11-777 Spring 2021 Class Project
ALFRED Background and Related Works

Christian Deverall* **Jingyuan Li** * **Artidoro Pagnoni** *
{cdeveral, jingyua4, apagnoni}@andrew.cmu.edu

1 Baselines

We present below the two baselines that we plan to evaluate against our proposed approach along with the metrics that will be used for the evaluation.

Sequence-to-Sequence with Progress Monitor Along with collecting the ALFRED dataset, Shridhar et al. (2020) proposed a baseline model for the task. They model the agent with a CNN-LSTM sequence-to-sequence architecture. A bidirectional-LSTM is used to generate a representation of the language instructions. A CNN encodes the visual input at each time step. And a decoder LSTM generates a sequence of low-level actions while attending over the encoded language and visual input. The model is trained to produce the actions and associated interaction masks from expert trajectories using imitation learning. The interaction mask is produced by a three layer deconvolution network trained with binary cross-entropy loss on ground-truth object segmentations.

The ALFRED task requires long action sequences to be completed. Shridhar et al. (2020) propose to use two auxiliary losses to monitor the progress of the agent towards the completion of the task. The first loss involves predicting the current process (a scalar between 0 and 1) using the decoder hidden state, the visual input, the attended language instructions, and the previous action. In addition to the global progress, the model is trained to predict the number of sub-goals completed so far.

Modular Object-Centric Approach (MOCA) The Modular Object-Centric Approach (MOCA) (Singh et al., 2020) decomposes the interactive instruction following procedure into action, and policy generation tasks. Two modularized networks, named action policy module (APM) and visual perception module (VPM), deal with the action planning and visual localization tasks respectively via attentive filtering visual and language information. The method also proposed to use an obstruction detection module to address the difficulties in being stuck around the obstacles. The method successfully gains improvements from previous methods and is of state-of-the-art position.

For training MOCA, the objective function is composed of the loss for APM and VPM. To guide the discovery of objects, the basic objective functions are two cross-entropy losses that enforce the model to discover objects and the corresponding action classes for each step. Two loss functions helping to achieve subgoals, and monitor the overall progress.

2 Metrics

Task Success To evaluate whether tasks have been successfully completed, the authors define two categories for success. For the task success metric, a score of 1 is given to a tasks trial when all of the goal conditions for that task have been met. If at least one goal condition is not met at the end of the sequence, a score of 0 is given. The main limitation of this metric is that the state-of-the-art models still perform poorly on the dataset, thus the percentage of tasks that meet every goal condition is very low.

Goal-Condition Success To overcome this limitation, goal-condition success measures the proportion of goal-conditions that have been met for the task. For example, if the completion of a task requires 5 goal-conditions and only 2 are met during testing, a score of 0.4 will be given. In general, tasks with longer sequences of instructions have a lower goal-condition success as the directive is more complex. As a result, it is important to use the same test splits across experiments to keep the

*Everyone Contributed Equally – Alphabetical order

	Seen		Unseen	
	Task (PW)	Goal-Cond (PW)	Task (PW)	Goal-Cond (PW)
Seq2Seq + PM	-	-	-	-
MOCA	-	-	-	-

Table 1: Results Table

task length constant.

Path Weighted Metrics A major limitation of both metrics is that one could attain a relatively high score by performing a long sequence of random acts. To overcome this, the authors perform path weighting (PW), which gives a lower score to action sequences that are long compared to the expert demonstration. The weighting follows the below equation where p_s is the path weighted score, s is the non-path weighted score, L^* is the length of the expert demonstration sequence and $\hat{L}$ is the length of the predicted sequence.

$$p_s = s * \frac{L^*}{max(L^*, \hat{L})}$$

Sub-Goal Evaluation The last metric is sub-goal evaluation. Initially, the model is forced to follow the beginning part of the expert demonstration. Afterwards, the model must predict actions to complete the remaining sub-goals given the full textual instruction and the most recent image input. This metric allows analysis into how previous action sequences affect the prediction of future action sequences.

3 Empty Results

Please refer to Table 1.

References

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. 2020. Moca: A modular object-centric approach for interactive instruction following. *arXiv preprint arXiv:2012.03208*.