figure 1. An example of the natural language video localization problem in the TVR dataset. The blue bar represents the ground truth segment, and the green bar shows the predictions of our proposed PEARL model. The red and purple bars show the results of the XML model [14] and EXCL model [8], respectively. The XML and EXCL models were reproduced by us using the code provided in [14] and have performance similar to that reported in the original papers.

the action "hands on hips" but there is no action "entering door" nearby, potentially leading to a false boundary point. As illustrated by the red bar, models using frame-wise consistency measurement [14] tend to fail for this scenario. Therefore, local temporal sequentiality and continuity should be emphasized for finding the correct boundary. Otherwise, false detections can arise if another non-contiguous part of the video is visually similar to the boundary frame.

- **CG2:** Handcrafted consistency metrics can fail when dealing with complex scenarios. Consistency metrics used in the existing literature include cosine similarity [25], squared distance [11] and dot-product [14]. This can limit the ability to learn complicated relationships when inferring the consistency between the video content and query sentence. A fully learnable metric is needed for performance optimization.

- **CG3:** In one of the datasets we study, the TVR dataset [14], 74% of the queries are only related to the video, 9% are only related to subtitles/closed captioning, and 17% require both visual and subtitle information. Capturing asynchronous visual and subtitle dependencies is required for a correct boundary detection. Element-wise sum [14] or multiplication may not be able to capture comprehensive relationships at different timestamps well.

Inspired by edge detectors in image processing such as the Laplacian or Sobel detectors, we think of the target segment boundaries as edge points along the timeline in the video. Similar to how edges in an image are locations with sharp changes in pixel brightness, our task is to detect the timestamps with abrupt changes in consistency between the video content in a local window and the given query.

In traditional action localization tasks such as detecting "jumping" or "surfing" segments, pixel values and motions in the segment itself can help distinguish whether or not it contains an action. On the other hand, in our task, there are no salient segments that are self-distinguishable if a query is not given. Therefore, the definition and the location of "edges" (i.e., abrupt changes in consistency) in the video are purely controlled by each specific query, and can vary significantly if the query is changed for the same video. In our approach, we learn a query-customized video boundary detector, which we call a "query filter". Unlike a Laplacian or Sobel filter that contains fixed values, or filters in traditional CNNs that are randomly initialized and trained from data, our query filter is constructed from each query sentence. In the same way edge detection is implemented using filters, we perform convolution across the timeline and obtain the boundary detection results for the query.

Our contributions can be summarized as follows:

- **Learnable consistency measurement and query-customized video edge detector.** Instead of handcrafted consistency metrics, we measure the consistency between the query sentence and each candidate through signal convolution, where the filter representation is learned from the query embedding and the target signal representation is learned from the visual and subtitle sequences.

- **Temporal dependency modeling emphasizing a controllable local window.** Since a correct target segment is expected to contain all sequential actions in the query, we use a sliding window-based pipeline, where the window size is adjustable. Compared with temporal encoding with LSTM only, our model architecture focuses on the local window around each candidate point, providing the ability to capture the local continuity of the key actions mentioned in the query.

- **Explainable model.** We visualize the response map for locating the target moments, analyze success and failure cases, and demonstrate that the model can solve the aforementioned challenges.

We evaluate our model on three benchmark datasets, TVR [14], TACoS [19] and Charades-STA [7], and demonstrate that it achieves state-of-the-art results. We discuss more details in Section 3.

2. Related Work

Our work is mainly related to two areas: boundary-oriented video moment retrieval using a query and temporal sequence modeling.

2.1. Boundary-oriented video moment retrieval

ExCL [8] is a boundary-oriented model that first encodes the whole video and query sentence features into the same vector space and concatenates them. A multilayer perceptron (MLP) or long-short term memory (LSTM) network is applied on top of the concatenated vector to generate a 2-dimensional value representing the start point and end point of the target temporal segment. VSLNet [24] applied a context-query attention mechanism that computes the similarity between the query and each clip within the whole video. The query-attended context and context-attended query are fused using element-wise multiplication and channel concatenation. The fused sequence is then fed into a set of LSTM layers to predict the boundary points.

DEBUG [16] considered each video frame as one training sample; all frames within a target segment are foreground (positive samples) and the others are considered background (negative samples). The query representation is combined with each set of frame features through concatenation and element-wise multiplication. Each frame is then classified as either background or foreground. All foreground predictions are merged and refined for final target segment prediction. Lei et al. [14] encoded cross-modal features between visual and subtitle information in the video. Query representations in the visual and subtitle spaces are generated, and consistency scores are computed between the video and query representations. A set of 1D convolutional layers is used for boundary point detection.

Since our method involves generating a query-controlled filter for signal convolution, existing models that are closer to our approach include the moment alignment network (MAN) [23] proposed by Zhang *et al*. and semantic conditioned dynamic modulation [22] proposed by Yuan *et al*. Specifically, MAN fuses the encoded video (with length l_v representing the timeline) and query sentence (with length l_q representing the number of words) through a matrix product between the feature vectors of the two at each timestamp-word pair, resulting in a matrix of size $l_q \times l_v$. On the other hand, in our approach, we apply a filter using the convolution operation from signal processing. Our query filter has a kernel size and stride, and we perform true convolution by sliding the kernel of the filter along the timeline in the context sequence to generate the response for each candidate frame. Thus for each time instant we consider a local neighborhood around the candidate frame and explore the relationship between different pieces of key information mentioned in the query with the local neighboring frames. Comparison between our model and MAN is included in Table 2.

In Yuan et al. [22], the query is used to modulate the video feature map at subsequent layers of the network. The video feature map is multiplied by a learned scale factor and added to a shifting factor. The final prediction is generated by feeding the modulated feature map into a set of stacked 1D convolution layers. The above framework is different from ours since the effect from the query is achieved by generating two parameters to reposition and rescale the feature, while our approach constructs query-controlled boundary detection filters.

2.2. Temporal sequence modeling

Temporal sequence modeling is crucial for video understanding and language perception tasks. Due to their ability to capture temporal dependencies over long periods, recurrent neural networks (RNN) including GRU [3] and LSTM [20] have been largely used in the existing literature [8, 9, 25]. Several variations of LSTM have also been developed to further improve its performance, such as hierarchical multimodal LSTM [17]. To put more attention on critical timestamps in the sequence, many existing studies investigate attention mechanisms such as transformers [21, 5, 13].

Recent research suggests that temporal convolutional neural (TCN) networks should also be considered for sequence modelling due to their smaller size compared with RNNs and their ability to capture memory over a long time range. This architecture has been used in natural language video moment localization tasks in several studies [22, 14] where the consistency score between each timestamp in the video and the query is fed into a set of stacked 1D convolutional layers. Our model is different from the existing 1D convolutional networks in that, instead of fusing video features and query language features together as a context representation and feeding them into convolutional layers, we treat the video sequence as a signal and convert the query into a filter, to investigate the effect of different queries on the signal.

3. Approach

3.1. Proposed model

In this paper, we propose **PEARL**: a novel framework for natural language video moment localization. Inspired by [4], one of the input signals is fed into an additional network to generate a dynamic representation. As shown in Figure 2, the **Perception and Abstraction** module is designed to convert each different query sentence into a filter based on a pre-defined kernel size. This will be used as a query-customized edge detector to be targeted at each video with the goal of abstracting critical information from the query. In the **Response** module, we convolve the query filter with

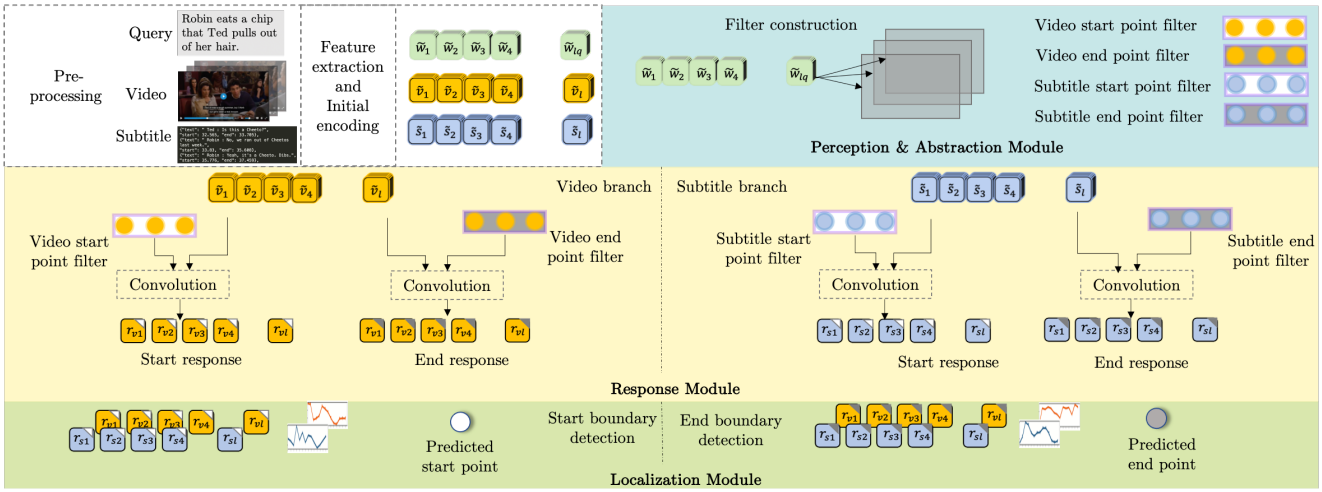

Figure 2. An overview of the framework of our approach: The raw query, video, and subtitle sequences are first fed into the feature extraction and pre-processing modules for initial temporal encoding. The encoded query is then processed by a perception and abstraction network to generate start and end point filters. We convolve the query filters with each of the context vectors and blend the responses from the visual and subtitle branches. The fused response is then integrated to finally infer the boundary points.

the video content including visual and subtitle sequences to detect query-specific edge points, i.e., the start and end boundaries controlled by the given query. The lengths of the original visual and subtitle sequences are maintained through zero-padding. When subtitle information is available, an additional **Localization** module is used to fuse information from both modalities and to provide the ability to capture visual and text dependencies at different timestamps for better predicting the boundaries. We discuss the details of each module in the following subsections.

3.2. Feature extraction and pre-processing

We adopt existing pre-trained models to obtain features from the raw video and subtitle sequences.

For the TVR dataset, we directly use the feature set provided in [14]. Specifically, the query sentence is processed by a BERT model [15]. The video sequences are processed by ResNet-152 [10] for visual appearance features and I3D [2] for motion features, and the subtitle sequences are processed by a BERT model to extract contextualized text features. While they are extracted at 3 frames per second, the feature sequences are downsampled to form short 1.5 second clips by max-pooling. For evaluation purposes, the final predicted boundary timestamps are obtained by converting the predicted clip indices into seconds using $t = \text{index} * 1.5$. Initial temporal encoding is constructed in a pre-processing step using the transformer encoder presented in [14], resulting in two context sequences $\{\tilde{v}_t\}_{t=1}^{T}$ and $\{\tilde{s}_t\}_{t=1}^{T}$ representing the visual branch and subtitle branch.

There is no subtitle data in the TACoS or Charades-STA datasets. We obtain pre-processed C3D visual features from [24, 7] for TACoS and I3D visual features from [24] for Charades-STA. The query features are extracted by a pre-trained GloVe model [18]. We follow the method in [24] to do initial temporal encoding through a 1D convolution layer.

After this module, we obtain the temporal sequences $\{\tilde{v}_t\}_{t=1}^{T}$ with feature dimension d_v and $\{\tilde{s}_t\}_{t=1}^{T}$ with feature dimension d_s for the video/subtitle and query $\{\tilde{q}_t\}_{t=1}^{T}$ with feature dimension d_q that are ready to be fed into the PEARL model for the video moment localization task.

3.3. Perception and abstraction module

The objective of this module is to construct a query-customized edge detector to locate moment boundaries in the video. The resulting filter has size $(1, k, d_f)$, where 1 is the height of the filter kernel, since we are dealing with a temporal sequence with dimensions $(1, T)$. The hyperparameter k is the width of the filter kernel, and d_f is the number of input channels of the filter, which should be equal to the number of video feature dimensions (d_v or d_s respectively) in the temporal sequence.

Within the window of a boundary frame, its left or right neighbors should have feature characteristics that correspond to sub-phrases in the query. Ideally, a start boundary is expected to have left-hand neighbors with low levels of correspondence with the query and right-hand neighbors with high levels of correspondence with the query. Additionally, neighboring frames that are closer to a boundary may have partial correspondences with the query and are expected to be higher-scoring than locations further away. Detecting this abrupt change in the correspondence results in a possible boundary detection. Therefore, each neighboring position in a local window around a boundary position is expected to have separate and distinctive relationships with the query.

As shown in Figure 3, we use multiple bi-directional

LSTMs as our filter element generators to fully exploit key actions and the sequence between them in the query sentence. Specifically, for k positions to be filled in the filter, we have k filter element generators corresponding to them.

Each filter element generator is designed to have input size d_q and hidden size $d_f/2$. When feeding the query sequence $\{\tilde{q}_t\}_{t=1}^{T}$ with feature dimension d_q into one generator, we combine the final hidden states in both directions $[\overrightarrow{h^{lq}}, \overleftarrow{h^{lq}}]$ resulting one filter element with size d_f. By constructing k separate element generators, we obtain k elements to be formed into one query-customized edge detector containing the position-aware information from the query for a local candidate window.

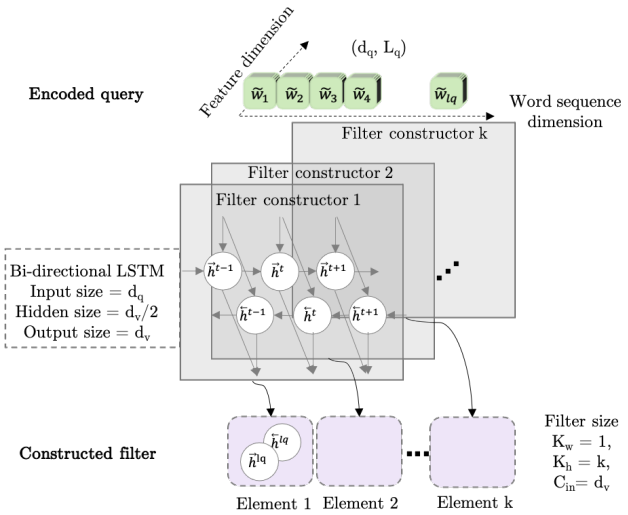

Figure 3. Details of the perception and abstraction network.

3.4. Response module

We have now constructed the filters F_{qv}^{start}, F_{qv}^{end} and F_{qs}^{start}, F_{qs}^{end} from the query for start point detection and end point detection targeting the two branches, and the context sequences $\{\tilde{v}_t\}_{t=1}^{T}$ and $\{\tilde{s}_t\}_{t=1}^{T}$ along the two branches. We first investigate the response of the context sequence to the query filter. For branch 1, the response r_v^{start} (for the visual-primary context sequence) indicating the probability for each point to be the start point is computed as

$$r_v^{start}[i] = \sum_{j=1}^{L} F_{qv}^{start}[j] \cdot c_v[i - j + L/2] \qquad (1)$$

The response r_s^{start} for branch 2 is obtained by

$$r_s^{start}[i] = \sum_{j=1}^{L} F_{qs}^{start}[j] \cdot c_s[i - j + L/2] \qquad (2)$$

Similarly, we compute the responses for the end point detection r_s^{end} and r_v^{end}.

3.5. Localization module

Two steps are performed in the **localization** module. The responses from the visual and subtitle branches are fused via a feature blender, and the final heatmap to infer the start point is generated using a boundary localizer. Specifically, we blend the intermediate response maps from the visual and subtitle branches via an LSTM module. Taking the start point detection as an example, the responses to the start point filter F_{qv}^{start} and F_{qs}^{start} are temporal maps with length T and dimension 1. We concatenate the two maps $[r_v^{start} : r_s^{start}]$ to have size $(2, T)$ and feed this into a bi-directional LSTM layer with input size 2 and hidden size d_h. At each timestamp, the response values from the visual and subtitle branches are fused along the timeline, and the temporal dependencies over a longer time range are captured. The output sequence of the LSTM with dimension (d_h, T) is then fed into a boundary localizer consisting of a layer normalization and a fully connected layer with size $(d_h, 1, T)$ to generate the final heatmap p_{start} with size $(1, T)$ to infer the most likely start point along the whole video timeline. Figure 4 illustrates the overall module.

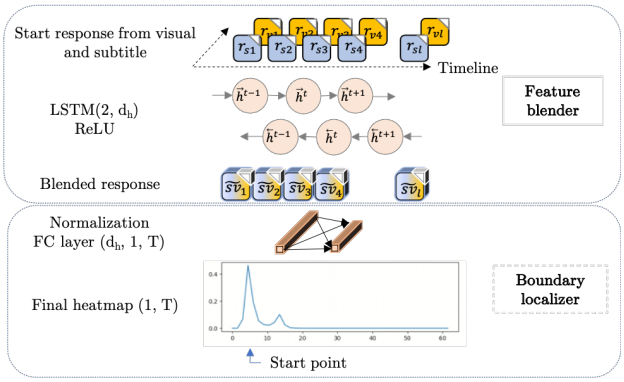

Figure 4. Details for the localization network.

3.6. End-to-end training

Given the contextualized heatmap, we sum the cross-entropy loss for the start point and end point for training:

$$\begin{aligned} L = &-(y_{start} log(p_{start}) + (1 - y_{start}) log(1 - p_{start})) \\ &-(y_{end} log(p_{end}) + (1 - y_{end}) log(1 - p_{end})) \end{aligned} \qquad (3)$$

where y_{start} and y_{end} represent the ground truth label of the start point and end point.

During inference, we adopt the post-processing step in [8, 14] that the confidence score of a segment is generated by the multiplication of the corresponding start probability and end probability with the constraints that the end point must occur after the start point.

4. Experimental Results

4.1. Datasets for Evaluation

To evaluate the performance of PEARL, we mainly focus on the TVR dataset, a recently proposed challenging dataset that requires both video and subtitles for moment localization. We also report performance metrics on the TACoS and Charades-STA datasets to further demonstrate the effectiveness of PEARL when subtitle information is not required. An overview of the three datasets is outlined in Table 1.

TVR contains 21793 videos extracted from various TV shows and 109000 natural language queries. The average length for query sentences is 13.4 words with average video length around 76 seconds, making it more difficult for the model to parse the complex information in the query sentence. The other challenging part of this benchmark is that multiple interactions between multiple people are involved in most of the videos. According to [14], 67% of the samples contain more than 1 action and 66% of them contain more than 1 person, posing challenges for the video understanding algorithm to determine the relationships and temporal dependencies between multiple behaviors.

TACoS is a dataset containing 17344 query-moment pairs involving indoor single-person cooking activities. The average length for query sentences is 11 words with average video length around 287 seconds.

Charades-STA is a widely-used benchmark that contains 16128 query-moment pairs involving daily indoor activities. Compared with TVR, the average length for query sentences is 7.2 words with average video length around 31 seconds.

4.2. Evaluation metrics

The Intersection over Union (IoU) is used to determine whether a prediction is correct, where

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{4}$$

We use ranking scores to evaluate the performance of a video moment retrieval algorithm. Specifically, a predicted temporal interval is correct if the IoU between the prediction and ground truth is larger than a threshold μ; thus $\alpha =$ Rank@n, IoU = μ means that when predicting n temporal intervals, α of them are correct.

Additionally, as reported in [24, 8], we also computed the mIoU metric for Rank@1, representing the mean ratio of intersection of union, to assess the precision of the predicted temporal window.

4.3. Implementation Details

We set the hidden size of the pre-processed features to be 512, 192, and 128 for TVR, TACoS, and Charades-STA, respectively. The size of the convolutional kernel is set to be 5, 13, and 13 on the 3 datasets. The optimal dropout rates are 0.365, 0.360, and 0.350.

4.4. Qualitative Results

In Figure 5 we visualize several examples from the TVR dataset, including one sample with a misleading point that also contains a partial query action located at a non-contiguous position in the video, one sample that has multiple simultaneous actions in the query, one sample in which the query only relates to the subtitle information, and one sample in which both visual and subtitle information are needed to infer the moment.

As shown in the top row of Figure 5, all neighbors around the correct boundary point have higher response scores because they together cover all the environmental constraints in the query *Entering the door* and *Hands on hips*. While the misleading point contains the action *Hands on hips*, its neighboring points do not meet all the constraints, so they all have a lower response. Therefore, by generating the position-aware query filter using the **perception and abstraction** module and measuring the consistency by convolving the filter with the video content sequence using the **response** module, our model produces good results for this challenge.

As shown in the second row, co-occurrent actions may actually be shown in close-up sequences in the video, so a single boundary point may not match all the constraints in the query; traditional frame-based similarity comparisons tend to fail. As we can see from Figure 5, using the **response network** that considers all the neighbors within a temporal window, the response score at the correct boundary has a higher score than at the other parts of the video, since all the key actions are covered in its neighborhood. The first row and the third row demonstrate that when the query is only related to either visual or subtitle information, our model can correctly focus on one modality and ignore the other unreliable modality in both scenarios, illustrating that the late fusion mechanism in the **localization** module provides the ability to flexibly choose from the two branches. Moreover, when the query requires both visual and subtitle data, as shown in the last row of Figure 5, the fused response map can correctly capture the critical correspondence change at different timestamps across the two modalities, making our model perform well for this case.

4.5. Quantitative comparisons

Tables 2, 3 and 4 show the results of PEARL compared with existing approaches on the TVR, TACoS, and Charades-STA datasets, respectively. We can see that PEARL surpasses existing approaches to a large extent on all 4 metrics on the TVR dataset. PEARL also achieves much higher ranking score (at IoU=0.5) and mean IoU on TACoS. The improvement of PEARL on Charades-STA is

Table 1. An overview of the three datasets. Partial statistics are directly obtained from [14].

Dataset	Example	Moments with 2 or more actions
TVR	Ross **points** to a shelf to **ask** Jack to **grab** something from the shelf.	67%
TACoS	The person **cuts** the leek, from the middle to the top, then **washes** it.	20%
Charades-STA	A person **opens** a door.	6%

Figure 5. **First row**: visualized result when there is a misleading frame in the video. **Second row**: visualized result when simultaneous actions are displayed in multiple frames. **Third row**: visualized result when the query is only related to the subtitles, **Fourth row**: visualized result when the query is related to both visual and subtitle information.

Table 2. Quantitative comparison on the TVR dataset. Results of existing approaches are directly obtained from [14].

	Model	IoU=0.5, R@1	IoU=0.5, R@5	IoU=0.7, R@1	IoU=0.7, R@5
Existing approaches	MCN [11]	16.86	40.55	7.96	21.45
	CAL [6]	17.61	42.08	8.07	21.40
	ExCL [8]	31.31	48.54	14.34	28.89
	XML [14]	31.43	51.66	13.89	31.11
Proposed	PEARL	**34.49**	52.20	**15.53**	31.43
Ablation study	RL	29.02	45.35	12.46	25.98
	PEAR-Dense	31.77	49.38	13.87	29.48
	PEAR-Add	32.88	<u>52.61</u>	<u>15.12</u>	**32.37**
	PEAR-Add-Dense	32.75	**53.09**	14.17	<u>32.09</u>
	PEARL-single	<u>33.84</u>	48.81	14.36	28.47

not as significant as for TVR and TACoS, which is reasonable since PEARL has the ability to emphasize sequential key actions in the query. As mentioned previously, 94% of the query-clip pairs in Charades-STA contain only a single action, making PEARL's strengths less apparent.

Table 3. Quantitative comparison on the TACoS dataset. Results of existing approaches are directly obtained from [24] and [8].

Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
DEBUG	23.45	11.72	–	16.03
ExCL [8]	**45.50**	<u>28.00</u>	13.80	–
VSLBase [24]	23.59	20.40	16.65	20.10
VSLNet [24]	29.61	24.27	**20.03**	<u>24.11</u>
PEARL	<u>42.94</u>	**32.07**	<u>18.37</u>	**31.08**

Table 4. Quantitative comparison on the Charades-STA dataset. Results of existing approaches are directly obtained from [24].

Model	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
ExCL [8]	65.1	44.1	23.3	-
VSLBase [24]	68.1	50.2	30.2	47.2
VSLNet [24]	<u>70.5</u>	**54.2**	<u>35.2</u>	<u>50.2</u>
PEARL	**71.9**	<u>53.5</u>	**35.4**	**51.2**

4.6. Ablation study

We verify the necessity and importance of each module in our model architecture by removing them and comparing the performance with the full model on the TVR dataset. Specifically, we consider the following baselines:

- **PEAR-Dense**: This is the baseline model by removing and replacing the LSTM-based localization module in the PEARL framework. After the temporal responses from the visual and subtitle branches are generated, following the methods used in [8], the visual and subtitle feature sequences are fused by concatenation.
- **PEAR-Add**: This is a variation of PEAR-Dense; following the strategy used in [14], visual and subtitle information are fused by averaging.
- **PEAR-Add-Dense**: This baseline is a combination of the above two, in which we average the two branches and then add dense layers to predict the final results.
- **RL**: This is the baseline model obtained by removing and replacing the Perception and Abstraction module in the PEARL framework. Following [14], we use a similar module to encode the query as a feature vector and compare the frame-wise similarity by element-wise multiplication.
- **PEARL-single**: This baseline model contains a unified filter for both start and end boundary detection.

As shown in the last 5 rows of Table 2, we observe that the **RL** model has the lowest performance, since the query-customized filter generation and edge detection mechanism in **PEARL** are removed and replaced. This demonstrates that the **Perception and Abstraction** module is crucial and necessary. **PEARL-Single** achieves lower scores, demonstrating that generating separate filters from the query for start and end boundaries is necessary. **PEAR-Add** and **PEAR-Add-Dense** achieve good performance on some metrics, but **PEARL** has the highest score for multiple metrics and achieves much higher performance at Rank@1, showing that we are able to capture critical information for detecting the query-controlled boundaries in the video.

5. Conclusions

We proposed a novel video moment retrieval framework based on convolution between pieces of the query and the video/subtitle content. The network is trained with cross-entropy loss using a classification framework. One drawback of the approach is that this loss function assigns equal penalties to all false predictions without considering the distances between different false predictions. For example, given a ground truth start point at the 6^{th} unit, a false prediction at the 7^{th} unit should have a higher score compared with a false prediction at the 20^{th} unit. Therefore, one future direction is to add a regression module to jointly reinforce the generated heatmap to have a larger value around the ground truth point to improve the performance.

6. Acknowledgements

This research was supported by a grant from the Strengthening Teamwork in Novel Groups' Collaborative Research Alliance of DEVCOM ARL (U.S. Army Research Lab) under Grant No. W911NF-19-2-0135.

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[4] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 667–675, 2016.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, June 2019.

[6] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*, 2019.

[7] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5267–5275, 2017.

[8] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. ExCL: extractive clip localization using natural language descriptions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 1984–1990, June 2019.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] Lisa A. Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5803–5812, 2017.

[12] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[13] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. 2020.

[14] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Proceedings of the European Conference on Computer Vision)*, 2020.

[15] Yinhan Liu et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[16] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5147–5156, 2019.

[17] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017.

[18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[19] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013.

[20] Jürgen Schmidhuber and Sepp Hochreiter. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

[21] Ashish Vaswani et al. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, 2017.

[22] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[23] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1247–1257, 2019.

[24] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6543–6554, July 2020.

[25] Lingyu Zhang and Richard J. Radke. Temporal attention and consistency measuring for video question answering. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, page 510–518, 2020.