Temporally Grounding Natural Sentence in Video

Jingyuan Chen[1*] **Xinpeng Chen**[2*] **Lin Ma**[2] **Zequn Jie**[2] **Tat-Seng Chua**[1]
[1]National University of Singapore [2]Tencent AI Lab
{jingyuanchen91, jschenxinpeng, forest.linma, zequn.nus}@gmail.com, dcscts@nus.edu.sg

Abstract

We introduce an effective and efficient method that grounds (*i.e.*, localizes) natural sentences in long, untrimmed video sequences. Specifically, a novel Temporal GroundNet (TGN)[1] is proposed to temporally capture the evolving fine-grained frame-by-word interactions between video and sentence. TGN sequentially scores a set of temporal candidates ended at each frame based on the exploited frame-by-word interactions, and finally grounds the segment corresponding to the sentence. Unlike traditional methods treating the overlapping segments separately in a sliding window fashion, TGN aggregates the historical information and generates the final grounding result in one single pass. We extensively evaluate our proposed TGN on three public datasets with significant improvements over the state-of-the-arts. We further show the consistent effectiveness and efficiency of TGN through an ablation study and a runtime test.

1 Introduction

We examine the task of Natural Sentence Grounding in Video (NSGV). Given an untrimmed video and a natural sentence, the goal is to determine the start and end timestamps of the segment in the video which corresponds to the given sentence, as shown in Figure 1 (a). Comparing with the other video researches, such as bidirectional video-sentence retrieval (Xu et al., 2015b), video attractiveness prediction (Chen et al., 2018, 2016), and video captioning (Pasunuru and Bansal, 2017; Wang et al., 2018a,b), NSGV needs to model not only the characteristics of sentence and video but also the fine-grained interactions between the two modalities, which is even more challenging.

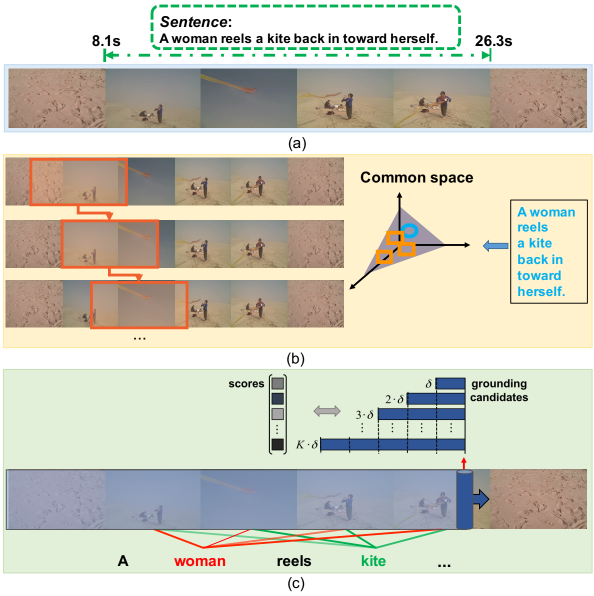


Figure 1: (a) The Natural Sentence Grounding in Video (NSGV) task. (b) A common space based matching method performs in a sliding window fashion. (c) Our proposed Temporal GroundNet (TGN) localizes the candidate video segments at multiple scales in a single processing pass. The frames in the video and the words in the sentence interact attentively to perform fine-grained frame-by-word matchings for grounding sentence in video.

Recently, several related works (Gao et al., 2017; Hendricks et al., 2017) leverage one temporal sliding window approach over video sequences to generate video segment candidates, which are then independently combined (Gao et al., 2017) or compared (Hendricks et al., 2017) with the given sentence to make the grounding prediction. Although the existing works have achieved promising performances, they are still suffering from inferior effectiveness and efficiency. First, existing methods project the video segment and sentence into one common space, as shown in Figure 1 (b),

[*] Work done while Jingyuan Chen and Xinpeng Chen were Research Interns with Tencent AI Lab.

[1] The project homepage is https://jingyuanchen.github.io/archive/tgn.html.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 162–171
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

where the two generated embedding vectors are used to perform the matching between video segment and sentence. Such a matching is only performed in the global segment and sentence level and thus not expressive enough, which ignores the fine-grained matching relations between video frames and the words in sentences. Second, in order to handle the diverse temporal scales and locations of the candidate segments, exhaustive matching between the large amount of overlapping segments and the sentence is required. As such, the sliding window methods are very computationally expensive.

In order to tackle the above two limitations, we introduce a novel Temporal GroundNet (TGN) model, the first dynamic single-stream deep architecture for the NSGV task that takes full advantage of fine-grained interactions between video frames and words in a sentence, as shown in Figure 1 (c). TGN sequentially processes video frames, where at each time step we rely on a novel multimodal interactor to exploit the evolving fine-grained frame-by-word interactions. Then, TGN works on the yielded interaction status to simultaneously score a set of temporal candidates of multiple scales and finally localize the video segment that corresponds to the sentence. More importantly, our proposed TGN is able to analyze an untrimmed video frame by frame without resorting to handling overlapping temporal video segments.

2 Related Work

2.1 Grounding Natural Language in Image

Grounding natural language in image is also known as natural language object retrieval. The task is to localize an image region described by natural language, which involves comprehending and modeling different spatial contexts, such as spatial configurations (Hu et al., 2016), attributes (Yu et al., 2018), and relationships between objects (Hu et al., 2017). Specifically, the task is usually formulated as a ranking problem over a set of candidate regions in a given image, where candidate spatial locations come from region proposal methods (Uijlings et al., 2013; Jie et al., 2016b,a; Ren et al., 2017) such as EdgeBox (Zitnick and Dollár, 2014). Earlier studies (Mao et al., 2016; Rohrbach et al., 2016) score the generated candidate regions according to their appearances and spatial features along with features of the entire image. However, these methods fail to incorporate the interactions between objects, because the scoring process of each region proposal is isolated. More recent studies (Hu et al., 2017; Nagaraja et al., 2016) improve the performance with the aid of modeling relationships between objects.

2.2 Grounding Natural Language in Video

Analogous to spatial grounding in image, this work studies a similar problem—temporal natural language grounding in video. Earlier works (Yu and Siskind, 2013; Lin et al., 2014) learn the semantics of sentences, which are then matched to visual concepts via exploiting object appearance, motion and spatial relationships. However, they are limited to a small set of objects. Recently, larger datasets (Gao et al., 2017; Hendricks et al., 2017) are constructed to support more flexible groundings. The methods proposed in (Gao et al., 2017; Hendricks et al., 2017) learn a common embedding space shared by video segment features and sentence representations, in which their similarities are measured. Specifically, moment context network (MCN) (Hendricks et al., 2017) learns a shared embedding for video clip-level features and language features. The video features integrate local video features, global features, and temporal endpoint features. Cross-modal temporal regression localizer (CTRL) (Gao et al., 2017) contains four modules, specifically a visual encoder extracting clip-level features with context, a sentence encoder yielding its embedding through LSTM, a multimodal processing network generating the fused representations via element-wise operations, and a temporal regression network producing the alignment scores and location offsets. One limitation of those common space matching methods is that the video segment generation process is computationally expensive, as they carry out overlapping sliding window matching (Gao et al., 2017) or exhaustive search (Hendricks et al., 2017). Another weakness is that they exploit the relationships between textual and visual modalities by conducting a simple concatenation (Gao et al., 2017) or measuring a squared distance loss (Hendricks et al., 2017), which ignores the evolving fine-grained video-sentence interactions. In this paper, a novel model TGN is proposed to deal with the aforementioned limitations for the task of natural sentence grounding in video.

3 Approach

Given a long and untrimmed video sequence V and a natural sentence S, the NSGV task is to localize a video segment $V_s = \{f_t\}_{t=t_b}^{t_e}$ from V, beginning at t_b and ending at t_e, which corresponds to and expresses the same semantic meaning as the given sentence S. In order to perform the grounding, each video is represented as $V = \{f_t\}_{t=1}^{T}$, where T is the total number of frames and f_t denotes the feature representation of the t-th video frame. Similarly, each sentence is represented as $S = \{w_n\}_{n=1}^{N}$, where w_n is the embedding vector of the n-th word in the sentence and N denotes the total number of words.

We propose a novel model, namely Temporal GroundNet (TGN), to tackle the NSGV problem. As illustrated in Figure 2, TGN consists of three modules. 1) Encoder: visual and textual encoders are used to compose the video frame representations and word embeddings, respectively. 2) Interactor: a multimodal interactor learns the frame-by-word interactions between the video and sentence. 3) Grounder: a grounder generates the temporal localization in one single pass. Please note that these three modules are fully coupled together, which can thus be trained in an end-to-end fashion.

3.1 Encoder

With the obtained video frame features $V = \{f_t\}_{t=1}^{T}$ and word embeddings of the sentence $S = \{w_n\}_{n=1}^{N}$, we employ two long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) to sequentially process the two different modalities, *i.e.*, video and sentence, independently. Specifically, one LSTM sequentially models the video V, yielding the hidden states $\{\mathbf{h}_t^v\}_{t=1}^{T}$, while the other LSTM processes the sequential words in the sentence S, resulting in its corresponding hidden states $\{\mathbf{h}_n^s\}_{n=1}^{N}$. Owing to natural behaviors and characteristics of LSTMs, both $\{\mathbf{h}_t^v\}_{t=1}^{T}$ and $\{\mathbf{h}_n^s\}_{n=1}^{N}$ can encode and aggregate the contextual evidences (Wang and Jiang, 2016b) from the sequential video frame representations and word embeddings of the sentence, respectively, meanwhile casting aside the irrelevant information.

3.2 Interactor

Based on the hidden states of the video and sentence yielded from the leveraged encoders, we design a multimodal interactor to perform the frame-by-word interactions between the video and sentence. First, the frame-specific sentence feature is generated through summarizing the sentence hidden states by considering their relationships with the specific video frame at each time step. Afterwards, an interaction LSTM, dubbed i-LSTM, is performed to aggregate frame-by-word interactions.

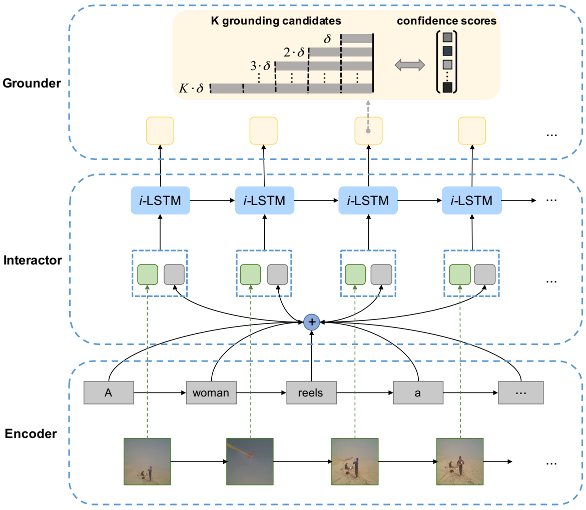


Figure 2: The architecture of our proposed TGN model. TGN consists of three modules. The visual and textural encoders aggregate the contextual evidences from the sequential video frame representations and word embeddings of the sentence, respectively. The multimodal interactor learns the fine-grained frame-by-word interactions between the video and sentence. The grounder yields the temporal grounding of the sentence in the video sequence via one single pass.

3.2.1 Frame-Specific Sentence Feature

Directly operating on the clip-level and sentence-level features generated by the encoders cannot well exploit the frame-by-word relationships between video and sentence that evolve over time. Inspired by (Wang and Jiang, 2016a; Feng et al., 2018), we introduce one novel frame-specific sentence feature, which adaptively summarizes the hidden states of the sentence $\{\mathbf{h}_n^s\}_{n=1}^{N}$ with respect to the t-th video frame:

$$\mathbf{H}_t^s = \sum_{n=1}^{N} \alpha_t^n \mathbf{h}_n^s, \tag{1}$$

where $\mathbf{H}_t^s$ denotes the summarized sentence representation specified by the t-th video frame. At each time step t, we utilize the hidden state $\mathbf{h}_t^v$ to selectively attend the words and summarize them accordingly. The attention weight α_t^n encodes the degree to which the n-th word in the sentence is aligned with the t-th video frame. As the processing of video frames proceeds, the attention weights dynamically change regarding to the current video frame. As such, the generated frame-specific sentence features $\{\mathbf{H}_t^s\}_{t=1}^T$ consider the frame-by-word relationships between all the video frames and all the words in the sentence.

As the generation of frame-specific sentence feature is deeply coupled with the following interaction LSTM, we will explain the calculation of the attention weight α_t^n later.

3.2.2 Interaction LSTM (i-LSTM)

In order to accurately ground the sentence in a video, the multimodal interation behaviors between the video and sentence need to be comprehensively modeled. Previous approaches on multimodal interactions were limited to concatenation (Zhu et al., 2016), element-wise product or sum (Gao et al., 2017), and bilinear pooling (Fukui et al., 2016). These methods are not expressive enough since they ignore the evolving fine-grained interactions across video and sentence, particularly the frame-by-word interactions. In this paper, we propose a novel multimodal interaction model, which is realized by LSTM. We term it interaction LSTM (i-LSTM), which sequentially processes the video sequence frame by frame, holding deep interactions with the words in the sentence.

In order to well capture the complicated temporal interactions between the video and sentence, at each time step t, the input of the i-LSTM is formed by concatenating the t-th video hidden state $\mathbf{h}_t^v$ and the t-th frame-specific sentence feature $\mathbf{H}_t^s$ as: $\mathbf{r}_t = \mathbf{h}_t^v \parallel \mathbf{H}_t^s$. $\mathbf{r}_t$ is then fed into the i-LSTM unit to yield the t-th intermediate interaction status between the video and sentence:

$$\mathbf{h}_t^r = i\text{-LSTM}(\mathbf{r}_t, \mathbf{h}_{t-1}^r), \tag{2}$$

where $\mathbf{h}_t^r$ is the yielded hidden state, encoding the fine-grained interactions between the word and video frame. $\mathbf{h}_t^r$ will be further used to perform the grounding process. Due to the inherent properties and characteristics of LSTMs, important cues regarding to grounding up to the current stage will be "remembered", while non-essential ones will be "forgotten".

Now we go back to the generation of attention weight α_t^n in Eq. (1), based on the obtained visual hidden states h_t^v and textual hidden state h_n^s as well as the yielded interaction status h_{t-1}^r in the previous step. The widely used soft-attention mechanism (Xu et al., 2015a; Chen et al., 2017) is used to generate the attention weights in a frame-by-word manner. As aforementioned, the i-LSTM models the evolving frame-by-word interactions between the sentence and video. Therefore, the attention weight between the n-th word $\mathbf{h}_n^s$ and the t-th video frame $\mathbf{h}_t^v$ is determined by not only the content of the video and sentence but also their interaction status. Thus, we design one network to compute the relevance score of one video frame with respect to each word:

$$\beta_t^n = \mathbf{w}^\intercal \tanh(\mathbf{W}^S\mathbf{h}_n^s + \mathbf{W}^V\mathbf{h}_t^v + \mathbf{W}^R\mathbf{h}_{t-1}^r + \mathbf{b}) + c, \tag{3}$$

where vector $\mathbf{w}$, matrices $\mathbf{W}^*$, bias vector $\mathbf{b}$, and bias c are the network parameters to be learned. $\mathbf{h}_{t-1}^r$ is the hidden state of the i-LSTM at $t-1$ time step. The final word-level attention weights are obtained by:

$$\alpha_t^n = \frac{\exp(\beta_t^n)}{\sum_{j=1}^N \exp(\beta_t^j)}. \tag{4}$$

The obtained attention weight α_t^n is thereafter to generate the frame-specific sentence feature as in Eq. (1).

3.3 Grounder

In this section, we introduce the grounder, which works on the yielded interaction status $\mathbf{h}_t^r$ from i-LSTM, to localize the video segment that corresponds to the sentence. Our proposed grounder works in one single pass without introducing overlapping sliding windows, which thus results in a fast runtime. As shown in Figure 2, at each time step t, the grounder efficiently scores a set of K grounding candidates by considering multiple time scales (Buch et al., 2017) that end at time step t. Specifically, we use different K for different datasets, which is determined by the distribution of the lengths of all ground-truth groundings in a certain dataset. To simplify the following discussions, the lengths of K time scales are assumed to be an arithmetic sequence with the common difference δ and all the temporal candidates are sorted by increasing lengths. In other words, the length of the k-th candidate is $k\delta$. Note that all grounding candidates considered at time t have a fixed ending boundary.

Specifically, at each time step t, the grounder will classify each temporal candidate in consideration as a positive grounding or a negative one with respect to the given sentence. Considering multiple time scales, the grounder will generate the confidence scores $\mathbf{C}_t = (c_t^1, c_t^2, ..., c_t^K)$ that correspond to the set of K visual grounding candidates, all ending at time step t. The hidden state $\mathbf{h}_t^r$ generated by i-LSTM at time t, representing the interaction status between the sentence and video sequence up to the current position, is naturally suited to yield the confidence scores for the different time scales ending at time step t. In this paper, the confidence scores, indicating the sentence grounding, are generated by a fully-connected layer with sigmoid nonlinearity:

$$\mathbf{C}_t = \sigma(\mathbf{W}^K \mathbf{h}_t^r + \mathbf{b}_t^r), \tag{5}$$

where $\mathbf{W}^K$ and $\mathbf{b}_t^r$ are the corresponding parameters, and σ denotes the nonlinear sigmoid function.

3.4 Training

The training samples collected in $\mathcal{X}$ for NSGV are video-sentence pairs. Specifically, each video V is temporally associated with a set of sentence annotations: $A = \{(S_i, t_i^b, t_i^e)\}_{i=1}^M$, where M is the number of annotated sentences of the video, and S_i is a sentence description of a video clip, with t_i^b and t_i^e indicating the beginning and ending time in the video. Each training sample corresponds to a ground-truth matrix $\mathbf{y} \in \mathbb{R}^{T \times K}$ with binary entries. We use y_t^k to denote the (t, k)-th entry of the ground-truth matrix. y_t^k is interpreted as whether the k-th grounding candidate at time step t corresponds to the given natural sentence. Concretely, the entry y_t^k is set as 1, indicating that the corresponding video segment (ends at time step t with length $k\delta$) has a temporal Intersection-over-Union (IoU) with (t^b, t^e) larger than a threshold θ. Otherwise y_t^k is set as 0.

For a training pair $(V, S) \in \mathcal{X}$, the objective at time step t is given by a weighted binary cross entropy loss $\mathcal{L}(t, V, S)$:

$$-\sum_{k=1}^{K} w_0^k y_t^k \log c_t^k + w_1^k (1 - y_t^k) \log(1 - c_t^k), \tag{6}$$

where the weights w_0^k and w_1^k are calculated according to the frequencies of positive and negative samples in the training set with length $k\delta$. y_t^k is the ground-truth value and c_t^k denotes the prediction results by our proposed model.

Our TGN backpropagates at every time step t to learn all the parameters of the fully-coupled three modules: encoder, interactor, and grounder. The objective of all training video-sentence pairs $\mathcal{X}$ is defined as:

$$\mathcal{L}_{\mathcal{X}} = \sum_{(V,S)\in\mathcal{X}} \sum_{t=1}^{T} \mathcal{L}(t, V, S). \tag{7}$$

3.5 Inference

During the inference stage, given a testing video V and a sentence S, the textual and visual encoders first generate hidden states for each word and video frame, respectively. Then, the interactor sequentially goes through the video frame by frame to yield the frame-by-word interaction status. At each position t, a K-dimensional score vector $\mathbf{C}_t$ is generated by the grounder. Therefore, after processing the last frame in the video, a $T \times K$ score matrix is obtained for the whole video, with the (t, k)-th entry in the matrix indicating the probability that the video segment ended at position t with length $k\delta$ in video V corresponds to sentence S. Eventually, the evaluation is reduced to a ranking problem over all the grounding candidates based on the generated scores.

4 Experiments

In this section, we evaluate the effectiveness of our proposed TGN on the NSGV task. We begin by describing the datasets used for evaluation, followed by the introduction of the experimental settings including the baselines, configurations, as well as the evaluation metrics. Afterwards, we demonstrate the effectiveness of TGN by comparing with the state-of-the-art approaches and efficiency through a runtime test.

4.1 Datasets

We experiment on three publicly accessible datasets: DiDeMo (Hendricks et al., 2017), TACoS (Regneri et al., 2013), and ActivityNet Captions (Fabian Caba Heilbron and Niebles, 2015). These datasets consist of videos as well as their associated temporally annotated sentences.

DiDeMo[2] consists of 10464 25-50 second long videos. The same split provided by (Hendricks et al., 2017) is used for a fair comparison, with 33008, 4180, and 4022 video-sentence pairs for training, validation, and testing, respectively.

[2] https://goo.gl/JpbAhg.

TACoS[3] consists of 127 videos selected from the MPII Cooking Composite Activities video corpus (Rohrbach et al., 2012). The same split as in (Gao et al., 2017) is used, consisting of 10146, 4589, and 4083 video-sentence pairs for training, validation, and testing, respectively.

ActivityNet Captions[4] consists of $19,209$ videos amounting to 849 hours. The public split is used for our experiments, which has 37421, 17505, and 17031 video-sentence pairs for training, validation, and testing, respectively.

4.2 Experimental Settings

4.2.1 Baselines

We compare our proposed TGN against the following two state-of-the-art models, specifically, the MCN (Hendricks et al., 2017), CTRL (Gao et al., 2017), visual-semantic alignment with LSTM (VSA-RNN) (Karpathy and Li, 2015), and visual-semantic alignment with skip thought vector (VSA-STV) (Kiros et al., 2015). For fair comparisons, we compare the results of MCN on DiDeMo and the results of CTRL, VSA-RNN, VSA-STV on TACoS reported in their papers.

4.2.2 Evaluation Metrics

A grounding of one natural sentence in a video is considered as "correct" if its temporal IoU with the ground-truth boundary is above a threshold θ. To be consistent with the baselines, we adopt R@N, IoU=θ, and mean IoU (mIoU) as our evaluation metrics. R@N, IoU=θ represents the percentage of testing samples which have at least one of the top-N results with IoU larger than θ. mIoU means the average IoU over all testing samples.

4.2.3 Configurations

Generally, the video frame features are usually extracted with a time resolution. For the videos in DiDeMo and TACoS, we sample every 5 second as done by (Hendricks et al., 2017). As the videos in DiDeMo are 25-30 second long, the video feature length is reduced to 6. For videos in ActivityNet Captions, we sample every second. To extract visual features, we consider both appearance and optical flow features. Specifically, we study four widely-used visual features: VGG16 (Simonyan and Zisserman, 2014), C3D (Tran et al., 2015), Inception-V4 (Szegedy et al., 2017), and optical flow (Wang et al., 2016). Please note that when comparing with specific baseline methods, we use the same features as baseline methods, specifically, VGG16 and optical flow for MCN and C3D for CTRL, VSA-RNN, and VSA-STV.

Table 1: Performance comparisons of different methods on DiDeMo. The best performance for each metric entry is highlighted in boldface.

Method	R@1 IoU=1	R@5 IoU=1	mIoU
MFP	19.40	66.38	26.65
MCN-VGG16	13.10	44.82	25.13
MCN-Flow	18.35	56.25	31.46
MCN-Fusion	19.88	62.39	33.51
MCN-Fusion+TEF	28.10	78.21	41.08
TGN-VGG16	24.28	71.43	38.62
TGN-Flow	27.52	76.94	42.84
TGN-Fusion	**28.23**	**79.26**	**42.97**

For sentences, we tokenize each sentence by Stanford CoreNLP (Manning et al., 2014) and use the 300-D word embeddings from GloVe (Pennington et al., 2014) to initialize the models. The words not found in GloVe are initialized as zero vectors. The hidden state dimensions of all LSTMs (including the video, sentence, and interaction LSTMs) are set as 512. We use the Adam (Kingma and Ba, 2014) optimizer with β_1 = 0.5 and β_2 = 0.999. The initial learning rate is set to 0.001. We train the network for 200 iterations, and the learning rate is gradually decayed over time. The mini-batch size is set to 64.

4.3 Experimental Results and Analysis

4.3.1 Comparisons with State-of-the-Arts

Experiments on DiDeMo. Table 1 illustrates the performance comparisons on the DiDeMo dataset. In addition to MCN, we also compare with the baseline Moment Frequency Prior (MFP) in (Hendricks et al., 2017), which selects segments corresponding to the positions of videos in the training dataset with most annotations. First, TGN with different features can significantly outperforms the "prior baseline" MFP, which retrieves segments corresponding to the most common start and end points in the dataset. Second, it can be observed that with the same visual features, specifically VGG16 and optical flow, TGN significantly outperforms MCN. And the performance of TGN with optical flow is better than that with VGG16. One possible reason is that the videos in DiDeMo are relatively short, which only contain a single event. In such a case, the action information plays

[3] https://goo.gl/ajmsva.

[4] https://goo.gl/N355bG.

Table 2: Performance comparisons of different methods on TACoS. The best performance for each metric entry is highlighted in boldface.

Method	R@1 IoU=0.5	R@1 IoU=0.3	R@1 IoU=0.1	R@5 IoU=0.5	R@5 IoU=0.3	R@5 IoU=0.1
VSA-RNN	4.78	6.91	8.84	9.10	13.90	19.05
VSA-STV	7.56	10.77	15.01	15.5	23.92	32.82
CTRL-C3D	13.30	18.32	24.32	25.42	36.39	48.73
TGN-C3D	**18.90**	**21.77**	**41.87**	**31.02**	**39.06**	**53.40**

a more critical role. This finding is also consistent with (Hendricks et al., 2017). By fusing the results obtained by VGG16 and optical flow together, the performance can be further boosted, as demonstrated by TGN-Fusion and MCN-Fusion. Third, MCN introduces the temporal endpoint feature (TEF) as prior knowledge, which indicates when a segment occurs in a video. With TEF, the performance of MCN can be significantly improved. However, it is still inferior to our proposed TGN.

MCN is designed as an enumeration-based approach. Each video in the DiDeMo dataset is split into six five-second chunks which are considered as the time unit for localization. Therefore, in total there are only $\mathrm{C}_7^2 = 7 \times 6/2 = 21$ different ways of localization for DiDeMo videos. Therefore, although MCN can be effectively applied to videos with several chunks due to the small search space, it is not practical for untrimmed long videos. In the Section 4.3.3, we will evaluate and compare the efficiencies of MCN, CTRL, and our proposed TGN.

Experiments on TACoS. Table 2 illustrates the experimental results on TACoS. First, it can be observed that CTRL performs much better than VSA-RNN and VSA-STV. The reasons lie in twofold (Gao et al., 2017). On one hand, CTRL utilizes a multilayer alignment network to learn better alignment. On the other hand, VSA-RNN and VSA-STV do not encode temporal context information of video. Second, with the same visual feature, specifically C3D, TGN-C3D significantly outperforms CTRL-C3D. This is due to the fact that TGN exploits not only the contextual information but also the fine-grained interaction behaviors. More concretely, TGN considers the frame-by-word correlations by introducing an attentive combinations of the words in the sentence, where each weight encodes the degree to which the word is aligned with each specific frame. This mechanism is beneficial to capturing the informative semantics in the sentences for alignment.

Table 3: Performance comparisons of different visual features on ActivityNet Captions. The best performance for each metric entry is highlighted in boldface.

Feature	R@1 IoU=0.5	R@1 IoU=0.3	R@1 IoU=0.1	R@5 IoU=0.5	R@5 IoU=0.3	R@5 IoU=0.1
C3D	27.93	43.81	69.59	**44.20**	54.56	78.66
VGG16	23.90	42.24	65.76	40.17	51.82	76.21
Inception-V4	**28.47**	**45.51**	**70.06**	43.33	**57.32**	**79.10**

Table 4: Ablation studies on TACoS. The best performance for each metric entry is highlighted in boldface.

Feature	R@1 IoU=0.5	R@1 IoU=0.3	R@1 IoU=0.1	R@5 IoU=0.5	R@5 IoU=0.3	R@5 IoU=0.1
NA	5.53	7.67	24.23	15.20	18.94	41.25
NM	13.89	18.60	41.41	26.60	31.74	47.70
TGN	**18.90**	**21.77**	**41.87**	**31.02**	**39.06**	**53.40**

Experiments on ActivityNet Captions. Besides the two benchmarks, we also evaluate our model on the ActivityNet Captions dataset. Different CNNs are used to encode video visual information. Specifically, we consider VGG16, C3D, and Inception-V4. The results are included in Table 3. First, our proposed TGN can perform effectively on long untrimmed videos. Second, Inception-V4 performs generally better than VGG16 and C3D, which is consistent with the finding in (Canziani et al., 2016). Therefore, more powerful visual representations of video features will undoubtedly improve the the performance of our proposed TGN on the NSGV task.

Some qualitative results of our proposed TGN on ActiveityNet Captions dataset is illustrated in Figure 3. It can be observed that with different visual features, different grounding results are obtained. For the first and second examples, TGN with VGG16 and Inception-V4 generates more accurate groundings than that with C3D, while TGN with C3D yields more accurate grounding results for the third example. More specifically, our proposed TGN with VGG16 and Inception-V4 can well identify the visual information related with the sentence, i.e. "`A man in a red shirt claps his hands`".

4.3.2 Effect of Frame-by-Word Attention

We examine the effect of the frame-by-word attention in interactor. We ablate TGN into two other methods. 1) NA: There is no attention layer in this model. After obtaining the sequential hidden states of the sentence, mean pooling is used to generate the representation for the whole sentence.

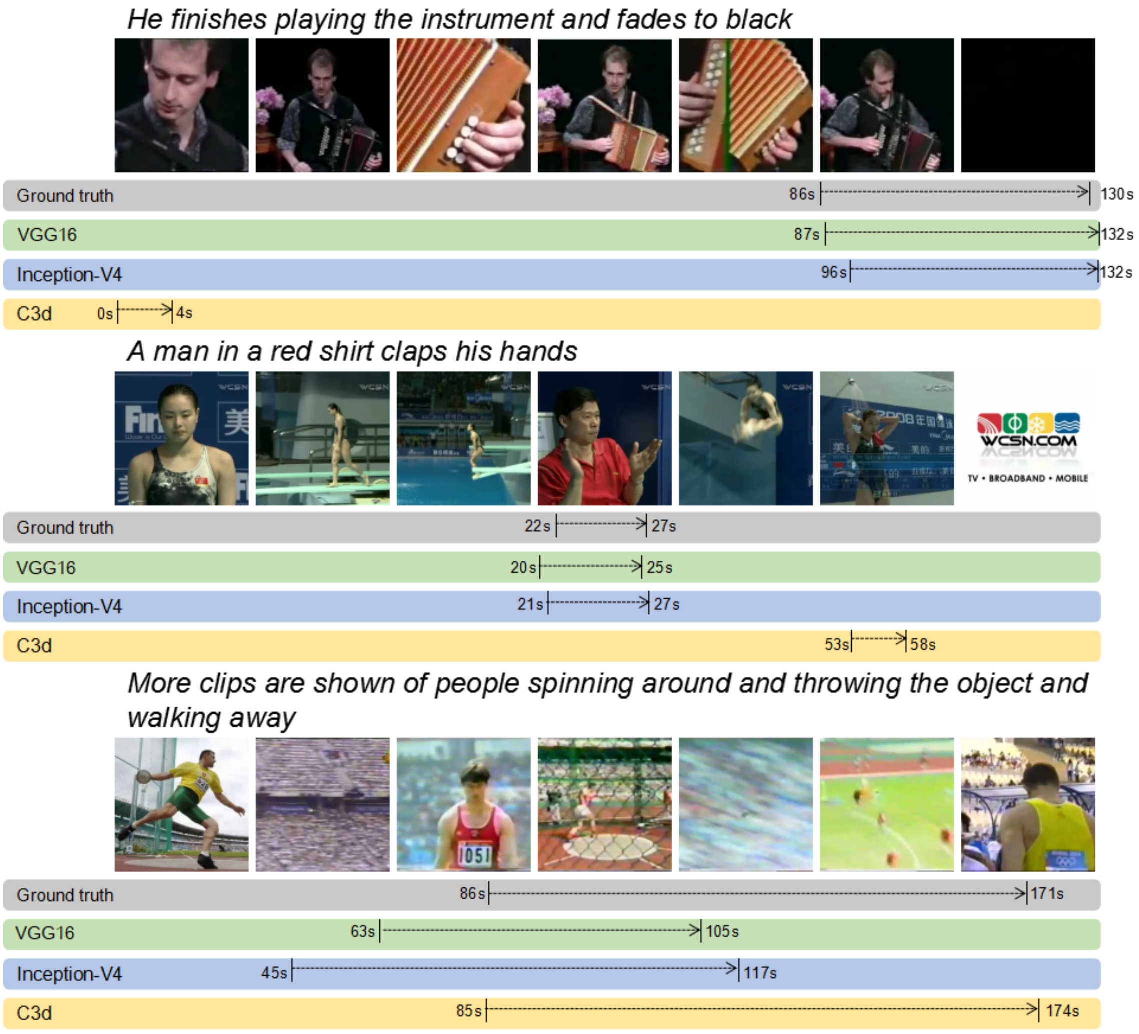
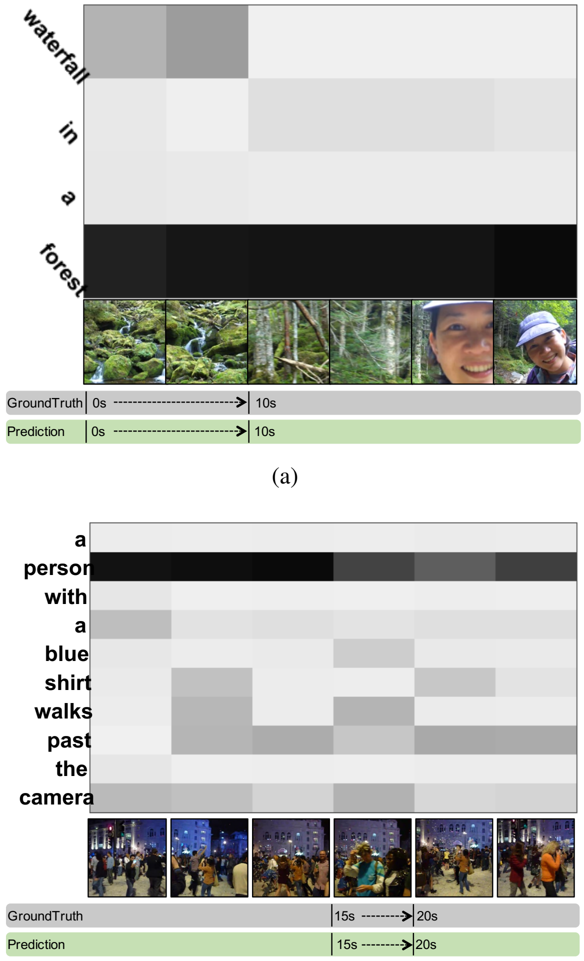


Figure 3: The qualitative grounding results of our TGN model on the ActiveityNet Captions dataset with different visual features.

Then the generated representation is concatenated with video representation, based on which the scores for multiple grounding candidates are predicted. 2) NM: The idea of generating frame-specific sentence feature is still reserved in the NM model. The difference between NM and TGN is that there is no interaction LSTM in NM. Specifically, when calculating the attention weight for each word as in Eq. (3), the hidden state $\mathbf{h}^r_{t-1}$ indicating the interaction status is not incorporated.

The quantitative results are displayed in Table 4. First, when the attention mechanism is applied (NM), the performance is improved as compared with utilizing mean pooling (NA) for sentence features. The better performance demonstrates that our assumption about the evolving frame-by-word correlations between two modalities is reasonable. This also indicates that it is necessary to discriminate the contribution of each word in a sentence to perform the NSGV task. Second, utilizing the interaction LSTM module (TGN) achieves better performance than simply concatenating the video representation and the attentive sentence representation (NM). This result indicates that the interaction LSTM yields better interaction status between these two modalities, which can thereby benefit the final grounding.

We provide some qualitative examples in Figure 4 for a better understanding of the frame-by-word attention. Meanwhile, the grounding results yielded by TGN-Fusion (considering both VGG16 and optical flow) are also illustrated. This experiment is designed to verify whether the frame-by-word attention mechanism in interactor is useful to highlight the representative concepts in the sentence. The attention weights α for two testing samples in DiDeMo are illustrated in Figure 4, where the darker the color is, the larger



(a)

(b)

Figure 4: Visualization results on frame-by-word attention. The darker the color is, the larger its represented attention value is.

Table 5: Efficiency comparison in terms of frame per second.

	CTRL	MCN	TGN
FPS	562	286	1,363

the attention weight is. It can be observed that some words well match the frames. For example, in Figure 4 (a), the concept "forest" appears across all the video frames presenting an evenly distributed attention weights, while the other concept "waterfall" only presents in the first two frames. In addition to nouns, the adjective "blue" in Figure 4 (b) also receives relatively higher attention weights in relevant frames. Lastly, for stop words like "a", "the" and "in", their attention weights, which are very small, also present an even distribution.

4.3.3 Efficiency

We evaluate the efficiency of our proposed TGN, by comparing its runtime with MCN and CTRL on a Tesla M40 GPU. The efficiency is measured by frames per second (FPS) as shown in Table 5. Please not that the feature extraction time is excluded. It can be observed that our TGN model achieves much faster processing speeds, with 1,363 fps vs. 562 and 286 for CTRL and MCN, respectively. The reason mainly attributes to that the proposed TGN only process each video in one single pass without processing overlapped sliding windows.

5 Conclusion

In this paper, we focused on the task of natural sentence grounding in video that is believed to offer a comprehensive understanding of bridging computer vision and natural language processing. Towards this task, we proposed an end-to-end Temporal GroundNet (TGN) by incorporating the evolving fine-grained frame-by-word interactions across video-sentence modalities to generate a visual grounding tailored to each given natural sentence. Moreover, TGN performs efficiently, which only needs to process the video sequence in one single pass. Extensive experiments on three real-world datasets clearly demonstrate the effectiveness and efficiency of the proposed TGN.

References

Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. 2017. Sst: Single-stream temporal action proposals. In *CVPR*, pages 5534–5542.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678.

Jingyuan Chen, Xuemeng Song, Liqiang Nie, Xiang Wang, Hanwang Zhang, and Tat-Seng Chua. 2016. Micro tells macro: Predicting the popularity of micro-videos via a transductive model. In *MM*, pages 898–907.

Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *SIGIR*, pages 335–344.

Xinpeng Chen, Jingyuan Chen, Lin Ma, Jian Yao, Wei Liu, Jiebo Luo, and Tong Zhang. 2018. Fine-grained video attractiveness prediction using multimodal deep learning on a large real-world dataset. In *WWW*.

Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970.

Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. 2018. Video re-localization. In *ECCV*.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. TALL: temporal activity localization via language query. In *ICCV*, pages 5277–5285.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2017. Localizing moments in video with natural language. In *ICCV*, pages 5804–5813.

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

Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, pages 4418–4427.

Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *CVPR*, pages 4555–4564.

Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. 2016a. Tree-structured reinforcement learning for sequential object localization. In *NIPS*, pages 127–135.

Zequn Jie, Xiaodan Liang, Jiashi Feng, Wen Feng Lu, Francis Eng Hock Tay, and Shuicheng Yan. 2016b. Scale-aware pixelwise object proposal networks. *IEEE Trans. Image Processing*, 25(10):4525–4539.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. 2014. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, pages 2657–2664.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60. The Association for Computer Linguistics.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20.

Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. 2016. Modeling context between objects for referring expression understanding. In *ECCV*, pages 792–807.

Ramakanth Pasunuru and Mohit Bansal. 2017. Multi-task video captioning with video and entailment generation. In *ACL*, pages 1273–1283.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *TACL*, 1:25–36.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *ECCV*, pages 817–834.

Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. 2012. Script data for attribute-based recognition of composite activities. In *ECCV*, pages 144–157.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284.

Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497.

Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. 2013. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171.

Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. 2018a. Reconstruction network for video captioning. In *CVPR*.

Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. 2018b. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36.

Shuohang Wang and Jing Jiang. 2016a. Learning natural language inference with LSTM. In *NAACL*, pages 1442–1451.

Shuohang Wang and Jing Jiang. 2016b. Machine comprehension using match-lstm and answer pointer. *CoRR*, abs/1608.07905.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015a. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057.

Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. 2015b. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, pages 2346–2352.

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from video described with sentences. In *ACL*, pages 53–63.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. *CoRR*, abs/1801.08186.

Yuke Zhu, Oliver Groth, Michael S. Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *CVPR*, pages 4995–5004.

C. Lawrence Zitnick and Piotr Dollár. 2014. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405.