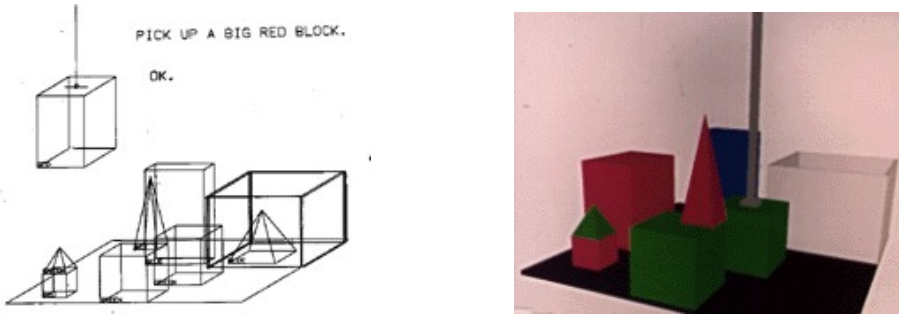
Literature Survey on NLI to Robots

Since the time of Golems and Galatea, humans have been dreaming of creating Artificial Intelligence. These artificial beings were, no doubt, creation of artists. With the advancement in technology, the field is now in the domain of computer scientists. Robotics, and hence true AI, involves expertise from a lot of discipline including vision, control systems, planning and recently natural language processing. Vast amount of work has been done in these fields. However, these fields have developed almost independently. It is only in recent few years that we find research, integrating them on real robots. Integrating NL to robots, working in real world, is currently a hot topic of research. This task is challenging, as it requires amalgamation of all the fields. Thus in this chapter we give a brief overview of the work done in all these system.

First we give a overview of the household robots (autonomous systems). Then we briefly describe, well known SHRDLU [12] system, which accepts command in natural language and manipulates block world. Then we describe the much celebrated subsumption architecture by R Brooks [2]. Household robots like roomba, cleaning robot, has been built using this architecture. After that we describe few recent system where a mobile robot accepts a navigation and object manipulation command in NL and acts accordingly. Lastly we describe the geometric and traditional symbolic planners.



(a) Original screen display

(b) Later color rendering (Univ. of Utah)

Figure 1: SHRDLU block world. From *http://hci.stanford.edu/winograd/shrdlu/*

SHRDLU

It is a system for understanding natural language, developed by T Winograd [12]. It carried out natural language commands in a simulated, but symbolic, block-world. Robot is represented as an arm capable of lifting and placing objects. Objects are 3D objects like blocks, boxes, cones *etc.,* as shown in Fig 1 below.

It carries out a dialogue with the user. Its capabilities include answering questions, executing commands and accepting information in English. It does syntactic analysis of the sentence with heuristics, combining information from discourse and world knowledge to decide the meaning of the sentence. It asks for clarification in English. Its language understanding capability are impressive till date [14]. Here is a part of the dialog that was used as a SHRDLU demo. It is taken from [13] : *http://hci.stanford.edu/winograd/shrdlu/*

Person	:	PICK UP A BIG RED BLOCK.
Computer	:	OK. (does it)
Person	:	GRASP THE PYRAMID.
Computer	:	I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person	:	FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.
Computer	:	BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING. OK. (does it) ...
Person	:	HOW MANY BLOCKS ARE NOT IN THE BOX?
Computer	:	FOUR OF THEM.
Person	:	IS AT LEAST ONE OF THEM NARROWER THAN THE ONE WHICH I TOLD YOU TO PICK UP?
Computer	:	YES, THE RED CUBE. ....
Person	:	CAN A PYRAMID BE SUPPORTED BY A BLOCK?
Computer	:	YES. ...
Person	:	HOW MANY THINGS ARE ON TOP OF GREEN CUBES?
Computer	:	I'M NOT SURE WHAT YOU MEAN BY "ON TOP OF "IN THE PHRASE "ON TOP OF GREEN CUBES " . DO YOU MEAN: 1. DIRECTLY ON THE SURFACE 2. ANWHERE ON TOP OF ?
Person	:	2
Computer	:	THREE OF THEM.
Person	:	HAD YOU TOUCHED ANY PYRAMID BEFORE YOU PUT THE GREEN
	:	ONE ON THE LITTLE CUBE?
Computer	:	YES, THE GREEN ONE...

It constituted a significant step in NLP research as it combined models of human linguistics and reasoning methods in the language understanding process. Its problem solving is highly oriented towards predicate calculus. However it was not able to scale

out into the real world, which is too complicated for a symbolic planner[4]. Also it has problem in dealing with the hypothesis.

Subsumption Architecture

It is a layered architecture advocated by Rodney Brooks. Each layer deals with a specific goal of the agent and higher layers are increasingly abstract. Higher layers can subsume and inhibit, input and output of the lower layer modules. In contrast with the SHRDLU system, there is no abstraction. In fact it is claimed that abstraction is dangerous [2]. It reduces the AI problem to essentially a search and the modules which brings abstraction are treated as black-box. The problem of recognition, spatial understanding, dealing with sensor noise, partial models *etc.,* are all ignored completely. The expert system, like MYCIN [7], are completely rule based and hence are highly constrained. For example MYCIN, although an expert at diagnosing human bacterial infection, has no human model and thus for any other problem like incision *etc.,* it tries to find the bacterial cause.

The architecture approaches intelligence in incremental manner. Instead of directly trying to approach human level intelligence and reasoning, it focuses first on behavior based intelligence exhibited by insects. It is argued, based on evolution history, that simple tasks, like the ability to move around in a dynamic environment, sensing the surrounding and taking appropriate action *etc.,* is much harder. This part of the intelligence is where evolution has concentrated most of its time. Once these are achieved than problem solving behavior, language, expert knowledge and application, and reason will be pretty simple. Hence all the systems which are build are actually deployed in the real world. Such systems has been called creatures. There are multiple layers of behavior present in such creatures.

For example, in [1], the lowest layer could just be to '*avoid an object*'. Thus when left in a confined space it will move itself towards center of it. As shown in Fig 2, it has simple modules like:-

- **Sonar** : It produces a robot centered map of obstacles in polar coordinates.
- **Collide** : If it detects an obstacle ahead, a halt signal is passed to the motor.
- **Feel force** : Each obstacle detected contributes to some repulsive force, which are summed to generate a single resultant force.
- **Runaway** : When the resultant force become significant it sends the command to the motor module.
- **motor** : Apart from commands from halt and runaway it also accepts command directly from the robot. However whenever a halt command is given the robot stops.

A higher layer like '*wander around*' works on top of it. It relies on its lower layer for most of its functionality. In addition it also plan ahead a little so that it can avoid potential collisions. It is shown in Fig 3, additional modules are :-

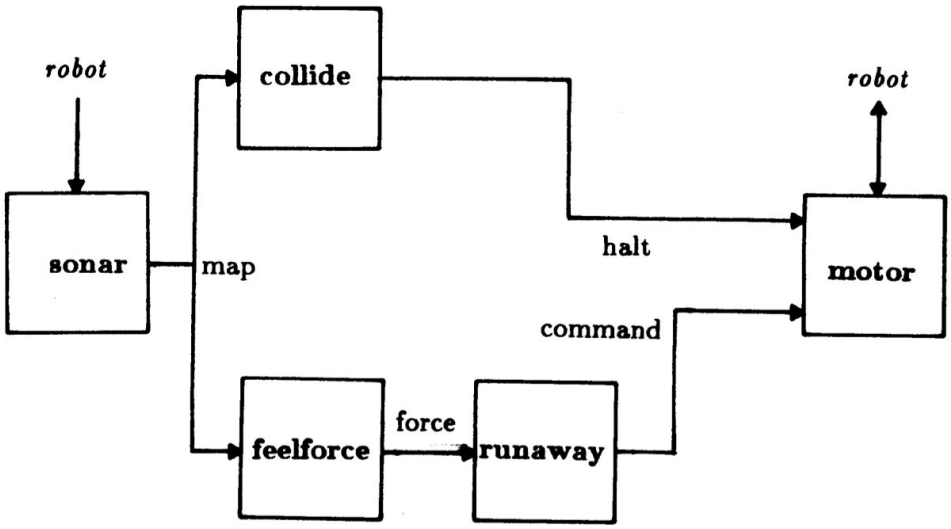


Figure 2: Lowest layer '*avoid an object*'

- **Wander** : Generates a new heading after a fixed time, say 10 seconds.
- **Avoid** : It perturbs the output of wander modules to avoid potential collisions. It combines feelforce module output as well and subsumes the input to the motor module.

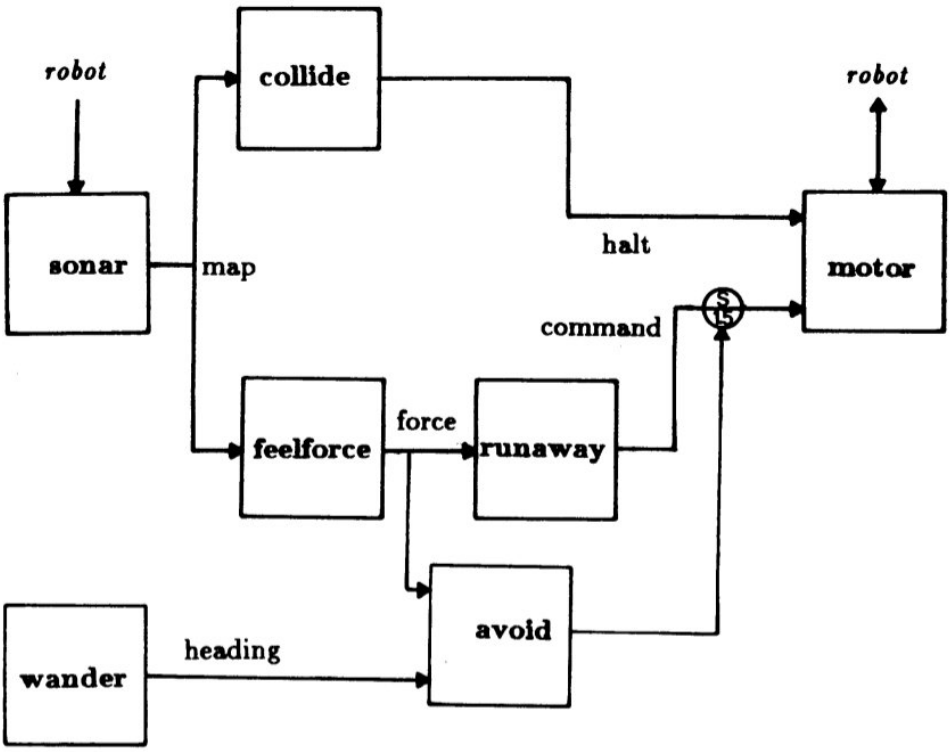


Figure 3: A higher layer '*wander around*'

On top of these layers another, seemingly more intelligent, layer like '*reach a goal*' can run. It has various modules like:-

- **Grabber** : It sends a halt signal to the motor to momentarily stop it. During which stable readings are taken and a long distance goal is fed to pathplan module.
- **Pathplan** : It subsumes the input to the avoid module by feeding it with the heading direction required to reach the goal. It also takes integrate module input to check how far it is from the goal, and once the goal is almost reached it sends command to straighten module for final alignment in the pose.
- **Monitor** : It constantly monitors, how far the robot has traveled.
- **Integrate** : It accumulates input from the monitor module and send them to pathplan and straighten module.
- **Straighten** : Once the robot is at its goal, this modules orient it to the desired degree.

The result of these layers is shown in the Fig 5.

When no goal is present the robot wander in the environment, avoiding obstacle. Once it gets a goal it tries to reach that goal, avoiding the obstacles. To an observer the behavior of the creature will look intelligent. In fact if the world will be more complicated, full of many obstacles, the behavior might seem more intelligent. Thus it explains that complexity lies in eyes of the observer[2]. Here no explicit representation of world is made (hence no abstraction is done) in-fact the world acts as its own representation.

House cleaning robots like Roomba [3] has been successfully build using this architecture. However higher level of intelligence has so far not been implemented on this. As more and more layers are build the goals begin to interfere with each other and thus the no of layers become limited. Also it is difficult to integrate language [14] to it. This is not surprising as language deals with abstracting real world (and representing it symbolically) and this architectures makes world as its own representation. Also language is a very recent phenomenon in the evolution history.

Work related to NL Interface

There are a no of systems, which performs a specific task, and has a NL interface for it. We have already seen SHRDLU which works in a simulated world. More recently there has been some work in developing systems which can take commands in natural language and work in real or real-like simulated environment.

- In [11] NL commands are interpreted for navigation of a robot. The system takes direction to a place in NL (as it is given to human) and outputs a sequence of region which it will take to reach there. First a map of the environment is created, by traversing a robot in it. Robot is mounted with camera and laser, using which

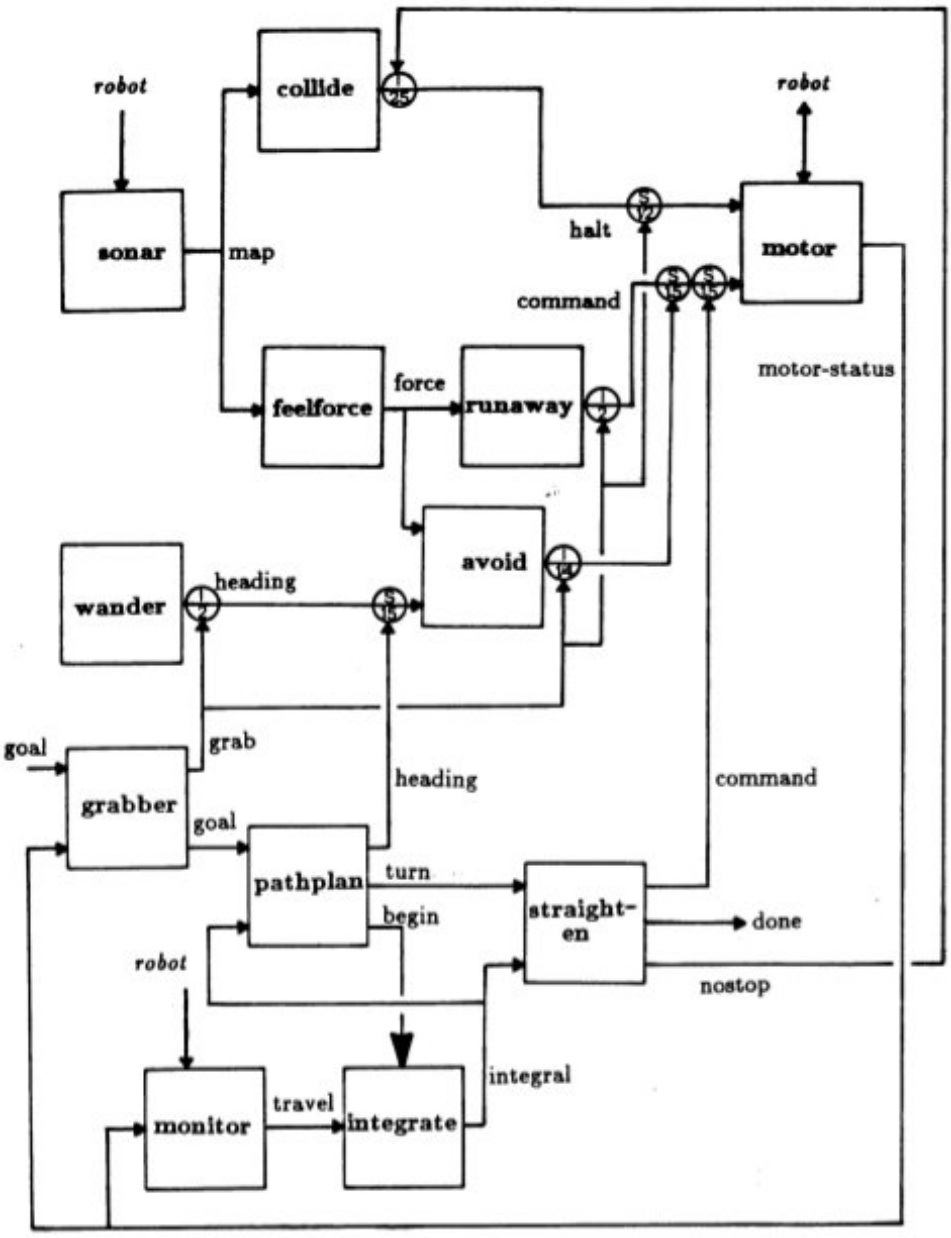
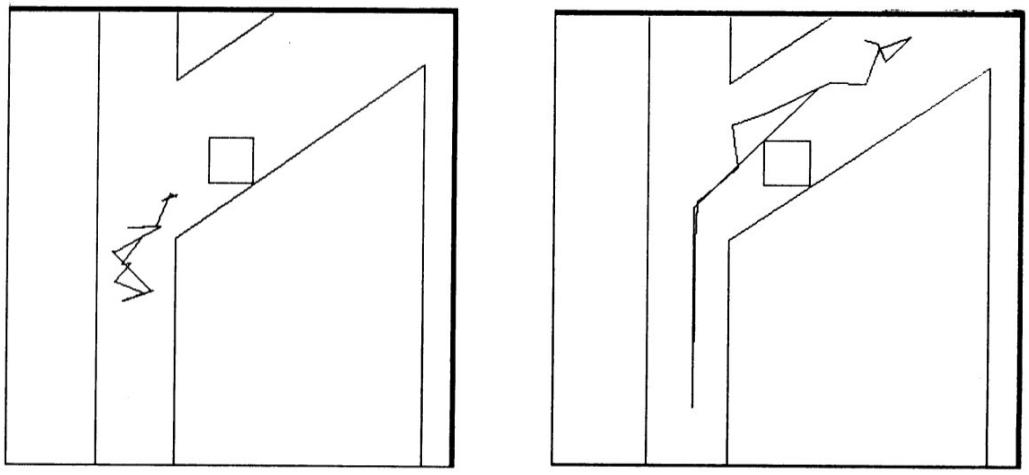


Figure 4: A still higher layer '*reach a goal*'

it identifies the structure and some novel objects like fridge, sofa *etc.,* as shown in Fig 6

The map can be created on the fly and even created by hand. While creating this map, few objects (z) in the given region (r_i) are also detected. From this $p(z|r_i)$ is calculated.

From NL sentence nouns are extracted (in order) and it is assumed that they will correspond to the landmarks (landmarks could be abstract objects like "kitchen", "office"or specific objects like "microwave","table"), through which traversal will happen. A variant of MRF is used to predict the most probable state (regions).



(a) Effect of first two layer

(b) Effect of exploring thw world when goal is given via two straight lines

Figure 5: Behavior of the creature governed by various layer

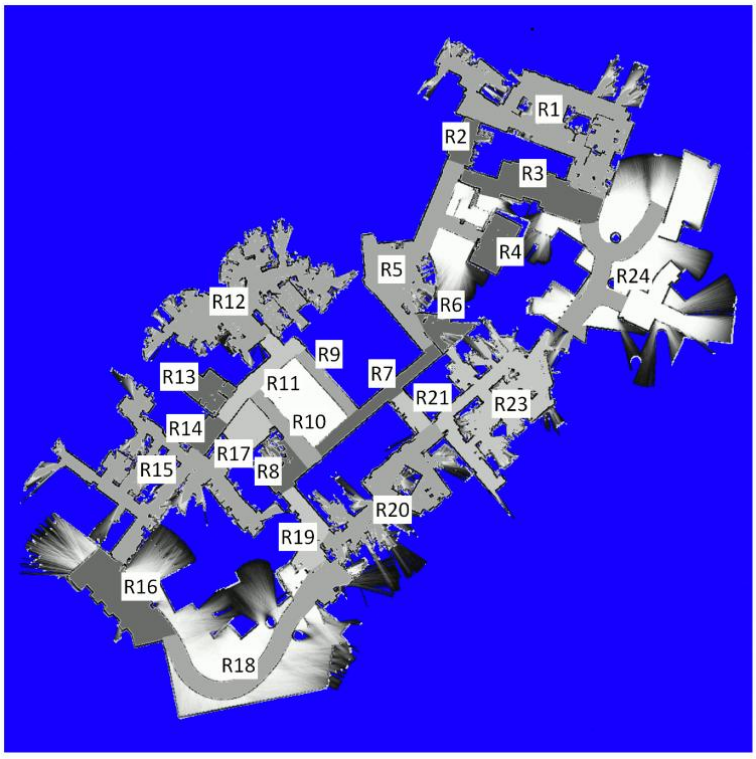


Figure 6: A sample map, segmented in different regions.

The model used is depicted in Fig 7

The goal is to determine the region (r_t referred as states s_t) to which the landmark belongs. The problem is posed as :

$$\arg \max_{s_1,...,s_T} \phi(s_1, ..., s_T | z_1, ..., z_T).$$

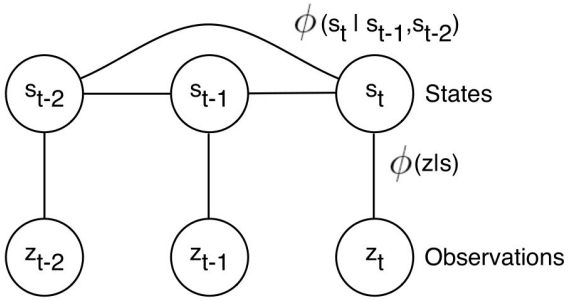


Figure 7: A variant of MRF model used.

Using Bayes rule, it is proportional to :

$$\arg \max_{s_1,...,s_T} \phi(z_1, ..., z_T|s_1, ..., s_T)\phi(s_1, ..., s_T)$$

Assuming that objects only depend on the current state and a region depends on N previous state. it can be expressed as :

$$\arg \max_{s_1,...,s_T} p(s_1)\phi(z_1|s_1) \prod_{t=2}^{T} \phi(z_t|s_t)\phi(s_t|s_{t-1}, ..., s_{t-N})$$

For the objects detected, $\phi(z_1|s_1)$ was calculated during the map preparation. For an unknown object this probability is approximated using tagged Flickr image dataset.

$\phi(s_t|s_{t-1}, ..., s_{t-N})$ can be easily calculated for N=1, as 1 if regions t and t-1 are connected and 0 otherwise. Now a Viterbi like algorithm can be run to get the regions $s_1, ..., s_T$

Although in a closed environment it follows 85% of the path correctly which is followed correctly by humans, in a larger setting the accuracy falls. Also if map is generated on the fly, then also accuracy falls.

- In [5] authors tries to ground linguistic categorization of space in aspects of visual perception. Specifically a score was given for the two objects, indicating how well they satisfy the terms like "object A is *above* object B". Four different models were considered and it was concluded that *AVS (Attention vector Sum)* model was able to fit to the experimental data more accurately than any other model. AVS model depends on two independent observations:-

  1. *Attention* : As humans also consider only *interesting* regions of an object, similarly the model also considers point (a beam is focused) of the landmark

(central object relative to which another object is referred) top, that is vertically aligned with the trajector (the object being referred).

2. *vector sum* : Overall direction is dependent upon the vector sum of a set of constitute direction. All the points in the beam are considered for vector-sum direction calculation, but their contribution decreases as they depart from the center of beam.

AVS model combines the two observation by focusing an attention beam to a point on landmark nearest to the trajector. It is argued that power in the AVS model comes as a fact that it combines two independent observations. This model is useful for calculating spatial relation between the objects. It has been used in [8].

- In [9] a model for understanding language about space and movement in realistic situations has been developed. The dissertation has two main contribution. Firstly it provides a set of features for detecting if a scene corresponds to the given prepositions. Thus if the path in natural language is say *"around kitchen"* then a path (marked in the figure) which does satisfy this criteria will get a higher weight-age as compared to other paths. The features are given in terms of "Ground" (landmark, which could be a point or area (say area of kitchen)) and path. Few examples of the features are:-

  1. Distance between the figure center of mass and centroid of the ground.
  2. Distance of minor axis inside ground.
  3. Ratio of distance between start and end point of figure and the axis, and so on

  The second contribution is introduction of SDC (Spatial Description Clauses). It consists of

  1. a figure (the subject of the sentence),
  2. a verb (an action to take),
  3. a landmark (an object in the environment),
  4. and a spatial relation (a geometric relation between the landmark and the figure).

  Though it is hierarchical like NL, but for automatic extraction they were linearized. Different components of SDC can be trained separately. While testing the system NL was converted into SDC and the path which satisfies the relations (preposition) most suitably was considered. When global inference was done it was found that system was able to follow 85% of the paths correctly as compared to humans.

- In [8] description of a system is presented which takes a command in NL and generates a sequence of actions for the robot (in forklifting domain). It uses a probabilistic approach to interpret NL command. It estimates a distribution over the mapping between the referenced objects, places and actions; given the command. The compositional structure in the command is used for estimating the distribution. Thus the problem of following instructions (generating action sequence) gets reduced to inferring the most likely grounded state sequence under the above setting.

  Defining $\Gamma = \{\gamma_i\}$ as set of all groundings and Φ as a binary variable which is true if the grounding is correct, then the objective is expressed as :-

  $$\arg\max_{\gamma}(\phi = T|command, \gamma)$$

  Given a NL command, first it is transformed to a hierarchical structure of SDCs (Spatial Description Clauses). This is done using dependencies extracted (exact details not described) using Stanford dependency parser. SDCs consists of :-

  1. EVENT (verb in previous setting) eg : *Move the tire pallet.*
  2. OBJECT (landmark in previous setting) eg : *Forklift, the tire pallet, the truck, the person.*
  3. PLACE (landmark along with spatial preposition as in previous setting) eg : *on the truck, next to the tire pallet.* These are places.
  4. PATH (landmark along with preposition describing the path) eg : *past the truck.*

  Then G^3 (Generalized Grounding Graphs) is constructed using the structure of SDCs. Its a bipartite factor graph with a factor Ψ for each SDC. The other nodes connected to the SDC_i are its grounding γ_i and binary variable ϕ_i. Thus we have

  $$p(\phi|command, \Gamma) = p(\phi|SDC_i, \Gamma) \propto \prod_i \Psi_i(\phi_i, SDC_i, \Gamma)$$

  Thus given a new command a Generalized Grounding Graph could be constructed by first making the SDC of the sentence and then converting that SDC to the graph. The conversion seems quite straight forward as prepositions will link each clique. Here clique is defined as a landmark, corresponding grounding (not known in advance) and a binary variable which is true if the grounding is correct. Depending upon the type of node (EVENT, OBJECT, PLACE or PATH) different features are used for giving weight-age to the grounding. Thus for the correct grounding the expression should give maximum value. Experimentally it was found that for hand coded SDC system is able to achieve a precision of 0.63.

Although SDCs are simple and intuitive they fail to capture many types of sentences. If a landmark (object) has some property like color, size , or say a side of object has some different color then we need extra node to capture it. Also it seems difficult to extend SDCs to more richer commands, like hypothesis.

RoboFrameNet

RoboFrameNet [10] is a ROS package which has been developed to execute commands given in NL by PR2 and turtlebot. It was developed for the fuerte version of ROS. It uses FrameNet's [6] frames as an intermediate representation between the NL and robot action. The sentence is first parsed using Stanford dependency parser. Then roles are extracted using lexical units associated with the root verb. Also the root verb is identified and all the frames for that verb are extracted from the FrameNet. The roles identified tries to fill all the FrameNet's frame. Then the filled frames are used for a semantic graph search, where other frames are identified for completing tasks. The search is based on specificity, where a specific child frame is discovered and sequentiality where sequences of frames are identified to complete the execution.

The use of frame from FrameNet make its coverage high and also domain independent, however the system needs lexical units for grounding the correct value for the frame elements. For example in command *Go to Bucks office.*, a lexical unit is needed which will search for the correct office. In the current implementation the command is equivalent to *Go to an office.* Also out of 10 different domains, considered for the evaluation of RoboFrameNet, the four domains which are related to the manipulation of the object were not completed. We also find object manipulation to be unreliable as of now.

Bibliography

[1] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE of*, 2, 1986.

[2] Rodney Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[3] Wiki community. from http://en.wikipedia.org/wiki/Roomba. 2013.

[4] Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical planning in the now. In *IEEE Conference on Robotics and Automation (ICRA)*, 2011.

[5] T. Regier and L. A. Carlson. Grounding spatial language in perception: an empirical and computational investigation. *Journal of Experimental Psychology: General*, 130(2):273–298, June 2001.

[6] Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. FrameNet II: Extended theory and practice. *Unpublished Manuscript*, 2006.

[7] E.H. Shortliffe. MYCIN: Computer-Based Consultations.

[8] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Interpreting robotic mobile manipulation commands expressed in natural language. *ICRA 2011 Workshop on Manipulation Under Uncertainty*, 2011.

[9] Stefanie Anne Tellex. Natural language and spatial reasoning, 2010.

[10] Brian J. Thomas and Odest Chadwicke Jenkins. Roboframenet: Verb-centric semantics for actions in robot middleware. In *ICRA*, pages 4750–4755, 2012.

[11] Yuan Wei, Emma Brunskill, Thomas Kollar, and Nicholas Roy. Where to go: Interpreting natural directions using global inference. In *ICRA*, pages 3761–3767. IEEE, 2009.

[12] T Winograd. Procedures as a representation for data in a computer program for understanding natural language. *Thesis, Massachusetts Institute of Technology*, 1971.

[13] T. winograd. from http://hci.stanford.edu/winograd/shrdlu/. 2013.

[14] Kai yuh Hsiao, Stefanie Tellex, Soroush Vosoughi, Rony Kubat, and Deb Roy. Object schemas for grounding language in a responsive robot, 2008.