TVR-Ranking: A Dataset for Ranked Video Moment Retrieval with Imprecise Queries

Renjie Liang*
liang.renjie@ufl.edu
Nanyang Technological University
Singapore, Singapore

Chongzhi Zhang
chongzhi001@e.ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Li Li*
li.li02@usc.edu
Nanyang Technological University
Singapore, Singapore

Jing Wang
jing005@e.ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Xizhou Zhu
zhuxizhou@sensetime.com
SenseTime Research
Beijing, China

Aixin Sun†
axsun@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Abstract

In this paper, we propose the task of *Ranked Video Moment Retrieval* (RVMR) to locate a ranked list of matching moments from a collection of videos, through queries in natural language. Although a few related tasks have been proposed and studied by CV, NLP, and IR communities, RVMR is the task that best reflects the practical setting of moment search. To facilitate research in RVMR, we develop the TVR-Ranking dataset, based on the raw videos and existing moment annotations provided in the TVR dataset. Our key contribution is the manual annotation of relevance levels for 94,442 query-moment pairs. We then develop the $NDCG@K, IoU \geq \mu$ evaluation metric for this new task and conduct experiments to evaluate three baseline models. Our experiments show that the new RVMR task brings new challenges to existing models and we believe this new dataset contributes to the research on multi-modality search. The dataset is available at https://github.com/Ranking-VMR/TVR-Ranking.

CCS Concepts

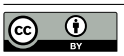
• **Information systems** → **Video search**; *Evaluation of retrieval results*.

Keywords

Ranked Video Moment Retrieval, Video Moment Ranking

ACM Reference Format:
Renjie Liang, Chongzhi Zhang, Li Li, Jing Wang, Xizhou Zhu, and Aixin Sun. 2025. TVR-Ranking: A Dataset for Ranked Video Moment Retrieval with Imprecise Queries. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP 2025), December 7–10, 2025, Xi'an, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3767695.3769516

*Renjie Liang is currently with University of Florida, Gainesville, United States, and Li Li is with University of Southern California, Los Angeles, United States.
†Aixin Sun is the corresponding author.



SIGIR-AP 2025, Xi'an, China

ACM ISBN 979-8-4007-2218-9/2025/12
https://doi.org/10.1145/3767695.3769516

1 Introduction

Given a query expressed in natural language, to retrieve or to locate a temporal moment from video(s) that semantically matches the query has many applications. A temporal moment refers to a segment within a source video with identified start and end timestamps. Examples of such applications include searching for a specific scene in security surveillance videos [25], locating a medical procedure within an educational tutorial [7], or identifying desired scenes for video editing purposes, among others.

A few tasks with different names have been studied for addressing similar objectives, including video retrieval (VR), video moment retrieval (VMR), natural language video localization (NLVL), temporal sentence grounding in video (TSGV), and video corpus moment retrieval (VCMR) [15, 28]. Among them, VR involves retrieving a video from a collection based on visual content, akin to video search on platforms like YouTube, but with the search criteria grounded in the visual content of videos. NLVL and TSGV, more commonly used in CV and NLP communities, refer to the same task as video moment retrieval (VMR) in the IR community. VMR aims to locate a moment within a given video that semantically matches the text query. The VCMR task is a direct extension of VMR, focusing on retrieving a moment from a collection of videos [22]. As depicted in Figure 1, existing tasks VR, VMR, and VCMR, all aim to find one answer, being either a video or a moment, for a given query.

The reason for expecting one exact answer to a query in existing datasets lies primarily in the annotation of benchmark datasets. During annotation, annotators watch a video, then provide textual descriptions of meaningful video moments in this video. Subsequently, each description serves as the query to retrieve the corresponding moment from this source video. Given that a query typically describes a specific moment precisely, a model trained on these datasets can assume the existence of the moment to be searched for, and all queries are from users who possess a good understanding of the source video.

In a practical setting, there exist multiple moments that can be described similarly, even in a single video. For example, one video may contain multiple moments for "Phoebe enters room and sits on sofa", or a very similar moment "Alice enters room and sits on sofa". If we assume a user has limited knowledge about the source video, then he/she may formulate a query like "a woman enters room and sits on sofa". In this case, all the moments that correspond to either

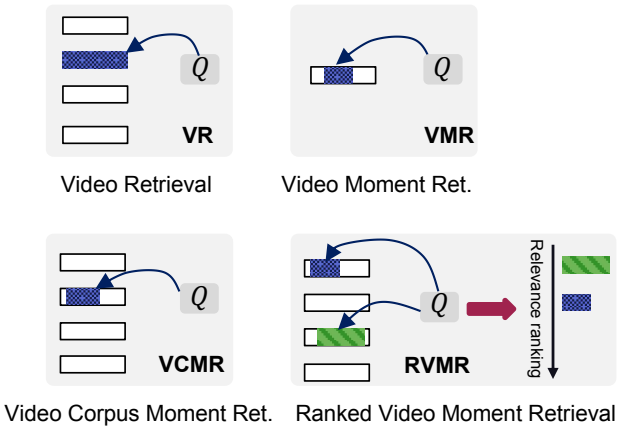


Figure 1: RVMR and its related tasks. A rectangle represents a video; the matching moment to text query Q is shaded. In VR VMR and VCMR, exactly one video/moment is retrieved.

Phoebe or Alice entering a room and sitting on sofa are perfect matches. Further, there could be relevant but non-perfect matches like "a man enters room".

In this paper, we define a new task named *Ranked Video Moment Retrieval* (RVMR) to better reflect the practical setting. **RVMR** is to retrieve a ranked list of moments matching a query from a collection of videos. We do not assume that users have fully watched all videos to be searched. Hence, users do not have a specific expectation for the retrieved results. Multiple moments from the same or different videos can be retrieved and ranked by their degrees of relevance to the query. Compared to existing tasks, RVMR exhibits two distinct characteristics: (i) it retrieves list of moments instead of a single moment, and (ii) the retrieved moments are ranked by their relevance to the query. While models designed for VCMR can potentially be repurposed for RVMR, they may lack the necessary moment ranking capability.

To date, no datasets cater to this novel task setting. Thus, we have curated the **TVR-Ranking** dataset to facilitate the RVMR task. As its name suggests, the dataset has its root in the TVR dataset [13]. Specifically, we reuse the source videos in TVR, i.e., video clips from six TV series, and the original moment annotations i.e., the begin/end timestamps of meaningful moments. We have made two main efforts in the development of the TVR-Ranking. The first is deriving queries from the original moment descriptions, which are less precise than their original versions. Therefore, we also refer to these newly derived queries as 'imprecise queries'. In the TVR dataset, moment descriptions are very detailed and often contain TV character names. These character names limit the query to matching only a specific moment. We substitute character names with pronouns using carefully crafted prompts to ChatGPT, with follow up quality control. This process entails replacing a total of 160,701 words across 72,842 moment descriptions. Our second effort involves *annotating the relevance scores of moments* to these imprecise queries. This task was undertaken by 23 annotators over 1,200 working hours. We have annotated relevant moments for 3,281 queries. Among them, 52% of queries were each annotated with 20 candidate moments, featuring five relevance levels ranging from irrelevant (0) to a perfect match (4). The remaining 48% of queries were each annotated with 40 candidate moments. The annotation of the relevance score for a candidate moment to a query (i.e., a query-moment pair) was conducted by either two annotators (in case of consensus can be reached by the two) or four annotators. A total of 94,442 relevance scores were annotated with consensus. The annotated queries are divided into 500 validation and 2,781 test queries. Due to the high cost of annotation, we use sentence similarity between the query and the moment caption (i.e., the imprecise version of moment description), as a proxy to generate pseudo-annotations as the training set.

For evaluating the retrieved results, the prior tasks VMR and VCMR only involve calculating the accuracy of predicted temporal boundaries for single moments. Therefore, Intersection over Union (IoU) [18] has been a suitable metric for these tasks. However, the results of RVMR are presented as a ranked list, requiring consideration of the quality of the ranking in addition to moment localization. To this end, we propose a new metric, $NDCG@K, IoU \geq \mu$. This metric evaluates the quality of the result from both the accuracy of moment localization, and the quality of moment ranking.

Finally, we adapted three baseline models for the RVMR task, and these models were trained with different pseudo-training datasets. Our contributions in this resource paper are summarized as follows. First, we **define the RVMR task** to reflect the practical scenario of retrieving moments from video collections using imprecise queries. Second, we have **annotated the TVR-Ranking** which provides 3,281 queries and their relevance annotations, each with 20 or 40 candidate moments. Third, we propose an **evaluation metric** for the new task named $NDCG@K, IoU \geq \mu$. This metric builds upon $NDCG$ [9], designed for ranking tasks, by incorporating the IoU metric to handle partial matches between the retrieved and the ground truth moments. Lastly, we **adapt three baseline models** (initially designed for VCMR) to RVMR, and evaluate their performance on the TVR-Ranking. Our results suggest that models effective on VCMR may not perform well on RVMR.

2 Related Work

Current VMR and VCMR datasets fail to simulate real-world moment search scenarios due to two key unrealistic assumptions: users have a deep understanding of the source video and there is only one "perfect match" moment for each query.

The first assumption stems from traditional annotation processes where annotators are required to watch the entire video and then describe meaningful moments therein. Most datasets listed in Table 1 are annotated in this way, including DiDeMo [2], TACoS [17], TVR [13], ActivityNet Caption [11], Ego4D(NLQ) [5], and UCA [25]. Besides, the Charades-STA dataset [3] extends the Charades dataset [20] by segmenting video descriptions into sentences and linking them to specific video timestamps via keywords. In contrast, queries in our dataset may or may not provide precise descriptions of moments, thus embracing users with different levels of understanding of the corpus.

Typically, standard VMR datasets generally link a query to a single relevant moment. For instance, the health-related queries in the MedVidQA dataset [6], sourced from WikiHow's 'Health'

Table 1: Existing VMR datasets and our annotated TVR-Ranking dataset. 'M.Dur' and 'V.Dur' mean the total duration of moments and videos, respectively, in hours. TVR-Ranking, as a dataset for retrieval like web search, all queries in training, validation, test sets share the same video collection. In the dataset, we select the top $N = 40$ moments by query-caption similarity for a query, to comprise the pseudo training set. In the annotated validation and test sets, the average number of relevant moments per query is 27. For all other datasets, the matching moment per query is 1.0 except QVHighlight which is 1.78.

Dataset	#Query	#Moment	#Video	Vocab.	#Verb	M.Dur	V.Dur
Charades-STA [3]	16,128	11,770	6,672	1,303	469	26.33	56.69
ActivityNet Captions [11]	54,559	54,559	14,926	13,645	4,510	560.37	487.60
TACoS [17]	18,227	7,069	127	2,287	994	2220.82	10.11
MAD (v2-unnamed) [21]	3,328,745	328,742	488	56,066	13,993	263.46	1,207.3
Ego4D(NLQ) [5]	18,399	18,374	1,685	3,337	823	56.85	231.82
UCA [25]	19,211	18,299	1,544	4,087	1,580	84.39	96.71
MedVidQA [6]	3,010	2,990	899	2,291	670	51.78	95.72
QVHighlight [12]	10,310	18,367	10,148	7,750	1,824	125.50	422.83
TVR [13]	98,070	97,442	19,614	18,856	6,104	240.70	414.86
TVR-Ranking (Training)	69,317	94,259	19,614	10,865	3,846	228.91	414.86
TVR-Ranking (Validation)	500	12,191	19,614	994	330	29.48	414.86
TVR-Ranking (Test Set)	2,781	42,472	19,614	2,517	837	101.60	414.86

category, are well-represented of real-world scenarios. Nonetheless, this dataset confines each query to just the most relevant moment. In contrast, real-life situations frequently encompass multiple moments that can be similarly described. The QVHighlight dataset [12] was pioneering in allowing queries to match multiple moments within a single video. However, it still restricts searches to single videos and focuses only on perfect matches. Our dataset aims to retrieve a ranked list of moments from a video corpus based on imprecise queries, functioning more like a search engine and accommodating both closely and loosely relevant matches. This paradigm not only broadens the utility of the results but also aligns more closely with practical search needs.

3 TVR-Ranking Dataset

In the ideal setting, a dataset shall well reflect the context of real-world applications, e.g., the data source and the information needs from users [24]. In the RVMR task setting, we assume there exists a collection of videos, and users search for relevant moments through textual descriptions as queries. However, such kinds of queries can only be collected from logs of video search services, which are not publicly accessible. Without access to such resources, we choose to derive user queries from existing data annotations, i.e., the datasets listed in Table 1.

Our immediate task is to choose which existing dataset to use as the raw data for annotation. To this end, we compare the existing datasets based on the following perspectives: accessibility to the raw videos, number of videos, variants of different activities/scenes, and number of moments annotated. Because existing annotations are mostly descriptions of scenes and/or actions, the number of verbs has been widely used to measure the number of activities covered in a dataset [19]. Based on these considerations, we adopt the TVR dataset as the raw source for our annotation; accordingly, our dataset is named **TVR-Ranking**.

The TVR dataset contains video clips from six different TV series, along with their corresponding subtitles and audio tracks. In the construction of the original TVR dataset, annotators were tasked with identifying the boundaries of events i.e., moments, within these clips, and describing their content. Workers also provided whether the description was purely based on the visual content, the subtitle, or both the video and subtitle. The TVR dataset comprises 72,842 video-only, 8,920 subtitle-only, and 16,308 video-subtitle moment descriptions.[1] In our TVR-Ranking construction, we aim to concentrate on the visual aspects; therefore, we only consider the annotations that are purely based on the visual content of the videos.

3.1 Query Construction via Rewriting

In TVR dataset, many moment descriptions contain character names and even their dressing details, making them precise descriptions of the moments. To make these queries match more relevant moments, we replace specific words with more general terms. In particular, we replace all character names with pronouns. This replacement is essential for our annotation because our annotators (also users) may not have knowledge about these characters. Table 2 lists three example descriptions before and after substitution.

The character name substitution is through carefully designed prompts to ChatGPT, with quality checks. The output of ChatGPT is a quality substitution if it successfully passes two validation checks. The first check is for semantic consistency, to ensure no significant change in terms of semantic meaning after substitution. The SimCSE [4] similarity of moment descriptions before and after the substitution is expected to be above a threshold (0.4 in our implementation). The second check is to ensure no person names appear in the substituted version. We detect person names in the substituted moment description using Flair [1]. If a substitution fails to pass both checks, the moment description undergoes human review and is fed to ChatGPT again for another substitution with a different temperature parameter setting, till it passes both checks.

The above procedure replaces 160,701 words across 72,842 moment descriptions, averaging 2.21 words per description. Table 3

[1]The numbers reported here are from the dataset version used in our annotation, with negligible differences from those reported in the original paper [13].

Table 2: Three example moment descriptions before and after word substitution.

No.	Original query before word substitution	Query after word substitution
1.	*Eric and Dr. Gregory* were having a conversation.	Two people were having a conversation.
2.	*Rachel Green and Ross* were having a conversation.	Two people were having a conversation.
3.	*Javier and the young man wearing checkered polo* was having a conversation.	Two people were having a conversation.

Table 3: The top 12 most frequent replacement words. M, F, and N denote male, female, and gender-neutral, respectively.

Word	Freq.	Gender	Word	Freq.	Gender
man	37,533	M	some	1,747	N
woman	34,708	F	doctor	1,263	N
person	28,109	N	other	1,149	N
two	6,426	N	they	1,064	N
people	6,035	N	her	572	F
someone	1,938	N	guy	533	M

presents the top 12 most frequently replaced words, which shows a diversity of personal pronouns in the descriptions.

To distinguish the moment descriptions before and after the substitution, we call the substituted version **moment caption**. As shown in Table 2, the three original moment descriptions — though referring to different character instances — are rewritten into the same imprecise caption: "Two people were having a conversation." This transformation demonstrates two key properties. First, it reflects our intended goal of semantic abstraction, where the essential event is retained while character-specific details are removed. Second, it increases the likelihood that multiple semantically similar but visually distinct moments are matched to the same query, thus enhancing the dataset's support for ranked multi-moment retrieval.

3.2 Relevance Annotation and Quality Control

From the 72,842 moment captions, we randomly select 500 and 2,781 moment captions as queries for the validation and test sets, respectively. The remaining moment captions are reserved for constructing the pseudo training set (see Section 3.4). As a retrieval task, all queries share the same large pool of source videos.

Next, we manually annotate ground truth moments, along with their degree of relevance, for both the 500 validation and 2781 test queries. The annotation pipeline is shown in Fig. 2. Manually annotating all matching moments from such a large video corpus for a given query is infeasible. Therefore, we utilize the moment annotations available in the original TVR dataset during the annotation process. These original annotations serve two purposes. First, we fully rely on the temporal boundaries of all moments in the original TVR annotations. This approach allows us to view the video corpus as a vast collection of moments each accompanied by a moment caption, during the data annotation process. Second, the moment caption provides a reasonably good description of a moment. Consequently, the semantic similarity between a query and a moment caption acts as a proxy for an initial estimation of their relevance.

Let $m.c$ and $m.v$ represent the caption and the visual content of moment m, respectively. To annotate the ground truth moments

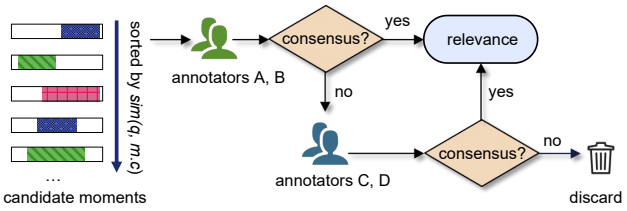


Figure 2: The annotation pipeline of scoring the relevance of moments for a given query. We first select the top-k candidate moments based on $sim(q, m.c)$, i.e., similarity between the query and a moment caption. Two annotators then score these moments. A consensus is reached if their relevance scores differ by 0 or 1. Moments without consensus are re-scored by another pair of annotators and discarded if no consensus is reached.

for a query q, we initially retrieve the top-K moment candidates based on their similarity to q by using SimCSE [4], denoted by using $sim(q, m.c)$. In our annotation process, we set $K = 20$ for the first batch. This batch of 20 query-moment pairs is then presented to two annotators.[2] Each annotator independently labels the degree of relevance of every query-moment pair $\langle q, m.v \rangle$ purely based on the moment's visual content, assigning a score from 0 for irrelevant, to 4 for a perfect match.

If the difference between the two relevance scores assigned by two annotators is either 1 or 0, then we considered the two annotators to have reached a consensus. The average relevance score was then rounded up to the nearest whole number as the final score for this query-moment pair. If the two annotators fail to reach a consensus, the same query-moment pair is assigned to another two annotators. Then we have a total of 4 scores. Among the 4 scores, we remove one highest score and one lowest relevance score. If the difference between the remaining two scores is either 1 or 0, we consider a consensus is reached; the average of the remaining two scores is rounded up to the nearest whole number as the final score. Otherwise, the pair is discarded.

After annotating all 20 moments for a query in the first batch, the lead annotator (the first author) checks the relevance score distribution of these 20 candidate movements. Recall that the 20 candidate movements are ranked by $sim(q, m.c)$. Movements ranked in the top few positions are likely to be more relevant than those ranked lower. However, if the last 5 candidates among the 20 remain very relevant, then it is a strong indication that the annotation so far has not fully covered all matching moments. The next batch of 20

[2]The query q is the same in the batch of 20 query-moment pairs $\langle q, m.v \rangle$. However, during annotation, an annotator is presented with one query-moment pair each time through the annotation interface.

candidate moments will be retrieved by $sim(q, m.c)$ for annotation. We observe that we can cover all relevant movements for nearly every query after annotating the second batch, totaling 40 candidate moments. Hence, at most, two batches or 40 candidate moments are annotated for a query.

At the completion of the annotation process, we obtained a total of 9,272 valid annotations for the 500 validation queries and 18,146 annotations for the 2,781 test queries. This resulted in a total annotation cost of approximately 13,000 USD for around 1,200 working hours contributed by 23 annotators, excluding the lead annotator's effort. All annotators underwent a tutorial and qualifying exam before participating in the annotation task.

3.3 Annotation Quality and Similarity-Based Sampling Analysis

In TVR-Ranking, we aim to retrieve all relevant video moments for a given query. However, annotating the entire video corpus is infeasible. Thus, we leverage the moment captions in the original TVR dataset as proxies for visual content, using their semantic similarity to the query, $sim(q, m.c)$, as a heuristic for selecting candidate moments. All annotations are performed based on visual inspection of the moment content $m.v$, but moment captions $m.c$ guide candidate selection.

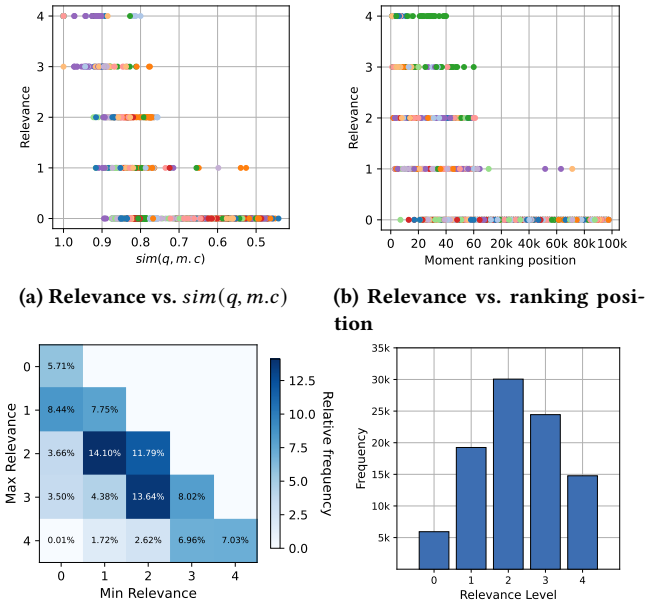
To validate this similarity-based sampling strategy and understand annotation consistency, we conduct an in-depth analysis using 10 randomly sampled queries. For each query, we annotate 120 candidate moments: the top 60 ranked by $sim(q, m.c)$ and 60 randomly sampled from the remaining moments. Figures 3a and 3b show that higher $sim(q, m.c)$ scores are correlated with higher annotated relevance scores $rel(q, m.v)$, and most relevant moments rank within the top 40, justifying our annotation cutoff.

In parallel, we examine the consistency of annotations. Each query-moment pair is annotated by two or four annotators, and relevance scores range from 0 (irrelevant) to 4 (perfect match). Figure 3c presents the raw score range distributions, showing that the vast majority of annotations have a score difference of 0 or 1, indicating strong consensus. Figure 3d shows the distribution of final relevance scores after applying consensus rules. Most moments have intermediate relevance (score = 2), while extreme scores are less frequent. These findings support the reliability of our annotation protocol and the use of similarity-based candidate selection.

3.4 Pseudo Training Set Generation

Due to the high annotation cost, we do not manually annotate training data. Instead, we rely on the query-caption similarity, i.e., $sim(q, m.c)$, as a proxy to generate pseudo annotations as the training set. Specifically, given a query, we collect the top-N moments based on $sim(q, m.c)$ as the training set. In our dataset, we include pseudo training sets with $N = 1$, $N = 20$, and $N = 40$. Datasets with other values of N can be easily generated as well.

As shown in Table 2, after the substitution, two moment descriptions may become identical. To ensure all queries in the validation and tests do not appear in training, we remove from the pseudo training set the queries that appear either in validation or test, a total of 244 queries. As a result, the pseudo training set contains a total of 69,317 queries.



(a) Relevance vs. $sim(q, m.c)$

(b) Relevance vs. ranking position

(c) Raw score min-max heatmap

(d) Final score distribution

Figure 3: Analysis of annotation quality and the effectiveness of query-caption similarity. (a) Relevance scores increase with higher $sim(q, m.c)$. (b) Relevant moments are mostly ranked in the top positions. (c) Most annotators agree within 1-point difference. (d) Final relevance scores follow a bell-shaped distribution.

Table 4: Statistics of TVR-Ranking. The average ratio of the moment duration to the entire video duration. 'M-V Duration Ratio' refers to the ratio of the video's duration that is occupied by the moment. Statistics regarding video durations pertain to videos that contain moments matching the queries in the specific set.

Statistic	Pseudo Training Set (N=40)	Validation Set	Test Set
Min. Query Length	4	7	6
Avg. Query Length	13.98	14.11	13.97
Max. Query Length	122	35	108
Min. Moment Duration (s)	0.26	0.27	0.26
Avg. Moment Duration (s)	8.74	8.71	8.61
Max. Moment Duration (s)	239.38	121.86	138.02
Min. Video Duration (s)	2.02	2.02	2.02
Avg. Video Duration (s)	76.14	76.59	76.23
Max. Video Duration (s)	272.02	272.02	272.02
Avg. M-V Duration Ratio	0.12	0.11	0.11
Avg. Rel. Moments per Query	N/A	27.1	27.0

3.5 Dataset Statistics for TVR-Ranking

Table 4 provides an overview of the annotated TVR dataset, with query length in number of words, moment duration in seconds, and the source video duration in seconds. The average ratio of moment length to its source video length is about one-tenth. In the

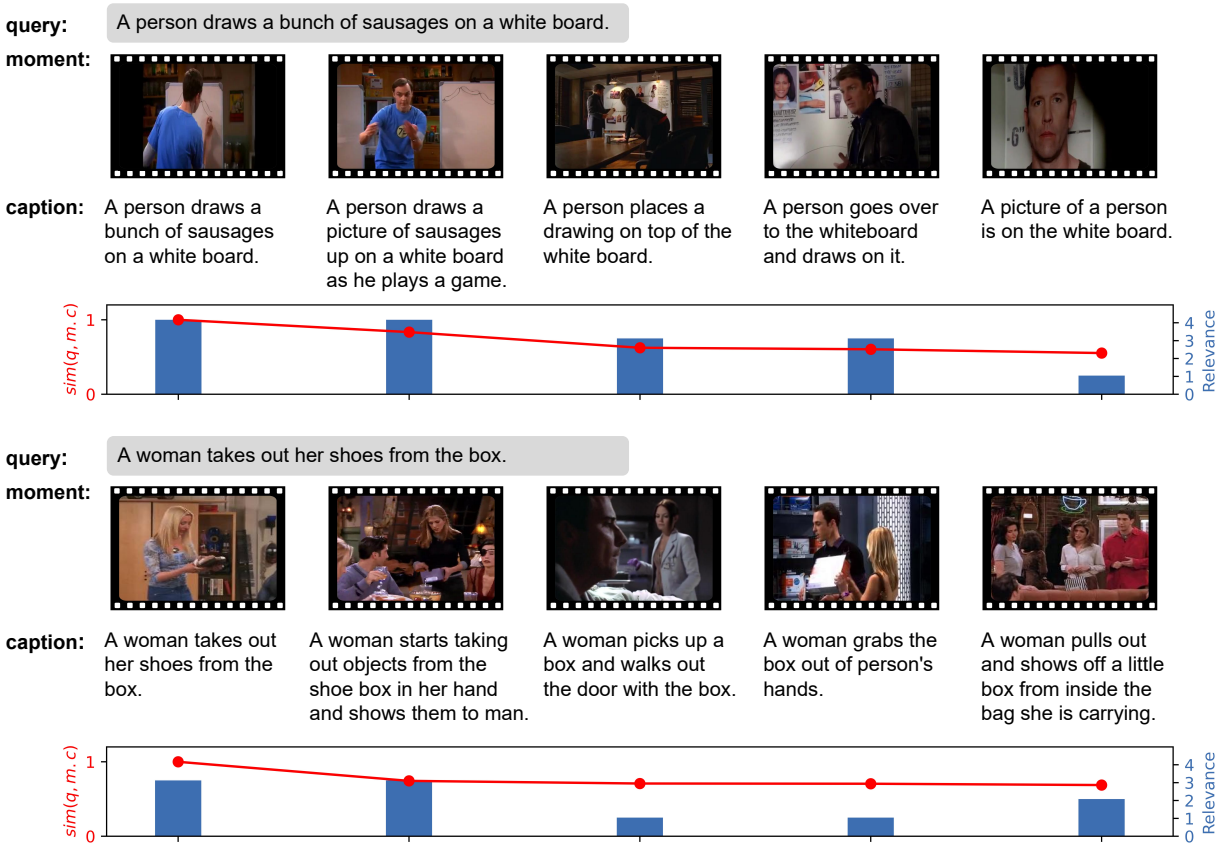


Figure 4: Illustration of similarity vs. human relevance scores for imprecise queries. The line chart shows $sim(q, m.c)$—the semantic similarity between the query q and each moment caption $m.c$, computed using SimCSE. The bar chart shows the final human-annotated relevance scores for the corresponding moments.

table, we also list the average number of relevant moments (with a relevance score of 1 to 4) annotated per query is around 27. Recall that, moments are annotated in batches to a query, with each batch containing 20 candidate moments. Specifically, in the validation set, 264 queries (52.80%) were annotated with 20 moments (i.e., one batch) and 236 queries (47.20%) with 40 moments (i.e., two batches). The test set follows a similar distribution, with 1,473 queries (52.97%) annotated with 20 moments and 1,308 queries (47.03%) with 40 moments.

Following the consensus verification process, 14,382 (97.79%) annotations in the validation set reached consensus with either two or four annotators, while 325 (2.21%) were found to be in disagreement and subsequently discarded. Each annotation here is a query-moment pair. Again, the test set shows a similar distribution; 80,060 (97.90%) annotations achieved consensus and 1,716 (2.10%) annotations were discarded. With the annotations in consensus, on average each query comes with 27.1 relevant moments in the validation set, and 27.0 in the test set, by counting the moments with relevance scores from 1 to 4.

3.6 Case Study of Example Queries

Figure 4 provides two example queries, each with 5 candidate moments ranked by $sim(q, m.c)$ in descending order. The relevance scores assigned by annotators (in bar chart) show a reasonable correlation with $sim(q, m.c)$ (in line chart), where larger $sim(q, m.c)$ also suggests high relevance scores. Yet, the moments with lower $sim(q, m.c)$'s can be annotated with higher relevance scores, and $sim(q, m.c) = 1.0$ does not guarantee a perfect match. The second example is a good illustration of this point. One possible reason for the discrepancy is the annotation methodology. The original TVR dataset required annotators to watch the full video before selecting and describing specific moments, providing them with full context in the video. In our annotation, only moments are presented to annotators, without the full video. Our annotators make judgments solely based on the provided moment. For query "A woman takes out her shoes from the box", the moment with $sim(q, m.c) = 1$ is not considered a perfect match because the moment does not show "the box", though "the box" might exist in the source video somewhere before this moment. Among the test query set, there are 2,635 queries where annotators reach consensus for the query moment pair with $sim(q, m.c) = 1.0$. Among them, 312 (11.76%) moments are not assigned with the perfect relevance score.

We clarify two points here. First, each query is a moment caption originated from the TVR dataset, hence there exists at least one candidate moment whose caption is identical to the query, i.e., $sim(q, m.c) = 1.0$. Second, the moment captions in Figure 4 are provided for reference purposes here. The captions of candidate moments are not shown to the annotators during the annotation process. Annotators judge the level of relevance purely based on moment's visual content to the query. Further, moment captions in the dataset shall only be used as queries, and not as additional information available in source videos. In a RMVR task, the videos are not segmented into moments and such high-quality captions do not exist as well.

4 Evaluation Metric for RVMR

RVMR can be evaluated from at least two aspects: (i) the quality of moment localization, i.e., to what extent the model correctly identifies the temporal boundaries of a moment, and (ii) the quality of ranking, i.e., to what extent the model correctly ranks the retrieved moments from most to least relevance to the query. Note that, even if a moment is correctly located with perfect start/end timestamps, the moment may not be the more relevant to the query.

***IoU* for Moment Localization**. Intersection over Union (*IoU*), denoted by μ, is a common metric widely used in moment retrieval tasks. Given a moment prediction with start and end timestamps, evaluated against the ground truth start and end timestamps, IoU measures the intersection along the timeline against the union along the timeline, illustrated in Figure 5a, where g_0 and p_0 denote the ground truth and predicted moments respectively. If there is no overlap between the two moments, then $\mu = 0$. *IoU* is commonly used as pre-selection criteria for qualifying moments before other measures are computed. For example, a model can be measured by the ability to locate moments with $\mu \geq 0.3$.

***NDCG* for Ranked Retrieval**. Normalized Discounted Cumulative Gain (*NDCG*) is delicately designed for evaluating ranking results with different relevance levels [16]. Specifically, the Discounted Cumulative Gain (*DCG*) of the top K ranked results is defined in Equation 1, where i is the ranking position with 1 being the top ranked position, rel_i is the level of relevance. For example, the left part of Figure 5b shows four ground truth moments with g_1 at rank 1 position and g_4 at rank 4 position. To their left are the relevance levels with $rel_1 = 4$ for g_1 and $rel_2 = 2$ for g_2.

$$DCG@K = \sum_{k=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{1}$$

NDCG@K is then defined as the *normalized DCG* against the *DCG@K* value of a perfect ranking e.g., all items with a relevance level of 4 are ranked before all items with a relevance level of 3, and so on, till the K cut.

The Proposed Metric $NDCG@K, IoU \geq \mu$ for RVMR. We process the matching of predicted moments following their ranking returned by a model. If a predicted moment fails to find a matching ground truth with an $IoU \geq \mu$, it is assigned a relevance score of 0. When multiple ground truths meet the $IoU \geq \mu$ criterion, we select the one with the highest *IoU* and remove it from the ground truth moment listing, to prevent duplicate matches. The *NDCG@K* is computed by the relevance scores of the predicted moments at cut K, against the perfect ranking of the top K ground truth moments, regardless these K moments are matched by any predicted moment or not.

Figure 5 shows the computation of $NDCG@3, IoU \geq 0.3$ as an example. Starting with top predicted moment p_1, where $IoU(p_1, g_1) = 0.35$ and $IoU(p_1, g_3) = 0.4$, both exceeding the threshold $\mu = 0.3$, see Figure 5b. Since the *IoU* with g_3 is higher, we consider p_1 matching with g_3 and assign g_3's relevance score to p_1. Then g_3 is removed from the ground truth due to being matched by a predicted moment. Assuming p_2 is very similar to p_1 (or a near duplicate),[3] with $IoU(p_2, g_1) = 0.35$ and $IoU(p_2, g_3) = 0.4$, as depicted in Figure 5c. Since g_3 has been removed from the ground truth, g_1 becomes the sole match for p_2, resulting in a relevance score of 4 for p_2, and the removal of g_1 from the ground truth. For p_3, $IoU(p_3, g_4) = 0.5$, indicating a match with g_4, thus p_3 is assigned a relevance score of 2, and g_4 is removed. The relevance scores of the predicted moments are 2, 4, and 2. To compute *NDCG* of the ground truth ranking, we consider a perfect ranking of the top 3 ground truth moments (4, 2, and 2), regardless of whether they are matched by predicted moments or not, as illustrated by $K = 3$ on the left side of Figure 5b.

5 Baseline Performance

5.1 Model Adaptation and Implementation

Illustrated in Figure 1, the closest task setting to RVMR is VCMR. In particular, if a VCMR model can compute a form of confidence for its retrieval result, then a ranking of the predicted moments can be naturally derived. Hence, we adapt three representative VCMR models to the RVMR task and evaluate them on the TVR-Ranking dataset: XML [13], CONQUER [8], and ReLoCLNet [27]. The main adaptation involves introducing a query-moment similarity weight $sim(q, m.c)$ into the training loss to downweight less relevant training pairs. We also experimented with another VCMR model, SQuiDNet [23], which employs exhaustive query-video fusion during inference. SQuiDNet failed to yield competitive performance and exhibited prohibitively high inference cost. Due to its limited scalability in a ranked retrieval setting, we exclude it from further baseline comparisons.

In addition, we explored adapting VMR models, which are originally designed for retrieving moments from a single video. These models similarly require dense query-video interactions at inference time, making them computationally impractical for large-scale RVMR evaluation.

XML. XML integrates video and subtitle features (ResNet+I3D and RoBERTa) and uses matrix-based similarity for retrieval. We modify its loss with a similarity-based decay factor $sim(q, m.c)$, and apply a positive pair mask [10] to support multiple relevant moments per query.

ReLoCLNet. Built on XML, ReLoCLNet adds a frame-level contrastive loss. We retain the same input features and loss modifications, and make no change to the frame-level component.

CONQUER. CONQUER separates retrieval and localization, using SlowFast+ResNet for video and RoBERTa for subtitles. It inherits

[3] This is a rare case, and it is unlikely to have two ground truth moments matching duplicate predictions as well. However, we would like to show that our proposed measure is able to handle such a rare case.

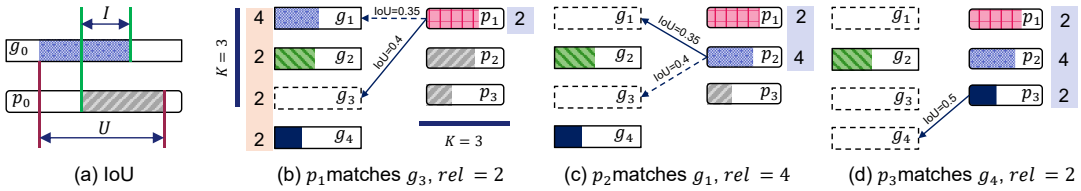


Figure 5: Illustration (a) IoU, and (b)–(d) for $NDCG@3, \mu = 0.3$. (b) p_1 matches g_3 with $rel = 2$ for the larger IoU, above the 0.3 threshold. (c) p_2 matches g_1 as g_3 is no longer available. (d) p_3 matches g_4, with $rel = 2$.

Table 5: Performance of the three baselines. N is the number of moments included in the pseudo training set for each query, sorted by the query-caption similarity $sim(q, m.c)$.

Model	N	$NDCG@10$						$NDCG@20$						$NDCG@40$					
		$IoU \geq 0.3$		$IoU \geq 0.5$		$IoU \geq 0.7$		$IoU \geq 0.3$		$IoU \geq 0.5$		$IoU \geq 0.7$		$IoU \geq 0.3$		$IoU \geq 0.5$		$IoU \geq 0.7$	
		val	test	val	test	val	test	val	test	val	test	val	test	val	test	val	test	val	test
XML [13]	1	0.102	0.092	0.075	0.066	0.024	0.027	0.101	0.092	0.074	0.066	0.026	0.027	0.108	0.102	0.078	0.073	0.027	0.029
	20	0.223	0.214	0.162	0.157	0.058	0.063	0.233	0.224	0.170	0.165	0.063	0.066	0.258	0.251	0.187	0.185	0.071	0.075
	40	0.200	0.204	0.146	0.150	0.054	0.059	0.211	0.217	0.153	0.159	0.058	0.064	0.241	0.243	0.174	0.179	0.067	0.072
CONQUER [8]	1	0.100	0.086	0.084	0.071	0.053	0.051	0.095	0.084	0.081	0.069	0.053	0.048	0.097	0.087	0.083	0.072	0.056	0.051
	20	0.241	0.225	0.222	0.210	0.167	0.152	0.213	0.200	0.198	0.187	0.153	0.137	0.203	0.191	0.189	0.179	0.148	0.133
	40	0.245	0.222	0.226	0.209	0.167	0.152	0.218	0.197	0.202	0.185	0.152	0.137	0.208	0.189	0.193	0.178	0.147	0.132
ReLoCLNet [27]	1	0.158	0.153	0.136	0.135	0.091	0.092	0.150	0.144	0.130	0.127	0.087	0.085	0.153	0.149	0.132	0.130	0.088	0.087
	20	0.375	0.375	0.341	0.340	0.232	0.234	0.382	0.379	0.346	0.343	0.238	0.239	0.404	0.403	0.366	0.365	0.254	0.257
	40	0.434	0.435	0.398	0.399	0.269	0.281	0.442	0.444	0.406	0.406	0.279	0.288	0.472	0.474	0.434	0.434	0.302	0.308

video retrieval from HERO [14]. While HERO is trained on TVR, the risk of leakage is minimized due to query rewriting and new annotations.

We generate pseudo training data using $N = \{1, 20, 40\}$ top-ranked moments per query based on $sim(q, m.c)$. All models are trained with a learning rate of 0.0001 using a warm-up schedule. For $N = 1$, we train 4000 epochs with validation every 20; for $N = 20$ and $N = 40$, we train 200 and 100 epochs respectively, with validation once or twice per epoch. Early stopping is applied if no improvement is observed after 10 validations. All models are implemented in PyTorch 2.2.1 with CUDA 12.1 and trained on a single NVIDIA V100 32GB GPU. No post-processing (e.g., non-maximum suppression) is applied. We evaluate using $NDCG@K$ under intersection-over-union (IoU) thresholds $\mu \in \{0.3, 0.5, 0.7\}$, where $K \in \{10, 20, 40\}$. Parameter tuning is conducted on validation using $NDCG@20, \mu = 0.5$. Table 5 presents the complete set of experimental results.

5.2 Performance Results and Analysis

Performance on Test and Validation Sets. We observe a consistent trend across all metrics, models, and pseudo-training sets: generally, the results on the test set exhibit slightly lower performance compared to the validation set, as expected. However, a few exceptions exist, but the differences in performance are marginal across all such cases.

The K Values for $NDCG$. When training with the top 1 pseudo training set, no clear pattern emerges across the three models as K values change. With the top 20 and top 40 pseudo training sets, the $NDCG$ slightly increases as K changes from 10 to 40 for XML and ReLoCLNet, while CONQUER shows the opposite trend.

The μ Values for IoU. As expected, elevating the value of μ poses a greater challenge to localization, resulting in a decline in performance in general. The impact to models is a bit different as well. In particular, XML experiences a bigger drop compared to other models, suggesting its limitations in achieving precise localization. ReLoCNet shows relatively a smaller drop with higher μ values.

The Choice of N in Pseudo Training: The $N = 1$ training set yields the lowest scores across all three models, suggesting insufficient training instances. For XML, the best performance is achieved with $N = 20$, while ReLoCLNet performs best with $N = 40$. The results for CONQUER are comparable for $N = 20$ and $N = 40$. These findings indicate that the models have different capabilities in handling noise in the training data.

Comparison of Baseline Models: All three models (XML, CONQUER, and ReLoCNet) exhibit consistent performance trends across different training sets and metrics. Among the three, ReLoCNet emerges as the top performer, notably when sufficient training moments are provided i.e., $N = 40$. However, the three baselines show different performance ranking on the VCMR task, as reported in the original papers [8, 13, 27], where CONQUER demonstrates superior performance against ReLoCNet, and XML is slightly better than ReLoCNet as well. The discrepancy on RVMR implies that the abilities required by our RVMR task differ from those of the VCMR task.

Although VCMR models can be easily adapted, directly applying them to an RVMR application may not be appropriate. Designing a new model tailored specifically to our RVMR task is necessary. These findings validate the significance and utility of our dataset for further research and development in the video retrieval field.

Table 6: Performance upper bound on the test set using similarity-based labels for both training and evaluation.

Measure	IoU=0.3	IoU=0.5	IoU=0.7
NDCG@10	0.759	0.758	0.758
NDCG@20	0.808	0.808	0.808
NDCG@40	0.891	0.891	0.891

5.3 Upper Bound of Pseudo-Training with Similarity-Based Labels

To further assess the utility of our pseudo-labeled training data, we conducted an experiment using the same sentence similarity method (used in generating training data) to also create the test set labels. This ensures that the training and testing distributions are aligned. The results, as shown in Table 6, demonstrate a significant gap between current model performance and this upper bound, indicating ample room for improvement and validating the usefulness of the pseudo training set in the early stage of research. Moreover, a recent model proposed by Zhang et al. [26] surpasses our baselines (achieving $NDCG@40$, $IoU = 0.7$ of 0.442), further suggesting that the noise in the training labels does not significantly hinder progress.

6 Conclusion

In this paper, we study the task of ranked moment retrieval from video collection by natural language queries. To facilitate the research in this new task, we develop the TVR-Ranking based on the raw videos and moment annotations of the TVR dataset. Our data annotation process considers query rewriting to best simulate the queries from users who may not have watched all videos in the search collection. The main effort is the manual annotation of relevance levels for a large number of candidate moments for validation and test queries. We then develop the evaluation metric by considering measures used in both ranking tasks i.e., *NDCG*, and in moment retrieval, i.e., IoU. Through experiments, we show that models that perform well on VCMR may not necessarily outperform others on this new RVMR task, indicating the lack of ranking capability of existing models. With the availability of the TVR-Ranking dataset, we expect new technologies to be developed for efficient and effective large-scale video moment search, in line with the development of large language and/or vision models.

Our work has some limitations. First, the queries are adopted from TV series, that might not perfectly mirror the real-world needs of users. Nonetheless, for the purpose of benchmarking and evaluating model capabilities, the dataset is adequate. Second, the pseudo training set was generated using simple sentence similarity. At the time of this work, there were no powerful large visual models available for our purpose. We plan to explore higher-quality annotations using Large Models in the future.

Acknowledgments

This research is supported by cash and in-kind funding from NTU S-Lab and industry partner(s).

References

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL*. 54–59.
[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing Moments in Video With Natural Language. In *ICCV*.
[3] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*. 5267–5275.
[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.
[5] Kristen Grauman, Andrew Westbury, and Others. 2022. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *CVPR*. 18995–19012.
[6] Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2022. A Dataset for Medical Instructional Video Classification and Question Answering. *CoRR* abs/2201.12888 (2022). arXiv:2201.12888
[7] Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2023. A dataset for medical instructional video classification and question answering. *Sci. Data* 10, 1 (2023), 158.
[8] Zhijian Hou, Chong-Wah Ngo, and Wing Kwong Chan. 2021. CONQUER: Contextual query-aware ranking for video corpus moment retrieval. In *MM*. 3900–3908.
[9] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *NeurIPS*, Vol. 33. 18661–18673.
[11] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*. 706–715.
[12] Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. *NeurIPS* 34 (2021), 11846–11858.
[13] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval. In *ECCV*.
[14] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training. In *EMNLP*.
[15] Meng Liu, Liqiang Nie, Yunxiao Wang, Meng Wang, and Yong Rui. 2023. A Survey on Video Moment Localization. *ACM Comput. Surv.* 55, 9, Article 188 (jan 2023). doi:10.1145/3556537
[16] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press. http://nlp.stanford.edu/IR-book/information-retrieval-book.html
[17] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *Trans. Assoc. Comput. Linguist.* 1 (2013), 25–36.
[18] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 658–666.
[19] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. 2017. What actions are needed for understanding human actions in videos?. In *ICCV*. 2137–2146.
[20] 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 *ECCV*. Springer, 510–526.
[21] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. 2022. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *CVPR*. 5026–5035.
[22] Escorcia Victor, Soldan Mattia, Sivic Josef, Ghanem Bernard, and Russell Bryan. 2019. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763* (2019).
[23] Sunjae Yoon, Ji Woo Hong, Eunseop Yoon, Dahyun Kim, Junyeong Kim, Hee Suk Yoon, and Chang D Yoo. 2022. Selective query-guided debiasing for video corpus moment retrieval. In *European Conference on Computer Vision*. Springer, 185–200.
[24] Mengying Yu and Aixin Sun. 2023. Dataset versus reality: Understanding model performance from the perspective of information need. *J. Assoc. Inf. Sci. Technol.* 74, 11 (2023). doi:10.1002/asi.24825
[25] Tongtong Yuan, Xuange Zhang, Kun Liu, Bo Liu, Chen Chen, Jian Jin, and Zhenzhen Jiao. 2023. Towards Surveillance Video-and-Language Understanding: New Dataset, Baselines, and Challenges. arXiv:2309.13925 [cs.CV]
[26] Chongzhi Zhang, Xizhou Zhu, and Aixin Sun. 2025. A Flexible and Scalable Framework for Video Moment Search. *arXiv preprint arXiv:2501.05072* (2025).
[27] Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. 2021. Video Corpus Moment Retrieval with Contrastive Learning. In *SIGIR*.
[28] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2023. Temporal Sentence Grounding in Videos: A Survey and Future Directions. *TPAMI* 45, 8 (aug 2023), 23 pages. doi:10.1109/TPAMI.2023.3258628