Table of Content

Introduction

Video Grounding

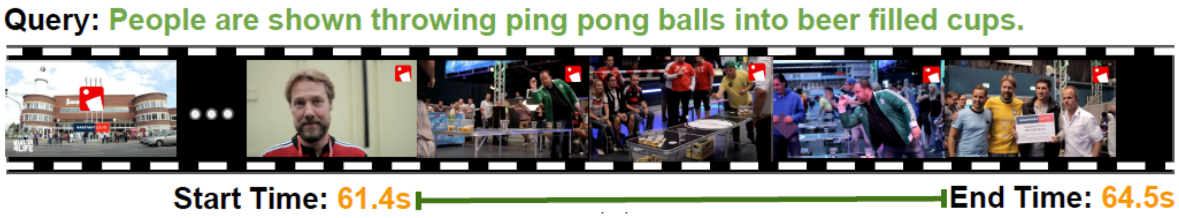

Query: People are shown throwing ping pong balls into beer filled cups.
Start Time: 61.4s
End Time: 64.5s

Figure 1: Illustration of video grounding

Video grounding [9,12] is the locating of temporal boundaries within a target video that match a given textual description. This task is useful in applications such as video question answering [23] and video summarization [24]. In Figure 1, the query “People are shown throwing ping pong balls into beer filled cups” contains three objects, “ping pong balls”, “beer”, and “cups” and two actions “shown” and “throwing”. The query would return the start time of 61.4s and end time of 64.5s within which the relevant events can be observed in the video.

Previous methods for video grounding can be categorised into three groups: 1) ranking methods that count on a two-stage propose-and-rank pipeline, or attention-based localisation approach to find the best scoring target video span among multiple options. 2) regression methods that directly predict the start and end times of the query reducing computational load for high numbers of candidates, and 3) reinforcement learning approaches that dynamically remove sequences of frames conditioned on the given textual query and final outputs temporal boundaries.

We aim to accomplish video grounding using a dual contrastive learning method to learn more informative feature representations by maximising the mutual information (MI) between query and video clips as well as the MI between start/end frames of a target moment.

Contrastive Learning

Contrastive learning is an approach to formulate the task of learning representations for similar and dissimilar instances by contrasting positive pairs against negative pairs. For a datapoint x, the data points that are similar to x are regarded as positive samples x^{+} while the data points that are dissimilar to x are regarded as negative samples x^{-}. The model is expected to learn representations for the data points where the similarity between x^{+} and x is large and the similarity between x^{-} and x is small.

Some prior works apply the maximization of mutual information (MI) between latent representations. MI measures how much information is communicated between two random variables. In our approach, we leveraged the idea of MI maximization in the two contrastive learning modules to facilitate the effectiveness of the visual and textual encoder such that they encode useful information. We aim to maximize the MI between the learned representation, i.e. the encoders' output, and input data. The details of the contrastive modules will be discussed later in this report.

Methodology

Model Architecture

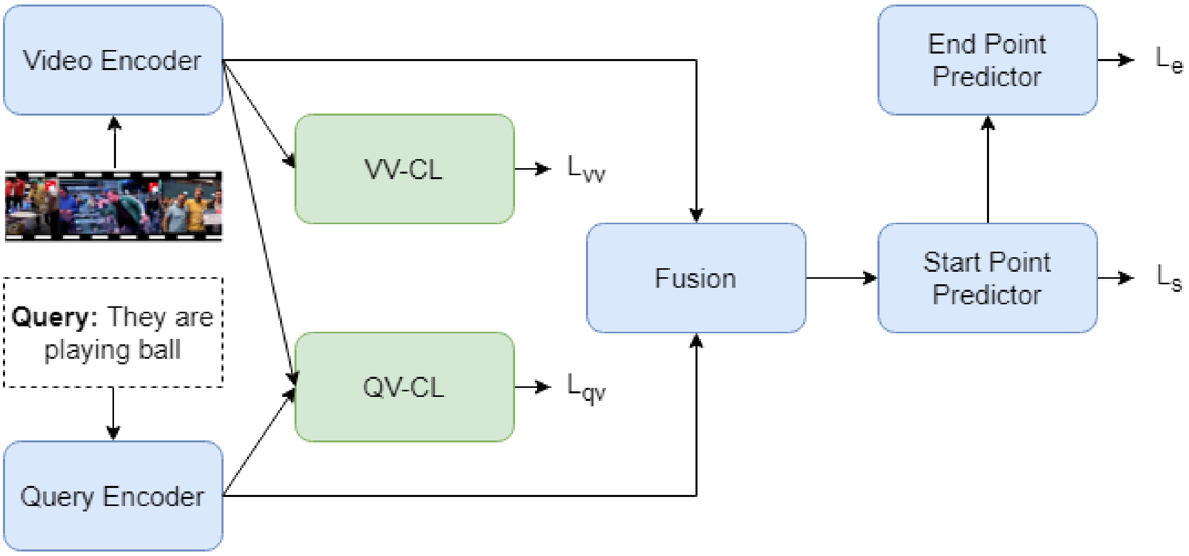


Figure 2. Model architecture. VV-CL and QV-CL are the two contrastive modules.

Our model consists of 4 main modules, feature extractors, contrastive learning module, a fusion module and the boundary predictors. Figure 2 illustrates the architecture of our model.

(1) Given a text query and video, the query encoder and video encoder extract the text and video representation respectively.
(2) The text and video representations are fed into two contrastive modules to learn better video representation and better query-video alignment.
(3) The text and video representations are fed into a fusion model with a context-query attention mechanism to capture the cross-modal interaction between visual and text features.
(4) The output from the fusion model is fed to a start point predictor to output the start index of the video sequence and the start point feature. Then the start point feature is fed to the end point predictor to output the end index of the video sequence.

The four losses, contrastive losses between video-video pair and video-query pair, and the losses for the start and end point predictor are combined to train the model.

Feature Encoder

We extracted the features for the video using the pre-trained video feature extraction model C3D. The extracted video representation can be denoted as $V \in R^{T \times d_v}$ where T indicates the number of extracted features from the video and d_v is the dimension of the video features. For the query input, we applied Glove embedding to obtain the query features. The query representation can be denoted as $Q \in R^{N \times d_w}$ where N refers to the text query length and d_w refers to the dimension of the embedding for a word.

We used two linear layers to project the video features V and query features Q to the same dimension d. To obtain the contextualized representation of video and query features respectively, we applied an encoder with four convolutional layers with the attention mechanism and a feed forward layer to generate contextualized $Q' \in R^{N\times d_v}$ and $V' \in R^{T\times d_v}$.

Contrastive Module

We applied contrastive learning to guide the encoder to better align the video and query representations and to learn better video representation. To address these two objectives, we used two contrastive learning modules.

V-Q Contrastive Learning

To better align the query and video representations, we regard the video clips that reside in the target boundary (with the boundary video clips) as positive samples V^+ with respect to the query q, and the ones that are outside of the target boundary as negative samples V^- with respect to the query q. To compute the contrastive loss, we applied the Jensen-Shannon mutual information (MI) estimator to maximize the MI between positive pairs $(V^+, q)$and minimize the MI between the negative pairs (V^-, q).
The contrastive loss L_{vq} is represented as:

$$L_{vq} = E_{V^-}[softplus(C(q, V'))] - E_{V^+}[softplus(C(q, V'))]$$

where the softplus function is:

$$softplus(z) = log(1 + e^z)$$

and the $C: R^{d_v} \times R^{d_v} \rightarrow R$ denotes the MI discriminator.

V-V Contrastive Learning

To facilitate the encoder to learn better video representation, we treat the video clips that reside in the target boundary (without the boundary video clips) as positive samples V^+ and those outside the target boundary as negative samples V^-. We extracted out the feature representation of the start boundary and end boundary as v'_s and v'_e respectively. We aim to maximize the MI between v'_s and V^+, as well as the MI between v'_e and V^+. We want to minimize the MI between v'_s and V^-, as well as the MI between v'_e and V^-. Therefore, similar to the L_{vq}, the contrastive loss L_{vv} is represented as:

$$L_{vv} = E_{V^-}[softplus(C(v'_s, V'))] - E_{V^+}[softplus(C(v'_s, V'))] + \\ E_{V^-}[softplus(C(v'_e, V'))] - E_{V^+}[softplus(C(v'_e, V'))]$$

Fusion Module

We used context-query-attention (CQA) to capture the bi-modal interaction between video and text features. We applied a feed-forward layer after the CQA to get the fusion module output X.

$$X = FFN(CQA(V', Q'))$$

Boundary Predictors

The start boundary predictor simply takes the output of the fusion module as input, and generates the logits for the start index. The predictor consists of a feature encoder to represent the input X as a hidden feature vector, and two convolutional layers applied on the concatenation of X and the hidden feature vector. A mask is then applied on the convolutional layers output to gain the logits.

Similarly, the end boundary predictor takes the start boundary's hidden feature vector and X as inputs, and output the logits by two convolutional layers.

We used cross entropy to calculate the loss for the start boundary and end boundary, denoted as L_s and L_e.

Training Objective

The overall training loss is:

$$L = \alpha L_{vq} + \beta L_{vv} + L_s + L_e$$

where αand βare hyperparameters representing the weights for the contrastive losses.

Experiments

Dataset

TACoS [10] is constructed from MPII Cooking Composite Activities dataset. We follow the same data split as previous works for fair comparisons. There are 10146, 4589 and 4083 instances in training, validation and testing dataset, respectively. Each video has 148 language sequences as the query on average.

Charades-STA [8] is a benchmark for the video grounding task, which is generated based on Charades [4] dataset mainly for various indoor activities. There are 12408 and 3720 moment annotations for the training and testing dataset, respectively.

Experimental Settings

Evaluation Metrics: We follow previous works to use ”R@n, IoU = μ” as our evaluation metrics, which denotes the percentage of testing samples that have at least one correct result. Correct here indicates that intersection over IoU with ground truth is larger than μ in top-n retrieved moments.

Settings: We implement our model in Pytorch. We follow the previous works to use the same pre-trained video features and Glove 300-dimension word embedding.

For the model parameters, we set the word embedding dimension as 300, visual feature dimension to 1024 for i3d pre-pretrained model and 4096 for c3d pre-trained model, hidden size of the model to 128, and drop out rate to 0.2 for the whole model.

For the training and evaluation parameters, random seed is set to 12345, number of training epochs to 45, batch size to 16, and initial learning rate to 0.0004.

For the objective, the loss weights α and β are configured as 0.3 and 0.3 respectively.

Model Zoo

There are previous works that work well for the video grounding task. They mostly fall under three different categories of approach - Ranking Methods, Regression Model, and Reinforcement Learning Methods.

Ranking methods rely on multi-modal matching architectures to obtain the target moment with the highest confidence score. Typical example models are *2D-TAN* [18] and *MAN* [22].

Regression Models directly regress the moment boundaries to avoid heavy computations. Typical example models include *ABLR* [21], *DEBUG* [2], *DRN* [15], and *VSLNet* [5].

Reinforcement Learning Methods progressively localize the moment boundaries for a given query. *SMRL* [19] and *RWM* [3] treat the problem of video grounding as a sequential decision-making process, which naturally can be resolved by the RL paradigm.

Performance Comparisons

The results of our method on TACoS [10] and Charades-STA [10] datasets are given in Table 1 and Table 2.

As shown in Table 1, our proposed Video Grounding model consistently outperforms the baselines under various settings and achieves the new state-of-the-art performance on TACoS dataset.

Table 2 summarizes the comparisons on the Charades-STA datasets and it shows that our model can have comparable performance with those baselines mentioned in the Model Zoo.

Model Categrory	Model Name	IoU=0.1	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
RL	SM-RL [19]	26.51	20.25	15.95	-	-
	TripNet [11]	-	23.95	19.17	-	-
Ranking	ROLE [12]	20.37	15.38	9.94	-	-
	MCN [9]	14.42	-	5.58	-	-
	CTRL [8]	24.32	18.32	13.30	-	-
	ACRN [13]	24.22	19.52	14.62	-	-
	QSPN [6]	25.31	20.15	15.23	-	-
	MAC [16]	31.64	24.17	20.01	-	-
	SAP [17]	31.15	-	18.24	-	-
	TGN [7]	41.87	21.77	18.90	-	-
	2D-TAN [18]	47.59	37.29	25.32	-	-
Regression	SLTA [1]	23.13	17.07	11.92	-	-
	VAL [20]	25.74	19.76	14.74	-	-

	ABLR [21]	34.70	19.50	9.40	-	-
	DEBUG [2]	-	23.45	11.72	-	16.03
	DRN [15]	-	-	23.17	-	-
	VSLBase [5]	-	23.59	20.40	16.65	20.10
	VSLNet [5]	-	29.61	24.27	**20.03**	24.11
Ours		**53.69**	**42.26**	**31.69**	19.87	**30.49**

Table 1: Performance comparisons on the TACoS dataset.

Model Categrory	Model Name	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
RL	SM-RL [19]	-	24.36	11.17	-
	RWM [3]	-	36.70	-	-
Ranking	CTRL [8]	-	23.63	8.89	-
	ACRN [13]	-	20.26	7..64	-
	SAP [17]	-	27.42	13.36	-
	MAC [16]	-	30.48	12.20	-
	QSPN [6]	54.70	35.60	15.80	-
	2D-TAN [18]	-	39.70	23.31	-
	MAN [22]	-	46.53	22.72	-
Regression	DEBUG [2]	54.95	37.39	17.69	36.34
	DRN [15]	-	45.40	26.40	-
	VSLBase [5]	61.72	40.97	24.14	42.11
	VSLNet [5]	64.30	47.31	30.19	45.15
Ours		**65.00**	**47.63**	**30.59**	**46.22**

Table 2: Performance comparisons on the Charades-STA dataset.

Loss and Accuracy Visualization

We record the total loss for every 100 batches during training. How the loss varies is shown in Figure 3.

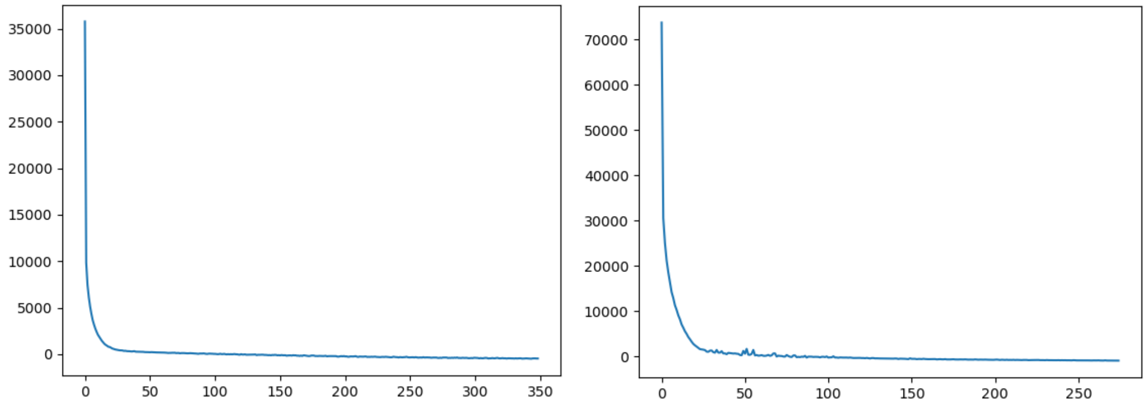
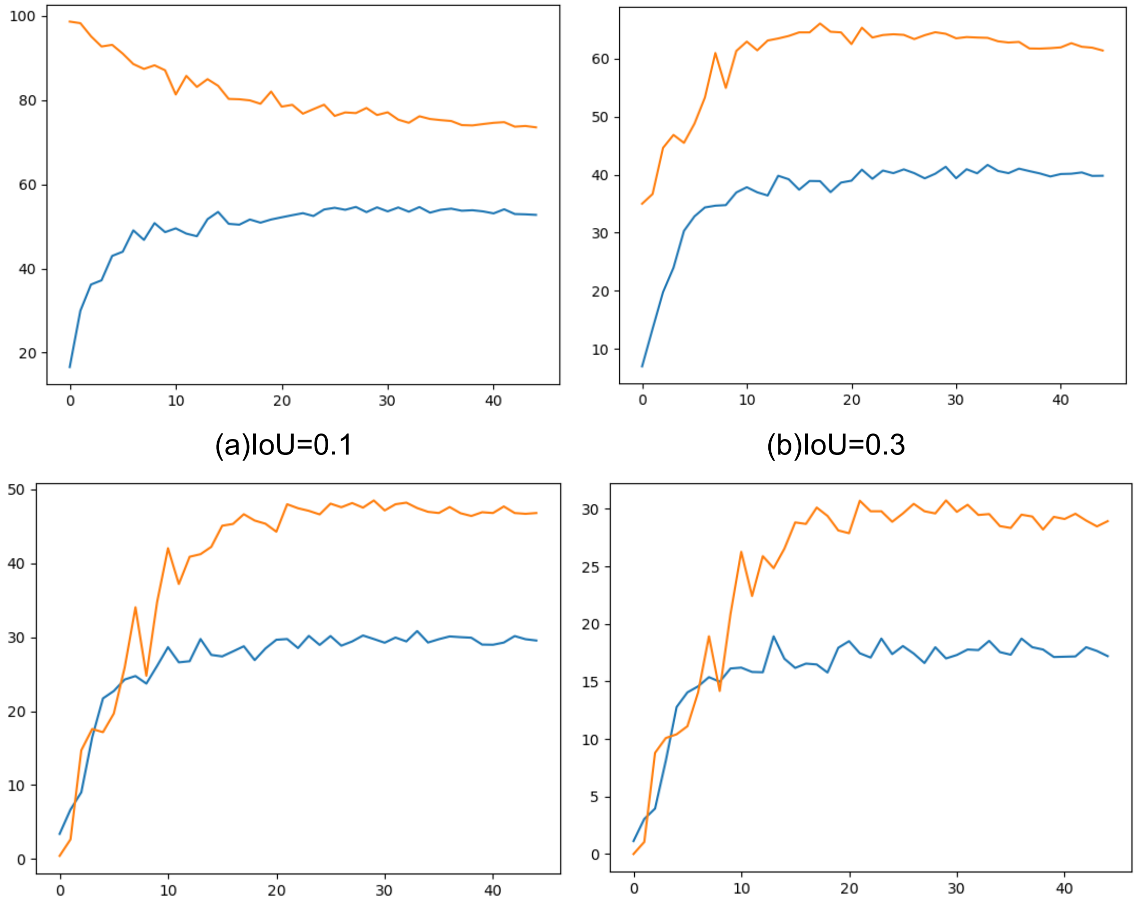


Figure 3: Loss curve during training. Left is training with Charades-STA, and right is with TACoS

We also save the evaluation result ”R@n, IoU = μ” for each epoch during training to have a better sense of whether the model is learning correctly. Figure 4 illustrates how our model's performance changes during training.



(a)IoU=0.1

(b)IoU=0.3

(c)IoU=0.5

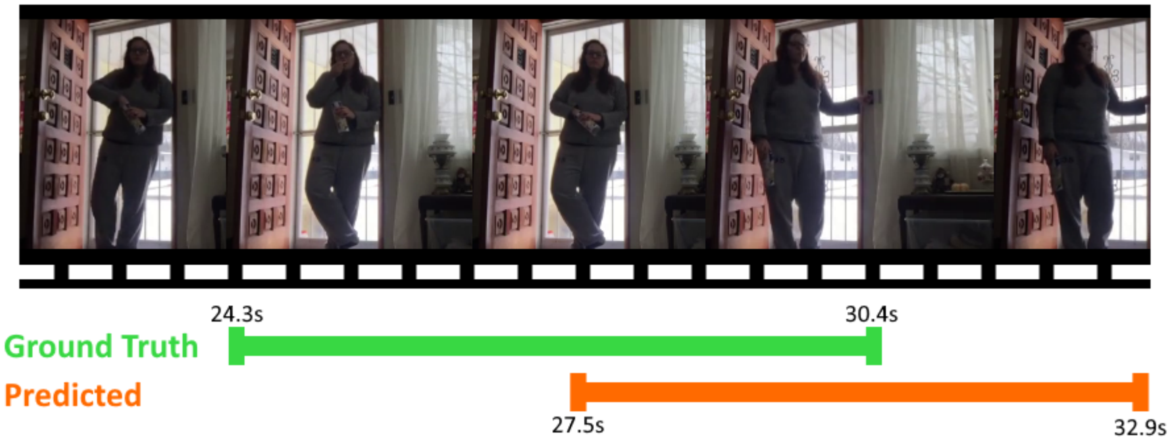
(d)IoU=0.7

Figure 4: Performance curve during training. Line with blue color represents TACoS dataset while another is Charades-STA

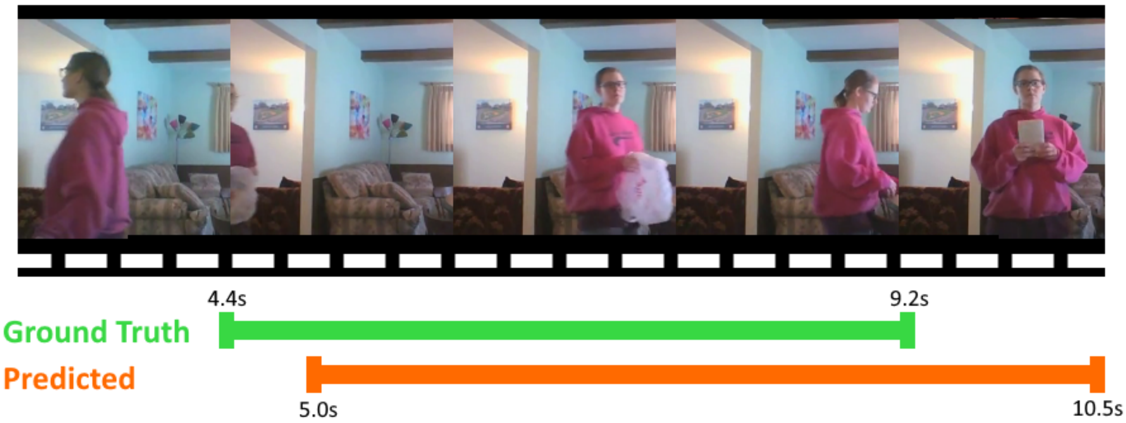
Case Study

In this section, we are illustrating several example videos from TACoS and Charades-STA datasets and the temporal indexes predicted by our model.

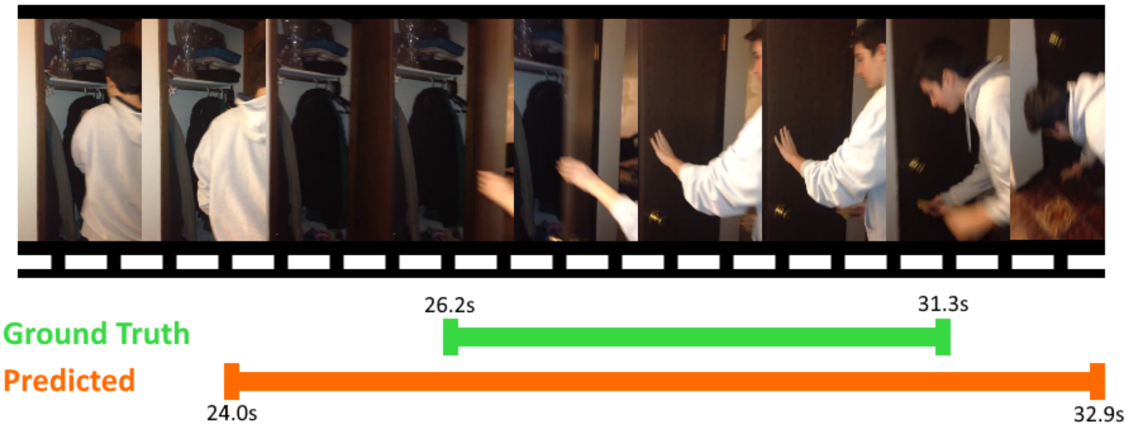
Query: "person flipped the light switch near the door"



Query: "the person puts down the bag"



Query: "person closes the door"



Conclusion

In this project, a new Video Grounding model that cooperates with various deep neural architectures and auxiliary contrastive loss is proposed and implemented. Experiments on two standard datasets show the effectiveness of our proposed model.

Team Contribution

Zachary Tan: Report Writing
Li Jiaxi: Idea Generation, Detail Discussion, Report Writing
Leng Sicong: Idea Generation, Detail Discussion, Report Writing, Code Implementation

References

[1] Bin Jiang, Xin Huang, Chao Yang, and Junsong Yuan. Crossmodal video moment retrieval with spatial and language temporal attention. In ICMR, 2019.

[2] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. Debug: A dense bottom-up grounding approach for natural language video localization. In EMNLP, pages 5147–5156, 2019.

[3] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In AAAI, 2019.

[4] Gunnar A Sigurdsson, G¨ul Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In ECCV, 2016.

[5] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In ACL, 2020.

[6] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In AAAI, 2019.

[7] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat Seng Chua. Temporally grounding natural sentence in video. In EMNLP, 2018.

[8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In CVPR, 2017.

[9] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In ICCV, 2017.

[10] Marcus Rohrbach, Michaela Regneri, Mykhaylo Andriluka, Sikandar Amin, Manfred Pinkal, and Bernt Schiele. Script data for attribute-based recognition of composite

activities, 2012.

[11] Meera Hahn, Asim Kadav, James M Rehg, and Hans Peter Graf. Tripping through time: Efficient localization of activities in videos. arXiv preprint arXiv:1904.09936, 2019.

[12] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, BaoquanChen, and Tat-Seng Chua. Cross-modal moment localizationin videos. In ACMMM, 2018.

[13] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In SIGIR, 2018.

[14] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In ICLR, 2018.

[15] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In CVPR, 2020.

[16] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In WACV, 2019.

[17] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In AAAI, 2019.

[18] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In AAAI, 2020.

[19] Weining Wang, Yan Huang, and Liang Wang. Language driven temporal activity localization: A semantic matching reinforcement learning model. In CVPR, 2019.

[20] Xiaomeng Song and Yahong Han. Val: Visual-attention action localizer. In PCM, 2018.

[21] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. In AAAI, 2019.

[22] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In CVPR, 2019.

[23] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In EMNLP, 2018.

[24] Jiaxin Wu, Sheng-hua Zhong, and Yan Liu. Dynamic graph convolutional network for multi-video summarization. Pattern Recognition, 107:107382, 2020.