MULTISCRIPT: Multimodal Script Learning for Supporting Open Domain Everyday Tasks

Jingyuan Qi*, Minqian Liu*, Ying Shen, Zhiyang Xu, Lifu Huang

Department of Computer Science, Virginia Tech
jingyq1@vt.edu, minqianliu@vt.edu, yings@vt.edu, zhiyangx@vt.edu, lifuh@vt.edu

Abstract

Automatically generating scripts (i.e. sequences of key steps described in text) from video demonstrations and reasoning about the subsequent steps are crucial to the modern AI virtual assistants to guide humans to complete everyday tasks, especially unfamiliar ones. However, current methods for generative script learning heavily rely on well-structured preceding steps described in text and/or images or are limited to a certain domain, resulting in a disparity with real-world user scenarios. To address these limitations, we present a new benchmark challenge – MULTISCRIPT, with two new tasks on task-oriented multimodal script learning: (1) multimodal script generation, and (2) subsequent step prediction. For both tasks, the input consists of a target task name and a video illustrating what has been done to complete the target task, and the expected output is (1) a sequence of structured step descriptions in text based on the demonstration video, and (2) a single text description for the subsequent step, respectively. Built from WikiHow, MULTISCRIPT covers multimodal scripts in videos and text descriptions for over 6,655 human everyday tasks across 19 diverse domains. To establish baseline performance on MULTISCRIPT, we propose two knowledge-guided multimodal generative frameworks that incorporate the task-related knowledge prompted from large language models such as Vicuna. Experimental results show that our proposed approaches significantly improve over the competitive baselines.

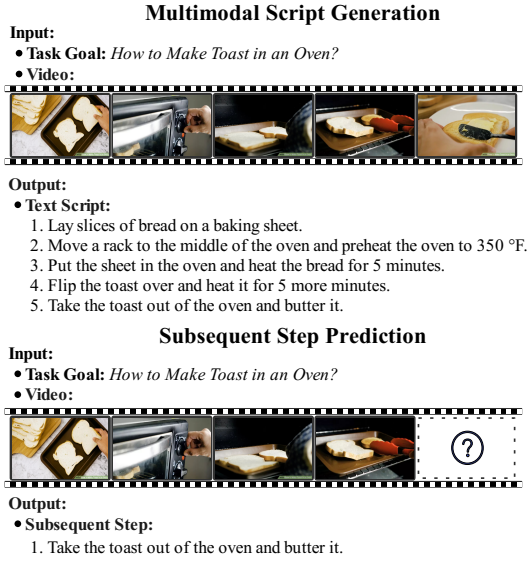

Figure 1: An example from MultiScript. In order to generate the subsequent step, the model needs to understand the key steps from the video and gain sufficient instructional knowledge about "how to make toast in an oven".

Introduction

Recently, there has been an increasing focus on studies aimed at enabling AI assistants to better assist individuals in completing daily tasks, and two crucial aspects of these studies are to automatically: (1) provide a well-structured script, i.e., a sequence of standardized steps, as guidance for the target task; and (2) reason about the subsequent steps based on what has been done by the individual. Taking the task "`Make a toast in an oven`" as an example, a script for achieving this task can be illustrated as: *preheating the oven*, *putting the sheet in the oven and heating the bread*, *flipping the toast over*, and *taking the toast out of the oven and buttering it*. While such script knowledge is typically authored by humans based on their experience or by watching video demonstrations, how to automatically acquire it and apply it to support the wide variety of everyday tasks is still an open research problem.

Existing studies on script learning can be categorized into two task formats: (1) *multiple-choice question answering*, which is to select the most plausible subsequent step from a few candidates by giving a task goal and a preceding step (Yang et al. 2021a). Though such candidate-based formulation is a reasonable way to test models' capability of selecting the correct steps, it is not feasible for real-world scenarios as the candidates are usually not available. (2) *generative script learning*, which requires the model to generate a future step given a task goal and a sequence of preceding steps described in text and/or images (Lyu, Zhang, and Callison-Burch 2020; Wang et al. 2022b). While this setup is more challenging, it assumes that the preceding steps are well-structured, which is not realistic either for actual AI assistants as in many use cases, they can only access preceding

*These authors contributed equally.

human actions from real-time or recorded video demonstrations that do not have clear boundaries between steps.

To address these limitations and enable AI Assistants to better guide humans to complete everyday tasks, we propose a new benchmark challenge – MULTISCRIPT on task-oriented multimodal script learning with two novel tasks: (1) **Multimodal Script Generation**, which aims to extract and summarize all the key and necessary steps into structured text based on the target task and a video demonstration provided as input; and (2) **Subsequent Step Prediction**, of which the goal is to generate the most plausible and logical subsequent step based on the preceding steps demonstrated in an input video and the target task. MULTISCRIPT consists of a large-scale dataset for each of the two tasks, covering a wide range of 6,655 everyday tasks across 19 domains, such as *Finance*, *Education*, etc. Figure 1 shows an example for each of the two tasks. We identify two critical challenges presented in MULTISCRIPT: first, it requires accurate understanding and identifying all the key steps contained in the video as well as the target goal; second, it requires the model to gain sufficient background knowledge on various tasks and their necessary steps to correctly generate a complete script or a subsequent step for the target task, especially for the unseen tasks during inference.

To tackle these two challenges and establish baseline performance for the two tasks presented in MULTISCRIPT, we propose a new knowledge-guided generative framework where we take a large-scale pre-trained language model (LLM) such as Vicuna (Zheng et al. 2023) as a task-agnostic universal knowledge source and dynamically induce and incorporate task-related knowledge to guide the model to generate a script or a subsequent step based on the input video demonstration. To relieve the effect of unrelated and/or incorrect external knowledge due to the randomness and hallucination of LLMs, we develop a natural language inference-based selector to selectively incorporate the prompt knowledge into the generation process. Extensive experiments demonstrate that our proposed methods substantially outperform the competitive baselines[1].

Dataset Design

Task Formulation

Multimodal Script Generation To automate the acquisition of script knowledge from massive video demonstrations for human everyday tasks, we introduce Multimodal Script Generation (MSG) task that aims to generate a structured text script by giving the task goal and a full video demonstration. More formally, we denote the task goal as T, input video demonstration as V, and the target text script as $\mathcal{S}$=$\{S_1,...,S_n\}$ involving n necessary and ordered steps. Compared to action anticipation (Girdhar and Grauman 2021; Zhong et al. 2022) or video captioning (Singh, Singh, and Bandyopadhyay 2020), the generated scripts in our task are expected to be well-structured descriptions for a sequence of actions that follow a temporal and logical order.

Subsequent Step Prediction To support the research of assisting humans to complete various daily tasks, we propose the Subsequent Step Prediction (SSP) task. Formally, given a task goal T and a partial video V_{i-1} that presents the preceding steps $\{S_1, ..., S_{i-1}\}$ that have been completed, a model needs to predict a subsequent step S_i. As there could be several subsequent steps following S_{i-1}, we define S_i as the *most plausible and logically reasonable step that is in a correct temporal order* given the task goal T and preceding steps shown in V_{i-1}.

Instructional Article Collection

We use WikiHow[2] as the source to build MULTISCRIPT as it contains multimedia instructional articles, including key step descriptions in text and/or images and optional video demonstrations, for diverse open-domain human everyday tasks. To collect the WikiHow articles, we develop data collection programs based on (Zhang, Lyu, and Callison-Burch 2020) and select 6,652 multimedia instructional articles, where each article must satisfy one of the following criteria: (1) the article contains a *full video* V^f demonstrating the whole process of completing the target task, or; (2) each step in the article is associated with an image or a video clip C_i that demonstrates the process of completing a particular action. These 6,652 articles cover diverse human everyday tasks from 19 domains (e.g., *food* and *finance*). Appendix shows the detailed statistics for the 19 domains.

Task Instance Construction

We further process the multimedia articles and construct instances for each of the two tasks we proposed. Each article may involve one or multiple methods that share the same task goal but have different sequences of steps. When an article contains multiple methods and a full video V^f, V^f likely covers all the methods. In this case, we further process V^f to extract a video demonstration for each method. Specifically, we leverage the *transition frames*[3] in the video where each *transition frame* indicates a transition occurs from one method to another. Specifically, for each frame in V^f, we check if its pixel values in the content-fixed region match those of the exemplar *transition frame*. If so, we label it as a *transition frame* and record its serial number as its position in V^f. We then segment the video V^f into N segmented videos based on the transition frames. If N is the same as the number of methods in the article, the segmented videos are sequentially associated with the methods. Otherwise, we deem the videos unavailable for these methods. After the processing, for each method M, we extract the following information: (1) title of the article which is used as the target task goal T; (2) n steps of text descriptions $\mathcal{S} = [S_1, ..., S_n]$ for the method M; (3) optional images or video clips for the n steps $[C_1, ..., C_n]$; and (4) an optional video demonstration for the method V^M. We then construct the instances for the two tasks as follows.

[1]The codes, model checkpoints, and datasets are publicly available at https://github.com/VT-NLP/MultiScript.

[2]https://www.wikihow.com/Main-Page

[3]See Appendix for an example of transition frame.

	Train	Dev	Test
Multimodal Script Generation			
# Articles	3,800	222	731
# Data Instance	4,955	294	947
Avg./Max. # Step per Instance	7.3/54	7.7/43	7.5/32
Avg./Max. Video Duration (s)	64/437	69/459	68/396
Subsequent Step Prediction			
# Articles	2,407	154	432
# Data Instance	11,409	862	2,156
Avg./Max. # Preceding Steps	3.62/35	3.80/19	3.86/26
Avg./Max. Video Duration (s)	33/977	34/439	36/568

Table 1: Statistics of the two tasks in MultiScript.

Multimodal Script Generation For each method M, we create an instance that takes a task goal T and a demonstration video V as input, and a sequence of step descriptions $\mathcal{S}$ as the target output. If each step $S_i \in \mathcal{S}$ is associated with a video clip, we concatenate the video clips for all the steps and use it as the demonstration video V for M. If there is no video clip for any of the steps but a video demonstration V^M is obtained from the article, we directly use V^M as V. Otherwise, we will ignore this method. We finally obtained 6,169 instances for this task.

Subsequent Step Prediction In this task, each instance contains a task goal T and a partial video V_k that demonstrates the k proceeding steps as input, and a subsequent step description S_{k+1} as target output. More specifically, for each method M with n steps of text descriptions $\mathcal{S} = [S_1, ..., S_n]$ and video clips $[C_1, ..., C_n]$, we take each step S_{k+1} $(0 \leq k \leq n-1)$ as the target output to create an instance where the input video V_k is concatenated from $[C_1, ..., C_k]$. If any video clip in $[C_1, ..., C_k]$ is unavailable, we extract V_k from the video demonstration V^M for method M by determining the end frame of step S_k in the video. The end frame is sourced from the last frame of video clip C_k or the beginning frame of C_{k+1}. If neither of these two video clips is available, we will ignore this step. We iterate each frame in V^M and seek matches for the end frames by comparing pixel values. If no match is found after the iteration, we cannot extract V_k from V^M and will ignore this step as a target step. We finally collected 14,427 instances for this task.

Train / Dev / Test Split We split the instances created for each task into training, development, and test sets. For each task, to ensure the coverage of various domains in each set, we randomly sample 80%, 5%, and 15% articles from each domain, and use the instances created from them as the training, development, and test sets. We name this benchmark challenge with two multimodal script learning tasks as MULTISCRIPT. Table 1 shows the statistics of the two tasks.

Method

Overveiew

As shown in Figure 2, we design two approaches for the two tasks introduced in MULTISCRIPT. For multimodal script generation, we propose an approach consisting of three main components: (1) a *Step Extractor* to extract a sequence of key frames from the input video and generate their captions to describe the sequence of key actions; (2) an *External Knowledge Prompter* that induces task-specific instructional knowledge from large language models (LLMs) such as Vicuna; and (3) a *Generator* to produce the final sequence of step descriptions by considering the input video, key frame captions as well as task-specific instructional knowledge. For subsequent step prediction, we first leverage the model traind on the previous multimodal script generation task to produce a script describing the preceding steps shown in the input partial video. Then, introduce a *Selective Generator* to dynamically select external knowledge prompted from LLMs and incorporate them together with the preceding step descriptions to generate a subsequent step. In the following, we discuss the details of these two approaches.

Multimodal Script Generation

Step Extractor Given a task goal T and a demonstration video V, the step extractor aims to detect a sequence of key action frames from V and generate their text descriptions. We employ Katna (Mayank Jain 2021), an open-source tool, to extract a set of keyframes from V and order them based on their timestamp in V, such that we obtain a sequence of chronologically ordered key frames $\mathcal{K} = \{K_1, \ldots, K_m\}$, where m is the total number of keyframes and is automatically determined by Katna (Mayank Jain 2021). For each keyframe K_i, we further employ the pre-trained OFA model[4] (Wang et al. 2022a) to generate a caption, yielding a sequence of keyframe captions $\mathcal{I} = \{I_1, \ldots, I_m\}$.

External Knowledge Prompter LLMs have been shown to be able to capture generic instructional knowledge for open-domain human everyday tasks, which is valuable to inform the generator to produce a sequence of reasonable and logically coherent step descriptions from the input video. Considering this, we leverage an LLM, such as Vicuna (Chiang et al. 2023), to produce task-specific instructional knowledge. Given the task goal T, we define a template prompt that instructs the LLM to generate a sequence of instructional steps $\mathcal{P} = \{P_1, ..., P_{n'}\}$, where n' is the number of steps in the generated sequence. The prompt includes an instruction of expert identity (Xu et al. 2023a), such as "*Imagine you are an expert on daily life tasks*" at the beginning, followed by two in-context examples where each example is a pair of question and answer to demonstrate the expected output format. Following the prompt, we append a question based on the task goal T to acquire a sequence of steps as the suggested procedures to complete the target task. Appendix presents the template prompt we defined for the knowledge prompter. Note that the instructions prompted by the LLM are typically not precisely aligned with the input demonstration video, but they can provide a general workflow to complete the target task.

Generator Given the input task goal T, video keyframe captions $\mathcal{I} = \{I_1, \ldots, I_m\}$, instructional knowledge from

[4]We employ the checkpoint "OFA-Sys/ofa-base" in this work.

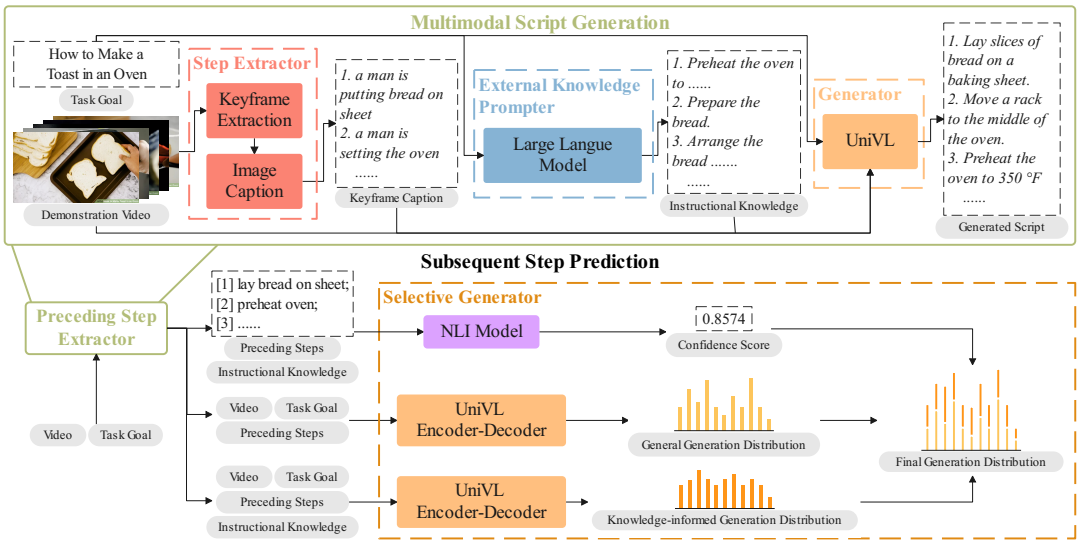

Figure 2: The overall framework to tackle our proposed two tasks. Three main components (1) a step extractor to determine and extract steps demonstrated in video, (2) a external knowledge prompter to acquire task-related knowledge, and (3) a generator.

LLMs $\mathcal{P} = \{P_1, ..., P_{n'}\}$, as well as the demonstration video V, the generator aims to produce a script, $\mathcal{S} = \{S_1, ..., S_n\}$, consisting of a sequence of action descriptions that can instruct humans to successfully complete the target task. Here, we employ UniVL (Luo et al. 2020), a pre-trained multimodal encoder-decoder model, as the generator. Specifically, we first concatenate all the text inputs $T, \mathcal{I}, \mathcal{P}$ with a split token `[SEP]` and encode the concatenated sequence using the text encoder of UniVL. The video V is first encoded as spatial and temporal representation feature (Xie et al. 2018) by a separable 3D CNN (S3D) model (Miech et al. 2019a). The S3D features for the video V are then encoded by the video encoder of UniVL. A cross encoder further takes in the features from the text encoder and video encoder and captures the interactions between the two modalities. Finally, we employ a Transformer-based decoder to produce a sequence of step descriptions $\mathcal{S}$.

Subsequent Step Prediction

Preceding Step Extractor Unlike prior studies on task planning (Wang et al. 2022b) which heavily rely on well-defined preceding steps, one significant challenge of our subsequent step prediction task lies in automatically inferring the preceding steps solely based on the input video. This objective is similar to the multimodal script generation task that produces a sequence of step descriptions from the input video. Considering this, we directly employ a model trained on the multimodal script generation task to generate a sequence of preceding steps $\mathcal{S}_k = [S_1, ..., S_k]$ given the task goal T and a partial video V_k.

Adaptive Knowledge-informed Generator We further generate a subsequent step by considering the input task goal T, preceding steps $\mathcal{S}_k = [S_1, ..., S_k]$, the partial video V_k, as well as the instructional knowledge $\mathcal{P} = \{P_1, ..., P_{n'}\}$ prompted from the LLM during the multimodal script generation. The prompted knowledge can provide the instructions related to the generator. However, it may not be well aligned with the preceding steps demonstrated in the input video since the prompted knowledge from LLMs can be too generic. To adaptively incorporate the prompted knowledge, we propose to generate the subsequent step by producing two separate probability distributions at each decoding step, where one distribution considers the instruction knowledge as input and the other does not. Then, we fuse these two distributions with a dynamic confidence score. Specifically, we employ two separate multimodal encoders based on UniVL: one encodes the concatenation of task goal T and preceding steps $\mathcal{S}_k = [S_1, ..., S_k]$ with a split token `[SEP]` as well as the spatial and temporal representation of the partial video V_k, and the other further considers the instructional knowledge $\mathcal{P} = \{P_1, ..., P_{n'}\}$ as part of the textual input. At each decoding step, based on the encoded features of the two separate UniVL encoders, we produce two separate probability distributions: D_g and D_k.

To combine the two probability distributions at each decoding step, we need to characterize how likely the prompted knowledge is aligned and beneficial to the subsequent step generation. Here, we design an approach based on natural language inference (NLI) to first predict a confidence score for each individual step included in the prompted knowledge and then output an overall score as the weight to combine the two distributions. The confidence score for each step indicates how likely the step follows the preceding steps demonstrated in the input video. Specifically, we take the concatenation of the task goal T and preceding steps $\mathcal{S}_k$ as a premise, each step in the instructional knowledge P_i as a hypothesis, and use a pre-trained Deberta (He et al. 2020) [5] as the backbone NLI model to predict a probability for the

[5] In this work, we employ the Deberta model with checkpoint "nli-deberta-v3-base".

Entailment label, which is then used as the confidence score. Note that to fine-tune the NLI model on our own dataset, we use the gold subsequence step as the positive hypothesis and sample a set of negative hypotheses from both preceding and future steps. Here, the "future steps" signify the steps that follow, but aren't immediately after, the preceding step. Finally, we use the highest confidence score c among all the steps as the weight to combine the two distributions:

$$D_f = D_g + c * D_k$$

Based on the combined distributions D_f at each decoding step, a Transformer-based decoder further predicts the output token and finally generates a subsequent step S_{k+1}.

Experiment Setup

Evaluation Metrics We evaluate our methods and baselines with eight standard metrics: BLEU (Papineni et al. 2002) including BLEU-1, BLEU-2, BLEU-3 and BLEU-4, METEOR (Denkowski and Lavie 2014), ROUGE-L (Lin 2004), BERTScore (Zhang et al. 2019), and STS (Thakur et al. 2021). Considering the variability in describing steps or task scripts, semantic similarity-based metrics hold greater significance in the proposed tasks.

Baselines For multimodal script generation, we first employ the video keyframe captions as the output script, serving as the most naive baseline (**Keyfram Caption**). Additionally, we assess the generic instructions prompted from the large language model, Vicuna-13B-1.1 (Chiang et al. 2023), as another baseline, to gauge the quality of the instructional knowledge (**Vicuna**). For subsequent step prediction, we compare our approach with two text-only based generation models, T5-Large with 7.7M parameters (**T5**) and Vicuna-13B-1.1 (Chiang et al. 2023) (**Vicuna**), which take in the task goal T and the preceding steps $\mathcal{S}_k$ and generates a subsequent step S_{k+1}. Both T5 and Vicuna are fine-tuned on the same dataset we created for subsequent step prediction.We also compare our approach with several ablated variants, including: **(1) UniVL**: directly takes in the task goal and input video, and generates a script or a subsequent step; **(2) UniVL+Step**: Besides the task goal T and input video, it also takes in the steps demonstrated in the video. Specifically, for multimodal script generation, we use the video keyframe captions $\mathcal{I}$ as the step description, while for subsequent step prediction, we use the preceding steps $\mathcal{S}_k$ generated by a well-trained model for multimodal script generation; **(3) UniVL+Knowledge(Know.)**: Besides the task goal T and input video, it also considers the instructional knowledge prompted from large language models for both tasks; **(4) UniVL+Step+Knowledge(Know.)**: complete variants of our proposed approaches for both tasks that consider the task goal T, input video, step descriptions generated from the input video as well as the instructional knowledge from the large language models.

Human Evaluation In order to comprehensively understand the challenges of the proposed tasks and assess the disparity between machine and human performance, we also conduct a human evaluation for both tasks. We hired five graduate students with decent NLP backgrounds as annotators to independently complete the multimodal script generation and step prediction tasks on 100 randomly sampled instances. For the multimodal script generation task, annotators are provided the task goal T and a demonstration video V, and were asked to summarize the script from the video. For the step prediction task, annotators were given the task goal T and a partial demonstration video V_k, and were asked to predict the immediate subsequent step.

Results and Discussions

Quantitative Comparison

Table 2 shows the performance of both baselines and our approaches for two proposed tasks. In most cases, our proposed approaches outperform all baselines and their variants with significant margins.

Multimodal Script Generation From the results of the **Keyframe Caption** baseline in Table 2, it is evident that for multimodal script generation, relying solely on video input and merely captioning the keyframes of the demonstration video does not accurately summarize the video content. Fine-grained information is missing when extracting the keyframes, and details of the keyframe images are compromised when converting them into text captions. In addition, scripts generated by LLMs based solely on task goals also lack accuracy, as shown by the performance of **Vicuna**. Due to the diverse solutions available for a human everyday task, solely relying on the task goal, Vicuna can only produce generic scripts that may not align with the methods demonstrated in the video, leading to scripts deviating significantly from the ground truth.

The results from the middle group of Table 2 present the performance of the variants of our approach. Our models that incorporate external knowledge (**UniVL+Knowledge (Know.)** and **UniVL+Script+Knowledge (Know.)**) provide average absolute gains of 0.83% and 4.67%, respectively, over the standard **UniVL** model, which demonstrates the benefit of the task-specific instructional knowledge prompted from LLMs. An interesting finding is that, the model that integrates only keyframe captions as the supplementary input (**UniVL+Step**) also exhibits remarkable improvements, surpassing even the knowledge-enhanced variant **UniVL+Knowledge (Know.)**. The possible reasons are (1) the key frames extracted from the video usually highlight the crucial steps to complete the target task, and (2) the instructions prompted by LLMs are not aligned with the steps demonstrated in the input video. It might elucidate why the improvements benefiting from the step extraction component do not mirror its significant performance in the multimodal script generation task.

Subsequent Step Prediction From Table 2, it is evident that the approaches based on single modality(**T5** and **Vicuna**) underperform all multimodal variants of our approach for subsequent step prediction, e.g., the baseline **UniVL** outperforms **T5** and **Vicuna** by 21.9% and 16.81%, respectively. A potential reason is that the preceding steps produced by the multimodal script generation approach are not

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	BERTScore	SBert	Ave
Multimodal Script Generation									
Keyframe Caption	21.03	4.93	0.95	0.17	15.58	19.82	83.84	45.05	23.92
Vicuna	28.38	20.63	14.74	5.78	28.97	24.56	86.76	60.61	33.80
UniVL	34.60	26.50	20.11	10.39	31.75	35.30	88.32	63.39	38.8
UniVL+Step	42.74	32.61	24.57	11.71	38.01	**38.39**	88.44	61.45	42.24
UniVL+Know.	36.33	27.68	20.93	11.00	33.32	34.95	88.32	64.48	39.63
UniVL+Step+Know.	**44.35**	**33.84**	**25.63**	**13.24**	**38.54**	38.04	**88.98**	**65.14**	**43.47**
Human Performance	22.20	7.86	2.94	0.51	20.92	28.21	88.33	67.89	29.78
Subsequent Step Prediction									
T5	7.63	0.38	0.00	0.00	5.42	11.81	80.85	14.74	15.10
Vicuna	9.32	2.94	0.38	0.1	12.06	11.94	83.49	34.78	19.38
UniVL	36.46	25.45	4.44	2.36	39.96	40.19	89.50	51.17	36.19
UniVL+Step	36.64	25.80	4.59	2.52	39.90	40.03	89.54	51.80	36.35
UniVL+Know.	37.08	26.04	4.63	2.52	39.98	**40.86**	89.7	52.04	36.60
UniVL+Step+Know.	**37.57**	**26.45**	**5.45**	**3.00**	**40.61**	40.73	**89.7**	**53.59**	**37.14**
Human Performance	8.63	2.00	0.83	0.00	7.55	11.77	85.76	37.71	19.28

Table 2: Automatic evaluation results on multimodal script generation and subsequent step prediction tasks. All metrics are reported in percentage (%). We highlight the best scores in bold and the second best with underline.

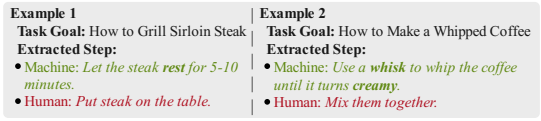

Figure 3: Example of extracted steps by human and machine in multimodal script generation task.

fully correct so the errors are accumulated in the subsequent step generation, while the multimodal variants of our approach can alleviate or correct the errors in preceding steps based on the input video. Among the variants of our approach, **UniVL+Knowledge (Know.)** outperforms both **UniVL** and **UniVL+Step** because (1) **Vicuna**, being substantially larger than **UniVL** and trained on a more extensive dataset, serves as a more robust knowledge source, enabling the generation of more precise and fine-grained steps. Although the instructional knowledge does not fully match the demonstrated method, it may share similar steps. Combined with the selective mechanism, this overlap allows the generator to integrate meaningful information while reducing the impact of misaligned instructional knowledge; (2) the extraction of the preceding steps, as mentioned earlier, is not fully correct.

Human Evaluation Table 2 reveals that, for multimodal script generation, machines outperform humans in terms of word accuracy and professionalism. The machine employs precise technical terminology for action descriptions and offers more standardized and accurate portrayals of entities and their states, e.g., *"rest", "whisk", "creamy"* in Figure 3, which is a challenge for humans without expertise. However, humans significantly outperform machines based on semantic-based evaluation. One possible reason is that humans might not recognize specific tool or entity names from a demonstration video, but it is easy for them to clearly elucidate the workflow to complete the target task. For subsequent step prediction tasks, humans consistently underperform machines across all metrics. This can be attributed to the essentiality of domain-specific knowledge in this task. In multimodal script generation, humans can describe the content demonstrated in the video and effectively achieve high semantic alignment with the golden script. However, in the subsequent step prediction task, human annotators without domain knowledge struggle to grasp the workflow and steps involved in an everyday task. Consequently, their predictions frequently diverge from the correct subsequent steps.

Qualitative Analysis

Table 3 and Table 4 present the qualitative results from various model variants in multimodal script generation and subsequent step prediction tasks, showing approaches that incorporate external instructional knowledge and video keyframe captions substantially enhance the quality of generation and prediction. Additional instructional knowledge provides finer-grained commonsense details that are hard to discern from demonstration videos. For instance, entities with comparable appearances, e.g., "*salt*", "*sugar*" in Table 3, are not easily distinguished from the video. With clear clarification and definitions from the instructional knowledge, our approach is able to extract more accurate and fine-grained steps. For the subsequent step prediction task, a well-defined workflow outline aids the model in discerning the current task stage. As demonstrated in Table 4, our approach accurately predicts the"*cut*" step, reflecting a deeper understanding of the task flow compared with UniVL.

Keyframe	Additional Information	Generated Script	Ground Truth
Task Goal: How to Make Porridge			
	**Instructional Knowledge:** "Add salt to the boiling water and turn down the heat to low.", "Stir in the sugar, cinnamon, nutmeg, vanilla, and sweetened condensed milk"	**UniVL:** [1] choose a pot with a lid [2] pour in the water [3] bring the water to a boil [4] simmer the porridge for 3-5 minutes [5] remove the pot from the heat and serve; **UniVL+Step+Knowledge:** [1] boil the water and salt in a large saucepan [2] add the oats and reduce the heat [3] stir in the porridge for 5 minutes [4] stir in the nuts and sugar [5] serve	[1] place the oats and water in a large pan over medium heat [2] add a pinch of salt and stir [3] bring to a steady simmer for 5 or so minutes [4] add in a dash of brown sugar and honey , if you ' d like [5] flavor with toppings and serve
Task Goal: How to Cook New York Strip Steak			
	**Keyframe Caption:** "a close up of a person holding a frying pan", "a piece of meat being drizzled with oil and seasoning", "a person pick up a piece of steak"	**UniVL:** [1] choose steaks [2] sear the steaks on both sides [3] sear the steaks [4] sear the steaks; **UniVL+Cap+History:** [1] heat the pan [2] heat the oil in a skillet over medium-high heat [3] add the steaks to the pan [4] cook the steaks for about 3 to 4 minutes [5] use a spatula to move the steaks in the pan	[1] heat a pan on the stove [2] season and oil your steaks [3] add your steaks to the pan [4] flip after about three or four minutes [5] allow to cook through and serve

Table 3: Qualitative result of different model variants in multimodal script generation task.

Keyframe	Preceding Step	Additional Information	Prediction	Ground Truth
Task Goal: How to Sew Oven Mitts				
	1. Select an insulating fabric for the inside of the oven mitts; 2. Trace your hand onto paper to make a pattern; 3. Align the paper pattern on the fabric as desired; 4.	**Instructional Knowledge:** [1] Cut out two pieces of fabric for the oven mitts, each 8 inches long by 4 inches wide [2] Cut out two pieces of fabric for the thumb, each 2 inches long by 1 inch wide [3] Sew	**UniVL:** Sew the outer edges of the mittens together; **UniVL+Step+Knowledge:** [1] Use scissors cut the fabric from the fabric.	Use a sharp pair of scissors to cut out your fabric pieces.

Table 4: Qualitative result of different model variants in subsequent step prediction task.

Related Work

Multimodal Script Learning Several multimodal approaches (Narasimhan et al. 2022; Wang et al. 2022b; Xu, Shen, and Huang 2023; Qi et al. 2023) have been proposed in recent years to tackle script learning tasks. While there is significant progress in predictive performance, several practical issues have yet to be adequately addressed. For example, in candidate-based tasks such as classification (Lin et al. 2022) and multiple-choice (Yang et al. 2021a), the acquisition of reliable candidates is often unstable due to the lack of user input and the limited number of documented tasks.

Datasets Related to Future Prediction Task Three types of datasets have been used in previous research on future prediction tasks: visual-only (Damen et al. 2022, 2018; Li, Liu, and Rehg 2018), text-only (Puig et al. 2018; Lyu, Zhang, and Callison-Burch 2021; Le et al. 2023), and multimedia datasets (Yang et al. 2021b; Tang et al. 2019; Miech et al. 2019b; Xu et al. 2023b). Our dataset distinguishes itself from these previous works in two aspects: (1) Compared with visual/text-only dataset, our dataset is a multimedia dataset, comprising video, image, and text descriptions for each instructional step. The data inclusion from multiple modalities provides unique and complementary information [6]. (2) Text description of our dataset is more fine-grained, using original instructional article step descriptions from WikiHow rather than using the transcript or summary of videos.

Conclusion

In this paper, we introduced a new benchmark challenge encompassing two task-oriented multimodal script learning tasks: multimodal script generation and subsequent step prediction. To support further research in this domain, we present a new dataset consisting of over 6,655 everyday human tasks across 19 domains. We proposed a knowledge-informed framework that adaptively integrates task-related instructional knowledge into the multimodal generative model. While our approach demonstrated encouraging results, there are still limitations and challenges that need to be addressed in future work on these tasks.

[6] See Appendix for data type comparison of existing work and MULTISCRIPT.

Acknowledgments

This research is based upon work supported by the U.S. DARPA ECOLE Program # HR001122S0052. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Damen, D.; Doughty, H.; Farinella, G. M.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; and Wray, M. 2018. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *European Conference on Computer Vision (ECCV)*.

Damen, D.; Doughty, H.; Farinella, G. M.; Furnari, A.; Ma, J.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; and Wray, M. 2022. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130: 33–55.

Denkowski, M.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. 376–380.

Girdhar, R.; and Grauman, K. 2021. Anticipative Video Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 13505–13515.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Le, D.; Guo, R.; Xu, W.; and Ritter, A. 2023. Improved Instruction Ordering in Recipe-Grounded Conversation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10086–10104. Toronto, Canada: Association for Computational Linguistics.

Li, Y.; Liu, M.; and Rehg, J. M. 2018. In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part V*, 639–655. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-01227-4.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. 74–81.

Lin, X.; Petroni, F.; Bertasius, G.; Rohrbach, M.; Chang, S.-F.; and Torresani, L. 2022. Learning To Recognize Procedural Activities With Distant Supervision. 13853–13863.

Luo, H.; Ji, L.; Shi, B.; Huang, H.; Duan, N.; Li, T.; Li, J.; Bharti, T.; and Zhou, M. 2020. UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation.

Lyu, Q.; Zhang, L.; and Callison-Burch, C. 2020. Reasoning about Goals, Steps, and Temporal Ordering with WikiHow. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 4630–4639.

Lyu, Q.; Zhang, L.; and Callison-Burch, C. 2021. Goal-oriented script construction. *arXiv preprint arXiv:2107.13189*.

Mayank Jain, A., Nitin Katyal. 2021. Katna: Tool for automating video keyframe extraction, video compression, Image Autocrop and Smart image resize tasks.

Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019a. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019b. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2630–2640.

Narasimhan, M.; Nagrani, A.; Sun, C.; Rubinstein, M.; Darrell, T.; Rohrbach, A.; and Schmid, C. 2022. TL;DW? Summarizing Instructional Videos with Task Relevance and Cross-Modal Saliency. *LNCS*, 13694: 540–557.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 311–318.

Puig, X.; Ra, K.; Boben, M.; Li, J.; Wang, T.; Fidler, S.; and Torralba, A. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8494–8502.

Qi, J.; Xu, Z.; Shen, Y.; Liu, M.; Jin, D.; Wang, Q.; and Huang, L. 2023. The Art of SOCRATIC QUESTIONING: Recursive Thinking with Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 4177–4199. Singapore: Association for Computational Linguistics.

Singh, A.; Singh, T. D.; and Bandyopadhyay, S. 2020. Nitsvc system for vatex video captioning challenge 2020. *arXiv preprint arXiv:2006.04058*.

Tang, Y.; Ding, D.; Rao, Y.; Zheng, Y.; Zhang, D.; Zhao, L.; Lu, J.; and Zhou, J. 2019. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1207–1216.

Thakur, N.; Reimers, N.; Daxenberger, J.; and Gurevych, I. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 296–310. Online: Association for Computational Linguistics.

Wang, P.; Yang, A.; Men, R.; Lin, J.; Bai, S.; Li, Z.; Ma, J.; Zhou, C.; Zhou, J.; and Yang, H. 2022a. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*.

Wang, Q.; Li, M.; Chan, H. P.; Huang, L.; Hockenmaier, J.; Chowdhary, G.; and Ji, H. 2022b. Multimedia Generative Script Learning for Task Planning.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, 305–321.

Xu, B.; Yang, A.; Lin, J.; Wang, Q.; Zhou, C.; Zhang, Y.; and Mao, Z. 2023a. ExpertPrompting: Instructing Large Language Models to be Distinguished Experts. *arXiv preprint arXiv:2305.14688*.

Xu, Z.; Ashby, T.; Feng, C.; Shao, R.; Shen, Y.; Jin, D.; Wang, Q.; and Huang, L. 2023b. Vision-Flan:Scaling Visual Instruction Tuning.

Xu, Z.; Shen, Y.; and Huang, L. 2023. MultiInstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11445–11465. Toronto, Canada: Association for Computational Linguistics.

Yang, Y.; Panagopoulou, A.; Lyu, Q.; Zhang, L.; Yatskar, M.; and Callison-Burch, C. 2021a. Visual Goal-Step Inference using wikiHow. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2167–2179.

Yang, Y.; Panagopoulou, A.; Lyu, Q.; Zhang, L.; Yatskar, M.; and Callison-Burch, C. 2021b. Visual Goal-Step Inference using wikiHow. *arXiv preprint arXiv:2104.05845*.

Zhang, L.; Lyu, Q.; and Callison-Burch, C. 2020. Reasoning about Goals, Steps, and Temporal Ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4630–4639.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. BERTScore: Evaluating Text Generation with BERT.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685.

Zhong, Z.; Schneider, D.; Voit, M.; Stiefelhagen, R.; and Beyerer, J. 2022. Anticipative Feature Fusion Transformer for Multi-Modal Action Anticipation. *arXiv preprint arXiv:2210.12649*.