Review12: Multi-Target Embodied Question Answering

Vivswan Shitole

CS539 Embodied AI

1 Summary

This paper extends the Embodied Question Answering (EQA) task to questions involving multiple targets (MT-EQA) that require the agent to navigate to the different targets and develop the specific relational understanding between the the targets that allows it to answer the question. The paper introduces a new MT-EQA dataset for the specified task built over the EQA-v1 dataset. The paper introduces a new modular agent for solving the task. The modular architecture is composed of a program generator, a controller, a navigator and a VQA module. Experimental results and ablations show the model outperform the baselines and that each module in the architecture is critical to performance.
More concretely, the MT-EQA task generalizes EQA-v1 and involves comparison of various attributes (color, size, distance) between multiple targets. All questions are generated from a fixed template of 6 question types. Each question can be decomposed into a series of functional programs which can be executed on the environment as a series of operations to yield the ground-truth answer. Infeasible questions involving targets that are not cross-navigable are removed. Entropy filtering is performed to remove questions with very certain answers. Finally the complexity of the MT-EQA dataset is verified by the trying to obtain the answers using question-only or prior-only baselines. These baselines yield close to random predictions. The program generator takes the question as input and sequential programs for execution. These sequential programs are selected from a sufficient template of 7 program types with placeholders for targets obtained from the question. The navigator executes the navigation programs. The navigator comprises of a LSTM that takes as input the current egocentric RGB image, a GloVE based encoding of the target phrase and the previous action to predict one of the 3 next actions (forward, turn left 30 degrees, turn right 30 degrees). The controller is the central module that receives the sequence of programs from the program generator and suitably invokes the navigator or the VQA module to execute each program. Moreover, the controller keeps track of the first person views during navigation, looking for the target. Hence the controller is implemented as a LSTM that receives the program sequence, the target embedding and the CNN encoded features of the current egocentric view. The controller predicts ”SELECT” if the target is found. When the controller has gathered all the targets for comparison, it invokes the VQA module with its the hidden state as the VQA input feature for target specific attributes. The compositional (c)VQA module embeds the stored features of multiple targets into the question attribute space using a FC layer followed by ReLU. The transformed features are then concatenated and fed into another FC+ReLU, which is conditioned on the comparison operator to yield the binary (yes/no) answer. Training is performed in two stages - (a) first stage is imitation learning with the objective function consisting of a navigation objective and a controller objective at every time step, and a VQA objective at the final step. (b) second stage is RL based finetuning using a dense reward signal corresponding to agent’s progress towards the goal and a sparse reward signal quantifying the agent’s view orientation towards the target object at episode termination.
Experiments are performed comparing the proposed model against strong baselines (such as REINFORCE) and model ablations. Following metrics are used for evaluation - (a) EQA accuracy as overall accuracy, accuracy for each of the 6 question types and accuracy by level of difficulty of the question defined task (easy, medium, hard). (b) Navigation accuracy: navigation performance is evaluated by computing the distance to the target object at navigation termination, change in distance to the target from initial spawned position to the termination position, the mean IOU ratio of the target at termination and the hit accuracy (percentage of IOU ratios greater than 0.5). Results show that: (a) RL helps both navigation and EQA accuracies. (b) Controller is critical since the performance deteriorates for a agent with only the navigator and VQA module. (c) Questions with shorter ground truth paths are easier, with the performance suffering for questions involving complex navigation.

2 Strengths

The paper is a great extension to the EQA task as it requires the agent to develop relational reasoning between targets along with learning complex navigation between them. The authors have done a good job in validating their proposed modular architecture for the agent as its shown to outperform the baselines and its modular ablations under several navigation and QA metrics by large margins. Each module of the architecture is well described (no ambiguities) and the results section is well structured. The authors have provided a figure (Figure 4) for the visualizing the dataset composition (which is a not so common good practise).

3 Weaknesses

Its not clear if the 2D birds view distance from the goal used as the dense RL reward signal for training is geodesic (good) or something else (bad). In table 7, its not clear why the BestView+cVQA agent (row 2) performs better than the ShortestPath+BestView+Ctrl+cVQA agent (row 3) for inroom object-distance-compare questions. Moreover, its not clear why BestView+cVQA agent (row 2) consistently performs better than ShortestPath+Ctrl+cVQA (row 5), when the authors claim the Controller to be a critical component for performance. The authors compare row 2 and row 3 to claim that the Controller's features help, but two things are changing between the agents at row 2 and row 3 (not so fair comparison). Finally, to claim that the Controller's SELECT decision matters, the authors compare performance with a sequential VQA module that uses the hidden state of a LSTM which encodes the whole sequence of frames along the shortest path. But even the VQA model in the original architecture uses the hidden features of the Controller LSTM to extract object attributes. This implies there should not be a big drop in performance when using the sequential VQA, but its found to yield random performance.

4 Reflections

This might be the best paper to read after the Embodied Question Answering paper as its a very good extension to that paper. This paper is an addition to the set of papers that show that inducing prior knowledge in the form of engineered modular architecture can yield better performance and generalization over end-to-end trained architectures. There's still room for improvement in the numbers obtained in the result section of the paper which can be taken up for further research.

5 Most Interesting Thought

It will be very interesting to see if there is a way to learn the sequence of programs, used by the controller as a plan to execute the task instead of creating them from a template.