Robot Route Instruction Learning

Praveen Dhinwa
Advisor: Dr. Amitabh Mukherjee
Dept. of Computer Science and Engineering
{dpraveen, amit} @ cse.iitk.ac.in

April 20, 2013

As robots are becoming more advanced and capable of performing of complex tasks , the importance of enabling untrained users to interact with them has increased. Hence unconstrained natural-language interaction of humans with robots has emerged as a significant research area. It has found various uses in Robot Navigation System , traffic guiding system or simulation games. We have trained the program to take natural language input from the user, in our case English and executing on the Robot Control System and estimating the accuracy of correctness. For this PCFG (Probabilistic Context Free Grammar) approach is used to understand the semantic intent of the words. A previously collected realistic corpus of complex English navigation instructions for these environments is used for training and testing data. By using a learned lexicon to refine inferred plans and a supervised learner to induce a semantic parser, the system is able to automatically learn to correctly interpret a reasonable fraction of the trivial and non trivial instructions in this corpus mentioned above. The system is tested and trained on the 3 complex maps.

Keywords: PCFG , Navigation Instructions , RCL , CCG , CRF , Robot Simulation System

1 Introduction

An important application of natural language processing is the interpretation of human instructions. The ability to parse instructions and perform the intended actions is essential for smooth interactions with a computer or a robot. The goal of the navigation task is to take a set of natural- language directions, transform it into a navigation plan that can be understood by the computer, and then execute that plan to reach the desired destination. Route direction is a unique form of instructions that specifies how to get from one place to another and understanding them depends heavily on the spatial context. Take example of such an natural language instruction turn so that the wall is on your right side. walk forward once. turn left. walk forward twice. As this instruction has use of landmarks in it. Essential objective is to convert natural language instruction into robot executable format. Given a start position and natural language input by the user, parse the input into a RCL structure. Then executed the parsed RCL structure output on the grid type navigation system. Idea is to know about how well the system to able to interpret the instruction and able to execute on the sytem.

2 Previous work in this field

There have been a lot of experiments in this area. Two of them are most important on which most of my work is based on. Those experiments are given below.

2.1 Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, Dieter Fox

They train a semantic parsing model that defines, for any natural language sentence, a distribution over possible robot control sequences in a LISP-like control language called Robot Control Language named as RCL. But their grid system is not so complex , their RCL also has really less amount of parse structures in it. So there has not been a use of landmarks in this. For their work, parsing is performed using an extended version of the Unification-Based Learner, UBL.The grammatical formalism used by UBL is a probabilistic version of combinatory categorial grammars, or CCGs, a type of phrase structure grammar. CCGs model both the syntax (language constructs such as NP for noun phrase , PP for phrase structure) and the semantics (expressions in -calculus) of a sentence. UBL creates a parser by inducing a probabilistic CCG (PCCG) from a set of training examples. UBL first generates a set of possibly useful lexical items, made up of natural language words, a -calculus expression, and a syntactic category. (An example lexical item might be < left, turn-left, S > .).

2.2 David L. Chen and Raymond J. Mooney's Work

Their robot execution system makes a exntensive use of landmarks. Their grid system is quite complex and contains a lot of objects. They perform parser learning over a sentence of route instructions through a complex indoor environment containing objects and landmarks with no prior linguistic knowledge. However, their work assumes initial knowledge of a map, for training and testing.

3 My Work

My work is a combination of 1 and 2. Data set for training and testing the system is taken from [2]. Data set is a xml file containing the paragraph with both trivial and non-trivial sentences. Trivial sentences are those sentences which contain only one line and are easy instructions. On the other hand non-trivial sentences might have 2-5 commands or instructions in that. Data set is given in form a xml file in which with each sentence the starting and ending location is also given for perticular map as mentioned below. So I am using this information to train and test the simulator.
The grid map Set is also constructed by [2] and taken by me for the project. It contains three kind of map strucutres. There are 3 grid maps given in the form of xml file. Grid maps are made of nodes and edges.Where nodes signify the objects at the particular locations. eg. Hatrack, easel, chair etc . On the other hand Sample robot simulation system is also MARCO is also by [4]. I have used this simulation system for my project. The algorithm for learning the features of a map is taken from Chen and Mooney Paper and implemented by me on the MARCO system. Then parse structure of route instructions is also learnt by implementing it on the grid data sets. Algorithm for this is PCFG(probabilisistic CFG) whose implementation is taken from KRISP.[7]
A simulation of the above project is created by me using python tktiner library. Basically the image of the map is taken and used as a back screen and it's pixels are matched to output by the MACRO on the input string that user gives. User can intereact with this simulation by entering the instructions that he wants the simulator to execute.

3.1 Different Aspects of problem

We assume that we do not have any prior linguistic knowledge of any kind , be it syntactic or semantic or lexical. It means we have to learn the meaning of every word, including object names, verbs, spatial relations, as well as the syntax and compositional semantics of the language. The only supervision we receive is in the form of observing how humans behave when following sample navigation instructions. This is implemented by taking the actual path given in the data sets.

Formally, the system is given training data in the following structural form:

{ $(e_1 , a_1 , w_1), (e_2 , a_2 , w_2), . . . , (e_n , a_n , w_n)$ }, where e_i is a natural language instruction, ai is an observed action sequence, and w_i is a description of the current state of the world including the patterns of the floors and walls and positions of any objects. The goal is then to build a system that can produce the correct aj given a previously unseen (e_j , w_j) pair.

Given observation (e_i, a_i, w_i) , we construct a navigation plan p_i based a_i and w_i. The goal is to build a system that can produce the correct actions sequence a_j on a new (previously unseen) (e_j , w_j) pair.

The resulting pair (e_i , p_i) is then used as supervised training data for learning a semantic parser. During testing, the semantic parser maps new instructions e_j into formal navigation plans pj which are then carried out by the execution module.

Once we obtain the supervised data in the form of (e_i , p_i),we use KRISP (Kate and Mooney 2006)[7] to learn a semantic parser that can transform novel instructions ej into navigation plans p_j (i.e. transform turn to face the sofa into Turn(), verify(front: SOFA).)

KRISP is a publicly available learning algorithm for translation of natural language strings / instructions to a formal language defined by CFG (context free grammar). Given parallel training data in form of natural language strings with their corresponding formal meaning representations , it learns a set of strings that decide how to construct meaning representations.

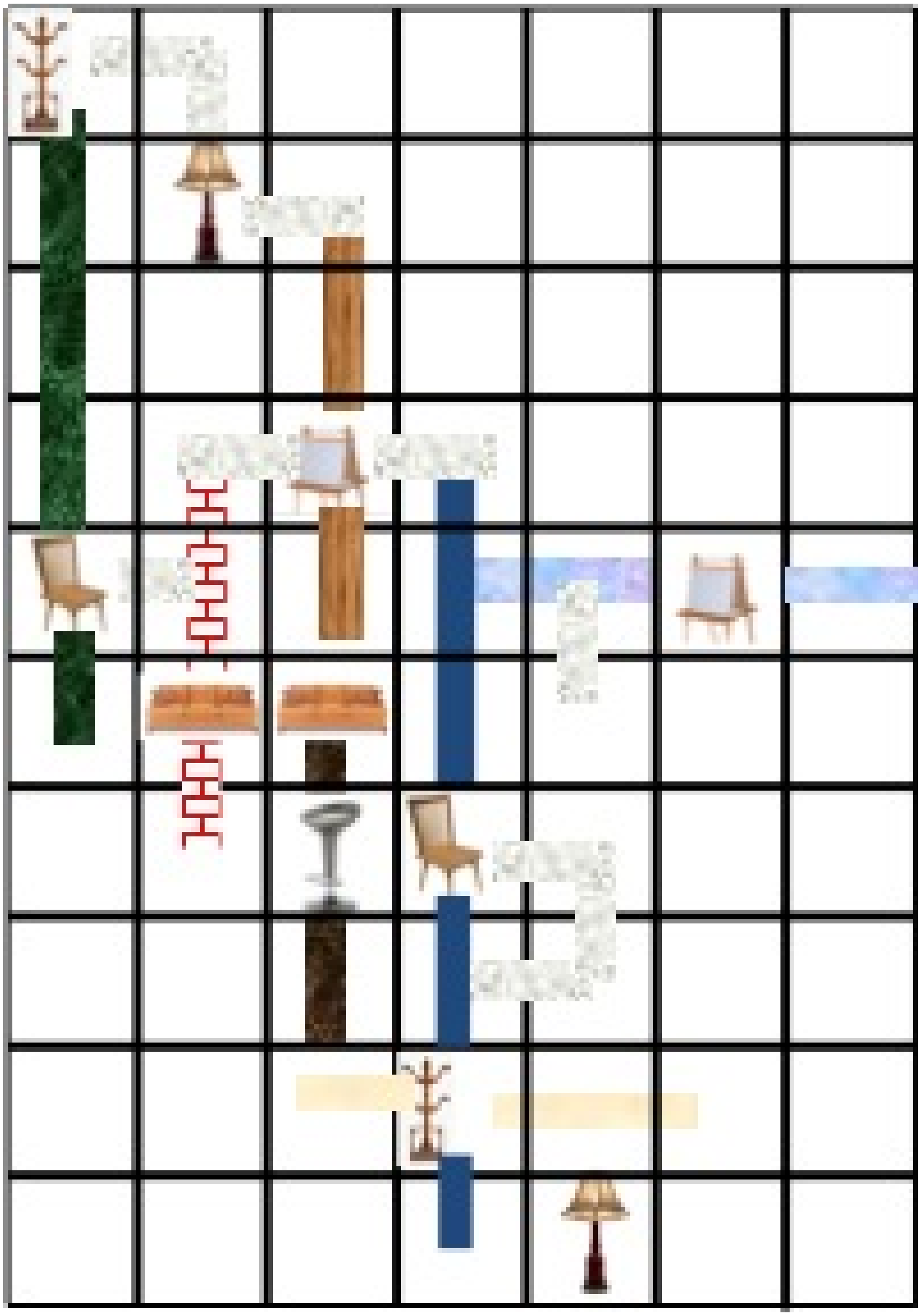
3.2 Lexicon Learning Algorithm

The lexicon Learning algorithm is taken from [2].
Input:
Navigation instructions and the corresponding navigation plans $(e_1 , p_1), . . . , (e_n , p_n)$.
Output: Lexicon, a set of phrase-meaning pairs

```
main:
    for n-gram w that appears in e = $(e_1 , . . . , e_n )$ do
        for instruction ei that contains w do
            Add navigation plan $p_i$ to meanings(w)
        end for
        repeat
            for every pair of meanings in meanings(w) do
                Add intersections of the pair to meanings(w)
            end for
            Keep k highest-scoring entries of meanings(w)
        until meanings(w) converges
        Add entries of meanings(w) with scores higher
        than threshold t to Lexicon
    end for
end main
```

4 Data Set

1. Data set for training is taken from [4]. It contain corpus of english grammar system used for navigation in the general kind of system.It contains 706 both trivial and non trivial route instructions. Instructions also have their corresponding path with them for learning purpose.
2. 3 grids as mentioned above are taken from [2].
3. 600 of the data set from above 706 is used for training on the above 3 grids. Remaining data set is used for testing.
4. Execution system is taken from [4].



5 Results

For the 106 instructions that were taken from the data set (as mentioned above). This instructions contained both the trivial and non-trivial sentences. Out of those 67 were parsed correctly. So overall the success rate of the our testing could be said as 67 / 106 : 63 . Note that this accuracy is tested by the actual path given in the collected data set. Note that simulation was working good also on a sentence containing even 3-4 instructions but it was not working so good in turning part , Hence the accuracy got reduced.
Also note that my accuracy is fairly low than actual accuracy of experiment(including landmarks). The accuracy for the later was around 81.46 . Though my experiment is a simpler portion of that.

6 Future Work

Recognition of extensive landmarks can help to improve the system a lot. For implementing this learning has to be changed and semantic intent of the objects has also has to be taken care of.

7 References

1. Learning to Parse Natural Language Commands to a Robot Control System
Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, Dieter Fox
2. Learning to Interpret Natural Language Navigation Instructions from Observations. David L. Chen and Raymond J. Mooney
AAAI Conference on Artificial Intelligence (AAAI), 2011
3. Y. Artzi and L.S. Zettlemoyer. Bootstrapping semantic parsers from conversations. In Proc. of the Conf. on Empirical Methods in Natural Language Processing, 2011.
4. Marco Code written by Matt MacMahon (matt@macmahon.org).
http://robotics.csres.utexas.edu/ adastra/papers/b2hd-macmahon-phd-07.html
5. A. Ferrein and G. Lakemeyer. Logic-based robot control in highly dynamic domains. Robotics and Autonomous Systems, 56(11), 2008.
6. T. Kwiatkowski, L.S. Zettlemoyer, S. Goldwater, and M. Steedman. Inducing probabilistic CCG grammars from logical form with higher-order
unification. In Proc. of the Conf. On Empirical Methods in Natural Language Processing, 2010 .
7. KRISP (Kate and Mooney 2006)