

Supplementary Materials: Reversed in Time: A Novel Temporal-Emphasized Benchmark for Cross-Modal Video-Text Retrieval

Anonymous Authors

1 TEMPORAL ANALYSIS OF MSR-VTT

MSR-VTT is one of the most extensively utilized benchmarks for video-text retrieval. We randomly sample 100 videos from the MSR-VTT test set to form a subset and manually assess the temporal relevance of videos based on the video and its corresponding text using the following rules: 1) the video contains a distinct temporal-related activity, such as open/close; or 2) the video contains consecutive activities with significant differences; or 3) the video involves an apparent change in the state of an object; or 4) the video contains observable changes in the position of an object; 5) the corresponding text fully describes the temporal changes presented in the video.

Following these rules, we find that only approximately 10% of the videos in the subset demonstrate temporal relevance. This observation highlights that the MSR-VTT test set mainly focuses on static information and lacks consideration of temporal aspects. Consequently, the absence of harder-negatives in the test set allows models to retrieve temporally relevant videos based solely on static cues, making it insufficient to evaluate the temporal understanding capability of video-text retrieval methods. We visualize some examples in Fig. 1.

2 HUMAN-IN-THE-LOOP VERIFICATION

We put human in the verification loop to control the data quality.

In **Seed Activity List Proposal**, we conduct a comprehensive examination of the action pairs generated by GPT-4. We eliminate actions that lack temporal relevance, cannot be detected through video, have mismatches between corresponding actions, or are rare.

In **Activity List Enrichment**, we conduct a comprehensive examination of the verb-noun phrases generated by GPT-4 and eliminate phrases that are rare or unreasonable.

In **Raw Video Acquisition**, to improve the overall quality, we recruit seven workers to search videos using a video search engine. They filter out activities that meet the following criteria: 1) the activity can be identified without relying on temporal information. 2) the number of videos retrieved using this activity as a query is less than 50. 3) less than 50% of all the videos retrieved based on this activity correctly match this activity.

In **Manual Annotation**, we employ the following processes to ensure the quality: 1) Training of Quality Assurance (QA) Personnel: Project manager provides training to the QA personnel, explaining the filtering and annotation guidelines while providing them with examples. 2) First Round of Trial by QA Personnel: The project manager meticulously review the samples annotated by the QA personnel, providing detailed feedback and revisions to ensure their understanding of the task aligned with the project manager’s expectations. 3) QA Personnel Supervision of Eight Annotators: Each annotator watches the training video provided by the project manager and underwent comprehensive QA inspection of their

annotated samples. Similar to the previous step, iterative feedback and revisions are given to rectify any misunderstandings and ensure consistency in the annotations.

3 PROMPTS FOR GPT-4

We leverage GPT-4 in our dataset construction process and we present our prompts for GPT-4 below.

Seed Activity List Proposal in Step1. In this phrase, we provide GPT-4 with a few action pairs in initial list and instruct it to generate more samples. Our prompt is demonstrated in Table 1.

Activity List Enrichment in Step2. In this phrase, we prompt GPT-4 to substitute [something] in each activity list with concrete objects to form a verb-noun activity list. Our prompt is demonstrated in Table 2.

Rewriting for Diversity in Step3. In this phrase, we provide GPT-4 with the human-written caption, and instruct it to rewrite nine extra sentences. Our prompt is demonstrated in Table 3.

4 MORE STATISTICS ABOUT THE RTIME

Some activities (verb-noun combinations) and a word-cloud based on the distribution of verb phrases are illustrated in Figure 2 and Figure 3.

To assess the quality of GPT-4 generated captions, we calculate the cosine similarity score between the manually annotated captions and the rewritten captions for the same videos base on their BERT [5] embedding. For comparison, we also randomly sample captions from other videos and compute the cosine similarity scores. As depicted in Figure 4, the captions generated by GPT-4 have higher similarity scores with the human-written captions, indicating that the rewritten captions relatively retain the original meaning.

5 DETAILS OF COMPARED SOTA METHODS

- **CLIP [14]** is an image-text model pre-trained on 400M image-text paired data. It includes a Visual Transformer (ViT) as image encoder and a Transformer with casual mask as text encoder. An image-text contrastive loss is used to cross-modal alignment. During inference, a mean pooling is applied to aggregate multi-frame features.

- **BLIP [8]** is an image-text pre-trained model with ViT as the image encoder and a Transformer as the text encoder. It employs image-text contrastive loss and image-text matching loss for cross-modal alignment. During inference, a mean pooling is applied to aggregate multi-frame features.

- **CLIP4Clip [12]** adds a temporal transformer on top of CLIP’s image encoder to enable cross-frame interaction, producing the video-level feature. A video-text contrastive loss is used to align video and text.

- **TS2Net [11]** is based on CLIP. It has a token shift module and token selection module in the video encoder to further enhance



Figure 1: Illustration of some video-text samples in MSR-VTT. (A): samples demonstrating temporal relevance. (B): samples without temporal relevance.

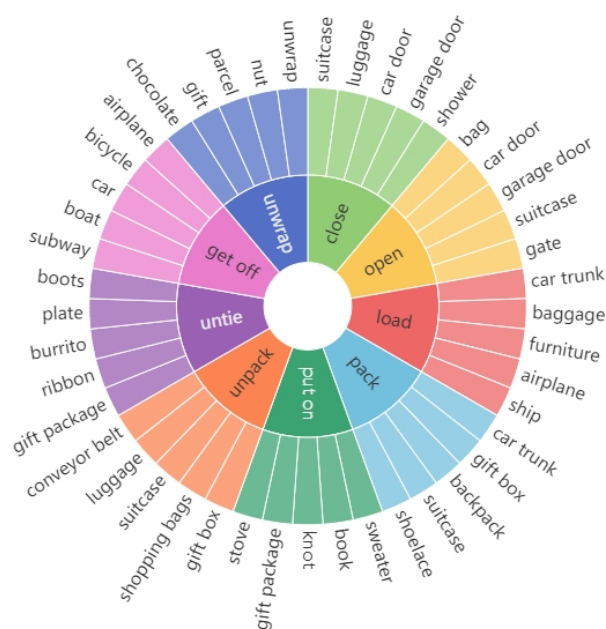


Figure 2: Some example verbs (inner circle) and their top 5 noun objects (outer circle) in the activity list from RTime

the video representation. It also uses video-text contrastive loss to align video and text.

- **Singularity** [7] uses ViT as the visual encoder and a Transformer as the text encoder. It employs video-text contrastive, masked language modeling, and video-text matching losses in training. It randomly samples a frame from a video in pre-training, and concatenates multi-frame features in inference. We use the checkpoint pre-trained on 17 million visual-text pairs, including WebVid-2M



Figure 3: Word-cloud of verb phrases in RTime dataset

[1], CC3M [15], COCO [10], Visual Genome [6], SBU Captions [13] and CC12M [3, 15].

- **VINDLU** [4] provides a video-and-language pre-training recipe. It implements several video encoders, text encoders, objective functions. It pre-trained on 25 million visual-text pairs, including CC3M [15], COCO [10], Visual Genome [6], SBU Captions [13], CC12M [3] and WebVid-10M [1].

- **UMT [9]** has the same architecture as VINDLU for video and text encoder. It utilizes a two-stage pretraining process with the CLIP image encoder as the teacher, employing a masking strategy to reduce training costs, and incorporates spatio-temporal attention mechanisms [2] to facilitate cross-frame interactions. It pre-trained on the same data as VINDLU.

For all the compared models, we follow their original experimental setup conducted on the MSR-VTT dataset. Regarding the fine-tuning process, we perform fine-tuning for 5 epochs with a batch size of 128.

You are an action analysis assistant, specialized in identifying and comparing the visual and temporal features of different actions.

Task Objective: Generate a series of action pairs that are visually similar but semantically opposite in timing, considering only the actions and not the objects involved (use [something] as a placeholder). Ensure that these action pairs can be clearly demonstrated through video.

Specific Steps:

1. Choose a common action, such as 'open [something]'.
2. Determine the direct antonym action, such as 'close [something]'.
3. Ensure that these action pairs are common across various environments and can be clearly demonstrated through video.
4. Repeat the above steps to generate more action pairs.

Output Format: Use JSON format for output. Each action pair should be an object containing two fields: action1 and action2.

Example Output:

```

...
[
  {"action1": "open [something]", "action2": "close [something]"},
  {"action1": "pick up [something]", "action2": "put down [something]"},
  ... // more action pairs
]
...

```

Table 1: Prompts used in Seed Activity List Proposal

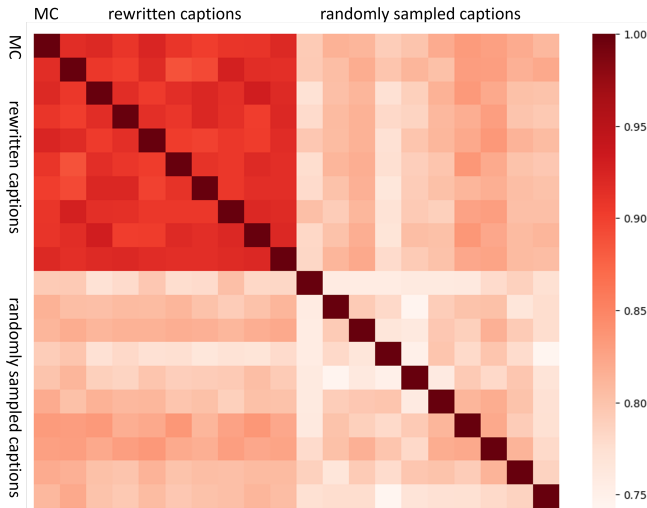


Figure 4: Cosine similarity score between different captions based on their BERT embeddings. 'MC' means manual captions for corresponding videos

REFERENCES

- [1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1728–1738.
- [2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In *ICML*, Vol. 2. 4.
- [3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3558–3568.
- [4] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. 2023. Vindlu: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10739–10750.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [6] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123 (2017), 32–73.
- [7] Jie Lei, Tamara Berg, and Mohit Bansal. 2023. Revealing Single Frame Bias for Video-and-Language Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 487–507.
- [8] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.
- [9] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. 2023. Unmasked Teacher: Towards Training-Efficient Video Foundation Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 19948–19960.
- [10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 740–755.
- [11] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. 2022. Ts2-net: Token shift and selection transformer for text-video retrieval. In *European Conference on Computer Vision*. Springer, 319–335.
- [12] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.
- [13] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems* 24 (2011).
- [14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

You are a professional motion analyst. Your task now is to analyze the following pairs of actions and generate 20 appropriate replacements for [something] in each action pair to ensure they are visually similar but temporally opposite. Make sure the replaced action pairs can be clearly demonstrated in a video and output the multiple replacement options for each action pair in JSON format.

Processing steps:

- 1.Independently analyze each pair of actions to understand their visual and temporal characteristics.
- 2.Choose 20 suitable objects or actions for [something].
- 3.Ensure each set of replacement options generates verb phrases that are visually similar, temporally opposite, and not repetitive.
- 4.Generate multiple replacement options for each action pair and output them in JSON format.

Example:

Input:

```
'''
[
  {"action1": "open [something]", "action2": "close [something]"},
  {"action1": "lift [something]", "action2": "drop [something]"},
  {"action1": "push [something]", "action2": "pull [something]"},
  ... // more action pairs
]
```

Output:

```
'''
{
  "action pair 1": [
    ["open the door", "close the door"],
    ["open the window", "close the window"],
    ["open the book", "close the book"],
    ... // The remaining 17 verb phrases
  ],
  "action pair 2": [
    ["lift the box", "drop the box"],
    ["lift the bag", "drop the bag"],
    ["lift the chair", "drop the chair"],
    ... // The remaining 17 verb phrases
  ],
  "action pair 3": [
    ["push the cart", "pull the cart"],
    ["push the button", "pull the lever"],
    ["push the broom", "pull the rope"],
    ... // The remaining 17 verb phrases
  ],
}
```

Table 2: Prompts used in Activity List Enrichment

[15] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic

image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2556–2565.

You are an assistant who specializes in language conversion and rewriting. Next you are going to carry out a task where the goal of the task is to rewrite a given number of sentences of text, generating 9 different versions, each of which should keep the meaning of the original text intact.

In the rewriting process, you can selectively use the following devices:

1. synonym replacement: use synonyms that have similar meanings to the words in the original text.
2. Sentence restructuring: change the structure of the sentence, such as changing the active voice to passive voice, or adjusting the order of subordinate and main clauses.
3. Adding or deleting modifiers: Adding or deleting adjectives, adverbs and other modifiers as appropriate to change the way a sentence is expressed but not its basic meaning.
4. use different grammatical structures: e.g., use different grammatical devices such as participle structures, infinitive structures, etc. to express the same meaning.

Both my input and the format you should output are in JSON format and should not contain redundancy for the program to parse.

```
'''
{
  "text1": "The rose flowers are placed on a turntable to rotate, and then the petals float down.",
  "text2": "On a white table there are four connected working gas stoves and then the flames go out
one by one.",
  // Other more input sentence cases
}
'''
```

Sample output is as follows:

```
'''
{
  "text1":{
    "original": "The rose flowers are placed on a turntable to rotate, and then the petals float down.",
    "rewrites": [
      "The rose blossoms are set on a revolving platform, causing the petals to drift to the ground.",
      "Placed upon a rotating turntable, the rose flowers spin, allowing their petals to fall gently.",
      "Rose petals descend gracefully as the flowers are spun on a turntable.",
      "... (the remaining 6 rewritten versions)"
    ]
  }
  "text2":{
    "original": "On a white table there are four connected working gas stoves and then the flames go out
one by one.",
    "rewrites": [
      "Four active gas stoves are positioned on a white table, with their flames extinguishing
sequentially.",
      "There are four gas stoves linked together on a white table, and their flames are snuffed out
successively.",
      "A quartet of operational gas stoves sits atop a white table, and extinguish in a one-after-another
fashion.",
      "... (the remaining 6 rewritten versions)"
    ]
  }
  // Other more output
}
'''
```

Table 3: Prompts used in Rewriting for diversity