TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval - Supplementary File

Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal

University of North Carolina at Chapel Hill
{jielei, licheng, tlberg, mbansal}@cs.unc.edu

Table of Contents

1 Additional TVR Data Details

1.1 Data Collection

TVR Data Collection Procedure. In Fig. 1 we show an overview of TVR data collection procedure. For details of each step, please refer to both our main text Sec. 3.1 and the following text.

Qualification Test. We designed a qualification test with 12 multiple-choice questions and only let workers who correctly answer at least 9 questions participate in our annotation task, ensuring that workers understand our task requirements well. In total, 1,055 workers participated in the test, with a pass rate of 67%. Adding this qualification test greatly improved data quality. In Fig. 2, we show an example question from our qualification test. This particular question is designed to make sure the annotators write relevant and correct descriptions (queries).

Post-Annotation Verification. To verify the quality of the collected data, we performed a post-annotation verification experiment. We set up another AMT task where workers were required to rate the quality of the collected query-moment pairs based on *relevance*, *is the query-moment pair a unique-match*, etc.

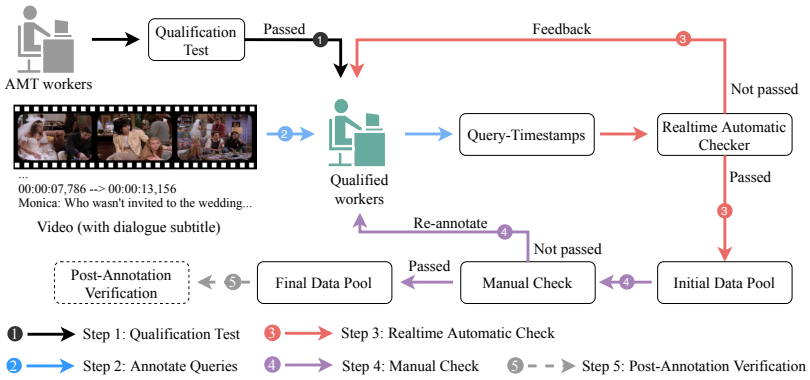

Fig. 1: TVR data collection procedure

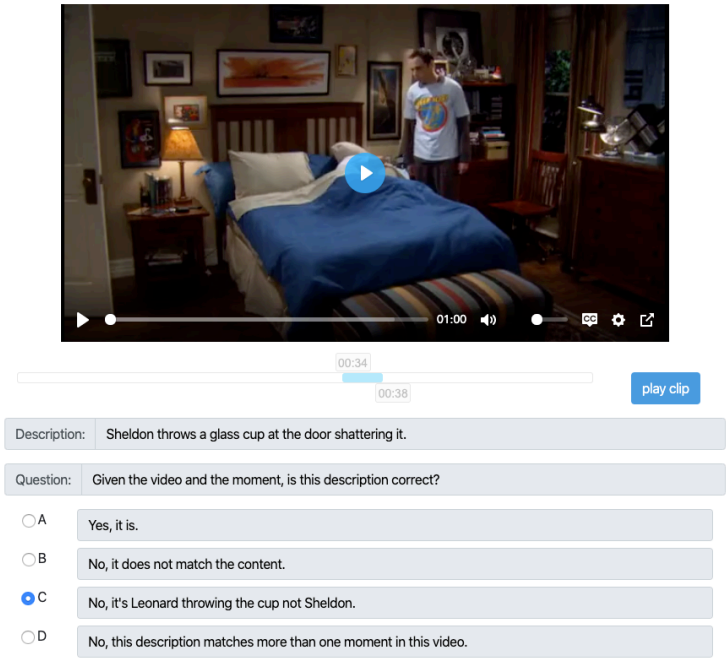

Fig. 2: Example question from our qualification test

The rating was done in a *likert-scale* manner with 5 options: *strongly agree*, *agree*, *neutral*, *disagree* and *strongly disagree*, as is shown in Fig. 3. Results show that 92% of the pairs have a rating of at least *neutral*. This verification was conducted on 3,600 query-moment pairs. Detailed rating distribution is shown in Fig. 4. We further analyzed the group of queries that were rated as *strongly disagree*, and found that 80% of them were still of acceptable quality: e.g., slightly mismatched

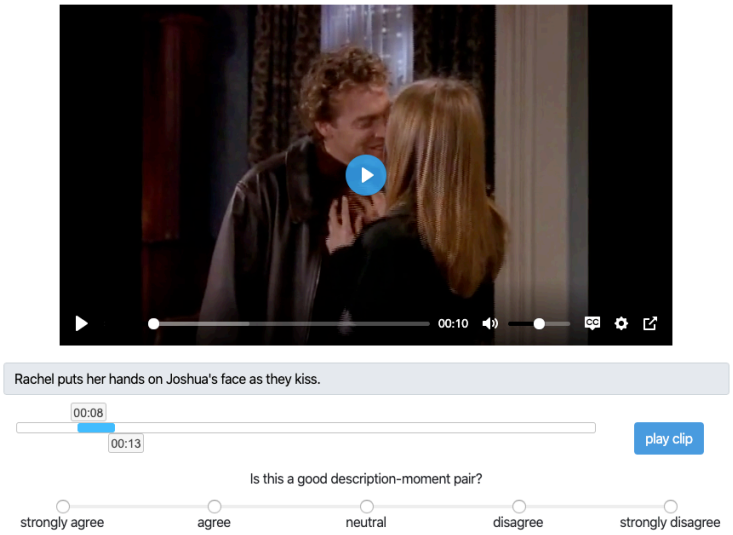

Fig. 3: Post-Annotation quality rating interface

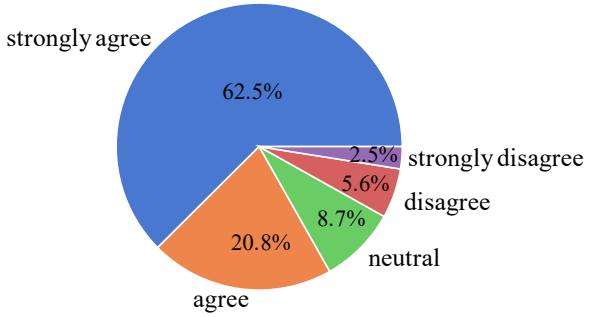

Fig. 4: Quality rating distribution on 3,600 query-moment pairs. 92% of the pairs have a rating of at least *neutral*

timestamps ($\leq$1 sec.). For the group of queries that were rated as *disagree*, this number is 90%. This verification demonstrates the high quality of the data.

1.2 Data Analysis

Statistics by TV Show. TVR is built on 21,793 videos (provided by TVQA [10]) from 6 long-running TV shows: *The Big Bang Theory*, *Friends*, *How I Met You Mother*, *Grey's Anatomy*, *House*, *Castle*. Table 1 shows detailed statistics.

Moments and Queries. Fig. 5 (*left*) shows TVR moment length distribution. The majority of the moments are relatively short, with an average length of 9.1 seconds. As a comparison, the average length of the videos is 76.2 seconds. Fig. 5

Table 1: Data Statistics for each TV show. BBT=*The Big Bang Theory*, HIMYM=*How I Met You Mother*, Grey=*Grey's Anatomy*, Epi=Episode, Sea.=Season

Show	Genre	#Sea.	#Epi.	#Clip	#Query
BBT	sitcom	10	220	4,198	20,990
Friends	sitcom	10	226	5,337	26,685
HIMYM	sitcom	5	72	1,512	7,560
Grey	medical	3	58	1,427	7,135
House	medical	8	176	4,621	23,105
Castle	crime	8	173	4,698	23,490
Total	—	44	925	21,793	108,965

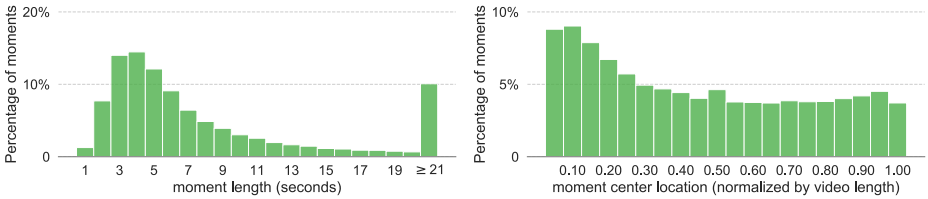

Fig. 5: Distribution of TVR moment lengths (*left*) and moment center locations (*right*)

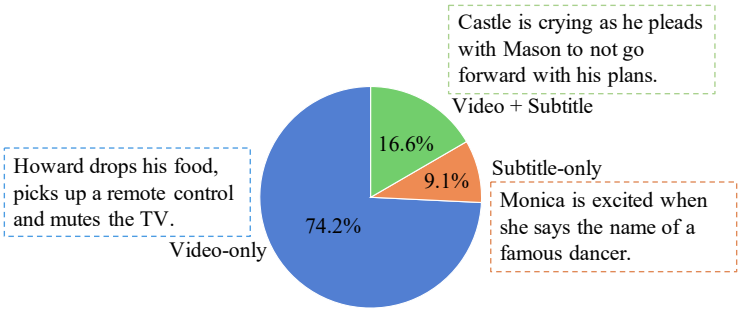

Fig. 6: Query type distribution based on reasoning types. Text inside *dashed boxes* are query examples for each query type

(*right*) shows the video-length normalized moment center distributions. More moments are located at the beginning of the videos. A similar phenomenon was observed in DiDeMo [1]. Fig. 6 shows TVR query type distribution, around 91% of the queries need video context, while 26% of the queries need subtitle context.

Frequent Words in Queries. In Fig. 7 we show frequent nouns (*left*) and verbs (*right*) in TVR queries in the form of word clouds. The words are lemmatized, stop words are removed. We notice that TVR covers a wide range of common

Fig. 7: TVR query word clouds for nouns (*left*) and verbs (*right*)

Table 2: Baseline comparison on TVR *test-public* set, VCMR task. Model references: *MCN* [1], *CAL* [3], *MEE* [12], *ExCL* [5]. This table includes models trained with Temporal Endpoint Feature (TEF) [1]. We show top-2 scores in each column in bold

Model	w/ video	w/ sub.	IoU=0.5				IoU=0.7				Runtime ↓
			R@1	R@5	R@10	R@100	R@1	R@5	R@10	R@100	(seconds)
Chance	-	-	0.00	0.02	0.04	0.33	0.00	0.00	0.00	0.07	-
Frequency	-	-	0.06	0.07	0.11	0.28	0.02	0.04	0.06	0.11	-
Proposal based Methods											
TEF-only	-	-	0.00	0.09	0.15	0.79	0.00	0.07	0.09	0.48	-
MCN	✓	✓	0.02	0.15	0.24	2.20	0.00	0.07	0.09	1.03	-
MCN (TEF)	✓	✓	0.04	0.11	0.17	1.84	0.02	0.06	0.07	1.10	-
CAL	✓	✓	0.09	0.31	0.57	3.42	0.04	0.15	0.26	1.89	-
CAL (TEF)	✓	✓	0.04	0.17	0.31	2.48	0.02	0.15	0.22	1.30	-
Retrieval + Re-ranking											
MEE+MCN	✓	✓	0.92	3.69	5.58	17.91	0.42	1.89	2.98	10.84	-
MEE+MCN (TEF)	✓	✓	1.36	3.89	5.79	19.34	0.62	2.04	3.21	11.66	66.8
MEE+CAL	✓	✓	0.97	3.75	5.80	18.66	0.39	1.69	2.98	11.52	-
MEE+CAL (TEF)	✓	✓	1.23	4.00	6.52	20.07	0.66	1.93	3.09	12.03	161.5
MEE+ExCL	✓	✓	0.92	2.53	3.60	6.01	0.33	1.19	1.73	2.87	-
MEE+ExCL (TEF)	✓	✓	1.01	2.50	3.60	5.77	0.40	1.21	1.73	2.96	1307.2
XML (sliding window)	✓	✓	3.82	10.38	14.20	35.89	1.91	5.25	8.12	23.47	-
XML	✓	✓	**7.25**	**16.24**	**21.65**	**44.44**	**3.25**	**8.71**	**12.49**	**29.51**	-
XML (TEF)	✓	✓	**7.88**	**16.53**	**21.84**	**45.51**	**3.32**	**9.46**	**13.41**	**30.52**	**25.5**

objects/scenes and actions, while also has many genre-specific words such as 'patient' and 'hospital'.

Video Comparison. TVR videos are from 6 TV shows of 3 different genres, which cover a diverse set of objects/scenes/activities. In Fig. 8, we compare TVR videos with videos from existing datasets [14,4,9,1]. Each TVR video typically has more visual diversity, i.e., more camera viewpoints, activities and people, etc.

2 Additional TVR Experiments

2.1 More VCMR Experiments

Frequency Baseline. Following prior works [1,3], we first discretize the video-length normalized start-end points, then use moments with most frequent start-end points as predictions. For video retrieval, we randomly sample videos from

dataset. The results of this baseline is presented in Table 2. We observe this baseline has slightly better performance than chance, we hypothesize it is mainly caused by the fact that the annotators tend to annotate the first few seconds of the video [1], as we shown in Fig. 5 (*Right*).

Models Trained with TEF. It is shown in [1,3] that adding Temporal Endpoint Feature (TEF) [1] improves models' performance in moment retrieval tasks. In Table 2, we compare models trained with TEF. In most cases, adding TEF increases models' performance, which suggests there exists a certain degree bias in the proposed dataset. This phenomenon is also observed by recent works [1,3] in various moment retrieval datasets, i.e., DiDeMo [1], CharadesSTA [4] and ActivityNet Captions [9]. We attribute this phenomenon into two aspects: (1)*moment distribution bias* - the moments are not evenly distributed over the video, e.g., in TVR and DiDeMo [1], there are more moments appear at the beginning of the video. (2)*language timestamp correlation bias* - some query words are highly indicative of the potential temporal location of the queries, e.g., the temporal connectives like 'first' strongly indicates the associated query might be located around the beginning of the video and pronouns like 'He' may suggest this query should not be placed at the beginning of the video as people would usually not use pronouns when they first mention someone. The second bias commonly exists in datasets that are built by converting paragraphs into separate sentences, i.e., CharadesSTA [4], TACoS [14] and ActivityNet Captions [9]. TVR avoids this bias by explicitly ask annotators to write queries as individual sentences without requiring the context of a paragraph.

XML with Sliding Windows. In main text, we compared XML variants with different proposal generation strategies. In Table 2, we further compare XML (sliding window) with MCN/CAL models. For details of this variant, please see main text Sec. 5.3. Compared to the best baseline (MEE+CAL), using the same set of sliding window proposals, we observe XML (sliding window) still perform much better (3.82 *vs.* 0.97, R@1 IoU=0.7). We hypothesize that the lower performance of MCN/CAL models compared to XML (sliding window) is mainly caused by the difficulties of training and ranking with a large pool of proposal candidates (1.5M proposals for TVR *train*). Both MCN and CAL are trained with a ranking objective, which relies on informative negatives to learn effectively. However, effective negative sampling in such a large pool of candidates can be challenging. In comparison, XML breaks the video corpus level moment retrieval problem into two sub-problems: video-level and moment-level retrieval. At video-level retrieval, XML performs ranking within a small set of videos (17.4K), which eases the aforementioned issue. At moment-level, XML (sliding window) utilizes Binary Cross Entropy to maximize the similarity scores of each ground-truth clip, eliminating the need for manually designing a negative sampling strategy.

Model Architecture. Table 3 presents a model architecture ablation. We first compare with different self-encoder architectures, replacing our transformer style encoder with a bidirectional LSTM encoder [10] or a CNN encoder [16,11]. We

Table 3: Model architecture ablation on TVR *val* set, VCMR task. Our full XML model in the last row is configured with transformer encoder and modular query. All models use both videos and subtitles

Model	IoU=0.7			
	R@1	R@5	R@10	R@100
Self-Encoder Type				
XML (LSTM)	2.12	4.97	6.86	18.06
XML (CNN)	2.45	5.53	7.77	19.88
Modular Query				
XML (No modular query)	2.46	5.87	8.56	22.00
XML	**2.62**	**6.39**	**9.05**	**22.47**

Table 4: Feature ablation on TVR *val* set, VCMR task. All models use both videos and subtitles

Model	IoU=0.7			
	R@1	R@5	R@10	R@100
XML (ResNet)	2.28	5.40	7.33	20.28
XML (I3D)	2.22	5.75	8.37	21.20
XML (ResNet+I3D)	**2.62**	**6.39**	**9.05**	**22.47**

observe worse performance after the change and attribute this performance drop to the ineffectiveness of LSTMs and CNNs to capture long-term dependencies [7,15]. Next, we compare XML with a variant that uses a single max-pooled query instead of two modularized queries. Across all metrics, XML performs better than the variant without modular queries, showing the importance of considering different query representations in matching the context from different modalities.

Feature Ablation. We tested XML model with different visual features, the results are shown in Table 4. The model that uses both static appearance features (ResNet [6]) and action features (I3D [2]) outperforms models using only one of the features, demonstrating the importance of recognizing both the objects and the actions in the VCMR task.

Retrieval Efficiency in 1M Videos. We consider Video Corpus Moment Retrieval in a video corpus containing 1M videos with 100 queries. Following [3], we conduct this experiment in a simulated setting with each video containing 20 clips with max moment length of 14 clips. Each query containing 15 words. We report the following metrics: (1) feature encoding time (*feat time*) - measures the time for encoding the context (video and subtitle) features offline. (2) encoded feature size (*feat size*) - measures the disk space needed to store the encoded context features. (3) retrieval time (*retrieval time*) - measures the time needed to retrieve relevant moments for 100 new queries. It includes time for encoding the queries and performing approximate nearest neighbor search [8] or matrix multiplication. The time spent on data loading, pre-processing, feature extraction on backend models (i.e., ResNet-152, I3D, RoBERTa) are not considered as they

Table 5: VCMR on 1M videos with 100 queries. TVR *test-public* set results are included as reference.Model references: *MCN* [1], *CAL* [3], *MEE* [12], *ExCL* [5]

Model	IoU=0.7		Search 100 queries in 1M videos ↓		
	R@1	R@5	feat time (s)	feat size (GB)	retrieval time (s)
Retrieval + Re-ranking					
MEE+MCN	0.42	1.89	131	326	0.090
MEE+CAL	0.39	1.69	841	2,235	0.166
MEE+ExCL	0.33	1.19	-	-	1.435
XML	**3.25**	**8.71**	**29**	**76**	**0.005**

Table 6: Impact of #retrieved videos on TVR *val* set, VCMR task.

Model	#retrieved videos	IoU=0.5				IoU=0.7			
		R@1	R@5	R@10	R@100	R@1	R@5	R@10	R@100
XML	10	5.29	11.82	15.83	31.05	2.62	6.54	9.14	21.19
	50	5.29	11.74	15.92	35.95	2.63	6.40	9.07	22.55
	100	5.28	11.73	15.90	36.16	2.62	6.39	9.05	22.47
	200	5.28	11.73	15.90	36.20	2.62	6.39	9.05	22.46

should be similar if not the same for all the methods. Note that the *retrieval time* here is different from the *runtime* in Table 2, which additional includes *feat time*. We do not report *feat time* and *feat size* for ExCL [5] as it does not have the ability to pre-encode the features - its context encoding depends on the input queries. This experiment was conducted on an RTX 2080Ti GPU and an Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz × 40, with PyTorch [13] and FAISS [8].

The results are shown in Table 5. Our XML model is more efficient than all the baselines. Compared to the best baseline methods MEE+MCN, XML is 18× faster in retrieval, 4.5× faster in feature encoding and needs 77% less disk space to store the encoded features. Besides, it also has 7.7× higher performance (3.25 *vs.* 0.42, IoU=0.7, R@1, on TVR *test-public* set). Note that MEE+ExCL has very poor *retrieval time* performance (287× slower than XML), as it requires early fusion of context and query features. In comparison, the other 3 methods are able to pre-encode the context features and only perform lightweight query encoding and highly optimized nearest neighbor search or matrix multiplication to obtain the moment predictions.

Impact of #Retrieved Videos. In previous experiments, we fix the number of videos retrieved by XML to be 100 for corpus level moment retrieval experiments. To study the impact of this hyperparameter, we perform experiments when #videos $\in [10, 50, 100, 200]$, the results are shown in Table 6. Overall, we notice XML is not sensitive to the number of retrieved videos in terms of R@1, R@5 and R@10 (IoU=0.5, 0.7) in the tested range. When we focus on R@100, IoU=0.5, we find that using more videos is helpful in improving the retrieval performance.

2.2 SVMR and Video Retrieval Experiments

Table 7: Baseline comparison on TVR *val* set, SVMR task. Model references: *MCN* [1], *CAL* [3], *MEE* [12], *ExCL* [5]. We show top-2 scores in each column in bold

Model	w/ video	w/ sub.	IoU=0.5		IoU=0.7	
			R@1	R@5	R@1	R@5
Chance	-	-	3.24	12.79	0.94	4.41
Moment Frequency	-	-	7.72	18.93	4.19	12.27
TEF-only	-	-	9.63	24.86	5.14	14.92
MCN	✓	✓	13.08	39.61	5.06	20.37
MCN (TEF)	✓	✓	16.86	40.55	7.96	21.45
CAL	✓	✓	12.07	39.52	4.68	20.17
CAL (TEF)	✓	✓	17.61	42.08	8.07	21.40
ExCL	✓	✓	**31.34**	47.40	**14.19**	28.01
ExCL (TEF)	✓	✓	31.31	48.54	**14.34**	28.89
XML	✓	✓	30.75	**51.20**	13.41	**31.11**
XML (TEF)	✓	✓	**31.43**	**51.66**	13.89	**31.11**

Table 8: Baseline comparison on TVR *val* set, video retrieval task. Model references: *MCN* [1], *CAL* [3], *MEE* [12]. We show top-2 scores in each column in bold

Model	w/ video	w/ sub.	R@1	R@5	R@10	R@100
Chance	-	-	0.03	0.22	0.47	4.61
MCN	✓	✓	0.05	0.38	0.66	3.59
MCN (TEF)	✓	✓	0.07	0.28	0.51	3.93
CAL	✓	✓	0.28	1.02	1.68	8.55
CAL (TEF)	✓	✓	0.06	0.34	0.63	5.26
MEE	✓	✓	7.56	20.78	29.88	73.07
XML	✓	✓	**16.54**	**38.11**	**50.41**	**88.22**
XML (TEF)	✓	✓	**16.08**	**37.92**	**50.38**	**88.62**

Single Video Moment Retrieval. Table 7 shows the Single Video Moment Retrieval (SVMR) results on TVR *val* set. The goal of the task is to retrieve relevant moments from a single video rather than from a video corpus as in VCMR. We observe XML achieves comparable performance with the state-of-the-art method ExCL [5]. However, note that XML significantly outperforms ExCL on the VCMR task with higher efficiency, as stated in the main text Sec. 5.2 and the supplementary file Sec. 2.1. We also noticed that adding TEF has minimal impact on the performance of XML and ExCL, while greatly improves MCN's and CAL's performance. This is not surprising as XML and ExCL directly model the complete video where the temporal information could be acquired, while MCN and CAL break the video into separate proposals where the temporal information is lost in the process.

Video Retrieval. Table 8 shows the Video Retrieval results on TVR *val* set. The goal of the task is to retrieve relevant videos from a large corpus. As MCN and CAL do not perform whole-video retrieval, we approximate their video retrieval

Table 9: Baseline comparison on DiDeMo [1] *test* set, Video Corpus Moment Retrieval task. Model references: *MCN* [1], *CAL* [3], *MEE* [12]. This table includes models trained with Temporal Endpoint Feature (TEF) [1]. We show top scores in each column in bold

Model	w/ video	IoU=0.5			IoU=0.7		
		R@1	R@10	R@100	R@1	R@10	R@100
Chance	-	0.00	0.10	1.99	0.00	0.02	0.64
Frequency	-	0.02	0.22	2.34	0.02	0.17	1.99
Proposal based Methods							
TEF-only	-	0.05	0.32	2.58	0.03	0.27	2.12
MCN (TEF)	✓	0.88	5.16	26.23	0.58	4.12	21.03
CAL (TEF)	✓	0.97	6.15	28.06	0.66	4.69	22.89
Retrieval + Re-ranking							
MEE+MCN (TEF)	✓	0.53	3.00	6.52	0.46	2.64	6.37
MCN+MCN (TEF)	✓	0.92	4.83	17.50	0.64	3.67	13.12
CAL+CAL (TEF)	✓	1.07	6.45	22.60	0.72	4.86	17.60
CAL+CAL (TEF,re-train)	✓	1.29	6.71	22.51	0.85	4.95	17.73
Approx. CAL+CAL (TEF,re-train)	✓	1.27	6.39	15.82	0.80	4.95	11.59
XML (TEF)	✓	**2.26**	**10.42**	**34.49**	**1.59**	**6.71**	**25.44**

predictions using the videos associated with the top-retrieved moments, as in [3]. MCN and CAL models perform rather poor (>50x lower performance than XML, R@1) on the video retrieval task, we summarize some possible reasons here: (1) MCN and CAL's video retrieval results are only an approximation as they are trained to differentiate moments rather than videos; (2) they need to rank a large number of proposals (187K proposals in TVR *val* set), which has many drawbacks, e.g., inefficient negative sampling in training. MEE gets less than half of XML's performance as it uses global pooled context features instead of more fine-grained local context features as XML.

2.3 More Qualitative Examples

We show more qualitative examples from our XML model in Fig. 9 and Fig. 10. We show top-3 predictions for the VCMR task, as well as associated predictions (with ConvSE filter responses) for the SVMR task.

3 DiDeMo Experiments

To show the effectiveness of XML for the video corpus moment retrieval task, we also tested it on the popular moment retrieval dataset DiDeMo [1]. Different from TVR experiments, we only use ResNet features for DiDeMo. Besides, we also switch off the subtitle channel as DiDeMo has only videos as context. The results are shown in Table 9. The baseline results are directly taken from [3]. We observe XML outperforms all the baseline methods on DiDeMo dataset by a large margin, showing XML is able to generalize well to datasets where only video is available.

Table 10: TVR data split detail

Split	#queries	#moments	#videos
train	87,175	87,175	17,435
val	10,895	10,895	2,179
test-public	5,445	5,445	1,089
test-private	5,450	5,450	1,090

4 Data Release and Public Leaderboards

TVR dataset and code are publicly available: `https://tvr.cs.unc.edu/`. Besides, we also extended TVR by collecting extra descriptions for each annotated moment. This dataset, named TV show Captions (**TVC**), is a large-scale multimodal video captioning dataset, with 262K captions. For TVC dataset, please check TVC website at `https://tvr.cs.unc.edu/tvc.html`. With the datasets, we host public leaderboards at the website to better compare the systems. In the following, we describe data split and usage in detail.

We split TVR into 80% *train*, 10% *val*, 5% *test-public* and 5% *test-private* such that videos and their associated queries appear in only one split. This setup is the same as TVQA [10]. Details of the splits are presented in Table 10. *test-public* will be used for a public leaderboard, *test-private* is reserved for future challenges. *val* set should only be used for parameter tuning, it should not be used in the training process in any means, including but not limited to pre-train the language features.

Fig. 8: Video comparison of TVR with existing moment retrieval datasets [14,4,9,1]. Ground truth moment is shown in *green box*. We see each TVR video is typically more diverse, containing more camera viewpoints, activities and people, etc.

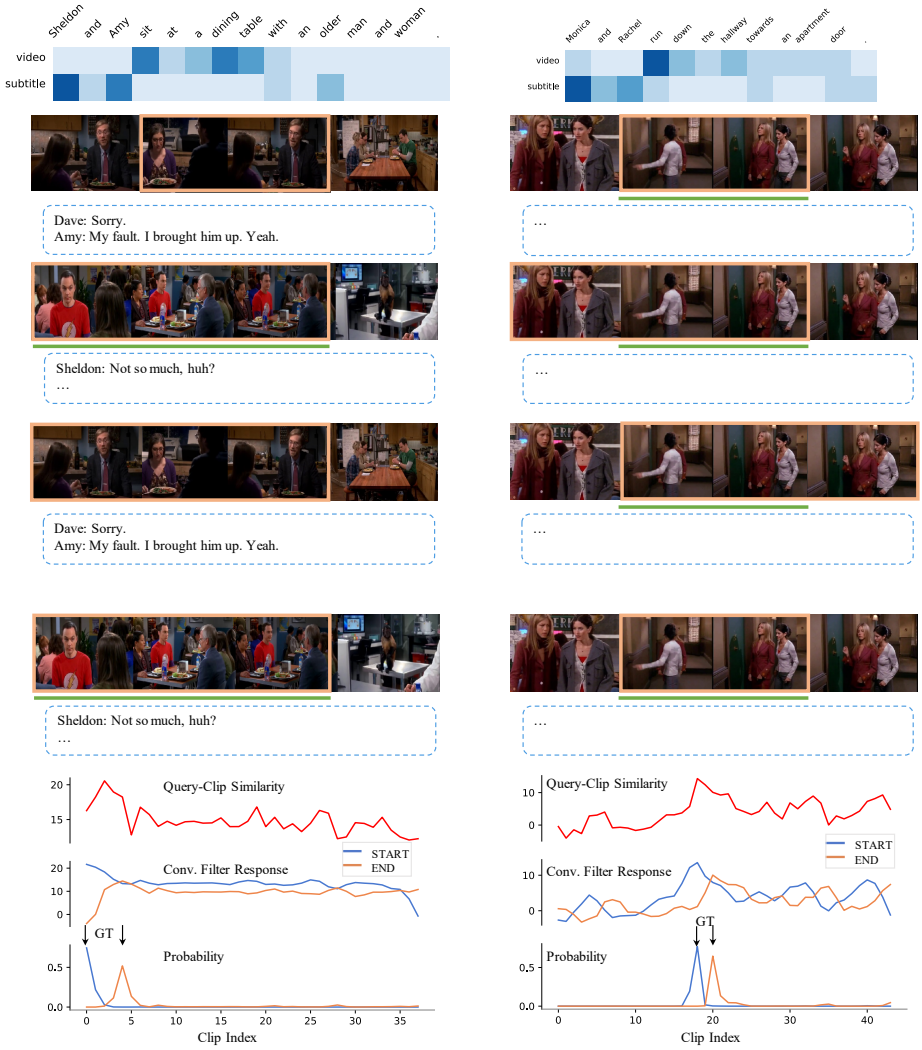

Fig. 9: Qualitative examples of XML. We show top-3 retrieved moments for VCMR (*top*) and SVMR results (*bottom*, with convolution filter responses) for each query. Text inside *dashed boxes* is the subtitles with the predicted moments. *Orange box* shows the predictions, *green bar* shows the ground truth. Best viewed in color

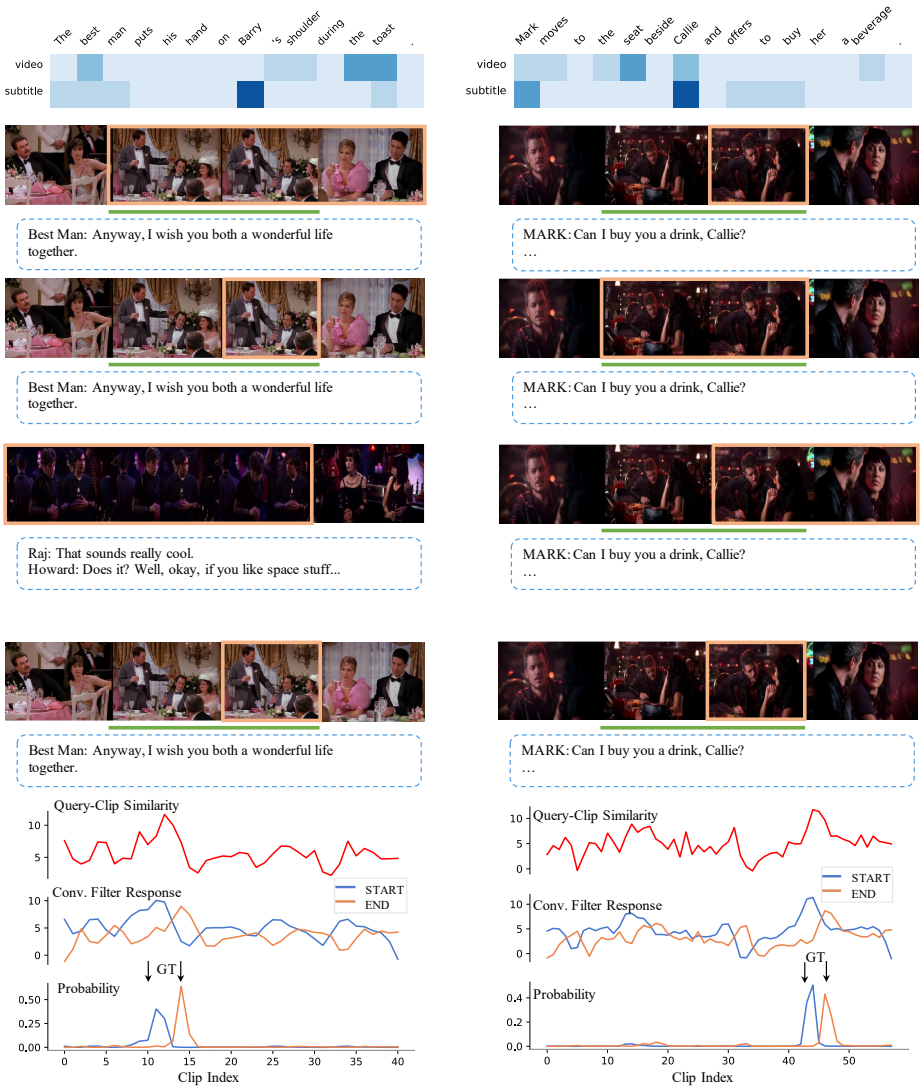

Fig. 10: Qualitative examples of XML. We show top-3 retrieved moments for VCMR (*top*) and SVMR results (*bottom*, with convolution filter responses) for each query. Text inside *dashed boxes* is the subtitles with the predicted moments. *Orange box* shows the predictions, *green bar* shows the ground truth. Best viewed in color

References

1. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: ICCV (2017)
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
3. Escorcia, V., Soldan, M., Sivic, J., Ghanem, B., Russell, B.: Temporal localization of moments in video collections with natural language. arXiv preprint arXiv:1907.12763 (2019)
4. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: ICCV (2017)
5. Ghosh, S., Agarwal, A., Parekh, Z., Hauptmann, A.: Excl: Extractive clip localization using natural language descriptions. In: NAACL (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
7. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
8. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data (2019)
9. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: ICCV (2017)
10. Lei, J., Yu, L., Bansal, M., Berg, T.L.: Tvqa: Localized, compositional video question answering. In: EMNLP (2018)
11. Lei, J., Yu, L., Berg, T.L., Bansal, M.: Tvqa+: Spatio-temporal grounding for video question answering. arXiv preprint arXiv:1904.11574 (2019)
12. Miech, A., Laptev, I., Sivic, J.: Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516 (2018)
13. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
14. Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., Pinkal, M.: Grounding action descriptions in videos. TACL (2013)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
16. Yu, A.W., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Norouzi, M., Le, Q.V.: Qanet: Combining local convolution with global self-attention for reading comprehension. In: ICLR (2018)