MULTIMEDIA GENERATIVE SCRIPT LEARNING FOR TASK PLANNING

Anonymous authors
Paper under double-blind review

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 in daily life. 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 steps with descriptive images. 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 quality and inductive quality. Experiment results demonstrate that our approach significantly outperforms strong baselines[1].

1 INTRODUCTION

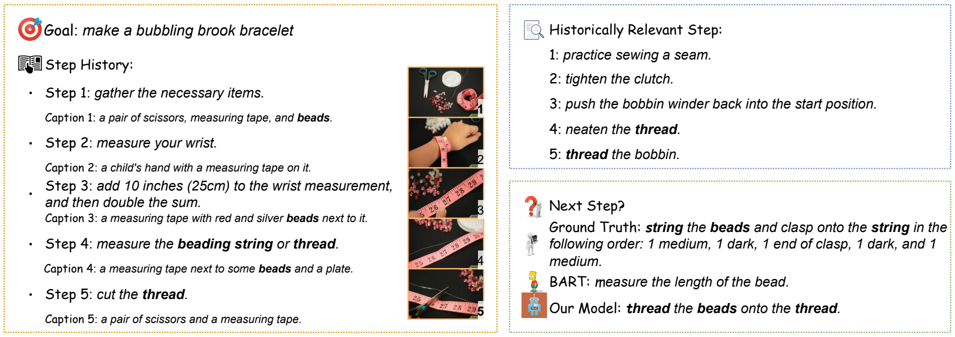
Robots rely on understanding the present real-world state and predicting the subsequent steps to better assist humans in daily stereotypical tasks such as meal preparation and gardening (Ruth Anita Shirley et al., 2021; Liu et al., 2022). As an example, Robohow (Beetz et al., 2016) uses articles from WikiHow[2] to assist robots in everyday tasks in human working and living environments. The problem, however, 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) 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 script learning focuses solely on text (Lyu et al., 2021; Huang et al., 2022), which is commonly affected by reporting bias (Gordon & Van Durme, 2013), since 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] Code and data resources will be made publicly available for research purposes.

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



Goal: *make a bubbling brook bracelet*

Step History:

- Step 1: *gather the necessary items.*

  Caption 1: *a pair of scissors, measuring tape, and* ***beads****.*
- Step 2: *measure your wrist.*

  Caption 2: *a child's hand with a measuring tape on it.*
- Step 3: *add 10 inches (25cm) to the wrist measurement, and then double the sum.*

  Caption 3: *a measuring tape with red and silver* ***beads*** *next to it.*
- Step 4: *measure the* ***beading string*** *or* ***thread****.*

  Caption 4: *a measuring tape next to some* ***beads*** *and a plate.*
- Step 5: *cut the* ***thread****.*

  Caption 5: *a pair of scissors and a measuring tape.*

Historically Relevant Step:

1: *practice sewing a seam.*

2: *tighten the clutch.*

3: *push the bobbin winder back into the start position.*

4: *neaten the* ***thread****.*

5: ***thread*** *the bobbin.*

Next Step?

Ground Truth: ***string*** *the* ***beads*** *and clasp onto the* ***string*** *in the following order: 1 medium, 1 dark, 1 end of clasp, 1 dark, and 1 medium.*

BART: *measure the length of the bead.*

Our Model: ***thread*** *the* ***beads*** *onto the* ***thread****.*

Figure 1: **Multimedia Generative Script Learning:** The left 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.

(1) *Visual-State Trackable*: 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 example, 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) *Inductive*: 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. As an example, the future step in Figure 1 closely resembles the scripts used in previous retrieved steps about *threading items*, thus enabling script knowledge to be transferred to an unseen task.

(3) *Diverse*: 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.

To evaluate task performance, in addition to traditional generation-based metrics, 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 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 steps with descriptive images. (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 Method

2.1 Problem Formulation

We propose a new multimedia generative script learning task as follows. Given an activity goal G, an optional subgoal M that specifies the concrete needs, and the previous multimedia step history

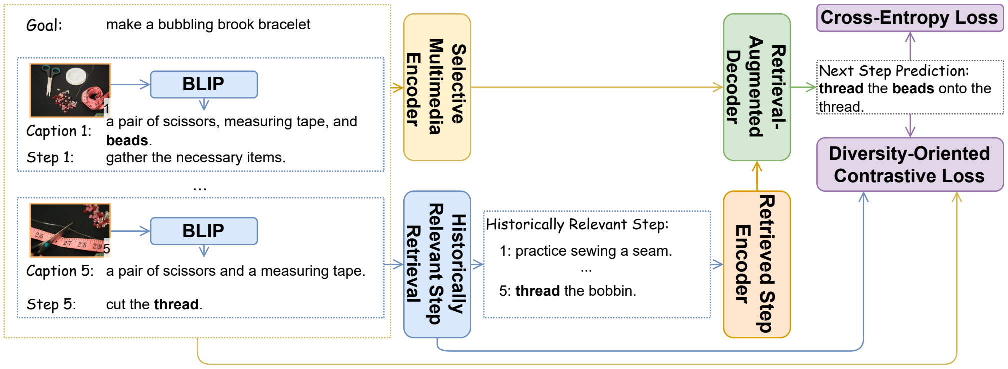

Goal: make a bubbling brook bracelet
1
BLIP
Caption 1: a pair of scissors, measuring tape, and **beads**.
Step 1: gather the necessary items.
...
5
BLIP
Caption 5: a pair of scissors and a measuring tape.
Step 5: cut the **thread**.
Selective Multimedia Encoder
Historically Relevant Step Retrieval
Historically Relevant Step:
1: practice sewing a seam.
...
5: **thread** the bobbin.
Retrieved Step Encoder
Retrieval-Augmented Decoder
Next Step Prediction: **thread** the **beads** onto the thread.
Cross-Entropy Loss
Diversity-Oriented Contrastive Loss

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

$\mathcal{H}_n = \{(S_1, V_1), ..., (S_n, V_n)\}$ with length h, a model is expected to predict the next possible step S_{n+1}, where $S_{i\in\{1,...,n+1\}}$ are sequences of words, and $V_{i\in\{1,...,n\}}$ are images.

2.2 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.

2.3 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 $X_{2i} = C_i$. We prepend a learnable [CLS] token to the sequence as a contextualized vector to summarize the step history. 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 $\mathbf{H} = \{\mathbf{h}_0, ..., \mathbf{h}^{2n}_{L_{X_{2n}}}\} = \text{Enc}(\mathcal{X})$. 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 sequence X_j.

Since the input sequence contains steps or captions that are 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 $(\mathrm{MultiHead})$ (Vaswani et al., 2017) for each sequence hidden states $\mathbf{H}_{X_j}$:

$$\hat{\mathbf{h}}_{X_j} = \mathrm{MultiHead}(\mathbf{h}_0, \mathbf{H}_{X_j}, \mathbf{H}_{X_j}) \tag{1}$$

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:

$$\alpha_j = \sigma(\mathbf{W}_\alpha[\mathbf{h_0}; \hat{\mathbf{h}}_{X_j}]) \tag{2}$$

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

$$\bar{\mathbf{H}}_{X_j} = \alpha_j \cdot \mathbf{emb}_{\texttt{[MASK]}} + (1 - \alpha_j)\mathbf{H}_{X_j} \tag{3}$$

where $\mathbf{emb}_{\texttt{[MASK]}}$ is the embedding of the `[MASK]` token. The final hidden state sequences from the selective multimedia encoder are therefore $\bar{\mathbf{H}} = [h_0; \bar{\mathbf{H}}_1; ...; \bar{\mathbf{H}}_{2n}]$.

2.4 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 & 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[3] from the representation given by SentenceBERT. 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 $\mathcal{R}_{n+1} = \{R_1, ..., R_k\}$ with length $L_{R_1}, ..., L_{R_k}$, 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 a 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 from historically relevant step encoder is $\bar{\mathbf{H}}_R = [\bar{\mathbf{H}}_{R_1}; ...; \bar{\mathbf{H}}_{R_k}]$.

2.5 Retrieval-Augmented Decoder

In the decoder, we compute the probability $P\left(s_{q,(n+1)}|s_{<q,(n+1)}, \hat{\mathcal{H}}, G, M\right)$ for the q-th token. 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_q^l which is the hidden state of $s_{q,(n+1)}$ in layer l, we first use a multi-head cross-attention to fuse the hidden states from the retrieved steps $\bar{\mathbf{H}}_{\mathbf{R}}$: ${z'_q}^l = \mathrm{MultiHead}(z_q^l, \bar{\mathbf{H}}_R, \bar{\mathbf{H}}_R)$. We also append a gating mechanism to control the knowledge from the retrieved steps and previous hidden states as:

$$\begin{aligned} \gamma &= \sigma(\mathbf{W}_\gamma[z_q^l; {z'_q}^l]) \\ \tilde{z}_q^l &= \gamma \cdot \mathrm{LN}({z'_q}^l) + (1 - \gamma) \cdot (z_q^l) \end{aligned} \tag{4}$$

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 as follows:

$$\mathcal{L}_{\mathrm{gen}} = \sum_{q=1}^{|s^{n+1}|} \log P\left(s_{q,(n+1)}|s_{<q,(n+1)}, \hat{\mathcal{H}}, G, M\right) \tag{5}$$

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

2.6 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 paraphrase 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 generation models suffer from the "exposure bias" problem (Dhingra et al., 2016; An et al., 2022) because they are trained by *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.

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):

$$\begin{aligned}\mathcal{L}_{\mathrm{cl}} &= \frac{\exp\left(y^+/\tau\right)}{\sum_k \exp\left(y_k^-/\tau\right) + \exp\left(y^+/\tau\right)} \\ y^+ &= \sigma(\mathrm{Avg}(\mathbf{W}_y\bar{\mathbf{H}}^+ + \mathbf{b}_y)) \\ y_k^- &= \sigma(\mathrm{Avg}(\mathbf{W}_y\bar{\mathbf{H}}_k^- + \mathbf{b}_y))\end{aligned} \tag{6}$$

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 $\mathrm{Avg}(*)$ denotes the average pooling function.

2.7 Training Objective

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

3 Evaluation Benchmark

3.1 Dataset

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 intermediate process to accomplish a *goal*, and *steps* which are specific actions to complete a *subgoal*. For each WikiHow article, we collect step-image pairs as well as their goals and methods[4]. We split the whole dataset based on the task categories. Therefore, the validation and test sets contain tasks that are not included in the training set. Table 1 shows the detailed data statistics.

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

Domain	Split	#Task	#Pair	$\overline{\#\mathbf{Step}}$	$\overline{\#\mathbf{Token}}$
Gardening	Train	1,857	20,258	3.10	11.6
	Validation	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
	Validation	3,33	4,061	6.12	9.10
	Test	3,33	3,937	5.91	9.00

3.2 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 & Lavie, 2014) scores to examine the content overlap between generated steps and ground truth.

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

Table 2: Results with automatic evaluation on next step prediction for the gardening domain (%). *CP* is models with caption input. *ME* is models with selective multimedia encoder. *RD* is models with historically relevant step encoder and retrieval-augment decoder. *CL* is models with diversity-oriented contrastive learning. *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	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+ME	17.8	9.36	5.30	3.19	9.61	**17.4**	-4.38	0.305
+CP+ME+RD	17.5	9.22	5.25	3.13	9.60	17.2	**-4.36**	0.309
+CP+ME+RD+CL	**18.4**	**9.72**	**5.51**	**3.31**	**9.91**	17.3	-4.37	**0.310**

Table 3: Automatic evaluation results on next step prediction for the crafts domain (%).

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+ME	20.5	11.1	6.61	4.40	10.79	20.1	-4.28	0.305
+CP+ME+RD	20.7	11.5	6.93	4.66	11.02	**20.5**	**-4.25**	0.309
+CP+ME+RD+CL	**21.3**	**11.8**	**7.12**	**4.85**	**11.25**	20.3	-4.26	**0.313**

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 2.4, we use SBERT (Reimers & 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 as 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 our evaluation setup are in the Appendix C.

Table 4: Multimodal-retrieval based future steps evaluation (%).

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+ME	**49.8**	41.0	**50.3**	**37.8**
+CP+ME+RD	48.1	38.9	48.9	31.8
+CP+ME+RD+CL	49.5	**43.0**	49.0	33.9

4 EXPERIMENTS

4.1 BASELINES

We first compare our model with **(1) state-of-the-art pretrained text-only generation models** 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 naive retrieval baseline which directly uses retrieved historically relevant sentences as discussed in Section 2.4, 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 better image-text alignment. Additionally, we compare our model with a state-of-the-art script learning model: GPT2-SIF Sancheti & Rudinger (2022) finetuned on our dataset. Finally, we include the variances of our model as **(4) baselines for ablation**. The hyperparameters and training details are presented in the Appendix A, B.

4.2 AUTOMATIC EVALUATION

As shown in Table 2 and 3, 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 straightforward 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 not sufficient to predict accurate next steps.

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

Model	**Gardening**				**Crafts**			
	1↓	**2↓**	**3↓**	**4↓**	**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+ME	43.6	**5.75**	**0.78**	**0.20**	36.4	1.97	**0.02**	0.01
+CP+ME+RD	44.2	6.32	1.12	0.38	36.9	2.03	0.06	**0.01**
+CP+ME+RD+CL	43.3	6.23	1.01	0.35	36.2	1.91	0.05	0.02

Among our model variants, adding selective encoding leads to a further performance increase, demonstrating that selective encoding helps the model concentrate 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 indicate the effectiveness of the guidance from historical relevant steps. Our contrastive learning model achieves larger gains compared to baselines with respect to BLEU and METEOR, suggesting that our contrastive loss helps the model generate results similar to the ground truth.

Table 6: Self-BLEU (%) for human written steps and system results.

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+ME	96.2	89.9	81.4	73.9	**95.9**	**87.8**	76.6	68.5
+CP+ME+RD	**92.3**	**80.5**	**67.9**	**57.8**	96.9	89.6	78.6	71.1
+CP+ME+RD+CL	95.1	87.2	77.1	68.6	96.3	88.0	**75.8**	**67.3**

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 4. 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, which further indicates the effectiveness of our diversity-based contrastive loss in generating more diverse steps.

Table 7: Unique n-grams appear in human written steps or system results for test set (%) .

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+ME	4.94	18.6	32.8	41.8	4.92	22.4	42.3	53.8
+CP+ME+RD	5.06	19.2	34.6	44.3	**5.23**	**23.3**	43.9	55.2
+CP+ME+RD+CL	5.02	**19.3**	**35.0**	**45.2**	5.07	23.3	**44.2**	**56.1**

4.3 Human Evaluation

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.

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

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+ME	1.77	1.95	2.41	1.37	2.15	2.04	4.11	1.77
+CP+ME+RD	1.48	1.55	2.66	1.29	1.93	2.13	2.89	1.63
+CP+ME+RD+CL	**1.31**	**1.37**	**1.27**	**1.18**	**1.55**	**1.84**	**1.57**	**1.52**

The human evaluation results[5] are shown in Table 8. Our contrastive learning model achieves the best performance 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.

4.4 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*

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

shrub" instead of *"keep your shrub healthy"*. The selective gate successfully filters out unrelated steps which are not directly related to the current subgoal. For example, in Figure 1, the selective gate successfully ignores Step 2 and generates *"thread the beads onto the thread"*.

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"*.

5 Related Work

Previous script learning tasks fall into two forms: selective and generative. The first category focuses on modeling the script interactions given a list of candidates. The selective script learning tasks include 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); Ghoddoosian et al. (2022); Zhou et al. (2018), 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; Sancheti & 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 & Yao (2019), which uses a video as input to generate the next step, our generative script learning 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.

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 (Liu et al., 2021; Lewis et al., 2020b). 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 & 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 the fact 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.

6 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. 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. 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. Additional external knowledge will be added in the future. Limitations and broader impact are in the Appendix.

REFERENCES

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. Cont: Contrastive neural text generation. *Computation and Language*, arXiv:2205.14690, 2022. URL http://arxiv.org/abs/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. Open robotics research using web-based knowledge services. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5380–5387, 2016. doi: 10.1109/ICRA.2016.7487749. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7487749&casa_token=1t93OuETwvIAAAAA:nklBGsyiIrn4SSJN6wcqPg6WQCKt8rB_3OdzuEc0tdZ6TprS-05L3qHpM4JnEcGgVpuQ4c3xJDU&tag=1.

Shuyang Cao and Lu Wang. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6633–6649, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.532. URL https://aclanthology.org/2021.emnlp-main.532.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 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)*, pp. 3558–3568, June 2021.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1931–1942. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/cho21a.html.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. URL http://openaccess.thecvf.com/content_ECCV_2018/html/Dima_Damen_Scaling_Egocentric_Vision_ECCV_2018_paper.html.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. URL https://doi.org/10.1007/s11263-021-01531-2.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 376–380, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3348. URL https://aclanthology.org/W14-3348.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. Sequence level training with recurrent neural networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2016. URL https://arxiv.org/pdf/1511.06732.pdf.

William Fedus, Ian Goodfellow, and Andrew M. Dai. MaskGAN: Better text generation via filling in the _. In *Proceedings of the 6th International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByOExmWAb.

Reza Ghoddoosian, Saif Sayed, and Vassilis Athitsos. 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*, pp. 1922–1932, 2022.

Yoav Goldberg. Neural language generation. Technical report, 2018. URL `https://inlg2018.uvt.nl/wp-content/uploads/2018/11/INLG2018-YoavGoldberg.pdf`.

Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pp. 25–30, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324113. doi: 10.1145/2509558.2509563. URL `https://doi.org/10.1145/2509558.2509563`.

Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. 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)*, pp. 2288–2305, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.163. URL `https://aclanthology.org/2022.acl-long.163`.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Machine Learning Repository*, arXiv:2201.07207, 2022. URL `http://arxiv.org/abs/2201.07207`. version 2.

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

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. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, may 2017. ISSN 0920-5691. doi: 10.1007/s11263-016-0981-7. URL `https://doi.org/10.1007/s11263-016-0981-7`.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. Contrastive learning with adversarial perturbations for conditional text generation. In *Proceedings of the 9th International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Wga_hrCa3P3`.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pp. 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL `https://aclanthology.org/2020.acl-main.703`.

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. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf`.

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

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://aclanthology.org/W04-1013`.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13853–13863, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Lin_Learning_To_Recognize_Procedural_Activities_With_Distant_Supervision_CVPR_2022_paper.pdf.

Junjia Liu, Yiting Chen, Zhipeng Dong, Shixiong Wang, Sylvain Calinon, Miao Li, and Fei Chen. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166, 2022. doi: 10.1109/LRA.2022.3153728.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 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*, pp. 2262–2272, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.173. URL https://aclanthology.org/2021.emnlp-main.173.

Ilya Loshchilov and Frank Hutter. 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, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=Bkg6RiCqY7.

Qing Lyu, Li Zhang, and Chris Callison-Burch. Goal-oriented script construction. In *Proceedings of the 14th International Conference on Natural Language Generation*, pp. 184–200, Aberdeen, Scotland, UK, August 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.inlg-1.19.

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

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

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. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Nils Reimers and Iryna Gurevych. 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)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5996, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00627. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00627.

Leonard Richardson. Beautiful soup documentation. *April*, 2007. URL https://beautiful-soup-4.readthedocs.io/en/latest/.

D. Ruth Anita Shirley, K. Ranjani, Gokulalakshmi Arunachalam, and D. A. Janeera. Automatic distributed gardening system using object recognition and visual servoing. In G. Ranganathan, Joy Chen, and Álvaro Rocha (eds.), *Inventive Communication and Computational Technologies*, pp. 359–369, Singapore, 2021. Springer Singapore. ISBN 978-981-15-7345-3.

Abhilasha Sancheti and Rachel Rudinger. What do large language models learn about scripts? In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pp. 1–11, Seattle, Washington, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.starsem-1.1. URL https://aclanthology.org/2022.starsem-1.1.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *Proceedings of Data Centric AI NeurIPS Workshop)*, 2021. URL https://datacentricai.org/neurips21/papers/159_CameraReady_Workshop_Submission_LAION_400M__Public_Dataset_with_CLIP_Filtered_400M_Image_Text_Pairs.pdf.

Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/papers/Sener_Zero-Shot_Anticipation_for_Instructional_Activities_ICCV_2019_paper.pdf.

Ayan Sengupta, Amit Kumar, Sourabh Kumar Bhattacharjee, and Suman Roy. Gated Transformer for Robust De-noised Sequence-to-Sequence Modelling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3645–3657, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.309. URL https://aclanthology.org/2021.findings-emnlp.309.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 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)*, pp. 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 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)*, pp. 5227–5237, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Sung_VL-Adapter_Parameter-Efficient_Transfer_Learning_for_Vision-and-Language_Tasks_CVPR_2022_paper.pdf.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6408–6417, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.520. URL https://aclanthology.org/2020.emnlp-main.520.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 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*, pp. 296–310, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.28. URL https://aclanthology.org/2021.naacl-main.28.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 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)*, pp. 4281–4294, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.295. URL https://aclanthology.org/2022.acl-long.295.

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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 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)*, pp. 4525–4542, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.310. URL https://aclanthology.org/2022.acl-long.310.

Yue Yang, Joongwon Kim, Artemis Panagopoulou, Mark Yatskar, and Chris Callison-Burch. Induce, edit, retrieve: Language grounded multimodal schema for instructional video retrieval. *Computer Vision and Pattern Recognition*, arXiv:2111.09276, 2021a. URL https://arxiv.org/pdf/2111.09276.pdf6.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. Visual goal-step inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2167–2179, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.165. URL https://aclanthology.org/2021.emnlp-main.165.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27263–27277. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Paper.pdf.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 23634–23651. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/c6d4eb15f1e84a36eff58eca3627c82e-Paper.pdf.

Li Zhang, Qing Lyu, and Chris Callison-Burch. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4630–4639, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.374. URL https://aclanthology.org/2020.emnlp-main.374.

He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G. Derpanis, Richard P. Wildes, and Allan D. Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2938–2948, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Zhao_P3IV_Probabilistic_Procedure_Planning_From_Instructional_Videos_With_Weak_Supervision_CVPR_2022_paper.pdf.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 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, 2018. ISBN 978-1-57735-800-8.

Shuyan Zhou, Li Zhang, Yue Yang, Qing Lyu, Pengcheng Yin, Chris Callison-Burch, and Graham Neubig. 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)*, pp. 2998–3012, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.214. URL https://aclanthology.org/2022.acl-long.214.

Yilun Zhou, Julie Shah, and Steven Schockaert. Learning household task knowledge from WikiHow descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pp. 50–56, Macau, China, August 2019. Association for Computational Linguistics. URL https://aclanthology.org/W19-5808.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pp. 1097–1100, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

A HYPERPARAMETERS

Our model is built based on the Huggingface framework (Wolf et al., 2020)[6]. We choose top 5 retrieved historically relevant steps as input for our retrieval model. For the gardening domain, we choose 5 negative samples for each step during contrastive learning. Specifically, 4 self-negative samples are randomly chosen from the title, method, and step history input, including steps and captions. The remaining 1 retrieved negative samples are randomly chosen from top-20 most similar

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

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. For the training objectives, we set τ as 1 for contrastive loss and λ as 0.5 based on validation performance. We optimize our model by AdamW (Loshchilov & Hutter, 2019) with Cosine Annealing Warm Restarts schedule (Loshchilov & 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 our results with a beam size of 5.

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 tokens. We only choose the closest 10 step-caption pairs. We use BLIP (Li et al., 2022) [7] 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 $\texttt{all} - \texttt{mpnet} - \texttt{base} - \texttt{v2}$ from SentenceBert (Reimers & Gurevych, 2019) which performs best in semantic search to retrieve similar steps.

Table 9: # of Model Parameters

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

We train out model with NVIDIA A6000 GPUs with 48G memory with full precision. We choose our best model based on 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 is 2.81 and ROUGE-L is 15.24 for the gardening domain and BLEU-4 is 4.85 and ROUGE-L is 20.25 for the crafts domain. The average training time for each model is 2 to 4 hours. Table 9 shows the number of the parameters for each model.

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

# 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

C Evaluation Metrics

We use BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Denkowski & Lavie, 2014) from Microsoft COCO Caption Evaluation package[8]. We use official implementation of BARTScore (Yuan et al., 2021)[9]. We use $\texttt{cross} - \texttt{encoder/stsb} - \texttt{roberta} - \texttt{large}$ 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: $\texttt{all} - \texttt{mpnet} - \texttt{base} - \texttt{v2}$ from SentenceBert (Reimers & Gurevych, 2019) for text retrieval, and

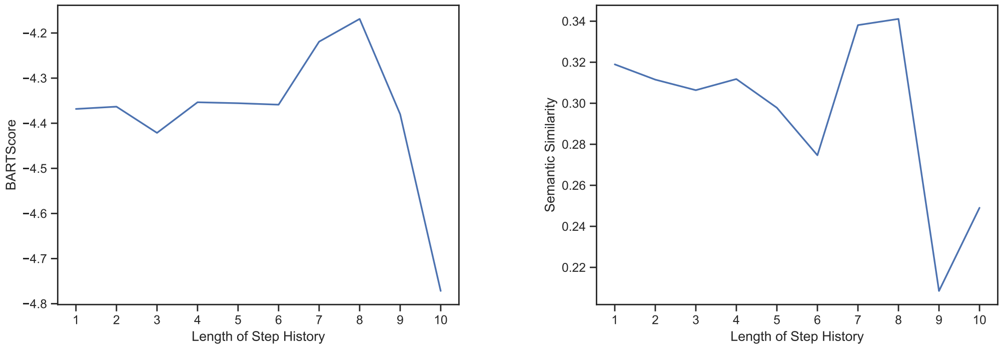
[7] The BLIP checkpoint we is https://storage.googleapis.com/sfr-vision-language-research/BLIP/models/model_base_capfilt_large.pth

[8] https://github.com/salaniz/pycocoevalcap

[9] https://github.com/neulab/BARTScore

the best language-image pretraining model `ViT − L/14@336px` from CLIP (Radford et al., 2021) for image retrieval.

D 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 length 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 even larger, 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 an clear decline in the number of instances because of our dataset construction strategy. Therefore, the model cannot generalize over long history input.

E 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.

Table 11: Krippendorff-α scores for human evaluation on with average ranking of next step correctness (N.), future steps correctness (F.), diversity (D.), executability (E.).

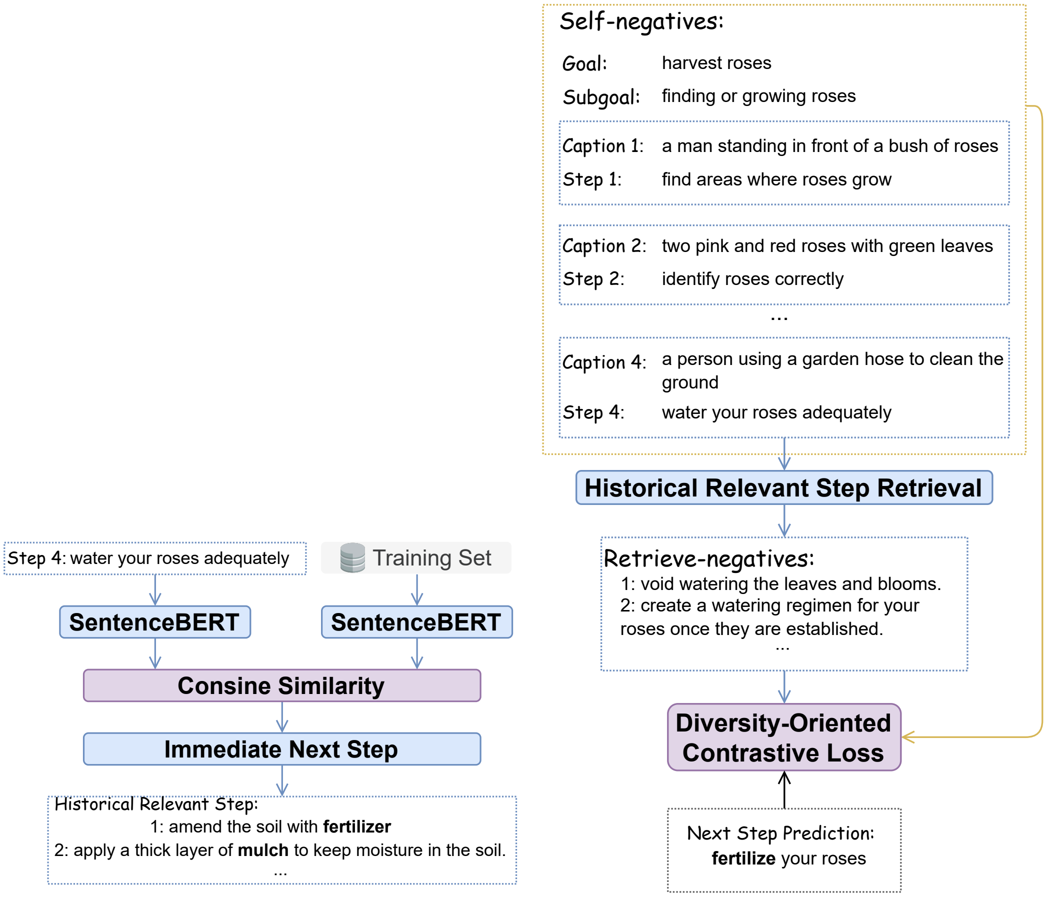
Model	**Gardening**				**Crafts**			
	N.	**F.**	**D.**	**E.**	**N.**	**F.**	**D.**	**E.**
BART	0.60	0.64	0.55	0.22	0.60	0.59	0.70	0.35
+CP	0.65	0.50	0.53	0.41	0.67	0.60	0.90	0.31
+CP+ME	0.70	0.74	0.86	0.31	0.45	0.40	0.76	0.41
+CP+ME+RD	0.53	0.50	0.68	0.37	0.62	0.46	0.78	0.31
+CP+ME+RD+CL	0.43	0.58	0.56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

F Human evaluation details

We measures inter-annotator agreement with Krippendorff-α scores (Krippendorff, 2018). The results are in Table 11.Table 12 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.

Table 12: Annotation examples

Type	**Content**
Instructions	(1) similarity to next step measures the correctness of generated results with 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 the these models output from 1(best)-5(worst), ties are allowed if both outputs are 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.


Step 4: water your roses adequately
Training Set
SentenceBERT
SentenceBERT
Consine Similarity
Immediate Next Step
Historical Relevant Step:
1: amend the soil with **fertilizer**
2: apply a thick layer of **mulch** to keep moisture in the soil.
...
Self-negatives:
Goal: harvest roses
Subgoal: finding or growing roses
Caption 1: a man standing in front of a bush of roses
Step 1: find areas where roses grow
Caption 2: two pink and red roses with green leaves
Step 2: identify roses correctly
...
Caption 4: a person using a garden hose to clean the ground
Step 4: water your roses adequately
Historical Relevant Step Retrieval
Retrieve-negatives:
1: void watering the leaves and blooms.
2: create a watering regimen for your roses once they are established.
...
Diversity-Oriented Contrastive Loss
Next Step Prediction:
fertilize your roses

(a) Details for historical relevant step retrieval

(b) Details for diversity-oriented contrastive loss

Figure 4: Additional model details

G Additional model architecture

Figure 4 shows additional details for our framework. The immediate next step refers to the step right after the previous given steps.

H Impact of Historical Relevant Steps

We analyze the relation between the quality of the retrieved historical relevant steps and the quality of the model predictions. The quality of retrieved steps and model predictions are evaluated by the semantic similarity score, which measures the embedding space similarity between a given text and the ground-truth next step. The Pearson's correlation between the semantic scores of historical relevant steps and the semantic scores of model predictions is 0.39 with a $p < 0.01$. We also illustrate their relation in Figure 5. The results suggest that the performance of our model is positively correlated with the relevance of the retrieved historical steps.

I Limitations

I.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 a case studies. Because we focus on multimedia

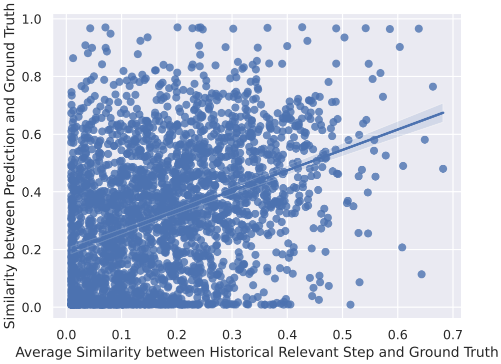


Figure 5: The semantic similarity scores (Thakur et al., 2021) between the model predictions and the ground truths versus the semantic similarity scores between the retrieved historical relevant steps and the ground truths in the gardening domain.

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. We plan to expand our model to other categories written in other languages.

I.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 description 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 thruth *"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 module and the sentence retrieval module. 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 compared to 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.

I.3 Limitations of Evaluation

The automatic metrics we chose, including BLEU Papineni et al. (2002), ROUGE Lin (2004), METEOR Denkowski & 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 metric are based on pretrained model including Augmented SBERT Thakur et al. (2021), SentenceBert Reimers & Gurevych (2019) and CLIP Radford et al. (2021). Those metrics might not align with human judgment and might be biased towards 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.

J Ethics and Broader Impact

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

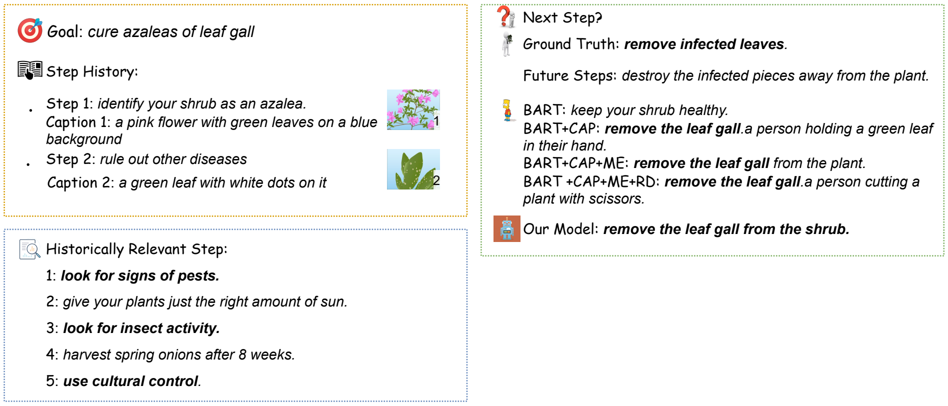
J.1 Usage Requirement

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.

J.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*[10]. 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 4.3, we perform human evaluation. All annotators involved in the human evaluation are voluntary participants and receive a fair wage.

K Sample Output



Goal: *cure azaleas of leaf gall*

Step History:

- Step 1: *identify your shrub as an azalea.*
  Caption 1: *a pink flower with green leaves on a blue background*
- Step 2: *rule out other diseases*
  Caption 2: *a green leaf with white dots on it*

Historically Relevant Step:

1: ***look for signs of pests.***

2: *give your plants just the right amount of sun.*

3: ***look for insect activity.***

4: *harvest spring onions after 8 weeks.*

5: ***use cultural control**.*

Next Step?

Ground Truth: ***remove infected leaves**.*

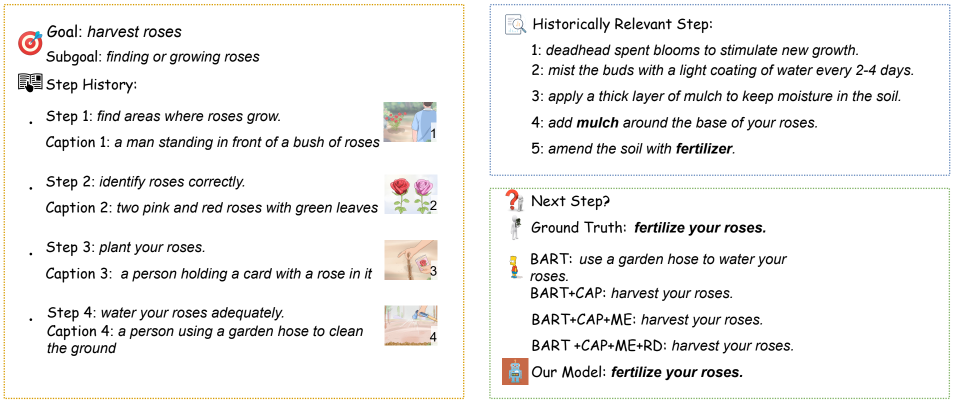
Future Steps: *destroy the infected pieces away from the plant.*

BART: *keep your shrub healthy.*
BART+CAP: ***remove the leaf gall**.a person holding a green leaf in their hand.*
BART+CAP+ME: ***remove the leaf gall** from the plant.*
BART +CAP+ME+RD: ***remove the leaf gall**.a person cutting a plant with scissors.*

Our Model: ***remove the leaf gall from the shrub.***

Figure 6: Human and System Step Prediction Results. It shows an example that our model benefits from selective multimedia encoder.

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



Goal: *harvest roses*

Subgoal: *finding or growing roses*

Step History:

- Step 1: *find areas where roses grow.*
  Caption 1: *a man standing in front of a bush of roses*
- Step 2: *identify roses correctly.*
  Caption 2: *two pink and red roses with green leaves*
- Step 3: *plant your roses.*
  Caption 3: *a person holding a card with a rose in it*
- Step 4: *water your roses adequately.*
  Caption 4: *a person using a garden hose to clean the ground*

Historically Relevant Step:

1: *deadhead spent blooms to stimulate new growth.*
2: *mist the buds with a light coating of water every 2-4 days.*
3: *apply a thick layer of mulch to keep moisture in the soil.*
4: *add* ***mulch*** *around the base of your roses.*
5: *amend the soil with* ***fertilizer****.*

Next Step?

Ground Truth: ***fertilize your roses.***

BART: *use a garden hose to water your roses.*

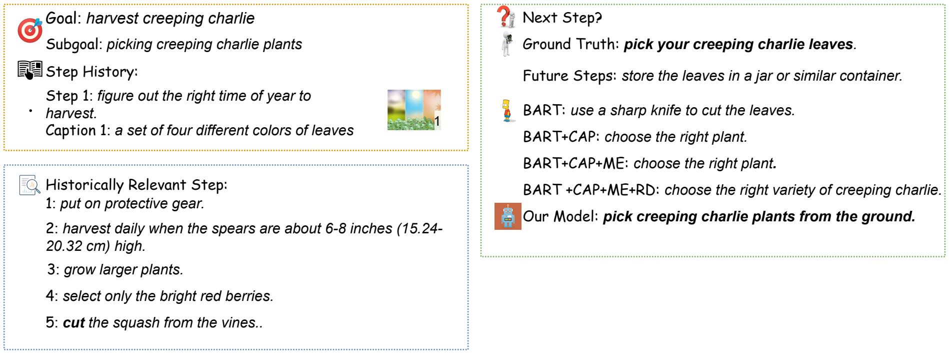
BART+CAP: *harvest your roses.*

BART+CAP+ME: *harvest your roses.*

BART +CAP+ME+RD: *harvest your roses.*

Our Model: ***fertilize your roses.***

Figure 7: Human and System Step Prediction Results. It shows an example that our model prediction results benefits from retrieval results and contrastive learning.



Goal: *harvest creeping charlie*

Subgoal: *picking creeping charlie plants*

Step History:

- Step 1: *figure out the right time of year to harvest.*
  Caption 1: *a set of four different colors of leaves*

Historically Relevant Step:

1: *put on protective gear.*
2: *harvest daily when the spears are about 6-8 inches (15.24-20.32 cm) high.*
3: *grow larger plants.*
4: *select only the bright red berries.*
5: ***cut*** *the squash from the vines..*

Next Step?

Ground Truth: ***pick your creeping charlie leaves****.*

Future Steps: *store the leaves in a jar or similar container.*

BART: *use a sharp knife to cut the leaves.*

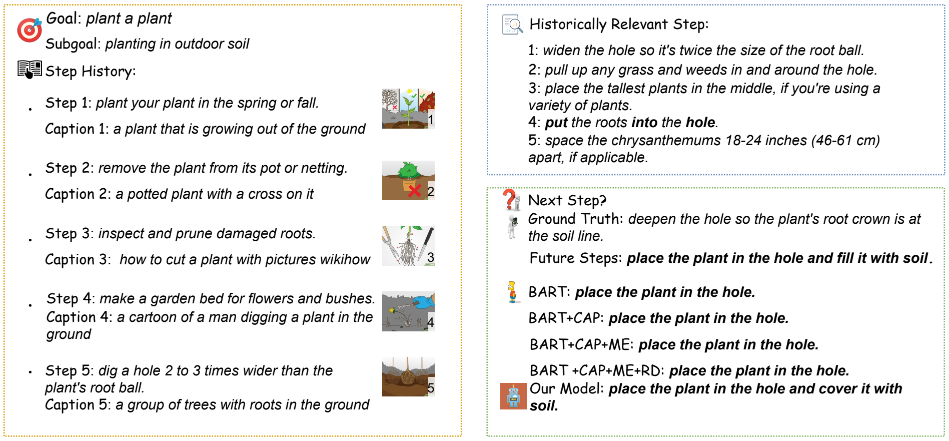
BART+CAP: *choose the right plant.*

BART+CAP+ME: *choose the right plant****.***

BART +CAP+ME+RD: *choose the right variety of creeping charlie.*

Our Model: ***pick creeping charlie plants from the ground.***

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



Goal: *plant a plant*

Subgoal: *planting in outdoor soil*

Step History:

- Step 1: *plant your plant in the spring or fall.*
  Caption 1: *a plant that is growing out of the ground*
- Step 2: *remove the plant from its pot or netting.*
  Caption 2: *a potted plant with a cross on it*
- Step 3: *inspect and prune damaged roots.*
  Caption 3: *how to cut a plant with pictures wikihow*
- Step 4: *make a garden bed for flowers and bushes.*
  Caption 4: *a cartoon of a man digging a plant in the ground*
- Step 5: *dig a hole 2 to 3 times wider than the plant's root ball.*
  Caption 5: *a group of trees with roots in the ground*

Historically Relevant Step:

1: *widen the hole so it's twice the size of the root ball.*
2: *pull up any grass and weeds in and around the hole.*
3: *place the tallest plants in the middle, if you're using a variety of plants.*
4: ***put*** *the roots* ***into*** *the* ***hole****.*
5: *space the chrysanthemums 18-24 inches (46-61 cm) apart, if applicable.*

Next Step?

Ground Truth: *deepen the hole so the plant's root crown is at the soil line.*

Future Steps: ***place the plant in the hole and fill it with soil*****.**

BART: ***place the plant in the hole.***

BART+CAP: ***place the plant in the hole.***

BART+CAP+ME: ***place the plant in the hole.***

BART +CAP+ME+RD: ***place the plant in the hole.***

Our Model: ***place the plant in the hole and cover it with soil.***

Figure 9: Human and System Step Prediction Results. It shows an example that our model prediction results matches future steps instead of immediate next step.