Grounded Language Learning with Uncertain Instructions

Ashish Rao,[1] Kyle Hatch, [1]Sylvia Yuan[1]

[1]Department of Computer Science, Stanford University

Computer Science

Abstract

This project explores the question of how one can train an Imitation Learning agent in an instruction-following environment when the instructions provided may be ambiguous. To answer this question, we design a system consisting of two modules: the Imitation Learning agent itself, and a classifier that can predict whether a provided instruction is ambiguous given the state of the environment. If the classifier classifies an instruction as ambiguous, the instruction must be clarified by a user. Otherwise, the Imitation Learning agent executes the instruction. Our system is evaluated on a variety of environments in the BabyAI platform that are modified to produce ambiguous instructions. We study and present results for two different classifier architectures -- one based on a fine-tuned version of the GPT-2 model, and another based on an LSTM. We also show results comparing different ways to train the Imitation Learning agent.

Stanford University

Problem Setting

We investigate how to train Imitation Learning agents in instruction following environments with ambiguous instructions. Our experiments are with the BabyAI Platform, in which an agent must perform a specified task in a gridworld.

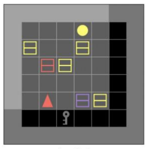

Figure 1: Example BabyAI task. The agent (red triangle) must follow the given instruction ("go to a yellow box")

We train a classifier to predict whether an instruction is ambiguous and an IL agent to execute instructions. In combination, this allows one to train IL agents with ambiguous instructions: if an instruction is classified as ambiguous, the user must clarify the instruction, and if it is classified as unambiguous, the IL agent executes the instruction.

Ambiguous Instructions

To test our method, we designed a method to automatically make BabyAI instructions ambiguous. BabyAI instructions have an internal tree-based representation:

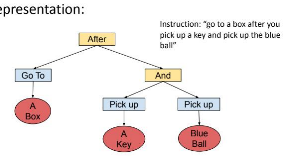

Figure 2: Internal Tree-Based Instruction Representation

We make instructions ambiguous by recursively making the subtrees of each node ambiguous. When we arrive at the leaves (which represent objects), we randomly drop certain descriptors of the object, such as its color, type, or location.

Classifier Network Architecture

Fine-tuned GPT-2: The classifier is a fine-tuned GPT-2 model from the HuggingFace library. The logits of the predicted distribution over next tokens is concatenated with a flattened state representation, and fed into fully connected layers to produce the classifier output.

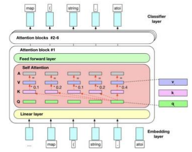

Figure 3: GPT-2 Architecture diagram from "Code prediction by feeding trees to transformers "[1]

LSTM: The LSTM is an RNN variant that can take into account long term dependencies in sequential data. We convert word vectors into embeddings, and feed them through LSTM layer and linear layer to output whether the instruction is ambiguous.

IL Agent Network Architecture

The instruction is fed through a GRU, and the state a CNN. The outputs are combined using FiLM (feature-wise linear modulation) layers, which is then fed into an LSTM that outputs actions and state values.

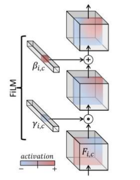

Figure 4: FiLM layer (performs affine transform of CNN outputs based on instruction)

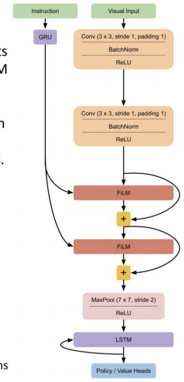

Figure 5: IL Agent Architecture

Classifier Experiments and Results

Dataset:

- 4000 instructions-state pairs(2800 in training, 800 in validation, and 400 in test)
- Randomly selected to be turned ambiguous, labelled by assessing whether the instruction is ambiguous with respect to the environment.
- Concatenate state and tokenized instruction, convert the vector to word index vectors

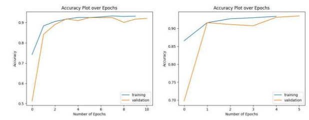

Figure 6: Accuracy Plots over Epochs. LSTM(Left), Fine-tuned GPT-2(Right)

Performance & Discussion:

- Similar accuracy
- GPT-2 converges faster (Pretrained)
- GPT-2 performance not significantly better: concatenated state and instruction as input, and GPT-2 has no experience in comprehending the state, which is necessary in judging whether the instruction is ambiguous.

Table 1: Highest Accuracy Achieved by the Models

	Training Accuracy	Validation Accuracy	Test Accuracy
LSTM	93.21%	92.50%	94.75%
Fine-tuned GPT-2	93.39%	93.50%	94.00%

RL Results

In addition to the classifier and IL experiments, we trained RL agents on the BabyAI environments. RL agents learned a policy of interacting with as many objects as possible to complete the task. Adding a penalty was insufficient to discourage this behavior.

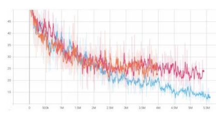

Figure 7: Identical performance of RL agents regardless of instruction type. Blue: plain, orange: ambiguous, pink: nonsense.

Imitation Learning Results

Imitation Learning agent is trained on demos collected by the BabyAI expert BOT.

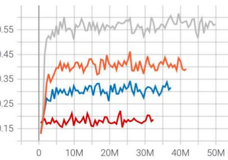

Figure 8: Training success rate of IL agents on the PutNextLocalS6N4 environment. Gray: plain, Orange: half ambiguous, Blue: all ambiguous, Red: nonsense

Evaluating the effect of the ambiguity classifier:

- If classifier detects ambiguity, clarification is requested, and the ambiguous instruction is replaced with the unambiguous instruction
- Effect of ambiguity classifier evaluated on three environments with 0.5 ambiguity rate
- Each agent evaluated for 1,000 episodes

Table 2: Evaluation Success Rates

	GoToLocal	PutNextLocal-S6N4	PickupLocal	Average
No Ambiguity (oracle)	95.9%	67.2%	62.7%	75.3%
No Classifier (baseline)	88.0%	51.4%	42.1%	60.5%
LSTM Classifier	89.7%	53.2%	51.6%	64.8%
GPT-2 Classifier	90.0%	55.0%	51.5%	65.5%

Results and Discussion:

- Using the ambiguity classifier results in slight increases in success rates
- Marginal benefits in easy environments
- Larger benefits in difficult environments

Selected References

- S. Kim, J. Zhao, Y. Tian, en S. Chandra, "Code prediction by feeding trees to transformers", in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, bll 150–162.
- Hui, David Yu-Tung, et al. "BabyAI 1.1." arXiv preprint arXiv:2007.12770 (2020).
- Perez, Ethan, et al. "Film: Visual reasoning with a general conditioning layer." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.
- Chevalier-Boisvert, Maxime, et al. "Babyai: A platform to study the sample efficiency of grounded language learning." arXiv preprint arXiv:1810.08272 (2018).