

Speech based Control of Articulated Robots

2014-15 2nd Semester Project - CS365

Rahul Gurjar, Rohun Tripathi
Supervised by: Prof Amitabh Mukherjee

April 17, 2015

Abstract

This report presents the methodology used to create a model for speech based control of the articulated robots. Speech Recognition tool, Google ASR has been used for transcription of the audio commands to the robot. Stanford NLP parser is used to generate the parse tree of the command, which is used for Semantic Role Labeling as implemented by the Illinois' 'The Curator'. The action requested is extracted and is executed on the Aldeberan Nao robot. This report introduces each of the key areas of the project and a step by step implementation of each. To conclude, we discuss the results obtained and the future directions for this project.

Contents

1	Introduction	3
2	Literary Review	3
2.1	RoboFrameNet	3
3	Methodology	5
3.1	Module 1 : Google ASR	5
3.2	Module 2 : Stanford NLP Parser	5
3.3	Modules 3-4 : Semantic Role labeling - Action Parser	6
3.4	Module 5 : Robot Control	7
4	Experiments and Results	7
4.1	Error	8
5	Future Work	8
6	Conclusion	9

1 Introduction

Collaborative robotics is about humans and robots working together in the same space. For this we require an efficient and robust communication medium. Robots are no longer put behind fences (as in large manufacturing plants) and so there is a need to increase their capability to work collaboratively with users. However, programming a robot to do every task is quite tedious work and it is difficult to employ them for household or medical purposes. With the rapid development in other robot technologies, facilitating the usage of these advanced robots by non robotics experts in all walks of life has become a high priority.

Speech based control of robots is one such promising mode of communication. Speech comes naturally to humans and is a viable mode of communication between robots and non robotics experts. Further, the voice command delivered to the robot can be in a prescribed notation or in natural language. Prescribed notation has a major pitfall in its need for trained users, effectively destroying the objective to enable the non robotics experts. Thus, there is a need for a system that works on commands in natural language.

In order to use natural language for the input commands, we need to extract the semantic roles of the parts of the sentence and extract the key elements of the action. FrameNet is a dataset used for the this purpose, as described later in the report. FrameNet is a collection of frames, each describing a situation along with all its key elements. Each frame is tagged by a lexical unit and contains many of the possible dependencies of the lexical unit in a sentence. These dependencies are covered by frame's elements.

Humanoid robots are robots that posses a body shape similar to a human and thus, are easy to relate to for humans. Possible applications of humanoids include assisting in daily chores of the humans, delivery of various items, disaster rescue operations etc. The Aldebaran Nao is a one such humanoid robot. Easily available, this robot has been used for a vast range of applications such as detecting an object and pick and place operations on the object.

We have worked with the above vision for speech based control of robots in natural language and have developed such an end to end system simulated on the Nao Aldebaran. The rest of this report is divided in six sections. Section 2 discusses present state of work in this field, section 3 covers out methodology. Section 4 discusses the results of various experiments on out system followed by sections 5 and 6, covering future work and the Conclusion.

2 Literary Review

2.1 RoboFrameNet

Research in robotics systems has produced frameworks for robot middleware such as ROS, as developed by Quigley et al. [8], which has been used in several domains

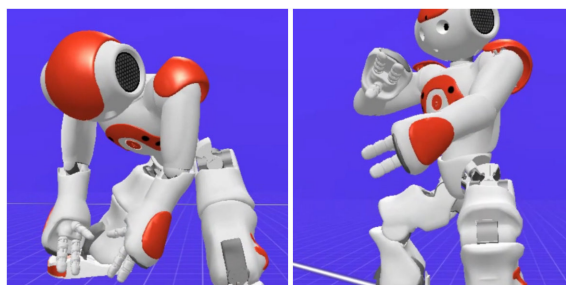


Figure 1: Picking action, an example motion by the Nao.

of modern robotics research. ROS is open-source platform for Robot software development which provides libraries and frameworks for robot applications. It possesses operating system like functionality with message-passing between processes. ROS architecture is similar to a graph where the processing happens in nodes. The Nao robot is a humanoid robot which can be simulated in ROS and also in external simulators using Choreograhe and WeRobots.

FrameNet [3] is a dataset of frames containing more than 10,000 word senses and more than 170,000 manually annotated sentences used for sense extraction from an input sentence. FrameNet is based on the model of Frame Semantics. A Semantic frame is a description of a situation or a type of event in its entirety, with all its elements and peripheral details. These frames can be used to extract the word sense.

Using natural language for robot control [9, 5] is a powerful tool for communication with a robot. The RoboFrameNet [9] is a HRI(Human Robot Interaction) based project and stack package for ROS authored by Brian J Thomas. The main idea of the project is verb-based semantics for actions which the robot is capable of doing. Semantic frame provides ground between natural language and robot actions. Natural language is parsed into verbs and their dependencies. These verbs plus their dependencies complete the semantic frame and are converted into ground actions according to scene depicted by semantic frame. Stack package for RoboFrameNet was developed for ROS-fuerte (2012) version ROS and is no-longer available.

Automatic semantic role labeling(SRL) [6, 4] systems provide semantic roles to parts of a sentence provided in Natural language. Work in SRL's was pioneered by Gildea and Jurafsky, [6] in 2002, and semantic role labeling was treated as a tagging problem on each constituent in a parse tree, solved using an argument identifier and an argument classifier. University of Illinois' SRL package, [4] is a part of their 'Curator' system, which encompass many NLP functionalities and is trained using the PropBank dataset.

We intend to implement an end-to-end system, from speech recognition to action implementation and in the process extend Nao's [2] action set.

3 Methodology

The method we applied to build our end to end system constitutes five modules, as described by the following flow chart. In each of the following subsections, we discuss one of these modules.

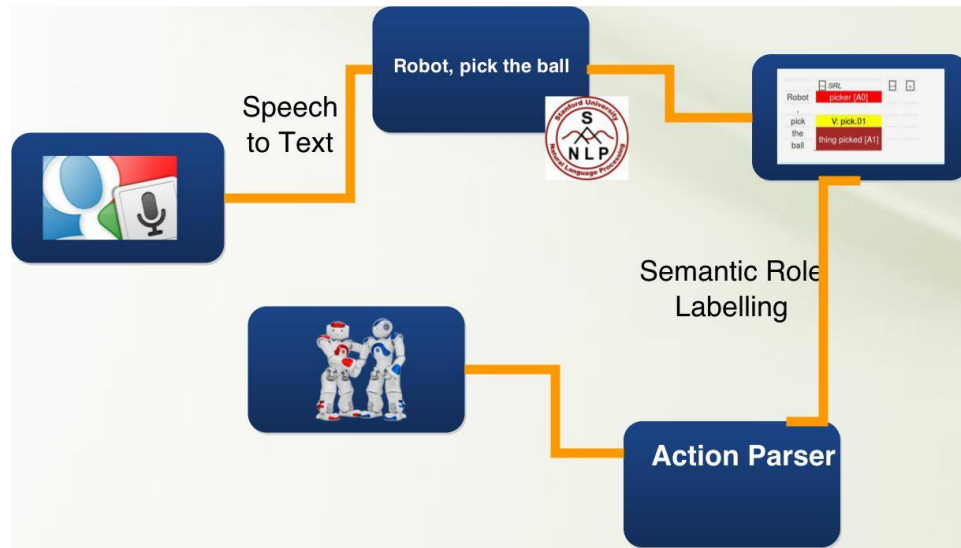


Figure 2: The five modules that describe our end to end system

3.1 Module 1 : Google ASR

Google ASR [1] provides us with the text form of the input voice command. To utilize the Google web speech API, we initialize a HTTP Post request with the audio input in .flac format. The API returns transcription with a confidence value for the same. A threshold has been set on this confidence value, on whether to accept the command. In case the value is below the threshold, the user is asked to input the command again. We worked a with threshold value of 0.75.

```
[{"result":[{"alternative":[{"transcript":"robot pick the ball","confidence":0.85152686}], "final":true}], "result_index":0}]
```

Figure 3: Transcription as provided by the Speech module, for command 'Robot, pick the ball'

3.2 Module 2 : Stanford NLP Parser

The output obtained from the Speech API is run through the Parser for generating the parse tree. This parse tree is further used by the SRL module. The NLP Parser [7] is

based on a Probabilistic Context Free Grammar and trained on the FrameNet Dataset. Probabilistic parsers use knowledge of language gained from hand-parsed sentences (from the FrameNet) to try to produce the most likely analysis of new sentences provided in our module.

```
(ROOT
  (S
    (VP (VB Pick)
      (VP
        (ADVP (RB up)
          (NP (DT the) (NN spoon))
          (PP (IN from)
            (NP (DT the) (NN table))))))
```

Figure 4: Parse Tree as generated by the NLP module, for command 'Pick up the spoon from the table'

3.3 Modules 3-4 : Semantic Role labeling - Action Parser

The FrameNet corpus is a database of semantic frames and manually annotated text which is frequently used for training the Semantic Role Labeler (SRL). Building an SRL is traditionally done in a two-stage architecture consisting of an argument identifier and an argument classifiers for the action verbs in the sentence. The Curator extends the SRL building process by adding a pruning module before the argument identifier. Pruning filters out simple constituents unlikely to be arguments and the inference module runs after the classification stage and incorporates global information and enforces constraints that reduce any overlap of arguments of different verb.

srl View

```
Predicate pick [sense: 01] [predicate: pick]
  <A0> Argument Robot
  <A1> Argument the ball
```

Figure 5: Labeled output as generated by the SRL module, for command 'Robot, pick the ball'

In module 4, the output from the SRL is used to create the corresponding action file for the Nao. This is done using a code to map the arguments provided for the verbs to the labels of the verb required by the Nao to perform that specific action. The present state of this module only allows one action verb per command to the robot. Development of this module and inspection of its robustness is a possible extension of this project.

3.4 Module 5 : Robot Control

For robot control, we started by trying to use the whole RoboFrameNet package on ROS but it is no-longer available for newer versions of ROS. Then we decided of implementing Robot action through ROS for PR2 robot. ROS has many stacks and packages for PR2 navigation and control of its arms but only few of them were up-to-date and after extensive research we got to know ROS-Indigo (latest version of ROS) doesnt have many running or workable nodes of PR2.

As a result, we moved to Aldebaran Nao as our target robot. Aldebaran Nao has library support in softwares, Choregraphe and Webots which we used for programming, simulation and testing. As we are required to extend the action set, we implemented various tasks like moving to predefined location, turning in place and picking a ball. Voice commands for the same were prepared by us. The following table is an exhaustive list of the actions and also a few sample voice commands for each of these actions. A link of the Nao executing these actions is here : <https://www.youtube.com/watch?v=CX6YKurqRq4>.

VERB	Successful Sample Commands	Failed Commands
PICK	Robot Pick the ball	Nao pick the ball
PLACE	Robot place the Spoon on the table	Place the spoon
MOVE TO ORIGIN	Robot move to the origin	-
TURN	Robot turn left	Robot turn 60° left
SIT DOWN	Robot sit down	(Only stability problems)
STAND UP	Robot stand up	(Only stability problems)
WAVE	Robot wave	-
RAISE ARM	Robot raise your arm	Raise your arm, Nao
WIPE FOREHEAD	Robot wipe your forehead	-

Figure 6: Actions of the Nao Robot we implemented using speech control.

4 Experiments and Results

In this section we discuss the results of some experiments on our system. In each of the experiments, we presented to the robot simple sample commands such as 'Robot pick the ball' and 'Robot move to origin'. These commands were restricted to only one major verb and executed successfully.


Speech	Google ASR Output	SRL Output	Robot Action
"Robot pick the ball"	['robot pick the ball', confidence:0.8515]	[predicate:pick] Argument1:Robot Argument2:the ball	

Figure 7: Results at each step for a sample command : 'Robot, pick the ball'

4.1 Error

Experiments that failed usually related to an error in one of the modules. A major issue was the error propagating in the system to all the modules after the module at which it occurs. The following are some instances of error generated by different modules during executions:

- Google ASR : Some sample outputs failed such as 'Nao pick the ball'. It was wrongly transcribed as 'Now, pick the ball' and lacked a root
- Semantic Role Labeling : Commands like 'Pick the ball' failed as the parser couldn't interpret the 'robot' as subject of the statements.
- Nao Robot : Execution of commands which required the Nao to bend could result in a failure (actions like picking off the ground)

5 Future Work

This project opens up new horizons on which work can be pursued. Here is a list of a few possible future applications of this project.

- As discussed in the poster presentation, we have extended the action set of the Nao Robot. This is the most important future work of this project, to further extend the action set of the Nao robot in our system.
- Presently, we modify the input data to make sure each input command is governed by only one major verb. This can be extended, quite easily, to take complex sentences as input.

6 Conclusion

In this report, we have presented an end to end system comprising five modules to control articulated robots using speech controls and tested our system on a simulated Aldebaran Nao robot. The first module converts audio to text and employs Google ASR. The second and third generate the semantic roles of the parts of the sentence for execution. The fourth module prepares the command for the Nao robot from the SRL output and the fifth implements the action. Each module in our system presents its own level of error, which propagates through the system affecting modules following it. Improvements to our system require improvements in accuracy at each module along with implementation of the future works discussed earlier.

This video : <https://www.youtube.com/watch?v=CX6YKurqRq4> captures the Nao executing a few of the commands from our system.

References

- [1] Google voice recognition.
- [2] Aldebaran. Nao robot: intelligent and friendly companion.
- [3] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project, 1998.
- [4] J. Clarke, V. Srikumar, M. Sammons, and D. Roth. An nlp curator (or: How i learned to stop worrying and love nlp pipelines), 2012.
- [5] Luke Zettlemoyer Dieter Fox Cynthia Matuszek, Evan Herbst. Learning to parse natural language commands to a robot control system, 2012.
- [6] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles, 2002.
- [7] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing, 2003.
- [8] Morgan Quigley. Ros: an open-source robot operating system, 2009.
- [9] Brian J Thomas and Odest Chadwicke Jenkins. Roboframenet: Verb-centric semantics for actions in robot middleware, 2012.