Semantic Proposal for Activity Localization in Videos via Sentence Query

Shaoxiang Chen, Yu-Gang Jiang*
Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University
Shanghai Institute of Intelligent Electronics & Systems
{sxchen13, ygj}@fudan.edu.cn

Abstract

This paper presents an efficient algorithm to tackle temporal localization of activities in videos via sentence queries. The task differs from traditional action localization in three aspects: (1) Activities are combinations of various kinds of actions and may span a long period of time. (2) Sentence queries are not limited to a predefined list of classes. (3) The videos usually contain multiple different activity instances. Traditional proposal-based approaches for action localization that only consider the class-agnostic "actionness" of video snippets are insufficient to tackle this task. We propose a novel Semantic Activity Proposal (SAP) which integrates the semantic information of sentence queries into the proposal generation process to get discriminative activity proposals. Visual and semantic information are jointly utilized for proposal ranking and refinement. We evaluate our algorithm on the TACoS dataset and the Charades-STA dataset. Experimental results show that our algorithm outperforms existing methods on both datasets, and at the same time reduces the number of proposals by a factor of at least 10.

1 Introduction

Recognizing "what's happening" in videos is a crucial task of visual understanding. Recent success of deep learning and computer vision has advanced this task from action classification (Simonyan and Zisserman 2014; Karpathy et al. 2014; Ng et al. 2015; Tran et al. 2015) to detection (Zhao et al. 2017; Yuan et al. 2017; Lin et al. 2018) i.e. temporal localization of actions in videos. Traditional action detection approaches makes one important assumption: the actions to be detected are atomic and in a predefined list (Karpathy et al. 2014; Heilbron et al. 2015; Monfort et al. 2018). Thus these approaches are insufficient to describe and detect the combination of a series of actions.

In this paper, we aim to tackle temporal localization of *activities* in videos via *sentence* queries, which is a more desirable setting. It has three major differences compared to traditional action localization: (1) Activities are more complex than atomic actions like boxing or drinking. The definition of activity we adopt here is actually the same as "high-level event" defined in (Jiang et al. 2013): an activity is composed by several actions or interactions. The latter is more common and is the process of a subject interacting with an object. They may happen in order or co-occur. (2) Sentences are not constrained to a predefined list. They are variable regarding both structure and content, thus can describe various activities. (3) The videos usually contain multiple different activity instances and each may span a long duration. Figure 1 shows an example in which the query sentence is "a person opens a cabinet to place his hat inside ", which describes an activity composed by two interactions involving two objects: cabinet and hat.

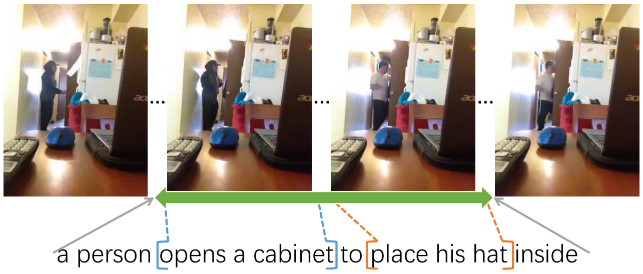


Figure 1: Sample of temporal activity localization via sentence query.

Current approaches to localization problems in computer vision, either spatial or temporal, are mostly based on "proposal and classification": candidate regions are first generated by a separate method, and then fed to a classifier to get the probabilities of containing the target classes. State-of-the-art action proposal generation methods (Zhao et al. 2017; Yuan et al. 2017; Lin et al. 2018) make predictions based on the "actionness" score of every short snippets in the videos. The actionness of a snippet is class-agnostic, just a quantification of the likelihood of containing a generic action instance (Wang et al. 2016). Moreover, the actionness judgment will assign low score to relatively static scenes which may contain objects. In our settings, static objects are as important as actions for localizing activities. Thus existing works of temporal activity localization via sentence queries choose not to use proposal generation methods, but use exhaustive enumeration of possible temporal regions (Hendricks et al. 2017) or naive sliding windows (Gao et al. 2017a). Due to the long duration of videos and activ-

*Corresponding author.

ities, the absence of proper activity proposals will lead to a large number of candidates, and result in inefficient algorithms.

To tackle the challenge of activity proposal generation for sentence query, we propose a novel Semantic Activity Proposal (SAP) framework to integrate the semantic information in sentences into the activity proposal generation process. We first train a visual concept detection CNN with paired sentence-clip training data. The visual concepts are selected from training sentence according to their frequencies. For proposal generation, the visual concepts extracted from the query sentence and video frames are used to compute visual-semantic correlation score for every frame. Activity proposals are generated by grouping frames with high visual-semantic correlation score. Finally, the visual features of video frames, the visual concept vectors and the textual features of sentence queries are utilized to compute a visual-textual alignment score and a refinement of the temporal boundaries for proposals.

Our contributions in this work are as follows:

(1) We propose a novel proposal generation algorithm and framework for temporal activity localization via sentence queries. The proposed Semantic Activity Proposal (SAP), to the best of our knowledge, is the first work to integrate semantic information of sentences into proposal generation.

(2) The proposed framework not only achieves superior localization performance over the state-of-the-art on the TACoS dataset and the Charades-STA dataset, but also reduced the average number of proposals by a factor of at least 10, which is a significant improvement of efficiency.

2 Related Work

Action Classification. There has been a large number of studies about action classification using deep convolutional neural networks (CNNs). (Tran et al. 2015) extend the 2D CNN architecture used in image classification tasks to 3D, which includes temporal dimension and can model short-term motion in video clips. (Simonyan and Zisserman 2014) combine two 2D CNNs which model RGB image and optical flow image to predict the actions in videos. 3D convolution and optical flow are not enough to model long-term motion information in untrimmed videos, thus later works focused on aggregating temporal information. (Karpathy et al. 2014) propose various kinds of temporal information fusion strategies for CNN inputs. (Ng et al. 2015) use the Long Short Term Memory (LSTM) as a feature aggregation technique. (Wang et al. 2017) integrate non-local operation (which can be viewed as a form of attention) into 3D CNN to model relations between consecutive frames. However, these methods deal with trimmed videos or untrimmed videos which contain single action instance. Hence they don't consider the temporal localization of actions.

Action Proposal and Temporal Localization. Temporal action localization methods are based on action proposals, which generates a limited number of candidate temporal regions. A major group of action proposal methods are based on "actionness grouping". (Zhao et al. 2017) train an actionness classifier to evaluate the binary actionness probabilities for individual snippets and then use the proposed temporal actionness grouping (TAG) the generate proposals. Such strategy is also adopted by later works: (Lin, Zhao, and Shou 2017b; Shou et al. 2017; Gao, Chen, and Nevatia 2018). (Yuan et al. 2017; Lin et al. 2018) devise algorithms to compose action proposals based on the probabilities of starting, course, and ending of every time point if the videos. Another group of works first generate anchors of variable length at every temporal position and then evaluate them by predicting 0/1 actionness label or action class label. DAP (Escorcia et al. 2016) and SST (Buch et al. 2017) use an LSTM or GRU unit to process the feature sequence of a video to produce K proposals at each time step. (Gao et al. 2017b) propose to generate and evaluate a clip pyramid at every anchor unit in the frame sequence. (Lin, Zhao, and Shou 2017a) also use convolutional layer to produce anchors hierarchically with different granularities. (Xu, Das, and Saenko 2017) design 3D convolutional network to map a video snippet to predictions of anchor segments.

Video/Image Retrieval with Sentence. Our work is also closely related to video retrieval with sentence, which requires retrieving the videos/images from a set of candidates that match the given sentence query. In (Wang, Li, and Lazebnik 2016), image and text are embedded into the same space via the proposed deep structure-preserving image-text embeddings. In (Karpathy and Li 2015), they embed object regions and words into the same multi-modal space, then region-word pairwise similarities are computed and reduced to image-sentence score for retrieval. To retrieve videos via complex textual queries, (Lin et al. 2014) parse the sentences into semantic graphs and match them to visual concepts in the videos. But the retrieval of whole videos is different from temporal localization in our settings.

Activity Localization with Sentence. In (Hendricks et al. 2017), the authors propose to localize moments in video via natural language with a dataset named DiDeMo. However, the annotated temporal boundaries are coarse since each video is segmented into 5-second segments. They propose a sentence-to-video retrieval method named Moment Contextual Network (MCN) to tackle the localization problem since the number of possible temporal segments are very limited. (Gao et al. 2017a) propose a Cross-modal Temporal Regression Localizer (CTRL), which use dense sliding window to produce activity proposals, then encode visual and textual information with a multi-modal processing network to produce visual-textual alignment score and location regression. But proposals produced by sliding window ignore the relation between temporal regions and the sentence queries. The Attention Based Location Regression (ABLR) in (Yuan, Mei, and Zhu 2018) does not rely on proposals to localize activities. They encode the visual and textual features with Bi-LSTM and directly regress the temporal locations based on the visual-textual co-attention weights. Thus, this method is unable to generate multiple predictions for a sentence query. These existing methods overlooked the im-

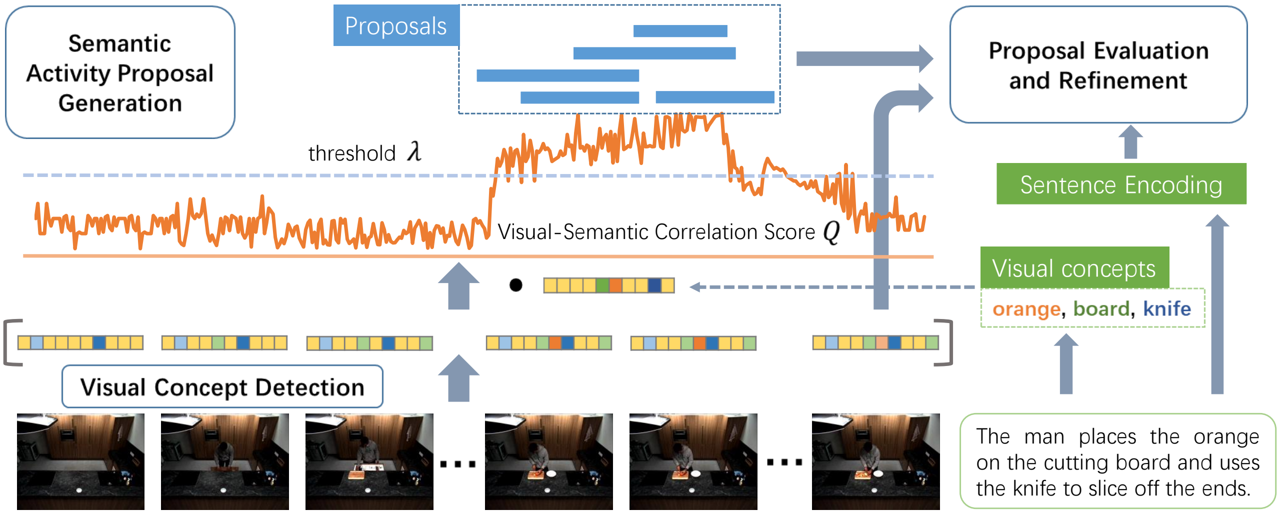

Semantic Activity Proposal Generation
Proposals
Proposal Evaluation and Refinement
threshold λ
Sentence Encoding
Visual-Semantic Correlation Score Q
Visual concepts
orange, board, knife
Visual Concept Detection
The man places the orange on the cutting board and uses the knife to slice off the ends.

Figure 2: The proposed framework for temporal activity localization via sentence query. There are three main components: (1) Visual Concept Detection produces vectors of probabilities of containing common visual concepts for sampled frames. These visual concept vectors dot-product with the one extracted from query sentence, results are the visual-semantic correlation score. (2) Semantic Activity Proposals are generated by temporally grouping frames with high correlation score. (3) Proposal Evaluation and Refinement takes the proposals, visual concept vectors and query sentences as input, and outputs alignment scores and refined boundaries for the proposals.

portance of generating activity proposals, let alone integrating textual information into proposal generation.

3 Approach

3.1 Problem Formulation

A video V is denoted as a sequence of frames: $V = \{f_t\}_{t=1}^{T}$. Each video is associated with a set of temporal annotations $A = \{(d_j, s_j, e_j)_{j=1}^{N}\}$, where N is the number of annotations, d_j is the sentence description, the corresponding clip of video V starts at frame s_j and ends at frame e_j. The task is to predict the start and end time for a given sentence query.

3.2 Framework

Visual Concept Detection. In this work, we define *visual concept* as the visible object and actions in the videos. They are described by the sentences, thus the words in the sentences correspond to visual concepts in the video frames, such as `orange`, `cup`, and `wash`. While there are no spatial bounding box annotations for the words, a visual concept detector can be trained using Multiple Instance Learning (MIL) (Maron and Lozano-Pérez 1997; Viola, Platt, and Zhang 2005; Fang et al. 2015). We first select K most common words (visual concepts) in all the training sentences. Each sentence description d_j can then be converted to a one-hot vector c_j, where c_j^k equals to 1 means word k is in the sentence, and 0 otherwise. Meanwhile, we assume that every frame of the corresponding clip will contain the visual concepts in d_j. Thus we randomly sample a frame f_j from the clip (s_j, e_j) as the input to visual concept detector. We define the visual concept detector as a function $F_{vcd}(f_j)$ that maps an image to a visual concept vector p_j. Inside F_{vcd}, a CNN F_{cnn} is utilized as visual feature extractor, whose input is an image f_j and output is a feature map $\mathcal{M}_j$. $\mathcal{M}_j^{h,w}$ is the feature vector of length m for image region indexed by h, w, which is transformed by a fully-connected layer:

$$\mathcal{P}_j^{h,w} = \texttt{sigmoid}(\mathcal{M}_j^{h,w}\mathbf{W} + \mathbf{b}), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{m\times K}$ and $\mathbf{b} \in \mathbb{R}^{K}$ are trainable parameters. $\mathcal{P}_j^{h,w}$ is then the word probability vector of image region indexed by h, w. We use the noisy-OR version of MIL (Viola, Platt, and Zhang 2005) to compute the final probability for the whole image:

$$p_j = 1 - \prod_{h,w}(1 - \mathcal{P}_j^{h,w}), \tag{2}$$

where p_j is a vector of length K and p_j^k stands for the probability of word k appearing in frame f_j. We denote p_j as the visual concept vector for frame f_j. Equation 1 and 2 conclude the details of the visual concept detector F_{vcd}. To learn the parameters of F_{vcd}, we adopt the cross-entropy loss:

$$Loss_{vcd} = -\sum_{k=1}^{K} c_j^k \log p_j^k. \tag{3}$$

Semantic Activity Proposal. With the visual concept extractor F_{vcd}, we can obtain visual concept vectors for each frame of video V: $P = \{p_t\}_{t=1}^{T}$, where $p_t = F_{vcd}(f_t)$. Then the visual-semantic correlation scores between the query

Algorithm 1 Semantic Activity Proposal Generation

```
function SEMANTIC ACTIVITY PROPOSAL($Q$, $\lambda$, $\tau$)
    $R \leftarrow \emptyset$                                  ▷ Grouped temporal regions
    $G \leftarrow \emptyset$                                  ▷ Generated proposals
    end ← True
    for $t$ = 1 to $T$ do
        if end then
            if $Q_i \geq \lambda$ then                        ▷ Start a new temporal group
                $s \leftarrow t$
                end ← False
        else
            $r \leftarrow$ ratio of $Q_i \geq \lambda$ in $[s, t]$     ▷ Get positive ratio
            if $r < \tau$ then                              ▷ Positive ratio under tolerance
                end ← True
                add $[s, t]$ to $R$                          ▷ End current temporal group
    for $s, t$ in $R$ do
        for $L$ in proposalLengths do                       ▷ List of lengths
            propL ← sliding windows of $L$ in $[s, t]$
            add propL to $G$
    return $G$
```

sentence d_j and frames can be represented as $Q_j = \{q_j^t\}_{t=1}^T$, where

$$q_j^t = p_t \cdot c_j. \tag{4}$$

q_j^t stands for the total probabilities of frame t containing all the visual concepts in query j. Q_j is then normalized to $[0, 1]$. As shown in Figure 2, frames with more visual concepts described in d_j tend to get higher correlation score. Frames with score above a threshold λ are considered positive, i.e. related to the sentence query. We use a binary-search algorithm to determine λ, such that the ratio of positive frames does not exceed 0.06. This value is chosen to make a balance between the number of generated proposals and the recall, and is decided on the validation set. When the positive frames are selected, we adopt an algorithm similar to (Zhao et al. 2017) to group the positive frames into consecutive temporal regions. To account for false negative frames, an extra parameter τ is introduced as the tolerance ratio, which controls the ratio of negative frames allowed in a temporal region. The activity proposals are finally generated with a predefined length inside the grouped temporal regions using sliding window. Algorithm 1 shows the details of the temporal grouping process.

Proposal Evaluation and Refinement. The details of proposal evaluation and refinement are shown in Figure 3. For a specific query (d_j, s_j, e_j), we denote the generated proposals by $G_j = \{(l_n, r_n)\}_{n=1}^{N_j}$, where N_j is the number of proposals and l_n, r_n are the temporal boundaries. G_j will be evaluated to produce alignment scores, which are then used to rank the proposals. Since the generated proposals have fixed lengths, their boundaries will be further refined to localize the activities more precisely. First, the visual feature vectors and concept vectors of the frames inside proposal region are extracted from a pre-trained CNN, denoted by $\mathbf{f}_v$ and $\mathbf{f}_c$, respectively. Next, these vectors are aggregated as a single feature vector. For visual features, we adopt the trainable VLAD encoding (Miech, Laptev, and Sivic 2017).

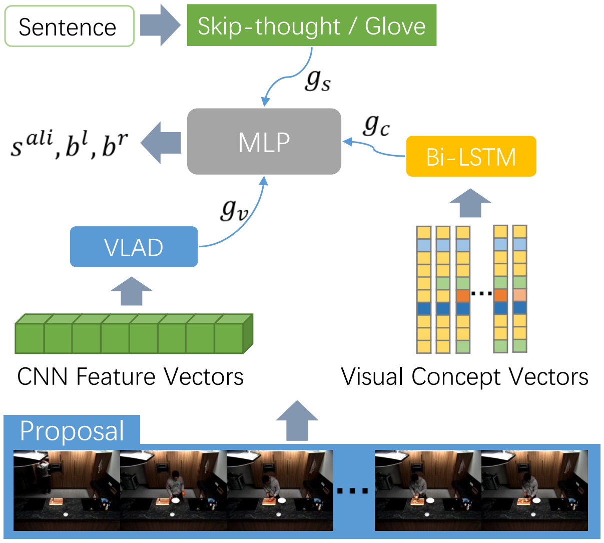


Figure 3: The proposal evaluation and refinement process.

For visual concept vectors, since the order of the sequence is important, we use a bi-directional LSTM to encode the sequence and concatenate the final state vectors of both directions. The feature aggregation is summarized as follows:

$$\begin{aligned} g_v &= \mathtt{VLAD}(\mathbf{f}_v), \\ g_c &= [\mathtt{LSTM_{fw}}(\mathbf{f}_c), \mathtt{LSTM_{bw}}(\mathbf{f}_c)]. \end{aligned} \tag{5}$$

For the query sentence, we experiment with two kinds of off-the-shelf sentence encoding method: Skip-thought (Kiros et al. 2015) and Glove (Pennington, Socher, and Manning 2014). Details can be found in Sec. 4.2. The encoded sentence feature vector is denoted as g_s.

Then the alignment score and boundary refinement is computed as:

$$\begin{aligned} s^{ali} &= \mathtt{MLP}(g_s \otimes g_v, 1), \\ b^l, b^r &= \mathtt{MLP}(g_s \otimes g_c, 2), \end{aligned} \tag{6}$$

where $\otimes$ is element-wise product, and $\mathtt{MLP}(, u)$ is a multi-layer perceptron whose final layer has u outputs. b^s and b^e are the predicted offset for the start and end points of the proposal. During training, we compute alignment scores between all the sentence-proposal pairs in a mini-batch, and encourage our model to output low scores for negative pairs and high scores for positive pairs. Thus the alignment loss is defined as:

$$\begin{aligned} Loss_{ali} = \frac{1}{B} \sum_{i=1}^{B} [\log(1 + \exp(-s_{i,i}^{ali})) + \\ \sum_{j=1, j \neq i}^{B} \alpha \log(1 + \exp(s_{i,j}^{ali}))], \end{aligned} \tag{7}$$

where α is a hyper-parameter to balance the loss of positive and negative pairs. The boundary refinement loss is defined as:

$$Loss_{ref} = \sum_{i=1}^{B} [H(b_i^l - (l_i - s_i)) + \\ H(b_i^r - (r_i - e_i))], \tag{8}$$

where l_i and s_i are the proposal and annotated starting points, likewise for r_i and e_i. $H()$ is the Huber loss function. The final loss for training the proposal evaluation module is:

$$Loss = Loss_{ali} + \beta Loss_{ref}, \tag{9}$$

where β is a hyper-parameter to balance the alignment and refinement loss.

4 Experiments

4.1 Datasets

TACoS (Regneri et al. 2013). The TACoS dataset is built on the MPII Cooking Composite Activities (Rohrbach et al. 2012b; 2012a), which contains fine-grained temporal annotations of cooking activities. There are 127 videos in the dataset. Following previous work, we split the dataset into training, validation and test sets with 75, 27 and 25 videos, respectively. Each annotation contains one sentence and the start and end time of the activity it describes in the video. The numbers of annotations in training, validation and test sets are 10146, 4589 and 4083, respectively. The average length of the sentences is 6.2 words, the average duration of the videos is 287.1 seconds, and the average number of activities per video is 21.4.

Charades-STA (Gao et al. 2017a). The Charades-STA dataset is built on the Charades (Sigurdsson et al. 2016) dataset, which contains 9848 videos of daily indoors activities collected through Amazon Mechanical Turk. There are 16128 clip-sentence pairs in the released Charades-STA dataset, which are split into training and test sets of 12408 and 3720 clip-sentence pairs, respectively. The average length of the sentences is 8.6 words, the average duration of the videos is 29.8 seconds, and the average number of activities per video is 2.3.

4.2 Implementation

For training the visual concept detector, we collect common visual concepts on both datasets. Concretely, we count the words of training sentences, discard stop words and keep words whose occurrence are at least 2 as the visual concepts. This results in 912 and 566 visual concepts on the TACoS dataset and Charades-STA dataset, respectively. We use the VGG16 network pre-trained on ImageNet as the backbone of our visual concept detector. We discard its layers after `fc6` and use the rest as the feature extractor. For each annotated temporal region, we uniformly sample one frame and resize it to 512x512 pixels as the input at every training step. We use the Momentum algorithm with a learning rate of 10^{-5} and batch size of 16 to train the visual concept detector.

In the proposal evaluation module, the visual feature is extracted from the visual concept detector's `fc6` layer. The number of clusters for VLAD is 64 and the number of units for LSTM is 1024. The Skip-thought encoding produces one vector of length 4800 for each sentence. The Glove encoding maps each word to a vector of length 300, and we further encode the sequence using an LSTM with 1024 units. The hyper-parameters in the losses, α and β are 0.015 and 0.01, respectively. During training, the proposals are generated by dense sliding window method. For each annotation, we generate sliding windows of length $[64, 128, 256, 512]$ frames for the video to cover the annotated temporal region. Only windows having temporal IoU$\geq$ 0.5 are used for training. Each mini-batch is sampled such that there does not exist any pair of sentences that describes the same clip, this ensures there is only one positive sentence for each proposal in the batch and $Loss_{ali}$ is correctly computed. The final loss is optimized by the Adam algorithm with a learning rate of 10^{-4} and batch size of 64. For evaluation, the generated proposal lengths are in $[128, 256]$ (decided based on the statistics of the datasets).

4.3 Evaluation Metrics

As in previous work (Gao et al. 2017a), we measure the performance of temporal localization by average recall rate of top-n results at certain temporal IoU (Intersection over Union), which is the "R@n, IoU=m" in Table 1 and 2, shown in percentage. The recall of one sentence query d_j, $r(n, m, d_j)$, is 1 if the top-n returned results contains at least one that has a temporal IoU$\geq$ m, otherwise 0. The average recall rate is the average over all the queries: $R(n, m) = \frac{1}{N}\sum_{j=1}^{N} r(n, m, d_j)$.

4.4 Compared Methods

- **Random.** We generate activity proposals by sparse sliding windows with [128, 256] frames and 20% stride, then randomly select temporal regions from proposals.
- **SST.** (Buch et al. 2017) as mentioned in Sec. 2. The original SST method generates dense proposals with various lengths as each time step. In our experiments, we train SST with dense proposal lengths. For evaluation, the proposals are [128, 256] frames and post-processed by non-maximum suppression.
- **CTRL.** (Gao et al. 2017a) as mentioned in Sec. 2.
- **MCN.** (Hendricks et al. 2017) as mentioned in Sec. 2. The original MCN enumerates all possible temporal regions as candidates, but this is impractical for our settings. We use the same proposal generation algorithm as CTRL for MCN.
- **ABLR.** (Yuan, Mei, and Zhu 2018) as mentioned in Sec. 2. We implemented ABLR and tested on the datasets. Note that this method can't produce Recall@5 results.

4.5 Results

Results on the TACoS dataset. Table 1 shows the recall of top $\{1, 5\}$ results at IoU threshold $\{0.1, 0.5\}$ of different methods on the TACoS dataset. It is clear that traditional action proposal method SST doesn't work well under this setting. The reasons are mainly twofold: (1) the proposals generate by SST are not aware of the specific activ-

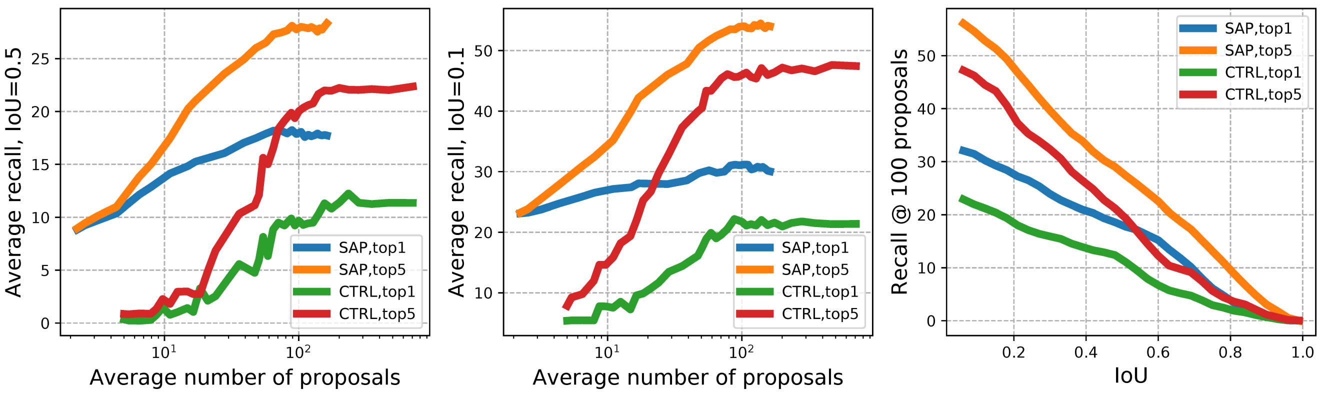


Figure 4: Comparison of our SAP with existing methods on the TACoS dataset. (**left & center**) SAP outperforms other methods on every metric and requires much less proposals. And SAP reaches peak performance at around 100 proposals. (**right**) When the number of proposals is fixed to 100, SAP also consistently has the highest recall.

Method	R@1	R@1	R@5	R@5
	IoU=0.5	IoU=0.1	IoU=0.5	IoU=0.1
Random	0.71	3.28	3.72	15.47
SST	0.97	3.46	4.57	14.54
CTRL	13.30	24.32	25.42	48.73
MCN	5.58	14.42	10.33	37.35
ABLR	9.4	**31.4**	-	-
SAP$_{glove}$	16.62	29.24	27.01	52.50
SAP$_{noref}$	14.45	29.51	23.78	52.09
SAP$_{sv}$	**18.24**	31.15	**28.11**	**53.51**

Table 1: Comparison of different methods on TACoS.

Method	R@1	R@1	R@5	R@5
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Random	8.51	3.03	37.12	14.06
SST	15.98	8.31	40.68	27.24
CTRL	23.63	8.89	58.92	29.52
MCN	17.46	8.01	48.22	26.73
ABLR	24.36	9.01	-	-
SAP$_{glove}$	26.96	12.36	63.20	35.83
SAP$_{sv}$	**27.42**	**13.36**	**66.37**	**38.15**

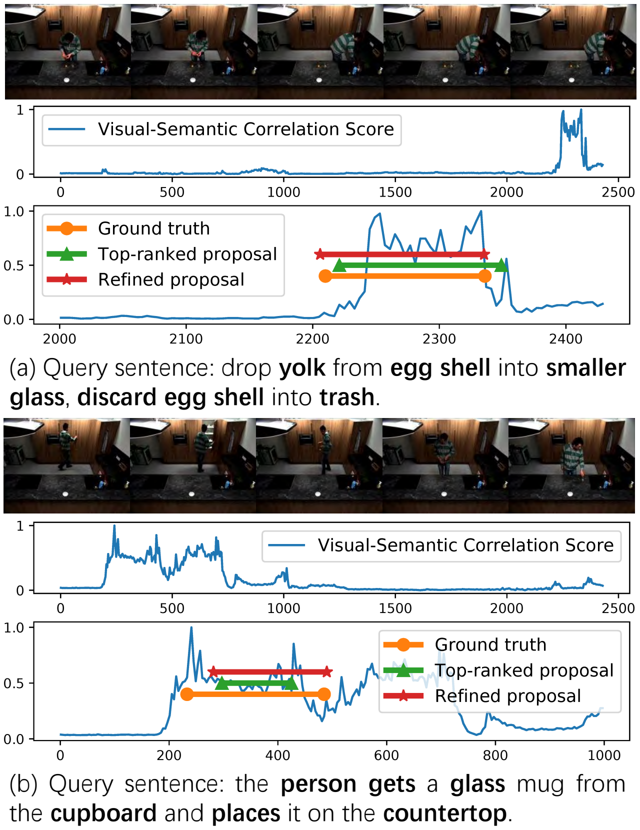
Table 2: Comparison of different methods on Charades-STA.

ity described in the query sentence; (2) the proposals are not ranked according to their correlation to the query sentence. Since the videos in the TACoS dataset have long durations and contain multiple activities, methods that do not consider sentence information in proposal generation will suffer from a large number of proposals. CTRL and MCN use naive proposal generation algorithm, and also have this problem. They integrate sentence information only in the proposal evaluation and ranking process, which still leads to inferior performance. ABLR discards proposal generation. However, the attention based approach may suffer from low accuracy at the boundaries, which we conjecture is the reason why ABLR gets lower recall at higher IoU threshold. The effectiveness of proposal refinement is demonstrated by ablation (SAP$_{noref}$), it is clear that adding proposal refinement leads to better localization performance. We also found that Skip-thought vectors (SAP$_{sv}$) performs consistently better than Glove embeddings (SAP$_{glove}$). We hypothesize the reason is that the number of training sentences is not large enough to train the encoding LSTM for Glove embeddings. Overall, the proposed method outperforms others by a significant margin. Notably, on the most important metric "R@1,IoU=0.5", SAP outperforms the best competitor CTRL by 37%.

Results on the Charades-STA dataset. Table 2 shows the recall of top {1, 5} results at IoU threshold {0.5, 0.7} of different methods on the Charades-STA dataset. We choose a higher IoU threshold on this dataset because the videos are shorter and the number of activities per video is less compared to TACoS. For this reason, SST achieves higher performance on this dataset, which in turn indicates the importance of discriminative proposals on datasets of long videos. And it is also observed that there isn't a large difference between SAP and other methods regarding the number of generated proposals. Thus the advantage of SAP can be attributed to the proposal evaluation and refinement process. Overall, the proposed SAP consistently outperforms other methods on this dataset. On the most important metric "R@1,IoU=0.7", SAP outperforms the best competitor ABLR by 48.3%.

4.6 Efficiency of Proposal Generation

Considering the video duration and number of activities in a video, a successful proposal method should be able to achieve high recall rate with *a small number of proposals*. We evaluate this with two measurements: average number of proposals and average recall for a fixed number of proposals. Figure 4(left and center) shows the advantage of SAP over CTRL is significant both at high and low IoU threshold. Notably, for IoU=0.5, SAP only needs around 20 proposals to achieve CTRL's peak top5 recall rate, which CTRL takes around 200 proposals. The result is likewise



(a) Query sentence: drop **yolk** from **egg shell** into **smaller glass**, **discard egg shell** into **trash**.

(b) Query sentence: the **person gets** a **glass** mug from the **cupboard** and **places** it on the **countertop**.

Figure 5: Qualitative results. The words shown in bold are defined as visual concepts. It can be observed that the visual-semantic correlation score is a good indication of the temporal region of the activity even in a long video.

for IoU=0.1. This demonstrates the high efficiency of SAP. Figure 4(right) plots the average recall rate for 100 proposals for SAP and CTRL. The advantage of SAP is again significant, it outperforms CTRL at every IoU region. Table 3 shows the comparison of time consumption for proposal generation and evaluation per query. Note that SST doesnt do proposal evaluation and ABLR generates single prediction without proposals, thus they are faster. The advantage of having less proposals can be seen from the comparison among SAP(ours), CTRL and MCN. Overall, these show that by integrating semantic information for activity proposal generation, SAP can produce a small number of discriminative proposals for faster evaluation and achieve high localization accuracy.

4.7 Effect of Visual Concepts

To demonstrate the effect of visual concepts, we conduct experiments on the TACoS dataset with various numbers of visual concepts. Table 4 shows the results. It can be observed that with a small number of concepts, the model is likely to lose some semantic information during proposal generation. Thus, as the number of concepts increase (from 93 to 912), the performance will continue to improve. But an even larger number of concepts (e.g. 1413) will introduce noise into the model and hurt the performance.

Method	SAP(ours)	CTRL	MCN	SST	ABLR
Time	0.35s	1.76s	0.88s	0.33s	0.01s

Table 3: Comparison of time consumption for proposal generation and evaluation per query.

#Concepts	R@1	R@1	R@5	R@5
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
1413	18.13	30.14	27.28	52.09
912	18.24	31.15	28.11	53.51
397	18.05	31.66	27.50	54.17
233	17.31	29.83	27.09	53.09
93	16.60	27.50	25.12	49.74

Table 4: Performances with different number of visual concepts on TACoS.

4.8 Qualitative Results

To gain an intuition about the effectiveness of the Semantic Activity Proposal, we present some qualitative results in Figure 5. On the TACoS dataset, each video contains over 20 different activities which could span the duration of the whole video. It can be observed that the visual-semantic correlation scores are high around the ground truth regions and low for other regions. Thus our SAP can generate a small number of proposals for a long video while having high localization accuracy. Furthermore, the boundaries of the proposals can be refined to more accurately localize the activities as shown in the bottom of Figure 5(a,b).

5 Conclusions

We have introduced a novel framework for activity localization in videos via sentence query, including an efficient activity proposal generation algorithm named Semantic Activity Proposal (SAP). We evaluate both the localization accuracy and number of proposals of our framework on the TACoS and Charades-STA dataset. Experiments show that our proposed framework outperforms existing methods with a significant margin, and at the same time reduces the number of needed proposals by a factor of at least 10. Our future work will consider analyzing the sentence structure to discover more discriminative visual concepts.

Acknowledgements. This work was supported in part by two projects from NSF China (61622204, 61572134) and one project from STCSM, China (16QA1400500).

References

Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; and Niebles, J. C. 2017. SST: single-stream temporal action proposals. In *CVPR*, 6373–6382.

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B. 2016. Daps: Deep action proposals for action understanding. In *ECCV*, 768–784.

Fang, H.; Gupta, S.; Iandola, F. N.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; Zitnick, C. L.; and Zweig, G. 2015. From captions to visual concepts and back. In *CVPR*, 1473–1482.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017a. TALL: temporal activity localization via language query. In *ICCV*, 5277–5285.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R. 2017b. TURN TAP: temporal unit regression network for temporal action proposals. In *ICCV*, 3648–3656.

Gao, J.; Chen, K.; and Nevatia, R. 2018. CTAP: complementary temporal action proposal generation. In *ECCV*, 70–85.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*, 5804–5813.

Jiang, Y.; Bhattacharya, S.; Chang, S.; and Shah, M. 2013. High-level event recognition in unconstrained videos. *IJMIR* 2(2):73–101.

Karpathy, A., and Li, F. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 3128–3137.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*, 1725–1732.

Kiros, R.; Zhu, Y.; Salakhutdinov, R.; Zemel, R. S.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*, 3294–3302.

Lin, D.; Fidler, S.; Kong, C.; and Urtasun, R. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2657–2664.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: boundary sensitive network for temporal action proposal generation. In *ECCV*, 3–21.

Lin, T.; Zhao, X.; and Shou, Z. 2017a. Single shot temporal action detection. In *ACM MM*, 988–996.

Lin, T.; Zhao, X.; and Shou, Z. 2017b. Temporal convolution based action proposal: Submission to activitynet 2017. *CoRR* abs/1707.06750.

Maron, O., and Lozano-Pérez, T. 1997. A framework for multiple-instance learning. In *NIPS*, 570–576.

Miech, A.; Laptev, I.; and Sivic, J. 2017. Learnable pooling with context gating for video classification. *CoRR* abs/1706.06905.

Monfort, M.; Zhou, B.; Bargal, S. A.; Andonian, A.; Yan, T.; Ramakrishnan, K.; Brown, L. M.; Fan, Q.; Gutfreund, D.; Vondrick, C.; and Oliva, A. 2018. Moments in time dataset: one million videos for event understanding. *CoRR* abs/1801.03150.

Ng, J. Y.; Hausknecht, M. J.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; and Toderici, G. 2015. Beyond short snippets: Deep networks for video classification. In *CVPR*, 4694–4702.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *TACL* 1:25–36.

Rohrbach, M.; Amin, S.; Andriluka, M.; and Schiele, B. 2012a. A database for fine grained activity detection of cooking activities. In *CVPR*, 1194–1201.

Rohrbach, M.; Regneri, M.; Andriluka, M.; Amin, S.; Pinkal, M.; and Schiele, B. 2012b. Script data for attribute-based recognition of composite activities. In *ECCV*, 144–157.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S. 2017. CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 1417–1426.

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*, 510–526.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 568–576.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 4489–4497.

Viola, P. A.; Platt, J. C.; and Zhang, C. 2005. Multiple instance boosting for object detection. In *NIPS*, 1417–1424.

Wang, L.; Qiao, Y.; Tang, X.; and Gool, L. J. V. 2016. Actionness estimation using hybrid fully convolutional networks. In *CVPR*, 2708–2717.

Wang, X.; Girshick, R. B.; Gupta, A.; and He, K. 2017. Non-local neural networks. *CoRR* abs/1711.07971.

Wang, L.; Li, Y.; and Lazebnik, S. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*, 5005–5013.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: region convolutional 3d network for temporal activity detection. In *ICCV*, 5794–5803.

Yuan, Z.; Stroud, J. C.; Lu, T.; and Deng, J. 2017. Temporal action localization by structured maximal sums. In *CVPR*, 3215–3223.

Yuan, Y.; Mei, T.; and Zhu, W. 2018. To find where you talk: Temporal sentence localization in video with attention based location regression. *CoRR* abs/1804.07014.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*, 2933–2942.