Temporally Grounding Language Queries in Videos by Contextual Boundary-aware Prediction

Jingwen Wang, Lin Ma, Wenhao Jiang
Tencent AI Lab
{jaywongjaywong, forest.linma, cswhjiang}@gmail.com

Abstract

The task of temporally grounding language queries in videos is to temporally localize the best matched video segment corresponding to a given language (sentence). It requires certain models to simultaneously perform visual and linguistic understandings. Previous work predominantly ignores the precision of segment localization. Sliding window based methods use predefined search window sizes, which suffer from redundant computation, while existing anchor-based approaches fail to yield precise localization. We address this issue by proposing an end-to-end boundary-aware model, which uses a lightweight branch to predict semantic boundaries corresponding to the given linguistic information. To better detect semantic boundaries, we propose to aggregate contextual information by explicitly modeling the relationship between the current element and its neighbors. The most confident segments are subsequently selected based on both anchor and boundary predictions at the testing stage. The proposed model, dubbed Contextual Boundary-aware Prediction (CBP), outperforms its competitors with a clear margin on three public datasets.

1 Introduction

Videos are increasingly popular in the social network. As most videos contain both activities of interest and complicated background content, temporal activity localization is of key importance for video analysis. Recently, the task of temporally grounding language queries in videos has been attracting research interest from the vision community (Gao et al. 2017; Hendricks et al. 2017). The task aims to localize the activity of interest corresponding to a language query. This task is challenging because both videos and sentences need to be deeply incorporated to differentiate fine-grained details of different video segments and to perform segment localization. In this paper, we identify and tackle the main challenge on this task, namely, how to improve the localization precision of the desired segment given a language query.

Prior work predominantly ignores the precision of segment boundaries. Sliding window based methods scan the video by predefined windows of different sizes (Gao et al. 2017; Hendricks et al. 2017; Liu et al. 2018c; Wu and Han 2018; Liu et al. 2018b; Ge et al. 2019; Xu et al. 2019). Because the desired segments are of varied durations, these methods cannot guarantee the complete coverage of all segments, and thus tend to produce inaccurate temporal boundaries. Other research tried to avoid this problem by designing single-stream models (Buch et al. 2017; Chen et al. 2018) using LSTMs. Although LSTMs effectively aggregate video information, the thresholding of positive and negative samples loses boundary information. As shown in Figure 1 (b), segments overlapped with the ground truth more than a predefined threshold (e.g., 0.5) are all labeled as positive samples during training stage. Therefore, the model could be confused to localize the best matched segment at prediction. A complementary approach to improve the precision of localization is to add a location offset regression branch to the anchor-based approaches (Gao et al. 2017; Xu et al. 2019; Ge et al. 2019; Liu et al. 2018b). However, the added offset regression could fail when the model is unable to localize the best anchor, since the calculated offsets need to be added to the predicted anchor to generate final grounding time stamp

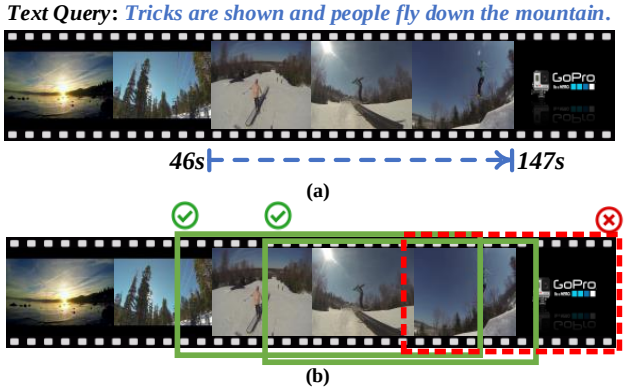

Figure 1: (a) The task of temporally grounding language queries in videos. (b) Positive and negative training segments defined in anchor-based approaches given the sentence query in (a).

(See Table 2 for comparison).

To improve temporal grounding precision, we propose a novel model that jointly predicts temporal anchors and boundaries at each time step, with a small computation overhead. At prediction stage, the anchors are modulated by boundary scores to generate boundary-aware grounding results. To detect semantic boundaries more accurately, contextual information is adaptively integrated into our architecture. As shown in Figure 1, the activity "fly down the mountain" exhibits different visual appearance compared to the background content. The activity is better localized with the aid of its surrounding information. To this end, we propose a self attention based contextual integration module, which is deeply embedded into the architecture. Different from (Gao et al. 2017; Hendricks et al. 2017; Wu and Han 2018; Ge et al. 2019) where context information is simply integrated by feature concatenation, we explicitly measure the different "contributions" by leveraging the self-attention technique. Noticeably, our proposed context module operates on the layer which already integrates query and video information. It thus enables our network to "perceive" the surrounding predictions and collect reliable contextual evidences before making predictions at the current step. This is different from previous context modeling, which only considers visual context but ignores the impact of language integration. Although LSTMs are also capable of summarizing contextual information, it suffers from the so-called "gradient vanishing/exploding" problem and could fail to memorize information for long segments. The proposed contextual model, however, shortens the path for remote elements and effectively aggregates useful contexts in the video.

To summarize, our main contributions are two-folds. First, we address the problem of temporally grounding language queries in videos with a simple yet effective boundary-aware approach, which effectively improves grounding precision in an end-to-end manner. Second, to better detect semantic boundaries, a self attention based module is designed to collect contextual clues. Based on interaction output of both language and video, it explicitly measures the contributions from different contextual elements. Our proposed contextual boundary-aware model (named as CBP) achieves compelling performance on three public datasets.

2 Related Work

The interdisciplinary research topics of vision and language have long been explored (Wang et al. 2018; 2019; 2016a; Yuan et al. 2019). Among them we emphasize the following two most relevant topics to our paper: grounding language queries in images, and grounding language queries in videos.

2.1 Grounding Language Queries in Images

Grounding language queries in images, also known as "grounding referring expressions in images", is to spatially localize the image region corresponding to a given language query. Most work follows the standard pipeline, which first generates candidate image regions using image proposal method like (Ren et al. 2015), then finds the matched one to the given query. In (Hu et al. 2016; Rohrbach et al. 2016), the target image regions were extracted based on description reconstruction error or probabilities. Some studies consider incorporating contextual information into the retrieval model (Hu et al. 2016; Yu et al. 2016; Chen et al. 2017; Zhang, Niu, and Chang 2018). These "contexts" include global contexts (Hu et al. 2016), and contexts from other candidate regions (Yu et al. 2016; Chen et al. 2017; Zhang, Niu, and Chang 2018). (Wang et al. 2016b) explored not only region-phrase relationship, but also modeled region-region and phrase-phrase structures. Some other methods exploit attention modeling in queries, images, or object proposals (Endo et al. 2017; Yu et al. 2018).

2.2 Grounding Language Queries in Videos

Temporally video grounding aims at extracting the corresponding video segment to a given language query. Early studies focus on constrained scenarios such as autonomous driving (Lin et al. 2014), or constrained setting such as alignment of multiple sentences (Bojanowski et al. 2015). Recently, (Gao et al. 2017) and (Hendricks et al. 2017) extended the task to more general scenarios. (Gao et al. 2017) proposed to jointly model video clips and text queries using multi-modal operations, then alignment scores and location offsets were predicted based on the multi-model representation. (Hendricks et al. 2017) proposed to embed both modalities into a common space and minimize the squared distances. Both (Gao et al. 2017) and (Hendricks et al. 2017) exploited temporal visual contexts for localization. (Wu and Han 2018) integrated multiple interactions between different modalities and proposed Multi-modal Circulant Fusion. (Liu et al. 2018b) designed a memory attention network to enhance the visual features. To avoid redundant computation caused by sliding windows, (Chen et al. 2018) dynamically matches language and video, and generates grounding results in one single pass. (Liu et al. 2018a) designed a temporal modular network that can exploit underlying language structure. (Ge et al. 2019) proposed to mine semantic activity concepts to enhance the temporal grounding task. (Xu et al. 2019) followed a two-stage pipeline to retrieve video clips. They first generated query-specific proposals from the videos, then leveraged caption reconstruction for training. In (Chen and Jiang 2019), a visual concept based approach was proposed to generate proposals, followed by proposal evaluation and refinement. (Wang, Huang, and Wang 2019; Hahn et al. 2019) explored reinforcement learning to find the corresponding segments to language queries.

3 Proposed Method

In this section we introduce our main framework for temporally grounding queries in videos, as shown in Figure 2. Our model consists of three main components: the query-video interaction module, the contextual integration module, and the localization module. The three components are deeply integrated and thus enable end-to-end training.

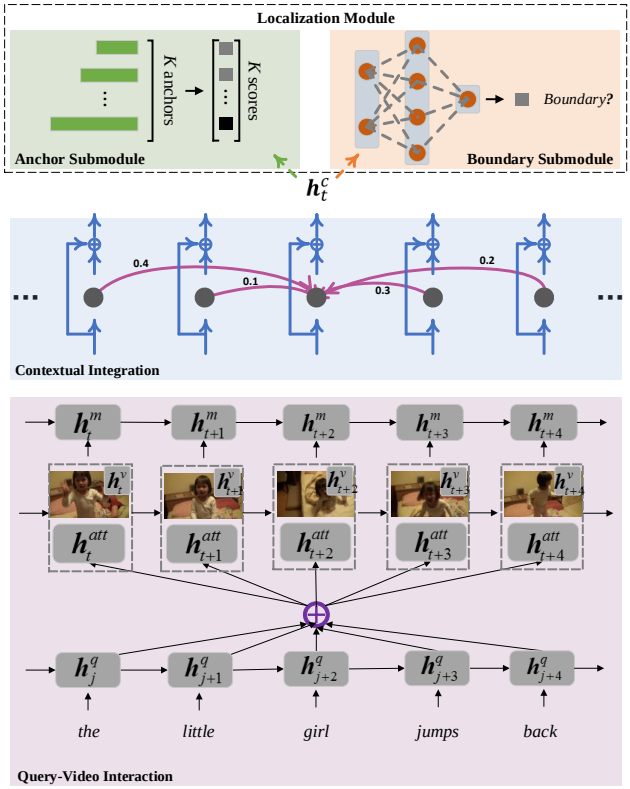

Figure 2: The main framework of our proposed method: Contextual Boundary-aware Prediction (CBP). It composes of three modules: a query-video interaction module to deeply integrate language query and video information, a contextual integration module to collect localization clues from neighboring elements, and a localization module to output segments. The localization module consists of an anchor submodule and a boundary submodule.

3.1 Problem Formulation

We denote a video as a sequence of frames $X = \{x_1, x_2, ..., x_L\}$. Each video is associated with a set of annotations: $\{(s_j, t_j^s, t_j^e)\}$, where s_j, t_j^s, t_j^e denote the query sentence, the start and end time of the annotated segment, respectively. Given the input video and the sentence query, our task is to localize the target segment. Each video is represented as a sequence of features $V = \{v_t\}_{t=1}^T$. The sentence query is represented by $Q = \{q_j\}_{j=1}^N$.

3.2 Query-Video Interaction Module

Intrinsically both videos and sentence queries are sequential signals. We incorporate Match-LSTM (Wang and Jiang 2016; Chen et al. 2018) as our backbone network to learn vision-language interaction. The Match-LSTM composes of three LSTM (Hochreiter and Schmidhuber) layers. The first LSTM incorporates textual information (denoted as "query LSTM"). The second LSTM encodes video motion and long-term dependencies from the input video (denoted as "video LSTM"). The third LSTM is responsible for summarizing video and language elements (denoted as "interaction LSTM"). The output states of the three LSTMs are $H^q = \{\mathbf{h}_j^q\}$, $H^v = \{\mathbf{h}_t^v\}$, and $H^m = \{\mathbf{h}_t^m\}$, respectively.

As shown in Figure 2, each video frame is attentively matched to different words from a query:

$$r_{tj} = \mathbf{w}_r^T \cdot \tanh(W_s \mathbf{h}_j^q + W_v \mathbf{h}_t^v + W_m \mathbf{h}_t^m + \mathbf{b}_r), \quad (1)$$

$$\alpha_{tj} = \exp(r_{tj}) / \sum_{k=1}^{N} \exp(r_{tk}), \quad (2)$$

$$\mathbf{h}_t^{att} = \sum_{j=1}^{N} (\alpha_{tj} \cdot \mathbf{h}_j^q), \quad (3)$$

$$\mathbf{h}_{t+1}^m = LSTM^m(\mathbf{h}_t^{att} || \mathbf{h}_t^v, \mathbf{h}_t^m), \quad (4)$$

where $\mathbf{h}_t^{att}$ is the attended query vector, which relies on current video LSTM state and interaction LSTM state. The attended query vector is concatenated ("||") with the video state ($\mathbf{h}_t^v$) to serve as input to the interaction LSTM to obtain next state $\mathbf{h}_{t+1}^m$.

By the above integration, we deeply summarize and integrate the query and the video.

3.3 Contextual Integration Module

To better capture the boundary information corresponding to the starting or ending of an activity, we explore contextual integration by leveraging the self attention technique (Vaswani et al. 2017) on top of the Match-LSTM. Different from pure visual contextual integration (Gao et al. 2017; Hendricks et al. 2017; 2018; Ge et al. 2019; Wu and Han 2018), our contextual integration module can strengthen and collect useful grounding clues as it operates on the layer which already integrates query and video information. We also explicitly model the different contributions from different "contexts" by assigning them with different attention weights. Formally, the input sequence to the contextual integration module is: $H^m = \{\mathbf{h}_t^m\}_{t=1,2,...,T}$, where $H^m \in \mathbb{R}^{T \times D}$. Since every pair from H^m needs to be matched, we use scaled dot-product operation to perform self attention as it enjoys high computational efficiency. The relevance matrix for H^m is:

$$Z = \frac{1}{\sqrt{d}} (H^m W^Q)(H^m W^V)^T, \quad (5)$$

where the projection matrices $W^Q, W^V \in \mathbb{R}^{D \times d}$ and $Z \in \mathbb{R}^{T \times T}$. **In practice, we keep $W^Q = W^V$ by sharing projection weights at training.** We find it helps improve the performance. The relevance matrix is then normalized to obtain the context weights α:

$$\alpha_{ij} = \exp(Z_{ij}) / \sum_{t=1}^{T} \exp(Z_{it}). \quad (6)$$

We summarize contextual elements using the learnt attention to obtain:

$$\hat{H}^c = \alpha H^m. \quad (7)$$

To avoid corrupting temporal dependency of LSTM, H^m and H^c are integrated by concatenation operation:

$$H^c = \hat{H}^c || H^m. \quad (8)$$

$H^c = \{\mathbf{h}_t^c\}_{t=1,2,...,T}$ is expected to strengthen reliable contextual evidence for localization. The operation faithfully preserves the temporal dependency of LSTM, which benefits the following prediction procedure.

3.4 Localization Module

The traditional anchor prediction focus more on coarse localization by recognizing segment content. We further propose to strengthen fine-grained semantic boundary information with an additional boundary module. The two modules share the common base network and could benefit each other at the training stage.

Anchor Submodule. We adopts similar idea as Buch *et al.* (Buch et al. 2017). We design K anchors to match different temporal durations. Each $\mathbf{h}_t^c$ in aggregates historical video information from position 0 to position t, after query-video integration. Each hidden state $\mathbf{h}_t^c$ will be fed into K independent binary classifiers and produces K confidence scores $C_t = \{c_t{}^i\}_{i=1,...,K}$ indicating the probabilities of K segments specified by $S_t = \{s_t{}^i\}_{i=1,...,K}$. $s_t{}^i$ denotes a video clip with end time as t and start time as $t - l_i$, where $\{l_i\}_{i=1}^K$ is the lengths of the predefined K anchors. The segment scores C_t are calculated by:

$$C_t = \sigma(W_c \mathbf{h}_t^c + \mathbf{b}_c), \tag{9}$$

where σ denotes *sigmoid* nonlinearity. W_c, $\mathbf{b}_c$ are shared across all time steps.

Boundary Submodule. Except for the anchor prediction, we also design a parallel branch to predict boundaries of segments. The idea of boundary modeling is simple. We take $\mathbf{h}_t^c$ as an indication of whether there is a semantic boundary at position t. Specifically, a binary classifier is trained with $\mathbf{h}_t^c$ as input. The output boundary score for current position t is:

$$B_t = \sigma(W_b \mathbf{h}_t^c + \mathbf{b}_b), \tag{10}$$

which measures how confident the LSTM is going through a semantic boundary. Intuitively, by comparing with its memory (historical video information), the LSTM decides whether the current step is a semantic boundary corresponding the start/end time of an activity (annotated segment).

3.5 Training

There are two main losses corresponding to the above two output modules.

Anchor Loss. Following (Buch et al. 2017), the anchor labels y_t (K-dim 0-1 vector) at time step t is determined by overlap threshold $\theta = 0.5$. We adopt weighted multi-label cross entropy as anchor loss $\mathcal{L}_a$. For a video X at time t:

$$\mathcal{L}_a = -\sum_{i=1}^{K} w_0^i y_t^i log c_t^i + w_1^i (1 - y_t^i) log(1 - c_t^i), \tag{11}$$

where w_0^i, w_1^i are determined based on the numbers of positive and negative samples.

Boundary Loss. Assume the training sample $V = \{v_i\}_{i=1}^T$ is associated with ground truth boundary labels $\{z_t\}_{t=1}^T$. The boundary loss is given by:

$$\mathcal{L}_b = w_+ z_t log b_t + w_- (1 - z_t) log(1 - b_t), \tag{12}$$

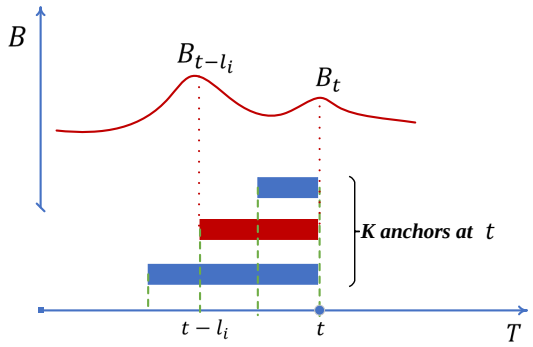

Figure 3: Local Boundary Score Fusion. The boundary prediction (red curve) helps modulate the score of each anchor. In this case the anchor in red will be selected as the most matched one.

where $b_t \in B_t$ is the boundary prediction score at temporal position t, w_+ and w_- are positive/negative weights.

Joint Training. We balance the anchor loss and the boundary loss by:

$$\mathcal{L} = \mathcal{L}_a + \lambda \times \mathcal{L}_b. \tag{13}$$

λ is determined by cross validation to balance the two loss terms. The CBP network can be trained in an end-to-end manner by minimizing the total loss $\mathcal{L}$.

3.6 Boundary-modulated Anchor Prediction

At inference stage, we calculate K anchor scores $C_t \in \mathbb{C}$ and boundary scores $B_t \in \mathbb{B}$ for each video temporal location $t \in \{1, 2, \ldots, T\}$.

Local Boundary Score Fusion. As illustrated in Section 1, the anchor module cannot well reflect boundary information and can produce high scores for many segments that have overlap with the ground truth segment. To precisely localize the target segment, we first apply local score fusion to combine both anchor scores and boundary scores at temporal location t. The new scores for the i-th anchor at time step t is:

$$\hat{c}_t^i = c_t^i + 0.5 \times (B_{t-l_i} + B_t), \tag{14}$$

where $c_t^i \in C_t = \{c_t{}^i\}_{i=1,...,K}$. By Equation (14), we obtain new scores $\hat{C}_t = \{\hat{c}_t^i\}_{i=1,...,K}$ at each time step t. As illustrated in Figure 3, we adjust the score of each anchor by taking its start boundary and end boundary into consideration.

Global Score Ranking. The final segment scores for a video are $\hat{\mathbb{C}} = \{\hat{C}_t\}_{t=1,2,...,T}$. M candidate segments with highest scores are selected and NMS (Non-Maximum Suppression) is performed to further remove redundant candidates. Please note that NMS does not affect top-1 result.

4 Experiments

We conduct extensive experiments on three public datasets: TACoS (Regneri et al. 2013), Charades-STA (Gao et al.

Table 1: Performance comparison on TACoS (Regneri et al. 2013) dataset. All results are reported in percentage (%).

Method	R@1 IoU=0.7	R@1 IoU=0.5	R@1 IoU=0.3	R@5 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.3	mIoU
Random Anchor	0.22	0.56	2.23	1.10	3.55	9.77	1.89
VSA-RNN (Karpathy and Fei-Fei 2015)	-	4.78	6.91	-	9.10	13.90	-
VSA-STV (Karpathy and Fei-Fei 2015)	-	7.56	10.77	-	15.50	23.92	-
CTRL (Gao et al. 2017)	6.96	13.30	18.32	15.33	25.42	36.69	11.98
MCF (Wu and Han 2018)	-	12.53	18.64	-	24.73	37.13	-
ACRN (Liu et al. 2018b)	-	14.62	19.52	-	24.88	34.97	-
TGN (Chen et al. 2018)	11.88	18.90	21.77	15.26	31.02	39.06	17.93
SM-RL (Wang, Huang, and Wang 2019)	-	15.95	20.25	-	27.84	38.47	-
TripNet (Hahn et al. 2019)	9.52	19.17	23.95	-	-	-	-
SAP (Chen and Jiang 2019)	-	18.24	-	-	28.11	-	-
ACL (Ge et al. 2019)	-	20.01	24.17	-	30.66	42.15	-
CBP (ours)	**19.10**	**24.79**	**27.31**	**25.59**	**37.40**	**43.64**	**21.59**

Table 2: Ablation study on TACoS (Regneri et al. 2013) dataset. All results are reported in percentage (%).

Method	R@1 IoU=0.7	R@1 IoU=0.5	R@1 IoU=0.3	R@5 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.3	mIoU
CBP baseline (Chen et al. 2018)	11.88	20.21	25.13	15.26	30.86	38.80	17.93
+ Boundary	16.02	22.26	25.52	22.90	34.90	41.76	19.46
+ Boundary, + Context (full model)	**19.10**	**24.79**	**27.31**	**25.59**	**37.40**	**43.64**	**21.59**
Replace: Concat-Context	18.37	22.97	24.88	25.77	36.42	43.35	19.98
Replace: Global-Context	16.56	21.21	25.01	23.19	35.17	43.18	19.87
Replace: Offset-Reg	17.68	24.69	27.31	22.73	36.08	42.59	20.79

2017), and ActivityNet Captions (Krishna et al. 2017). For fair comparison, we use the same settings for all baselines, including initial learning rate, segment sampling, NMS threshold, and other hyper-parameters.

4.1 Datasets

TACoS. TACoS is widely used on this task. The videos from TACoS were collected from cooking scenarios. They are around 7 minutes on average. The same split as (Gao et al. 2017) is used, which includes 10146, 4589, 4083 query-segment pairs for training, validation and testing.

Charades-STA. Charades-STA was built on Charades dataset (Sigurdsson et al. 2016), which focus on indoor activities. The temporal annotations of Charades-STA were generated in a semi-automatic way, which involved sentence decomposition, keyword matching, and human check. The videos are 30 seconds on average. The train/test split is 12408/3720.

ActivityNet Captions. ActivityNet Captions was built on ActivityNet v1.3 dataset (Caba Heilbron et al. 2015). The videos are 2 minutes on average. Different from the above three datasets, the annotated video clips in this dataset have much larger variation, ranging from several seconds to over 3 minutes. Since the test split is withheld for competition, we merge the two validation subsets "val_1", "val_2" as our test split, as (Chen et al. 2018). The numbers of query-segment pairs for train/test split are thus 37421 and 34536.

4.2 Metrics

Following prior work, we mainly adopt "R@N, IoU=θ" and "mIoU" as the evaluation metrics. "R@N, IoU=θ" represents the percentage of top N results that have at least one segment with higher IoU (Intersection over Union) than θ. "mIoU" computes the average IoU of top 1 result with ground truth segment over all testing queries.

4.3 Implementation Details

For fair comparison, C3D (Tran et al. 2015) features are adopted for all compared methods. Each word from the query is represented by GloVe (Pennington, Socher, and Manning 2014) word embedding vectors pre-trained on Common Crawl. We set hidden neuron size of LSTM to 512.

We generally design the K anchors to cover at least 95% of training segments. Therefore, we empirically set K to 32, 20 and 100 for TACoS, Charades-STA and ActivityNet Captions, respectively. The NMS thresholds are 0.3, 0.55 and 0.55, respectively.

4.4 Compared Methods

We compare our proposed CBP against the following methods: **Random Anchor**: the confidence score for each anchor is randomly generated, followed by NMS. **VSA-RNN** (Karpathy and Fei-Fei 2015): visual-semantic alignment with LSTM. **VSA-STV** (Karpathy and Fei-Fei 2015): similar as VSA-RNN, except using skip-thought vectors (Kiros et al. 2015) as query representations. **CTRL** (Gao et al. 2017): Cross-model Temporal Regression Localizer. **ACRN**

Table 3: Performance comparison on ActivityNet Captions (Krishna et al. 2017). All results are reported in percentage (%).

Method	R@1 IoU=0.7	R@1 IoU=0.5	R@1 IoU=0.3	R@5 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.3	mIoU
Random Anchor	4.54	13.28	26.64	17.95	43.40	63.65	18.40
TGN (Chen et al. 2018)	11.86	27.93	43.81	24.84	44.20	54.56	29.17
Xu *et al.* (Xu et al. 2019)	13.60	27.70	45.30	38.30	59.20	75.70	-
TripNet (Hahn et al. 2019)	13.93	32.19	48.42	-	-	-	-
CBP (ours)	**17.80**	**35.76**	**54.30**	**46.20**	**65.89**	**77.63**	**36.85**

(Liu et al. 2018b): Attentive Cross-Model Retrieval Network. **TGN** (Chen et al. 2018): Temporal GroundNet. **MCF** (Wu and Han 2018): Multi-modal Circulant Fusion. **ACL** (Ge et al. 2019): Activity Concepts based Localizer. **Xu *et al.*** (Xu et al. 2019): a two-stage method (generation + reranking) exploiting re-captioning. **SAP** (Chen and Jiang 2019): a two-stage approach based on visual concept grouping. **SM-RL** (Wang, Huang, and Wang 2019): based on reinforcement learning. **TripNet** (Hahn et al. 2019): leverages RL to perform efficient grounding.

4.5 Comparison with State-of-the-Arts

TACoS. Table 1 summarizes performances of different approaches on the test split of TACoS. "Random Anchor" is a stronger baseline than uniform random as it eliminates candidates with "impossible" durations. However, it achieves very low recalls on all the metrics, indicating that it is quite challenging to accurately localize the desired segment on TACoS. As shown in Table 1, the performance degenerates for all the methods when IoU gets higher. VSA-RNN and VSA-STV achieve unsatisfactory performance compared to the others, mainly because they do not exploit any contextual information for localization. CTRL (Gao et al. 2017), MCF (Wu and Han 2018), ACRN (Liu et al. 2018b), TripNet (Hahn et al. 2019) and ACL (Ge et al. 2019) use sliding windows to match sentences and video segments, while TGN (Chen et al. 2018), SM-RL (Wang, Huang, and Wang 2019) and our proposed method CBP adopt LSTMs to eliminate the need of sliding windows. Most sliding window based approaches perform inferior to the single-stream methods (TGN, SM-RL, CBP). ACL (Ge et al. 2019) and SAP (Chen and Jiang 2019) perform better than other sliding-window based methods, thanks to the detected visual concepts. Finally, the proposed CBP outperforms all the other methods. Noticeably, CBP maintains much better recall rates at high IoUs. For example, for the important metric "R@1, IoU=0.7" which indicates high precision, CBP outperforms the others with over 60% relative gain.

Charades-STA. The results on Charades-STA are shown in Table 4. Compared to TACoS dataset, the annotated segments from Charades-STA have a much larger coverage ratio in the video. Therefore, "Random Anchor" has much higher recall rates (e.g., 14.65 *vs* 0.22 for "R@1, IoU=0.5"). We notice that for "R@5, IoU=0.5", "Random Anchor" obtains a surprisingly high recall (54.35%). Therefore, we argue that it is better to compare different methods at high IoUs (IoU=0.7 or even higher) on this dataset. Xu *et al.* (Xu et al. 2019) leverages multiple useful techniques to enhance the grounding performance, and its results are better than CTRL (Gao et al. 2017), ACL (Ge et al. 2019), SAP (Chen and Jiang 2019), SM-RL (Wang, Huang, and Wang 2019) and TripNet (Hahn et al. 2019). For the important metric "R@1, IoU=0.7", our method obtains a recall of 18.87%, surpassing the previous best result (15.80%). For the metric "R@5, IoU=0.5", Xu *et al.* achieves better recall. One possible reason is that our model finds more false positive boundaries on this dataset.

Table 4: Performance comparison on Charades-STA (Gao et al. 2017) dataset. All results are reported in percentage (%).

Method	R@1 IoU=0.7	R@1 IoU=0.5	R@5 IoU=0.7	R@5 IoU=0.5	mIoU
Random Anchor	3.95	14.65	20.65	54.35	20.38
VSA-RNN	4.32	10.50	20.21	48.43	-
VSA-STV	5.81	16.91	23.58	53.89	-
CTRL	7.15	21.42	26.91	59.11	-
ACL	12.20	30.48	35.13	64.84	33.84
SAP	13.36	27.42	38.15	66.37	-
SM-RL	11.17	24.36	32.08	61.25	-
TripNet	14.50	36.61	-	-	-
Xu *et al.*	15.80	35.60	45.40	**79.40**	-
CBP (ours)	**18.87**	**36.80**	**50.19**	70.94	**35.74**

ActivityNet Captions. As can been seen from Table 3, our CBP surpasses both TGN (Chen et al. 2018) and Xu *et al.* (Xu et al. 2019) on all the metrics with a clear margin. The proposed CBP obtains 17.04% at "R@1, IoU=0.7" while Xu *et al.* and TripNet can only achieves 13.60% and 13.93% respectively. This provides strong evidences on the superiority of the proposed CBP. Similar to Charades-STA, many annotated segments on ActivityNet Captions occupy large portion of the video duration. Therefore, for low IoUs (e.g., IoU=0.3), many approaches perform similarly to the "Random Anchor". We also notice that CBP achieves less relative improvement over Xu *et al.* and TripNet for lower IoUs (e.g., IoU=0.3). This is because our model focus more on localization precision.

4.6 Ablation Study

We conduct ablation study on TACoS, as shown in Table 2. We observe substantial performance improvement when applying the proposed boundary module, especially for the metrics of high IoUs (e.g., "R@1,IoU=0.7",

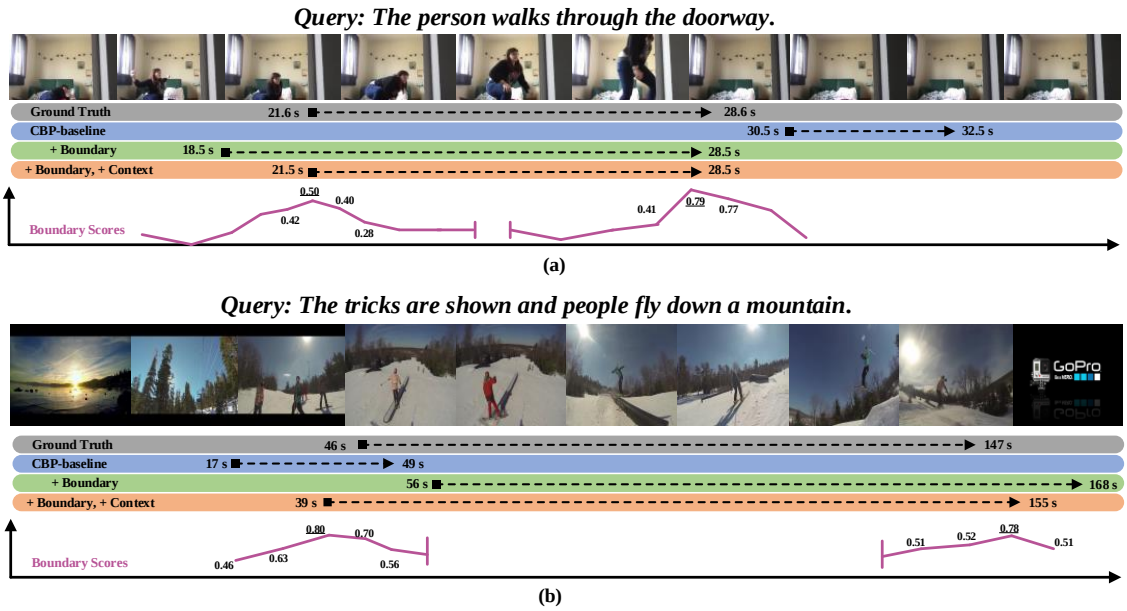

Figure 4: Prediction examples of our CBP and the baselines. The boundary scores are computed using our full CBP model.

"R@5,IoU=0.7"). Equipping with the boundary module greatly improves the grounding precision. CBP outperforms other methods when further integrating the context module ("+ Boundary, + Context"). Moreover, each module of CBP is compared to existing techniques by replacement in order to further verify the effectiveness of the proposal. The first experiment is to replace our proposed self-attention based contextual integration module with the commonly-adopted concatenation-based contextual module (Gao et al. 2017; Hendricks et al. 2017; Wu and Han 2018; Ge et al. 2019) or the global contextual module (Wang, Huang, and Wang 2019; Hendricks et al. 2017). The second one is to replace our boundary module with an offset regression branch (Gao et al. 2017; Xu et al. 2019; Ge et al. 2019; Liu et al. 2018b). The performance degeneration observed in Table 2 verifies the superiority of our proposed modules over their corresponding competitors.

4.7 Qualitative Analysis

We provide some qualitative examples to validate the effectiveness of the proposed CBP. As shown in Figure 4, the boundary prediction module exploits boundary information and modulates the anchors by combining predictions from both output modules. This makes it perform better than the CBP baseline. By contextual integration, the boundaries of the desired segment can be further recognized.

We also visualize the learnt context weights in Figure 5. Each blue box represents the ground-truth segment to be localized and each red box corresponds to the segment with the highest context weight. In Figure 5 (a), our model successfully pinpoint the desired activity "`jumps back up`" (in blue box) by attending to its precursor action "`falling down`" (in red box). In Figure 5 (b), to accurately localize the desired segment in blue box, the model resorts to the

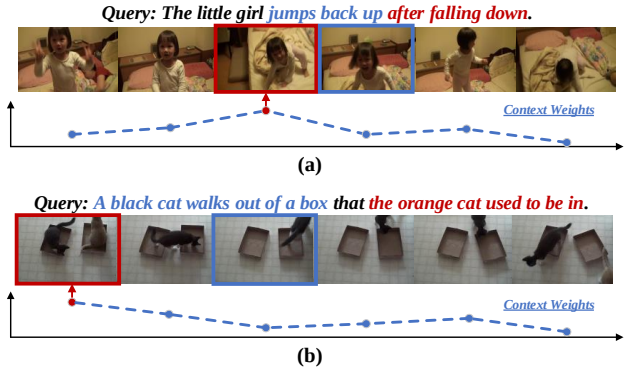

Figure 5: Visualization of the learnt context weights. Ground-truth segments are outlined in blue boxes. Contextual segments corresponding to the highest context weights are outlined in red boxes.

segment in red box, which shows the visual content of "`a box that the orange cat used to be in`". We notice that the best context is not necessarily the nearest segment to the queried segment, as evidenced by Figure 5 (b).

5 Conclusion

In this paper, we proposed a contextual boundary-aware model (CBP) to address the task of temporally grounding language queries in videos. Different from most prior work, CBP was built with a single-stream architecture, which processes a video in one single pass. The idea of boundary prediction is simple yet effective. The promising experimental results obtained on three widely-used datasets demonstrated the effectiveness of our model.

References

Bojanowski, P.; Lajugie, R.; Grave, E.; Bach, F.; Laptev, I.; Ponce, J.; and Schmid, C. 2015. Weakly-supervised alignment of video with text. In *ICCV*.

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Niebles, J. C. 2017. Sst: Single-stream temporal action proposals. In *CVPR*.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Chen, S., and Jiang, Y. 2019. Semantic proposal for activity localization in videos via sentence query. In *AAAI*.

Chen, K.; Kovvuri, R.; Gao, J.; and Nevatia, R. 2017. Msrc: Multimodal spatial regression with semantic context for phrase grounding. In *ICMR*.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018. Temporally grounding natural sentence in video. In *EMNLP*.

Endo, K.; Aono, M.; Nichols, E.; and Funakoshi, K. 2017. An attention-based regression model for grounding textual phrases in images. In *IJCAI*.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Ge, R.; Gao, J.; Chen, K.; and Nevatia, R. 2019. Mac: Mining activity concepts for language-based temporal localization. In *WACV*.

Hahn, M.; Kadav, A.; Rehg, J. M.; and Graf, H. P. 2019. Tripping through time: Efficient localization of activities in videos. *arXiv preprint arXiv:1904.09936*.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2018. Localizing moments in video with temporal language. In *EMNLP*.

Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation*.

Hu, R.; Xu, H.; Rohrbach, M.; Feng, J.; Saenko, K.; and Darrell, T. 2016. Natural language object retrieval. In *CVPR*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-captioning events in videos. In *ICCV*.

Lin, D.; Fidler, S.; Kong, C.; and Urtasun, R. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*.

Liu, B.; Yeung, S.; Chou, E.; Huang, D.-A.; Fei-Fei, L.; and Niebles, J. C. 2018a. Temporal modular networks for retrieving complex compositional activities in videos. In *ECCV*.

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S. 2018b. Attentive moment retrieval in videos. In *SIGIR*.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T.-S. 2018c. Cross-modal moment localization in videos. In *ACMMM*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *TACL*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Rohrbach, A.; Rohrbach, M.; Hu, R.; Darrell, T.; and Schiele, B. 2016. Grounding of textual phrases in images by reconstruction. In *ECCV*.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wang, S., and Jiang, J. 2016. Learning natural language inference with lstm. In *NAACL-HLT*.

Wang, J.; Fu, J.; Xu, Y.; and Mei, T. 2016a. Beyond object recognition: Visual sentiment analysis with deep coupled adjective and noun neural networks. In *IJCAI*.

Wang, M.; Azab, M.; Kojima, N.; Mihalcea, R.; and Deng, J. 2016b. Structured matching for phrase localization. In *ECCV*.

Wang, J.; Jiang, W.; Ma, L.; Liu, W.; and Xu, Y. 2018. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*.

Wang, B.; Ma, L.; Zhang, W.; Jiang, W.; Wang, J.; and Liu, W. 2019. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*.

Wu, A., and Han, Y. 2018. Multi-modal circulant fusion for video-to-language and backward. In *IJCAI*.

Xu, H.; He, K.; Sigal, L.; Sclaroff, S.; and Saenko, K. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *ECCV*.

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L. 2018. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Yuan, Y.; Ma, L.; Wang, J.; Liu, W.; and Zhu, W. 2019. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *NeurIPS*.

Zhang, H.; Niu, Y.; and Chang, S.-F. 2018. Grounding referring expressions in images by variational context. In *CVPR*.