Department of
Robotics and Mechatronics Engineering

University of Dhaka

RME 410: Dissertation

Robot Navigation from Natural Language Instructions

Submitted by :
Foysal Khandakar Joy
Roll: 410

Harish Pasha Dipto
Roll: SH-092-013

Supervised by:
Sujan Sarker
Lecturer
Dept. of Robotics and
Mechatronics Engineering
University of Dhaka

Acknowledgment

Thanking is the best way to appreciate ourself for performing a great task. The reason We can claim that We are successful in completing our work with such ease is that the Almighty gave us the ability, capability, chance, cooperating hands, and enough stamina to walk on this path. Almighty enriched us with very good luck on this path of work.

The reason We are talking about luck is that We had our respected teacher Sujan Sarker as our supervisor in this work and We would like to take the opportunity to express our sincere gratitude and indebtedness to our respected supervisor. Although he is a self-motivated and hard-working person always loaded with research and administrative works, he gave us more than enough time in this work.

We are thankful to the Bsc thesis committee members for their suggestion, corrections and guidelines. It is worth mentioning that, their help and support have influence us reorganize our work, furnish our ideas, find out errors and correcting them.

Foysal Khandakar Joy
Harish Pasha Dipto

Signature of Supervisor

Contents

Chapter 1

Introduction

Artificial intelligence is playing more prominent role in our daily lives by demonstrating human intelligence by machines, especially computer systems. The field includes learning, reasoning and self-correction. These can be described as the acquisition of information and rules for using the information, using the rules to reach approximate or definite conclusions. It has become more important to make machines able to communicate through natural language with human or among machines themselves. Learning algorithms for natural language understanding in language translation, reading comprehension have progressed at a rate in recent years that never done before, but that lack ultimate aspects of how humans understand and produce natural language. Mainly humans develop language understanding and producing by being embodied in an environment which they can realize and interact with other humans [4].

In many tasks understanding compositional language in context is very complex. Reasoning about sets of objects, quantities, comparisons and spatial relations are required in visual question answering and robot instruction systems. Robust language understanding is required when instructing assembly-line or home assistance robots to manipulate objects in random environments. And this is only partially addressed by existing datasets [10].

Since the early days of artificial intelligence, the problem of interpreting instructions written in natural language has been widely studied. The automation of tasks that currently require human participation would be enabled by mapping instructions to a sequence of executable actions [9].

"Natural Language" refers to a human language that is used for everyday communication by humans; languages like English, Bengali or Portuguese as distinct from the typically artificial command. Artificial language like programming language and mathematical transcripts, natural languages have evolved over generations, and hard to keep down with explicit rules. NLP based technologies are becoming increasingly widespread. For example, phones and handheld computers support text suggestion and handwriting recognition;web search engines provide access to information locked up in noisy text data;machine translation allows us to recover written texts in different language and read them in another language; text analysis enables us to classify sentiment in different reviews or blogs post. By providing a more natural human-machine interfaces and more sophisticated access to stored data, a central role is played by language processing in multilingual information society [3].

It is straightforward to induce our hands on countless words of text. What will we have a tendency to do with it, forward we are able to write some easy programs? We're all terribly accustomed to text, since we have a tendency to scan and write it on a daily basis. Here we'll treat text as data for the programs we have a tendency to write, programs that manipulate and analyze it in an exceedingly type of attention-grabbing ways.

1.1 Robot Navigation

Navigation refers to the strategy of crucial aspects like position, speed, and direction throughout travel. within the pre-modern era, direction associate degreed position were determined mistreatment an measuring instrument, a compass, and a map; these square measure currently thought of primitive kinds of navigation. As a results of fashionable developments in science and technology, precise positions and speeds square measure determined mistreatment instrumentation like artificial satellites, international navigation satellite system (GNSS), direction systems (INS), etc [7].

1.2 Automatic Natural Language Understanding

At a strictly sensible level, we tend to all want facilitate to navigate the universe of knowledge fast up in text on the net. Search engines are crucial to the expansion and recognition of the net, however have some shortcomings. It takes talent, knowledge, and a few luck, to extract answers to such queries as: *What tourist sites can I visit between Philadelphia and Pittsburgh on a limited budget?What do experts say about digital SLR cameras? What did the trusted commentators predict about the steel market last week?* Getting a machine to answer them automatically involves a variety of language process tasks, as well as info extraction, inference, and account, and would want to be dispensed on a scale and with grade of hardiness that's still on the far side our current capabilities.

On a a lot of philosophical level, a long-standing challenge at intervals AI has been to create intelligent machines, and a serious a part of intelligent behavior is knowing language. for several years this goal has been seen as too troublesome. However, as information science technologies become a lot of mature, and sturdy strategies for analyzing unrestricted text become a lot of widespread, the prospect of language understanding has re-emerged as a plausible goal [3].

In this section we tend to describe some language understanding technologies, to allow a way of the attention-grabbing challenges that area unit associated with NLP.

1.2.1 Word Sense Disambiguation:

In word meaning disambiguation we wish to figure out that sense of a word was supposed in an exceedingly given context. Contemplate the ambiguous words serve and dish:

a. serve: help with lunch or drink; hold an office; put football into play

b. dish: glass; course of a meal; communications devices

In a sentence containing the phrase: he served the dish, square measure able to notice that each serve and dish are being employed with their food meanings. It's unlikely that the subject of dialogue shifted from sports to crockery within the space of 3 words. This might force us to create outre pictures, sort of a professional tennis player removing his or her frustrations on a china tea-set arranged out beside the court. In alternative words, we have a tendency to automatically clear up words exploitation context, exploiting the straightforward incontrovertible fact that close words have closely connected meanings. As another example of this discourse result, take into account the word by, that has many meanings, e.g.: the book by Russel (agentive – Russel was the author of the book); the match by the stove (locative – the stove is where the match is); and submit by Saturday (temporal – Saturday is the time of the submission). Observe in (c) that the meaning of the italicized word helps us interpret the meaning of by.

a. The lost women were found by the searchers (agentive)

b. The lost women were found by the mountain (locative)

c. The lost women were found by the afternoon (temporal)

1.2.2 Pronoun Resolution

A deeper reasonably language understanding is to figure out ”who did what to whom” — i.e., to observe the subjects and objects of verbs. You learnt to try and do this in grammar school, however it's tougher than you may assume. In the sentence the thieves stole the paintings it's simple to detect who performed the stealing action. Consider 3 doable following sentences in (c), and check out to see what was sold, caught, and found (one case is ambiguous).

a. The thieves stole the paintings. They were subsequently sold.

b. The thieves stole the paintings. They were subsequently caught.

c. The thieves stole the paintings. They were subsequently found.

Answering this question involves looking for the preposition of the pronoun they, either thieves or paintings. One of the calculation techniques to address this problem includes anaphora resolution – identifying what a pronoun or noun phrase means – and labeling semantic role – identifying the relation between a noun phrase and the verb (as agent, patient, instrument, and so on).

1.2.3 Generating Language Output

If we are able to automatically solve such issues of language understanding, we are going to solve the tasks that involve generating language output, like responding to a question and translating language from one form to another. In the initial case, a machine ought to be ready to answer a user's query regarding to text collection.

a. Text: ... The thieves stole the paintings. They were subsequently sold. ...

b. Human: Who or what was sold?

c. Machine: The paintings.

The answer of machine proves that it has correctly worked out that *they* doesn't refer to the thieves but to paintings.

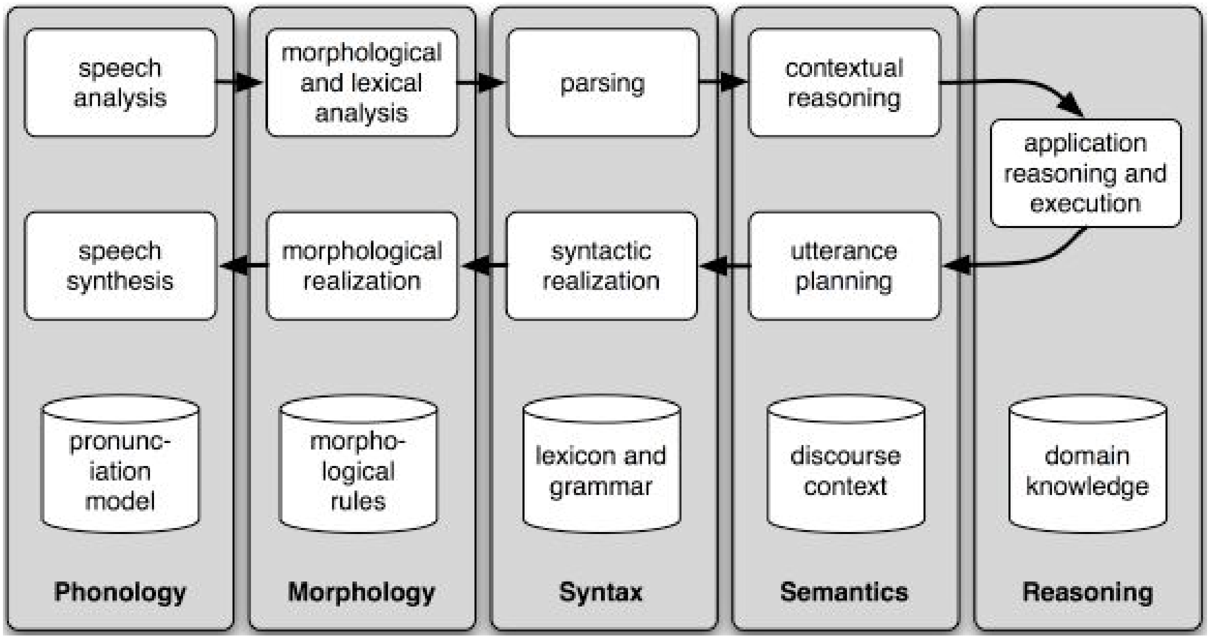


Figure 1.1: Simple Pipeline Architecture for a Spoken Dialogue System[3]

1.2.4 Spoken Dialog Systems

In the study of past of artificial intelligence, the main measure of intelligence has been a linguistic, Turing test: if a dialog system, responding to a user's text input, can perform so naturally that we can't differentiate it from human-generated reactions. Can't i? On the contrary, today's commercial dialog systems are very limited but still perform useful functions in short-term domains, as we can see here:
S: How may I help you?
U: Deliver some food from the list to Shahidullah Hall after buying these from TSC.
S: Buying food from TSC first and then deliver these to Shahidullah Hall?
U: Yes.

We could not ask the machine to produce driving directions or details of nearby restaurants unless the specified info had already been hold on and appropriate question-answer pairs had been incorporated into the language processing system.

Observe that this system understands the sequence of working plan based on the instruction: the user tells an instruction in complex way and the system correctly determines from this that the user wants to deliver some food but it needs to buy these foods first. This assumption seems so obvious that you probably didn't notice that it was trained, yet a natural language system needs to be completed with this skill to communicate naturally. WIthout it, when asked *Deliver some food from the list to Shahidullah Hall after buying these from TSC.*, a system might plan to deliver first and to buy food from TSC. Usually contextual assumptions and business logic is used in commercial dialogue systems to ensure that a user can request or provide information that pays for specific applications.

In Figure 1.1, *Spoken signal (top left) is taken, speech is recognized, words are parsed and taken in context, application-specific actions occur (top right); a reaction is planned, is realized as a syntactic structure, then to the words that are properly replaced and finally to the spoken output; The different types of linguistic knowledge inform each level of the* *process.*

Dialogue systems give us the fundamental pipeline for NLP. Figure 1.1 shows the architecture of a normal dialog system. One of the few language-understanding elements that move left to right at the top of the diagram is ”Pipeline”. This map is a kind of money-making from the spicinput through the syntactic parsing. In the middle, the opposite pipeline of elements to convert the concept from right to left. These elements create dynamic aspects of the system. Below the diagram there are a few representative organizations of static information: the collection of language-related data for processing materials to do their job.

1.2.5 Information Extraction

With rise of digital age, there’s associate degree explosion of data within the sort of news, articles, blogs, social media, and so on. A lot of of this knowledge lies in unstructured type and manually managing and effectively creating use of it’s tedious, boring and labor intensive. This explosion info of data of knowledge and want for a lot of subtle and economical information handling tools provides rise to data Extraction(IE) and knowledge Retrieval(IR) technology. data Extraction systems takes linguistic communication text as input and produces structured data mere by bound criteria, that’s relevant to a selected application. numerous sub-tasks of i.e. like Named Entity Recognition, grammatical relation Resolution, Named Entity Linking, Relation Extraction, mental object reasoning forms the building blocks of assorted high finish linguistic communication process (NLP) tasks like MT, Question-Answering System, linguistic communication Understanding, Text summarisation and Digital Assistants like Siri, Cortana and Google Now [8].

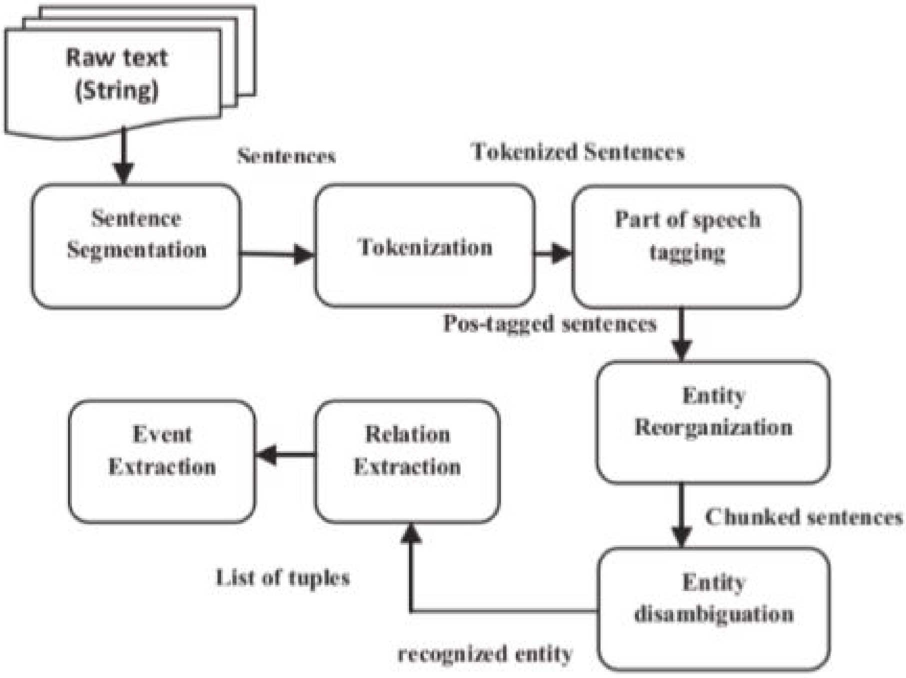


Figure 1.2: Simple Pipeline Architecture for an Information Extraction System [8].

Figure 1.2shows the design for a straightforward info extraction system. It begins by process a document victimization many of the procedures: 1st, the raw text of the document is split into sentences employing a sentence segmenter, and every sentence is any divided into words employing a tokenizer. Next, every sentence is labeled with part-of-speech tags, which can prove terribly useful within the next step, named entity detection. during this step, we tend to look for mentions of probably attention-grabbing entities in every sentence. Finally, we tend to use relation detection to look for doubtless relations between completely different entities within the text.

1.3 Task Definition

In our project, we are developing an agent that will follow Bengali instructions to navigate in real-life visual environment.

- Inputs: Text based instructions in Bengali language.
- Outputs: Mapping instructions and visual observations to actions and execute them in the environment.

1.4 Motivation

Computers are good at operating with structured information like spreadsheets and info tables. However we humans typically communicate in words, not in tables. That's unfortunate for computers. Loads of data within the world is unstructured raw text in English or another human language. however will we have a tendency to get a pc to know unstructured text and extract information from it? Advances in AI are sanctioning increasingly more refined, capable technologies to achieve massive client populations. Such systems provide unprecedented potential for AI to assist in a very type of human-centric applications like elder care and home maintenance. However, natural, easy-touse interfaces to such systems, like those using linguistic communication, square measure insulation behind. As robots become additional prevalent—and because the would like for the services they will provide grows—the importance of permitting non-expert users to act with them naturally and well will increase. linguistic communication is a superb modality for finish users to offer directions and teach robots concerning their environments [5].

As long as computers are around, programmers are attempting to write down programs that perceive languages like English. the rationale is pretty obvious humans are writing things down for thousands of years and it'd be extremely useful if a laptop might scan and perceive all that knowledge. NLP is very important for scientific, economic, social, and cultural reasons. NLP is experiencing rapid climb as its theories and strategies square measure deployed in a very style of new language technologies. For this reason it's necessary for a good vary of individuals to possess a operating data of NLP. among business, this includes folks in human-computer interaction, business data analysis, and internet software system development. among academe, it includes folks in areas from humanities computing and corpus linguistics through to engineering and computing [3].

1.5 Objectives

By developing the project, we will learn:

- To manipulate large corpora, explore linguistic models and analyze language data.
- To use the key concepts from NLP and linguistics to describe and analyse language.
- To use data structures and linguistics algorithms in robust language processing software.
- To extract knowledge from natural language.
- To map instructions from natural language to actions.
- To deploy an intelligent agent to execute actions in a particular visual environment.

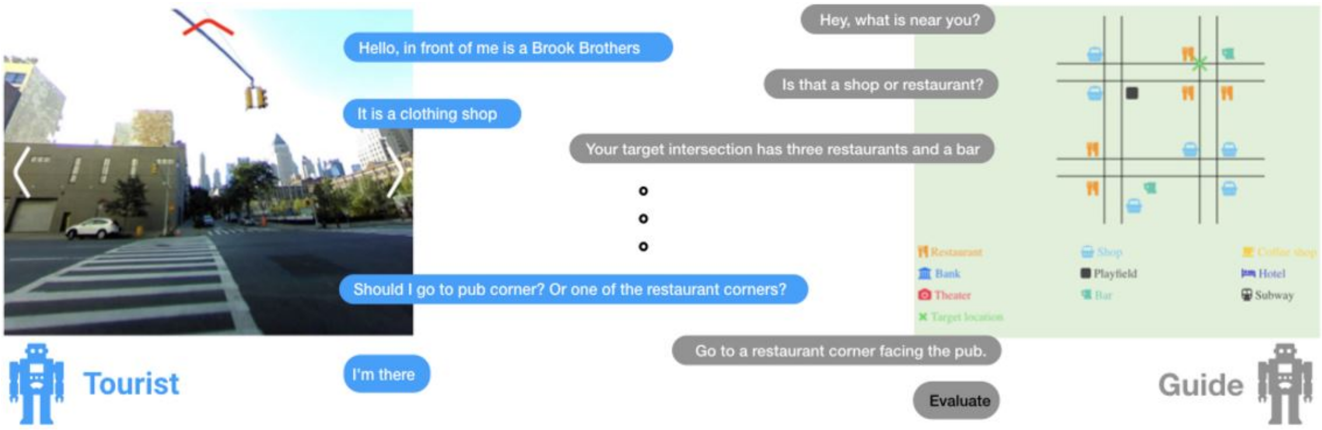
Chapter 2

Related Works

Navigation needs the agent to reason regarding its relative position to things and the way these relations modification because it moves through the setting. whereas in each learning needs counting on indirect oversight to accumulate spatial information and language grounding, for navigation, the coaching knowledge includes incontestable actions, and for spatial description resolution, annotated target locations. we've studied the matter of reasoning regarding linguistic communication directions to navigate and located some works associated with our project.

2.1 Talk The Walk

In [4], they introduce the Talk the Walk, wherever the aim is for 2 agents, a "guide" and a "tourist", to communicate with one another via natural language so as to attain a standard goal: having the tourist navigate towards the right location. The guide has access to a map and is aware of the target location, however doesn't have idea wherever the tourist is; the tourist features a 360-degree read of the environment, however is aware of neither the target location on the map nor the path to it. The agents got to work along through interaction so as to solve the task successfully. Associate degree example of the task is given in Figure 2.1.



Hey, what is near you?

Hello, in front of me is a Brook Brothers

Is that a shop or restaurant?

It is a clothing shop

Your target intersection has three restaurants and a bar

Should I go to pub corner? Or one of the restaurant corners?

Go to a restaurant corner facing the pub.

I'm there

Figure 2.1: Example of the Talk The Walk task [4]

In Figure 2.1: in order to help the tourist navigate towards the exact location a "guide" is interacting with the "tourist" via natural language. The target location is known to the guide but not the location of the tourist. The guide also has access to the map but the tourist doesn't have. Finally the tourist navigate through a 360-degree view of the street environment.

Example:

Guide: Hello, what are surrounding you?
Tourist: ACTION:TURNLEFT ACTION:TURNLEFT ACTION:TURNLEFT
Tourist: Hello, in front of me is a Brooks Brothers
Tourist: ACTION:TURNLEFT ACTION:FORWARD ACTION:TURNLEFT ACTION:TURNLEFT
Guide: Is that a coffee shop or hotel?
Tourist: ACTION:TURNLEFT
Tourist: It is a stationary shop.
Tourist: ACTION:TURNLEFT
Guide: You need to move to the junction in the northwest corner of the map
Tourist: ACTION:TURNLEFT

Talk The Walk is the first task to combine all three aspects together: in order to observe the environment the tourist first do perception, a dialogue system for achieving common goal through interaction via natural language, action for the tourist to navigate through the environment.

As the main focus of their task is on interactive dialogue, they limit the difficulty of the control problem by having the tourist navigating a 2D grid via discrete actions (turning left, turning right and moving forward) [4].

2.2 Mapping Instructions and Visual Observations to Actions

In [6], they propose to directly map raw visual observations and text input to actions for instruction execution. While existing approaches assume access to structured environment representations or use a pipeline of separately trained models, they learn a single model to jointly reason about linguistic and visual input. They use reinforcement learning in a contextual bandit setting to train a neural network agent. To guide the agent's exploration, they use reward shaping with different forms of supervision.

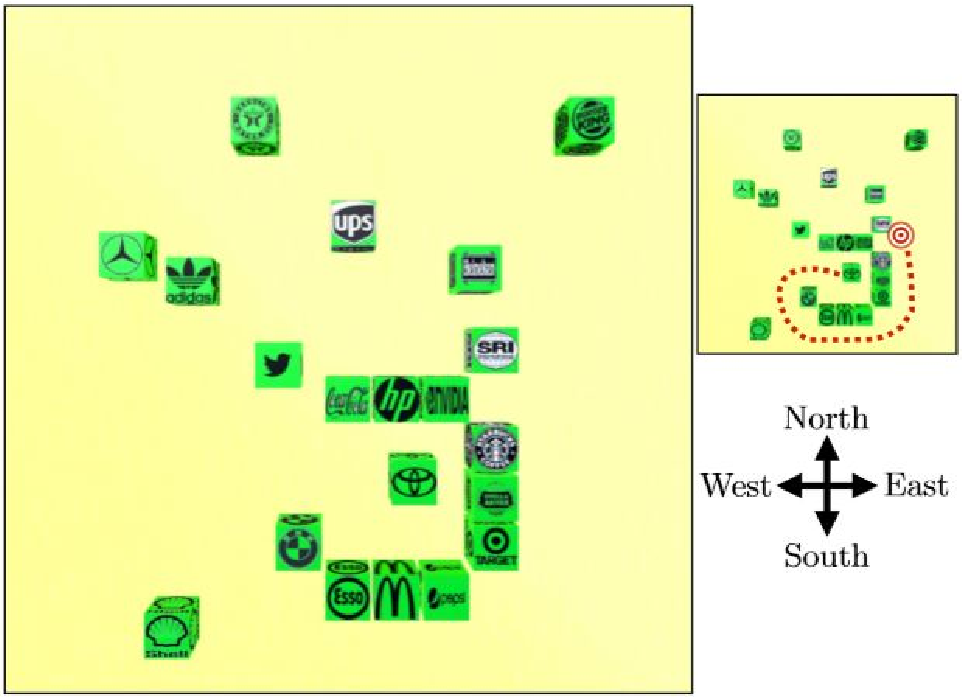


Figure 2.2: Blocks environment [6]

Put the Toyota block in the same row as the SRI block, in the first open space to the right of the SRI block.
Move Toyota to the immediate right of SRI, evenly aligned and slightly separated.
Move the Toyota block around the pile and place it just to the right of the SRI block.
Place Toyota block just to the right of The SRI Block.
Toyota, right side of SRI.

Table 2.1: Instructions in the Blocks environment [6]

In Table 2.1, illustrates the matter within the Blocks surroundings. The agent observes the surroundings as RGB image employing a camera sensing element. Given the RGB input, the agent should acknowledge the blocks and their layout. To grasp the instruction, the agent should determine the block to maneuver (Toyota block) and also the destination (just right side of the SRI block). This needs determination linguistics and grounding issues. For instance, contemplate the top instruction within the figure. The agent has to determine the phrase pertaining to the block to maneuver, Toyota block, and ground it. It should resolve and ground the phrase SRI block as a reference position, that is then changed by the spatial which means recovered from a similar row as or initial open area to the correct of, to spot the goal position. Finally, the agent has to generate actions, for instance moving the Toyota block around obstructing blocks.

To address these challenges with one model, they create a neural network agent. The agent executes directions by generating a sequence of actions. At every step, the agent takes as input the instruction text, observes the planet as RGB image, and selects consequent actions. Action execution changes the state of the environment. Given observation of the new environment state, the agent selects consequent action. This method continues till the agent indicates execution completion. Once choosing actions, the agent together reasons concerning its observations and also the instruction text. This permits selections supported shut interaction between observations and linguistic input.

They study the problem of learning to execute instructions in a situated environment given only raw visual observations. Supervised approaches do not explore adequately to handle test time errors, and reinforcement learning approaches require a large number of samples for good convergence. Their solution provides an effective combination of both approaches: reward shaping to create relatively stable optimization in a contextual bandit setting.

This combination is designed for a few-samples regime, as we address. When the number of samples is unbounded, the drawbacks observed in this scenario for optimizing longer term reward do not hold.

2.3 Vision-and-Language Navigation

The idea in [2] that they might be able to provide general, verbal directions to a robot and have a minimum of an affordable likelihood that it'll do the specified task is one in all the long-held goals of artificial intelligence, and robots. Despite vital progress, there area unit variety of major technical challenges that require to be over precede robots are going to be able to perform general tasks within the universe. One in all the first needs are going to be new techniques for linking language to vision and action in *unstructured, previously unseen environments.* They refer it as Vision-and-Language Navigation (VLN) because of the navigation version of this challenge [2].

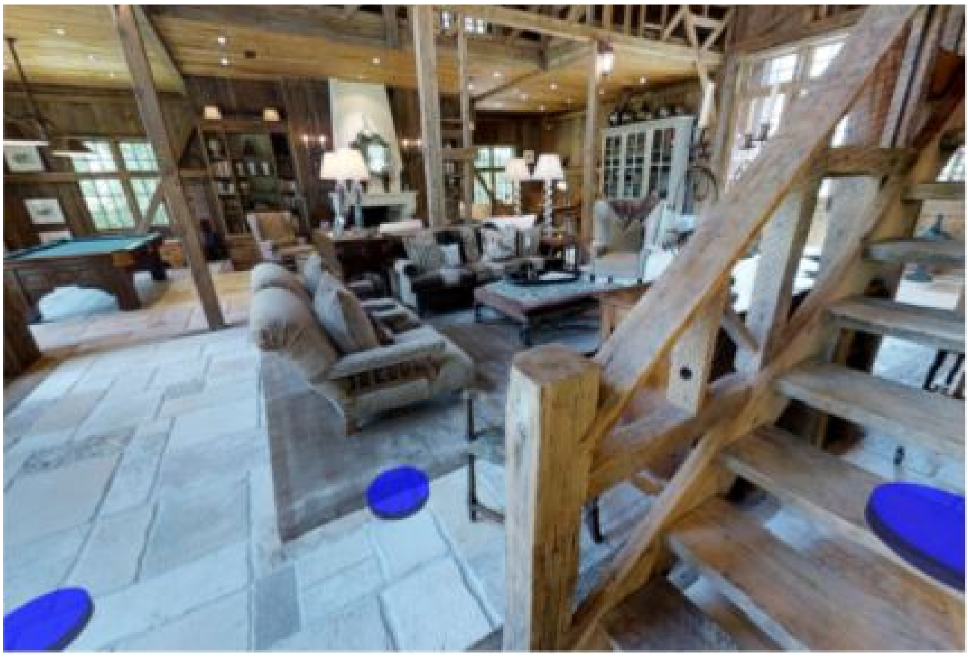


Figure 2.3: Room-to-Room (R2R) navigation task [2]

Instruction

Head upstairs and walk past the piano through an entryway directly before. Turn right once corridor ends at footage and table. Wait by the moose antlers hanging on the wall.

Previous approaches to natural language command of robots have often neglected the visual information processing aspect of the problem. The R2R dataset is the first dataset to evaluate the capability to follow natural language navigation instructions in previously unseen real images at building scale. To explore this task they investigated several baselines and a sequence-to-sequence neural network agent. The process used to generate R2R is applicable to a host of related vision and language problems, particularly in robotics.

2.4 Enabling Robots to Understand Incomplete Natural Language Instructions Using Commonsense Reasoning

In [11], they introduce Language-Model-based commonsense Reasoning (LMCR), a replacement technique that permits a robot to concentrate to a linguistic communication instruction from a person's, observe the surroundings around it, and mechanically fill in info missing from the instruction mistreatment environmental context and a replacement common sensible reasoning approach. Their approach initial converts associate degree instruction provided as free linguistic communication into a type that a robot will perceive by parsing it into verb frames.

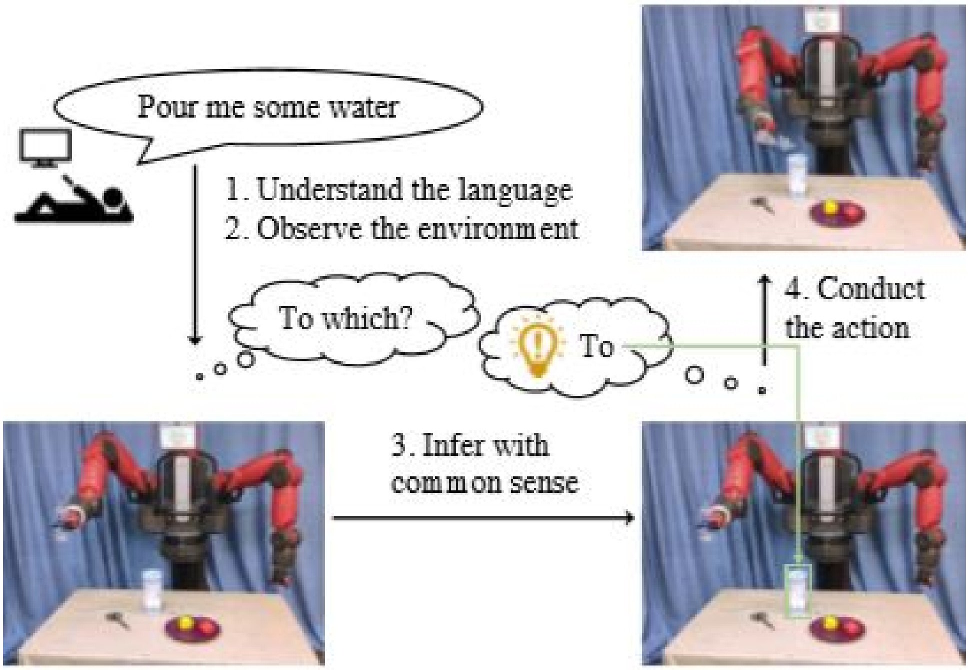

Pour me some water
1. Understand the language
2. Observe the environment
To which?
To
4. Conduct the action
3. Infer with common sense

Figure 2.4: A natural language controlled robot employing commonsense knowledge to interpret an instruction with missing information [11]

Their approach then fills in missing data inside the instruction by observant objects in its neck of the woods and finance common wise reasoning. To be told common wise reasoning automatically, their approach distills data from huge unstructured matter corpora by coaching job a language model. Our results show the reasonableness of a automaton learning common wise data automatically from web-based matter corpora, and additionally the facility of learned common wise reasoning models in serving to a automaton to autonomously perform tasks supported incomplete communication directions.

A person provide instruction “pour him some water” however the automaton cannot perform the action while not knowing wherever to pour. When scanning the surroundings, the robot uses common sensible information to work out the missing parameters and with success perform the action.

Chapter 3

Methodology

The work that we proposed here for navigating a robot using natural language based instruction has 3 major sections. They are:

- Web Environment
- Dataset and the Model
- Map for Providing Output

Now we'll go through the elaboration of these sections.

3.1 The Web Environment

The main work that we've done to demonstrate our task is designing a web interface. Initially this runs inside local server but it could easily be deploy-able to actual web server. We design this web interface so that the module could easily be used in real life scenario.

An image of the designed interface is as follow.

The functionality we added inside the interface is:

- Taking voice input
- Converting voice to text
- Sending converted text to the model
- Showing output on map from the output of the model

Now we'll go through details of these functionalities.

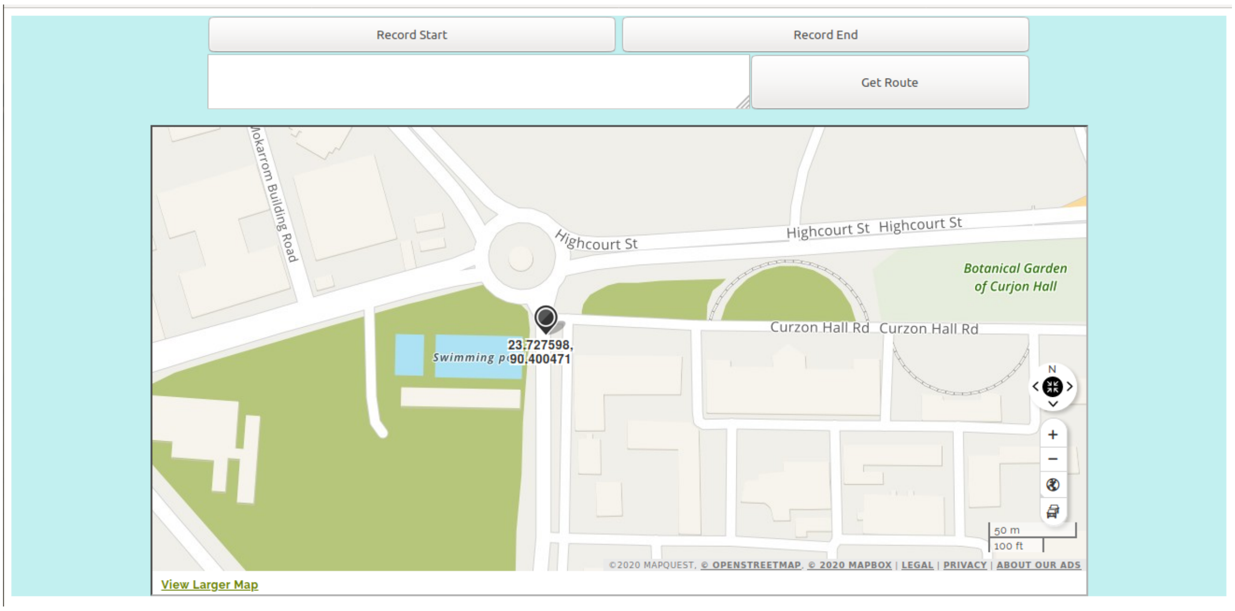


Figure 3.1: Used Web Interface

3.1.1 Taking Voice Input

As we can see in the figure above, there are two buttons: ‘Record Start’ and ‘Record End’. The name of the buttons describe the functionalities. If we click on the button ‘Record Start’ then we go to the recording mode. After entering the recording mode we speak to the connected microphone and give navigation instruction in language that we use our day to day life.

There is no time limit added with voice input. It means as long as we want we can give instruction. What happened here is we basically record the voice which is spoken.

3.1.2 Converting Voice to Text

When our instruction ended, we will click on ‘Record End’ button. After pressing the button the recording will stop which was started when we pressed ‘Record Start’ button. After the audio recording ended, we saved the audio in WAV format.

Immediately we convert the audio to the text. For this conversion we used Google Cloud Speech-to-Text service. There are several other ways to convert the voice into text but we find Google’s service more reliable and accurate. As converting voice to text is a huge task which takes much time and large dataset so that we didn’t develop our own version of voice-to-text. Competing with Google’s dataset and algorithm is also a great challenge. As our project is neither voice-to-text conversion focused so we used Google’s service here.

After converting the audio to text, the converted text appeared in the text box in our interface. We kept the option to further modify the text. Though the converted voice mostly accurate but sometimes some Bengali names spoken in Bangladeshi accent becomes difficult to convert in English. If some words converted wrongly so that we can manually edit it for our navigation purpose.

3.1.3 Sending Converted Text to Model

After conversion and/or necessary modification done, if we press the 'Get Route' button, then the text is sent to the model which is already saved in local server. We trained the model previously and saved the tokenizer and the model in our server. Once we clicked on the button then the text is first tokenized and then applied on the model.

3.1.4 Showing Output on the Map from the Output of the Model

After applying the text on the model, the model gives us the list of the places sequentially that is said to be navigated in the text. Based on the places list, the required path is generated and shown on the map.

3.2 Dataset and Model

Dataset and Model are required for learning purpose. First have a look where and for what purpose we used learning.

As we said we'll navigate a robot using natural language instruction, so understand the natural language instruction we'll use machine learning method. So the basic question is, what was the main challenge we encounter so we pick the learning method?

As we want to enable our robot/agent to navigate according to natural language based instruction, so our robot/agent should have the ability to understand all kind of instruction. As instruction format could be different from person to person like some person may use complex sentence structure, some person may could use complex grammar or some person may could use very simple instruction. As we don't know how our input instruction would look like so that we can't use naive string matching type algorithm.

Lets consider an example. Lets we have an instruction: 'Go to point A through point B'. In this instruction the robot has to go to point B first then point A. The same instruction could be in another format: 'Go to point B first then go to point A'. The robot has to accomplish same task but it is given in different format. The given instruction could be even further complex where the robot may have to navigate through three or more points. If we try to implement a 'rule based model' like if we got this type of sentence structure then we'll do this else we'll do something different, this type of approach will be very naive and difficult to implement. Furthermore, the robot won't have the capability to understand a new instruction what it never seen before.

So situation like described above need the approach of machine learning. So here instead of implementing rule based model we implemented a learning approach for robot navigation.

3.2.1 Dataset

As we've already described why we are using learning. For learning, we need a dataset. Here we'll look how we've designed our dataset.

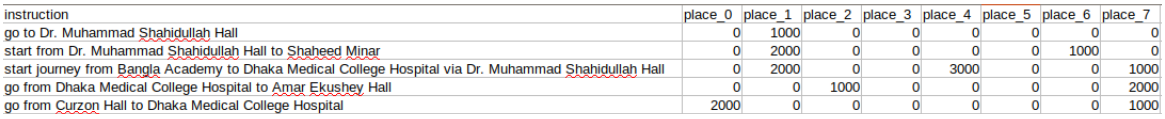
We are navigating through 8 points on Dhaka University campus. This points are: Curzon Hall, Dr. Muhammad Shahidullah Hall, Amar Ekushey Hall, Fazlul Haque Muslim Hall, Bangla Academy, TSC, Shaheed Minar and Dhaka Medical College Hospital. So our navigation instruction will be based on these eight points. In other word, all of navigation instruction that we included in our dataset have at last one of these points.

As we'll look in next section, the model we used here is basically a classifier. The classifier will give us the place names and their order, which place have to navigate before which place.

Our dataset has nine columns. First column holds the instruction. The other eight columns are marked as: 'place0' through 'place7'. 'place0' represent Curzon Hall. If we have Curzon Hall mentioned in the instruction the column 'place0' will contain a value else it will contain a 0. If Curzon hall in mentioned in the instruction, according to the order, the 'place0' column will contain a value. If the robot has to navigate Curzon Hall first among two other points then the row associated with the instruction will contain a value 3000. If Curzon Hall is the second point then the row has a value of 2000. If none of this true then the row will contain a value of 1000.

As we mentioned place0 holds values for Curzon Hall. Similarly place1 holds values for Dr. Muhammad Shahidullah Hall, place2 for Amar Ekushay Hall, place3 for Fazlul Haque Muslim Hall, place4 for TSC, place5 for Shaheed Minar, place6 for Bangla Academy and place7 holds values for Dhaka Medical College Hospital consecutively.

Here is a quick view of our dataset that we've used.



instruction	place_0	place_1	place_2	place_3	place_4	place_5	place_6	place_7
go to Dr. Muhammad Shahidullah Hall	0	1000	0	0	0	0	0	0
start from Dr. Muhammad Shahidullah Hall to Shaheed Minar	0	2000	0	0	0	0	1000	0
start journey from Bangla Academy to Dhaka Medical College Hospital via Dr. Muhammad Shahidullah Hall	0	2000	0	0	3000	0	0	1000
go from Dhaka Medical College Hospital to Amar Ekushey Hall	0	0	1000	0	0	0	0	2000
go from Curzon Hall to Dhaka Medical College Hospital	2000	0	0	0	0	0	0	1000

Figure 3.2: A Quick View of Our Dataset

Data Preprocessing

Before describing the model, we first need to prepare our data so that it can be fit to the model.

We used some cleaning operation they are: lowering the letters, removing unnecessary spacing, removing bad symbols like # or * which has no purpose in navigation instruction. After that we make apply our instruction data to a Tokenizer. This tokenizer allows us to fit our text data to a trainable model.

When we evaluate our model or applying new data to our model, we do this same data preprocessing operation.

3.2.2 Model

The problem we are dealing with is basically a classification problem. As text is our training data so its better to use Sequential model here. Traditional neural network architecture is not suitable for our problem. In this problem we are dealing with text data. For text data, when we are processing a particular word we also have to recall previous words. Recognizing previous data in a traditional neural network is an impossible task.

As we are not dealing with image data so Convolutional Neural Network(CNN) is neither a good choice. For our intended task, neural network architecture of sequential model is the best choice.

After choosing sequential model as our classifier, now the question is what architecture of sequential model will we use? Should we use Recurrent Neural Network(RNN) or Long Short Term Memory(LSTM) architecture? First lets have a look on both of these architecture.

Recurrent Neural Network(RNN)
RNN is helpful to tackle problem when previous information also needed to consider for processing current information. RNN is an architechture of networks with loop [1]

The following figure shows a basic building block of RNN.

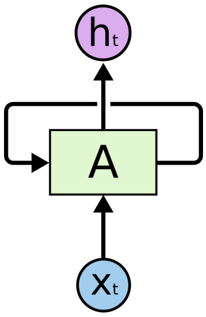


Figure 3.3: Single Building Block of RNN [1]

In the above figure, block **A** represent a group of neural networks. It takes some input x_t and provide value h_t as output.

We can think RNN as same network with multiple copies, each of which passes a message to its successor [1]. If we unroll the RNN it will look like as bellow.

Through previous year in research and industry RNN solved many exciting problems like speech recognition, image captioning, language modeling etc. But RNN has its limitation too. Long-term dependency is the problem that we encounter with RNN.

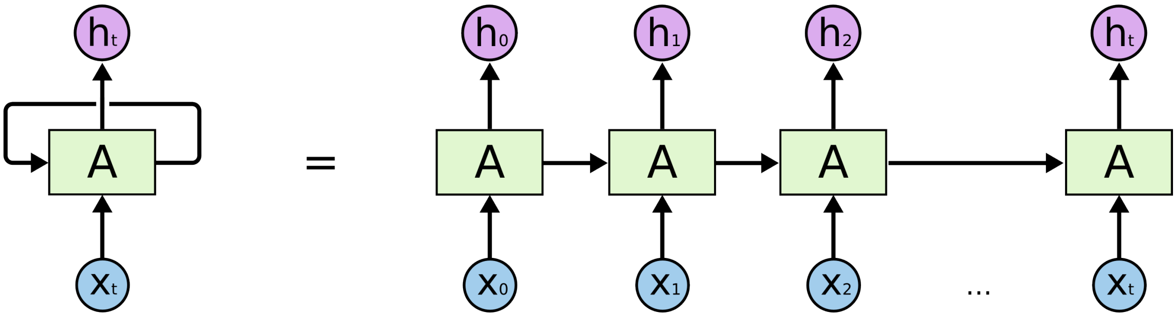


Figure 3.4: Unrolled RNN [1]

Long-term dependency problem arises when a gap created between a relevant information and the place where it needed. Though theoretically RNN supposed not to have long-term dependency problem but practically it actually has. To solve the addressed problem, a special version of RNN, called LSTM is used.

For our problem that we are trying to solve, at first it seems that RNN is sufficient but soon we'll see LSTM provide us better result.

Long Short Term Memory (LSTM)
LSTMs are special structure of RNN which have the capability to learn long-term dependencies. Remembering an information which is may or may not be relevant for long period of is default behavior of LSTM networks.

In all type of RNNs, there is a chain of repeating modules. In a standard RNN, the repeating module have very simple structure like tanh layer.

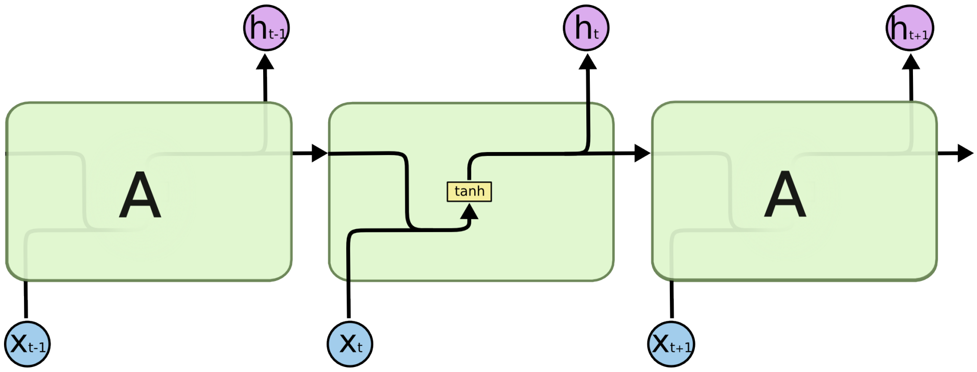


Figure 3.5: Single Layer of RNN in Repeating Module [1]

LSTMs also have RNN like chain structure, but in repeating module of LSTM a different structure is used. Instead of a single NN(neural network) layer, four layers are used and they interact a very special way. The following figure shows repeating module of LSTM.

Cell state is the key contributor of LSTM. Cell state works like kind of conveyor belt.

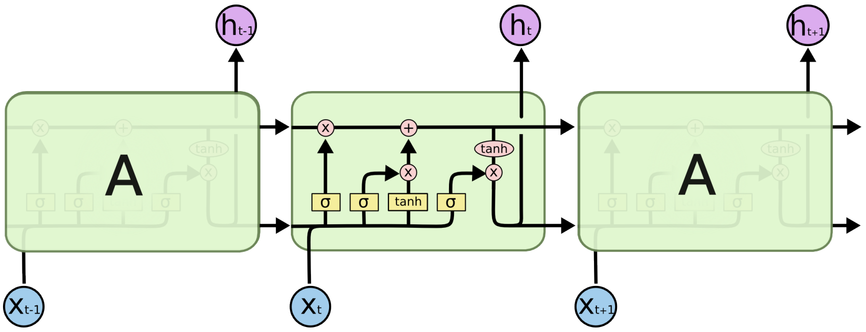
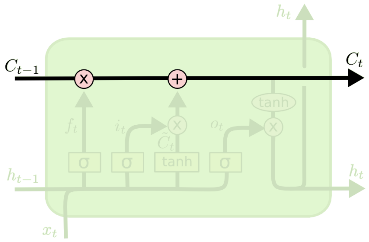


Figure 3.6: Repeating LSTM module with Four Interacting Layers [1]



(a) (b)

Figure 3.7: LSTM's Cell State and Gate

This runs straight through the chain with only having some minor linear interaction. The LSTM posses the ability of adding or removing information to the cell state. This ability is carefully regulated by another structure called Gate.

Gates have the ability to modify the flow of information. Gates are composed of a sigmoid neural network layer and LSTM's pointwise multiplication operation.

The operations of LSTM are done in following three steps:

- Forgettring unnecessary information from cell state
- Adding information to cell state
- Calculating output

For brevity we skip the details of each step. A full and quick overview of a LSTM cell can be easily understood from the figure below.

3.2.3 Designed Architecture

As said earlier, we used LSTM for our classification task. We designed our architecture such a way, first we take s Sequential model. Then we add an Embedding layer with $30X100$ shape. After that We added a spatial dropout of 0.2. Then we add LSTM layer in our Sequential model with 128 hidden units and 0.2 recurrent dropout. At the end of the model, we added a Dense layer with 8 hidden units and we used $softmax$ as our

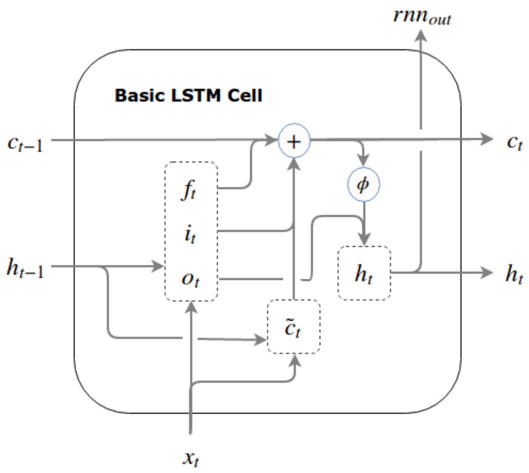


Figure 3.8: Complete LSTM Cell [1]

Layer	Output Shape
Embedding	(30, 100)
Spatial (Dropout)	(30, 100)
LSTM	(128)
Dense	(8)

Table 3.1: Used Model Architecture

activation function in the final Dense layer.

At the end of our model, we calculate loss using ‘categorical cross entropy’. We used *adam* optimizer. Finally we select ‘accuray’ as our metric of the model.

A quick view of our model is as follow.

We designed our model as described previous table. With this architecture we have 618,280 parameters to train.

During training, we used 5 epochs, and 64 size of batches. We used 0.1 validation split. Early stopping was also used with validation loss monitoring, patience value of 3 and minimum delta of 0.0001.

3.3 Map for Providing Output

At the final stage of the task, we shown the output of navigation path on a map. This map is a real world map. Though we used only a little area of Dhaka University campus and eight points but this problem can be scaled on whole Dhaka city or even on whole world!

We use the map service from Mapquest.[Reference 8]

Why Mapquest?
The obvious question is why we used the service of Mapquest instead of others?

First we considered Google Map but Google Map is not entirely free. To get the service of Google Map we need the paid version of Google Cloud Platform so that we couldn't use Google Map.

Then comes Open Street Map. Open Street Map has more location marked but OSM(Open Street Map) doesn't have any built-in routing services. For finding location OSM could be a better choice but for routing purpose it is not a good service at all.

Then we find the option Mapquest. Though Mapquest have less location marked but Mapquest is really helpful for routing from one point to another. Multi-location routing is also possible in Mapquest.

How We Used Mapquest
For getting route between two or several location, we fist collected the latitude and longitude of our desired locations. We collected this location data from Google Map. After finding the sequences of desired destination, which comes as a output from the model, we send a query to Mapquest server and then Mapquest gives us the optimal path between desired points.

3.4 Flowchart

An overview of the whole procedure could be easily understood from a flowchart that is added in the next page.

In this flowchart, it can be seen that first our program is started. When we start our server, the program stared here.

Then the person who using the program speak some navigation instruction. This voice data is sent to speech recognizer module.

After that this module will give us the converted text. If the voice is not properly detected then the user have to speak the instruction again.

After that, the text instruction is applied on Tokenizer and then applied on the Model.

After that the model will give us specific locations that the person instructed to be navigated. And finally the map is shown on the map section.

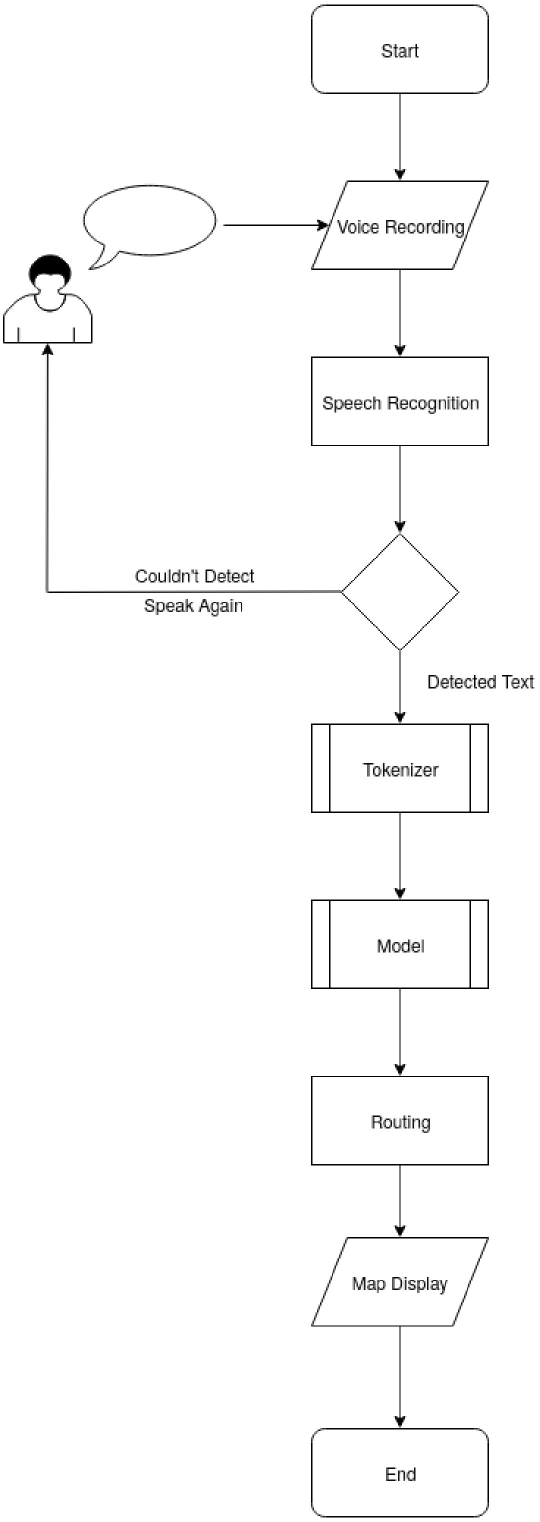


Figure 3.9: Flowchart of the Procedure

Chapter 4

Comparative Result

4.1 Used Model's Performance

As described in *Methodology* section, we've used LSTM model here. A quick summery of LSTM performance is as follow.

Accuracy Type	Accuracy
Train Accuracy	89.72%
Validation Accuracy	97.78%
Test Accuracy	97.90%

Table 4.1: LSTM Accuracy

The following figure shows the graph of model's loss during 5 epochs.

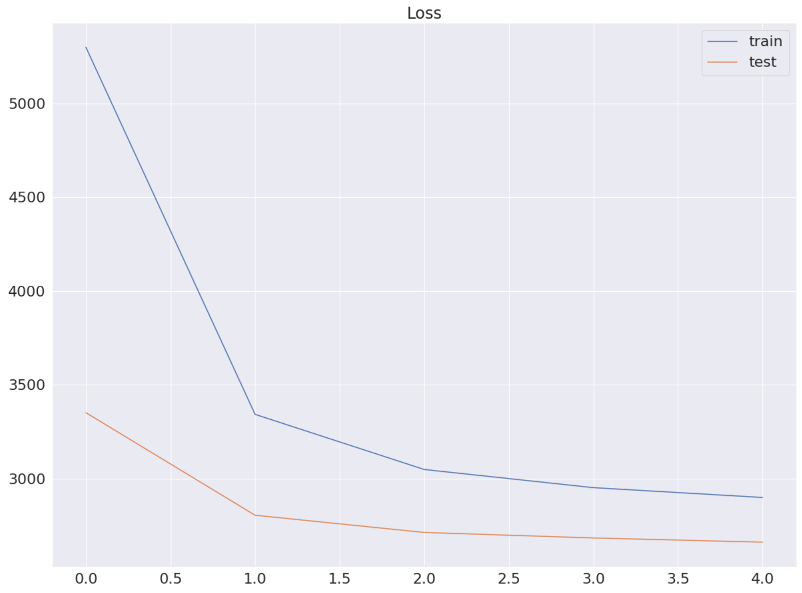


Figure 4.1: LSTM Loss Graph

The following figure shows the graph of the model's test accuracy during 5 epochs.

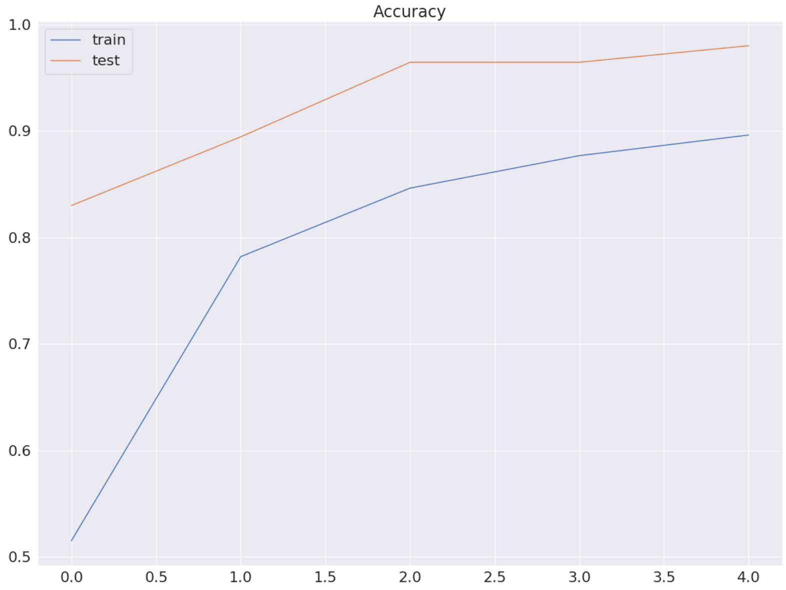


Figure 4.2: LSTM Accuracy Graph

4.2 Another Model's Performance

We try with other two model architecture, RNN and GRU. Here we'll look the performance of these two model.

4.2.1 RNN Performance

A quick summery of RNN model performance is as follow.

Accuracy Type	Accuracy
Train Accuracy	57.53%
Validation Accuracy	63.00%
Test Accuracy	63.00%

Table 4.2: RNN Accuracy

The following figure shows the graph of RNN's loss during 5 epochs.

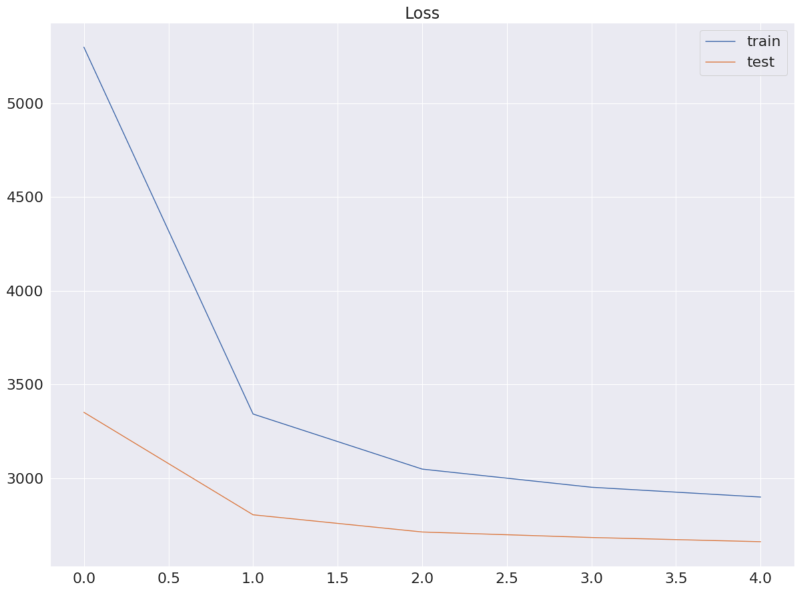


Figure 4.3: RNN Loss Graph

The following figure shows the graph of RNN's test accuracy during 5 epochs.

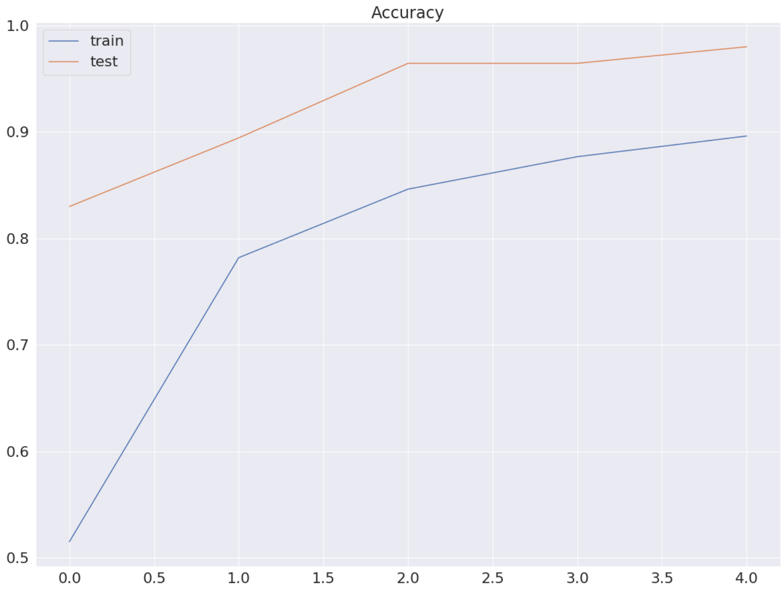


Figure 4.4: RNN Accuracy Graph

4.2.2 GRU Performance

A quick summery of GRU model performance is as follow.

Accuracy Type	Accuracy
Train Accuracy	83.59%
Validation Accuracy	37.30%
Test Accuracy	37.30%

Table 4.3: GRU Accuracy

The following figure shows the graph of GRU's loss during 5 epochs.

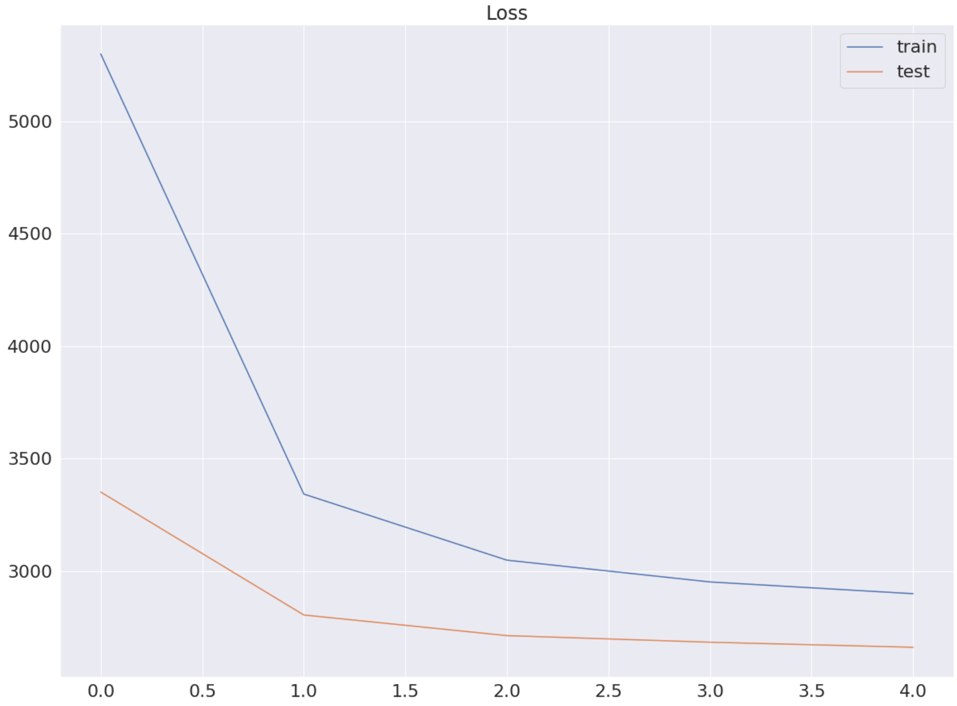


Figure 4.5: GRU Loss Graph

The following figure shows the graph of GRU's test accuracy during 5 epochs.

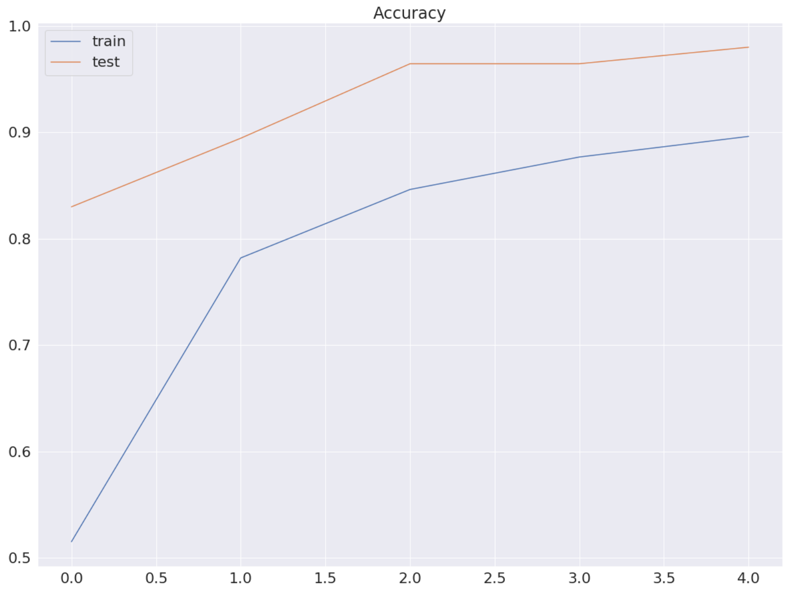


Figure 4.6: GRU Accuracy Graph

4.3 Comparison Altogether

Lets have look on both of these model and their accuracy altogether.

Accuracy Type \Model	LSTM	RNN	GRU
Train Accuracy	89.72%	57.53%	83.59%
Validation Accuracy	97.78%	63.00%	37.30%
Test Accuracy	97.90%	63.00%	37.30%

Table 4.4: Accuracy Comparison Between LSTM, RNN and GRU

The table above shows various accuracy comparison between various models. And we can easily see that LSTM is the obvious champion here.

Chapter 5

Simulation

We provided the initial output in a web interface. The following figure shows the first appearance of our simulation environment when we access to the local server.

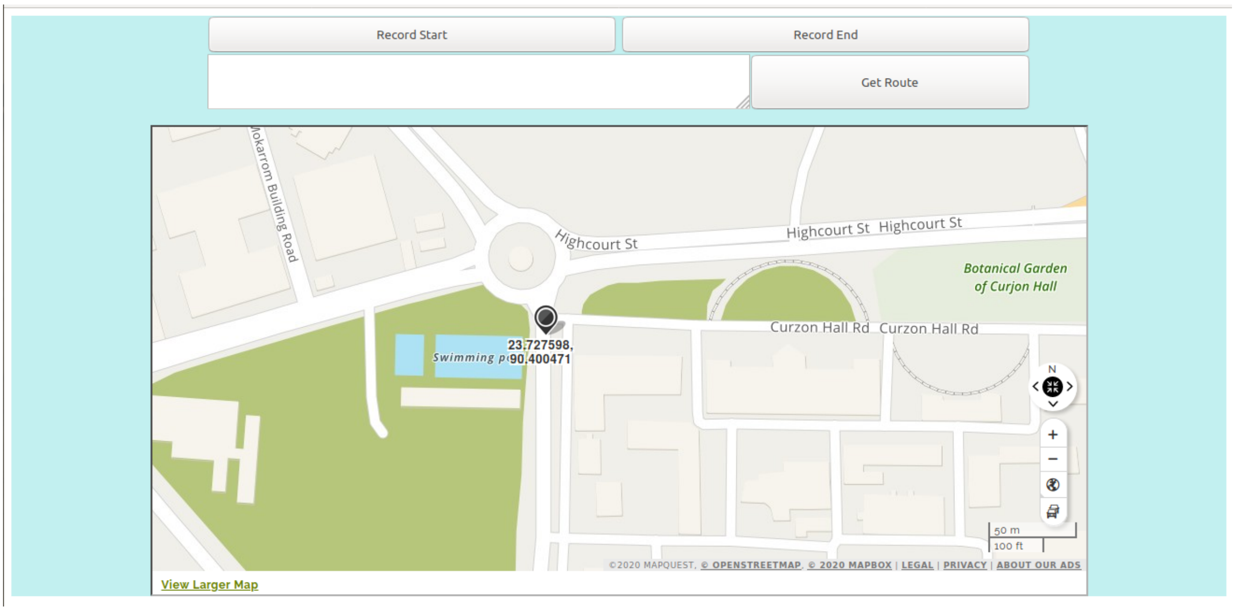
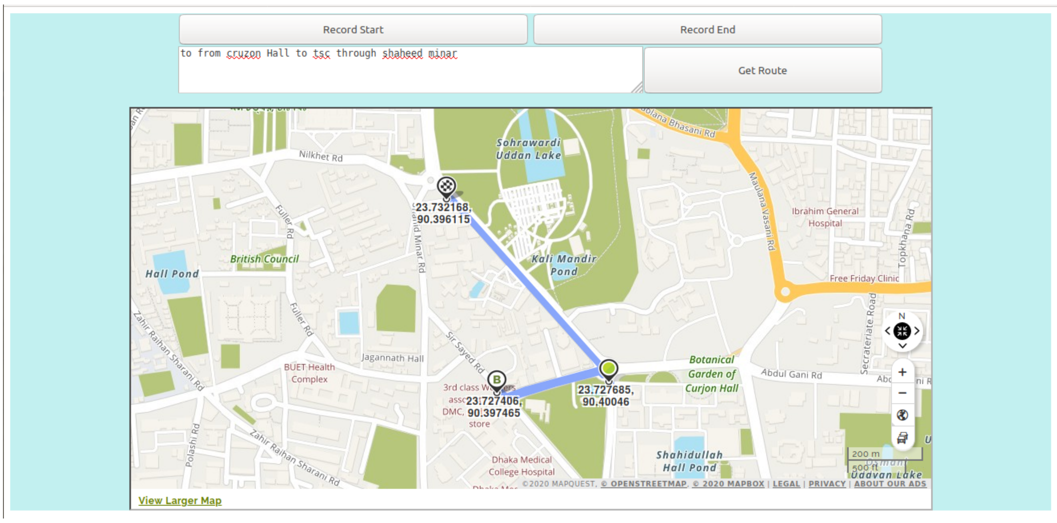


Figure 5.1: First Appearance of the Environment

The instruction we give in voice not all are properly detected by our used speech recognition module. In this simulation we modified the detected text here.

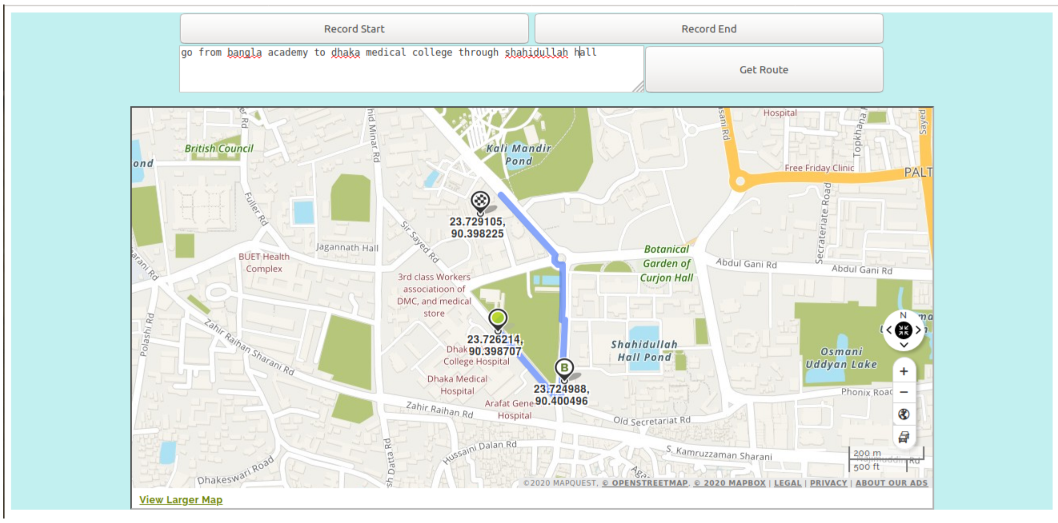
If we give voice instruction: ‘Go from Cruzon Hall to TSC through Shaheed Minar’, the output in the environment will look like as follow.



to from cruzon Hall to tsc through shaheed minar

Figure 5.2: Output for an Instruction

As the models sometimes makes error, here is a erroneous case. For instruction ‘Go from Bangla Academy to Dhaka Medical college through Shahidullah Hall’, the model detected Dhaka Medical College as source instead of destination and similarly detect Bangla Academy as destination instead of source.



go from bangla academy to dhaka medical college through shahidullah hall

Figure 5.3: A Wrong Output for Another Instruction

Hardware

We also have designed a simple line following robot which is able to follow the instruction we gave in voice. We make a dummy track where the robot assumes some node as some real location. The following figure shows our designed line following robot.

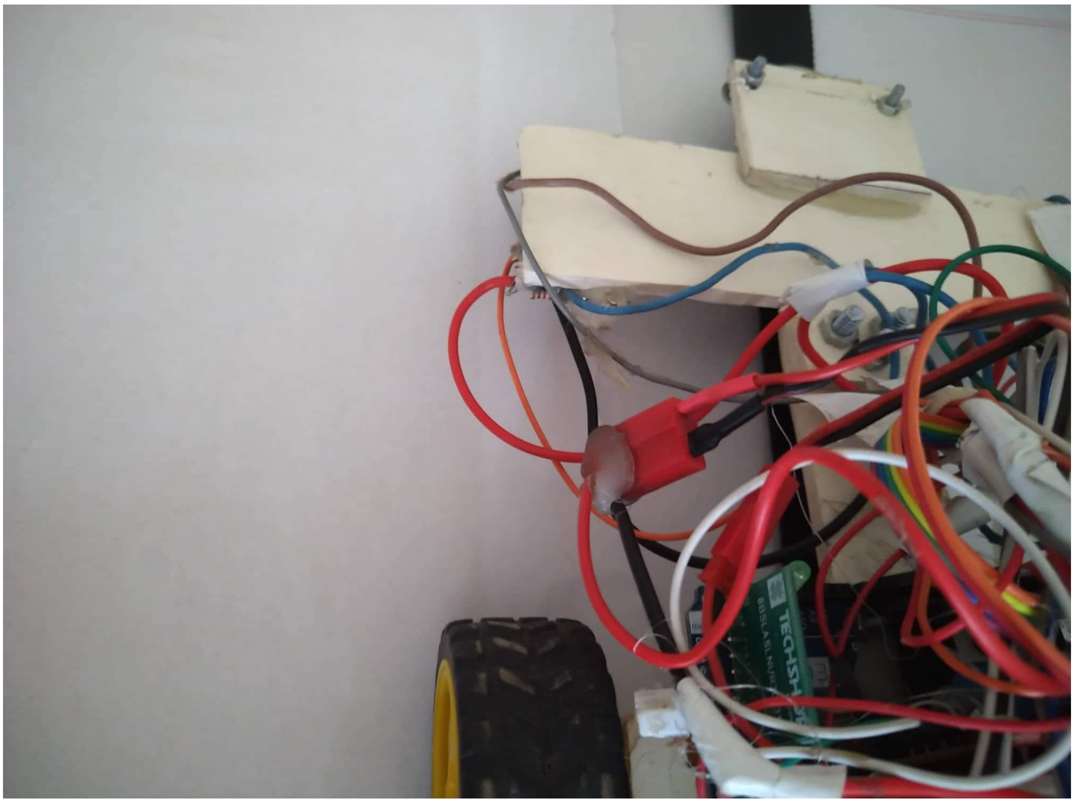


Figure 5.4: A Simple Line Follower for Simulation

Chapter 6

Conclusion

As our collective human knowledge getting richer and the purpose of these knowledge is to make human life easy so it is a key component that a robot or an agent that works on behalf of human should have the ability to understand human level language so that humans don't have to take the burden of making sure the instructions s/he gives that is properly translated by the robot.

In this work, we introduced robot navigation task using natural language. Typically to navigate a robot we have to use the navigation command that the robot can understand like turn left, turn right etc. But in this work we proposed a method where in an known environment we'll just talk to robot for navigating from one point to another point and we don't have to think about will the robot understand our instruction or not.

If one human gives some navigation instruction to another human, the words or language are used between the navigation focused conversation between two humans should also be used for instructing robot. In other words, a robot should supposed to have the ability to understand human level language for navigation related task.

Limitations

Work that we already have done certainly have some limitations. Here we're addressing some of those limitations.

As we used learning method so that it certainly have limitations. We used 10000 data points but still some kind of natural language instruction are missing that we couldn't assign in our dataset. So in future if we give some instruction that is not in our dataset, the model will face some difficulties to extract proper information.

Sequence of locations that we are providing as output is an important issue during navigation. But our model sometimes provide some ambiguous output like which is destination it treat that as starting point and which is starting point it treat that as destination. Here some more works left that we couldn't accomplish here.

We used only eight locations. So if someone talk to navigate some unknown location rather than indicating that its an unknown location, the robot treat the new location as it is within those eight locations. Which is certainly a major limitations of our work.

We only simulate a simple hardware based robot which don't have any access to GPS. But our project shows output in actual world map. So its also a limitations that we didn't prepare proper hardware prototype for the proposed work.

Future Scope
Based on the limitations, we could easily detect what are some future scope left for our project.

Firstly as we didn't develop any speech recognition system here so adding or developing a specialized speech recognition system is one of major section of future work.

If more variation of instruction with complex grammar structure can be added to the dataset then certainly the model will perform better than our current work.

To properly detect the sequence of the model it possibly could use a regression along with classification. Classification is used to detect whether a certain location is mentioned in the instruction or not. A regression can be used to predict that which is source and which is destination. So adding a regression functionality is surely a big issue associate with future scope.

The hardware we demonstrated here didn't had any GPS access. So making a hardware with GPS access that can be navigated according to the actual map would certainly be a great work that still remain incomplete.

We actually proposed a procedure of working with geographical navigation data and natural language instructions. This module can be used in several application like food delivery robot or smart wheelchair etc.

Bibliography

[1] Understanding lstm, Aug 2015.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *CoRR*, abs/1711.07280, 2017.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Sebastopol, CA, United States, 2009.

[4] Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. Talk the walk: Navigating new york city through grounded dialogue. *CoRR*, abs/1807.03367, 2018.

[5] Cynthia Matuszek. Grounded language learning: Where robotics and nlp meet. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5687–5691. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[6] Dipendra Kumar Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. *CoRR*, abs/1704.08795, 2017.

[7] Rustam B. Rustamov, Sabina Hasanova, and Mahfuza H. Zeynalova. *Introduction to Navigation Systems, Multi-purposeful Application of Geospatial Data*. IntechOpen, 2017.

[8] Sonit Singh. Natural language processing for information extraction. *CoRR*, abs/1807.02383, 2018.

[9] S.R.K.Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay. Reinforcement learning for mapping instructions to actions. 2009.

[10] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *ACL*, 2017.

[11] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J. Mankowitz, and Shie Mannor. A deep hierarchical approach to lifelong learning in minecraft. *CoRR*, abs/1604.07255, 2016.