Abstract

We address the problem of training an artificial agent to follow verbal instructions. Our training data is a set of natural language instructions paired with demonstrations of optimal behavior. From the behavior, the agent obtains a reward function associated with each task by Inverse Reinforcement Learning. From the verbal instructions, the agent obtains a parsing of each sentence by Semantic Parsing. We add an abstraction component that enables the agent to learn parameterized tasks.

Introduction

Learning how to follow verbal instructions comes naturally to humans, but it has been a challenging problem to automate. In this paper, we address the following problem: given a verbal instruction, what is the sequence of actions that the instructed agent needs to perform to successfully carry out the corresponding task? There are many challenges such an agent is faced with: what is the meaning of the spoken sentence and how should it be parsed? How do words map to objects in the real world? Once the task is identified, how should the task be executed? Once the task is learned and executed, how can we generalize it to a new context, with objects and properties that the agent has not seen in the past? How should the agent’s learning be evaluated? and so on.

The goal of our project is to develop techniques that will permit a computer or robot to learn from examples to carry out multipart tasks specified in natural language on behalf of a user. Our approach uses three main subcomponents: recognizing intentions from observed behavior using extensions of Inverse Reinforcement Learning (IRL), translating instructions to task specifications using Natural Language Processing (NLP) techniques, and creating generalized task specifications to match user intentions using probabilistic methods for creating and managing Task Abstraction (TA).

Our paper is structured as follows: Section presents related work, Section describes the architecture of our system and we show a preliminary experiment in Section . We make a few concluding remarks in Section .

Related Work

Our work of grounding language in a real or simulated environment has strong connections with the problem of finding data-efficient methods for semantic parsing. Semantic parsing is the task of obtaining the meaning representation for a given text. Environment information can significantly reduce the amount of annotated language data used by supervised learning methods, traditionally used in Semantic Parsing. This observation has been applied to various problems, such as inducing a semantic parser from conversational logs (Atrzi and Zettlemoyer 2011), eliminating ambiguity by using a grounded concept labeling of the words in a text (Bordes et al. 2010), question answering (Liang, Jordan, and Klein 2011), and using a feed-back signal to learn the meaning of words in a database (Clarke et al. 2010). In this paper we are interested in the problem of interpreting verbal commands. We eliminate the explicit supervision needed when learning a semantic parser by pairing verbal commands with behavior.

Other papers have addressed the problem of teaching computers how to accomplish a new task, given natural language instructions and demonstrations of desirable behavior. Some approaches address learning navigational instructions (Chen and Mooney 2011), (Vogel and Jurafsky 2010), (Grubb et al. 2011), while others extend it beyond the navigation problem (Hewlett, Walsh, and Cohen 2010), (Antoine Bordes 2010), (Tellex et al. 2011), citegoldwasser11. One possible approach is to derive a task plan from each given demonstration and use pairs of plans and the corresponding verbal instructions to train a parser (Chen and Mooney 2011). Another approach is to see sentences as bags-of-words, and use these words, together with environment information to build a state space (Vogel and Jurafsky 2010). If the agent is given a feed-back signal proportional to its closeness to the demonstrated behavior, a reinforcement learning algorithm can help it learn optimal actions in any given state. An alternative sentence model (Grubb et al. 2011) assumes a fixed structure for the verbal instructions: (verb, spatial relation, landmark) and uses an inverse optimal control approach to find a plan that most likely generated the demonstrated behavior. In contrast with these approaches, we aim to learn more complex tasks, and allow for richer language. We exploit the fact that imperative sentences have a certain structure, thus using information about
the parts of speech for each word, while, at the same time, allowing for flexibility in the sentence structure.

The problem of learning other kinds of tasks has been addressed (Hewlett, Walsh, and Cohen 2010), (Antoine Bordes 2010), (Tellex et al. 2011), (Goldwasser and Roth 2011).

System Architecture

The architecture of our system has three major components, presented in detail below. The training data is a set of verbal instructions paired with demonstrations of corresponding optimal behavior. For example, one of these pairs could be the instruction /emphMove the chair to the green room with a demonstration of how the task would be accomplished in a given environment, with various pieces of furniture and rooms of different colors. We also assume the availability of a set of features for each state. These features play an important role in defining the tasks we are interested in learning. For example, if a robot is taught to move around furniture, it will, at any moment in time, need information about whether or not it is currently carrying a piece of furniture, what piece of furniture it needs to be moving, which room it is currently in, which room contains each piece of furniture, etc. For each task, IRL takes the demonstrations and information about the environment, and produces a set of features that are relevant to the goal of the task, according to the demonstrations. For example the most important features for a furniture moving robot that is instructed to put a chair in the green room could be a) there is a chair in the green room and b) the robot’s arm is empty. The NLP component takes the verbal instructions and produces the corresponding logical form. For example, if the robot receives the following verbal instruction: Move the chair to the green room, NLP will produce the parse: move(chair1, room1), P1(room1, green), together with what are the parts of speech for each word. The TA component takes the set of features selected by the IRL and the logical form produced by the NLP and compiles a set of parameterized tasks, one level of abstraction higher than the tasks in the training set. We present each of these components in more detail below.

Inverse Reinforcement Learning

Inverse Reinforcement Learning (Abbeel and Ng 2004) addresses the task of learning a reward function from demonstrations of expert behavior, given information about the environment, such as a set of states (what is going on), actions (what an agent can do), and a transition function (how do an agent’s actions change the state). Recently, data efficient IRL methods have been proposed, such as Maximum Likelihood Inverse Reinforcement Learning (MLIRL) (Babes-Vroman et al. 2011). Given a small number of trajectories, MLIRL finds a weighing of the state features that maximizes the probability of the demonstrations. For a given task, MLIRL can also be used to select a set of possible reward functions and their probabilities. This set can be passed to TA and TA can combine the information in receives from IRL and NLP to adjust the probabilities and send the results back to the other two components, so the probabilities over reward functions for IRL and parses for NLP get further refined.

Natural Language Processing


