ExCL: Extractive Clip Localization Using Natural Language Descriptions

Soham Ghosh[1,*] **Anuva Agarwal**[1,*] **Zarana Parekh**[1,*] **Alexander Hauptmann**[1]

[1]Language Technologies Institute
Carnegie Mellon University

{sohamg,anuvaa,zpp,alex}@cs.cmu.edu

Abstract

The task of retrieving clips within videos based on a given natural language query requires cross-modal reasoning over multiple frames. Prior approaches such as sliding window classifiers are inefficient, while text-clip similarity driven ranking-based approaches such as segment proposal networks are far more complicated. In order to select the most relevant video clip corresponding to the given text description, we propose a novel extractive approach that predicts the start and end frames by leveraging cross-modal interactions between the text and video - this removes the need to retrieve and re-rank multiple proposal segments. Using recurrent networks we encode the two modalities into a joint representation which is then used in different variants of start-end frame predictor networks. Through extensive experimentation and ablative analysis, we demonstrate that our simple and elegant approach significantly outperforms state of the art on two datasets and has comparable performance on a third.

1 Introduction

Clip Localization is the task of selecting the relevant span of temporal frames in a video corresponding to a natural language description and has recently piqued interest in research that lies at the intersection of visual and textual modalities. An example of this task is demonstrated in Figure 1. It requires cross-modal reasoning to ground free-form text inside the video and calls for models capable of segmenting a video into action segments (Singh et al., 2016; Yeung et al., 2016; Xu et al., 2017) as well as measuring multi-modal semantic similarity (Karpathy and Fei-Fei, 2015).

This task is inherently discriminative, i.e., there is only a single most relevant clip pertaining to a given query in the corresponding video. However, most prior works (Hendricks et al., 2017, 2018; Liu et al., 2018; Chen et al., 2018; Zhang et al., 2018) explore this as a ranking task over a fixed number of moments by uniformly sampling clips within a video. Moreover, these approaches are restrictive in scope since they use predefined clips as candidates for a video and cannot be easily extended to videos with considerable variance in length.

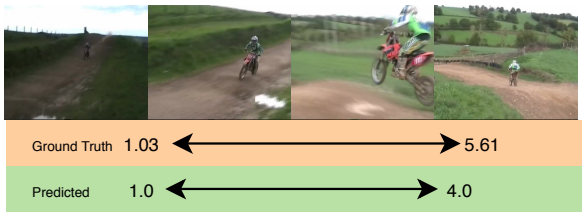


Figure 1: Clip Extraction task for the given query '*the biker jumps to another ramp near the camera*'

Gao et al. (2017); Xu et al. (2019) apply two-stage methods which rank candidate clips using a learned similarity metric. Gao et al. (2017); Ge et al. (2019) propose a sliding window approach with alignment and offset regression learning objective, but it is limited by the coarseness of the windows and is thus inefficient and inflexible. Xu et al. (2019) address this through a query-guided segment proposal network (QSPN). However, the similarity metric used by these approaches is difficult to learn as it is sensitive to the choice of negative samples (Yu et al., 2018) and it still does not consider the discriminative nature of the task.

Hence, we propose an elegant and fairly simple extractive approach. Our technique is similar to text-based Machine Comprehension (Chen et al., 2017) but in a multimodal setting where the video is analogous to the text passage and the target-clip is analogous to the text span corresponding to the correct answer. We verify empirically that

*Equal contribution, randomly ordered.

Proceedings of NAACL-HLT 2019, pages 1984–1990
Minneapolis, Minnesota, June 2 - June 7, 2019.

our method significantly outperforms prior work on two benchmark datasets - TACoS, ActivityNet and comparably well on the third, Charades-STA. Our flexible, modular approach to *Extractive Clip Localization (ExCL)* can easily be extended to incorporate attention models and different variants of encoders for both visual and text modality to improve performance further.

2 Approach

Our model comprises of three modular parts - a text encoder, a video encoder and a span predictor as shown in Figure 2. We use a bidirectional text LSTM as the text encoder with pre-trained GloVE (Pennington et al., 2014) embeddings as input, and use the last hidden state ($\mathbf{h}^T$) as sentence-embedding. Simpler variants such as bag-of-words, and other approaches such as InferSent (Conneau et al., 2017) or Skip-Thought (Kiros et al., 2015) could also be used. We use a bidirectional LSTM with I3D features (Carreira and Zisserman, 2017) as the video encoder which captures temporal context at each time-step ($\mathbf{h}_t^V$). Note that this could also be substituted by C3D features (Tran et al., 2014) or self-attentive networks (Kim et al., 2018). Finally, we compare the variants of span-predictor networks which output scores $S_{\text{start}}(t), S_{\text{end}}(t) \in \mathbb{R}^1$ respectively.

2.1 Training objectives

We consider two modes of training our networks, one which uses a classification loss (ExCL-clf) and another which uses a regression loss (ExCL-reg).

ExCL-clf: The scores are normalized using SoftMax to give $P_{\text{start}}(t), P_{\text{end}}(t)$, and trained using negative log-likelihood loss:

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N} \log(P_{\text{start}}(t_s^i)) + \log(P_{\text{end}}(t_e^i)) \tag{1}$$

where N is number of text-clip pairs and t_s^i, t_e^i are the ground-truth start and end frame indices for the i^{th} pair. During inference we predict the span of frames $(\hat{t}_s, \hat{t}_e)$ for each query by maximizing the joint probability of start and end frames.

$$\text{span}(\hat{t}_s, \hat{t}_e) = \underset{\hat{t}_s, \hat{t}_e}{\text{argmax}}\, P_{\text{start}}(\hat{t}_s) P_{\text{end}}(\hat{t}_e) \tag{2}$$

$$= \underset{\hat{t}_s, \hat{t}_e}{\text{argmax}}\, S_{\text{start}}(\hat{t}_s) + S_{\text{end}}(\hat{t}_e) \tag{3}$$

$$\text{s.t. } \hat{t}_s \leq \hat{t}_e \tag{4}$$

ExCL-reg: The classification approach is limited to predicting start and end times of clip at discretized intervals, while the true target is a continuous value. In order to directly model this as a regression problem, we formulate start and end time prediction by computing an expectation over the probability distribution given by SoftMax outputs:

$$t_s = \mathbb{E}_{P_{\text{start}}}[t] \tag{5}$$

$$= \sum_{t_s=1}^{T} t_s P_{\text{start}}(t_s) \tag{6}$$

$$t_e = \mathbb{E}_{P_{\text{start}}}[\mathbb{E}_{P_{\text{end}|\text{start}}}[t]] \tag{7}$$

$$= \sum_{t_s=1}^{T} P_{\text{start}}(t_s) \sum_{t_e=1}^{T} P_{\text{end}|\text{start}}(t_e) \tag{8}$$

Here the above equations t_s, t_e refer to the actual time values corresponding to each index. $P_{\text{end}|\text{start}}(t_e)$ is computed by a SoftMax over masked logits:

$$P_{\text{end}|\text{start}} = \text{SoftMax}(\mathbb{1}[t_e \geq t_s] S_{\text{end}}(t))$$

Finally we train the networks using regression losses such as mean squared error and absolute error. We find that using absolute error and first normalizing the values of time to $t_s, t_e \in [0, 1]$ yields better results.

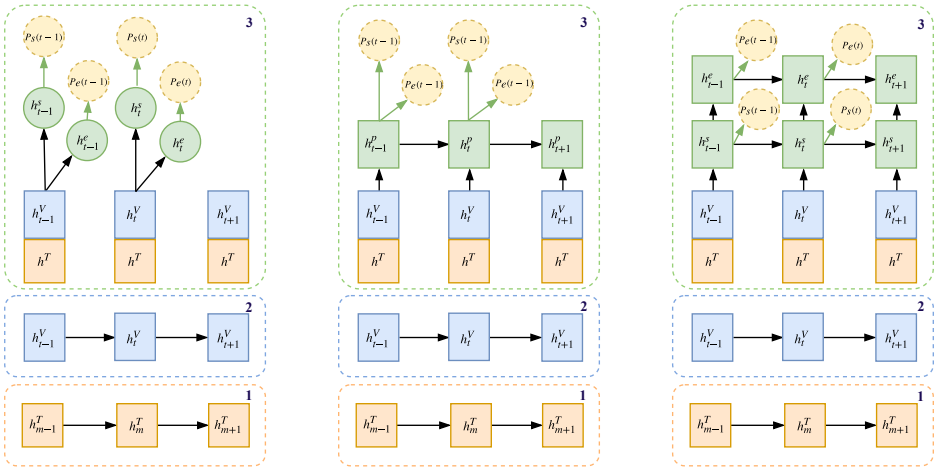
2.2 Span Predictor Variants

We implement the following variants of the span predictor network:

MLP predictor: At each time step t, we pass concatenated video-encoder features with sentence embeddings into two multi-layered perceptrons (MLPs) to obtain scores $S_{\text{start}}(t), S_{\text{end}}(t)$.

$$S_{\text{start}}(t) = \text{MLP}_{\text{start}}([\mathbf{h}_t^V; \mathbf{h}^T]) \tag{9}$$

$$S_{\text{end}}(t) = \text{MLP}_{\text{end}}([\mathbf{h}_t^V; \mathbf{h}^T]) \tag{10}$$



(a) MLP predictor (b) Tied-LSTM predictor (c) Conditioned-LSTM predictor

Figure 2: Our model consists of three modules: a text sentence encoder (**orange**, denoted by [1]), a video encoder (**blue**, denoted by [2]) and three variants of span predictor (**green**, denoted by [3]) - MLP, Tied-LSTM and Conditioned-LSTM to predict start and end probabilities for each frame. We use bidirectional LSTMs for the text and video encoders.

Tied LSTM predictor: Here, we send concatenated video encoder output and sentence embedding to a bidirectional LSTM as input at each step in order to capture recurrent cross-modal interactions. The hidden states ($\mathbf{h}_t^P$) concatenated with the original inputs are then fed to a MLP with $\tanh$ activation in hidden layers to predict start and end scores.

$$h_t^P = \text{LSTM}([\mathbf{h}_t^V; \mathbf{h}^T], \mathbf{h}_{t-1}^P) \quad (11)$$

$$S_{\text{start}}(t) = \text{MLP}_{\text{start}}([\mathbf{h}_t^P; \mathbf{h}_t^V; \mathbf{h}^T]) \quad (12)$$

$$S_{\text{end}}(t) = \text{MLP}_{\text{end}}([\mathbf{h}_t^P; \mathbf{h}_t^V; \mathbf{h}^T]) \quad (13)$$

Conditioned LSTM predictor: Note that in the previous two approaches the end-frame predictor is not conditioned in any way on the start predictor. Here we use two bidirectional LSTMs: LSTM_{end} takes as input the hidden states $\mathbf{h}_t^{P_0}$ of $\text{LSTM}_{\text{start}}$ and produces $\mathbf{h}_t^{P_1}$ as output. The respective hidden states are then used in a similar way as the tied LSTM method to generate start and end scores.

$$\mathbf{h}_t^{P_0} = \text{LSTM}_{\text{start}}([\mathbf{h}_t^V; \mathbf{h}^T], \mathbf{h}_{t-1}^{P_0}) \quad (14)$$

$$\mathbf{h}_t^{P_1} = \text{LSTM}_{\text{end}}(\mathbf{h}_t^{P_0}, \mathbf{h}_{t-1}^{P_1}) \quad (15)$$

$$S_{\text{start}}(t) = \mathbf{W}_s([\mathbf{h}_t^{P_0}; \mathbf{h}_t^V; \mathbf{h}^T]) + \mathbf{b}_s \quad (16)$$

$$S_{\text{end}}(t) = \mathbf{W}_e([\mathbf{h}_t^{P_1}; \mathbf{h}_t^V; \mathbf{h}^T]) + \mathbf{b}_e \quad (17)$$

3 Datasets

We evaluate our models on three datasets. Note that we do not evaluate our models on DiDeMo/TEMPO (Hendricks et al., 2017, 2018) as these datasets only provide coarse fixed-size moments, thus reducing the problem essentially to ranking a fixed set of candidates. We choose the following datasets because each has unique properties such as visual variance, richness of vocabulary and variance in query lengths.

MPII TACoS: This dataset (Rohrbach et al., 2014) has been built on top of the MPII Cooking Activities dataset. It consists of detailed temporally aligned text descriptions of cooking activities. The average length of videos is 5 minutes. A significant challenge in TACoS dataset is that descriptions span over only a few seconds because of the atomic nature of queries such as 'takes out the knife' and 'chops the onion' (8.4% of them are less than 1.6s long). Such short queries allow a smaller margin of error. We use the train/test split as provided by Gao et al. (2017) *here*. Coupled with lesser visual variance to distinguish activities, fine-grained actions and descriptions which can often have high word overlap, this is a challenging dataset.

ActivityNet Captions: ActivityNet (Caba Heilbron et al., 2015) is a large-scale open domain activity recognition, segmentation and prediction dataset based on YouTube videos additionally augmented with dense temporally annotated captions (Krishna et al., 2017). The average length is 2 minutes, but they are much more diverse in content, with videos that span over 200 activity

classes and annotations which use a richer vocabulary. There are 10,024 and 5,044 train and test set videos. We use the same train/test split as provided by the authors. Our reported results have 3,370 missing videos which could not be downloaded.

Charades-STA: The Charades dataset was introduced with action classification, localization and video description annotations (Sigurdsson et al., 2016). Gao et al. (2017) extend this to include sentence level temporal annotation to create the Charades-STA dataset which contains 12,408 training, and 3,720 test sentence level annotations. In Charades-STA, average length of videos is 30 seconds and query length (number of frames) has much lower variance (3.7s) as compared to TACoS (39.5s) and ActivityNet (78.1s).

4 Experiments

4.1 Feature Extraction

We downsample the videos at a frame rate of 5 frames per second and extract I3D RGB (Carreira and Zisserman, 2017) visual features pretrained on Kinetics dataset. We use a fine-tuned version for Charades available *here*.

4.2 Implementation Details

We use pre-trained GloVE embeddings of 300 dimensions. The extracted visual features have 1024 dimensions. For TACoS, Charades-STA, and ActivityNet we use a vocabulary size of 1438, 3720 and 10,000 respectively. We considered batch sizes of 16, 32, 64 and single-layer LSTM hidden sizes of 128, 256, 512. The video and span predictors are bidirectional LSTMs with 256 and 128 hidden units respectively while queries are encoded by a 256-dimensional bidirectional LSTM. The MLP based span predictors have 256 hidden dimensions and use $\tanh$ activation in hidden layers. For all datasets, we train the model with a batch size of 32 for 30 epochs using Adam optimizer with a learning rate of 0.001 and early stopping. We apply a dropout of 0.5 to all the abovementioned LSTMs during training. We measure our model performance primarily using *localization accuracy* which is defined to be intersection over union (IoU) at threshold values of 0.3, 0.5, 0.7 to compare with past work which reports Recall@1, IoU=$\{0.3, 0.5, 0.7\}$.

4.3 Experimentation and Ablative Analysis

We compare the different training objectives (labeled ExCL-clf for classification and ExCL-reg for regression) and evaluate the usefulness of recurrent encoders for video representations by removing the video LSTM (labeled ExCL-clf/reg 1-$\{a, b, c\}$). We also perform ablative analysis of a range of span predictor networks. We compare the performance of our proposed model with three baselines which are the current SOTA for the different datsets. (i) Activity Concepts based Localizer (ACL) proposed in Ge et al. (2019) for TACoS (ii) Segment Proposal Network approach proposed in (Xu et al., 2019) for ActivityNet and (iii) Moment Alignment Network (MAN) via Iterative Graph Adjustment approach proposed in (Zhang et al., 2018) for Charades-STA. While we significantly beat the first two baselines for TACoS and Activity Net respectively, we attain comparable performance with the third for Charades-STA. It is to be noted that since the approach in (Zhang et al., 2018) depends on ranking fixed number of segments in each video, it is not scalable to the other two datasets which have longer videos with greater variance in their lengths.

4.4 Results

Our results in Table 1 confirm our hypothesis that extractive models work better.

We find that across all datasets, models without recurrent architectures (ExCL 1-a) perform significantly worse, demonstrating the importance of temporal context provided by LSTMs in either video encoder (ExCL 2-a) or span predictors (ExCL 1-b). With ExCL 2-a we see that if the visual context is captured well using recurrent architectures in the video LSTM, then even with a MLP span predictor we get a significant improvement in performance. Furthermore, if we add LSTM span predictors along with the video LSTM (ExCL 2-$\{b, c\}$) we obtain an additional boost in performance. However, without a recurrent visual encoder, a recurrent span predictor is essential to capture both uni-modal and cross-modal interactions (ExCL 1-$\{b, c\}$). When comparing the regression learning objective with classification, we do not notice a substantial gain in performance thereby indicating that not much information is lost if the continuous nature of the labels is not considered. In terms of span predictors, tied LSTM generally performs well across all datasets,

	TACoS			Charades-STA			ActivityNet		
IoU	0.3	0.5	0.7	0.3	0.5	0.7	0.3	0.5	0.7
Ge et al. (2019)	24.2	20.0	–	–	30.5	12.2	–	–	–
Xu et al. (2019)	–	–	–	54.7	35.6	15.8	45.3	27.7	13.6
Zhang et al. (2018)	–	–	–	–	46.5	22.7	–	–	–
ExCL-clf 1-a	22.6	12.6	5.1	55.4	30.4	14.8	42.5	23.8	12.1
ExCL-clf 1-b	42.0	25.0	12.3	64.7	43.8	22.1	61.7	40.4	23.0
ExCL-clf 1-c	41.9	25.5	13.6	64.2	43.9	**23.3**	60.7	40.9	23.4
ExCL-clf 2-a	41.7	26.0	12.9	64.6	41.5	20.3	60.4	40.5	23.1
ExCL-clf 2-b	44.2	**28.0**	**14.6**	**65.1**	**44.1**	22.6	61.1	41.3	23.4
ExCL-clf 2-c	**44.4**	27.8	**14.6**	61.4	41.2	21.3	**62.1**	**41.6**	**23.9**
ExCL-reg 1-a	26.2	11.9	4.8	54.7	34.0	14.5	48.4	27.0	11.0
ExCL-reg 1-b	45.2	27.5	12.9	60.1	42.6	21.6	**63.0**	**43.6**	23.6
ExCL-reg 1-c	41.4	24.8	11.4	59.0	43.1	20.7	61.5	42.7	23.4
ExCL-reg 2-a	42.2	27.2	11.7	59.6	41.9	20.2	61.5	41.9	23.3
ExCL-reg 2-b	**45.5**	**28.0**	**13.8**	**61.5**	**44.1**	**22.4**	62.3	42.7	**24.1**
ExCL-reg 2-c	42.3	27.3	12.5	58.0	41.8	20.5	61.4	41.7	22.4

Table 1: Clip Localization Accuracy at IoU = {0.3, 0.5, 0.7} for TACoS, Charades-STA and ActivityNet. Here ExCL-clf represents the classification loss model and ExCL-reg represents the regression loss model. ExCL-{clf/reg} 1-m refer to models run without a video LSTM encoder, while ExCL-{clf/reg} 2-m include the video LSTM. $m = a, b, c$ refer to MLP, tied LSTM and conditioned LSTM span predictor networks respectively.

and any difference with conditioned LSTM is negligible. This benefit is more pronounced when using the regression objective, possibly because conditioning is already captured in the formulation as given by Equation 7.

While Xu et al. (2019) note a significant difference in performance between Charades-STA and ActivityNet captions, performance is similar on both the datasets in our model. We hypothesize that their model fails to work well when there is large variability in the query lengths, as explained in Section 3. We also find TACoS to be a significantly more challenging benchmark, similar to Ge et al. (2019).

5 Conclusion

In conclusion, our main contribution is an extractive model for clip localization based on text queries as opposed to ranking driven approaches used in the past. Our results show that this elegant and fairly simple approach works much better empirically on three very different benchmark datasets, with tied LSTM span predictor generally giving best results. It is to be noted that these datasets previously had three different architectures as their respective SOTA, and our work naturally lays the foundation for training a single generalizable model across datasets and possibly related tasks as the next step. Furthermore, our approach is modular, making it trivial to insert different architectures for the encoders and span-predictors. Other future directions of this work include adding temporal attention in order to handle more complicated temporal references and extending this approach to work for longer, and thereby more challenging videos such as movies.

Acknowledgements

The authors would also like to thank anonymous reviewers for their comments and Junwei Liang for helpful discussions. This research was supported in part by DARPA grant FA8750-18-2-0018 funded under the AIDA program. This work is also supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340.

References

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 961–970.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 162–171.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. *arXiv preprint arXiv:1705.02101*.

Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. 2019. Mac: Mining activity concepts for language-based temporal localization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 245–253. IEEE.

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

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2018. Localizing moments in video with temporal language. *arXiv preprint arXiv:1809.01337*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Kyung-Min Kim, Seong-Ho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. 2018. Multimodal dual attention memory for video story question answering. *Lecture Notes in Computer Science*, page 698713.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*, pages 706–715.

Bingbin Liu, Serena Yeung, Edward Chou, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. 2018. Temporal modular networks for retrieving complex compositional activities in videos. In *European Conference on Computer Vision*, pages 569–586. Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*, pages 184–195. Springer.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer.

Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970.

Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2014. C3D: generic features for video analysis. *CoRR*, abs/1412.0767.

Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-c3d: region convolutional 3d network for temporal activity detection. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 5794–5803.

Huijuan Xu, Kun He, L Sigal, S Sclaroff, and K Saenko. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, volume 2, page 7.

Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. 2016. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687.

Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. 2018. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–87.

Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. 2018. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. *arXiv preprint arXiv:1812.00087*.