Understanding Multimodal Procedural Knowledge by Sequencing Multimodal Instructional Manuals

Anonymous ACL submission

Abstract

The ability to sequence unordered events is evidence of comprehension and reasoning about real world tasks/procedures, and is essential for applications such as task planning and multi-source instruction summarization. It often requires thorough understanding of temporal common sense and multimodal information, since these procedures are often conveyed by a combination of texts and images. While humans are capable of reasoning about and sequencing unordered procedural instructions, the extent to which the current machine learning methods possess such a capability is still an open question. In this work, we benchmark models' capability of reasoning over and sequencing *unordered multimodal* instructions by curating datasets from online instructional manuals and collecting comprehensive human annotations. We find current state-of-the-art models not only perform significantly worse than humans but also seem incapable of efficiently utilizing multimodal information. To improve machines' performance on multimodal event sequencing, we propose sequence-aware pretraining techniques exploiting the sequential alignment properties of both texts and images, resulting in >5% improvements on perfect match ratio.

1 Introduction

Instructions are essential sources for agents to learn how to complete complex tasks composed of multiple steps (e.g., "making a wood sign from scratch"). However, instructions do not always come in a proper sequential order, for example, when instructions must be combined across sources. Therefore, *sequencing* ***unordered*** *task-steps* is crucial for comprehending and inferring task procedures, which requires thorough understanding of event causal and temporal common sense. It is essential for applications such as multi-source instruction summarization and robot task planning (Garattoni and Birattari, 2018).

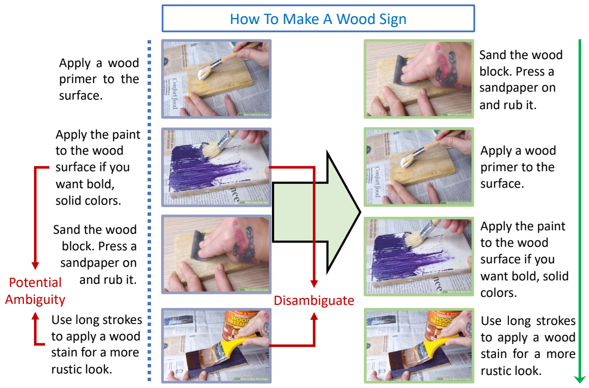


Figure 1: **Multimodal task procedure sequencing:** The left column shows unordered instruction steps from the manual *How To Make Wood Signs*. Each step is a text description and its associated image. Without the complementary information from the visuals, a novice may have difficulty inferring the proper task order. Considering multimodal information, the proper order can be correctly inferred (right column).

Existing work has studied sequencing unordered texts from paper abstracts or short stories (Chen et al., 2016; Cui et al., 2018). However, real-life tasks are often complex, and multimodal information is usually provided to supplement textual descriptions to avoid ambiguity or illustrate details that are hard to narrate, as illustrated in Figure 1.

To investigate whether current AI techniques can efficiently leverage multimodal information to sequence unordered task instructions, we curate two datasets from *online instructional manuals* (Hadley et al.; Yagcioglu et al., 2018). We consider two representative instruction domains: cooking recipes and "How-To" instructions (WikiHow). We establish human performance for the sequencing task on a subset of each data resource. As certain steps to perform a task can potentially be interchangeable[1], we collect annotations of possible **orders** alternative to the originally authored ones to create multiple references. Such additional annotation provides not only better measurement of human

[1] For example, without special requirements, preparing certain ingredients of a dish, such as slicing carrots or cucumbers, does not necessarily need to follow a specific order.

and model performance by alleviating unintended biases from content creators, but also a useful resource for future research of models that are aware of task-step dependencies and interchangeability.

To measure the ability of state-of-the-art AI techniques to sequence instruction steps, we construct models consisting of: (1) an **input encoder** which encodes image, text, or multimodal inputs, and (2) an **order decoder** which predicts step order using the encoded representations. They are jointly trained with the order supervisions.

Our preliminary studies show that multimodal information is consistently helpful for the sequencing task. However, compared to humans, current models are less efficient in utilizing multimodal information. We hypothesize that it is because the models do not effectively capture the sequential information in the vision modality as well as the sequential alignment between multimodal contents. To address this, we propose to equip models with capabilities of performing **multimodal grounding with sequential awareness**. Specifically, we propose several self-supervised objectives, including sequence-based masked language modeling, image region modeling, and content swapped prediction, to pretrain the models before finetune them on the downstream sequencing task.

The proposed pretraining techniques are shown to be effective in improving multimodal performance, enjoying a >5% improvement on the perfect match ratio metric. However, it is still significantly behind human performance ($\sim$ 15% in perfect match ratio metric). The same conclusion holds when alternative orders are considered.

Our key contributions are two-fold: (1) We propose a multimodal sequencing task with two curated instructional manuals, and collected comprehensive human annotations. (2) We investigate model performance on sequencing unordered manuals, and propose sequence-aware pretraining techniques to more effectively use the multimodal information. Our experiments and extensive analysis provide insights on which task categories are most challenging and reveal that more sophisticated sequential multimodal grounding can potentially further improve the performance on our task.

2 Problem Definition

Given a task procedure S consisting of N steps, where each step $S_i \in S$ can consist of two types of contents: a textual description T_i of tokens $\{T_{i,k}\}_{k=1}^{n_T}$ and/or image(s) $I_i = \{I_{i,k}\}_{k=1}^{n_I}$.[2] A model is required to take as inputs a random permutation of S, *i.e.* $S_p = \{S_{p_1}, ..., S_{p_N}\}$, where p is a permutation (S_{p_j} can take one of the following three modalities: T_{p_j}, I_{p_j}, and $\{T_{p_j}, I_{p_j}\}$), and predict the correct order of S_p, *i.e.* argsort(S_p).

3 Datasets and Human Annotation

We are interested in understanding the current state-of-the-art models' performance on this multimodal instruction sequencing task. To this end, we curate instruction datasets to support our study.

3.1 Instruction Manual Datasets

There are three major features we require for the target datasets: (1) It is multimodal. (2) It consists of task procedures as sequences of steps. (3) Different modalities are used intentionally to complement each other. In light of these, we consider the following two datasets:

RecipeQA. We start from a popular as well as intuitive choice of instruction manuals, recipes, which fully fulfill the aforementioned criteria. RecipeQA is a multimodal question answering dataset consists of recipes scraped from *Instructables.com* (Yagcioglu et al., 2018). We utilize the recipes collected in RecipeQA and convert each unique recipe into sequential multimodal steps for our task.

WikiHow. To expand the types of instruction manuals for our task beyond recipes, we also consider a popular "How To ..." type of instructions, WikiHow, which is an online knowledge base that consists of human-created articles describing procedures to accomplish a desired task. Each article contains a high level goal of a task, a short summary of the task procedures, and several *multimodal* steps where each step consists of a description paired with one or a few corresponding images.

We scrape the entire WikiHow knowledge resource, containing more than 100k unique articles (mostly) with multimodal contents , as well as the hierarchically structured category for each article. See Append. Sec. A for more dataset details.

3.2 Human Performance Benchmark

To ensure the validity of our proposed multimodal sequencing task, we establish the human performance via Amazon Mechanical Turk. Since our dataset is constructed from resources that are not

[2]For computational concerns, we set $n_I = 1$ in this work.

designed for the sequencing task, the quality of random samples is unverified. Specifically, some articles in WikiHow may not have a notion of proper order among the steps.[3] As a result, to construct a high quality test set particularly for WikiHow for establishing human performance, we first identify a set of categories which are more likely to feature proper order, *e.g*. *Home and Garden* and *Hobbies and Crafts*.[4] A random proportion is then sampled and the co-authors further downsample the subset to 300 samples with the aforementioned criteria via majority vote. For RecipeQA, we randomly sample 100 recipes from the dataset. And hence, the resulting two subsets will serve as our **golden-test-set** for the performance benchmarking.

Human Performance. Prompted with a task goal and a randomly scrambled sequence of the task-steps (can be one of the following modalities: multimodal or text/image-only), workers are asked to examine the contents and decide the proper performing order. Human performance are then computed against the original authored orders as the ground truths, averaged across the whole set.[5]

Alternative Orders. When performing a task, some steps can be interchangeable. To take this into consideration in our benchmark task, we also collect possible alternative orders to the original ones to create multiple references. For each instance in our golden-test-set, given the instruction steps sequenced in their original order, we ask workers to annotate alternative orders if the presented task-steps can be performed following a different order.[6] More details of the two human annotation tasks can be found in Append. Sec. B.

4 Models

To benchmark the proposed task, we construct models comprising: (1) an **encoder** which encodes multimodal or text/image-only inputs, and (2) an **order decoder** which utilizes the encoded representations to predict the orders. To help models capture **sequentiality in task-steps** better as well as adapt to our target task domains, we pretrain the encoders with several self-supervised objectives on the instructions before integrating them with the decoder.

4.1 Input Encoders

Text-Only Encoders. We use *RoBERTa* (Liu et al., 2019) for text-only inputs. Although the next-sentence prediction in BERT (Devlin et al., 2019) can potentially be exploited for sequencing, we empirically find that RoBERTa performs better.

Multimodal Encoders. We consider the following two V&L models mainly due to their easy adaptation to our proposed sequencing task:

VisualBERT (Li et al., 2019) grounds object detected image regions (*e.g*. by Faster-RCNN (Ren et al., 2016)) to language with a single transformer model (Vaswani et al., 2017). VisualBERT is pretrained with: (1) multimodal masked language modeling (MLM)[7], and (2) image-text matching prediction (ITM), where the image in an image-caption pair is randomly replaced with another one to create misalignment, and the model is required to predict whether the current pair is aligned.

CLIP-ViL (Shen et al., 2021) is also a single-stream V&L model similar to VisualBERT, while the visual encoder is replaced by a patch-based model inspired by the ViT (Dosovitskiy et al., 2021) in CLIP (Radford et al., 2021), where the image features are taken as *gridded-image-patches* as shown in Figure 2. The pretraining objectives remain the same as VisualBERT. Empirically, both Shen et al. (2021) and this work find such patch-based model tends to yield better downstream performance.

Image-Only Encoders. We attempt to provide an image-only baseline on our sequencing task with two visual encoders: (1) *ResNet*-based (He et al., 2016) Faster-RCNN model (also the visual encoder in VisualBERT) where both the detected regional features and the whole-image-feature are used, and (2) the aforementioned patch-based *CLIP* model.[8]

4.2 Sequence-Aware Pretraining

The standard multimodal grounding techniques (Li et al., 2019; Lu et al., 2019; Su et al., 2020; Chen et al., 2020a) do not explicitly concern the sequentiality of text and associated image sequences, and

[3] No temporal or other dependencies among the task-steps, *e.g*. "How to be a good person", where each step depicts a different aspect and tips of being a good person.

[4] Although the data used for training is not cleansed and thus can be noisy, we believe models can still learn to sequence from many of the articles designed to have proper order.

[5] We design an algorithm to compute the inter-annotator agreements (IAAs), see Append. Sec. B.3 for details. The IAAs for (*multimodal*, *text-only*, *image-only*) versions in WikiHow is: (0.84, 0.82, 0.69), and (0.92, 0.87, 0.81) in RecipeQA.

[6] The alternative order annotation IAAs for (*multimodal*, *text-only*, *image-only*) versions in WikiHow is: (0.73, 0.71, 0.78), and (0.79, 0.76, 0.79) in RecipeQA.

[7] RoBERTa is used to initialize VisualBERT and CLIP-ViL.

[8] Without confusion, throughout the paper we term the ViT- and CLIP-inspired visual encoder simply as CLIP.

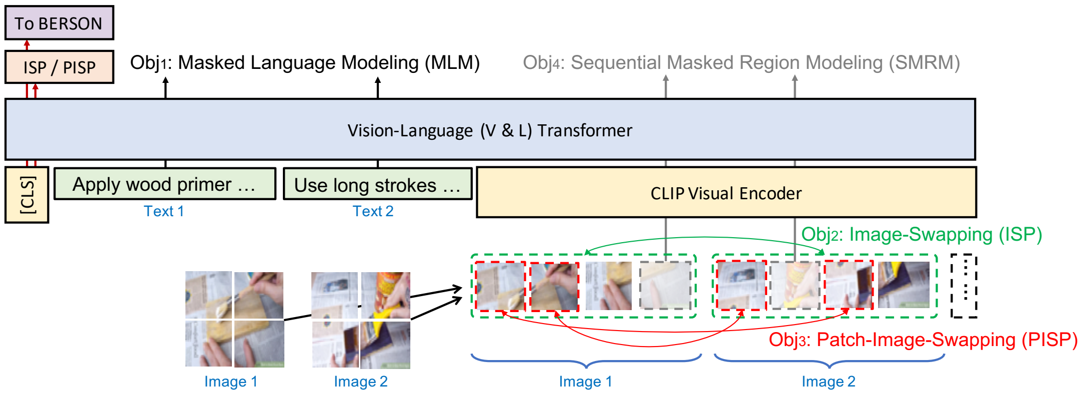


Figure 2: **Sequence-aware pretraining** includes: (1) masked language modeling (MLM), (2) image-swapping prediction (ISP/PISP) which requires the model to predict if some images (image-patches) are swapped, and (3) sequential masked region modeling (SMRM) where models are asked to reconstruct masked regions in each image within the input sequence.

hence may fall short of effectively utilizing the sequential properties in multimodal inputs. To encourage models to have better awareness of the sequential alignments in multimodal instruction steps, we propose to pretrain the encoders with the following self-supervised objectives: (1) masked language modeling (**MLM**), (2) (patch-based) image-swapping predictions (**ISP/PISP**), and (3) sequential masked region modeling (**SMRM**). Figure 2 illustrates an overview of the pretraining paradigm.

For the proposed objectives, the inputs to the models are generally *ordered* instruction step sequences, which can be further sub-sampled to produce length-varying subsequences. Although we do not find this necessarily benefit the downstream performance, it is observed that the sub-sampling helps the model converge faster. Without loss of generality and for simplicity, the following sections assume the sub-sampled sequence is of length 2.

4.2.1 Masked Language Modeling

The standard MLM (Devlin et al., 2019) is employed by the text-only models to adapt a pretrained language model to the target domain (task instructions). Following prior V&L works, we apply MLM to multimodal models. Specifically, we ensure that the textual description of each step T_i gets similar amount of maskings such that the models can potentially exploit the image sequences more.[9]

4.2.2 Swapping-Based Prediction

This objective concerns, with certain probability, randomly swapping a pair of items in a sequence and asking the model to judge whether the resulting sequence is properly ordered or not (*i.e.* binary classification). We mainly perform the swapping in the image modality and hence it can be viewed as a sequence-aware version of ITM objective in most V&L models. As in ITM, the output representation at the `[CLS]` token is used to make the prediction.

Standard. For an ordered sequence S, we can randomly swap two[10] items of S, $\{S_i, S_j\}$, where $i < j$, to $\{S_j, S_i\}$, with a certain probability δ. Our preliminary studies find that swapping the textual contents does not necessarily help the downstream performance for either text-only or multimodal models, so we only perform the swapping on the images $\{I_i, I_j\}$ in both multimodal and image-only models. For patch-based image inputs (or regional features), the whole patches of an image are swapped with those of another one within the same sequence, as illustrated in **Obj**$_2$ in Figure 2.

Patch-Based. We can perform the aforementioned swapping prediction with a finer granularity, directly on the image patches. Assuming each image I_i is cropped into w patches (or w detected regions), *i.e.* $\{\mathbf{i}_{i,k}\}_{k=1}^{w} = \{\mathbf{i}_{i,1}, ..., \mathbf{i}_{i,w}\}$, we randomly select M (ranging from 1 to w) number of patches each from the two images I_i, I_j (*i.e.* $\{\mathbf{i}_{i,p}\}, \{\mathbf{i}_{i,q}\}, p, q \in M$-sized sampled indices) to be swapped with probability δ. Specifically, for each image patch $\mathbf{i}_{i,m} \in I_i$, a randomly selected image patch $\mathbf{i}_{j,n} \in I_j$ is sampled to be swapped with. The sampled M-sized indices do not need to be the same set of integers for each image. The **Obj**$_3$ in Figure 2 illustrates the patch-based swapping prediction with $w = 4$ and $M = 2$.

4.2.3 Sequential Masked Region Modeling

Prior works extend the masked learning to the visual modality, where the masked target is either a predefined discrete visual vocabulary (Sun et al.,

[9]As higher chances that the complementary textual information is also masked out from different steps.

[10]Two is our minimum number for a valid subsequence.

2019; Bao et al., 2021) or (soft) object class labels (Lu et al., 2019; Su et al., 2020; Chen et al., 2020a). In this work, we construct a feature-based target vocabulary dynamically in each training mini-batch. We first randomly select same amount of $X\%$ ($X = 15$) patches for **each image** to be masked out (replaced with 0-tensor), and then construct a target vocabulary from the original output representations (before masking) of these patches.

Concretely, denote the output representation of an input image-patch $\mathbf{i}_{i,m}$ as $h(\mathbf{i})_{i,m}$ and the masked positions of I_i as D_i, we can construct a candidate list from all the output representations of the patches at the masked positions of each image, *i.e.* $C = \{h(\mathbf{i})_{i,m}\} \cup \{h(\mathbf{i})_{j,n}\}, m, n \in D_i, D_j$. Denote the masked image patches as $\textbf{mask(i)}_{i,m}$, for each output masked representation $h(\textbf{mask(i)})_{i,m}$, we concatenate it with all the candidates, *i.e.* $h(\textbf{mask(i)})_{i,m} || h(\textbf{i'}), \forall \textbf{i'} \in C$, which results in $|C|$ concatenated representations for each masked position. A $|C|$-way multi-class classification can then be performed by maximizing the probability of $p(\mathbf{i}_{i,m} | h(\textbf{mask(i)})_{i,m}; C)$. For robust training, we additionally: (1) shuffle the candidate set C for each masked position to prevent overfitting, and (2) ensure the overlapping of masked positions in each pair of images, $D_i \cap D_j$, is $< 50\%$, allowing the models to utilize information of similar regions from other images in the sequence.

4.2.4 Overall Training Objective

As the mechanism in some objectives cannot guarantee mutually exclusive impacts (*e.g.* performing ISP and PISP simultaneously may create confusing swapped patches), we employ a turn-taking fashion, with uniform probability, one of the objectives (**Obj**) is sampled for each training mini-batch. The overall pretraining objective is defined as below:

$$L = L_{\text{MLM}} + L_{\textbf{Obj}}, \textbf{Obj} \sim \{\text{ISP}, \text{PISP}, \text{SMRM}\} \quad (1)$$

4.3 Order Decoder – BERSON

BERSON is a recently proposed state-of-the-art neural sentence ordering framework (Cui et al., 2020), where a pointer network (Vinyals et al., 2016) exploits both the local (relative pairwise order) and global (self-attentions on top of the entire input sequence) information of the inputs to decode the predicted order. BERSON mainly exploits the `[CLS]` output representations for relational understanding, which aligns well with how our encoders are pretrained (Figure 2). We integrate our encoders (with or without sequence-aware pretraining) into BERSON, replacing its original BERT encoder. The BERSON-module-specific components are freshly initialized and then the entire integrated module is finetuned on our sequencing task.

5 Experiments and Analysis

Our experiments seek to answer these questions: (1) How valid is the proposed task for humans to complete? (2) Is multimodality helpful? (3) Can the proposed sequence-aware pretraining utilize multimodality more effectively? (4) How would results differ when alternative orders are considered?

5.1 Evaluation Metrics

We adopt metrics from sentence ordering works:

Position-Based metrics concern the correctness of the absolute position of each item in a sequence, including: (1) **Accuracy (Acc)** which computes the ratio of absolute positions in the ground truth order that are correctly predicted; (2) **Perfect Match Ratio (PMR)** which measures the percentage of predicted orders exactly matching the ground truth orders; and (3) **Distance (Dist.)** which measures the average distance[11] between the predicted and ground truth positions for each item.

Longest Common Subsequence computes the average longest subsequences in common (Gong et al., 2016) between the predicted and ground truth orders ($\mathbf{L}_q$). We also consider a stricter version, longest common substring, which requires the consecutiveness for the comparisons ($\mathbf{L}_r$).

Kendall's Tau (τ) (Lapata, 2003) is defined as $1 - 2 \times (\#\ inversions)/(\#\ pairs)$, where the inversion denotes that the predicted relative order of a pair of items is inverted compared to the corresponding ground truth relative order, and $\#\ pairs = \binom{N}{2}$ for N-length sequence.

Each metric focuses on different perspectives of the predictions, *i.e.* position metrics concern the absolute correctness, while common subsequence and τ metrics measure if general sequential tendency is preserved despite incorrect absolute positions.

5.2 Implementation Details

We use the original data splits for RecipeQA. For WikiHow, to prevent models' exploiting knowledge from similar articles, we split the data so that certain (sub)categories do not overlap in each split.

[11] Except for distance metric, higher scores are better.

Modality	Encoders	Pretrain	WikiHow Golden-Test-Set						RecipeQA Golden-Test-Set					
			Acc↑	PMR↑	L_q ↑	L_r ↑	τ ↑	Dist↓	Acc↑	PMR↑	L_q ↑	L_r ↑	τ ↑	Dist↓
Image-Only	ResNet	N	21.73	2.00	2.81	1.73	0.01	7.87	31.20	5.00	3.27	2.07	0.27	6.10
	CLIP	N	24.92	3.33	2.95	1.84	0.08	7.32	38.40	8.00	3.39	2.02	0.35	5.44
	CLIP	Y	28.24	5.00	3.09	1.96	0.16	6.80	47.20	16.00	3.68	2.40	0.52	4.12
	Human Performance		68.16	47.49	4.27	3.51	0.72	2.43	80.40	64.50	4.54	4.02	0.86	1.29
Text-Only	RoBERTa	N	74.75	56.67	4.47	3.78	0.82	1.71	74.00	52.00	4.45	3.68	0.83	1.64
	RoBERTa	Y	75.68	58.67	4.50	3.87	0.82	1.69	77.00	57.00	4.49	3.81	0.84	1.48
	Human Performance		83.35	66.91	4.63	4.11	0.89	1.06	88.92	78.56	4.76	4.41	0.93	0.70
Multimodal	VisualBERT	N	75.30	57.33	4.45	3.83	0.81	1.65	76.20	58.00	4.49	3.85	0.83	1.58
	VisualBERT	Y	77.30	59.67	4.50	3.86	0.83	1.58	78.20	60.00	4.56	3.91	0.85	1.44
	CLIP-ViL	N	76.15	59.00	4.49	3.87	0.82	1.68	79.20	60.00	4.57	3.93	0.85	1.29
	CLIP-ViL	Y	**79.87**	**65.67**	**4.57**	**4.05**	**0.85**	**1.44**	**82.60**	**68.00**	**4.61**	**4.10**	**0.88**	**1.10**
	Human Performance		91.03	79.61	4.78	4.46	0.94	0.52	92.12	83.13	4.82	4.53	0.95	0.45

Table 1: **Golden-test-set performance:** Models which take multimodal inputs (for both VisualBERT and CLIP-ViL encoders) consistently outperform the ones that only take unimodal inputs. Our proposed sequence-aware pretraining is shown consistently helpful throughout the three modality variants. Humans show larger performance gain when both modalities of inputs are provided, and are more robust to the local ordering as implied by the smaller gaps between L_q and L_r.

Modality	Pretrain	WikiHow Golden-Test-Set						RecipeQA Golden-Test-Set					
		Acc↑	PMR↑	L_q ↑	L_r ↑	τ ↑	Dist↓	Acc↑	PMR↑	L_q ↑	L_r ↑	τ ↑	Dist↓
Image-Only	ISP	27.31	4.00	3.02	1.82	0.12	7.00	43.20	9.00	3.49	2.05	0.47	4.46
	ISP + PISP	27.57	4.67	3.07	1.93	0.16	6.85	43.40	12.00	3.57	2.24	0.48	4.46
Multimodal	MLM	77.08	61.33	4.52	3.96	0.83	1.65	79.60	61.00	4.55	3.93	0.86	1.29
	MLM + ISP	77.61	62.00	4.54	3.97	0.83	1.60	80.00	61.00	4.56	3.93	0.86	1.26
	MLM + SMRM	77.94	62.33	4.54	3.98	0.84	1.60	80.00	59.00	4.53	3.89	0.87	1.26
	MLM + ISP + PISP	78.14	63.33	4.55	4.03	0.84	1.56	80.80	63.00	4.57	3.99	0.87	1.24
	MLM + ISP + SMRM	79.47	63.67	4.57	4.03	0.85	1.54	81.40	63.00	4.57	4.00	0.87	1.20

Table 2: **Model ablation studies:** We provide a performance breakdown for incremental combinations of the pretraining objectives, ablated on the best performing models (CLIP and CLIP-ViL) from Table 1 for each dataset and modality.

Details are in Append. Sec. A. Preliminary studies show that joint training with both RecipeQA and WikiHow data does not necessarily improve the downstream performance, thus the models evaluated in the two datasets are trained simply using their respective training sets for faster convergence.

We cap the overall sequence length at 5 and each step description with maximally 5 sentences for both models and humans. The maximum input length per step is 60 tokens (overall maximum length $= 300$) for training and GPU memory efficiency. $\delta = 0.5$ for both ISP and PISP. All images are resized to 224×224, and 32×32 patch is used for CLIP-based models, resulting in $7 \times 7 = 49$ patches per image. Aside from standard positional embedding, we only supplement a modality token type embedding (text:=0, image:=1) to the multimodal models. Pretrained weights for each encoder is obtained either from their corresponding code bases or by running their codes on our setup.[12]

5.3 Standard Benchmark Results

Table 1 summarizes both the human and model performance for each input modality evaluated using the original ground truth orders on the golden-test-set, whereas Table 2 summarizes a more detailed breakdown of the model performance when incrementing combinations of pretraining objectives.

As is shown, multimodal information is verified consistently helpful for humans. Compared under same scenario with or without the sequence-aware pretraining, the two multimodal models consistently outperform their text-only counterparts, where the proposed pretraining technique is shown particularly effective for the patch-based multimodal model (CLIP-ViL). However, our top-performing models still exhibit significant gaps below human performance, especially in PMR.

Additionally, we observe a different trend in the two datasets where the multimodality benefits more in RecipeQA than WikiHow. The gap between the multimodal human and model performance is larger than the text-only counterparts in WikiHow, while a reversed trend is shown in RecipeQA. We hypothesize that recipes may contain less common language usages and/or words for the pretrained language models and hence benefits more from the pretraining. Humans, on the other hand, benefit more from the images in WikiHow as its texts are hypothesized to contain more ambiguities.

WikiHow Category Analysis. We are interested in which categories of WikiHow our models perform closer to humans, as well as in which the multimodality is most efficiently utilized. In Figure

[12] We initialize CLIP-ViL with our pretrained CLIP.

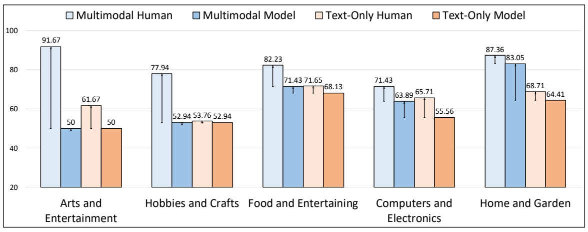


Figure 3: **Top-3 and least-2 categories of human-model performance difference (in PMR):** The selected categories have >10 samples. The difference bars on the multimodal model series are compared against the text-only model series.

3 we select categories with the top and least performance gaps (with PMR metric, top=3, least=2) between the human and our best performing models. We observe that the categories of which multimodal model outperforms text-only one the most are also the categories the models perform closest to humans, *e.g*. *Home and Garden*. We hypothesize that the images in these categories are well complementary to the texts and that our sequence-aware grounding performs effectively. In contrast, in categories such as *Arts and Entertainment* and *Hobbies and Crafts* where humans still enjoy benefits from multimodality, our models have difficulty utilizing the multimodal information. We hypothesize that better visual understanding may alleviate the potentially suboptimal grounding as images of these categories can contain many non-common objects.

5.4 Evaluating with Alternative Orders

For each instance where alternative ground truth orders exist, the performance is computed by the best each predicted order can obtain against all the ground truth orders[13], denoted by **multi-reference performance**, and the subset containing these instances is denoted as the **multi-reference subset**.[14]

Multi-Reference Performance. The noticeable main competitors in Table 1 are multimodal and text-only models, and hence for conciseness in Table 3 we mainly report the multi-reference version of their best performing variants with the selected metrics. Several trends still hold: (1) Multimodal models still outperform the text-only counterparts. (2) Human performance is still well above models' even under multi-reference setups. Additionally, both humans and models perform significantly worse in the multi-reference subset when single (original) ground truth is enforced, implying the validity of our alternative order annotations.

[13]Jointly considered from all the evaluation metrics.

[14]The overall average number of ground truth references becomes 1.19, 1.23, 1.09 for multimodal, text-only, and image-only versions in WikiHow; and 1.10, 1.17, 1.14 in RecipeQA.

We originally question that whether enforcing the original authored order to be the only ground truth can cause unfairness to text-only models, as images can often better represent the detailed scene changes omitted by the texts, while in reality certain steps may not need to strictly follow the authored order. Judging from the number of instances that improve after evaluating with alternative orders, the text-only model indeed benefits more from the multi-reference setup. Examining the general trends in Table 3, one can conclude that the textual contents indeed posses certain levels of ambiguities where images can help to alleviate, however, as the performance gaps between multimodal and text-only models are still significant under the multi-reference settings, enforcing the original order as the only ground truth should not be the major reason justifying advantages of multimodality.

WikiHow Categories. Table 4 lists the WikiHow categories with the most (top-5) annotated multi-reference ground truths. Note that the categories with more annotated alternative ground truths are also among the worse performance from both humans and models (Figure 3). We provide sample qualitative inspections in Append. Sec. C.1.

6 Related Work

Sequence Ordering. Story sequencing test is a popular way of examining children's abilities on sequential reasoning which is shown evident for procedural understanding (Tomkins, 1952; Baron-Cohen et al., 1986; Loucks et al., 2017). In NLP, existing works attempt the sequencing task as sorting a series of unordered sentences (Chen et al., 2016; Cui et al., 2018; Logeswaran et al., 2018; Oh et al., 2019; Lee et al., 2020; Calizzano et al., 2021) from paper abstracts or short paragraphs. While certain prior work also attempts to extend it to incorporate multimodality (Agrawal et al., 2016), the dataset used, Visual StoryTelling (Huang et al., 2016), features album images that were not intended to be procedural nor supply unstated details to complement the texts. In computer vision, existing work leverages shuffle frame prediction for learning video representations (Lee et al., 2017; Xu et al., 2019; Wang et al., 2020; Li et al., 2020) as well as cycle consistency constraints for learning temporal dynamics (Epstein et al., 2021). (Zellers et al., 2021) also features a pairwise relative frame re-ordering objective to learn temporal common sense from scripted videos, however, as their downstream

Modality	Subset	WikiHow Golden-Test-Set (Size: 300)						RecipeQA Golden-Test-Set (Size: 100)					
		Acc↑		PMR↑		L_r ↑		Acc↑		PMR↑		L_r ↑	
		single	multi	single	multi	single	multi	single	multi	single	multi	single	multi
Text-Only	Single	77.30	—	61.75	—	3.98	—	79.32	—	60.23	—	3.90	—
	Multi.	67.35	80.00	40.82	59.18	3.35	3.86	60.00	75.00	33.33	58.33	3.17	3.92
		(% of instances benefit w. multi-reference: 34.7%)						(% of instances benefit w. multi-reference: 50.0%)					
	All	75.68	77.74	58.67	61.67	3.87	3.96	77.00	78.80	57.00	60.00	3.81	3.90
	Single†	85.57	—	71.41	—	4.24	—	90.27	—	80.41	—	4.47	—
	Multi.†	72.03	85.51	43.84	71.38	3.46	4.14	79.00	87.00	65.00	80.00	3.95	4.40
		(% of instances benefit w. multi-reference: 42.9%)						(% of instances benefit w. multi-reference: 41.6%)					
	All†	83.35	85.56	66.91	71.40	4.11	4.22	88.92	89.88	78.56	80.36	4.41	4.46
Multimodal	Single	81.68	—	69.90	—	4.15	—	83.71	—	69.07	—	4.12	—
	Multi.	70.98	78.82	47.05	61.22	3.59	3.90	46.67	60.00	33.33	33.33	3.67	3.78
		(% of instances benefit w. multi-reference: 21.6%)						(% of instances benefit w. multi-reference: 66.6%)					
	All	79.87	81.19	65.67	68.00	4.05	4.11	82.60	83.00	68.00	68.00	4.10	4.11
	Single†	92.86	—	83.67	—	4.56	—	91.88	—	82.61	—	4.52	—
	Multi.†	82.09	92.22	59.80	83.33	3.99	4.54	100.00	100.00	100.00	100.00	5.00	5.00
		(% of instances benefit w. multi-reference: 41.18%)						(% of instances benefit w. multi-reference: 0.0%)					
	All†	91.03	92.75	79.61	83.61	4.46	4.55	92.12	92.12	83.13	83.13	4.53	4.53

∗ The size of the **Multi.** subsets in (*text-only*, *multimodal*) are: (49, 51)/300 in WikiHow and (12, 3)/100 in RecipeQA.

Table 3: **Multi-reference performance:** († denotes human performance) Our golden-test-set can be decomposed into two subsets: **Single** where each instance in this subset only has one single originally authored ground truth, and **Multi.** where each instance features multiple ground truths from alternative orders. For the **Multi.** subset, two types of performance can be computed: **single** considers only the originally authored ground truth and **multi** computes the multi-reference performance. **All** denotes the entire test-set combining the results from **Single** and **Multi.** subsets. Results are reported on the two main competitors: multimodal and text-only using the best performing models from Table 1 in each modality. **% of instances benefit w. multi-reference** indicates that of what percentage of instances *in each multi-reference subset* humans and the models benefit (for each instance if its performance improves *in any of the metrics*) from alternative ground truth orders.

Categories	Mean Per-Instance Refs. (Cnt)		
	Multimodal	Text	Image
Home and Garden	2.00 (7)	2.14 (7)	2.00 (3)
Hobbies and Crafts	2.00 (5)	2.73 (11)	2.00 (2)
Food and Entertaining	2.20 (15)	2.22 (14)	2.17 (12)
Others	2.28 (7)	2.67 (5)	2.00 (4)
Personal Care and Style	2.33 (3)	2.00 (1)	2.00 (1)

Table 4: **Top-5 mean alternative orders by categories:** We list top-5 categories in WikiHow according to the number of average ground truth references in their multi-reference subset. We again only list the categories with total instance count >10.

tasks mainly concern visual reasoning and ordering by frame-text-matching (also on Visual Story-Telling), the re-ordering objective is more focused on the visual modality. Our work takes a different perspective to tackle a comprehensive multimodal sequencing task with a focus on the procedural task-solving knowledge and gauging the helpfulness of complementary information in different modalities.

Task/Procedure Understanding. Other works have utilized WikiHow for learning task knowledge. In NLP, textual descriptions of WikiHow have been used for abstractive summarization (Koupaee and Wang, 2018), procedural understanding (Zhou et al., 2019; Tandon et al., 2020), and intent estimation (Zhang et al., 2020a). Prior work (Zhang et al., 2020b) considers WikiHow for learning event temporal ordering, but limited to only pairwise relations. A concurrent work uses WikiHow to infer visual goals (Yang et al., 2021). We hope our curation can help advancing the goal of comprehensive multimodal procedural understanding.

Multimodality. Beside models used in this work, there are several recent advanced multimodal grounding techniques (Tan and Bansal, 2019; Li et al., 2019; Lu et al., 2019; Su et al., 2020; Chen et al., 2020b; Huang et al., 2020; Wen et al., 2021). We utilize VisualBERT and CLIP-ViL for their simplicity to be adapted to our task and easier integration to our proposed pretraining techniques, however, our framework is able to incorporate any of the aforementioned multimodal models.

7 Conclusions

In this work we present studies of language and multimodal models on procedure sequencing, leveraging popular online instructional manuals. Our experiments show that both multimodality and our proposed sequence-aware pretraining are helpful for multimodal sequencing, however, the results also highlight significant gaps below human performance ($\sim$ 15% on PMR).

We provide insights as well as resources, such as the multi-reference annotations of the sequencing task, to spur future relevant research. We also anticipate that the alternative orders defined and annotated in our work can benefit more comprehensive task-procedure understanding. Future work such as predicting task steps which can be parallel or interchangeable, and understanding step dependencies can be explored.

8 Ethics and Broader Impacts

We hereby acknowledge that all of the co-authors of this work are aware of the provided *ACM Code of Ethics* and honor the code of conduct. This work is mainly about sequencing a given series of multimodal task procedures, represented by text descriptions along with their images. The followings give the aspects of both our ethical considerations and our potential impacts to the community.

Dataset. We collect the human performance on our sequencing task (both the standard human performance and the alternative order annotations) via Amazon Mechanical Turk (MTurk) and ensure that all the personal information of the workers involved (e.g., usernames, emails, urls, demographic information, etc.) is discarded in our dataset. While the sequence orders either from the original author intended ones or those annotated by the workers for the standard performance may possess unintended biases against certain population group of people (*e.g*. due to cultural differences or educational differences, some tasks may be performed differently from the original intended orders), we anticipate the additional multi-reference annotation can alleviate such an issue as well as provide a broader view to approach procedural understanding, *i.e*. certain task-steps can be interchanged.

This research has been reviewed by the **IRB board** and granted the status of an **IRB exempt**. The detailed annotation process (pay per amount of work, guidelines) is included in the appendix; and overall, we ensure our pay per task is above the the annotator's local minimum wage (approximately $12 USD / Hour). We primarily consider English speaking regions for our annotations as the task requires certain level of English proficiency.

Techniques. We benchmark the proposed sequencing task with the state-of-the-art large-scale pretrained language and multimodal models with our novel sequence-aware pretraining techniques. As commonsense and task procedure understanding are of our main focus, we do not anticipate production of harmful outputs, especially towards vulnerable populations, after training models on our proposed task.

References

Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. 2016. Sort story: Sorting jumbled images and captions into stories. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hangbo Bao, Li Dong, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Simon Baron-Cohen, Alan M Leslie, and Uta Frith. 1986. Mechanical, behavioural and intentional understanding of picture stories in autistic children. In *British Journal of developmental psychology*, volume 4, pages 113–125. Wiley Online Library.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2020. Multimodal pretraining unmasked: Unifying the vision and language BERTs. *arXiv preprint arXiv:2011.15124*.

Rémi Calizzano, Malte Ostendorff, and Georg Rehm. 2021. Ordering sentences and paragraphs with pretrained encoder-decoder transformers and pointer ensembles. In *Proceedings of the 21st ACM Symposium on Document Engineering*, pages 1–9.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Neural sentence ordering. *arXiv preprint arXiv:1607.06952*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020a. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. Uniter: Learning universal image-text representations. In *European Conference on Computer Vision (ECCV)*.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep attentive sentence ordering network. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4340–4349.

Baiyun Cui, Yingming Li, and Zhongfei Zhang. 2020. BERT-enhanced relational sentence ordering network. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 6310–6320. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4171–4186.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Dave Epstein, Jiajun Wu, Cordelia Schmid, and Chen Sun. 2021. Learning temporal dynamics from cycles in narrated video. In *International Conference on Computer Vision (ICCV)*.

Lorenzo Garattoni and Mauro Birattari. 2018. Autonomous task sequencing in a robot swarm. In *Science Robotics*, volume 3. Science Robotics.

Jingjing Gong, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953*.

Chris Hadley, Katiana Uyemura, Kyle Hall, Kira Jan, Sean Volavong, and Natalie Harrington. Wikihow.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ting-Hao K. Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Aishwarya Agrawal, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Association for Computational Linguistics (ACL)*, pages 545–552.

Haejun Lee, Drew A Hudson, Kangwook Lee, and Christopher D Manning. 2020. Slm: Learning a discourse language representation with sentence unshuffling. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2017. Unsupervised representation learning by sorting sequences. In *International Conference on Computer Vision (ICCV)*, pages 667–676.

Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Jeff Loucks, Christina Mutschler, and Andrew N Meltzoff. 2017. Children's representation and imitation of events: How goal organization influences 3-year-old children's memory for action sequences. In *Cognitive Science*, volume 41, pages 1904–1933. Wiley Online Library.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neural Information Processing Systems (NeurIPS)*, pages 13–23.

Byungkook Oh, Seungmin Seo, Cheolheon Shin, Eunju Jo, and Kyong-Ho Lee. 2019. Topic-guided coherence modeling for sentence ordering by preserving global and local information. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2273–2283.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: Towards real-time object detection with region proposal networks. 39(6):1137–1149.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations (ICLR)*.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

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 *Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417.

Silvan S Tomkins. 1952. The tomkins-horn picture arrangement test. In *Transactions of the New York Academy of Sciences*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations (ICLR)*.

J Wang, B Hu, Y Long, and Y Guan. 2020. Order matters: Shuffling sequence generation for video prediction. In *30th British Machine Vision Conference 2019, BMVC 2019*. Newcastle University.

Keyu Wen, Jin Xia, Yuanyuan Huang, Linyang Li, Jiayan Xu, and Jie Shao. 2021. Cookie: Contrastive cross-modal knowledge sharing pre-training for vision-language representation. In *International Conference on Computer Vision (ICCV)*, pages 2208–2217.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10334–10343.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. Visual goal-step inference using wikihow. *arXiv preprint arXiv:2104.05845*.

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 *Neural Information Processing Systems (NeurIPS)*.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020a. Intent detection with WikiHow. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333, Suzhou, China. Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639.

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.

A Details of Datasets

A.1 Image Contents

For simplicity and computational concerns, in this work we only pair one image to each of its associated task-step textual descriptions. However, in both WikiHow and RecipeQA, each task-step can have more than one associated images or visual contents represented by short clips or GIFs. We simply select the first image, which is supposed to be the most representative, for those step featuring multiple images; and sample the frame in the middle of time interval for clips or GIFs. Nevertheless, our framework does not assume any limitation on how many images per step to be processed.

A.2 WikiHow Categories

The category in WikiHow generally forms a hierarchical directed acyclic graph. Each category can have its relevant subcategory, which usually spans finer-granularity of category types. For example, a possible category traversal path is: *Cars and Vehicles* $\rightarrow$*Public Transport* $\rightarrow$*Air Travel*, which can lead to the article *How to Overcome the Fear of Flying*. We attach these full category traversal paths as an additional feature to each of the article in our dataset, and we also will provide a complete list of the taxonomy composed by all the categories and subcategories in WikiHow. We include the category-data counts in Table 5 for a reference, where we only show the top-level category here. The more in-depth categories can be referred to in the full released version of the dataset.

Categories	Counts
Arts and Entertainment	4675
Cars and Other Vehicles	2044
Computers and Electronics	15023
Education and Communications	7406
Family Life	1747
Finance and Business	6228
Food and Entertaining	7670
Health	8800
Hobbies and Crafts	9217
Holidays and Traditions	736
Home and Garden	9460
Personal Care and Style	6523
Pets and Animals	5281
Philosophy and Religion	828
Relationships	2877
Sports and Fitness	3271
Travel	746
Work World	1579
Youth	2389
Others	21

Table 5: **Top-Level Categories of WikiHow:** Number of unique articles in each top-level category of the WikiHow dataset. The categories are sorted by alphabetical order. In total there are 19 top-level categories (same as what this page indicates: https://www.wikihow.com/Special:CategoryListing), and one "others" category for standalone leaf nodes without real linkages to these top-level categories.

A.3 Train-Dev Splits

For RecipeQA we use the original data splits which ensure no identical recipe appears in more than one set (each recipe has its unique recipe-id), as this dataset only has one category and the data quality is much more uniform than that of WikiHow, *i.e.* most recipes fulfill our target dataset criteria.

For WikiHow, we split the data according to the third level category to prevent models from exploiting too similar task knowledge in the same category, where the level (three) is empirically decided. Specifically, we ensure that the third-level categories where the articles in our golden-test-set belong to, do not appear in the train set. We first split the WikiHow dataset into train, development, and test set following this strategy, and then construct our golden-test-set by sub-sampling a subset of this (larger) test set followed by manual inspections, to ensure its quality. And then, we simply join the remaining test set samples to the development set. Table 6 presents the more detailed essential statistics of the two datasets, WikiHow in Table 6a, and RecipeQA in Table 6b.

B Details of Human Annotation

B.1 Golden-Test-Set Selections

In order to construct a high-quality test set for humans to evaluate, we manually select the samples which meet our general criteria: (1) the tasks are procedural in both texts and images (2) the task's images are designed to complement the textual descriptions or provide a more illustrative information for some unstated implicit knowledge. We ask three of our internal members (co-authors) to perform such manual selection, and preserve ones that have majority votes. In total, we select 300 samples for WikiHow and 100 samples for RecipeQA.

B.2 General Annotation Procedure

B.2.1 Standard Performance Benchmark

We collect the human performance via Amazon Mechanical Turk (MTurk). Each MTurk worker is required to read the provided instruction carefully, as shown in Figure 5a, and then perform the task,

Type	Counts			
Total Unique Articles	109486			
Total Unique Images	1521909			
Train / Dev / Golden-Test	98268 / 11218 / 300			
Type-Token Ratio	216434 / 82396591 = 0.0026			
Type	**Mean**	**Std**	**Min**	**Max**
Tokens in a Step Text	52.95	26.25	0	5339
Sentences in a Step Text	3.36	1.3	0	50
Number of Steps of a Task	5.27	2.62	0	75

(a) WikiHow

Type	Counts			
Total Unique Articles	10063			
Total Unique Images	87840			
Train / Dev / Golden-Test	8032 / 2031 / 100			
Type-Token Ratio	91443 / 5324859 = 0.017			
Type	**Mean**	**Std**	**Min**	**Max**
Tokens in a Step Text	82.08	84.72	0	998
Sentences in a Step Text	4.19	4.22	0	73
Number of Steps of a Task	6.45	2.57	4	20

(b) RecipeQA

Table 6: **General statistics of the two datasets**: We provide the detailed component counts of the datasets used in this work, including the statistics of tokens and sentences from the instruction steps (lower half of the two tables).

which is designed to be done in an intuitive *drag-n-drop* (illustrated in Figure 5b) fashion.

Each MTurk HIT is designed to have five sets of sequencing tasks followed by a few additional questions such as confidence level of the worker when inferring the order, and whether different modalities are helpful in a particular task. For each unique sample in the selected golden-test-set, we construct three annotation sets each for one modality version: multimodal, text-only, and image-only. We launch the HITs containing the same sample but with different modalities with a week gap to prevent potential memorization if the same worker happens to annotate the exactly identical data sample. We estimate the time required to complete each of our HITs to be 10-15 minutes, and adjust our pay rate accordingly to $2 or $3 USD depending on the length of the task. This roughly equates to a $12 to $15 USD per hour wage, which is above the local minimum wage for the workers. In total we receive annotated HITs from around 80 workers for WikiHow, and 14 workers for RecipeQA.

In order to ensure annotation quality and filter potential MTurk spammers, we design a few sets to be our *qualification rounds* for later on worker pool selection. The Pearson correlation between the performance of the qualification samples and the overall HIT performance is 0.6 with p-value < 0.05. Since it is positive correlated and significant, we censor assignments with substantially low overall performance (<20% on accuracy metric), and re-launch the HITs containing those samples for a few more rounds for higher quality annotations.

Finally, since the agreement is sufficiently high (see Section 3.2), we simply compute the human performance using all of the collected annotated orders from all the participated workers, which result in reasonably high human performance upper bound for our proposed sequencing task.

B.2.2 Annotating Alternative Orders

We deliberately ask a different set of MTurk workers than those participated in the standard performance benchmark round for annotating the alternative orders. In total we receive HITs from around 70 workers for WikiHow, and 40 workers for RecipeQA. The monetary rewards and other general settings follow the same procedure as in the standard performance collection. We compute pairwise IAAs for each worker against every other workers, using the method described in Append. Sec. B.3, and then we place a threshold to filter out workers that tend to have too low IAAs (which is a likely indicator that a worker is either a spammer or not understanding our task well). As the final IAAs among the selected pool of workers are sufficiently high (see Section 3.2), for each instance we perform a majority vote on the annotated alternative orders to serve as the final multi-references.

B.3 Inter-Annotator Agreements (IAA)

B.3.1 Standard Performance

As orders concern not only positioning of the items but also more complicated relative information among the items in a sequence, we propose to measure the agreements among orders centering around the concept of **pairwise relationship**. Specifically, we transform an integer sequence order to an one-hot encoded representation of the $\binom{N}{2}$ pairs of relative relations. Consider an example: suppose three items (`1, 2, 3`) are to be ordered, and all the pairwise relations are {`12, 13, 21, 23, 31, 32`}. The transformed one-hot representation is defined as: R_{123} = {`12: 1, 13: 1, 21: 0, 23: 1, 31: 0, 32: 0`} = {110100}, *i.e.* , $R(ij) = 1$ iff ij is a valid relatively ordered pair. Similarly, R_{231} = {001110}.

Using the aforementioned definition of R, we can compute Cohen's Kappa inter-annotator agreement score for a pair of annotated order per each instance. The overall scores can be computed by firstly taking the average of pairwise Kappa scores of annotations for each instance, and then taking the average across the entire dataset.

B.3.2 Alternative Orders

To evaluate the agreements for the alternative orders, we focus on the *differences* between an order and the ground truth in their transformed representations. We first compute the *one-hot difference* between an alternative order to the ground truth order, *e.g*. suppose ground truth order is simply $o_g = \texttt{123}$, and an alternative order is $o_1 = \texttt{132}$, then $R^{diff}_{o_g,o_1}$ = $abs|\{110100\}$ - $\{110001\}|$ = $\{000101\}$. To focus on the agreements of *the differences to the original ground truth*, we apply the Kappa score on a pair of orders by retaining the union of the positions where each order differ from the ground truth in their one-hot representations. For example, if $o_2 = \texttt{213}$, then $R^{diff}_{o_g,o_2}$ = $abs|\{110100\}$ - $\{011100\}|$ = $\{101000\}$, and hence the differences to the ground truth are at positions $4, 6$ from o_1 and $1, 3$ from o_2, *i.e*. the union is $\{1, 3, 4, 6\}$. Computing the Kappa scores on $R^{diff}_{o_g,o_1}$ and $R^{diff}_{o_g,o_2}$ at these positions leads to computing the scores on lists $\{0011\}$ and $\{0110\}$.

To compute the agreements of two series of alternative orders from two annotators (the series can have different lengths), we first iteratively find all the best matching pair of orders from the two series (each order in a series can only be matched **once**). When one series contain more orders than the other, the remaining unmatched orders will be compared to the ground truth to serve as the penalty. For a particular instance, we take the mean of all the Kappa scores (the best-matching-pair and penalty scores) as the IAA for the two annotators, as detailed in Algorithm 1. The overall IAA is computed similarly to the standard case.

Annotation Statistics. Table 7 lists the essential statistics of the multi-reference subsets.

B.4 Additional Statistics

Apart from the main sequencing task, we also ask the annotators for their confidence of predictions and if multimodality is helpful for deciding the order in the standard benchmark round. We hereby provide two more statistics obtained from the workers: the percentages of confidence levels and which modality (modalities) helps for deciding the order.

Modality Helps. As which modality is potentially more helpful, we include the percentages of each answer category in Table 8. It can be noticed that majority of workers (> 60%) think that multimodal (both modalities) is helpful, and especially in the recipe data, there are > 90% of workers indicating the effectiveness of utilizing multimodal inputs.

Confidence Levels. As shown in Table 9, majority of workers feel at least fairly confident (score of 4) about their predictions, which can justify the validity of our selection of golden-test-set.

Algorithm 1 Alternative Order IAA Per Instance

Require: $\{A_n\}_{n=1}^{N}$: A list of annotation series, where $A_n = \{a_{n,k}\}_{k=1}^{K_n}$ denotes K_n orders annotated by nth worker for an instance.
Require: $f(x, y)$: IAA scoring function.

Initialize S: empty score list
for i = 1 to N **do**
  for $j = i + 1$ to N **do**
    One-hot encode $\{a_{i,k}\}$, and $\{a_{j,k}\}$
    Assume $K_i < K_j$ *// otherwise swap*
    while $\{a_{i,k}\}$ not empty **do**
      Find best match according to R^{diff}
      $\hat{m}, \hat{n} = \arg\max_{m,n} f(R^{diff}_{o_g,o_{i,m}}, R^{diff}_{o_g,o_{j,n}})$
      $\{a_{i,k}\}$.pop($\hat{m}$); $\{a_{j,k}\}$.pop($\hat{n}$)
      $S = S \cup$ score
    end while
    while $\{a_{j,k}\}$ not empty **do**
      $S = S \cup f(o_g, o_{j,m})$; $\{a_{j,k}\}$.pop(m)
    end while
  end for
end for
return mean(S)

Modality	WikiHow (300) Cnt	Min/Max	Avg/Std	RecipeQA (100) Cnt	Min/Max	Avg/Std
Image-Only	24	2/4	2.1/1.4	13	2/3	2.1/0.3
Text-Only	49	2/6	2.4/0.9	12	2/6	2.4/1.1
Multimodal	51	2/4	2.1/0.5	3	2/6	4/1.6

Table 7: **Multi-reference subset statistics:** We report the count of multi-reference instances in each dataset across the three modalities, and their basic statistics.

Dataset	Both	Text-Only	Image-Only	Neither
RecipeQA	90.4	1.0	8.6	0.0
WikiHow	62.9	33.7	2.4	1.0

Table 8: **Which modality helps?** We compute the percentage of each answer category. In both datasets, majority of the annotations indicate that both modality are helpful for deciding the orders.

Confidence Level	WikiHow	RecipeQA
5 (Very)	54.61	64.75
4 (Fairly)	27.38	23.00
3 (Moderately)	12.24	7.00
2 (Somewhat)	5.21	4.75
1 (Not-At-All)	0.56	0.50

Table 9: **Confidence Level Statistics (%):** In both datasets, majority (> 80%) of the annotators indicate at least > 4 (fairly) confidence level, which can help justify the validity of the human performance.

C Additional Results

C.1 Qualitative Inspections

Figure 4 shows a few qualitative examples in different categories. Figure 4a shows that while step 1 and 3 may seem confusing if only looking at the texts, the images can help deciding the proper order, whereas models may fail to grasp such multimodal information in Figure 4b. In Figure 4c we show an example where multi-reference benefits both humans and the models, although in reality it should be more commonsensical to *stir* before *refrigerating* the mixtures.

C.2 Image-Only Multi-References

We also provide the detailed multi-reference performance break down on the image-only modality using the best performing models in Table 1, CLIP, in Table 10 for references.

D More Model Details

Multimodal Model Considerations. Bugliarello et al. (2020) suggests that many V&L models can achieve similar downstream performance if well trained, and thus we consider the models presented in this work, VisualBERT and CLIP-ViL, due to their simplicity of adapting to our sequencing task, as well as their main differences being how the visual inputs are encoded (via standard object detector networks or patch-based models like CLIP), which suits our proposed objectives well.

Swapping-Based Predictions. In Section 4.2.2 we mention that we do not observe necessary improvements when swapping the textual contents. Our hypothesis is that the pairwise loss function applied in the BERSON module already takes care of this especially for the textual contents. And that the stronger discourse-level hints inherent in the textual descriptions may make this operation unnecessary. On the other hand, both image and multimodal alignment does not share this similar property with the texts, and hence this reasons why swapping the visual modality suffices this particularly pretraining objective.

D.1 Training & Implementation Details

Training Details. All the models in this work are trained on a single Nvidia A100 GPU[15] on a Ubuntu 20.04.2 operating system. The hyperparameters for each model are manually tuned against different datasets, and the checkpoints used for testing are selected by the best performing ones on the held-out development set, which is constructed using the method described in Append. Sec. A.3.

Implementation Details. The implementations of the transformer-based models are extended from the HuggingFace[16] code base (Wolf et al., 2020), and our entire code-base is implemented in PyTorch.[17] The computer vision detector model used in one of our image-only encoders, ResNet-based Faster-RCNN (Ren et al., 2016), adopts the detectron2 open sourced module, and their pretrained weights are obtained from the official implementation from Facebook AI Research.[18] Implementation of BERSON modules are adapted from the original author's implementation, where more details can be found in their paper. Implementation of the VisualBERT is obtained from the *MMF*[19] framework from Facebook AI Research, and CLIP-ViL model is obtained and adapted from the original author's released code repository.[20] We use this same repository for the image-only encoder CLIP.

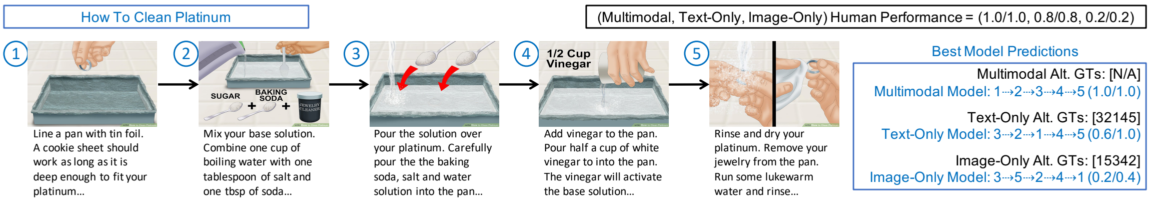
D.2 Hyperparameters

For the sequencing task, we train all the models for 5 or 10 (for multimodal models) epochs for all the model variants, where the training time varies from 2-4 hours for the text-only models and 6-8 hours for the multimodal models. We list all the hyperparameters used in Table 11. We also include the search bounds and number of trials in Table 12, that all of our models adopt the same search bounds and ranges of trials.

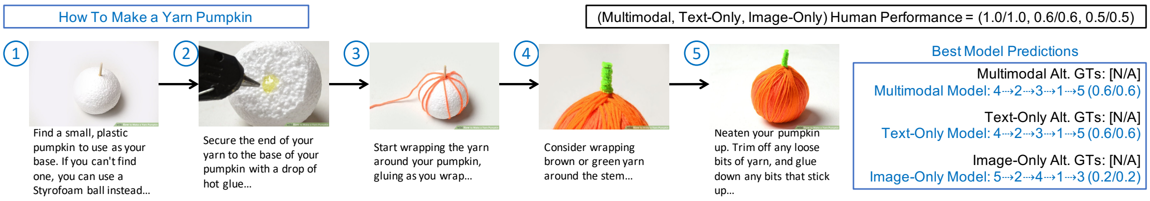
D.3 WikiHow Images

Although the images in WikiHow can often be synthetic or "cartoon-ish", we observe that modern

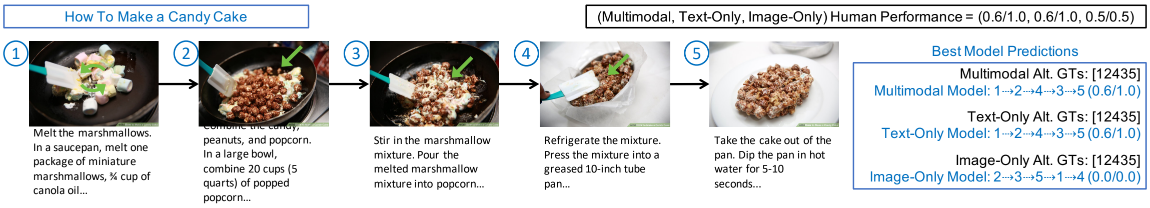
[15] https://www.nvidia.com/en-us/data-center/a100/
[16] https://github.com/huggingface/transformers
[17] https://pytorch.org/
[18] https://github.com/facebookresearch/detectron2
[19] https://github.com/facebookresearch/mmf
[20] https://github.com/clip-vil/CLIP-ViL



(a) Home and Garden Sample



(b) Hobbies and Crafts Sample



(c) Recipe Sample

Figure 4: **Qualitative examples:** We show some qualitative samples of our dataset associated with human and model predictions, and the annotated multi-reference ground truths. The texts are truncated to fit into the box shown in each sample. The performance are: (single-reference, multi-reference) **accuracy** metric respectively.

Modality	Subset	WikiHow Golden-Test-Set (Size: 300)						RecipeQA Golden-Test-Set (Size: 100)					
		Acc↑		PMR↑		L_r ↑		Acc↑		PMR↑		L_r ↑	
		single	multi	single	multi	single	multi	single	multi	single	multi	single	multi
Image-Only	Single	28.38	—	5.07	—	1.97	—	49.89	—	17.24	—	2.47	—
	Multi.	26.67	39.17	4.17	8.33	1.83	1.92	29.23	40.00	7.69	7.69	1.92	2.31
	All	28.24	29.24	5.00	5.33	1.96	1.97	47.2	48.60	16.00	16.00	2.40	2.45
	Single†	68.47	—	48.36	—	3.54	—	81.61	—	66.67	—	4.10	—
	Multi.†	64.58	75.83	37.50	56.25	3.19	3.71	72.31	79.23	50.00	61.54	3.50	3.88
	All†	68.16	69.06	47.49	48.99	3.51	3.55	80.40	81.30	64.50	66.00	4.02	4.07

∗ The size of the **Multi.** subsets are: 24/300 in WikiHow and 13/100 in RecipeQA.

Table 10: **Multi-reference performance on image-only modality:** † denotes human performance. The denotations are same as the Table 3. Results are reported using the best performing image-only models from Table 1.

object detectors can still propose meaningful regions, regardless of whether the object class prediction is sensible or not. We include some predicted bounding boxes in Figure 6 for references. And hence, although there may be concerns on suboptimal visual understanding from these images, we do believe both of our ResNet and CLIP visual encoders can extract reasonably useful features.

E Releases & Codes

The scraped WikiHow dataset will be released upon acceptance, along with a clearly stated documentation for usages. We will also release the code for processing the RecipeQA dataset particularly for our procedure sequencing task, where the original dataset can be obtained from their project website.[21] If permitted by the authors of the BERSON model, we will also release the cleaned code repository which encompasses the majority of the implementations in this work upon acceptance. We hope that by sharing the datasets and their essential tools, more interest could be drawn into research on multimodal procedure understanding and its future research directions.

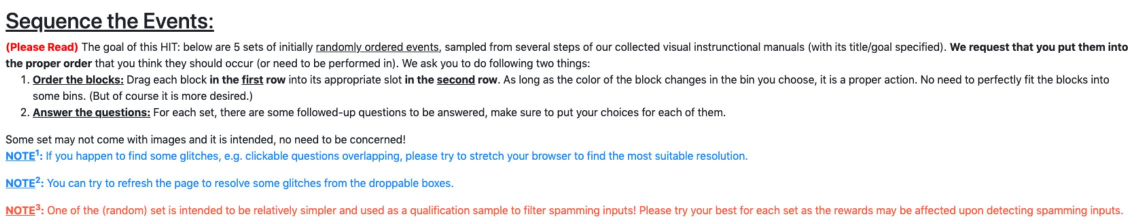
[21] https://hucvl.github.io/recipeqa/

Modalities	Models	Batch Size	Initial LR	# Training Epochs	Gradient Accumulation Steps	# Params
Image-Only	ResNet	4	5×10^{-6}	5	1	112.98M
	CLIP	4	5×10^{-6}	5	1	88.08M
Text-Only	RoBERTa	4	5×10^{-6}	5	1	393.16M
Multimodal	VisualBERT	4	5×10^{-6}	10	1	421.32M
	CLIP-ViL	4	5×10^{-6}	10	1	497.40M
Image-Only Pretrain	CLIP	4	1×10^{-5}	5	1	68.09M
Text-Only Pretrain	RoBERTa	4	1×10^{-5}	5	1	355.36M
Multimodal Pretrain	VisualBERT	4	1×10^{-5}	5	1	383.52M
	CLIP-ViL	4	1×10^{-5}	5	1	465.50M

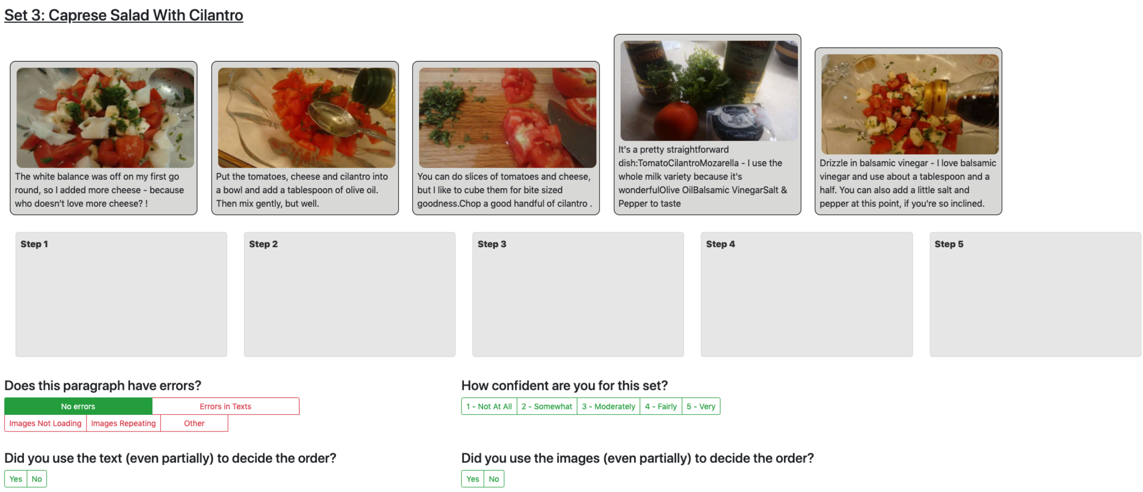
Table 11: **Hyperparameters in this work:** *Initial LR* denotes the initial learning rate. All the models are trained with Adam optimizers (Kingma and Ba, 2015). We include number of learnable parameters of each model in the column of *# params*.

Type	Batch Size	Initial LR	# Training Epochs	Gradient Accumulation Steps
Bound (lower–upper)	2–8	1×10^{-5}–1×10^{-6}	3–10	1–2
Number of Trials	2–4	2–3	2–4	1–2

Table 12: **Search bounds** for the hyperparameters of all the models.

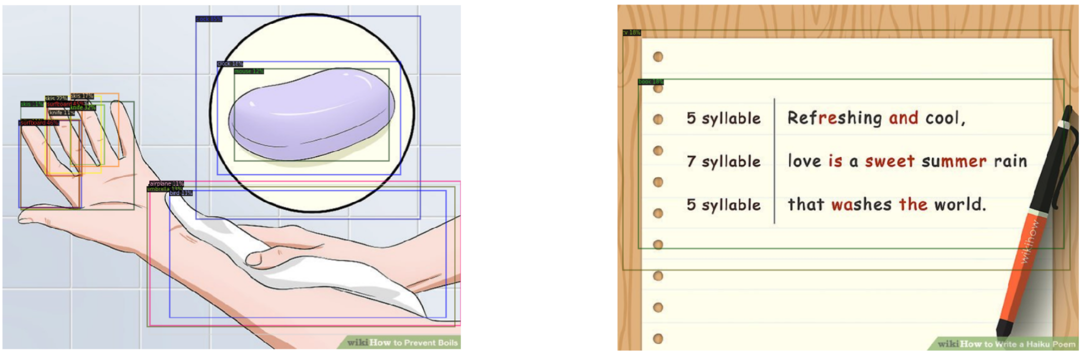


(a) Human Annotation Instruction



(b) Sample Annotation Interface

Figure 5: **MTurk Annotation User Interface: (a)** We ask the annotator to follow the indicated instruction, and perform the sequencing task. **(b)** The annotation task is designed for an intuitive *drag-and-drop* usage, followed by a few additional questions such as confidence level and whether each modality helps. (This example is obtained from RecipeQA dataset.)



(a) Detected Image Regions 1

(b) Detected Image Regions 2

Figure 6: **Proposed image regions by Detectron2:** We show some examples that even these synthetic and cartoon-ish images in the WikiHow dataset can provide meaningful representations which can be utilized by strong pretrained object detection modules. We show few top-detected objects with their bounding boxes and predicted classes. Note that while the classes may be wrongly predicted, the proposed regions are all meaningful.