

Open-source Human Evaluation Framework for Video-to-Text and Video-to-Audio Systems

Goran Topić¹, Graham Neubig², Katsuhito Sudoh⁵, Yuki Saito³, Shinnosuke Takamichi⁴,
Ryosuke Matsushita⁴, Kota Iura³, Hiroya Takamura¹, Tatsuya Ishigaki¹
¹AIST, ²CMU, ³The University of Tokyo, ⁴Keio University, ⁵Nara Women’s University

Abstract

We present a framework that streamlines the preparation of human evaluation process for text or audio automatically generated from video. In such evaluation tasks evaluators often assess the generated text or audio while watching a video. Consequently, preparing for these evaluations can be highly resource-intensive because the process typically involves several steps cutting a video and audio segments, synthesizing speech with text-to-speech tools, merging audio and video, and developing a user interface for crowdsourcing annotation collection. Our framework automates these steps, reducing the researchers’ workload.

1 Introduction

Video-to-text systems, such as dense video captioning [1] and commentary generation [2, 3, 4], have made remarkable progress. Recently, there has been growing interest in combining these systems with text-to-speech (TTS) technologies to also produce audio outputs, enabling a more immersive evaluation of video content [5, 6]. Evaluating these models is a key area of research. While automatic evaluation metrics, which are mainly based on word overlap or embedding similarities, are commonly used, it is common practice to combine human evaluation with automatic evaluation for assessing these models.

Preparing an evaluation process is resource-intensive for researchers. It typically involves several labor-intensive steps, including: 1) video preparation, 2) defining cuts for video segments, 3) generating text using language models, 4) creating subtitles, 5) synthesizing speech with text-to-speech technology, 6) merging audio and video, and 7) developing a user interface for crowdsourcing annotation collection. Currently, these tasks are often carried out independently by each research institution, leading to redundant

efforts and inefficiencies. The lack of a standardized framework not only increases costs but also hinders progress in evaluating video-to-text systems effectively.

Evaluating video-to-text and video-to-audio tasks using existing annotation tools is challenging. While tools like brat [7] and commercial platforms such as VoTT¹⁾ and LabelBox²⁾ may be suitable for annotation tasks involving single-modal data, such as text or video alone, they are often not well-suited for multimodal tasks. These tools are primarily designed for annotations involving one modality—either text or visual content—and typically require significant preprocessing to handle multimodal inputs. For example, they may need additional steps like adding audio or subtitles to videos before annotation, which increases the overall workload and makes them less efficient for evaluating video-to-text and video-to-audio systems.

In contrast, this paper presents an open source framework that streamlines the human evaluation workflow. Our framework automates key steps in the preparation, such as muxing, cutting, and UI development, which are essential for creating evaluation tasks presented to evaluators.

2 Related Work

Various video-to-text and video-to-audio tasks and datasets have been proposed including dense video captioning [1], commentary generation [2, 3, 4, 5], speech corpus of commentary [8]. For video-to-text generation tasks, existing studies employ three main strategies for evaluation: 1) using only text outputs, 2) using both video and text outputs, and 3) using both video and audio outputs.

For the first strategy, common in dense video captioning tasks, generated texts are often aligned with gold standard texts using IoU [1] or by solving an optimization problem [9]. Metrics such as BLEU [10], METEOR [11],

1) <https://github.com/microsoft/VoTT>

2) <https://labelbox.com>

and CIDEr [12] are then used for scoring. Preparation of evaluation process in this strategy is easier, however, these metrics have limitations in capturing nuanced aspects of language generation, so many studies supplement them with human evaluations.

Recent studies have adopted the second or third strategies, where evaluators assess outputs by viewing the video alongside subtitles/audio. While these approaches offer more comprehensive evaluations, they also require more effort and resources to prepare the evaluation process.

Sometimes, existing annotation tools are used for human annotation process, but most of them are aimed at either textual [7] or visual annotation, not multimodal.

3 Framework

3.1 Overview

Conventional Evaluation Flow: As shown in Figure 1, the conventional evaluation process involves several steps. First, a video is prepared, assuming it to be a full-length video for evaluation. Then, is generated using a model tailored for a specific task, such as dense video captioning or commentary generation. Since evaluating a long full-length video is often impractical for human evaluators, the video is divided into shorter segments. To do this, the length of each segment is defined, and the full-length video is manually cut using video editing software or by writing scripts using libraries such as MoviePy³⁾ or FFmpeg⁴⁾. If subtitles should be shown, the subtitles will also need to be split accordingly. If audio commentary is to be included for the evaluators, it might be synthesized by text-to-speech systems and merged with the video. Finally, a user interface needs to be created for the annotation platform for local evaluators or crowdworkers in e.g., Amazon Mechanical Turk⁵⁾ or Lancers⁶⁾.

Evaluation Flow in Our System: Our framework is a Django-based web application that simplifies these steps. We ask the user, i.e., the person requesting the evaluation tasks (not the evaluators), to upload several types of files:

1. A video file, which may include an audio track e.g., sound effects and background music for a game.
2. Automatically generated commentary or captions in

the format of subtitles and/or audio, which are synchronized with the video. We can upload both subtitles and audio.

3. A JSON file defining the start and end timestamps for each segment of the video to be evaluated.

Our framework then automates cutting the full-length video, merging each segment with audio or subtitles, and creating a web-based interface, as shown in Figure 2 for crowdsourcing services. This automation significantly reduces the workload for researchers.

The proposed system serves two types of users: 1) evaluators and 2) administrators. Evaluators perform evaluation tasks by annotating video segments with scores or comments based on the audio and/or subtitles. Administrators, on the other hand, design the evaluation tasks, assign evaluators, and oversee task management. We describe our framework from these two perspectives in the following subsections. We also present other functionalities including a flexible user management and connecting other services e.g., crowdsourcing services and AWS S3 storage.

3.2 The Admin Page:

On this page, admin users can create and manage evaluation tasks. Creating evaluation tasks involves two main steps: 1) uploading the full-length videos and a JSON file that defines the segments to be evaluated, and 2) specifying the evaluation criteria.

Uploading contents to be evaluated: As shown in Figure 3, an administrator uploads a full-length video and a JSON file specifying the start and end timestamps (in seconds) of each cut segment. An example of the JSON format is as `[[0, 7.4], [10, 16], [16]]`. This JSON represents a cut definition where the first segment starts at a timestamp of 0s and ends at 10s, followed by two additional segments. If no end is specified, the rest of the video is used.

Admins can optionally upload a subtitle file and/or an audio file, which are assumed to contain synchronized captions or audio commentary for the video. The subtitle file should be in one of WebVTT, SRT, SBV and CSV/TSV formats, and the audio file should be MP3, AAC, or WAV.

Defining Evaluation Criteria: The next step for the admin user is to define the evaluation questions that will be presented to the evaluators. For example, when evaluating a video and its associated audio commentary for delay,

3) <https://github.com/Zulko/moviepy>

4) <https://ffmpeg.org>

5) <https://www.mturk.com>

6) <https://www.lancers.jp>

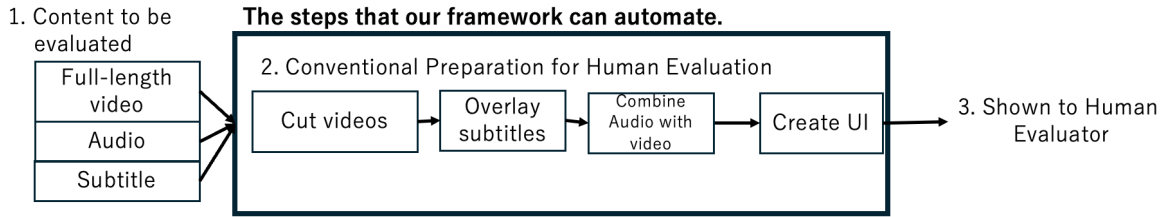


Figure 1: Conventional evaluation flow and the parts that our frame work can automate.



Please watch the video while listening to the audio. Determine if the audio is delayed or acceptable.

☐ without any problems in terms of delay
☐ audio is slightly delayed
☐ audio is obviously delayed
☐ audio is slightly ahead of video
☐ audio is obviously ahead of video.

If you encountered a problem, please describe it here.

Submit

Figure 2: The evaluation page automatically generated by our system⁷⁾.

Dataset Video

Name

Video file (.mp4)

Choose File

No file chosen

Subtitle file (.vtt, .srt, .sbv, .csv or .tsv)

Choose File

No file chosen

Audio file (.mp3, .aac or .wav)

Choose File

No file chosen

Cut file (.json)

Choose File

No file chosen

Submit

Figure 3: The page for uploading video, the cut definition, subtitles, and audio commentary.

the system might present a question such as, “Determine if the audio is delayed or acceptable,” along with multiple choice options like “audio is slightly delayed” or “audio is slightly ahead of the video.”. To define these criteria, the administrator prepares a JSON file containing the question definitions:

```

1 [
2   {
3     "id": "delay",
4     "instruction": "<p>Please watch
the video while listening to

```

7) The sample is CC-BY video “The Cutest Octopus” by vlogbrothers at <https://www.youtube.com/watch?v=xHYTSJWzpbs>

```

the audio. Determine if the
audio is delayed or acceptable
.</p>",
5  "type": "radio",
6  "options": [
7    { "value": 0, "text": "without
any problems in terms of
delay" },
8    { "value": 1, "text": "audio is
slightly delayed" },
9    { "value": -1, "text": "audio
is obviously delayed" },
10   { "value": 2, "text": "audio is
slightly ahead of video" },
11   { "value": -2, "text": "audio
is obviously ahead of video
." }
12 ]
13 },
14 {
15   "id": "problem",
16   "instruction": "<p>If you
encountered a problem, please
describe it here.</p>",
17   "type": "text"
18 }
19 ]

```

After uploading the dataset files and defining the evaluation criteria as described in this section, our system automatically cuts the full-length video into smaller segments and combines them with audio and/or subtitles. Finally, the system creates web-based annotation user interface with the video, questions, and choices.

3.3 The Evaluator Page

Figure 2 shows the page displayed to evaluators. The target video segment defined by the administrator is shown on the left, while the questions and choices are displayed on the right. The video is shown with the commentary audio, and subtitles are overlaid if available. Once the evaluator submits their answers, the evaluation is saved. The page for the next video segment is displayed next.

3.4 Filtering and Downloading Results

When the evaluation work is done, the administrators inspect the submitted evaluations, and either approve or reject them, to filter out the cases where e.g. the worker misunderstood the task or the form was submitted empty. For convenience, an administrator can approve all non-rejected assignments at once. The results of evaluation can be downloaded as a JSON file:

```
1 {
2   "project": "Evaluation in terms of
3   "tasks": [
4     {
5       "name": "Demo Video",
6       "start": 0.0,
7       "end": 5.0,
8       "evaluations": [
9         {"delay": -1, "problem": "N/A"}
10        {"delay": 2, "problem": "No
11          audio"}
12      ]
13    }
14  ]
15 }
```

3.5 Support for Large-scale Evaluations

We provide two functions to support large scale evaluations; and flexible user management. In real-world scenarios, crowdsourcing services like Amazon Mechanical Turk are often utilized. Additionally, large-scale cloud storage solutions, such as AWS S3, are frequently required to stream videos to these crowdsourcing platforms. Managing such extensive evaluation tasks typically requires two or more administrators to oversee the project. To address these needs, our system includes the functionalities described in this subsection.

Connecting to Crowdsourcing: Crowdsourcing services

like Amazon Mechanical Turk (MTurk) and Lancers allow researchers to recruit and compensate workers beyond their immediate environment. MTurk, as one of the most widely known platforms, enables administrators to create tasks, approve or reject assignments, and retrieve results. Our system directly supports MTurk, simplifying the process through integration with the AWS API. By entering their AWS credentials and configuring MTurk settings in the project properties, administrators can seamlessly manage crowdsourcing tasks.

Cloud Storage Support: Effective use of crowdsourcing services often requires providing workers with access to media files, such as videos and subtitles. This requires hosting the files on publicly accessible URLs, which can be challenging if the application is not deployed on a public server. Additionally, video files can be large, making storage a significant concern. To address these challenges, our system integrates with Amazon S3 (Simple Storage Service). If AWS credentials and an S3 bucket location are specified, the system automatically uploads segment files (e.g., muxed video and optional subtitle files) to S3. This ensures that files are accessible to workers and alleviates storage limitations on local servers.

Scriptable data upload: Uploading videos one-by-one through a web interface could be not suitable when the dataset is large. To this end, we provide an endpoint where videos can be uploaded via the 'curl' command. This should make it simple to upload videos in bulk.

Flexible User management: A large evaluation projects often require collaborations by several administrators. Therefore, this administrator can create further admin users who can upload videos and create evaluation projects.

4 Conclusion

In this paper, we introduced an open-source framework aimed at simplifying and standardizing the preparation process for human evaluations in video-to-text and video-to-audio tasks. By addressing the complexity and resource-intensive nature of current evaluation workflows, our framework integrates subtitle creation, TTS synthesis, audio-video merging, and crowdsourcing interface development into a unified process. This approach helps reduce both the cost and effort required for preparation. In the future, we plan to evaluate the system from the perspectives of user experience.

Acknowledgments

This study is based on results obtained from a project, Programs for Bridging the gap between R&D and the IDEal society (society 5.0) and Generating Economic and social value (BRIDGE)/Practical Global Research in the AI × Robotics Services, implemented by the Cabinet Office, Government of Japan.

References

- [1] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In **Proceedings of the IEEE international conference on computer vision**, pp. 706–715, 2017.
- [2] Zihan Wang and Naoki Yoshinaga. Commentary generation from data records of multiplayer strategy esports game. In Yang (Trista) Cao, Isabel Papadimitriou, Anaëlia Ovalle, Marcos Zampieri, Francis Ferraro, and Swabha Swayamdipta, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)**, pp. 263–271, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [3] Edison Marrese-Taylor, Yumi Hamazono, Tatsuya Ishigaki, Goran Topić, Yusuke Miyao, Ichiro Kobayashi, and Hiroya Takamura. Open-domain video commentary generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 7326–7339, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [4] Tatsuya Ishigaki, Goran Topic, Yumi Hamazono, Hiroshi Noji, Ichiro Kobayashi, Yusuke Miyao, and Hiroya Takamura. Generating racing game commentary from vision, language, and structured data. In Anya Belz, Angela Fan, Ehud Reiter, and Yaji Sripada, editors, **Proceedings of the 14th International Conference on Natural Language Generation**, pp. 103–113, Aberdeen, Scotland, UK, August 2021. Association for Computational Linguistics.
- [5] Tatsuya Ishigaki, Goran Topić, Yumi Hamazono, Ichiro Kobayashi, Yusuke Miyao, and Hiroya Takamura. Audio commentary system for real-time racing game play. In C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß, editors, **Proceedings of the 16th International Natural Language Generation Conference: System Demonstrations**, pp. 9–10, Prague, Czechia, September 2023. Association for Computational Linguistics.
- [6] Erica Kido Shimomoto, Edison Marrese-Taylor, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. Introducing spatial information and a novel evaluation scheme for open-domain live commentary generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 10352–10370, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [7] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In Frédérique Segond, editor, **Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 102–107, Avignon, France, April 2012. Association for Computational Linguistics.
- [8] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. SMASH corpus: A spontaneous speech corpus recording third-person audio commentaries on gameplay. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 6571–6577, Marseille, France, May 2020. European Language Resources Association.
- [9] Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Soda: Story oriented dense video captioning evaluation framework. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, **ECCV (6)**, Vol. 12351 of **Lecture Notes in Computer Science**, pp. 517–531. Springer, 2020.
- [10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [11] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, **Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization**, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.
- [12] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In **CVPR**, pp. 4566–4575. IEEE Computer Society, 2015.