Multimedia Generative Script Learning for Task Planning

Qingyun Wang[1], **Manling Li**[1], **Hou Pong Chan**[2], **Lifu Huang**[3],
Julia Hockenmaier[1], **Girish Chowdhary**[1], **Heng Ji**[1]
[1] University of Illinois at Urbana-Champaign [2] University of Macau [3] Virginia Tech
[1]{qingyun4,manling2,juliahmr,girishc,hengji}@illinois.edu
[2]hpchan@um.edu.mo,[3]lifuh@vt.edu

Abstract

Goal-oriented generative script learning aims to generate subsequent steps to reach a particular goal, which is an essential task to assist robots or humans in performing stereotypical activities. An important aspect of this process is the ability to capture historical states visually, which provides detailed information that is not covered by text and will guide subsequent steps. Therefore, we propose a new task, Multimedia Generative Script Learning, to generate subsequent steps by tracking historical states in both text and vision modalities, as well as presenting the first benchmark containing 5,652 tasks and 79,089 multimedia steps. This task is challenging in three aspects: the multimedia challenge of capturing the visual states in images, the induction challenge of performing unseen tasks, and the diversity challenge of covering different information in individual steps. We propose to encode visual state changes through a selective multimedia encoder to address the multimedia challenge, transfer knowledge from previously observed tasks using a retrieval-augmented decoder to overcome the induction challenge, and further present distinct information at each step by optimizing a diversity-oriented contrastive learning objective. We define metrics to evaluate both generation and inductive quality. Experiment results demonstrate that our approach significantly outperforms strong baselines[1].

1 Introduction

Robots rely on understanding the present real-world state and predicting the subsequent steps to better assist humans in daily stereotypical tasks such as meal preparation and gardening (Ruth Anita Shirley et al., 2021; Liu et al., 2022). As an example, Robohow (Beetz et al., 2016) uses articles

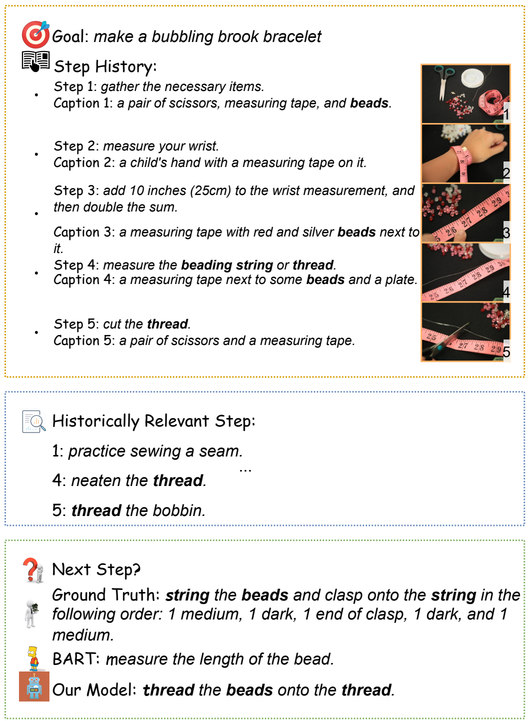


Figure 1: **Multimedia Generative Script Learning:** The upper box shows the task input, including the goal and multimedia step history. Each step contains a text description and an illustrative image. The output is the next step. We retrieve historically relevant steps from the training corpus.

from WikiHow[2] to assist robots in everyday tasks in human working and living environments. However, the problem is that not all daily tasks are well documented. Thus, generating a sequence of steps that lead to a given goal (i.e., goal-oriented generative script learning) (Lyu et al., 2021; Huang et al., 2022; Li et al., 2023; Zhou et al., 2023; Liu et al., 2023) has a fundamental importance in allowing robots to perform unseen tasks by understanding the patterns in previously observed similar tasks.

Despite this, previous goal-oriented generative

[1]The programs, data, and resources are publicly available for research purposes at: https://github.com/EagleW/Multimedia-Generative-Script-Learning.

[2]https://www.wikihow.com contains steps for a variety of tasks.

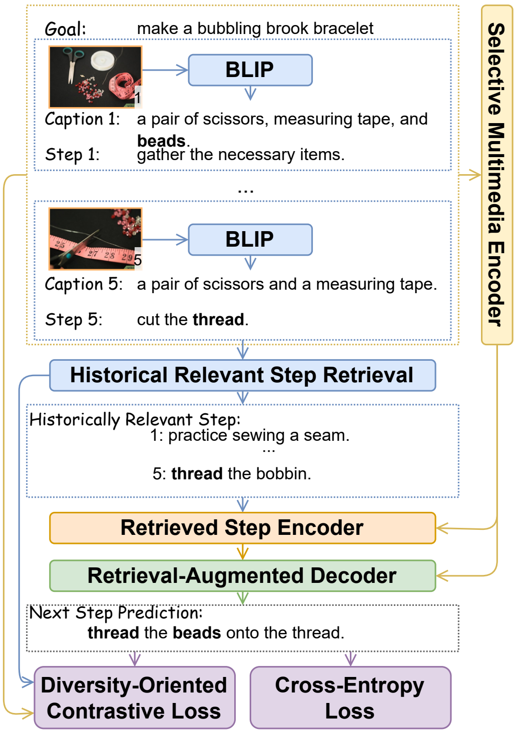


Figure 2: Architecture overview. We use the example in Figure 1 as the walking-through example.

script learning focuses solely on text (Lyu et al., 2021; Huang et al., 2022), which is commonly affected by reporting bias (Gordon and Van Durme, 2013) as important details may be omitted in the source text. However, such information is often implicitly contained in images. For example, in Figure 1, the image of Step 1 illustrates the items needed to *make a bracelet*, which is not mentioned in the text but helps predict the action of *threading beads* as a future step. Existing multimedia script learning work seeks to bridge this cross-media gap, but the task settings are multi-choice selection (Yang et al., 2021b) or ordering (Wu et al., 2022), which require candidate steps as input so it is not a practical setting for real-life robots.

To address these problems, we propose a new task, **Multimedia Generative Script Learning** (Figure 1), that requires systems to generate future steps based on the goal and previous steps with visual scenes depicting their states. Specifically, given the goal and previous step history in the form of natural language sentences paired with descriptive images, the model should automatically generate the natural language instruction for the next step. A good script has three hallmarks:

(1) <u>*Visual-State Trackable*</u>: it records the historical visual scenes and recognizes significant changes that impact future steps. We call it *multimedia challenge*. To address this challenge, we focus on salient differences in visual scenes, and propose a novel **selective multimedia encoder**. Rather than learning directly from the visual details of each object, we first leverage an image captioner as an abstract summary of the image about global interactions among multiple objects. We then introduce a selection gate to focus on the selected captions and steps closely related to the future step. For instance, the second caption *"a child's hand with a measuring tape on it"* in Figure 1 can be filtered out by the selection gate because it is not closely related to the future steps.

(2) <u>*Inductive*</u>: it transfers knowledge from a previously observed task to similar unseen tasks. We call it *induction challenge*. To induce procedural knowledge from previously observed tasks, we propose a **retrieval augmented decoder** to obtain relevant steps to guide the subsequent step generation. For example, the future step in Figure 1 closely resembles the scripts used in previous retrieved steps about *threading items*, thus transferring script knowledge to an unseen task.

(3) <u>*Diverse*</u>: it displays distinct information at each step. We call it *diversity challenge*. Existing pre-trained transformer-based language models such as T5 (Raffel et al., 2020), BART (Lewis et al., 2020a), and GPT-2 (Radford et al., 2019) tend to generate repeated or highly similar future steps as shown in Figure 1. Therefore, we introduce a novel **diversity-oriented contrastive learning objective** to control all subsequent steps to convey different information. We treat all other steps in the given input and retrieved steps in other tasks similar to the given input as *hard* negatives.

In addition to traditional generation-based metrics to evaluate task performance, we propose a new *multimodal-retrieval based metric* to capture cross-modal semantic similarity. While the model design can be applied to any domain of interest, we experiment with the model on two domains *Gardening* and *Crafts*, where task planning has not been well researched. Automatic evaluation shows that our generated step predictions are close to the human written ground truth. Human evaluation further confirms that our diversity-oriented contrastive learning objective leads to diverse and correct steps.

The contributions are threefold:

1. We propose the first *multimedia goal-oriented*

generative script learning task to record historical steps in both text and images. We also release a new benchmark from WikiHow, featuring 5,652 tasks and 79,089 multimedia steps.

2. We propose a novel approach to produce *visually trackable*, *inductive*, and *diverse* scripts through a selective multimedia encoder, a retrieval augmented decoder, and a diversity-oriented contrastive learning objective.

3. We propose a new *multimodal-retrieval based metric* to evaluate the cross-modal semantic similarity and the inductive ability by checking factual correctness.

2 Problem Formulation

We propose a new multimedia generative script learning task: given an activity goal G, an optional subgoal M that specifies the concrete needs, and the previous multimedia step history $\mathcal{H}_n = \{(S_1, V_1), ..., (S_n, V_n)\}$ with length n, a model is expected to predict the next possible step S_{n+1}, where S_i is a text sequence and V_i is an image.

Domain	Split	#Task	#Pair	$\overline{\#\textbf{Step}}$	$\overline{\#\textbf{Token}}$
Gardening	Train	1,857	20,258	3.10	11.6
	Valid.	237	2,428	3.03	10.6
	Test	238	2,684	2.88	11.2
Crafts	Train	2,654	32,082	6.06	8.98
	Valid.	3,33	4,061	6.12	9.10
	Test	3,33	3,937	5.91	9.00

Table 1: Statistics of our dataset. $\overline{\#\textbf{Step}}$ denotes average number of steps per sample. $\overline{\#\textbf{Token}}$ denotes average number of words per step.

3 Dataset Collection

Using articles from *Gardening* and *Crafts* categories as case studies, we create a new dataset based on the English WikiHow dump (2021/05). There are typically three levels of hierarchy in a WikiHow article: *goals* which describe the overall task, *subgoals* which represent the intermediate process to accomplish a *goal*, and *steps* which are the specific actions to complete a *subgoal*. For each WikiHow article, we collect step-image pairs as well as their goals and methods[3]. We split the whole dataset based on the task categories. Therefore, the validation and test sets contain tasks not included in the training set. Table 1 shows the detailed data statistics.

[3] We only keep steps that contain both images and texts.

4 Method

4.1 Model Architecture

The overall framework is illustrated in Figure 2. Given the activity goal G, optional subgoal M, and multimedia step history $\mathcal{H}_n$, we first use an image captioner to map each input image into a precise caption and produce the caption-enhanced step history $\hat{\mathcal{H}}_n$. Then we propose a *selective multimedia encoder* by extending the BART encoder with a gated fusion layer to learn contextualized representations for the step history. After that, a retrieval module retrieves historically relevant steps from the training corpus and encodes them with a *retrieved step encoder*. Finally, we introduce a *retrieval-augmented decoder*, which enhances the BART decoder with a retrieval gate fusion layer to fuse the representations of the input step history and retrieved steps to generate the next step. The entire model is trained by our proposed *diversity-oriented contrastive loss* and cross-entropy loss.

4.2 Selective Multimedia Encoder

Image Encoding Compared to step descriptions which focus more on action description, captions provide more visual environment/object information such as *beads* in Step 1 from Figure 2. Because we are more concerned with the overall semantics of the salient objects in the image rather than the details of every object, we adopt image captioners to encode visual features and track visual state changes. For instance, while multiple objects are present in Step 3 in Figure 1, the *finger* object can be ignored in the third step as it does not represent the key information conveyed by the image. Specifically, we use the state-of-the-art image captioner BLIP (Li et al., 2022), which is pretrained on a large-scale vision-and-language corpus with 129M images to generate a caption C_i for each image V_i in the input step history $\mathcal{H}_n$. After that, we obtain the *caption-enhanced step history* $\hat{\mathcal{H}}_n = \{(S_1, C_1), ..., (S_n, C_n)\}$, where C_i is the caption of the image V_i in step i.

Selective Multimedia Encoding To help the encoder capture the activity goal and subgoal information, we concatenate goal G and optional subgoal M to serve as the first sequence in the history $X_0 = [G, M]$. For the subsequent steps in the history, we concatenate each step and caption as $X_{2i-1} = S_i$

and $X_{2i} = C_i$. To summarize the step history, we prepend a learnable [CLS] token to the sequence as a contextualized vector. The entire text sequence is then represented as $\mathcal{X} = \{[\texttt{CLS}], X_0, X_1, ..., X_{2n}\}$. We pass the text sequence $\mathcal{X}$ into a BART encoder to get the contextualized hidden representation $\mathbf{H} = \{\mathbf{h}_0, ..., \mathbf{h}^{2n}_{L_{X_{2n}}}\} = \mathrm{Enc}(\mathcal{X})$. We denote $\mathbf{H}_{X_j} = \{\mathbf{h}^j_1, ..., \mathbf{h}^j_{L_{X_j}}\}$ as the hidden states for sequence X_j, where L_{X_j} is the length of X_j.

Since the input sequence contains steps or captions not directly relevant to the future step, we need to mask those sentences based on the step/caption representations. For instance, in Figure 2, the step description for Step 1 is vague and needs to be masked. We treat the representation of the [CLS] token, $\mathbf{h}_0$, as the contextualized representation of the entire step history and use it to compute a mask that filters out the irrelevant step/caption information. Specifically, we use $\mathbf{h}_0$ as query and $\mathbf{H}_{X_j}$ as both the key and value to compute Multi-Headed Attention (MultiHead) (Vaswani et al., 2017) for each sequence hidden states $\mathbf{H}_{X_j}$: $\hat{\mathbf{h}}_{X_j} = \mathrm{MultiHead}(\mathbf{h}_0, \mathbf{H}_{X_j}, \mathbf{H}_{X_j})$, where $\hat{\mathbf{h}}_{X_j}$ is the weighted representation for text sequence X_j. Then, for each sequence X_j, we can calculate the mask probability as:$\alpha_j = \sigma(\mathbf{W}_\alpha[\mathbf{h_0}; \hat{\mathbf{h}}_{X_j}])$, where $\mathbf{W}_\alpha$ is a learnable parameter. Similar to Sengupta et al. (2021), we update the hidden states for each sequence X_j as $\bar{\mathbf{H}}_{X_j} = \alpha_j \cdot \mathbf{emb}_{[\texttt{MASK}]} + (1 - \alpha_j)\mathbf{H}_{X_j}$, where $\mathbf{emb}_{[\texttt{MASK}]}$ is the embedding of the [MASK] token. The final hidden state sequences are $\bar{\mathbf{H}} = [h_0; \bar{\mathbf{H}}_1; ...; \bar{\mathbf{H}}_{2n}]$.

4.3 Step Retrieval Augmentation

Historically Relevant Step Retrieval In addition to the caption-enhanced step history, $\hat{\mathcal{H}}_n$, we retrieve historically relevant steps $\mathcal{R}_{n+1} = \{R_1, ..., R_k\}$ from the training tasks, where k is the number of retrieved relevant steps. We first use SentenceBERT (Reimers and Gurevych, 2019) to encode all steps. We then retrieve k steps from the training corpus, which have the top-k highest cosine similarity to the previous step S_n from the representation given by SentenceBERT[4]. Finally, we consider the immediate next step for each of those k steps as potential relevant steps $\mathcal{R}_{n+1}$. For instance, because Step 5 in Figure 2 is similar to *pull the thread out* in the training corpus, we choose its immediate next step *thread the bobbin* as a historically relevant step.

Retrieved Step Encoder For historically relevant steps $\mathcal{R} = \{R_1, ..., R_k\}$, we apply the BART encoder to get hidden states $\mathbf{H}_R = \{\mathbf{H}_{R_1}; ....; \mathbf{H}_{R_k}\}$. Similarly, we use $\mathbf{h}_0$ in multimedia encoder as the query and $\mathbf{H}_{R_i}$ as both the key and value to compute multi-headed attention for each sequence hidden states: $\hat{\mathbf{h}}_{R_i} = \mathrm{MultiHead}(\mathbf{h}_0, \mathbf{H}_{R_i}, \mathbf{H}_{R_i})$, where $\hat{\mathbf{h}}_{R_i}$ is the weighted representation for step sequence R_i. Similarly, we can calculate the mask probability as: $\beta_j = \sigma(\mathbf{W}_\beta[\mathbf{h}_0; \hat{\mathbf{h}}_{R_j}])$, where $\mathbf{W}_\beta$ is a learnable parameter. We then update the hidden states for each sequence R_j as $\bar{\mathbf{H}}_{R_i} = \beta_j \cdot \mathbf{emb}_{[\texttt{MASK}]} + (1 - \beta_j)\mathbf{H}_{R_i}$. The final hidden state sequences is $\bar{\mathbf{H}}_R = [\bar{\mathbf{H}}_{R_1}; ...; \bar{\mathbf{H}}_{R_k}]$.

4.4 Retrieval-Augmented Decoder

In the decoder, we compute the probability $P\left(s_q|s_{<q}, \hat{\mathcal{H}}, G, M\right)$ for the q-th token $s_q \in S_{n+1}$. Our retrieval-augmented decoder is similar to (Liu et al., 2021), which aims to capture historically relevant steps related to the next step based on previous decoder hidden states. Given z^l_q which is the hidden state of s_q in layer l, we first use a multi-head cross-attention to fuse the hidden states from the retrieved steps $\bar{\mathbf{H}}_\mathbf{R}$: ${z'}^l_q = \mathrm{MultiHead}(z^l_q, \bar{\mathbf{H}}_R, \bar{\mathbf{H}}_R)$. We also append a gating mechanism to control the knowledge from the retrieved steps and previous hidden states:

$$\begin{aligned} \gamma &= \sigma(\mathbf{W}_\gamma[z^l_q; {z'}^l_q]) \\ \tilde{z}^l_q &= \gamma \cdot \mathrm{LN}({z'}^l_q) + (1 - \gamma) \cdot (z^l_q) \end{aligned} \tag{1}$$

where $\mathbf{W}_\gamma$ is a learnable parameter and $\mathrm{LN}(*)$ is the layer norm function. Finally, the fused hidden states in the top layer are used to compute the generation probability. We supervise the next step generation using the standard cross-entropy loss:

$$\mathcal{L}_{\text{gen}} = \sum_{q=1}^{|S_{n+1}|} \log P\left(s_q|s_{<q}, \hat{\mathcal{H}}, G, M\right) \tag{2}$$

4.5 Diversity-Oriented Contrastive Learning

In the experiment, we observe that the model tends to keep generating similar future steps in a row given the beginning steps as input or just paraphrases the input steps. Therefore, we propose a contrastive learning-based loss to encourage the model to return diverse step prediction results.

Negative Sampling Sequence-to-sequence models suffer from the "exposure bias" problem (Dhingra

[4]We use the previous step S_n instead of all history since it is more temporally correlated to the next step.

et al., 2016; An et al., 2022) because of *teacher forcing*. Contrastive loss provides an additional sequence level loss which can help models increase the diversity of the output steps. We adopt two types of negative sampling strategies to discourage the model from paraphrasing the previous step as the future step: *self-negatives* (Wang et al., 2022) where we consider the input steps as negative samples and *retrieved-negatives* where we consider the retrieved steps from training corpus which are similar to the input step as negative samples. For example, in Figure 1, the goals and steps from the step history serve as the self-negatives. Given the last step, “cut the thread”, we retrieve similar steps from the training set as retrieved negatives which include “cut your thread”, "cut off the extra thread", etc.

Diversity-Oriented Contrastive Loss Since the model needs to distinguish between the ground truth and those negative samples, we design a novel diversity-oriented contrastive loss. Specifically, given an input sequence $\hat{\mathcal{H}}, G, M$, the ground truth next step S_{n+1}, and a set of K negative samples $\{S^1_{n+1}, S^2_{n+1}, ..., S^K_{n+1}\}$, we aim to maximize the probability of classifying the positive sample correctly with the InfoNCE loss (Oord et al., 2018):

$$\begin{aligned}
\mathcal{L}_{\text{cl}} &= \frac{\exp\left(y^+/\tau\right)}{\sum_k \exp\left(y_k^-/\tau\right) + \exp\left(y^+/\tau\right)} \\
y^+ &= \sigma(\text{Avg}(\mathbf{W}_y \bar{\mathbf{H}}^+ + \mathbf{b}_y)) \\
y_k^- &= \sigma(\text{Avg}(\mathbf{W}_y \bar{\mathbf{H}}_k^- + \mathbf{b}_y))
\end{aligned} \tag{3}$$

where $\bar{\mathbf{H}}^+$ and $\bar{\mathbf{H}}_k^-$ are decoder hidden states from the positive and k-th negative samples, $\mathbf{W}_y$ is a learnable parameter, τ is the temperature, and $\text{Avg}(*)$ denotes the average pooling function.

4.6 Training Objective

We jointly optimize the cross-entropy loss and our proposed diversity-oriented contrastive loss: $\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda \mathcal{L}_{\text{cl}}$, where λ is a hyperparameter that controls the weight of the contrastive loss.

5 Evaluation Metrics

Generation Quality Evaluation Following common practice in text generation, we first evaluate our model with BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Denkowski and Lavie, 2014) scores to examine the content overlap between generated steps and ground truth.

Inductive Quality Evaluation In order to determine whether the inferred subsequent steps are factually correct, we further evaluate the models with BARTScore (Yuan et al., 2021) and the semantic similarity score (Thakur et al., 2021). The semantic similarity score uses a cross-encoder pretrained on STSBenchmark (Cer et al., 2017) to calculate the semantic similarity between two sentences.

In addition to evaluating whether the generated step matches the next step, we also check whether the generated step matches any subsequent step. This enables the model to earn credit if it generates a step that appears in the future. We propose a *Multimodal-Retrieval based metric*: for each generated step, we use it as a query to search all corresponding step-image pairs under the same subgoal/goal from the testing set. We then compute HIT@1 for results that fall into ground-truth future step-image pairs. Similar to Section 4.3, we use SBERT (Reimers and Gurevych, 2019) to rank the most similar steps under the same subgoal to get Text@1 (T@1). To compute Image@1 (I@1), we use CLIP (Radford et al., 2021) to rank the most similar images under the same subgoal. If the top-1 retrieval results appear in the subsequent steps, we consider it a HIT. The retrieval-based metric captures normalized semantic similarity concerning all related steps under certain subgoals. The CLIP-based retrieval metric also enables the evaluation of the cross-modality semantic similarity. Additional details of the evaluation setup are in the Appendix C.

Model	**Gardening**		**Crafts**	
	I@1↑	**T@1↑**	**I@1↑**	**T@1↑**
BART	44.6	40.0	48.2	29.9
+CP	48.5	39.2	48.2	31.5
+CP+M	**49.8**	41.0	**50.3**	**37.8**
+CP+M+R	48.1	38.9	48.9	31.8
+CP+M+R+CL	49.5	**43.0**	49.0	33.9

Table 2: Multimodal-retrieval based evaluation (%). *CP* is models with caption input. *M* is models with selective multimedia encoder. *R* is models with historically relevant step encoder and retrieval-augment decoder. *CL* is models with diversity-oriented contrastive learning.

6 Experiments

6.1 Baselines

We first compare our model with **(1) state-of-the-art pretrained text-only generation models**

Model	B-1↑	B-2↑	B-3↑	B-4↑	METEOR↑	R-L↑	BARTScore↑	Semantic↑
GPT-2	13.2	5.03	1.87	0.72	7.38	12.5	-4.73	0.239
T5	17.6	9.05	4.92	2.87	9.41	16.5	-4.45	0.300
Naive Retrieval	10.9	4.14	1.93	1.10	6.33	10.0	-4.88	0.180
CLIP-BART	14.4	7.10	3.77	2.22	8.28	13.8	-4.44	0.256
Retrieval BART	16.8	8.68	4.80	2.24	9.15	16.0	-4.43	0.295
GPT2-SIF	11.6	5.10	2.43	1.28	6.85	10.8	-4.80	0.233
BART	17.0	8.21	4.45	2.61	8.93	15.7	-4.52	0.277
+CP	16.9	8.79	4.99	3.03	9.23	16.5	-4.41	0.300
+CP+M	17.8	9.36	5.30	3.19	9.61	**17.4**	-4.38	0.305
+CP+M+R	17.5	9.22	5.25	3.13	9.60	17.2	**-4.36**	0.309
+CP+M+R+CL	**18.4**	**9.72**	**5.51**	**3.31**	**9.91**	17.3	-4.37	**0.310**

Table 3: Results with automatic evaluation on next step prediction for the gardening domain (%). *B-n* denotes the BLEU-n score. *R-L* denotes the ROUGE-L score. *Semantic* denotes semantic similarity score.

Model	B-1↑	B-2↑	B-3↑	B-4↑	METEOR↑	R-L↑	BARTScore↑	Semantic↑
GPT-2	15.5	5.40	1.98	0.93	7.63	14.0	-4.67	0.218
T5	20.8	11.1	6.43	4.07	10.54	19.6	-4.38	0.300
Naive Retrieval	13.5	5.26	2.38	1.28	6.81	12.3	-4.83	0.163
CLIP-BART	17.9	9.13	5.21	3.40	9.37	16.4	-4.56	0.245
Retrieval BART	18.7	9.78	5.52	3.52	9.89	18.2	-4.38	0.285
GPT2-SIF	14.8	6.70	3.05	1.58	7.74	13.2	-4.69	0.234
BART	19.7	10.8	6.22	4.11	10.44	20.0	-4.29	0.299
+CP	20.1	11.1	6.48	4.24	10.61	20.1	-4.29	0.303
+CP+M	20.5	11.1	6.61	4.40	10.79	20.1	-4.28	0.305
+CP+M+R	20.7	11.5	6.93	4.66	11.02	**20.5**	**-4.25**	0.309
+CP+M+R+CL	**21.3**	**11.8**	**7.12**	**4.85**	**11.25**	20.3	-4.26	**0.313**

Table 4: Automatic evaluation results on next step prediction for the crafts domain (%).

to examine the results without tracking visual states, including GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020a). We then compare our model with the **(2) retrieval baselines** including a naive retrieval baseline which directly uses retrieved historically relevant sentences as discussed in Section 4.3, and retrieval BART which takes in the concatenation of the retrieved historically relevant sentences with the original text input. We also include **(3) multi-modal generation baselines** that can take image embedding instead of captions as input, which is equivalent to CLIP-BART (Sung et al., 2022). The CLIP-BART has a similar backbone as VL-BART (Cho et al., 2021) but instead replacing the Faster R-CNN (Ren et al., 2015) with ViT-B/32 CLIP encoder (Radford et al., 2021) which has a better image-text alignment. Additionally, we compare our model with a state-of-the-art script learning model: GPT2-SIF (Sancheti and Rudinger, 2022) finetuned on our dataset. Finally, we include the variances of our model as **(4) baselines for ablation**. We select BART over T5 as the base model due to the performance and parameter size. Due to the large number of parameters in T5 (222M) compared to BART (139M), given similar model performance in Table 3 and 4, we choose BART instead of T5. The hyperparameters, training details, and additional ablation study are presented in the Appendix A, B, and D.

Model	Gardening 1↓	2↓	3↓	4↓	Crafts 1↓	2↓	3↓	4↓
Ground Truth	37.0	3.08	0.42	0.18	30.6	1.07	0.05	0.00
BART	45.2	6.94	1.39	0.73	39.2	2.18	0.26	0.10
+CP	**43.1**	5.88	1.00	0.39	**36.0**	**1.81**	0.05	0.02
+CP+M	43.6	**5.75**	**0.78**	**0.20**	36.4	1.97	**0.02**	0.01
+CP+M+R	44.2	6.32	1.12	0.38	36.9	2.03	0.06	**0.01**
+CP+M+R+CL	43.3	6.23	1.01	0.35	36.2	1.91	0.05	0.02

Table 5: Percent (%) of n-grams in step history which appear in human or system steps.

6.2 Automatic Evaluation

As shown in Table 3 and 4, our model outperforms baselines. Since our task is open-ended and we are testing on unseen activities, our generated sentences usually contain paraphrases. Therefore, the BLEU scores, which rely on the exact word n-grams match (Goldberg, 2018), are not high. In particular, because our ground truth only has an average length of 11 which contains less 4-grams than the text in other tasks, our BLEU-4 is lower than other text generation tasks. The substantial gap between CLIP-BART and BART or BART with caption indicates that captions usually carry more specific information than images, and the current multimodal encoders still cannot perfectly embed text and images into the same semantic space. Meanwhile, the low performance of the retrieval baselines shows that simple retrieval methods are insufficient to predict accurate next steps.

Model	Gardening				Crafts			
	1↓	2↓	3↓	4↓	1↓	2↓	3↓	4↓
Ground Truth	87.1	60.1	36.1	23.6	91.3	68.7	41.6	27.7
BART	93.7	84.3	72.9	64.2	96.9	90.6	80.6	73.5
+CP	92.8	81.3	68.9	60.5	96.3	89.3	79.2	72.5
+CP+M	96.2	89.9	81.4	73.9	**95.9**	**87.8**	76.6	68.5
+CP+M+R	**92.3**	**80.5**	**67.9**	**57.8**	96.9	89.6	78.6	71.1
+CP+M+R+CL	95.1	87.2	77.1	68.6	96.3	88.0	**75.8**	**67.3**

Table 6: Self-BLEU (%) for human or system steps.

Among our model variants, adding selective encoding leads to a further performance increase, showing that selective encoding helps the model focus on the content in step history that is most related to future steps. The superior performance on BARTScore and semantic similarity of the retrieval-augmented model indicates the effectiveness of the guidance from historically relevant steps. Our contrastive learning model achieves larger gains compared to baselines for BLEU and METEOR, suggesting that our contrastive loss helps the model generate results similar to the ground truth.

Automatic Evaluation with Future Steps We evaluate whether the predicted step is related to any future steps. Our contrastive learning model outperforms other ablations significantly on text retrieval for the Gardening domain, as shown in Table 2. These results imply that the contrastive learning objective encourages the model to generate more informative future steps. The decrease in n-gram overlap between input step history and step predictions (Table 5) suggests that the contrastive learning objective also decreases the model's paraphrasing tendency. Interestingly, the performance decreases when adding the retrieval augmentation to the model because the retrieval model introduces additional information related to the step history, which makes the model generate results similar to previous steps (Table 5).

Automatic Evaluation on Diversity To evaluate the diversity between generated steps in the test sets, we employ two diversity metrics: self-BLEU (Zhu et al., 2018) (Table 6) and unique n-grams (Fedus et al., 2018) (Table 7). The self-BLEU evaluates whether a model produces similar n-grams in different samples by measuring the similarity between one sentence and the rest in the test set. The retrieval model achieves the best results for the Gardening domain because it acquires additional knowledge from the retrieved steps and thus diversifies the output. The contrastive learning model achieves the best self-BLEU for 3,4 grams for the Crafts domain, implying our model's effectiveness. The unique n-grams calculate the percentage of distinct n-grams. It considers the repetition of n-grams within a generated step and across samples. The contrastive learning model achieves the highest distinct scores for 3,4 grams for both domains, indicating the effectiveness of our diversity-based contrastive loss in generating more diverse steps.

Model	Gardening				Crafts			
	1↑	2↑	3↑	4↑	1↑	2↑	3↑	4↑
Ground Truth	11.4	50.9	80.8	92.2	8.46	44.4	77.9	90.9
BART	4.75	17.7	32.4	42.6	5.11	22.6	42.8	53.8
+CP	**5.17**	19.2	33.7	42.7	5.12	22.6	42.7	53.8
+CP+M	4.94	18.6	32.8	41.8	4.92	22.4	42.3	53.8
+CP+M+R	5.06	19.2	34.6	44.3	**5.23**	**23.3**	43.9	55.2
+CP+M+R+CL	5.02	**19.3**	**35.0**	**45.2**	5.07	23.3	**44.2**	**56.1**

Table 7: Unique n-grams in human or system steps(%).

6.3 Human Evaluation

Model	Gardening				Crafts			
	N.↓	F.↓	D.↓	E.↓	N.↓	F.↓	D.↓	E.↓
BART	1.92	2.05	2.43	1.60	1.90	2.03	2.29	1.76
+CP	1.78	1.93	2.70	1.39	1.70	1.85	2.86	1.65
+CP+M	1.77	1.95	2.41	1.37	2.15	2.04	4.11	1.77
+CP+M+R	1.48	1.55	2.66	1.29	1.93	2.13	2.89	1.63
+CP+M+R+CL	**1.31**	**1.37**	**1.27**	**1.18**	**1.55**	**1.84**	**1.57**	**1.52**

Table 8: Human evaluations on with average ranking of next step correctness (N.), future steps correctness (F.), diversity (D.), executability (E.). Ties are allowed.

Since script learning is an open-ended task that is inherently difficult for automatic metrics to measure the correctness of generated scripts (Huang et al., 2022), we further conduct a human evaluation. We hire four proficient English speakers as human annotators to independently rank the generation results from 1 (best) to 5 (worst) for: (1) *next step correctness* which measures whether the generated results match the next step; (2) *future* *steps correctness* measuring whether the generated results match any of the future steps; (3) *diversity* which measures the diversity of generated results under the same subgoal; (4) *executability* which checks the generated results repeat or conflict with step history. We randomly select ten subgoals, including 41 and 44 generated steps from the test set for Gardening and Crafts separately.

The human evaluation results[5] are shown in Table 8. Our contrastive learning model performs best over all metrics on two datasets. By adding each component of our model, we observe a consistent trend in correctness to ground truth. However, we also observe that scores for selective encoding decrease because the output space with selective encoding is more constrained than the BART baseline, and the length of our generated sequence is not very long.

6.4 Discussions

Impact of Selective Multimedia Encoder The caption input helps the model understand the general step descriptions better. For example, given the activity *"cure azaleas of leaf gall"*, the step text only shows a generic instruction: *"rule out other diseases"*. However, the BLIP captioner generates *"a green leaf with white dots on it"* which helps the model generate *"remove the leaf gall from the shrub"* instead of *"keep your shrub healthy"*. Furthermore, in Figure 1, the finger object is absent from caption 3, indicating that the caption model has the ability to eliminate extraneous information from the image. The selective gate can filter out unrelated steps which are not directly related to the current subgoal. For example, in Figure 1, our model successfully predicts a low masking weight of 0.049324 for the step "cut the thread", while assigning a much higher masking weight of 0.134498 to its uninformative caption "a pair of scissors and a measuring tape". The results imply that the selective gate successfully guides the model to focus on the related information.

Impact of Retrieval Augmentation The retrieved steps provide relevant knowledge from similar tasks: given the subgoal *"finding or growing roses"* because the retrieved sentence mentioned *"fertilizer"* and *"mulch"*, the model successfully generates *"fertilize your roses"*. Additionally, the model also benefits from retrieval augmentation with an analogy, e.g., the model generates *"know when to harvest"* given the retrieved step *"plant the bulbs when you get them"*.

Impact of Contrastive Learning In addition to the improvement in diversity from the previous section, we observe that contrastive learning helps the model generate results closer to ground truth compared to other baselines. For example, it generates *"pick creeping charlie plants from the ground"*, similar to ground truth *"pick your creeping charlie leaves"*. The addition of contrastive learning also helps our model generates instructions with more details than other baselines by stating *"place the plant in the hole and cover it with soil"* instead of *"place the plant in the hole"*.

7 Related Work

Previous script learning tasks fall into two forms: selective and generative. The selective script learning tasks focus on modeling the script interactions given a list of candidates, including multi-choice goal step inference/ordering (Zhou et al., 2019; Zhang et al., 2020), script retrieval (Lyu et al., 2021; Zhou et al., 2022), action anticipation (Damen et al., 2018, 2021), procedure segmentation (Richard et al., 2018; Zhou et al., 2018; Ghoddoosian et al., 2022), multi-choice visual goal-step inference (Yang et al., 2021b), multimedia procedure planning (Zhao et al., 2022), multimedia step ordering (Zellers et al., 2021; Wu et al., 2022), instructional video retrieval (Yang et al., 2021a), and step classification (Lin et al., 2022). Despite promising results, their performance heavily relies on the given candidates, making them difficult to generalize for unseen activities. The second category is text-based generative script learning (Tandon et al., 2020; Lyu et al., 2021; Huang et al., 2022; Li et al., 2020, 2021; Jin et al., 2022; Sancheti and Rudinger, 2022). However, this is the first work to provide a multimedia goal-oriented generative script learning along with a new multimodal-retrieval based metric. Different from Sener and Yao (2019), which uses a video to generate the next step, our new task uses step image-text pairs as input. Unlike previous multimedia script learning frameworks with a multimedia encoder to capture visual and textual information, we use a captioner to convert images into captions summarizing the important objects in images. The GOSC dataset (Lyu et al., 2021) contains the steps of daily stereotypical tasks, but most of the steps (52.6%) in

[5] The Krippendorff-α inter-annotator agreement scores (Krippendorff, 2018) and detailed guidelines of human evaluations are in the Appendix K

this dataset are unordered, making it infeasible to evaluate the next-step prediction. Consequently, we adapted the best model mT5 (Xue et al., 2021) in GOSC to our settings, i.e., the monolingual version T5, and used it as an equivalent baseline to show the comparison with the state-of-the-art model.

To handle irrelevant sentences in the input, instead of using a token-level gating mechanism that only depends on the token itself (Sengupta et al., 2021), we introduce a sentence (step/caption) level gating mechanism whose gates depend on global context and weighted sentence representations. Our work is also related to retrieval-augmented text generation models (Wang et al., 2019; Lewis et al., 2020b; Liu et al., 2021). However, instead of retrieving knowledge from an external corpus, we use steps from similar tasks in training data to guide the generation process. Moreover, we introduce a new contrastive learning loss to increase diversity. Previous contrastive learning-based text generation methods usually use negative samples constructed by sequence manipulation (Cao and Wang, 2021; Hu et al., 2022) or perturbation (Lee et al., 2021). Inspired by Wang et al. (2022) which uses self-negatives for knowledge graph completion and that the generation output tends to repeat the input, we extend self-negatives for sequence-to-sequence contrastive learning. We also retrieve similar steps from the training set as additional hard negatives.

8 Conclusion

We propose a novel Multimedia Generative Script Learning task with the first benchmark featuring step and descriptive image pairs to generate subsequent steps given historical states in both text and vision modalities. Moreover, we build a new script learning framework consisting of a selective multimedia encoder, a retrieval-augmented decoder, and a diversity-oriented contrastive learning objective to generate the next steps. Furthermore, we define a new *multimodal-retrieval based metric* which can be used for multimedia script learning tasks. Automatic and human evaluation results demonstrate consistent performance improvements.

9 Limitations

9.1 Limitations of Data Collection

Regarding data collection, we crawled the English WikiHow website from Jan 2021 to May 2021. The number of available activities is limited by the data we crawled from WikiHow. We currently only choose *Gardening* and Crafts categories as case studies. Because we focus on multimedia image-step pairs, we remove steps *that* are not attached to any illustrative images. We also observe that a small portion of activities in the dataset do not follow chronological order.

Since our task focuses on the daily stereotypical tasks which usually require the model to understand the visual environment, the model design can be directly applied to support other domains, such as steps in the cooking videos. In addition, our model can also adapt to scenarios without visual images because the performance of our model only decreases slightly if no caption is provided. We plan to expand our model to other categories written in other languages.

9.2 Limitations of System Performance

The model might generate incorrect nouns because of the occurrence of patterns (e.g., *"refrigerate the **slane** for up to 1 year"* instead of *"refrigerate the **purslane** for up to 1 year"*). In addition, our model sometimes tends to generate generic step descriptions because of insufficient input information, e.g., given the last step *"lay the t-shirt out on a clean, flat surface."*, the model generates *"cut the shirt out"* which is vague compared to ground truth *"carefully cut around the sleeve"*. Moreover, the pretrained model might focus more on language modeling instead of inherent logic: for the activity of *"make paint can planters"*, after *"removing the label"* from the paint can, the BART+CAP generates *"read the label"*. In addition, there is still a small chance that the model generates the same output for various similar inputs.

Because we rely on image captions and retrieval results for step prediction, the upper bound of our generation quality is limited by the performance of the image caption and sentence retrieval modules. Our framework also needs to improve on imbalanced topics in the dataset. For example, the dataset contains more activities about *tree* for the gardening domain than other gardening-related plants. Because our multimedia generative script learning is a new task, we cannot compare our model with other established state-of-the-art models. Moreover, because WikiHow is a crowd-sourcing website, some everyday activities might have better human annotations than the remaining activities. We plan to include a fine-grained human written step prediction as an upper bound to address this issue.

9.3 Limitations of Evaluation

The automatic metrics we chose, including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Denkowski and Lavie, 2014), BARTScore (Yuan et al., 2021), self-BLEU (Zhu et al., 2018), and unique n-grams (Fedus et al., 2018), might not be the best metrics to evaluate our results. Some other metrics, such as semantic similarity and multimodal-retrieval based metrics, are based on pretrained models, including Augmented SBERT (Thakur et al., 2021), SentenceBert (Reimers and Gurevych, 2019), and CLIP (Radford et al., 2021). Those metrics might not align with human judgment and might be biased toward pretrained datasets. While we complement it with human evaluation, we only focus on relevance to ground truth and diversity. Although we found fluency is not an issue, it is likely we still need to cover all aspects of generation results.

10 Ethics and Broader Impact

The type of multimedia script learning framework we have designed in this paper is limited to WikiHow articles, and they might not be applicable to other scenarios.

10.1 Usage Requirement

Our multimedia script learning framework provides investigative leads for multimedia script prediction. Therefore, it is not intended to be used for any activity related to any human subjects. Instead, our system aims to generate step predictions with unseen activities similar to those in the training set. Accordingly, domain experts might use this tool as an assistant to write more constructive instructional scripts that would be too time-consuming for a human to create from scratch. Experts can also use this system to improve writing instruction by adding missing instructions. However, our system does not perform fact-checking or incorporate any external knowledge, which we leave as future work. The IRB board should first approve human subjects who follow instructions generated by our system.

10.2 Data Collection

We collect data by crawling the raw official English WikiHow website, which is under *Attribution-Noncommercial-Share Alike 3.0 Creative Commons License*[6]. We ensure that our data collection procedure follows the Terms of Use located at https://www.wikihow.com/wikiHow:Terms-of-Use. Therefore our dataset can only be used for non-commercial purposes. As mentioned in Section 6.3, we perform the human evaluation. All annotators involved in the human evaluation are voluntary participants and receive a fair wage.

[6]https://www.wikihow.com/wikiHow:Creative-Commons

Acknowledgement

This work is supported by Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture, and by U.S. DARPA KAIROS Program No. FA8750-19-2- 1004. 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 the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. Hou Pong Chan was supported in part by the Science and Technology Development Fund, Macau SAR (Grant Nos. FDCT/060/2022/AFJ, FDCT/0070/2022/AMJ) and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST).

References

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *Computation and Language*, arXiv:2205.14690.

Michael Beetz, Daniel Beßler, Jan Winkler, Jan-Hendrik Worch, Ferenc Bálint-Benczédi, Georg Bartels, Aude Billard, Asil Kaan Bozcuoğlu, Zhou Fang, Nadia Figueroa, Andrei Haidu, Hagen Langer, Alexis Maldonado, Ana Lucia Pais Ureche, Moritz Tenorth, and Thiemo Wiedemeyer. 2016. Open robotics research using web-based knowledge services. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5380–5387.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings* of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

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 (CVPR)*, pages 3558–3568.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2021. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the 5th International Conference on Learning Representations*.

William Fedus, Ian Goodfellow, and Andrew M. Dai. 2018. MaskGAN: Better text generation via filling in the _. In *Proceedings of the 6th International Conference on Learning Representations*.

Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. 2022. Hierarchical modeling for task recognition and action segmentation in weakly-labeled instructional videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1922–1932.

Yoav Goldberg. 2018. Neural language generation. Technical report.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, page 25–30, New York, NY, USA. Association for Computing Machinery.

Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. 2022. PLANET: Dynamic content planning in autoregressive transformers for long-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2305, Dublin, Ireland. Association for Computational Linguistics.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Machine Learning Repository*, arXiv:2201.07207. Version 2.

Xiaomeng Jin, Manling Li, and Heng Ji. 2022. Event schema induction with double graph autoencoders. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2013–2025.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In *Proceedings of the 9th International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Computer Vision and Pattern Recognition*, arXiv:2201.12086.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695.

Sha Li, Ruining Zhao, Manling Li, Heng Ji, Chris Callison-Burch, and Jiawei Han. 2023. Open-domain hierarchical event schema induction by incremental prompting and verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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*, pages 740–755, Cham. Springer International Publishing.

Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. 2022. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13853–13863.

Jiateng Liu, Sha Li, Zhenhailong Wang, Manling Li, and Heng Ji. 2023. A language first approach for procedural planning. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics.

Junjia Liu, Yiting Chen, Zhipeng Dong, Shixiong Wang, Sylvain Calinon, Miao Li, and Fei Chen. 2022. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 2021. A Three-Stage Learning Framework for Low-Resource Knowledge-Grounded Dialogue Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2262–2272, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

Qing Lyu, Li Zhang, and Chris Callison-Burch. 2021. Goal-oriented script construction. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 184–200, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *Machine Learning Repository*, arXiv:1807.03748.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages

3982–3992, Hong Kong, China. Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A. Richard, H. Kuehne, and J. Gall. 2018. Action sets: Weakly supervised action segmentation without ordering constraints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5996, Los Alamitos, CA, USA. IEEE Computer Society.

Leonard Richardson. 2007. Beautiful soup documentation. *April*.

D. Ruth Anita Shirley, K. Ranjani, Gokulalakshmi Arunachalam, and D. A. Janeera. 2021. Automatic distributed gardening system using object recognition and visual servoing. In *Inventive Communication and Computational Technologies*, pages 359–369, Singapore. Springer Singapore.

Abhilasha Sancheti and Rachel Rudinger. 2022. What do large language models learn about scripts? In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 1–11, Seattle, Washington. Association for Computational Linguistics.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *Proceedings of Data Centric AI NeurIPS Workshop)*.

Fadime Sener and Angela Yao. 2019. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Ayan Sengupta, Amit Kumar, Sourabh Kumar Bhattacharjee, and Suman Roy. 2021. Gated Transformer for Robust De-noised Sequence-to-Sequence Modelling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3645–3657, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5227–5237.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 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*, pages 296–310, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental draft generation of scientific ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1980–1991, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2022. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4525–4542, Dublin, Ireland. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya

Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yue Yang, Joongwon Kim, Artemis Panagopoulou, Mark Yatskar, and Chris Callison-Burch. 2021a. Induce, edit, retrieve: Language grounded multimodal schema for instructional video retrieval. *Computer Vision and Pattern Recognition*, arXiv:2111.09276.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021b. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. In *Advances in Neural Information Processing Systems*, volume 34, pages 23634–23651. Curran Associates, Inc.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.

He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G. Derpanis, Richard P. Wildes, and Allan D. Jepson. 2022. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2938–2948.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 2022. Show me more details: Discovering hierarchies of procedures from semi-structured web data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2998–3012, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhou, Julie Shah, and Steven Schockaert. 2019. Learning household task knowledge from WikiHow descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 50–56, Macau, China. Association for Computational Linguistics.

Yu Zhou, Sha Li, Manling Li, Xudong Lin, Shih-Fu Chang, Mohit Bansal, and Heng Ji. 2023. Non-sequential graph script induction via multimedia grounding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Hyperparameters

Our model is built based on the Huggingface framework (Wolf et al., 2020)[7]. We choose top 5 retrieved historically relevant steps as input for our retrieval model. We choose 5 negative samples for each step during contrastive learning for the gardening domain. Specifically, 4 self-negative samples, including steps and captions, are randomly chosen from the title, method, and step history input. The remaining 1 retrieved negative samples are randomly chosen from top-20 most similar steps retrieved from the training set based on the last step. For the crafts domain, we choose 5 self-negative samples and 5 retrieved negative samples. We set τ as 1 for contrastive loss and λ as 0.5 based on validation performance for the training objectives. We optimize our model by AdamW (Loshchilov and Hutter, 2019) with Cosine Annealing Warm Restarts schedule (Loshchilov and Hutter, 2017). Our learning rate is 1×10^{-5} with $\epsilon = 1 \times 10^{-6}$ for gardening domain and 2×10^{-5} with $\epsilon = 1 \times 10^{-6}$ for crafts domain. The number of warm-up steps is 2000. The batch size is set to 16 for both domains, and the maximum training epoch is set as 30 with 10 patience. During decoding, we use beam-search to generate results with a beam size of 5.

	# of Parameters
BART	139.425M
+CP	139.425M
+CP+M	141.788M
+CP+M+R	158.346M
+CP+M+R+CL	158.347M

Table 9: # of Model Parameters

B Training details

We use BART-base from Huggingface (Wolf et al., 2020) for our method and baselines. We normalize all our input sentences into lower case. We add 5 special tokens for BART-base model including <title>, <method>, <step>, <caption>, <template>, and <cls>. We prepend <title> to goal, <method> to subgoal, <step> to text step, <caption> to step caption, <template> to retrieved step, and <cls> to the beginning of step history input. We truncate our step, caption, goal, and subgoal to 30 tokens and target step to 40. We only choose the closest 10 step-caption pairs. We use BLIP (Li et al., 2022) [8] pretrained with 129M images including including COCO (Lin et al., 2014), Visual Genome (Krishna et al., 2017), Conceptual Captions (Sharma et al., 2018), Conceptual 12M (Changpinyo et al., 2021), SBU (Ordonez et al., 2011), and LAION (Schuhmann et al., 2021). We use $\mathtt{all-mpnet-base-v2}$ from SentenceBert (Reimers and Gurevych, 2019), which performs best in semantic search to retrieve similar steps.

# History	# Instance	BARTScore↑	Semantic↑
1	685	-4.3683	0.3189
2	680	-4.3633	0.3115
3	545	-4.4213	0.3064
4	346	-4.3535	0.3118
5	207	-4.3556	0.2748
6	104	-4.3588	0.2746
7	56	-4.2192	0.3381
8	26	**-4.1687**	**0.3411**
9	12	-4.3800	0.2085
10	23	-4.7718	0.2491

Table 10: The average number of BARTScore/ Semantic Similarity Score and the number of instances given the different lengths of step history for the gardening domain

We train our model with NVIDIA A6000 GPUs with 48G memory with full precision. We choose our best model based on the validation score with BLEU-4 (Papineni et al., 2002) and ROUGE (Lin, 2004). The best validation scores for our contrastive learning model are: BLEU-4 with 2.81 and ROUGE-L with 15.24 for the gardening domain; BLEU-4 with 4.85 and ROUGE-L with 20.25 for the crafts domain. The average training time for each model is 2 to 4 hours. Table 9 shows the number of parameters for each model.

C Evaluation Metrics

We use BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Denkowski and Lavie, 2014) from Microsoft COCO Caption Evaluation package[9]. We use official implementation of BARTScore (Yuan et al., 2021)[10]. We use $\mathtt{cross-encoder/stsb-roberta-large}$ which performs best on STSBenchmark (Cer et al., 2017) to compute semantic similarity score from Augmented SBERT (Thakur et al., 2021). For multimodal-retrieval based metric, we use the best sentence embedding model: $\mathtt{all-mpnet-base-v2}$ from SentenceBert (Reimers and Gurevych, 2019) for text retrieval, and the best language-image pretraining model $\mathtt{ViT-L/14@336px}$ from CLIP (Radford et al., 2021) for image retrieval. Specifically, we compute the CLIP similarity between the image embedding and the sentence embedding of the target step to retrieve images. All results are based on a single run. We have opted not to include a human performance baseline in our evaluation. This decision was made due to the inherent challenges of assessing human performance in generative script learning, which requires annotators to possess domain knowledge in order to predict the next steps accurately. Moreover, different tasks may require different levels of expertise, experience, or background knowledge, making it difficult to establish a consistent baseline for human performance evaluation.

D Additional Ablation Study

We conducted further ablation experiments, the results of which are presented in Table 11. Our findings show that all ablated models performed worse than our proposed model.

[7] https://github.com/huggingface/transformers

[8] The BLIP checkpoint we is `https://storage.googleapis.com/sfr-vision-language-research/BLIP/models/model_base_capfilt_large.pth`

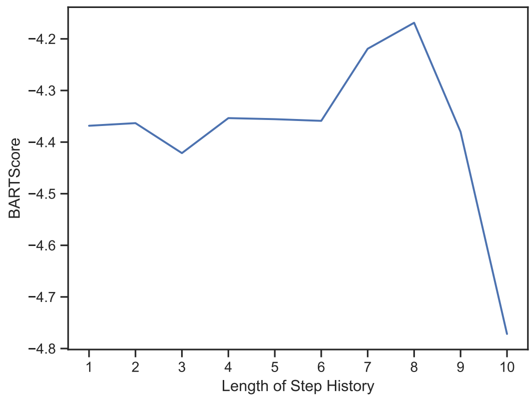
[9] `https://github.com/salaniz/pycocoevalcap`

[10] `https://github.com/neulab/BARTScore`

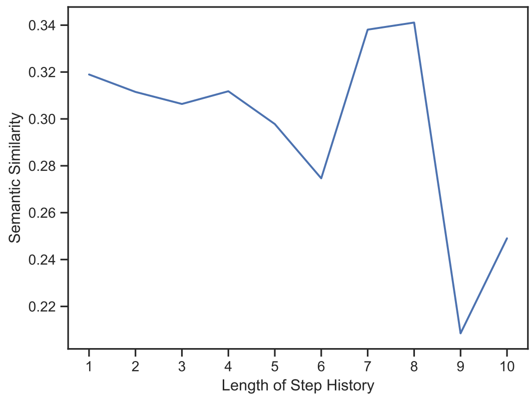
Domain	Model	B-1↑	B-2↑	B-3↑	B-4↑	METEOR↑	R-L↑	BARTScore↑	Semantic↑
Gardening	BART+CP+M+CL	**17.9**	**9.30**	**5.20**	**3.07**	**9.72**	**17.1**	**-4.39**	0.304
	BART+CP+R+CL	17.6	9.16	5.16	3.03	9.54	16.7	-4.41	0.299
	BART+M+R+CL	17.7	9.11	4.98	2.92	9.71	17.0	-4.37	**0.306**
Crafts	BART+CP+M+CL	20.6	10.9	6.12	3.89	10.8	19.3	-4.30	**0.307**
	BART+CP+R+CL	20.3	11.0	6.36	4.12	10.8	19.8	-4.29	0.301
	BART+M+R+CL	**20.8**	**11.5**	**6.78**	**4.49**	**10.9**	**20.1**	**-4.27**	0.306

Table 11: Automatic evaluation results on next step prediction for the gardening and crafts domain (%).

E Prediction for different history length



(a) The average number of BARTScore in test set given the different lengths of step history



(b) The average number of Semantic Similarity Score in test set given the different lengths of step history

Figure 3: Prediction for different history lengths for the gardening domain

In Figure 3a and Figure 3b, we show the averaged BARTScore and semantic similarity scores of our contrastive learning models in the next step prediction task over different step history lengths. In both figures, we observe that the results with eight step-caption pairs obtain the highest scores. We analyze the reasons as follows. For the instances that contain less than eight history steps, increasing the step history introduces more information than noise from the step text and corresponding captions. However, as the step length grows, the additional step-caption pairs introduce more noise than information relevant to the future step. Empirically, the eight-step length achieves an optimal balance between noise and relevant information. Another potential reason is related to the number of instances. In Table 10, we see a clear decline in the number of instances because of our dataset construction strategy. Therefore, the model cannot generalize over long history input.

F Dataset Collections

We crawled the English WikiHow website from Jan 2021 to May 2021. We extract all articles from the crawled website dump in the *Gardening* and *Crafts* categories. Each article contains a unique activity. We use BeautifulSoup (Richardson, 2007) to parse the article and obtain JSON files. Each JSON file contains a gardening activity. For each gardening activity, we remove those steps without paired images or steps whose images do not exist in the dump. Then, we use a regular expression to remove the URLs in the steps. We remove those steps that are too short (less than two words) or contain no values. Finally, we remove the activity containing only one step in each subgoal.

G Parallel Steps

In this paper, we focus on predicting correct orders for sequential step prediction since we find that only 18% of the subgoals have one parallel step by random checking 50 subgoals, and 14% contain more than one parallel step. It is more critical to predict correct orders for non-interchangeable steps, such as step 4 and 5 in Figure 1. By using generative methods, multiple steps can be predicted with different probabilities, which can support parallel processes. We also propose the multimodal-retrieval-based metric by treating the future steps as a set and checking whether the generation steps fall into the future steps.

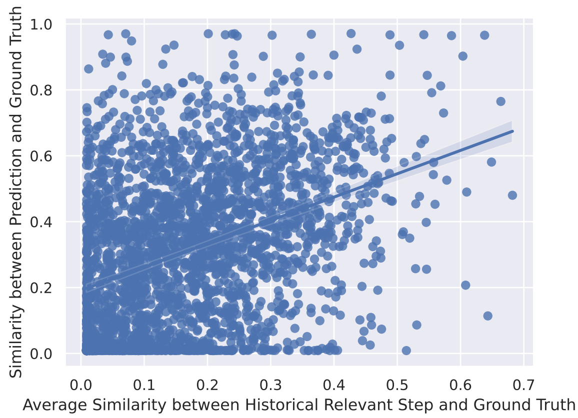


Figure 4: The semantic similarity scores (Thakur et al., 2021) between the model predictions and the ground truths versus the semantic similarity scores between the retrieved historical relevant steps and the ground truths in the gardening domain.

H Impact of Historical Relevant Steps

We analyze the relation between the quality of the retrieved historically relevant steps and the quality of the model predictions. The semantic similarity score evaluates the quality of retrieved steps and model predictions, which measures the embedding space similarity between a given text and the ground-truth next step. Pearson's correlation between the semantic scores of historically relevant steps and the semantic scores of model predictions is 0.39 with a $p < 0.01$. We also illustrate their relation in Figure 4. The results suggest that the performance of our model is positively correlated with the relevance of the retrieved historical steps.

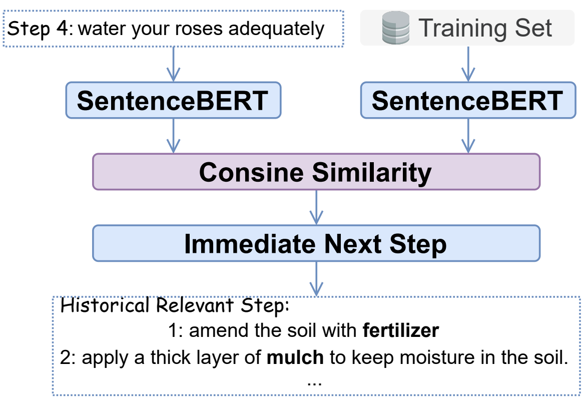


Figure 5: Details for historical relevant step retrieval

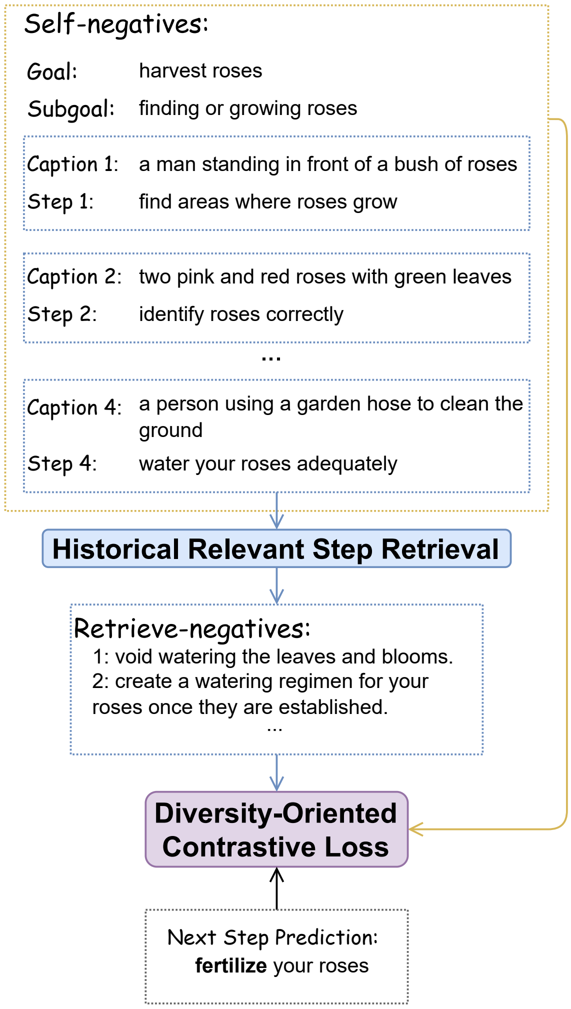


Figure 6: Details for diversity-oriented contrastive loss

I Additional Model Architecture

Figure 5 and 6 show additional details for our framework. The immediate next step refers to the step right after the previously given steps.

J Scientific Artifacts

We list the licenses of the scientific artifacts used in this paper: WikiHow (Attribution-Noncommercial-Share Alike 3.0 Creative Commons License), Huggingface Transformers (Apache License 2.0), SBERT (Apache-2.0 license), BARTScore (Apache-2.0 license), CLIP (MIT license), and BLIP (BSD-3-Clause license).

K Human evaluation details

Model	Gardening				Crafts			
	N.	F.	D.	E.	N.	F.	D.	E.
BART	0.60	0.64	0.55	0.22	0.60	0.59	0.70	0.35
+CP	0.65	0.50	0.53	0.41	0.67	0.60	0.90	0.31
+CP+M	0.70	0.74	0.86	0.31	0.45	0.40	0.76	0.41
+CP+M+R	0.53	0.50	0.68	0.37	0.62	0.46	0.78	0.31
+CP+M+R+CL	0.43	0.58	0.56	0.26	0.58	0.48	0.13	0.35

Table 12: Krippendorff-α scores for human evaluation on with average ranking of next step correctness (N.), future steps correctness (F.), diversity (D.), executability (E.).

We measure inter-annotator agreement with Krippendorff-α scores (Krippendorff, 2018). The results are in Table 12. Table 13 shows the annotation examples. Because we do not have a virtual environment to execute those steps, we do not have a good inter-annotator agreement on the executability.

L Sample Output

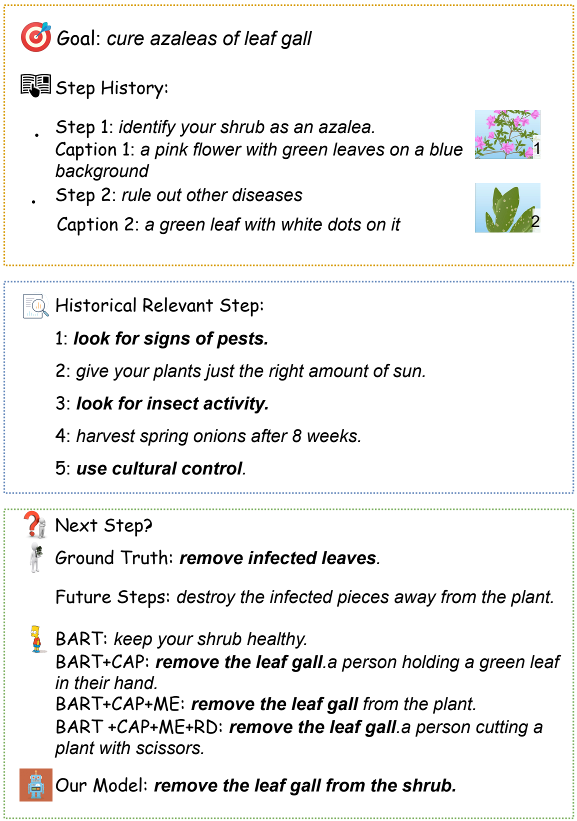


Figure 7: Human and System Step Prediction Results. It shows an example that our model benefits from selective multimedia encoder.

Type	Content
Instructions	(1) similarity to the next step measures the correctness of generated results with the next step; (2) similarity to future steps measures whether the generated results are relevant to the future steps; (3) diversity measures the diversity of generated results under the same subgoal (4) executability which checks the generated results repeat or conflict with step history/ Please rank these models' output from 1(best)-5(worst), ties are allowed if both outputs are the same.
Similarity and executability annotation examples	*Title:* protect garden berries *Subgoal:* setting up decoys *Step History:* use plastic snakes. ———————————- *Ground Turth Target:* *Next Step:* put out shiny pinwheels. *Future Steps:* put out shiny pinwheels. create a decoy food area. ———————————- *Predictions:* *0's prediction:* wrap the snake in a plastic bag. *1's prediction:* set up a trellis. *2's prediction:* cut the berries down to the ground. *3's prediction:* set up a trap. *4's prediction:* choose a sturdy piece of string.
Diversity	*0's predictions:* wrap the snake in a plastic bag. place the flowers on a stick in the dirt. *1's predictions:* choose the right plant. set up a trap. *2's predictions:* cut the berries down to the ground. create a trap. *3's predictions:* set up a trap. create a trap. *4's predictions:* choose a sturdy piece of string. set up a trap.

Table 13: Annotation examples

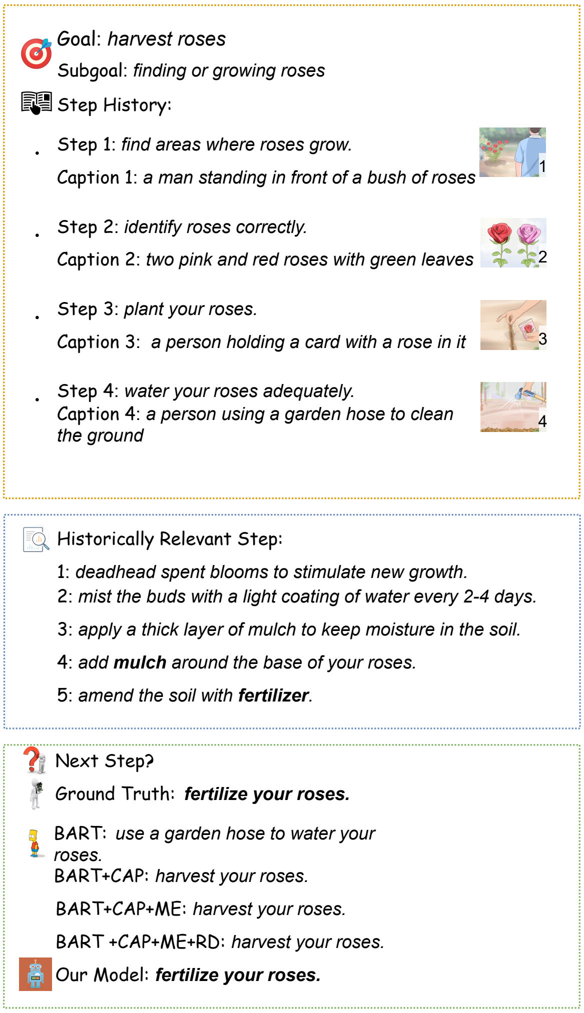


Figure 8: Human and System Step Prediction Results. It shows an example that our model prediction results benefits from retrieval results and contrastive learning.

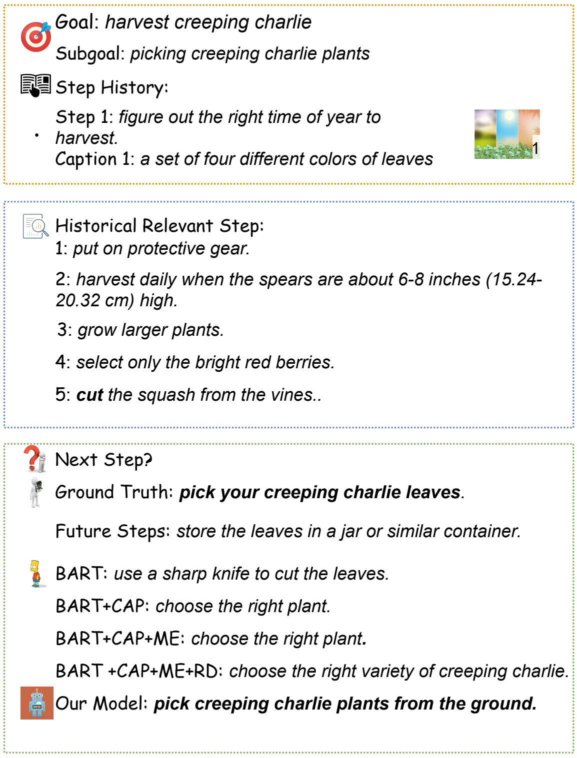


Figure 9: Human and System Step Prediction Results. It shows an example that our model prediction results benefits from retrieval results and contrastive learning.

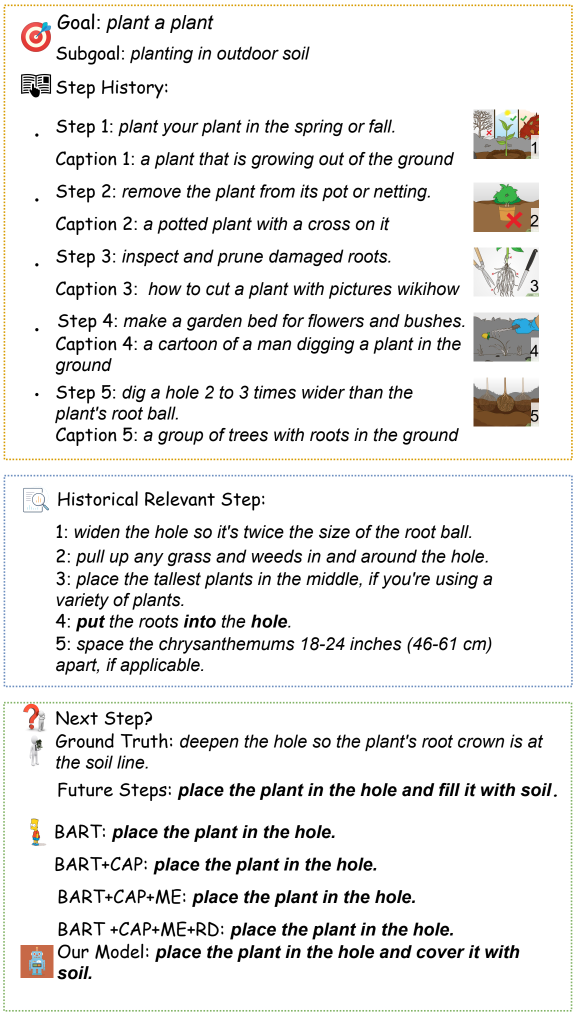


Figure 10: Human and System Step Prediction Results. It shows an example that our model prediction results matches future steps instead of immediate next step.