Lighthouse: A User-Friendly Library for Reproducible Video Moment Retrieval and Highlight Detection

Taichi Nishimura **Shota Nakada** **Hokuto Munakata** **Tatsuya Komatsu**
LY Corporation
{tainishi,shota.nakada,hokuto.munakata,komatsu.tatsuya}@lycorp.co.jp

Abstract

We propose Lighthouse, a user-friendly library for reproducible video moment retrieval and highlight detection (MR-HD). Although researchers proposed various MR-HD approaches, the research community holds two main issues. The first is a lack of comprehensive and reproducible experiments across various methods, datasets, and video-text features. This is because no unified training and evaluation codebase covers multiple settings. The second is user-unfriendly design. Because previous works use different libraries, researchers set up individual environments. In addition, most works release only the training codes, requiring users to implement the whole inference process of MR-HD. Lighthouse addresses these issues by implementing a unified reproducible codebase that includes six models, three features, and five datasets. In addition, it provides an inference API and web demo to make these methods easily accessible for researchers and developers. Our experiments demonstrate that Lighthouse generally reproduces the reported scores in the reference papers. The code is available at https://github.com/line/lighthouse.

1 Introduction

With the rapid advance of digital platforms, videos become ubiquitous and popular on the web. Although they offer rich, informative, and entertaining content, watching entire videos can be time-consuming. Hence, there is a high demand for multimodal tools that enable users to quickly find specific moments within videos and browse through highlights in the moments from natural language queries. The former is called moment retrieval (MR) and the latter is called highlight detection (HD). Given a video and a language query, MR retrieves relevant moments (start and end timestamps), and HD detects highlighted frames within these moments by calculating saliency scores representing frame-level highlightness (Figure 1). Note that HD calculates saliency scores for all frames in the video, but the frames with the highest saliency scores are detected within the moments.

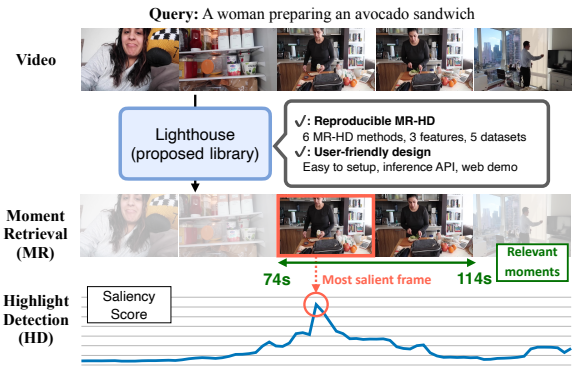

Figure 1: Overview of MR-HD and Lighthouse. Given a video and query, the model predicts relevant moments for MR and saliency scores for HD. Lighthouse achieves reproducible MR-HD by supporting multiple settings. In addition, it aims at a user-friendly design with an easy-to-setup environment, inference API, and web demo.

Although MR and HD share common characteristics, such as learning the similarity between input queries and video frames, they were separately treated due to the lack of annotations supporting both tasks (Zhang et al., 2020; Song et al., 2015). To address this, Lei et al. (2021) proposed the QVHighlights dataset comprising videos, language queries, and moment/highlight annotations, enabling researchers to tackle both tasks simultaneously. We refer to this unified task of MR and HD as *MR-HD* to distinguish it from the individual tasks of MR and HD. Based on this dataset, various approaches have been proposed to perform MR-HD. Note that most methods are applicable for single tasks of either MR or HD as well as MR-HD.

Despite the rapid development of MR-HD, the research community holds two issues. The first is a lack of comprehensive and reproducible experiments across various methods, datasets, and fea-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing
System Demonstrations, pages 53–60

	MR-HD	MR			HD			
	QVHighlights	**ActivityNet Captions**	**Charades-STA**	**TaCoS**	**TVSum**	**Features**	**API?**	**Web demo?**
Moment DETR (Lei et al., 2021)	✓					C+S		
QD-DETR (Moon et al., 2023b)	✓				✓	C+S		
EaTR (Jang et al., 2023)	✓	✓	✓			C+S		
TR-DETR (Sun et al., 2023)	✓				✓	C+S		
UVCOM (Xiao et al., 2024)	✓		✓			C+S		
CG-DETR (Moon et al., 2023a)	✓		✓	✓	✓	C+S+V+G		
Lighthouse (ours)	✓	✓	✓	✓	✓	C+S+R+G	✓	✓

Table 1: Comparison of Lighthouse and existing publicly available MR-HD repositories. C, S, V, R, and G in the "Features" column represent CLIP (Radford et al., 2021), Slowfast (Feichtenhofer et al., 2019), VGGNet16 (Simonyan and Zisserman, 2014), ResNet152 (He et al., 2016), and GloVe (Pennington et al., 2014), respectively.

tures. This is because there is no unified training and evaluation codebase covering multiple settings. While previous work reported scores for their methods on individual tasks for MR, HD, and MR-HD, researchers release their code only for QVHighlights, without necessarily providing training codes for other datasets. In addition, datasets and features are not standardized. Researchers use different MR and HD datasets to demonstrate their approach's effectiveness (Table 1). Hence, to fully reproduce experiments, researchers should set up individual environments and write additional code, ranging from video-text feature extraction preprocessing to modifications to the training and evaluation codes. This is time-consuming and cumbersome.

The second is user-unfriendly design. Because previous works use different libraries for their method, MR-HD researchers should set up individual environments. In addition, most previous works release only training codes, requiring users to implement the whole inference process of MR-HD and apply it to their videos. This includes frame extraction from videos, video-text feature extraction, and forwarding them into the trained model. Implementing all of these steps accurately is challenging for developers who are interested in MR-HD but lack expertise in video-text processing.

Our goal is to address these issues and foster the MR-HD research community. To this end, we propose ***Lighthouse***, a user-friendly library for reproducible MR-HD. Lighthouse unifies training and evaluation codes to support six recent MR-HD methods, three features, and five datasets for MR-HD, MR, and HD, resolving the reproducibility issue. While this results in 90 possible configurations (6 methods $\times$ 3 features $\times$ 5 datasets), the configuration files are written in YAML format, allowing researchers to easily reproduce experiments by specifying the necessary file. Our experiments demonstrate that Lighthouse mostly reproduces

```yaml
model_name: tr_detr
dset_name: activitynet
ctx_mode: video_tef
v_feat_types: slowfast_clip
t_feat_type: clip
train_path: data/activitynet/activitynet_train_release.jsonl
eval_path: data/activitynet/activitynet_val_release.jsonl
eval_split_name: val
v_feat_dirs: ['features/ActivityNet/clip', 'features/ActivityNet/slowfast']
t_feat_dir: features/ActivityNet/clip_text
v_feat_dim: 2818
t_feat_dim: 512
aux_loss: True
results_dir: results/clip_slowfast_tr_detr/activitynet
ckpt_filename: best.ckpt
train_log_filename: train.log
eval_log_filename: val.log
clip_length: 2

# TR-DETR specific losses
VTC_loss_coef: 0.3
CTC_loss_coef: 0.5
```

Figure 2: YAML configuration example.

the original experiments in the referenced six papers. In addition, to resolve the user-unfriendliness, Lighthouse provides an inference API and web demo. The inference API covers the entire MR-HD process and provides users with easy-to-use code for MR-HD. The web demo, built upon the API, enables users to confirm the results visually. The codes are under the Apache 2.0 license.

2 Highlights of Lighthouse

Table 1 shows a comparison of Lighthouse and public MR-HD repositories. We describe them in terms of reproducibility and user-friendly design.

2.1 Reproducibility

Support for multiple methods, datasets, and features: As shown in Table 1, previous works support different datasets and features for MR and HD tasks. Lighthouse supports all of them by integrating all these MR-HD methods, features, and datasets into a single codebase. We extract video-text features from all datasets, train models using these features, and release reproducible code along with the features and pre-trained weights. This significantly reduces the effort required to write additional code for conducting experiments across multiple settings.

```python
import torch
from Lighthouse.models import CGDETRPredictor

device = 'cuda' if torch.cuda.is_available() else '
    cpu'

# Initialize model instance
model = CGDETRPredictor('checkpoint.ckpt',
                        device=device,
                        feature_name='clip')

# Encode video features
model.encode_video('video.mp4')

# Moment retrieval & highlight detection
query = 'A man is speaking in front of the camera'
pred = model.predict(query)
```

Listing 1: Example usage of the inference API.

Reproducible training and evaluation: Lighthouse enables researchers to reproduce the training process with a single Python command by specifying the configuration files, where hyper-parameters are written in YAML format (Figure 2). The Lighthouse users can easily test different hyper-parameters by modifying these files. We release all of the files used or generated during experiments, including video-text features, trained weights, and logs during the training. Therefore, to reproduce the experiments, researchers can obtain the same results by downloading the necessary files and running a single Python evaluation command with the trained weights.

2.2 User-friendly design

Easy to set up: Lighthouse allows researchers and developers to install it easily with "`pip install .`" after cloning the repository. Because the libraries used in previous work vary between repositories, researchers need to set up individual environments by cloning each repository and installing the dependency libraries. Lighthouse streamlines this process by summarizing the necessary libraries and carefully removing any unnecessary ones that are imported but not used in the codebase.

Easy to use: Lighthouse provides an inference API and a web demo, enabling researchers and developers who are not well-versed in detailed MR-HD pipelines, to use MR-HD. Listing 1 shows the inference API, which hides the detailed implementation of video-text processing and provides users with three main steps: model initialization, `encode_video()`, and `predict()`. First, the user initializes the model instance by specifying the model weight, device type (i.e., CPU or GPU), and feature name. Second, given a video path, `encode_video()` extracts frames from the video, converts them into features, and stores them as instance variables. Finally, given a query, `predict()` encodes the query and forwards both the video and query features into the model to obtain results. Figure 3 shows a web demo built upon the inference API to visualize the model's outputs. By clicking on the moment panes, the video seek bar jumps to the corresponding timestamps, enabling users to view those specific moments. Hovering over the saliency scores lets users see both the values and the corresponding timestamps in the video.

3 Architecture of Lighthouse

Figure 4 shows an overview of Lighthouse architecture, consisting of four components: datasets, video-text feature extractor, models, and evaluation metrics.

3.1 Datasets

We utilize five commonly-used datasets: QVHighlights (Lei et al., 2021), ActivityNet Captions (Krishna et al., 2017), Charades-STA (Hendricks et al., 2017), TaCoS (Regneri et al., 2013), and TVSum (Song et al., 2015). The QVHighlights dataset is an MR-HD dataset comprising videos, queries, and annotations for both moments and highlights. It is the only dataset that includes annotations for both moments and highlights. Moments are represented as start and end timestamps for each query, while highlights are represented as saliency scores ranging from 1 (very bad) to 5 (very good) for each frame of the video. ActivityNet Captions, Charades-STA, and TaCoS are MR datasets because they contain only moment annotations, whereas TVSum is an HD dataset as it includes 50 videos from ten domains (e.g., news and documentary) and highlight annotations. Note that we do not release the original videos due to copyright issues. Instead, we release the pre-processed video-text features to allow researchers to reproduce experiments.

3.2 Video-text feature extractor

Given video frames and a query, the video-text encoders convert them into frame- and word-level features $\mathbf{V} \in \mathbb{R}^{L \times D_v}$, $\mathbf{T} \in \mathbb{R}^{T \times D_t}$, where L and T represent the numbers of frames and words, and D_v and D_t represent the dimensions of the vision and text features. We utilize three feature extractors: CLIP (Radford et al., 2021), CLIP+Slowfast (Feichtenhofer et al., 2019), and ResNet152+GloVe (He et al., 2016; Pennington et al., 2014). CLIP employs vision and

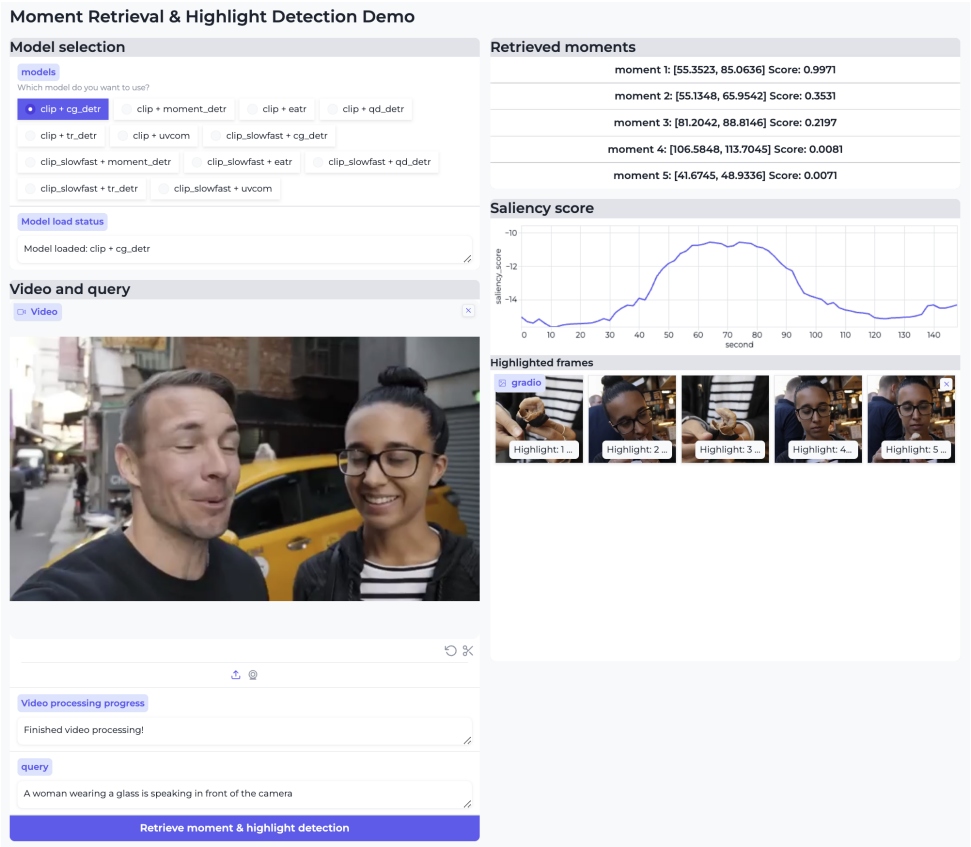

Figure 3: A screenshot of the web demo. In the web demo, you can select a model and feature in the model selection pane. Then, in the video and query pane, you can upload a video and input a text query. By clicking the 'Retrieve Moment & Highlight Detection' button, the retrieved moments and highlighted frames will be displayed in the right panes. Hugging face spaces: https://huggingface.co/spaces/awkrail/lighthouse_demo.

text encoders, based on the Transformer architecture (Vaswani et al., 2017), pre-trained on extensive web image-text pairs. These encoders transform frames and queries into feature vectors. CLIP+Slowfast combines CLIP vision features with Slowfast features to enhance motion awareness, as Slowfast is pre-trained on the Kinetics400 action recognition dataset (Kay et al., 2017) and is adept at recognizing motion in videos. ResNet152+GloVe uses ResNet152 for frame-wise visual features and GloVe for word-level text features. ResNet152 and GloVe are pre-trained on ImageNet (Deng et al., 2009) and English Wikipedia, respectively. While CLIP is the standard in MR-HD, this setup allows us to assess the superiority of CLIP's vision-language encoders by comparing them with models trained separately on visual and textual data. Note that we extract video-text features as a preprocessing step before training, rather than during training because extracting features during training is costly and time-consuming. For this process, we use the HERO video extractor library (Li et al., 2020).

3.3 Models

We implement six recent MR-HD models: Moment DETR (Lei et al., 2021), QD-DETR (Moon et al., 2023b), EaTR (Jang et al., 2023), TR-DETR (Sun et al., 2023), UVCOM (Xiao et al., 2024), and CG-DETR (Moon et al., 2023a). These models are extensions of DETR (Carion et al., 2020), Transformer-based object detectors, adapted for MR-HD. Given a video and language query, they can predict both moments and saliency scores. Note that, except for TR-DETR, these models are designed to be trainable on a single task of MR or HD[1].

We describe briefly by focusing on the difference between these methods. Moment DETR is first proposed with QVHighlights as an MR-HD

[1]Note that TR-DETR is unavailable for single MR and HD tasks because the official code necessitates MR-HD annotations for loss calculation. See: https://github.com/mingyao1120/TR-DETR/issues/3 for details.

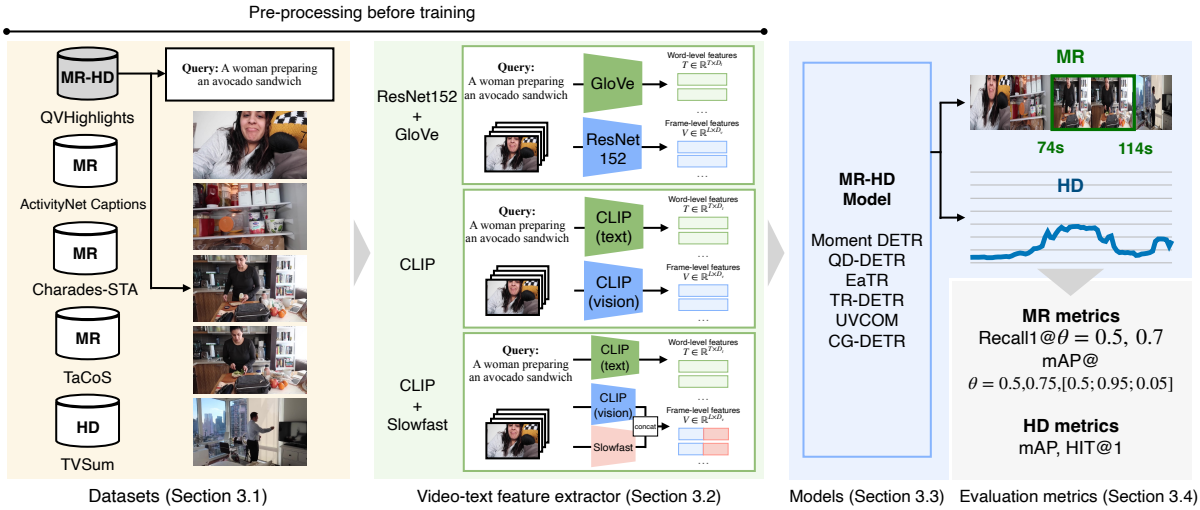

Figure 4: Overview of Lighthouse architecture for MR-HD training and evaluation. It consists of four components: datasets, video-text feature extractor, models, and evaluation metrics.

baseline. Given video and text features, the Transformer encoder concatenates and encodes them, then the Transformer decoder with query slots predicts both moments and saliency scores. Based on Moment DETR, QD-DETR focuses on enhancing query-moment similarity by introducing contrastive learning using query and different video pairs. EaTR improves Moment DETR by incorporating video and query information into the query slots. TR-DETR explores the reciprocal relationship between MR and HD to improve performance. UVCOM devises local and global encoding approaches based on the observation that a model shows different attention maps for MR and HD. Specifically, the attention map for MR emphasizes local moments in the videos, whereas, for HD, it highlights a global pattern. CG-DETR also focuses on the attention heatmap between video frames and queries. To achieve this, CG-DETR introduces an adaptive Cross Attention layer, which adds dummy tokens to the key in the multi-head attention to adjust relevancy between words and moments.

Extension to other model types. Currently, the models used are based on DETR, and the inference APIs are specifically designed for it. However, research by (Meinardus et al., 2024) has shown that the BLIP2-style (Li et al., 2023) auto-regressive approach outperforms DETR-based models, though it requires significantly more GPU resources (e.g., 8x NVIDIA A100 80GB GPUs for training). To integrate this into Lighthouse, we believe the frame and video-text feature extraction modules can be shared, and a wrapper class will be needed for the model's forward module. Extending support to other model types is planned for future work.

3.4 Evaluation metrics

We follow the evaluation metrics described in Lei et al. (2021). For MR, we provide Recall1@θ and mAP@θ. Recall1@θ represents the percentage of the top 1 retrieved moment with an IoU greater than θ with the ground-truth moment, where θ is set to be 0.5 and 0.7. mAP@θ denotes the mean average precision with θ set to 0.5 and 0.75, as well as the average mAP across multiple θ values ranging from 0.5 to 0.95 in increments of 0.05. For HD, we provide mAP, and HIT@1, which computes the hit ratio for the highest scored frame. Note that the frame is regarded as positive if it has a score of "Very Good (= 5)." QVHighlights consists of saliency scores from three annotators, HIT@1 is computed as the average of these annotators.

4 Experiments

We perform experiments on MR-HD, MR, and HD tasks individually. We used 1 NVIDIA A100 GPU (48GB) for all experiments. The hyperparameters used in this paper are the same as in the reference papers.

4.1 MR-HD results

Table 2 presents MR-HD results on the validation and test splits of QVHighlights, revealing three key insights. First, when comparing the reproduced results using CLIP+Slowfast with the reported scores, Lighthouse generally reproduces the

	val							test						
	MR					**HD**		**MR**					**HD**	
	R1		mAP			Very Good		R1		mAP			Very Good	
	@0.5	@0.7	@0.5	@0.75	avg	mAP	HIT@1	@0.5	@0.7	@0.5	@0.75	avg	mAP	HIT@1
ResNet152+GloVe														
Moment DETR	41.5	25.2	45.9	22.6	24.7	29.1	41.4	40.0	22.0	44.9	21.6	23.8	30.0	42.9
QD-DETR	53.2	37.5	55.4	34.5	34.5	34.1	52.1	52.7	36.1	55.4	33.9	33.7	33.8	50.7
EaTR	**54.9**	36.0	**56.7**	33.5	34.1	**35.1**	**54.7**	**57.2**	**38.9**	**59.6**	**35.6**	**36.7**	**36.3**	**57.4**
TR-DETR	48.3	32.9	49.5	28.6	29.6	34.2	51.4	47.7	31.6	49.8	29.3	29.4	34.3	52.0
UVCOM	53.7	**39.7**	55.9	**36.5**	**36.1**	34.9	53.0	53.8	37.6	55.1	33.4	34.0	34.8	53.8
CG-DETR	51.9	39.0	54.3	36.0	35.5	34.1	53.2	53.1	38.3	55.7	35.1	35.1	34.5	52.9
CLIP														
Moment DETR	53.5	34.1	56.2	30.8	32.4	35.3	54.0	55.8	33.8	58.2	31.2	32.7	35.7	55.8
QD-DETR	59.7	42.3	60.4	37.5	37.5	38.0	59.2	60.8	41.8	62.3	37.1	38.3	38.2	60.7
EaTR	54.9	36.0	56.7	33.5	34.1	35.1	54.7	54.6	34.0	57.1	32.6	33.2	34.9	54.7
TR-DETR	63.6	43.9	62.9	39.7	39.6	**40.1**	63.2	60.2	41.4	60.1	37.0	37.2	38.6	59.3
UVCOM	64.8	48.0	64.2	42.7	42.3	38.7	62.2	62.7	**46.9**	63.6	**42.6**	**42.1**	**39.8**	**64.5**
CG-DETR	**66.6**	**49.9**	**66.2**	**44.2**	**43.9**	39.9	**64.3**	**64.5**	46.0	**64.8**	41.6	41.8	39.4	64.3
CLIP+Slowfast (Reproduced scores)														
Moment DETR	54.2	36.1	55.3	31.5	32.6	35.9	56.7	54.4	33.9	55.2	29.7	31.5	32.6	56.7
QD-DETR	63.0	46.4	63.3	41.1	41.3	39.1	61.3	62.1	44.6	63.0	41.0	40.6	38.8	61.6
EaTR	59.6	40.3	60.9	38.1	38.0	36.6	57.9	57.2	38.9	59.6	35.6	36.7	36.6	57.9
TR-DETR	66.5	48.8	65.3	44.3	43.4	**40.8**	66.2	**65.2**	**48.8**	64.4	**43.0**	42.6	39.8	62.1
UVCOM	64.0	49.4	63.3	44.8	43.9	39.7	64.3	62.6	47.6	62.4	42.4	42.5	39.6	62.8
CG-DETR	**65.6**	**52.1**	**65.6**	**46.3**	**45.3**	40.7	**67.0**	64.9	48.1	**64.8**	42.8	**43.3**	**40.7**	**67.0**
Reported scores in the reference papers (CLIP+Slowfast)														
Moment DETR	53.9	34.8	-	-	32.2	35.7	55.6	52.9	33.0	54.8	29.4	30.7	35.7	55.6
QD-DETR	62.7	46.7	62.2	41.8	41.2	39.1	63.0	62.4	45.0	62.5	39.9	39.9	38.9	62.4
EaTR	61.4	45.8	61.9	41.9	41.7	37.2	58.7	-	-	-	-	-	-	-
TR-DETR	-	-	-	-	-	-	-	64.6	**48.9**	63.9	**43.7**	42.6	39.9	63.4
UVCOM	-	-	-	-	-	-	-	63.6	47.5	63.4	42.7	43.2	39.7	64.2
CG-DETR	**67.4**	**52.1**	**65.6**	**45.7**	**44.9**	**40.8**	**66.7**	**65.4**	48.4	**64.5**	42.8	**42.9**	**40.3**	**66.2**

Table 2: MR-HD results on the QVHighlights dataset. **Bold** values represent the best scores among methods with the same video-text feature.

reported scores. The models proposed in 2024, TR-DETR, UVCOM, and CG-DETR, achieve competitive performance among the methods. Second, CLIP+Slowfast generally achieves higher performance than CLIP alone, indicating that sequential motion information in videos is effective for MR-HD tasks in addition to frame-level appearance representations. Finally, CLIP-based features outperform ResNet152+GloVe, demonstrating the effectiveness of CLIP in the MR-HD task.

4.2 MR results

Table 3 presents the MR results. Although the insights gained are similar to the MR-HD results, we observe one different finding; later methods do not consistently outperform older ones across different datasets and features. For instance, in Charades-STA, QD-DETR with CLIP+Slowfast and ResNet152+GloVe achieves higher performance than CG-DETR and UVCOM. This suggests that there is no one-size-fits-all solution. To apply the methods to a custom MR dataset, users need to test multiple methods with different features. Lighthouse facilitates this trial-and-error process to achieve the best performance settings.

4.3 HD results

Table 4 presents the HD results on the TVSum dataset. In addition to our three backbones, we tested I3D+CLIP (Text) because previous studies used I3D (Carreira and Zisserman, 2017) and CLIP as visual and textual backbones. The findings are consistent with the MR results. First, the results demonstrate that Lighthouse can reproduce the reported scores. Second, we observe that newer methods do not always outperform older ones across different features. For example, when using CLIP, Moment DETR outperforms other approaches. Thus, Lighthouse is valuable for the HD community to test multiple methods with various features.

5 Conclusion

In this paper, we proposed Lighthouse, a user-friendly library for reproducible MR-HD. It supports six methods, five datasets, and three features. Lighthouse includes the inference API and web demo, enabling users to try MR-HD methods eas-

	ActivityNet Captions					Charades-STA					TaCoS				
	R1		mAP			R1		mAP			R1		mAP		
	@0.5	@0.7	@0.5	@0.75	avg	@0.5	@0.7	@0.5	@0.75	avg	@0.5	@0.7	@0.5	@0.75	avg
ResNet152+GloVe															
Moment DETR	34.2	19.5	46.3	24.4	26.2	38.4	22.9	52.4	22.2	26.2	20.0	8.6	24.2	6.9	10.1
QD-DETR	35.4	20.3	47.4	24.9	26.6	**42.1**	**24.0**	56.7	**24.5**	**28.7**	30.6	15.1	35.1	12.3	16.1
EaTR	32.4	18.2	44.3	21.9	24.1	37.6	20.1	53.5	23.6	27.0	22.5	9.2	26.3	7.9	10.7
UVCOM	34.4	19.9	46.1	24.4	25.9	38.1	18.2	54.4	21.1	25.6	24.1	10.7	28.1	8.6	12.0
CG-DETR	**37.0**	**21.2**	**48.6**	**26.5**	**28.0**	39.7	19.4	**56.9**	23.2	27.5	**34.2**	**17.4**	**39.7**	**14.6**	**18.7**
CLIP															
Moment DETR	36.1	20.4	48.2	25.7	27.5	47.9	26.7	61.0	28.8	31.9	18.0	7.9	21.3	6.7	9.3
QD-DETR	36.9	21.4	48.4	26.3	27.6	52.0	31.7	63.6	29.4	33.4	32.3	17.2	36.0	14.1	17.5
EaTR	34.6	19.7	45.1	23.1	24.9	48.4	27.5	59.9	26.9	30.9	24.7	10.0	28.8	8.7	11.8
UVCOM	37.0	21.5	48.3	25.7	27.4	48.4	27.1	60.9	27.9	31.4	**36.8**	**20.0**	**41.5**	**16.3**	**20.1**
CG-DETR	**38.8**	**22.6**	**50.6**	**27.5**	**28.9**	**54.4**	**31.8**	**65.5**	**30.5**	**34.5**	34.3	19.8	38.6	15.8	19.0
CLIP+Slowfast (Reproduced scores)															
Moment DETR	36.5	21.1	48.4	26.0	27.4	53.4	30.7	62.0	29.1	32.6	25.5	12.9	29.1	10.3	13.3
QD-DETR	37.5	22.1	48.9	26.4	27.8	**59.4**	**37.9**	**66.6**	**33.8**	**36.4**	38.7	22.1	42.9	16.7	20.9
EaTR	34.6	19.3	45.2	22.3	24.6	55.2	33.1	65.4	30.4	34.2	31.7	15.6	37.4	14.0	17.2
UVCOM	37.3	21.6	48.9	25.7	27.3	56.9	35.9	65.6	33.6	36.2	40.2	23.3	43.5	19.1	22.1
CG-DETR	**40.0**	**23.2**	**51.0**	**27.7**	**29.2**	57.6	35.1	65.9	30.9	35.0	**39.8**	**25.1**	**44.2**	**19.6**	**22.9**
Reported scores in the reference papers (CLIP+Slowfast)															
Moment DETR	-	-	-	-	-	52.1	30.6	-	-	-	24.7	12.0	-	-	-
QD-DETR	-	-	-	-	-	57.3	32.6	-	-	-	-	-	-	-	-
EaTR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UVCOM	-	-	-	-	-	**59.3**	**36.6**	-	-	-	36.4	**23.3**	-	-	-
CG-DETR	-	-	-	-	-	58.4	36.3	-	-	-	**39.6**	22.2	-	-	-

Table 3: MR results on the ActivityNet Captions, Charades-STA, and TaCoS datasets.

	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	avg
ResNet152+GloVe											
Moment DETR	87.5	**93.3**	91.5	79.7	**92.6**	85.1	70.0	91.8	87.9	79.7	85.9
QD-DETR	**90.8**	89.8	90.8	83.6	88.8	85.3	**79.6**	**95.1**	89.7	78.4	87.2
EaTR	87.9	87.2	89.0	**87.9**	85.8	90.1	73.2	92.3	89.4	78.7	86.2
UVCOM	87.8	92.6	**94.7**	80.7	88.7	**91.3**	76.0	94.0	90.1	**80.1**	**87.6**
CG-DETR	89.3	89.3	93.6	84.8	89.5	86.5	76.4	93.6	**90.2**	77.9	87.1
CLIP											
Moment DETR	**92.0**	**95.8**	**96.5**	87.3	89.0	89.9	**80.4**	92.6	87.8	79.5	**89.1**
QD-DETR	88.5	92.6	94.4	86.2	88.0	**91.9**	78.6	94.0	90.0	79.6	88.4
EaTR	86.4	94.1	90.9	84.9	83.8	88.9	77.9	92.5	**90.8**	76.8	86.7
UVCOM	90.1	92.4	95.8	86.5	86.8	89.2	76.5	**95.4**	87.7	76.1	87.7
CG-DETR	89.7	86.3	91.0	**90.6**	**90.6**	89.4	75.4	95.1	90.0	**83.2**	88.1
CLIP+Slowfast											
Moment DETR	85.0	**95.8**	91.6	88.2	85.8	85.2	76.3	91.8	88.0	81.3	86.9
QD-DETR	**90.3**	93.2	91.3	85.0	**90.9**	88.9	**78.6**	94.0	88.7	**82.9**	88.4
EaTR	87.1	93.7	89.5	84.6	88.5	84.5	73.4	91.4	88.8	79.9	86.1
UVCOM	89.6	92.8	91.4	87.4	87.9	86.9	76.3	**95.4**	**90.2**	79.5	87.7
CG-DETR	89.0	92.6	**96.3**	**92.0**	88.9	**89.2**	77.0	94.0	87.4	81.9	**88.8**
I3D+CLIP (Text) (Reproduced scores)											
Moment DETR	84.6	93.5	91.7	80.8	88.4	**91.4**	77.3	92.5	88.6	78.1	86.7
QD-DETR	89.9	86.6	91.1	85.9	88.7	88.9	74.2	**97.1**	88.3	80.0	87.1
EaTR	86.9	80.3	91.4	75.2	88.9	86.1	76.8	93.1	88.6	**82.5**	85.0
UVCOM	89.2	**92.4**	**94.4**	91.1	84.4	89.9	**77.8**	94.0	87.3	78.8	**87.9**
CG-DETR	**90.5**	83.1	94.2	**91.9**	**90.6**	88.6	76.1	94.0	**89.1**	81.0	**87.9**
Reported scores in the reference papers (I3D+CLIP (Text))											
Moment DETR	-	-	-	-	-	-	-	-	-	-	-
QD-DETR	**88.2**	87.4	85.6	85.0	85.8	86.9	76.4	91.3	**89.2**	73.7	85.0
EaTR	-	-	-	-	-	-	-	-	-	-	-
UVCOM	87.6	**91.6**	91.4	86.7	**86.9**	86.9	**76.9**	92.3	87.4	75.6	86.3
CG-DETR	86.9	88.8	94.8	**87.7**	86.7	**89.6**	74.8	**93.3**	**89.2**	**75.9**	**86.8**

Table 4: HD results on TVSum. mAP scores for each domain are displayed.

ily. Our experiments showed that Lighthouse reproduces the reported scores. In addition, we found that newer MR-HD methods do not consistently outperform older ones across MR/HD datasets and various features. Lighthouse aids researchers in the trial-and-error process, helping them achieve optimal performance settings.

6 Limitation and future work

This paper has two main limitations. First, we did not conduct a usability study to assess how the developed demos assist end users. We plan to address this in future work. Second, our models are based on DETR, and we did not implement other types of models. Recently, autoregressive approaches have been introduced in MR (Meinardus et al., 2024) based on large language models (Raffel et al., 2020). One of our future directions is to enhance Lighthouse by incorporating these approaches.

Acknowledgment

We grateful to Dr. Yusuke Fujita, Dr. Park Byeongseon, and Mr. Takuya Hasumi for providing insightful comments with this work. In addition, we thank anonymous reviewers and the area chair for providing reviews with us, which significantly improves our paper.

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Proc. ECCV*.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proc. ICCV*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*.

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

Jinhyun Jang, Jungin Park, Jin Kim, Hyeongjun Kwon, and Kwanghoon Sohn. 2023. Knowing where to focus: Event-aware transformer for video grounding. In *Proc. ICCV*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The kinetics human action video dataset. *arXiv*.

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

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. In *Proc. NeurIPS*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. ICML*.

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 *Proc. EMNLP*.

Boris Meinardus, Anil Batra, Anna Rohrbach, and Marcus Rohrbach. 2024. The surprising effectiveness of multimodal large language models for video moment retrieval. In *arXiv*.

WonJun Moon, Sangeek Hyun, SuBeen Lee, and Jae-Pil Heo. 2023a. Correlation-guided query-dependency calibration in video representation learning for temporal grounding. *arXiv preprint arXiv:2311.08835*.

WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. 2023b. Query-dependent video representation for moment retrieval and highlight detection. In *Proc. CVPR*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proc. EMNLP*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, and Sandhini Agarwal. 2021. Learning transferable visual models from natural language supervision. In *Proc. ICML*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *TACL*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv*.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *Proc. CVPR*.

Hao Sun, Mingyao Zhou, Wenjing Chen, and Wei Xie. 2023. Tr-detr: Task-reciprocal transformer for joint moment retrieval and highlight detection. In *Proc. AAAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

Yicheng Xiao, Zhuoyan Luo, Yong Liu, Yue Ma, Hengwei Bian, Yatai Ji, Yujiu Yang, and Xiu Li. 2024. Bridging the gap: A unified video comprehension framework for moment retrieval and highlight detection. In *Proc. CVPR*.

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