CONQUER: Contextual Query-aware Ranking for Video Corpus Moment Retrieval

Zhijian Hou
City University of Hong Kong
zjhou3-c@my.cityu.edu.hk

Chong-Wah Ngo
Singapore Management University
cwngo@smu.edu.sg

W. K. Chan
City University of Hong Kong
wkchan@cityu.edu.hk

ABSTRACT

This paper tackles a recently proposed Video Corpus Moment Retrieval task. This task is essential because advanced video retrieval applications should enable users to retrieve a precise moment from a large video corpus. We propose a novel CONtextual QUery-awarE Ranking (CONQUER) model for effective moment localization and ranking. CONQUER explores query context for multi-modal fusion and representation learning in two different steps. The first step derives fusion weights for the adaptive combination of multi-modal video content. The second step performs bi-directional attention to tightly couple video and query as a single joint representation for moment localization. As query context is fully engaged in video representation learning, from feature fusion to transformation, the resulting feature is user-centered and has a larger capacity in capturing multi-modal signals specific to query. We conduct studies on two datasets, TVR for closed-world TV episodes and DiDeMo for open-world user-generated videos, to investigate the potential advantages of fusing video and query online as a joint representation for moment retrieval.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**;
Video search.

KEYWORDS

Moment Localization With Natural Language; Cross-Modal Retrieval

1 INTRODUCTION

Impressive progress has been witnessed for Video Retrieval (VR) [4, 7, 9, 11] or Single Video Moment Retrieval (SVMR) [6, 34] tasks, where a natural language query is matched with either a video from
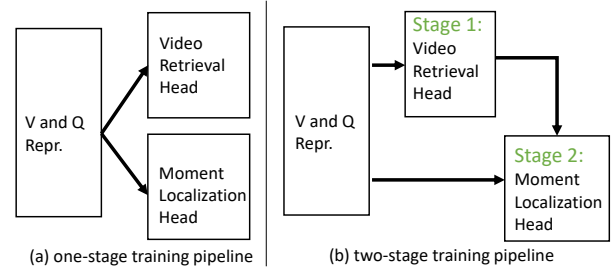
Figure 1: Comparisons between one-stage and two-stage training pipeline. V and Q Repr. means the video and query representations extracted from the backbone. One-stage training pipeline uses a model to train a video retrieval head and a moment localization head simultaneously whereas two-stage training pipeline uses the first model to train a video retrieval head and the second model to train a moment localization head.

a large corpus or a moment within a video. However, those two tasks are not entirely realistic. Current VR benchmarks [5, 35, 41] trim video clips by human intervention from the original lengthy video to form the video corpus. Current SVMR benchmarks [1, 12, 17] assume a priori ground-truth video and consider only visual source. This paper tackles a more realistic *Video Corpus Moment Retrieval* (VCMR) task. For a natural language query, the system aims to retrieve the ground-truth video from the whole corpus and predict the moment with high Intersection-of-Union (IoU) with the ground-truth moment. It is a challenging problem because a video corpus typically contains many videos, each video contains many possible moments, queries are open-ended, and moments are query-dependent and vary in time duration.

State-of-the-art models [19, 20] adopt the one-stage training pipeline with parallel heads as shown in Fig. 1a. A video is first divided into a sequence of non-overlapping clips. Once the backbone produces the video and query representations, those representations are inputted to its video retrieval head to output VR scores and its moment localization head to output the start and end probability distributions for each clip index. Nonetheless, one-stage training pipeline has inherent limitation on precise moment localization. Current moment localization head integrates each clip and query feature by late-fusion to generate 1D query-clip similarity scores and then uses a 1D Convolution to predict the start and end score distributions. The similarity scores are merely summarized through cosine similarity without any attention, and the convolution operator only captures local temporal dependencies within a sliding window.

Early fusion, which alters the video representation based on query attention for similarity scoring, is known to be more effective [23,

24]. Nevertheless, as video representation can only be generated when query is known, videos cannot be indexed for real-time search. Furthermore, early fusion usually consumes considerable inference time and memory space, causing it less preferable.

In general, early fusion of video and query feature involves two key steps. First, the multi-modality features in a video are fused depending on query context. In the literature, multi-modal fusion is practically carried out by weighted sum of different modality features. Before the deep-learning era, there have been numerous empirical studies conducted to investigate query-dependent fusion [21, 33, 39]. These works include classification of user information need into visual or text oriented queries and derivation of optimal fusion weights for different query classes [42]. These efforts, however, are in no avail as practically average fusion is known to perform better in some large datasets despite the inherent limitation. The second step is the coupling of video and query features as a joint feature, which we refer to query-aware feature in this paper, for similarity scoring. Since the deep-learning era, especially due to the popularity of attention mechanism [29] and transformer [38], learning query-aware or question-aware features have been explored, for example, in multimedia question-answering [2, 18].

To the best of our knowledge, both steps of early fusion, i.e., query-dependent fusion (QDF) and query-aware feature learning (QAL), have never been fully explored and jointly exploited. For example, in video search, QDF is a common practice for multi-modal fusion but QAL is seldom employed due to requirement of long inference time in feature generation. This paper addresses the problem, specifically the integrated use of QDF and QAL, in the context of VCMR. The problem is more complex than VR because the query context is expected to couple locally with video for moment localization. We present a new architecture, composed of various off-the-shelf sub-networks [3, 11, 19, 36, 38], to study the problem. The architecture, as depicted in Fig. 2, connects QDF and QAL in two separate steps. QDF derives fusion weights from user query for derivation of video feature. The weights represent the perspective of a query on how different modalities of video content should be composed as a feature for further processing. QAL, subsequently, couples the QDF derived video feature with user query into a single feature for retrieval. We argue that this QDF+QAL derived representation is user-centered and has more potential to answer diverse types of queries. In addition, the representation is more resilient for encapsulating different levels of details in a progressive manner. Specifically, this refers to the processing pipeline where a feature locally extracted from a clip is continuously evolved when being gradually embedded with video and query contexts at different steps from QDF, QAL to a representation readily for moment localization.

The presented work is envisioned to be a two-stage retrieval pipeline as in Fig. 1b, where the 1st-stage VR search engine supplies top-k video candidates for the proposed architecture to perform moment localization and ranking. The main contribution of this paper is the proposal of a novel learning-based architecture, CONQUER, and the provision of empirical insight on joint exploitation of query-dependent fusion and query-aware feature learning in two steps. We study the performance of CONQUER on both the closed-world (i.e., TVR [19]) and open-world (i.e., DiDeMo [1]) datasets to show how retrieval performance is improved when video features are gradually evolved to capture query context.

2 RELATED WORK

Video Retrieval. The task ranks videos among corpus based on the similarity with a language query. Standard pipeline learns global embedding of video and query feature separately, maps those two features to joint embedding space, and finally computes similarity scores by cosine distance. Some recent works tackle only visual context and propose separate embedding for each part-of-speech [40], or multi-level encoding [9] and hierarchical graph reasoning [7] for both visual and text modalities, others focus on multi-modal context and propose mixture of embedding experts [30], collaborative experts [25], multi-modal transformer [11]. Unlike those works, we aim to retrieve the moment rather than the whole video.

Singe Video Moment Retrieval. The task localizes a video segment by a distinct and describable sentence from a video. One kind of dichotomy is early-fusion [1] and late-fusion [6, 12, 13, 23, 24, 28, 34, 43, 45]. Late-fusion approach computes offline query-agnostic video feature while early-fusion approach computes query-aware video features. Due to better localization performance, the majority of papers adopt the early-fusion approach. Another kind of dichotomy is proposal-based [1, 12, 15] and proposal-free [6, 13, 28, 34, 43, 45]. Proposal-based approach first generates handcraft heuristic [15] or sliding window [1, 12] proposals while proposal-free approach directly predicts start and end timestamp. Recent papers mainly follow the proposal-free approach due to less computation cost and better results. Unlike the works mentioned above, we aim to perform retrieval from a large corpus rather than a single video.

Video Corpus Moment Retrieval. Compared to VR and SVMR, relatively few researches – XML [19], HERO [20], HAMMER [44] – are conducted for this problem. These works mainly explore cross-modal attention [29] or transformer [38] to fuse multi-modal features and provide video context for derivation of frame or clip-level features. As most of these works [19, 20] target for one-stage retrieval (as in Fig. 1a), query context is only lightly used with little computation overhead to provide basic information for moment localization. For example, XML [19] applies self-attention to learn two query vectors that respectively capture the visual and textual attentions of user query words. During retrieval, the two online query vectors evaluate text and visual query-clip similarities separately to produce probability scores indicating moment locations. We argue that the late fusion of video and query features for computing similarity as in XML [19] does not fully exploit query context because video features are derived independent of query. Similarly, in HERO [20], query context is not exploited for multi-modal fusion but, instead, is embedded in a joint space as the video feature for video and clip similarity measures. Different from XML and HERO which employ ConvSE [19] for moment localization, HAMMER treats VCMR as a frame classification problem and tags each frame in a video with labels to facilitate the prediction of moment span. Like CONQUER, HAMMER is envisioned as a two-stage retrieval pipeline with the 2nd-stage performs only for top-ranked videos. Nevertheless, HAMMER also does not exploit the query context for early fusion with video features. In this paper, we adopt HERO [20] as the 1st-stage search engine for CONQUER. HERO is pre-trained on HowTo100M instructional videos [31] and TVR [19] TV episodes training data, and reported the state-of-the-art performances on datasets such as TVR [19] and DiDeMo [1].
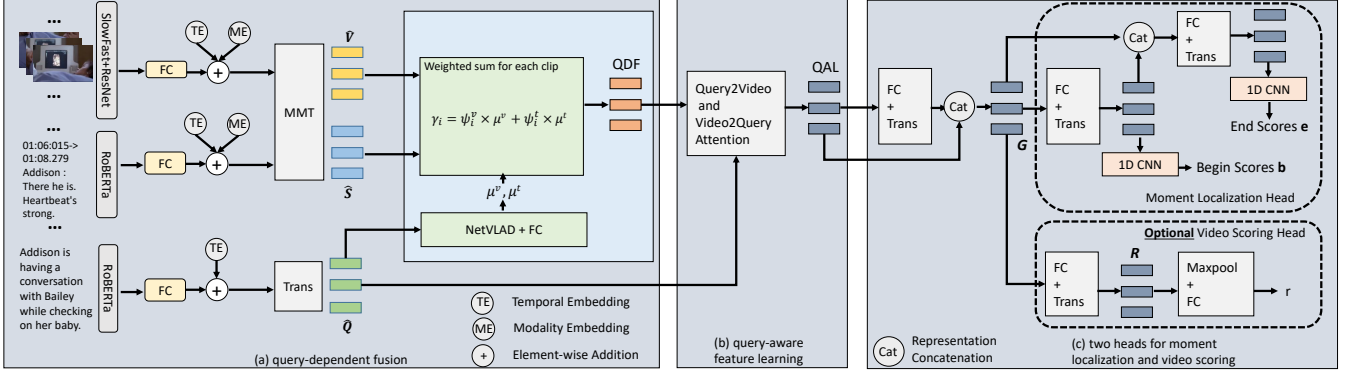
Figure 2: CONQUER: (a) query-dependent fusion, (b) query-aware feature learning, (c) two heads for moment localization and video scoring. *Trans = Transformer*. For illustration, each video (query) is assumed to have three clips (tokens).

3 PROBLEM DEFINITION

Given a query $\mathbf{Q}$, the problem of VCMR is to locate a moment $\mathbf{M}$ in a video $\mathbf{V}$ retrieved from a large dataset D. The query $\mathbf{Q} = [q_1, q_2, ..., q_m]$ is a textual query composed of m tokens. Similarly, a video $\mathbf{V} = [c_1, c_2, ..., c_n]$ is composed of n non-overlapping clips and additionally associated with the text descriptions such as subtitles or ASR (automatic speech transcript). Each description is timestamped but does not necessarily align to clip boundaries. An example of a one-second-long subtitle is "Chase: That's all, is it?" which states the person name and conservation content. Each clip is associated with text descriptions that have time-overlapped with the clip. Hence, a clip c_i is visually and textually described by two different modalities. A moment $\mathbf{M}$ is a subsequence of frames in a video $\mathbf{V}$, spanning one or multiple clips. In VCMR, each query is paired with one moment as the only ground-truth answer. Hence, the goal is to rank the moment as top as possible such that the user can quickly browse and locate the right answer of a query. Different from VR which concerns the retrieval of video that contains query moment from D, VCMR also requires the moment of that video to be retrieved and with sufficient temporal overlap with the ground-truth.

VCMR is more complex for requiring the localized analysis of clip-level information for moment extraction. The retrieval speed also decreases linearly with the number of clips in a video, |$\mathbf{V}$|, and the number of videos, |D|, in a dataset. Hence, a two-stage pipeline is generally helpful in filtering irrelevant videos from clip-level analysis. To this end, given the top-k videos where $k \ll |D|$, the problem of VCMR is to extract the potential moments from k videos and rank them based on their similarity to $\mathbf{Q}$.

4 CONQUER

CONQUER entangles video clips and query progressively to encompass different level of details in two subsequent steps. As depicted in Fig. 2, the two steps are query-dependent fusion (QDF) to combine within-clip modalities and query-aware learning (QAL) to learn new representation that aggregates clip and query information. The query-aware feature is utilized by two heads for moment localization and video scoring. Inter-clip and inter-token contextualization is also performed along with each step and head. Table 1 summarizes

the notations used by CONQUER at different steps to represent the video and query features. For simplicity, the bold notation indicates video or query level features, while the notation with subscript refers to clip or token level features. The various off-the-shelf networks being leveraged to derive the features are also listed in the table 1.

4.1 Query-dependent Fusion (QDF)

Each video clip is described by visual and textual modalities using the deep models stated in Table 1. Instead of concatenating both modalities as a long feature vector as the network input [32], we fuse the modalities with linear weighted sum. The weights are adaptively derived dependent on the user query. Specifically, a query such as "Penny admits Sheldon has said something that she agrees with and haven't thought about before" emphasizes content of verbal discussion that is expected to be out of video content. In this case, textual features deriving from the textual modality should be given higher priority in fusion. On the other hand, a query such as "Rachel is shoving a tie into Chandler's mouth" is more visually than textually appealing, and hence higher visual weight should be assigned. Given ψ_i^v and ψ_i^t as the visual and text modalities feature of a clip, respectively, the QDF feature is generated by two learnable weights μ^v and μ^t, as follows,

$$\gamma_i = \psi_i^v \times \mu^v + \psi_i^t \times \mu^t. \tag{1}$$

To enhance the discriminative power of visual and textual modalities, we first convert the input features v_i and s_i into the same dimension. Specifically, both modalities are initially undergone linear projection to have the same dimensionality. Following the feature pre-processing of Multi-Modal Transformer (MMT) [11], each of the projected feature is summed with two vectors that are self-learnt to capture the temporal evolution and modality-specific information. The two vectors named as temporal and modality embedding, respectively, are randomly initialized. As the number of clips in a video varies, a total of L_v=100 temporal embeddings are learnt, assuming a maximum of 100 clips per video. Each embedding encodes a time-dependent feature. For a video with less than 100 clips, the non-existing clips are represented by zero vectors summed with the temporal and modality embeddings. Next, to learn the inter-clip

Table 1: Notations used by CONQUER.

	Video		Query	Network
Input feature	Visual: $\mathbf{V} = [v_1, ..., v_n] \in \mathbb{R}^{n \times 4352}$			SlowFast [10], ResNet [14]
	Textual: $\mathbf{S} = [s_1, ..., s_n] \in \mathbb{R}^{n \times 768}$		$\mathbf{Q} = [q_1, ..., q_m] \in \mathbb{R}^{m \times 768}$	RoBERTa [26]
Query-dependent	Visual: $\widehat{\mathbf{V}} = [\psi_1^v, ..., \psi_{L_v}^v] \in \mathbb{R}^{L_v \times 768}$			Multi-Modal Transformer [11]
	Textual: $\widehat{\mathbf{S}} = [\psi_1^t, ..., \psi_{L_v}^t] \in \mathbb{R}^{L_v \times 768}$		$\widehat{\mathbf{Q}} = [\phi_1, ..., \phi_{L_q}] \in \mathbb{R}^{L_q \times 768}$	Transformer [38]
	QDF: $\gamma_i = \psi_i^v \times \mu^v + \psi_i^t \times \mu^t \in \mathbb{R}^{768}$			NetVLAD [3]
Query-aware	QDF: $\gamma_i \in \mathbb{R}^{768}$			
	V2Q: $\eta_i \in \mathbb{R}^{768}$			
	Query contextualized: $\gamma_i \odot \eta_i \in \mathbb{R}^{768}$		—	Bi-directional Attention Flow [36]
	Video contextualized: $\gamma_i \odot q_v \in \mathbb{R}^{768}$			
Moment	$\mathbf{b} = [b_1, ..., b_n] \in \mathbb{R}^n$		—	Transformer [38]
Localization	$\mathbf{e} = [e_1, ..., e_n] \in \mathbb{R}^t$			ConvSE [19]

relationship, the sets of features are fed in parallel to the MMT transformer to produced the contextualized visual and textual features, $\widehat{\mathbf{V}}$ and $\widehat{\mathbf{S}}$, respectively. At this phase, the clip-level features not only encode intra-modality signals but are also globally contextualized with video-level information.

Similar transformation is also conducted for the user query, by treating query token as a clip. Note that only temporal embedding is learnt since the query is expressed in text description. A total of L_q=30 embedding are learnt, assuming a maximum of 30 tokens in a query. Note that different from the textual modality of a video clip, each query token is encoded as a separate feature and summed with the temporal embedding. The reason is that the content of a user query usually spans across multiple video clips. Keeping each token feature individually instead of collapsing them allows modeling of clip-token relationship, which will be further elaborated in Section 4.2. Finally, like video clips, the token-level feature is contextualized by a transformer with all other tokens in a query.

To derive query-dependent weights, the transformed query token features are further aggregated by NetVLAD [3] using a learnt codebook of 32 clusters or code words. The resulting vector globally represents the entire user query Q, and captures the residuals between the query and codebook. Such encoded feature has been proven to be more discriminative as reported in [3]. We add a 1-layer fully-connected network with softmax on top of NetVLAD to learn the query-dependent weights, i.e., $\mu^v, \mu^t \in \mathbb{R}^1$ (see Eq. 1). The adaptively fused and globally contextualized multi-modal feature is expected to capture a good mixture of two different modalities prior on the query expectation.

4.2 Query-aware Feature Learning (QAL)

Up to now, the clip-level fusion reflects the relative importance between two modalities, as adaptively determined based on the query nature. To couple both video and query features for moment localization, we adopt a bi-directional memory-less attention mechanism proposed in [36]. The mechanism appends the QDF feature (γ_i) of each video clip in the context with a set of query-aware features. Specially, the appended features are the result of performing attention in two directions: element-wise weighting clip features by query tokens (i.e., query-to-video attention or Q2V) and vice versa (i.e., video-to-query attention or V2Q). The core of attention is a similarity matrix $\in \mathbb{R}^{L_v \times L_q}$, in which an element $a_{i,j}$ represents the

similarity between i-th clip and j-th token, learnt by,

$$a_{i,j} = \mathbf{w_1} \cdot \gamma_i + \mathbf{w_2} \cdot \phi_j + \mathbf{w_3} \cdot (\gamma_i \odot \phi_j),$$

where $\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_3} \in \mathbb{R}^{768}$ are trainable vectors and $\odot$ represents element-wise multiplication. V2Q determines the relevance of query tokens to a video clip. The attention weight on j-th token by i-th clip is denoted as $p_{i,j}$ and each attended query vector η_i of the clip is a linear weighted sum of query-token features, as follows,

$$p_{i,j} = \frac{e^{a_{i,j}}}{\sum_{j=1}^{L_q} e^{a_{i,j}}}, \qquad \eta_{\mathbf{i}} = \sum_{j=1}^{L_q} \phi_j p_{i,j}.$$

Q2V plays attention to the clips that are more similar to one of the query tokens. In this way, clips that are not similar to any token play a relative minor role in the later phase of feature aggregation. Let b_i as the maximum similarity score among all the tokens to i-th clip. The attended video vector q_v, which is a weighted sum of clip features with respect to query context, is calculated as the follows,

$$b_i = \max_{1 \le j \le L_q} a_{i,j}, \qquad p_i = \frac{e^{b_i}}{\sum_{j=1}^{L_v} e^{b_i}}, \qquad q_v = \sum_{i=1}^{L_v} \gamma_i p_i.$$

The query-aware features can be appended to each query-dependent clip feature, for example, by concatenating γ_i with η_i from V2Q and q_v from Q2V. Nevertheless, as q_v is a vector global to all clips, we follow the suggestion in [36] to append the following triplets (η_i, $\gamma_i \odot \eta_{\mathbf{i}}, \gamma_i \odot q_v$) as QAL feature for each clip.

4.3 Moment Localization (ML) Head

The ML head essentially extracts a sequence of clips $[c_b, .., c_e]$, where $b \le e$, as the query moment. This is conducted by estimating the begin and end score distributions of a moment, respectively, for each clip based on its QAL feature. As a pre-processing, we first contextualize QAL features with a transfomer to learn the inter-clip relationship. The transformed features are concatenated with the features in the previous layer as $G \in \mathbb{R}^{L_v \times 3072}$ to reduce the information loss caused by contextualization.

To estimate the "begin" probabilities of clips, a second round of transformation is performed before feeding the feature to ConvSE [19], which is a 1D CNN that produces a score distribution $\mathbf{b}$. To couple the estimation of begin and end probabilities, a third round of transformation is further performed before feeding to ConvSE for

prediction of "end" score distribution **e**. For details, please refer to Fig. 2(c) for the exact workflow of ML head.

4.4 Video Scoring (VS) Head

CONQUER includes an optional head for scoring the similarity of a video to the user query. The head is optional since the similarity can also be derived by consolidating **b** and **e** score distributions. Like ML head, the video scoring (VS) head also contextualizes QAL feature with two transformers. The resulting clip features $R \in \mathbb{R}^{L_v \times 768}$ are max-pooled and then undergone linear regression to predict video similarity score.

4.5 Training Strategy: Shared Normalization

CONQUER is trained end-to-end to optimize the network parameters, by minimizing the moment cross-entropy loss for the ML head and optional video cross-entropy loss for the VS head. Instead of minimizing the moment loss for each video individually, the softmax operator is shared between the positive video and selected negative videos. In other words, the **b** and **e** from different videos are considered jointly such that the learnt begin and end scores are comparable across multiple videos. The technique is referred to as shared normalization originally proposed in [8] for document-level reading comprehension. The begin and end timestamps of a ground-truth moment are first floored and ceiled, respectively, to align with the index of the clip. Denote the aligned ground-truth moment as $\mathbf{M_{gt}} = [c_{b_{gt}}, .., c_{e_{gt}}]$, where b_{gt} and e_{gt} are the aligned begin and end clip index inside the ground-truth video. The shared softmax normalization produces a begin probability distribution for multiple videos in a mini-batch, as follows,

$$P_{begin} = softmax([\mathbf{b}^{pos}, \mathbf{b}^{neg_1}; \mathbf{b}^{neg_2}; \ldots]), \quad (2)$$

where the superscript indicates whether the video in a mini-batch is positive or negative example. The loss of begin timestamp is $-\log P_{begin}(b_{gt})$ (see Eq. 2). The shared normalization is also computed similarly for P_{end} as the end probability distribution. The moment cross-entropy loss is the sum of the begin and end timestamp losses. The loss function for VS head is also computed in a similar way using shared normalization, which involves multiple videos in a mini-batch. The video cross-entropy loss is the negative log-likelihood of the positive video similarity score.

Note that the negative examples are not sampled in random but according to the video rank list provided by the 1st-stage search engine. To increase the sample size while emphasizing more on hard negative examples, empirically, we sample videos up to a search depth of $d = p + 500$, where p is the rank of the ground-truth video. Please refer to Appendix A for further details.

4.6 Discussion

Although CONQUER is built on top of various networks in the literature, the integration novelly explores a rigorous approach of coupling query context in video representation learning. Compared to XML [19] which uses attention model and HERO [20] as well as HAMMER [44] which uses cross-modal transformer as encoders, CONQUER which learns temporal and modality embeddings simultaneously as in MMT [11] can model temporal and modality

signals more effectively. The use of NetVLAD [3] also makes CONQUER different from XML. Specifically, VLAD and NetVLAD are known to be a powerful feature representation technique for large-scale retrieval [3, 16]. In our work, NetVLAD is utilized to enrich query representation with contextually relevant word clusters before learning query-dependent weights. This is also largely different from XML which performs self-attention to learn visual and textual dependent queries for multi-modality similarity scoring. In addition, while ConvSE [19] is also used in XML and HERO, the input to ConvSE is a sequence of similarity scores. In contrast, CONQUER input the learnt QDF+QAL features contextualized with transformers as input directly to ConvSE. As the inputs are more discriminative than similarity scores, better performance from ConvSE is expected. Finally, the use of shared normalization [8] is also unique in CONQUER, where the hard negative videos can be sampled from top-d videos for model training. Different from XML and HERO which only uses ground-truth videos for training, this objective function allows CONQUER to distinguish the moment timestamp within the ground-truth video and all the timestamps in hard negative videos.

5 MOMENT RETRIEVAL

CONQUER parses each of the top-k videos recommended by the 1st-stage search engine, and outputs **b** and **e** score distributions for each video. By outer product of normalized score distributions, $\hat{\mathbf{b}}$ and $\hat{\mathbf{e}}$, within each video, a matrix is formed where each element indicates the probability of a moment candidate. We post-process the matrix to eliminate short and lengthy moment candidates from consideration. Non-maximum suppression (NMS) [22] is then performed to further remove candidates with lower probability while having more than 70% of IoU with an existing moment.

The surviving moment candidates of different videos are subsequently ranked and presented to the user. There are, nevertheless, three possible ways of quantifying moment similarity for ranking. The most *general* way is by multiplying the moment probability obtained by CONQUER with the video similarity score inherited from the 1st-stage search engine. This way is general for naturally leveraging the scores from the two-stage pipeline for retrieval. A more *exclusive* way is by forgetting the video score from the 1st-stage search engine, and directly multiplying the moment probability with the video similarity produced by the VS head of CONQUER. Instead of prior on a global similarity score, a *disjoint* way is to simply rank moment candidates by their probabilities. We experiment all three different ways. In short, denote r^1 and r^2 as the video similarity score obtained from 1st and 2nd stage retrieval respectively for the i-th moment $\mathbf{M}$, the three ways of similarity measure are summarized as

$$\begin{aligned} \text{General:} \quad & \hat{b}_i \times \hat{e}_i \times r^1, \\ \text{Exclusive:} \quad & \hat{b}_i \times \hat{e}_i \times r^2, \\ \text{Disjoint:} \quad & b_i + e_i, \end{aligned} \quad (3)$$

where $\hat{b}_i$ and $\hat{e}_i$ are normalized begin and end scores within the video at a given begin and end index for the i-th moment $\mathbf{M}$, respectively. Note that *disjoint* does not use the normalized probabilities because the normalized scores are not comparable when not being prior on the video similarity score.

Table 2: Dataset statistics showing the number of videos in different splits.

Dataset	Training	Validation	Testing
TVR	17,435	2,179	1,089
DiDeMo	8,395	1,065	1,004

6 EXPERIMENT

6.1 Dataset

We conduct experiment on two datasets: TVR [19] and DiDeMo [1]. The statistics of both datasets are listed in Table 2. TVR is a closed-world dataset where the videos are drawn from 6 long-running TV shows. Videos are accompanied with timestamped subtitles and are on average 76.2 seconds. There are 5 queries designed for each videos, resulting in 5,445 queries in the testing set. The moment length ranges from 0.29 seconds to 123.02 seconds and on average lasts for 9.1 seconds. In contrast, the DiDeMo videos are open-world, unedited and randomly sampled from YFCC100M Flickr videos [37]. The average length is shorter with 54 seconds. Note that the videos in DiDeMo are not accompanied with text descriptions. Instead, ASR is provided by [20] for about 50% of the videos. There are on average 4 queries for each video. The average length for query and moment is 8 tokens and 6.5 seconds, respectively.

While the focus of this paper is on VCMR, we also perform evaluation on VR and SVMR. The evaluation metric follows [19], which is the average recall of queries at the search depth of k or $R@k$. For VCMR and SVMR, an additional IoU threshold (0.5 or 0.7) with ground-truth moment is set. Only moment candidates with IoU equal to or more than the threshold are counted as positives. Note that IoU is measured based on the overlapped timestamps between the retrieved and ground-truth moments.

6.2 Implementation

Following TVR [19], we uniformly divide each video into a sequence of 1.5-second clips. As the clip boundaries do not align with timestamps of text descriptions, we simply assign descriptions to clips that have timestamp overlap between them. The assignment is not one-to-one, or precisely, a textual description may associate with consecutive video clips. In moment localization, we adopt the same assumption as [19] where the prior knowledge of moment length is known to a system. Specifically, the moment length is constrained within $[L_{min}, L_{max}]$. The values correspond to the minimum and maximum number of clips in a moment, respectively. Therefore, moment candidates which are shorter or longer than this range are directly pruned from further processing. We empirically set $L_{min} = 1$ and $L_{max} = 24$ for TVR dataset, and $L_{min} = 3$ and $L_{max} = 7$ for DiDeMo dataset in the following experiments. Please refer to Appendix B for further details.

The learning rate is set as 1e-4. There are two cross-entropy loss functions, respectively, correspond to the ML and VS heads of CONQUER. The weights for moment and video loss are set to 1e-2, 5e-2, respectively. Note that we only use the moment loss for general and disjoint similarity scoring function (see Eq. 3), whereas we use the combination of the two for exclusive similarity scoring function. We adopt early stopping approach, and compute the sum of R@1 values at IoU={0.5, 0.7} after each training epoch. The training only involves queries whose ground-truth videos can be retrieved within the search depth of 100 by the 1st-stage search engine. This is simply because some queries are ambiguous and could have multiple moments as ground-truths although only one of them is marked as positive. Involving these queries can negatively impact learning effectiveness. There are about 0.5% of queries being excluded from training. The training is ended when the value of sum consecutively drops for 3 epochs in the validation set. The training time depends on the number of negative videos being sampled. When keeping the number to 3, CONQUER converges in about 7.5 to 9 hours depending on the similarity scoring function for the TVR dataset. The experiments are conducted on one 2080 Ti GPU using AdamX optimizer [27].

6.3 Ablation Study

We conduct experiments to justify the contribution of three major components (QDF, QAL and shared normalization) in CONQUER, the setting of top-k and the similarity measure for moment ranking. The experiments are conducted on the validation set of TVR because the ground-truth of the test set is not publicly available. The results are presented in tables 3, 4 and 5 respectively. As a reference, when considering the top-10 videos returned by HERO, the oracle VCMR performance in terms of recall is 63.07. This is based on the assumption that, for all the queries, the oracle model manages to extract the ground-truth moments from top-10 videos and rank them within the search depth. This result also indicates that, among the 10,895 validation queries, HERO manages to rank the ground-truth videos of 6,872 queries within the search depth of $k = 10$. Note that when employing HERO as the 1-st stage search engine, we only use the search result produced by its video retrieval head.

Table 3 details the contribution of each component, by ranking the moment candidates extracted from top-10 videos with *general* scoring (see Eq. 3). The baseline is an "empty" CONQUER where modalities are transformed by MMT, averagely fused, multiplied by aggregated query features with self-attention [19], and finally forwarded to ConvSE for moment localization. When comparing this baseline to CONQUER with either QDF or QAL component, it is noted that QAL plays a more significant role in boosting the performances of both VCMR and SVMR. QDF alone manages to push more ground-truth moments within top-5 depth. The result shows that even a relatively simple way of aggregating video and query features, such as the self-attention adopted by baseline, actually works fairly well in ranking. Compared to baseline, CONQUER, which adopts transformer for bi-directional attention modeling, shows much better performance. When combining both QDF and QAL, substantial improvement is noted when comparing to baseline. Shared normalization also contributes to the performance, showing the strategy of using softmax over a positive video and multiple hard negative videos is practically effective in model training. Overall, there are 2,488 out of 10,895 queries where the moments are correctly extracted with at least 70% IoU overlap with the ground-truth and ranked at top-1 position for SVMR. Although some of these moments are not ranked within top-5 for VCMR task, the result still indicates the effectiveness of CONQUER, despite still having a big performance gap compared to the oracle result.

QDF	QAL	Shared normalization	VCMR IoU=0.7			SVMR IoU=0.7	
			R1	R5	R10	R1	R5
			5.89	13.62	18.55	16.92	37.82
✓			5.88	13.78	18.8	16.69	38.12
	✓		6.98	16.24	21.04	21.18	42.96
✓	✓		7.21	16.1	21.09	22.15	43.06
✓	✓	✓	**7.76**	**16.46**	**21.47**	**22.84**	**43.64**

	VCMR IoU=0.7	VR	SVMR IoU=0.7
General	**7.76**	29.01	**22.84**
Disjoint	7.18	26.67	21.84
Exclusive	7.02	**29.29**	20.10

top-k	VCMR IoU=0.7			speed (ms)
	R1	R10	R100	
1	7.24	18.25	22.26	3
5	7.73	22.29	32.74	8.8
10	7.76	22.49	35.17	16.2
100	7.77	22.73	37.67	141.9
200	7.77	22.73	37.67	283.4

Table 4 contrasts the impact of three similarity measures for ranking of moment candidates. Note that, for *general* and *disjoint*, CONQUER is trained with ML head only. Comparing *disjoint* and *exclusive*, the former shows better R@1 for VCMR while the latter shows better R@1 for VR. The result indicates that joint learning of ML and VS heads could hurt the performance of moment localization, despite improving the performance of video search. In SVMR, the fact that *disjoint* outperforms *exclusive* when ranking moment candidates within the ground-truth video further confirms this finding. The result gives clue to the difficulty of training two ML and VS heads simultaneously, and is consistent with our speculation that one-stage training with two parallel heads could result in suboptimal moment localization. Overall, *general*, which computes similarity score prior on the result of 1st-stage search engine, offers the overall best performance in both VCMR and SVMR.

Next, we study the impact of video pool size, i.e., top-k, towards VCMR performance. On TVR validation set, the performance indeed saturates quickly with the increase value of k and becomes stable at around $k = 10$. From top-1 to top-5, the video retrieval (VR) performance of HERO is improved from 29.01% to 52.82% in terms of recall. The VCMR performances of CONQUER at R@1 and R@10 are also boosted accordingly but the performance margin is relatively narrow compared to VR. The smaller degree of improvement is not surprising. This is because the number of moment candidates increases substantially with a larger video pool size, which also

inherently raises the challenge of pushing ground-truth moment to better rank. When further enlarging pool size to beyond $k = 5$, the margin of improvement gets smaller. The result could imply that, for queries where the ground-truth videos are not managed to be ranked within top-5, the corresponding ground-truth moments are also equally hard to be ranked in better positions. Nevertheless, our result shows that the challenge is not just due to moment localization but also the difficulty of ranking the positively extracted moments when video pool size increases. For example, when performing SVMR on the ground-truth videos of queries whose ground-truth moment ranks are outside of 10, CONQUER manages to rank about 25% of them with their ground-truth moments within top-5 position. Among them, more than 50% are ranked within top-3. On the other hand, while the performance saturates rapidly, the VCMR performance of CONQUER is not adversely impacted. This is evidenced from the fact that, when top-k is increased from $k = 10$ to 200, CONQUER still maintains the same level of performance at R@1 and R@10. In terms of speed efficiency, CONQUER only incurs little computation overhead with an average speed of 142 milliseconds per query even when top-100 videos are considered. Overall, HERO+CONQUER can process almost 30 queries per second when $k = 10$. In the remaining experiments, we set $k = 10$ and use *general* similarity measure as the default setting, unless otherwise stated.

6.4 Performance Comparison

Table 6 lists the results of various approaches on TVR dataset. CONQUER significantly outperforms all approaches across different recall levels on both validation and testing sets. XML, which contextualizes video-level features with attention models, generally shows lower performance than others which adopt more advanced transformers for contextualizing features. HERO, which is pre-trained on large dataset (Howto100M [31]), offers stronger performance than XML and HAMMER. Compared to CONQUER, however, the performance is still limited for not contextualizing video feature with query. The improvement introduced by CONQUER is considered impressive. For example, at R@10, the performance closely approaches the performance of HERO at R@100 on the validation set, and the performance of XML at R@100 on the testing set.

Figure 3 shows four typical queries contrasting the performances of CONQUER and HERO. In Query-A, HERO only localizes the short segment where "Phoebe walks in to join" while CONQUER also manages to locate the prior segments showing "Leslie performs a song in Central Perk". For this query, CONQUER gives higher weight to visual ($\mu^v = 0.62$). Hence, even Central Perk or singing is not mentioned in the transcript, CONQUER is still able to locate the moment quite precisely. In Query-B, HERO can not capture the query phrase "a young woman in a purple hoodie" and locate an incorrect moment where Dr. Hadley talks to a woman. In contrast, by intertwining video and query representations into a single representation, CONQUER manages to locate the ground-truth moment where a woman with purple hoodie is clearly visible. In Query-C, both HERO and CONQUER localize a segment that fully cover the target moment. However, both segments also include extra clips that showing "Sheldon is putting down the spray". Nevertheless, the moment localized by CONQUER is in higher quality as it includes only the scene where Sheldon holds spray bottle. Query-D shows an
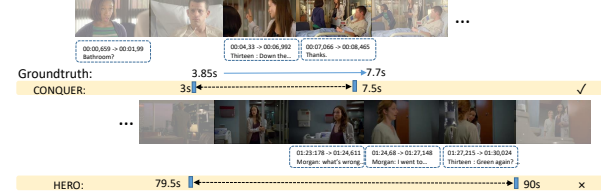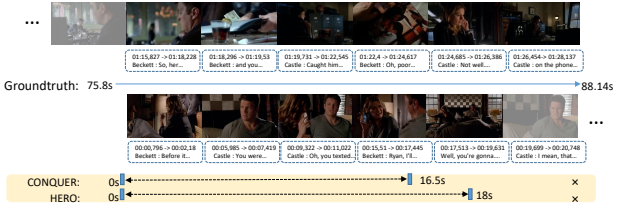
Figure 3: Four example queries showing the moments localized by CONQUER and HERO. A tick is given to the correctly localized moment, while a cross indicates negative localization because of IoU ≤ 0.7. Best view by zooming to cross-read the query, subtitle and video timeline.

Table 6: Comparison of VCMR performance (IoU=0.7) with various state-of-the-art approaches on TVR.

Model	Validation			Testing		
	R1	R10	R100	R1	R10	R100
XML [19]	2.62	9.05	22.47	3.32	13.41	30.52
HAMMER [44]	5.13	11.38	16.71	-	-	-
HERO [20]	5.13	16.26	24.56	6.21	19.34	36.66
CONQUER (general)	**7.76**	22.49	**35.17**	**9.24**	**28.67**	41.98
CONQUER (disjoint)	7.18	**23.0**	33.94	8.5	26.72	39.12
CONQUER (exclusive)	7.02	22.44	34.69	7.33	26.81	**42.19**

Table 7: Results on DiDeMo Testing set.

Model	VCMR IoU=0.5			VCMR IoU=0.7		
	R1	R5	R10	R1	R5	R10
XML [19]	2.26	-	10.42	1.59	-	6.77
HERO [20]	**3.37**	8.79	13.26	2.76	7.73	11.78
CONQUER (general)	3.31	**9.27**	**13.99**	**2.79**	**8.04**	**11.9**
CONQUER (disjoint)	3.09	8.49	12.86	2.66	6.93	10.27
CONQUER (exclusive)	3.14	8.16	12.05	2.61	7.03	11.02

example query which is considered challenging because the pharse "extra marital affair" is an abstract concept which cannot be easily inferred from subtitle or visual content. For this query, both CON-QUER and HERO localize a moment where Castle and Beckett talk about their potential marital affair.

Table 7 compares the performances on DiDeMo dataset [1]. Note that only 50% of videos are supplied with text descriptions, and almost all the queries are visual-oriented. For these reasons, CON-QUER only considers visual modality. In other words, the QDF in CONQUER is turned off for this experiment. Nevertheless, HERO[1] still exploits the extracted ASR information and benefits from it.

---

[1]The HERO result is based on our reproduced result since there is an error in the original HERO code for evaluating the DiDeMo dataset.

As shown, CONQUER still significantly outperforms XML. Nevertheless, the improvement of CONQUER over HERO is not as significant as in TVR. We speculate that, as the videos in DiDeMo are not as correlated as in TVR, QAL in CONQUER cannot take advantage of closed-world knowledge to calibrate video feature with query context. Nevertheless, the result of CONQUER is still better or comparable to that of HERO, despite not exploiting text modality.

7 CONCLUSION AND FUTURE WORK

We have presented CONQUER as a module designated for two-stage moment retrieval. Compared to one-stage retrieval, CONQUER has more capacity to exploit query context for adaptive multi-modal fusion and representation learning. As demonstrated on TVR dataset, CONQUER is particularly effective and introduces a large margin of improvement compared to the current state-of-the-arts. The improvement due to QAL is most significant, and the best result is attained when combining QDF, QAL and shared normalization. While CON-QUER obtains overall encouraging results on TVR and DiDeMo datasets, the result also indicates that query context is relatively easier to exploit in close-world than open-world dataset. In terms of retrieval efficiency, with HERO as the 1-st stage search engine, the overall speed of HERO+CONQUER manages to process about 30 queries per second on a dataset with slightly more than 2000 videos of about 46 hours.

Furthermore, integration with audio is also possible except the increase in computational cost. Currently, the clip duration is set based on the setting in HERO. We do believe there are other better options, for example to have variable clip length that aligns with the subtitle timestamp. We will leave these issues as future work.

REFERENCES

[1] 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*. 5803–5812.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. 2425–2433.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5297–5307.

[4] George Awad, Cees GM Snoek, Alan F Smeaton, and Georges Quénot. 2016. Trecvid semantic indexing of video: A 6-year retrospective. *ITE Transactions on Media Technology and Applications* 4, 3 (2016), 187–208.

[5] Fabian Berns, Luca Rossetto, Klaus Schoeffmann, Christian Beecks, and George Awad. 2019. V3c1 dataset: An evaluation of content characteristics. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 334–338.

[6] Da Cao, Yawen Zeng, Meng Liu, Xiangnan He, Meng Wang, and Zheng Qin. 2020. STRONG: Spatio-Temporal Reinforcement Learning for Cross-Modal Video Moment Localization. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4162–4170.

[7] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-Grained Video-Text Retrieval With Hierarchical Graph Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10635–10644.

[8] Christopher Clark and Matt Gardner. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 845–855.

[9] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9346–9355.

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slow-Fast Networks for Video Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 6201–6210.

[11] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal Transformer for Video Retrieval. In *Proceedings of the European Conference on Computer Vision*. 214–229.

[12] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. TALL: Temporal Activity Localization via Language Query. In *Proceedings of the IEEE International Conference on Computer Vision*. 5267–5275.

[13] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander G. Hauptmann. 2019. ExCL: Extractive Clip Localization Using Natural Language Descriptions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 1984–1990.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[15] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2018. Localizing Moments in Video with Temporal Language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1380–1390.

[16] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3304–3311.

[17] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 706–715.

[18] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, Compositional Video Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1369–1379.

[19] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval. In *Proceedings of the European Conference on Computer Vision*. 447–463.

[20] 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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2046–2065.

[21] Zhonghua Li, Bingjun Zhang, Yi Yu, Jialie Shen, and Ye Wang. 2013. Query-document-dependent fusion: A case study of multimodal music retrieval. *IEEE transactions on multimedia* 15, 8 (2013), 1830–1842.

[22] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision*. 3–19.

[23] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. 2018. Attentive Moment Retrieval In Videos. In *Proceeding of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 15–24.

[24] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. 2018. Cross-modal Moment Localization In Videos. In *Proceedings of the 26th ACM international conference on Multimedia*. 843–851.

[25] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use What You Have: Video retrieval using representations from collaborative experts. In *Proceedings of the British Machine Vision Conference*. 279.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[27] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[28] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. 2019. DEBUG: A dense bottom-up grounding approach for natural language video localization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 5147–5156.

[29] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[30] Antoine Miech, Ivan Laptev, and Josef Sivic. 2018. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv preprint arXiv:1804.02516* (2018).

[31] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *Proceedings of the IEEE International Conference on Computer Vision*. 2630–2640.

[32] Liqiang Nie, Wenjie Wang, Richang Hong, Meng Wang, and Qi Tian. 2019. Multi-modal dialog system: Generating responses via adaptive decoders. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1098–1106.

[33] Yuxin Peng and Chong-Wah Ngo. 2006. Clip-based similarity measure for query-dependent clip retrieval and video summarization. *IEEE Trans. Circuits Syst. Video Technol.* 16, 5 (2006), 612–627.

[34] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Fine-grained Iterative Attention Network for Temporal Language Localization in Videos. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4280–4288.

[35] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for Movie Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3202–3212.

[36] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *International Conference on Learning Representations*.

[37] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[39] Peter Wilkins, Alan F Smeaton, and Paul Ferguson. 2010. Properties of optimally weighted data fusion in CBMIR. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 643–650.

[40] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. 2019. Fine-Grained Action Retrieval Through Multiple Parts-of-Speech Embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*. 450–459.

[41] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5288–5296.

[42] Rong Yan, Jun Yang, and Alexander G Hauptmann. 2004. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th annual ACM International Conference on Multimedia*. 548–555.

[43] Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. 9159–9166.

[44] Bowen Zhang, Hexiang Hu, Joonseok Lee, Ming Zhao, Sheide Chammas, Vihan Jain, Eugene Ie, and Fei Sha. 2020. A Hierarchical Multi-Modal Encoder for Moment Localization in Video Corpus. *arXiv preprint arXiv:2011.09046* (2020).

[45] Songyang Zhang, Jinsong Su, and Jiebo Luo. 2019. Exploiting temporal relationships in video moment localization with natural language. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1230–1238.

APPENDIX

A ADDITIONAL INFORMATION ON TRAINING

A.1 The 1-st Stage Search Engine

In order to make CONQUER independent of HERO (in case of experiment bias), we indeed use a simplified version of CONQUER as the 1-st stage search engine. Referring to Fig. 2(a), we still use the MMT and transformer to compute $\widehat{V}$, $\widehat{S}$ and $\widehat{Q}$ for QDF. NetVLAD is used to derive the fusion weight μ^v, μ^t. Same as XML [19], we also learn to aggregate $\widehat{Q}$ into two vectors as query vectors for visual and textual modalities, respectively. The vectors are directly used to compute the video similarities with $\widehat{V}$, $\widehat{S}$. The two similarities scores are further weighted sum using the fusion weights μ^v, μ^t. Note that QAL and ML heads are not used in this simplified CONQUER.

A.2 Negative Video Sampling

During training, we sample videos of a mini training batch up to a search depth of $d = p + x$, where p is the rank of the ground-truth video, and x is the extension range. Fig. 4 shows the VCMR result sum curves for different similarity scoring functions when the negative video number is set as 3. As we can see, $x = 500$ achieves the optimal performance and is the final adopted setting in this work. Compared with the performance of $x = 500$, the performances of randomly sampled negative videos (see $x = 17,435$) are always inferior. It indicates the importance to sample top-ranked videos as negative videos.

B THE MOMENT LENGTH CONSTRAINS

Same as XML [19] and HERO [20], we assume the knowledge of moment length for pruning short and lengthy moments. We use $L_{min} = 1$ and $L_{max} = 24$, instead of $L_{min} = 2$ and $L_{max} = 16$ as suggested by HERO for TVR dataset. Table 8 provides the experimental results if we set the moment length constrains as HERO, i.e., $L_{min} = 2$ and $L_{max} = 16$. As shown in Table 8, CONQUER still outperform HERO and XML significantly by using this setting. Note that the setting ($L_{min} = 3$ and $L_{max} = 7$) on DiDeMo is the same for both HERO and CONQUER.

C MORE RESULTS ON TVR

Table 9 provides the full set of results comparing CONQUER with XML, HAMMER and HERO. The results include performance for VCMR, VR and SVMR.

Table 8: Results of CONQUER if using the setting of $L_{min} = 2$ and $L_{max} = 16$ suggested by HERO for result pruning.

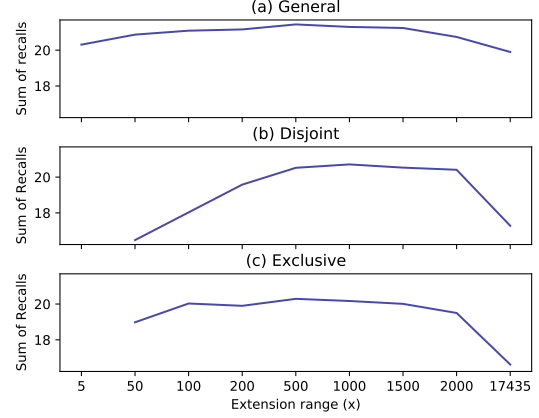| Model	VCMR				SVMR IoU=0.7	
	R1	R5	R10	R100	R1	R5
XML [19]	2.62	6.39	9.05	22.47	13.89	31.11
HERO [20]	5.13	12.24	16.26	24.56	15.36	32.27
CONQUER (general)	**7.25**	16.49	**21.66**	**33.88**	**21.87**	46.18
CONQUER (disjoint)	6.89	**16.88**	22.38	33.29	21.03	**46.98**
CONQUER (exclusive)	6.79	16.35	21.57	33.8	19.55	43.83

Figure 4: Effects of negative video sampling. The y-axis shows the sum of R@1 for both IoU={0.5, 0.7} and x-axis is the extension range of search depth.

Table 9: Full results on TVR validation set.

Model	VCMR IoU=0.7				VR		SVMR IoU=0.7	
	R1	R5	R10	R100	R1	R5	R1	R5
XML [19]	2.62	6.39	9.05	22.47	16.08	37.92	13.89	31.11
HAMMER	5.13	-	11.38	16.71	-	-	-	-
HERO [20]	5.13	12.24	16.26	24.56	29.01	52.82	15.36	32.27
CONQUER (general)	**7.76**	17.22	22.49	**35.17**	29.01	52.82	**22.84**	47.72
CONQUER (disjoint)	7.18	**17.4**	**23.0**	33.94	26.67	55.89	21.84	**48.63**
CONQUER (exclusive)	7.02	17.03	22.44	34.69	**29.29**	**56.15**	20.1	45.12