11-777 Spring 2021 Class Project

Andrew Singh[*] **Ankit Ramchandani**[*] **Vashisth Parekh**[*]
{andrewsi, aramchan, vparekh}@andrew.cmu.edu

Abstract

Template for 11-777 Reports using the ACL 2021 Style File

1 Introduction and Problem Definition (1-1.25 pages)

Thesis statement or Hypothesis we are aiming to prove
"Our approach is better is not a hypothesis"

[*] Everyone Contributed Equally – Alphabetical order

2 Related Work and Background (1-1.5 pages)

Literature 1

Literature 2

Literature 3

Literature 4

3 Task Setup and Data

3.1 Task Definition

We plan to work with the ALFRED dataset (Shridhar et al., 2020) with the goal of learning a set of actions in an indoor household setting which will help an agent complete a task described by natural language. The tasks require navigation and interaction with multiple objects in the scene. Each interaction action requires a pixel-wise interaction mask to specify the object of interest. The agent receives high-level and low-level natural language instructions at the beginning of the episode, and can use egocentric visual observation (i.e. access to current RGB image, depth map, and instance segmentation map) at each time step as input. The agent produces one or two outputs at each time step: the current action to take, and, if the action involves interaction, the interaction mask of an object of interest.

We intend to predict the interaction mask pixel-wise instead of using any other coarser representations like bounding boxes. We also intend to use inputs in their rawest representation (e.g. raw image data instead of extracted features) for maximum generality and flexibility of downstream methods. Furthermore, we plan to develop a method to solve the full task of navigation and interaction in the ALFRED dataset. We clarify this to convey that we are not working with a small sub-task or a sub-problem of the dataset. Since current methods (Corona et al., 2020; Singh et al., 2020; Shridhar et al., 2020) struggle with generalization to novel objects and environments, we will attempt to primarily focus on improving generalization performance, which is measured by an "unseen" split of the test set which contains new environments and objects.

3.2 Dataset Statistics

In this section, we present a subset of the analysis we performed. **We encourage the reader to see the Jupyter notebook stored in the *Analysis* folder for full list of figures pertaining to the analysis since this report only includes a subset.**

3.2.1 Metadata Analysis

In this section, we analyze various metadata in ALFRED. Table 1 shows average values for the quantitative metrics we chose to measure on ALFRED. We can see that the averages are fairly consistent across splits. The navigation-interaction ratio indicates that for every interaction action in a demonstration, there are roughly 9 navigation actions. The mask coverage indicates that on average, the ground-truth interaction mask covers a rather small (15-17%) proportion of the image. The step-object coverage of nearly 1 indicates that for almost all interactions, the name of the object of interest is mentioned in the corresponding language directive. By manually inspecting examples with low interaction step coverage, we find that the object's name are usually substituted with a synonym (e.g."rag" for "cloth" and "scoop" for "ladle").

	Train	**Valid (seen)**	**Valid (unseen)**
Steps per directive	6.68	6.64	6.27
Tokens per step	12.39	12.18	12.63
Task desc. tokens	10.02	10.09	10.04
Images	286.75	287.24	277.72
Actions	49.78	50.12	46.98
Images per action	6.08	6.02	6.12
Actions per step	7.6	7.72	7.72
Nav-interact ratio	9.19	9.25	8.13
Total objects	33.2	32.84	38.44
Mask coverage	0.17	0.17	0.15
Step-object coverage	0.86	0.85	0.88

Table 1: Average values of various quantitative aspects of ALFRED by split. See appendix for definitions.

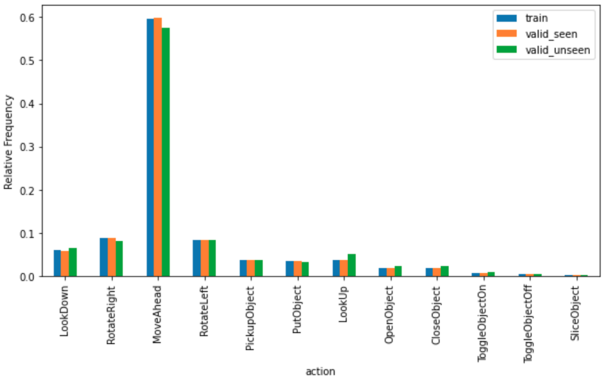


Figure 1: Relative frequency of each action type by split

Furthermore, Figure 1 shows the frequency of the 12 different actions (5 navigation actions + 7 interaction actions). Note that all splits have fairly equal frequency for all actions. Also, note that 60% of actions are "move ahead" actions which signifies the importance of a navigation module we may need in our model.

3.2.2 Textual Analysis

For textual analysis, we first identified out of vocabulary (OOV) words in the training and validation set using a vocabulary of 685k words defined by spaCy. We found that less than 0.002% of all words were OOV in any split, indicating the dataset is already quite clean. Most of the OOV words were just misspelled (eg: ”stovve”), indicating that it would be important to preprocess text using a simple spell checker before using it for downstream tasks.

We also found the top few synonyms used to describe objects in the dataset. This was performed by comparing the similarity of word vectors of all objects in the dataset with all common nouns identified in all task descriptions. The complete results are in the Jupyter notebook, but a few results are shown in Table 2. This reveals that our model will need to be robust enough to recognize synonyms of different words in order to be successful.

Furthermore, we analyzed how many objects present in the scene are directly referred to in the step-by-step instructions. The results are plotted in a histogram in Figure 2, which reveals that much less than 40% of objects in the scene are actually referred to in the instructions.

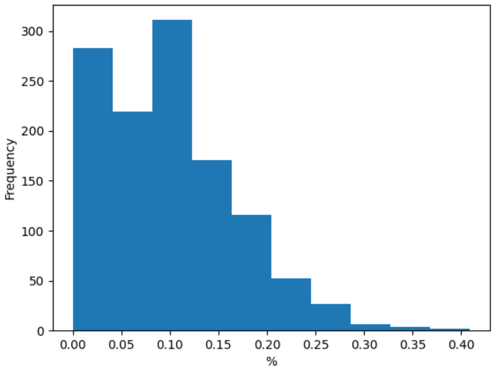


Figure 2: Percentage of objects referred to in instructions compared to objects in the scene

3.2.3 Visual Analysis

Due to compute constraints, most visual analysis was performed qualitatively by inspecting visual quality of different types of images. Figure 3 shows some sample RGB, depth and instance segmentation images. All images retrieved during simulation are of size 300 x 300.

Object Name	Synonyms used in task descriptions
Coffee Machine	Espresso Machine, Beverage Machine
Chair	Couch Chair, Sofa Chair
CD	DVD
Side Table	Corner Table
Butter Knife	Bread knife
Ottomon	Loveseat, Recliner
Fridge	Kitchen Fridge, Refrigerator
Poster	Wall Photo, Picture
Safe	Safety Box
Soap Bottle	Lotion Bottle

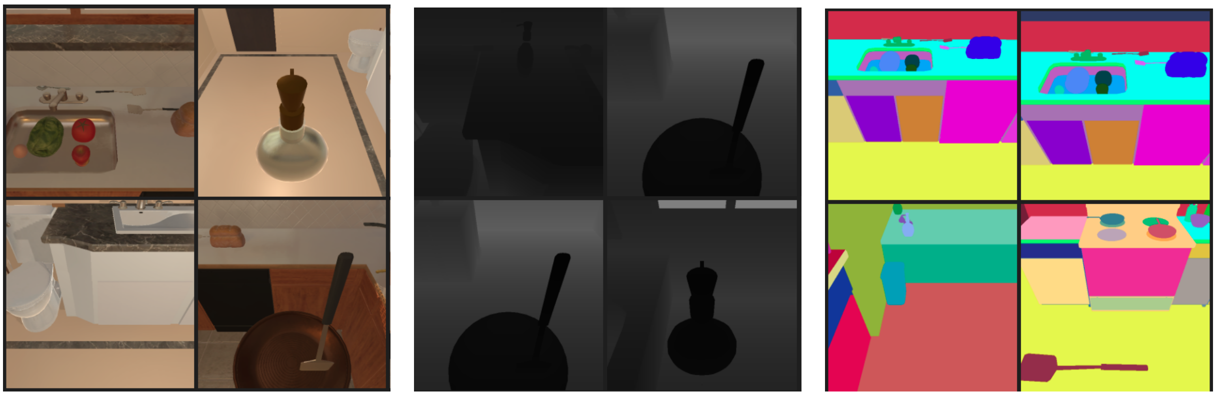
Table 2: Synonyms (i.e. closest words in embedding space) used in task descriptions of some objects in the dataset

3.2.4 Other Qualitative Analysis

In addition to the quantitative analysis, we also analyzed the solvability of the task. Our analysis in Figures 4 and 5 (in Appendix) shows that the unseen split of the validation set contains objects from the same classes as the training data, but could contain novel instances of those objects in novel environments. Since classes remain the same during training and testing times, the training data contains full information to solve the task, meaning a sufficiently intelligent agent should be able to solve the task, given training data.

3.3 Metrics

There are two primary metrics for evaluation. The first is “task success”, which is a binary value indicating if the object positions and state changes correspond correctly to the goal-conditions of the task at the end of the action sequence. The second is “goal-condition success”, which is the fraction of required goal-conditions that were completed at the end of the episode. Note that “task success” is true only if “goal-condition success” is 100%. Additionally, there exists a path-weighted version of these two metrics that considers the length of the episode, penalizing longer action sequences. For example, in the path-weighted version, an agent would receive half the score for taking twice as long as an expert to complete the task.



(a) (b) (c)

Figure 3: Sample RGB, depth, and instance segmentation images retrieved from AI2Thor simulator

4 Models (2 pages)

4.1 Baselines

Both existing baselines explained with citations and novel ones missing from the current literature

4.2 Proposed Approach

5 Results (1 page)

The columns above are just examples that should be expanded to include all metrics and baselines.

	Dev		Test	
Methods	Accuracy ↑	L_2 Error ↓	Accuracy ↑	L_2 Error ↓
Previous Approach 1 ()				
Previous Approach 2 ()				
Previous Approach 3 ()				
Proposed Method				

6 Analysis (2 pages)

This section should include at least two to three plots

6.1 Ablations and Their Implications

6.2 Qualitative Analysis and Examples

This section should likely contain a table of examples demonstrating how the current approach succeeds/fails.

References

Rodolfo Corona, Daniel Fried, Coline Devin, Dan Klein, and Trevor Darrell. 2020. Modularity improves out-of-domain instruction following. *arXiv preprint arXiv:2010.12764*.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

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*.

7 Appendix

7.1 Data Analysis Plots

Description of fields in Table 1:

1. Steps per directive: number of steps in each language directive

2. Tokens per step: number of words in each directive step

3. Task description tokens: number of words in directive task description

4. Images: number of images per demonstration

5. Actions: number of actions per demonstration

6. Images per action: number of images divided by number of actions per demonstration

7. Actions per step: number of actions divided by number of directive steps per demonstration

8. Nav-interact ratio: number of navigation actions divided by number of interaction actions per demonstration

9. Total objects: number of total objects in a scene per demonstration

10. Mask coverage: proportion of the image that is covered by the interaction mask per demonstration

11. Step-object coverage: proportion of interaction actions whose object of interest is mentioned in the step-by-step instructions, averaged over all interaction actions and language directives in the demonstration

In Figure 6, we see that ALFRED contains a roughly equal number of demonstrations for each type of task, and for the most part, a roughly equal proportion for each split. The validation data, especially the unseen portion, does have relatively less "Pick Two & Place" tasks than the training data. Additionally, the unseen portion has a significantly higher proportion of "Examine in Light" tasks than the other splits.

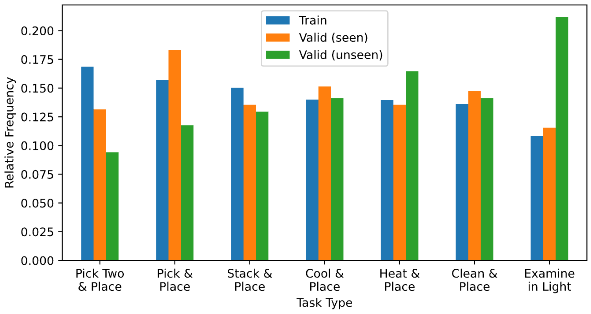


Figure 6: Relative frequency of each task type by split

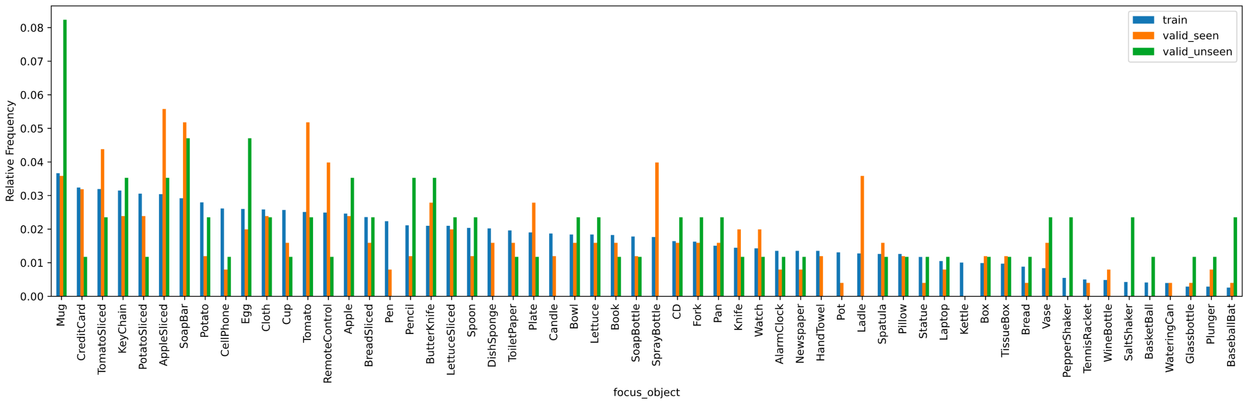


Figure 4: Relative frequency of focus objects used in different splits

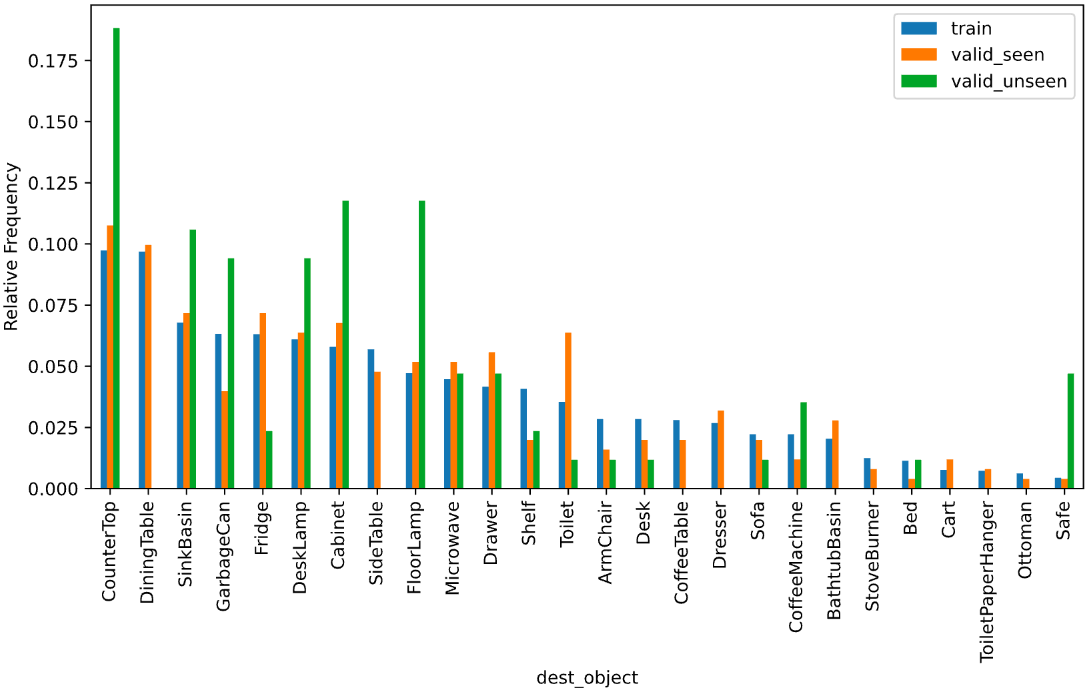


Figure 5: Relative frequency of destination objects used in different splits